HAL open science

Combination of gene regulatory networks and sequential machine learning for drug repurposing

Clémence Réda

▶ To cite this version:

Clémence Réda. Combination of gene regulatory networks and sequential machine learning for drug repurposing. Artificial Intelligence [cs.AI]. Université Paris Cité, 2022. English. NNT : . tel-03846072v1

HAL Id: tel-03846072

https://hal.science/tel-03846072v1

Submitted on 9 Nov 2022 (v1), last revised 30 Oct 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NoDerivatives 4.0 International License

Université Paris Cité
ED Bio Sorbonne Paris Cité (562)
Laboratoire Inserm (UMR_1141)
Équipe Integrative Genomics in Neurodevelopment

Combination of gene regulatory networks and sequential machine learning for drug repurposing

par Clémence RÉDA

Thèse de doctorat de Génétique
dirigée par Andrée DELAHAYE-DURIEZ

Présentée et soutenue publiquement le 09/09/2022
devant un jury composé de

M. Junya HONDA	Associate Professor	Kyoto University	Rapporteur
Mme. Élisabeth REMY	DR	Institut de Mathématiques de Marseille	Rapportrice
M. Wouter KOOLEN	Research Fellow	Centrum Wiskunde & Informatica	Examinateur
M. Denis THIEFFRY	PU	Institut de Biologie ENS de Paris	Examinateur
M. Bruno VILLOUTREIX	DR	Université Paris Cité	Examinateur
Mme. Andrée DELAHAYE–DURIEZ	PU–PH	Université Sorbonne Paris Nord, AP-HP	Directrice de thèse
Mme. Émilie KAUFMANN	CR-HDR	Inria Lille	Co-encadrante

Titre : Combinaison de réseaux de régulation génique et d'apprentissage statistique séquentiel pour le repositionnement de médicaments

Résumé : À cause du coût toujours croissant de la conception de molécules thérapeutiques *de novo*, et la masse considérable de données biologiques disponibles actuellement, la création de méthodes d'exploration systématique pour le développement de thérapies est devenue un enjeu crucial. Lors de ma thèse, je me suis concentrée sur le paradigme du repositionnement de médicaments, qui cherche à identifier de nouvelles indications thérapeutiques pour des molécules chimiques connues. J'ai cherché des méthodes qui traitent de façon reproductible la quantité importante de données transcriptionnelles disponibles (relative à la production de protéines à travers la transcription des séquences ADN géniques) pour le criblage de molécules. Une revue de l'état de la recherche en développement de médicaments montre que de telles approches génériques peuvent permettre de considérablement accélérer la découverte de thérapies prometteuses, plus particulièrement contre les maladies rares. Premièrement, en remarquant que les mesures d'activité transcriptionnelle résultent d'un réseau dynamique complexe d'interactions régulatrices entre gènes, j'ai travaillé sur l'intégration d'information biologique de différent types afin de construire un modèle de ces régulations géniques. C'est là que les réseaux de régulation génique, et, plus spécifiquement, les réseaux booléens, interviennent. Ces modèles permettent à la fois d'expliquer les mesures d'origine transcriptionnelle observées, et de prédire le résultat de perturbations de l'activité de certains gènes par des molécules. Ensuite, ces modèles permettent d'effectuer des essais cliniques *in silico* de médicaments. Tandis que l'utilisation des prédictions faites par des réseaux booléens peut s'avérer coûteuse, l'hypothèse centrale de ma thèse est que leur combinaison avec des algorithmes d'apprentissage séquentiel, comme les bandits à bras multiples, peuvent non seulement réduire ce coût, mais également contrôler le taux d'erreur dans les recommandations de candidats thérapeutiques. Cette démarche est la procédure d'essai clinique *in silico* analysée tout au long de mon travail de thèse. Le problème de l'intégration des vecteurs caractéristiques connues des composants chimiques dans les bandits à bras multiples a également été étudié. Enfin, j'ai appliqué une partie de mon travail de thèse au classement de différents protocoles de traitement pour de la neuroréparation dans le cas d'encéphalopathies chez des enfants prématurés. J'ai également contribué à la conception d'un algorithme qui permet d'étendre la procédure d'essai médicamenteux *in silico* à un cadre collaboratif, où plusieurs sous-populations de patients sont considérées simultanément.

Mots-clefs : repositionnement de médicaments ; analyse de données transcriptionnelles ; inférence de réseaux de régulation génique ; réseaux booléens ; essai de médicaments ; apprentissage séquentiel ; bandits à bras multiples.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Title: Combination of gene regulatory networks and sequential machine learning for drug repurposing

Abstract: Given the ever increasing cost of designing *de novo* therapeutic molecules, and the huge amount of currently available biological data, the development of systematic explorative pipelines for drug development has become of paramount importance. In my thesis, I focused on drug repurposing, which is a paradigm that aims at identifying new therapeutic indications for known chemical compounds. I investigated how to leverage in a reproducible way for drug screening the huge collection of available transcriptomic data –relative to protein production through the transcription of gene DNA sequences. The current state of research in drug development indicates that such generic approaches might considerably fasten the discovery of promising therapies, especially for rare diseases research. First, noting that transcriptomic measurements are the product of a complex dynamical system of regulatory interactions on genes, I worked on integrating diverse types of biological information in order to build a model of these regulations. That is where gene regulatory networks, and more specifically, Boolean networks, intervene. Such models are useful for both explaining observed transcription levels, and for predicting the result of gene activity perturbations through molecules. Second, these models allow online *in silico* drug testing. While using the predictive features of Boolean networks can be costly, the core assumption of this thesis is that, combining them with sequential learning algorithms, such as multi-armed bandits, might mitigate that effect, and help control the error rate in recommended therapeutic candidates. This is the drug testing procedure suggested throughout my PhD. The question of the proper integration of known side information about the chemical compounds into multi-armed bandits is crucial, and has also been investigated. Finally, I applied part of my work to ranking different treatment protocols for neurorepair in the case of encephalopathy in premature infants. On the theoretical side, I also contributed to the design of an algorithm which is able to extend the drug testing procedure to a collaborative setting, where personalized recommendation of drug candidates can be made to heterogeneous subpopulations of patients, while being learnt using observations made on the whole set of subpopulations.

Keywords: drug repurposing ; transcriptomic data analysis ; gene regulatory network inference ; Boolean networks ; drug testing ; sequential learning ; multi-armed bandits.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Résumé substantiel en français

Le développement de médicaments est un processus connu pour son coût en temps de recherche, pouvant prendre près de dix ans, et financier important, avec un coût chiffré en millions de dollars. De plus, il est sujet à d'assez forts taux d'échec de développement,[1] même tardivement dans le développement. Seulement 64 % des traitements entrés en phase III d'un essai clinique après 2007 ont été commercialisés en 2012.[2]

Cependant, grâce aux améliorations technologiques dans le domaine de la bioinformatique, une large quantité de données biologiques est disponible publiquement ; par exemple, la base de données Gene Expression Omnibus (GEO) pour les données transcriptomiques.[3] Cela permet d'accélérer la phase de recherche de thérapie en explorant ces données. Afin de garantir la reproducibilité et la transparence des résultats obtenus (et donc, leur robustesse), il est nécessaire pour traiter ces données de faire appel à des méthodes *in silico* automatisées.[4] De plus, le développement de molécules thérapeutiques *de novo* est à la fois complexe (puisque requérant la conception de molécules synthétisables ayant la fonction désirée) et relativement risqué, car il n'y a d'informations ni sur leur capacité d'administration, ni sur leurs éventuels effets secondaires néfastes, ce qui explique une partie des échecs dans le développement.[5]

Ce constat m'a incitée à m'intéresser à un paradigme nommé le « repositionnement de médicaments » (*drug repurposing*), qui propose de rechercher de nouvelles thérapies parmi des composants chimiques d'ores et déjà commercialisés, soit en tant que molécules-outils, soit en tant que médicaments

[1] T. Burki (2020). "A new paradigm for drug development". *The Lancet Digital Health*, 2(5), e226–e227.

[2] D. Lowe (2019). *The Latest on Drug Failure and Approval Rates*. https://www.science.org/content/blog-post/latest-drug-failure-and-approval-rates. Accessed: [March 23, 2022].

[3] T. Barrett et al. (2012). "NCBI GEO: archive for functional genomics data sets—update". *Nucleic acids research*, 41(D1), pp. D991–D995.

[4] P. Zucchelli (2018). *Lab Automation Increases Repeatability, Reduces Errors in Drug Development*. https://www.technologynetworks.com/drug-discovery/articles/lab-automation-increases-repeatability-reduces-errors-in-drug-development-310034. Accessed: [March 23, 2022].

[5] V. D. Mouchlis et al. (2021). "Advances in de novo drug design: from conventional to machine learning methods". *International journal of molecular sciences*, 22(4), p. 1676.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

approuvés.[6] Cela permet à la fois de limiter la phase préclinique et la phase I de développement, mais également de réduire la probabilité d'émergence d'effets secondaires non désirables inconnus. Ces observations sont basées sur une revue de l'état-de-l'art effectuée en début de thèse.[7]

En particulier, cette thèse s'intéresse à une approche spécifique, nommé l'inversion de signature (*signature reversion*).[8] Cette méthode a déjà démontré son efficacité, par exemple, pour le repositionnement de médicaments contre la grippe,[9] où les auteurs ont pu vérifier *in vitro* que l'administration de nifurtimox ou de chrysine permettait de réduire la quantité de virions grippaux.

Cette approache exploite des données transcriptomiques d'expression de gène. Le niveau d'expression d'un gène donné, dans un contexte spatial et temporal fixé, correspond à une mesure du nombre de transcriptions de la séquence génique. Cette mesure est corrélée au nombre de copies effectuées du produit encodé par ce gène (par exemple, une protéine). Ce produit est ce par quoi le gène exécute sa fonction dans l'organisme, en participant à des réactions chimiques au sein de l'organisme. Plus le nombre de copies de ce produit est élevé, plus l'effet de la fonction du gène se fait ressentir. La mesure du niveau d'expression d'un gène consiste d'abord à dénombrer les transcrits (nombres d'ARN messager, ou ARNm associés à ce gène) présents dans un échantillon de cellules, ce qui peut être fait par les technologies de séquençage à très haut débit. Une normalisation de cette quantité donne le niveau d'expression, c'est-à-dire le nombre moyen de transcriptions du gène par échantillon.

L'approche d'inversion de signature utilise alors cette donnée dans le but de trouver un candidat thérapeutique contre une pathologie fixée :

(1). D'abord, le niveau d'expression de gène chez des échantillons traités par diverses molécules, avec leur population de référence (« contrôle »), est mesuré. À partir de ces profils transcriptionnels, on en déduit une *signature* « traitement », c'est-à-dire une quantification de la variation d'expression de chaque gène des échantillons traités par rapport aux contrôles due à la différence de traitement. Par exemple, si l'expression d'un gène A est globalement plus (respectivement, moins) importante chez les échantillons traités par rapport à son expression chez les contrôles, alors ce gène sera noté

[6] D. W. Carley (2005). "Drug repurposing: identify, develop and commercialize new uses for existing or abandoned drugs. Part I". *IDrugs: the investigational drugs journal*, 8(4), pp. 306–309.

[7] Réda, Kaufmann, and Delahaye-Duriez (2020). "Machine learning applications in drug development". *Computational and structural biotechnology journal*, 18, pp. 241–252.

[8] J. Lamb et al. (2006). "The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease". *science*, 313(5795), pp. 1929–1935.

[9] Y. Xin et al. (2022). "Identification of Nifurtimox and Chrysin as Anti-Influenza Virus Agents by Clinical Transcriptome Signature Reversion". *International Journal of Molecular Sciences*, 23(4), p. 2372.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

comme étant « up-régulé » (resp., « down-régulé »). L'avantage de considérer des signatures, plutôt que les profils transcriptionnels eux-mêmes, est de permettre de comparer les évolutions au niveau de l'expression de gène corrélées au traitement, indépendamment du contexte spatial et temporel des mesures transcriptomiques ;

(2). De même, le niveau d'expression de gène est mesuré chez des individus sains pour la pathologie considérée et des individus malades. Pareillement, une signature « maladie » est construite, représentant la différence d'expression entre individus sains et malades due à la maladie ;

(3). Pour chaque signature « traitement », une comparaison avec la signature « maladie » est effectuée. Si une signature « traitement » est similaire à la signature « malade », alors cela signifie que l'évolution au niveau de l'expression de gène chez un échantillon traité est corrélée à celle chez un échantillon malade ; autrement dit, le traitement semble « reproduire » un profil malade. Si au contraire, il existe une forte différence entre signatures « traitement » et « maladie », alors, par un même raisonnement, on en déduit que le traitement semble « restaurer » des niveaux d'expression de gène similaires à ceux observés chez les individus sains ;

(4). Traditionnellement, les candidats recommandés à la suite de cette procédure sont ceux dont la dissimilarité entre leur signature associée et la signature « maladie » est la plus grande.

Exemple d'application du principe de l'inversion de signature. J'ai appliqué[10] cette procédure dans un projet de sélection d'un protocole de traitement par cellules souches mésenchymateuses humaines chez le modèle murin d'encéphalopathie de la prématurité.

Dans cette étude, deux voies d'administration, trois âges d'injection des cellules souches et trois grades de doses (dépendant du poids de l'animal) ont été testés sur des rats modèles dont les cellules microgliales ont été séquencées, ainsi que celles d'animaux modèles n'ayant pas reçu de cellules souches, et d'animaux sains injectés avec un traitement factice. En me basant sur les profils transcriptionnels obtenus et l'approche d'inversion de signature, j'ai calculé la signature « traitement » associée à chaque protocole de traitement (en comparant échantillons modèles traités selon le protocole par cellules souches, et échantillons modèles non traités) ainsi que la signature « contrôle », en comparant échantillons sains et échantillons modèles non traités par cellules souches. Enfin, j'ai classé les différents protocoles de traitement par similarité décroissante entre chaque signature « traitement » et la signature « contrôle ».

Le classement obtenu ne semble pas dépendre de facteurs confondants

[10] Bokobza, Réda, et al. (*in prep.*). "Therapeutic evaluation of Hu-MSCs in a rat model of perinatal inflammation: a systematic outcome scoring".

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

liés à la mesure des données et non au traitement (« effet batch »), et ne présente pratiquement que des scores positifs, dont certains sont proches de la valeur 1. Cela suggère que l'injection de cellules souches est globalement bénéfique pour le traitement de l'encéphalopathie de la prématurité chez le modèle murin. L'approche par inversion de signature semble donc être intéressante en pratique. Dans ce cas, elle permet de déterminer le mode d'administration (voie, âge, dose relative au poids) qui maximise l'effet des cellules souches mésenchymateuses humaines sur la restauration d'un phénotype sain.

Limites de l'approche d'inversion de signature. Cependant, l'utilisation de cette approche pour le repositionnement de médicaments est confrontée à plusieurs défis.

Premièrement, **(A)** il n'est pas clair que les signatures obtenues respectivement aux étapes **(1)** et **(2)** dans le paragraphe précédent soient le reflet des conséquences *directes* du traitement (ou de la pathologie) au niveau transcriptionnel. En particulier, l'existence bien connue de cascades transcriptionnelles de régulation,[11] avec un effet « domino » de la perturbation sur l'expression des gènes, semble plutôt suggérer que ces signatures sont le fruit de profils transcriptionnels « stabilisés ». Cela pose donc le problème de distinguer la cause des symptômes de la pathologie étudiée, de même que les processus biologiques réellement ciblés par un traitement ;

Deuxièmement, **(B)** d'un point de vue pratique, cette méthode requiert d'assez grandes quantités de données transcriptomiques. Il est alors capital d'exploiter les données de traitement et de profils patients de façon générique pour augmenter la reproductibilité (*c.à.d.*, compatible avec n'importe quelle pathologie considérée) ;

Troisièmement, **(C)** la procédure de recommandation de candidats, qui consiste à considérer les candidats les mieux classés (par exemple, les N premiers), ne permet pas ni de contrôler le taux d'erreur, ni de fournir une interprétation des résultats au niveau transcriptionnel ;

Enfin, **(D)** la démarche d'inversion de signature ne prend en compte qu'une tendance globale au niveau de l'évolution de l'expression à travers la population de patients : définir un cadre personnalisé pour la recommandation de candidats thérapeutiques pourrait être intéressant pour l'étude de maladies avec de multiples sous-catégories, comme par exemple l'épilepsie, ou le cancer du sein.

Introduction des réseaux de régulation pour le repositionnement de médicaments. Tout d'abord, dans le but de résoudre les problèmes

[11] H. Bolouri and E. H. Davidson (2003). "Transcriptional regulatory cascades in development: initial rates, not steady state, determine network kinetics". *Proceedings of the National Academy of Sciences*, 100(16), pp. 9371–9376.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

évoquées en **(A)** et en **(B)**, j'ai considéré[12] l'intégration de réseaux de régulation génique dans l'approche d'inversion de signature.

En effet, un réseau de régulation génique (*gene regulatory network*) modélise l'effet d'interactions régulatrices de l'expression sur un ensemble de gènes. Ce réseau est un graphe dont les nœuds sont les gènes, et ses arêtes sont les interactions régulatrices entre les gènes.

Cependant, ce modèle reste statique, et dans le but de démêler les causes des conséquences **(A)**, nous avons besoin d'un modèle prédictif de l'expression de gène, en particulier, en réponse à des perturbations externes. Plusieurs types de réseaux satisfont ce critère, et dans cette thèse, je considère le formalisme des réseaux booléens.[13] D'abord, parce que ce type de modèle permet d'obtenir des prédictions d'ordre qualitatif, ce qui en facilite l'interprétation ; ensuite, parce que le calcul des « états stables » dans le réseau a été bien étudié.

Un réseau booléen est défini par, premièrement, un graphe d'interactions $\mathcal{B}(\mathcal{G}, \mathcal{E})$ qui connecte les nœuds dans l'ensemble $\mathcal{G}$ agissant les uns sur les autres à travers l'ensemble d'arêtes $\mathcal{E}$; et, deuxièmement, un système logique $\mathcal{S}(V, F, [\![\cdot]\!])$ qui décrit la dynamique du modèle. Ce système comporte un ensemble de variables booléennes (c'est-à-dire, pouvant prendre la valeur 0 ou 1 uniquement) V, et de formules logiques F, en nombre égal à la taille de $\mathcal{G}$. On confond en général l'ensemble des nœuds $\mathcal{G}$ et des variables associées V. Pour tout nœud g dans $\mathcal{G}$ ($g \in \mathcal{G}$), la formule logique $\phi_g \in F$ comporte des connecteurs logiques (« et » $\wedge$, « ou » $\vee$, « non » $\neg \ldots$) qui combinent les variables associées à des régulateurs *directs* du nœud g dans le graphe $\mathcal{B}$. Si un régulateur $r \in \mathcal{G}$ est un activateur (resp., un inhibiteur) de l'expression du nœud g, alors il existe une arête de r vers g activatrice (resp., inhibitrice) dans l'ensemble des arêtes du graphe d'interactions $\mathcal{E}$.

À un moment donné, l'état du réseau est décrit par l'ensemble des valeurs booléennes attribuéees à toutes les variables associées aux nœuds. Pour obtenir l'état du réseau après application de l'effet des interactions régulatrices sur les valeurs des variables, on utilise la fonction $[\![\cdot]\!]$ (« sémantique »). Elle définit la façon d'évaluer la proposition logique $\phi_g \implies g$ pour chaque gène g, en fonction des valeurs actuelles des variables du réseau. Si pour une variable g donnée, cette proposition est vraie ($[\![\phi_g \implies g]\!] = 1$) alors l'état du nœud associé g est égal à 1 (interprété biologiquement comme « le gène g est exprimé »). Sinon, cet état est égal à 0 (« non exprimé »). On

[12] Réda and Delahaye-Duriez (2022). "Prioritization of Candidate Genes Through Boolean Networks". *International Conference on Computational Methods in Systems Biology*. Springer, pp. 89–121.

[13] S. A. Kauffman (1969). "Metabolic stability and epigenesis in randomly constructed genetic nets". *Journal of theoretical biology*, 22(3), pp. 437–467; R. Thomas (1973). "Boolean formalization of genetic control circuits". *Journal of theoretical biology*, 42(3), pp. 563–585.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

ne détaillera pas dans ce résumé les sémantiques possibles. La sémantique définit alors une relation entre deux états s_1, s_2 du réseau : $s_1 \rightarrow_{[\![\cdot]\!]} s_2$ si et seulement si évaluer ces propositions logiques avec $[\![\cdot]\!]$ aboutit à l'état s_2. On peut alors construire un diagramme de transition entre états, c'est-à-dire, un graphe dont les nœuds sont les états possibles du réseau et les arêtes les relations $\rightarrow_{[\![\cdot]\!]}$. En particulier, on s'intéresse aux états qui « bouclent sur eux-mêmes » dans ce diagramme : ce sont les états attracteurs stables. Autrement dit, une fois que le réseau se trouve dans cet état, l'application de la sémantique laisse cet état invariant. Ces états sont intéressants parce que, biologiquement, on les considère comme les phénotypes « stabilisés » des types de cellules modélisés.[14]

On peut alors utiliser ce réseau pour prédire des profils transcriptionnels « stabilisés », en particulier, après perturbation de certains gènes. Une perturbation du gène g est définie par le fait de rajouter dans la sémantique la règle $[\![g]\!] = 0$ (knock-out) ou $[\![g]\!] = 1$ (surexpression, ou knock-in) qui supplante l'évaluation de $[\![\phi_g]\!]$. En partant d'un état initial du réseau et en appliquant itérativement la sémantique choisie, on peut prédire l'état attracteur stable dans lequel on aboutit au bout d'un certain nombre d'itérations de la sémantique, s'il existe. Le nombre d'itérations est en général choisi suffisamment grand pour pouvoir détecter cet état attracteur s'il existe.

La construction d'un réseau booléen, en particulier sans la présence d'un modèle de régulation déjà existant ni de données pertinentes d'expression de gène, est souvent un processus manuel de collecte d'interactions entre paires de gènes (dans la littérature ou dans des bases de données), et de amélioration itérative du modèle en comparant les prédictions d'états obtenues avec des données transcriptomiques issues d'expériences de perturbation de gène. Cette procédure atteint rapidement ses limites à partir du moment où l'on considère quelques centaines de gènes dans le réseau, ou dans le cas où l'on étudie les régulations dans le contexte d'une maladie rare.

Construction automatisée de réseau booléen associé à une maladie donnée. Une première étape a porté sur le développement d'une méthode de construction automatique de réseau booléen associé à une maladie donnée, dans le but de résoudre le problème évoqué dans **(B)**.

Cette méthode part d'un ensemble de gènes associés à une maladie, de données d'expression dans des expériences de perturbation de ces gènes (un gène à la fois est perturbé) issues de la base de données LINCS L1000,[15] et des interactions répertoriées entre protéines codées par ces gènes, *via*

[14] P. Bloomingdale et al. (2018). "Boolean network modeling in systems pharmacology". *Journal of pharmacokinetics and pharmacodynamics*, 45(1), pp. 159–180.

[15] Subramanian et al. (2017). "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles". *Cell*, 171(6), pp. 1437–1452.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

la base de données STRING.[16] En utilisant les données d'expression, j'ai pu construire un ensemble d'interactions possibles entre gènes. Puis j'ai utilisé la méthode d'inférence de réseau booléen BoNeSiS[17], qui calcule les formules logiques et le sous-graphe d'interactions qui permettent de satisfaire toutes les observations expérimentales extraites de LINCS L1000. Enfin, j'ai sélectionné un réseau parmi les solutions retournées par la méthode d'inférence sur des critères topologiques, en maximisant une fonction de désirabilité.[18] Les seules entrées requises sont une liste de gènes à utiliser dans le réseau et les lignées cellulaires d'intérêt.

J'ai appliqué cette méthode à une liste de gènes présélectionnés[19] pour construire un réseau booléen sur des lignées cérébrales, afin d'étudier les mécanismes liés à l'épilepsie, et mieux comprendre l'épilepsie réfractaire, qui touche 25 % des patients épileptiques, qui ne réagissent pas aux traitements actuels.

Priorisation de régulateurs-clés par le réseau booléen. Pour évaluer l'intérêt de ce réseau, j'ai cherché à déterminer les nœuds régulateurs-clés dans le réseau, *c.à.d.*, les gènes qui sont en amount de la régulation d'un grand nombre de gènes, donc dont la perturbation a des répercussions sur l'ensemble du réseau. Ces gènes pourraient donc constituer des cibles thérapeutiques intéressantes, en particulier, si leur inhibition ou leur surexpression permet d'inverser le profil malade.

Actuellement, une des méthodes les plus courantes pour identifier ces régulateurs-clés est de calculer, pour chaque gène, une mesure relative à la centralité de la position de ce gène dans le réseau. Cependant, d'une part, cette méthode n'exploite que la topologie du réseau lorsqu'elle est appliquée à un réseau booléen. D'autre part, cette définition ne colle pas exactement à la définition d'un régulateur-clé.

J'ai donc conçu une méthode qui retourne, pour chaque gène, un score (appelé « spread ») proportionnel à la perturbation de l'état du réseau par la perturbation de ce seul gène. L'idée derrière ce score est de calculer la dissimilarité entre les états attracteurs stables atteints sans perturbation du gène, et ceux dans lesquels le système peut aboutir lors de la perturbation du gène dans la lignée cellulaire d'intérêt.

[16] D. Szklarczyk et al. (2021). "The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets". *Nucleic acids research*, 49(D1), pp. D605–D612.

[17] S. Chevalier et al. (2019). "Synthesis of boolean networks from biological dynamical constraints using answer-set programming". *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, pp. 34–41.

[18] S. Babichev et al. (2019). "Technique of Gene Regulatory Networks Reconstruction Based on ARACNE Inference Algorithm." *IDDM*, pp. 195–207; E. C. Harrington (1965). "The desirability function". *Industrial quality control*, 21(10), pp. 494–498.

[19] Delahaye-Duriez, Srivastava, et al. (2016). "Rare and common epilepsies converge on a shared gene regulatory network providing opportunities for novel antiepileptic drug discovery". *Genome biology*, 17(1), pp. 1–18.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Le classement par « spread » décroissant est corrélé à la centralité du gène dans le réseau (Control Centrality,[20] degré total et sortant du nœud). Cependant, ce classement est davantage corrélé à des mesures de pathogénicité de gène, par exemple, à l'intolérance d'un gène à une mutation perte-de-fonction (pLI). Dans l'application à l'épilepsie, des gènes associés aux encéphalopathies épileptiques et à de graves troubles du développement sont retrouvés de manière significative (« enrichis ») parmi les 12 premiers gènes du classement par « spread ».

Repositionnement de médicaments par le réseau booléen. Ensuite, j'ai cherché à résoudre les points évoqués dans la partie **(A)**, en définissant une procédure de repositionnement de médicaments, basée sur l'approche d'inversion de signature, qui exploite les propriétés prédictives du réseau booléen.

Cette procédure a été appliquée au réseau construit pour l'épilepsie, et les résultats obtenus ont été comparés à une méthode traditionnelle d'inversion de signature, appelée L1000 CDS^2. Pour ce faire, j'ai récupéré des profils transcriptionnels d'hippocampes épileptiques et sains[21] ainsi qu'une signature construite sur des échantillons traités et contrôles pour chaque molécule à cribler, à partir de la base de données LINCS L1000.[22] Puis, pour chaque échantillon patient perturbé et chaque molécule, le réseau prédit un ensemble d'états attracteurs stables atteints à partir du profil patient perturbé selon la signature de la molécule. Les gènes notés comme up- (resp., down-) régulés dans la signature sont surexprimés (resp., knocked-out) tout au long de la simulation. Pour chaque état attracteur prédit, on calcule un score associé à la distance entre cet état et la frontière de classification qui sépare échantillons contrôles des échantillons patients dans un espace en 2D bien choisi. Ce score est positif si l'état attracteur est dans l'hyperplan des échantillons contrôles, négatif sinon. Un score final pour ce profil patient et cette molécule est obtenu en pondérant les scores par attracteur par la probabilité d'aboutir dans cet attracteur. Enfin, un seul score par traitement est obtenu en moyennant les scores sur l'ensemble des patients.

Cette approche a été testée sur un ensemble de 22 proconvulsivants et 12 antiépileptiques connus. On observe que cette méthode a des performances similaires à L1000 CDS^2[23], avec une faible amélioration au niveau de la mesure d'aire sous la courbe (AUC) qui quantifie la relation entre le taux

[20] Liu, J.-J. Slotine, and A.-L. Barabási (2012). "Control centrality and hierarchical structure in complex networks".

[21] N. Mirza, R. Appleton, et al. (2017). "Genetic regulation of gene expression in the epileptic human hippocampus". *Human molecular genetics*, 26(9), pp. 1759–1769.

[22] Subramanian et al. (2017). "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles". *Cell*, 171(6), pp. 1437–1452.

[23] Q. Duan et al. (2016). "L1000CDS2: LINCS L1000 characteristic direction signatures search engine". *NPJ systems biology and applications*, 2(1), pp. 1–12.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

d'antiépileptiques détectés par la méthode, et le taux de proconvulsivants prédits comme antiépileptiques dans le classement obtenu (L1000 CDS^2 : AUC≈ 0.55 contre AUC≈ 0.63 avec notre méthode, arrondi à la seconde décimale). Plus ce score est grand, meilleure est la méthode. De même, le score F_1 quantifie la relation entre sensitivité (le ratio du nombre d'antiépileptiques détectés sur le nombre total d'antiépileptiques présents) et la spécificité (le ratio du nombre de proconvulsivants détectés sur le nombre total de proconvulsivants présents). Le score F_1 est de 0.44 pour L1000 CDS^2, contre $F_1 \approx 0.58$ (arrondi à la seconde décimale) pour notre méthode.

Variabilité entre individus dans la réponse au traitement. Cependant, il y a une variation assez importante au niveau des scores obtenus par un traitement à travers les profils patients (écart-type de l'AUC à travers les patients : 0.01, arrondi à la seconde décimale). Cela fait écho au problème évoqué en **(C)** sur le calcul du taux d'erreur dans les recommandations, et l'intérêt de recommandations personnalisées en **(D)**. De plus, pour calculer ce score moyen, il faut itérer la procédure de prédiction d'états attracteurs (qui est assez coûteuse en temps et en puissance de calcul) sur l'ensemble des profils patients disponibles, et ce, pour chaque traitement à cribler.

J'ai donc réfléchi à une procédure adaptative, qui calculerait les scores de façon parcimonieuse, dans le but de déterminer les meilleurs candidats à repositionner. Cette procédure pourrait éventuellement reposer sur les signatures des traitements, qui sont informatives sur leur effet transcriptionnel. Cela m'a donc amené à m'intéresser aux algorithmes de bandit.

Introduction des algorithmes de bandit pour le repositionnement de médicaments. Les algorithmes de bandit appartiennent de la catégorie d'algorithmes d'apprentissage statistique par renforcement. L'algorithme (l'« agent ») doit interagir avec un environnement dont il peut tirer des observations, à partir desquelles il prend des décisions ; il adapte alors sa prise de décision en fonction des retours reçus de l'environnement. Dans cette thèse, l'environnement est le réseau booléen qui retourne le score (« observation ») du traitement sur un ensemble de gènes, et la décision à prendre est le choix du prochain traitement (« bras ») à à évaluer. Le but est d'ultimement identifier les meilleurs candidats thérapeutiques à partir de leur effet moyen sur le réseau. Les bandits sont surtout populaires dans le domaine des systèmes de recommendation,[24] même s'ils ont été introduits pour décrire les essais cliniques adaptatifs,[25] c'est-à-dire avec un processus d'allocation des patients aux bras de traitement qui n'est pas uniforme.

[24] D. Bouneffouf, A. Bouzeghoub, and A. L. Gançarski (2012). "A contextual-bandit algorithm for mobile context-aware recommender system". *International conference on neural information processing*. Springer, pp. 324–331.

[25] W. R. Thompson (1933). "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". *Biometrika*, 25(3-4), pp. 285–294.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Maintenant, introduisons le concept de bandit (ici, on ne considèrera que les « bandits stochastiques structurés à nombre de bras finis »). On considère un nombre K de bras, associés à des vecteurs caractéristiques $(X_k)_{k\in[K]}$. Un agent sélectionne séquentiellement ces bras pendant un nombre de tours donnés ou lorsqu'un critère d'arrêt est satisfait. En tirant le bras $I_t \in [K] := \{1, 2, \ldots, K-1, K\}$ au tour t, l'agent exécute une action sur l'environnement, et perçoit son impact sur l'environnement via une observation bruitée Y_t. On fait l'hypothèse que cette observation est une réalisation d'une distribution de probabilité ν_{I_t} ($Y_t \sim \nu_{I_t}$), c'est-à-dire, un nombre tiré selon cette distribution. Celle-ci est considérée fixée et inconnue de l'agent ν_{I_t} (hypothèse de stochasticité), et dépendante du vecteur caractéristique associé X_{I_t} (hypothèse de structure). On note l'espérance de la distribution ν_a de probabilité $\mu_a := \mathbb{E}_{\{Y\sim\nu_a\}}[Y]$ pour tout bras $a \in [K]$. En particulier, ici, on suppose que le modèle de bruit pour tout bras $a \in [K]$ est additif et suit une distribution sous-gaussienne de moyenne μ_a de variance fixée σ^2

$$Y_t = \mu_{I_t} + \psi_t, \text{ où } \psi_t \text{ satisfait pour tout réel } \lambda \in \mathbb{R}^+, \mathbb{E}[\exp(\lambda\psi_t)] \leq \exp\left(\frac{\lambda^2\sigma^2}{2}\right) .$$

Pour la suite du résumé, on se restreint aux distributions gaussiennes de variance fixée σ^2. Sans perte de généralité, on considère qu'une action positive a sur l'environnement émet une observation moyenne μ_a plus grande que celle émise par une action négative sur l'environnement. Cependant, l'agent ne peut estimer l'observation moyenne associée à un bras qu'en tirant plusieurs fois certains bras. Un bras optimal $a^\star$ est alors un bras dont l'observation moyenne est la plus grande de tous les bras :

$$\mu_\star = \mu_{a^\star} := \max_{a\in[K]} \mu_a$$

(il peut y en avoir plusieurs) ; similairement, on définit le $N^{\text{ème}}$ meilleur bras

$$\mu_{(N)} = \mu_{a_{(N)}} := \max_{a\in[K]}^{N} \mu_a .$$

L'agent peut avoir différents objectifs. Par exemple, l'objectif le plus couramment défini est d'obtenir la plus grande somme cumulative d'observations en T tours, ce qui s'appelle la minimisation du regret cumulatif (*cumulative regret minimization*).[26] Dans ce cas, l'agent doit souvent jouer le bras que l'agent estime optimal à partir des observations précédentes, au risque d'être pénalisé si on tire un bras sous-optimal. Dans notre cadre de reposi-

[26] P. Auer, N. Cesa-Bianchi, Y. Freund, et al. (2002). "The nonstochastic multiarmed bandit problem". *SIAM journal on computing*, 32(1), pp. 48–77; Li, W. Chu, et al. (2010). "A contextual-bandit approach to personalized news article recommendation". *Proceedings of the 19th international conference on World wide web*, pp. 661–670.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

tionnement de médicaments (et en particulier, sachant que l'on utilise des simulations *in silico*), on n'a pas d'intérêt à pénaliser l'exploration, c'est-à-dire l'observation de bras sous-optimaux pour acquérir de l'information sur l'environnement. L'objectif de l'agent en utilisant l'algorithme de bandit $\mathfrak{A}$ est donc l'identification des N meilleurs bras (*Top-N identification*) avec un plus petit nombre d'observations $\tau_{\mathfrak{A}}$ possible. Ce nombre est défini par le critère d'arrêt de l'algorithme. Pour répondre au problème en **(C)**, l'algorithme $\mathfrak{A}$ doit retourner un ensemble de bras $\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}})$ de taille N, avec une probabilité d'erreur de recommandation inférieure à un seuil fixé δ

$$\mathbb{P}\left[\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \not\subseteq \mathcal{S}_N^\star\right] \leq \delta, \text{ où } \mathcal{S}_N^\star := \left\{a \in [K] : \mu_a \geq \mu_{(N)}\right\} \ , \tag{1}$$

est l'ensemble des bras optimaux. On peut aussi définir l'ensemble des bras ε-optimaux (*c.à.d.*, optimaux à ε près) $\mathcal{S}_N^\star(\varepsilon) := \left\{a \in [K] : \mu_a \geq \mu_{(N)} - \varepsilon\right\}$.

Combinaison des réseaux géniques et des algorithmes de bandit pour le repositionnement de médicaments. La méthode de repositionnement de médicaments que je propose combine les scores obtenus par le réseau booléen et les algorithms de bandit. En réponse à la requête de l'agent, un score lié au traitement t est calculé par le réseau booléen, sur un profil patient tiré aléatoirement. Ce score est retourné à l'agent. Celui-ci doit alors choisir le prochain traitement à tester, de sorte à limiter le nombre de scores nécessaires avant de trouver les N meilleurs candidats. Contrairement aux travaux publiés avant le début de ma thèse,[27] l'algorithme de bandit peut exploiter les signatures associées aux traitements. Premièrement, cette information supplémentaire permet de réduire le nombre d'évaluations nécessaires, en supposant que des traitements ayant des signatures similaires auront des scores de repositionnement similaires. Deuxièmement, cela permet d'apprendre un modèle mathématique des scores en fonction des signatures, et de pouvoir alors interpréter l'importance d'un gène sur la valeur du score.

Résolution du problème d'identification des N meilleurs bras avec vecteurs caractéristiques. D'abord, je me suis concentrée[28] sur l'identification des N bras $\varepsilon-$optimaux sous taux d'erreur fixé δ se reposant sur les vecteurs caractéristiques.

N'importe quel tel algorithme $\mathfrak{A}$ est défini par trois règles :

(a) la règle d'échantillonnage (*sampling rule*), qui choisit le bras à tirer en fonction des observations précédentes ;

[27] V. Gabillon, M. Ghavamzadeh, and A. Lazaric (2012). "Best arm identification: A unified approach to fixed budget and fixed confidence". *Advances in Neural Information Processing Systems*, 25; S. Kalyanakrishnan et al. (2012). "PAC subset selection in stochastic multi-armed bandits." *ICML*. vol. 12, pp. 655–662.

[28] Réda, Kaufmann, and Delahaye-Duriez (2021). "Top-m identification for linear bandits". *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1108–1116.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

(b) le critère d'arrêt (*stopping rule*), qui choisit d'arrêter la phase d'apprentissage (et donc, choisit le temps $\tau_{\mathfrak{A}}$) pour entrer dans la phase de décision ;

(c) la règle de décision (*decision rule*), qui construit l'ensemble de candidats $\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}})$.

L'analyse de cet algorithme se décompose en deux parties. D'une part, on prouve sa correction, c'est-à-dire si la Condition (1) est bien satisfaite. En pratique, on construit un « bon » évènement $\mathcal{E}$ de probabilité supérieure à $1-\delta$ sur lequel l'évènement $\{\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \subseteq \mathcal{S}_N^\star\}$ est toujours vrai (*c.à.d.*, l'algorithme ne se trompe pas). D'autre part, on calcule une borne supérieure sur le nombre $\tau_{\mathfrak{A}}$ d'observations suffisant pour déclencher le critère d'arrêt (*sample complexity*).

J'ai donc défini une famille d'algorithmes de bandit appelée GIFA (*Gap-Index Focused Algorithms*), qui suppose une relation linéaire entre les scores et les vecteurs caractéristiques des bras, avec un paramètre inconnu à estimer θ

$$\exists \theta \in \mathbb{R}^d \forall a \in [K], \mu_a := \theta^\top X_a \ . \tag{2}$$

En particulier, si le vecteur caractéristique X_a est la signature du traitement a, alors le paramètre θ estimé à la fin de la phase d'apprentissage donne les cœfficients de la combinaison linéaire $\theta^\top X_a$. J'ai proposé une analyse unifiée de *sample complexity* pour ces algorithmes, dont la correction peut être montrée même sur des algorithmes partiellement définis (*c.à.d.*, l'une des trois règles ci-dessus n'est pas définie). Cette famille d'algorithmes se fonde sur les valeurs (« indices ») $(\mathcal{B}_{a,b}(t))_{(a,b)\in[K]^2, t\geq 0}$, tels que $\mathcal{B}_{a,b}(t)$ surapproxime l'écart (*gap*) $\mu_a - \mu_b$ entre les bras a et b à n'importe quel tour t, avec probabilité $1-\delta$. J'ai montré que cette condition constitue un « bon » évènement pour tout algorithme de la famille GIFA, ce qui en garantit sa correction. Au tour t, un algorithme de la famille GIFA définit à l'aide de ces indices un ensemble de candidats $J(t)$; un représentant de l'ensemble des candidats $J(t)$ noté $b(t)$; et un compétiteur $c(t)$ qui n'est pas dans $J(t)$. Ce dernier est défini comme le bras qui n'est pas dans $J(t)$ qui maximise l'indice sur l'écart avec $b(t)$: $c(t) \in \arg\max_{a \notin J(t)} \mathcal{B}_{a,b(t)}(t)$. Un algorithme GIFA $\mathfrak{A}$ retourne à la fin de sa phase d'apprentissage l'ensemble $J(\tau_{\mathfrak{A}})$.

Deux types d'algorithmes GIFA, nommés LUCB-GIFA et Gap-GIFA selon les définitions de $J(t)$ and $b(t)$, sont particulièrement intéressants car ils généralisent des travaux antérieurs[29]. Le premier traque la valeur de l'écart minimal entre un bras dans $J(t)$ et un autre qui n'y est pas, *via* l'indice

[29] V. Gabillon, M. Ghavamzadeh, and A. Lazaric (2012). "Best arm identification: A unified approach to fixed budget and fixed confidence". *Advances in Neural Information Processing Systems*, 25; S. Kalyanakrishnan et al. (2012). "PAC subset selection in stochastic multi-armed bandits." *ICML*. vol. 12, pp. 655–662.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

$\mathcal{B}_{c(t),b(t)}(t)$, et s'arrête au tour

$$\tau^{\text{LUCB}} := \inf\left\{t \geq 0 : \mathcal{B}_{c(t),b(t)}(t) < \varepsilon\right\} .$$

Le second suit la valeur de l'écart entre le bras $b(t)$ et le $N^{\text{ème}}$ meilleur bras $a_{(N)}$ ($\mu_{a_{(N)}} := \max_{k\in[K]}^{N} \mu_k$, qui est la $N^{\text{ème}}$ plus grande observation moyenne), en considérant $\max_{a\neq b(t)}^{N} \mathcal{B}_{a,b(t)}(t)$. Le critère d'arrêt correspondant est

$$\tau^{\text{UGapE}} := \inf\left\{t \geq 0 : \max_{a\in J(t)} \max_{b\neq a}^{N} \mathcal{B}_{b,a}(t) < \varepsilon\right\} .$$

Ces deux types d'algorithmes sont parfaitement définis une fois leur règle d'échantillonnage implémentée. J'ai montré que le critère d'arrêt τ^{UGapE} était plus agressif que τ^{LUCB} (c'est-à-dire, qu'avec les mêmes observations au cours de la phase d'apprentissage, les algorithmes Gap-GIFA tirent moins de bras que les algorithmes LUCB-GIFA).

Enfin, j'ai testé les performances de ces algorithmes avec deux algorithmes qui ne tiennent pas compte des vecteurs caractéristiques des bras. J'ai considéré un petit problème de repositionnement pour $N = 3$ candidats et un taux d'erreur maximal de 10 % sur un ensemble de 21 médicaments pour l'épilepsie (avec 10 anti-épileptiques, et 11 pro-convulsivants), en transformant les signatures de médicaments pour obtenir un modèle linéaire. Les algorithmes n'utilisant pas les signatures échantillonnent toujours plus de $10\,000$ bras, alors que le nombre d'échantillons pour les deux algorithmes GIFA proposés est inférieur à 400, ce qui montre que tenir compte de ces vecteurs caractéristiques permet effectivement d'efficacement réduire le nombre d'échantillons pendant l'apprentissage.

Intégration des vecteurs caractéristiques dans le modèle de bandit. Cependant, on peut remettre en cause l'hypothèse de linéarité (Condition (2)) dans le modèle de scores, en particulier dans le contexte de données réelles. En effet, si le modèle sous-jacent n'est pas adapté aux données, alors la régression effectuée sur le paramètre θ à partir des observations peut donner une approximation $\hat{\theta}$ trop mauvaise (*i.e.*, la distance entre $\hat{\theta}^\top X_a$ et μ_a est importante). Alors, le « bon » évènement, défini précédemment avec l'hypothèse de linéarité, peut être faux avec une probabilité supérieure à δ. Donc l'algorithme peut retourner des résultats incorrects avec une probabilité plus grande que δ. Toutefois, le modèle linéaire est intéressant car la procédure de régression ainsi que l'interprétation du modèle y sont simples.

J'ai donc travaillé[30] par la suite sur le problème des erreurs de spécifi-

[30] Réda, Tirinzoni, and Degenne (2021). "Dealing With Misspecification In Fixed-Confidence Linear Top-m Identification". *Advances in Neural Information Processing*

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

cation (de modèle) pour le problème d'identification des N meilleurs bras. Pour ce faire, j'ai étudié un type de modèle, appelé « déviés de la linéarité » (*linear misspecified models*). Un tel modèle comprend une partie linéaire (avec un paramètre inconnu θ comme dans la Condition (2)) et une partie sans structure (avec un paramètre inconnu η) qui n'utilise pas le vecteur caractéristique associé au bras :

$$\exists \theta \in \mathbb{R}^d \exists \eta \in \mathbb{R}^K \forall a \in [K], \mu_a = \theta^\top X_a + \eta_a \,. \tag{3}$$

La valeur absolue des cœfficients θ est comme précédemment associée à l'influence de chaque gène sur le score moyen de repositionnement, en introduisant une valeur de biais η_t propre au traitement t. On suppose comme seule hypothèse supplémentaire que l'on connaît une borne supérieure Ψ nommée « déviation maximale » sur la valeur absolue de la partie non structurée η_a pour tout bras a : $\|\eta\|_\infty := \max_{a \in [K]} \|\eta_a\| \leq \Psi$. Un tel modèle est dit « Ψ-dévié ». Le but est de construire un algorithme qui suppose que les scores résultent d'un modèle dévié de la linéarité (Condition (3)), et qui identifie les N meilleurs bras avec une probabilité supérieure à $1-\delta$, avec le moins d'observations possible. Dans le travail fait dans cette partie de la thèse, une borne inférieure sur le nombre d'observations nécessaire pour qu'un algorithme $\mathfrak{A}$ retourne un ensemble de candidats correct avec probabilité $1-\delta$ (Condition (1)) a été trouvée, et peut être calculée numériquement

$$\mathbb{E}[\tau_{\mathfrak{A}}] \geq (\mathcal{C}^\star)^{-1} \log\left(\frac{1}{2.4\delta}\right) \text{ où } \mathcal{C}^\star := \sup_{\omega \in \triangle_K} \min_{\substack{i \in \mathcal{S}_N^\star \\ j \notin \mathcal{S}_N^\star}} \inf_{\lambda \text{ alternatif}} \frac{1}{2\sigma^2} \sum_{a \in [K]} \omega_a (\mu_a - \lambda_a)^2 \,, \tag{4}$$

est appelé « temps caractéristique ». $\mathcal{C}^\star$ est proportionnel à une combinaison convexe de différences (bras à bras) entre l'observation moyenne μ_a dans l'environnement que l'on considère, et celle λ_a dans un environnement « alternatif ». Cet environment alternatif satisfait la Condition (3), et est tel que l'ensemble des N meilleurs candidats dans cet environment soit différent de l'ensemble des meilleurs bras $\mathcal{S}_N^\star$. $\triangle_K := \{p \in [0,1]^K : \sum_{k \in [K]} p_k = 1\}$ est l'ensemble des allocations sur un ensemble fini à K éléments. L'allocation $\omega \in \triangle_d$ donne alors la probabilité avec laquelle tirer chaque bras $a \in [K]$ pour se rapprocher de la borne inférieure. Étant donné la Condition (4), si de plus un algorithme $\mathfrak{A}$ satisfait la Condition (1) et qu'il atteint cette borne asymptotiquement pour de petites valeurs de δ, alors l'algorithme $\mathfrak{A}$ est dit « asymptotiquement optimal ». L'avantage d'avoir une borne inférieure calculable numériquement, c'est qu'elle peut être utilisée dans le corps de l'algorithme afin que celui-ci puisse être asymptotiquement optimal. J'ai donc travaillé sur la conception et à l'analyse d'un tel algorithme nommé

Systems, 34.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

MisLid. Si $N_a(t)$ est le nombre de fois où l'agent sélectionne le bras a jusqu'au tour t inclus, le principe de l'algorithme est de tirer des bras tant que le critère d'arrêt suivant, lié à l'Équation (4)

$$\inf_{\lambda \text{ alternatif}} \frac{1}{2\sigma^2} \sum_{a \in [K]} N_a(t)(\lambda_a - \tilde{\mu}_a(t))^2 < \mathcal{T}_{t-1}(\delta) \ ,$$

n'est pas vérifié au tour $t \geq 1$. $(\tilde{\mu}_a(t))_{a \in [K]}$ est la projetée, sur l'ensemble des modèles Ψ-déviés, des moyennes empiriques $(\widehat{\mu}_a(t))_{a \in [K]}$ pour chaque bras a au tour t. $\mathcal{T}_\cdot(\delta)$ est un seuil qui quantifie la distance maximale tolérée entre le modèle empirique et un modèle Ψ-dévié alternatif. La définition de $\mathcal{T}_\cdot(\delta)$ garantit que MisLid satisfait la Condition (1). À la fin de sa phase d'apprentissage, MisLid retourne les N bras optimaux selon $(\widehat{\mu}_a(\tau_{\text{MisLid}}))_{a \in [K]}$

$$\hat{\mathcal{S}}_N(\tau_{\text{MisLid}}) := \underset{k \in [K]}{\overset{N}{\arg\max}} \ \widehat{\mu}_k(\tau_{\text{MisLid}}) \ .$$

J'ai comparé cet algorithme avec un algorithme GIFA (N-LinGapE) et un autre algorithme (LUCB) n'utilisant pas la structure des scores sur le même problème de repositionnement sur 21 médicaments, en transformant les signatures pour satisfaire la Condition (3) et connaître la valeur de la déviation maximale Ψ. La connaissance de cette déviation est malheureusement indispensable pour que l'algorithme soit plus performant qu'un algorithme n'utilisant pas la structure des scores. La table 1 présente les résultats obtenus. En comparant les nombres d'observations entre LUCB et MisLid, on observe, comme précédemment, qu'il est avantageux de prendre en compte les signatures des médicaments pour limiter le nombre d'observations. De plus, sur ce modèle des scores non linéaires, N-LinGapE se trompe systématiquement sur les recommandations de candidats, contrairement à MisLid.

Algo.	$\hat{\delta}$	$\hat{s}$
MisLid	0 %	$158\ 869 \pm 126\ 209$
LinGapE	100 %	161 ± 159
LUCB	0 %	$222\ 969 \pm 22\ 798$

Table 1: La table des résultats associée. $\hat{\delta}$ est la fréquence d'erreur dans les 100 itérations, $\hat{s}$ est le nombre moyen ($\pm$ l'écart-type), arrondi à l'entier le plus proche, d'échantillons utilisés.

Extension à différentes populations de patients. Dans la dernière partie de la thèse, je m'intéresse[31] au problème en **(D)** relatif à la définition d'une recommandation personnalisée. En effet, toutes les approches de

[31] Réda, Vakili, and Kaufmann (2022). "Near-Optimal Federated Learning in Bandits". 36th *Conference on Neural Information Processing Systems*. In press.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

bandit précédentes supposent l'existence d'une population relativement uniforme dans ses réponses à un médicament (avec des observations bruitées avec une variance $\sigma^2 = 1$ dans mes expériences) pour laquelle on veut trouver un ensemble de candidats.

Maintenant, comment procéder si les populations sont disparates, par exemple au niveau génomique, ou au niveau du sous-type de maladie dont elles sont atteintes ? Dans ce cas-là, on aimerait pouvoir recommander des ensembles de candidats spécifiques à chaque population, tout en exploitant les observations obtenues sur d'autres populations, afin de limiter le nombre d'observations. Toutefois, partager chaque nouvelle information entre toutes les populations peut aussi être coûteux, et donc le coût de communication doit également être réduit, en plus du nombre d'observations.

Similairement au concept d'apprentissage de bandit fédéré introduit dans le cadre de la minimisation de regret cumulatif,[32] j'ai travaillé sur une version (plus générale) de l'identification centralisé de N meilleurs bras, qui n'utilise pas [33] les vecteurs caractéristiques. J'ai prouvé une borne inférieure sur l'espérance du nombre d'observations pour retourner un ensemble de candidats correct avec probabilité $1-\delta$ dans ce cadre centralisé. J'ai également travaillé sur une nouvelle approche pour obtenir un algorithme quasi optimal. Au lieu de résoudre de manière répétée le problème dans l'Équation (4), ou de le faire de façon incrémentale comme pour MisLid, l'idée est de procéder par phases, dans laquelle l'on résout un problème plus simple qui est proche, et l'on utilise l'allocation calculée sur ce problème pour tirer les bras. L'information sur les différentes populations n'est alors partagée qu'à la fin des échantillonnages de la phase courante pour tous les groupes.

J'ai prouvé que l'algorithme correspondant donnait des recommendations correctes avec probabilité $1-\delta$, et l'ai ensuite appliqué cet algorithme au problème de repositionnement des 21 médicaments, où les patients sont clusterisés en M groupes en fonction de leur profil transcriptionnel sur $d = 194$ gènes de M30. J'ai comparé ces résultats avec une méthode de référence inspiré du cadre fédéré qui a motivé ce travail[34]. Les résultats sont présentés dans la Table 2.

[32] C. Shi and C. Shen (2021). "Federated multi-armed bandits". *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*; C. Shi, C. Shen, and Yang (2021). "Federated multi-armed bandits with personalization". *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2917–2925.

[33] Encore...

[34] C. Shi and C. Shen (2021). "Federated multi-armed bandits". *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

α	0.4		0.6		0.8	
	$\hat{s}$	$\hat{r}$	$\hat{s}$	$\hat{r}$	$\hat{s}$	$\hat{r}$
CPE	$47\ 418 \pm 1\ 016$	6 ± 0	$59\ 537 \pm 1\ 677$	6 ± 0	$70\ 201 \pm 2\ 300$	6 ± 0
Réf.	$230\ 836 \pm 55\ 904$	13 ± 0	$92\ 278 \pm 1\ 071$	12 ± 0	$64\ 393 \pm 376$	11 ± 0

Table 2: La table des expériences dans le cadre de l'apprentissage centralisé ($\delta = 10$ %, $N = 1$, $K = 21$, $M = 3$, 100 itérations). La fréquence d'erreur $\hat{\delta}$ est à 0 % dans toutes les expériences. $\hat{s}$ est le nombre moyen (arrondi à l'entier le plus proche) d'échantillons utilisés ($\pm$ l'écart-type), et $\hat{r}$ est le nombre moyen de phases de communication. Notre algorithme est celui dénommé CPE. La référence n'est valable que pour $N = 1$ et pour $W = \alpha I_M + \frac{1-\alpha}{M}\mathbb{1}_{M\times M}$, avec $\alpha \in [0, 1]$, et $\mathbb{1}_{M\times M}$ est la matrice remplie de 1.

Conclusion. Ma thèse se situe à l'interface entre la bioinformatique et le traitement de données génomiques, à travers l'utilisation de réseaux booléens, et l'apprentissage statistique, et plus précisément les algorithmes de bandit, qui permettent d'exploiter le score retourné par le réseau booléen. Ce travail de thèse s'est concentré sur quatre objectifs :

(A) la construction d'un modèle simulant l'effet de médicaments sur un réseau de régulation transcriptionnel ;

(B) l'automatisation quasi complète de cette procédure ;

(C) la conception d'une méthode de recommandation de candidats thérapeutiques exploitant le modèle en (A) ;

(D) l'extension de cette méthode à un contexte où l'apprentissage est centralisé.

Ce travail est une nouvelle étape vers l'automatisation du processus de recherche pour le développement de médicaments, qui permet de diminuer le temps d'identification de molécules d'intérêt, mais aussi, en exploitant les données disponibles de façon transparente et reproductible, de faire face au problème des maladies rares.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Publications

Articles de journaux scientifiques / Journal papers

Delahaye-Duriez, Réda, and Gressens (2019). "Identification de cibles thérapeutiques et repositionnement de médicaments par analyses de réseaux géniques". *médecine / sciences*, 35(6-7), pp. 515–518

Réda, Kaufmann, and Delahaye-Duriez (2020). "Machine learning applications in drug development". *Computational and structural biotechnology journal*, 18, pp. 241–252

Actes de conférences / Conference proceedings

Réda, Kaufmann, and Delahaye-Duriez (2021). "Top-m identification for linear bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1108–1116

Réda, Tirinzoni, and Degenne (2021). "Dealing With Misspecification In Fixed-Confidence Linear Top-m Identification". *Advances in Neural Information Processing Systems*, 34

Réda and Delahaye-Duriez (2022). "Prioritization of Candidate Genes Through Boolean Networks". In: *International Conference on Computational Methods in Systems Biology*. Springer, pp. 89–121

Réda, Vakili, and Kaufmann (2022). "Near-Optimal Federated Learning in Bandits". In: 36th *Conference on Neural Information Processing Systems*. In press

Articles en préparation / Papers in preparation :

Cet article est en cours de préparation pour soumission cette année, et fait l'objet du Chapitre 7.

This paper is in preparation for submission this year, and is the topic of Chapter 7.

Bokobza, Réda, et al. (*in prep.*). "Therapeutic evaluation of Hu-MSCs in a rat model of perinatal inflammation: a systematic outcome scoring"

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Remerciements

À ma famille et mes amis.

Je remercie :

- mes deux directrices de thèse, Andrée Delahaye-Duriez et Émilie Kaufmann, pour l'aide et le temps qu'elles m'ont consacré tout au long de ma thèse ainsi que dans la rédaction du manuscrit ;

- mes (anciens comme actuels) collègues de travail de l'unité Inserm Neurodiderot à Paris (et plus particulièrement, [35] Cindy Bokobza, Thomas Bourgeois, David-Peter Cohen [36], Adrien Dufour, Jorge Gallego, Pierre Gressens, Christophe Le Priol, Amélia Madani, Boris Matrot, Amazigh Mokhtari, Baptiste Porte, Nélina Ramanantsoa, Éléonore Sizun, Juliette Van Steenwinckel) ainsi que ceux de l'unité Inria SCOOL à Lille (dont [37] Omar Darwiche Domingues, Rémy Degenne, Andrea Tirinzoni, Xuedong Shang, Jill-Jênn Vie) pour leurs conseils avisés et leurs enseignements ;

- mes co-auteurs externes (dont [38] Sophie Foulon, Sophie Lemoine, Philippe Nghe, Sattar Vakili) qui m'ont partagé leurs connaissances ;

- et enfin, les membres du jury pour avoir accepté d'examiner (et de rapporter) cette thèse.

[35] Par ordre alphabétique.
[36] Who has provided a comprehensive feedback on a large part of the manuscript.
[37] Itou.
[38] Itou (*bis repetita*).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

List of Figures

1 Drug development timeline and principle of drug repurposing . 2
2 Transcriptomics 5

1.1 Lac operon regulation as a gene regulatory network 17
1.2 Gene regulatory networks and Boolean networks 18
1.3 Dynamics in a Boolean network 20

2.1 Automated building of Boolean networks 29
2.2 Inferred network for the application to epilepsy 36
2.3 Comparison of the inferred networks 37
2.4 Illustration of the collapsed network 37
2.5 Comparison between *spread* values and other gene measures . 41
2.6 Gene ranking by decreasing *spread* value 41

3.1 Visualization of gene targets for antiepileptic and proconvulsant drugs 51
3.2 Comparison between the different types of drug signatures . . . 53
3.3 Drug repurposing on a set of antiepileptics and proconvulsant drugs 57
3.4 Comparison between the drug repurposing method and the baseline 57

4.1 Multi-armed bandits 64
4.2 Drug repurposing with bandits 75

5.1 Drug repurposing instance with linear bandits 95
5.2 Comparison between the two stopping rules 96

6.1 Drug repurposing with misspecified bandits 112

7.1 Timeline of the experiments 119
7.2 Application of signature reversion to encephalopathy of prematurity 121
7.3 Comparison between Characteristic Direction signatures and univariate differential analysis 123

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

7.4 Correlation between score and number of common differentially expressed genes 124
7.5 Enrichment analysis of differentially expressed genes in the best recommended treatment 126
7.6 Enrichment analysis of differentially expressed genes in the second best recommended treatment 127
7.7 Rankings from cosine scores and repurposing scores on a subset of drugs 129
7.8 Performance of cosine scores and repurposing scores on a subset of drugs 129

8.1 Clustering of epileptic profiles 151

9.1 Comparison of the 50 inferred networks 167
9.2 Enrichment analyses for *spread*-selected genes 171

11.1 Cosine method applied to the whole set of 34 drugs 177

14.1 Visualization of the "real life" repurposing instance 195
14.2 Results for the drug repurposing in epilepsy 196

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

List of Tables

1.1 Examples of regulatory interactions 13

2.1 Distribution statistics for the inferred networks 35
2.2 Distribution statistics of the *spread* values 40
2.3 Scores for genes prioritized by *spread* values 44

3.1 Hit ratios on the proposed drug scoring and the baseline 56

4.1 Prior results on the sample complexity for unstructured bandits 72

5.1 Adaptive sampling-based algorithms for linear Top-N identification 82
5.2 Sample complexity results for linear bandits 91

6.1 Running a linear bandit algorithm on an increasingly non linear model 97
6.2 Results for drug repurposing with misspecified bandits 112

7.1 Experimental batches for rat transcriptional profiles 118
7.2 Ranking of the treatment protocols 122

8.1 Results for collaborative drug repurposing 152

9.1 Distribution statistics for the 50 inferred networks 167
9.2 Experimental profiles used for network inference 168
9.3 Parameters for the network inference 168
9.4 Parameters for the retrieval of disease-associated genes 168
9.5 Regulatory interactions present in all inferred networks 170

10.1 List of antiepileptics 173
10.2 List of proconvulsants 174
10.3 Bandit instance with 21 drugs 175

14.1 Parameters values for the drug repurposing in epilepsy 196
14.2 Results for the drug repurposing in epilepsy 197

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

List of Algorithms

1 Gene influence maximization 32
2 Single gene perturbation scoring 42
3 Drug perturbation scoring 55
4 Sequential drug perturbation scoring 59
5 Top-N identification 71
6 Structure of GIFA 81
7 MisLid algorithm for misspecified models 105
8 CPE algorithm for the collaborative setting 144
9 PF-UCB-BAI 149

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Acronyms and abbreviations

i.e., (fr : *c.à.d.*, c'est-à-dire) *id est*.
e.g., *exempli gratia*.
2**D** Two dimensional.
A Adenine.
API Application programming interface.
AUC Area under the curve.
BAI Best arm identification.
BH Benjamini-Hochberg (Benjamini and Hochberg, 1995).
C Cytosine.
CC Control Centrality (Liu, Slotine, and Barabási, 2012).
CD Characteristic Direction (Clark et al., 2014).
cDNA Complementary DNA.
Centr (Network) centralization.
CID Concept identifier (ID) (Doğan, Leaman, and Lu, 2014).
CL (Network) clustering coefficient.
CMap Connectivity map (Lamb et al., 2006).
Covid-19 Coronavirus 2019.
CPE Collaborative phased elimination (Réda, Vakili, and Kaufmann, 2022).
CPU Central processing unit.
CRISPR Clustered regularly interspaced short palindromic repeats.
DMSO Dimethyl sulfoxide.
DNA (fr : ADN, acide désoxyribonucléique) Deoxyribonucleic acid.
DS (Network) density.
EDS Enhancer-domain score (Wang and Goldstein, 2020).
EHR Electronic health records.
EoP Encephalopathy of prematurity.
G Guanine.
GDSC Genomics of Drug Sensitivity in Cancer (database) (Yang, Soares, et al., 2012).
GEO Gene Expression Omnibus (Barrett et al., 2012).
GIFA Gap-index focused algorithms (Réda, Kaufmann, and Delahaye-Duriez, 2021).
GLRT Generalized likelihood ratio test.
GO Gene Onthology (Ashburner et al., 2000).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

GRN Gene regulatory network.
GT (Network) heterogeneity.
GTP General topological parameter.
HN-h Hippocampal neuron (in) human.
HR Hit ratio.
IL1β Interleukin-1-beta.
INAS Intranasal (injection).
IV Intravenous (injection).
KD Knockdown (of a gene).
KL Kullback-Leibler divergence (Kullback and Leibler, 1951).
MBP Myelin basic protein.
MisLid Misspecified linear identification algorithm (Réda, Tirinzoni, and Degenne, 2021).
mRNA (fr : ARNm, acide ribonucléique messager) Messenger ribonucleic acid.
MSC Mesenchymal stem cell.
NP-hardness Non-deterministic polynomial-time hardness.
ORA Overrepresentation analysis (Yaari et al., 2013).
P Postnatal day.
PAC Probably approximately correct (Valiant, 1984).
PBS Phosphate-buffered saline.
PCA Principal component analysis.
pLI Probability of loss-of-function intolerance (Lek et al., 2016).
PR Precision-recall.
RF (Gene) regulatory function.
ROC Receiver operating characteristic.
RNA (fr : ARN, acide ribonucléique) Ribonucleic acid.
RVIS Residual variation intolerance score (Petrovski et al., 2013).
shRNA Short (or small) hairpin RNA.
STD Standard deviation.
STG State-transition graph.
T Thymine.
TF Transcription factor.
TLE Temporal lobe epilepsy.
Top-N Top-N identification.
TTD Therapeutic Target Database (Zhou et al., 2022).
U Uracil.
UCB Upper confidence bound.
WMI White matter injury.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Contents

Abstract / Résumé

Substantial abstract (in French) / Résumé substantiel en français i

List of papers / Liste des publications xviii

Acknowledgements / Remerciements xix

Figures / Liste des figures xxi

Tables / Liste des tableaux xxii

Algorithms / Liste des algorithmes xxiii

Acronyms and abbreviations / Liste des sigles et abréviations xxiv

Table of contents / Table des matières xxviii

Introduction 2

I Analyzing a disease-specific regulatory network 10

1 Introduction to Boolean networks 12

1.1 Modelling gene expression with regulatory networks 13
1.2 Overview of gene regulatory networks 15
1.3 Introduction of dynamics with Boolean networks 16
1.4 Boolean networks in drug repurposing 21

2 Prioritization of candidate genes through Boolean networks 23

2.1 Related work 24
2.2 Reproductible inference of a cell-line specific Boolean network . 26
2.3 Detection of master regulators in a specific disease-context . . 29
2.4 Results 34
2.5 Discussion 45

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

3 Drug efficacy scoring using a Boolean network **46**
3.1 Related work 47
3.2 Selection of drug signatures 50
3.3 Scoring with the Boolean network 54
3.4 Results 56
3.5 Discussion 58

II Adaptive drug testing using bandits **60**

4 Introduction to Top-N identification **62**
4.1 Structured stochastic bandits 64
4.2 Bandit algorithms for Top-N identification 70
4.3 Sample complexity in bandits 71
4.4 Integration of bandits to drug repurposing 75

5 Sequential identification in linear models **76**
5.1 Related work 77
5.2 General structure for efficient algorithms 78
5.3 Theoretical guarantees in GIFA 87
5.4 Application to drug repurposing 93
5.5 Discussion 96

6 Dealing with non-linear models **97**
6.1 Related work 99
6.2 Assumptions 100
6.3 Tractable lower bound for general Top-N identification 100
6.4 Misspecified Top-N identification 103
6.5 Application to drug repurposing 110
6.6 Discussion 111

III Application & extension of drug repurposing **113**

7 Application of systematic drug scoring **115**
7.1 Related work 116
7.2 Setting and experimental data 117
7.3 Systematic outcome scoring 119
7.4 Results 122
7.5 Discussion 128

8 Extension to a collaborative setting **130**
8.1 Collaborative Top-N (identification) 131
8.2 Related work 134

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

8.3 Lower bound for collaborative Top-N 136
8.4 Structure for nearly-optimal Top-N 140
8.5 Application to drug repurposing 148
8.6 Discussion 152

Conclusion **155**

Appendix / Appendices **159**

9 Chapter 2 : Building the cell line specific Boolean network **159**
9.1 Building the Boolean network 159
9.2 Robustness on a larger set of 50 solutions 167
9.3 Tables 168
9.4 Implementation of the influence maximization algorithm 169
9.5 Additional results 170

10 Chapter 3: Drug repurposing scoring **172**
10.1 List of antiepileptic and proconvulsant drugs 172
10.2 Drug repurposing instances in bandits 175

11 Chapter 7 : Results for the cosine method applied to the "stabilized" profiles **176**

12 Chapter 5: Theoretical guarantees for GIFA **178**
12.1 Conjecture 5.3.9: sample complexity analysis for Gap-GIFA . 178
12.2 Conjecture 5.3.10: analysis of the greedy selection rule 182

13 Chapter 8 : Proofs for CPE **186**
13.1 Technical lemmas 186
13.2 Correctness analysis 188
13.3 Sample complexity analysis 188

14 Chapters 5 & 6: Real-life conditions for drug repurposing **194**

Bibliography / Bibliographie **222**

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Introduction

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

The contents of this chapter rely on some of my publications.[a]

[a]Réda, Kaufmann, and Delahaye-Duriez (2020). "Machine learning applications in drug development". *Computational and structural biotechnology journal*, 18, pp. 241–252.

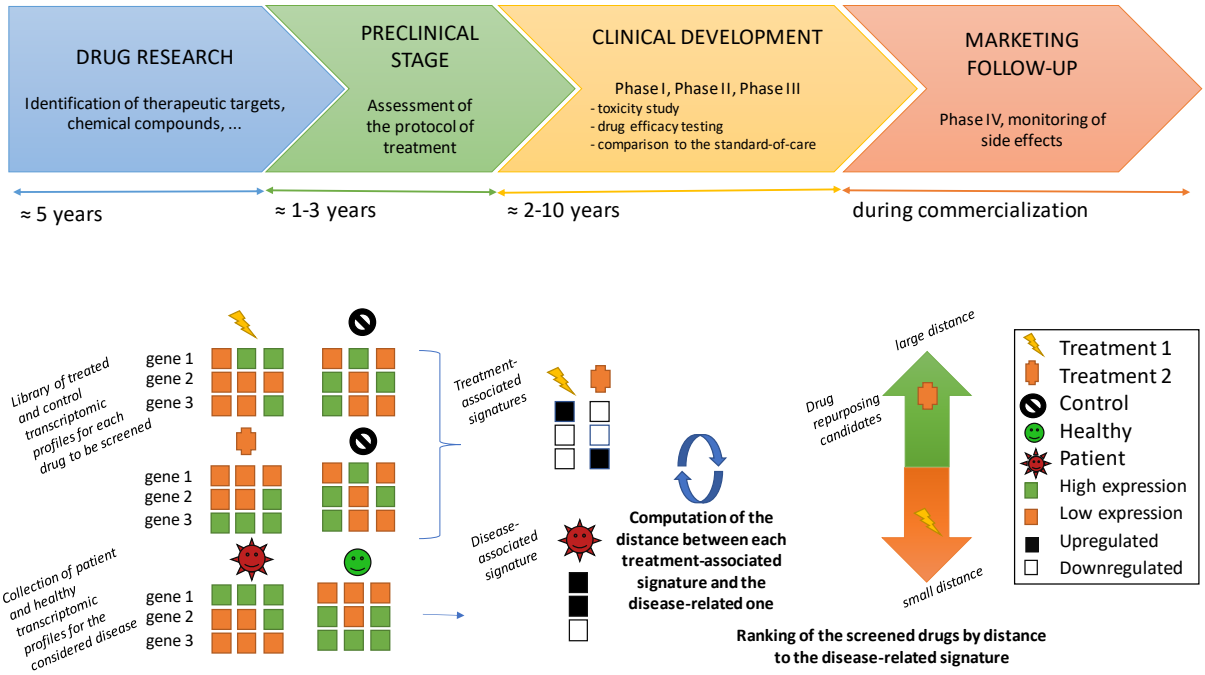

Figure 1: **Drug development timeline and principle of drug repurposing. Top plot** : Timeline for a typical drug development pipeline, comprising of four main research stages followed by post-commercialization monitoring. **Bottom plot** : Illustration of the approach of "signature reversion" for drug repurposing.

Development of new drugs is a time-consuming and costly process (Réda, Kaufmann, and Delahaye-Duriez, 2020). Indeed, in order to ensure both the patients' safety and drug effectiveness, prospective drugs must undergo a competitive and long procedure. This process, from the identification of a molecule of interest to its commercialization, is completed on average in 5 years, and can take up to 10 years, depending on the considered disease, and have millions of dollars spent -still next to $\$2$ billion on average for major pharmaceutical labs in 2021 (Deloitte Centre for Health Solutions, 2022). Figure 1 illustrates the typical pipeline for drug development. Drug development is roughly split into four major stages, called phases. The preclinical phase comprises of basic research, drug discovery and preclinical tests, which aim at assessing the efficiency and body processing of the drug candidate. The last three stages are clinical trials : study of dose-toxicity, short-lived side effects, and kinetic relationships (Phase I) ; determination

Except where otherwise noted, this work is licensed under https://creativecommons.org/licenses/by-nd/3.0/fr/

of drug performance (Phase II) ; and comparison of the molecule to the standard-of-care (Phase III). An optional Phase IV (post-drug marketing) can be set to monitor long-lasting side effects and drug combination with other therapies. In addition to being time-consuming and expensive, drug development is subject to high failure rates, event in the latest stages of the pipeline (Burki, 2020). Only 64% of the molecules which reached Phase III after 2007 –the third block on the timeline in Figure 1– were marketed by 2012 (Lowe, 2019). Main reasons for failure were the lack of efficacy (57% of the failing drug candidates), and safety concerns (17%). Among safety concerns were increased risk of death or of serious side effects, which were still the main reasons of failure in Phases II and III in 2012 (Schuhmacher, Gassmann, and Hinder, 2016), and in 2019 (Lowe, 2019).

Although political efforts have been made to promote orphan disease research –the Orphan Drug Act in 1983 in the United States, three national reasearch plans in France, with the latest one passed in 2018 (Ministère des solidarités et de la santé, 2018)– with some success (Institute for Clinical and Economic Review (ICER), 2022), this situation has led the pharmaceutical industry to focus on the most profitable diseases. Cancer still represents the most studied therapeutic area in 9 of the biggest 10 pharmaceutical companies ; although a shift has been observed towards coronavirus 2019 (Covid-19) treatments in all of these companies. At least 4 drug pipelines related to Covid-19 were launched per company for treatment, disease prevention, or complications. This observation raises the issue of finding therapies for rarer, complex diseases, where the limited number of patients might hinder meaningful studies to be carried on ; or for tropical "neglected" diseases, such as malaria, where the drug development cost might be too prohibitive with respect to the estimated selling profits (Walker, Hamley, et al., 2021). As highlighted by Carbonell, Radivojevic, and Garcia Martin (2019); Mak and Pichika (2019), one rather inexpensive way to improve these numbers might be to automate some important but repetitive data processing and analysis tasks. For instance, artifical intelligence is massively featured in drug development methods during the Covid-19 outbreak (Chen, See, et al., 2020; Jamshidi et al., 2020; Vaishya et al., 2020). Such collaborations between researchers in artificial intelligence, pharmaceutical labs, and hospitals slowly bridge the gap in bioinformatics between applied mathematics, computer sciences and biology. This would accelerate drug development pipelines as they might be computationally, thus automatically, performed and less prone to human-related technical mistakes.

Moreover, thanks to technological breakthroughs in bioinformatics, a large amount of biological information and experiments is publicly available ; for

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

instance, the Gene Expression Omnibus (GEO) database (Barrett et al., 2012), which collects various types of data. This allows a lesser reliance on the access to wet-lab experiments to study biological phenomena. Again, automated methods intervene in the transparent and reproducible processing of this data. This data availability, combined with the rise of machine learning techniques, has led researchers to consider screening known molecules for a specific therapeutic indication, and finding them a new *purpose*, instead of designing *de novo* molecules. This approach to drug development is called *drug repurposing* –or *drug repositioning*– and might allow to decrease the failure rates and the development time. Indeed, repurposed drugs have well-documented safety-profiles –that is, side effects are known, thus avoiding the discovery of negative side effects in the later stages of development (Mouchlis et al., 2021)– and the preclinical phase of dose and administration route testing have already been performed. Different approaches have been used to tackle the drug repurposing problem. For example, some rely on automatic processing of Electronic Health Records (EHR), clinical trial data, and text mining methods to identify correlations between drug molecules and gene or protein targets in literature (Andronis et al., 2011; Bisgin et al., 2012; Tari and Patel, 2014). However, this approach might be sensitive, but not really specific, since text interpretation is still a hard problem, and the relationship between disease factors and drugs might not be clear. The current state-of-the-art methods seem to have turned to different paradigms of repurposing, as emphasized by the following review papers (Alaimo, Giugno, and Pulvirenti, 2016; Hodos et al., 2016; Sardana et al., 2011). However, most of these methods rely on a rather strong hypothesis, which is that similarity between elements –for instance, chemical composition of drug molecules– implies correlation at drug target level. Nonetheless, some counter-examples to this hypothesis have led to health disasters : for instance, thalidomide exists as two chiral forms –that is, with the same chemical composition but having mirrored structures. One of these forms can treat morning sickness ; the other form can have teratogen effects (Vargesson, 2015).

An attempt at quantifying more accurately drug effects is *signature reversion*, also called *connectivity mapping*, which focuses on transcriptomic measurements, that is, relative to the production of molecules as encoded by gene sequences. Indeed, the transcriptome is the set of the different types of ribonucleic acids (RNA), which contain a part of the genomic information encoded in the deoxyribonucleic acid (DNA). RNA is key to the production of proteins, and, as such, ensures that vital chemical reactions occur, as illustrated in Figure 2. While DNA is a double helix comprising of four different small unit called nucleobases : adenine, guanine, cytosine, thymine,

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

which are usually denoted by the letters A, G, C, T and paired (Watson and Crick, 1953), RNA are single-strand molecules composed of four nucleobases (adenine, guanine, thymine, and uracil, which "replaces" cytosine). First, immature messenger RNAs are usually produced by "copying" a strand of DNA in the cell nucleus, where DNA resides, during a process called transcription. If a gene sequence is represented in the RNA, the sequence of RNA corresponding to that gene sequence is called gene transcript. After maturation, messenger RNA (mRNA) is transported to the cell cytoplasm, outside the nucleus, to be read by a small unit called ribosome during the process of translation. During translation, the RNA sequence is read by triplets of nucleobases (called codons). Once the ribosome encounters a start codon (usually in human, AUG), each following codon is successively matched to a single aminoacid, which is concatenated to previous aminoacids, until the ribosome reaches one of the stop codons. This procedure generates a single protein. Then RNAs are interesting (proxy) snapshots of the state of protein production at a given time. In particular, in order to quantify this, one needs to detect gene transcripts, and count them, assuming that the number of transcripts associated with a given gene will be proportional to the quantity of the product encoded by the gene that would have been produced post-translation. This count is called *gene expression*, and can be measured through RNA sequencing for instance. The basic idea of RNA sequencing is to align a genomic sequence, obtained from the mRNAs, to a reference genome, and then count the number of successful matches to a gene sequence of the reference genome.

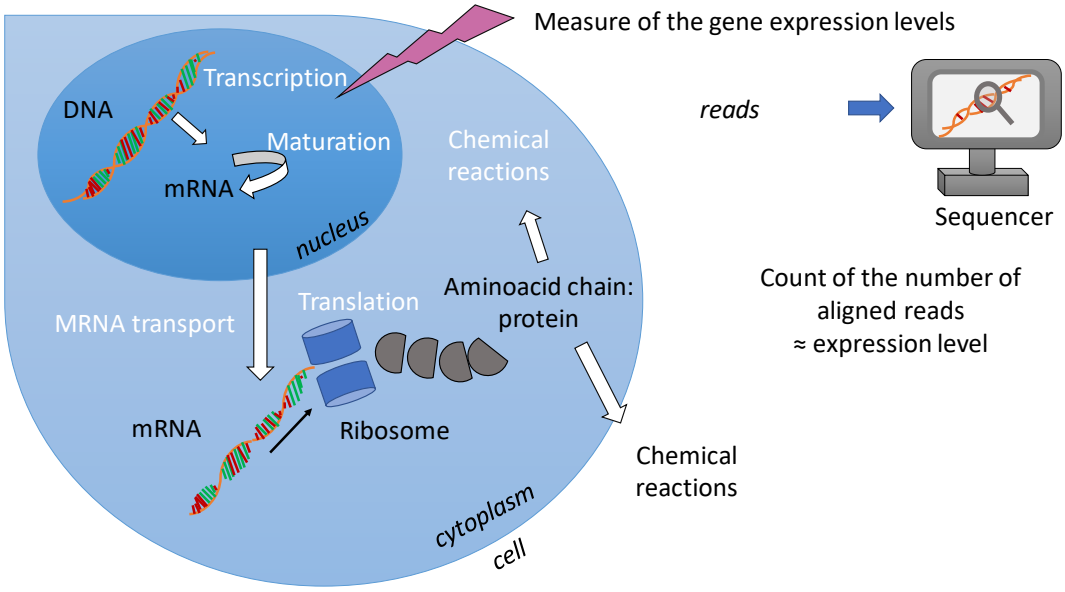

Figure 2: **Transcriptomics.** Transcriptomics : relationship between deoxyribonucleic and ribonucleic acids (DNA and RNA), and the measure of gene expression.

