Models and methods for genome-resolved metagenomics

Benjamin Churcheward

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Nantes Université

THÈSE DE DOCTORAT DE

NANTES UNIVERSITÉ

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Benjamin CHURCHEWARD

Models and methods for genome-resolved metagenomics

Thèse présentée et soutenue à Nantes, le 13 Décembre 2022
Unité de recherche : LS2N - Laboratoire des Sciences du Numérique de Nantes

Rapporteurs avant soutenance :

Lucie BITTNER	Maîtresse de conférences Sorbonne Université
Eric PELLETIER	Directeur de Recherche CEA

Composition du Jury :

Président :	Dominique LAVENIER	Directeur de Recherche CNRS
Examinateurs :	Silvia ACINAS	Associate Professor ICM-CSIC Barcelona
	Mathieu ALMEIDA	Chargé de Recherche INRAE
Dir. de thèse :	Guillaume FERTIN	Professeur Nantes Université, LS2N
Encadrant de thèse :	Samuel CHAFFRON	Chargé de Recherche CNRS, LS2N

Acknowledgement

I do not have the pretention to have come through this journey alone, and the work presented in this document is the result of numerous helps, either directly implicated in this work, or of any indirect nature.
First, and obviously, I deeply thank Samuel and Guillaume, my supervisors, without who this project would not have begun. Thank you two for the time you gave me, for both your human qualities, to have been responsive through this journey, and for your scientific rigor.
I also thank all the ComBi team members which I have encountered during this thesis. I thank Nils Giordano, my former lab roomate, which has helped me a lot through the understanding and handling of many computer aspects. I thank Damien Éveillard for his wide knowledge which has led to some interesting conversations about science reasoning, but also teaching. Thank you to Marko Budinich.
This PhD was also the occasion to give some courses in different computer science courses. I thank Jérémie Bourdon, Géraldine Jean, Christophe Jermann, Evgueny Gurevsky, Stéphane Télétchéa, Mathieu Perrin, and also Guillaume, to this regards.
Even if I had met ASP in my previous academic journey, I was far from being a specialist in this language at the beginning of this project, and I do not think I am yet, but I thank Carito Guziolowski for her help in the understanding of this language, with her lecture I followed at Centrale Nantes and her help at the beginning of the design of the model. I also thank Torsten Schaud and his team embers for their invitation to talk about my work, and their help in the enhancement of the implementation. I thank Flavio Everardo, who took the time to explain me deeplier some aspects of this language during regular meetings.
I deeply thank all the PhD students from ComBi team, and from the different teams in the building of the department of Informatics. There was a friendly atmosphere, catalyzed by the resurrected Login association, allowing for great evenings and events. I thus thank in disorder Anna, Marinna, Émile, Mathieu B., Mathieu V., Rémi, Josselin, Hugo, Thibault, Jolan.
A special thank to Johanna, who began her PhD at the same time than me, for her nat-

ural kindness and her sweet madness. An also deep thank to Albane and Sophie, for our many evenings and parties, and our hikes around Nantes. Thank you to Antoine, for our video games plays and our sometimes passionate discussions about any subject around a beer. Also, a journey that long would not have come without downsides, and this project proves that well.
And, last but not least, a deep thank to my family to have remained supportive during the long years of this journey.

Table of Contents

RÉSUMÉ FRANÇAIS

Depuis leur découverte en 1671 par van Leeuwenhoek, et leur association avec des maladies humaines existantes, l'étude des microbes a représenté un intérêt majeur en biologie. Bénéficiant des avancées technologiques au cours des siècles, la microbiologie a permis de mieux comprendre l'écologie des microbes, mettant en évidence leur ubiquité et le nombre élevé d'organismes présents dans l'environnement, constituant de grandes communautés. Notamment, le fait que la majorité des molécules antimicrobiennes soient issues de composés naturels synthétisés par les microbes du sol [1] souligne l'importance de l'étude à grande échelle des communautés microbiennes. Une caractérisation profonde et précise de ces communautés microbiennes est donc devenue nécessaire. Cette caractérisation a été permise en répondant à trois questions : "Qui est là ?", "Que font-ils ?" et "Qui fait quoi ?" [2]. Le développement du séquençage génomique et de la métagénomique a permis de fournir des méthodes pour répondre à ces trois questions, la reconstruction des génomes des organismes à partir des données métagénomiques, aussi appelés Metagenome-Assembled Genomes (MAGs), ayant pour but de répondre à cette troisième question, et de définir les rôles au sein de la communauté.

L'étude des communautés microbiennes est une discipline assez récente, qui a longtemps souffert de plusieurs limitations techniques. La culture bactérienne étant le seul moyen d'accéder aux microbes, l'étude des bactéries était limitée à une poignée d'organismes, en raison de la faible proportion de taxons bactériens qui se développent bien en culture. Le développement du séquençage du génome dans les années 1970 [3], et sa récente amélioration du débit avec le développement des outils de séquençage de nouvelle génération (NGS), ont permis d'accéder directement au matériel génomique environnemental, créant ainsi un domaine appelé métagénomique [4]. La métagénomique a révolutionné la microbiologie en offrant un moyen direct et plus complet d'étudier les communautés bactériennes environnementales. L'exploration d'une diversité bactérienne jusqu'alors inconnue a permis d'appréhender leur importance dans les processus biogéochimiques globaux [5], ainsi que leurs rôles dans la physiologie des organismes multicellulaires [6]. Les études métagénomiques ont également révélé que les génomes microbiens n'expriment pas la même plasticité [7], que les génomes des organismes multicellulaires eurkaryotes.

