Conceptions de phénotypes computationnels pour la recherche en santé publique

Pegdwendé Sawadogo, Thomas Guyet, Etienne Audureau

Conceptions de phénotypes computationnels pour la recherche en santé publique

Pegdwendé N. Sawadogo[1], Thomas Guyet[2] et Etienne Audureau[3]

[1] FONDATION DE L'AP-HP, Paris, France
nicolas.sawadogo-ext@aphp.fr

[2] INRIA, Centre de Lyon, France

[3] AP-HP, Hôpital Henri Mondor, Université Paris Est Créteil, France

Résumé : Au cours des dernières décennies, la digitalisation des systèmes médicaux a ouvert la voie à de nouvelles opportunités de recherche en santé publique en rendant accessibles des données de santé à grande échelle via les entrepôts de données de santé. Cependant, l'exploitation de ces données dans des études épidémiologiques nécessite une étape primordiale de préparation des données, qui consiste notamment à identifier et caractériser les patients ciblés par l'étude. Au travers de ces données numériques, il faut retrouver des phénotypes d'intérêt pour une étude. On parle alors de « phénotype computationnel ». Or, il n'existe pas à ce jour de consensus sur l'approche à suivre pour concevoir un phénotype computationnel, ni même sur sa définition. D'ailleurs, les définitions existant dans la littérature se contredisent parfois, générant ainsi des confusions autour de ce concept. Pour y remédier, nous proposons dans cet article une analyse de l'état de l'art centrée sur le concept de phénotype computationnel. Nous identifions ainsi cinq dimensions à travers desquels les phénotypes computationnels peuvent être caractérisés, classifiés ou différenciés.

Mots-clés : Phénotypes computationnels, Ontologies, Données médicales.

1 Introduction

Au cours des dernières décennies, l'utilisation des outils numériques dans les hôpitaux et plus globalement dans les systèmes sanitaires a permis de répondre à plusieurs problèmes de santé publique. Ces outils digitaux permettent à l'échelle individuelle d'améliorer et de faciliter les diagnostics à l'aide par exemple d'algorithmes d'intelligence artificielle (Awad *et al.*, 2021). Ils permettent également un meilleur ajustement de l'offre de soins aux besoins des patients à travers la conception de prothèses et d'implants par impression 3D (Awad *et al.*, 2021). À l'échelle collective, la digitalisation des systèmes médicaux offre d'immenses opportunités de recherche en santé publique, dans la mesure où elle rend disponibles des données de santé à grande échelle via les entrepôts de données de santé (De Moor *et al.*, 2014). L'existence de ces entrepôts de données de santé (EDS) ouvre en effet la voie à des études cliniques et épidémiologiques comme l'analyse des interactions entre les médicaments, la surveillance des effets secondaires de médicaments, ou encore l'évaluation de l'efficacité des traitements (De Moor *et al.*, 2014; Rivault *et al.*, 2019). En France, les deux grandes sources de données utilisées à cet usage sont les données hospitalières et les données de l'assurance maladie (SNIIRAM).

Tous ces cas d'usage nécessitent une étape primordiale de constitution de cohorte de patients, c'est-à-dire, une liste des patients à inclure dans une étude. Cela consiste plus concrètement à identifier les patients correspondant à un phénotype. Loebe *et al.* (2012) décrit un phénotype comme un ensemble de caractéristiques observables et qui peuvent être reliés à une condition (e.g. maladie, exposition à un traitement). Nous appelons cela des phénotypes « traditionnels ». L'utilisation de ces phénotypes dits « traditionnels » sur des EDS n'est pas aisée. En effet, les phénotypes sont historiquement conçus dans le seul but d'être lisibles et compréhensibles par l'Homme et nécessitent donc d'être traduits sous une forme adaptée à la machine pour être utilisés efficacement dans un contexte de grands volumes de données, comme c'est le cas dans les EDS (Mo *et al.*, 2015). De plus, les caractéristiques observables ne sont pas nécessairement les caractéristiques accessibles via les données. Il faut donc parfois les adapter pour correspondre aux informations effectivement disponibles.