Except where otherwise noted, this work is licensed under https://creativecommons.org/licenses/by-nd/3.0/fr/

Signature reversion leverages gene expression data in order to screen molecules for a specific indication. Given a "signature", that is, a subset of genes which are perturbed in patients at expression level –for instance, with increased or decreased expression levels in patients with respect to healthy individuals– the objective is to identify which treatments are most able to revert this signature ; that is, increasing the expression levels of genes with decreased expression in patients, and vice-versa. This operation is performed via comparisons of the query signature with "drug signatures", that is, summaries of genewise expression changes, this time due to the drug treatment. This approach has recorded some successes in drug repurposing, as noticed in Musa et al. (2018). For example, Xin et al. (2022) recently applied this approach for drug repurposing against influenza. They identified among potential drug candidates nifurtimox, which is usually prescribed to treat Chagas' disease, and could successfully decrease the number of influenza virions in later experiments (World Health Organization (WHO), 2021). The bottom diagram in Figure 1 illustrates each step of signature reversion to find novel drugs against a fixed pathology of interest :

(1). First, gene expression levels are measured in each group of treated samples and their controls. [40] (for every molecule to screen) A drug signature is computed from these transcriptomic profiles, which reports genewise expression changes due to the effect of the treatment. A gene which expression is increased (resp., decreased) in treated samples with respect to controls is said *up-regulated* (resp., *down-regulated*). The advantage of drug signatures over directly considering transcriptomic profiles is that the changes in gene expression described in the signature only depends on the treatment, and not on other possibly confounding factors.

(2). A disease-specific signature –that is, the aforementioned query signature– is built in a similar fashion by comparing transcriptomic samples from patients to healthy individuals (or another appropriate control group).

(3). A series of comparisons are performed between the disease signature and each of the drug signatures. If a drug signature is deemed similar to the disease signature, then this means that the treatment incurs the same type of perturbations at gene expression level than the pathology itself, since the same genes are affected by both the treatment and the disease in a similar way. To the contrary, if a drug signature and the disease signature are opposite, then the drug seems to restore the gene expression levels conversely to the effects of the disease. Then the considered drug might be a good candidate for repurposing. Finally, drugs can be ranked according to

[40] All parameters are similar but for the treatment, which can be a sham or no treatment at all.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

their similarity to the disease signature.

In this thesis, I exhibit an application of signature reversion applied to the treatment of encephalopathy of prematurity, a devastating condition due to premature birth, in Chapter 7 (Bokobza, Réda, et al., *in prep.*). This project aimed at selecting the optimal treatment protocol for the injection of stem cells in a rat model of encephalopathy of prematurity : age of injection, administration route, and weight-dependent dose level. Disease and drug signatures were computed as previously described, from the transcriptomic profiles of rats exposed to the treatment, and from appropriate controls. The final ranking of the treatment protocols emphasized the global positive effect of the injection of stem cells in rat models, and further bioinformatical analyses showed that the first two candidates were indeed promising.

However, in that project, I was lucky to get access to appropriate control groups for each treatment, which allowed me to build drug and disease signatures of good quality. That was possible because that work was a collaboration which involved custom wet-lab experiments. However, most existing molecules have not been tested on every pathology and every cell line. This raises the question of designing proper drug signatures when several factors (outside the treatment) might mismatch between samples from patient, healthy individuals, treated and control individuals. Then, I have identified four main questions to further enable the use of automated signature reversion, which constitute the core work in my PhD :

(A). There are online databases which compile drug signatures, based on *in vitro* treatment of human cells in standardized pipelines for transcriptomics (Lamb et al., 2006; Subramanian et al., 2017). These signatures could be alternatives to drug signatures specifically run on diseased profiles. However, the existence of regulatory cascades (Bolouri and Davidson, 2003) [41] rather suggests that appropriate drug signatures are the result of "stabilized" transcriptional profiles, after potential molecular interactions between the perturbed levels of gene expression in the initial patient sample, and the perturbations due to the treatment itself. The method developed in Chapter 3 aims at modelling these regulatory cascades, and stable *in silico* treated profiles, by combining the information about drug-induced gene expression and a cohort of transcriptional profiles from patient and healthy individuals. In order to achieve this goal, I have built a model of gene expression regulation, which is called a *gene regulation network*, that allowed the prediction of gene expression levels under gene perturbations. Drugs can be scored according to the proximity between the corresponding

[41] That is, gene expression changes that trickle downstream of the initial gene perturbations incurred by a treatment.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

in silico treated profiles and profiles from healthy individuals.

(B). In practice, the method mentioned in paragraph **(A)** requires a large amount of transcriptional data. A reproducible and transparent pipeline for collecting and processing these data, in a generic way that is disease-agnostic, is crucial to obtain drug signatures of good quality, and to ensure the replicability of drug repurposing. Chapter 2 (Réda and Delahaye-Duriez, 2022) deals with building the aforementioned gene regulatory network used for predicting drug signatures in a fully automated, reproducible and transparent fashion, from publicly available data.

(C). In the traditional pipeline for signature reversion, as shown at the bottom of Figure 1, drug candidates are ranked by increasing score, which might be the distance to a disease signature. Then, the top-N candidates are selected. However, this recommendation does not control for the error in recommendation. Moreover, a single scalar (score) might not be informative enough about the transcriptional interactions which accounts for its high (or low) value, which hinders the interpretability of the results. Chapters 5 (Réda, Kaufmann, and Delahaye-Duriez, 2021) and 6 (Réda, Tirinzoni, and Degenne, 2021) describe the introduction of sequential learning algorithms to solve these two issues. More specifically, these algorithms adaptively select the drugs to be scored and aim at reducing the number of selections needed to make a recommendation, by leveraging information about drug-induced perturbations of gene expression. Chapter 5 assumes that a linear model connects the drug-associated perturbations to the scores, which increases interpretability, at the cost of a sometimes bad approximation ; whereas Chapter 6 relies on a type of structure which generalizes linear models. These algorithms sequentially interact with the gene regulatory network, by sequentially observing the noisy score obtained for a specific treatment until they can make a guess about the top candidates.

(D). Finally, the pipeline shown at the bottom of Figure 1 only takes into account a global trend at gene expression level in the whole population of patients. However, for some diseases (for instance, epilepsy or brain cancer), there might be several subpopulations of patients with distinct disease subtypes. One might be interested in making personalized drug recommendations for each subpopulation, while leveraging the information about the response to treatment for all subpopulations. Chapter 8 (Réda, Vakili, and Kaufmann, 2022) describes the extension of the framework previously studied in (C) to a collaborative setting, where the information about subpopulations should be shared, but only at times. Indeed, running experiments per batches is more practical and cost-effective, instead of having to wait until all tests have been run on all subpopulations. This issue of

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

delayed feedback, and the need for a finite number of interim analyses, is indeed one of the main issues in adaptive clinical trials (Pallmann et al., 2018; Villar, Bowden, and Wason, 2015), which are the *in vivo* counterpart to drug repurposing.

The ultimate objective of this work is to combine sophisticated techniques for modelling gene expression and recommending candidates to enable automated and transparent *in silico* drug repurposing.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Part I

Analyzing a disease-specific regulatory network

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

The contents of this chapter rely on some of my publications.[a]

[a]Réda and Delahaye-Duriez (2022). "Prioritization of Candidate Genes Through Boolean Networks". *International Conference on Computational Methods in Systems Biology*. Springer, pp. 89–121; Réda, Kaufmann, and Delahaye-Duriez (2020). "Machine learning applications in drug development". *Computational and structural biotechnology journal*, 18, pp. 241–252.

Boolean networks provide a qualitative summary of regulatory interactions at molecular level. Provided the current huge amount and diversity of biological data, being able to integrate all this data to automatize and control each step of the model inference procedure becomes crucial. On the one hand, it might help targeting true causal regulatory interactions, and replay probable regulatory cascades and mechanisms which had lead to a given experimental, observed, phenotype. On the other hand, this systematization of the inference allows higher replication, lower time and wet-lab costs.

We propose a fully automated Boolean network inference pipeline. In order to illustrate our method, we focus on a gene module named M30, which global gene expression has been shown to be anti-correlated to epileptic phenotypes (Delahaye-Duriez, Srivastava, et al., 2016). Thus, targeting one of the M30 genes might potentially be of interest for therapeutic purposes. Advanced graph analysis of a graph of regulatory interactions in this module might help identifying master regulators related to epilepsy factors ; give indications about the potential of molecules –for instance, whether they have an antiepiletic or proconvulsant effect ; and predict their effect on patients.

First, we describe a reproducible pipeline to identify the network associated with M30. It relies on perturbation experiments of interest from several public databases, and on the integration of supplementary biological information to further constraint the network inference procedure. Second, we will suggest a method using influence maximization to find a set of maximally perturbating set of genes. In particular, we exploit this to identify master regulator genes, at the top of the regulatory hierarchy. These genes might be potential therapeutic candidates, and might shed a light on the mechanisms of the considered disease. Last, we propose a method which directly uses this inferred model, in order to predict and to compare *in silico* the effect of potential drug candidates on patient phenotypes.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 1

Introduction to Boolean networks

The power of systems biology and network-based approaches comes from the analysis of multiple genes in functionally enriched pathway, as opposed to traditional single gene and single target approaches. In an effort to encode the effects of a molecule on the transcriptional activity of genes, we consider gene regulatory networks. These graphs connect genes according to their regulatory interactions. An arrow is drawn from one gene to another if the former regulates the expression level of the latter, for instance by preventing or encouraging its transcription. Such a model describes a dynamical system, which can be used for predicting transcriptional profiles. Integration of system biology-related methods to drug development has been implemented for epilepsy in Delahaye-Duriez, Srivastava, et al. (2016), and has allowed the identification of a gene module which global expression is highly anti-correlated to epileptic phenotypes (Delahaye-Duriez, Srivastava, et al., 2016). Then, a whole set of genes can be targeted for treatment, instead of screening drugs against a single relevant target. Prior literature emphasized on the importance of small-effect gene in a system biology model, as their belonging to highly interconnected gene regulatory networks implies that any slight perturbation on these genes might impact significantly "core" disease genes (Dugger, Platt, and Goldstein, 2018). A large and active literature (Ahmed, Roy, and Kalita, 2018) has emerged about the formalization, the building and the validation of such gene regulatory networks, along with the identification of gene modules highly correlated with pathological phenotype. As gene regulatory networks are assumed to mirror gene activity with regard to other genes' expression, building them usually require (time-series) expression data. These data can be extracted from databases recording measurements of expression after genewise perturbations – for instance, in Young, Yeung, and Raftery (2016), which relies

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Type	Example	Implication
protein-gene (transcriptional)	In Escherichia Coli, when the concentration in solute is too high, protein OmpR activates the transcription of gene ompC	Gene ompC codes for protein OmpC which blocks cell pores and does not allow solute to enter the cell : solute concentration decreases
protein-protein (post-transcriptional)	In the case of an oncogenic insult, protein USP42 prevents degradation of protein p53	Tumor suppressor protein p53 is active during oncogenic threat

Table 1.1: **Examples of regulatory interactions.** Two examples of regulatory interactions of various types, and their implications (Aiba et al., 1989; Hock et al., 2011).

on knock down gene expression measurements collected in the LINCS L1000 database (Subramanian et al., 2017).

In this thesis, we focus on a specific type of gene regulatory networks, called Boolean networks. This class of networks allows to model in a qualitative way the effects of the interconnected gene regulations. Moreover, what makes them particularly attractive is the ability to easily simulate transcriptional profiles that result from the perturbation of one or several genes in the network. This property is key to our solution to tackle drug repurposing while taking into account the effect of regulatory cascades (Bolouri and Davidson, 2003).

1.1 Modelling gene expression with regulatory networks

As illustrated in Figure 2 in introduction, the main connection between our genomic information –encoded into our DNA and shared by almost all cells in our body– and the state of our organism at some point in time is described by the "central dogma" (Crick, 1958) : DNA material (that is a double helix comprising of nucleotides) is transcribed into immature messenger RNA (mRNA, which is a single strand of nucleotides) within the cell nucleus. Immature mRNA matures, and then exits the nucleus to enter the cytoplasm. At this point, if sequencing occurs, then the mRNAs are retrieved and aligned against a reference genome in order to identify their matching gene transcripts. The number of successfully aligned mRNAs corresponds to the expression level of the considered gene. Later in the cytoplasm, mature

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

mRNA is translated into aminoacid chains (*i.e.*, proteins) via ribosomes, which "read" mRNA by triplets of nucleotides. Each triplet of nucleotides uniquely encodes for an aminoacid (or for a signal that the current aminoacid chain should start or end). However, several combinations of nucleotides can lead to the same aminoacid. Although this dogma is now often challenged, for instance by the existence of reverse transcription (Baltimore, 1970), where RNA-like molecules can be turned into genomic material, this dogma interestingly highlights the link between our more-or-less static genetic information, and the adaptation of our organism to its environment.

If there were no regulatory interplay on gene expression, *i.e.*, no regulatory interaction of a given gene on the expression level of another gene, then there would be no to little adaptability to the environment. Indeed, almost every cell possess the same genomic information, yet there are different cell behaviors, implying a change in expression, with hundreds of existing different cell types in humans. Table 1.1 gives examples of such interactions, and their implications in terms of cell behavior and structure. Such regulatory relationships between genes, proteins, or other molecules produced from DNA can be summed up into a graph, that is called regulatory network. This model allows the identification of upstream regulators of specific proteins, that have an impact on the cellular property or properties of interest, and their regulatory patterns, that is, the dynamics of expression levels in this network. In Zhao, Sun, and Zhao (2012), this model is used to report a competiting co-regulatory mechanism between tumor-suppressor genes and oncogenes –*i.e.*, genes that promote cancer growth. Tumor-suppressor genes, which are primarily involved in DNA repair and apoptosis, target genes that are functionally linked to the response to hormone stimuli ; whereas the downstream targets of oncogenes, which regulate response to hormone stimuli, act upon apoptosis. This model might cast a light upon patient response to hormonotherapy. In other domains, such as evolutionary development (evo-devo), networks help understand the embryonic development, that is, the process that leads an embryo to a fully-fledged organism. For instance, Dunn, Martello, et al. (2014); Dunn, Li, et al. (2019) successfully model the reprogramming of differentiated (*i.e.*, "specialized") cells into pluripotent cells in mice by the Yamanaka transcription factors (Takahashi and Yamanaka, 2006). Transcription factors are genes that bound to the DNA to modulate the transcription of some sequences, and often play a key role in regulatory networks.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

1.2 Overview of gene regulatory networks

A gene regulatory network is a model of the regulatory interactions which modulate the expression level of a set of genes. Those regulatory interactions may intervene at transcription, or by a subsequent mechanism. As a general rule, this network is represented as a diagram which can be decomposed into three main components, respectively called "Inputs", "Regulatory network" and "Outputs". The second component "Regulatory network" connects the first part of the network "Inputs" to the "Outputs" component. It is a graph which nodes are genes or proteins, or any actively regulating compound, and edges (if the graph is undirected) or arrows (otherwise) between the nodes are regulatory interactions of gene expression. This decomposition is illustrated in Figure 1.1 for one of the most well-known examples of gene regulation, that is, the adaptation of Escherichia Coli to an environment lacking in glucose (Pardee, Jacob, and Monod, 1959). The inputs to a gene regulatory network usually refer to the "actable" points of the network, to which an external perturbation can be applied, whereas outputs represent interesting biological states (called "phenotypes") resulting from the gene regulatory interplay. Phenotypes might be cell behaviors (*e.g.*, apoptosis, that is, cell death) or structures (*e.g.*, differenciation of a pluripotent stem cell into a more specialized cell line).

Several methods have been investigated to implement these networks in practice, especially the "Regulatory network" component, which correspond to different venues of research and can achieve different goals. Karlebach and Shamir (2008) classify these frameworks into three main groups, in increasing order of complexity.

Discrete models. First, they consider discrete models of gene expression, for instance Boolean networks (Kauffman, 1969; Thomas, 1973) which will be discussed at length in the next section. The principle of discrete models is to consider discrete values of expression and a finite set of types of regulatory interactions. This simple description –to the price of some loss of information– has already been useful to show the relationship between the structure of the network, and the presence of steady stable attractor configurations, in which the network remain under no external perturbations (Thomas, 1978).

Deterministic continuous models. Second, continuous models of gene expression, where values which amount to expression levels are updated for each species (gene, protein, ...) of interest, are considered (Goodwin et al., 1963). The idea is to find, for each species, a mathematical function which

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

connects the expression level of regulator genes to the decay and synthesis rate for the considered species. The resulting set of functions yield a system of coupled differential equations which can be solved, for instance in order to identify basins of attraction. This system represent chemical kinetics between species, and can be formally described and executed using the κ-language (Danos and Laneve, 2004) for instance. However, the predicting power of this framework greatly rely on accurate estimations of the kinetic constants in the chemical equations, which can be difficult to obtain in the first place.

Stochastic continuous models. Last, a regulatory network can also be viewed as a global system, which comprises of several type of small molecules. Each interaction between two types of molecules occurs with an associated probability (McAdams and Arkin, 1997), which can also be seen as chemical reaction rates. Then, given the probabilities associated with the interactions, and the initial number of each molecular type, the Gillespie algorithm (Gillespie, 1977) can compute the probability of a phenotype, along with the associated number of each species. This framework shows an accurate description of the regulatory dynamics, but can be hard to analyze (Karlebach and Shamir, 2008).

Because of its straightforward qualitative interpretation, and its relatively low computational cost, my PhD revolved around discrete models of gene regulation, and more specifically, around Boolean networks.

1.3 Introduction of dynamics with Boolean networks

We now focus on Boolean networks to model gene regulation networks. In the graph associated with a Boolean network, called "interaction graph", only two types of interactions are considered, either activatory or inhibitory. [1] The type of an interaction is called its sign. Interactions are also directed, meaning that in the two genes involved in the interaction, one is set as the regulator, which causes the regulation, and the other is the regulated gene.

[1]Note that, in this thesis, we will not consider non monotonous regulatory interactions, which can be either inhibiting or activatory given the local context. This assumption is based on the fact that, in most models, it has been shown that non-monotonous regulations could be successfully modelled via post-transcriptional mechanisms, such as the involvement of non-coding DNA sequences, as mentioned in Réda and Wilczyński (2020). Dynamics encoded by multi-level formalisms (*i.e.*, with more than two expression states) can also be encompassed by carefully selecting the type of dynamics in the network (Paulevé et al., 2020).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

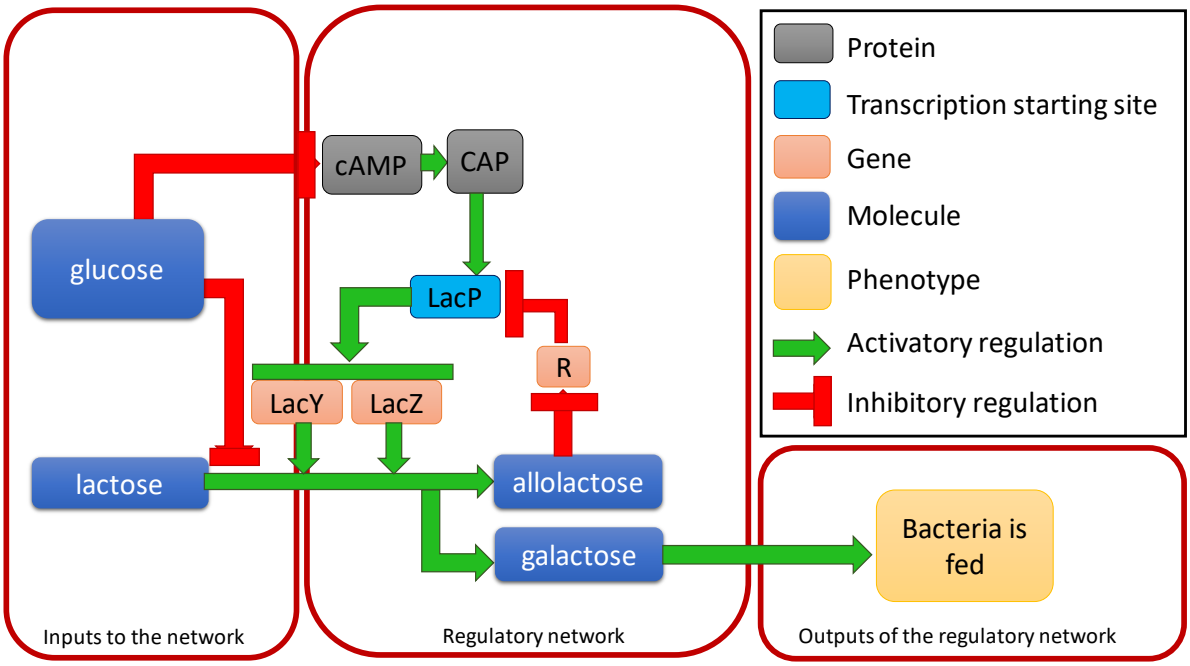

Figure 1.1: **Lac operon regulation as a gene regulatory network.** Decomposition of the well-known *lac operon* regulatory network in the Escherichia Coli (E. Coli) bacteria from Pardee, Jacob, and Monod (1959), based on the regulatory network in Santillán and Mackey (2008). It shows the adaptation of the bacteria to the type of nutrients present in its environment. E. Coli preferably feeds on glucose, but when it is missing from its environment, the inhibition on the transcription starting site LacP through repressor R, that blocks the transcription of genes LacY and LacZ, do not hold anymore, which allows the bacteria to feed on lactose. Gene LacY is involved in the transport of external lactose to the bacteria, and LacZ is involved in the transformation of lactose (and allolactose) into galactose. In the presence of glucose, the production of protein cAMP and CAP is inhibited, which further blocks the transcription of LacY and LacZ due to the presence of repressor R. To allow a readable figure, I omitted an activatory link between glucose and the output node, and the reaction between allolactose and LacZ which results in the production of galactose.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

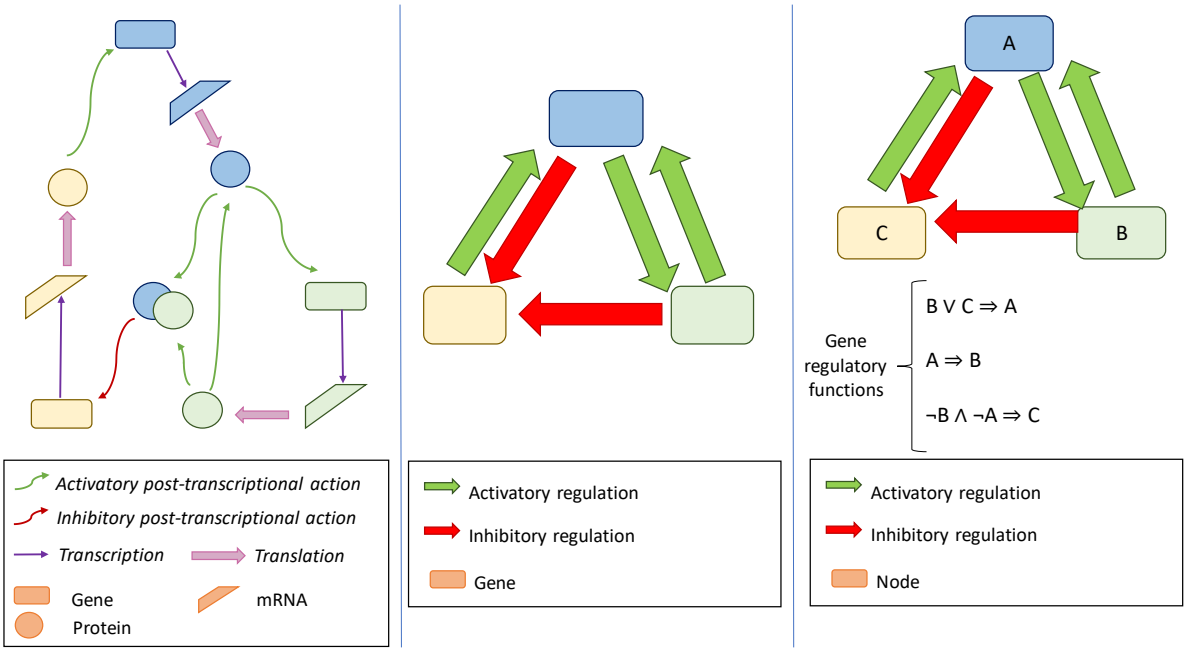

Figure 1.2: **Gene regulatory networks and Boolean networks.** Going from a set of regulatory interactions between genes, to a Boolean network with both the interaction graph and the set of regulatory functions for each node in the network. **Left plot** : Regulatory interactions (of different types) are shown between three genes (nodes of rectangular shape). **Center plot** : The gene regulation network between those three genes. **Right plot** : One possible Boolean model associated with the gene regulatory network. There might be potentially several sets of regulatory functions which can match the regulatory network, in the absence of additional information about gene expression.

The expression level of a gene has two states : either it is equal to 0, which stands for absent to low expression, or it is equal to 1, which corresponds to a state of high expression. These states belong to the Boolean domain, that is, $\{0, 1\}$, hence the name of the framework. A so-called "regulatory function" is assigned to each node in the regulatory network. This function describes the behavior of the expression state of a given gene at a given time point, depending on the expression states of its *direct* regulators, that is, the predecessor nodes in the network. This function is encoded as a Boolean (logical) function of the variables associated with gene expression states, which can be evaluated to get the current gene expression state. Figure 1.2 illustrates how to go from a set of regulatory interactions to a Boolean network, on a dummy example with three genes. From now on, we will often abuse notation by indifferently using "genes" and "variables" to designate gene expression states or nodes in the network. In order to determine how the configuration of the network –*i.e.*, the set of expression states for every node in the network– changes at a given time point, one has to define the dynamics of the system. That is, the type of update

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

between network configurations, in addition to the interaction graph and the regulatory functions. [2] Several update types have been proposed, listed in Chatain, Haar, and Paulevé (2018) for example. The two most simple types of updates are the *synchronous* and the *asynchronous* updates. The synchronous update means that, at each update step, all the gene regulatory functions are simultaneously evaluated, based the gene expression states at the current time step ; whereas in the asynchronous update step, a single gene regulatory function is selected at time for evaluation. Different update types mean different dynamics. In particular, depending on the chosen update type, the network can reach different configurations. Once the update type is set, a state-transition graph (STG) can be built, connecting network configurations to each other with directed links, such that, if there exists an arrow going from a network configuration A to another one B, then the network can go from the initial configuration A to final configuration B in a single update step. A state-transition graph thus depends both on the set of gene regulatory functions and the definition of the update step. The STG is particularly interesting because steady attractors –that is, configurations that loop upon themselves in the STG– and unsteady attractor states –cycles in the STG– can be directly read from this diagram. The attractor states in a Boolean network –and in gene regulatory networks in general– usually correspond to interesting biological phenotypes (Bloomingdale et al., 2018; Wery et al., 2019). Examples of STGs, with different types of attractor states, are shown in Figure 1.3.

A Boolean network can be straightforwardly built from a known static gene regulatory network, or maps of molecular interactions, by automatically assigning gene regulatory functions based on the activators, resp. the inhibitors, of each gene, for instance using CaSQ (Aghamiri et al., 2020). In order to infer the Boolean network "from scratch", that is, from a set of genes, one must define both an interaction graph and an set of gene regulatory functions which match experimental observations *under the selected update type*. One way to find the appropriate Boolean network requires to know a set of possible genepairwise interactions –sometimes known as the "prior knowledge network" (Ostrowski et al., 2016; Vaginay, Boukhobza, and Smail-Tabbone, 2021)– and a set of time-series (binarized) transcriptional profiles which (hopefully) exhibits enough of the true dynamics of the biological system. Then gene regulatory functions are tuned until all the experimental constraints are satisfied by the model. However, there are two main hurdles in this inference problem. First, the binarization step to go from transcriptional profiles of continuous gene expression to binary profiles –with

[2] When it is obvious, the term "state" can also be used to designate network configurations instead of gene expression states.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

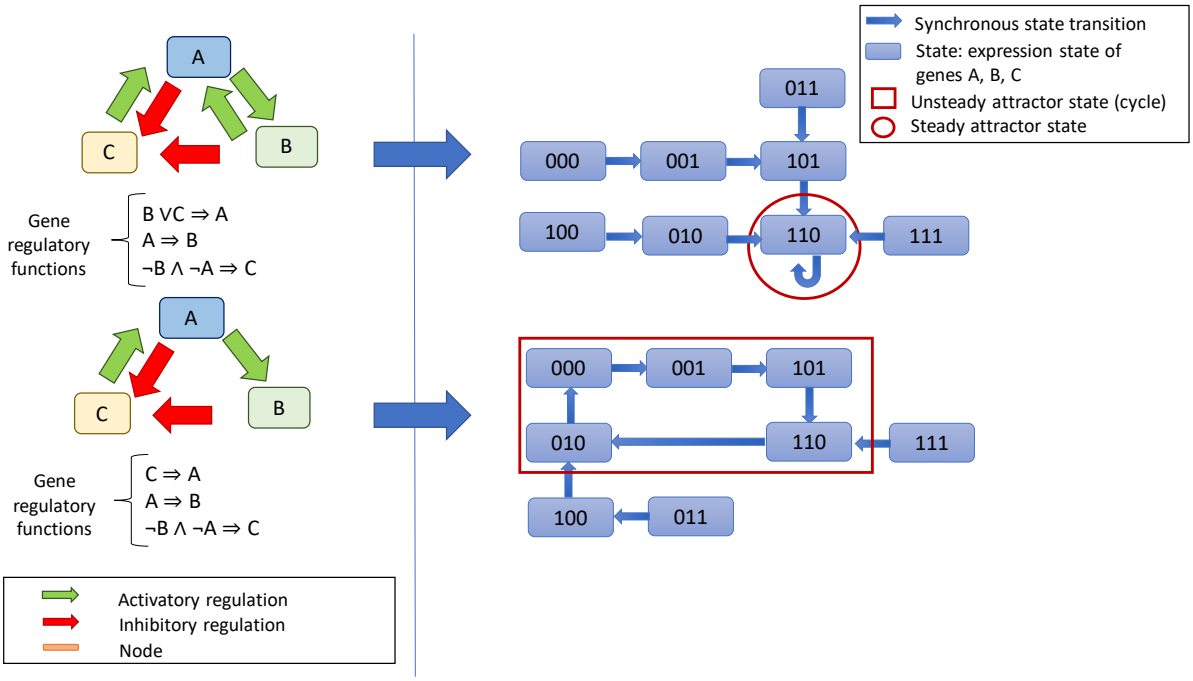

Figure 1.3: **Dynamics in a Boolean network.** Dynamics in a Boolean network. **Left plot** : Two instances of Boolean networks, with the same interaction graph, but different sets of regulatory functions. **Right plot** : Corresponding state-transition graphs (STG), assuming synchronous updates, that is, simultaneous update of all gene regulatory functions.

potentially genes which have no known binary state, if their expression is not high or low enough– incurs an unavoidable loss of information, and should be performed carefully. Binary profiles might also been built using prior knowledge on the expected behavior of the system, for instance, the phenotypes that should be observed under specific perturbations, but this requires supplementary access to custom wet-lab experiments or to the literature, at the risk of propagating scientific bias. Second, the problem of infering a network, which satisfies both the topological (related to interactions) and the dynamical (related to observations) constraints, is usually underdetermined. As the number of genes in the network grows, so does the amount of experimental data needed in order to pinpoint a single model which can satisfy all the constraints. This fact makes unbiased model building difficult when done manually. Some tools allow the automation of the tuning step, such as Re:In (Dunn, Martello, et al., 2014) or BoNeSiS (Chevalier et al., 2019). These methods are based on enumerating putative network solutions, and testing whether the network satisfies all constraints using a solver of Boolean equations (SAT solver).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

1.4 Boolean networks in drug repurposing

Now, using the same tools as in Boolean network inference, a Boolean network can be predict (binary) transcriptional profiles under the perturbation of one or several genes.

The perturbation of a given gene, either by knockout –*i.e.*, the expression of the gene is totally inhibited– or by overexpression –that is, the gene is forcibly expressed– is defined by setting the corresponding gene variable to either 0 or 1. Then, under these new Boolean constraints, and starting from a fixed initial state, update steps are iteratively executed, until an attractor state is reached, or until a maximum number of update steps is performed (in order to get results in finite time).

This maximum number of steps, that is, the length of the (configuration) trajectory, can be chosen large enough to end up in an attractor state, if any exists. However, the network might end up in an unsteady attractor state, that is, a cycle of configurations between which the network indefinitely oscillates in the absence of supplementary perturbations. This can be checked by considering the whole trajectory of configurations from the initial to the last reached configuration, and matching the corresponding cycle –if it exists– to one of those present in the STG. Nonetheless, note that computing the whole STG, especially as the number of considered nodes is large, becomes quickly non computationally tractable. Fortunately, model checking tools such as Z34Bio (Yordanov et al., 2013) –which uses SAT solvers– or BioModelAnalyzer (BMA) (Benque et al., 2012), can prove the stabilization of the system or that the final configurations belong to a cycle.

This property of Boolean networks will be useful in the drug repurposing method proposed in this thesis. Indeed, it allows to explicitly model the transcriptomic cascades, and to compute the "stabilized" qualitative biological profile the regulatory network ends up in after a single gene or drug-induced perturbation.

However, as previously mentioned, the accurate building of a Boolean network requires a rather large amount of data, which is often manually retrieved from the literature (Collombet et al., 2017). When modelling a network on a large number of genes, the process is often incremental and based on iterative improvements of the model (Niarakis and Helikar, 2021). At best, it requires at least collecting a set of putative regulatory interactions, and a set of time-series experiments of good quality, which reports the initial configurations, the perturbations (if needed), and the final observed configurations ; and then feeding these inputs to a Boolean

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

network inference algorithm (Chevalier et al., 2019; Dunn and Yordanov, 2019; Réda and Wilczyński, 2020). This approach might show its limits when facing hundreds of genes in a relatively poorly studied cell line.

This led me to first design an automated, reproducible method for building Boolean networks, as described in the next chapter.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 2

Prioritization of candidate genes through Boolean networks

This chapter focuses on *in silico* detection of master regulator genes, which is a popular approach to speed up drug development. Master regulator genes might be directly related to the onset of the disease, or may act on one pathway which counteracts the associated symptoms. Then, one could perhaps screen drugs to select chemical compounds targeting these genes. In prior works, the detection of these candidates was performed through the identification of the regulatory interactions between genes of interest for the disease. Indeed, system biology approaches have proven a useful tool to integrate transcriptomic data and predict transcriptional profiles under gene perturbations. In particular, Boolean networks, where gene expression levels are reduced to two values, are a flexible framework for qualitatively modelling gene expression. However, for rare diseases, building such a regulatory model can become a tedious and time-consuming task.

In this work, we show how to build, in a reproducible fashion, a Boolean network related to a subset of interesting genes for the disease, using publicly available data. Then, we describe a method to identify master regulatory genes, that is, genes which have an impact on the dynamics of the gene regulation in a specific disease-related transcriptional context. As a proof-of-concept, we focus on a subset of genes related to various epileptic phenotypes, in order to find novel master regulator genes associated with epilepsy. These genes might help casting a light on the causes of refractory epilepsy, which are epilepsies which cannot be managed by conventional antiseizure medication. We show that our method for the identification of master regulatory genes is consistent with network controllability measures, while targeting genes which are significantly enriched for epilepsy-related terms. We show that *in silico* perturbation of these candidates can reproduce

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

epileptic phenotypes. Our pipeline allows for systematic and automated synthesis of a Boolean network and identification of putative drug targets. This work was accepted at the 20th conference on Computational Methods in Systems Biology (CMSB 2022) (Réda and Delahaye-Duriez, 2022). [1]

2.1 Related work

We propose a novel generic method for the detection of master regulator genes, which can be applied to any disease, and relies on a dynamic interplay between a gene regulatory network and gene expression data. We focus here, as a proof-of-concept, on an application to epilepsy.

Epilepsy actually encompasses various neurological diseases and syndromes, which can originate from brain injury or genetic background, that have in common a propensity to trigger chronic epileptic crises. Epileptic crises are characterized by a transitory abnormal neuron electric discharge, which might lead to unconsciousness, seizures, and/or body stiffness. Epilepsy is one of the most common neurological diseases worldwide, with around 50 million people living with this disease (World Health Organization (WHO), 2022). Moreover, more than 25% of epileptic patients are afflicted with drug-resistant epilepsy (González et al., 2015), also called refractory epilepsy. Symptoms in those patients could not be managed by at least two different antiepileptic therapies. This shows the limits of conventional antiepileptic medication, which are often molecules with antiseizure effects, and emphasizes the need to look for novel therapeutic candidates. Epilepsy-related genes are shown to be usually mainly expressed in a specific brain region, called hippocampus (Mirza, Appleton, et al., 2017). This region is also affected by morphological changes linked to neuronal discharges in some epileptic patients (Ogren et al., 2009). The exact relationship between lesions in the hippocampus and epilepsy-associated symptoms is still unclear, but might be related to the fact that hippocampus is one of the most excitable parts of the brain (Kuruba, Hattiangady, and Shetty, 2009). Several animal models of epilepsy exist, including a mouse model where injection of pilocarpine induce symptoms similar to temporal lobe epilepsy (Srivastava, Bagnati, et al., 2017), or another involving sodium channels, which play a role to convey electric potentials. For instance, a genetically modified animal with invalidation of gene *Scn1a*, which code for sodium channels, models a severe form of epilepsy called Dravet syndrome (Kalume et al., 2013).

[1] Related code is located at `https://github.com/clreda/PrioritizationMasterRegulators`.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

In prior works, the identification of master regulators in gene networks has been a powerful method to detect novel genes of interest for a given disease. Master regulator genes are genes which might have a large, global influence on the expression of a group of genes in a specific pathway. For instance, *SESTRIN3* (Johnson et al., 2015) and *CSF1R* (Srivastava, van Eyll, et al., 2018) were prioritised as candidate antiepileptic drug targets using different systems-biology approaches dedicated to identifying master regulators of epilepsy-associated networks of gene expression. Such genes might be forcibly expressed or knocked out –*i.e.*, no more expressed– by molecules, which might be interesting antiepileptic drug target candidates. Other approaches exploit the location of a given gene inside a gene regulatory network. It is assumed that the more central it is, the most regulatory it should be. Other approaches use the concept of "network controllability" (Liu, Slotine, and Barabási, 2011) which is loosely related to the centrality of the gene in the network, as it estimates the number of downstream targets possibly affected by a perturbation of this gene. Yet, most of the cited approaches for the detection of interesting regulatory genes only leverage topological knowledge about the network, without considering the actual dynamics of the regulatory system. In order to find master regulator genes related to rheumatoid arthritis based on expression data, Zerrouk et al. (2020) have considered an approach which combines a transcription factor (TF) co-regulatory network and gene expression in fibroblast-like synoviocytes in patients afflicted with rheumatoid arthritis. [2]TF influence in these samples was assessed using the tool CoRegNet (Nicolle, Radvanyi, and Elati, 2015), which computes a score of influence per TF on a set of transcriptional profiles. This score is defined for TF t, that activates a set $\mathcal{A}_t$ of genes and inhibits a set $\mathcal{I}_t$ of genes, and a matrix of transcriptional profiles M [3] as follows

$$\text{Influence}(t) := \frac{\left(\frac{1}{|\mathcal{A}_t|}\sum_{a\in\mathcal{A}_t} M[a,:]\right) - \left(\frac{1}{|\mathcal{I}_t|}\sum_{i\in\mathcal{I}_t} M[i,:]\right)}{\sqrt{(s_{\mathcal{A}_t})^2/|\mathcal{A}_t| + (s_{\mathcal{I}_t})^2/|\mathcal{I}_t|}} \tag{2.1}$$

where $s_{\mathcal{A}_t}$ (resp., $s_{\mathcal{I}_t}$) is the standard deviation in gene expression of all genes in $\mathcal{A}_t$ (resp., $\mathcal{I}_t$) across all profiles in M. However, such a computation does not take into account downstream transcriptional cascades beyond genes directly regulated by the TF. Yet these regulatory cascades might allow to identify off-target genes –*i.e.*, genes subject to non specific and involuntary changes– which might lead to serious side effects (Huang et al., 2019). These regulatory cascades can be modelled through a dynamical

[2]Fibroblast-like synoviocytes are a cellular subtype which is present in conjunctive tissues.

[3]Rows are genes, and columns are samples.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

gene regulatory network. In that case, setting up a reproducible pipeline for building this regulatory network –especially for the retrieval of raw biological data– is crucial to ensure replicability in drug development research. Finally, this gene regulatory network can be represented by a Boolean network. This type of model comprises of discrete, qualitative, gene regulatory interactions, which promote an increased interpretability of the predicted changes and less expensive computations compared to continuous models of gene expression.

To this aim, we developed a fully automated pipeline to infer a Boolean network which models the regulatory interactions in a subset of genes. Our method is based on perturbation experiments of genes that may participate in our network, and on the integration of supplementary biological information to further constraint our inference procedure. In our work, we exploit the database LINCS L1000, which provides a large number of gene expression profiles for various combinations of cell types, and genetic perturbations (Subramanian et al., 2017).

In our application to epilepsy, we focus on a module of 320 genes, called M30, which global expression was shown to be anticorrelated with various epileptic profiles and with the severity of epilepsy (Delahaye-Duriez, Srivastava, et al., 2016). Using the Boolean network selected by this method, we rank genes in M30 according to their *in silico* estimated ability to permanently modify the expression of the whole network, and prioritized top genes. In favor of the important role in epilepsy-related biological processes of these genes, this prioritised set of candidate genes is significantly enriched in terms related to epilepsy and neurodevelopmental issues when compared to the whole M30 module.

2.2 Reproductible inference of a cell-line specific Boolean network

This part of our work aims at designing a method which, given a set of genes of interest, is able to retrieve a Boolean network that allows the prediction of transcriptomic profiles. We consider the formalism of Boolean networks, introduced in Kauffman (1969); Thomas (1973), which are popular models to describe gene-gene expression regulations as gene regulatory networks.

Boolean networks. A Boolean network is characterized: first, by a graph –*i.e.*, the network– which connects genes by their regulatory interactions. Such connections are enriched with the direction of the interaction,

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

which distinguishes between regulator and regulated genes, and with the sign of this interaction, that is, whether the regulator inhibits or activates the expression of its target. Second, by the dynamics of the regulatory system : a logical function, called "gene regulatory function", is assigned to each gene. Regulatory functions are logical Boolean formulas where variables correspond to the expression of genes. The expression of a given variable is set to 1 if the associated gene is expressed, otherwise 0. For a given gene, a logical formula contains in its premise variables associated with *direct* regulators of the considered gene –*i.e.*, direct predecessors of this gene in the network– and, in its conclusion, the variable related to this gene. Then, given the expression states of the regulators, one can obtain the expression state of the considered gene by evaluating the corresponding formula. The vector collecting all the binary gene expression states at a given time point is called network state, or sometimes network configuration. From this model, one can build a *state-transition diagram*, which is a graph where an edge goes from a given network state A to another network state B if and only if one can reach state B from state A in a single update step. One can read from this diagram attractor states, that is, self-looping nodes, which are defined as steady stable network configurations. That is, the application of the update step to this configuration will lead to itself. Attractor states are interesting because they are commonly related to observable biological phenotypes (Bloomingdale et al., 2018; Wery et al., 2019). This diagram also displays cycles of configurations, which correspond to unsteady stable configurations ; the application of the update step makes the system oscillate between a set of configurations in a cyclic way. A state-transition diagram is associated with a given model *and* a type of update. Several types have been suggested in the literature (Chatain, Haar, and Paulevé, 2018), the most well-known ones being the synchronous and the asynchronous updates. In the former, all regulatory functions are evaluated in a single step, whereas in the asynchronous update, only one regulatory function is evaluated at one update step. Recently, Paulevé et al. (2020) have introduced new dynamics for Boolean networks, which was shown to be flexible enough to represent (a)synchronous dynamics as well as multi-level formalisms, that is, beyond boolean values for gene expression.

Building a Boolean network from scratch. Our work focused on combining several trusted public data sources, and published methods for the design of an end-to-end pipeline for the synthesis of a Boolean network, represented in Figure 2.1. This network models the regulatory dynamics on a subset of genes, in the absence of external perturbation, from a well-chosen cell line ; for instance, the regulations between M30 genes in brain cell lines for our application to epilepsy. Contrary to the contemporaneous work

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

of Montagud et al. (2022) applied to cancer, here we do not have any access to a generic model which could model any type of epilepsy to start with. Moreover, relying too much on prior epilepsy-oriented knowledge might lead us to find already known gene targets. The big picture of this pipeline comprises of the following three main steps in chronological order, respectively denoted (A), (B) and (C) in Figure 2.1 :

(A) Data collection. Step **(A)** represents the collection and filtering of information from public, large databases : measurements of transcriptomic data are retrieved from the LINCS L1000 database (Subramanian et al., 2017) using careful filtering and quality control measures ; known unsigned, undirected protein pairwise regulatory interactions involving genes in M30 are obtained from the STRING database (Szklarczyk et al., 2021).

(B) Data processing. Step **(B)** represents the processing of this information into appropriate inputs for the inference of Boolean networks. First, a set of binarized phenotypes is built, corresponding to profiles from single gene perturbations and their associated controls, from the LINCS L1000 transcriptomic data. Then, a signed network of admissible regulatory interactions is constructed from the STRING-extracted regulatory interactions, by filtering out and signing edges based on Spearman's ρ gene pairwise expression correlations computed on LINCS L1000 profiles.

(C) Network inference. Finally, step **(C)** is the inference of a set of Boolean networks which satisfy all the experimental and topological constraints given by the phenotypes and the signed network. The experimental constraints comprise of each knockout experiment, where the control phenotype is considered the initial condition, and the perturbed one the final configuration reached after stabilization of the system perturbed by the gene knockout or overexpression. A Boolean network solution should satisfy all of these time-series constraints by only considering a subset of admissible regulatory interactions, represented in the signed network. The final step of the procedure is the selection of an optimal Boolean network among these solutions, according to its topology. This final inferred network is selected through a desirability function maximization (Babichev et al., 2019) which aims at maximizing a parameter called *general topological parameter* (GTP), which depends on several network topology measures.

Details in the implementation are available in Appendix 9.1. Note that, although this issue did not have to be addressed for epilepsy, since the method is meant to be generic (*i.e.*, usable for any disease), which is why, if no gene set is provided, the method automatically retrieves genes from DisGeNet (Piñero et al., 2020) from the disease Concept ID (CID)

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

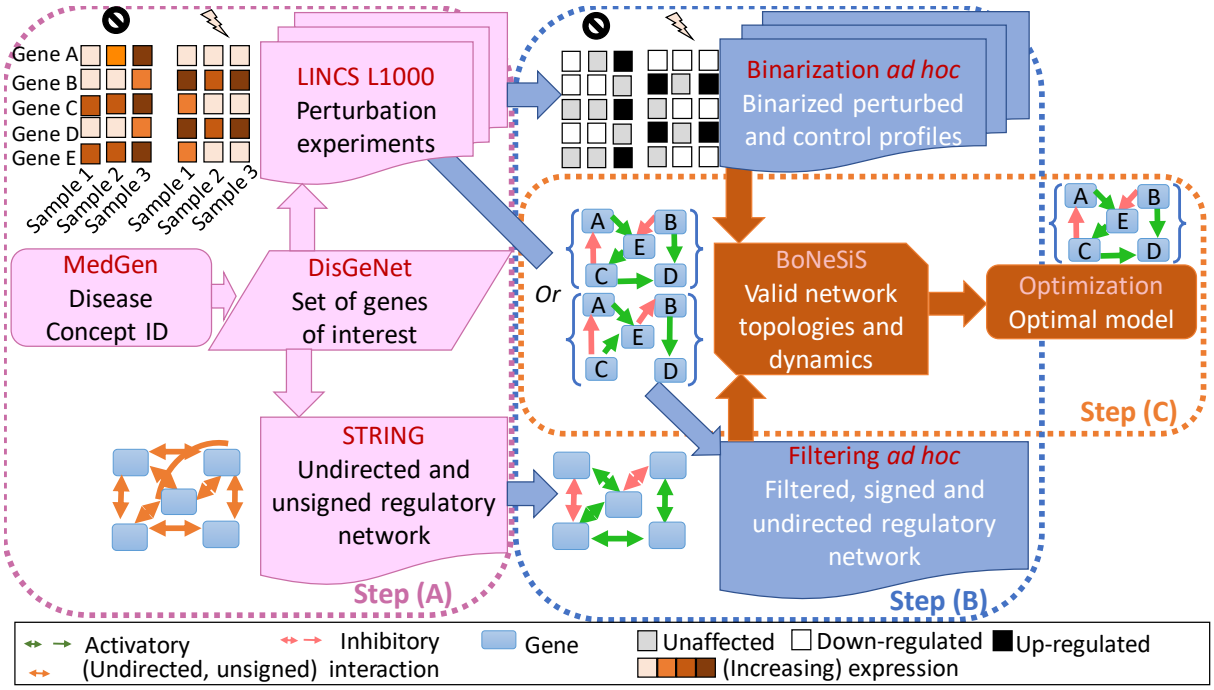

Figure 2.1: **Automated building of Boolean networks.** Overview of the pipeline for the automated building of Boolean networks. Blocks are colored according to the step they belong to: step (A) for the lightest blocks, step (B) for blocks at the center of the picture, and step (C) for the darkest blocks.

defined in PubMed (Doğan, Leaman, and Lu, 2014). Table 9.4 in Appendix reports the values used to filter out genes from the DisGeNet database. The single network obtained at the end of step (C) is a dynamical system which can predict the behavior of gene expression under one or several gene perturbations, by considering the stable states (attractors and cycles) reachable from a given initial state under these perturbations. We now describe on how we leveraged this network model to rank genes according to their regulatory influence on other genes.

2.3 Detection of master regulators in a specific disease-context

As mentioned in Section 2.1, when looking for therapeutic candidates, one might be interested in master regulators, that is, genes at the top of the gene regulation hierarchy. Change in expression in a master regulator gene generally induces a large change in downstream gene expression ; for instance, this master regulator gene encodes for a transcription factor which affects the transcription of other genes (Mattick, Taft, and Faulkner, 2010). In practice, it is frequently quantified using the node (outgoing) degree and by the detection of hub nodes in the network. Many measures

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

defining this "centrality" property for nodes in a graph exist, and can be computed using for instance Cytoscape (Shannon et al., 2003), modules NetworkAnalyzer (Assenov et al., 2008) and CytoCtrlAnalyser (Wu, Li, et al., 2018). For example, control centrality (Liu, Slotine, and Barabási, 2012) has been recently used to identify regulations between *NFATC4* and Type 2 diabetes-associated genes (Sharma et al., 2018). However, these measures only use the topological information in the network, whereas our network inference pipeline allows, along with the identification of regulatory relationships, the inference of gene regulatory functions, which encode how regulators influence the expression of their target genes. This is why we designed a method which uses this supplementary information for the detection of master regulators.

Gibbs and Shmulevich (2017) used a machine learning technique called "influence maximization" in order to identify key genes in their continuous model of the yeast regulatory network. In our work, we adapted influence maximization to generic Boolean networks. For long, online recommendation and advertising researchers have been interested in influence maximization (Kempe, Kleinberg, and Tardos, 2003), which aims at finding a subset of fixed size of nodes which influences most the remainder of the network. The most well-known use cases for influence maximization are in online sponsoring of influencers on social media (Leskovec, Adamic, and Huberman, 2007). We will describe below the general setting for the problem of influence maximization.

Influence maximization. Considering a (un)directed graph $\mathcal{G}$ on a set of nodes $\mathcal{V}$ connected through a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, the goal is to determine a *seed set* N of fixed size k –that is, a subset of k nodes which start propagating some quantity (*influence*) throughout the network from their direct neighbors– which maximizes the *influence spread*. This influence spread is the expected number of vertices ultimately (in)directly influenced by seed set N. k is selected by the user depending on the final application case –for instance, the number of influencers to sponsor on social media to increase the advertisement of a product. Then, we study the random variable $\mathbf{I}(N)$ which is the random number of vertices influenced by seed set N by the cascades of propagating influence.

In order to make this technique applicable to Boolean networks, we need to explicitly define the concept of influence on gene expression in this type of regulatory networks, that is called "spread process". This quantity is proportional to the influence that propagates along the edges of the network. We define influence in an iterative way ; first we consider a single gene and initial network state, then we proceed to define multi-gene influence, and

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

finally influence for a set of genes across a set of initial network states.

Genewise influence in a Boolean network. The most intuitive definition of influence (denoted in the remainder of the paper "spread value") $\mathrm{SV}_{\mathcal{B}}(\{n\},\{i\})$ of a given node n on the other nodes in a Boolean network, in a given initial state i, would be that any perturbation of this node would "greatly change" the attractor states reachable from state i, compared to the set of attractor states reachable from state i in the absence of any perturbation. We define this great change (*i.e.*, a positive spread value) by the fact that those two sets of attractors have an empty intersection. Let us denote $\mathcal{A}(i,P)$ the set of attractor states reachable from state i, under perturbations in set P. Set P contains pairs of gene names and their perturbation (either 0 for knockout, or 1 for overexpression). Let us also denote $\mathcal{O}$ the set of output genes, that we define here as the set of genes with a positive in-degree. Then, considering any similarity measure $\mathcal{S}$ between network states, this definition of spread value $\mathrm{SV}_{\mathcal{B}(\{n\},i)}$ for node n and initial state i is defined as :

$$\mathrm{SV}_{\mathcal{B}}(\{n\},\{i\}) = 1 - \max\left\{\mathcal{S}\left(a^1_{|\mathcal{O}}, a^2_{|\mathcal{O}}\right) : a^1, a^2 \in \mathcal{A}(i,\emptyset)\times \in \mathcal{A}(i,\{(n,\neg i[n])\})\right\} \ ,$$

where we restrict the attractor states a^1 and a^2 to the set of output genes $\mathcal{O}$ when computing their similarity.

The perturbation denoted by $(n,\neg i[n])$ means that gene n is perturbed in the opposite direction to its expression state $i[n]$ in i : for instance, if n is expressed in state i, then we consider knockouts of gene n. The restriction $a_{|\mathcal{O}}$ of any attractor state a to output genes in $\mathcal{O}$ is actually important in order to have consistent results when considering isolated nodes.

Note that, if n does not have a determined expression state in initial state i, we set the associated perturbation set to $\emptyset$. This implies that some genes with individual spread value equal to 0 can either have no true influence on the network, or have no determined expression state in initial state s, which means that they are not measured during the generation of transcriptomic data.

The value $\mathrm{SV}_{\mathcal{B}}(\{n\},\{i\})$ is equal to 0 if and only if $\mathcal{A}(i,\emptyset)\cap\mathcal{A}(i,\{(n,\neg i[n])\}) \neq \emptyset$ (that is, if there is any attractor state in common). But, note that, if $\mathcal{A}(i,\emptyset)\cap\mathcal{A}(i,\{(n,\neg i[n])\}) = \emptyset$, then $\mathrm{SV}_{\mathcal{B}}(\{n\},i)$ is not necessarily equal to 1, as reachable attractors might still be close to those obtained without any external perturbation.

Geneset influence in a Boolean network. When considering several nodes in set $\mathcal{N}$ instead of a single one n, that is, to assess the influence of

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

a subset of nodes all simultaneously perturbed, we consider :

$$\mathsf{SV}_{\mathcal{B}}(\mathcal{N}, \{i\}) = 1 - \max\left\{\mathcal{S}\left(a^1_{|\mathcal{O}}, a^2_{|\mathcal{O}}\right) : a^1, a^2 \in \mathcal{A}(i, \emptyset) \times \mathcal{A}(i, \{(n, \neg i[n]) : n \in \mathcal{N}\})\right\} \ .$$

Aggregation of values for several initial states. When considering several putative initial states in set $\mathcal{I}$ instead of a single one i, for a given gene set $\mathcal{N}$, we consider the geometric mean of their spread values for each initial state $i \in \mathcal{I}$:

$$\mathsf{SV}_{\mathcal{B}}(\mathcal{N}, \mathcal{I}) = \left(\prod_{i \in \mathcal{I}} (\mathsf{SV}_{\mathcal{B}}(\mathcal{N}, \{i\}) + 1)\right)^{1/|\mathcal{I}|} - 1 \ ,$$

where $|\mathcal{I}|$ is the number of considered initial states. Note that we need to correct for zeroes to order to avoid the collapse of this measure when one perturbation does not trigger a change in reachable attractors for one of the initial states, while keeping spread values between 0 and 1 for better interpretability.

Algorithm 1 Gene influence maximization. Greedy influence maximization algorithm for Boolean networks

---

Input: $\mathcal{B} = (V, E, F)$ a Boolean network on node set V with edges in E and regulatory functions F ; K the number of simultaneous perturbations on the network ; $\mathcal{I}$ set of initial Boolean states

---

Initialize $\mathcal{N} = \emptyset$, $k = 0$
repeat
  $k \leftarrow k + 1$
  # Adding to set $\mathcal{N}$ nodes that maximize the spread value
  $\mathcal{N} \leftarrow \mathcal{N} \cup N_k$

$$\text{where } N_k \leftarrow \arg\max_{n \in V \setminus \mathcal{N}} \mathsf{SV}_{\mathcal{B}}(\mathcal{N} \cup \{n\}, \mathcal{I})$$

  # Ensuring submodularity
until $k = K$ or the following condition holds

$$\max_{n \in V \setminus \mathcal{N}} \mathsf{SV}_{\mathcal{B}}(\mathcal{N} \cup \{n\}, \mathcal{I}) \leq \mathsf{SV}_{\mathcal{B}}(\mathcal{N}, \mathcal{I})$$

Output: $\mathcal{N}$ and observed spread values

Matching to the general setting of influence maximization. The graph that describes the connections between nodes (genes) is the interaction graph of the considered Boolean network $\mathcal{B}$. Random variable $I(N)$ is no longer an integer (the number of influenced nodes) but directly the minimum frequency of change in binary expression across all downstream genes in reachable attractor states due to the perturbation(s). Note that computing $\mathsf{SV}_{\mathcal{B}}(\cdot, \cdot)$ in practice runs several stochastic trajectories to compute possibly

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

reachable attractor states.

Once the concept of influence is defined, we propose an algorithm which greedily builds the set of nodes which influence most the remainder of the network.

Influence maximization algorithm on Boolean networks. We describe how to use this spread value to identify a subset of master regulators in the network. Current literature on influence maximization (Perrault et al., 2020), which is NP-hard, relies on the fact that the spread value function is submodular : roughly, as the considered subset increases, the difference in the value of this function due to adding another single element to the subset decreases. However, no such property can be assessed for the definition of spread value defined in the previous paragraph.

Then, we slightly adapted the greedy algorithm designed by Kempe, Kleinberg, and Tardos (2003), which determines the set of nodes of minimal size K which are the most influent, where K is a predefined fixed value. Algorithm 1 goes as follows: starting from an empty set of nodes $\mathcal{N}_0$, a fixed set of initial states $\mathcal{I}$, at each step $k \in \{1, 2, \ldots, K\}$, the algorithm selects the node $n \notin \mathcal{N}_{k-1}$ which maximizes spread value $\mathrm{SV}_{\mathcal{B}}(\mathcal{N}_{k-1} \cup \{n\}, \mathcal{I})$ and computes the set $\mathcal{N}_k = \{n\} \cup \mathcal{N}_{k-1}$, until step $k = K$, or until the first step k when the spread value $\mathrm{SV}_{\mathcal{B}}(\mathcal{N}_{k-1}, \mathcal{I})$ is no longer increasing, that is,

$$\max\{\mathrm{SV}_{\mathcal{B}}(\mathcal{N}_{k-1} \cup \{n\}, \mathcal{I}) : n \notin \mathcal{N}_{k-1}\} \leq \mathrm{SV}_{\mathcal{B}}(\mathcal{N}_{k-1}, \mathcal{I}) \ , \tag{2.2}$$

which is necessarily to compensate for the fact that the function might not be submodular. If, at a given step k, several nodes maximize the spread value, they are all added to set $\mathcal{N}_k$. The iteratively built set $\mathcal{N}_K$ is then the set of possible K-sized gene subsets to simultaneously perturb on the network, such that the set of attractors reachable from initial set $\mathcal{I}$ is greatly modified. In this work, we selected $K = 1$, that is, we looked at individual contributions of genes to the changes, and we ranked gene n according to its spread value $\mathrm{SV}_{\mathcal{B}}(\{n\}, \mathcal{I})$.

Set of initial network states ($\mathcal{I}$). We consider transcriptomic profiles from human hippocampi afflicted with temporal lobe epilepsy (TLE) in Mirza, Appleton, et al. (2017) for the initial states, such that genes are ranked according to their influence in an epileptic context. Temporal lobe epilepsy is one of the most common forms of partial epilepsy, where seizures affect one part of the brain, and is often associated with cases of refractory epilepsy that cannot be surgically treated (Han et al., 2014). Details about the implementation and initial states are available in Appendix 9.4. 207 genes out of 232 genes both from the M30 module and present in the network are mapped

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

to expression levels in these states, which means that for these genes, we are sure that any spread value equal to 0 for any of these genes truly means that the gene has no influence over the remainder of the network.

Similarity between attractor states ($\mathcal{S}$). The definition of influence relies on a similarity function $\mathcal{S}$ between two network states, that we left to be defined in the concept of influence. In the implementation of the method, we used a distance function which is relevant for binary vectors. The main difference with other binary distances is that we wanted to compute the differences in the presence of 1's, but also of 0's in the vectors, which prevented us from directly using Jaccard's score. Based on previous surveys of the state-of-the-art on binary distances (Choi, Cha, and Tappert, 2010), we decided to use a "normalized" ℓ_1-norm distance. That is, if a^1 and a^2 are the two binary vectors to compare (of size d), then the resulting similarity $\mathcal{S}(a^1, a^2)$ between a^1 and a^2 is :

$$\mathcal{S}(a^1, a^2) = 1 - \frac{1}{d} \sum_{i=1}^{d} |a^1[i] - a^2[i]| \ . \tag{2.3}$$

This expression is exactly the percentage of row-wise equal coefficients in a^1 and a^2, and yields 1 when $a^1 = a^2$, and 0 for $a^2 = (a^1 + 1) \equiv [2]$ (modulo 2). It penalizes in a symmetric way differences in 1's and 0's.

2.4 Results

Networks obtained from the inference procedure

We discuss the networks resulting from the inference procedure described in Section 2.2.

Inferred network. The final network obtained at the end of step (C) is shown in Figure 2.2. In this figure, nodes are colored by their degree ; the darker the color, the higher the degree. Edges in Figure 2.2 are colored according to their source of evidence as reported by the STRING database (Szklarczyk et al., 2021). One can notice that there are a lot of undirect gene-to-gene regulatory interactions in this network. This actually is not very surprising, since few gene pairwise interactions are experimentally tested compared to all possibly existing interactions. Moreover, our model does not aim at taking into account exclusively transcriptomic interactions, but possibly non-physical, post-transcriptomic effects.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

	Min.	25th quantile	Median	Mean	75th quantile	Max.
# RFs	1	1	2	2.202	3	11
GTP	0.796	0.798	0.800	0.800	0.800	0.802

Table 2.1: **Distribution statistics for the inferred networks.** Distribution statistics on the number of *unique* regulatory functions (RFs) across solutions per gene, and on the value of the general topological parameter (GTP) used for network selection in step (C) of the inference procedure. All values are rounded up to the 3rd decimal place.

Comparison of the different solutions. Now, we consider all 25 [4] solutions generated at the end of step (B), and estimate how far they are from each other, in terms of node degree distribution, edge numbers, redundancy in interactions, unicity of regulatory functions for each node across those solutions, and values of general topological parameter (GTP). GTP is a value comprised between 0 and 1 that is used to select the final network among the 25 ones (as further described in Section 9.1 in Appendix) and characterizes the proximity of a network topology to a scale-free topology.

Table 2.1 shows distribution statistics about the values of GTP and the unicity of gene regulatory functions across solutions. Note that all solutions present similar topologies, with similar GTP scores quite close to 1, which matches what can be expected from biochemical interaction networks in non-fungi systems (Broido and Clauset, 2019). Moreover, except for less than 25% of the genes in the network, genes are assigned at most 3 different regulatory functions across all solutions, which shows that their function in the network is globally preserved.

Figure 2.3 displays the boxplots of edge number and node degree distributions across solutions. These two plots show that, as mentioned before, the typical scale-free topology, with a few "hub nodes" with large degree and a large number of genes with few regulatory interactions, is present in all solutions. There are 74 interactions (that is, around $30 - 34$% of edges) which are present in at least 75% of the solutions, among which 25 are present in all of them. They are shown in Table 9.5 located at the Appendix. These numbers are confirmed by plotting the network comprising of all genepairwise interactions which are present in at least one solution, shown in Figure 2.4.

All in all, the networks obtained just before the model selection step mostly seem similar, both functionally (at the level of regulatory functions) and topologically (considering the node degrees, the number of edges, and the GTP scores).

[4] That number was chosen for reasons related to computational cost and time.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

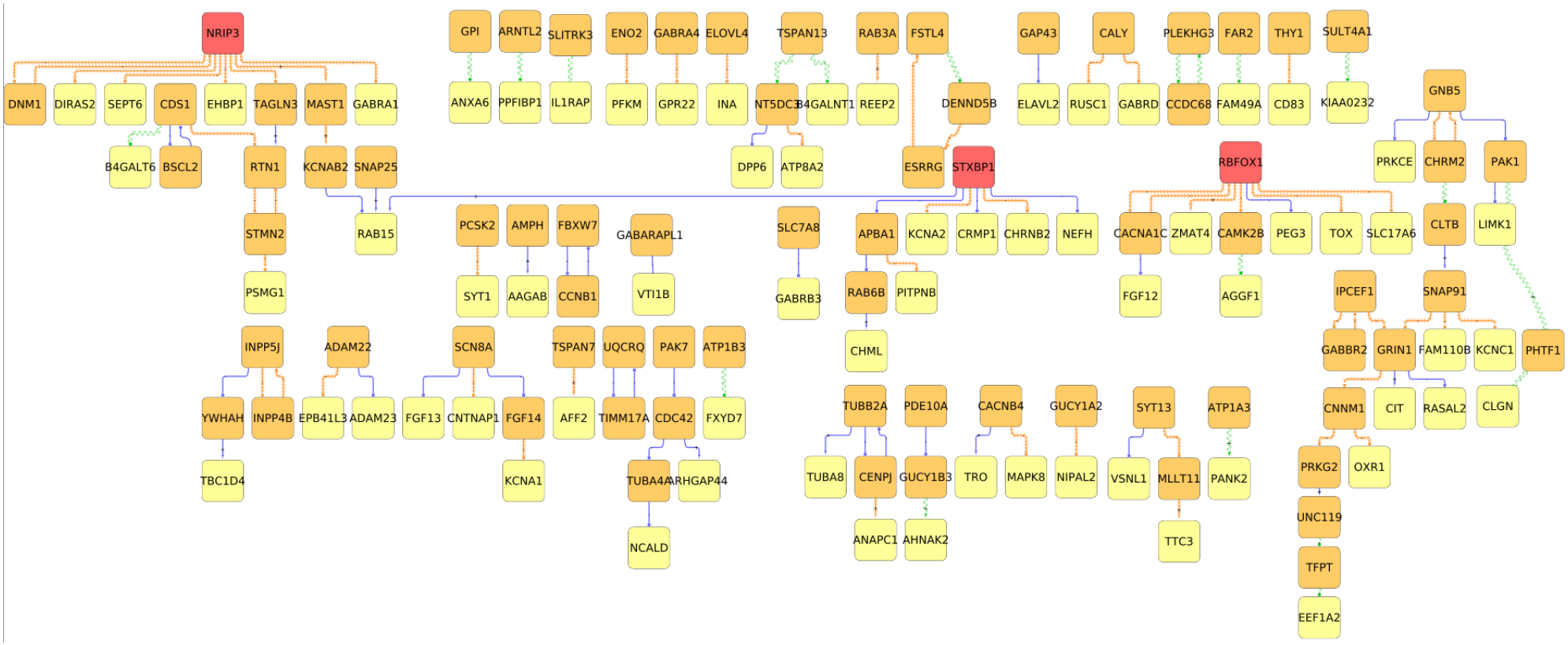

Figure 2.2: **Inferred network for the application to epilepsy.** Inferred network resulting from our pipeline applied to the M30 gene module. Tee-headed arrows represent inhibitory regulatory interactions, whereas triangle-headed arrows are activatory regulations. Edges are drawn according to their source of evidence (on the *undirected* interactions) as reported by the STRING database: contiguous arrows denote coexpression, solid line denote experimental proofs of interaction, and sinewave interactions are derived from text-mining procedures. Gene nodes are colored according to their out-degree: lightest color for genes with outdegree equal to 0, darkest for genes with an outdegree higher than 5. Isolated nodes are not shown.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

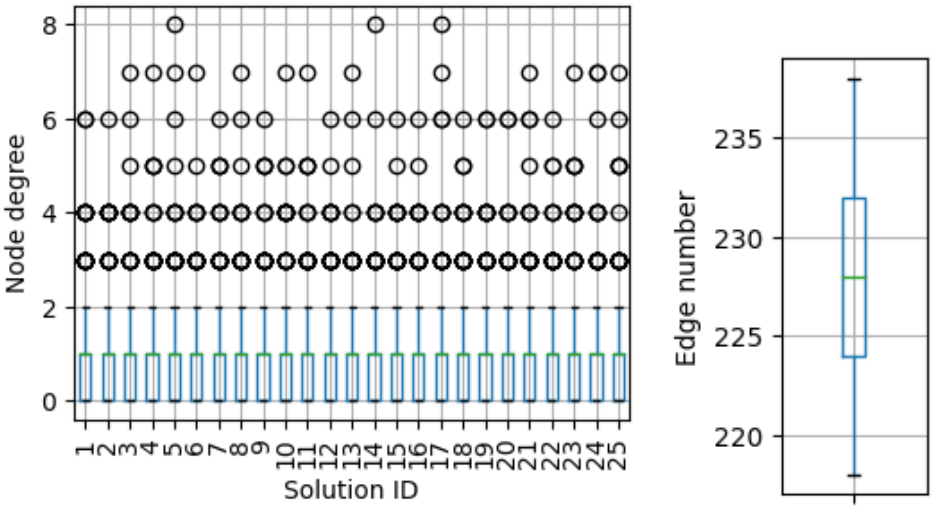

Figure 2.3: **Comparison of the inferred networks.** *Left-hand plot*: Boxplots of node total degrees (in- and out-degree) per solution. The green lines represent median values. *Right-hand plot*: Boxplot of the number of edges across solutions. Again, the green line represent the median value.

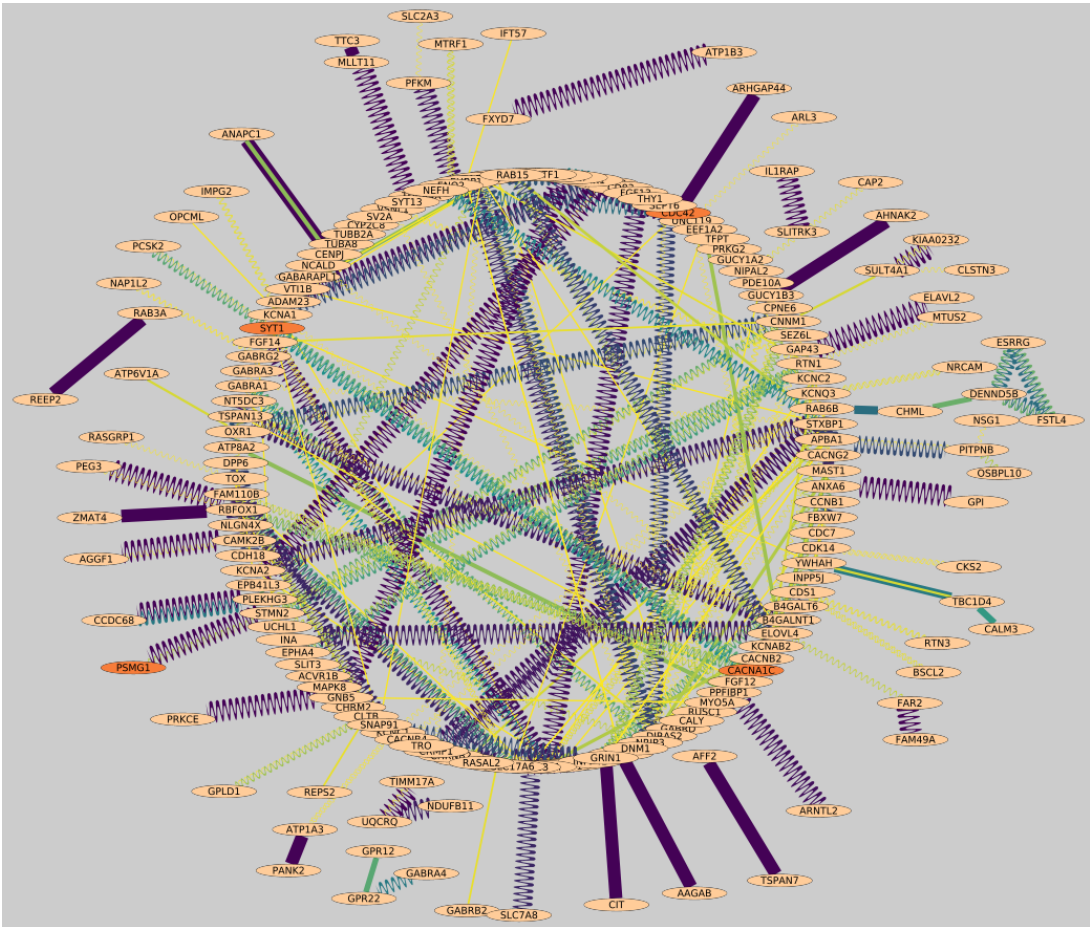

Figure 2.4: **Illustration of the collapsed network.** Network comprising of all gene-to-gene interactions which are present in at least one solution. The darker and thicker an edge is, the more frequent it is across solutions. Sinewave edges are inhibitory interactions, whereas solid lines denote activatory interactions. Orange nodes correspond to the genes which are perturbed in the LINCS L1000 experimental profiles used for inference.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Recommended master regulator candidates

We now study the results from the method of detection of master regulators.

Comparison to known gene measures. We computed the correlation between spread values for our application to epilepsy, genewise Control Centrality (Liu, Slotine, and Barabási, 2012) values computed with CytoCtrlAnalyser (Wu, Li, et al., 2018), and genewise outgoing degrees. The outgoing degree ("outdegree") is the number of direct downstream targets, whereas Control Centrality is the number of nodes which are affected by a change in the considered node, based on the (directed) network topology. More specifically, in order to compute the Control Centrality for any gene g in the setting we consider, at some time step t, (continuous) expression levels $x(t) \in \mathbb{R}^N$ are time-invariant and depend linearly on those $x(t-1) \in \mathbb{R}^N$ at the previous time step $t-1$, where N is the number of genes in the network

$$\frac{\partial x(t)}{\partial t} = Ax(t) + u_g(t) \ , \tag{2.4}$$

where A is the adjacency matrix in $\mathbb{R}^{N \times N}$ associated with the network, and $u_g(t) \in \mathbb{R}$ is the external signal imposed on node g at time t (either overexpression if it is positive, or knockout otherwise). In such a system, computing the number of nodes which can be controlled by gene g boils down to getting the rank of the so-called controllability matrix related to A and g, which is a function of powers of matrix A. This rank can be computed by solving a combinatorial optimization problem described in Equation (3) in Liu, Slotine, and Barabási (2012). Moreover, since, as a general rule, the true nonzero values in A as well as $u_g(\cdot)$ are unknown, Control Centrality aims at quantifying *structural* controllability, independently from the values of nonzero coefficients in A and $u_g(\cdot)$. All in all, Control Centrality is a solid counterpart to our method. It does not take into account neither the set of regulatory functions nor the gene expression levels in patients, but models regulatory cascades through the differential equation in Equation (2.4).

We compared these measures, related to the influence of a node in the network, to scores associated with the pathogenicity of genes :

- probability of loss of function intolerance (pLI) (Lek et al., 2016), which quantifies the intolerance to a deleterious mutation of a given gene.

- enhancer-domain score (EDS) (Wang and Goldstein, 2020), which studies the conservation of the regulatory domain around genes ;

- residual variation intolerance score (RVIS) (Petrovski et al., 2013), which quantifies functional genetic variation, and is anticorrelated to gene

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

pathogenicity.

Finally, we computed influence scores (Nicolle, Radvanyi, and Elati, 2015) as well, which expression is reported in Equation (2.1).

Figure 2.5 displays the Spearman's ρ correlation heatmap between these different measures. We observed that, contrary to influence values, spread values were consistent and strongly correlated with Control Centrality and the outgoing degree, that is, network-dependent measures. Moreover, spread values have a stronger correlation to the gene pathogenicity-related measures pLI and (opposite of) RVIS.

We tested whether the spread value was actually totally determined by the number of downstream (not necessarily direct) regulated genes. To do so, we performed a Spearman's ρ linear correlation test on the spread values and the number of downstream regulated genes. We confirmed that there is a strong, significative correlation between the two –which is expected, given the definition of the spread value– but that the spread value is not completely determined by this value, that is, the correlation value is not equal to 1 ($\rho = 0.82$, $p = 3.10^{-57}$).

Enrichment analysis in epilepsy-related terms of genes with high spread values. From Figure 2.6, it can be noticed that there is a lot of discrepancy between pLI scores and spread values on M30 genes. Nonetheless, it should be noted that Ziegler et al. (2019) warns against genes which are involved in recessive forms of diseases, while having a low pLI score. That is actually the case for gene *GNB5*, which has a central place in our network (see Figure 2.1) with spread value 0.024, and a pLI score close to 0, and is involved in a recessive form of epileptic encephalopathy (Poke et al., 2019).

In order to statistically test if the top genes for spread values are related to epilepsy-related symptoms and mechanisms, we performed a pathway enrichment analysis. This analysis allows the identification of the gene functions most represented among these top genes. An over-representation analysis (ORA) (Yaari et al., 2013) determines which sets of functionally similar genes are statistically surrepresented in the subset of genes, if they exist. Surrepresentation of a gene function set is quantified by comparing the gene function categories represented in the subset of genes (enrichments) to the categories present in a background set of genes, which contains the subset of interest. The statistical tests –one per gene function category– were run by the online tool WebGestalt (Liao et al., 2019).

In this case, the subset of considered genes is the set of 14 genes with spread value greater than 0.01 (displayed in Figure 2.6). The selected back-

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Minimum	25th quantile	Median	Mean	75th quantile	Maximum
0.0	0.0	0.0	0.00254	0.0	0.0556

Table 2.2: **Distribution statistics of the *spread* values.** Distribution statistics (rounded up to the 5th decimal place) of the spread values obtained for M30 genes present in the inferred network.

ground genes were the 233 genes present in the Boolean network. We considered the gene categories annotated in the DisGeNet database (Piñero et al., 2020). [5]

Indeed, the subset of genes is (weakly) significantly enriched in genes related to the term "Epileptic encephalopathy" (odds ratio $OR = 7.5$, Benjamini-Hochberg (BH) (Benjamini and Hochberg, 1995)-adjusted $p \approx 0.038$), and more strongly enriched with (neuro)developmental issues, for instance, "Loss of developmental milestones" ($OR = 10.5$, BH-adjusted $p \approx 0.012$), as reported in Figure 2.5. Similar results can be observed on another family of gene annotations, GLAD4U (Jourquin et al., 2012), as shown in Figure 9.2 in Appendix. These enrichment results go beyond the fact that M30 is globally enriched in epilepsy-related *de novo* mutations compared to the *whole* measured genome, as shown in Delahaye-Duriez, Srivastava, et al. (2016) : what is shown is that, among genes in the M30 module, ranking by spread values pinpoints epilepsy-related genes.

Selection of a list of candidates. Based on the results shown in Figure 2.5 and Table 2.2, we have selected a shorter list of candidate genes, comprised of rather large spread value (greater than 0.01) and pLI score greater than 0.9, and of genes with very large spread value (greater than 0.02), in order to avoid the shortcoming in the pLI score mentioned above. Candidate genes are *CACNA1C*, *RBFOX1*, *STXBP1*, *DNM1*, *NRIP3*, *SCN8A*, *CHRM2*, *GNB5*, *TUBB2A*, *PAK7* and *GRIN1*, shown in Figure 2.6. Most of these candidates –except for *NRIP3*, which is notably mainly expressed in the hippocampus– have a relationship to epilepsy-related symptoms in humans as shown in prior works (Appenzeller et al., 2014; Butler et al., 2017; Cushion et al., 2014; Al-Eitan et al., 2019; Lal et al., 2013; Myers et al., 2016; Ohba et al., 2015; Poke et al., 2019; Stamberger et al., 2016), as expected due to their membership to the M30 module.