Les progrès informatiques ont accompagné l'essor des études métagénomiques avec la capacité d'extraire des génomes individuels à partir de données métagénomiques. La dernière décennie s'est concentrée sur le développement d'outils permettant de regrouper des séquences génomiques appartenant au même génome pour reconstruire des génomes assemblés par métagénome (MAG). Ces outils suivent pour la plupart des protocoles standards, dans lesquels le problème principal est de regrouper les séquences métagénomiques en génomes putatifs en se basant soit sur des références taxonomiques, soit sur des caractéristiques communes inhérentes qui augmentent la probabilité qu'un ensemble de séquences appartiennent au même génome. Les outils de binning ont été régulièrement utilisés dans les études métagénomiques récentes [5, 8, 9], reconstruisant des milliers de génomes microbiens [9, 10], et contribuer à améliorer la description taxonomique des communautés microbiennes de divers environnements, comme l'intestin humain [11], les sols ou les environnements marins [12, 13]. Cependant, les protocoles de reconstruction des MAGs souffrent encore de limitations, telles que leur incapacité à traiter des portions de génomes bactériens, ou leur difficulté à classer les séquences de faible abondance. En raison de ces limitations, les MAGs ont souvent été fragmentés, ont manqué pour des taxons ou des variants rares, et leur qualité n'a pas toujours été évaluée avec précision [14]. De plus, les bases de données MAGs contiennent encore une grande partie des gènes qui n'ont pas été annotés taxonomiquement [15]. Le premier objectif de ce travail était de développer une nouvelle méthode qui permettrait de reconstruire les MAGs d'un plus grand nombre de taxons présents dans les métagénomes. Pour cela, nous avons choisi de nous appuyer sur un paradigme de programmation logique. Avec cette approche, nous nous sommes concentrés sur la description du problème, sa résolution étant prise en charge par le solveur. Nous avons inclus la procédure de regroupement dans un problème de regroupement et d'optimisation, deux types de problèmes couramment résolus par la programmation logique. Le principal avantage du paradigme de la programmation logique est la certitude d'atteindre la solution optimale du problème, si elle existe, et la sortie de toutes les solutions optimales du problème, par rapport à la sortie d'un seul ensemble de reconstruction MAGs avec des outils de binning classiques.

Cette exploration d'un grand nombre de solutions de binning est envisagée comme une bonne réponse aux limitations des outils de binning classiques, qui peuvent souffrir lors de la reconstruction de génomes d'organismes rares, ou pour reconstruire des souches étroitement liées. Pour cela, nous avons développé un modèle de binning de contigs reposant sur l'Answer Set Programming (ASP), un langage de logique déclarative. Nous avons en-

suite comparé les performances du modèle ASP à un outil de binning de la littérature, metabat2 [16]. Le principal résultat de notre approche a été sa capacité à reconstruire avec succès deux souches de génomes étroitement liés, alors que cette discrimination n'a pas été du tout détectée par metabat2, qui a fusionné les souches en un seul MAG. La séparation supposément meilleure permise par notre modèle doit cependant être reconsidérée à travers le cadre très contraint de notre modèle. Notamment, lorsque le modèle ASP ne donnait aucune information sur le nombre de MAG à reconstruire, ses performances ont fortement chuté. Ainsi, une évaluation préalable plus précise du nombre putatif de MAG à reconstruire serait grandement bénéfique à notre modèle ASP, et permettrait de faire ressortir sa meilleure capacité de reconstruction des souches. Si la détermination des souches bactériennes présentes dans les métagénomes représente encore une tâche difficile en métagénomique, elle constitue également l'un de ses objectifs majeurs pour le futur proche, avec plusieurs études récentes se concentrant sur la détermination des souches [15, 17]. La possibilité d'explorer plusieurs solutions au problème de binning pour identifier les souches de génomes et les pan-génomes soulignerait alors davantage la pertinence de l'approche de programmation logique pour résoudre le problème de binning de contigs.
Cependant, l'efficacité globale du modèle de binning ASP pour reconstruire les MAGs s'est révélée être surpassée par l'outil avec lequel nous nous comparons, metabat2. L'ajout de plus de contraintes à notre modèle pourrait augmenter la précision du regroupement des contigs. De plus, la capacité de rejeter certains contigs de l'ensemble de données représente également un avantage majeur de metabat2 par rapport à notre modèle. L'inclusion du rejet des contigs dans notre modèle permettrait alors d'augmenter la précision, en éliminant les contigs provenant de régions génomiques qui diffèrent significativement dans leur composition nucléotidique, et qui peuvent être difficiles à inclure dans un bin génomique. Le principal problème restait cependant le nombre trop élevé de solutions à explorer, qui dépassait la capacité du solveur de fermetures. La conception d'un modèle plus complet, avec l'ajout de plus de contraintes, permettrait de réduire encore l'espace de recherche du solveur. L'autre point principal du développement du cadre ASP a été la réduction significative du temps de calcul permise par l'amélioration de l'implémentation du code. Il y a sans aucun doute encore de la place pour d'autres améliorations techniques, afin d'accélérer la résolution des problèmes. Parmi les améliorations possibles, l'utilisation de propagateurs, qui sont des logiciels écrits dans des langages impératifs recouvrant le programme ASP, devrait être envisagée. Ils peuvent traiter plus facilement des procédures qui pourraient être très coûteuses dans le processus de résolution effectué par le solveur