Pour remédier à cela, plusieurs approches ont été proposées au cours de la dernière décennie pour la conception de phénotypes dits « computationnels », c'est-à-dire, des phénotypes liés à des EDS (Ahmad *et al.*, 2020; Sharma *et al.*, 2019). Si ces approches se rapportent toutes à l'utilisation de phénotypes dans un contexte d'EDS, elles restent cependant diverses dans le « comment ». Cette hétérogénéité d'approches rend d'ailleurs ambigu le concept de phénotype computationnel. Ainsi, certains auteurs conçoivent le phénotype computationnel comme un modèle d'apprentissage machine (Sharma *et al.*, 2019) quand d'autres le présentent comme une requête (De Moor *et al.*, 2014; Ahmad *et al.*, 2020) ou un *workflow* (Mo *et al.*, 2015). De ce fait, il est difficile pour des chercheurs ou praticiens de comparer des phénotypes computationnels entre eux, et donc de réaliser un choix éclairé de l'approche de phénotypage la plus adaptée à leur contexte.

Pour remédier aux ambiguïtés autour du concept de phénotype computationnel, nous réalisons dans cet article une analyse comparative des principales définitions et conceptions de phénotypes computationnels. À travers cette étude, nous contribuons à la littérature en introduisant une définition multi-dimensionnelle du concept de phénotype computationnel. Ainsi, nous offrons non seulement une vision plus claire du concept, mais aussi un outil de comparaison des approches de phénotypage computationnel.

2 Phénotypes computationnels dans la littérature

Dans cette partie, nous analysons les principales définitions et conceptions de phénotypes computationnels dans la littérature. Pour ce faire, nous commençons par présenter les principales visions existant dans la littérature avant d'en discuter les limites.

2.1 Notion de phénotype computationnels

Le terme « phénotype » a été introduit en 1909 par Whilhelm Johannsen. Il est généralement défini comme un ensemble de caractéristiques observables permettant de distinguer des organismes (Johannsen, 1911; Loebe *et al.*, 2012; Richesson & Smerek, 2014; Uciteli *et al.*, 2020). Les caractéristiques entrant dans la définition d'un phénotype peuvent être de plusieurs ordres, notamment morphologiques (couleur des yeux, taille), bio-chimiques (gènes, groupes sanguins), ou cliniques (symptômes) (Johannsen, 1911; Uciteli *et al.*, 2020).

La notion de « phénotype computationnel » est quant à elle beaucoup plus récente. Elle est étroitement liée à l'émergence des EDS. Il existe dans la littérature plusieurs visions du phénotype computationnel, dont certaines sont contradictoires.

De Moor *et al.* (2014) et Ahmad *et al.* (2020) présentent ainsi le phénotype computationnel comme la matérialisation d'un phénotype « traditionnel », c'est-à-dire, un ensemble de caractéristiques, sous la forme d'une requête exécutable sur un EDS. Mo *et al.* (2015) conçoivent le phénotype computationnel sous la forme d'un *workflow*. Ceci peut être vu comme une version plus flexible de la notion de requête puisque moins contrainte par une algèbre. Sharma *et al.* (2019) proposent une vision alternative en concevant le phénotype computationnel sous la forme d'un modèle d'apprentissage supervisé (arbre de classification, règles de décision) qui apprend à constituer une cohorte de patients à partir d'un ensemble de caractéristiques.

En plus de ces désaccords sur la forme matérielle que doit prendre un phénotype computationnel, il existe une diversité de visions sur les interactions entre les phénotypes computationnels, et les EDS. Ainsi, la définition de Mo *et al.* (2015) spécifie que le phénotype computationnel doit être appliqué à (ou exécuté sur) un EDS. A *contrario*, De Moor *et al.* (2014) et Uciteli *et al.* (2020) soulignent qu'un phénotype computationnel doit être conçu à partir des données d'un EDS. Enfin, Richesson & Smerek (2014) intègrent à la fois ces deux exigences dans leur définition du phénotype computationnel.

2.2 Discussion

En analysant les différentes visions du concept de phénotype computationnel dans la littérature, nous constatons de nombreuses différences et limites. Une première différence

concerne les exigences édictées sur la forme que doit prendre le phénotype computationnel (*workflow*, modèle d'apprentissage machine, requête). En effet, la diversité de visions à ce niveau démontre qu'un phénotype computationnel peut se matérialiser de diverses manières. Nous voyons là l'opportunité d'apporter une spécification d'un phénotype computationnel qui englobe cette diversité.