***In silico* validation of the candidates.** In order to further confirm our list of candidate genes, we exploited the Boolean network inferred in Section 2.2 to predict whether a knockout of each of these genes would trigger a transcriptional profile close to epileptic states. Knockouts are easier

[5] Remember that, in the application to epilepsy, we *do not* use genes from DisGeNet, but the preselected set of genes M30.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

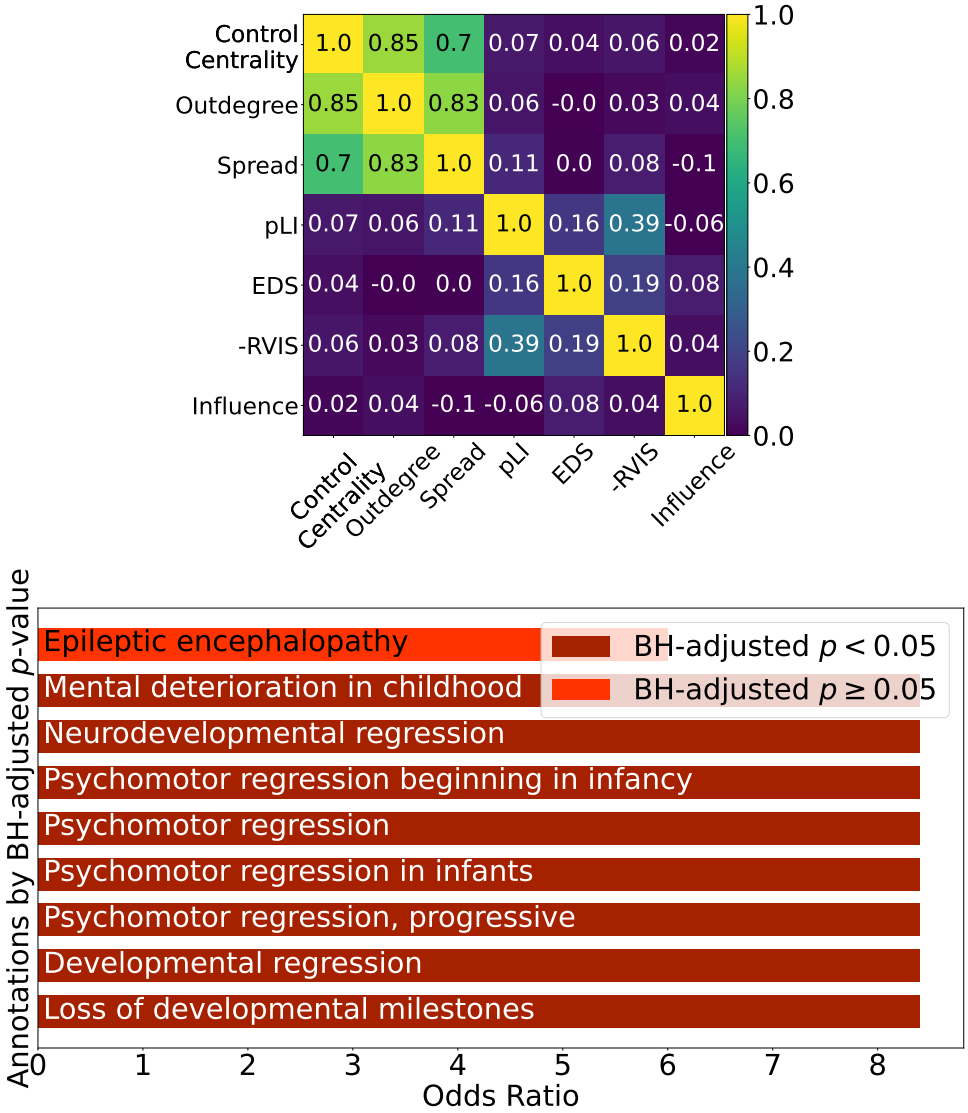

Figure 2.5: **Comparison between *spread* values and other gene measures.** *Top plot*: Spearman's ρ correlation heatmap between different gene measures either related to the influence of a node on a network, or to the genetic variations associated with pathogenicity. *Bottom plot*: Enrichment results from the ORA analysis. All reported enrichments have an adjusted p-value lower than 20%.

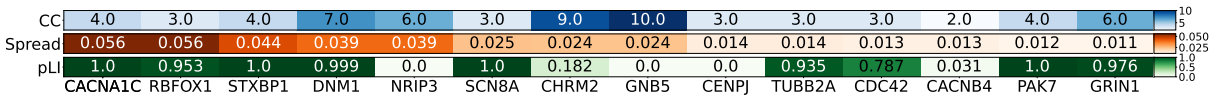

Figure 2.6: **Gene ranking by decreasing *spread* value.** Genes ranked by decreasing spread value, restricted to spread value greater than 0.01 (*center bar*), with their associated Control Centrality (CC) (*top bar*), and pLI scores (*bottom bar*).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Algorithm 2 Single gene perturbation scoring. Scoring of the effect towards pseudo-epileptic transcriptional profiles

Input: $\mathcal{C}$ set of control transcriptional profiles and $\mathcal{P}$ set of epileptic profiles ; a list of genes $\mathcal{L}$; $\mathcal{B} = (V, E, F)$ a Boolean network on node set V with edges in E and regulatory functions F

Initialize scores$(g) \leftarrow 0$ for any gene $g \in \mathcal{L}$
Binarize profiles in $\mathcal{C}$ and in $\mathcal{P}$ in matrix $X \in \{-1, 1\}^{(|\mathcal{C}|+|\mathcal{P}|)\times|M30|}$

Normalize the matrix of control and patient samples per feature (gene)
$\tilde{X} \leftarrow (\frac{1}{\sigma_g}(X_g - \mu_g))_{g\in\{1,\ldots,|M30|\}}$, where $\mu_g =$ mean(X_g) and $\sigma_g =$ std(X_g)

Learn a mapping from the initial high-dimensional space to 2D
Fit a Principal Component Analysis (PCA) $\mathcal{M} : \{-1, 1\}^{|M30|} \mapsto \mathbb{R}^2$ on matrix $\tilde{X}$

Run Characteristic Direction on the set of control and patient samples
to determine the classification frontier between the two groups
Compute $S \leftarrow$ CD$[\mathcal{P}\|\mathcal{C}] \in \mathbb{R}^{|M30|}$
for $g \in \mathcal{L}$ **do**
  for $c \in \mathcal{C}$ **do**
    Initialize scores_profile$(g) \leftarrow 0$ for any gene $g \in \mathcal{L}$
    Enumerate a set of attractors $\mathcal{A}_{(c,g)}$ which are reachable from binarized control profile c under the knockout of gene g
    Initialize list$(a) \leftarrow 0$ for $a \in \mathcal{A}_{(c,g)}$
    for $a \in \mathcal{A}_{(c,g)}$ **do**
      $\pi(\mathcal{M}(a)) \leftarrow$projection of $\mathcal{M}(a)$ onto hyperplane of normal vector $\mathcal{M}(S)$

      # Compute the "signed distance" between the considered attractor
      # and the classification frontier defined by $\mathcal{S}$ in 2D
      list$(a) \leftarrow$ signed$(\mathcal{M}(a), \pi(\mathcal{M}(a)))$
    end for

    # Compute a score based on the probabilities of presence of attractors
    scores_profile$(c, g) \leftarrow \sum_{a\in\mathcal{A}_{(c,g)}} p_{(a,c,g)}$list$(a)$ where $p_{(a,c,g)}$ is the probability associated with attractor a, initial profile c and knockout of g
  end for

  # Obtain a single score per gene
  scores$(g) \leftarrow \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}$ scores_profile(c, g)
end for
Output: scores

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

to experimentally confirm than overexpression perturbations, and most of the selected genes have a reported pro-epileptic deletion mutation (Appenzeller et al., 2014; Griffin et al., 2021; Lal et al., 2013). Our procedure is summarized in Algorithm 2. The key idea behind this algorithm is that the repurposing score aims at determining in which part of the 2D plane an *in silico* treated profile located : either on the hyperplane globally assigned to control samples, or the one mainly associated with patient profiles. Then, the "signed" distance of this treated profile to the frontier which separates these two hyperplanes is computed. The sign of this distance is indicative of the hyperplane the treated profile belongs to (positive if in the "control" hyperplane, negative otherwise). We now describe Algorithm 2. We consider this time both sets of binarized epileptic and control (healthy for epilepsy) hippocampi samples in Mirza, Appleton, et al. (2017), respectively denoted $\mathcal{P}$ ($|\mathcal{P}| = 24$) and $\mathcal{C}$ ($|\mathcal{C}| = 23$). These profiles are restricted to the set of M30 genes present in the network. The binarization step is performed similarly to what is described in Section 9.1 in Appendix. In order to score the perturbation related to a given gene, first, we fit a Principal Component Analysis (PCA) model $\mathcal{M}$ with the standard-normalized matrix of binary control and patient profiles (with values in $\{-1, 1\}$). The use of PCA allows to renormalize the distances on the 2D plane, and transform the profiles to make them more informative, by considering the two axes on which the most variance across samples is observed. Second, we consider the PCA-transformed Characteristic Direction (CD) signature (Clark et al., 2014) $\mathcal{M}(S)$ computed on patient and control profiles.

I refer the reader to the corresponding paper (Clark et al., 2014) for the details about Characteristic Direction. The main principle that should be kept in mind is that the signature CD[G1‖G2], computed through Characteristic Direction for two condition groups G1 and G2, is actually the vector normal to the decision frontier in a high-dimensional space, which classifies samples into either G1 or G2, and that is oriented in the direction from G2 to G1. That means in particular that the changes reported in the signature are changes in G1 compared to the reference group G2.

Then, for each candidate gene g and each control profile c available, we retrieve a set of attractor states denoted $\mathcal{A}_{(c,g)}$ reachable from state c under knockout of gene g, as described in Section 9.4 in Appendix. To compute the score associated with an attractor state a (that is, how close to a control profile it is), first, we PCA-transform this attractor into a 2D vector $\mathcal{M}(a)$; second, we compute the associated projection $\pi(\mathcal{M}(a))$ onto the frontier between control and patient samples (which normal vector is $\mathcal{M}(S)$) ; finally,

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

CACNA1A	*DNM1*	*GNB5*	*STXBP1*	*SCN8A*	*PAK7*	*TUBB2A*	*CHRM2*
-0.17	-0.17	-0.17	-0.20	-0.20	-0.20	-0.22	-0.24

NRIP3	*GRIN1*	*RBFOX1*
-0.25	-0.29	-0.30

Table 2.3: **Scores for genes prioritized by *spread* values.** *In silico* perturbation scores, rounded up to the second decimal place, resulting from the comparison of attractor states, reachable from control samples, to the set of control samples, after the knockout of one of the candidate genes in the Boolean network in Figure 2.2.

we compute the signed distance between $\mathcal{M}(a)$ and its projection $\pi(\mathcal{M}(a))$:

$$\text{signed}(\mathcal{M}(a), \pi(\mathcal{M}(a))) := \sum_{i=1}^{n} (\pi(\mathcal{M}(a))_i - \mathcal{M}(a)_i) \ .$$

Note that, contrary to the distance function used in Section 2.3, this quantity is signed. This value is positive if $\mathcal{M}(a)$ is in the zone delimited by the hyperplan which contains control samples, and negative otherwise. The large it is, the more the attractor state is considered in the control or patient zone. Such a score allows a quick interpretation of the results and a straightforward visualization of the improvement (or worsening) of the patient profile after treatment.

In the presence of several attractors, we use the probability $p_{(a,c,g)}$ of presence of attractor a reachable from the initial control state c under knockout of gene g, as provided by PyMaBoSS (Stoll et al., 2017), using the same parameters as in Section 9.4 in Appendix. For an attractor which is not reachable from profile c, we have $p_{(a,c,g)} = 0$. The final score for a given gene g based on $|\mathcal{C}|$ control profiles is then :

$$\text{score}(g) := \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \sum_{a \in \mathcal{A}_{(c,g)}} p_{(a,c,g)} \times \text{signed}(\mathcal{M}(a), \pi(\mathcal{M}(a))) \ .$$

If no attractor state is found, then we set this value to "not a number". [6] Table 2.3 shows the scores obtained for each candidate gene. These scores show that any knockout of these genes will globally turn a control transcriptomic profile into one which is similar to an epileptic one, with some of these genes which may never been investigated in the research related to epilepsy, such as *NRIPS3*, and some of them, notably, *STXBP1* and *GRIN1*, which orthologuous genes were associated with epileptic seizures in zebrafish (Griffin et al., 2021).

[6] This case never happened in practice.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

2.5 Discussion

We introduced in this work two main contributions to *in silico* disease research. First, we designed a method for the inference of gene regulatory network only from a subset of genes. This method carefully combines information from several public databases and methods, and infers a dynamical model of gene regulation adapted to the considered set of genes. This method yields to quite robust network solutions, and is easily reproducible. Second, we showed how to exploit the inferred network model, and more specifically its dynamics, to detect master regulatory genes, which may be interesting candidates to investigate a disease. We applied our methodology to tackle epilepsy, and to find novel gene candidates to investigate epilepsy. It allowed us to get a list of candidate genes which perturbations greatly impact the whole network in an epileptic transcriptomic context. The experimental validation of the knockout of some of these candidates in the zebrafish is an ongoing collaboration with Nadia Soussi-Yanicostas at Inserm UMR1141. This methodology allows reproducible and transparent research, while reducing the amount of data needed as input, which is one of the main caveats of researching on rare or tropical neglected diseases. Next, we will investigate, using the inferred network, the simulation of the effect of molecules –that is, which potentially targets several genes at a time.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 3

Drug efficacy scoring using a Boolean network

I now introduce how to use the Boolean network inferred in Chapter 2 in order to simulate and score the effect of a given drug at transcriptomic level. Had we had access to several clinical trials involving the considered disease and molecules –for instance, in the form of a binary matrix with molecules as rows and diseases as columns, where ones would indicate successful trials– we could have used traditional machine learning techniques for classification tasks (He, Yang, Gong, et al., 2020; Jarada, Rokne, and Alhajj, 2020; Xue et al., 2018; Yang, Luo, et al., 2019) to solve drug repurposing. However, when facing rare diseases, we actually lack of information about putative successful drugs, and even sometimes information about the disease itself (*e.g.*, classification, biological mechanism, ...). This problem bears more than a passing resemblance to the "cold start" problems in recommender systems. Nonetheless, when no prior information can be collected before running the recommender system, the methods used to circumvent this hurdle mostly rely on the guilt-by-association principle, where diseases with similar transcriptomic profiles may have common successful treatments.

However, when it comes to medical purposes, simple similarity is not enough, as mentioned in Réda, Kaufmann, and Delahaye-Duriez (2020), as a change in chirality [1] is enough to turn a morning illness treatment into a teratogen molecule. In particular, in order to perform signature reversion on a fixed pair of disease and drug, simply comparing the signature built from transcriptomic profiles from healthy individuals and patients on the one hand, and the one obtained on control and *in vitro* treated cells on the other hand, might not be enough to estimate the actual reversing

[1] That is, the orientation of the 3D molecular structure. Two molecules which only difference lies in their chirality cannot overlap, as the conformation of one is the mirror inverted version of the other.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

power of the considered molecule. Genes interact on the expression level of others through regulatory cascades (Bolouri and Davidson, 2003), and they sometimes undirectly act upon their own expression. When screening thousands of molecules, the experimental measure of the actual post-treatment transcriptomic profiles on diseased patients or animal models is too time-consuming and expensive to be effectively implemented ; hence the proposed drug repurposing method, which exploits the Boolean network inferred in Chapter 2 to model regulatory cascades, and predict post-treatment transcriptomic profiles.

In this chapter, I will describe how to leverage the Boolean network to predict post-treatment transcriptomic profiles. These predictions will help scoring the therapeutic effect of the treatment, by comparing *in silico* treated profiles and appropriate control ones. A special attention is brought to the selection of the drug-induced perturbations at transcriptomic level. These scores allow ranking putative drug candidates by their performance, which effectively solves the problem of drug repurposing. As a proof-of-concept, this procedure has been applied to epilepsy, for which a Boolean network was inferred in Chapter 2. The resulting ranking on a subset of known epilepsy-related molecules was compared to a baseline method, which does not model regulatory cascades.

3.1 Related work

In this section, I briefly recall important concepts related to this chapter.

Drug repurposing and signature reversion. Drug repurposing is a paradigm of drug development which aims at discovering new therapeutic indications for already commercially available molecules. The reuse of well-documented and already approved drugs allows decreasing the allocated budget and time for drug discovery research, while limiting the possibility of undiscovered negative side effects (Hwang et al., 2016).

A promising method is the method of "signature reversion" (Lamb et al., 2006; Sirota et al., 2011) : drug candidates are ranked according to the similarity between disease- and drug-associated signatures. Signatures describe the changes in gene expression which were induced by the disease, resp. by the treatment. The key idea is that, if the same genes are affected in both signatures in a different direction –*i.e.*, up-regulated in the drug signature, and down-regulated in the disease one, or vice-versa– then this treatment could be an interesting therapy. Many examples of applications

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

of signature reversion are available in the following review papers (Hodos et al., 2016; Musa et al., 2018). This approach has further been facilitated by the creation of databases of treated transcriptomic profiles, which are systematically produced through a standard RNA-sequencing pipeline on immortalized human cells. One of the first databases collecting this data is the Connectivity map (CMap) database (Lamb et al., 2006) in 2006, followed by the LINCS L1000 database in 2017 (Subramanian et al., 2017) at a larger scale. Each drug signature is built by comparing significant genewise changes in expression between cells *in vitro* treated by a control molecule or by a specific active molecule. The main asset of LINCS L1000 is that the database is extremely large : up to $30,000$ drugs or genetic perturbations are tested across 98 cell lines, going from $1,309$ chemical compounds applied to 5 cell lines in CMap. This is due to the reduced cost in the generation of transcriptomic data. Only around a thousand genes are truly assayed, and the expression of the remainder of the genome is inferred through a linear regression model. This computational inference seems to give satisfying results compared to the fully-assayed genome (Cheng and Li, 2016; Subramanian et al., 2017). However, these databases might provide discordant drug signatures, and thus rankings, probably due to differences in the technical tools used in their profile generation pipeline (Lim and Pavlidis, 2021). The latter work also highlights the importance of careful selection of drug signatures –in particular, in terms of cell lines– from these databases, as we will develop in the next sections.

The LINCS L1000 database has been exploited in a signature reversion method called "L1000 CDS^2" (Duan et al., 2016). First, this method computes the drug signatures on the whole LINCS L1000 genome ($\approx 12,000$ genes) with Characteristic Direction (CD) (Clark et al., 2014). CD is run over a set of treated and control samples from the same plate. Control samples are treated with a "sham" treatment, for instance dimethyl sulfoxide (DMSO). Then, CD signatures are averaged across replicates. In our application, in order to have a single drug signature per drug, I have also averaged across the two cell lines NPC (neural progenitor cell) and SH-SY5Y (neublastoma line), doses and exposure times. [2] Second, a CD signature $\mathcal{S}$ is obtained from applying CD to healthy and patient profiles (respectively denoted $\mathcal{C}$ and $\mathcal{P}$). [3] Last, L1000 CDS^2 ranks drugs by computing a cosine distance score between $\mathcal{S}$ and each drug signature. That is, for any drug signature s,

[2]This choice was made in order to ensure fairness when comparing this method to ours, since our drug signatures are computed over brain cell lines, as described later in this chapter.

[3]Characteristic Direction (CD) has been introduced in the previous chapter. In a nutshell, this procedure creates a vector CD$[\mathcal{G}_\infty \| \mathcal{G}_\in]$ which reports the genewise magnitude and direction of change in expression from reference sample group $\mathcal{G}_\in$ to treated group $\mathcal{G}_\infty$.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

the method computes

$$\cos(s, \mathcal{S}) := 1 - \frac{\sum_{\text{gene } g} \mathcal{S}[g] \times s[g]}{\sqrt{\sum_{\text{gene } g} \mathcal{S}[g]^2} \sqrt{\sum_{\text{gene } g} s[g]^2}} .$$

The higher the cosine score is, the more epilepsy-affected genes are perturbed by the drug in the opposite direction with respect to the disease ; then the considered molecule is a putative drug candidate. This method is the baseline for the suggested drug repurposing procedure presented in this chapter.

Gene regulatory networks as Boolean networks. A gene regulatory network (GRN) is a summary of gene regulatory interactions, which is depicted as a graph : nodes are genes or proteins, (directed) edges are regulatory interactions from a regulator to a regulated gene. We implement GRNs as Boolean networks. In this type of models, nodes can have two expression states : 0 for low expression, and 1 otherwise. A "regulatory function" is assigned to each node. This function computes the new expression state of the associated node, depending on the expression states of its direct regulators at the previous time step. A network state is the concatenation of all the expression states at a time point. The procedure to update a network state using the regulatory functions is defined by the dynamics of the model. Examples of dynamics are described in Chatain, Haar, and Paulevé (2018); Paulevé et al. (2020). A Boolean model can predict the final binary network state(s) under the perturbation of one or several genes in the network, by iteratively applying the update step. These final states are attractor states. An attractor state is such that either (a) applying the update step leaves that state invariant (steady attractor) ; (b) the system oscillates in a cycle to which this state belong (unsteady attractor).

Application to epilepsy. In Chapter 2, we focused on a module of 320 genes called M30, which has been shown to have a global gene expression that is anti-correlated to various epileptic transcriptomic profiles (Delahaye-Duriez, Srivastava, et al., 2016) ; and we inferred a Boolean network which models the regulatory interactions in this gene module.

At the end of the previous chapter, we actually described a method which allowed us to score the effect of a single gene perturbation on the network. The goal of the current chapter is to explain how we have extended Algorithm 2 to scoring chemical compounds ; and, more specifically, how we have modelled the effect of the treatment, based on a careful selection of drug signatures from the LINCS L1000 database (Subramanian et al., 2017).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

3.2 Selection of drug signatures

Extension of Algorithm 2 to drug perturbations. Algorithm 2 in Chapter 2 predicts (binary) transcriptomic profiles under the perturbation of a single gene. These predictions are (steady) attractor states reachable from a control profile [4] under the knockout of a given candidate gene g. To each knocked-out gene, a score was computed as a weighted sum of "signed distances" from each attractor state to the classification frontier between control and patient samples. This frontier was defined by the Characteristic Direction signature (Clark et al., 2014) obtained from healthy and epileptic patient profiles, which corresponds to the normal vector to this frontier. However, potentially several genes can simultaneously be perturbed.

Nonetheless, current lists of drug targets –*i.e.*, genes which are directly affected by a drug at transcriptional level– are not completely satisfying for our purpose. This motivates the use of drug signatures, which are a proxy to estimate which genes are specifically affected by a treatment and in which direction, *i.e.*, whether they would be overexpressed or underexpressed after the treatment.

Gene targets. Several online databases list drug-associated gene targets : DrugBank (Wishart et al., 2018), MINERVA (Hoksza et al., 2019), Drug Central (Ursu et al., 2016), Therapeutic Targets Database (TTD) (Zhou et al., 2022) or LINCS L1000 (Subramanian et al., 2017). Let us check the relevance of these lists with regards to our application to epilepsy.

We compiled a set of 71 known antiepileptics and proconvulsant drugs –*i.e.*, which trigger seizures, which is shown in Table 10.1 (36 antiepileptics) and in Table 10.2 (35 proconvulsant drugs) in Appendix. The objective is to determine how well the information about the drug targets in the M30 module discriminates between antiepileptics and proconvulsant drugs, from the comparisons in Figure 3.1. Each list of gene targets (per database of origin, independently of the direction of change) is turned into a binary vector of size (71×320), where ones represent gene targets to a given molecule prior to the comparisons. In the right-hand plot in Figure 3.1, an "aggregated" target matrix in $\{0,5\}^{71\times320}$ was built, such that coefficient in position (i,j) is the number of times gene j is denoted as a target to drug i across the listed five databases.

Indeed, as shown in the right-hand heatmap in Figure 3.1, the heatmap regroups on one hand (right-most cluster at the top of the plot) antiepileptics which target gamma-aminobutyric acid (GABAergic) receptors : *GABRA4*,

[4] That is, the transcriptional profile of a healthy individual with respect to epilepsy.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

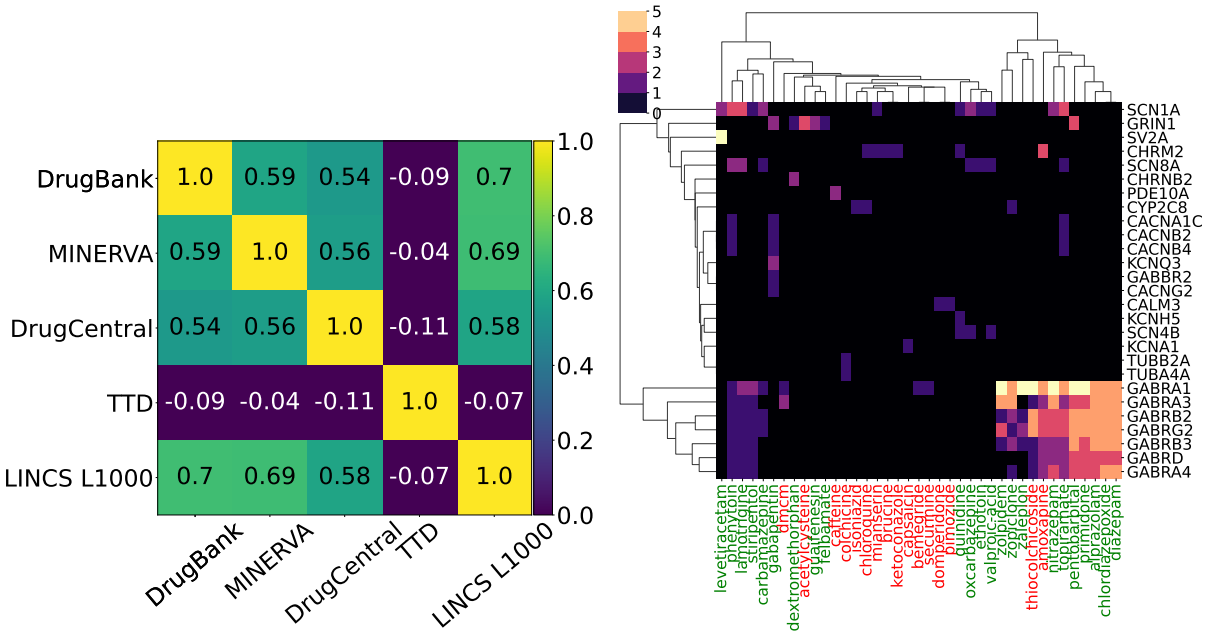

Figure 3.1: **Visualization of gene targets for antiepileptic and proconvulsant drugs.** *Left plot* : Spearman's ρ correlation matrix heatmap which represents the redundancy of targets across databases. *Right plot* : Heatmap associated with the aggregated target matrix described in the main text. This matrix is restricted to genes ($39/320$) and drugs ($27/71$) with at least one nonzero coefficient in the matrix. Antiepileptic, resp. proconvulsant, drugs are written using a green, resp. red, font. Rows and columns are ordered according to two hierarchical clusterings represented on the plot.

GABRD, *GABRB3*, . . . , which are related to synaptic transmission and notably known for intervening in epilepsy-linked mechanisms (Treiman, 2001) ; and on the other hand, drugs targeting sodium channel genes : *SCN1A*, *SCN8A*, *SCN4B*, which are involved in the initiation and propagation of action potentials in the central nervous system (Meisler, O'brien, and Sharkey, 2010) ; and calcium genes : *CACNA1C*, *CACNB4*, . . . , which might induce neuron hyperexcitability (Steinlein, 2014). However, these lists are discordant across databases, as shown by the left-hand plot in Figure 3.1. Moreover, they do not usually provide the direction of expression changes (except for TTD). Last, these lists are too sparse, *i.e.*, a large number of drugs have no reported gene targets, and some genes cannot be linked to any drug, as shown in the caption for the right-hand plot in Figure 3.1.

Nonetheless, these gene targets constitute an interesting ground truth to guide the feature selection to build drug signatures.

Types of drug signatures. We considered three different methods, denoted "BrainCell", "AllCell" and "BestCell", to build drug signatures from the transcriptional profiles in the LINCS L1000 database (Subramanian et al., 2017) for the 71 selected antiepileptics and proconvulsant drugs. These

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

three approaches are mentioned in both plots in Figure 3.2. They only differ on the considered type of cell lines. As mentioned in Lim and Pavlidis (2021), the correct selection of the cell line(s) on which differential expression is considered is key in signature reversion.

Any of these methods proceeds as follows : for any molecule, transcriptional profiles from control and treated cells, with at least two technical replicates, are retrieved. Depending on the selected approach, these profiles are possibly filtered on the cellular type on which they were measured. Then, at most 30 of the most reproducible profiles are kept for each group of profiles (treated and control groups). [5] Last, we run Characteristic Direction (Clark et al., 2014) on these two groups, which outputs a single real-valued vector. This vector reports the magnitude of change in expression for each gene. Based on the p-values (per gene) computed by CD, [6] this vector is binarized by assigning ones to genes with positive magnitude and significant p-value at level 5%, resp., zeroes to genes with negative magnitude and significant p-value at level 5%. In particular, it means that ones correspond to genes which are significantly upregulated after treatment –with respect to the control group– whereas zeroes correspond to significantly downregulated genes. Genes which are not represented in this vector are non significantly differentially expressed, that is, almost unaltered by the treatment.

As previously claimed, we consider the three following filters on cell lines

- In "BrainCell", we further restrict the set of transcriptional profiles to profiles obtained in one of the two brain cell lines in LINCS L1000 : neural progenitor cells (NPCs) and a neuroblastoma cell line (SHSY-5Y).

- "AllCell" does not use any filter on the cell line.

- "BestCell" only considers the cell line in which a treatment experiment with this drug has yielded the highest Transcriptomic Activity Score ("cell_tas"), as reported by the LINCS L1000 database. The highest this score is, the most reliable the profiles with this treatment in this cell line are. [7]

In all three methods, 194 genes (out of 320 genes present in M30) are significantly differentially expressed in at least one of the drug signatures computed for the 71 epilepsy-related molecules. Due to the fact that experiments involving drugs do not necessarily cover all 98 cell lines present in

[5] This is done in practice by considering the reported value `distil_ss` in LINCS L1000 for each profile. This quantity is correlated to the number of differentially expressed genes (genes altered by the treatment), and is predictive of the reproductibility of the profile (Lim and Pavlidis, 2021).

[6] We set the number of permutations performed by CD at $10,000$.

[7] This metric (like others such as `distil_ss`) is further discussed at https://clue.io/connectopedia/glossary.

Except where otherwise noted, this work is licensed under https://creativecommons.org/licenses/by-nd/3.0/fr/

LINCS L1000, there are only 34 drug signatures in "BrainCell", whereas there are 68 drug signatures in "BestCell" and "AllCell". The list of 34 antiepileptic and proconvulsant drugs having a "BrainCell" signature is reported in Tables 10.1 (12 antiepileptics) and 10.2 (22 proconvulsants) in Appendix (column "BrainCell"). Note that not all 71 drugs are present, due to the condition on having at least two technical replicates.

Feature selection with respect to gene targets. We compare all three methods to the lists of gene targets mentioned earlier in Figure 3.2. We denote "Aggregated" the binary matrix in $\{0,1\}^{71\times320}$, where the coefficient at position (i,j) is equal to 1 if and only if gene j is reported as a target of drug i in at least one of the following 5 databases : DrugBank, DrugCentral, LINCS L1000, TTD, and MINERVA. Similarly, for each type of drug signature, the associated drug signature matrix is converted into a binary matrix in $\{0,1\}^{71\times320}$, where ones represent significantly differentially expressed genes (up-regulation or down-regulation) according to the drug signatures.

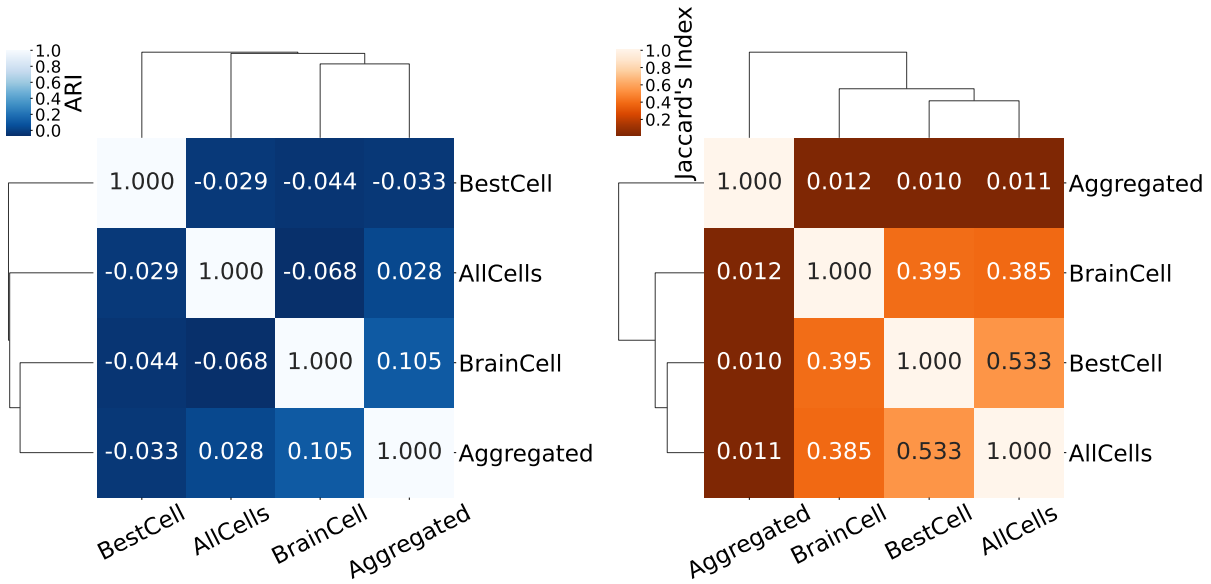

Figure 3.2: **Comparison between the different types of drug signatures.** Similarity between the three types of drug signatures retrieved from LINCS L1000 and the lists of gene targets, on the set of 71 epilepsy-related drugs. Associated matrices are converted into binary vectors of size (71×320). Each pairwise comparison is restricted to common genes and drugs. *Left plot* : We compare the similarities of clusterings (obtained by running k-means++ (Arthur and Vassilvitskii, 2006) with 3 clusters) by computing the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985).[a] *Right plot* : The Jaccard index (Jaccard, 1912) is computed between approaches. In both heatmaps, the higher the score, the most similar two methods are. Rows and columns of the heatmaps are ordered by a hierarchical clustering represented on the plots.

[a]The choice of 3 clusters was motivated by the minimum number of clusters which could be found using any of the three types of features.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Figure 3.2 shows that, out of the three methods for retrieving drug signatures, the one exclusively using profiles from brain cell lines ("BrainCell") is the most similar to the aggregated list of gene targets ; both in terms of regrouping drugs (left-hand plot), and in terms of intersection of genes (right-hand plot). However, there is still little correlation between the drug signatures from LINCS L1000 and the gene target lists. This might be due to the fact that a gene might be differentially expressed –up- or downregulated– even when it is not directly targeted by a given drug. Moreover, the number of currently reported targets might be a lot smaller than the actual number of targets. Based on these plots, from now on, we only consider only "BrainCell" drug signatures.

3.3 Scoring with the Boolean network

Now, we lay out the outlines of the extension of Algorithm 2 in Chapter 2 to scoring chemical compounds, instead of single gene perturbations. Algorithm 3 describes the associated pseudo-code, with differences with Algorithm 2 written in bold type. Notice that one of the main differences is that we consider as initial conditions perturbed *patient* profiles –instead of profiles of healthy individuals– and that perturbations on the initial states are defined by drug signatures, and no longer by single gene perturbations. That means in particular that, if a gene is denoted as upregulated in a drug signature, then the corresponding perturbation of this gene in the algorithm will be an overexpression. Conversely, a downregulated gene denotes a knockout perturbation of this gene.

In our application to epilepsy, we considered the sets of binarized epileptic and control hippocampi samples for temporal lobe epilepsy (TLE) in Mirza, Appleton, et al. (2017), respectively denoted $\mathcal{P}$ ($|\mathcal{P}| = 24$) and $\mathcal{C}$ ($|\mathcal{C}| = 23$). The use of these profiles is justified by the link between TLE and refractory epilepsies (refer to Chapter 2). This procedure was run on the 34 drugs related to epilepsy with the "BrainCell" signatures, as defined in Section 3.2. This took several hours to complete. [8] Indeed, the runtime is proportional to the number of nodes in the Boolean network $\mathcal{B}$ and the number of patient samples $|\mathcal{P}|$. Solving the problem of finding a single attractor state reachable from an initial state, using the most permissive semantics (Paulevé et al., 2020), is PSPACE-complete as a general rule. [9] It is coNP-complete under some conditions on the behaviour of regulatory functions. [10]

[8] On a personal computer (processor Intel Core i7-8750H, 12 cores @2.20GHz, RAM 16GB).

[9] *i.e.*, this problem can be solved using an amount of memory at most polynomial in the number of nodes in the network.

[10] *i.e.*, the proof that the considered state is not a reachable attractor state is verifiable in polynomial time (in the number of nodes) by a deterministic algorithm.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Algorithm 3 Drug perturbation scoring. Scoring the effect of a drug candidate on epileptic transcriptional profiles

Input: $\mathcal{C}$ set of control transcriptional profiles and $\mathcal{P}$ set of epileptic profiles ; **a list of drug candidates** $\mathcal{L}$ **along with their drug signatures** ; $\mathcal{B} = (V, E, F)$ a Boolean network on node set V with edges in E and regulatory functions F

Initialize scores$(d) \leftarrow 0$ for any drug $d \in \mathcal{L}$
Binarize profiles in $\mathcal{C}$ and in $\mathcal{P}$ in matrix $X \in \{-1, 1\}^{(|\mathcal{C}|+|\mathcal{P}|)\times|M30|}$
Normalize the set of control and patient profiles, per feature (gene)
$\tilde{X} \leftarrow (\frac{1}{\sigma_g}(X_g - \mu_g))_{g\in\{1,\ldots,|M30|\}}$, where $\mu_g =$ mean(X_g) and $\sigma_g =$ std(X_g)
Learn a mapping from the initial high-dimensional space and the
2D plane
Fit a Principal Component Analysis (PCA) $\mathcal{M} : \{-1, 1\}^{|M30|} \mapsto \mathbb{R}^2$ on matrix $\tilde{X}$

Run Characteristic Direction on the set of control and patient samples
Compute $S :=$ CD$[\mathcal{P}\|\mathcal{C}] \in \mathbb{R}^{|M30|}$
for $d \in \mathcal{L}$ **do**
  # We now consider patient profiles in initial conditions
  for $p \in \mathcal{P}$ **do**
    Initialize scores_profile$(d) \leftarrow 0$ for any drug $d \in \mathcal{L}$
    Enumerate a set of attractors $\mathcal{A}_{(p,d)}$ which are reachable from binarized **patient** profile p under the **perturbations by drug** d**, as defined by its drug signature**
    Initialize list$(a) \leftarrow 0$ for $a \in \mathcal{A}_{(p,d)}$
    for $a \in \mathcal{A}_{(p,d)}$ **do**
      $\pi(\mathcal{M}(a)) \leftarrow$ projection of $\mathcal{M}(a)$ onto hyperplane of normal vector $\mathcal{M}(S)$

      # Compute the "signed distance" between the considered attractor
      # and the classification frontier defined by $\mathcal{S}$ in 2D
      list$(a) \leftarrow$ signed$(\mathcal{M}(a), \pi(\mathcal{M}(a)))$
    end for
    # Compute a score based on the probabilities of presence of an
    # attractor
    scores_profile$(c, g) \leftarrow \sum_{a\in\mathcal{A}_{(p,d)}} p_{(a,p,d)}$list$(a)$ where $p_{(a,p,d)}$ is the probability associated with attractor a, initial profile p and perturbation by d
  end for
  # Get a single value per drug candidate
  scores$(d) \leftarrow \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}$ scores_profile(p, d)
end for
Output: scores

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Then, each call to the Boolean network to solve the reachable attractor problem in Algorithm 3 is polynomial (in memory space) in the number of nodes in the Boolean network, up to a multiplicative constant corresponding to the maximum total number of attractor states to enumerate. There are $|\mathcal{P}| \times |\mathcal{L}|$ such calls in total, where $|\mathcal{L}|$ is the number of considered drugs.

In Chevalier et al. (2019), using their implementation of the most permissive semantics, they have been able to solve a single call to the solver –with complex constraints, that is, a potentially even harder problem than the reachability of attractor states– in a couple hours of CPU time in networks with up to 200 nodes. In a nutshell, these problems are tractable in practice for a limited number of nodes in the network, but cannot be extended to a large-scale screening of hundreds of drugs. We will discuss the consequences of this high computational cost at the end of the chapter.

3.4 Results

We compare repurposing results on the set of 34 drugs mentioned in Section 3.2 from Algorithm 3 and L1000 CDS^2 (Duan et al., 2016), described in Section 3.1. Recall that L1000 CDS^2 computes a cosine distance score on CD signatures derived from LINCS L1000 profiles. In both approaches, the higher the score is, the best the candidate. The rankings on the 34 drugs are shown in Figure 3.3. Their associated receiver operating characteristic and precision-recall curves are represented in Figure 3.4.

From Figure 3.4, we observe that Algorithm 3 slightly improves over L1000 CDS^2 in predictability. In order to additionally assess the relevance of the proposed drug scoring method, we compute the hit ratio at fixed rank k (Koren, 2008), denoted HR@k (or Recall@k), which is the accuracy restricted to the top-k elements. This quantity can be computed from the rankings in Figure 3.3. The hit ratio metric is useful for evaluating recommendations, as one might only be interested in the first top items. Hit ratios at rank 2, 3, 5 and 10 –which are reasonable ranks for drug repurposing– for both methods are reported in Table 3.1. This table confirms that, indeed, the proposed drug scoring fares better than the baseline.

	HR@2	HR@3	HR@5	HR@10
Scoring	0.5	0.7	0.4	0.3
L1000 CDS^2	0.0	0.3	0.2	0.4

Table 3.1: **Hit ratios on the proposed drug scoring and the baseline.** Hit ratios at ranks 2, 3, 5, 10 in the proposed scoring and L1000 CDS^2 (Duan et al., 2016) , rounded up to the 1st decimal place.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

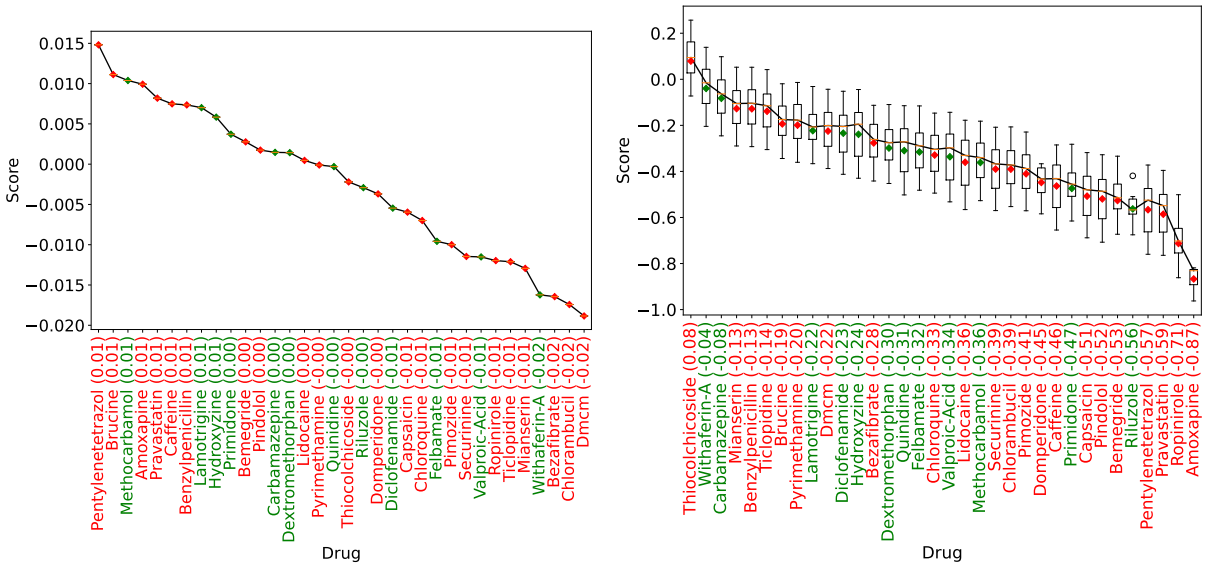

Figure 3.3: **Drug repurposing on a set of antiepileptics and proconvulsant drugs.** Boxplots of rewards across patient profiles on the set of 34 drugs, sorted by decreasing average score, as reported in the x-axis. [a] The dot (average score) and labels on the x-axis are colored in green (resp., in red) if they match an antiepileptic (resp., a proconvulsant) drug. The black solid line is the function of median score values. *Left plot* : ranking from L1000 CDS^2 (baseline). *Right plot* : ranking from Algorithm 3.

[a] In L1000 CDS^2, there is a single value equal to the average value.

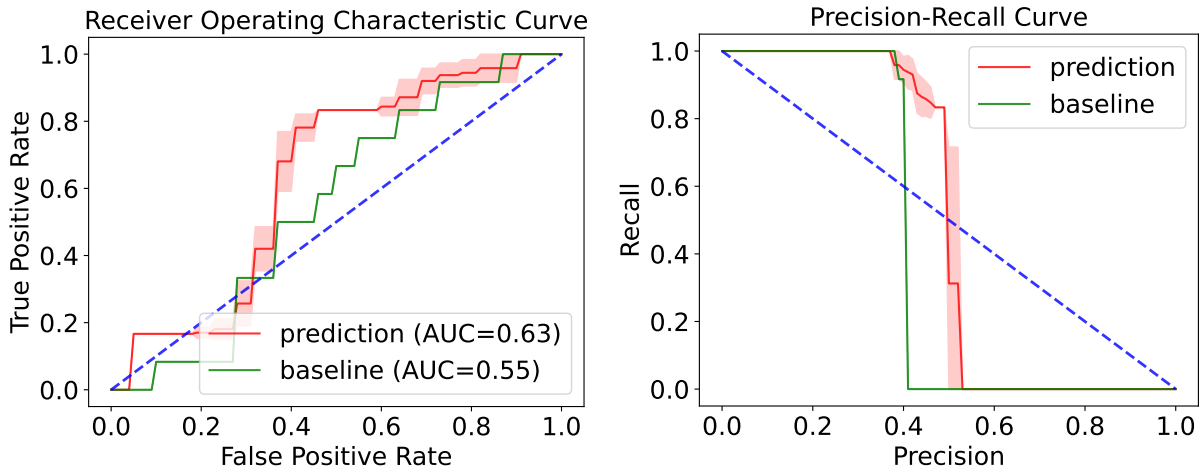

Figure 3.4: **Comparison between the drug repurposing method and the baseline.** Receiver operating characteristic (ROC) and precision-recall (PR) curves, representative of the performance of a method (prediction : Algorithm 3, baseline : L1000 CDS^2) on the set of 34 drugs. *Left plot* : ROC curves. *Right plot* : PR curves.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

3.5 Discussion

In this chapter, we described how the algorithm in Chapter 2 for scoring single gene perturbations was extended to drug-induced perturbations. The proposed method relies on a careful selection of the drug signatures, which highlights the importance of the choice of the cell line, as previously remarked by Lim and Pavlidis (2021). This method slightly improves over a baseline approach on a dataset of 34 antiepileptics and proconvulsants. Yet it is unfortunately still not satisfying in terms of interpretability. Indeed, most of the drug candidates are assigned negative scores ; according to the way the score was designed in Chapter 2, it means that these drugs do not truly "treat" patient profiles. Indeed, a negative score means that, in average, *in silico* treated profiles remain in the part of the 2D plane that is globally associated with epileptic profiles.

Nonetheless, as discussed in the introduction to this thesis, except for the use case described in Chapter 7, there is no transcriptomic database on individual patient responses to treatment at a large scale for any pathology, [11] similar to what LINCS L1000 is to drug perturbations on *in vitro* human immortalized cells. This work is an attempt at bridging the gap between what is known about the drug-induced changes in expression and modelling *in silico* patient responses to treatment.

Yet the proposed method might be improved with further work on the feature selection ; *e.g.*, by incorporating the influence of the treatment dose. Indeed, higher doses might be more effective and lead to more reliable transcriptional profiles –which was our primary criterion for feature selection. However, they might be toxic, and trigger unwanted side effects. Sometimes, high doses might even trigger paradoxical effects, that is, reinforcing the symptoms of the disease, as reported for some antiepileptics in *in vivo* experiments (Nakken et al., 2003; Osorio et al., 1989).

Moreover, as highlighted by the plots associated with Algorithm 3 (Figures 3.3 and 3.4), there exists a quite large variation, in terms of scores and predictability, across patient profiles. Indeed, on the right-hand plot in Figure 3.3, we can notice that some patients seem to respond well to the treatment (*i.e.*, with positive scores). For instance, at least one patient outputs a score close to 0.18 when treated by Whitaferin-A, even if the related average score across patients is negative. This raises the question on how to properly aggregate the scores across patients.

[11] *e.g.*, the Genomics of Drug Sensitivity in Cancer (GDSC) database (Yang, Soares, et al., 2012) for cancer subtypes.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Finally, as previously evoked when discussing the computational complexity of Algorithm 4, large-scale screening might not be tractable in practice. This fact led me to consider sequential learning methods (namely, multi-armed bandits) in the next chapters : combining an online sampler, which asks for the evaluation of a given drug at points in time, and Algorithm 4 (below) which performs the evaluation of a single drug on a single patient profile. Additionally, this paves the way for a personalized drug repurposing method, which suggests drug candidates which are the most adapted to a specific patient. Personalized drug recommendation in cancer was recently investigated in Montagud et al. (2022), showing promising results.

In the next chapters, we will apply multi-armed bandit algorithms to a drug repurposing instance, on a subset of 21 antiepileptics and proconvulsant drugs. This instance comprises of "BrainCell" drug signatures and associated average repurposing scores derived from Algorithm 3. This instance is further described in Table 10.3 in Appendix. This restricted instance, which displays a larger gap in scores between the set of antiepileptic and proconvulsant drugs, allows us to illustrate our proof-of-concept on the application of multi-armed bandit algorithms to drug repurposing. In practice, the actual gap in score is of course unknown, and such a transformation only holds to support our proof-of-concept.

Algorithm 4 Sequential drug perturbation scoring. Socoring of the effect of a drug candidate in a sequential setting

In practice, a patient profile is sampled at random from the available
pool of patient profiles
Input: a patient profile $p \in \{0,1\}^{|M30|}$ and a drug candidate d with drug signature $s_d \in \{-1,0,1\}^{|M30|}$
Parameters: $\mathcal{B} = (V, E, F)$ a Boolean network on node set V with edges in E and regulatory functions F, fitted Principal Component Analysis model $\mathcal{M}$, classification frontier defined by normal vector $\mathcal{M}(\mathcal{S})$ in the 2D plane

Enumerate a set of attractors $\mathcal{A}_{(p,d)}$ which are reachable from binarized patient profile p under the perturbations by drug d, as defined by its drug signature s_d
Initialize list$(a) \leftarrow 0$ for $a \in \mathcal{A}_{(p,d)}$
for $a \in \mathcal{A}_{(p,d)}$ **do**
  $\pi(\mathcal{M}(a)) \leftarrow$ projection of $\mathcal{M}(a)$ onto hyperplane of normal vector $\mathcal{M}(S)$
  # Compute the "signed distance" between the considered attractor
  # and the classification frontier defined by $\mathcal{S}$ in 2D
  list$(a) \leftarrow$ signed$(\mathcal{M}(a), \pi(\mathcal{M}(a)))$
end for
# Compute a score based on the probabilities of presence of an attractor
score $\leftarrow \sum_{a \in \mathcal{A}_{(p,d)}} p_{(a,p,d)}$list$(a)$ where $p_{(a,p,d)}$ is the probability associated with attractor a, initial profile p and perturbation by d
Output: score

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Part II

Adaptive drug testing using bandits

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

The contents of this chapter rely on some of my publications.[a]

[a] Réda, Kaufmann, and Delahaye-Duriez (2020). "Machine learning applications in drug development". *Computational and structural biotechnology journal*, 18, pp. 241–252; Réda, Kaufmann, and Delahaye-Duriez (2021). "Top-m identification for linear bandits". *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1108–1116; Réda, Tirinzoni, and Degenne (2021). "Dealing With Misspecification In Fixed-Confidence Linear Top-m Identification". *Advances in Neural Information Processing Systems*, 34.

In Chapter 3, we described how to build a model which simulates the transcriptional effect of a given drug, based on a Boolean network. In particular, this procedure can be seen as a black box, which takes as input a drug signature –as previously defined previously– and a patient transcriptomic profile, and returns a score proportional to its signature reversing power. However, in practice, we know that this model retrieves a set of reachable steady attractor states in order to output a single score. Moreover, since we know there can be a high variability across patients' scores, estimating robustly the average score (across patients) of a given drug would require iterating it for all patients. All in all, this might be a time-consuming and computationally expensive process, especially when performed for a large number of drugs. I wondered if there was some adaptive, "smart", procedure for exploring the space of drugs. This is why I resorted to multi-armed bandit algorithms, which are introduced in the first sections of this chapter. For a given disease, we are interested in identifying a subset of drugs that may have a therapeutic interest. Providing a group of 5 or 10 drugs, rather than a single one, can ease the decision of further investigation, as many leads are provided. The problem of determining the N best options out of K ones was introduced in Kalyanakrishnan et al. (2012), and is named EXPLORE-N (Kalyanakrishnan et al., 2012), Top-N identification (Chen, Li, and Qiao, 2017), or Best-N identification (Jiang, Li, and Qiao, 2017). Moreover, drug signatures can be leveraged to discriminate between drug candidates, and make the algorithm more efficient. Hence, in my PhD, I worked on *structured* Top-N identification, [13] which exploits the structure of the model, that is, the relationship between drug signatures and scores.

[13] Which is the name we will retain for the remainder of the manuscript.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 4

Introduction to Top-N identification

We will use machine learning techniques to refine our drug repurposing method. Machine learning is a subfield of artificial intelligence in Computer Science. Here, a machine learning algorithm designates any computational method where results from past actions or decisions, or past observations, are used to improve predictions or future decision-making. Machine learning techniques are now extremely popular in drug development (Aliper et al., 2016; Ekins et al., 2019; Hodos et al., 2016) as they allow automation of highly-dimensional, noisy biological data analysis. Many different machine learning tasks have been studied, which fall broadly into three categories. The first one is supervised learning, in which the goal is to predict the label of new observations given a large database of labelled examples. Several supervised learning algorithms have been applied in a biological context, such as Support Vector Machines (Noble et al., 2004) or (Deep) Neural Networks (Chen, Engkvist, et al., 2018). The second task is unsupervised learning, and it aims at detecting underlying relationships or patterns in unlabeled data. Dimension reduction methods, like Principal Component Analysis (PCA), fall in this class. But other unsupervised problems are also studied in the context of drug development, such as density estimation and clustering (Zhang et al., 2018). [1] The third type of task is renforcement learning, where algorithms rely on trial-and-error, and iteratively use external observations in order to find the best decision with respect to the environment they interact with. A large literature has dwelled on the use of sequential learning algorithms, where an agent, *i.e.*, a goal-oriented entity which interacts with its environment, must make one choice at a time according to previous observations of the environment. Offline –or batch–

[1] That is, grouping data.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

methods use batches of data in order to learn, whereas online –or sequential learning– algorithms process one data point at a time, and update their prediction or decision accordingly.

In this thesis, I am interested in multi-armed bandit algorithms. They constitute a popular and versatile family of sequential decision-making algorithms. In multi-armed bandit algorithms, a fixed set of actions, called arms, is available. An agent sequentially interacts with the environment by selecting arms, as illustrated in Figure 4.1 below. Each arm selection produces some noisy observation, often interpreted as a reward. However, the average reward associated with each arm is initially unknown to the agent, and has to be learnt in the process while achieving a certain objective. Typically, this objective could be to discover the most efficient arm(s), that is, the arm(s) with highest average reward (Kalyanakrishnan et al., 2012), or to maximize the total reward accrued across iterated arm selections (Auer, Cesa-Bianchi, and Fischer, 2002). See the following referenced paper (Kaufmann and Garivier, 2017) for a comparison between these two bandit objectives.

In our application to drug repurposing, multi-armed bandit algorithms will help decreasing the number of calls to the Boolean network, introduced in Chapter 3, needed in order to find the best drug candidates with respect to their score. In terms of drug development, one might restrict their search for new therapies to N leads associated with the highest repurposing score. What is left to do is to design an appropriate recommender system to automatically test and recommend drug candidates. Recommender systems actually belong to different families of machine learning methods, since a recommender system broadly designates an algorithm which aims at predicting rating of a given user which tests a given object (Adomavicius and Tuzhilin, 2005). A large part of the literature about recommender systems is motivated by commercial purposes (Brynjolfsson, Hu, and Smith, 2010; Smith and Linden, 2017), including works on multi-armed bandit algorithms (Guillou, Gaudel, and Preux, 2016; Mary, Gaudel, and Preux, 2015). However, we will demonstrate that multi-armed bandit algorithms can actually be applied to solve drug repurposing.

As described in Chapter 3, one evaluation –that is, one call to the Boolean network– outputs a score, which is computed by comparing the attractor states reached after a drug-induced perturbation on a single patient profile, and the attractors states reached in the absence of perturbation. Computation of reachable steady states in Boolean networks, although easier than in continuous dynamical systems, [2] can still be expensive, especially as the

[2] Even easier with the dynamics described in Paulevé et al. (2020).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

network grows larger. Moreover, we only consider one patient profile at a time, and plots in Chapter 3 show that there is a high variability in scores across patients. In a nutshell, a single call to the Boolean network might be computationally expensive and noisy, hence the need to reducing the number of calls, especially when considering a large number of drugs to screen.

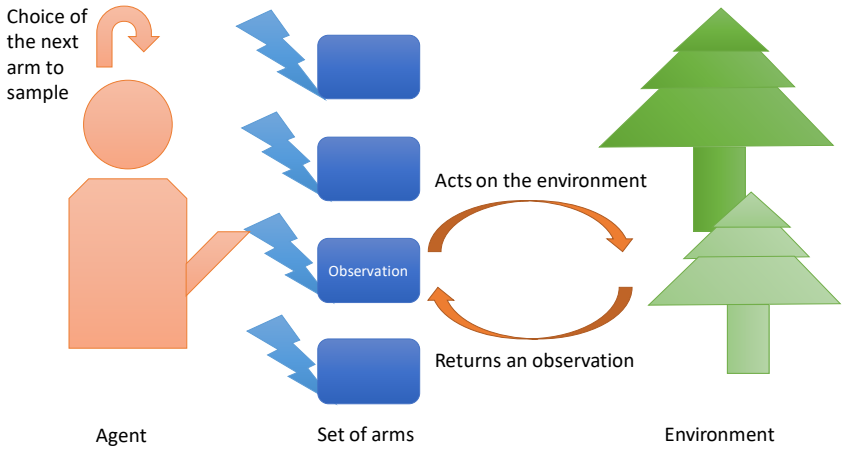

Figure 4.1: **Multi-armed bandits.** Illustration of the principle of multi-armed bandit algorithms.

4.1 Structured stochastic bandits

In this thesis, I consider a specific type of multi-armed bandits, [3] which are adapted to the use in real-life applications. We denote throughout the manuscript $[L] := \{1, 2, \ldots, L-1, L\}$ for any positive integer $L \in \mathbb{N}^*$. We consider a finite set of K arms, associated with feature vectors $(X_a)_{a \in [K]}$ which belong to $\mathbb{R}^d$. Let us denote the feature matrix $X \in \mathbb{R}^{K \times d}$ such that the feature vectors are the rows of this matrix : $X := [X_1^\top, X_2^\top, \ldots, X_K^\top]$. An agent sequentially selects arms for a given number of rounds or when a specific stopping criterion is satisfied. Sampling arm $I_t \in [K]$ at round t, based on past observations, [4] yields a noisy observation $Y_t \in \mathbb{R}$. Y_t is a realization of ν_{I_t} drawn independently from all past observations, denoted $Y_t \sim \nu_{I_t}$. Let $\mu_a := \mathbb{E}_{Y \sim \nu_a}[Y]$ be the expected reward associated with probability law ν_a for any arm $a \in [K]$. The probability law ν_a, for any arm $a \in [K]$, is assumed fixed and unknown to the agent (assumption of *stochasticity*), and depends on feature vector X_a (assumption of *structure*). In the cases studied during my PhD, I focused on zero-mean σ^2-subgaussian additive noise, for a fixed

[3] I denote "(multi-armed) bandit" the setting, and "bandit *algorithm*" the algorithm itself.
[4] And also some internal randomization variable $U_t \in [0, 1]$ in practice.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

value of σ that is known to the agent. [5] In order to reconcile all the works done on the structure of bandits, I introduce the following definition

Definition 4.1.1. ***$\mathcal{M}$-structured stochastic bandit with σ^2-subgaussian noise.*** *Let σ be a positive real number, and $\mathcal{M} \subseteq (\mathbb{R}^d \to \mathbb{R})$ a class of functions mapping from $\mathbb{R}^d$ to $\mathbb{R}$. If μ is $\mathcal{M}$-structured, then there is a deterministic function $f \in \mathcal{M}$ such that for any arm a, $\mu_a = f(X_a)$. At round t, the agent then observes*

$$Y_t := f(X_{I_t}) + \psi_t \ ,$$

where ψ_t is a centered random variable independent from past observations, and is σ^2-subgaussian, [6] that is, satisfies for any $\lambda \in \mathbb{R}$

$$\mathbb{E}[\exp(\lambda\psi_t)] \leq \exp\left(\frac{\lambda^2\sigma^2}{2}\right) \ .$$

The idea of this generic definition is to put an emphasis on the structure of the expected rewards for all arms, that is, the relationship between the expected reward μ_a and the feature vector X_a for any arm a. In particular, the key ingredient in such a structure is that parameters of functions $f \in \mathcal{M}$ are shared across arms. This is what makes speeding up the bandit algorithm possible, [7] since the set of putative models matching the observations is possibly reduced to $\mathcal{M}$-structured models. What should be kept in mind is that, here, features are a fixed, known, input to the problem, and one of the main questions tackled in this PhD is about their integration in the underlying (drug repurposing) model. Three familiar structures of bandit models are presented below.

Unstructured bandits ($\mathcal{M} = (\mathbb{R}^d \to \mathbb{R})$). This means that any function mapping from $\mathbb{R}^d$ to $\mathbb{R}$ can be considered, which implies that we do not actually restrict the set of possible models, since we do not exploit feature vectors. This is the most straightforward model for bandits –in regret minimization (Auer and Ortner, 2010; W. R. Thompson, 1933)– and was the first class of models considered for the identification of the N arms with highest expected rewards, see for instance Chen, Li, and Qiao (2017); Gabillon, Ghavamzadeh, and Lazaric (2012); Jiang, Li, and Qiao (2017); Kalyanakrishnan et al. (2012). Additional assumptions are often made, for instance, bounds on the value of any expected reward : for any arm a,

[5] In practice, such an assumption could be discussed, especially for real-life applications. Nonetheless, it still remains an interesting simplification for our drug repurposing problem. Some theoretical work on the case of unknown variance is present in the literature, see for instance Chowdhury et al. (2022).

[6] Familiar examples of σ^2-subgaussian distributions are Gaussian distributions with fixed variance set to σ^2, or bounded distributions.

[7] That is, reducing the number of calls to the Boolean network, in our case.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

$\mu_a \in [0,1]$, or $|\mu_a| \leq 1$.

Linear bandits ($\mathcal{M} = \{x \mapsto \theta^\top x \mid \theta \in \mathbb{R}^d\}$). In this case, we know that the expected reward of an arm a is the product of a linear function of the associated feature vector. As mentioned in the previous paragraph, parameter θ is common to all arms in $[K]$. Such a setting has been studied in the context of regret minimization (Abeille and Lazaric, 2017; Agrawal and Goyal, 2013), but also for best arm identification (Jedra and Proutiere, 2020; Soare, Lazaric, and Munos, 2014; Xu, Honda, and Sugiyama, 2018). We will mention this class of models in Chapter 5. Supplementary assumptions might be that feature vectors span $\mathbb{R}^d$, which is justified by numerical concerns related to matrix singularity.

Remark 4.1.2. *Given all the expected rewards $(\mu_a)_{a\in[K]}$, there is not necessarily a unique θ such that for any arm a, $\mu_a = \theta^\top X_a$, depending on the set of feature vectors $(X_a)_{a\in[K]}$. For instance, considering two arms with respective feature vectors $[1,1]^\top$ and $[0.5,0.5]^\top$ and expected rewards $\mu_1 = 1$ and $\mu_2 = 0.5$, both $\theta = [1,0]^\top$ and $\theta' = [0,1]^\top$ could be considered as valid parameters. Without supplementary assumptions, potentially several distinct functions in the class $\mathcal{M}$ can represent the same model μ (however, a function $f \in \mathcal{M}$ represent a single model). In particular, this is why I chose to parametrize expectations and probabilities with model functions in $\mathcal{M}$ instead of models μ themselves. This underdetermination raises questions, especially for the interpretability of the inferred parameter(s).*

Generalized linear bandits ($\mathcal{M} = \{x \mapsto \ell(\theta^\top x) \mid \theta \in \mathbb{R}^d, \ell$ link function$\}$). Generalized linear models are known beyond the multi-armed bandit field, and comprise of a *link function* –denoted ℓ here– which belongs to the exponential family, *i.e.*, ℓ is any probability distribution parametrized by some scalar $\gamma \in \mathbb{R}$ [8] which density, with respect to a reference measure, is of the following form

$$\mathbb{P}_{\{X\sim\ell\}}[X = x] = h(x)\exp(\Gamma(\gamma) \times T(x) - A(\gamma)) \ ,$$

where h is a function with nonnegative values, and Γ, T, and A (log-partition function) are known functions. Such a family includes the normal, exponential, and Poisson distributions. They have already been studied extensively in the literature (Filippi et al., 2010; Li, Lu, and Zhou, 2017; Russo and Van Roy, 2013). In particular, when $\ell : y \mapsto \frac{1}{1+\exp(-y)}$, the class of models becomes

$$\mathcal{M} = \left\{x \mapsto \frac{1}{1+\exp(-\theta^\top x)} \mid \theta \in \mathbb{R}^d\right\} \ ,$$

[8] We will only consider scalar parametrization here.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

which correspond to logistic bandits (Dumitrascu, Feng, and Engelhardt, 2018; Faury et al., 2020).

The agent can estimate the expected rewards from a given arm by sampling it repeatedly, or, when exploiting feature vectors, by sampling other arms. We denote the optimal arm, that is, the arm yielding the highest expected reward, $a^\star$: $\mu_\star = \mu_{a^\star} := \max_{a\in[K]} \mu_a$. In a similar fashion, we define the Nth best arm, that is, the arm which yields the Nth highest reward $a_{(N)}$ such that $\mu_{(N)} := \overset{N}{\max}_{a\in[K]} \mu_a$. Note that there can be several of them, but we assume in the remaining of the manuscript that it is unique. [9]

Several types of objectives might be satisfied by the agent : for instance, the most commonly studied is the maximization of the cumulative sum of rewards in T rounds, where T is fixed, which is called *regret minimization*. (Bubeck and Cesa-Bianchi, 2012) In that case, the agent has to play a large number of times the estimated optimal arm, from the past observations. Sampling a suboptimal arm incurs a penalty in terms of payoff, while the agent needs to have a good idea of the performance of each arm in the set to correctly estimate the best arm : this is the well-known *exploration-exploitation dilemma*. Inspired by the image of one-armed bandits in casinos –which gave their names to multi-armed bandits– let us try to illustrate this principle : a player enters a casino and considers a row of one-armed bandit machines, numbered from 1 to K. This player yields an average payoff μ_a of winning on machine a, for any $a \in [K]$. That player then wants to win as much as possible with its fixed budget of T tokens. If they knew which was the most rewarding machine, playing it repeatedly would ensure to win the greatest average return. Of course, the owner of the casino would not disclose such an information, which drives the player to sample all machines enough so that they can rule out probably suboptimal arms, while trying to maximize their profits within their limited token budget.

Other types of objectives belong to in the field of *pure exploration* problems, that aim at answering a question about the set of arms, which have also received a lot of attention (Bubeck, Munos, and Stoltz, 2009; Bubeck, Wang, and Viswanathan, 2013; Degenne and Koolen, 2019; Garivier and Kaufmann, 2016; Jun and Nowak, 2016). In our setting of drug repurposing, since we are considering *in silico* simulations, the agent has to explore as much as possible in order to detect which molecules are the best candidates for repurposing. Consider once again the casino example. Let us assume that the player decides to sneak at night into the casino, having stolen a

[9] Because it simplifies the proof of correctness of the algorithms (refer to the next sections). This assumes that the expected rewards of the Nth and the $(N+1)^{\text{th}}$ arms are distinct, which is a reasonable assumption in practice.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

large number of tokens, and plays all night long the K one-armed bandit machines so that to estimate which one is the best. They cannot leave the casino with their payoff this time, but they might escape before sunset with the information about the best arm to sample the next day. This pure exploration setting is the closest to our drug repurposing use case.

Therefore, in my thesis, I studied the design of algorithms $\mathfrak{A}$ which tackle the pure exploration problem of identifying the N best arms with as few observations as possible –called in this manuscript *Top-N identification* [10]. At final round $\tau_{\mathfrak{A}}$, these algorithms return the set $\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}})$, of size N, based on past observations. The number of samples $\tau_{\mathfrak{A}}$ is a random variable which depends on a stopping rule, which ensures that each arm in $\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}})$ belongs to the true set of N best arms with probability higher than $1-\delta$. This is the *fixed-confidence setting*.

Definition 4.1.3. ***Fixed confidence Top-N identification for bandits with structure $\mathcal{M}$.*** *At fixed class $\mathcal{M}$, the goal of fixed-confidence Top-N identification is to design an algorithm $\mathfrak{A}$ which is δ-correct for any $f \in \mathcal{M}$*

$$\mathbb{P}_{\{f\}}\left[\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \not\subseteq \mathcal{S}^\star_{N,f}\right] \leq \delta \ , \ \textit{where } \mathcal{S}^\star_{N,f} := \left\{a \in [K] : f(X_a) \geq \max_{i\in[K]}^{N} f(X_i)\right\} \ .$$

$\mathcal{S}^\star_{N,f}$ is the true set of N best arms (again, considered unique) in model μ associated with function $f \in \mathcal{M}$. I drop the subscript f in the notation $\mathcal{S}^\star_{N,f}$ when we consider a single $\mathcal{M}$-structured model. One might also be interested in the set of (N,ε)-best arms, that is, the set of arms which are among the N-best arms up to some margin $\varepsilon \geq 0$

$$\mathcal{S}^\star_N(\varepsilon) \approx \mathcal{S}^\star_{N,f}(\varepsilon) := \left\{a \in [K] : f(X_a) \geq \max_{i\in[K]}^{N} f(X_i) - \varepsilon\right\} \ . \tag{4.1}$$

In that case, algorithms which satisfy the inequality constraint in Definition 4.1.3 with set $\mathcal{S}^\star_N(\varepsilon)$ are said to be $(\varepsilon,\delta)-$Probably Approximately Correct (PAC) (Valiant, 1984)

Definition 4.1.4. ***(ε,δ,N)-Probably Approximately Correct (PAC) for $\mathcal{M}$-structured models.*** *Algorithm $\mathfrak{A}$ is (ε,δ,N)-PAC* [11] *if and only if, for any $f \in \mathcal{M}$, $\mathfrak{A}$ satisfies*

$$\mathbb{P}_{\{f\}}[\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \not\subseteq \mathcal{S}^\star_{N,f}(\varepsilon)] \leq \delta \ .$$

Then, we aim at minimizing the number of samples $\tau_{\mathfrak{A}}$ among all δ-(PA)Correct algorithms $\mathfrak{A}$, either in high probability (what is proven in Chap-

[10] Sometimes called *best arm identification* in the case where $N=1$.
[11] I will often abuse notation by only mentioning "δ-PAC".

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

ters 5 and 8)

$$\forall f \in \mathcal{M}, \mathbb{P}_{\{f\}}[\tau_{\mathfrak{A}} \leq C(f, \delta, \varepsilon, N)] \leq \delta \,, \tag{4.2}$$

or in expectation (what is proven in Chapter 6)

$$\forall f \in \mathcal{M}, \mathbb{E}_{\{f\}}[\tau_{\mathfrak{A}}] \leq E(f, \delta, \varepsilon, N) \,, \tag{4.3}$$

where C and E are some functions depending on the inputs of the problem.

In practice, as illustrated in the next sections of this chapter, in order to show that an algorithm is $\delta-$PAC, we build a minimally "good" event which is defined as an event $\mathcal{E}$ such that $\mathbb{P}_{\{\mathfrak{A}\}}(\mathcal{E}) \geq 1 - \delta$ and $\left\{\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \subseteq \mathcal{S}^\star_N(\varepsilon)\right\} \subseteq \mathcal{E}$. Then it directly follows that

$$\mathbb{P}\left[\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \not\subseteq \mathcal{S}^\star_N(\varepsilon)\right] \leq \mathbb{P}[(\mathcal{E})^c] \leq \delta \,,$$

where $(\mathcal{E})^c$ is the complementary event to $\mathcal{E}$: given the universe Ω (*i.e.*, the set of all possible events),

$$\mathcal{E} \cap (\mathcal{E})^c = \emptyset \text{ and } \mathcal{E} \cup (\mathcal{E})^c = \Omega \,.$$

Typically, to prove a high-probability sample complexity bound as in Equation (4.2), we prove an upper bound on the total number of samples $\tau_{\mathfrak{A}}$ when this event $\mathcal{E}$ holds, as done in Kalyanakrishnan et al. (2012); Réda, Kaufmann, and Delahaye-Duriez (2021) and in Chapter 8 for instance. Proving an sample complexity upper bound in expectation (Equation (4.3)) is a bit more complex and is done in Degenne and Koolen (2019); Kaufmann and Kalyanakrishnan (2013); Réda, Tirinzoni, and Degenne (2021). The goal is to carefully craft a time-dependent event $\mathcal{E}_t$ which probability exponentially increases as t increases (such that the complementary event seldom happens), combining events that allow the algorithm to stop and be δ-correct. Then, finding a suitable time $T_0 > 0$ which ensures that, for any $t \geq T_0$, $\mathcal{E}_t \subseteq \{\tau_{\mathfrak{A}} \leq t\}$, yields

$$\mathbb{E}[\tau_{\mathfrak{A}}] = \sum_{t=0}^{+\infty} \mathbb{P}(\tau_{\mathfrak{A}} > t) \leq \sum_{t=0}^{T_0-1} 1 + \underbrace{\sum_{t=T_0}^{+\infty} \mathbb{P}((\mathcal{E}_t)^c)}_{\text{finite value E}} \leq \underbrace{T_0}_{\text{to upper bound}} + E \,.$$

Remark 4.1.5. *Note that, in high probability, we do not actually end up with a true upper bound on the total number of samples needed, that is, an upper bound in the worst-case scenario. Indeed, when the "good event" does not hold anymore, it is likely that the algorithm is no longer δ-correct, and we do not control what happens in that case (i.e., the upper bound does not hold anymore). However, Kalyanakrishnan et al. (2012, Theorem 8) showed*

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

that the worst case in Top-N identification for unstructured bandit models yields an upper bound on the number of samples of order $\mathcal{O}(K\varepsilon^{-2}\log(N/\delta))$ (for $0 < \delta < \frac{1}{4}$, $0 < \varepsilon \leq \sqrt{1/32}$, $N \geq 6$ and $K \geq 2N$). That means that, for any bandit algorithm, we could add to the stopping rule the following condition : regardless of the satisfaction of the initial stopping criterion, if the number of samples is higher than $K\varepsilon^{-2}\log(N/\delta)$, then the algorithm should stop and apply its decision rule.

Remark 4.1.6. *In addition to the error probability and the sample complexity, we could also evaluate algorithm $\mathfrak{A}$ on their simple regret (Bubeck, Munos, and Stoltz, 2008; Gabillon, Ghavamzadeh, and Lazaric, 2012), defined as follows for Top-N identification $r_N := \max_{k\in[K]}^{N} \mu_k - \min_{a\in\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}})} \mu_a$. Note that, by definition, that quantity is always non negative ($|\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}})| = N$), and positive if and only if $\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \not\subseteq \mathcal{S}_N^\star$. Then, studying the error probability is estimating the expectation $\mathbb{E}_{\mathfrak{A},\mu}[\mathbb{1}(r_N > \varepsilon)]$. However, in the context of drug repurposing, dealing with recommendation errors is more straightforward, especially in the case where $\varepsilon = 0$.*

4.2 Bandit algorithms for Top-N identification

Any algorithm for Top-N identification [12] comprises of three distinct parts which fully define it. At each round t, algorithm $\mathfrak{A}$ uses a *sampling rule* to select arm I_t, based on past observations $Y_1, Y_2, \ldots, Y_{t-1}$ obtained from arms $I_1, I_2, \ldots, I_{t-1}$ conditioned on some internal randomization $U_1, U_2, \ldots, U_{t-1}$. [13] Then, considering the σ-algebra [14] $\mathcal{F}(t) := \sigma(\{I_s, Y_s, U_s \mid s \leq t\})$ generated by all observations up to round t included, random variable I_t is then $\mathcal{F}(t-1)$-measurable. That is, intuitively, all the information needed to set the value of I_t is contained in the subsets of random variables present in $\mathcal{F}(t-1)$. Algorithm $\mathfrak{A}$ keeps sampling arms until the adaptive *stopping rule* is triggered at round t. As it is adaptive, this stopping rule is a stopping time with respect to the filtration $\mathcal{F}(t)$. This means that the time when the algorithm stops sampling is a random variable which depends on the variables in the filtration which contains all variables of interest collected at the previous time steps. At stopping time $\tau_{\mathfrak{A}}$, algorithm $\mathfrak{A}$ outputs its answer $\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}})$ which is computed

[12] It still holds for any algorithm for pure exploration.

[13] The generation of noisy observations, and selection of arms (for instance, when choosing between two arms maximizing the same criterion) rely on the seed of a pseudo-random generator.

[14] The σ-algebra ("tribu" in French) $\sigma(X)$ associated with a set X is a collection of subsets of X, such that : $X \in \sigma(X)$; if subset $A \in \sigma(X)$, so is $(A)^c$ (closure under complement) ; if a finite number n of subsets A_1, A_2, ..., A_n belong to $\sigma(X)$, then so is $\bigcup_{i\leq n} A_i$ (closure under countable unions) and $\bigcap_{i\leq n} A_i$ (closure under countable intersections).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

from the *decision rule*. Set $\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}})$ is of size N and is $\mathcal{F}(\tau_{\mathfrak{A}})$-measurable. The interactions between these three rules are illustrated in Algorithm 5.