ASP, facilitant ainsi la résolution du problème. Leur utilisation limitée dans ce travail n'a pas permis une amélioration significative du processus de résolution. D'autres approches, telles que le développement d'un ASP à programmation par contraintes (CASP), incluraient des caractéristiques qui sont plus facilement exécutées dans la résolution complète de la PC que dans l'ASP. Notamment, le CASP supprimerait la nécessité de lister toutes les solutions possibles [18], ce qui représenterait une amélioration majeure de la technique ASP elle-même. Le développement de telles approches constitue cependant une discipline assez récente, avec des applications actuellement limitées, mais a montré des résultats prometteurs [18, 19].

Le second objectif de ce travail de thèse était de développer un nouveau workflow pour reconstruire les MAGs. L'extraction de connaissances à partir de données métagénomiques brutes nécessite de traiter plusieurs tâches spécifiques, de l'assemblage à l'appel de gènes et à l'annotation, chacune d'entre elles étant souvent réalisée à l'aide de logiciels dédiés. Le choix, la configuration et l'utilisation de ces différents outils peuvent alors représenter une tâche difficile pour l'utilisateur. Récemment, plusieurs workflows de métagénomique ont été développés [20–23], utilisant souvent des paramètres spécifiques par défaut pour chaque logiciel intégré. Cependant, ces flux de travail souffrent généralement de limites, soit vers l'étape d'assemblage, soit vers l'étape de regroupement des génomes. Notamment, la question de savoir comment réaliser le co-assemblage de plusieurs métagénomes ensemble a été peu explorée dans la littérature. Nous avons donc développé un flux de travail intégrant un module entièrement automatisé (module de novo) pour déterminer quels métagénomes devraient être co-assemblés, afin d'augmenter l'efficacité du co-assemblage. Ce module a été intégré dans le workflow de reconstruction MAGs nommé MAGNETO. Ce flux de travail permet également aux utilisateurs de configurer soit l'assemblage, soit l'étape de regroupement, présentant ainsi quatre stratégies différentes de reconstruction MAGs. Une comparaison entre ces quatre stratégies de reconstruction a été effectuée.

Le co-assemblage dans la reconstruction des MAGs a en effet été largement utilisé, en raison de son avantage à augmenter l'abondance des variants rares présents dans les communautés microbiennes. Cependant, l'utilisation du co-assemblage s'est révélée être une stratégie du "tout ou rien", avec de nombreuses études qui co-assemblent systématiquement tous les métagénomes dans leurs ensembles de données. En raison de l'augmentation de la consommation de ressources informatiques, le co-assemblage de tous les métagénomes en une seule fois peut ne pas être possible, en particulier lorsque les métagénomes proviennent de communautés bactériennes complexes telles que les communautés marines. Cer-

taines études choisissent donc d'effectuer plusieurs co-assemblages de sous-ensembles de métagénomes de leur jeu de données. Cette pratique soulève cependant la question du choix des ensembles de métagénomes. Les études réalisant des co-assemblages de sous-ensembles de métagénomes marins ont à cette fin utilisé la localisation géographique [5, 10]. Les connaissances a priori nécessaires pour déterminer les ensembles à co-assembler ne sont pas toujours disponibles ou pertinentes, ce qui constitue la raison d'être de notre travail. Les ensembles de métagénomes extraits de la matrice de distance métagénomique (MD) ont permis de rassembler des métagénomes suivant des caractéristiques environnementales communes, comme la température. Cette observation a souligné la pertinence de notre approche MD, car l'effet de la température dans le façonnement des communautés bactériennes a déjà été démontré [8]. Cette approche MD a également permis de reconstruire plus de MAGs et avec une meilleure qualité que l'approche reposant sur la localisation géographique. Des précisions sont toutefois nécessaires, notamment en raison de l'apparente contradiction entre deux mesures de qualité différentes. Une explication possible serait que le co-assemblage de métagénomes étroitement liés facilite la reconstruction des parties accessoires du pan-génome, tout en détériorant la reconstruction des parties centrales du génome.

La comparaison des quatre stratégies de reconstruction des MAGs a révélé un effet positif important du calcul de la couverture différentielle parmi un grand nombre de métagénomes. L'effet de la couverture différentielle a été mesuré comme étant plus efficace que l'effet du co-assemblage, car l'assemblage simple combiné au co-binage pourrait surpasser les stratégies de co-assemblage. Cette observation peut refléter le fait que les effets délétères du co-assemblage, à savoir la probabilité plus élevée de produire des MAG fragmentés, ne peuvent être surmontés qu'avec une couverture différentielle calculée sur un nombre suffisamment élevé de métagénomes. Cela concerne essentiellement les communautés complexes, car ces observations ont été faites sur des MAG reconstruits à partir d'ensembles de données sur le microbiome intestinal humain, qui peuvent contenir plusieurs souches et variantes.

Le calcul de la couverture différentielle sur tous les métagénomes disponibles peut cependant ne pas être nécessaire, et s'est révélé coûteux, en raison du nombre quadratique d'opérations à effectuer. Une amélioration de cette approche pourrait consister à évaluer le nombre optimal d'échantillons sur lesquels calculer la couverture différentielle. Une telle évaluation a déjà été faite dans le passé [24], mais il serait intéressant de savoir si ce nombre optimal dépendrait de la complexité de la communauté bactérienne considérée [24].