En plus d'être pour certaines trop spécifiques, les définitions du concept de phénotype computationnel dans la littérature sont pour la plupart partielles. Alors que certaines définitions se focalisent sur les entrées d'un phénotype computationnel (De Moor *et al.*, 2014; Uciteli *et al.*, 2020), d'autres ciblent ses sorties (Sharma *et al.*, 2019), ou encore son exécution (Mo *et al.*, 2015). À notre connaissance, aucune définition n'intègre à la fois l'ensemble de ces dimensions.

Une nouvelle définition plus complète nous paraît donc nécessaire pour remédier aux confusions qui existent dans la littérature autour des phénotypes computationnels. C'est ce que nous proposons dans la suite de cet article.

3 Vision multi-dimensionnelle des phénotypes computationnels

Dans cette partie, nous proposons une vision plus complète des phénotypes computationnels en les caractérisant à travers cinq dimensions. Pour chacune de ces dimensions, nous définissons et motivons un ensemble d'exigences et de critères de différenciation possibles pour un phénotype computationnel. Ces dimensions et exigences nous servent par la suite de grille d'analyse de l'état de l'art. Ces cinq dimensions sont 1) les interactions avec l'EDS ($\mathcal{I}$), 2) les entrées ($\mathcal{E}$), 3) les sorties ($\mathcal{S}$), 4) la matérialisation ($\mathcal{M}$) et 5) la complétude ($\mathcal{C}$) d'un phénotype. Ainsi, nous définissons un phénotype ($\mathcal{P}$) comme un assemblage de ces cinq dimensions.

$$\mathcal{P} = \langle \mathcal{I}, \mathcal{E}, \mathcal{S}, \mathcal{M}, \mathcal{C} \rangle$$

3.1 Interactions avec les entrepôts de données de santé

Il existe deux types d'interactions possibles entre les phénotypes computationnels et les EDS. Les phénotypes computationnels peuvent être conçus à partir des données d'un EDS d'une part (De Moor *et al.*, 2014; Uciteli *et al.*, 2019) et, d'autre part, ils peuvent être utilisés pour requêter les mêmes EDS (Mo *et al.*, 2015; Richesson & Smerek, 2014).

Application d'un phénotype computationnel à un EDS (*Exc*) Les phénotypes computationnels se distinguent principalement des phénotypes « traditionnels » par la présence d'une forme exécutable (Section 3.4). En ce sens, cette capacité à être exécuté sur un EDS paraît primordiale. C'est d'ailleurs ce qui rend les phénotypes computationnels appropriés dans un contexte d'EDS.

Génération du phénotype computationnel à partir d'un EDS (*Gen*) Ce mode d'interaction pose la question de l'apprenabilité d'un phénotype computationnel. En utilisant l'EDS sur lequel le phénotype computationnel sera exécuté, il est en effet possible d'ajuster le phénotype computationnel en intégrant par exemple les terminologies présentes dans l'entrepôt (Bacry *et al.*, 2020). On peut y voir deux limites : 1) une forte spécificité au jeu de données (et donc peu réutilisable) et 2) une limitation dans l'expressivité des phénotypes (les contraintes liées à l'apprenabilité des phénotypes réduisant probablement leur expressivité). Cet autre type d'interaction est certes utile, mais reste tout de même optionnel.

3.2 Entrées d'un phénotype computationnel

Comme les phénotypes « traditionnels », les phénotypes computationnels se conçoivent à partir d'un ensemble de caractéristiques sur les patients. Ces caractéristiques que nous considérons comme des « entrées » peuvent être des évènements ou des faits définis individuellement ou combinés entre eux.

Faits (*Fai*) Ce sont des éléments factuels et observables sur un patient. Certaines de ces caractéristiques n'ont pas vocation à évoluer avec le temps. Nous parlons donc de « faits permanents ». C'est par exemple le cas du sexe biologique, du groupe sanguin, ou encore de la date de naissance d'un patient. *A contrario*, d'autres faits comme le poids ou la taille fluctuent au cours de la vie du patient. On parle de « faits évolutifs ».