Algorithm 5 Top-N identification. Skeleton of a bandit algorithm for Top-N identification

```
t ← 0
repeat
    # Selection of an arm to sample
    I_t ← sampling_rule({I_s, Y_s, U_s : s < t})
    # Observation of sampled arm I_t drawn from law ν_{I_t}
    Y_t ~ ν_{I_t}
    t ← t + 1
until stopping_rule(t, {I_s, Y_s, U_s : s ≤ t})
# Making a decision for the Top-N set
Ŝ_N(τ_𝔄) ← decision_rule({I_s, Y_s, U_s : s ≤ τ_𝔄})
return Ŝ_N(τ_𝔄)
```

4.3 Sample complexity in bandits

As previously mentioned, the goal in the fixed-confidence setting is to design an algorithm $\mathfrak{A}$ with an (expected) number of samples as small as possible, while keeping the error rate under the threshold δ on a given class of models $\mathcal{M}$. Depending on what we want to achieve (Equation (4.3), resp. Equation (4.2)) $\mathbb{E}_{\{\cdot\}}[\tau_{\mathfrak{A}}]$, resp. an upper bound on $\tau_{\mathfrak{A}}$, is called *sample complexity*, as observations are sometimes called samples. We would like to assess whether our algorithms perform well by checking out the behavior of the sample complexity depending on the expected rewards (when we know them). In particular, consider the extreme case where the difference between the expected rewards of the Nth best arm and of any of the $(K-N)$ worst-performing arms is equal to some small value $\varepsilon > 0$, that is, the minimal gap in performance between the top N arms and the others. It is easy to guess that, as ε decreases, the harder it is to distinguish between these two arms. Then, a large number of observations will be needed to return the correct set of best arms (and vice-versa, had we considered one of the N best arms and the $(N+1)^{\text{th}}$ best one). Hence, we already suspect that the sample complexity needed to return a δ-correct answer crucially depends on the vector of expected rewards μ. As suggested by the previous example, (expected reward) gaps to the Nth or $(N+1)^{\text{th}}$ will play a significant role in the analysis of structured bandit models in Chapter 5. We define the gap associated with any of the arms $a \in [K]$ as follows

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

	Sample complexity bound in high probability
LUCB	$292\mathcal{H}^{\mathsf{LUCB}}(f)\ln\left(\frac{\mathcal{H}^{\mathsf{LUCB}}(f)}{\delta}\right)+16$ where $\mathcal{H}^{\mathsf{LUCB}}(f):=\sum_{a\in[K]}\max\left(\Delta_a,\frac{\varepsilon}{2}\right)^{-2}$
KLLUCB	$2\exp(1)\mathcal{H}^{\mathsf{KL}}(f)\left[\log\left(\frac{6K}{\delta}\right)+2\log\left(\frac{\mathcal{H}^{\mathsf{KL}}(f)}{\delta}\right)\right]$ where $\mathcal{H}^{\mathsf{KL}}(f):=\sum_{a\in[K]}\max\left(\Delta_a^2,(\varepsilon/2)^2\right)^{-2}$
UGapE	$2\mathcal{H}^{\mathsf{UGapE}}(f)\log\left(\frac{K}{\delta}\right)+12\mathcal{H}^{\mathsf{UGapE}}(f)\log\left(K+2\mathcal{H}^{\mathsf{UGapE}}(f)\log\left(\frac{K}{\delta}\right)+6\mathcal{H}^{\mathsf{UGapE}}(f)\right)$ where $\mathcal{H}^{\mathsf{UGapE}}(f):=\sum_{a\in[K]}\max\left(\frac{\Delta_a+\varepsilon}{2},\varepsilon\right)^{-2}$

Table 4.1: **Prior results on the sample complexity for unstructured bandits.** Sample complexity results for unstructured bandits, for any $f:\mathbb{R}^d\to\mathbb{R}$ (unstructured case). References : LUCB (Kalyanakrishnan et al., 2012, Corollary 7), KL-LUCB and KL-Racing (Kaufmann and Kalyanakrishnan, 2013) ($\alpha=2$,$C_0(\alpha)=2\exp(1)$,$k_1=3$), UGapE (Gabillon, Ghavamzadeh, and Lazaric, 2012, Theorem 2) ($b=1$). To find the upper bound with the correct constants for UGapE, I have used Kaufmann, Ménard, et al. (2021, Lemma 15). I have omitted factors in $\mathcal{O}(K)$.

Definition 4.3.1. ***Characteristic individual arm gap for $\mathcal{M}$-structured bandit.*** *For any $f\in\mathcal{M}$ such that, for any arm $k\in[K]$, $\mu_a=f(X_a)$, we define the positive gap quantity for arm k as follows*

$$\Delta_k:=\max\left(\mu_k-\mu_{(N+1)},\mu_{(N)}-\mu_k\right)\ .$$

In particular, it implies that

$$\Delta_k:=\begin{cases}\mu_{(N)}-\mu_k & \text{if } k\notin\mathcal{S}^\star_N(\varepsilon)\ ,\\ \mu_k-\mu_{(N+1)} & \text{otherwise}\ .\end{cases}$$

We further illustrate the importance of arm gaps by considering sample complexity results (in high probability, as in Equation (4.2)) from prior work on Top-N identification in unstructured bandits in Table 4.1.

Moreover, had we have a guess on the lower bound on the sample complexity needed to solve our pure exploration problem on any model function $f\in\mathcal{M}$, and if we managed to upper bound the sample complexity of our algorithm on that model by some quantity not too far from that lower bound, then we could ensure that our algorithm was performant with regards to sample efficiency. There exists an instance-dependent lower bound on the minimum expected number of samples needed to make a δ-PAC decision in a pure exploration problem (Kaufmann, Cappé, and Garivier, 2016, Lemma 1), in the regime of small, finite values of error rate δ. This lower bound relies on the definition of $\mathcal{M}$-structured alternative model functions to f, which are the set of model functions in $\mathcal{M}$ where the set of N best arms differs from $\mathcal{S}^\star_{N,f}$. Intuitively, if there exists an alternative, similarly structured, model function $\widetilde{f}$ which is close enough to f such as $\mathcal{S}^\star_{N,f}\not\subseteq\mathcal{S}^\star_{N,\widetilde{f}}$, then the associated

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

pure exploration problem will be hard, that is, will need a large number of samples to output a probably correct answer. That distance between models is expressed in terms of Kullback-Leibler divergence (Kullback and Leibler, 1951), denoted KL, which is defined as follows

Definition 4.3.2. ***Kullback-Leibler divergence.*** *For a pair of probability densities ν_a and α_a (for a fixed arm $a \in [K]$)*

$$KL(\nu_a, \alpha_a) = D_{KL}(\nu_a \| \alpha_a) := \mathbb{E}_{\{Y\}}\left[\nu_a(Y) \log\left(\frac{\nu_a(Y)}{\alpha_a(Y)}\right)\right] .$$

In particular, for continuous probability densities ν_a and α_a with respect to Lebesgue measure,

$$\mathrm{KL}(\nu_a, \alpha_a) = \int_{-\infty}^{+\infty} \nu_a(x) \log\left(\frac{\nu_a(x)}{\alpha_a(x)}\right) dx .$$

It has a closed-form in the case of Gaussian distributions that is symmetric, contrary to the general definition of Kullback-Leibler divergence in Definition 4.3.2. In the remainder of the manuscript, since we consider a single class of models $\mathcal{M}$ at a time, we will systematically abuse notation by considering model functions $f, \widetilde{f} \in \mathcal{M}$ such that $\mathbb{E}_{\{Y\sim\nu_a\}}[Y] = f(X_a)$ and $\mathbb{E}_{\{Y\sim\alpha_a\}}[Y] = \widetilde{f}(X_a)$, instead of probability laws ν_a, α_a, and then denote

$$\mathrm{KL}_a(f, \widetilde{f}) := \mathrm{KL}(\nu_a, \alpha_a) .$$

The lower bound stated in Kaufmann, Cappé, and Garivier (2016, Lemma 1) is as follows : for any small δ, any δ-PAC algorithm $\mathfrak{A}$ on any model function $f \in \mathcal{M}$ will sample in expectation a number of observations at least as large as some constant $\mathcal{C}^\star(f)$ inflated by a factor $\mathcal{O}(\log(1/\delta))$. Constant $\mathcal{C}^\star(f)$ is called "characteristic time". It depends on f and reflects the intrinsic hardness of the problem in terms of alternative model functions.

Known result 4.3.3. ***Lower bound on the expected number of samples in a pure exploration problem in Kaufmann, Cappé, and Garivier (2016, Lemma 1).*** *For any function $f \in \mathcal{M}$ and any $\delta \leq 1/2$, δ-PAC algorithm $\mathfrak{A}$ satisfies the following inequality*

$$\mathbb{E}_{\{f\}}[\tau_{\mathfrak{A}}] \geq \left(\underbrace{\sup_{\omega\in\triangle_K} \inf_{\widetilde{f}\in Alt(f)} \sum_{a\in[K]} \omega_a KL_a(f, \widetilde{f})}_{=\mathcal{C}^\star(f)}\right)^{-1} \log\left(\frac{1}{2.4\delta}\right) ,$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

where

$$\textit{Alt}(f) := \left\{ \widetilde{f} \in \mathcal{M} \mid \mathcal{S}^\star_{N,f}(\varepsilon) \not\subseteq \mathcal{S}^\star_{N,\widetilde{f}}(\varepsilon) \right\} \text{ , and}$$

$$\triangle_K := \left\{ p \in [0,1]^K \mid \sum_{k \in [K]} p_k = 1 \right\} \text{ is the simplex on } [K] \text{ .}$$

The lower bound in Result (4.3.3) was shown to be rather tight for $\varepsilon = 0$ and Gaussian bandits : indeed, for Gaussian bandits with fixed variance σ^2, the Kullback-Leibler divergence has the following closed-form

$$\forall f, \widetilde{f} \in \mathcal{M} \ \forall a \in [K], \ \mathsf{KL}_a(f, \widetilde{f})^2 = \frac{(f(X_a) - \widetilde{f}(X_a))^2}{2\sigma^2} \text{ .} \tag{4.4}$$

Then Garivier and Kaufmann (2016, Theorem 5) have shown that

$$\mathcal{H}^{\mathsf{opt}}(f) \leq (\mathcal{C}^\star(f))^{-1} \leq 2\mathcal{H}^{\mathsf{opt}}(f) \text{ where } \mathcal{H}^{\mathsf{opt}}(f) := \sum_{a \in [K]} \frac{2\sigma^2}{\Delta_a^2} \text{ ,}$$

which is, up to a factor $\log(1/\mathcal{C}^\star(f))$, what is shown in Table 4.1. However, this lower bound is probably not reachable as a general rule. In order to compare algorithms to this lower bound, the concept of *asymptotic optimality* is defined as the ability of a δ-PAC algorithm $\mathfrak{A}$ of recovering the optimal scaling $(\mathcal{C}^\star(f))^{-1}$ in the asymptotic regime of small values of δ (*i.e.*, δ tends to 0).

Known result 4.3.4. ***Asymptotic optimality for pure exploration bandit algorithms in Degenne and Koolen (2019, Theorem 11).*** *δ-PAC algorithm $\mathfrak{A}$ is asymptotically optimal on $\mathcal{M}$-structured models if and only if for any $f \in \mathcal{M}$*

$$\liminf_{\delta \to 0} \frac{\mathbb{E}_{\{f\}}[\tau_{\mathfrak{A}}]}{\log(1/\delta)} = (\mathcal{C}^\star(f))^{-1} \text{ .}$$

In order to prove such a result for a δ-PAC candidate algorithm $\mathfrak{A}$, we have to exhibit an upper bound on the expected sample complexity $\mathbb{E}_{\{f\}}[\tau_{\mathfrak{A}}]$ for any model function $f \in \mathcal{M}$, which ratio to $\log(1/\delta)$ tends to $(\mathcal{C}^\star(f))^{-1}$ as δ tends to 0. For example, it has been shown in Degenne and Koolen (2019, Theorem 11) that Track-and-Stop (Garivier and Kaufmann, 2016, Section 3), for the best arm identification ($N = 1$) problem, is asymptotically optimal. [15] In Chapter 6, an algorithm is proven asymptotically optimal for a specific structure which encompasses linear bandits.

[15] For best arm identification, Sticky Track-and-Stop (mentioned in Degenne and Koolen (2019, Algorithm 1)) and Track-and-Stop coincide.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

4.4 Integration of bandits to drug repurposing

In my thesis, I considered the integration of multi-armed bandits in drug repurposing. First, in order to reduce the number of calls to the gene regulatory network in Chapter 3. Second, in order to control the error rate (through the formalism of δ-PAC in Definition 4.1.3). Last, in order to infer an explanable model, based on the drug signatures, which are the feature vectors associated with screened molecules. This last point was something rather novel at the beginning of my PhD, since the few papers about Top-N identification were dealing with unstructured bandits (Chen, Li, and Qiao, 2017; Gabillon, Ghavamzadeh, and Lazaric, 2012; Kalyanakrishnan et al., 2012; Kaufmann and Kalyanakrishnan, 2013). Figure 4.2 illustrates the connection between Boolean networks and multi-armed bandits for drug repurposing.

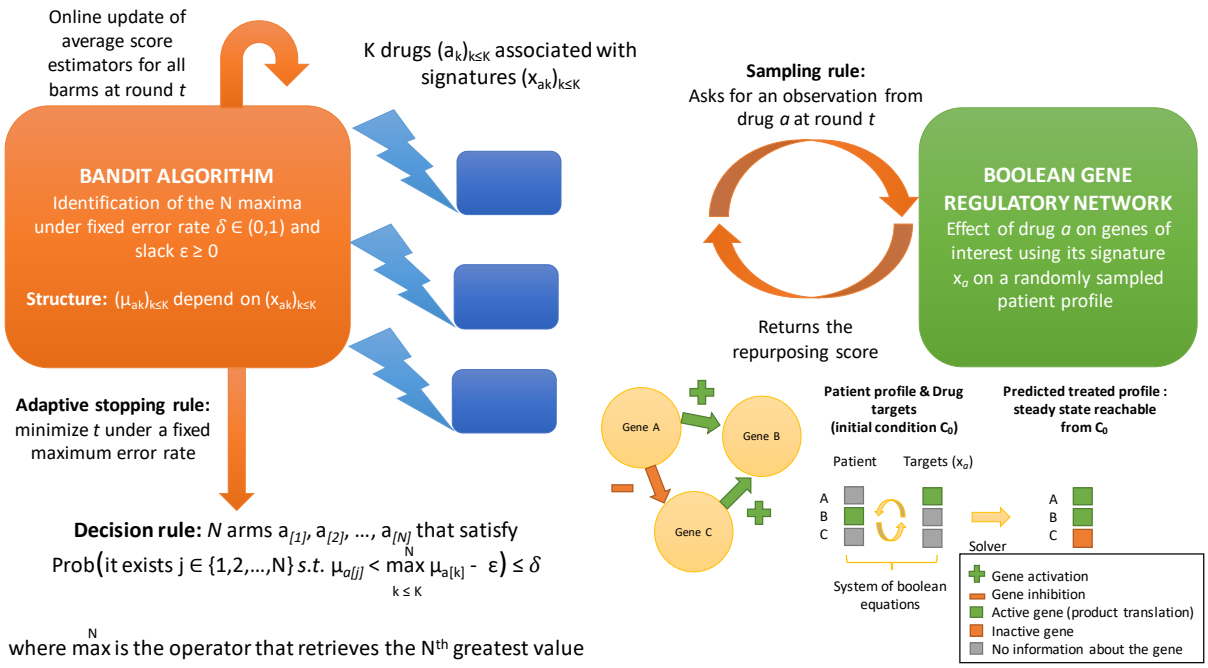

Figure 4.2: **Drug repurposing with bandits.** Illustration of the drug repurposing method integrating both Boolean networks and multi-armed bandits.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 5

Sequential identification in linear models

The most crucial question, with respect to the state-of-the-art at the start of my PhD, was about the integration of feature vectors into the bandit model. Inspired by prior works on structured best-arm identification and regret maximization (Auer, 2002; Fiez et al., 2019; Li, Chu, et al., 2010; Xu, Honda, and Sugiyama, 2018), I first considered the simpler setting of *linear bandits* : bandits where, for any arm $a \in [K]$, the associated expected reward vector μ_a linearly depends on feature vector $X_a \in \mathbb{R}^d$, with a parameter θ shared by all arms

$$\exists \theta \in \mathbb{R}^d \ \forall a \in [K], \ \mu_a = X_a^\top \theta \ .$$

In the terminology introduced in Chapter 4, such models have structure $\mathcal{M}_{\text{lin}}$, where $\mathcal{M}_{\text{lin}} := \left\{x \mapsto \theta^\top x \mid \theta \in \mathbb{R}^d\right\}$ (Definition 4.1.1). Linear models are interesting because they are easily interpretable ; if feature vector $X_a \in \mathbb{R}^d$ corresponds to the gene expression signature of molecule a, where d is the number of considered genes, then parameter θ exactly encodes the genewise contributions to the repurposing score. In this chapter, I will describe my contributions in solving the problem of Top-N identification in linear bandits. In particular, I introduced a set of generic algorithms with adaptive sampling, called gap-index focused algorithms (GIFA), which are $\delta-$PAC (Definition 4.1.3) under weak assumptions about their definition (Section 4.2), and are on par with the state-of-the-art for Top-N identification in terms of empirical performance. This work was published in Réda, Kaufmann, and Delahaye-Duriez (2021) at the 24th Artificial Intelligence and Statistics conference (AISTATS 2021). [1]

[1] Related code is available at https://github.com/clreda/linear-top-m.

Except where otherwise noted, this work is licensed under https://creativecommons.org/licenses/by-nd/3.0/fr/

5.1 Related work

Roughly two types of fixed-confidence algorithms have been proposed for Top-N identification in an unstructured bandit :

- based on adaptive sampling such as LUCB (Kalyanakrishnan et al., 2012), or UGapE (Gabillon, Ghavamzadeh, and Lazaric, 2012).

- based on uniform sampling and eliminations (Chen, Li, and Qiao, 2017; Kaufmann and Kalyanakrishnan, 2013).

At the time of publication, the only efficient algorithms for linear bandits were proposed for the best arm identification (BAI) problem, which corresponds to $N = 1$. This setting, first investigated by Soare, Lazaric, and Munos (2014), recently received a lot of attention. An efficient adaptive sampling algorithm called LinGapE was proposed by Xu, Honda, and Sugiyama (2018). Subsequent elimination-based works such as Fiez et al. (2019); Hassidim, Kupfer, and Singer (2020) sought to achieve the minimal sample complexity. In particular, the LinGame algorithm of Degenne, Ménard, et al. (2020) was shown to exactly achieve the problem-dependent sample complexity lower bound for linear BAI, in the regime in which δ goes to zero (Result 4.3.4). We note that, in principle, LinGame can be used for any pure exploration problem in a linear bandit, which includes Top-N identification for $N > 1$. However, this algorithm relies on a game theoretic formalism which needs the computation of a best response for Nature in response to the player's selection ; a computable expression of this strategy was (at that time) not available to our knowledge for Top-N ($N > 1$). Besides, naively computing the information-theoretic lower bound for Top-N identification is computationally hard. [2]

These remarks led us to investigate efficient adaptive sampling algorithms for Top-N identification ($N \geq 1$) in linear bandits, which are still missing in the literature, instead of trying to propose asymptotically optimal algorithms, as done in linear BAI. First, by carefully looking at existing adaptive sampling bandits for unstructured Top-N identification, we propose a generic algorithm structure based on *gap indices*, called gap-index focused algorithms (GIFA), which encompasses existing adaptive algorithms for unstructured Top-N identification and linear BAI. This structure allows a higher order and modular understanding of the learning process, and correctness properties can readily be inferred from a *partially specified* bandit algorithm. It allows us to define two interesting new algorithms, called N-LinGapE and LinGIFA. Then, we present a unified sample complexity analysis of a

[2] Note that we tackled both issues in Chapter 6.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

subclass of GIFA which comprises existing methods. Finally, we empirically show that N-LinGapE and LinGIFA perform better than their counterparts for unstructured bandits, which showcases the importance of side information in sample efficiency.

5.2 General structure for efficient algorithms

As mentioned in Definition 4.1.1, we consider one-dimensional subgaussian distributions of fixed variance $\sigma^2 > 0$, which are uniquely defined by their vector of expected rewards per arm. At each round t, the agent observes $Y_t \sim \nu_{I_t}$ from pulled arm I_t. We solve Top-N identification with error rate $\delta \in (0,1)$ and any margin $\varepsilon \geq 0$, that is, for a model function $f \in \mathcal{M}_{\text{lin}}$

$$\text{find } \mathcal{S}_N^\star(\varepsilon) := \left\{ a \in [K] \mid f(X_a) \geq \max_{i \in [K]}^{N} f(X_i) - \varepsilon \right\} .$$

Estimation of the linear structure

The learner needs to estimate the unknown linear parameter $\theta \in \mathbb{R}^d$, which can be done *via* a (regularized) least-squares estimator, defined as follows. For any arm $a \in [K]$ and at round t, we define the number of times a is sampled up to round t included as

$$N_a(t) := \sum_{s \leq t} \mathbb{1}(I_s = a) ,$$

and the κ-regularized design matrix and least-squares estimate of θ as

$$\hat{V}^\kappa(t) := \kappa I_d + \sum_{a \in [K]} N_a(t) X_a X_a^\top \text{ and } \hat{\theta}(t) := \left(\hat{V}^\kappa(t)\right)^{-1} \left(\sum_{s \leq t} Y_s X_{I_t} \right) .$$

Note that matrix $\sigma^2 \left(\hat{V}^\kappa(t)\right)^{-1}$ can be interpreted as the posterior covariance in a Bayesian linear regression model, in which the covariance of the prior is $\sigma^2/\kappa I_d$.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Introduction to gap indices

Any adaptive sampling-based algorithm mentioned in Section 5.1 –along with the GIFA family that we introduce– crucially relies on estimating the gaps $(\Delta_{i,j})_{(i,j)}$ between pairs of arms $(i,j) \in [K]^2$ for any model $f \in \mathcal{M}_{\text{lin}}$

$$\forall i \in [K]\ \forall j \in [K],\ \Delta_{i,j} := f(X_i) - f(X_j)\ .$$

One way to achieve this estimation is by building upper confidence bounds (UCBs) on these quantities ; *i.e.*, indices $(U_{i,j}(t))_{i,j,t}$ such that the following inequalities all hold with high probability

$$\forall i \in [K]\ \forall j \in [K]\ \forall t > 0,\ \Delta_{i,j} \leq U_{i,j}(t)\ .$$

For this purpose, we respectively introduce the empirical arm-individual mean and the empirical arm-pairwise gap at round t

$$\forall a \in [K],\ \widehat{\mu}_a(t) := \left(\hat{\theta}(t)\right)^\top X_a \text{ and } \forall (a,b) \in [K]^2,\ \hat{\Delta}_{a,b}(t) := \widehat{\mu}_a(t) - \widehat{\mu}_b(t)\ .$$

A first option to build UCBs on pairwise gaps $(\Delta_{i,j})_{i,j}$ consists in building individual confidence intervals –that is, lower and upper bounds for $\Delta_{i,j}$ for any i,j– on the mean of each arm $a \in [K]$. These (symmetrical) confidence intervals are of the form $\{\widehat{\mu}_a(t) \pm \mathcal{W}_a(t)\}$, where $\mathcal{W}_a(t) := \sigma^2 \mathcal{T}_\delta(t) \|X_a\|_{(\hat{V}^\kappa(t))^{-1}}$, for some threshold function $\mathcal{T}_\delta(\cdot)$ to be specified later. $\|\cdot\|_M$ is the Malahanobis norm for any positive definite matrix M, such that, for any vector x, $\|x\|_M := \sqrt{x^\top M x}$. For a well-chosen $\mathcal{T}_\delta(\cdot)$, and for any pair of arms (i,j) and round t, the following quantity

$$\mathcal{B}^{\text{ind}}_{i,j}(t) := \hat{\Delta}_{i,j}(t) + \mathcal{W}^{\text{ind}}_{i,j}(t) \text{ where } \mathcal{W}^{\text{ind}}_{i,j}(t) := \mathcal{W}_i(t) + \mathcal{W}_j(t)$$

is an upper bound on $\Delta_{i,j}$ ("individual" UCB). Yet, using the linear model, one can also directly build a "paired" UCB on the difference

$$\mathcal{B}^{\text{pair}}_{i,j}(t) := \hat{\Delta}_{i,j}(t) + \mathcal{W}^{\text{pair}}_{i,j}(t) \text{ where } \mathcal{W}^{\text{pair}}_{i,j}(t) := \sigma^2 \mathcal{T}_\delta(t) \|X_i - X_j\|_{(\hat{V}^\kappa(t))^{-1}}\ .$$

Both constructions lead to symmetrical bounds, called *gap indices*, of the general form $\mathcal{B}_{i,j}(t) := \hat{\Delta}_{i,j}(t) + \mathcal{W}_{i,j}(t)$. To sum it up,

Definition 5.2.1. ***(Symmetrical) gap index.*** *Let us define $\mathcal{B}_{\cdot,\cdot}(\cdot) = (\mathcal{B}_{i,j}(t))_{i,j,t}$ so that $\mathcal{B}_{i,j}(t) := \hat{\Delta}_{i,j}(t) + \mathcal{W}_{i,j}(t)$ for any arms (i,j), where $\mathcal{W}_{\cdot,\cdot}(\cdot) \in ([K] \times [K] \times \mathbb{N}^* \mapsto \mathbb{R}^+)$. $\mathcal{B}_{\cdot,\cdot}(\cdot)$ is a gap index if, with probability $1-\delta$*

$$\forall i \in [K]\ \forall j \in [K]\ \forall t > 0,\ \Delta_{i,j} \leq \mathcal{B}_{i,j}(t)\ .$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Remark 5.2.2. *As shown in Section 5.2, in order to ensure that an algorithm using gap indices is δ-PAC, we only need those upper confidence bounds to hold with probability $1-\delta$ for specific gaps, namely the gaps between the true good arms and arms which are not in the set of N best arms up to ε*

$$\forall j \in (\mathcal{S}_N^\star(\varepsilon))^c \ \ \forall i \in \mathcal{S}_N^\star \ \forall t > 0, \ \Delta_{i,j} \leq \mathcal{B}_{i,j}(t) \ .$$

However, to upper bound the sample complexity, we will need Definition 5.2.1.

This generic form of gap index allows us to study and compare in a modular way the gain (or loss) in performance incurred by different definitions. For instance, let us compare individual and paired indices. Paired indices may indeed increase sample-efficiency. First, it directly follows from the triangular inequality for the Mahalanobis norm $\|\cdot\|_{(\hat{V}^\kappa(t))^{-1}}$ that

$$\forall (i,j) \in [K]^2 \ \forall t > 0, \ \mathcal{W}_{i,j}^{\text{pair}}(t) = \mathcal{W}_{i,j}(t) \leq \mathcal{W}_i(t) + \mathcal{W}_j(t) = \mathcal{W}_{i,j}^{\text{ind}}(t) \ .$$

Therefore, if both types of gap indices use the same threshold function $\mathcal{T}_\delta(\cdot)$, paired indices are smaller

$$\mathcal{B}_{i,j}^{\text{pair}}(t) = \hat{\Delta}_{i,j}(t) + \mathcal{W}_{i,j}^{\text{pair}}(t) \leq \hat{\Delta}_{i,j}(t) + \mathcal{W}_{i,j}^{\text{ind}}(t) = \mathcal{B}_{i,j}^{\text{ind}}(t) \ .$$

Moreover, the following lemma implies that paired or individual indices using arm features can yield smaller bounds on the gaps than individual indices which do not exploit the structure. This is proven in Réda, Kaufmann, and Delahaye-Duriez (2021, Lemma 2) by induction, combining the Sherman-Morrison formula and the Cauchy-Schwartz inequality.

Remark 5.2.3. *Note that, in the unstructured case, one can consider the vectors of the canonical basis of $\mathbb{R}^K$ as feature vectors : $X_a = (\mathbb{1}(i=a))_{i\in[K]}$. As a consequence, for any round t, $\hat{V}^\kappa(t) = \text{diag}(N_1(t)+\kappa, N_b(t)+\kappa, \ldots, N_K(t)+\kappa)$, and $\|X_a\|_{(\hat{V}^\kappa(t))^{-1}} = 1/\sqrt{N_a(t)+\kappa}$.*

Lemma 5.2.4. ***Individual versus paired gap indices.*** *The following inequality holds*

$$\forall t > 0 \ \forall a \in [K] \ \forall y \in \mathbb{R}^d, \ \|y\|_{(\hat{V}^\kappa(t))^{-1}} \leq \|y\|_2 / \left(\sqrt{N_a(t)\|X_a\|_2^2 + \kappa}\right) \ .$$

In particular, when $\kappa = 0$, for any $t > 0$, and arms $a, b \in [K]$

$$\|X_a - X_b\|_{(\hat{V}^\kappa(t))^{-1}} \leq \|X_a\|_{(\hat{V}^\kappa(t))^{-1}} + \|X_b\|_{(\hat{V}^\kappa(t))^{-1}} \leq 1/\sqrt{N_a(t)} + 1/\sqrt{N_b(t)} \ .$$

As we will see in the next sections, the gap index plays an important part in the stopping rule, and the tighter it is, the faster the stopping time will be

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

reached. We will show in Section 5.4 that empirically, the choice of paired indices over individual ones considerably speeds up the sampling phase. We will describe later how to build these gap indices in practice.

Remark 5.2.5. *For any symmetrical gap index $\mathcal{B}_{\cdot,\cdot}(\cdot)$, we actually get a full confidence interval for free. Indeed, for any pair of arms i, j and round t, the following inequalities hold*

$$-\mathcal{B}_{j,i}(t) \leq \Delta_{i,j} \leq \mathcal{B}_{i,j}(t) \ .$$

Gap-index focused algorithms (GIFA)

Algorithm 6 Structure of GIFA. GIFA for (ε, δ)-PAC Top-N identification, with four undefined rules `stopping`, `compute_Jt`, `compute_bt` and `selection` and gap indices.

$t \leftarrow 1$
For unstructured bandits
`initialization()`
while `stopping`$\big((\mathcal{B}_{i,j}(t))_{i,j\in[K]^2}; \varepsilon, t\big) > \varepsilon$ **do**
  # $J(t)$ is the set of estimated N best arms at t
  $J(t) \leftarrow$ `compute_Jt`$((\mathcal{B}_{i,j}(t))_{i,j}, (\widehat{\mu}_i(t))_i)$
  # $b(t)$ is the estimated N^{th} best arm $a_{(N)}$ and $c(t)$ its challenger at t
  $b(t) \leftarrow$ `compute_bt`$\Big(J(t), (\mathcal{B}_{i,j}(t))_{i,j}\Big)$ and $c_t \leftarrow \arg\max_{a \notin J(t)} \mathcal{B}_{a,b(t)}(t)$
  # Selection of the sampled arm I_t
  $I_t \leftarrow$ `selection`$\Big(b(t), c(t); \hat{V}^{\kappa}(t-1)\Big)$
  $Y_t \sim \nu_{I_t}$
  Update design matrix $\hat{V}^{\kappa}(t)$, means $(\widehat{\mu}_i(t))_i$ (if needed), $(\mathcal{B}_{i,j}(t+1))_{i,j}$
  $t \leftarrow t+1$
end while
$\hat{\mathcal{S}}_N(\tau_{\text{GIFA}}) \leftarrow J(\tau_{\text{GIFA}})$
return $\hat{\mathcal{S}}_N(\tau_{\text{GIFA}})$

The gap indices introduced in Subsection 5.2 allow us to introduce a generic family of algorithms that encompasses the state-of-the-art in adaptive sampling-based algorithms for BAI and Top-N identification. Using this generic structure, we can easily extend an efficient BAI algorithm, LinGapE, to Top-N identification : this extension is called N-LinGapE. We also define another algorithm, called LinGIFA, for Top-N identification which only needs to keep and track their (symmetrical) gap indices, regardless of their actual definition. We will discuss the consequences of this flexibility later in this section. In this section, unless specified, we consider any type of symmetrical gap index, as defined in Definition 5.2.1.

Algorithm 6 sums up the idea behind GIFA. The principle of GIFA is to estimate in each round t a set of candidate N best arms, denoted by $J(t)$,

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

GIFA	compute_Jt	compute_bt	selection	stopping
LUCB	$\arg\max_{j\in[K]}^{[N]} \widehat{\mu}_j(t)$	$\arg\max_{j\in J(t)} \max_{i\notin J(t)} \mathcal{B}_{i,j}(t)$	$\{b(t), c(t)\}$	τ^{LUCB}
UGapE	$\arg\min_{j\in[K]}^{[N]} \max_{i\neq j}^{N} \mathcal{B}_{i,j}(t)$	$\arg\max_{j\in J(t)} \max_{i\notin J(t))} \mathcal{B}_{i,j}(t)$	$\arg\max_{a\in\{b(t),c(t)\}} \mathcal{W}_a(t)$	τ^{UGapE}
LinGapE	$\arg\max_{j\in[K]} \widehat{\mu}_j(t)$	$J(t)$	greedy, optimized	τ^{LUCB}
N**-LinGapE**	$\arg\max_{j\in[K]}^{[N]} \widehat{\mu}_j(t)$	$\arg\max_{j\in J(t)} \max_{i\notin J(t)} \mathcal{B}_{i,j}(t)$	greedy, optimized	τ^{LUCB}
LinGIFA	$\arg\min_{j\in[K]}^{[N]} \max_{i\neq j}^{N} \mathcal{B}_{i,j}(t)$	$\arg\max_{j\in J(t)} \max_{i\neq j}^{N} \mathcal{B}_{i,j}(t)$	greedy, $\arg\max_{a\in\{b(t),c(t)\}} \mathcal{W}_a(t)$	τ^{UGapE}

Table 5.1: **Adaptive sampling-based algorithms for linear Top-N identification.** Adaptive sampling-based algorithms for (ε, δ)-PAC Top-N identification (our proposals are in bold type ; except for LUCB and UGapE, all algorithms use paired indices instead of individual indices). References : LUCB (Kalyanakrishnan et al., 2012), UGapE (Gabillon, Ghavamzadeh, and Lazaric, 2012), LinGapE (Xu, Honda, and Sugiyama, 2018). Greedy and optimized sampling rules are defined in the main text.

and to select the two most ambiguous arms : $b(t) \in J(t)$, which can be viewed as a guess for the Nth-best arm $a_{(N)}$, and a challenger $c(t) \notin J(t)$. $c(t)$ is defined as a potentially misassessed $\mu_{(N)}$. $c(t)$ is defined as having the largest possible gap index to $b(t)$ among the estimated $(K-N)$ worst arms

$$c(t) := \arg\max_{c\in[K]} \mathcal{B}_{c,b(t)}(t) \ .$$

The idea of using two ambiguous arms goes back to the LUCB algorithm (Kalyanakrishnan et al., 2012) for Top-N identification in unstructured bandits. [3] Selected arm at round t I_t should help discriminate between $b(t)$ and $c(t)$. A naive idea is to either draw $b(t)$ or $c(t)$, but alternative selection rules will be discussed later. At the end of the learning phase, at stopping time τ_{GIFA}, the final set $\hat{\mathcal{S}}_N(\tau_{\text{GIFA}})$ is equal to $J(\tau_{\text{GIFA}})$. The part denoted `initialization` is an optional phase where all arms are sampled once. [4] We assume that ties are randomly broken. The yet unspecified parts of the bandit algorithm remain in the choice of the rules `compute_Jt` (definition of $J(t)$), `compute_bt` (definition of $b(t)$), selection rule (choice of I_t), and the definition of the stopping rule `stopping`. Some of these rules may also rely on the gap indices.

[3] That is, when $\mathcal{M}$ is the entire set of functions mapping from $\mathbb{R}^d$ to $\mathbb{R}$, see Definition 4.1.1.

[4] Which is a mandatory step for unstructured bandits in order not to sample at random in the first rounds, but might be ignored for structured ones, provided $\kappa > 0$.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Rules `compute_Jt`, `compute_bt`**.** Table 5.1 presents some instanciations of the GIFA structure presented in Algorithm 6. Two types of algorithms can be distinguished :

- the LUCB-GIFA type, which most intuitively computes $J(t)$ as the empirical set of $\mu_{(N)}$ arms and $b(t)$ as the arm in $J(t)$ closest to $[K] \setminus J(t)$ in terms of (estimated) expected reward. It comprises of algorithms LUCB (Kalyanakrishnan et al., 2012), LinGapE (Xu, Honda, and Sugiyama, 2018), and N-LinGapE.

- the Gap-GIFA type. This subclass of GIFA includes the unstructured bandit algorithm UGapE (Gabillon, Ghavamzadeh, and Lazaric, 2012) and LinGIFA.

To better understand the rules associated with Gap-GIFA, we first prove the following result for any gap index

Lemma 5.2.6. ***Upper bound on the gap to*** $a_{(N)}$**.** *If $\mathcal{B}_{\cdot,\cdot}(\cdot)$ is a gap index in the sense of Definition 5.2.1, then with probability $1-\delta$*

$$\forall t > 0 \ \forall a \in [K], \ \mu_{(N)} - \mu_a \leq \max_{b \in [K]}^{N} \mathcal{B}_{b,a}(t) \ .$$

Proof. Except for some degenerate cases where two arm features are equal and the past observations made from both arms are exactly the same, we can assume that, at fixed arm $a \in [K]$ and round $t > 0$, values $(\mathcal{B}_{b,a}(t))_{b \in [K]}$ are distinct. Then, assume towards contradiction that there exists a round $t > 0$ and an arm $i \in [K]$ such that

$$\max_{j \in [K]}^{N} \mathcal{B}_{j,i}(t) < \mu_{(N)} - \mu_i = \Delta_{a_{(N)},i} \ .$$

Then, using the definition of gap index in Definition 5.2.1, with probability greater than $1-\delta$, for any $k \in \mathcal{S}_N^\star = \arg\max_{a \in [K]}^{N} \mu_a$

$$\mathcal{B}_{k,i}(t) \geq \Delta_{k,i} \geq \Delta_{a_{(N)},i} > \max_{j \in [K]}^{N} \mathcal{B}_{j,i}(t) \ .$$

That means that at least N distinct values of $(\mathcal{B}_{j,i}(t))_{j \in [K]}$ are strictly greater than $\max_{j \in [K]}^{N} \mathcal{B}_{j,i}(t)$, which contradicts the definition of operator $\max_{a \in [K]}^{N}$. □

Remark 5.2.7. *In UGapE, which only uses individual indices, the following result (Gabillon, Ghavamzadeh, and Lazaric, 2012, Lemma 1) was shown*

$$\forall t > 0 \ \forall a \in [K], \ \mu_{(N)} - \mu_a \leq \max_{b \neq a}^{N} \mathcal{B}_{b,a}(t) \ .$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

However, it only holds for general gap indices when $a \notin \mathcal{S}_N^\star$ (see the previous proof). Nonetheless, for any Gap-GIFA $\mathfrak{A}$, at any time $t < \tau_{\mathfrak{A}}$

$$\max_{c \neq b(t)}^{N} \mathcal{B}_{c,b(t)}(t) > 0 \ ,$$

meaning that if $b(t) \in \mathcal{S}_N^\star$, then $\max_{c \neq b(t)}^{N} \mathcal{B}_{c,b(t)}(t) > 0 \geq \mu_{(N)} - \mu_{b(t)}$ which makes the lemma hold in that important case, which justifies the stopping rule, as shown below.

Then, in Gap-GIFA, at round t, the set $J(t)$ is the set of arms which *true* expected rewards are more likely to minimize the (signed) distance to $\mu_{(N)}$. In a similar fashion, arm $b(t)$ is the arm which *true* expected reward $\mu_{b(t)}$ is likely to maximize the distance to $\mu_{(N)}$ among arms in $J(t)$. Then $b(t)$ is a proxy for arm $a_{(N)}$. Indeed, using the definition of $b(t)$ in LinGIFA, and Lemma 5.2.6

$$\max_{j \in J(t)} \max_{i \neq j}^{N} \mathcal{B}_{i,j}(t) = \max_{i \neq b(t)}^{N} \mathcal{B}_{i,b(t)}(t) \geq \mu_{(N)} - \mu_{b(t)} \ .$$

Stopping times. We restrict our attention to two stopping times already proposed for the LUCB (Kalyanakrishnan et al., 2012) and UGapE (Gabillon, Ghavamzadeh, and Lazaric, 2012) algorithms, respectively

$$\tau^{\text{LUCB}} := \inf \left\{ t \in \mathbb{N}^* \mid \mathcal{B}_{c(t),b(t)}(t) \leq \varepsilon \right\} \text{ and } \tau^{\text{UGapE}} := \left\{ t \in \mathbb{N}^* \mid \max_{j \in J(t)} \max_{i \neq j}^{N} \mathcal{B}_{i,j}(t) \leq \varepsilon \right\} \ .$$

Why do these stopping times work? Let us deconvolute their expression.

τ^{LUCB} relies on quantity $\mathcal{B}_{c(t),b(t)}(t)$, which is defined in all algorithms using stopping time τ^{LUCB} as

$$\mathcal{B}_{c(t),b(t)}(t) = \max_{j \in J(t)} \max_{i \notin J(t)} \mathcal{B}_{i,j}(t) \geq \max_{j \in J(t)} \max_{i \notin J(t)} \Delta_{i,j} = \max_{i \notin J(t)} \mu_i - \min_{j \in J(t)} \mu_j \ ,$$

by definition of $b(t)$ and $c(t)$, and using the fact that $\mathcal{B}_{\cdot,\cdot}(\cdot)$ is a gap index. If the algorithm correctly identifies $J(\tau^{\text{LUCB}}) = \mathcal{S}_N^\star(\varepsilon)$ at stopping time, then

$$\varepsilon \geq \mathcal{B}_{c(\tau^{\text{LUCB}}),b(\tau^{\text{LUCB}})}(\tau^{\text{LUCB}}) \geq \mu_{(N+1)} - \mu_{(N)} \ .$$

So stopping time τ^{LUCB} actually tracks the value $\mu_{(N+1)} - \mu_{(N)}$, that is, the separation (*i.e.*, the minimum gap) between arms in $J(t)$ and in $[K] \setminus J(t)$ at each round t.

What about τ^{UGapE}? Using Gabillon, Ghavamzadeh, and Lazaric (2012, Lemma 1) for individual indices at stopping time τ^{UGapE}, if $J(\tau^{\text{UGapE}}) = \mathcal{S}_N^\star(\varepsilon)$,

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

then

$$\varepsilon \geq \max_{j \in \mathcal{S}_N^\star(\varepsilon)} \max_{i \neq j}^{N} \mathcal{B}_{i,j}(\tau^{\text{UGapE}}) \geq \mu_{(N)} - \min_{j \in \mathcal{S}_N^\star(\varepsilon)} \mu_j = 0 \ ,$$

that is, stopping time τ^{UGapE} tracks the distance of the worst-performing arm in $J(t)$ to $a_{(N)}$ at each round t. Now, intuitively, since this distance can reach 0 –contrary to what is tracked by τ^{LUCB}– it seems that this stopping rule τ^{UGapE} stops earlier than τ^{LUCB}, given the same past observations. This claim can be proven for any gap index

Lemma 5.2.8. ***τ^{UGapE} stops earlier than τ^{LUCB}.*** *For any $t > 0$, for any subset $J \subseteq [K]$ of size N, for any $j \in J$, and any values $(\mathcal{B}_{i,j}(t))_{i,j}$*

$$\max_{i \neq j}^{N} \mathcal{B}_{i,j}(t) \leq \max_{i \notin J} \mathcal{B}_{i,j}(t) \ , \ \textit{which implies} \ \max_{j \in J} \max_{i \neq j}^{N} \mathcal{B}_{i,j}(t) \leq \max_{j \in J} \max_{i \notin J} \mathcal{B}_{i,j}(t) \ .$$

Proof. Indeed, for any round $t > 0$ and arm $j \in J$

$$\max_{i \neq j}^{N} \mathcal{B}_{i,j}(t) = \min_{\substack{S \subseteq [K] \\ |S| = N-1}} \max_{i \notin (S \cup \{j\})} \mathcal{B}_{i,j}(t) \ ,$$

since the set S matching the outer bound is $\arg\max_{i \in [K]}^{[N-1]} \mathcal{B}_{i,j}(t)$, meaning that we consider the maximum value over the set of $(\mathcal{B}_{i,j}(t))_{t>0, i \in [K]}$ from which the $N-1$ largest values (and $\mathcal{B}_{j,j}(t)$, if j does not already belong to $\arg\max_{i \in [K]}^{[N-1]} \mathcal{B}_{i,j}(t)$) are removed. Then consider $S = J \setminus \{j\} \subseteq [K]$ of size $N-1$ (since $j \in J$). Then

$$\max_{i \in [K]}^{N} \mathcal{B}_{i,j}(t) \leq \max_{i \notin ((J \setminus \{j\}) \cup \{j\})} \mathcal{B}_{i,j}(t) = \max_{i \notin J} \mathcal{B}_{i,j}(t) \ .$$

□

We will empirically compare these two stopping rules in Section 5.4.

Selection rule. Besides the gap indices, this is the part where all information about the structure is concentrated. Regarding the selection rules shown in Table 5.1, Gap-GIFA algorithms select the least sampled arm among $b(t)$ and $c(t)$, which coincides with the following rule that we propose for a general setting, that is, at round t

$$I_t \leftarrow \arg\max_{a \in \{b(t), c(t)\}} \mathcal{W}_a(t) \ , \ \text{(largest variance)} \tag{5.1}$$

which is still defined even when considering paired indices. In the original version of LUCB (Kalyanakrishnan et al., 2012), both arms $b(t)$ and $c(t)$ are sampled at time t, but the analysis that we propose in this paper obtains similar guarantees for LUCB using the largest variance rule, so we only

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

consider this selection rule in the remainder of the chapter. In LinGapE (Xu, Honda, and Sugiyama, 2018), authors propose two different selection rules to possibly sample another arm that would reduce at round t the variance on the estimation of gap $\Delta_{c(t),b(t)}$

$$I_t \leftarrow \operatorname*{arg\,min}_{a\in[K]} \|X_{c(t)} - X_{b(t)}\|_{(\hat{V}^\kappa(t-1)+X_a X_a^\top)^{-1}} \ , \text{ (greedy)} \tag{5.2}$$

$$\text{and } I_t \leftarrow \operatorname*{arg\,max}_{\substack{a\in[K] \\ \omega_a^\star(b(t),c(t))>0}} N_a(t)\frac{\|\omega^\star(b(t),c(t))\|_1}{|\omega_a^\star(b(t),c(t))|} \ , \text{ (optimized)} \tag{5.3}$$

where $\omega^\star(b(t),c(t)) \in \mathbb{R}^K$ is a minimizer to the following optimization problem

$$\min_{\omega\in\mathbb{R}^K} \|\omega\|_1 \text{ s.t. } X_{b(t)} - X_{c(t)} = \omega X \ . \tag{5.4}$$

As explained by Xu, Honda, and Sugiyama (2018), these two rules are meant to bring the empirical proportions of arm selections close to an optimal design $(\omega_1, \omega_2, \ldots, \omega_K)$ which asymptotically minimizes quantity $\|X_{b(t)} - X_{c(t)}\|_{(\sum_{a\in[K]}\omega_a X_a X_a^\top)^{-1}}$. This quantity is present in the lower bound for best arm identification in linear bandits (Fiez et al., 2019).

Novel algorithms. As previously mentioned, we introduce two new algorithms : an extension of LinGapE (Xu, Honda, and Sugiyama, 2018) to Top-N identification named N-LinGapE, and a new algorithm of type Gap-GIFA named LinGIFA, with a completely new rule for the computation of arm $b(t)$. The strength of LinGIFA is that it is completely defined in terms of gap indices, and, as such, may be improved by deriving tighter bounds on the gaps.

Moreover, we emphasize that, provided that the regularizing constant κ in the design matrix $\hat{V}^\kappa(t)$ is positive at any round t, both LinGapE and LinGIFA can be run without an initialization phase. This permits to avoid the initial sampling cost when the number of arms is large that was noticed by Fiez et al. (2019). However, both algorithms still need to compute maximizers or minimizers over the whole set of arms during the learning rounds (for instance, when computing set $J(t)$). This might be mitigated by considering only a subset of arms when computing these rules, as mentioned in the experimental part of Réda, Tirinzoni, and Degenne (2021).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

5.3 Theoretical guarantees in GIFA

In this section, we present theoretical guarantees on GIFA. The fact that these algorithms are (ε, δ)-PAC (Definition 4.1.4) is a consequence of generic correctness guarantees that can be obtained for partially specified GIFA algorithms, which rely on the definition of gap indices (Definition 5.2.1). We further analyze the sample complexity of LUCB-GIFA algorithms, for instance N-LinGapE, in a unified proof.

Correctness analysis

We justify that the two stopping rules τ^{LUCB} and τ^{UGapE} introduced in Section 5.2 lead to (ε, δ)-PAC algorithms, in the sense of Definition 4.1.4. This holds provided the following condition on the gap indices

Definition 5.3.1. ***Good gap indices.*** *Let us denote*

$$\mathcal{E}^{GIFA} := \bigcap_{t>0} \bigcap_{i \notin \mathcal{S}_N^\star(\varepsilon)} \bigcap_{j \in \mathcal{S}_N^\star(\varepsilon)} \{\mathcal{B}_{j,i}(t) \geq \Delta_{j,i}\} \ .$$

Then a good choice of gap indices $(\mathcal{B}_{i,j}(t))_{i,j,t>0}$ satisfies $\mathbb{P}[\mathcal{E}^{GIFA}] \geq 1 - \delta$.

We will show that such a choice of good gap indices exist, and any symmetrical gap index as in Definition 5.2.1 will satisfy this constraint. First, we observe that on event $\mathcal{E}^{\text{GIFA}}$, for any GIFA $\mathfrak{A}$ (Algorithm 6), both stopping rules τ^{LUCB} and τ^{UGapE} output a (ε, δ)-correct answer.

Theorem 5.3.2. *On event $\mathcal{E}^{GIFA}$, if $\mathfrak{A}$ is from the family GIFA, then*
LUCB-GIFA. *If $b(t) := \arg\max_{j \in J(t)} \max_{i \notin J(t)} \mathcal{B}_{i,j}(t)$, then $J(\tau^{LUCB}) = \hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \subseteq \mathcal{S}_N^\star(\varepsilon)$.*
Gap-GIFA. *If $b(t) \in J(t)$, then $J(\tau^{UGapE}) = \hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \subseteq \mathcal{S}_N^\star(\varepsilon)$.*

Proof. Assume towards contradiction that at stopping time $\tau_{\mathfrak{A}} \in \{\tau^{\text{LUCB}}, \tau^{\text{UGapE}}\}$, there exists an arm $b \in J(\tau_{\mathfrak{A}}) \cap (\mathcal{S}_N^\star(\varepsilon))^c$ (meaning that there is a mistake in the recommendation). If there is another arm $c \in \mathcal{S}_N^\star$ such that $\mathcal{B}_{c,b}(\tau_{\mathfrak{A}}) \leq \varepsilon$, [5] then using event $\mathcal{E}^{\text{GIFA}}$ and $c \in \mathcal{S}_N^\star(\varepsilon)$

$$\mu_{(N)} - \mu_b \leq \Delta_{c,b} \leq \mathcal{B}_{c,b}(\tau_{\mathfrak{A}}) \leq \varepsilon \implies \mu_{(N)} - \varepsilon \leq \mu_b \ ,$$

which is a contradiction to the fact that $b \notin \mathcal{S}_N^\star(\varepsilon)$. We will actually show that such an arm c always exists. Assuming towards contradiction that it does

[5] Be careful that here we consider $c \in \mathcal{S}_N^\star$ (one of the N best arms), and *not* $c \in \mathcal{S}_N^\star(\varepsilon)$ (one of the N best arms *up to* ε).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

not exist, and introducing set $\mathcal{C}_N(b) := \{a \in [K]\backslash\{b\} \mid \mathcal{B}_{a,b}(\tau_{\mathfrak{A}}) > \varepsilon\}$, the following two claims hold

$$\forall c \in \mathcal{S}_N^\star,\ \mathcal{B}_{c,b}(\tau_{\mathfrak{A}}) > \varepsilon \implies \mathcal{S}_N^\star \subseteq \mathcal{C}_N(b)\ , \tag{5.5}$$

$$\forall c \in \mathcal{S}_N^\star,\ \mathcal{B}_{c,b}(\tau_{\mathfrak{A}}) > \varepsilon \text{ and } |\mathcal{S}_N^\star| = N \implies \arg\max_{a\neq b}^{N} \mathcal{B}_{a,b}(\tau_{\mathfrak{A}}) \in \mathcal{C}_N(b)\ . \tag{5.6}$$

Let us split the proof in two parts for each stopping rule

LUCB-GIFA ($\tau_{\mathfrak{A}} = \tau^{\textbf{LUCB}}$). Using the definition of τ^{LUCB}, $c(t)$, $b(t)$, and $b \in J(\tau^{\text{LUCB}})$, it holds that for any $c \notin J(\tau^{\text{LUCB}})$

$$\mathcal{B}_{c,b}(\tau^{\text{LUCB}}) \leq \max_{j\in J(\tau^{\text{LUCB}})} \max_{i\notin J(\tau^{\text{LUCB}})} \mathcal{B}_{i,j}(\tau^{\text{LUCB}}) = \mathcal{B}_{c(\tau^{\text{LUCB}}),c(\tau^{\text{LUCB}})}(\tau^{\text{LUCB}}) \leq \varepsilon\ .$$

Then $\big(J(\tau^{\text{LUCB}})\big)^c \subseteq ([K]-\{b\})\backslash\mathcal{C}_N(b)$. Then, by Equation (5.5), $\mathcal{S}_N^\star \cap \big(J(\tau^{\text{LUCB}})\big)^c = \emptyset$, which means $\mathcal{S}_N^\star \cap J(\tau^{\text{LUCB}}) \neq \emptyset$ since both sets are not empty, of size N, which implies that $\mathcal{S}_N^\star = J(\tau^{\text{LUCB}})$. Then

$$J(\tau^{\text{LUCB}}) \cap (\mathcal{S}_N^\star(\varepsilon))^c = \mathcal{S}_N^\star \cap (\mathcal{S}_N^\star(\varepsilon))^c = \emptyset\ ,$$

which is a contradiction to the fact that $b \in J(\tau^{\text{LUCB}}) \cap (\mathcal{S}_N^\star(\varepsilon))^c$.

Gap-GIFA ($\tau_{\mathfrak{A}} = \tau^{\textbf{UGapE}}$). Using the definition of τ^{UGapE}, and $b \in J(\tau^{\text{UGapE}})$

$$\max_{a\neq b}^{N} \mathcal{B}_{a,b}(\tau^{\text{UGapE}}) \leq \max_{j\in J(\tau^{\text{UGapE}})} \max_{i\neq j}^{N} \mathcal{B}_{i,j}(\tau^{\text{UGapE}}) \leq \varepsilon\ ,$$

but, according to Equation (5.6), $\max_{a\neq b}^{N} \mathcal{B}_{a,b}(\tau^{\text{UGapE}}) > \varepsilon$, which is a contradiction. Hence, there always is $c \in \mathcal{S}_N^\star$ such that $\mathcal{B}_{c,b}(\tau_{\mathfrak{A}}) \leq \varepsilon$, which allows us to prove the theorem. □

It easily follows from Theorem 5.3.2 that if LinGapE or LinGIFA are based on good gap indices in the sense of Definition 5.3.1, both algorithms are δ-PAC. We exhibit below a threshold function $\mathcal{T}_\delta(\cdot)$, which defines the width of the upper confidence bounds, for which the corresponding gap indices are good gap indices.

Lemma 5.3.3. ***Example of good gap indices.*** *If for any pair of arms (i,j) and round $t > 0$, gap index $\mathcal{B}_{i,j}(t)$ is of the form*

$$\hat{\Delta}_{i,j}(t) + \sigma^2\mathcal{T}_\delta(t)\tilde{\mathcal{W}}_{i,j}(t)\ ,\ \textit{where}\ \tilde{\mathcal{W}}_{i,j}(t) := \begin{cases} \|X_i - X_j\|_{(\hat{V}^\kappa(t))^{-1}} & \textit{(paired)}\ , \\ \|X_i\|_{(\hat{V}^\kappa(t))^{-1}} + \|X_j\|_{(\hat{V}^\kappa(t))^{-1}} & \textit{(individual)}\ , \end{cases}$$

and

$$\mathcal{T}_\delta(t) := \sqrt{2\ln\left(\frac{1}{\delta}\right) + d\ln\left(1 + \frac{(t+1)L^2}{\kappa d}\right)} + \frac{\sqrt{\kappa}}{\sigma}S\ ,$$

with $L := \max_{a\in[K]} \|X_a\|_2$ and $S \in \mathbb{R}^+$ is such that $\|\theta\|_2 \leq S$, then $\mathcal{B}_{\cdot,\cdot}(\cdot)$ is a good

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

gap index in the sense of Definition 5.3.1.

Proof. The proof follows from the fact that

$$\bigcap_{t\in\mathbb{N}^*}\left\{\|\hat{\theta}(t)-\theta\|_{\hat{V}^{\kappa}(t)}\leq\sigma^2\mathcal{T}_\delta(t)\right\}\subseteq\mathcal{E}^{\text{GIFA}}\ ,$$

together with Kaufmann (2014, Lemma 4.1) which yields

$$\mathbb{P}\left[\forall t\in\mathbb{N}^*,\|\hat{\theta}(t)-\theta\|_{\hat{V}^{\kappa}(t)}\leq\sigma^2\mathcal{T}_\delta(t)\right]\geq 1-\delta\ .$$

For paired indices, the inclusion follows from the fact that, for any pair of arms (i,j) and round $t>0$

$$\left|\hat{\Delta}_{i,j}(t)-\Delta_{i,j}\right|=\left|(\hat{\theta}(t)-\theta)^\top(X_i-X_j)\right|\leq_{(1)}\|\hat{\theta}(t)-\theta\|_{\hat{V}^{\kappa}(t)}\|X_i-X_j\|_{(\hat{V}^{\kappa}(t))^{-1}}\leq_{(2)}\tilde{\mathcal{W}}_{i,j}(t)\ ,$$

where (2) is either an equality for paired indices, or the application of the triangle inequality for individual indices. (1) is shown by proving that, as a general rule, for any pair of vectors $x,y\in\mathbb{R}^d$ and positive definite matrix Σ

$$\left|x^\top y\right|\leq\|x\|_\Sigma\|y\|_{\Sigma^{-1}}\ .$$

Indeed, using the Cauchy-Schwarz inequality

$$\left|x^\top y\right|=\left|(\Sigma^\top x)^\top\Sigma^{-1}y\right|\leq\|\Sigma^\top x\|_2\|\Sigma^{-1}y\|_2=\|x\|_\Sigma\|y\|_{\Sigma^{-1}}\ .$$

□

Remark 5.3.4. *Note that L can easily be computed, since the agent has access to all the feature vectors. In order to provide a good value for S, if we know an upper bound M on the maximum absolute value of any of the expected rewards $(\mu_a)_{a_in[K]}$ (which is slightly easier to obtain depending on the application case), then using the fact that $\mu_a=\theta^\top X_a$ for any arm $a\in[K]$*

$$\forall a\in[K],\ |\mu_a|=\left|\theta^\top X_a\right|\leq L\times S\leq M\ ,$$

and then use $S=L/M$ as a proxy.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Unified sample complexity analysis

We derive below a high-probability upper bound on the sample complexity – that is, on the maximum value of τ_{GIFA}– of the subclass LUCB-GIFA, combined with different selection rules, and a conjecture on the sample complexity upper bound for Gap-GIFA algorithms. More precisely, we upper bound the sample complexity on event

$$\mathcal{E} := \bigcap_{t>0} \bigcap_{\substack{i\in[K]\\ j\in[K]}} (\Delta_{i,j} \in [-\mathcal{B}_{j,i}(t), \mathcal{B}_{i,j}(t)]) \ , \tag{5.7}$$

which is included in $\mathcal{E}^{\text{GIFA}}$. Considering any gap index of the form described in Definition 5.2.1, combined with Remark 5.2.2, event $\mathcal{E}$ holds with probability $1-\delta$.

Theorem 5.3.5. ***Sample complexity of LUCB-GIFA algorithms.*** *On event $\mathcal{E}$, stopping time τ^{LUCB} satisfies on model function $f \in \mathcal{M}_{\text{lin}}$*

$$\tau^{\text{LUCB}} \leq \inf\left\{u \in \mathbb{R}^{*+} \mid u > 1 + \mathcal{H}^{\text{LUCB-GIFA}}(f)(\mathcal{T}_\delta(u))^2\right\} \ ,$$

where, depending on the selection rule
Largest variance rule (Eq. (5.1)***)/sampling both*** $b(t)$ ***and*** $c(t)$***.***

$$\mathcal{H}^{\text{LUCB-GIFA}}(f) := 4\sigma^2 \sum_{a\in[K]} \max\left(\varepsilon, \frac{\varepsilon+\Delta_a}{3}\right)^{-2} \ ,$$

Optimized rule (Eq. (5.3)***).***

$$\mathcal{H}^{\text{LUCB-GIFA}}(f) := \sigma^2 \sum_{a\in[K]} \max_{i,j\in[K]^2} \frac{|\omega_a^\star(i,j)|}{\max\left(\varepsilon, \frac{\varepsilon+\Delta_i}{3}, \frac{\varepsilon+\Delta_j}{3}\right)^2} \ ,$$

where $\omega^\star(i,j)$ is the minimizer of Problem (5.4) *for any pair of arms (i,j).*

We compare all these claims to known results from the literature in Table 5.2, in particular, with regard to their complexity constant $\mathcal{H}^{\text{GIFA}}(f)$ for $f \in \mathcal{M}_{\text{lin}} := \{x \mapsto \theta^\top x \mid \theta \in \mathbb{R}^d\}$. We note that all GIFA algorithms have a scaling which is roughly of order $\mathcal{O}(\sum_{a\in[K]}(\varepsilon+\Delta_a)^{-2})$ when using the largest variance selection rule, or, interchangeably, the selection of both arms $b(t)$ and $c(t)$ at each round t. This is on par with what is observed in unstructured bandits in prior works.

Moreover, more sophisticated rules –namely, greedy and optimized selection rules– allow a possibly smaller scaling which might account for their empirical performance compared to the more naive largest variance rule.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Algorithm	**Upper bound**
Unstructured	$\inf\{u \in \mathbb{R}^{*+} \mid u > 1 + \mathcal{H}^{\text{uns}}(f)(\mathcal{T}_\delta(u))^2\}$
LUCB	$\mathcal{H}^{\text{uns}}(f) := 2\sum_a \max(\varepsilon/2, \Delta_a)^{-2}$
UGapE	$\mathcal{H}^{\text{uns}}(f) := 2\sum_a \max(\varepsilon, (\varepsilon+\Delta_a)/2)^{-2}$
GIFA*	$\inf\{u \in \mathbb{R}^{*+} \mid u > 1 + \mathcal{H}^{\text{lin}}(f)(\mathcal{T}_\delta(u))^2\}$
Largest var.	$\mathcal{H}^{\text{lin}}(f) := 4\sigma^2\sum_a \max(\varepsilon, (\varepsilon+\Delta_a)/3)^{-2}$
Optimized	$\mathcal{H}^{\text{lin}}(f) := \sigma^2\sum_a \max_{i,j\in[K]^2} \lvert\omega_a^\star(i,j)\rvert (\max(\varepsilon, (\varepsilon+\Delta_i)/3, (\varepsilon+\Delta_j)/3))^{-2}$
Greedy*	$\mathcal{H}^{\text{lin}}(f) := \sigma^2\sum_a \max_{i,j\in[K]^2} \frac{\lVert X_i - X_j\rVert_2^2}{\lVert X_a\rVert_2^2} (\max(\varepsilon, (\varepsilon+\Delta_i)/3, (\varepsilon+\Delta_j)/3))^{-2}$

Table 5.2: **Sample complexity results for linear bandits.** Sample complexity results for unstructured and linear bandits on any model function $f \in \mathcal{M}_{\text{lin}}$. $\omega^\star$ is defined as the minimizer of Problem (5.4). Our proposed algorithms are in bold type, and conjectures are denoted with a superscript star. I omit factors in $\mathcal{O}(K)$ in unstructured bandit algorithms.

However, at fixed arm a, no problem-dependent upper bound is known on quantity $|\omega_a^\star(i,j)|$ for any arms (i,j), where $\omega^\star(i,j)$ is solution to Problem (5.4), which intervenes in the complexity constant for the optimized selection rule. If it exists, let us denote it $\Gamma_a(X)$. Then the associated constant would be upper-bounded by a quantity of order $\mathcal{O}\left(\sigma^2 \max_{b\in[K]} \Gamma_b(X) K \max_a (\varepsilon+\Delta_a)^{-2}\right)$. If $1 \gg \sigma^2 \max_{b\in[K]} \Gamma_b(X)$, then this quantity depends on something indeed smaller than the sample complexity of unstructured bandit algorithms. This bound for the optimized selection rule is similar to what was derived in the original paper of LinGapE (Xu, Honda, and Sugiyama, 2018) for best arm identification. [6]

For the conjectured bound for the greedy rule, consider the following ratio between the ℓ_2 norms of feature vectors at fixed arm $a \in [K]$ [7]

$$\Gamma'_a(X) := \max_{i,j\in[K]^2} \frac{\lVert X_i - X_j\rVert_2}{\lVert X_a\rVert_2} \leq 2L(\lVert X_a\rVert_2)^{-1} .$$

Th associated complexity constant is then upper bounded by a factor

$$\mathcal{O}\left(\sigma^2 \max_{b\in[K]} \Gamma'_b(X) K \max_a (\varepsilon+\Delta_a)^{-2}\right) .$$

[6] However, note that the gaps depend on N and thus are defined differently.
[7] We suspect that, if X is well-conditioned (*w.r.t.* a well-chosen norm), $\Gamma'_a(X) \approx \mathcal{O}(1)$.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Sketch of proof. The proof of Theorem 5.3.5 generalizes and extends the proofs for unstructured and linear Top-N identification, with both paired and individual gap indices. Only a big picture of the proof is provided. The full proof is contained in the supplementary part in Réda, Kaufmann, and Delahaye-Duriez (2021). Appendix 12 comprises of the computations that guided Conjectures 5.3.9 and 5.3.10. As mentioned in Table 5.1, the LUCB-GIFA type of algorithms comprises of the following two rules : at round t

$$J(t) := \arg\max_{j\in[K]} \widehat{\mu}_a(t) \ (\texttt{compute_Jt}) \text{ and } b(t) := \arg\max_{j\in J(t)} \max_{i\notin J(t)} \mathcal{B}_{i,j}(t) \ (\texttt{compute_bt}).$$

This class of algorithms includes LinGapE (Xu, Honda, and Sugiyama, 2018) for best arm identification, our proposal N-LinGapE for Top-N identification, but also LUCB (Kalyanakrishnan et al., 2012). The key ingredient in the proof is a lemma that gives an upper bound for any round t on the stopping quantity $\mathcal{B}_{c(t),b(t)}(t)$. This lemma holds for any gap index satisfying Definition 5.2.1

Lemma 5.3.6. ***Upper bound on the stopping quantity (LUCB-GIFA).*** *On event $\mathcal{E}$, for any round $t > 0$*

$$\mathcal{B}_{c(t),b(t)}(t) \leq \min\left(-\max(\Delta_{b(t)}, \Delta_{c(t)}) + 2\mathcal{W}_{c(t),b(t)}(t), 0\right) + \mathcal{W}_{c(t),b(t)}(t) \ .$$

This result is shown by case disjunction on the membership of $b(t)$ and $c(t)$ to $\mathcal{S}^\star_N(\varepsilon)$. This is a counterpart to Lemma 4 in Xu, Honda, and Sugiyama (2018), but does not require $|J(t)| = 1$ at round $t > 0$, notably by noticing that, by definition of $b(t)$ and $c(t)$, for any $N \geq 1$,

$$\mathcal{B}_{c(t),b(t)}(t) = \max_{j\in J(t)} \max_{i\notin J(t)} \mathcal{B}_{i,j}(t) \ .$$

In order to get the upper bound in Theorem 5.3.5 using the optimized rule (Equation (5.3)), one can straightforwardly apply Xu, Honda, and Sugiyama (2018, Lemma 1) to the inequality in Lemma 5.3.6. For the selection rule which either selects both $b(t)$ and $c(t)$, or the largest variance rule (Equation (5.1)), by combining Lemma 5.3.6 with the definition of the stopping time τ^{LUCB}, we obtain the following upper bound on the number of times $N_{I_t}(t)$ that arm I_t has been sampled up to round t included, at any round $t < \tau^{\text{LUCB}}$

Lemma 5.3.7. ***Upper bound on the number of selections (LUCB-GIFA, largest variance rule).*** *For any round $t > 0, t < \tau^{LUCB}$*

$$N_{I_t}(t) \leq 4\sigma^2(\mathcal{T}_\delta(t))^2 \max\left(\varepsilon, \frac{\varepsilon + \Delta_{I_t}}{3}\right) \ .$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Finally, we apply the following result

Lemma 5.3.8. ***Inversion lemma.*** *Let $f : [K] \times (0,1) \times \mathbb{N}^* \to \mathbb{R}^{*+}$ be a nondecreasing function in its last argument, and $\mathcal{I}_t$ the set of sampled arms at exactly round t. Let $\mathcal{E}$ be an event such that for any $t < \tau_{\mathfrak{A}}$, for any $\delta \in (0,1)$, there is an arm $I_t \in \mathcal{I}_t$ such that $N_{I_t}(t) \leq f(I_t, \delta, t)$. Then, on event $\mathcal{E}$,*

$$\tau_{\mathfrak{A}} \leq \inf \left\{ u > 0 \mid u > 1 + \sum_{a \in [K]} f(a, \delta, t) \right\} .$$

The proof for both lemmas is available in Réda, Kaufmann, and Delahaye-Duriez (2021). I also propose a few conjectures present in Table 5.2, as listed below

Conjecture 5.3.9. ***Sample complexity of Gap-GIFA.*** *On event $\mathcal{E}$, stopping time τ^{UGapE} satisfies on model function $f \in \mathcal{M}_{\text{lin}}$*

$$\tau^{\text{UGapE}} \leq \inf \left\{ u > 0 \mid u > 1 + \mathcal{H}^{\text{Gap-GIFA}}(f)(\mathcal{T}_\delta(u))^2 \right\} ,$$

where, depending on the selection rule
Largest variance rule (Eq. (5.1)***)/sampling both*** $b(t)$ ***and*** $c(t)$***.***

$$\mathcal{H}^{\text{Gap-GIFA}}(f) := 4\sigma^2 \sum_{a \in [K]} \max \left(\varepsilon, \frac{\varepsilon + \Delta_a}{3} \right)^{-2} .$$

Conjecture 5.3.10. ***Greedy rule (Eq.*** (5.2)***).*** *On event $\mathcal{E}$ and model function $f \in \mathcal{M}_{\text{lin}}$, using the same notations as in Theorem 5.3.5*

$$\mathcal{H}^{\text{LUCB-GIFA}}(f) := \sigma^2 \sum_{a \in [K]} \max_{i,j \in [K]^2} \frac{\|X_i - X_j\|_2^2}{\|X_a\|_2^2} \max \left(\varepsilon, \frac{\varepsilon + \Delta_i}{3}, \frac{\varepsilon + \Delta_j}{3} \right)^{-2} .$$

5.4 Application to drug repurposing

We compared the empirical performances of both GIFA proposals –namely, N-LinGapE and LinGIFA– to the two adaptive sampling algorithms for unstructured models in GIFA, that is, LUCB (Kalyanakrishnan et al., 2012) and UGapE (Gabillon, Ghavamzadeh, and Lazaric, 2012). These algorithms were applied to a drug repurposing instance created using the drug scoring in Chapter 3 on 21 drugs –comprised of 10 antiepileptics and 11 proconvulsants, that is, that trigger seizures. In practice, the scores for each drug and patient profile have been obtained prior to the experiments, and saved in a matrix. When an agent asks for the evaluation of a given arm, a patient index is

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

sampled at random, and the score related to the arm and this patient is returned. At first, drug signatures directly computed from LINCS L1000, as described in Chapter 3, were considered as feature vectors in $\mathbb{R}^{194}$. However, the underlying model is very far from being linear, and this fact incurred empirical error rates higher than δ. The next chapter (Chapter 6) is actually motivated by this issue. In the meanwhile, in order to test our theoretical results on GIFA, a feature transformation procedure is performed on the drug signatures, as done in Papini et al. (2021, Appendix $F.4$). This procedure only transforms the feature vectors such that the resulting model is –roughly– linear. This procedure goes as follows : a linear model is extracted from the data by first fitting a neural network that regresses a linear model from each feature vector X_a to score μ_a for any arm a. Then, the activations from the last layer are selected as feature vectors (of dimension $d = 15$), and the associated network parameters constitute the linear parameter θ. The ℓ_∞ norm of the difference between the predictions of the resulting linear model and the scores $(\mu_a)_{a\in[K]}$ is equal to 0.132, which is lower than the minimum gap $\Delta_{a_{(N)}a_{(N+1)}} \approx 0.14$ for $N = 3$, which is enough to assume in practice that this transformed model is quasi-linear (Ghosh, Chowdhury, and Gopalan, 2017; Réda, Tirinzoni, and Degenne, 2021). To sum it up, the feature vectors of dimension $d = 15$ are considered, and observed rewards are computed *true* scores from a randomly selected patient for the sampled treatment.

Remark 5.4.1. *Of course, in a real-life setting, one does not have access to the true labels (average scores) associated with candidate drugs. Such a setting is however useful to check if our theoretical results empirically hold. In practice, either a non-linear algorithm should be considered on the untransformed data (see next Chapter 6), or either the (quasi) linearity of the data should be assessed on some datapoints (drugs) such that the recommendations from linear algorithms can be trusted. For instance, if the number of features is a lot larger than the number of arms/candidate drugs, the problem is intrinsically linear –but remember that a large number of features will result with high computing cost, due to the matrix multiplications which are cubic in the number of features. Application to "real life" drug repurposing is performed in Appendix 14.*

In this experiment, we consider $\sigma^2 = 1$, $\delta = 0.1$, $N = 3$ and we set ε to 0.2. For each algorithm, we use the corresponding theoretically-supported gap index, and we run it 100 times on the drug repurposing problem, in order to estimate its average empirical error rate and sample complexity.

One boxplot per algorithm is displayed in Figure 5.1. The y-axis corresponds to the sample complexity while the x-axis shows the algorithmic variants, with their reported empirical error rate $\hat{\delta}$ computed across the 100

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

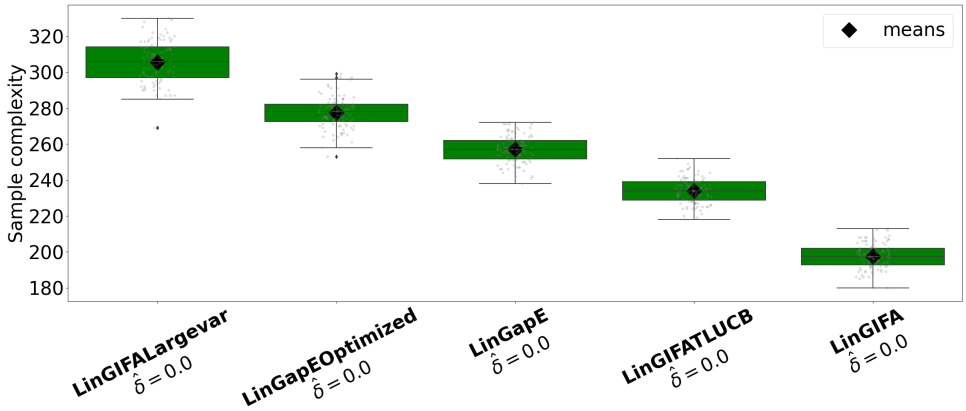

Figure 5.1: **Drug repurposing instance with linear bandits.** Drug repurposing instance with linear bandit algorithms (number of dimensions $d = 15$, $\sigma = 1$, $\delta = 10$ %, $\varepsilon = 0.2 > \Delta_{a_{(N)}a_{(N+1)}} \approx 0.14$, $N = 3$, 100 iterations) with their reported empirical error rates $\hat{\delta}$.

iterations, and rounded up to the 5th decimal place. Individual outcomes are shown as gray dots. Strikingly, both LUCB and UGapE, but also the version of LinGIFA which uses individual gap indices instead of paired ones, need more than $10,000$ samples at each iteration of the experiment –which is why they are not shown in the boxplot. We have tested several versions of our two proposals :

- LinGIFA with the largest variance selection rule (**LinGIFALargevar**).

- LinGIFA (that is the version mentioned in Table 5.1 with the greedy selection rule).

- LinGIFA with the stopping rule τ^{LUCB} instead of τ^{UGapE} (**LinGIFATLUCB**).

- N-LinGapE with the optimized selection rule (**LinGapEOptimized**).

- The default version of N-LinGapE shown in Table 5.1 with the greedy selection rule (**LinGapE**).

This plot hightlights several interesting points :

1. Taking into account feature vectors is crucial for sample-efficient algorithms, at the condition of selecting the appropriate structure. Especially in this drug repurposing instance, linear algorithms perform better by several orders of magnitude than unstructured ones.

2. The experiment with the version of LinGIFA using individual gap indices shows that upper confidence bounds built with individual gap indices can be a lot larger than those associated with paired indices, which was predicted by our Lemma 5.2.4. Moreover, this suggests that the loss of performance between unstructured and linear bandit algorithms is mainly due to the use

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

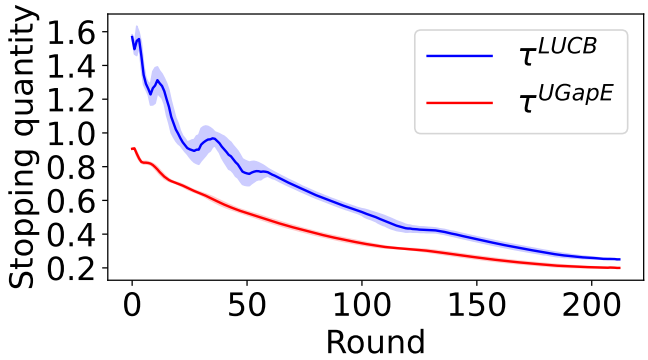

Figure 5.2: **Comparison between the two stopping rules.** Plotting τ^{LUCB} against τ^{LUCB} on a same set of 100 trajectories of the version of LinGIFA using the stopping rule τ^{LUCB} (**LinGIFATLUCB**) on the drug repurposing instance. The blue curve ("τ^{LUCB}") plots $\mathcal{B}_{c(t),b(t)}(t)$ at each round t : the solid line being the average value at round t across the 100 iterations, the blue lighter zone the interval around the average value $\pm$ the standard deviation. Similarly, the red curve ("τ^{UGapE}") plots $\max_{j \in J(t)} \max_{i \neq j}^{N} \mathcal{B}_{i,j}(t)$.

of individual gap indices, in addition to less sophisticated selection rules.

3. As expected from the conclusion of our Lemma 5.2.8, **LinGIFA** (the default version using the stopping rule τ^{UGapE}) is more sample-efficient than **LinGIFATLUCB** (the version of LinGIFA which only differs by the use of stopping rule τ^{LUCB}). Figure 5.2 plots the two curves associated with stopping quantities $\mathcal{B}_{c(t),b(t)}(t)$ (for τ^{LUCB}), $\max_{j \in [K]} \max_{i \neq j}^{N} \mathcal{B}_{i,j}(t)$ (for τ^{UGapE}) on the same 100 trajectories, which empirically confirms Lemma 5.2.8.

4. On average, **LinGapE** (N-LinGapE with the greedy selection rule) ends before **LinGapEOptimized** (N-LinGapE which uses the optimized selection rule), although the speedup is roughly of 7%, which seems to confirm our Conjecture 5.3.10.

5.5 Discussion

This chapter introduced a family of algorithms aimed at solving Top-N identification for linear models. According to Table 5.2, the sample complexity-related constant is always at most of order $\mathcal{O}(\sum_k (\varepsilon + \Delta_k)^{-2})$, which is similar to what holds for unstructured bandits. However, can we exhibit an algorithm for Top-N identification which would be theoretically optimal (that is, asymptotically optimal, in the sense of Result 4.3.4)? Can we even get an easily computable version of the lower bound in Result 4.3.3? Moreover, how robust are these algorithms to non-linearity in the model? These points motivate the next chapter.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 6

Dealing with non-linear models

In Chapter 5, I studied the integration of feature vectors into a Top-N identification problem for bandits. As discussed in the conclusion of Chapter 5, although linear models are extremely attractive for their interpretability and the existence of associated efficient algorithms, real-life models might deviate from linearity. In particular, there might not be any valid value of parameter θ in the linear regression described in Subsection 5.2. As a consequence, the confidence intervals might not hold anymore, and the algorithm might return errors with rate higher than δ. In Table 6.1, I exhibit a synthetic experiment that shows that an efficient linear bandit algorithm can fail a lot more often than δ% of the time when facing non-linear models.

Ψ	0	1	2	2.7	2.8	3	4	5
$\hat{\delta}$	0%	0%	0%	1%	6%	28%	100%	100%

Table 6.1: **Running a linear bandit algorithm on an increasingly non linear model.** Empirical error $\hat{\delta}$ from running N-LinGapE for Top-N identification (introduced in Chapter 5) on synthetic models of the form $\theta^\top X + \Psi\left(\mathbb{1}(i = a_{(N+1)})\right)_{i\in[K]}$, using $\delta = 5$%, $d = 5$, $K = 10$, $N = 3$, $\sigma = 1$, and the theoretically-supported gap index in Lemma 5.3.3 across 100 iterations. Feature vectors were drawn at random. [a] The minimum gap in the associated linear model $\theta^\top X$ is $\Delta_{a_{(N)},a_{(N+1)}} \approx 0.2786$. The larger Ψ is, the more the underlying model deviates from linearity, and the error $\hat{\delta}$ increases.

[a] The exact value of matrix X is available at https://github.com/clreda/misspecified-top-m/tree/main/data ("simulated misspecified instance" in Réda, Tirinzoni, and Degenne (2021)).