L'implémentation de MAGNETO permet l'ajout de modules supplémentaires, dont certains sont déjà considérés, afin de réaliser une analyse plus complète des MAGs. Notamment, un module estimant la croissance optimale des MAGs basée sur leur composition nucléotidique a été développé et a déjà contribué à l'étude des MAGs arctiques [25]. D'autres ajouts concerneraient le logiciel d'assemblage et les outils de binning, afin d'augmenter la flexibilité de l'utilisateur. Par ailleurs, le protocole de regroupement combinant plusieurs outils pour récupérer des MAGs en mosaïque a donné des résultats prometteurs [26, 27], et pourrait constituer un moyen facile d'améliorer l'efficacité du regroupement. Notre flux de travail suit un protocole de reconstruction MAGs qui peut être caractérisé comme classique, convoquant des outils qui ont été largement utilisés dans les études métagénomiques récentes. A cet égard, il souffre des limites de ces outils, et l'obstacle de la reconstruction des régions accessoires du pan-génome est resté sans réponse. Ces dernières années, plusieurs études ont visé une meilleure caractérisation des pan-génomes dans les études métagénomiques [28–30]. Ces travaux seraient une source d'inspiration pour permettre une analyse plus complète des communautés bactériennes dans MAGNETO.

INTRODUCTION

Since their discovery in 1671 by van Leeuwenhoek, and their association with existing human diseases, the study of microbes has represented a major interest in biology. Benefitting from technological advances through centuries, microbiology has allowed to better understand the ecology of microbes, highlighting their ubiquitousness and the high number of organisms present in the environment, constituting large communities. Notably, the fact that the majority of antimicrobial molecules are derived from natural compounds synthetized by microbes from the soil [1] highlights the importance of large-scale study of microbial communities. A deep and accurate characterization of these microbial communities has become thus necessary. This characterization has been allowed by answering three questions: "Who is there ?", "What are they doing ?" and "Who is doing what ?" [2]. The development of genome sequencing and metagenomics has allowed to provide methods to answering those three questions, the reconstruction of organisms' genomes through metagenomic data, also called Metagenome-Assembled Genomes (MAGs), constituting the approach aiming to determine the different roles within the community.

The study of microbial communities is a rather recent discipline, which has suffered for a long time from several technical limitations. With the bacterial culture as the only way to access to microbes, the study of bacteria were limited to a handful of organisms, because of the tiny proportion of bacteria taxa that develop well in culture. The development of genome sequencing in the 1970s [3], and its recent improvement of throughput with the development of Next-Generation Sequencing (NGS) tools, have led the opportunity to directly access environmental genomic material, creating a field called metagenomics [4]. Metagenomics has revolutionized microbiology by giving a direct and more comprehensive way to study bacterial environmental communities. The exploration of a hitherto unknown bacterial diversity has allowed to apprehend their importance in global biogeochemical processes [5], as well as their roles in the physiology of multicellular organisms [6]. Metagenomics studies have also revealed that the microbial genomes do not express the same plasticity [7], than the genomes of multicellular eurkaryotic organisms. A more complete description of the development of metagenomics, and the discoveries allowed by metagenomic studies, are presented in the Chapter 1 of this thesis.

Computational advances have accompanied the rise of metagenomic studies with the capacity to retrieve individual genomes from metagenomic data. The last decade has focused on the development of tools allowing to bin genomic sequences belonging to the same genome to reconstruct Metagenome-Assembled Genomes (MAGs). These tools mostly follow standard protocols, in which the main problem is to cluster metagenomic sequences into putative genomes relying either on taxonomic references, or on inherent common features which increase the likelihood for a set of sequences to belong to the same genome. Binning tools have been routinely used in recent metagenomic studies [5, 8, 9], reconstructing thousands of microbial genomes [9, 10], and contributing to enhance the taxonomic description of microbial communities from diverse environments, such as the human gut [11], soils or marine environments [12, 13]. However, MAGs reconstruction protocols still suffer from limitations, such as their inability to handle portions of bacterial genomes, or its difficulty to bin sequences with low abundance. Because of these limitations, MAGs have often been fragmented, have been missing for rare taxons or variants, and their quality has not been always accurately assessed [14]. Moreover, MAGs databases still contain a large portion of genes which have not been taxonomically annotated [15]. A more complete description and a review of the MAGs reconstruction process is the object of the Chapter 2.

The first objective of this work was to develop a new method that would allow to reconstruct MAGs from more taxa present in metagenomes. For that purpose, we choose to rely on a logic programming paradigm. With this approach, our main focus was the description of the problem, with its resolution being handled by the solver. We included the binning procedure into a clustering and optimization problem, both of which being kinds of problems routinely resolved in logic programming. The main advantage of the logic programming paradigm is the certainty to reach the optimal solution of the problem, if it exists, and the output of all optimal solutions of the problem, compared to the output of only one MAGs reconstruction set with classic binning tools. This work, combined with a description of logic programming, is presented in details in Chapter 3.