Évènements (*Evn*) Ils traduisent des actions médicales réalisées ou subies par le patient. Les évènements sont associés à un horodatage ou à un intervalle de temps. Nous parlons d'« évènements ponctuels » dans le cas où ils sont associés à un horodatage unique (prescription de médicament, examen clinique), et d'« évènements continus » lorsqu'ils sont associés à une fenêtre de temps (hospitalisation) (Bacry *et al.*, 2020).

3.3 Sorties d'un phénotype computationnel

Les « sorties » désignent les résultats de l'exécution d'un phénotype computationnel. Elles peuvent prendre la forme de cohortes de patients ou de scores de correspondance.

Cohorte de patients (*Coh*) C'est le cas le plus commun (De Moor *et al.*, 2014; Bache *et al.*, 2015; Ahmad *et al.*, 2020; Bacry *et al.*, 2020). Cela consiste à retourner une liste de patients selon qu'ils correspondent ou non aux caractéristiques définies par le phénotype computationnel. Il s'agit donc du résultat d'une suite d'évaluation binaire.

Scores de correspondance (*Sco*) Ce type de sortie est beaucoup moins envisagé dans la littérature, mais n'en demeure pas moins pertinent. En effet, la correspondance du patient peut être partielle vis-à-vis du phénotype (Ahmad *et al.*, 2020). Il semble donc utile d'évaluer la correspondance des patients de façon floue. Il s'agit dans ce cas d'établir à quel point chaque patient correspond aux faits et évènements intégrés dans le phénotype computationnel.

Contextes temporels (*Con*) Ce type de sortie consiste à préciser pour des patients correspondant à un phénotype, le moment auquel cette correspondance a commencé et/ou s'est arrêtée. Par exemple, pour des patients atteints de diabète, il s'agirait d'identifier le temps (jour, mois, années) auquel la pathologie a été diagnostiquée. Selon le besoin d'étude, ce temps peut être exprimé en termes d'âge du patient ou en date calendaire.

3.4 Matérialisation des phénotypes computationnels

Les phénotypes computationnels peuvent être vus comme une traduction de phénotypes « traditionnels » sous un format adapté aux EDS (Sharma *et al.*, 2019; Ahmad *et al.*, 2020). Autrement dit, il s'agit d'un phénotype rendu compréhensible par la machine. En ce sens, plusieurs types de matérialisation computationnelle sont possibles :

Workflow (*Wkf*) Ce type de matérialisation représente le phénotype computationnel à travers une liste séquentielle d'opérations, présentée sous une forme graphique (Chapman *et al.*, 2021). Contrairement aux autres types de matérialisation, le *workflow* a l'avantage d'être compréhensible et facilement utilisable par des non-informaticiens.

Script (*Scr*) Il se distingue du *workflow* par l'absence d'éléments graphiques. Le phénotype computationnel est alors représenté exclusivement à travers une suite d'instructions décrite dans un langage de programmation. Le projet SciKit-EDS [1] s'inscrit dans ce type d'approche. Cette solution technique a un fort pouvoir expressif, mais peut rendre difficile la réutilisabilité des solutions mises en place.

Requête (*Req*) Ici, le phénotype computationnel est matérialisé sous la forme d'un ensemble de restrictions à travers un langage de requête. Le langage de requête utilisé est étroitement lié au format de stockage adopté pour l'EDS. Par exemple, le langage SPARQL est utilisé quand les données sont stockées au format RDF (Jiang *et al.*, 2017) alors que le langage SQL sied dans le cas de données stockées en bases de données relationnelles (Bakalara *et al.*, 2019).

1. `https://www.inria.fr/fr/scikiteds`

Modèle apprenable (*App*) Un phénotype computationnel obtenu comme le résultat d'un algorithme d'apprentissage a généralement une forme assez spécifique, propre à la méthode d'apprentissage elle-même. La question qui se pose sur ce type de modèle est leur interprétabilité par les épidémiologistes.

3.5 Complétude d'un phénotype computationnel

En plus des caractéristiques ci-dessus, nous identifions un ensemble de fonctionnalités usuelles des phénotypes computationnels. La prise en charge ou non de ces fonctionnalités permet de juger de la complétude, c'est-à-dire la richesse expressive et l'adaptabilité des phénotypes computationnels.