However, we might still want to preserve the linear part of the model : first, for better interpretability ; second, to design a flexible bandit algorithm that can switch between perfectly linear and completely unstructured models, when feature vectors no longer bring supplementary information about the scores. In this chapter, I consider *misspecified linear* models, that

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

can interpolate between linear and unstructured models. The corresponding class $\mathcal{M}_\Psi$ of Ψ-misspecified models is

$$\mathcal{M}_\Psi := \{x \mapsto \theta^\top x + \eta(x) \mid \theta \in \mathbb{R}^d,\ \eta \in (\mathbb{R}^d \to \mathbb{R}) \text{ s.t. } \max_{k\in[K]} |\eta(x)| \leq \Psi\} .$$

As their name alludes to, the deviation of these models to a linear model –in terms of the maximum absolute coefficient of the unstructured term η– is always smaller than a fixed $\Psi \geq 0$. When $\Psi = 0$, this class is the class of linear models, whereas, as Ψ grows, it matches the class of unstructured models –with bounded means. For large values of Ψ, the contribution η_a to the expected reward μ_a of arm a can be a lot larger than the linear part $\theta^\top X_a$: most of the information about arm a is present in η_a.

Remark 6.0.1. *In practice, we consider the vector of expected rewards μ*

$$\mu = \theta^\top X + \eta\ ,\ \textit{where } \theta \in \mathbb{R}^d,\ \eta \in \mathbb{R}^K\ .$$

In particular, here, compared to Equation 6, possibly two arms a, b with the same feature vectors ($X_a = X_b$) might have a different "unstructured" coefficient $\eta_a \neq \eta_b$. However, in our drug repurposing problem, the rewards result from a function of the signatures, which prevents that case from arising.

Remark 6.0.2. *Notice that the class of Ψ-misspecified models is different from the following class*

$$\mathcal{M} = \{x \mapsto \theta^\top \tilde{x} \mid \theta \in \mathbb{R}^{d+1}\}\ ,$$

where $\tilde{\cdot}$ is the operator that turns a vector in $\mathbb{R}^d$ into a vector in $\mathbb{R}^{d+1}$ by appending vector $[1]$ to the initial vector. Indeed, in that case, the last coefficient θ_{d+1} is shared by all arms, whereas coefficients $\eta_1, \eta_2, \ldots, \eta_K$ can possibly have different values.

In this chapter, we focus on fixed-confidence *exact* Top-N identification ($\varepsilon = 0$). First, a tractable lower bound on the sample complexity of any δ-correct algorithm (Definition 4.1.4) is derived. Then, an algorithm adapted to misspecified linear models is designed. Finally, the performance of this algorithm is compared to unstructured and linear bandit algorithms on a drug repurposing instance. This is a joint work with Andrea Tirinzoni and Rémy Degenne (SCOOL Inria team), and it led to a publication at the 35th Neural Information Processing Systems conference (NeurIPS 2021) in Réda, Tirinzoni, and Degenne (2021). [1]

[1]The code related to experiments is available at https://github.com/clreda/misspecified-top-m.

Except where otherwise noted, this work is licensed under https://creativecommons.org/licenses/by-nd/3.0/fr/

6.1 Related work

Learning a model under misspecification of its structure requires adapting to the scale of misspecification, typically under the assumption that some information about the latter is known –*e.g.*, an upper bound Ψ to $\|\eta\|$. Due to its importance, this problem has recently gained increasing attention in the bandit community for regret minimization, but has not been addressed in the context of pure exploration. Ghosh, Chowdhury, and Gopalan (2017) introduced the class of misspecified models for regret minimization. In that work, the authors show that if T is the learning horizon, for any bandit algorithm which enjoys an optimal regret scaling on linear models, there exists a misspecified instance where the regret is necessarily linear. As a workaround, they design a statistical test based on sampling a subset of arms prior to learning, to decide whether a linear or an unstructured bandit algorithm should be run on the data. Similar ideas are suggested in Chatterji, Muthukumar, and Bartlett (2020), where the authors design a sequential test to sequentially switch between linear and unstructured algorithms. More recently, elimination-based algorithms (Lattimore, Szepesvari, and Weisz, 2020; Takemura et al., 2021) and model selection methods (Foster et al., 2020; Pacchiano et al., 2020) were popular approaches to tackle model misspecification for regret minimization. Notably, these regret minimization algorithms adapt to the amount of misspecification Ψ without knowing it beforehand, at the cost of an additive linear term that scales with Ψ.

Moreover, while best-arm identification has been the focus of many prior works in the realizable linear setting, some suggesting asymptotically-optimal algorithms (Degenne, Ménard, et al., 2020; Jedra and Proutiere, 2020), Top-N identification has been seldom studied in terms of problem-dependent lower bounds. Lower bounds on the sample complexity for the unstructured [2] Top-N problem have been derived previously, either focusing on explicit bounds (Kaufmann, Cappé, and Garivier, 2016, Theorem 4), or on worst-case lower bounds (Chen, Li, and Qiao, 2017, Theorem 1.1), or on getting the correct dependence in the problem parameters for any error rate δ (Simchowitz, Jamieson, and Recht, 2017, Proposition 6). Because of the combinatorial nature of the Top-N identification problem, obtaining a tractable, tight, problem-dependent lower bound is not straightforward.

[2] *i.e.*, $\mathcal{M} := (\mathbb{R}^d \to \mathbb{R})$, the whole set of functions mapping from $\mathbb{R}^d \to \mathbb{R}$.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

6.2 Assumptions

We denote $L := \max_{k\in[K]} \|X_k\|_2$, and we assume that the feature vectors span $\mathbb{R}^d$ –otherwise, those vectors could be rewritten in a subspace of smaller dimension. Furthermore, we assume that there exists some $C \in \mathbb{R}^+$ such that, for any model function $f \in \mathcal{M}_\Psi$, $|f(X)|_\infty := \max_{k\in[K]} |f(X_k)| \leq C$. [3] Since f is a Ψ-misspecified model function, let us consider associated parameters θ and η, such that for any arm $a \in [K]$, $f(X_a) = \theta^\top X_a + \eta_a$. With regard to interpretability, we can observe that the absolute values of the coefficients in $\theta \in \mathbb{R}^d$ are still related to the influence of each of the d features on the average score $f(X_a)$; with the addition of an arm-specific bias term η_a.

As previously mentioned, we focus on the case where $\varepsilon = 0$, that is, we only aim at returning arms in the true top-N set $\mathcal{S}^\star_{N,f}$. However, the set of N best arms of model function f might not be well defined since the set $\mathcal{S}^\star_{N,f} := \left\{k \in [K] \mid f(X_k) \geq \max_{i\in[K]}^{N} f(X_i)\right\}$ might contain more than N elements if some arms have the same mean. Thus, let $\mathcal{S}_{N,f} := \left\{S \subseteq \mathcal{S}^\star_{N,f} \mid |S| = N\right\}$ be the set containing all subsets of N elements of $\mathcal{S}^\star_{N,f}$. Moreover, suppose that the true model function f has exactly N arms that are among the top-N, *i.e.*, that $\left|\mathcal{S}^\star_{N,f}\right| = N$ and $\mathcal{S}_{N,f} = \{\mathcal{S}^\star_{N,f}\}$. Remember that any δ-correct algorithm $\mathfrak{A}$ satisfies Definition 4.1.4, that is, for any $f \in \mathcal{M}_\Psi$

$$\mathbb{P}_{\{f\}}\left(\hat{\mathcal{S}}_N(\tau_{\mathfrak{A}}) \not\subseteq \mathcal{S}^\star_{N,f}\right) \leq \delta \,.$$

6.3 Tractable lower bound for general Top-N identification

In this section, we consider any class of models $\mathcal{M}$, not necessarily $\mathcal{M}_\Psi$. As mentioned in Section 4.3, being able to asymptotically match –in the sense of Result 4.3.4– the lower bound on the sample complexity might lead to performant algorithms. In prior works that tackle pure exploration tasks, authors have focused on recovering (an estimation of) the oracle allocation $\arg\min_{\omega\in\triangle_K}$ in the general pure exploration lower bound (Result 4.3.3) to design asymptotically optimal algorithms. This oracle allocation can be approached either by using tracking (Du et al., 2021; Garivier and Kaufmann, 2016; Russac et al., 2021), or through online optimization algorithms (Degenne and Koolen, 2019; Degenne, Ménard, et al., 2020).

[3] In practice, the algorithm proposed in this work will not need to use the actual value of C.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

The computation of an oracle is tractable and well-studied for best arm identification, *i.e.*, $N = 1$, since there are $K - 1$ sets of alternative models when the optimal arm $a^\star$ is unique. Each alternative corresponds to one of the $K - 1$ alternative best arms. However, in Top-N identification, the naive computation of the lower bound-related optimization problem is done over $\binom{K}{N} = \frac{K!}{(K-N)!N!}$ sets, where $k! = 1 \times 2 \times 3 \times \cdots \times k$ for any positive integer k. In this problem, the set of alternative models to $f \in \mathcal{M}$ –which can be any structure– is

$$\text{Alt}(f) := \left\{ \widetilde{f} \in \mathcal{M} \mid \mathcal{S}_{N,\widetilde{f}} \cap \mathcal{S}_{N,f} = \emptyset \right\} .$$

This is the set of all $\mathcal{M}$-structured model functions $\widetilde{f}$, such that the set of top-N arms differs from $\mathcal{S}^\star_{N,f}$. Note that, while we assumed that the set of top-N arms under f is unique, this might not be the case for alternative $\widetilde{f}$. Given any $\delta-$correct algorithm $\mathfrak{A}$ tackling Top-N identification in $\mathcal{M}$-structured models, for any $\delta \leq 1/2$, we can combine Result 4.3.3, which gives a lower bound on the sample complexity for any pure exploration problem, with the following lemma

Lemma 6.3.1. ***Expression of Alt***(f) ***in Top-***N ***identification.*** *For any model functions* $f, \widetilde{f} \in \mathcal{M}$ *such as* $|\mathcal{S}_{N,f}| = 1$

$$\widetilde{f} \in \textit{Alt}(f) \Leftrightarrow \forall S \in \mathcal{S}_{N,f} \ \exists i \notin S \ \exists j \in S, \ \widetilde{f}(X_i) > \widetilde{f}(X_j) .$$

Proof. **Implication.** If $\widetilde{f} \in \text{Alt}(f)$, then it means that $\mathcal{S}_{N,f} \cap \mathcal{S}_{N,\widetilde{f}} = \emptyset$. Suppose that

$$\exists S \in \mathcal{S}_{N,f} \ \forall i \notin S \ \forall j \in S, \ \widetilde{f}(X_i) \leq \widetilde{f}(X_j) . \tag{6.1}$$

Then, this implies that for any arm $i \notin S$, $\widetilde{f}(X_i) \leq \min_{j \in S} \widetilde{f}(X_j)$. Since, by definition, $|S| = N$, then S is a valid Top-N set under model function $\widetilde{f}$, which means that $\mathcal{S}_{N,f} \cap \mathcal{S}_{N,\widetilde{f}} \neq \emptyset$, and then $\widetilde{f} \notin \text{Alt}(f)$. In conclusion, Equation (6.1) implies that $\widetilde{f} \notin \text{Alt}(f)$. Then the contraposition also holds.

Inverse implication. Consider $S \in \mathcal{S}_{N,f}$ (unique top-N set associated with f), and arms i, j that satisfy the right-hand expression in Lemma 6.3.1. We now consider two cases :

- If there exists a top-N set $\widetilde{S}$ associated with $\widetilde{f}$, *i.e.*, $\widetilde{S} \in \mathcal{S}_{N,\widetilde{f}}$, such that $j \in \widetilde{S}$, then by definition of i, $i \in \widetilde{S} \cap (S)^c$.

- Otherwise, j is not among the best N arms under model function $\widetilde{f}$. Then for any $\widetilde{S} \in \mathcal{S}_{N,\widetilde{f}}$, $j \in S \cap \left(\widetilde{S}\right)^c$.

This means that, in both cases, $S \notin \mathcal{S}_{N,\widetilde{f}}$, and then $\mathcal{S}_{N,\widetilde{f}} \cap \mathcal{S}_{N,f} = \emptyset$. □

The combination of these two results yields the following lower bound for

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

any structured Top-N identification problem

Theorem 6.3.2. ***Characteristic time for Top-N identification.*** *Let KL be the Kullback-Leibler divergence on continuous probability densities (Definition 4.3.2). For any δ-correct algorithm $\mathfrak{A}$ for Top-N identification where $\delta \leq 1/2$, and any $f \in \mathcal{M}$, the following lower bound holds on the sample complexity*

$$\mathbb{E}_{\{f\}}[\tau_{\mathfrak{A}}] \geq (\mathcal{C}^\star(f))^{-1} \log\left(\frac{1}{2.4\delta}\right) ,$$

where

$$\mathcal{C}^\star(f) = \sup_{\omega \in \triangle_K} \min_{i \notin \mathcal{S}_f^\star} \min_{j \in \mathcal{S}_f^\star} \inf_{\substack{\widetilde{f} \in \mathcal{M} \\ \widetilde{f}(X_i) > \widetilde{f}(X_j)}} \sum_{a \in [K]} \omega_a KL_a(f, \widetilde{f}) ,$$

and $KL_a(f, \widetilde{f}) := KL(\nu_a, \alpha_a)$ where ν_a, α_a are two Gaussian laws with fixed variance σ^2 that satisfy $\mathbb{E}_{\{Y \sim \nu_a\}}[Y] = f(X_a)$ and $\mathbb{E}_{\{Y \sim \alpha_a\}}[Y] = \widetilde{f}(X_a)$, and

$$\triangle_K := \left\{ p \in [0,1]^K \mid \sum_{k \in [K]} p_k = 1 \right\}$$

is the simplex on $[K]$.

Computing this quantity now requires to : **(a)** maximize once over the simplex –which can be still hard ; **(b)** minimize over the $N(K-N)$ half-spaces $\{\widetilde{f} \in \mathcal{M} \mid \widetilde{f}(X_i) > \widetilde{f}(X_j)\}$ for $(i,j) \in \left(\mathcal{S}_f^\star\right)^c \times \mathcal{S}_f^\star$. Those minimizations are convex optimization problems, and can be solved efficiently for $\mathcal{M} = \mathcal{M}_\Psi$. Our algorithm inspired from that bound will need to perform only those minimizations, since it uses an online approach to estimate the oracle allocation $\omega \in \triangle_K$. For Gaussian bandits with fixed variance σ^2 –*i.e.*, when rewards follow a Gaussian distribution with variance σ^2– using the expression of the Kullback-Leibler divergence in that case (Equation (4.4)) yields the following optimization problem related to characteristic time $\mathcal{C}^\star$

$$\min_{i \notin \mathcal{S}_f^\star} \min_{j \in \mathcal{S}_f^\star} \inf_{\substack{\widetilde{f} \in \mathcal{M}_\Psi \\ \widetilde{f}(X_i) > \widetilde{f}(X_j)}} \frac{1}{2\sigma^2} \sum_{a \in [K]} \omega_a (f(X_a) - \widetilde{f}(X_a))^2 . \tag{6.2}$$

This problem can be solved in Python by iterating over all pairs $(i,j) \in \left(\mathcal{S}_f^\star\right)^c \times \mathcal{S}_f^\star$, and independently solving at fixed (i,j) the quadratic program derived in Réda, Tirinzoni, and Degenne (2021, Appendix $C.3$) using module `quadprog` based on Goldfarb and Idnani (1983). Finally, the solution which yields the minimum value across all pairs of arms is selected.

Note that a lower bound for Top-N identification using Result 4.3.3 has been obtained in Kaufmann, Cappé, and Garivier (2016, Theorem 4). Aiming to be more explicit, they defined the set of alternative models as the set of

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

models where one of the best arms is switched with the $(N+1)^{\text{th}}$ best arm under f, or one of the $K-N$ worst ones with the Nth best one under f. These models are a strict subset of Alt(f) for any $f \in \mathcal{M}$. Hence this bound is not as tight as the one in Theorem 6.3.2, which is why the algorithm we introduce in the next section will rely on the latter instead. Note that for $\Psi = 0$ (linear models) and $N = 1$ (best arm identification), this lower bound is exactly the one derived by Fiez et al. (2019, Theorem 1).

Moreover, it was shown in Réda, Tirinzoni, and Degenne (2021, Section 3.1) using Theorem 6.3.2 that knowing that a problem is misspecified without knowing the upper bound Ψ on $\|\eta\|_\infty$ is the same as not knowing anything about the structure of that problem. This means that there is no optimal δ-correct algorithm which is oblivious to Ψ ; if an algorithm $\mathfrak{A}$ is δ-correct and optimal –in terms of sample complexity– on the set of model functions $\mathcal{M}_\Psi$, then it cannot be optimal as well on a strict subset of model functions $\mathcal{M}_{\Psi'}$, where $\Psi' < \Psi$. Hence, the algorithm described in the next section will crucially need a good estimate Ψ of the deviation to linearity.

In a related work on δ-correct best arm identification dealing with misspecified model structures (Zhu, D. Zhou, et al., 2021), the proposed algorithm also relies on an estimated upper bound on the approximation error between the misspecified model (represented by a neural network) and the true expected rewards.

6.4 Misspecified Top-N identification

Now we introduce an algorithm for fixed-confidence Ψ-misspecified Top-N identification, named misspecified linear identification (MisLid). Its structure is outlined in Algorithm 7. This algorithm is only well-defined for Gaussian bandits with fixed variance $\sigma^2 = 1$. On the one hand, the design of MisLid builds on top of recent approaches for constructing pure exploration algorithms from lower bounds (Degenne, Koolen, and Ménard, 2019; Degenne, Ménard, et al., 2020; Jedra and Proutiere, 2020; Zaki, Mohan, and Gopalan, 2020). On the other hand, its main components and their analysis introduce several technical novelties to deal with misspecified Top-N identification, that might be of independent interest for other settings. We describe these components below. Let us define for any vector $v \in \mathbb{R}^K$

$$D_v := \text{diag}(v_1, v_2, \dots, v_K) \text{ and } \hat{V}^\kappa(t) := \sum_{s \le t} X_{I_t} X_{I_t}^\top \text{ (note that } \kappa = 0 \text{ here)} .$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

The core principle of the algorithm is to sample arms according to the oracle allocation $\omega(t)$ at round t. This allocation is estimated by an online learner $\mathcal{L}$, which is iteratively updated based on the expression of characteristic time $\mathcal{C}^\star(f)$ in Theorem 6.3.2. Sampling stops at round τ_{MisLid} so that

$$\tau_{\text{MisLid}} := \inf_{t>0} \left\{ \inf_{f' \in \text{Alt}(\widetilde{f}_t)} \|\widetilde{f}_t(X) - f'(X)\|^2_{D_{N.(t)}} > 2\mathcal{T}_\delta(t) \right\} ,$$

where $\widetilde{f}_t \in \mathcal{M}_\Psi$ is a model function which corresponds to the projection of the vector of empirical means $(\widehat{\mu}_a(t))_{a\in[K]}$ onto the set of Ψ-misspecified model functions ; $\mathcal{T}_\delta(\cdot)$ is a time-dependent threshold function to be defined. Remember that the empirical mean $\widehat{\mu}_a(t)$ for arm $a \in [K]$ at round t is defined as

$$\widehat{\mu}_a(t) := \frac{1}{N_a(t)} \sum_{s\leq t} \mathbb{1}(I_s = a) Y_t ,$$

where $N_a(t)$ is the number of selections of arm a up to round t. For short, we denote $\widetilde{f}_t(X)$ vector $(\widetilde{f}_t(X_a))_{a\in[K]}$. Provided that function $\mathcal{T}_\delta(\cdot)$ is well-chosen, the stopping rule ensures that MisLid is δ-correct, and asymptotically optimal, as defined in Result 4.3.4

$$\begin{aligned} \inf_{f' \in \text{Alt}(\widetilde{f}_t)} \|\widetilde{f}_t(X) - f'(X)\|^2_{D_{N.(t)}} &= \inf_{f' \in \text{Alt}(\widetilde{f}_t)} (\widetilde{f}_t(X) - f'(X))^\top D_{N.(t)} (\widetilde{f}_t(X) - f'(X)) \\ &= \inf_{f' \in \text{Alt}(\widetilde{f}_t)} \sum_{a\in[K]} N_a(t) (\widetilde{f}_t(X_a) - f'(X_a))^2 . \end{aligned}$$

If the $(N_a(t))_{a\in[K]}$ were replaced by some allocation $\omega \in \triangle_K$ and $\widetilde{f}_t$ by the true model function f, the expression would have matched the definition of $\mathcal{C}^\star$ in the lower bound for Gaussian bandits with fixed variance $\sigma^2 = 1$, up to constant $1/2$ (Equation (6.2)). This quantity is the statistic associated with a parallel generalized likelihood ratio test (GLRT), as described in Garivier and Kaufmann (2021). [4] Here, the GLRT is a sequential statistical test of the following two non-overlapping hypotheses

$$\mathcal{H}_0 : \left(\mathcal{S}^\star_{N,f} \not\subseteq \mathcal{S}^\star_{N,\widetilde{f}_t}\right) \text{ (null hypothesis)} , \text{ versus } \mathcal{H}_1 : \left(\mathcal{S}^\star_{N,f} \subseteq \mathcal{S}^\star_{N,\widetilde{f}_t}\right) \text{ (alternative)} ,$$

and rejects the null hypothesis when the statistic is large enough. If $\mathcal{H}_0$ is rejected, then we cannot rule out the possibility that the top-N arms under current empirical model $\widetilde{f}_t$ includes the true set $\mathcal{S}^\star_{N,f}$. At the end of the sampling phase in round τ_{MisLid}, MisLid returns the N-best arms of the projected estimated mean vector $\widetilde{f}_{\tau_{\text{MisLid}}}(X)$. I will now explicit the different steps in Algorithm 7.

[4] However, GLRTs date back at least to Wilks (1938).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Algorithm 7 MisLid algorithm for misspecified models.

Require: Online learner $\mathcal{L}$, stopping thresholds $\{\mathcal{T}_\delta(t)\}_{t\geq 1}$

Initialization
$L \leftarrow \max_{a\in[K]} \|X_a\|_2$
Compute a sequence of arms $I_1, I_2, \ldots, I_{t_0}$ such that

$$\sum_{t=1}^{t_0} X_{I_t} X_{I_t}^T \succeq 2L^2 I_d$$

for $t = 1, \ldots, t_0$ **do**
  # Pull spanner
  Pull I_t, receive Y_t, and set $\omega(t) \leftarrow (\mathbb{1}(i = I_t)_{i\in[K]}$
end for
Compute empirical mean $\widehat{\mu}_\cdot(t_0) := (\widehat{\mu}_a(t_0))_{a\in[K]}$ and the model function associated with its projection onto the set of Ψ-misspecified models

$$\widetilde{f}_{t_0} \leftarrow \arg\min_{f'\in\mathcal{M}_\Psi} \|f'(X) - \widehat{\mu}_\cdot(t_0)\|^2_{D_{N,(t_0)}} .$$

for $t = t_0+1, t_0+2, \ldots,$ **do**
  # Stopping rule
  if $\inf_{f'\in\mathsf{Alt}(\widetilde{f}_{t-1})} \|\widetilde{f}_{t-1}(X) - f'(X)\|^2_{D_{N,(t-1)}} > 2\mathcal{T}_\delta(t-1)$ **then**
    Stop and return $\hat{\mathcal{S}}_N(\tau_{\mathsf{MisLid}}) := \mathcal{S}^\star_{N,\widetilde{f}_{\tau_{\mathsf{MisLid}}}}$
  end if
  Obtain $\omega(t)$ from learner $\mathcal{L}$
  Compute closest alternative $f(t) \leftarrow \arg\min_{f'\in\mathsf{Alt}(\widetilde{f}_{t-1})} \|\widetilde{f}_{t-1}(X) - f'(X)\|^2_{D_{\omega_t}}$
  # Update learner
  Update $\mathcal{L}$ with gain

$$g(t) : \omega \mapsto \sum_{k\in[K]} \omega_k \left(\left| \widetilde{f}_{t-1}(X_k) - f(t)(X_k) \right| + \sqrt{c(t-1)_k} \right)^2$$

  # Action sampling
  Pull $I_t \sim \omega(t)$ and receive reward Y_t
  # Estimation
  Update $\widehat{\mu}_\cdot(t)$ and compute its projection $\widetilde{f}_t \leftarrow \arg\min_{f'\in\mathcal{M}_\Psi} \|f'(X) - \widehat{\mu}_\cdot(t)\|^2_{D_{N,(t)}}$
end for

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Initialization phase. Algorithm 7 starts by sampling a deterministic sequence of t_0 arms that make the minimum eigenvalue of the resulting design matrix $\hat{V}^\kappa(t_0)$ larger than $2L^2$. Since the rows of feature matrix X are assumed to span $\mathbb{R}^d$ (Section 6.2), such sequence can be easily found by taking any subset of d arms that span the whole space –*e.g.*, by computing a barycentric spanner. A barycentric spanner of $\mathbb{R}^d$ of size d is a set of d arms if any element in $\mathbb{R}^d$ can be expressed as a linear combination of these arms with coefficients in $[-1, 1]$. An approximation of this set can be computed in polynomial time (Awerbuch and Kleinberg, 2004). Once this barycentric spanner is built, arms in the spanner are sampled in a round robin fashion until the desired condition is met. This is required to make design matrix $\hat{V}^\kappa(\cdot)$ invertible, and to ensure that concentration inequalities hold. While prior works typically avoid this step by regularizing (Abbasi-Yadkori, Pál, and Szepesvári, 2011), in our misspecified setting it is crucial not to do so to obtain tight concentration results for the estimator of $f(X)$. In order to get an upper bound on the length t_0 of the initialization phase, let us denote $\sigma_{\min}(M)$ the minimal singular value of matrix M. Let us consider $\mathcal{B} = \{b_1, b_2, \dots, b_d\} \subseteq [K]$, such that $\mid B \mid = d$, the barycentric spanner of size d computed on feature matrix X. Then, if we stopped the round-robin sampling at round t_0 such that each arm in the barycentric spanner is sampled a number u_0 of times,

$$\hat{V}^\kappa(t_0) = u_0 \sum_{b \in \mathcal{B}} X_b X_b \ .$$

The following condition is enough to ensure that $\hat{V}^\kappa(t_0) \succeq 2L^2 I_d$ –that is, that matrix $(\hat{V}^\kappa(t_0) - 2L^2 I_d)$ is positive definite

$$u_0 \sigma_{\min}\left(\sum_{b \in \mathcal{B}} X_b X_b^\top\right) \geq 2L^2 \ . \tag{6.3}$$

Let us denote $\Gamma'_d(X) := \min_{B \ d\text{-sized spanner}} \sigma_{\min}(\sum_{b \in B} X_b X_b)$. Then $u_0 = \left\lceil \frac{2L^2}{\Gamma'_d(X)} \right\rceil$ satisfies Condition (6.3), and then $t_0 \leq d \left\lceil \frac{2L^2}{\Gamma'_d(X)} \right\rceil$.

Remark 6.4.1. *When the condition number of the matrix is very large –e.g., feature vectors are collinear– this initialization phase can be time-consuming. Specifically in that case, feature selection (or transformation) prior to the application of bandits is of paramount importance.*

Estimation. At each round $t \geq t_0$, MisLid updates an estimator $\widetilde{f}_t$ of the true bandit model f, by first computing the empirical mean for each arm a

$$\widehat{\mu}_a(t) := \frac{1}{N_a(t)} \sum_{s \leq t} \mathbb{1}(I_s = a) Y_t \ .$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Then this empirical model is projected onto the set of Ψ-misspecified models, according to the norm weighted by the individual number of draws $N_a(t)$ of each arm a

$$\widetilde{f}_t := \underset{f' \in \mathcal{M}_\Psi}{\arg\min} \, \|f'(X) - \widehat{\mu}_\cdot(t)\|_{D_{N_\cdot(t)}} \, .$$

In Réda, Tirinzoni, and Degenne (2021), this projection can be computed efficiently as two independent quadratic optimization problems, one for each of the two parameters θ and η.

Stopping rule. MisLid uses the standard stopping rule adopted in many existing algorithms for pure exploration (Degenne, Koolen, and Ménard, 2019; Garivier and Kaufmann, 2016; Shang et al., 2020), which is based on generalized likelihood ratio tests. But the difficulty lies in its calibration, that is, the choice of an appropriate threshold function $\mathcal{T}_\delta(\cdot)$ for which the algorithm is δ-correct. MisLid requires a careful combination of concentration inequalities. First, for linear bandit models, to make the algorithm adapt well to (quasi) linear models with low misspecification Ψ ; and, second, for unstructured bandits, to guarantee asymptotic optimality in the sense of Result 4.3.4. The precise definition of $\mathcal{T}_\delta(\cdot)$ is given in the following result.

Theorem 6.4.2. ***Correctness of* MisLid.** *For the following expression of $\mathcal{T}_\delta(\cdot)$, where $\delta \in (0,1)$, Algorithm 7 is δ-correct in the sense of Definition 4.1.4*

$$\forall t \geq 0, \;\; \mathcal{T}_\delta(t) := \min\left(\beta_\delta^{uns}(t), \beta_\delta^{lin}(t)\right) \;\; , \;\; \textit{where}$$

$$\beta_\delta^{uns}(t) := 2K\overline{W}\left(\frac{1}{2K}\log\left(\frac{2\exp(1)}{\delta}\right) + \frac{1}{2}\log\left(8\exp(1)K\log(t)\right)\right) \; ,$$

$$\beta_\delta^{lin}(t) := \left(\frac{4}{\sqrt{2}}\sqrt{t}\Psi + \sqrt{1 + \log(\delta^{-1}) + \left(1 + \frac{1}{\log(\delta^{-1})}\right)\frac{d}{2}\log\left(1 + \frac{t}{2d}\log(\delta^{-1})\right)}\right)^2 \; ,$$

and $\overline{W}(x) := -W_{-1}(-\exp(-x)) \approx x + \log(x)$,

where W_{-1} is defined such that, if $z \in [-\exp(-1), 0)$, $x \leq -1$, and $x = W_{-1}(z)$ then $x\exp(x) = z$ (W_{-1} is the negative branch of the Lambert W function) .

Note that this stopping rule crucially relies on the knowledge of misspecification level Ψ.

Proof sketch. The objective is to show that MisLid is δ-correct on Ψ-misspecified models, that is, according to Definition 4.1.4, for any $f \in \mathcal{M}_\Psi$, $\mathbb{P}_f\left[\hat{\mathcal{S}}_{N,f}(\tau_{\text{MisLid}}) \not\subseteq \mathcal{S}^\star_{N,f}\right] \leq \delta$.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

If $\tau = \tau_{\mathsf{MisLid}}$, then

$$\mathbb{P}_f\left[\hat{\mathcal{S}}_{N,f}(\tau) \not\subseteq \mathcal{S}^\star_{N,f}\right] = \mathbb{P}_f\left[\inf_{f' \in \mathsf{Alt}(\widetilde{f}_\tau)} \|\widetilde{f}_\tau(X) - f'(X)\|^2_{D_{N.(\tau)}} > 2\mathcal{T}_\delta(\tau) \text{ and } \widetilde{f}_\tau \in \mathsf{Alt}(f)\right]$$

And then

$$\begin{aligned}
\mathbb{P}_f\left[\hat{\mathcal{S}}_{N,f}(\tau) \not\subseteq \mathcal{S}^\star_{N,f}\right] &\leq \mathbb{P}_f\left[\exists t > 0, \inf_{f' \in \mathsf{Alt}(\widetilde{f}_t)} \|\widetilde{f}_t(X) - f'(X)\|^2_{D_{N.(t)}} > 2\mathcal{T}_\delta(t) \text{ and } \widetilde{f}_t \in \mathsf{Alt}(f)\right] \\
&\leq \mathbb{P}_f\left[\exists t > 0, \inf_{f' \in \mathsf{Alt}(\widetilde{f}_t)} \|\widetilde{f}_t(X) - f'(X)\|^2_{D_{N.(t)}} > 2\mathcal{T}_\delta(t)\right] ,
\end{aligned}$$

By the definition of threshold function $\mathcal{T}_\delta(\cdot)$ and applying a union bound

$$\begin{aligned}
\mathbb{P}_f\left[\hat{\mathcal{S}}_{N,f}(\tau) \not\subseteq \mathcal{S}^\star_{N,f}\right] &\leq \underbrace{\mathbb{P}_f\left[\exists t > 0, \inf_{f' \in \mathsf{Alt}(\widetilde{f}_t)} \|\widetilde{f}_t(X) - f'(X)\|^2_{D_{N.(t)}} > 2\beta^{\mathsf{uns}}_\delta(t)\right]}_{\leq \delta/2} \\
&\quad + \underbrace{\mathbb{P}_f\left[\exists t > 0, \inf_{f' \in \mathsf{Alt}(\widetilde{f}_t)} \|\widetilde{f}_t(X) - f'(X)\|^2_{D_{N.(t)}} > 2\beta^{\mathsf{lin}}_\delta(t)\right]}_{\leq \delta/2} \\
&\leq \delta ,
\end{aligned}$$

where the last two inequalities respectively stem from the unstructured and the linear concentration inequalities derived in Réda, Tirinzoni, and Degenne (2021, Appendix $F.1$).

Sampling strategy and online learners. The sampling strategy of MisLid aims at achieving the optimal sample complexity from the lower bound from Theorem 6.3.2. Instead of using a tracking procedure as done in Garivier and Kaufmann (2016), MisLid uses an online regret minimization learner $\mathcal{L}$, which incrementally estimates the optimal allocation $\omega \in \triangle_K$ in the expression of the characteristic time in Theorem 6.3.2. [5] Arm I_t is selected according to this allocation $\omega(t)$. Learner $\mathcal{L}$ iteratively updates its estimation $\omega(t)$ at round t through gains $g(t)$. Gains translate the regret incurred by a bad approximation of the true allocation ω. If we knew the true model f, we would use the gradient at $\omega(t)$ of the following gain function to quantify the

[5] The use of online learners is a standard approach in bandits ; however, their description is outside the scope of this thesis. The interested reader can refer to De Rooij et al. (2014).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

regret incurred by estimate $\omega(t)$

$$g^\star : \omega \mapsto \inf_{f' \in \mathsf{Alt}(f)} \|f(X) - f'(X)\|^2_{D_\omega} \,.$$

Since f is not available, we settle for an upper bound on $\arg\min_{f' \in \mathsf{Alt}(\widetilde{f}_t)} \|\widetilde{f}_t(X) - f'(X)\|_{D_\omega}$ for any $\omega \in \triangle_K$, using minimizer $f(t)$, and optimistic bonuses $c(t) \in \mathbb{R}^K$ such that, with high probability

$$\forall \omega \in \triangle_K \ \forall t > 0, \ g_t(\omega) = \sum_{k \in [K]} \omega_k \left(\mid \widetilde{f}_t(X_k) - f(t)(X_k) \mid + \sqrt{c(t)_k} \right)^2 \geq g^\star(\omega) \,.$$

In pratice, $c(t)_k$ is defined for any arm $k \in [K]$ as follows

$$c(t)_k := \min \left\{ 8(LK+1)^2 \Psi^2 + 4\alpha^{\mathsf{lin}}_{t^2} \|X_k\|^2_{(\hat{V}^\kappa(t))^{-1}}, \frac{2\alpha^{\mathsf{uns}}_{t^2}}{N_k(t)}, 4C^2 \right\} \,,$$

where $\alpha^{\mathsf{uns}}_t := \beta^{\mathsf{uns}}_{1/(5t^3)}(t)$ and $\alpha^{\mathsf{lin}}_t := \log(5t^2) + d\log(1 + t/(2d))$. As briefly mentioned in Section 6.3, using an online learner avoids having to rely on min-max oracles which might be computationally inefficient –due to the maximization over the simplex $\triangle_K$– and might be prone to numerical approximations that prevent convergence.

Given these parameters, Algorithm MisLid is provably asymptotically optimal (in the sense of Result 4.3.4). The full proof of the theorem is available in the supplementary section of Réda, Tirinzoni, and Degenne (2021).

Theorem 6.4.3. ***Sample complexity of*** **MisLid.** *The following inequality about the expected sample complexity of* MisLid *holds*

$$\mathbb{E}_{\{\mu\}}[\tau_{\mathsf{MisLid}}] \leq \tau^\delta + 2 \,,$$

where τ^δ satisfies the following inequality in t ($\ell_t := \log(t)$ and $\widehat{\mathcal{O}}$ is the sum of $\mathcal{O}$ of each argument)

$$\mathcal{T}_\delta(t) \geq t\mathcal{C}^\star + \widehat{\mathcal{O}}\left(\min\{tK^2\Psi^2 + d\sqrt{t}\ell_t, \sqrt{Kt}\ell_t\}; \log K\sqrt{t}; \sqrt{\min\{tK^2\Psi^2 + d\ell_t, K\ell_t\}} \log(1/\delta) \right) .$$

In particular, for $\Psi \approx 0$ (quasi linear models)

$$\tau^\delta \leq (\mathcal{C}^\star)^{-1} \left[\log(\delta^{-1}) + \left(\Psi^2\tau^\delta + \log(K)\sqrt{\tau^\delta} + d\sqrt{\tau^\delta}\log(\tau^\delta) + \sqrt{d\log(\tau^\delta)} \right) o(\log(\delta^{-1})) \right] \,,$$

and for $\Psi \gg 0$ (close to unstructured models)

$$\tau^\delta \leq (\mathcal{C}^\star)^{-1} \left[\log(\delta^{-1}) + \left(\log(K)\sqrt{\tau^\delta} + \sqrt{K\tau^\delta}\log(\tau^\delta) + \sqrt{K\log(\tau^\delta)} \right) o(\log(\delta^{-1})) \right] \,.$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Based on Theorem 6.4.3, for small values of δ,

$$\tau^\delta \approx (\mathcal{C}^\star)^{-1} \log(\delta^{-1}) + C_\mu o(\log(\delta^{-1})) \,,$$

where C_μ is a problem-dependent constant. Then

$$\liminf_{\delta \to 0} \frac{\mathbb{E}_{\{\mu\}}[\tau_{\text{MisLid}}]}{\log(\delta^{-1})} = \liminf_{\delta \to 0} \frac{\tau^\delta}{\log(\delta^{-1})} = (\mathcal{C}^\star)^{-1} \,,$$

which confirms that MisLid is indeed asymptotically optimal. Note that when $\Psi \approx 0$ (*i.e.*, , the model is quasi linear) there is only a logarithmic dependence on the number of arms K, which is on par with the state-of-the-art (Jedra and Proutiere, 2020; Kirschner et al., 2021; Tirinzoni et al., 2020). Moreover, the bound exhibits an adaptation to the value of Ψ. As Ψ grows, the upper bound transitions to terms matching the optimal unstructured bound without any dependence in the number of dimensions d and deviation Ψ.

6.5 Application to drug repurposing

Since our algorithm is the first to solve Top-N identification in Ψ-misspecified models, we compare it against an efficient linear algorithm, N-LinGapE, introduced in Chapter 5 –which is an extension to algorithm LinGapE described in Xu, Honda, and Sugiyama (2018)– and an unstructured one, LUCB (Kalyanakrishnan et al., 2012). In all experiments, we consider $\delta = 10\%$ and $\sigma = 1$. It has frequently been noted in the fixed-confidence literature that stopping thresholds which guarantee δ-correctness tend to be too conservative and to yield empirical error frequencies that are actually much lower than δ. Moreover, these thresholds are different between linear, misspecified and unstructured bandit models. In order to ensure a good trade-off between performance and computing speed, and fairness between the tested algorithms, as in prior works (Kaufmann and Kalyanakrishnan, 2013; Réda, Kaufmann, and Delahaye-Duriez, 2021), we use the following heuristic value

$$\forall t > 0, \;\; \mathcal{T}_\delta(t) := \ln\left(\frac{1 + \ln(t+1)}{\delta}\right) \,.$$

We consider the drug repurposing problem on 10 antiepileptics and 11 proconvulsants mentioned in Chapter 5. However, contrary to what was previously done, this time we truly stick to the theoretical setting, and make the environment return a realization of a Gaussian distribution of fixed variance $\sigma^2 = 1$ when the agent samples arm $I_t \in [K]$

$$Y_t \sim \mathcal{N}(\mu_{I_t}, 1) \,.$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Moreover, knowing that our algorithm relies on a good knowledge of the maximum misspecification Ψ, we apply the same type of feature transformation procedure as in Chapter 5. We compute Ψ as the ℓ_∞ norm of the difference between the predictions of the regressed linear model, and the average drug repurposing scores. Selecting the most deviated model to linearity yields $\Psi = 0.236$ with $d = 21$ dimensions. Since the misspecification Ψ is a lot larger than the minimum gap $\min_{k\in[K]} \Delta_k$ –contrary to the model built in Chapter 5– this model is highly misspecified, which will allow us to observe the same behavior as in the experiment in Table 6.1.

Figure 6.1 shows one boxplot per algorithm, which reports the sample complexity on the y-axis, and the error frequency δ across 100 iterations rounding up to the 5th decimal place. Individual outcomes are shown as gray dots. In order to speed up LUCB, we consider the PAC version of Top-N identification –that is, $\varepsilon > 0$– choosing as stopping threshold $\varepsilon = 0.042 \approx \min_{k\in[K]} \Delta_k$ (for $N = 4$), so that the algorithm stops earlier while returning the exact set of N best arms.

As previously mentioned, we expect N-LinGapE to dramatically fail since the misspecification scale Ψ is large. Moreover, from the sample complexity bound derived in Theorem 6.4.3, we expect MisLid to have an average sample complexity at most as large as the one incurred by a good unstructured bandit algorithm. As shown by the boxplot in Figure 6.1, linear algorithm N-LinGapE dramatically fails, whereas LUCB (for unstructured models) and MisLid remain δ-correct. Moreover, even with the early stopping rule using $\varepsilon > 0$, in average LUCB needs 40% more samples than MisLid, which does not use the information about ε.

6.6 Discussion

We have designed the first algorithm to tackle misspecification in fixed-confidence Top-N identification. The proposed algorithm can be applied to misspecified models which can deviate from linearity (*i.e.*, $\Psi \geq 0$), which encompass both unstructured (for large values of Ψ) and linear models (*i.e.*, $\Psi = 0$).

The main limitation of MisLid is its computational complexity : at each round, $\mathcal{O}(KN)$ convex optimization problems need to be solved for both the sampling and stopping rules, which can be expensive if the number of arms is large. Moreover, since the sampling of our algorithm is designed to aim at a lower bound, we can expect it to suffer from the same shortcomings as

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

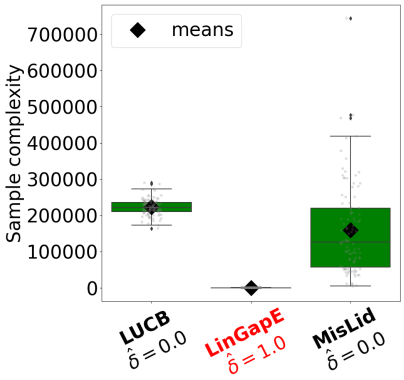

Figure 6.1: **Drug repurposing with misspecified bandits.** Drug repurposing instance with reduced dimension (#arms $K = 21$, #dimensions $d = 21$, $\sigma = 1$, $\delta = 10$ %, $\varepsilon = 0.042$, $\Psi = 0.236$, $N = 4$, 100 iterations) with empirical error rates $\hat{\delta}$.

Algorithm	$\hat{\delta}$	$\hat{s}$
MisLid	0%	$158,869 \pm 126,209$
LinGapE	100%	161 ± 159
LUCB	0%	$222,969 \pm 22,798$

Table 6.2: **Results for drug repurposing with misspecified bandits.** Table of results associated with the boxplots. $\hat{\delta}$ is the empirical error rate across 100 iterations, $\hat{s}$ is the average number of samples ($\pm$ the standard deviation) rounded up to the closest integer.

that bound. Indeed, it is known that the bound in question does not capture some lower order (in $\mathcal{O}(\delta^{-1})$) effects, in particular those due to the multiple-hypothesis nature of the test we perform, which can be very large for small times. Work to take these effects into account to design algorithms has started recently (Katz-Samuels, Jain, Jamieson, et al., 2020; Katz-Samuels and Jamieson, 2020; Wagenmaker, Katz-Samuels, and Jamieson, 2021) and we believe that it is an essential avenue for further improvements in pure exploration.

Finally, a good estimation of the maximum deviation to linearity, that we denoted Ψ, is crucial to obtain a sample-efficient algorithm. Unfortunately, this information is usually not available when dealing with real-life datasets. If the estimate $\widehat{\Psi}$ is smaller than the true value Ψ, then there is a risk of losing the property of δ-correctness ; however, if this estimate is too large, then the performance of MisLid should be similar to a good bandit algorithm for unstructured models. Finding a good procedure to heuristically estimate the deviation level Ψ –without having access to the expected rewards– at the price of some loss in accuracy in the estimation of the true scores could be as well an interesting subsequent work. Indeed, determining the relative ranking of arms is more important in drug repurposing than estimating *exactly* the scores. [6]

[6]Even though the latter can be useful for interpretability, as previously mentioned.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Part III

Application & extension of drug repurposing

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

The contents of this chapter rely on some of my publications.[a]

[a]Bokobza, Réda, et al. (*in prep.*). "Therapeutic evaluation of Hu-MSCs in a rat model of perinatal inflammation: a systematic outcome scoring"; Réda, Vakili, and Kaufmann (2022). "Near-Optimal Federated Learning in Bandits". 36th *Conference on Neural Information Processing Systems*. In press.

In this part of my thesis, I present two works related to the drug repurposing method I have proposed. Chapter 7 exhibits a instance of systematic treatment protocol ranking procedure through signature reversion when the appropriate conditions are reunited to build good quality signatures. Chapter 8 deals with the extension of any adaptive clinical trial to a collaborative setting.

In Chapter 7, I have collaborated with a team of biologists in order to select the optimal treatment protocol of stem cell injections in a rat model of infant encephalopathy. Transcriptional data obtained through sequencing comprised of samples subject to varying doses, modes and ages of injection. The goal was to rank the different treatment protocols depending on the estimated recovery of treated samples through signature reversion. This project is a proof-of-concept that present a systematic method to rank treatments based on transcriptomic data.

Chapter 8 deals with the theoretical setting where there are several heterogeneous subpopulations of patients which are recruited to join a collaborative adaptive clinical trial, in order to find the most interesting drug candidates *for their own population of patients*, while exploiting the results from other populations. This work is a step towards personalized clinical trials, which are a topical question in drug research, especially in cancer (Maitland and Schilsky, 2011), where there is a plethora of subtypes and biomarkers which are sometimes unique to a patient.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 7

Application of systematic drug scoring

This chapter focuses on an application of signature reversion in practice, in a particularly favorable context for computing drug signatures : indeed, the drug signatures will exactly represent the transcriptional changes due to the drug treatment applied *on patients*. The simulating procedure using the Boolean network will not be needed in this case. This application highlights how signature reversion allows to discriminate between treatments.

The work presented in this chapter aims at validating umbilical cord-derived human mesenchymal stem cells (MSCs) as a regenerative therapy for encephalopathy of prematurity (EoP), which is due to brain lesions in premature birth. Several treatment protocols –with varying weight-dependent doses, modes and ages of injection of MSCs– are performed on rat models of EoP, from which transcriptomic profiles are obtained, for control rats (which do not receive stem cells), rat models, and rat models treated with stem cells. The signature reversion method outputs a score for each treatment protocol, based on this transcriptomic data. These scores allow ranking treatment protocols according to their rescuing of a healthy transcriptomic profile. Further pathway analyses seem to confirm the impact of top-ranked treatment protocols on inflammation-related and developmental genes. A manuscript is in preparation for the Journal of Neuroinflammation (Bokobza, Réda, et al., *in prep.*).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

7.1 Related work

Preterm birth represents an estimated 15 million births every year, and related complications are the leading cause of death in children before the age of 5 in 2018, according to the World Health Organization (World Health Organization (WHO), 2018). Deliveries before 37 on 40 weeks of gestation lead to a constellation of neurodevelopmental injuries, regrouped under the term of encephalopathy of prematurity (EoP) (Bokobza, Van Steenwinckel, et al., 2019). White matter injury (WMI) is the most common brain injury associated to preterm birth. WMI is due to a myelination deficit during development. Myelination is an important step in nervous system development, as layers of myelin are produced and wrapped around the neuronal axons. This allows "insulating" the transmission of electric action potentials along the axon, and faster information processing and more complex brain processes. Myelin is produced by oligodendrocytes, that differentiate between 20 weeks of gestation to the first year of infant life. Preterm birth disturbs normal differentiation processes by blocking oligodendrocytes to a precursor state, which is unable to produce myelin. Moreover, microglial cells, which are a subtype of glial cells and the brain resident macrophages, produce cytotoxic molecules affecting precursor oligodendrocytes, which are highly vulnerable to such insults. This induces WMI through neuroinflammation in the white matter. In a good healthcare setting, more than half of babies who survive will suffer from lifelong disabilities, including cerebral palsy, severely impaired cognitive functions, and psychiatric disorders, such as attention-deficit and hyperactivity disorder, or autism spectrum disorder (Crump, Sundquist, and Sundquist, 2021). There is no existing treatment yet to repair brain damage incurred by EoP (Chung, Chou, and Brown, 2020).

In addition to their high regenerative capacity, mesenchymal stem cells (MSCs) [1] are reported as anti-inflammatory, and with a low immunogenicity (Passera et al., 2021), meaning that they do not trigger a strong immune response. Moreover, several studies report that MSCs and molecule-derived MSCs contribute to reduce microglial reactivity both *in vitro* and *in vivo* (Barati et al., 2019; Go et al., 2020; Liu, Zhang, et al., 2014). However, there is no current data which compiles effects of MSCs on microglial reactivity in a same animal model, at different time points. Indeed, for a common time of insult, MSCs will not correct the same microglial pathways depending on administration age –denoted by postnatal day (P) 5, P10, ..., which are known to model prenatal humans.

[1] That is, adult stem cells, which can be sourced from different body tissues.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

This project introduces a new model of EoP, on rats, by neuroinflammation induced by the injection of interleukin 1 beta (IL1β) at postnatal day 5 (P5). The characterization of the microglial transcriptome has already been carried out in an identical mouse model by Krishnan et al. (2017). In that work, they have shown the existence of clusters of genes involved with inflammatory processes hyper-regulated at postnatal day 1 (P1) and P5, as well as clusters negatively regulated by IL1β and grouping together developmental genes.

The objective of this project is to assess the optimal administration settings for MSCs injections in the rat model of EoP, in terms of weight-dependent dose, route of administration and age of administration. In order to achieve this, we consider using the paradigm of signature reversion, reviewed in particular in Musa et al. (2018), where post-treatment transcriptional levels are compared with control transcriptional profiles. The more similar these two groups of profiles are, the most interesting the associated treatment is. This method has already been successful, for instance, in drug repurposing against influenza, where candidates obtained through signature reversion have been validated *in vitro* (Xin et al., 2022). Combining this paradigm with the experimental transcriptional profiles generated for every selected treatment protocol allows us to rank the different treatment protocols depending on their similarity score to a healthy profile. This work was a collaboration with the Neurokines team (Inserm U1141), which performed the experimental part of this project, and the PREMSTEM consortium.

7.2 Setting and experimental data

All rat pups were intraperitoneally injected twice daily from post-natal day (P)1 to P4 and once in the morning of P5 with recombinant mouse IL1β, or the same volume of phosphate-buffered saline (PBS). Figure 7.1 shows the timeline of the experimental part. Rats were injected once with MSCs, according to their treatment protocol. Table 7.1 shows all parameters for the samples considered for treatment outcome scoring. In particular, note that there are 6 sequencing batches, with $N = 3$ replicates per condition. A condition represents the set of the following parameters : dose level, age and mode of administration, and treatment. IL1β samples are the rat models of EoP, whereas whereas PBS samples are the rat control group (without any neuroinflammation). The rats with a neuroinflammation induced by the injections of IL1β, and then treated with MSCs are considered the treated groups. Note that, since the doses of MSCs are weight-dependent, since we compare different ages of administration, the actual doses can be very dissimilar across groups. The numbering of doses 1, 2, 3 respectively stand for low, moderate and high doses of MSCs. In a nutshell, in addition to

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

the injection with PBS, IL1β and IL1β followed by a treatment with MSCs, we consider three different dose levels (low, moderate and high doses), two administration routes (intranasal and intravenous), and three ages of administration (postnatal days 5, 10 and 20).

Age*	Dose (#cells)	Age**	Mode* †	Treatment	#repl.	Batch
P5		P7	INAS	PBS,	3,	1
				IL1β,	3,	
	Dose 1 : $200,000$			IL1β+Dose 1,	3,	
	Dose 2 : $500,000$			IL1β+Dose 2,	3,	
	Dose 3 : $1,000,000$			IL1β+Dose 3	3	
			IV	PBS, IL1β,	3,	2
				IL1β,	3,	
				IL1β+Dose 1,	3,	
				IL1β+Dose 2,	3,	
				IL1β+Dose 3	3	
P10		P12	INAS	PBS,	3,	3
				IL1β,	3,	
	Dose 1 : $350,000$			IL1β+Dose 1,	3,	
	Dose 2 : $900,000$			IL1β+Dose 2,	3,	
	Dose 3 : $2,000,000$			IL1β+Dose 3	3	
			IV	PBS, IL1β,	3,	4
				IL1β,	3,	
				IL1β+Dose 1,	3,	
				IL1β+Dose 2,	3,	
				IL1β+Dose 3	3	
P20		P22	INAS	PBS,	3,	5
				IL1β,	3,	
	Dose 1 : $750,000$			IL1β+Dose 1,	3,	
	Dose 2 : $2,000,000$			IL1β+Dose 2,	3,	
	Dose 3 : $4,500,000$			IL1β+Dose 3	3	
			IV	PBS,	3,	6
				IL1β,	3,	
				IL1β+Dose 1,	3,	
				IL1β+Dose 2,	3,	
				IL1β+Dose 3	3	

Table 7.1: **Experimental batches for rat transcriptional profiles.** Samples used for the outcome scoring. * of administration of human MSCs. ** of microglial cell sorting before RNA sequencing. † INAS stands for intranasal, whereas IV stands for intravenous. "repl" stands for "replicates".

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

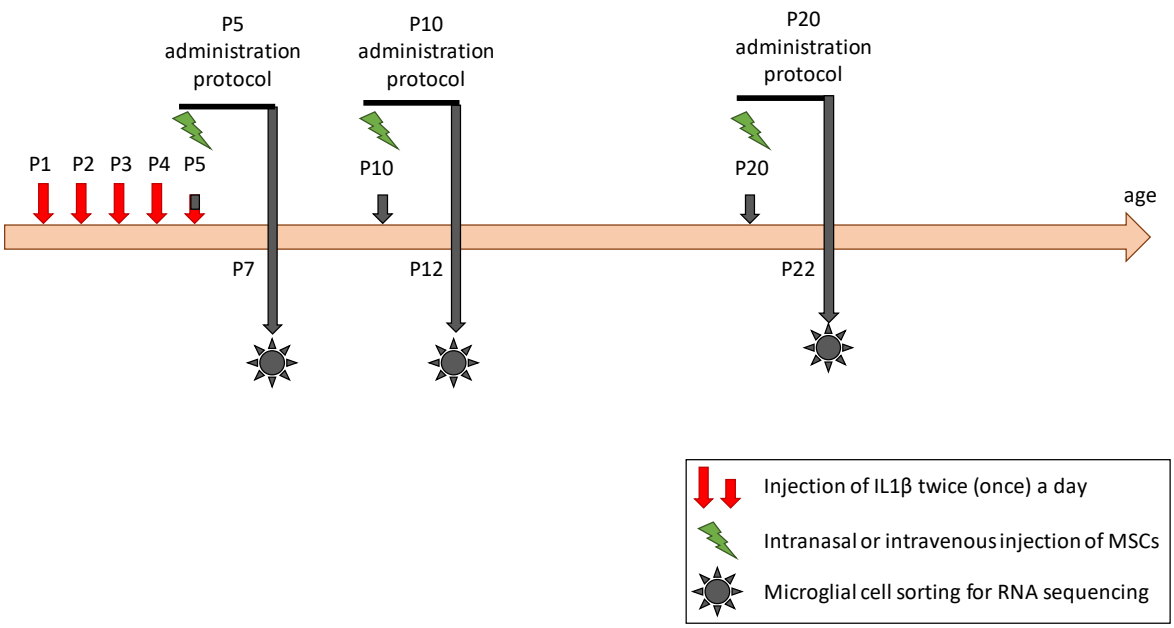

Figure 7.1: **Timeline of the experiments.** Timeline of the experimental part.

7.3 Systematic outcome scoring

Once the transcriptional profiles from all condition groups in Table 7.1 were generated, I applied the principle of signature reversion in order to rank the different condition groups depending on their ability to rescue the IL1β-induced inflammatory profiles. In signature reversion, the *in silico* identification of the most appropriate treatment protocols relies on a scoring of the changes at gene expression level. These changes are computed between the samples treated associated with a single condition, and a carefully determined reference group. As mentioned in previous chapters (Chapter 3), changes in expression for a group of genes between treated and reference groups can be quantified by a vector of numbers called signature, which is as large as the number of genes in the group. A signature accounts for the magnitude and the direction of the genewise expression change that is only due to this treatment. It can be used to characterize the treatment effect at transcriptional level. In particular, the magnitude of the change in expression of gene g is inferred from the absolute value of the coefficient at position g in the signature. Likewise, the direction of the change of expression of gene g –up-regulation versus down-regulation– is encoded in the sign of this very same coefficient. Genes which are not subject to expression changes are either absent from the signature, or present with a coefficient equal to 0. Actually, signatures can be built from the results of a (univariate) differential expression analysis. In that case, the absolute

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

values of coefficients are $\log_2$ fold-changes : *i.e.*, the binary logarithm of the ratio of gene expression levels in the treated and the reference groups. The set of genes associated with nonzero coefficients is restricted to significantly differentially expressed genes –*i.e.*, associated with an adjusted [2] p-value lower than some threshold, for instance 5%. However, we will consider here another method for differential expression analysis, which is not univariate. Indeed, we suppose that the changes in expression are mainly driven by groups of genes rather than individual genes. Leaving out the group interaction might incur important information loss about the perturbed pathways, as we will see in Section 7.4.

For each treatment protocol, the goal is to compute the similarity of the associated treatment signature to a reference signature. This reference signature is defined as the signature computed between corresponding PBS-control rats and IL1β-EoP model samples ; it represents the transcriptional changes expected from a good treatment protocol that is able to rescue the disease phenotype.

Figure 7.2 illustrates the ranking procedure. In order to build a signature for a given treatment protocol T of MSCs injection (Step $1.a$ in Figure 7.2), we rely on a computational method called Characteristic Direction (CD) (Clark et al., 2014). [3] To build the treatment signature, this method takes as input RNA-sequencing (DESeq2 (Love, Huber, and Anders, 2014) normalized) profiles from two condition groups in the same sequencing batch. The first group is the IL1β-EoP model samples treated with T, and the second group comprises of IL1β-EoP model samples without any injection of MSCs. We denote the resulting vector CD$[T\|\text{IL}1\beta]$. The reference signature, denoted CD$[\text{PBS}\|\text{IL}1\beta]$, to which treatment signature CD$[T\|\text{IL}1\beta]$ should be compared, is built in a similar way (Step $1.b$). CD is run on the condition groups of PBS-control samples and IL1β-EoP model samples without any injection of MSCs *from the same sequencing batch* as the previous two groups. Ensuring to use samples from the same sequencing aims at limiting the effect of the sequencing batch, which could otherwise be a confounding factor for the effect of the treatment. Moreover, the correction of batch effects from

[2]The adjustment is necessary to compare p-values resulting from a multiple hypothesis testing, as done in a univariate differential expression analysis, here, one hypothese per gene.

[3]Characteristic Direction was introduced in Chapter 2. In a nutshell, signature CD$[\text{G}1\|\text{G}2]$, computed through Characteristic Direction for two condition groups G1 and G2, is the vector normal to the decision frontier in a high-dimensional space, which classifies samples into either G1 or G2, and that is oriented in the direction from G2 to G1. That means in particular that the changes reported in the signature are changes in G1 compared to the reference group G2. Associated empirical p-values, relying on data permutations, might be computed using this method to assess the significance of the change in expression, and we used threshold 5% and 100 permutations in the computations.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

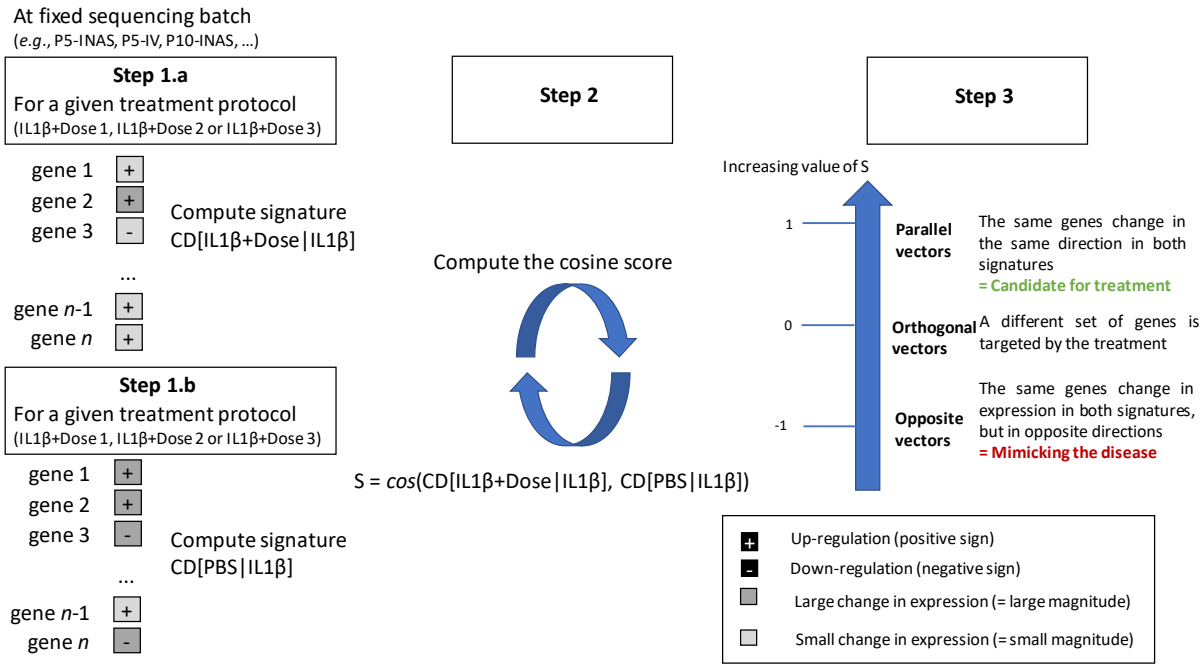

Figure 7.2: **Application of signature reversion to encephalopathy of prematurity.** The procedure of scoring treatments protocols using signature reversion.

raw data is a tricky procedure, which might lead to serious signal loss. This is why computations were run on normalized expression matrices (per batch), instead of batch-corrected raw expression data. Finally, in order to assign a score to treatment protocol T (Step 2), we compute the cosine similarity between the signature $\mathcal{S}_T = \text{CD}[T\|\text{IL}1\beta]$, corresponding to T, and the associated reference signature $\mathcal{R} = \text{CD}[\text{PBS}\|\text{IL}1\beta]$ on the same set of genes $\mathcal{G}$ –even if it means adding zero coefficients to the signatures. This cosine score is defined as

$$\cos\left(\mathcal{S}_T, \mathcal{R}\right) := \frac{\sum_{g\in\mathcal{G}} \mathcal{S}_T[g] \times \mathcal{R}[g]}{\sqrt{\sum_{g\in\mathcal{G}} \mathcal{S}_T[g]}\sqrt{\sum_{g\in\mathcal{G}} \mathcal{R}[g]}} \,. \tag{7.1}$$

This score is comprised between -1 (strong dissimilarity) and 1 (strong similarity). As shown in Step 3 of the figure, the underlying idea is that, the higher the similarity score is, the most the injection of MSCs is able to reproduce transcriptional genewise changes, in the right direction, from the diseased to a healthy phenotype.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

7.4 Results

The final ranking of treatment protocols is shown in Table 7.2. First, this ranking shows that the method described in Section 7.3 seems robust to batch effect –meaning that groups treated with different doses at the same age and through the same mode of administration may be associated with dissimilar scores ; for instance, samples treated at P20 by intranasal injection yield score 0.80 when treated with a moderate weight-dependent dose, 0.51 when treated with a high dose, and 0.08 when treated with a low dose. This brings us to a second claim : dose does not have a monotonic effect on the reversion of the inflammation-induced gene expression change. We can notice, as a sanity check, that most of the scores are indeed positive, some being even rather close to 1 (up to 0.80). This supports the fact that most human MSC injections have a positive, therapeutic effect at transcriptomic level. One can notice that treatments with the intranasal route seem to rank relatively higher than treatments with intravenous injections.

Batch number	Age	Mode	Dose level	Dose name	Score
5	P20	INAS	Moderate	Dose 2	0.80
3	P10	INAS	High	Dose 3	0.79
1	P5	INAS	High	Dose 3	0.68
6	P20	IV	Low	Dose 1	0.55
1	P5	INAS	Moderate	Dose 2	0.54
5	P20	INAS	High	Dose 3	0.51
3	P10	INAS	Low	Dose 1	0.36
3	P10	INAS	Moderate	Dose 2	0.33
1	P5	INAS	High	Dose 3	0.31
6	P20	IV	High	Dose 3	0.27
4	P10	IV	Low	Dose 1	0.24
6	P20	IV	Moderate	Dose 2	0.20
2	P5	IV	High	Dose 3	0.11
4	P10	IV	Moderate	Dose 2	0.08
5	P20	INAS	Low	Dose 1	0.08
4	P10	IV	High	Dose 3	0.01
2	P5	IV	Low	Dose 1	−0.02
2	P5	IV	Moderate	Dose 2	−0.12

Table 7.2: **Ranking of the treatment protocols.** Ranking of the treatment protocols (dose, age and mode of administration) resulting from the method described in section 7.3. Scores are rounded up to the 2nd decimal place.

We checked our assumption on the impact on gene expression made in Section 7.3 : that is, the fact that changes are rather driven by groups of genes than by individual genes. In order to do so, instead of building signatures based on Characteristic Direction (Clark et al., 2014), we considered signatures which coefficients were $\log_2$ fold-change values associ-

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

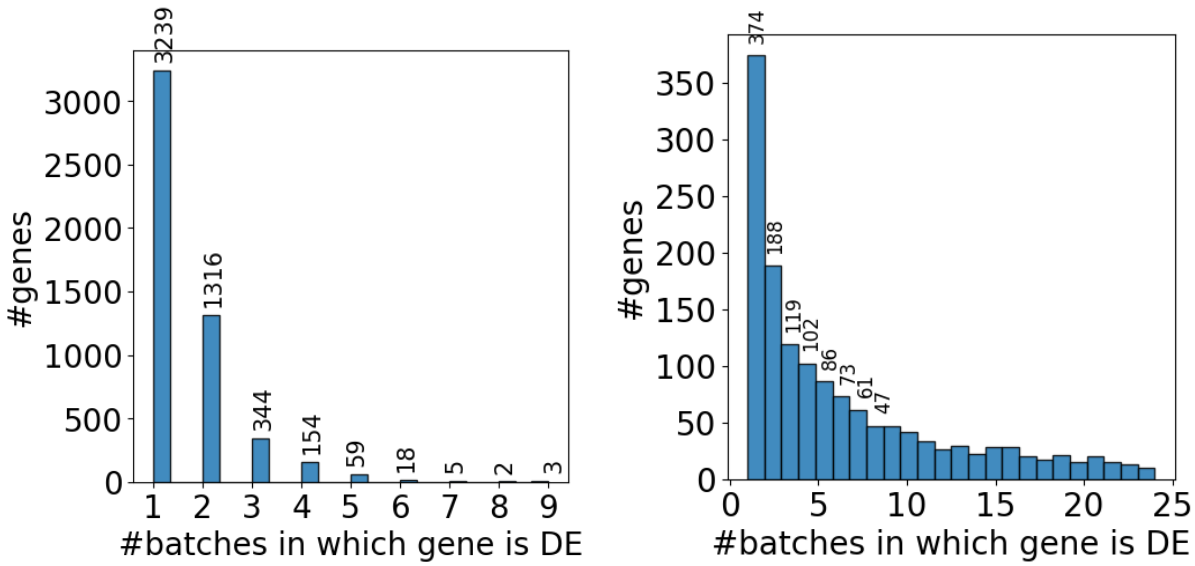

Figure 7.3: **Comparison between Characteristic Direction signatures and univariate differential analysis.** Comparison between the numbers of common differentially expressed (DE) genes across groups of conditions, when differential gene expression analysis is performed through DESeq2 (Love, Huber, and Anders, 2014) (*top plot*) or through Characteristic Direction (Clark et al., 2014) (*bottom plot*). DE genes are genes which are associated with nonzero coefficients in the signatures. The number of genes contained in the most populous bars are reported in both plots.

ated with Benjamini-Hochberg-adjusted p-values (Benjamini and Hochberg, 1995) lower than 5%, as derived with the classic (univariate) differential analysis approach DESeq2 (Love, Huber, and Anders, 2014). The top plot in Figure 7.3 shows that, for DESeq2-derived signatures, few genes were in common (that is, with a nonzero coefficient in the signatures) across condition groups. That means that cosine scores computed on these signatures have a value close to 0. As such, these cosine score are weakly informative of the relevance of a given treatment protocol. To the contrary, the bottom plot of Figure 7.3 –that considers signatures computed through Characteristic Direction– shows that there is a small core subset of genes which have a nonzero coefficient in all signatures. These genes are *Sparc*, *Csf1r*, *Lgmn*, *Ctsb*, *Cst3*, *AABR07006310.1*, *Mt-co1*, *Hexb*, *Ttr*, and *Apoe*. It means that every cosine score at least relies on changes on this core set ; such, the score is indeed reflective of some transcriptional changes due to the treatment. These two plots highlight the fact that reversal in expression is essentially driven by groups of genes, and confirm our choice of using Characteristic Direction to build signatures.

Moreover, since only nonzero coefficients are involved in the computation of the cosine score (Equation (7.1)), one might think that there might be a bias of higher cosine scores for treatment protocols which signature has

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

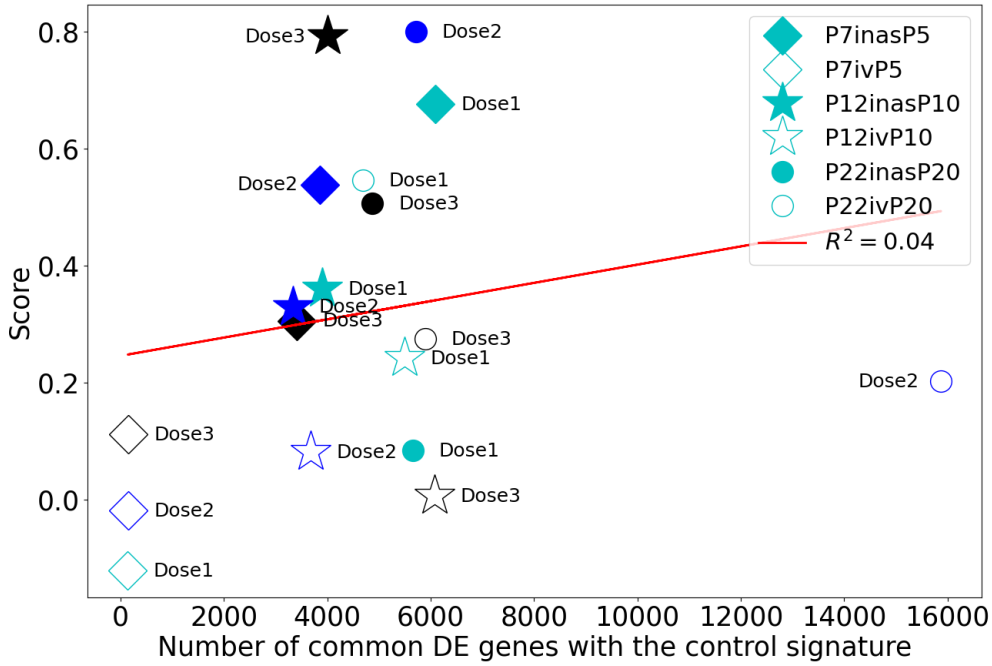

Figure 7.4: **Correlation between score and number of common differentially expressed genes.** Plot of the cosine score depending on the number of common differentially expressed (DE) genes of the associated signature (CD[IL1β + treatment||IL1β]) with the control signature (that is, CD[PBS||IL1β]). R^2 is the residual error of the linear regression (red line).

a large number of common differentially expressed genes with the control signature CD[PBS||IL1β]. We have tested this assumption by running a Spearman's ρ correlation test between the scores and the number of common differentially expressed genes with the control group. The test showed no significative linear correlation between the scores and the number of genes on which this score is computed (Spearman's ρ : 0.19, $p = 0.44$). Moreover, a linear regression from the number of common differentially expressed genes to the score has shown a bad fit (quantified by the residual $R^2 = 0.04$), as displayed by Figure 7.4. It is still interesting to note that this plot most likely gives away the reason why groups from the batch treated with intravenous injections at P5 are associated with low, sometimes even negative, scores in Table 7.2 ; the signatures associated with these groups have only a few genes in common with the control signature CD[PBS||IL1β]. As such, the support of genes on which the associated scores are computed might not be informative enough.

Finally, in order to assess the relevance of the Top-2 candidates, that is, P20-INAS-Dose 2 and P10-INAS-Dose 3, I performed a pathway analysis on the Top-2 treatment protocols. Both candidates are associated with a score around 0.80 in Table 7.2. A pathway analysis allows the identification of the biological pathways which are most perturbed by these treatment protocols. I considered the over-representation analysis (ORA) (Yaari et al., 2013),

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

which determines which sets of functionally similar genes –if there are any– are statistically surrepresented among a given subset of genes. Surrepresentation of a gene subset is computed by comparing the gene function categories represented in the subset (enrichments) to those present among a background set of genes, which contains the subset of interest. The gene function categories are annotated in Gene Onthology (GO) (Ashburner et al., 2000). In these annotations, genes are regrouped by biological processes in which they are involved (category GO Biological Process NoRedundant). The statistical tests –one per gene function category– were run by the online tool WebGestalt (Liao et al., 2019).

The gene subset associated with a treatment protocol T was built from the list of one-to-one orthologuous human genes with nonzero coefficients in the associated signature CD[T‖IL1β]. [4] We consider the human genes instead of rat genes because the gene categories for *Homo sapiens* are more robust. We also built two other gene lists, which respectively contains genes with positive (up-regulated), resp. negative (down-regulated), coefficients in the signature. This procedure output three lists of genes per treatment protocol : $5,263$ genes for P20-INAS-Dose 2, with $2,631$ up-regulated genes and $2,632$ down-regulated genes ; $6,233$ genes for P10-INAS-Dose 3, with $3,534$ up-regulated genes, and $2,699$ down-regulated genes. The background set of genes comprises of the $16,699$ human genes which are orthologuous to rat genes measured in all sequencing batches shown in Table 7.1.

Figure 7.5 shows the ORA results for P20-INAS-Dose 2, and Figure 7.6 those for P10-INAS-Dose 3. Similarly to differential expression analysis, the p-values associated with the enrichments should be adjusted for multiple hypothesis testing, for each gene function category –*e.g.*, using Benjamini-Hochberg's correction (Benjamini and Hochberg, 1995). According to Figure 7.5, intranasal injection of a moderate dose of MSCs at a later stage (P20-INAS-Dose 2), as well as the intranasal injection of a high dose at postnatal day 10 (P10-INAS-Dose 3) perturb the expression of genes involved in the activation of granulocytes and neutrophils, which are cellular subtypes which play an important role in the immune system, and in the response to inflammation. Moreover, both treatments significantly down-regulate the expression of genes involved in phagocytosis and autophagia. These processes are activated in microglia exposed to the IL1β-induced inflammation, and are partly responsible for the deficit in myelin.

[4] That is, we match rat genes with human genes which have a close common genetic ancestor in the phylogenetic tree. As a general rule, orthologuous genes have the same function across species, which is the basis for running the analysis on human genes.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

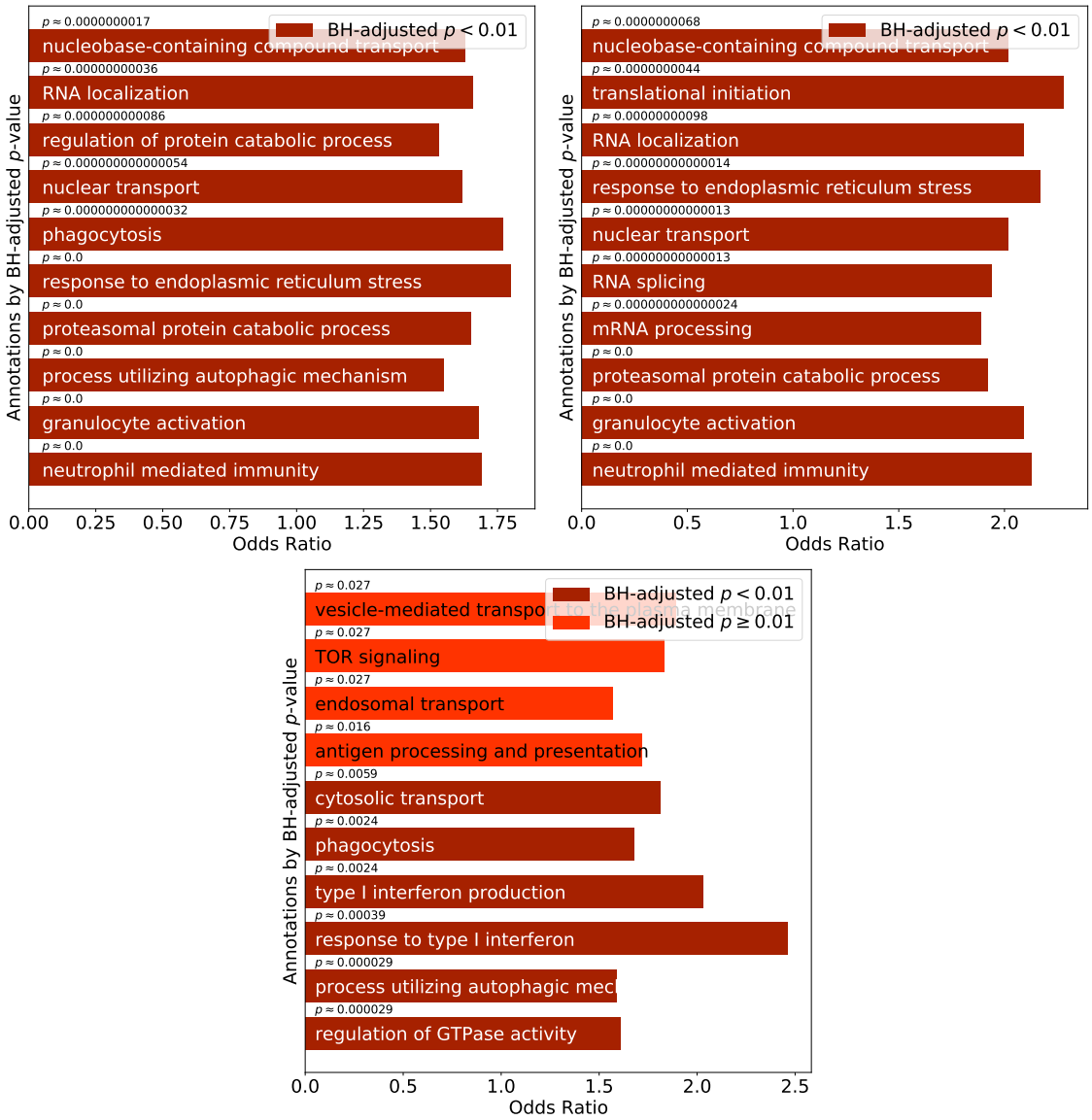

Figure 7.5: **Enrichment analysis of differentially expressed genes in the best recommended treatment.** *Top left plot* : enrichments (Top-10 in terms of decreasing p-value) of the subset of genes differentially expressed (both up- and down-regulated genes) by treatment protocol P20-INAS-Dose 2. *Top right plot* : considering the subset of up-regulated genes. *Bottom plot* : considering the subset of down-regulated genes.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

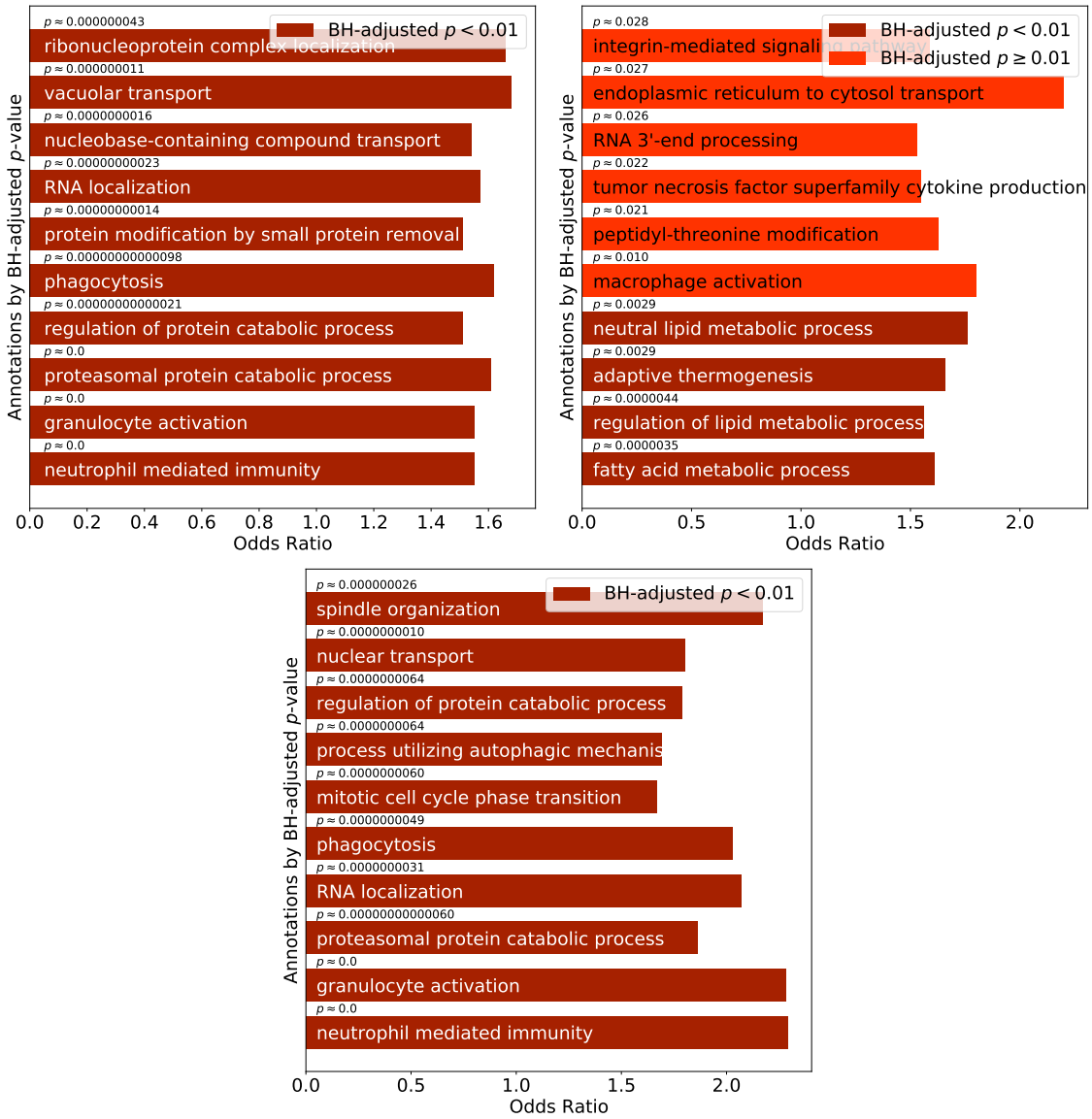

Figure 7.6: **Enrichment analysis of differentially expressed genes in the second best recommended treatment.** *Top left plot* : enrichments (Top-10 in terms of decreasing p-value) of the subset of genes differentially expressed (both up- and down-regulated genes) by treatment protocol P10-INAS-Dose 3. *Top right plot* : considering the subset of up-regulated genes. *Bottom plot* : considering the subset of down-regulated genes.