The second objective of this thesis work was to develop a new workflow to reconstruct MAGs. Extracting knowledge from raw metagenomics data requires to handle several specific tasks, from assembly to gene calling and annotation, each of them often performed using dedicated software. The choice, configuration and the use of these different tools may then represent a difficult task for the user. Recently, several metagenomics workflows have been developed [20–23], often using specific default parameters for each

integrated software. However, these workflows usually suffer from limits towards either the assembly step or the genome binning step. Notably, the question of how to perform the co-assembly of several metagenomes together was poorly explored in the literature. We thus developed a workflow integrating a fully-automated, *de novo* module to determine which metagenomes should be co-assembled together, in order to increase the efficiency of co-assembly. This module was integrated in the MAGs reconstruction workflow named MAGNETO. This workflow also allows the users to configure either the assembly and the binning step, exhibiting four different strategies of MAGs reconstruction. A comparison between these four reconstruction strategies was performed. The development of MAGNETO is presented in Chapter 4 and has led to the publication of an article in an international journal [31].

Chapter 1

From genomics to metagenomics

Preamble

In this chapter we will introduce metagenomics, a recent sub-field of microbiology, to understand the context of its development. We will then present in which context metagenomics appeared, how it has answered the main limitations of environmental microbiology, and the several discoveries it has allowed in recent years.

1.1 The study of microbial communities *in situ*

1.1.1 The great plate count anomaly

The development of bacterial culture by Koch during the 19th century had allowed considerable advances in the study of microbes, notably pathogenic bacteria. Direct observations based on microscope observations were later evidence of this limitation: in 1932, Razumov counted several orders of magnitude of difference between the number of organisms he could count by directly observing water samples under the microscope, and organisms present in culture [32]. This difference was later confirmed with the work of Staley and Konopka [33] which named "great plate count anomaly" this observed discrepancy. The filtering effect of the bacterial culture in the description of the microbial diversity became the next barrier to overcome in environmental microbial genomics. DNA sequencing was developed in 1977 by Sanger [3]. Considered as a revolution for genomics study at its time, its high efficiency, able to sequence several hundreds of bases in a single day, the Sanger method (fig. 1.1) helped to sequence many reference organisms, beginning with the phage ϕ X-174 as early as 1977 [3]. However, due to the limits of the technology at this time, pessimistic Staley and Konopka stated that "no breakthrough in determining species diversity seems likely in the near future" [33], and the assessment of microbial

diversity from environmental samples seemed to face a deadlock. However, the recent development of bio-molecular techniques to directly sequence genomic material, combined with the work of the Woese group, which identified the 16S ribosomal RNA (RNA) gene as a marker molecule to estimate microbial diversity, helped to resolve this issue. 16S RNA genes are universal in Prokaryotes and present in multiple copies located in hypervariable genomic regions, and as such, are easily recognizable, constituting primary targets to gather information about organisms.

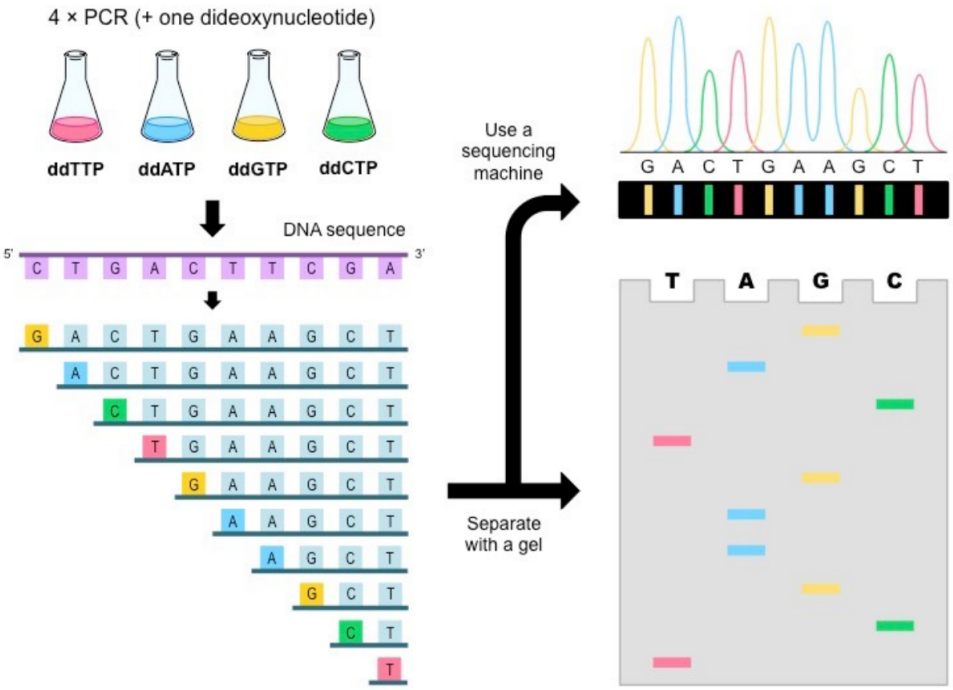

Figure 1.1 **Sanger sequencing method.**
After DNA extraction, preparation of library and an amplification phase, sequencing is performed. On the left, the de-deoxy-triphosphate nucleosides (ddTNP) stops the strand replication once incorporated in the mix. As each ddTNP is linked with a specific fluorescent marker, the observer may determine which nucleotide was added in the final position of a sequence. On the right, synthesized fragments are then disposed on a polyacrylamide gel, and separated through electrophoresis, allowing to read the order of nucleotides of the sequence. Source: https://microbenotes.com/dna-sequencing-maxam-gilbert-and-sanger-dideoxy-method