Contraintes temporelles (*Tmp*) D'après Bache *et al.* (2015), des contraintes temporelles sont exprimées dans 47% des phénotypes « traditionnels ». Ces contraintes doivent donc être transposables aux phénotypes computationnels. On devrait ainsi pouvoir spécifier la nécessité qu'un évènement existe (ou non) avant ou après un autre, ou encore dans un certain délai dans le parcours d'un patient.

Entrées hétérogènes (*Het*) Par définition, un phénotype computationnel se base sur des données contenues dans des EDS (Sharma *et al.*, 2019; Ahmad *et al.*, 2020). Dans l'idéal, le phénotype devrait donc pouvoir intéragir avec les différents types et formats de données de santé disponibles. Nous en distinguons trois : 1) les données médicales textuelles (comptes-rendus de consultation, interprétation de résultats), 2) les données d'imageries médicales et 3) les données structurées (prescriptions, résultats de biologie, hospitalisations, achats ou remboursements de médicaments, etc.). De manière encore marginale, d'autres types de données (données omiques, données vocales) émergent, mais ne sont pas pris en considération ici.

Support de plusieurs modèles de données (*Mdl*) Les EDS sont organisés suivant des modèles hétérogènes. Certes, des modèles standards comme FHIR (Jiang *et al.*, 2017) et OMOP (Bathelt, 2021) sont communs à plusieurs EDS, mais ils ne sont pas les seuls. Il est alors souhaitable pour un phénotype computationnel d'être adapté au maximum de modèles de données.

Moindre technicité d'utilisation (*Tec*) Les phénotypes computationnels, à l'image des phénotypes traditionnels, s'adressent aux épidémiologistes et cliniciens qui s'en servent pour la définition de cohortes de patients. De ce fait, il est souhaitable que ces phénotypes computationnels ne nécessitent pas de compétences techniques (au sens informatique) trop spécifiques. En ce sens, la matérialisation sous forme de *workflow* pourrait par exemple être préférée à la présentation sous forme de requête ou de script.

Extension sémantique (*Sem*) Il existe généralement un écart sémantique entre les données médicales stockées dans l'EDS et les caractéristiques exprimées dans le phénotype computationnel (Bakalara *et al.*, 2021). Par exemple, le phénotype pourrait exprimer une famille de médicaments, tandis que l'information stockée dans l'EDS se rapporte à un médicament spécifique. Une des solutions pour y remédier consiste à ajouter une couche sémantique au phénotype computationnel (Rivault *et al.*, 2019).

La Figure 1 récapitule les caractéristiques ci-dessus définies pour un phénotype computationnel en précisant pour chaque dimension les options possibles.

4 Vers une preuve de concept

Dans cette section, nous démontrons l'effectivité des cinq dimensions proposées pour caractériser un phénotype computationnel. Pour ce faire, nous analysons et comparons des exemples issus de six systèmes de phénotypage computationnels à la lumière de ces dimensions. Nous avons choisi ces systèmes car ils sont suffisamment détaillés pour être analysés et qu'ils couvrent une grande partie des approches existantes. Leurs caractéristiques sont résumées dans la Table 1.