Moreover, a similar enrichment analysis was performed on the core set of 9 genes (excluding *AABR07006310.1*, having no human orthologuous gene) which are significantly differentially expressed in all signatures. The following terms are significantly enriched at level 5% with an adjusted p-value of 0.021 : "regulation of cell morphogenesis", "neutrophil mediated immunity", "granulocyte activation" and "extracellular structure organization". The four genes that contribute to the enrichments in neutrophil-mediated immunity and granulocyte activation –which are important terms with regards to EoP, as discussed above– are *CST3*, *CTSB*, *HEXB* and *TTR*. The fact that this core set of genes comprises of genes involved in the response to inflammation is quite conforting with respect to the relevance of our ranking.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

7.5 Discussion

Other experimental analyses, through the detection of the presence of myelin basic protein (MBP) via Western blots, showed a restoration of the levels of the MBP in stem cell-treated samples. It means that what we observe at transcriptomic level –positive scores of signature reversion– seems to match what is measured at the level of proteomics. This project showcases signature reversion, which is the key to ranking treatments. Signature reversion crucially relies on the definition of appropriate reference groups, in order to accurately quantify the transcriptional changes due to the treatment. This approach is agnostic to the disease, which enables its use in a generic fashion for other drug research investigations. Furthermore, this application highlights the potential of transcriptomics for drug repurposing. Even if transcriptomics might not be as informative as proteomics –*i.e.*, the direct study of protein production– analysis of transcriptomic data still provides important clues about groups of genes affected by a disease. This confirms our choice of focusing on transcriptomic data throughout the thesis.

Lastly, this project allows us to take a step back to contemplate the big picture of this thesis. Contrary to the drug repurposing method proposed in my PhD, the main focus of this project was the design of a score associated with drug signatures, which already matched the appropriate cell lines and diseased tissues. The main difficulty of the drug repurposing method, as mentioned in introduction, lies in the design of appropriate drug signatures when the only available data is a set of *in vitro* treated profiles in immortalized cell lines –most likely cancerous, in the case of the LINCS L1000 database (Subramanian et al., 2017). This issue was tackled in Chapter 3, which combines the Boolean network in Chapter 2 with carefully built drug-induced perturbation signatures.

One might wonder why we did not apply the cosine score directly on the "stabilized" binary profiles predicted by the Boolean network and signature CD[Healthy||Patients] built on patient and control profiles, as done in the project on EoP. This scoring method was actually shown to be less predictive than the current score function on a smaller subset of 10 molecules (4 antiepileptic and 6 proconvulsant drugs), where the proconvulsant or antiepileptic effect is provably assessed, except for Withaferin-A. The resulting rankings and receiver operating characteristic (ROC) curves for both methods are respectively shown in Figures 7.7 and 7.8. [5] This can actually be explained by the fact that the profiles predicted by the Boolean network

[5]Note that we replaced zeroes by -1's in the binary profiles, in order to appropriately reflect down-regulation when computing the cosine score.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

are not exactly drug signatures *per se* : they are *in silico* treated patient profiles. As such, they do not represent changes in expression, but predicted expression levels in treated profiles. This is why the repurposing score described in Algorithm 3 aims at determining in which part of the 2D plane the *in silico* treated profile is –either on the hyperplane globally assigned to control samples, or the one mainly associated with patient profiles– and then computes the distance of that point to the frontier separating the two sides of the plane. For the sake of completeness, we also report the ranking and associated ROC curve when the cosine score method is applied to the whole set of 34 drugs mentioned in Chapter 3 in Figure 11.1 in Appendix.

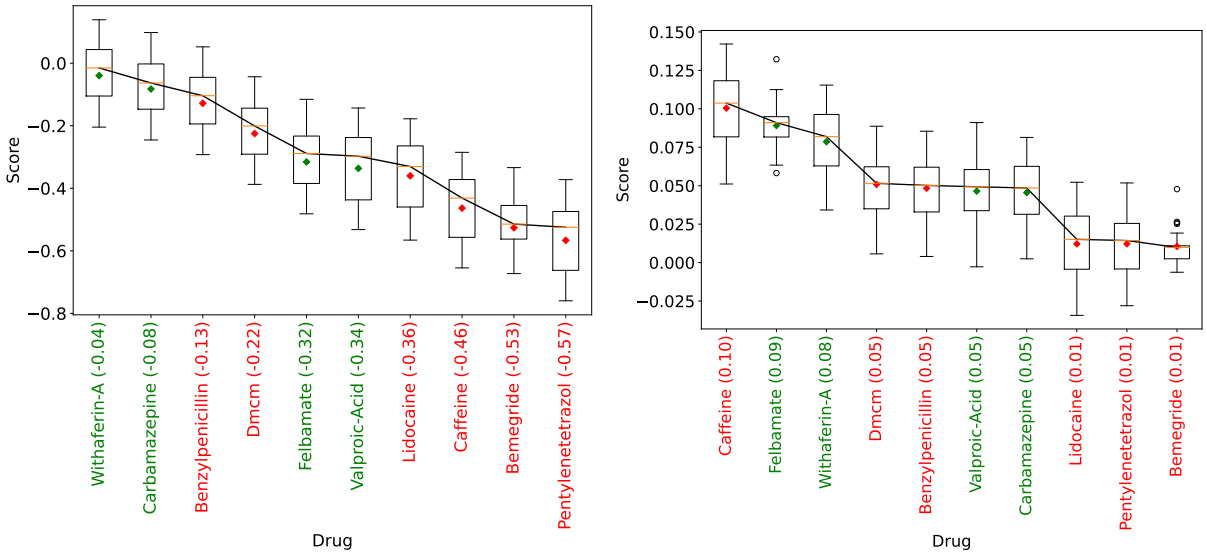

Figure 7.7: **Rankings from cosine scores and repurposing scores on a subset of drugs.** Boxplots of scores (*left* : repurposing score described in Algorithm 3 ; *right* : cosine score on "stabilized" profiles) on the smaller subset of 10 drugs, sorted by decreasing average score.

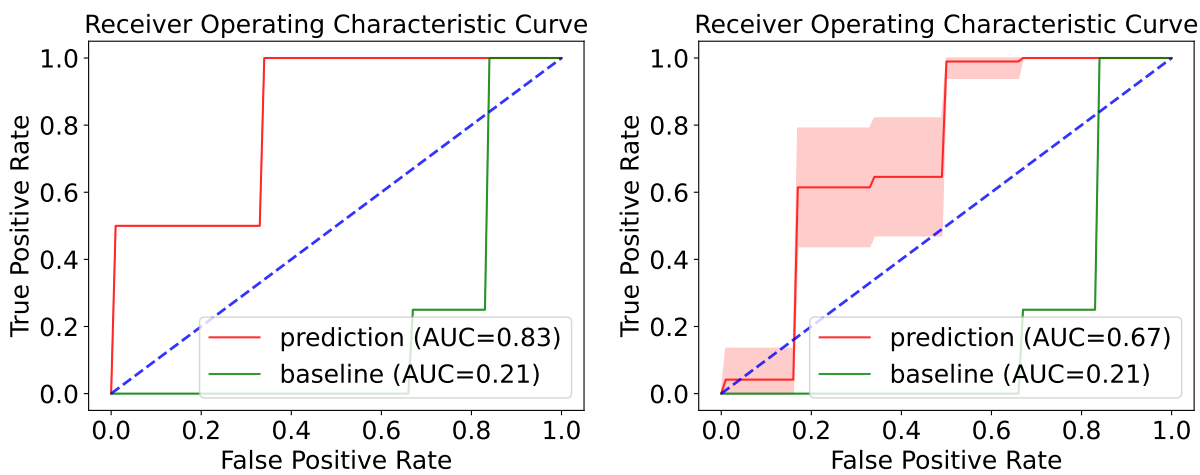

Figure 7.8: **Performance of cosine scores and repurposing scores on a subset of drugs.** ROC curves for the repurposing score (Algorithm 3, *left*) and the cosine score applied on "stabilized" profiles (*right*) on the smaller subset of 10 drugs. The baseline is L1000 CDS^2 (Duan et al., 2016).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 8

Extension to a collaborative setting

In the last part of my PhD, I was interested in extending to a multi-agent setting the principles laid out for a single agent facing a problem of identification. That setting might be of interest in *personalized* drug repurposing, in which potentially different sets of drug candidates are recommended to different subpopulations of patients. Heterogeneous patient subpopulations are a common sight for versatile diseases which actually regroup a large number of pathologies : *e.g.*, in breast cancer (Sims et al., 2007) or in epilepsy (Mirza, Stevelink, et al., 2021; Walker, Mirza, et al., 2015). In these diseases, a few number of patients might be afflicted with a specific subtype of disease, hence the idea of exploiting drug responses from other related patient subpopulations, while looking for the subpopulation-specific best drug candidates.

In the single-agent setting, the observations from a given action on the environment were assumed to be relatively uniform, as observations are "true" average scores with some additive noise of fixed variance equal to σ^2. Now, in order to model several subpopulations, a set of M agents is considered, across which expected rewards from the same arm might differ. Taking into account the observations made by other agents might decrease the total number of samples needed to determine the best drug candidates for each agent –especially as the reliance on others subpopulations grows– contrary to the case where each agent works separately.

However, in practice, broadcasting every new information about the observations to every agent might also be costly : indeed, launching a batch of trials is easier than having to wait for all other subpopulations to determine the next arm allocation. Moreover, with respect to patient data privacy –particularly in the context of adaptive clinical trials– sharing the raw patient

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

drug response might also be harmful.

Therefore, inspired by the "federated" [1] bandit learning setting which was proposed for regret minimization (Shi, Shen, and Yang, 2021), a generalized framework for collaborative bandit Top-N identification is introduced. An associated novel information theoretic lower bound on the sample complexity was derived. Furthermore, a phased elimination-based algorithm, which is nearly optimal with respect to the proposed lower bound, [2] was designed. This algorithm features two novel key ideas

- First, the tracking of an oracle allocation through a relaxed optimization problem, which is related to the characteristic time involved in the lower bound.

- Second, a data-dependent sampling scheme that selects the arms to sample in a more adaptive way than in prior works.

The combination of these two elements defines a new computationally cheaper approach to design near-optimal algorithms. This work has been accepted at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022) (Réda, Vakili, and Kaufmann, 2022), and was done in collaboration with Sattar Vakili (MediaTek Research). [3]

8.1 Collaborative Top-N (identification)

In this chapter, we do not consider specific model structures, and we consider exact Top-N identification ($\varepsilon = 0$). This work introduces a general multi-agent bandit model in which each agent is facing a finite set of K arms, and may communicate with the other $M-1$ agents through a central controller. The objective for an agent is to identify its own N best arms, in a context where optimality is defined with respect to *mixed* rewards. The mixed reward of an arm is a weighted sum of the rewards of this arm across agents, where the weight assigned by one agent to rewards observed by another is known. If positive (nonzero) weights are set on other agents' rewards, then communication between agents becomes necessary. This general setting recovers and extends several recent models for (centralized) collaborative bandit learning.

Let us now discuss the differences between this framework and the single-

[1] Their setting does not satisfy some requirements in federated learning, *e.g.*, dealing with communication interruption.
[2] That is, up to logarithmic multiplicative factors.
[3] Related code is located at `https://github.com/clreda/near-optimal-federated`.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

agent one discussed in Chapters 4 and 5. At each round, an agent can either sample an arm, or remain idle. In our case –dealing with a pure exploration problem– remaining idle means not sampling any arm, or, equivalently, sampling a "ghost" arm denoted 0, which returns observations equal to the smallest possible reward, so that it is quickly discarded from the set of top arms.

When agent $m \in [M]$ samples an arm $I_t^m \in [K]$ at round t based on prior observations, m observes *local* reward Y'^m_t, which only she can observe. This local reward is drawn from a σ^2-subgaussian distribution of mean $\mu_{(k,m)}$, independently from past observations and from other agents' observations –just like what happens in the single-agent context described in Definition 4.1.1.

However, this agent does not seek to identify the set of N arms $\arg\max_{k\in[K]}^{N} \mu_{(k,m)}$ that maximizes her *local* expected reward ; but rather, the set that maximizes a mixed reward, which definition relies on a known weight matrix $W := (w_{n,m})_{n,m\in[M]} \in [0,1]^{M\times M}$. The mth column of matrix W quantifies the normalized contributions of each agent to the mixed reward of agent m, such that for any agent $m \in [M]$

$$\sum_{n\in[M]} w_{n,m} = 1 \ .$$

In practice, this weight matrix could be obtained by considering a column-stochastic similarity matrix between patient subpopulations, based on their biomarkers, or transcriptional profiles. Then, as previously alluded to, the mixed reward at round $t > 0$ for agent $m \in [M]$ and arm $I_t^m \in [K]$ is defined as a weighted sum of the local rewards across agents, had they sampled the same arm I_t^m at round t

$$Y'^m_t := \sum_{n\in[M]} w_{n,m} Y'^n_t \ .$$

The expectation of this (unobserved) mixed reward, called *expected mixed reward*, is

$$\mu'_{(k,m)} := \sum_{n\in[M]} w_{n,m}\mu_{(k,n)} \ .$$

We denote by $\mathcal{S}^m_{N,\mu} := \arg\max_{k\in[K]}^{N} \mu'_{(k,m)}$ the set of N arms with largest expected mixed rewards for agent m, which is assumed unique. When it is obvious, we refer to this set as $\mathcal{S}^m_N$. Besides the degenerated case in which $w_{n,m} = \mathbb{1}(n = m)$, that is, each agent can solve their own bandit problem in isolation, agents need to communicate to fulfill their own identification targets ; *i.e.*, to share information about their local rewards to other agents.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

The communication model for an agent $m \in [M]$ is defined as follows : at each round t, this agent either remains idle ; or samples arm I_t^m and observes Y'^m_t ; or communicates information to a central server –*e.g.*, the empirical means of its past local observations. This central server broadcasts this information to all other agents, at the price of some fixed communication cost. Similarly to the communication model described in Shi, Shen, and Yang (2021); Tao, Zhang, and Zhou (2019), communication between agents, *i.e.*, broadcasting from the server, only happens at the end of local sampling phases for all agents ; that is, when all agents are idle at the same time. Like in single-agent settings, the analysis of the strategy adopted by the M agents will rely on the definition of *mixed* gaps related to pairs (agent, arm)

Definition 8.1.1. ***Characteristic gap for collaborative bandits.*** *For agent $m \in [M]$ and arm $k \in [K]$, the gap is defined as follows*

$$\Delta'_{k,m} := \max\left(\mu'_{(k,m)} - \max_{a\in[K]}^{N+1} \mu'_{(a,m)}, \max_{a\in[K]}^{N} \mu'_{(a,m)} - \mu'_{(k,m)}\right) .$$

In particular, it implies that

$$\Delta'_{k,m} := \begin{cases} \max_{a\in[K]}^{N} \mu'_{(a,m)} - \mu_k & \text{if } k \notin \mathcal{S}_N^m , \\ \mu'_{(k,m)} - \max_{a\in[K]}^{N+1} \mu'_{(a,m)} & \text{otherwise} . \end{cases}$$

The analysis will also depend on arm-pairwise gaps related to an agent, that is, for any pair of arms (k, ℓ) and agent m

$$\Delta'^m_{k,\ell} := \mu'_{(k,m)} - \mu'_{(\ell,m)} .$$

We denote the matrix in $\mathbb{R}^{K^2\times M}$, $\Delta' := (\Delta'^m_{k,\ell})_{k,\ell,m}$.

Our goal is to construct a δ-correct algorithm $\mathfrak{A}$, *i.e.*, a set of sampling, stopping and decision rules, so that, for any model of local expected rewards $\mu \in \mathbb{R}^{K\times M}$ and weight matrix W, the algorithm returns at stopping time $\tau_{\mathfrak{A}}$ M sets of N arms $(\widehat{\mathcal{S}}_N^1(\tau_{\mathfrak{A}}), \ldots, \widehat{\mathcal{S}}_N^M(\tau_{\mathfrak{A}}))$. These sets satisfy the property of correctness

$$\mathbb{P}_{\{\mu,W\}}\left(\forall m \in [M], \widehat{\mathcal{S}}_N^m(\tau_{\mathfrak{A}}) \subseteq \mathcal{S}_N^m\right) \geq 1 - \delta ,$$

while : first, achieving a small exploration cost

$$\mathsf{Exp}_{\{\mu,W\}}(\mathfrak{A}) := \sum_{m\in[M]} \sum_{k\in[K]} N_{(k,m)}(\tau_{\mathfrak{A}}) ,$$

where $N_{(k,m)}(t) := \sum_{s\leq t} \mathbb{1}(Y'^m_s = k)$ is the number of selections of arm k by agent m up to round t included ; and, second, aiming at a small communi-

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

cation cost, defined as

$$\mathrm{Com}_{\{\mu,W\}}(\mathfrak{A}) := \sum_{t \leq \tau_{\mathfrak{A}}} \mathbb{1}(\mathcal{I}_t) \ ,$$

where $\mathcal{I}_t$ is the event that the central server broadcasts information to all agents at round t.

In the algorithm we propose in this chapter, $\mathrm{Com}_{\{\mu,W\}}(\mathfrak{A})$ will be equal to the number of sampling phases, where agents observe local rewards from selected arms before sending information to the server. Similarly to the single-agent setting, a bound on the exploration cost can be expressed either in high probability (Equation (4.2)) or in expectation (Equation (4.3)).

8.2 Related work

The weighted collaborative model proposed above in the context of Top-N identification –and of pure exploration as a general rule– is novel. However, it encompasses different recent frameworks in collaborative learning, which formalized the case where several multi-armed bandits collaborate to efficiently perform sequential decision making (Zhu, Zhu, et al., 2021).

In particular, Shi, Shen, and Yang (2021) study a special case in which, given some personalization coefficient $\alpha \in [0,1]$, the mixed reward for arm $k \in [K]$ and agent $m \in [M]$ is an interpolation between the local expected reward $\mu_{(k,m)}$ and the average reward across agents $\frac{1}{M}\sum_{n\in[M]}\mu_{(k,n)}$. This amounts to choosing $W = (1-\alpha)I_M + \frac{1-\alpha}{M}\mathbb{1}_{M\times M}$, where $\mathbb{1}_{M\times M}$ is the matrix of size $M \times M$ with all coefficients equal to 1. In that work, authors consider the objective of minimizing regret in as few communication rounds as possible. Another work aimed at regret minimization considers a similar weighted model (Wu, Wang, et al., 2016) in a different setting in which a central controller chooses in each round an arm for the unique agent (corresponding to a sub-population) that arrives. [4]

The counterpart pure exploration problem of fixed-confidence Top-N identification in a collaborative context is the focus of this chapter. The weighted collaborative framework described in Section 8.1 extends the well-studied fixed-confidence Top-N identification problem (Chen, Li, and Qiao, 2017; Even-Dar et al., 2006; Gabillon, Ghavamzadeh, and Lazaric, 2012; Kalyanakrishnan et al., 2012).

[4] This type of bandits is an instance of *contextual bandits*, which are outside the scope of this thesis.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Another related setting is the "single-model" collaborative pure exploration (Hillel et al., 2013; Tao, Zhang, and Zhou, 2019; Yang, Chen, et al., 2021; Zhu, Mulle, et al., 2021), where M agents face the *same* best arm identification problem. Variants of this problem have been investigated. For instance, Yang, Chen, et al. (2021) consider asynchronous agents, which can only sample at some times ; whereas in Zhu, Mulle, et al. (2021); Zhu, Zhu, et al. (2021), agents can only communicate information to some of the other agents. The goal in these papers is to reduce the sample complexity needed to solve a single pure exploration problem, at the cost of some communication rounds. Our model recovers the synchronous setting when considering $\mu_{(k,m)} = \mu_{(k,n)}$ for any arm $k \in [K]$ and agents $m, n \in [M]$, and by setting the weight matrix to $W = I_M$.

An interesting kernelized collaborative pure exploration problem was recently studied by Du et al. (2021). In that work, both agents and arms are described by feature vectors, and there is a known kernel encoding the similarity between the mean reward of each (agent,arm) pair. This independent work follows a similar approach as ours and also propose a near-optimal phased elimination algorithm inspired by a lower bound, although the models and related lower bounds are significantly different. Finally, Russac et al. (2021) considers a pure exploration task in which the expected reward of an arm is the weighted sum of its rewards across M distinct subpopulations. A parallel can be drawn between their setting and the weighted collaborative framework by setting each weight $w_{n,m}$ to some α_n independent from m, so that the best arm is common to all subpopulations. However, the proposed algorithms do not aim at a low communication cost across subpopulations. Moreover, in the setting described in Section 8.1, an agent can potentially consider *any* linear combination of the others' observations in its mixed rewards, that is, any degree of personalization across agents.

In bandits working in collaboration, the need for a small communication cost makes algorithms based on *phased eliminations* appealing. In this category of algorithms, agents maintain a set of a active arms that are candidates for being optimal. At the end of each sampling phase, arms are possibly eliminated from this set, until the stopping criterion is met. Adaptivity to the observed rewards –and, in our case, communication– is only needed between sampling phases, which are typically long. This type of structure has been used in various bandit settings, both for regret minimization (Auer and Ortner, 2010; Shi, Shen, and Yang, 2021) and pure exploration (Fiez et al., 2019; Hassidim, Kupfer, and Singer, 2020; Hillel et al., 2013). In some of these algorithms, including Shi, Shen, and Yang (2021), the number of samples gathered from an arm which is active in some phase r is fixed in

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

advance. Going beyond such a deterministic sampling scheme might be crucial to achieve optimal performance with phased algorithms. In order to achieve (near)-optimality, others phased algorithms rely on the computation of an oracle allocation from the optimization problem related to a lower bound on sample complexity (Du et al., 2021; Fiez et al., 2019; Garivier and Kaufmann, 2016; Russac et al., 2021). In Du et al. (2021); Fiez et al. (2019), based on this allocation, a total number of samples to get in the current round is computed, depending on the identity of the surviving arms, that is, arms that remain in the candidate set. The distribution of samples across arms for the current round is proportional to the oracle allocation, and is obtained through a rounding procedure. Compared to these works, the algorithm proposed in this chapter is built on an allocation inspired by a *relaxation* of a lower bound, which does not only depend on the identity of surviving arms, and an alternative to the rounding procedure.

In our setting, we do not put constraints on the type of information that is exchanged in each communication round –which can be interesting when we consider privacy issues (Dubey and Pentland, 2020; Zhu, Zhu, et al., 2021)– nor on the lengths of the messages. Each communication round has a unit cost. In a communication round, all agents send messages to the central server –*e.g.*, estimates of their local means– and the server can send back arbitrary quantities or instructions –*e.g.*, how many times each arm should be sampled in the next exploration phase, and when to communicate next.

Moreover, contrary to the works of Hillel et al. (2013); Tao, Zhang, and Zhou (2019) on collaborative learning, we do not look at strategies that *explicitly* minimize for the number of communication rounds. Instead, our approach consists in proving a lower bound on the smallest possible exploration cost of a δ-correct algorithm which would communicate at every round ; and then, finding an algorithm which exploration cost matches this lower bound, while suffering a reasonable communication cost.

8.3 Lower bound for collaborative Top-N

We first present a lower bound on the exploration cost, *i.e.*, the total number of samples across arms and agents, needed for any δ-correct algorithm $\mathfrak{A}$ to make a decision $(\widehat{\mathcal{S}}_N^m(\tau_{\mathfrak{A}}))_{m\in[M]}$. This lower bound holds on the exploration cost of a collaborative exact Top-N identification algorithm, in which all agents communicate to the central server their latest observation as soon as they received it. Moreover, it assumes that we consider unstructured Gaussian bandits with fixed variance σ^2 ; meaning that the reward Y'^m_t from arm $I^m_t \in$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

$[K]$ observed at round $t > 0$ by agent $m \in [M]$ will be drawn from $\mathcal{N}(\mu_{(I_t^m,m)}, \sigma^2)$

$$Y'^m_t \sim \mu_{(I_t^m,m)} + \psi_t^m \text{ where } \psi_t^m \sim \mathcal{N}(0, \sigma^2) \ .$$

Similarly to the lower bound in single-agent settings (Result 4.3.3), this lower bound will be expressed as the value [5] of an optimization problem involving the characteristic gaps, inflated by a factor of order $\mathcal{O}(\delta^{-1})$.

Definition 8.3.1. ***Oracle problem.*** *Optimization problem $\mathcal{P}^\star$ on any vector $\Delta \in \mathbb{R}^{K\times K\times M}$ is defined as follows*

$$\mathcal{P}^\star(\Delta) := \underset{T\in(\mathbb{R}^+)^{K\times M}}{\text{arg min}} \ \sum_{m,k} T_{k,m} \qquad (1)$$

$$\text{s.t. } \forall m \in [M] \ \forall k \in \mathcal{S}_N^m \ \forall \ell \notin \mathcal{S}_N^m, \ \sum_{n\in[M]} w_{n,m}^2 \left(\frac{1}{T_{k,n}} + \frac{1}{T_{\ell,n}}\right) \leq \frac{(\Delta[k,\ell,m])^2}{2} \ . \qquad (2)$$

If allocation $T \in \mathbb{R}^{K\times M}$ satisfies $T \in \mathcal{P}^\star(\Delta)$, it means that T is a minimizer of Problem $\mathcal{P}^\star$, which minimizes the objective (1) while satisfying the constraints (2).

Before stating the lower bound, we further assume that the weight matrix W satisfies $w_{n,m} \neq 0$ for any agent $m \in [M]$. This assumption means that the mixed reward of any agent –at least partially– depends on her own local reward.

Theorem 8.3.2. ***Lower bound on the exploration cost for collaborative Top-N identification.*** *Let μ be a fixed matrix of expected rewards in $\mathbb{R}^{K\times M}$. For any $\delta \in (0, 1/2]$, let $\mathfrak{A}$ be a δ-correct algorithm under which each agent communicates each reward to the central server after it is observed. Then the expected exploration cost of algorithm $\mathfrak{A}$ satisfies*

$$\mathbb{E}[\mathit{Exp}_\mu(\mathfrak{A})] \geq \mathcal{N}^\star(\mu) \log\left(\frac{1}{2.4\delta}\right) \ ,$$

where $\mathcal{N}^\star(\mu) := \sum_{m\in[M]} \sum_{k\in[K]} \tau^\star_{k,m}$ and $\tau^\star \in \mathcal{P}^\star(\Delta')$, and $\Delta' := (\Delta'^m_{k,\ell})_{k,\ell,m}$.

The full proof (displayed below) uses standard change-of-distribution arguments, together with classical results from constrained optimization. Since the setting only deals with unstructured models, we drop all notations related to model functions, and directly consider bandit models instead.

Proof. Let us use Lemma 6.3.1 to define the set of alternative models to μ

[5] The value of a minimization problem is the objective evaluated at a minimizer (a similar definition can be given for maximization problems).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

in Top-N identification

$$\begin{aligned}\mathsf{Alt}(\mu) \quad &:= \quad \left\{\lambda \in \mathbb{R}^{K\times M} \mid \exists m \in [M],\ \mathcal{S}_N^m(\mu) \not\subseteq \mathcal{S}_N^m(\lambda)\right\} \\ &= \quad \left\{\lambda \in \mathbb{R}^{K\times M} \mid \exists m \in [M]\ \exists k \in \mathcal{S}_N^m(\mu)\ \exists \ell \notin \mathcal{S}_N^m(\mu),\ \lambda'_{(\ell,m)} > \mu'_{(k,m)}\right\} .\end{aligned}$$

where $\lambda'_{(k,m)} := \sum_{n\in[M]} w_{n,m}\lambda_{(k,n)}$ for any arm k and agent m. If, for δ-correct algorithm $\mathfrak{A}$, we assume that stopping time $\tau_{\mathfrak{A}}$ is almost surely finite under model μ, then let event $\mathcal{E}_\mu := \{\exists m \in [M],\ \widehat{\mathcal{S}}_N^m(\tau) \not\subseteq \mathcal{S}_N^m\}$. Since $\mathfrak{A}$ is δ-correct, this event holds with probability at most δ under model μ and at least $1-\delta$ under any alternative model to μ. We define $\tau_{k,m} := \mathbb{E}_{\{\mu\}}[N_{(k,m)}(\tau_{\mathfrak{A}})]$, where $N_{(k,m)}(t)$ is the number of samples arm k is sampled by agent m up to round t included, and $\mathsf{kl} : (x,y) \to x\log(x/y)+(1-x)\log((1-x)/(1-y))$ the binary relative entropy function. Since $\mathfrak{A}$ is δ-correct with $\delta \le 1/2$, by Kaufmann, Cappé, and Garivier (2016, Lemma 1) and the expression of Kullback-Leibler divergence for Gaussian distributions with fixed variance σ^2 in Equation (4.4), for any alternative model λ to μ

$$\frac{1}{2\sigma^2}\sum_{m\in[M]}\sum_{k\in[K]} \tau_{k,m}(\mu_{(k,m)} - \lambda_{(k,m)})^2 \ge \mathsf{kl}(\delta, 1-\delta) . \tag{8.1}$$

Since all diagonal coefficients of W are positive, for any $k \in [K]$, $m \in [M]$, $\tau_{k,m} > 0$. Indeed, if $w_{m,m} \neq 0$, it is possible to pick λ that only differs from μ by the entry $\lambda_{(k,m)}$, in such a way that arm k becomes optimal –or sub-optimal– for agent m

$$\begin{cases} \lambda_{(q,n)} &= \mu_{(q,n)} \text{ if } q \neq k \text{ or } n \neq m , \\ \lambda_{(k,m)} &= w_{m,m}^{-1}\left(\mu_{(k,m)} + \Delta'_{k,m}\right) \text{ if } k \notin \mathcal{S}_N^m(\mu) , \\ \lambda_{(k,m)} &= w_{m,m}^{-1}\left(\mu_{(k,m)} - \Delta'_{k,m}\right) \text{ otherwise } . \end{cases}$$

From Equation (8.1) and $\delta \in (0,1)$, we get

$$\tau_{k,m}\underbrace{(\mu_{(k,m)} - \lambda_{(k,m)})^2}_{>0} \ge \mathsf{kl}(\delta, 1-\delta) > 0 \implies \tau_{k,m} > 0 .$$

Consider now a fixed agent $m \in [M]$, and two arms $k \in \mathcal{S}_N^m(\mu)$ and $\ell \notin \mathcal{S}_N^m(\mu)$. We will build an alternative model λ, similar enough to μ, where only arms k and ℓ are modified for all agents, so that $k \notin \mathcal{S}_N^m(\lambda)$ and $\ell \in \mathcal{S}_N^m(\lambda)$. Given two nonnegative sequences $\gamma, \gamma' \in (\mathbb{R}^+)^M$, we define λ such that for all $n \in [M]$

$$\begin{cases} \lambda_{(q,n)} &= \mu_{(q,n)} \text{ if } q \notin \{k,\ell\} , \\ \lambda_{(k,n)} &= \mu_{(k,n)} - \gamma_n , \\ \lambda_{(\ell,n)} &= \mu_{(\ell,n)} + \gamma'_n , \end{cases}$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

and λ is an alternative model to μ if and only if

$$(\lambda'_{(\ell,m)} - \mu'_{(\ell,m)}) - (\lambda'_{(k,m)} - \mu'_{(k,m)}) = \sum_{n\in[M]} w_{n,m}(\gamma'_n + \gamma_n) \geq \Delta'^{m}_{k,\ell} := \mu'_{(k,m)} - \mu'_{(\ell,m)} \ . \quad (8.2)$$

Then, finding λ such that Equation (8.1) is as tight as possible –that is, building the closest alternative model to μ– yields the following constrained optimization problem in γ, γ'

$$\inf_{\substack{\gamma,\gamma'\in(\mathbb{R}^+)^M \\ \text{Equation 8.2 holds}}} \left[\sum_{n\in[M]} \tau_{k,n} \frac{\gamma_n^2}{2\sigma^2} + \sum_{n\in[M]} \tau_{\ell,n} \frac{\gamma_n'^2}{2\sigma^2} \right] \ .$$

The infimum can be computed in closed form using constraint optimization. Introducing a Lagrange multiplier $\Gamma \in \mathbb{R}^+$ and the minimizer $(\gamma^\star, \gamma'^\star)$, from the KKT conditions (Karush, 1939; Kuhn and Tucker, 2014) and the fact that $(\tau_{q,n})_{q,n}$ are positive, we get that, for any agent n,

$$\begin{aligned}
\sigma^{-2}\tau_{k,n}\gamma_n^\star - \Gamma w_{n,m} = 0 \implies \gamma_n^\star &= \frac{\Gamma w_{n,m}}{\sigma^2 \tau_{k,n}} \\
\sigma^{-2}\tau_{\ell,n}\gamma_n^\star - \Gamma w_{n,m} = 0 \implies \gamma_n'^\star &= \frac{\Gamma w_{n,m}}{\sigma^2 \tau_{\ell,n}} \\
\Gamma \left(\Delta'^{m}_{k,\ell} - \sum_{n\in[M]} w_{n,m}(\gamma_n^\star + \gamma_n'^\star) \right) &= 0 \ .
\end{aligned}$$

Solution $\Gamma = 0$ is not acceptable, because otherwise $\gamma_n^\star = \gamma_n'^\star = 0$ for any agent $n \in [M]$, and then $\lambda \notin \text{Alt}(\mu)$. If $\Gamma \neq 0$, then

$$\Delta'^{m}_{k,\ell} = \frac{\Gamma}{\sigma^2} \sum_n w_{n,m}^2 \left(\tau_{k,n}^{-1} + \tau_{\ell,n}^{-1}\right) \implies \Gamma = \frac{\sigma^2 \Delta'^{m}_{k,\ell}}{\sum_n w_{n,m}^2 (\tau_{k,n}^{-1} + \tau_{\ell,n}^{-1})} \ ,$$

And finally, plugging this expression into $\gamma^\star$ and $\gamma'^\star$ yields the following minimizer

$$\forall n \in [M],\ \gamma_n^\star = \frac{\Delta'^{m}_{k,\ell} w_{n,m} \tau_{k,n}^{-1}}{\sum_{n'\in[M]} w_{n',m}^2 (\tau_{k,n'}^{-1} + \tau_{\ell,n'}^{-1})} \text{ and } \gamma_n'^\star = \frac{\Delta'^{m}_{k,\ell} w_{n,m} \tau_{\ell,n}^{-1}}{\sum_{n'\in[M]} w_{n',m}^2 (\tau_{k,n'}^{-1} + \tau_{\ell,n'}^{-1})} \ ,$$

and the conclusion follows by plugging these expressions to get the expression of the infimum. □

Note that this theorem only focuses on the exploration cost, without attempting to minimize the number of communication rounds. Indeed, the number of communication rounds for an algorithm in which the central server shares information after each new observation is equal to the exploration cost, and is thus suboptimal in this regard.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Note that, for the single-agent case ($M = 1$), Theorem 8.3.2 recover the complexity of best arm identification in a Gaussian bandit model (Garivier and Kaufmann, 2016).

8.4 Structure for nearly-optimal Top-N

In single-agent pure exploration tasks, lower bounds can guide the design of optimal algorithms, as they allow to recover an oracle allocation –*i.e.*, the arg min for $\tau \in (\mathbb{R}^+)^{K\times M}$ in Problem (8.3.1). Algorithms may try to achieve this allocation by using some tracking (Du et al., 2021; Garivier and Kaufmann, 2016; Russac et al., 2021). Yet, these approaches may be computationally expensive, as they solve the optimization problem featured in the lower bound in every phase.

In this section, a third approach to design near-optimal algorithms in collaborative settings is introduced. This approach leverages the knowledge of the lower bound within a phased elimination algorithm. This is crucial to maintain a small communication cost $\mathrm{Com}_\mu(\mathfrak{A})$, and also to reduce the computational complexity compared to a pure tracking approach. The algorithm derived from this approach relies on a *relaxed* complexity term $\widetilde{\mathcal{N}}(\mu)$. As we will see, this term is within constant factors of the true constant $\mathcal{N}^\star(\mu)$. This implies that the upper bound on the exploration cost incurred by algorithm $\mathfrak{A}$ will ultimately depend on $\mathcal{N}^\star(\mu)$.

Relaxation of the lower bound optimization problem

The main issue with Problem (8.3.1) is its dependence on the knowledge of the *true* set $\mathcal{S}_N^m$ for any agent $m \in [M]$. In other algorithms which aim at asymptotic optimality –for instance, MisLid for Top-N identification for misspecified bandits in Chapter 6– it requires using the empirical best arms as proxy, and controlling the gap between this empirical estimate and the true one.

A simpler approach is actually to consider the following problem $\widetilde{\mathcal{P}}$ instead, for any matrix $\Delta \in (\mathbb{R}^+)^{K\times M}$

$$\widetilde{\mathcal{P}}(\Delta) := \operatorname*{arg\,min}_{\tau\in(\mathbb{R}^+)^{K\times M}} \sum_{m,k} \tau_{k,m} \text{ s.t. } \forall m \in [M]\ \forall k \in [K],\ \sum_{n\in[M]} \frac{w_{n,m}^2}{\tau_{k,n}} \leq \frac{(\Delta_{k,m})^2}{2}\,. \tag{8.3}$$

In Top-N identification, the value of Problem $\widetilde{\mathcal{P}}$ on the characteristic gaps is

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

at least upper bound by the quantity $\mathcal{N}^\star(\mu)$ from the oracle problem $\mathcal{P}^\star$ in Theorem 8.3.2. This result gets even more exciting for $N=1$, as the value $\widetilde{\mathcal{N}}(\mu)$ of Problem $\widetilde{\mathcal{P}}(\Delta')$ is equal up to a multiplicative absolute constant to the characteristic time $\mathcal{N}^\star(\mu)$.

Lemma 8.4.1. ***Bounds on the value of relaxed Problem*** $\widetilde{\mathcal{P}}$***.*** *Consider*

$$\widetilde{\mathcal{N}}(\mu) := \sum_{k,m} \tilde{\tau}_{k,m} \text{ and } \mathcal{N}^\star(\mu) := \sum_{k,m} \tau^\star_{k,m} \ ,$$

where $\tilde{\tau} \in \widetilde{\mathcal{P}}\left(\Delta'\right)$ *and* $\tau^\star \in \mathcal{P}^\star\left(\Delta'\right)$*, then*

$$\textbf{\textit{(i).}}\ \widetilde{\mathcal{N}}(\mu) \leq \mathcal{N}^\star(\mu)\ .$$

Moreover, for $N=1$

$$\textbf{\textit{(ii).}}\ \widetilde{\mathcal{N}}(\mu) \leq \mathcal{N}^\star(\mu) \leq 2\widetilde{\mathcal{N}}(\mu)\ .$$

Proof. Since we consider a single model μ here, we drop the indexing by μ in all notations in this proof.

(i). The following set of constraints of the oracle problem

$$\text{Const}^\star_N := \left\{ t \in (\mathbb{R}^+)^{K\times M} \mid \forall m \in [M]\ \forall k \in \mathcal{S}^m_N\ \forall l \notin \mathcal{S}^m_N,\ \sum_n w^2_{n,m}\left(t^{-1}_{k,n} + t^{-1}_{l,n}\right) \leq \frac{\left(\Delta'^m_{k,l}\right)^2}{2} \right\}$$

is included in the set of constraints of the relaxed problem in Equation 8.3

$$\widetilde{\text{Const}}_N := \left\{ t \in (\mathbb{R}^+)^{K\times M} \mid \forall m \in [M]\ \forall k \in [K],\ \sum_n \frac{w^2_{n,m}}{t_{k,n}} \leq \frac{\left(\Delta'_{k,m}\right)^2}{2} \right\}\ .$$

Indeed, if $t \in \text{Const}^\star_N$, then for any $m \in [M]$, and any $l \notin \mathcal{S}^m_N$,

$$\forall k \in \mathcal{S}^m_N,\ \sum_n \left(0 + \frac{w^2_{n,m}}{t_{l,n}}\right) \leq \sum_n w^2_{n,m}\left(\frac{1}{t_{k,n}} + \frac{1}{t_{l,n}}\right) \leq \frac{\left(\Delta'^m_{k,l}\right)^2}{2}$$

$$\Longrightarrow \sum_n \frac{w^2_{n,m}}{t_{l,n}} \leq \min_{k\in\mathcal{S}^m_N} \frac{(\Delta'^m_{k,l})^2}{2} = \frac{\left(\max\limits_{q\in[K]}^{N} \mu_{(q,m)} - \mu'_{(l,m)}\right)^2}{2} = \frac{\left(\Delta'_{k,m}\right)^2}{2}\ ,$$

Symmetrically, for any agent m and $k \in \mathcal{S}^m_N$, one can check that $\sum_n w^2_{n,m} t^{-1}_{k,n} \leq \frac{1}{2}\left(\Delta'_{k,m}\right)^2$, hence $t \in \widetilde{\text{Const}}_N$. Then, by minimality –that is, since $\tilde{\tau} \in \widetilde{\text{Const}}_N$, $t = \tau^\star \in \widetilde{\text{Const}}_N \cap \text{Const}^\star_N$, and $\tilde{\tau} \in \widetilde{\mathcal{P}}(\Delta')$– we conclude that $\widetilde{\mathcal{N}} \leq \mathcal{N}^\star$.

(ii). The lower bound is provided by **(i)** applied to the case $N=1$. Now let us consider the solutions $\tau^\star \in \mathcal{P}^\star(\Delta')$ and $\tilde{\tau} \in \widetilde{\mathcal{P}}(\Delta')$. Let us denote $k^m_\star := \arg\max\limits_{k\in[K]} \mu'_{(k,m)}$ the (unique) optimal arm for mixed expected rewards

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

for agent m. Then, for any agent $m \in [M]$ and any arm $k \neq k_\star^m$

$$\sum_{n\in[M]} \left(\frac{w_{n,m}^2}{2\tilde{\tau}_{k,n}} + \frac{w_{n,m}^2}{2\tilde{\tau}_{k_\star^m,n}} \right) = \frac{1}{2} \underbrace{\left(\sum_{n\in[M]} \frac{w_{n,m}^2}{\tilde{\tau}_{k,n}} \right)}_{\leq \left(\Delta'_{k,m}\right)^2/2} + \frac{1}{2} \underbrace{\left(\sum_{n\in[M]} \frac{w_{n,m}^2}{\tilde{\tau}_{k_\star^m,n}} \right)}_{(*)} \leq \left(\Delta'_{k,m}\right)^2/2 \,.$$

where $(*)$ uses the fact that by Definition 8.1.1 for $N = 1$

$$\Delta'_{k_\star^m,m} := \mu'_{(k_\star^m,m)} - \max_{q\in[K]}{}^{2} \mu'_{(q,m)} = \min_{q\neq k_\star^m} \Delta'_{q,m} \leq \Delta'_{k,m} \,.$$

Then $2\tilde{\tau} \in \mathrm{Const}_1^\star$, therefore once again by minimality, $\mathcal{N}^\star \leq 2\widetilde{\mathcal{N}}$. □

Compared to Problem $\mathcal{P}^\star$, a nice feature of $\widetilde{\mathcal{P}}$ is that its constraint set does not depend on the knowledge of $(\mathcal{S}_N^m)_{m\in[M]}$, which will allow us to design algorithms that do not suffer too much from bad empirical guesses on the N-best arms, especially in early phases.

Remark 8.4.2. *We further note that solving $\widetilde{\mathcal{P}}$ is slightly easier than solving $\mathcal{P}^\star$. Indeed, this optimization problem can be decoupled across arms, and it is sufficient to compute, for any arm $k \in [K]$*

$$\operatorname*{arg\,min}_{\tau^k\in(\mathbb{R}^+)^M} \sum_{m\in[M]} \tau_m^k \text{ s.t. } \forall m \in [M], \sum_{n\in[M]} \frac{w_{n,m}^2}{\tau_n^k} \leq \frac{\left(\Delta'_{k,m}\right)^2}{2} \,,$$

and then consider as the oracle allocation for the problem on the whole set of arms $(\tau_{k,m})_{k,m} := (\tau_m^k)_{k,m}$. To solve this convex optimization problem, we may rely on solvers for disciplined convex optimization, such as those implemented in CVXPy (Agrawal, Verschueren, et al., 2018; Diamond and Boyd, 2016).

However, we cannot get access to the true gaps $(\Delta'_{k,m})_{k,m}$. Thus, our algorithm will rely on gap proxies $(\widetilde{\Delta'_{k,m}})_{k,m}$ which will depend on past samplings and observations. The following lemma will be crucial in the analysis of the algorithm, in order to compare values, under some condition, from $\widetilde{\mathcal{P}}(\Delta')$ and $\widetilde{\mathcal{P}}(\widetilde{\Delta})$, where $\widetilde{\Delta} := (\widetilde{\Delta'_{k,m}})_{k,m}$. Its proof is given in Appendix (Lemma 13.1.1).

Lemma 8.4.3. ***Comparison of values of Problem $\widetilde{\mathcal{P}}$ with different gaps.*** *Consider $\Delta, \Delta' \in (\mathbb{R}^+)^{K\times M}$, such that $\tau \in \widetilde{\mathcal{P}}(\Delta)$ and $\tau' \in \widetilde{\mathcal{P}}(\Delta')$. Moreover, assume that there is a positive constant β such that for any agent m and arm k, $\Delta'_{k,m} \leq \beta\Delta_{k,m}$. Then*

$$\frac{1}{\beta^2} \sum_{k,m} \tau_{k,m} \leq \sum_{k,m} \tau'_{k,m} \,.$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Collaborative Phased Elimination (CPE)

We now introduce an algorithm for collaborative Top-N identification, called CPE and stated as Algorithm 8.

CPE proceeds in phases, indexed by r. In phase r, we let $B_m(r)$ be the set of active arms for agent m –that is, the set of candidate arms that agent m keeps sampling at phase r– and $B(r) := \bigcup_{m\in[M]} B_m(r)$ be the set of arms that are active for at least one agent at phase r. The algorithm maintains proxy gaps $(\widetilde{\Delta'_{k,m}}(r))_{k,m}$ for the true gaps $(\Delta'_{k,m})_{k,m}$ that are halved at the end of each phase for arms that remain active, hence the dependence in past observations. At the beginning of each phase r, the oracle allocation $t(r)$, with respect to the proxy gaps, is computed, as well as the number of new samples $(d_{k,m}(r))_{k,m}$ that player m should get from arm k in phase r. For any arm k and agent m, $d_{k,m}(r)$ is defined such that the total number of selections $n_{k,m}(r)$ of arm k by agent m up to round r included becomes close to –a quantity slightly larger than– $t_{k,m}(r)\log(1/\delta)$.

We observe that any arm $k \notin B(r)$ will not get any new samples in phase r, as the associated proxy gaps $(\widetilde{\Delta'_{k,n}})_{n\in[M]}$ are identical to those in the previous phase $r-1$, hence $t_{k,n}(r) = t_{k,n}(r-1)$ and $d_{k,n}(r) = 0$ for any agent $n \in [M]$.

In contrast to prior works, where the allocation in each round only depends on the identity of the surviving arms and the round index (Du et al., 2021; Fiez et al., 2019), in CPE it also depends on when the arms have been eliminated (which condition which gaps are frozen). After each agent m samples arm k $d_{k,m}(r)$ times, they all send their empirical local expected rewards $\left(\widehat{\mu_{(k,m)}}(r)\right)_k$ to the central server, which computes the empirical mixed expected rewards $\left(\widehat{\mu'_{(k,m)}}(r)\right)_{k,m,r}$

$$\forall m \in [M]\ \forall k \in [K]\ \forall r \geq 0,\ \widehat{\mu'_{(k,m)}}(r) := \sum_{n\in[M]} w_{n,m}\widehat{\mu_{(k,m)}}(r)\ .$$

The active sets $(B_m(r))_{m\in[M]}$ of all agents are then updated by removing arms whose empirical mixed expected rewards are too small. The number of communication rounds in Algorithm 8 is then exactly equal to the number of phases needed until the stopping criterion in Line 29 is fulfilled.

As in several prior works (Kaufmann and Kalyanakrishnan, 2013; Shi, Shen, and Yang, 2021), we rely on confidence intervals to perform these eliminations. However, constructing confidence intervals on the mixed expected rewards under our adaptive sampling rule is more challenging than when the number of samples from an active arm in phase r is fixed in advance. We build an confidence interval of the form $\left\{\widehat{\mu_{(k,m)}}(r) \pm \Omega_{(k,m)}(r)\right\}$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Algorithm 8 CPE algorithm for the collaborative setting. Collaborative Phased Elimination (CPE)

Initialize $r \leftarrow 0$, $\forall k, m, \widetilde{\Delta'_{k,m}}(0) \leftarrow 1, n_{k,m}(0) \leftarrow 1$, $\forall m, B_m(0) \leftarrow [K]$
Draw each arm k by each agent m once
repeat

  # * **Central server side** *
  $B(r) \leftarrow \bigcup_{m\in[M]} B_m(r)$
  Compute $t(r) \leftarrow \widetilde{\mathcal{P}}\left(\left(\sqrt{2}\widetilde{\Delta'_{k,m}}(r)\right)_{k,m}\right)$
  For all $k \in [K]$, compute

$$(d_{k,m}(r))_{m\in[M]} \leftarrow \arg\min_{d\in\mathbb{N}^M} \sum_m d_m \text{ s.t. } \forall m \in [M],\ \frac{n_{k,m}(r-1)+d_m}{\mathcal{T}_\delta(n_{k,\cdot}(r-1)+d)} \geq t_{k,m}(r)$$

  Send to each agent $m \in [M]$ $(d_{k,m}(r))_{k\in[K]}$ and $d_{\max}(r) := \max_{n\in[M]} \sum_{k\in[K]} d_{k,n}(r)$

  # * **Agent** $m \in [M]$ **side** *
  Sample arm $k \in B(r)$ $d_{k,m}(r)$ times, so that $n_{k,m}(r) = n_{k,m}(r-1) + d_{k,m}(r)$
  Remain idle for $d_{\max}(r) - \sum_{k\in[K]} d_{k,m}(r)$ rounds
  Send to server empirical local mean $\widehat{\mu_{(k,m)}}(r)$ based on the $n_{k,m}(r)$ samples

  # * **Central server side** *
  Compute each empirical mixed mean $\widehat{\mu'_{(k,m)}}(r)$ based on the empirical local means
  # Update set of candidate best arms for each user
  for $m = 1$ **to** M **do**
    $B_m(r+1) \leftarrow \left\{k \in B_m(r) \mid \widehat{\mu'_{(k,m)}}(r) + \Omega_{(k,m)}(r) \geq \max_{j\in B_m(r)}^{N} \left(\widehat{\mu'_{(j,m)}}(r) - \Omega_{(j,m)}(r)\right)\right\}$
  end for
  # Update the gap estimates
  for $k = 1$ **to** K **do**
    for $m = 1$ **to** M **do**
      if $k \in B_m(r+1)$ **and** $|B_m(r+1)| > N$ **then**
        $\widetilde{\Delta'_{k,m}}(r+1) \leftarrow \widetilde{\Delta'_{k,m}}(r)/2$
      end if
    end for
  end for
  $r \leftarrow r + 1$

until $\forall m \in [M], |B_m(r)| \leq N$
Output: for any agent m, $\widehat{\mathcal{S}}_N^m(\tau_{\text{CPE}}) := B_m(r)$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

to control empirical mixed expected rewards. The width $\Omega_{(k,m)}(r)$ of the confidence interval scales with the following quantity

Definition 8.4.4. ***Confidence interval for CPE.*** *For any arm k, agent m, and round $r \geq 0$, we define*

$$\Omega_{(k,m)}(r) := \sqrt{\mathcal{T}_\delta(n_{k,\cdot}(r)) \sum_{n \in [M]} \frac{w_{n,m}^2}{n_{k,n}(r)}} \,,$$

where $\mathcal{T}_\delta(\cdot) : (\mathbb{R}^+)^M \to \mathbb{R}^+, N \mapsto \mathcal{T}_\delta(N)$ is a threshold function to be defined.

A choice of threshold that gives valid confidence intervals can be derived by leveraging some time-uniform concentration inequalities from Kaufmann and Koolen (2021, Proposition 24). The proof is postponed to the Appendix (Lemma 13.1.2).

Lemma 8.4.5. ***Choice of threshold function $\mathcal{T}_\delta(\cdot)$.*** *For any $N \in (\mathbb{N}^+)^M$*

$$\mathcal{T}_\delta(N) := 2\left(g_M\left(\frac{\delta}{KM}\right) + 2\sum_{m \in [M]} \ln\left(4 + \ln(N_m)\right)\right) \,,$$

where g_M is a function that satisfies $g_M(\delta) \approx \log(1/\delta) + M \log\log(1/\delta)$. Then the good event

$$\mathcal{E}^{\text{CPE}} := \left\{\forall r \geq 0 \ \forall m \in [M] \ \forall k \in [K], \ \left|\mu'_{(k,m)} - \widehat{\mu'_{(k,m)}}(r)\right| \leq \Omega_{(k,m)}(r)\right\}$$

holds with probability larger than $1 - \delta$.

From this lemma and the elimination criterion at Line 17 in Algorithm 8, it easily follows that CPE for Top-N identification is δ-correct for this choice of threshold function, as, for any agent $m \in [M]$, no good arm $k \in \mathcal{S}_N^m$ can ever be eliminated from $B_m(r)$ at any round r on event $\mathcal{E}^{\text{CPE}}$.

Theorem 8.4.6. ***CPE is δ-correct.*** *On event $\mathcal{E}^{\text{CPE}}$, CPE outputs the correct set of optimal arms $\mathcal{S}_N^m$ for each agent m.*

The fact that the sample complexity of CPE scales with $\widetilde{\mathcal{N}}$ (and $\mathcal{N}^\star$ as well, thanks to Lemma 8.4.1) stems from the interplay between the confidence interval width $\Omega_{(k,m)}(t)$ –which, up to the threshold function $\mathcal{T}_\delta(\cdot)$, is exactly one of the constraints featured in the lower bound (Theorem 8.3.2)– and the definition of the allocation $t(r)$. This leads to the following crucial result

Lemma 8.4.7. *On event $\mathcal{E}^{\text{CPE}}$, for any arm k, agent m, and round r*

$$\Omega_{(k,m)}(r) \leq \widetilde{\Delta'_{k,m}}(r) \,.$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Proof. For any round r and arm k, using first the definition of $d(r)$ at Line 7 and then the definition of oracle allocation $t(r)$ at Line 6 in Algorithm 8

$$\Omega_{(k,m)}(r) := \sqrt{\mathcal{T}_\delta(n_{k,\cdot}(r)) \sum_{n\in[M]} \frac{w_{n,m}^2}{n_{k,n}(r)}} = \sqrt{\sum_{n\in[M]} w_{n,m}^2 \frac{\mathcal{T}_\delta(n_{k,\cdot}(r-1)+d_{k,\cdot})}{n_{k,n}(r-1)+d_{k,n}(r)}}$$
$$\leq \begin{cases} \sqrt{\sum_{n\in[M]} \frac{w_{n,m}^2}{t_{k,n}(r)}} \leq \widetilde{\Delta'_{k,m}}(r) & \text{if } k \in B(r) \\ \sqrt{\sum_{n\in[M]} \frac{w_{n,m}^2}{t_{k,n}(r'_k)}} \leq \widetilde{\Delta'_{k,m}}(r'_k) = \widetilde{\Delta'_{k,m}}(r) & (*) \text{ otherwise} . \end{cases}$$

where $r'_k := \sup\{r' \geq 0 \mid k \in B(r')\}$ when $k \notin B(r)$ and $(*)$ uses the fact that $d_{k,m}(r) = 0$ when $k \notin B(r)$. □

All in all, the following upper bound in high probability on the exploration cost of CPE is proven

Theorem 8.4.8. ***Upper bound on Exp$_\mu$(CPE).*** *On any model μ, with probability $1-\delta$, CPE outputs the set of N best arms for each agent with an exploration cost at most*

$$32\mathcal{N}^\star(\mu) \log_2\left(\frac{8}{\Delta'_{\min}}\right) \log(1/\delta) + o(1/\delta) \text{ where } \Delta'_{\min} := \min_{m\in[M]} \min_{k\in[K]} \Delta'_{k,m} ,$$

and at most $\left\lceil \log_2\left(\frac{8}{\Delta'_{\min}}\right)\right\rceil$ communication rounds.

A sketch of the proof will be given in this section. The full proofs of Theorems 8.4.6 and 8.4.8 are available in Appendix (Theorems 13.2.1 and 13.3.1).

This theorem proves that CPE is matching the exploration lower bound of Theorem 8.3.2 in the asymptotic regime of small values of δ, up to a logarithmic term in $\mathcal{O}(1/\Delta'_{\min})$. It achieves this using only $\lceil \log_2(8/\Delta'_{\min})\rceil$ communication rounds. We note that a similar extra multiplicative logarithmic factor is present in the analysis of near-optimal phased algorithms in other contexts (Du et al., 2021; Fiez et al., 2019). Such a quantity appears as an upper bound on the number of phases, and may be a price to pay for the phased structure.

We argue that the communication cost of CPE is actually of the same order of magnitude as that featured in some related work. In Shi, Shen, and Yang (2021), which is the closest setting to our framework, the equivalent number of communication rounds p needed to solve the regret minimization problem is upper bounded by $\mathcal{O}\left(2\log_2\left(\frac{8}{\sqrt{M}\Delta'_{\min}}\right)\right)$. In the setting of collaborative learning, where M agents face the same set of arm distributions and $W = I_M$, Hillel et al. (2013, Theorem 4.1) prove that an improvement of multiplicative factor $1/M$ on the exploration cost for a traditional best arm identification

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

algorithm can be reached by using at most $\lceil \log_2(1/\Delta_{\min}) \rceil$ communication rounds, where $\Delta_{\min}$ is the gap between the best and second best arms.

Sketch of proof. We let R denote the random number of phases used by Algorithm 8 before stopping. On the good event $\mathcal{E}^{\text{CPE}}$, considering any agent $m \in [M]$, we have proven that the algorithm never eliminates any good arm $k \in \mathcal{S}_N^m$ (Theorem 8.4.6), hence $R := \max_{m \in [M]} \max_{k \notin \mathcal{S}_N^m} R_{k,m}$, where $R_{k,m}$ is the last phase r in which $k \in B_m(r)$. Using Lemma 8.4.7, we can easily establish that

$$R_{k,m} \leq r_{k,m} := \min \left\{ r \geq 0 \mid 4 \times 2^{-r} < \Delta'_{k,m} \right\} ,$$

which satisfies $r_{k,m} \leq \log_2 \left(8/\Delta'_{k,m} \right)$. This yields $R \leq \log_2 \left(8/\Delta'_{\min} \right)$, and further permits to prove that the proxy gaps $\left(\widetilde{\Delta'_{k,m}}(r) \right)_{k,m,r}$ can be lower bounded by the true gaps

$$\forall r \leq R \; \forall k \in [K] \; \forall m \in [M], \; \widetilde{\Delta'_{k,m}}(r) \geq 1/8 \Delta'_{k,m} .$$

Using the monotonicity properties of the oracle that are stated in Lemma 8.4.3, we establish that the allocation $t(r)$ computed from the proxy gaps at Line 7 in Algorithm 8 satisfies

$$\forall r \leq R, \; \sum_{m \in [M]} \sum_{k \in [K]} t_{k,m}(r) \leq 32 \widetilde{\mathcal{N}}(\mu) . \tag{8.4}$$

To upper bound the exploration cost $\text{Exp}_\mu(\text{CPE})$, the next step is to relate the number of selections of arm k by agent m up to phase R $n_{k,m}(R)$ to the oracle allocations. To do so, we observe that if $\widehat{R}_{k,m}$ is the last phase r before R such that the number of selections of arm k by agent m *in* phase r $d_{k,m}(r)$ is positive, then $n_{k,m}(R) = n_{k,m}(\widehat{R}_{k,m})$, and by definition of the quantities $(d_{k,m}(r))_{k,m,r}$ and $\mathcal{T}_\delta(\cdot)$

$$n_{k,m}(\widehat{R}_{k,m}) \leq t_{k,m}(\widehat{R}_{k,m}) \mathcal{T}_\delta(n_{k,\cdot}(\widehat{R}_{k,m})) + 1 \leq t_{k,m}(\widehat{R}_{k,m}) \mathcal{T}_\delta(n_{k,\cdot}(R)) + 1 . \tag{8.5}$$

Then $\tau_{\text{CPE}} := \sum_{m,k} n_{k,m}(R)$ is bounded from above by

$$\begin{aligned} \tau_{\text{CPE}} &\leq \sum_{m,k} t_{k,m}(\widehat{R}'_{k,m}) \mathcal{T}_\delta(n_{k,\cdot}(R)) + KM \\ &\leq \sum_{m,k} \sum_{r \leq R} t_{k,m}(r) \beta^\star(\tau_{\text{CPE}}) + KM \\ &\leq 32 R \widetilde{\mathcal{N}}(\mu) \beta^\star(\tau_{\text{CPE}}) + KM , \end{aligned}$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

where we use Equation 8.4 and introduce

$$\beta^\star(\tau_{\mathsf{CPE}}) := 2\left(g_M\left(\frac{\delta}{KM}\right) + 2M\ln(4+\ln(\tau_{\mathsf{CPE}}))\right) .$$

Then, using the upper bound on the total number of rounds R and Lemma 8.4.1

$$\tau_{\mathsf{CPE}} \leq 32\log\left(8/\Delta'_{\mathsf{min}}\right)\mathcal{N}^\star(\mu)\beta^\star(\tau_{\mathsf{CPE}}) + KM . \tag{8.6}$$

The end of the proof consists in using the known upper bound on R, and finding an upper bound for the largest $\tau_{\mathfrak{A}}$ satisfying Equality 8.6.

8.5 Application to drug repurposing

Our general framework for any weight matrix W and $\mu \in \mathbb{R}^{K\times M}$ has not been studied in pure exploration prior to this work. However, we can compare it in known settings, namely, in the personalized collaborative best arm identification, which is a straightforward counterpart to personalized federated regret minimization introduced in Shi, Shen, and Yang (2021). As we did throughout this chapter, we always consider Gaussian bandits with fixed variance $\sigma^2 = 1$.

Baseline algorithm for collaborative best arm identification (BAI). We state below as PF-UCB-BAI (Algorithm 9) a straightforward adaptation of the PF-UCB algorithm in Shi, Shen, and Yang (2021) to *personalized* collaborative best arm identification ; meaning that only weight matrices of the form $w_{n,m} = \alpha\mathbb{1}(n=m) + \frac{1-\alpha}{M}$ for any pair of agents $(n,m) \in [M]^2$ are considered.

The original regret algorithm uses phased eliminations designed for each agent to identify their best arm together with *exploitation* : when all best arms have been found, or when some agent is waiting for others to finish their own exploration rounds, agents keep playing their empirical best arm.

To turn this into a δ-correct algorithm, we remove the exploitation rounds ; keep the same sampling rule within each phase –in which the number of samples from each arm is proportional to some rate function $f(r)$; and calibrate the size of the confidence intervals used to perform eliminations slightly differently, introducing for any $\delta \in (0,1)$ function

$$\forall r \geq 0,\ B_r(\delta) := \sqrt{\frac{2\log\left(KM\zeta(\beta)r^\beta/\delta\right)}{MF(r)}} .$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

where $F(r) = \sum_{p=1}^{r} f(p)$, for some $\beta > 1$, and where ζ is the Riemann zeta function. In practice, we use $\beta = 2$. This algorithm follows the same general structure as our algorithm, with the notable difference that the number of samples of an arm $k \in B_m(r)$ in phase r is fixed in advance. PF-UCB-BAI is indeed δ-correct. On the good event $\mathcal{G}$ introduced in the following Lemma 8.5.1, the optimal arm for any agent m can never be eliminated from the set $B_m(r)$ at phase r.

Algorithm 9 PF-UCB-BAI. Benckmark algorithm for Collaborative Top-N

$f(r)$: sampling effort in phase r, $B_r(\delta)$: size of the confidence intervals in phase r.
Initialize $r \leftarrow 0$, $\forall k, m, n_{k,m}(0) \leftarrow 0$, $\forall m, B_m(0) \leftarrow [K]$, $\hat{k}_m \leftarrow 0$.
repeat

  # * **Central server side** *
  if $|B_m(r)| = 1$ **then**
    $\hat{k}_m \leftarrow$ the unique arm in $B_m(r)$
    $B_m(r) \leftarrow \emptyset$
  end if
  $B(r) \leftarrow \bigcup_{m \in [M]} B_m(r)$
  for $k \in [K], m \in [M]$ **do**
    if $k \in B(r)$ **then**
      $d_{k,m}(r) \leftarrow \lceil (1-\alpha) f(r) \rceil + \lceil \alpha M f(r) \rceil \mathbb{1}\left(k \in B_m(r)\right)$ **else** $d_{k,m}(r) \leftarrow 0$
    end if
  end for
  Send $(d_{k,m}(r))_{k \in [K]}$ to agent m

  # * **Agent** $m \in [M]$ **side** *
  Sample arm $k \in [K]$ by agent m $d_{k,m}(r)$ times, so that the total number of samples is $n_{k,m}(r) = n_{k,m}(r-1) + d_{k,m}(r)$
  Compute the empirical mixed means $(\hat{\mu}'_{k,m}(r))_{k,m}$ based on the $(n_{k,m}(r))_{k,m}$ samples and send them to the central server

  # * **Central server side** *
  // Update set of candidate best arms for each user
  for $m = 1$ **to** M **do**

$$B_m(r+1) \leftarrow \left\{ k \in B_m(r) \mid \hat{\mu}'_{k,m}(r) + B_r(\delta) \geq \max_{j \in B_m(r)} \left(\hat{\mu}'_{j,m}(r) - B_r(\delta) \right) \right\}$$

  end for
  $r \leftarrow r + 1$

until $|B(r)| = \emptyset$
Output: $\left\{ \hat{k}_m : m \in [M] \right\}$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Lemma 8.5.1. *Event*

$$\mathcal{G} := \left\{ \forall r \in \mathbb{N}^* \ \forall m \in [M] \ \forall k \in B_m(r), \ \left| \widehat{\mu'_{(k,m)}}(r) - \mu'_{(k,m)} \right| \leq B_r(\delta) \right\}$$

holds with probability $1 - \delta$.

This lemma is a simple adaptation of Shi, Shen, and Yang (2021, Lemma 1) combined with a union bound on $r \in \mathbb{N}^*$. Moreover, on event $\mathcal{G}$, similarly to the analysis of PF-UCB (Shi, Shen, and Yang, 2021), we can upper bound the number of rounds where arm k is sampled by agent m by

$$p_{k,m} := \inf\{r : B_r(\delta) \leq \Delta'_{k,m}/4\} \ .$$

When $f(p) = 2^p$, one can prove that

$$\sum_{p=1}^{p_{k,m}} f(p) = \mathcal{O}\left(M^{-1} \Delta'^{-2}_{k,m} \log(\delta^{-1})\right) \ .$$

Summing the (deterministic) global and local exploration cost over rounds, arms and agents, yields an exploration cost of order

$$\mathcal{O}\left(\sum_{k \in [K]} \left[\left(\frac{1-\alpha}{\min\limits_{n \in [M]} \left(\Delta'_{k,n}\right)^2} \right) + \left(\sum_{m \in [M]} \frac{\alpha}{\left(\Delta'_{k,m}\right)^2} \right) \right] \log\left(\frac{1}{\delta}\right) \right) \ .$$

This algorithm can therefore serve as a baseline to be compared to our proposal in this particular case.

Drug repurposing. As mentioned in Chapter 6, in fixed-confidence BAI as well, algorithms using symmetric confidence intervals of width

$$\sqrt{2\mathcal{T}_\delta(n_{k,\cdot}(r)) g\left(\frac{1}{n_{k,\cdot}(r)}\right)}$$

for some arm k and some linear function g –which ensures that the confidence interval holds with probability $1 - \delta$– are usually quite conservative. Indeed, their reported empirical error frequency is usually a lot lower than the expected bound δ. As often done in other fixed-confidence best arm identification papers (Kaufmann and Kalyanakrishnan, 2013; Réda, Kaufmann, and Delahaye-Duriez, 2021), to decrease the exploration cost, instead of implementing the theoretical value of $\mathcal{T}_\delta(\cdot)$, we consider for any $N \in \mathbb{N}^M$

$$\mathcal{T}^{\text{heur}}_\delta(N) := \log\left(\left(1 + \log\left(\sum_{n \in [M]} N_n \right) \right) / \delta \right) \ .$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

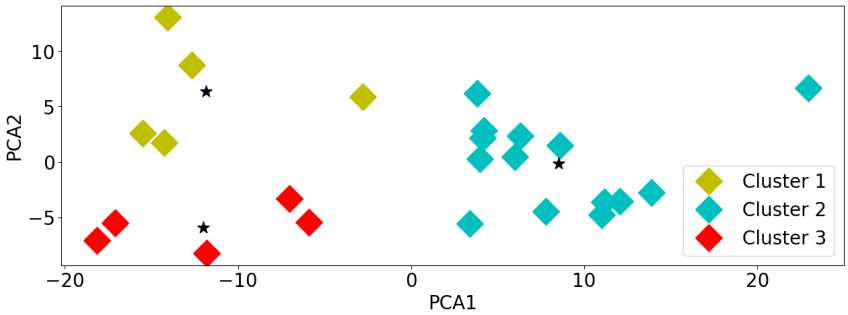

Figure 8.1: **Clustering of epileptic profiles.** PCA plot with 2 principal components of patient profiles (centers are denoted by a star).

For the baseline PF-UCB-BAI introduced above, which relies on a function f of the length of the exploration phase at phase p, we consider $f(p) := 2^p \log(1/\delta)$ for $p \geq 0$. Our experiment compares the performance of Algorithm 8 with the baseline algorithm PF-UCB-BAI on the drug repurposing instance mentioned in Chapter 3 with $K = 21$ drugs –note that here, we do not take into account the feature vectors in $\mathbb{R}^{194}$, so we do not need to apply feature transformations as we did in Chapters 5 and 6. We consider the setting where $N = 1$ and the weight matrix is of the form $\alpha I_M + (1-\alpha)/M \mathbb{1}_{M\times M}$, since the correctness of PF-UCB-BAI only holds in that case. We set $\delta = 0.1$, and change the degree of personalization $\alpha \in \{0.4, 0.6, 0.8\}$.

In order to determine the M clusters of patients, we first applied a Principal Component Analysis (PCA) transformation (using the first two principal components) on the standard-normalized matrix of transcriptomic profiles from epileptic patients (Mirza, Appleton, et al., 2017). Then, we ran k-means++ (Arthur and Vassilvitskii, 2006) with M clusters. The clusters and the patient datapoints on the PCA-transformed standard-normalized matrix are represented on the scatter plot on Figure 8.1. According to the scatter plot of the datapoints, the choice $M = 3$ seemed the most appropriate. Then, per cluster $m \in [M]$ and treatment $k \in [K]$, we defined $\mu_{(k,m)}$ as the average repurposing score (as defined in Chapter 3) obtained by treatment $k \in [K]$ across all patients in cluster m.

Results are shown in Table 8.1. Indeed, using the "heuristic" confidence interval has no consequences on the empirical error frequency $\hat{\delta}$. In terms of exploration and communication cost, CPE (Algorithm 8) improves considerably over the baseline, while being –for this specific instance of drug repurposing– extremely robust to changes in α, and thus to changes in Δ'_{min}.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

α	0.4		0.6		0.8	
Δ'_{min}	≈ 0.083		≈ 0.082		≈ 0.079	
	$\hat{s}$	$\hat{r}$	$\hat{s}$	$\hat{r}$	$\hat{s}$	$\hat{r}$
CPE	$47,418 \pm 1,016$	6	$59,537 \pm 1,677$	6	$70,201 \pm 2,300$	6
PF-UCB-BAI	$230,836 \pm 55,904$	13	$92,278 \pm 1,071$	12	$64,393 \pm 376$	11

Table 8.1: **Results for collaborative drug repurposing.** Experiments on collaborative Top-N identification ($\delta = 10$ %, $N = 1$, $K = 21$, $M = 3$, 100 iterations). Empirical error rate $\hat{\delta} = 0$ everytime. $\hat{s}$ is the average sample complexity (rounded up to the closest integer, $\pm$ standard deviation), and $\hat{r}$ is the average number of communication rounds (which standard deviation is always strictly smaller than 1).

8.6 Discussion

This work introduced a general framework for collaborative, or centralized, learning in multi-armed bandits, along with a lower bound on the associated exploration cost. Furthermore, we proposed a phased elimination algorithm for fixed-confidence Top-N identification. This algorithm tracks the optimal allocation from the pure exploration lower bound by considering a relaxed optimization problem instead, which is in contrast with prior works. Its exploration cost is matching the lower bound up to logarithmic factors, within a reasonable number of communication rounds.

As mentioned in introduction, our collaborative setting was motivated by the design of collaborative adaptive clinical trials for personalized drug recommendations, where several patient subpopulations (for instance, representing several subtypes of cancer) are considered and sequentially treated. However, in practice, especially when dealing with patient data, disclosing the mean response values to the central servor should be handled with care to preserve the anonymity of the patients. As such, the current solution is useful only to trustful partners. A possible solution to overcome this problem would be to carefully combine our algorithm with a data privacy-preserving method, for instance by adding some noise to the data (Dubey and Pentland, 2020).

Finally, one might wonder why a given subpopulation should be interested in outputs from other subpopulations (and, as such, be associated with a non-entirely diagonal weight matrix). In the context of rare diseases or subtypes, the number of patients might not be large to give a robust estimate of the success rate/score of the selected arm, hence the use of outcomes from other populations, weighted by their similarity to the original set of patients. A first approximation of this setting is the suggested "fixed weighted collaborative setting" introduced in this chapter. An interesting

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

subsequent work could rely on the relaxation of these fixed weights to a more adaptive weighting scheme. Another possibility is considering a kernel function instead of the fixed weights, as done in Du et al. (2021) ; however, it raises the issue of selecting an appropriate kernel, as opposed to computing the similarity coefficients in our use case.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Conclusion

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Summary

My PhD aims at tackling the problem of drug repurposing. This is the product of an interdisciplinary work, combining bioinformatics, through the modelling and analysis of drug-induced transcriptomic changes, and Computer Science, *via* the use of machine learning techniques for recommendation of treatments. I have focused on the approach of signature reversion, which compares the treatment-induced genewise expression changes to the expected changes associated with a good treatment. The main objective of this PhD was to develop a method which is disease-agnostic and easily reproducible –that is, that can be used in a generic and systematic fashion.

Chapter 7 aimed at describing a real-life application of signature reversion, to identify the optimal treatment protocol for stem cells against encephalopathy of prematurity, based on *in vivo* rat models. However, the main issue in drug repurposing based on signature reversion is data availability, especially for specific cell lines. To overcome this hurdle, Chapter 3 describes a method that is able to score the effect of a treatment on specific cell lines, based on drug signatures carefully built on *in vitro* experiments, and a dynamical system (Boolean network) inferred from scratch in Chapter 2. This network plays an important role in scoring treatments, as this dynamical system is able to score the effect of a treatment on a given patient for the considered disease. This score was slightly more predictive than traditional methods for signature reversion. Nonetheless, the previous method, when run on a set of repurposable drugs –which can be as large as millions of drugs to be screened– can be time-consuming. This motivates the works described in Chapters 5 and 6, which investigate the use of a specific type of sequential learning algorithms, called multi-armed bandit algorithms. Multi-armed bandits might be helpful to reduce the number of calls to the simulations run by the Boolean network, in order to identify the N most promising drug candidates. This approach can be extended to a collaborative context, where drug candidates can be simultaneously identified for heterogeneous subpopulations of patients, as described in Chapter 8.

Application in a more realistic setting is laid out in Appendix 14. It confirms the theoretical and empirical observations made throughout the thesis, and shows that the methods developed during the time course of the PhD can be robust in the face of real life data. This thesis is a new step towards the full automation of drug development research, which might allow the speed-up of early clinical phases for the identification of interesting molecules, and further increase reprodubility of clinical results. Tackling poor data availability is particularly of interest when dealing with rare diseases.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Next steps

The importance of feature selection is highlighted throughout the thesis ; more particularly in Chapter 3, where the choice of the appropriate cell line to compute drug signatures impacts the identification of gene targets ; and in Chapter 6, where the number of samples needed at the start of the algorithm is correlated with feature collinearity. Several other venues for research could be explored :

Adding new types of features in the bandit algorithm. In order to better estimate the similarity between two drugs with close rewards from the gene regulatory network, one might be interested in adding supplementary information in feature matrix X. For instance, information about the chemical structure of the molecules, or about their drugability, that is, their ability to be administered as a drug. Even if that supplementary information is not exploited in the simulation of regulatory cascades, it may have an impact both on the sample-efficiency of the algorithm, and on discriminating between two drug candidates with similar transcriptomic effects but different potentials for commercialization.

Improving the inference of the Boolean network by integrating non-coding elements. Non-coding elements in the DNA might have a regulatory impact by binding to specific transcription factors for instance, without actually coding for a product. Inspired by Réda and Wilczyński (2020), adding the regulatory interactions between genes and non-coding elements might add a biologically meaningful structure which additionally constraints the network. The main challenges for this integration are the retrieval of relevant non-coding elements across cell lines, and assessing their true impact on regulation.

Interpretation of the parameters learnt in the bandit algorithm. Although frequently mentioned in order to justify the choice of a given structure for a bandit model, a proper framework to analyze and exploit the parameters learnt during the sampling phase is missing. In the algorithm for misspecified linear models (Chapter 6), a sequential statistical test is performed to assess whether the null hypothesis, which claims that the current empirical model is an alternative model to the true model, should be rejected. In that case, if the empirical model is out of the alternative set at some point in time, one might be interested in leveraging the values of the coefficients associated with feature vectors to determine the role of a given feature –in our case, the change in expression for a gene– on the observed scores.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Controlling for the dose-related and exposure time effects. I have focused most of my PhD on carefully crafting the selection process of transcriptomic profiles to obtain the most relevant drug signatures. This process relied on the selection of the appropriate cell line, as shown in Chapter 3. However, as demonstrated by the case study in Chapter 7, other parameters might significantly impact the changes on transcription, such as the dose or time of administration. Again, new challenges arise as there is a need to simulate the dose and time-related effect for drugs which have never been tried on patients.