1.1.2 A first description of microbial communities

Benefiting from the development of DNA sequencing, Carl Woese and others started to analyse and sequence 16S RNA genes of various bacteria as early as the late 1970s, using it as a marker for phylogeny, enabling the discrimination at molecular level of the three domains of the tree of life [34]. The invention of the Polymerase Chain Reaction (PCR) and the automation of DNA sequencing allowed a growing number of 16S RNA (and Eukaryotic 18S RNA) studies, leading to the accumulation of RNA sequences belonging to a wide spectrum of living organisms in databases. Comparison of these sequences allowed to understand that the RNA gene sequences are highly conserved within living organisms of the same genus and species, but that they differ between organisms belonging to different genera and species. This observation has made taxonomic assignments using 16S RNA very straightforward [35], thus recognising 16S RNA gene for taxonomic profiling of microbial communities, initiating a sub-field of microbial ecology called "molecular ecology". In the 1990s, molecular ecology studies then proceeded to explore diversity in microbial communities belonging to different environments, such as oceans [36–38], hot springs [39, 40] and soils [41]. These studies allowed the discovery of several hitherto unknown phylogenetic lineages, such as the SAR11 cluster, first discovered through a study of the bacterioplankton of the Sargasso Sea [42], followed one year later by the discovery of new gene groups belonging to proteobacteriae lineages unrelated to any reference genomes [36, 37]. Comparison between the composition of two marine communities belonging to two different oceanic regions also helped to understand the global distribution of bacterioplankton [37]. It also revealed the widespread occurrence at the surface of coastal marine environments of Archaeal species [38], a group that was previously thought to colonize preferentially environments facing extreme physicochemical characteristics. Similarly, eight novel uncultured bacterias were discovered in a hot spring habitat [39], also the study of an acidic soil located in Australian rainforest revealed new bacterial taxa, some of which could be related to known reference lineages, and others completely unrelated to reference genomes [41]. These first studies drove the environmental microbiology toward its first purpose, the description of previously unknown, uncultured microbial lineages in their environments. While the order of magnitude of the unknown proportion of global bacterial diversity remained imprecisely assessed, these studies confirmed than the diversity known from cultured reference lineages represented only a tiny proportion of the global microbial diversity.

As these studies were pioneers in their domain, they also focused on highly constrained

environments: the Sargasso Sea, for instance, is known for its high limitations in nutriments supply [42], while the low pH of rainforest's soil represents a hostile characteristics for most common taxa [41]. The communities living in these environments have often a lower biodiversity than average, which means a lower number of taxa, and relationships easier to model.
However the RNA gene approach has also limitations to catch the whole environmental microbial diversity, as it may ignore distant lineages such as those belonging to viruses taxa. The design of the primers itself may also be biased towards specific taxa, at the expense of the exclusion of others [43]. They were thus not able to predict the physiological attributes of newly described phylotypes with the same efficiency as with already well-known phylotypes. This is partly due to the wide array of physiological and metabolic diversity encompassed within all phylogenetic lineages. The impossibility or at least the difficulty to enrich and isolate in culture these newly phylotypes did represent an additional impediment to that prediction.

1.1.3 Accessing a greater portion of the genome

In 1992, Shizuya and colleagues succeeded to clone a human genomic fragment 300 kilobases long into a bacterial artificial chromosome (BAC) inserted into a bacterial host, and to maintain the clone stable through a long serial growth of the bacteria [44]. BACs offer advantages to sequence uncultured microbes' DNA, as it was easier to control the growth of the vector organism, which was most of the time *Escherichia coli*, a model organism whose culture is well known and completely harnessed [45]. They can carry large portions of genes, with an insert size typically ranging from 150 to 300 kb. Metagenomic studies quickly used this technology too, in order to access a wider proportion of the microbial genomes than when relying on a few marker genes alone, constituting an environmental DNA library [46]. These microbial studies, rather than being limited to the pure description of the diversity of the studied environment, could then analyze sets of genes present in microbial communities [47, 48]. While studying the genetic content of a soil microbial community, Handelsman and colleagues conceptualized the functional analysis of the collective genomes belonging to a bacterial community, and its interdependent metabolic pathways, as an entity, and named it a "metagenome" [4]. To better describe the genomic analysis of uncultured microorganisms, Schloss and Handelsman later proposed to name this field of study "metagenomics" [49].

1.1.4 The contribution of high-throughput sequencing

If the Sanger sequencing had allowed to determine precisely the bases composition of nucleic acid molecules, contributing to the development of molecular biology as a field of study, it quickly showed its own limits. The laborious library preparation, the limited throughput of the method, impeded the efficiency of studies based on this technology.