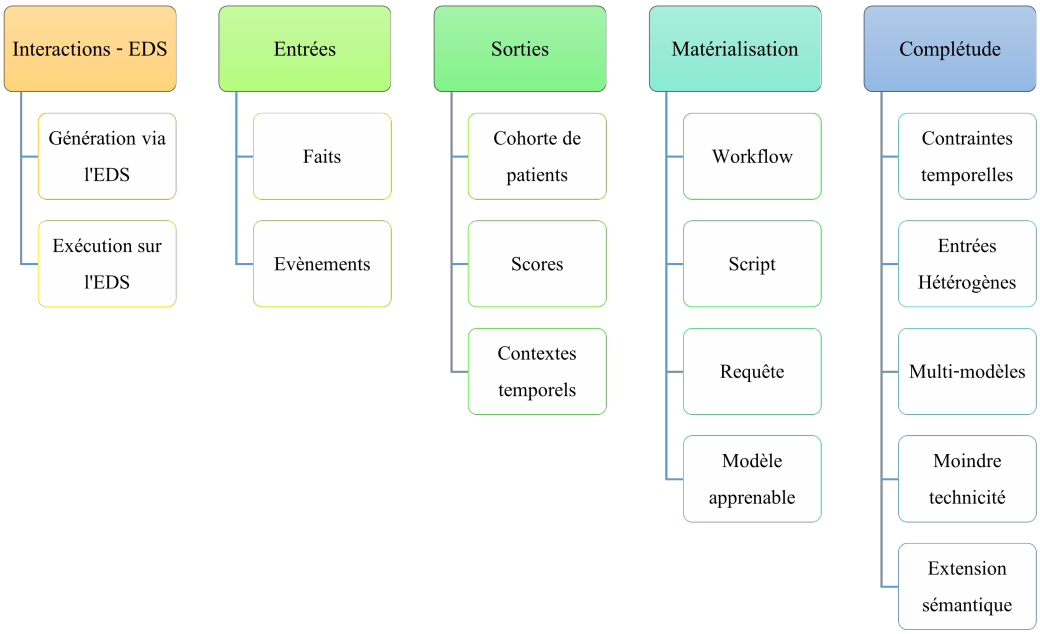

FIGURE 1 – *Caractéristiques potentielles d'un phénotype computationnel*

4.1 Systèmes de phénotypage ciblés

Dans le système **ECLECTIC** proposé par Bache *et al.* (2015), un phénotype computationnel prend la forme d'une requête produisant une cohorte de patients. Du point de vue de la complétude, les phénotypes résultants de ce système intègrent notamment une extension sémantique qui leur permet de prendre en compte les évènements définis à travers des systèmes de codage (ICD-10, LOINC, ATC, etc.). Par ailleurs, ECLECTIC supporte des contraintes temporelles entre les évènements. Son adoption par des utilisateurs non-techniques reste cependant limitée dans la mesure ou cela requiert pour eux d'apprendre un langage de requêtage.

L'approche de Jiang *et al.* (2017) conçoit les phénotypes computationnels sous la forme de requêtes SPARQL. Cette requête est automatiquement générée à l'aide d'une description fournie par l'utilisateur. De ce fait, la définition du phénotype computationnel requiert un faible niveau de technicité de la part de l'utilisateur. L'approche de Jiang *et al.* (2017) reste cependant limitée du point de vue des contraintes temporelles, et surtout du point de vue des modèles de données supportés qui se limitent au modèle FHIR.

Uciteli *et al.* (2020) conçoivent le phénotype computationnel sous la forme d'une ontologie qui est traitée comme une requête. Cette ontologie intègre alors un ensemble de faits et d'évènements combinés via des conjonctions ontologiques. Une interface graphique assiste les utilisateurs dans la conception de l'ontologie-phénotype. Comme dans l'approche de Jiang *et al.* (2017), ce système est spécifique au modèle de données FHIR.

Dans le système **Phenoflow** proposé par Chapman *et al.* (2021), les phénotypes computationnels sont conçus comme des *workflows*, c'est-à-dire des suites d'étapes présentées via une interface graphique. Cette représentation graphique permet une prise en main par des utilisateurs ayant peu de compétences techniques. Parmi les limites de Phenoflow, on peut citer la spécificité au modèle de données OMOP ainsi que le non prise en charge explicite des contraintes temporelles.

Le système **SCALPEL** de Bacry *et al.* (2020) intègre quant à lui un système de conception de phénotypes computationnels spécifique au SNIIRAM, c'est-à-dire aux données du système national de santé français. Les phénotypes computationnels présents dans SCALPEL prennent la forme de scripts de traitements permettant de générer des cohortes de patients.