Towards adaptive clinical and drug repurposing trials. Finally, Chapter 8 offers a glimpse into a new framework of uncovering new drug candidates, by sharing information across patient subpopulations in an communication-efficient way. However, this raises the issue of data privacy, as potentially, the actual outputs shared from agents to the central controller might be enough to threaten the anonymity of patients. This is one of the main topics studied in distributed or federated learning, which would be another approach to investigate the use of the drug repurposing method proposed in this thesis.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Appendix

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 9

Chapter 2 : Building the cell line specific Boolean network

9.1 Building the Boolean network

This section of the Appendix describes in details the procedure used to infer a Boolean network in our application to epilepsy.

Step (A) Building a undirected unsigned graph

This step builds an undirected, unsigned network of putative gene-to-gene regulations. The M30 module, as defined by Delahaye-Duriez, Srivastava, et al. (2016), comprises of 320 genes which global expression anticorrelates with epileptic phenotypes. We retrieved all 320 genes of the M30 module from the additional file 1 in Delahaye-Duriez, Srivastava, et al. (2016). Undirected and unsigned protein-pairwise interactions are then retrieved from the STRING database (Szklarczyk et al., 2021) on this specific set of genes for the human (NCBI taxon ID 9606). In order to perform the inference, it is necessary for computational reasons to restrict the set of edges to consider ; however, (weak) connectivity in the graph of interactions should also be preserved to fully exploit the dynamical constraints provided later on.

Considering the full network retrieved from the STRING database, we trimmed out isolated genes (*i.e.*, without any interactions with any gene, not even themselves). For the M30 module, 318 genes out of 320 were retained after this procedure, with a total of 14,662 undirected regulatory interactions. The STRING database also provides scores associated with

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

each undirected edge named "combined scores", comprised between 0 and $1,000$, which are an aggregation of various scores related to the type of evidence supporting these edges (Von Mering et al., 2005). This higher this score is, the more strongly supported the associated edge is. Provided a user-provided threshold η, we built the resulting gene regulatory network (GRN) by first preserving all edges with a STRING "combined score" greater than η ; second, considering all edges which contain at least one gene that do not appear in the set of edges at Step (1), we sorted them in the order of decreasing STRING "combined scores", and added them sequentially (by batch of edges with the same STRING "combined score") to the network until the number of weakly connected components is 1. We tested (weak) connectivity by performing a Depth-First Search (Cormen et al., 2009), which is a well-known procedure that explores all the nodes in a graph by favoring the exploration of child nodes instead of sibling nodes, until all nodes have been visited.

In order to select the threshold $\eta = 400$, we performed a gridsearch on $[\![100; 1,000]\!]$ with a step of 5, and selected the value which minimized the number of edges. This step is automatically performed the first time the inference procedure in the repository [1] is run, such that the user can use the threshold value η recommended by the grisearch. Choosing $\eta = 400$ allowed reducing the number of undirected edges from $14,662$ to $1,633$.

Step (A) Gene perturbation experiments

After that step, we restricted the set of genes (and thus, of interactions) to genes present in the database of transcriptional profiles LINCS L1000 (Subramanian et al., 2017). In order to filter out genes, we first converted all gene identifiers in M30 into EntrezGene IDs using BioDBnet (Mudunuri et al., 2009). Then, we filtered out genes for which EntrezGene ID was not present in LINCS L1000 (using the API in LINCS L1000). After this step, 236 genes, out of 318, were retained. We selected all experiments present in LINCS L1000 such that at least one gene from M30 has been perturbed in a genetic experiment (knockdown or overexpression, along with control samples). All experiments involved any brain cell line in LINCS L1000 in our application to epilepsy. Unfortunately, there are no hippocampal neuron human (HN-h) cell lines in LINCS L1000, which are able to differenciate into neurons and glial cells as shown in the rat (Eves et al., 1992) ; although we can assume the neural progenitor cell (NPC) line present in LINCS L1000 might be appropriate. Among the genetic perturbation experiments listed in

[1] https://github.com/clreda/PrioritizationMasterRegulators

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

the database, we selected those which satisfy all following conditions

- which are the richest, by considering metric `distil_ss` provided by the LINCS L1000 API, which is correlated to the number of significantly differentially expressed transcripts found in the differential analysis between the matching genetically treated and the control groups. In practice, this measure is correlated to the reproducibility of a drug signature (Lim and Pavlidis, 2021).

- where there is at least two replicates from the same plate for the perturbed (of type `pert_sh`) and control (of type `ctl_vector`) conditions.

- which interference scale, as described in Cheng and Li (2016), is positive. This ensures that the associated genetic perturbation experiment was successful, meaning that a gene which has been perturbed by a knockdown (resp., an overexpression) has an expression lower (resp., greater) in treated profiles than in controls, compared to an appropriate housekeeping gene. The expression of the housekeeping gene should not dramatically change in both groups of profiles. A list of housekeeping genes is provided by Cheng and Li (2016).

- where the associated experiment is either using shRNA (knockdown perturbation), cDNA, also known as knock-in (overexpression perturbation), or CRISPR (knockout perturbation).

- where the associated cell line is either SHSY-5Y (neuroblastoma) or NPC (neural progenitor cells), which are the only brain cell lines in LINCS L1000.

The result of this step is a matrix of M30 genes by experimental profiles, which contains Level 3 LINCS L1000 data (normalized expression data for the whole genome) for each perturbation experiment. See Table 9.2 in Appendix for the list of experimental profiles used in the application to epilepsy.

Step (B) Binarization of experiments into binary profiles

Although there are known methods for the binarization of (single) RNA-seq data (Béal et al., 2019; Finak et al., 2015), probably due to the fact Level 3 LINCS L1000 data is a combination of measured and inferred expression data, for different platforms, [2] there were issues with the model fitting. Only a few genes were assigned a binary value 0 or 1 –the alternative being that they are not considered expressed "enough", according to the thresholds computed by these methods, to be assigned a state equal to 1, nor too

[2] RNA-sequencing data for the most recent version, microarray for the first generated profiles.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

weakly expressed to be assigned a state equal to 0. A data-driven method to tune the granularity of the binarization, adaptive to the selected perturbation expression data, was necessary in order to explicitly enforce a trade-off between a full reliance on the undirected edges provided by the STRING database, and on the experimental profiles from LINCS L1000.

Binarization. We designed an *ad hoc* binarization method to satisfy these constraints. This binarization was performed independently on each cell line. Gene expression (in normalized RNA counts) data was first quantile-normalized and clipped to the interval $[0, 1]$. Control samples, for the same cell line, were aggregated by considering the genewise median expression value. Given the threshold ζ, all genes with expression greater than $1 - \zeta$ were considered greatly expressed (with assigned state 1), whereas genes with expression lower than ζ were considered non-expressed (with assigned state 0). Genes which expression levels were in the interval $[\zeta, 1 - \zeta]$ had an undetermined expression state. Note that the quantile normalization is necessary, even though the initial expression data was normalized, in order to apply a same threshold ζ on all profiles. The higher ζ is, the more constrained the experiments are, as more genes have a determined expression state 0 or 1. Lower thresholds mean less constrained experiments, and a higher preference for the regulatory interactions filtered from the STRING database over expression data from LINCS L1000.

Using a bisection method in interval $[0; 0.5]$ with precision 0.005, we identified $\zeta = 0.265$ as the maximum threshold such that the inference of Boolean networks described in the next sections admits at least one solution. We recommend using this bisection method to determine the threshold ζ when using the pipeline with another dataset.

Background expression data. However, this method relies on having enough data to compute reliable statistics of expression (minimum, maximum, mean) for each gene, which is why, for each cell line, we automatically retrieved from LINCS L1000 a "background" expression matrix, which we concatenated to the set of profiles before binarization. After binarization, we removed samples associated with the background dataset. In order to collect the background expression matrix, we selected all experiments in the considered cell line, with type `pert_sh` (knockdown experiments), and we filtered out experiments with less than two replicates, with metric `distil_cc_q75` greater or equal to 0.2, and with metric `pct_self_rank_q25` lower than 0.05. Metrics `distil_cc_q75` and `pct_self_rank_q25` are two measures associated with experimental profiles which quantify the reproducibility based on the correlation between the same technical replicates (`distil_cc_q75`) and the diversity of profiles for a given experimental setting (`pct_self_rank_q25`).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

These rules correspond to the requirements for reproducible and distinct ("gold") profiles according to LINCS L1000 documentation. Finally, we selected the same-plate replicates with the highest value of `distil_ss`. [3]

Step (B) Implementation of topological constraints

The inference of a Boolean network relies on a set of admissible regulatory interactions and a set of time-series expression constraints. Indeed, solution networks only comprise of admissible interactions, such that all constraints provided by the experimental transcriptional observations are valid.

First, to build the set of admissible interactions (topological constraints), we restricted the selection of interactions to the network extracted from the STRING database. Since STRING-extracted interactions are unsigned, we decided to reduce the number of possible interactions by using the gene perturbation expression matrix retrieved from LINCS L1000 (Table 9.2), in order to improve the computational cost. A Pearson's r (Bravais, 1844) gene correlation matrix was computed from these profiles, and raised to the power of β coefficient-wise, which allowed signing the interactions using pairwise correlation signs. To preserve connectivity, we built the filtered signed undirected network similarly to what we previously did with a threshold value equal to $\tau = 0.4$.

β was chosen as it is known that raising an adjacency matrix A to the power of β yields coefficients $A[i,j]$ in position (i,j) equal to the number of paths (with eventually repeated edges) between node i and node j of length β. τ was chosen as a compromise between richness of the network (number of edges) and computational cost, by a bisection search in interval $[0.01; 1]$ with step 0.005, which would be the way to go to apply our method to other datasets.

After this procedure, we removed isolated genes in the network (that is, with both in-degree and out-degree equal to 0). After this step, 232 genes were left in the network, with 637×2 putative genepairwise interactions (one for each direction between two genes). We stress on the fact that preserving connectivity will be crucial for properly exploiting the experimental data, which is why we trim out isolated genes.

[3] All these measures are further described at https://clue.io/connectopedia/glossary.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Step (B) Implementation of experimental constraints

Second, we turned to building dynamical constraints, that is, the time-series binary expression states of genes in the network, according to the gene perturbation experiments from LINCS L1000. The experiments shown in Table 9.2 comprise of control and perturbed profiles in single gene perturbation experiments (either by knockdown through shRNA, or by overexpression using cDNA).

First, these profiles were binarized using the binarization procedure described above. Then, for each single gene perturbation experiment, we considered as initial condition the profile obtained from control samples, and as final condition those from perturbed samples, which are set as a (steady) attractor states. In order to implement the new dynamics in Paulevé et al. (2020), we used the Python package BoNeSiS (Chevalier et al., 2019), which infers by answer-set programming Boolean networks (that is, both the set of regulatory interactions and regulatory functions) which satisfy the experimental constraints by only using the provided set of possible interactions. BoNeSiS is fed the binarized experimental profiles, along with their annotated gene perturbations, and the set of valid interactions determined in the previous section. Thus, the inferred GRN should satisfy all these constraints by assigning logical functions to genes and selecting regulatory interactions.

We use the procedure in BoNeSiS which randomizes the search for network solutions. Moreover, in order to avoid trivial solutions without interactions, we also implemented the constraint that the state where all genes were not expressed (*i.e.*, with expression state 0) cannot lead to any of the reported final attractor states. This constraint can be challenged, as one might assume that a network could end up in the state where all genes are turned off in a transient way, if there are some genes which are only regulated by inhibitors. However, in practice, the inference procedure without this constaint yields singularly trivial and poorly connected solutions (*i.e.*, most genes ending up without any regulators). We conjecture that it is linked to the procedure of answer-set programming, as similar methods, for instance Re:In (Dunn and Yordanov, 2019), give the option of adding supplementary constraints about the presence of an activator for some genes.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Step (C) Inference solutions and model selection

Inference of putative Boolean network solutions. In BoNeSiS, we asked for the enumeration of at most 1 solution to the set of topological and experimental constraints defined above. In the implementation of the most permissive semantics in Paulevé et al. (2020) by Chevalier et al. (2019), the size of the Boolean function specification can be upper-bounded by a pre-specified value. In my application, I have used the maximum total (ingoing and outgoing) degree of the underlying network in order to avoid spurious gene regulatory functions. Due to the intrinsic randomness stemming from the solver clingo (Gebser et al., 2016), and the randomized search procedure used in BoNeSiS, we iterated this enumeration, such that we obtained 25 Boolean network solutions (among which 25 are unique in terms of regulatory functions).

Selection of an optimal model. In order to select a "representative" network consistent with what is known about the topology of biological networks, Babichev et al. (2019) compiled a list of network measures to maximize in biological networks. Then, they computed a single scalar criterion value comprised in the interval $[0, 1]$ (called in their paper "general topological parameter") to maximize through the Harrington desirability index (Harrington, 1965). In practice, using notations from Babichev et al. (2019), we considered the following weights

$$a_{\texttt{DS}} = 3,\ a_{\texttt{CL}} = 3,\ a_{\texttt{Centr}} = 3,\ \text{and}\ a_{\texttt{GT}} = 1\ ,$$

where

- $\texttt{DS}$ corresponds to the *network density*, that is, the ratio of the number of edges to the maximum number of possible connections between the nodes in the network (that is, if the network was fully connected) ; for a network of n nodes, this maximum number is equal to $(n-1)n/2$.

- $\texttt{CL}$ corresponds to the *network clustering coefficient* which is the average of node-wise clustering coefficients. The clustering coefficient of a node is the ratio of the degree of the considered node and the maximum possible number of connections such that this node and its current neighbors form a clique (*i.e.*, form a fully connected graph).

- $\texttt{Centr}$ corresponds to the *network centralization*, which is correlated to the similarity of the network to a graph with a star topology.

- $\texttt{GT}$ corresponds to the *network heterogeneity*, which quantifies the nonuniformity of the node degrees across the network by computing the ratio

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

between the standard deviation of the node degrees and the average degree across the network.

The higher the weights, the more importance is given to having a large associated coefficient. Finally, for every network solution N returned by BoNeSiS, we computed

$$\exp\left(\mathtt{mean}\left\{-\exp\left(x \times a - 1\right) : (x, a) \in \mathcal{V}(N)\right\}\right) \ ,$$

where $\mathcal{V}(N)$ is the set of pairs (value, weight) associated with each topological measure

$$\mathcal{V}(N) := \{(\mathtt{DS}(N), a_{\mathtt{DS}}), (\mathtt{CL}(N), a_{\mathtt{CL}}), (\mathtt{Centr}(N), a_{\mathtt{Centr}}), (\mathtt{GT}(N), a_{\mathtt{GT}})\} \ .$$

The final network was the one which maximized this quantity.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

9.2 Robustness on a larger set of 50 solutions

Since the enumeration of solutions is still computationally expensive and time-consuming, we focused our work on a collection of 25 solutions. However, in order to assess the robustness of our inference procedure, we enumerated an additional set of 25 solutions, and reproduced the two plots shown in the results. Note that these 25 solutions were different from the first 25 ones, yielding a set of 50 unique solutions (in terms of gene regulatory functions). The selection of the optimal model run on these 50 models still returned the same network as shown in the main paper. Table 2.1 and Figure 2.3 allows us to conclude similarly to the main paper that the networks obtained just before the network selection step are mostly functionally and topologically similar.

	Min.	25th quantile	Median	Mean	75th quantile	Max.
# RFs	1	1	2	2.635	3	17
GTP	0.794	0.796	0.797	0.797	0.799	0.802

Table 9.1: Distribution statistics on the number of *unique* regulatory functions (RFs) across solutions per gene, and on the value of the general topological parameter (GTP) used for network selection in step (C) of the inference procedure. All values are rounded up to the 3rd decimal place. Applied on the set of 50 solutions.

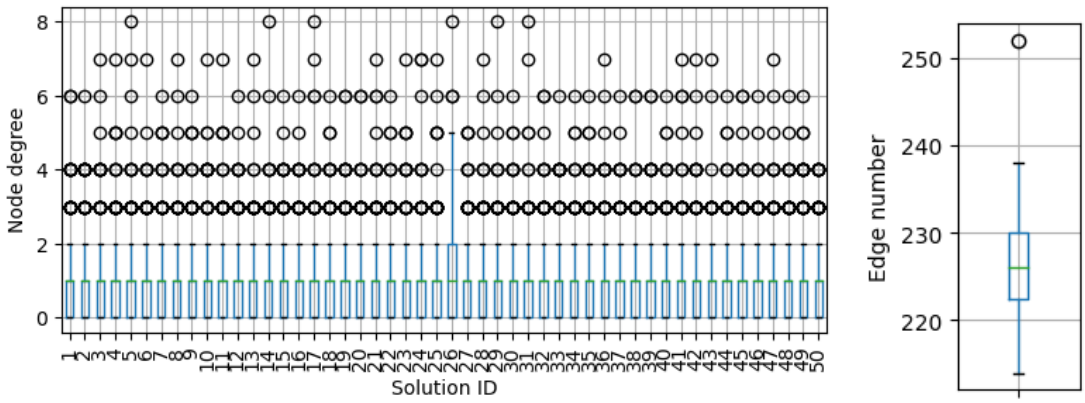

Figure 9.1: *Left-hand plot*: Boxplots of node total (ingoing and outgoing) degrees per solution. The green lines represent median values. *Right-hand plot*: Boxplot of the number of edges across solutions (each solution comprises of 232 nodes). The green line represent the median value. Applied on the set of 50 solutions. Note that the first 25 boxplots match the plot in Figure 2.3.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

9.3 Tables

Experimental profiles from LINCS L1000

Profile brew identifier (suffix)	Cell line	Gene	Type	Time**	Dose	Nb#
KDB003_NPC_96H	NPC	PSMG1	KD*	96	$1.5\mu L$	4
Samples {X1,2.A2,X2,X3.A2}						
T B6_DUO52HI53LO:K16						
C B6_DUO52HI53LO:F13						
EKW001_SHSY5Y_120H	SHSY-5Y	SOD1	KD	120	N/A	3
Samples {X1,X2,X3}						
T F1B3_DUO52HI53LO:J20						
C F1B3_DUO52HI53LO:I05						
T F1B3_DUO52HI53LO:H17		SYT1	KD	120	N/A	3
T F1B3_DUO52HI53LO:I19		CACNA1C	KD	120	N/A	3
T F1B3_DUO52HI53LO:A03		CDC42	KD	120	N/A	3

Table 9.2: Experimental profiles retrieved from LINCS L1000 for the application to epilepsy, as annotated in LINCS L1000. * KD stands for knockdown. ** Time (in hours) of exposure to the perturbagen. # number of replicates. **T** (resp., **C**) stands for **treated** (resp., **control**).

Parameters

	Definition	Value
η	Threshold for selecting edges from STRING	400
ζ	Threshold for the binarization step	0.265
β	Power applied to the matrix of genepairwise correlations	1
τ	Threshold for filtering out edges in the putative network	0.4

Table 9.3: Parameter values for the synthesis of Boolean networks (in the application to drug-resistant epilepsy). STRING refers to the STRING database (Szklarczyk et al., 2021).

	Score	EI	DSI	DPI	Source
Value	0	0	0.25	0	CURATED

Table 9.4: Parameter values (minimal values) for retrieving genes associated with a specific disease from DisGeNet (Piñero et al., 2020). The full definitions of these indices are reported at this page.[a] EI : Evidence Index. DSI : Disease Specificity Index. DPI : Disease Pleiotropy Index.

[a] DisGeNet (2022). *FAQ : Original Data Sources*. https://www.disgenet.org/. Accessed: [May 4, 2022].

Except where otherwise noted, this work is licensed under https://creativecommons.org/licenses/by-nd/3.0/fr/

9.4 Implementation of the influence maximization algorithm

This section deals with supplementary data about the implementation of the influence maximization procedure.

Iteration of attractor states

In order to enumerate attractors under perturbations, we used PyMaBoSS (Stoll et al., 2017). We ran PyMaBoSS with $1,000$ trajectories, for reachable attractors within 50 time steps, and parameters $\texttt{time_tick} = 1$, $\texttt{use_physrandgen} = 0$. Unfortunately, this method does not guarantee the similarity of attractors from one iteration to another, but our tests showed that, although there is some noticeable change in the resulting spread values, it does not affect the final ranking on genes. We never had to deal with the case where no attractor state is retrieved with these parameter values.

Choice of initial states

In our application, we considered the integration of a disease-specific context by considering 24 hippocampi normalized transcriptional profiles of humans affected with Temporal Lobe Epilepsy (TLE) (Mirza, Appleton, et al., 2017) (EMTAB 3123 on ArrayExpress). The main idea is that we specifically target genes which perturbative power is high in a transcriptional context for epilepsy. We restricted these epileptic profiles to genes present in the network, by retrieving their associated Entrez ID identifiers and by matching with identifiers in LINCS $\mathrm{L}1000$. Then, we binarized the profiles according to the binarization procedure described in the first section, with corresponding threshold ζ equal to 0.5, so that all genes have a determined binary expression state.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

9.5 Additional results

This section shows additional results related to the inference of the Boolean network and the spread values.

Regulator	Regulated	Sign	Evidence source*
RBFOX1	PEG3	Inhibitory	Coexpression
SLITRK3	IL1RAP	Inhibitory	Text-mining
TSPAN7	AFF2	Activatory	Coexpression
UQCRQ	TIMM17A	Inhibitory	Coexpression
CENPJ	ANAPC1	Activatory	Coexpression
SYT13	MLLT11	Inhibitory	Coexpression
GUCY1B3	AHNAK2	Activatory	Text-mining
PLEKHG3	CCDC68	Inhibitory	Text-mining
MLLT11	TTC3	Activatory	Text-mining
SULT4A1	KIAA0232	Inhibitory	Text-mining
GRIN1	CIT	Activatory	Interaction
GPI	ANXA6	Inhibitory	Text-mining
RBFOX1	ZMAT4	Activatory	Coexpression
FAR2	FAM49A	Inhibitory	Text-mining
RAB3A	REEP2	Activatory	Coexpression
CAMK2B	AGGF1	Inhibitory	Association in databases
GAP43	ELAVL2	Inhibitory	Coexpression
ADAM22	EPB41L3	Inhibitory	Coexpression
CDC42	ARHGAP44	Activatory	Interaction
GNB5	PAK1	Inhibitory	Association in databases
STMN2	PSMG1	Inhibitory	Coexpression
AMPH	AAGAB	Activatory	Interaction
GNB5	PRKCE	Inhibitory	Association in databases
ATP1B3	FXYD7	Inhibitory	Association in databases
ATP1A3	PANK2	Activatory	Text-mining

Table 9.5: Regulatory interactions present in all of the 25 solutions. * strongest evidence source from the STRING database. "Association in databases" means associated in curated pathway databases.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

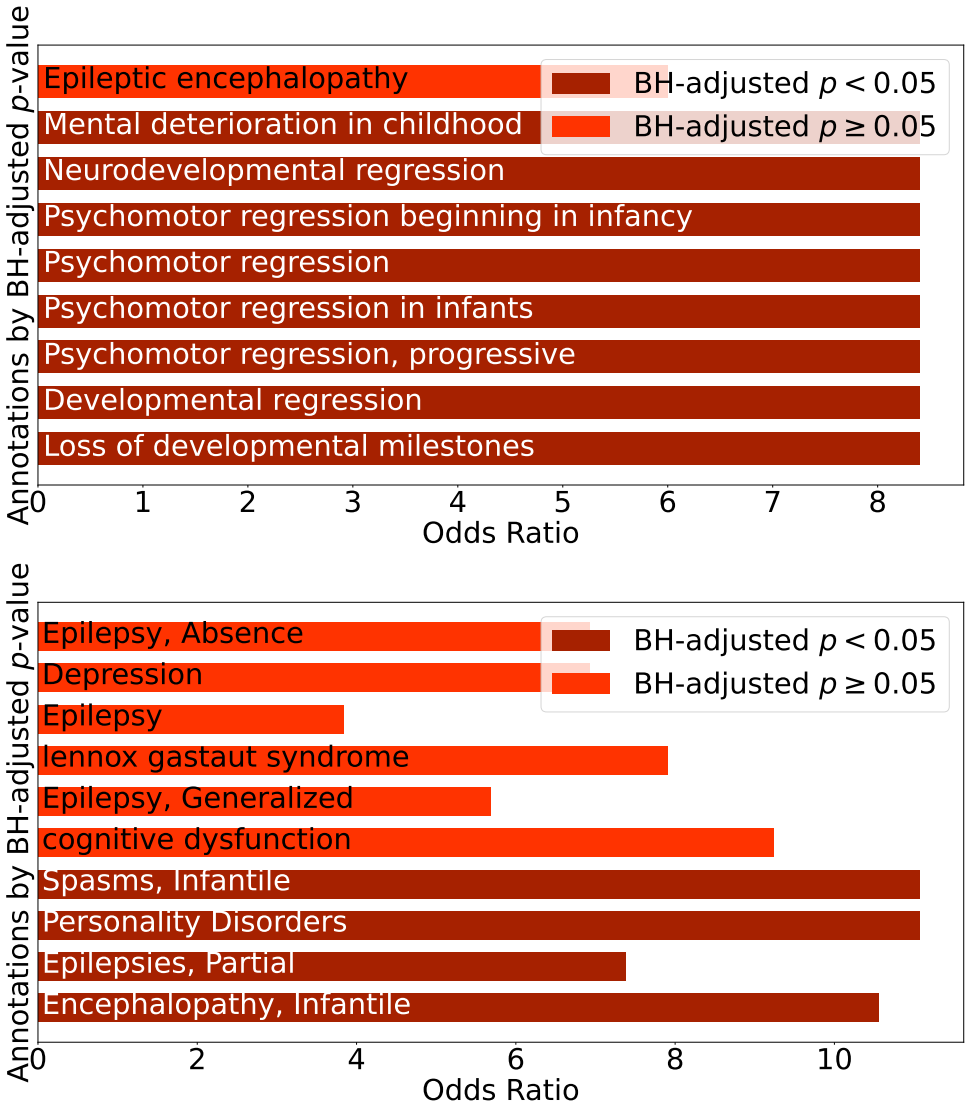

Figure 9.2: Enrichment results from the ORA analysis on the filtered list of genes based on spread values from the DisGeNet annotations (Piñero et al., 2020) (*top*) and GLAD4U (Jourquin et al., 2012) (*bottom*). The top-10 annotations (in increasing order of BH-adjusted p-value) are reported. All of these adjusted p-values are lower than 20%.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 10

Chapter 3: Drug repurposing scoring

10.1 List of antiepileptic and proconvulsant drugs

The initial tables of 71 epilepsy-related drugs on the next page (Tables 10.1 and 10.2) have been compiled thanks to Baptiste Porte.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

PubChem ID (best match)	Drug name	Reference	"BrainCell"
5070	Riluzole	Ground truth	Yes
3121	Valproic-Acid	Ground truth	Yes
3878	Lamotrigine	Ground truth	Yes
2554	Carbamazepine	Ground truth	Yes
4909	Primidone	Ground truth	Yes
3331	Felbamate	Ground truth	Yes
5734	Hydroxyzine	Ground truth	Yes
5360696	Dextromethorphan	Exploratory	Yes
4107	Methocarbamol	Exploratory	Yes
265237	Withaferin-A	Exploratory	Yes
3038	Diclofenamide	Proven	Yes
441074	Quinidine	Proven	Yes
4843	Piracetam	Ground truth	
2118	Alprazolam	Ground truth	
3292	Ethotoin	Ground truth	
5719	Zaleplon	Ground truth	
4506	Nitrazepam	Ground truth	
5311454	Stiripentol	Ground truth	
1775	Phenytoin	Ground truth	
4737	Pentobarbital	Ground truth	
5576	Trimethadione	Ground truth	
6839	Phensuximide	Ground truth	
5284627	Topiramate	Ground truth	
3291	Ethosuximide	Ground truth	
34312	Oxcarbazepine	Ground truth	
1986	Acetazolamide	Ground truth	
3016	Diazepam	Ground truth	
3446	Gabapentin	Ground truth	
5665	Vigabatrin	Ground truth	
5284583	Levetiracetam	Ground truth	
5732	Zolpidem	Proven	
3516	Guaifenesin	Exploratory	
2733	Chlorzoxazone	Exploratory	
2712	Chlordiazepoxide	Exploratory	
5735	Zopiclone	Exploratory	
3440	Furosemide	Exploratory	

Table 10.1: Initial list of antiepileptics with their associated references, and annotated presence of existing "BrainCell" signatures as described in Chapter 3. Drugs with existing "BrainCell" signatures are included in the ranking shown in Figure 3.3. Antiepileptics are classified in three categories: "Ground truth" (well-known antiepileptics, according to Epilepsy.com (2022) for instance), "Proven" (antiepileptics listed for epileptic encephalopathies in recent works (Johannessen Landmark et al., 2021; Pepi et al., 2021) and/or proven antiepileptic effect), and "Exploratory" (observed antiepileptic effect in animal).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

PubChem ID (best match)	Drug name	Class	"BrainCell"
2310	Bemegride	Ground truth	Yes
104999	Dmcm	Ground truth	Yes
5917	Pentylenetetrazol	Ground truth	Yes
5904	Benzylpenicillin	Ground truth	Yes
4993	Pyrimethamine	Proven	Yes
3676	Lidocaine	Proven	Yes
442021	Brucine	Proven	Yes
2519	Caffeine	Proven	Yes
2170	Amoxapine	Proven	Yes
2719	Chloroquine	Proven	Yes
3151	Domperidone	Proven	Yes
9915886	Thiocolchicoside	Proven	Yes
54687	Pravastatin	Exploratory	Yes
16362	Pimozide	Exploratory	Yes
442872	Securinine	Exploratory	Yes
5472	Ticlopidine	Exploratory	Yes
4828	Pindolol	Exploratory	Yes
39042	Bezafibrate	Exploratory	Yes
1548943	Capsaicin	Exploratory	Yes
5095	Ropinirole	Exploratory	Yes
4184	Mianserin	Exploratory	Yes
2708	Chlorambucil	Exploratory	Yes
3767	Isoniazid	Proven	
2153	Theophylline	Proven	
6167	Colchicine	Proven	
3156	Doxapram	Proven	
92722	Argatroban	Exploratory	
938	Niacin	Exploratory	
441130	Meropenem	Exploratory	
5479	Tinidazole	Exploratory	
4583	Ofloxacin	Exploratory	
53025	Cefotetan	Exploratory	
12035	Acetylcysteine	Exploratory	
753	Glycerin	Exploratory	
3823	Ketoconazole	Exploratory	

Table 10.2: Initial list of proconvulsants with their associated references, and annotated presence of existing "BrainCell" signatures as described in Chapter 3. Drugs with existing "BrainCell" signatures are included in the ranking in Figure 3.3. Similarly to Table 10.1, I rank drugs in three categories : "Exploratory" (observed proconvulsant effect in animal or human), "Proven" (proven proconvulsant effect), "Ground truth" (in the class of proconvulsant drugs).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

10.2 Drug repurposing instances in bandits

Note that the scores shown in Table 10.3 do not match the average scores in the right-hand boxplot in Figure 3.3, although the resulting ranking is similar to the one shown in the figure. This model has been obtained on the very same set of signatures as the one shown in Chapter 3.3. The one difference between the two models is in the initial patient states. On the one shown in Chapter 3, gene identifiers for genes measured in the profiles from Mirza, Appleton, et al. (2017) were converted into gene symbols present in LINCS L1000, which guaranteed a greater number of matches (207 matched genes instead of 194) than using the gene identifiers provided in the initial dataset (which were the profiles used for computing the scores shown in Table 10.3).

Drug name	Effect	Average repurposing score
Withaferin-A	Antiepileptic	-0.110999
Carbamazepine	Antiepileptic	-0.196116
Quinidine	Antiepileptic	-0.207422
Hydroxyzine	Antiepileptic	-0.353716
Diclofenamide	Antiepileptic	-0.395736
Dextromethorphan	Antiepileptic	-0.456390
Lamotrigine	Antiepileptic	-0.492804
Felbamate	Antiepileptic	-0.503488
Valproic-Acid	Antiepileptic	-0.518016
Primidone	Antiepileptic	-0.526466
Pimozide	Proconvulsant	-0.595459
Chlorambucil	Proconvulsant	-0.605096
Domperidone	Proconvulsant	-0.605607
Securinine	Proconvulsant	-0.607568
Caffeine	Proconvulsant	-0.676425
Pindolol	Proconvulsant	-0.694313
Capsaicin	Proconvulsant	-0.757241
Pentylenetetrazol	Proconvulsant	-0.760638
Pravastatin	Proconvulsant	-0.819149
Ropinirole	Proconvulsant	-1.001871
Amoxapine	Proconvulsant	-1.151488

Table 10.3: Bandit instance selected from the set of 34 ranked drugs using their "BrainCell" drug signatures (refer to Chapter 3). This instance is considered as it is in Chapter 8, whereas feature transformation (to make the model linear, resp. misspecified linear) is applied to the drug signatures in Chapter 5, resp. Chapter 6.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 11

Chapter 7 : Results for the cosine method applied to the "stabilized" profiles

We report on Figure 11.1 the ranking, the ROC and PR curves obtained by computing a cosine similarity score between "stabilized" binary profiles predicted by the Boolean network, and the "disease" signature for epilepsy CD[Healthy||Patient] based on Characteristic Direction (Clark et al., 2014). The reported baseline method is L1000 CDS^2 (Duan et al., 2016), performed as described in Chapter 3. Note that on this larger dataset, the AUC is equal to the one reported in Chapter 3 for the drug repurposing method we proposed. However, the precision-recall curve is significantly worse.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

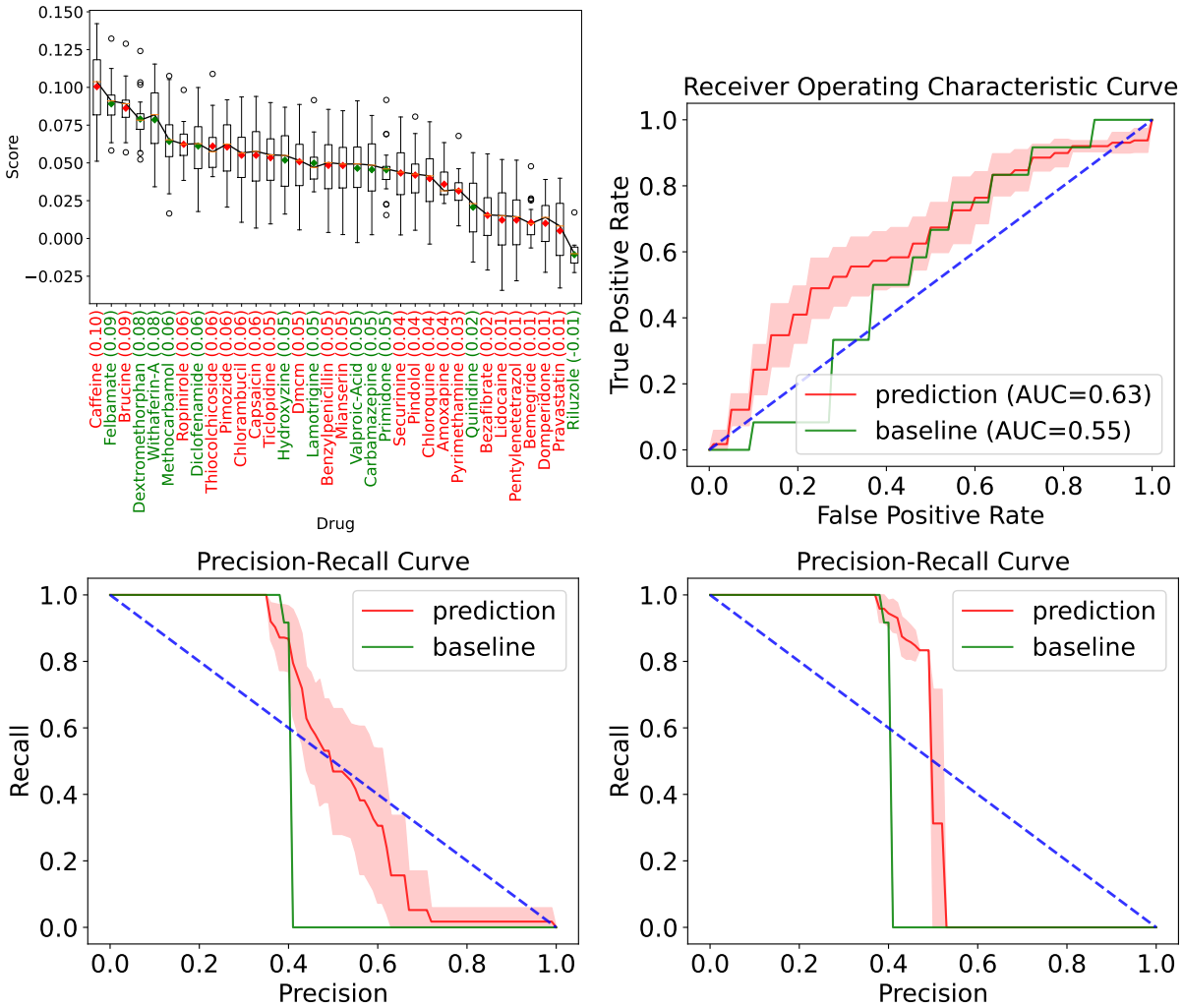

Figure 11.1: **Cosine method applied to the whole set of 34 drugs.** *Top left plot* : Ranking obtained by computing cosine scores on the "stabilized" profiles predicted by the Boolean network. *Top right plot* : the associated ROC curve, where the baseline is L1000 CDS^2 (Duan et al., 2016). *Bottom left plot* : the associated PR curve. *Bottom right plot* : the PR curve in Chapter 3.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 12

Chapter 5: Theoretical guarantees for GIFA

12.1 Conjecture 5.3.9: sample complexity analysis for Gap-GIFA

We consider now Gap-GIFA, where the following rules are fixed (see Table 5.1)

$$J(t) := \arg\min_{j\in[K]}^{[N]} \max_{i\neq j}^{N} \mathcal{B}_{i,j}(t) \ , \ (\texttt{compute_Jt})$$

$$\text{and } b(t) := \arg\max_{j\in J(t)} \max_{i\neq j}^{N} \mathcal{B}_{i,j}(t) \ , \ (\texttt{compute_bt})$$

and the stopping rule is

$$\tau^{\text{UGapE}} := \inf\left\{ t \in \mathbb{N}^* \mid \max_{j\in J(t)} \max_{i\neq j}^{N} \mathcal{B}_{i,j}(t) = \max_{i\neq b(t)}^{N} \mathcal{B}_{i,b(t)}(t) \leq \varepsilon \right\} \ .$$

Similarly to the proof for LUCB-GIFA in Réda, Kaufmann, and Delahaye-Duriez (2021), the objective is first to upper bound the stopping quantity $\tilde{\mathcal{B}}_{b(t)}(t) := \max_{i\neq b(t)}^{N} \mathcal{B}_{i,b(t)}(t)$.

Lemma 12.1.1. ***Gap-independent upper bound on*** $\tilde{\mathcal{B}}_{b(t)}(t)$***.*** *For any* $t > 0$*, for any gap index satisfying Definition 5.2.1,*

$$\tilde{\mathcal{B}}_{b(t)}(t) \leq \mathcal{B}_{c(t),b(t)}(t) \leq 2\max\left(\mathcal{W}_{c(t)}(t), \mathcal{W}_{b(t)}(t)\right) \ .$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Proof. Using successively Lemma 5.2.8, the triangle inequality (when considering paired gap indices), and Corollary 1 from Gabillon, Ghavamzadeh, and Lazaric (2012)

$$\begin{aligned}\tilde{\mathcal{B}}_{b(t)}(t) = \max_{i \neq b(t)}^{N} \mathcal{B}_{i,b(t)}(t) &\leq \max_{i \notin J(t)} \mathcal{B}_{i,b(t)}(t) = \mathcal{B}_{c(t),b(t)}(t) \\ &\leq \mathcal{B}^{\text{ind}}_{c(t),b(t)}(t) \\ &\leq 2 \max\left(\mathcal{W}_{c(t)}(t), \mathcal{W}_{b(t)}(t)\right) .\end{aligned}$$

□

Lemma 12.1.2. ***Upper bound on*** $(\tilde{\mathcal{B}}_{b(t)}(t))_{t>0}$ ***(Gap-GIFA).*** *For any round* $t > 0$*, on event*

$$\mathcal{E} := \bigcup_{t>0} \bigcup_{(i,j)\in[K]^2} \{\Delta_{i,j} \in [-\mathcal{B}_{j,i}(t), \mathcal{B}_{i,j}(t)]\} ,$$

$\tilde{\mathcal{B}}_{b(t)}(t) \leq \min\left(-\max\left(\Delta_{b(t)}, \Delta_{c(t)}\right) + 6\mathcal{W}_{a_t}(t), 0\right) + 2\mathcal{W}_{a_t}(t)$, *where* $a_t \in \arg\max_{a\in\{b(t),c(t)\}} \mathcal{W}_a(t)$.

Proof. Lemma 12.1.1 implies that $\tilde{\mathcal{B}}_{b(t)}(t) \leq 2\mathcal{W}_{a_t}(t)$. It means that we only have to prove that in all cases,

$$\tilde{\mathcal{B}}_{b(t)}(t) \leq -\max\left(\Delta_{b(t)}, \Delta_{c(t)}\right) + 6\mathcal{W}_{a_t}(t) .$$

Also note that, by triangle inequality, for any $t > 0$, for any pair of arms $(i, j) \in [K]^2$

$$\mathcal{W}_{i,j}(t) \leq 2\max_{a\in\{i,j\}} \mathcal{W}_a(t) .$$

Then, let us consider four cases :

(i). $b(t) \in \mathcal{S}_N^\star$ **and** $c_t \in (\mathcal{S}_N^\star)^c$**.** Using Lemma 5.2.8

$$\tilde{\mathcal{B}}_{b(t)}(t) \leq \mathcal{B}_{c_t,b_t}(t) = \mu_{c_t} - \mu_{b_t} + \mathcal{W}_{c_t,b_t}(t) .$$

Then, because $c(t) \in (\mathcal{S}_N^\star)^c$ and $b(t) \in \mathcal{S}_N^\star$

$$\begin{aligned}\tilde{\mathcal{B}}_{b(t)}(t) &\leq (\mu_{(N+1)} - \mu_{b(t)}) + \mathcal{W}_{c(t),b(t)}(t) = -\Delta_{b(t)} + \mathcal{W}_{c(t),b(t)}(t) , \\ \text{and } \tilde{\mathcal{B}}_{b(t)}(t) &\leq (\mu_{c_t} - \mu_{(N)}) + \mathcal{W}_{c(t),b(t)}(t) = -\Delta_{c(t)} + \mathcal{W}_{c(t),b(t)}(t) , \text{ and then}\end{aligned}$$

$$\tilde{\mathcal{B}}_{b(t)}(t) \leq -\max\left(\Delta_{b(t)}, \Delta_{c(t)}\right) + 2\mathcal{W}_{a_t}(t) .$$

(ii). $b(t) \in (\mathcal{S}_N^\star)^c$ **and** $c(t) \in \mathcal{S}_N^\star$**.** Using Lemma 5.2.6 (since $\mathcal{E}^{\text{GIFA}} \subseteq \mathcal{E}$)

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

combined with Lemma 12.1.1

$$\Delta_{b(t)} \leq \max_{i \neq b(t)}^{N} \mathcal{B}_{i,b(t)}(t) = \tilde{\mathcal{B}}_{b(t)}(t) \leq 2\mathcal{W}_{a_t}(t) \ ,$$

and using the fact that $c(t) \in \mathcal{S}_N^\star$, $b(t) \in (\mathcal{S}_N^\star)^c$, event $\mathcal{E}$ and Lemma 12.1.1

$\Delta_{c(t)} = \Delta_{c(t),b(t)} + \Delta_{b(t),a_{(N+1)}} < \Delta_{c(t),b(t)} + 0 \leq \mathcal{B}_{c(t),b(t)}(t) \leq 2\mathcal{W}_{a_t}(t)$, which implies that

$$\begin{aligned} \tilde{\mathcal{B}}_{b(t)}(t) &\leq 0 + 2\mathcal{W}_{a_t}(t) \leq (2\mathcal{W}_{a_t}(t) - \max\left(\Delta_{b(t)}, \Delta_{c(t)}\right)) + 2\mathcal{W}_{a_t}(t) \\ \Longrightarrow \tilde{\mathcal{B}}_{b(t)}(t) &\leq -\max\left(\Delta_{b(t)}, \Delta_{c(t)}\right) + 4\mathcal{W}_{a_t}(t) \ . \end{aligned}$$

(iii). $b(t) \in (\mathcal{S}_N^\star)^c$ **and** $c(t) \in (\mathcal{S}_N^\star)^c$**.** Since $b(t) \in (\mathcal{S}_N^\star)^c$, using Lemma 5.2.6 combined with Lemma 12.1.1

$$\Delta_{b(t)} \leq \max_{i \neq b(t)}^{N} \mathcal{B}_{i,b(t)}(t) = \tilde{\mathcal{B}}_{b(t)}(t) \leq 2\mathcal{W}_{a_t}(t) \ .$$

Moreover, there exists $c \in (J(t))^c \cap \mathcal{S}_N^\star$; otherwise, for any $c \in J(t)$, $c \in \mathcal{S}_N^\star$, and since $|J(t)| = |\mathcal{S}_N^\star| = N$, then $J(t) = \mathcal{S}_N^\star$. However, $b(t) \in J(t) \cap (\mathcal{S}_N^\star)^c$, hence there is a contradiction. Then, using successively event $\mathcal{E}$ twice, the definition of $c(t)$, event $\mathcal{E}$ again and the fact that $c \in \mathcal{S}_N^\star$

$$\begin{aligned} \Delta_{c(t),b(t)} &\geq -\mathcal{B}_{b(t),c(t)}(t) = \mathcal{B}_{c(t),b(t)}(t) - 2\mathcal{W}_{b(t),c(t)}(t) \\ &\geq \mathcal{B}_{c,b(t)}(t) - 2\mathcal{W}_{b(t),c(t)}(t) \\ &\geq \Delta_{c,b(t)} - 2\mathcal{W}_{b(t),c(t)}(t) \\ &\geq \mu_{(N)} - \mu_{b(t)} - 2\mathcal{W}_{b(t),c(t)}(t) \\ \Longrightarrow \Delta_{c(t),a_{(N)}} + 2\mathcal{W}_{b(t),c(t)}(t) &\geq 0 \\ \Longrightarrow -\Delta_{c(t)} + 4\mathcal{W}_{a_t}(t) &\geq 0 \ , \end{aligned}$$

and then we conclude similarly to **(ii)**

$$\tilde{\mathcal{B}}_{b(t)}(t) \leq -\max\left(\Delta_{b(t)}, \Delta_{c(t)}\right) + 6\mathcal{W}_{a_t}(t) \ .$$

(iv). $b(t) \in \mathcal{S}_N^\star$ **and** $c(t) \in \mathcal{S}_N^\star$**.** This is the part I haven't been able to show so far. I assume from now on that the lemma is correct. My conjecture is that, in this specific case

$$\tilde{\mathcal{B}}_{b(t)}(t) \leq -\max\left(\Delta_{b(t)}, \Delta_{c(t)}\right) + 8\mathcal{W}_{a_t}(t) \ .$$

□

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Then, similarly to the proof for LUCB-GIFA, I show an upper bound on the number of times sampled arm I_t has been sampled up to round t included

Lemma 12.1.3. ***Upper bound on the number of samples from I_t (Gap-GIFA, largest variance rule).*** *For any round $t > 0$, $t < \tau^{UGapE}$*

$$N_{I_t}(t) \leq 4\sigma^2(\mathcal{T}_\delta(t))^2 \max\left(\varepsilon, \frac{\varepsilon + \Delta_{I_t}}{3}\right)^{-2}.$$

Proof. Using successively the definition of the stopping rule τ^{UGapE} at time $t < \tau^{\text{UGapE}}$, if $a_t \in \arg\max_{a \in \{b(t), c(t)\}} \mathcal{W}_a(t)$ and I_t is sampled according to the largest variance rule (Equation (5.1)), then

$$\begin{aligned}
\varepsilon < \tilde{\mathcal{B}}_{b(t)}(t) &\leq \min\left(-\max(\Delta_{b(t)}, \Delta_{c(t)}) + 6\mathcal{W}_{a_t}(t), 0\right) + 2\mathcal{W}_{a_t}(t) \\
\max\left(\varepsilon, \frac{\varepsilon + \Delta_{b(t)}}{3}, \frac{\varepsilon + \Delta_{c(t)}}{3}\right) &\leq 2\mathcal{W}_{a_t}(t) = 2\sigma^2\mathcal{T}_\delta(t) \max_{a \in \{b(t), c(t)\}} \|X_a\|_{(\hat{V}^\kappa(t))^{-1}} \\
\max\left(\varepsilon, \frac{\varepsilon + \Delta_{I_t}}{3}\right) &\leq 2\sigma^2\mathcal{T}_\delta(t)\|X_{I_t}\|_{(\hat{V}^\kappa(t))^{-1}}
\end{aligned}$$

Then using Lemma 5.2.4, provided that $\kappa \geq 0$

$$\max\left(\varepsilon, \frac{\varepsilon + \Delta_{I_t}}{3}\right) \leq 2\sigma^2\mathcal{T}_\delta(t)\frac{\|X_{I_t}\|_2}{\sqrt{N_{I_t}(t)\|X_{I_t}\|_2^2 + \kappa}} \leq \frac{2\sigma^2\mathcal{T}_\delta(t)}{\sqrt{N_{I_t}(t)}}.$$

$$\text{which yields } N_{I_t}(t) \leq 4\sigma^2(\mathcal{T}_\delta(t))^2 \max\left(\varepsilon, \frac{\varepsilon + \Delta_{I_t}}{3}\right)^{-2}.$$

Then we apply Lemma 5.3.8, which gives us the final upper bound. □

Note that, empirically, LinGIFA has the same performance than N-LinGapE in terms of sample complexity, which makes us more confident in this conjecture.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

12.2 Conjecture 5.3.10: analysis of the greedy selection rule

Note that, for some $\kappa \geq 0$, $\forall t > 0$, $\hat{V}^\kappa(t) := \kappa I_d + \sum_{a\in[K]} N_a(t) x_a x_a^\top$. Greedy selection rule is selecting at round $t > 0$ the arm a_t which satisfies

$$a_t \in \underset{a\in[K]}{\arg\min} \, \|x_{b_t} - x_{c_t}\|_{(\hat{V}^\kappa(t-1)+x_a x_a^\top)^{-1}} \, .$$

The goal is to upper bound the quantity $\|x_{b_t} - x_{c_t}\|_{(\hat{V}^\kappa(t-1)+x_{a_t} x_{a_t}^\top)^{-1}}$ with something depending on $N_{a_t}(t)$, in order to apply Lemma 5.3.8.

Lemma 12.2.1. *For any $t > 0$, for any $a \in [K]$,*

$$||x_{b_t} - x_{c_t}||_{(\hat{V}^\kappa(t-1)+x_a x_a^\top)^{-1}} \leq \frac{\|x_{b_t} - x_{c_t}\|_2}{\sqrt{\lambda_{b_t,c_t,a}}} \, ,$$

where $(\lambda_i)_{i\leq d}$ are the eigenvalues of $(\hat{V}^\kappa(t-1) + x_a x_a^\top)$ associated with respective eigenvectors $(v_i)_{i\leq d}$, and $\lambda_{b_t,c_t,a} := \min\limits_{\substack{j\leq d \\ v_j^\top(x_{b_t}-x_{c_t})\neq 0}} \lambda_j$.

Proof. We denote $V_a := \hat{V}^\kappa(t-1) + x_a x_a^\top$. Matrix V_a is real, symmetric, and positive definite, thus all its eigenvalues $(\lambda_i)_{i\leq d}$ are positive. Then, due to the spectral theorem, these eigenvalues are all real-valued, and there exists a orthonormal basis of associated eigenvectors $(v_j)_{j\leq d}$. In particular,

$$\forall j \leq d, \; v_j = V_a^{-1} V_a v_j = \lambda_j V_a^{-1} v_j \Leftrightarrow V_a^{-1} v_j = \lambda_j^{-1} v_j \, .$$

Moreover, $\text{Span}(\{v_j\}_{j\leq d}) = \mathbb{R}^d$, which means that there exists at least one $j \leq d$ such that $v_j^\top(x_{b_t} - x_{c_t}) \neq 0$. Then

$$\begin{aligned}
\|x_{b_t} - x_{c_t}\|^2_{V_a^{-1}} &= (x_{b_t} - x_{c_t}) V_a^{-1} \sum_{j\neq d} v_j^\top (x_{b_t} - x_{c_t}) v_j \\
&= (x_{b_t} - x_{c_t})^\top \sum_{j\leq d} v_j^\top (x_{b_t} - x_{c_t}) \lambda_j^{-1} v_j \\
&\leq \lambda^{-1}_{b_t,c_t,a} \|x_{b_t} - x_{c_t}\|_2^2 \, .
\end{aligned}$$

□

Now, we will find a lower bound for the quantity λ_{b_t,c_t,a_t} depending on $N_{a_t}(t)$, for any $t > 0$, with a_t as defined by the greedy selection rule.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Lemma 12.2.2.

$$\lambda_{b_t,c_t,a_t} \geq \kappa + (N_{a_t}(t-1)+1)\|x_{a_t}\|_2^2 \,.$$

Proof. For any arm $a \in [K]$, let us denote for any $b \in [K]$, $b \neq a$,

$$\tilde{V}_b := \frac{\kappa}{K} I_d + N_b(t-1) x_b x_b^\top \text{ and } \tilde{V}_a := \frac{\kappa}{K} I_d + (N_a(t-1)+1) x_a x_a^\top \,,$$

such that $\sum_{b \neq a} \tilde{V}_b + \tilde{V}_a = V_a$. $\tilde{V}_b$, for any $b \neq a$, and $\tilde{V}_a$ are all real, symmetric, positive definite square matrices (if $\kappa > 0$ or if there is an initialization phase with uniform samplings). We will use these simple lemma to analyze the eigenspace of $xx^\top$, for any $x \in \mathbb{R}^d$

Lemma 12.2.3. *For any matrix A and for any constant $\alpha \in \mathbb{R}$, if ζ is an eigenvalue of A, then $\alpha\zeta$ is an eigenvalue of αA, and $\eta + \alpha$ is an eigenvalue of $A + \alpha I_d$.*

Proof. For any eigenvector v associated with eigenvalue ζ of A,

$$(\alpha A)v = (\alpha\zeta)v \text{ and } (A + \alpha I_d)v = \zeta v + \alpha v = (\zeta + \alpha)v \,.$$

□

We will now analyze the eigenspace of $xx^\top$, where $x \neq 0$. For any eigenvalue ζ of $xx^\top$ associated with eigenvector v, $\zeta v = (xx^\top)v = x(x^\top v) = (x^\top v)x$, which means that

- any vector in Span$(x)^\perp := \{v : x^\top v = 0\}$ (of dimension $d-1$) is an eigenvalue of $xx^\top$ associated with eigenvalue 0 (which then has a multiplicity of $d-1$).

- any nonzero eigenvalue ζ (necessarily of multiplicity 1) satisfies $v = \frac{x^\top v}{\zeta} x \in$ Span$(x) := \{\alpha x : \alpha\mathbb{R}\}$. Moreover, we can show that $(xx^\top)x = x(x^\top x) = \|x\|_2^2 x$, which means that this eigenvalue is $\|x\|_2^2$.

Finally, we recall Weyl's matrix inequality (Weyl, 1912)

Lemma 12.2.4. ***Weyl's matrix inequality.*** *If M and P are hermitian matrices of dimension $d \times d$,* [1] *and $\lambda_k(A)$ denotes the kth largest (with multiplicity) eigenvalue of any hermitian matrix A, then*

$$\max_{\substack{i,j \leq d \\ i+j=k+d}} \lambda_i(M) + \lambda_j(P) \leq \lambda_k(M+P) \leq \min_{\substack{i,j \leq d \\ i+j=1+k}} \lambda_i(M) + \lambda_j(P) \,.$$

[1] For real matrices, it means symmetric.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

By induction, if we consider K hermitian matrices P_a, $a \in [K]$, then

$$\forall a' \in [K],\ \lambda_k\left(\sum_{a\in[K]} P_a\right) \geq \lambda_k(P_{a'}) + \sum_{a\neq a'} \min_{j\leq d} \lambda_j(P_a)\ .$$

Now, let us apply this to eigenvalue $\lambda_{b_t,c_t,a}$. Consider the permutation $p_a \in \mathbb{S}_d$ such that the eigenvalues of V_a ordered by p_a are in the order of decreasing value: $\lambda_{p_a(1)} \geq \lambda_{p_a(2)} \geq \cdots \geq \lambda_{p_a(d)}$.

Let us denote $j'_a := \max_{\substack{j\leq d \\ v_{p_a(j)}^\top (x_{b_t}-x_{c_t})\neq 0}} j$ such that $\lambda_{b_t,c_t,a} = \lambda_{p_a(j'_a)}(V_a)$.

$$\begin{aligned}
\lambda_{b_t,c_t,a} &\geq \lambda_{p_a(j'_a)}(\tilde{V}_a) + \sum_{b\neq a} \min_{j\leq d} \lambda_j(\tilde{V}_b) \\
&= \lambda_{p_a(j'_a)}\left(\frac{\kappa}{K} I_d + (N_a(t-1)+1)x_a x_a^\top\right) + \sum_{b\neq a} \min_{j\leq d} \lambda_j\left(\frac{\kappa}{K} I_d + N_b(t-1)x_b x_b^\top\right) \\
&= \frac{\kappa}{K} + (N_a(t-1)+1)\lambda_{p_a(j'_a)}(x_a x_a^\top) + \sum_{b\neq a}\left(\frac{\kappa}{K} + N_b(t-1)\min_{j\leq d}\lambda_j(x_b x_b^\top)\right) \\
&= \frac{\kappa}{K} + (N_a(t-1)+1)\lambda_{p_a(j'_a)}(x_a x_a^\top) + \frac{\kappa}{K}(K-1) + \sum_{b\neq a} N_b(t-1)\times 0 \\
&= \kappa + (N_a(t-1)+1)\lambda_{p_a(j'_a)}(x_a x_a^\top)\ .
\end{aligned}$$

The second equality sign is due to Lemma 12.2.3 and the third one is due to the analysis of the eigenspace of $xx^\top$, where $x \neq 0$.

We know that $\lambda_{p_a(j'_a)}(x_a x_a^\top) \in \{0, \|x_a\|_2^2\}$ for any $a \in [K]$. Now all that is left is to prove that $\lambda_{p_a(j'_{a_t})}(x_{a_t} x_{a_t}^\top) = \|x_{a_t}\|_2^2$ where a_t is the arm selected by the greedy rule, which means that any eigenvector v associated with the largest eigenvalue λ_{max} of matrix V_{a_t} satisfies $v^\top(x_{c_t} - x_{b_t}) \neq 0$ and that, for any strictly smaller eigenvalue, associated eigenvector v satisfies $v^\top(x_{b_t} - x_{c_t}) = 0$ (such that, necessarily, $\lambda_{b_t,c_t,a_t} = \lambda_{\text{max}}$). This is the trickiest part of the proof, and is left unproven.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

□

Combining this last lemma with stopping rule $\tau^{\text{LUCB}} := \inf_{t>0}\{\mathcal{B}_{c_t,b_t}(t) < \varepsilon\}$, and the upper bound on $\mathcal{B}_{c_t,b_t}(t)$ for Algorithm N-LinGapE, on event $\mathcal{E}^{\text{GIFA}}$ at $t < \tau^{\text{LUCB}}$:

$$\begin{aligned}
\varepsilon &\leq \mathcal{B}_{c_t,b_t}(t) \leq \min(-\max(\Delta_{b_t}, \Delta_{c_t}) + 3W_t(b_t, c_t), W_t(b_t, c_t)) \\
\Leftrightarrow \max\left(\epsilon, \frac{\Delta_{b_t}+\epsilon}{3}, \frac{\Delta_{c_t}+\epsilon}{3}\right) &\leq W_t(b_t, c_t) = \sigma\mathcal{T}_\delta(t)||x_{c_t} - x_{b_t}||_{\left(\hat{V}^\kappa(t-1)+x_{a_t}x_{a_t}^\top\right)^{-1}} \\
&\leq \sigma\mathcal{T}_\delta(t)\frac{\|x_{b_t} - x_{c_t}\|_2}{\sqrt{\kappa + N_{a_t}(t)\|x_{a_t}\|_2^2}} \text{ (a_t pulled at t)} \\
\Leftrightarrow N_{a_t}(t) &\leq \sigma^2\mathcal{T}_\delta(t)^2\frac{\|x_{b_t} - x_{c_t}\|_2^2}{\|x_{a_t}\|_2^2}\max\left(\varepsilon, \frac{\Delta_{b_t}+\varepsilon}{3}, \frac{\Delta_{c_t}+\varepsilon}{3}\right)^{-2} .
\end{aligned}$$

Applying Lemma 5.3.8 allows us to conclude on the conjecture similarly to what was previously done.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 13

Chapter 8 : Proofs for CPE

13.1 Technical lemmas

Lemma 13.1.1. ***Comparison of values of Problem $\widetilde{\mathcal{P}}$ from different gaps.*** *Consider Δ , $\Delta' \in (\mathbb{R}^+)^{K\times M}$, $\tau \in \widetilde{\mathcal{P}}(\Delta)$ and $\tau' \in \widetilde{\mathcal{P}}(\Delta')$. Then* ***(i).*** *If there is a constant $\alpha > 0$ so that $\forall k \in [K]\ \forall m \in [M],\ \alpha\Delta_{k,m} \leq \Delta'_{k,m}$ then*

$$\sum_{k,m} \tau'_{k,m} \leq \frac{1}{\alpha^2} \sum_{k,m} \tau_{k,m} \ .$$

(ii). *If there is a constant $\beta > 0$ so that $\forall k \in [K]\ \forall m \in [M],\ \Delta'_{k,m} \leq \beta\Delta_{k,m}$ then*

$$\frac{1}{\beta^2} \sum_{k,m} \tau_{k,m} \leq \sum_{k,m} \tau'_{k,m} \ .$$

Proof. The proof follows from the fact that τ and τ' are minimal. In particular, to prove **(ii)**, let $\tau''_{k,m} = \beta^2\tau'_{k,m}$ for any $k \in [K]$, $m \in [M]$. Then, for any agent m and arm k,

$$\sum_{n\in[M]} \frac{w^2_{n,m}}{\tau''_{k,n}} = \sum_{n\in[M]} \frac{w^2_{n,m}}{\beta^2\tau'_{k,n}} \leq \frac{1}{2}\left(\frac{\Delta'_{k,m}}{\beta}\right)^2 \leq \frac{\Delta^2_{k,m}}{2} \ .$$

By minimality of τ ,

$$\sum_{m\in[M]} \tau_{k,m} \leq \sum_{m\in[M]} \tau''_{k,m} = \beta^2 \sum_{n\in[M]} \tau'_{k,m} \ .$$

(i) similarly follows. □

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Lemma 13.1.2. ***Choice of threshold function*** $\mathcal{T}_\delta(\cdot)$***.*** *For any* $N \in (\mathbb{N}^+)^M$

$$\mathcal{T}_\delta(N) := 2\left(g_M\left(\frac{\delta}{KM}\right) + 2\sum_{m\in[M]} \ln\left(4 + \ln(N_m)\right)\right) ,$$

where g_M *is a function that satisfies* $g_M(\delta) \approx \log(1/\delta) + M \log\log(1/\delta)$*. Then the good event*

$$\mathcal{E}^{\text{CPE}} := \left\{\forall r \geq 0\ \forall m \in [M]\ \forall k \in [K],\ \left|\mu'_{(k,m)} - \widehat{\mu'_{(k,m)}}(r)\right| \leq \Omega_{(k,m)}(r)\right\}$$

holds with probability larger than $1-\delta$*.*

Proof. We prove that, if we define the threshold function for $N \in (\mathbb{N}^*)^M$ as

$$\mathcal{T}_\delta(N) := 2\left(g_M\left(\frac{\delta}{KM}\right) + 2\sum_{m=1}^{M} \ln(4 + \ln(N_m))\right) ,$$

where

- $\forall \delta \in (0,1), g_M(\delta) := M\mathcal{C}^{g_G}\left(\log(1/\delta)/M\right)$,
- $\forall x > 0, \mathcal{C}^{g_G}(x) := \min_{\lambda\in(0.5,1)}(g_G(\lambda) + x)/\lambda$,
- $\forall \lambda \in (0.5,1), g_G(\lambda) := 2\lambda - 2\lambda\log(4\lambda) + \log(\zeta(2\lambda)) - 0.5\log(1-\lambda)$,

and ζ is the Riemann zeta function, then, the good event

$$\mathcal{E}^{\text{CPE}} := \left\{\forall r \in \mathbb{N}\ \forall m \in [M]\ \forall k \in [K],\ \left|\widehat{\mu'_{(k,m)}}(r) - \Delta'_{k,m}\right| \leq \Omega_{(k,m)}(r)\right\} .$$

holds with probability larger than $1-\delta$. Using Kaufmann and Koolen (2021, Proposition 24) on $\mu'_{(k,m)}$, for any arm k and agent m, directly yields

$$\mathbb{P}\left(\exists r \geq 0, \left|\mu'_{(k,m)}r - \Delta'_{k,m}\right| > \sqrt{\mathcal{T}_\delta(n_{k,\cdot}(r))\sum_n \frac{w_{n,m}^2}{n_{k,n}(r)}}\right) \leq \frac{\delta}{KM}$$

(using the notation of the paper, consider $\mu = \mu_{(k,\cdot)}$ and $c = W_{\cdot,m}$). Then all that is needed to conclude is to apply a union bound on $[K]\times[M]$

$$\mathbb{P}\left(\mathcal{E}^c\right) = \mathbb{P}\left(\exists m \in [M]\ \exists k \in [K]\ \exists r \geq 0,\ \left|\mu'_{(k,m)}r - \Delta'_{k,m}\right| > \sqrt{2\mathcal{T}_\delta(n_{k,\cdot}(r))\sum_n \frac{w_{n,m}^2}{n_{k,n}(r)}}\right)$$
$$\leq \sum_{m\in[M]}\sum_{k\in[K]} \frac{\delta}{KM} \leq \delta .$$

□

Remark 13.1.3. *Although the expression of* g_M *is not in closed-form, its value can easily be retrieved through any scalar minimization procedure.*

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

13.2 Correctness analysis

Theorem 13.2.1. ***CPE is δ-correct.*** *On event $\mathcal{E}^{\text{CPE}}$, CPE outputs the correct set of optimal arms $\mathcal{S}_N^m$ for each agent m.*

Proof. If CPE was not δ-correct on event $\mathcal{E}^{\text{CPE}}$, then, for some agent m, there would be an arm $\ell \in \mathcal{S}_N^m$ which is eliminated at round r from $B_m(r+1)$. But, on event $\mathcal{E}^{\text{CPE}}$, Lemma 8.4.5 implies that, for any $r \geq 0$, $m \in [M]$, and $(i,j) \in [K]^2$,

$$\widehat{\mu'_{(i,m)}}(r) - \widehat{\mu'_{(j,m)}}(r) + \Omega_{(i,m)}(r) + \Omega_{(j,m)}(r) \geq \Delta'^m_{i,j} \geq \widehat{\mu'_{(i,m)}}(r) - \widehat{\mu'_{(j,m)}}(r) - \Omega_{(i,m)}(r) - \Omega_{(j,m)}(r) \ .$$

Then, combining the right-hand inequality for $j = \ell$ with the elimination criterion at Line 17 in Algorithm 8

$$\begin{aligned} \max_{i \in [K]}^{N} \Delta'^m_{i,\ell} &\geq \max_{i \in [K]}^{N} \left(\widehat{\mu'_{(i,m)}}(r) - \widehat{\mu'_{(\ell,m)}}(r) - \Omega_{(i,m)}(r) - \Omega_{(\ell,m)}(r) \right) \\ &\geq \max_{i \in B_m(r) \subseteq [K]}^{N} \left(\widehat{\mu'_{(i,m)}}(r) - \widehat{\mu'_{(\ell,m)}}(r) - \Omega_{(i,m)}(r) - \Omega_{(\ell,m)}(r) \right) > 0 \ , \end{aligned}$$

which is absurd because $\ell \in \mathcal{S}_N^m$. □

13.3 Sample complexity analysis

Theorem 13.3.1. ***Upper bound on Exp$_\mu$(CPE).*** *On any model μ, with probability $1-\delta$, CPE (Algorithm 8) outputs the set of N best arms for each agent with an exploration cost at most*

$$32\mathcal{N}^\star(\mu) \log_2 \left(\frac{8}{\Delta'_{\text{min}}} \right) \log\left(1/\delta\right) + o\left(1/\delta\right) \ ,$$

and at most $\left\lceil \log_2 \left(\frac{8}{\Delta'_{\text{min}}} \right) \right\rceil$ communication rounds, where $\Delta'_{\text{min}} := \min_{m \in [M]} \min_{k \in [K]} \Delta'_{k,m}$.

Proof. Thanks to Theorem 8.4.6, CPE is shown to be δ-correct on event $\mathcal{E}^{\text{CPE}}$, of probability greater than $1-\delta$ (Lemma 8.4.5). Then we upper bound the exploration cost when $\mathcal{E}^{\text{CPE}}$ holds. We denote by R the (random) number of rounds used by the algorithm, and, for all $m \in [M]$ and $k \notin \mathcal{S}_N^m$, by $R_{k,m}$ the (random) last round in which k is still a candidate arm for player m

$$R_{k,m} := \sup\left\{ r \geq 0 \mid k \in B_m(r) \right\} \ .$$

By definition of Algorithm 8, $R = \max_{m \in [M]} \max_{k \notin \mathcal{S}_m^\star N} R_{k,m}$. We first provide upper bounds on $R_{k,m}$ and R. To achieve this, we introduce the following notation

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

for any arm $k \in [K]$ and agent $m \in [M]$

$$r_{k,m} \;:=\; \min\left\{r \geq 0 \mid 4 \times 2^{-r} < \Delta'_{k,m}\right\} \text{ and } r_{\max} := \max_{m\in[M]} \max_{k\notin\mathcal{S}_N^m} r_{k,m} \;.$$

From the definitions of $r_{k,m}$ and $r_{\max}$, the following upper bounds can be easily checked.

Lemma 13.3.2. *For any arm $k \in [K]$ and agent $m \in [M]$, $r_{k,m} \leq \log_2\left(8/\Delta'_{k,m}\right)$ and $r_{\max} \leq \log_2\left(8/\Delta'_{\min}\right)$, where $\Delta'_{\min} := \min_{m\in[K]} \min_{k\in[M]} \Delta'_{k,m}$.*

Using the fact that CPE only halves the gap proxies of arms that are not eliminated, we can write down the value of the gap proxies for these arms

Lemma 13.3.3. $\forall m \in [M]\ \forall k \in B_m(r),\ \widetilde{\Delta'_{k,m}}(r) = 2^{-r}$.

Using the important relationship between gap proxies and the confidence width as established in Lemma 8.4.7, we can further show that

Lemma 13.3.4. *On $\mathcal{E}^{\text{CPE}}$, for any $m \in [M]$ and any $k \notin \mathcal{S}_N^m$, $R_{k,m} \leq r_{k,m}$.*

Proof. Assume $\mathcal{E}^{\text{CPE}}$ holds. For any suboptimal arm k for agent m, at round $r = r_{k,m}$, if $k \notin B_m(r)$, then trivially $R_{k,m} < r_{k,m}$. Otherwise, if $k \in B_m(r)$, then

$$\begin{aligned}
\widehat{\mu'_{(k,m)}}(r) + \Omega_{(k,m)}(r) &\leq_{(1)} \mu'_{(k,m)} + 2\Omega_{(k,m)}(r) \\
&\leq_{(2)} \mu'_{(k,m)} + 2\widetilde{\Delta'_{k,m}}(r) = \mu'_{(k,m)} + 4\widetilde{\Delta'_{k,m}}(r) - 2\widetilde{\Delta'_{k,m}}(r) \\
&<_{(3)} \max_{i\in[K]}^{N} \mu'_{(i,m)} - 2\widetilde{\Delta'_{k,m}}(r) =_{(4)} \max_{i\in B_m(r)}^{N} \mu'_{(i,m)} - 2\widetilde{\Delta'_{k,m}}(r) \\
&\leq_{(1)} \max_{i\in B_m(r)}^{N} \left(\widehat{\mu'_{(i,m)}}(r) - \Omega_{(i,m)}(r) + 2\Omega_{(i,m)}(r)\right) - 2\widetilde{\Delta'_{k,m}}(r) \\
&\leq_{(2)} \max_{i\in B_m(r)}^{N} \left(\widehat{\mu'_{(i,m)}}(r) - \Omega_{(i,m)}(r) + 2\widetilde{\Delta'_{i,m}}(r)\right) - 2\widetilde{\Delta'_{k,m}}(r) \\
&=_{(5)} \max_{i\in B_m(r)}^{N} \left(\widehat{\mu'_{(i,m)}}(r) - \Omega_{(i,m)}(r)\right) + 2 \times 2^{-r} - 2 \times 2^{-r} \;, \\
\implies \widehat{\mu'_{(k,m)}}(r) + \Omega_{(k,m)}(r) &< \max_{i\in B_m(r)}^{N} \left(\widehat{\mu'_{(i,m)}}(r) - \Omega_{(i,m)}(r)\right) \;,
\end{aligned}$$

where (1) is because event $\mathcal{E}^{\text{CPE}}$ holds ; (2) uses Lemma 8.4.7 ; (3) uses $r = r_{k,m}$ and $k \notin \mathcal{S}_N^m$; (4) uses event $\mathcal{E}^{\text{CPE}}$ and Theorem 8.4.6, that implies that for any $\ell \in \mathcal{S}_N^m$, $\ell \in B_m(r)$; (5) holds because of Lemma 13.3.3. It follows that $k \notin B_m(r_{k,m} + 1)$ and $R_{k,m} \leq r_{k,m}$.

□

The previous lemma straightforwardly implies that

Corollary 13.3.4.1. $R \leq r_{\max} \leq \log_2(8/\Delta'_{\min})$.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Moreover, it also permits to prove that, in the last round R, the gap proxies are lower bounded by the true characteristic gaps.

Corollary 13.3.4.2. *At final round R, and for any agent m and suboptimal arm $k \notin \mathcal{S}_N^m$, if $\mathcal{E}^{CPE}$ holds,*

$$\widetilde{\Delta'_{k,m}}(R) \geq \frac{1}{8}\Delta'_{k,m} \ .$$

Proof. If $R < r_{k,m}$, by definition of $r_{k,m}$, $\widetilde{\Delta'_{k,m}}(R) \geq (1/4)\Delta'_{k,m} \geq (1/8)\Delta'_{k,m}$. If $R \geq r_{k,m}$, we first observe that $\widetilde{\Delta'_{k,m}}(R) = \widetilde{\Delta'_{k,m}}(R_{k,m}) = (1/2)\widetilde{\Delta'_{k,m}}(R_{k,m}-1)$ by definition of the algorithm (the gaps remain frozen when an arm is eliminated, and they are halved otherwise). As $R_{k,m} - 1 < r_{k,m}$ by Lemma 13.3.4, by definition of $r_{k,m}$, it follows that

$$4\widetilde{\Delta'_{k,m}}(R_{k,m}-1) > \Delta'_{k,m}$$

and we conclude that $\widetilde{\Delta'_{k,m}}(R) \geq (1/8)\Delta'_{k,m}$. □

Now, for any $m \in [M]$ and $k \in [K]$, using Corollary 13.3.4.2, and the fact that the gap proxies are nonincreasing between two consecutive phases, we get

$$\forall k \in [K]\ \forall m \in [M]\ \forall r \leq R,\ \widetilde{\Delta'_{k,m}}(r) \geq \widetilde{\Delta'_{k,m}}(R) \geq \Delta'_{k,m}/8 \ .$$

Using Lemma 8.4.3 and once again the fact that gap proxies are nonincreasing, for any round $r \leq R$, the optimal allocation $t(r) \in \widetilde{\mathcal{P}}\left(\sqrt{2}\widetilde{\Delta}\right)$ satisfies

$$\sum_{k,m} t_{k,m}(r) \leq 32 \sum_{k,m} t'_{k,m} \ ,$$

where $t' \in \widetilde{\mathcal{P}}(\Delta')$, hence

$$\max_{r \leq R} \left[\sum_{k,m} t_{k,m}(r)\right] \leq 32\widetilde{\mathcal{N}} \ . \tag{13.1}$$

For every arm $k \in [K]$ and agent $m \in [M]$, we now introduce

$$r'_{k,m} := \sup\{r \leq R : d_{k,m}(r) \neq 0\} \ ,$$

so that $n_{k,m}(R) = n_{k,m}(r'_{k,m})$. Using Lemma 13.3.5 stated below, and the fact that threshold function $\mathcal{T}_\delta(\cdot)$ is nondecreasing in each coefficient of its argument (see its definition in Lemma 8.4.5),

$$\begin{aligned} n_{k,m}(R) = n_{k,m}(r'_{k,m}) &\leq t_{k,m}(r'_{k,m})\mathcal{T}_\delta(n_{k,\cdot}(r'_{k,m})) + 1 \\ &\leq t_{k,m}(r'_{k,m})\mathcal{T}_\delta(n_{k,\cdot}(R)) + 1 \ . \end{aligned}$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Lemma 13.3.5. *For any $k, m, r \geq 0$, either $d_{k,m}(r) = 0$, or*

$$n_{k,m}(r) = n_{k,m}(r-1) + d_{k,m}(r) < t_{k,m}(r)\mathcal{T}_\delta(n_{k,\cdot}(r)) + 1 \ .$$

Proof. At fixed $r \geq 0$, for any set $S \subseteq [K] \times [M]$, let us prove by induction on $|S| \geq 1$ [1]

$$\begin{aligned} \forall k \in [K]\ \forall m \in [M], \qquad & d'_{k,m}(r) := (d_{k,m}(r) - \mathbb{1}_S((k,m)))_+ \\ \implies \quad & \forall (k,m) \in S, \frac{n_{k,m}(r-1) + d'_{k,m}(r)}{\mathcal{T}_\delta(n_{k,\cdot}(r-1) + d_{k,\cdot}(r))} < t_{k,m}(r) \\ & \text{or } d_{k,m}(r) = 0 \ . \end{aligned}$$

At $|S| = 1$: Let us denote $S = \{(k', m')\}$. If $d_{k',m'}(r) = 0$, then it is trivial. Otherwise, $\sum_{k,m} d'_{k,m}(r) < \sum_{k,m} d_{k,m}(r)$, and then, by minimality of solution $d(r)$, at least one constraint from the optimization problem of value $\sum_{k,m} d_{k,m}(r)$ has to be violated. For any $(k,m) \notin S$, by definition of $d(r)$ and nondecreasingness of $\mathcal{T}_\delta(\cdot)$

$$\begin{aligned} n_{k,m}(r-1) + d'_{k,m}(r) &= n_{k,m}(r-1) + d_{k,m}(r) \\ &\geq t_{k,m}(r)\mathcal{T}_\delta(n_{k,\cdot}(r-1) + d_{k,\cdot}(r)) \\ &\geq t_{k,m}(r)\mathcal{T}_\delta(n_{k,\cdot}(r-1) + d'_{k,\cdot}(r)) \ . \end{aligned}$$

That means, necessarily the only constraint that is violated is the one on (k', m'). Using the nondecreasingness of $\mathcal{T}_\delta(\cdot)$

$$\begin{aligned} n_{k',m'}(r-1) + d_{k',m'}(r) - 1 &= n_{k',m'}(r-1) + d'_{k',m'}(r) \\ &< t_{k',m'}(r)\mathcal{T}_\delta(n_{k',\cdot}(r-1) + d'_{k,\cdot}(r)) \\ &\leq t_{k',m'}(r)\mathcal{T}_\delta(n_{k',\cdot}(r-1) + d_{k,\cdot}(r)) \ . \end{aligned}$$

Combining the two ends of the inequality proves the claim.