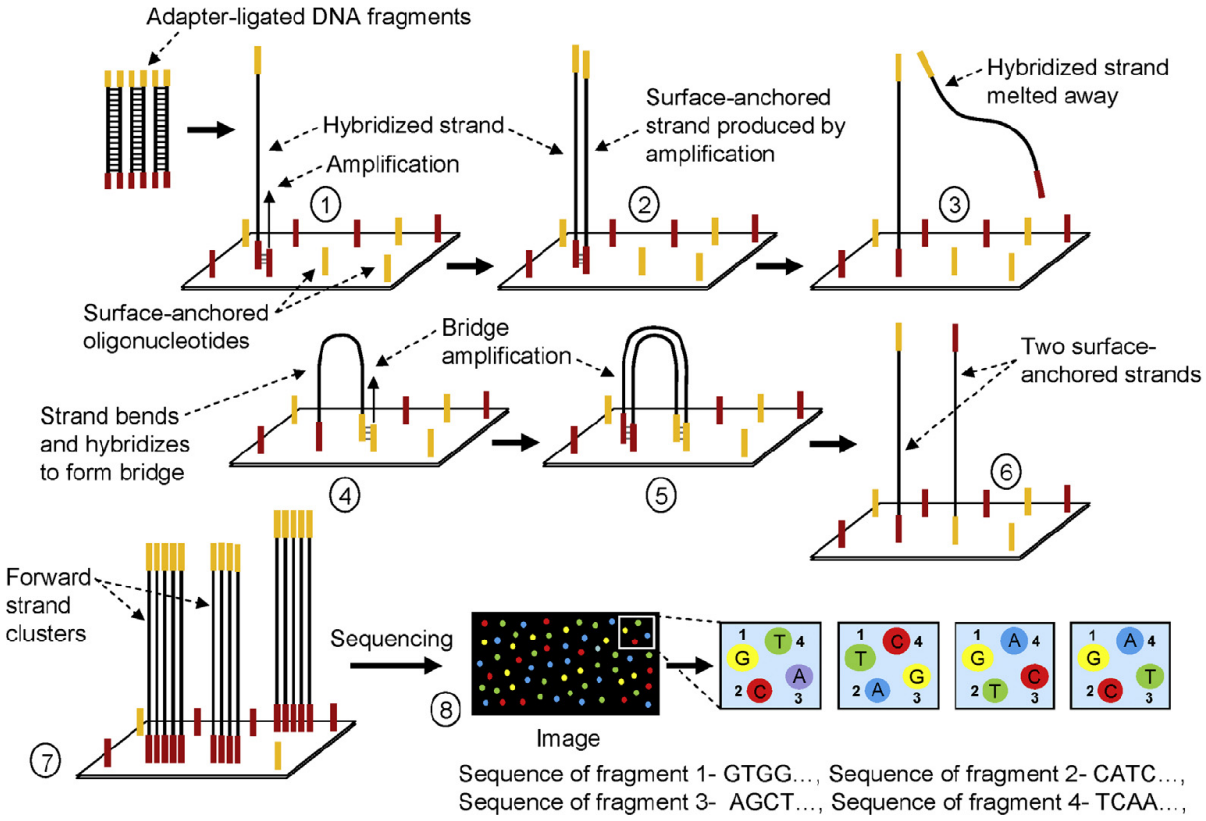

Figure 1.2 **Illumina sequencing.**
1) Hybridization step: The DNA strands are bound to the adaptors; 2) Synthesis of the complementary strand; 3) The hybridized stand is evacuated; 4) The synthesized strand is hybridized with an other anchor to form a bridge; 5) The bridge is then amplified, *i.e.*, the complement strand is synthesised; 6) The bridge is then relaxed: the two strands are still attached to their anchors; 7) Repeat to further amplify the DNA strands; 8) Each nucleotide being associated to a fluorescent molecule, whenever a nucleotide is incorporated during the elongation of strand, a specific light is emitted, which allows to read the DNA sequence. Source: [50]

The development of high-throughput sequencing technologies, also called Next-Generation Sequencing (NGS) technologies, during the 2000s, has revolutionized genomics research.

The main difference between the old electrophoretic methods and these new ones were the massive parallelization of the sequencing process, with not only one tube per reaction, but a complex DNA templates library, densely disposed on a solid support. Besides, these approaches also performed the amplification step *in situ*, directly on the sequencing support, facilitating the library preparation. At the end of this process, the total volume of sequences produced is several orders of magnitude above Sanger method [51, 52]. Also, contrary to the previous decades during which Sanger remained the only sequencing approach, the second generation of DNA sequencing saw the concurrence of several companies and the development of different protocols and technologies [53]. However, several of these approaches disappeared, and currently the Illumina sequencing protocol (fig. 1.2) represents the dominant approach of this generation.
Due to their highly-parallel processing, the second generation of sequencing technologies has dramatically increased the amount of genomic data produced in the 2010s [52]. Compared to the Sanger sequencing era, sequencing a genome quickly became a routine, and the second generation approaches allowed an enhancement in the rhythm of sequencing new reference genomes. They have also allowed a sharp reduction of the sequencing costs in the last 20 years, plummeting even faster than Moore's law since 2008. The price to sequence a human genome dropped with a four-orders of magnitude between 2008 and 2011 (fig. 1.3). Although the utility of metagenomics to observe the prokaryotic world *in situ* had already become apparent, and thus the number of metagenomic studies already began to grow, the development of NGS strongly contributed to increase their application. As a result, metagenomic analysis of complex environmental samples became affordable even for small laboratories, leading to a sharp increase of the number of metagenomic studies: the number of published papers in the field has grown from one in 1998 to several thousands today.