Enfin, Bakalara *et al.* (2021) proposent un système où les phénotypes se matérialisent

TABLE 1 – *Comparaison de systèmes de conception de phénotypes computationnels*

Système	Interactions	Entrées	Sorties	Matérialisation	Complétude
ECLECTIC	Gen, Exc	Fai, Evn	Coh	Req	Temp, Sem
Jiang *et al.* (2017)	Exc	Fai, Evn	Coh	Req	Tec
Uciteli *et al.* (2020)	Gen, Exc	Fai, Evn	Coh	Req	Tec, Sem
PhenoFlow	Exc	Fai, Evn	Coh	Wkf	Tec
SCALPEL	Exc	Fai, Evn	Coh	Scr	Tmp, Sem
Bakalara *et al.* (2021)	Exc	Evn	Coh	Req	Tec, Tmp, Sem

sous forme de requêtes SPARQL. Ces requêtes sont automatiquement générées à partir de graphes particuliers appelés « chroniques », qui servent à définir et combiner les évènements à prendre en compte. Grâce aux chroniques, le système de Bakalara *et al.* (2021) supporte les contraintes temporelles tout en restant adapté à des utilisateurs non-techniques.

4.2 Résultats

En analysant la synthèse présentée dans la Table 1 on constate sur le plan des interactions que l'ensemble des phénotypes computationnels s'exécutent sur un EDS. De même, les entrées sont presque toujours des combinaisons d'évènements et de faits (à l'exception de Bakalara *et al.* (2021) qui considèrent uniquement des évènements). Quant aux sorties, elles sont partout et exclusivement des cohortes de patients.

A *contrario*, seulement une minorité des systèmes de phénotypage prévoit une génération des phénotypes à partir des données de santé. On constate également une diversité de matérialisations des phénotypes computationnels. Ils sont ainsi exécutés sous forme de requêtes, de *workflow* ou encore de script. Certains systèmes proposent même une représentation additionnelle du phénotype computationnel, par exemple, à travers des chroniques pour Bakalara *et al.* (2021), et via une description logique dans le système de Jiang *et al.* (2017).

Enfin, du point de vue des critères de complétude, On constate une prise en charge quasi-systématique de l'extension sémantique, et dans une moindre mesure des contraintes temporelles et de l'adaptation aux utilisateurs non-techniques. En revanche, on constate que la quasi-totalité des systèmes présentés ne sont pas génériques, ni du point de vue des modèles de données, ni du point des types de données pris en compte.

5 Conclusion

Dans cet article, nous avons proposé une analyse de l'état de l'art autour du concept de phénotype computationnel. Pour ce faire, nous avons identifié cinq dimensions à travers desquelles les phénotypes computationnels peuvent être définis. Ce sont les types d'interactions entre le phénotype computationnel et les données de santé, les entrées, les sorties, la matérialisation et la complétude du phénotype computationnel. Nous avons ainsi identifié des similitudes, des divergences et surtout des limites dans les approches existantes de conception de phénotypes computationnels.

Dans nos travaux futurs, nous envisageons de proposer une nouvelle approche de conception de phénotypes computationnels qui surpasse les limites de la littérature. Pour cela, nous souhaiterions tout d'abord formaliser notre proposition de phénotype computationnel sous la forme d'une ontologie permettant de décrire de manière interopérable des phénotypes. Plus concrètement, il s'agirait par exemple de prendre en comptes des sorties autres que les « simples » cohortes de patients (scores, contextes temporels). Un autre axe de recherche serait la conception de phénotypes computationnels nativement agnostiques aux modèles de données (OMOP, FHIR, PCORnet, etc.) et/ou supportant des données hétérogènes en entrée. Nous pourrions également proposer des phénotypes computationnels conçus à partir des systèmes d'information médicaux en général et non plus uniquement des entrepôts de données de santé.

Remerciements

Une partie des recherches présentées dans cet article est subventionnée par la Fondation de l'AP-HP, dans le cadre de la Chaire AI-RACLES et a reçu l'accord du Comité scientifique et éthique du CDW de l'AP-HP (CSE-20-11-COVIPREDS).

Références

AHMAD F. S., RICKET I. M., HAMMILL B. G., ESKENAZI L., ROBERTSON H. R., CURTIS L. H., DOBI C. D., GIROTRA S., HAYNES K., KIZER J. R. *et al.* (2020). Computable phenotype implementation for a national, multicenter pragmatic clinical trial : Lessons learned from ADAPTABLE. *Circulation : Cardiovascular Quality and Outcomes*, **13**(6), 355–364.

AWAD A., TRENFIELD S. J., POLLARD T. D., ONG J. J., ELBADAWI M., MCCOUBREY L. E., GOYANES A., GAISFORD S. & BASIT A. W. (2021). Connected healthcare : Improving patient care using digital health technologies. *Advanced Drug Delivery Reviews*, **178**, 113958.