At $|S| > 1$: At fixed $(k', m') \in S$, we can apply the claim to $S \setminus \{(k', m')\}$. Moreover, if $d_{k',m'}(r) = 0$, then the claim is proven. Otherwise, towards contradiction

$$n_{k',m'}(r-1) + d'_{k',m'}(r) \geq t_{k',m'}(r)\mathcal{T}_\delta(n_{k',\cdot}(r-1) + d_{k,\cdot}(r)) \ .$$

Let us then consider the following allocation

$$\forall k \in [K]\ \forall m \in [M],\ d''_{k,m}(r) := d_{k,m}(r) - \mathbb{1}_{\{(k',m')\}}((k,m)) \ .$$

It can be checked straightforwardly – using the nondecreasingness of $\mathcal{T}_\delta()$ –

[1] For any $x \in \mathbb{N}^M$, $(x)_+ := (\max(0, x_m))_{m \in [M]}$, and $\mathbb{1}_S$ is the indicator function of set S.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

that d'' satisfies all required constraints for any pair $(k,m) \in [K] \times [M]$, and that $\sum_{k,m} d''_{k,m}(r) = \sum_{k,m} d_{k,m}(r) - 1$, which, by minimality of d, is absurd. Then the claim is proven for $|S| > 1$. Then Lemma 13.3.5 is proven by considering $S = [K] \times [M]$. $\square$

By summing the upper bound on $(n_{k,m}(R))_{k,m}$ over $[K] \times [M]$, we can upper bound the exploration cost τ as

$$\begin{aligned}
\tau := \sum_{k,m} n_{k,m}(R) &\leq \sum_{k,m} t_{k,m}(r'_{k,m}) \mathcal{T}_\delta(n_{k,\cdot}(R)) + KM \\
&\leq \sum_{k,m} t_{k,m}(r'_{k,m}) \beta^*(\tau) + KM \\
&\leq \sum_{k,m} \sum_{r \leq R} t_{k,m}(r) \beta^*(\tau) + KM \\
&\leq R \max_{r \leq R} \left[\sum_{k,m} t_{k,m}(r) \right] \beta^*(\tau) + KM \\
&\leq \log_2 (8/\Delta'_{\min}) \max_{r \leq R} \left[\sum_{k,m} t_{k,m}(r) \right] \beta^*(\tau) + KM ,
\end{aligned}$$

where we use Corollary 13.3.4.1 and introduce the quantity

$$\beta^*(\tau) := \mathcal{T}_\delta(\tau \mathbb{1}_M) = 2 \left(g_M \left(\frac{\delta}{KM} \right) + 2M \ln (4 + \ln (\tau)) \right) \text{ where } \forall n \in [M], \mathbb{1}_M(n) = 1 .$$

Using Equation (13.1) and Lemma 8.4.1,

$$\tau \leq 32 \widetilde{\mathcal{N}} \log_2 (8/\Delta'_{\min}) \beta^*(\tau) + KM \leq 32 \mathcal{N}^\star \log_2 (8/\Delta'_{\min}) \beta^*(\mu) + KM .$$

Therefore, τ is bounded from above by

$$\sup \{ n \in \mathbb{N}^* : n \leq 32 \mathcal{N}^\star \log_2 (8/\Delta'_{\min}) \beta^*(n) + KM \} .$$

Applying Kaufmann, Ménard, et al. (2021, Lemma 15) with

$$\begin{aligned}
\Delta &= \left(\sqrt{32 \mathcal{N}^\star \log_2 (8/\Delta'_{\min})} \right)^{-1} , \\
a &= KM + 2 g_M \left(\frac{\delta}{KM} \right) , \\
b &= 4M , \\
c &= 4 , \\
d &= e^{-1} \text{ using } \forall n, \log(n) \leq n e^{-1} ,
\end{aligned}$$

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

yields an explicit upper bound on τ

$$\widehat{T}(\mu) := 32\mathcal{N}^\star \log_2\left(8/\Delta'_{\min}\right)\left[KM + 2g_M\left(\frac{\delta}{KM}\right)\right.$$
$$\left.+4M\ln\left(4 + 1,024\frac{(\mathcal{N}^\star \log_2\left(8/\Delta'_{\min}\right))^2}{e}\left(KM + 2g_M\left(\frac{\delta}{KM}\right) + 4M(2+\sqrt{e})\right)^2\right)\right],$$

which satisfies $\tau\widehat{T}(\mu) \leq a + b\ln(c + d\tau\widehat{T}(\mu))$. Using that $g_M(x) \simeq x + M\log\log(x)$ in the regime of small values of δ, we obtain that

$$\widehat{T}(\mu) = 32\mathcal{N}^\star \log_2\left(8/\Delta'_{\min}\right)\log(1/\delta) + o_{\delta\to 0}\left(\log(\delta^{-1})\right).$$

The upper bound on the communication cost follows from the upper bound on the number of phases given in Corollary 13.3.4.1. □

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Chapter 14

Chapters 5 & 6: Real-life conditions for drug repurposing

One shortcoming of the experiments in Chapters 5 and 6 is that the setting is actually made closer to the studied frameworks –respectively linear and misspecified linear structures, by transforming the drug features in a supervised fashion, Gaussian rewards in Chapter 6 instead of raw patient-specific scores. In this chapter, we actually run the algorithms developed in this thesis (N-LinGapE, LinGIFA, MisLid, in their default versions) and the benchmark (unstructured) algorithm LUCB (Kalyanakrishnan et al., 2012) in a setting closer to the drug repurposing application.

We consider again the set of $K = 21$ drugs which associated average scores are displayed in Table 10.3. A boxplot of these reward values is reported in the left-hand plot in Figure 14.1. The gap between the lowest average reward among antiepileptics and the highest average reward across proconvulsant drugs is around 0.07, which means that the associated instance (for $N = 10$) is moderately hard to solve.

We want to determine the $N = 3$ best treatments ($\Delta_{a_{(N)}a_{(N+1)}} \approx 0.14$), based on $d = 10$ features. [1] In order to compute these features from the original 194 ones, we apply Principal Component Decomposition (PCA) and select the d largest components. The *a posteriori* heatmap built on these PCA-reduced features, according to the true reported class (antiepileptic or proconvulsant) is displayed in Figure 14.1 (right-hand plot). This heatmap shows that using these features allows to cluster very roughly the two class of drugs (ARI $\approx 0.080 > 0$), which tends to show that the underlying model is far from linear (with a full dependence on the features to infer the scores). Note that this feature transformation is reward-agnostic.

[1] The choice of the value of d is guided by computational considerations, whereas $N = 3$ is based on the choice made in Chapter 5.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

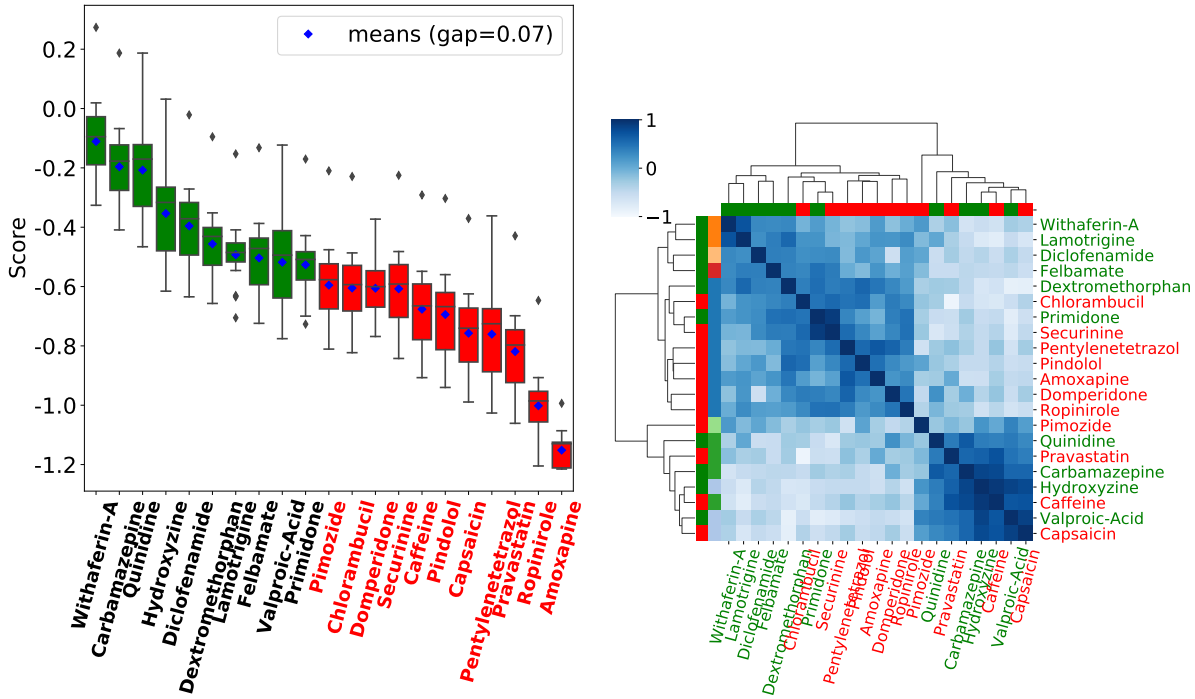

Figure 14.1: **Visualization of the "real life" repurposing instance. Left plot :** Boxplot of rewards, by increasing order of average scores. **Right plot :** Pearson's r correlation heatmap based on the $d = 10$ PCA-reducred features, ranked by the order obtained through hierarchical clustering. The first row of colors (red/green) corresponds to the class (resp., proconvulsant drug/antiepileptic), whereas the second row of colors (7 different colors) corresponds to the clusters found through hierarchical clustering.

Similarly to the experimental setting in Chapter 5, each time a treatment is selected, a patient is selected at random, and the score associated with this patient and the selected treatment is observed.

When using MisLid, we need to determine the value to assign to parameter Ψ (please refer to Chapter 6). In practice, the repurposing scores (see Table 10.3) are roughly comprised between -1 and 1. In order to ensure that the property of correctness is satisfied, we use the reasoning laid out in the discussion of Chapter 6 ; we start by considering $\Psi = 1$, and decreasing its value by a fixed step size ($\texttt{size} = 0.1$) until the sample complexity is equal or lower than the one incurred by an unstructured algorithm on the same instance (*e.g.*, LUCB). Of course, such a procedure is (1) not theoretically supported, (2) not tractable in practice because it would imply running first an unstructured instance. As previously mentioned in the thesis, an interesting subsequent work would be the design of an adequate finetuning method for Ψ.

The parameter values are reported in Table 14.1, whereas the final error frequencies and sample complexities are reported in Table 14.2.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

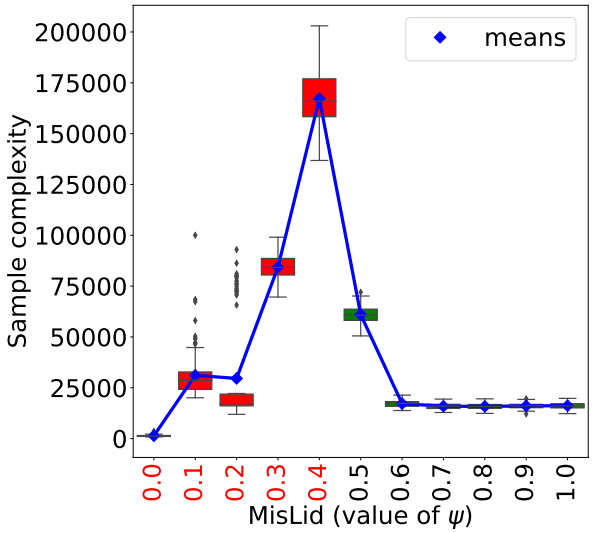

Figure 14.2: **Results for the drug repurposing in epilepsy.** Boxplots of the sample complexity for MisLid for $\Psi \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ for the "real life" drug repurposing instance. Green (resp., red) boxplots correspond to (resp., non) δ-correct algorithms.

Parameter	Value
K	21
d	10
σ	1^*
δ	10%
ε	0.14
N	3
#iterations	100

Table 14.1: **Parameters values for the drug repurposing in epilepsy.** Parameter values for the "real-life" drug repurposing setting. * Although this parameter is not related to the actual observations, it is used to build the confidence intervals in the algorithms.

MisLid was also run with a few lower values of Ψ in order to illustrate the effect of Ψ on the performance of the algorithm. The trend in sample complexity depending on Ψ is illustrated in the boxplot in Figure 14.2.

The comments made in the experimental part of Chapter 6 are still valid in this "real life" setting : linear bandit algorithms (including MisLid with $\Psi = 0$) do not preserve correctness anymore, whereas unstructured algorithms are not sample-efficient. In practice, as illustrated by Figure 14.2, selecting values around a educated guess about the absolute value of the upper bound on rewards seems rather robust and sample-efficient, while ensuring that the correctness property is preserved.

Considering MisLid and Figure 14.2, while the error rate drastically increases as the misspecification level ψ in input decreases (as described in the conclusion of Chapter 6), the trend of the sample complexity depending on ψ displays a different, non-monotonic pattern. In Table 14.2, the average sample complexity globally increases in the range $\psi \in [0, 0.4]$, globally decreases in the range $\psi \in [0.5, 0.8]$, and then slowly increases again starting from $\psi = 0.8$.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Alg.	$\hat{\delta}$	$\hat{s}$
LUCB	0%	$37,425 \pm 271$
MisLid ($\Psi = 1$)	0%	$16,189 \pm 1,492$
MisLid ($\Psi = 0.9$)	0%	$16,070 \pm 1,287$
MisLid ($\Psi = 0.8$)	0%	$15,846 \pm 1,371$
MisLid ($\Psi = 0.7$)	0%	$15,942 \pm 1,311$
MisLid ($\Psi = 0.6$)	0%	$17,205 \pm 1,610$
MisLid ($\Psi = 0.5$)	0%	$60,992 \pm 4,433$
MisLid ($\Psi = 0.4$)	100%	$167,233 \pm 13,922$
MisLid ($\Psi = 0.3$)	100%	$84,337 \pm 5,966$
MisLid ($\Psi = 0.2$)	100%	$29,586 \pm 24,571$
MisLid ($\Psi = 0.1$)	100%	$31,144 \pm 11,606$
MisLid ($\Psi = 0$)	100%	$1,393 \pm 491$
N-LinGapE	100%	157 ± 13
LinGIFA	100%	91 ± 14

Table 14.2: **Results for the drug repurposing in epilepsy.** Results (in terms of empirical error frequency and sample complexity) for the "real-life" drug repurposing example. $\hat{\delta}$ is the empirical error frequency across 100 iterations, $\hat{s}$ is the average sample complexity across iterations ($\pm$ the standard deviation), rounded up to the closest integer.

The last "increasing" section of the plot is consistent with the conclusion in Chapter 6, which predicts that, the greater ψ is, the largest the set of models $\mathcal{M}_\psi$ which the algorithm focuses on is, and gets closer to the set of all models (*i.e.*, unstructured models). Then, the expected sample complexity of MisLid should be closer and closer to the expected sample complexity of a good unstructured bandit algorithm (*e.g.*, LUCB). However, the observed behaviour of sample complexity for smaller values of ψ does not follow a linear pattern, and its study would be an interesting subsequent theoretical work.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Bibliography

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). "Improved algorithms for linear stochastic bandits". *Advances in neural information processing systems*, 24 (cited on p. 106).

Abeille, M. and Lazaric, A. (2017). "Linear thompson sampling revisited". In: *Artificial Intelligence and Statistics*. PMLR, pp. 176–184 (cited on p. 66).

Adomavicius, G. and Tuzhilin, A. (2005). "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions". *IEEE transactions on knowledge and data engineering*, 17(6), pp. 734–749 (cited on p. 63).

Aghamiri, S. S., Singh, V., Naldi, A., Helikar, T., Soliman, S., and Niarakis, A. (2020). "Automated inference of Boolean models from molecular interaction maps using CaSQ". *Bioinformatics*, 36(16), pp. 4473–4482 (cited on p. 19).

Agrawal and Goyal, N. (2013). "Thompson sampling for contextual bandits with linear payoffs". In: *International conference on machine learning*. PMLR, pp. 127–135 (cited on p. 66).

Agrawal, Verschueren, R., Diamond, S., and Boyd, S. (2018). "A rewriting system for convex optimization problems". *Journal of Control and Decision*, 5(1), pp. 42–60 (cited on p. 142).

Ahmed, S. S., Roy, S., and Kalita, J. (2018). "Assessing the effectiveness of causality inference methods for gene regulatory networks". *IEEE/ACM transactions on computational biology and bioinformatics*, 17(1), pp. 56–70 (cited on p. 12).

Aiba, H., Nakasai, F., Mizushima, S., and Mizuno, T. (1989). "Phosphorylation of a bacterial activator protein, OmpR, by a protein kinase, EnvZ, results in stimulation of its DNA-binding ability". *The Journal of Biochemistry*, 106(1), pp. 5–7 (cited on p. 13).

Alaimo, S., Giugno, R., and Pulvirenti, A. (2016). "Recommendation techniques for drug–target interaction prediction and drug repositioning". In:

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Data Mining Techniques for the Life Sciences. Springer, pp. 441–462 (cited on p. 4).

Aliper, A., Plis, S., Artemov, A., Ulloa, A., Mamoshina, P., and Zhavoronkov, A. (2016). "Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data". *Molecular pharmaceutics*, 13(7), pp. 2524–2530 (cited on p. 62).

Andronis, C., Sharma, A., Virvilis, V., Deftereos, S., and Persidis, A. (2011). "Literature mining, ontologies and information visualization for drug repurposing". *Briefings in bioinformatics*, 12(4), pp. 357–368 (cited on p. 4).

Appenzeller, S., Balling, R., Barisic, N., Baulac, S., Caglayan, H., Craiu, D., De Jonghe, P., Depienne, C., Dimova, P., Djémié, T., et al. (2014). "De novo mutations in synaptic transmission genes including DNM1 cause epileptic encephalopathies". *The American Journal of Human Genetics*, 95(4), pp. 360–370 (cited on pp. 40, 43).

Arthur, D. and Vassilvitskii, S. (2006). *k-means++: The advantages of careful seeding*. Tech. rep. Stanford (cited on pp. 53, 151).

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). "Gene ontology: tool for the unification of biology". *Nature genetics*, 25(1), pp. 25–29 (cited on pp. xxiv, 125).

Assenov, Y., Ramírez, F., Schelhorn, S.-E., Lengauer, T., and Albrecht, M. (2008). "Computing topological parameters of biological networks". *Bioinformatics*, 24(2), pp. 282–284 (cited on p. 30).

Auer, P. (2002). "Using confidence bounds for exploitation-exploration trade-offs". *Journal of Machine Learning Research*, 3(Nov), pp. 397–422 (cited on p. 76).

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). "Finite-time analysis of the multiarmed bandit problem". *Machine learning*, 47(2), pp. 235–256 (cited on p. 63).

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002). "The nonstochastic multiarmed bandit problem". *SIAM journal on computing*, 32(1), pp. 48–77 (cited on p. x).

Auer, P. and Ortner, R. (2010). "UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem". *Periodica Mathematica Hungarica*, 61(1-2), pp. 55–65 (cited on pp. 65, 135).

Awerbuch, B. and Kleinberg (2004). "Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches". In: *Proceedings of*

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

the thirty-sixth annual ACM symposium on Theory of computing, pp. 45–53 (cited on p. 106).

Babichev, S., Durnyak, B., Senkivskyy, V., Sorochynskyi, O., Kliap, M., and Khamula, O. (2019). "Technique of Gene Regulatory Networks Reconstruction Based on ARACNE Inference Algorithm." In: *IDDM*, pp. 195–207 (cited on pp. vii, 28, 165).

Baltimore, D. (1970). "Viral RNA-dependent DNA polymerase: RNA-dependent DNA polymerase in virions of RNA tumour viruses". *Nature*, 226(5252), pp. 1209–1211 (cited on p. 14).

Barati, S., Ragerdi Kashani, I., Moradi, F., Tahmasebi, F., Mehrabi, S., Barati, M., and Joghataei, M. T. (2019). "Mesenchymal stem cell mediated effects on microglial phenotype in cuprizone-induced demyelination model". *Journal of cellular biochemistry*, 120(8), pp. 13952–13964 (cited on p. 116).

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., et al. (2012). "NCBI GEO: archive for functional genomics data sets—update". *Nucleic acids research*, 41(D1), pp. D991–D995 (cited on pp. i, xxiv, 4).

Béal, J., Montagud, A., Traynard, P., Barillot, E., and Calzone, L. (2019). "Personalization of logical models with multi-omics data allows clinical stratification of patients". *Frontiers in physiology*, p. 1965 (cited on p. 161).

Benjamini, Y. and Hochberg, Y. (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing". *Journal of the Royal statistical society: series B (Methodological)*, 57(1), pp. 289–300 (cited on pp. xxiv, 40, 123, 125).

Benque, D., Bourton, S., Cockerton, C., Cook, B., Fisher, J., Ishtiaq, S., Piterman, N., Taylor, A., and Vardi, M. Y. (2012). "BMA: Visual tool for modeling and analyzing biological networks". In: *International Conference on Computer Aided Verification*. Springer, pp. 686–692 (cited on p. 21).

Bisgin, H., Liu, Z., Kelly, R., Fang, H., Xu, X., and Tong, W. (2012). "Investigating drug repositioning opportunities in FDA drug labels through topic modeling". In: *BMC bioinformatics*. Vol. 13. 15. Springer, pp. 1–9 (cited on p. 4).

Bloomingdale, P., Nguyen, V. A., Niu, J., and Mager, D. E. (2018). "Boolean network modeling in systems pharmacology". *Journal of pharmacokinetics and pharmacodynamics*, 45(1), pp. 159–180 (cited on pp. vi, 19, 27).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Bokobza, Réda, Guenoun, Faivre, Charpentier, L., Sautet, Schwendimann, Benchouaia, Dias, Lemoine, Fleiss, Steenwinckel, V., Delahaye-Duriez, and Gressens (*in prep.*). "Therapeutic evaluation of Hu-MSCs in a rat model of perinatal inflammation: a systematic outcome scoring" (cited on pp. iii, xviii, 7, 114, 115).

Bokobza, Van Steenwinckel, J., Mani, S., Mezger, V., Fleiss, B., and Gressens, P. (2019). "Neuroinflammation in preterm babies and autism spectrum disorders". *Pediatric Research*, 85(2), pp. 155–165 (cited on p. 116).

Bolouri, H. and Davidson, E. H. (2003). "Transcriptional regulatory cascades in development: initial rates, not steady state, determine network kinetics". *Proceedings of the National Academy of Sciences*, 100(16), pp. 9371–9376 (cited on pp. iv, 7, 13, 47).

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, B., et al. (2019). "Towards federated learning at scale: System design". *Proceedings of Machine Learning and Systems*, 1, pp. 374–388.

Bouneffouf, D., Bouzeghoub, A., and Gançarski, A. L. (2012). "A contextual-bandit algorithm for mobile context-aware recommender system". In: *International conference on neural information processing*. Springer, pp. 324–331 (cited on p. ix).

Bravais, A. (1844). *Analyse mathématique sur les probabilités des erreurs de situation d'un point*. Impr. Royale (cited on p. 163).

Broido, A. D. and Clauset, A. (2019). "Scale-free networks are rare". *Nature communications*, 10(1), pp. 1–10 (cited on p. 35).

Brynjolfsson, E., Hu, Y. J., and Smith (2010). "The longer tail: The changing shape of Amazon's sales distribution curve". *Available at SSRN 1679991* (cited on p. 63).

Bubeck, S. and Cesa-Bianchi, N. (2012). "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". *arXiv preprint arXiv:1204.5721* (cited on p. 67).

Bubeck, S., Munos, R., and Stoltz, G. (2008). "Bandit Exploration" (cited on p. 70).

— (2009). "Pure exploration in multi-armed bandits problems". In: *International conference on Algorithmic learning theory*. Springer, pp. 23–37 (cited on p. 67).

Bubeck, S., Wang, and Viswanathan, N. (2013). "Multiple identifications in multi-armed bandits". In: *International Conference on Machine Learning*. PMLR, pp. 258–265 (cited on p. 67).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Burki, T. (2020). "A new paradigm for drug development". *The Lancet Digital Health*, 2(5), e226–e227 (cited on pp. i, 3).

Butler, Silva, C. da, Shafir, Y., Weisfeld-Adams, J. D., Alexander, J. J., Hegde, M., and Escayg, A. (2017). "De novo and inherited SCN8A epilepsy mutations detected by gene panel analysis". *Epilepsy research*, 129, pp. 17–25 (cited on p. 40).

Carbonell, P., Radivojevic, T., and Garcia Martin, H. (2019). *Opportunities at the intersection of synthetic biology, machine learning, and automation* (cited on p. 3).

Carley, D. W. (2005). "Drug repurposing: identify, develop and commercialize new uses for existing or abandoned drugs. Part I". *IDrugs: the investigational drugs journal*, 8(4), pp. 306–309 (cited on p. ii).

Chatain, T., Haar, S., and Paulevé, L. (2018). "Boolean networks: beyond generalized asynchronicity". In: *International Workshop on Cellular Automata and Discrete Complex Systems*. Springer, pp. 29–42 (cited on pp. 19, 27, 49).

Chatterji, N., Muthukumar, V., and Bartlett, P. (2020). "Osom: A simultaneously optimal algorithm for multi-armed and linear contextual bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1844–1854 (cited on p. 99).

Chen, Engkvist, O., Wang, Y., Olivecrona, M., and Blaschke, T. (2018). "The rise of deep learning in drug discovery". *Drug discovery today*, 23(6), pp. 1241–1250 (cited on p. 62).

Chen, Li, and Qiao, M. (2017). "Nearly instance optimal sample complexity bounds for top-k arm selection". In: *Artificial Intelligence and Statistics*. PMLR, pp. 101–110 (cited on pp. 61, 65, 75, 77, 99, 134).

Chen, See, K. C., et al. (2020). "Artificial intelligence for COVID-19: rapid review". *Journal of medical Internet research*, 22(10), e21476 (cited on p. 3).

Cheng, L. and Li (2016). "Systematic quality control analysis of LINCS data". *CPT: pharmacometrics & systems pharmacology*, 5(11), pp. 588–598 (cited on pp. 48, 161).

Chevalier, S., Froidevaux, C., Paulevé, L., and Zinovyev, A. (2019). "Synthesis of boolean networks from biological dynamical constraints using answer-set programming". In: *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, pp. 34–41 (cited on pp. vii, 20, 22, 56, 164, 165).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Choi, S.-S., Cha, S.-H., and Tappert, C. C. (2010). "A survey of binary similarity and distance measures". *Journal of systemics, cybernetics and informatics*, 8(1), pp. 43–48 (cited on p. 34).

Chowdhury, S. R., Saux, P., Maillard, O.-A., and Gopalan, A. (2022). "Bregman Deviations of Generic Exponential Families". *arXiv:2201.07306* (cited on p. 65).

Chung, Chou, J., and Brown, K. A. (2020). "Neurodevelopmental outcomes of preterm infants: a recent literature review". *Translational pediatrics*, 9(Suppl 1), S3 (cited on p. 116).

Clark, N. R., Hu, K. S., Feldmann, A. S., Kou, Y., Chen, E. Y., Duan, Q., and Ma'ayan, A. (2014). "The characteristic direction: a geometrical approach to identify differentially expressed genes". *BMC bioinformatics*, 15(1), pp. 1–16 (cited on pp. xxiv, 43, 48, 50, 52, 120, 122, 123, 176).

Collombet, S., Oevelen, C. van, Ortega, J. L. S., Abou-Jaoudé, W., Di Stefano, B., Thomas-Chollier, M., Graf, T., and Thieffry, D. (2017). "Logical modeling of lymphoid and myeloid cell specification and transdifferentiation". *Proceedings of the National Academy of Sciences*, 114(23), pp. 5792–5799 (cited on p. 21).

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to algorithms*. MIT press (cited on p. 160).

Crick, F. H. (1958). "On protein synthesis". In: *Symp Soc Exp Biol*. Vol. 12. 138-63, p. 8 (cited on p. 13).

Crump, C., Sundquist, and Sundquist (2021). "Preterm or early term birth and risk of autism". *Pediatrics*, 148(3) (cited on p. 116).

Cushion, T. D., Paciorkowski, A. R., Pilz, D. T., Mullins, J. G., Seltzer, L. E., Marion, R. W., Tuttle, E., Ghoneim, D., Christian, S. L., Chung, S.-K., et al. (2014). "De novo mutations in the beta-tubulin gene TUBB2A cause simplified gyral patterning and infantile-onset epilepsy". *The American Journal of Human Genetics*, 94(4), pp. 634–641 (cited on p. 40).

Danos, V. and Laneve, C. (2004). "Formal molecular biology". *Theoretical Computer Science*, 325(1), pp. 69–110 (cited on p. 16).

De Rooij, S., Van Erven, T., Grünwald, P. D., and Koolen, W. M. (2014). "Follow the leader if you can, hedge if you must". *The Journal of Machine Learning Research*, 15(1), pp. 1281–1316 (cited on p. 108).

Degenne and Koolen, W. M. (2019). "Pure exploration with multiple correct answers". *Advances in Neural Information Processing Systems*, 32 (cited on pp. 67, 69, 74, 100).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Degenne, Koolen, W. M., and Ménard, P. (2019). "Non-asymptotic pure exploration by solving games". *Advances in Neural Information Processing Systems*, 32 (cited on pp. 103, 107).

Degenne, Ménard, P., Shang, X., and Valko, M. (2020). "Gamification of pure exploration for linear bandits". In: *International Conference on Machine Learning*. PMLR, pp. 2432–2442 (cited on pp. 77, 99, 100, 103).

Delahaye-Duriez, Réda, and Gressens (2019). "Identification de cibles thérapeutiques et repositionnement de médicaments par analyses de réseaux géniques". *médecine / sciences*, 35(6-7), pp. 515–518 (cited on p. xviii).

Delahaye-Duriez, Srivastava, Shkura, K., Langley, S. R., Laaniste, L., Moreno-Moral, A., Danis, B., Mazzuferi, M., Foerch, P., Gazina, E. V., et al. (2016). "Rare and common epilepsies converge on a shared gene regulatory network providing opportunities for novel antiepileptic drug discovery". *Genome biology*, 17(1), pp. 1–18 (cited on pp. vii, 11, 12, 26, 40, 49, 159).

Deloitte Centre for Health Solutions (2022). *Nurturing growth : Measuring the return from pharmaceutical innovation 2021*. https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/life-sciences-health-care/Measuring-the-return-of-pharmaceutical-innovation-2021-Deloitte.pdf. Accessed: [May 5, 2022] (cited on p. 2).

Diamond, S. and Boyd, S. (2016). "CVXPY: A Python-embedded modeling language for convex optimization". *The Journal of Machine Learning Research*, 17(1), pp. 2909–2913 (cited on p. 142).

DisGeNet (2022). *FAQ : Original Data Sources*. https://www.disgenet.org/. Accessed: [May 4, 2022] (cited on p. 168).

Doğan, R. I., Leaman, R., and Lu (2014). "NCBI disease corpus: a resource for disease name recognition and concept normalization". *Journal of biomedical informatics*, 47, pp. 1–10 (cited on pp. xxiv, 29).

Du, Y., Chen, W., Yuroki, Y., and Huang, L. (2021). "Collaborative Pure Exploration in Kernel Bandit". *arXiv preprint arXiv:2110.15771* (cited on pp. 100, 135, 136, 140, 143, 146, 153).

Duan, Q., Reid, S. P., Clark, N. R., Wang, Z., Fernandez, N. F., Rouillard, A. D., Readhead, B., Tritsch, S. R., Hodos, R., Hafner, M., et al. (2016). "L1000CDS2: LINCS L1000 characteristic direction signatures search engine". *NPJ systems biology and applications*, 2(1), pp. 1–12 (cited on pp. viii, 48, 56, 129, 176, 177).

Dubey, A. and Pentland, A. (2020). "Differentially-private federated linear bandits". *Advances in Neural Information Processing Systems*, 33, pp. 6003–6014 (cited on pp. 136, 152).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Dugger, S. A., Platt, A., and Goldstein, D. B. (2018). "Drug development in the era of precision medicine". *Nature reviews Drug discovery*, 17(3), pp. 183–196 (cited on p. 12).

Dumitrascu, B., Feng, K., and Engelhardt, B. (2018). "Pg-ts: Improved thompson sampling for logistic contextual bandits". *Advances in neural information processing systems*, 31 (cited on p. 67).

Dunn, S.-J., Martello, G., Yordanov, B., Emmott, S., and Smith, A. (2014). "Defining an essential transcription factor program for naive pluripotency". *Science*, 344(6188), pp. 1156–1160 (cited on pp. 14, 20).

Dunn, S.-J., Li, Carbognin, E., Smith, A., and Martello, G. (2019). "A common molecular logic determines embryonic stem cell self-renewal and reprogramming". *The EMBO journal*, 38(1), e100003 (cited on p. 14).

Dunn, S.-J. and Yordanov, B. (2019). "Automated Reasoning for the Synthesis and Analysis of Biological Programs". In: *Automated Reasoning for Systems Biology and Medicine*. Springer, pp. 37–62 (cited on pp. 22, 164).

Al-Eitan, L. N., Al-Dalalah, I. M., Mustafa, M. M., Alghamdi, M. A., Elshammari, A. K., Khreisat, W. H., Al-Quasmi, M. N., and Aljamal, H. A. (2019). "Genetic polymorphisms of CYP3A5, CHRM2, and ZNF498 and their association with epilepsy susceptibility: a pharmacogenetic and case–control study". *Pharmacogenomics and personalized medicine*, 12, p. 225 (cited on p. 40).

Ekins, S., Puhl, A. C., Zorn, K. M., Lane, T. R., Russo, D. P., Klein, J. J., Hickey, A. J., and Clark, A. M. (2019). "Exploiting machine learning for end-to-end drug discovery and development". *Nature materials*, 18(5), pp. 435–441 (cited on p. 62).

Epilepsy.com (2022). *Summary of Antiepileptic Drugs*. https://www.epilepsy.com/article/2014/3/summary-antiepileptic-drugs. Accessed: [May 16, 2022] (cited on p. 173).

Even-Dar, E., Mannor, S., Mansour, Y., and Mahadevan, S. (2006). "Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems." *Journal of machine learning research*, 7(6) (cited on p. 134).

Eves, E. M., Tucker, M. S., Roback, J. D., Downen, M., Rosner, M. R., and Wainer, B. H. (1992). "Immortal rat hippocampal cell lines exhibit neuronal and glial lineages and neurotrophin gene expression." *Proceedings of the National Academy of Sciences*, 89(10), pp. 4373–4377 (cited on p. 160).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Faury, L., Abeille, M., Calauzènes, C., and Fercoq, O. (2020). "Improved optimistic algorithms for logistic bandits". In: *International Conference on Machine Learning*. PMLR, pp. 3052–3060 (cited on p. 67).

Fiez, T., Jain, L., Jamieson, K. G., and Ratliff, L. (2019). "Sequential experimental design for transductive linear bandits". *Advances in neural information processing systems*, 32 (cited on pp. 76, 77, 86, 103, 135, 136, 143, 146).

Filippi, S., Cappe, O., Garivier, A., and Szepesvári, C. (2010). "Parametric bandits: The generalized linear case". *Advances in Neural Information Processing Systems*, 23 (cited on p. 66).

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A. K., Slichter, C. K., Miller, H. W., McElrath, M. J., Prlic, M., et al. (2015). "MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data". *Genome biology*, 16(1), pp. 1–13 (cited on p. 161).

Fleiss, B., Gressens, and Stolp, H. B. (2020). "Cortical gray matter injury in encephalopathy of prematurity: link to neurodevelopmental disorders". *Frontiers in Neurology*, 11, p. 575.

Foster, D. J., Gentile, C., Mohri, M., and Zimmert, J. (2020). "Adapting to misspecification in contextual bandits". *Advances in Neural Information Processing Systems*, 33, pp. 11478–11489 (cited on p. 99).

Fraser, H. B. (2013). "Gene expression drives local adaptation in humans". *Genome research*, 23(7), pp. 1089–1096.

Gabillon, V., Ghavamzadeh, M., and Lazaric, A. (2012). "Best arm identification: A unified approach to fixed budget and fixed confidence". *Advances in Neural Information Processing Systems*, 25 (cited on pp. xi, xii, 65, 70, 72, 75, 77, 82–84, 93, 134, 179).

Garivier, A. and Kaufmann (2016). "Optimal best arm identification with fixed confidence". In: *Conference on Learning Theory*. PMLR, pp. 998–1027 (cited on pp. 67, 74, 100, 107, 108, 136, 140).

— (2021). "Nonasymptotic sequential tests for overlapping hypotheses applied to near-optimal arm identification in bandit models". *Sequential Analysis*, 40(1), pp. 61–96 (cited on p. 104).

Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., and Wanko, P. (2016). "Theory solving made easy with clingo 5". In: *Technical Communications of the 32nd International Conference on Logic Programming (ICLP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (cited on p. 165).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Ghosh, A., Chowdhury, S. R., and Gopalan, A. (2017). "Misspecified linear bandits". In: *Thirty-First AAAI Conference on Artificial Intelligence* (cited on pp. 94, 99).

Gibbs, D. L. and Shmulevich, I. (2017). "Solving the influence maximization problem reveals regulatory organization of the yeast cell cycle". *PLoS computational biology*, 13(6), e1005591 (cited on p. 30).

Gillespie, D. T. (1977). "Exact stochastic simulation of coupled chemical reactions". *The journal of physical chemistry*, 81(25), pp. 2340–2361 (cited on p. 16).

Go, V., Bowley, B. G., Pessina, M. A., Zhang, Z. G., Chopp, M., Finklestein, S. P., Rosene, D. L., Medalla, M., Buller, B., and Moore, T. L. (2020). "Extracellular vesicles from mesenchymal stem cells reduce microglial-mediated neuroinflammation after cortical injury in aged Rhesus monkeys". *Geroscience*, 42(1), pp. 1–17 (cited on p. 116).

Goldfarb, D. and Idnani, A. (1983). "A numerically stable dual method for solving strictly convex quadratic programs". *Mathematical programming*, 27(1), pp. 1–33 (cited on p. 102).

González, F. L., Osorio, X. R., Rein, A. G.-N., Martínez, M. C., Fernández, J. S., Haba, V. V., Pedraza, A. D., and Cerdá, J. M. (2015). "Drug-resistant epilepsy: definition and treatment alternatives". *Neurología (English Edition)*, 30(7), pp. 439–446 (cited on p. 24).

Goodwin, B. C. et al. (1963). "Temporal organization in cells. A dynamic theory of cellular control processes." *Temporal organization in cells. A dynamic theory of cellular control processes.* (cited on p. 15).

Griffin, A., Carpenter, C., Liu, J., Paterno, R., Grone, B., Hamling, K., Moog, M., Dinday, M. T., Figueroa, F., Anvar, M., et al. (2021). "Phenotypic analysis of catastrophic childhood epilepsy genes". *Communications biology*, 4(1), pp. 1–13 (cited on pp. 43, 44).

Guillou, F., Gaudel, R., and Preux, P. (2016). "Large-scale bandit recommender system". In: *International Workshop on Machine Learning, Optimization, and Big Data*. Springer, pp. 204–215 (cited on p. 63).

Hall, B. A. and Niarakis, A. (2021). "Data integration in logic-based models of biological mechanisms". *Current Opinion in Systems Biology*, 28, p. 100386.

Han, C.-L., Hu, W., Stead, M., Zhang, T., Zhang, J.-G., Worrell, G. A., and Meng, F.-G. (2014). "Electrical stimulation of hippocampus for the treatment of refractory temporal lobe epilepsy". *Brain research bulletin*, 109, pp. 13–21 (cited on p. 33).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Harrington, E. C. (1965). "The desirability function". *Industrial quality control*, 21(10), pp. 494–498 (cited on pp. vii, 165).

Hassidim, A., Kupfer, R., and Singer, Y. (2020). "An optimal elimination algorithm for learning a best arm". *Advances in Neural Information Processing Systems*, 33, pp. 10788–10798 (cited on pp. 77, 135).

He, J., Yang, Gong, Z., et al. (2020). "Hybrid attentional memory network for computational drug repositioning". *BMC bioinformatics*, 21(1), pp. 1–17 (cited on p. 46).

Hillel, E., Karnin, Z. S., Koren, T., Lempel, R., and Somekh, O. (2013). "Distributed exploration in multi-armed bandits". *Advances in Neural Information Processing Systems*, 26 (cited on pp. 135, 136, 146).

Hock, A. K., Vigneron, A. M., Carter, S., Ludwig, R. L., and Vousden, K. H. (2011). "Regulation of p53 stability and function by the deubiquitinating enzyme USP42". *The EMBO journal*, 30(24), pp. 4921–4930 (cited on p. 13).

Hodos, Kidd, B. A., Khader, S., Readhead, B. P., and Dudley, J. T. (2016). "Computational approaches to drug repurposing and pharmacology". *Wiley interdisciplinary reviews. Systems biology and medicine*, 8(3), p. 186 (cited on pp. 4, 48, 62).

Hoksza, D., Gawron, P., Ostaszewski, M., Smula, E., and Schneider, R. (2019). "MINERVA API and plugins: opening molecular network analysis and visualization to the community". *Bioinformatics*, 35(21), pp. 4496–4498 (cited on p. 50).

Huang, Y., Furuno, M., Arakawa, T., Takizawa, S., Hoon, M. de, Suzuki, H., and Arner, E. (2019). "A framework for identification of on-and off-target transcriptional responses to drug treatment". *Scientific reports*, 9(1), pp. 1–9 (cited on p. 25).

Hubert, L. and Arabie, P. (1985). "Comparing partitions". *Journal of classification*, 2(1), pp. 193–218 (cited on p. 53).

Hwang, T. J., Carpenter, D., Lauffenburger, J. C., Wang, B., Franklin, J. M., and Kesselheim, A. S. (2016). "Failure of investigational drugs in late-stage clinical development and publication of trial results". *JAMA internal medicine*, 176(12), pp. 1826–1833 (cited on p. 47).

Institute for Clinical and Economic Review (ICER) (2022). *The Next Generation of Rare Disease Drug Policy : Ensuring Both Innovation and Affordability*. https://download2.eurordis.org/ertc/ertc33/ICER-White-Paper_The-Next-Generation-of-Rare-Disease-Drug-Policy_040722.pdf. Accessed: [May 5, 2022] (cited on p. 3).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Islam, M. Z. and Brankovic, L. (2011). "Privacy preserving data mining: A noise addition framework using a novel clustering technique". *Knowledge-Based Systems*, 24(8), pp. 1214–1223.

Jaccard, P. (1912). "The distribution of the flora in the alpine zone. 1". *New phytologist*, 11(2), pp. 37–50 (cited on p. 53).

Jamshidi, M., Lalbakhsh, A., Talla, J., Peroutka, Z., Hadjilooei, F., Lalbakhsh, P., Jamshidi, M., La Spada, L., Mirmozafari, M., Dehghani, M., et al. (2020). "Artificial intelligence and COVID-19: deep learning approaches for diagnosis and treatment". *Ieee Access*, 8, pp. 109581–109595 (cited on p. 3).

Jarada, T. N., Rokne, J. G., and Alhajj, R. (2020). "A review of computational drug repositioning: strategies, approaches, opportunities, challenges, and directions". *Journal of cheminformatics*, 12(1), pp. 1–23 (cited on p. 46).

Jedra, Y. and Proutiere, A. (2020). "Optimal best-arm identification in linear bandits". *Advances in Neural Information Processing Systems*, 33, pp. 10007–10017 (cited on pp. 66, 99, 103, 110).

Jiang, H., Li, and Qiao, M. (2017). "Practical algorithms for best-k identification in multi-armed bandits". *arXiv preprint arXiv:1705.06894* (cited on pp. 61, 65).

Johannessen Landmark, C., Potschka, H., Auvin, S., Wilmshurst, J. M., Johannessen, S. I., Kasteleijn-Nolst Trenité, D., and Wirrell, E. C. (2021). "The role of new medical treatments for the management of developmental and epileptic encephalopathies: Novel concepts and results". *Epilepsia*, 62(4), pp. 857–873 (cited on p. 173).

Johnson, Behmoaras, J., Bottolo, L., Krishnan, M. L., Pernhorst, K., Santoscoy, P. L. M., Rossetti, T., Speed, D., Srivastava, P. K., Chadeau-Hyam, M., et al. (2015). "Systems genetics identifies Sestrin 3 as a regulator of a proconvulsant gene network in human epileptic hippocampus". *Nature communications*, 6(1), pp. 1–11 (cited on p. 25).

Jourquin, J., Duncan, D., Shi, Z., and Zhang, B. (2012). "GLAD4U: deriving and prioritizing gene lists from PubMed literature". *BMC genomics*, 13(8), pp. 1–12 (cited on pp. 40, 171).

Jun, K.-S. and Nowak, R. (2016). "Anytime exploration for multi-armed bandits using confidence information". In: *International Conference on Machine Learning*. PMLR, pp. 974–982 (cited on p. 67).

Kalume, F., Westenbroek, R. E., Cheah, C. S., Frank, H. Y., Oakley, J. C., Scheuer, T., Catterall, W. A., et al. (2013). "Sudden unexpected death in

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

a mouse model of Dravet syndrome". *The Journal of clinical investigation*, 123(4), pp. 1798–1808 (cited on p. 24).

Kalyanakrishnan, S., Tewari, A., Auer, P., and Stone, P. (2012). "PAC subset selection in stochastic multi-armed bandits." In: *ICML*. Vol. 12, pp. 655–662 (cited on pp. xi, xii, 61, 63, 65, 69, 72, 75, 77, 82–85, 92, 93, 110, 134, 194).

Karlebach, G. and Shamir, R. (2008). "Modelling and analysis of gene regulatory networks". *Nature reviews Molecular cell biology*, 9(10), pp. 770–780 (cited on pp. 15, 16).

Karpov, N., Zhang, Q., and Zhou, Y. (2020). "Collaborative top distribution identifications with limited interaction". In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, pp. 160–171.

Karush, W. (1939). "Minima of functions of several variables with inequalities as side constraints". *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago* (cited on p. 139).

Katz-Samuels, J., Jain, L., Jamieson, et al. (2020). "An empirical process approach to the union bound: Practical algorithms for combinatorial and linear bandits". *Advances in Neural Information Processing Systems*, 33, pp. 10371–10382 (cited on p. 112).

Katz-Samuels, J. and Jamieson (2020). "The true sample complexity of identifying good arms". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1781–1791 (cited on p. 112).

Kauffman, S. A. (1969). "Metabolic stability and epigenesis in randomly constructed genetic nets". *Journal of theoretical biology*, 22(3), pp. 437–467 (cited on pp. v, 15, 26).

Kaufmann (2014). "Analysis of bayesian and frequentist strategies for sequential resource allocation". PhD thesis. Télécom ParisTech (cited on p. 89).

Kaufmann, Cappé, O., and Garivier, A. (2016). "On the complexity of best-arm identification in multi-armed bandit models". *The Journal of Machine Learning Research*, 17(1), pp. 1–42 (cited on pp. 72, 73, 99, 102, 138).

Kaufmann and Garivier, A. (2017). "Learning the distribution with largest mean: two bandit frameworks". *ESAIM: Proceedings and surveys*, 60, pp. 114–131 (cited on p. 63).

Kaufmann and Kalyanakrishnan, S. (2013). "Information complexity in bandit subset selection". In: *Conference on Learning Theory*. PMLR, pp. 228–251 (cited on pp. 69, 72, 75, 77, 110, 143, 150).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Kaufmann and Koolen, W. M. (2021). "Mixture martingales revisited with applications to sequential tests and confidence intervals". *Journal of Machine Learning Research*, 22(246), pp. 1–44 (cited on pp. 145, 187).

Kaufmann, Ménard, P., Domingues, O. D., Jonsson, A., Leurent, E., and Valko, M. (2021). "Adaptive reward-free exploration". In: *Algorithmic Learning Theory*. PMLR, pp. 865–891 (cited on pp. 72, 192).

Kempe, D., Kleinberg, and Tardos, É. (2003). "Maximizing the spread of influence through a social network". In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146 (cited on pp. 30, 33).

Kirschner, J., Lattimore, T., Vernade, C., and Szepesvári, C. (2021). "Asymptotically optimal information-directed sampling". In: *Conference on Learning Theory*. PMLR, pp. 2777–2821 (cited on p. 110).

Knutson, A. and Tao, T. (2001). "Honeycombs and sums of Hermitian matrices". *Notices Amer. Math. Soc*, 48(2).

Kolesnikov, N., Hastings, E., Keays, M., Melnichuk, O., Tang, Y. A., Williams, E., Dylag, M., Kurbatova, N., Brandizi, M., Burdett, T., et al. (2015). "ArrayExpress update—simplifying data submissions". *Nucleic acids research*, 43(D1), pp. D1113–D1116.

Koren, Y. (2008). "Factorization meets the neighborhood: a multifaceted collaborative filtering model". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434 (cited on p. 56).

Krishnan, M. L., Van Steenwinckel, J., Schang, A.-L., Yan, J., Arnadottir, J., Le Charpentier, T., Csaba, Z., Dournaud, P., Cipriani, S., Auvynet, C., et al. (2017). "Integrative genomics of microglia implicates DLG4 (PSD95) in the white matter development of preterm infants". *Nature communications*, 8(1), pp. 1–11 (cited on p. 117).

Kuhn, H. W. and Tucker, A. W. (2014). "Nonlinear programming". In: *Traces and emergence of nonlinear programming*. Springer, pp. 247–258 (cited on p. 139).

Kullback, S. and Leibler, R. A. (1951). "On information and sufficiency". *The annals of mathematical statistics*, 22(1), pp. 79–86 (cited on pp. xxv, 73).

Kuruba, R., Hattiangady, B., and Shetty, A. K. (2009). "Hippocampal neurogenesis and neural stem cells in temporal lobe epilepsy". *Epilepsy & Behavior*, 14(1), pp. 65–73 (cited on p. 24).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Lal, D., Trucks, H., Møller, R. S., Hjalgrim, H., Koeleman, B. P., Kovel, C. G. de, Visscher, F., Weber, Y. G., Lerche, H., Becker, F., et al. (2013). "Rare exonic deletions of the RBFOX1 gene increase risk of idiopathic generalized epilepsy". *Epilepsia*, 54(2), pp. 265–271 (cited on pp. 40, 43).

Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., Lerner, J., Brunet, J.-P., Subramanian, A., Ross, K. N., et al. (2006). "The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease". *science*, 313(5795), pp. 1929–1935 (cited on pp. ii, xxiv, 7, 47, 48).

Lattimore, T. and Szepesvari, C. (2017). "The end of optimism? an asymptotic analysis of finite-armed linear bandits". In: *Artificial Intelligence and Statistics*. PMLR, pp. 728–737.

Lattimore, T., Szepesvari, C., and Weisz, G. (2020). "Learning with good feature representations in bandits and in rl with a generative model". In: *International Conference on Machine Learning*. PMLR, pp. 5662–5670 (cited on p. 99).

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Lek, M., Karczewski, K. J., Minikel, E. V., Samocha, K. E., Banks, E., Fennell, T., O'Donnell-Luria, A. H., Ware, J. S., Hill, A. J., Cummings, B. B., et al. (2016). "Analysis of protein-coding genetic variation in 60,706 humans". *Nature*, 536(7616), pp. 285–291 (cited on pp. xxv, 38).

Leskovec, J., Adamic, L. A., and Huberman, B. A. (2007). "The dynamics of viral marketing". *ACM Transactions on the Web (TWEB)*, 1(1), 5–es (cited on p. 30).

Li, Chu, W., Langford, J., and Schapire, R. E. (2010). "A contextual-bandit approach to personalized news article recommendation". In: *Proceedings of the 19th international conference on World wide web*, pp. 661–670 (cited on pp. x, 76).

Li, Lu, and Zhou (2017). "Provably optimal algorithms for generalized linear contextual bandits". In: *International Conference on Machine Learning*. PMLR, pp. 2071–2080 (cited on p. 66).

Liao, Y., Wang, J., Jaehnig, E. J., Shi, Z., and Zhang, B. (2019). "WebGestalt 2019: gene set analysis toolkit with revamped UIs and APIs". *Nucleic acids research*, 47(W1), W199–W205 (cited on pp. 39, 125).

Lim, N. and Pavlidis, P. (2021). "Evaluation of connectivity map shows limited reproducibility in drug repositioning". *Scientific reports*, 11(1), pp. 1–14 (cited on pp. 48, 52, 58, 161).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Liu, Slotine, J.-J., and Barabási, A.-L. (2011). "Controllability of complex networks". *nature*, 473(7346), pp. 167–173 (cited on p. 25).

— (2012). "Control centrality and hierarchical structure in complex networks" (cited on pp. viii, xxiv, 30, 38).

Liu, Zhang, Yan, K., Chen, F., Huang, W., Lv, B., Sun, C., Xu, L., Li, F., and Jiang, X. (2014). "Mesenchymal stem cells inhibit lipopolysaccharide-induced inflammatory responses of BV2 microglial cells through TSG-6". *Journal of neuroinflammation*, 11(1), pp. 1–12 (cited on p. 116).

Love, M. I., Huber, W., and Anders, S. (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". *Genome biology*, 15(12), pp. 1–21 (cited on pp. 120, 123).

Lowe, D. (2019). *The Latest on Drug Failure and Approval Rates*. https://www.science.org/content/blog-post/latest-drug-failure-and-approval-rates. Accessed: [March 23, 2022] (cited on pp. i, 3).

Maitland, M. L. and Schilsky, R. L. (2011). "Clinical trials in the era of personalized oncology". *CA: a cancer journal for clinicians*, 61(6), pp. 365–381 (cited on p. 114).

Mak, K.-K. and Pichika, M. R. (2019). "Artificial intelligence in drug development: present status and future prospects". *Drug discovery today*, 24(3), pp. 773–780 (cited on p. 3).

Mary, J., Gaudel, R., and Preux, P. (2015). "Bandits and recommender systems". In: *International Workshop on Machine Learning, Optimization and Big Data*. Springer, pp. 325–336 (cited on p. 63).

Mattick, J. S., Taft, R. J., and Faulkner, G. J. (2010). "A global view of genomic information–moving beyond the gene and the master regulator". *Trends in genetics*, 26(1), pp. 21–28 (cited on p. 29).

McAdams, H. H. and Arkin, A. (1997). "Stochastic mechanisms in gene expression". *Proceedings of the National Academy of Sciences*, 94(3), pp. 814–819 (cited on p. 16).

Meisler, M. H., O'brien, J. E., and Sharkey, L. M. (2010). "Sodium channel gene family: epilepsy mutations, gene interactions and modifier effects". *The Journal of physiology*, 588(11), pp. 1841–1848 (cited on p. 51).

Ministère des solidarités et de la santé, F. G. (2018). *PLAN NATIONAL MALADIES RARES 2018-2022: Partager l'innovation, un diagnostic et un traitement pour chacun*. https://www.nouvelle-aquitaine.ars.sante.fr/media/35359/download?inline. Accessed: [May 5, 2022] (cited on p. 3).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Mirza, N., Appleton, R., Burn, S., Plessis, D. du, Duncan, R., Farah, J. O., Feenstra, B., Hviid, A., Josan, V., Mohanraj, R., et al. (2017). "Genetic regulation of gene expression in the epileptic human hippocampus". *Human molecular genetics*, 26(9), pp. 1759–1769 (cited on pp. viii, 24, 33, 43, 54, 151, 169, 175).

Mirza, N., Stevelink, R., Taweel, B., Koeleman, B. P., and Marson, A. G. (2021). "Using common genetic variants to find drugs for common epilepsies". *Brain communications*, 3(4), fcab287 (cited on p. 130).

Montagud, A., Béal, J., Tobalina, L., Traynard, P., Subramanian, V., Szalai, B., Alföldi, R., Puskás, L., Valencia, A., Barillot, E., et al. (2022). "Patient-specific Boolean models of signalling networks guide personalised treatments". *Elife*, 11, e72626 (cited on pp. 28, 59).

Mouchlis, V. D., Afantitis, A., Serra, A., Fratello, M., Papadiamantis, A. G., Aidinis, V., Lynch, I., Greco, D., and Melagraki, G. (2021). "Advances in de novo drug design: from conventional to machine learning methods". *International journal of molecular sciences*, 22(4), p. 1676 (cited on pp. i, 4).

Mudunuri, U., Che, A., Yi, M., and Stephens, R. M. (2009). "bioDBnet: the biological database network". *Bioinformatics*, 25(4), pp. 555–556 (cited on p. 160).

Musa, A., Ghoraie, L. S., Zhang, S.-D., Glazko, G., Yli-Harja, O., Dehmer, M., Haibe-Kains, B., and Emmert-Streib, F. (2018). "A review of connectivity map and computational approaches in pharmacogenomics". *Briefings in bioinformatics*, 19(3), pp. 506–523 (cited on pp. 6, 48, 117).

Myers, C. T., McMahon, J. M., Schneider, A. L., Petrovski, S., Allen, A. S., Carvill, G. L., Zemel, M., Saykally, J. E., LaCroix, A. J., Heinzen, E. L., et al. (2016). "De novo mutations in SLC1A2 and CACNA1A are important causes of epileptic encephalopathies". *The American Journal of Human Genetics*, 99(2), pp. 287–298 (cited on p. 40).

Nakken, K. O., Eriksson, A.-S., Lossius, R., and Johannessen, S. I. (2003). "A paradoxical effect of levetiracetam may be seen in both children and adults with refractory epilepsy". *Seizure*, 12(1), pp. 42–46 (cited on p. 58).

Niarakis, A. and Helikar, T. (2021). "A practical guide to mechanistic systems modeling in biology using a logic-based approach". *Briefings in Bioinformatics*, 22(4), bbaa236 (cited on p. 21).

Nicolle, R., Radvanyi, F., and Elati, M. (2015). "CoRegNet: reconstruction and integrated analysis of co-regulatory networks". *Bioinformatics*, 31(18), pp. 3066–3068 (cited on pp. 25, 39).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Noble, W. S. et al. (2004). "Support vector machine applications in computational biology". *Kernel methods in computational biology*, 14, pp. 71–92 (cited on p. 62).

Ogren, J. A., Wilson, C. L., Bragin, A., Lin, J. J., Salamon, N., Dutton, R. A., Luders, E., Fields, T. A., Fried, I., Toga, A. W., et al. (2009). "Three-dimensional surface maps link local atrophy and fast ripples in human epileptic hippocampus". *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society*, 66(6), pp. 783–791 (cited on p. 24).

Ohba, C., Shiina, M., Tohyama, J., Haginoya, K., Lerman-Sagie, T., Okamoto, N., Blumkin, L., Lev, D., Mukaida, S., Nozaki, F., et al. (2015). "GRIN 1 mutations cause encephalopathy with infantile-onset epilepsy, and hyperkinetic and stereotyped movement disorders". *Epilepsia*, 56(6), pp. 841–848 (cited on p. 40).

Osorio, I., Burastine, T. H., Render, B., Manon-Espaillat, R., and Reed, R. C. (1989). "Phenytoin-induced seizures: a paradoxical effect at toxic concentrations in epileptic patients". *Epilepsia*, 30(2), pp. 230–234 (cited on p. 58).

Ostrowski, M., Paulevé, L., Schaub, T., Siegel, A., and Guziolowski, C. (2016). "Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming". *Biosystems*, 149, pp. 139–153 (cited on p. 19).

Pacchiano, A., Phan, M., Abbasi Yadkori, Y., Rao, A., Zimmert, J., Lattimore, T., and Szepesvari, C. (2020). "Model selection in contextual stochastic bandit problems". *Advances in Neural Information Processing Systems*, 33, pp. 10328–10337 (cited on p. 99).

Paciorkowski, A. R., McDaniel, S. S., Jansen, L. A., Tully, H., Tuttle, E., Ghoneim, D. H., Tupal, S., Gunter, S. A., Vasta, V., Zhang, Q., et al. (2015). "Novel mutations in ATP1A3 associated with catastrophic early life epilepsy, episodic prolonged apnea, and postnatal microcephaly". *Epilepsia*, 56(3), pp. 422–430.

Pallmann, P., Bedding, A. W., Choodari-Oskooei, B., Dimairo, M., Flight, L., Hampson, L. V., Holmes, J., Mander, A. P., Odondi, L., Sydes, M. R., et al. (2018). "Adaptive designs in clinical trials: why use them, and how to run and report them". *BMC medicine*, 16(1), pp. 1–15 (cited on p. 9).

Papini, M., Tirinzoni, A., Restelli, M., Lazaric, A., and Pirotta, M. (2021). "Leveraging good representations in linear contextual bandits". In: *International Conference on Machine Learning*. PMLR, pp. 8371–8380 (cited on p. 94).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Pardee, A. B., Jacob, F., and Monod, J. (1959). "The genetic control and cytoplasmic expression of "inducibility" in the synthesis of β-galactosidase by E. coli". *Journal of Molecular Biology*, 1(2), pp. 165–178 (cited on pp. 15, 17).

Passera, S., Boccazzi, M., Bokobza, C., Faivre, V., Mosca, F., Van Steenwinckel, J., Fumagalli, M., Gressens, P., and Fleiss, B. (2021). "Therapeutic potential of stem cells for preterm infant brain damage: Can we move from the heterogeneity of preclinical and clinical studies to established therapeutics?" *Biochemical Pharmacology*, 186, p. 114461 (cited on p. 116).

Paulevé, L., Kolčák, J., Chatain, T., and Haar, S. (2020). "Reconciling qualitative, abstract, and scalable modeling of biological networks". *Nature communications*, 11(1), pp. 1–7 (cited on pp. 16, 27, 49, 54, 63, 164, 165).

Pepi, C., Palma, L. de, Trivisano, M., Pietrafusa, N., Lepri, F. R., Diociaiuti, A., Camassei, F. D., Carfi-Pavia, G., De Benedictis, A., Rossi-Espagnet, C., et al. (2021). "The role of KRAS mutations in cortical malformation and epilepsy surgery: a novel report of nevus sebaceous syndrome and review of the literature". *Brain Sciences*, 11(6), p. 793 (cited on p. 173).

Perrault, P., Healey, J., Wen, Z., and Valko, M. (2020). "Budgeted online influence maximization". In: *International Conference on Machine Learning*. PMLR, pp. 7620–7631 (cited on p. 33).

Petrovski, S., Wang, Q., Heinzen, E. L., Allen, A. S., and Goldstein, D. B. (2013). "Genic intolerance to functional variation and the interpretation of personal genomes". *PLoS genetics*, 9(8), e1003709 (cited on pp. xxv, 38).

Pharma Intelligence (2022). *Navigate the landscape in our Pharma R&D Annual Review*. https://pages.pharmaintelligence.informa.com/rdreview-globaldigital?utm_source=RDReview2022. Accessed: [May 5, 2022].

Piñero, J., Ramìrez-Anguita, J. M., Saüch-Pitarch, J., Ronzano, F., Centeno, E., Sanz, F., and Furlong, L. I. (2020). "The DisGeNET knowledge platform for disease genomics: 2019 update". *Nucleic acids research*, 48(D1), pp. D845–D855 (cited on pp. 28, 40, 168, 171).

Poke, G., King, C., Muir, A., Valles-Ibáñez, G. de, Germano, M., Moura de Souza, C. F., Fung, J., Chung, B., Fung, C. W., Mignot, C., et al. (2019). "The epileptology of GNB5 encephalopathy". *Epilepsia*, 60(11), e121–e127 (cited on pp. 39, 40).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Réda and Delahaye-Duriez (2022). "Prioritization of Candidate Genes Through Boolean Networks". In: *International Conference on Computational Methods in Systems Biology*. Springer, pp. 89–121 (cited on pp. v, xviii, 8, 11, 24).

Réda, Kaufmann, and Delahaye-Duriez (2020). "Machine learning applications in drug development". *Computational and structural biotechnology journal*, 18, pp. 241–252 (cited on pp. ii, xviii, 2, 11, 46, 61).

— (2021). "Top-m identification for linear bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1108–1116 (cited on pp. xi, xviii, xxiv, 8, 61, 69, 76, 80, 92, 93, 110, 150, 178).

Réda, Tirinzoni, and Degenne (2021). "Dealing With Misspecification In Fixed-Confidence Linear Top-m Identification". *Advances in Neural Information Processing Systems*, 34 (cited on pp. xiii, xviii, xxv, 8, 61, 69, 86, 94, 97, 98, 102, 103, 107–109).

Réda, Vakili, and Kaufmann (2022). "Near-Optimal Federated Learning in Bandits". In: 36th *Conference on Neural Information Processing Systems*. In press (cited on pp. xv, xviii, xxiv, 8, 114, 131).

Réda and Wilczyński, B. (2020). "Automated inference of gene regulatory networks using explicit regulatory modules". *Journal of Theoretical Biology*, 486, p. 110091 (cited on pp. 16, 22, 156).

Russac, Y., Katsimerou, C., Bohle, D., Cappé, O., Garivier, A., and Koolen, W. M. (2021). "A/B/n Testing with Control in the Presence of Subpopulations". *Advances in Neural Information Processing Systems*, 34 (cited on pp. 100, 135, 136, 140).

Russo, D. and Van Roy, B. (2013). "Eluder dimension and the sample complexity of optimistic exploration". *Advances in Neural Information Processing Systems*, 26 (cited on p. 66).

Santillán, M. and Mackey, M. C. (2008). "Quantitative approaches to the study of bistability in the lac operon of Escherichia coli". *Journal of The Royal Society Interface*, 5(suppl_1), S29–S39 (cited on p. 17).

Sardana, D., Zhu, C., Zhang, M., Gudivada, R. C., Yang, L., and Jegga, A. G. (2011). "Drug repositioning for orphan diseases". *Briefings in bioinformatics*, 12(4), pp. 346–356 (cited on p. 4).

Schuhmacher, A., Gassmann, O., and Hinder, M. (2016). "Changing R&D models in research-based pharmaceutical companies". *Journal of translational medicine*, 14(1), pp. 1–11 (cited on p. 3).

Shang, X., Heide, R., Menard, P., Kaufmann, E., and Valko, M. (2020). "Fixed-confidence guarantees for Bayesian best-arm identification". In: *Interna-*

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

tional Conference on Artificial Intelligence and Statistics. PMLR, pp. 1823–1832 (cited on p. 107).

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). "Cytoscape: a software environment for integrated models of biomolecular interaction networks". *Genome research*, 13(11), pp. 2498–2504 (cited on p. 30).

Sharma, A., Halu, A., Decano, J. L., Padi, M., Liu, Y.-Y., Prasad, R. B., Fadista, J., Santolini, M., Menche, J., Weiss, S. T., et al. (2018). "Controllability in an islet specific regulatory network identifies the transcriptional factor NFATC4, which regulates type 2 diabetes associated genes". *NPJ systems biology and applications*, 4(1), pp. 1–11 (cited on p. 30).

Shi, C. and Shen, C. (2021). "Federated multi-armed bandits". In: *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)* (cited on p. xvi).

Shi, C., Shen, C., and Yang (2021). "Federated multi-armed bandits with personalization". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2917–2925 (cited on pp. xvi, 131, 133–135, 143, 146, 148, 150).

Simchowitz, M., Jamieson, and Recht, B. (2017). "The simulator: Understanding adaptive sampling in the moderate-confidence regime". In: *Conference on Learning Theory*. PMLR, pp. 1794–1834 (cited on p. 99).

Sims, A. H., Howell, A., Howell, S. J., and Clarke, R. B. (2007). "Origins of breast cancer subtypes and therapeutic implications". *Nature Clinical Practice Oncology*, 4(9), pp. 516–525 (cited on p. 130).

Sirota, M., Dudley, J. T., Kim, J., Chiang, A. P., Morgan, A. A., Sweet-Cordero, A., Sage, J., and Butte, A. J. (2011). "Discovery and preclinical validation of drug indications using compendia of public gene expression data". *Science translational medicine*, 3(96), 96ra77–96ra77 (cited on p. 47).

Smith and Linden, G. (2017). "Two decades of recommender systems at Amazon.com". *Ieee internet computing*, 21(3), pp. 12–18 (cited on p. 63).

Soare, M., Lazaric, A., and Munos, R. (2014). "Best-arm identification in linear bandits". *Advances in Neural Information Processing Systems*, 27 (cited on pp. 66, 77).

Spinelli, E., Christensen, K. R., Bryant, E., Schneider, A., Rakotomamonjy, J., Muir, A. M., Giannelli, J., Littlejohn, R. O., Roeder, E. R., Schmidt, B., et al. (2021). "Pathogenic MAST3 variants in the STK domain are associated with epilepsy". *Annals of neurology*, 90(2), pp. 274–284.

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Srivastava, Bagnati, M., Delahaye-Duriez, A., Ko, J.-H., Rotival, M., Langley, S. R., Shkura, K., Mazzuferi, M., Danis, B., Eyll, J. van, et al. (2017). "Genome-wide analysis of differential RNA editing in epilepsy". *Genome research*, 27(3), pp. 440–450 (cited on p. 24).

Srivastava, van Eyll, J., Godard, P., Mazzuferi, M., Delahaye-Duriez, A., Van Steenwinckel, J., Gressens, P., Danis, B., Vandenplas, C., Foerch, P., et al. (2018). "A systems-level framework for drug discovery identifies Csf1R as an anti-epileptic drug target". *Nature communications*, 9(1), pp. 1–15 (cited on p. 25).

Stamberger, H., Nikanorova, M., Willemsen, M. H., Accorsi, P., Angriman, M., Baier, H., Benkel-Herrenbrueck, I., Benoit, V., Budetta, M., Caliebe, A., et al. (2016). "STXBP1 encephalopathy: a neurodevelopmental disorder including epilepsy". *Neurology*, 86(10), pp. 954–962 (cited on p. 40).

Steinlein, O. K. (2014). "Calcium signaling and epilepsy". *Cell and tissue research*, 357(2), pp. 385–393 (cited on p. 51).

Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., Kroemer, G., Barillot, E., and Calzone, L. (2017). "MaBoSS 2.0: an environment for stochastic Boolean modeling". *Bioinformatics*, 33(14), pp. 2226–2228 (cited on pp. 44, 169).

Subramanian, Narayan, R., Corsello, S. M., Peck, D. D., Natoli, T. E., Lu, X., Gould, J., Davis, J. F., Tubelli, A. A., Asiedu, J. K., et al. (2017). "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles". *Cell*, 171(6), pp. 1437–1452 (cited on pp. vi, viii, 7, 13, 26, 28, 48–51, 128, 160).

Szklarczyk, D., Gable, A. L., Nastou, K. C., Lyon, D., Kirsch, R., Pyysalo, S., Doncheva, N. T., Legeay, M., Fang, T., Bork, P., et al. (2021). "The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets". *Nucleic acids research*, 49(D1), pp. D605–D612 (cited on pp. vii, 28, 34, 159, 168).

Takahashi, K. and Yamanaka, S. (2006). "Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors". *cell*, 126(4), pp. 663–676 (cited on p. 14).

Takemura, K., Ito, S., Hatano, D., Sumita, H., Fukunaga, T., Kakimura, N., and Kawarabayashi, K.-i. (2021). "A parameter-free algorithm for misspecified linear contextual bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 3367–3375 (cited on p. 99).

Tao, Zhang, and Zhou (2019). "Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits". In:

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, pp. 126–146 (cited on pp. 133, 135, 136).

Tari, L. B. and Patel, J. H. (2014). "Systematic drug repurposing through text mining". *Biomedical Literature Mining*, pp. 253–267 (cited on p. 4).

Thomas, R. (1973). "Boolean formalization of genetic control circuits". *Journal of theoretical biology*, 42(3), pp. 563–585 (cited on pp. v, 15, 26).

— (1978). "Logical analysis of systems comprising feedback loops". *Journal of Theoretical Biology*, 73(4), pp. 631–656 (cited on p. 15).

Thompson, W. R. (1933). "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". *Biometrika*, 25(3-4), pp. 285–294 (cited on pp. ix, 65).

Tirinzoni, Pirotta, M., Restelli, M., and Lazaric, A. (2020). "An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits". *Advances in Neural Information Processing Systems*, 33, pp. 1417–1427 (cited on p. 110).

Treiman, D. M. (2001). "GABAergic mechanisms in epilepsy". *Epilepsia*, 42, pp. 8–12 (cited on p. 51).

Ursu, O., Holmes, J., Knockel, J., Bologa, C. G., Yang, J. J., Mathias, S. L., Nelson, S. J., and Oprea, T. I. (2016). "DrugCentral: online drug compendium". *Nucleic acids research*, gkw993 (cited on p. 50).

Vaginay, A., Boukhobza, T., and Smail-Tabbone, M. (2021). "Automatic synthesis of boolean networks from biological knowledge and data". In: *International Conference on Optimization and Learning*. Springer, pp. 156–170 (cited on p. 19).

Vaishya, R., Javaid, M., Khan, I. H., and Haleem, A. (2020). "Artificial Intelligence (AI) applications for COVID-19 pandemic". *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 14(4), pp. 337–339 (cited on p. 3).

Valiant, L. G. (1984). "A theory of the learnable". *Communications of the ACM*, 27(11), pp. 1134–1142 (cited on pp. xxv, 68).

Van der Maaten, L. and Hinton, G. (2008). "Visualizing data using t-SNE." *Journal of machine learning research*, 9(11).

Vargesson, N. (2015). "Thalidomide-induced teratogenesis: History and mechanisms". *Birth Defects Research Part C: Embryo Today: Reviews*, 105(2), pp. 140–156 (cited on p. 4).

Villar, S. S., Bowden, J., and Wason, J. (2015). "Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges". *Statis-*

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

tical science: a review journal of the Institute of Mathematical Statistics, 30(2), p. 199 (cited on p. 9).

Vitaliti, G. and Falsaperla, R. (2021). "Chorioamnionitis, Inflammation and Neonatal Apnea: Effects on Preterm Neonatal Brainstem and on Peripheral Airways: Chorioamnionitis and Neonatal Respiratory Functions". *Children*, 8(10), p. 917.

Von Mering, C., Jensen, L. J., Snel, B., Hooper, S. D., Krupp, M., Foglierini, M., Jouffre, N., Huynen, M. A., and Bork, P. (2005). "STRING: known and predicted protein–protein associations, integrated and transferred across organisms". *Nucleic acids research*, 33(suppl_1), pp. D433–D437 (cited on p. 160).

Wagenmaker, A., Katz-Samuels, J., and Jamieson (2021). "Experimental design for regret minimization in linear bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 3088–3096 (cited on p. 112).

Walker, Hamley, J. I., Milton, P., Monnot, F., Kinrade, S., Specht, S., Pedrique, B., and Basáñez, M.-G. (2021). "Supporting Drug Development for Neglected Tropical Diseases Using Mathematical Modeling". *Clinical Infectious Diseases*, 73(6), e1391–e1396 (cited on p. 3).

Walker, Mirza, N., Yip, V., Marson, A., and Pirmohamed, M. (2015). "Personalized medicine approaches in epilepsy". *Journal of internal medicine*, 277(2), pp. 218–234 (cited on p. 130).

Wang and Goldstein, D. B. (2020). "Enhancer domains predict gene pathogenicity and inform gene discovery in complex disease". *The American Journal of Human Genetics*, 106(2), pp. 215–233 (cited on pp. xxiv, 38).

Watson, J. D. and Crick, F. H. (1953). "The structure of DNA". In: *Cold Spring Harbor symposia on quantitative biology*. Vol. 18. Cold Spring Harbor Laboratory Press, pp. 123–131 (cited on p. 5).

Wery, M., Dameron, O., Nicolas, J., Remy, E., and Siegel, A. (2019). "Formalizing and enriching phenotype signatures using Boolean networks". *Journal of Theoretical Biology*, 467, pp. 66–79 (cited on pp. 19, 27).

Weyl, H. (1912). "Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung)". *Mathematische Annalen*, 71(4), pp. 441–479 (cited on p. 183).

Wilks, S. S. (1938). "The large-sample distribution of the likelihood ratio for testing composite hypotheses". *The annals of mathematical statistics*, 9(1), pp. 60–62 (cited on p. 104).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., Sajed, T., Johnson, D., Li, C., Sayeeda, Z., et al. (2018). "DrugBank 5.0: a major update to the DrugBank database for 2018". *Nucleic acids research*, 46(D1), pp. D1074–D1082 (cited on p. 50).

World Health Organization (WHO) (2018). *Preterm birth*. https://www.who.int/news-room/fact-sheets/detail/preterm-birth. Accessed: [April 26, 2022] (cited on p. 116).

— (2021). *WHO model list of essential medicines - 22nd list, 2021*. https://www.who.int/publications/i/item/WHO-MHP-HPS-EML-2021.02. Accessed: [May 5, 2022] (cited on p. 6).

— (2022). *Epilepsy*. https://www.who.int/news-room/fact-sheets/detail/epilepsy. Accessed: [April 29, 2022] (cited on p. 24).

Wu, Li, Wang, J., and Wu, F.-X. (2018). "CytoCtrlAnalyser: a Cytoscape app for biomolecular network controllability analysis". *Bioinformatics*, 34(8), pp. 1428–1430 (cited on pp. 30, 38).

Wu, Wang, Gu, Q., and Wang, H. (2016). "Contextual bandits in a collaborative environment". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 529–538 (cited on p. 134).

Xin, Y., Chen, S., Tang, K., Wu, Y., and Guo, Y. (2022). "Identification of Nifurtimox and Chrysin as Anti-Influenza Virus Agents by Clinical Transcriptome Signature Reversion". *International Journal of Molecular Sciences*, 23(4), p. 2372 (cited on pp. ii, 6, 117).

Xu, L., Honda, J., and Sugiyama, M. (2018). "A fully adaptive algorithm for pure exploration in linear bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 843–851 (cited on pp. 66, 76, 77, 82, 83, 86, 91, 92, 110).

Xue, H., Li, J., Xie, H., and Wang, Y. (2018). "Review of drug repositioning approaches and resources". *International journal of biological sciences*, 14(10), p. 1232 (cited on p. 46).

Yaari, G., Bolen, C. R., Thakar, J., and Kleinstein, S. H. (2013). "Quantitative set analysis for gene expression: a method to quantify gene set differential expression including gene-gene correlations". *Nucleic acids research*, 41(18), e170–e170 (cited on pp. xxv, 39, 124).

Yang, Chen, Pasteris, S., Hajiesmaili, M., Lui, J., and Towsley, D. (2021). "Cooperative stochastic bandits with asynchronous agents and constrained feedback". *Advances in Neural Information Processing Systems*, 34, pp. 8885–8897 (cited on p. 135).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Yang, Luo, H., Li, Y., and Wang, J. (2019). "Drug repositioning based on bounded nuclear norm regularization". *Bioinformatics*, 35(14), pp. i455–i463 (cited on p. 46).

Yang, Soares, J., Greninger, P., Edelman, E. J., Lightfoot, H., Forbes, S., Bindal, N., Beare, D., Smith, J. A., Thompson, I. R., et al. (2012). "Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells". *Nucleic acids research*, 41(D1), pp. D955–D961 (cited on pp. xxiv, 58).

Yordanov, B., Wintersteiger, C. M., Hamadi, Y., and Kugler, H. (2013). "Z34Bio: An SMT-based framework for analyzing biological computation". In: *SMT Workshop 2013 11th International Workshop on Satisfiability Modulo Theories* (cited on p. 21).

Young, W. C., Yeung, K. Y., and Raftery, A. E. (2016). "A posterior probability approach for gene regulatory network inference in genetic perturbation data". *arXiv preprint arXiv:1603.04835* (cited on p. 12).

Zaki, M., Mohan, A., and Gopalan, A. (2020). "Explicit best arm identification in linear bandits using no-regret learners". *arXiv preprint arXiv:2006.07562* (cited on p. 103).

Zerrouk, N., Miagoux, Q., Dispot, A., Elati, M., and Niarakis, A. (2020). "Identification of putative master regulators in rheumatoid arthritis synovial fibroblasts using gene expression data and network inference". *Scientific reports*, 10(1), pp. 1–13 (cited on p. 25).

Zhang, Yue, X., Huang, F., Liu, R., Chen, Y., and Ruan, C. (2018). "Predicting drug-disease associations and their therapeutic function based on the drug-disease association bipartite network". *Methods*, 145, pp. 51–59 (cited on p. 62).

Zhao, Sun, J., and Zhao (2012). "Distinct and competitive regulatory patterns of tumor suppressor genes and oncogenes in ovarian cancer" (cited on p. 14).

Zhou, Zhang, Y., Lian, X., Li, F., Wang, C., Zhu, F., Qiu, Y., and Chen, Y. (2022). "Therapeutic target database update 2022: facilitating drug discovery with enriched comparative data of targeted agents". *Nucleic Acids Research*, 50(D1), pp. D1398–D1407 (cited on pp. xxv, 50).

Zhu, Mulle, E., Smith, C. S., and Liu, J. (2021). "Decentralized Multi-Armed Bandit Can Outperform Classic Upper Confidence Bound". *arXiv preprint arXiv:2111.10933* (cited on p. 135).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/

Zhu, Zhou, D., Jiang, R., Gu, Q., Willett, R., and Nowak, R. (2021). "Pure Exploration in Kernel and Neural Bandits". *Advances in Neural Information Processing Systems*, 34 (cited on p. 103).

Zhu, Zhu, Liu, J., and Liu, Y. (2021). "Federated bandit: A gossiping approach". In: *Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, pp. 3–4 (cited on pp. 134–136).

Ziegler, A., Colin, E., Goudenège, D., and Bonneau, D. (2019). "A snapshot of some pLI score pitfalls". *Human mutation*, 40(7), pp. 839–841 (cited on p. 39).

Zucchelli, P. (2018). *Lab Automation Increases Repeatability, Reduces Errors in Drug Development*. https://www.technologynetworks.com/drug-discovery/articles/lab-automation-increases-repeatability-reduces-errors-in-drug-development-310034. Accessed: [March 23, 2022] (cited on p. i).

Except where otherwise noted, this work is licensed under
https://creativecommons.org/licenses/by-nd/3.0/fr/