BACHE R., TAWEEL A., MILES S. & DELANEY B. C. (2015). An eligibility criteria query language for heterogeneous data warehouses. *Methods of information in medicine*, **54**(01), 41–44.

BACRY E., GAÏFFAS S., LEROY F., MOREL M., NGUYEN D.-P., SEBIAT Y. & SUN D. (2020). SCALPEL3 : A scalable open-source library for healthcare claims databases. *International Journal of Medical Informatics*, **141**, 104203.

BAKALARA J., GUYET T., DAMERON O., HAPPE A. & OGER E. (2021). An extension of chronicles temporal model with taxonomies-application to epidemiological studies. In *Proceedings of the International Conference on Health Informatics, (HEALTHINF)*, p. 133–142 : ScitePress.

BAKALARA J., GUYET T., DAMERON O., OGER E. & HAPPE A. (2019). Temporal models of care sequences for the exploration of medico-administrative data. In *Doctoral Consortium of the Conference on Artificial Intelligence in Medicine (AIME)*, p. 1–8.

BATHELT F. (2021). The usage of OHDSI OMOP – a scoping review. In *Proceedings of the German Medical Data Sciences (GMDS)*, p. 95–95 : IOS Press.

CHAPMAN M., RASMUSSEN L. V., PACHECO J. A. & CURCIN V. (2021). PhenoFlow : A microservice architecture for portable workflow-based phenotype definitions. In *Proceedings of the Joint Summits on Translational Science*, p. 142–151 : AMIA.

DE MOOR G., SUNDGREN M., KALRA D., SCHMIDT A., DUGAS M., CLAERHOUT B., KARAKOYUN T., OHMANN C., LASTIC P.-Y., AMMOUR N., KUSH R., DUPONT D., CUGGIA M., DANIEL C., THIENPONT G. & COOREVITS P. (2014). Using electronic health records for clinical research : The case of the EHR4CR project. *Journal of Biomedical Informatics*, **53**, 162–173.

JIANG G., PRUD'HOMMEAUX E. & SOLBRIG H. (2017). Developing a semantic web-based framework for executing the clinical quality language using FHIR. In *Proceedings of the International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences (SWAT4LS)*.

JOHANNSEN W. (1911). The genotype conception of heredity. *The American Naturalist*, **45**(531), 129–159.

LOEBE F., STUMPF F., HOEHNDORF R. & HERRE H. (2012). Towards improving phenotype representation in OWL. *Journal of Biomedical Semantics*, **3**(2), 1–17.

MO H., THOMPSON W. K., RASMUSSEN L. V., PACHECO J. A., JIANG G., KIEFER R., ZHU Q., XU J., MONTAGUE E., CARRELL D. S. *et al.* (2015). Desiderata for computable representations of electronic health records-driven phenotype algorithms. *Journal of the American Medical Informatics Association*, **22**(6), 1220–1230.

RICHESSON R. & SMEREK M. (2014). Electronic health records-based phenotyping. https ://sites.duke.edu/rethinkingclinicaltrials/ehr-phenotyping/.

RIVAULT Y., DAMERON O. & LE MEUR N. (2019). queryMed : Semantic web functions for linking pharmacological and medical knowledge to data. *Bioinformatics*, **35**(17), 3203–3205.

SHARMA H., MAO C., ZHANG Y., VATANI H., YAO L., ZHONG Y., RASMUSSEN L., JIANG G., PATHAK J. & LUO Y. (2019). Developing a portable natural language processing based phenotyping system. *BMC Medical Informatics and Decision Making*, **19**(3), 79–87.

UCITELI A., BEGER C., KIRSTEN T., MEINEKE F. A. & HERRE H. (2019). Ontological modelling and reasoning of phenotypes. In *Proceedings of the Joint Ontology Workshops (JOWO)*, volume 2518 : CEUR.

UCITELI A., BEGER C., KIRSTEN T., MEINEKE F. A. & HERRE H. (2020). Ontological representation, classification and data-driven computing of phenotypes. *Journal of Biomedical Semantics*, **11**(1), 1–17.