1.2 Metagenomics and its discoveries

1.2.1 The possibilities of metagenomics: a better exploration of uncultiv ale lineages

The term "biological dark matter" [7, 54, 55] has been proposed as a descriptor of a wide prokaryotic world which dominates the biosphere, by analogy with the dark matter substance in astrophysics. Indeed, prokaryotes represent about half of the living biomass [56],

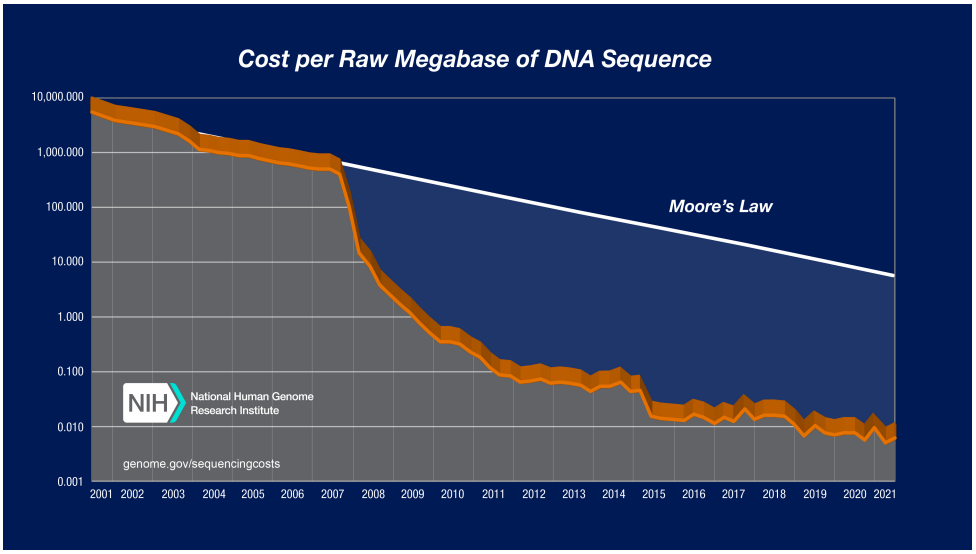

Figure 1.3 **Sequencing cost per raw Mb.**
After 2007, sequencing cost plummeted due to the arrival of high-throughput sequencing technologies such as the Illumina Miseq in laboratories, drifting sensibly below the degrowth expected if it would follow the Moore's Law. Source: https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data.

excluding viruses. In order to better describe this whole hitherto ill-known biodiversity, and benefiting from the development of bio-informatics, metagenomics studies described more and more complex communities, from acidic mine drainage [47, 57], geothermal hot springs [58], followed with human gut [59], rhizosphere [60] and oceanic samples [5, 61, 62]. They helped to better characterize the gene content, the diversity and the relative abundance of environmental microbes [48]. They gave better insights about niche-specific adaptations, like the description of pathways for nitrogen fixation, and iron-oxidizing process to obtain energy, in acid mine drainage biofilms, as well as the survival strategies adopted to resist to low pH [47].

Studies focusing on the gut microbiome, more specifically in human, helped to better understand the important role played in the microbiome in several pathways. They notably uncovered that several pathologies may be related with specific bacterial composition, such as inflammatory bowel disease [11], obesity [63], diabetes [64], malnutrition [65] or diarrhea [66], but also to pathologies which would *a priori* not be related to the intestinal tractus, such as kidney disease [67], cardiovascular disorders [68] or neurodevelopmental troubles [69].

The exploration of the bacterial diversity in oceanic communities helped to better con-

sider the role of the ocean in key global biogeochemical processes. It led to a better understanding of the diazotrophs organisms, and notably that this function could be expressed by previously unsuspected bacterial lineages [5]. It also revealed the importance of Ammonia-Oxidizing Archaea [70] and revealed the presence of photosynthesis genes within viral genomes [71].
Another question that metagenomics could help to answer was to understand what is the predominant force structuring microbial communities' composition. A tenet, known as the Bass Becking's hypothesis, stated that "everything is everywhere, but the environment selects" [72], underlining the capacity of dispersal of microbes, but that the organisms which proliferate are the most fitted to the specific conditions they face. The analysis of 339 metagenomes, from different environments, based on sequence similarity networks showed that similarities in metagenomes' composition is more explained by similarities in environmental conditions [73]. The effect of temperature had already been observed as a key driver of the composition of oceanic bacterial communities [8]. The role of the geographical distance was however underlined recently, with the observation that genetic distance between communities is also driven by oceanic currents [74].

1.2.2 From genomes to pan-genomes

Bacteria exhibit a remarkable and adaptive plasticity of their genomic content, compared to the stability of the genomic content observed amongst multicellular eukaryotes organisms [7]. During early whole genome sequencing (WGS) efforts, the re-sequencing of specific species led to the discovery of entirely new, previously undetected genes. This observation led to the concept of a genomic part that would be shared amongst all individuals belonging to a particular taxon, the *core genome*, opposed to a set of genes present in only some individuals, the *accessory genome*. These two components form a taxon's *pan-genome* [75] (fig. 1.4).
The core genome represents the set of genes that are mandatory for the organism's survival, and is representative of the phylogenetic lineage of the organism. Functions present in the core genome are thus related to housekeeping functions, construction of the cell envelope, regulatory roles, transport and binding proteins [76]. The accessory genome, on the other hand, contains key genes to face the conditions of a specific environment, notably genes related to defense mechanisms, such as antibiotic resistance [76].
In 2002, a comparison between the genomes of three pathogenic *E. coli* strains found that only 39.2% of the total set of proteins are shared amongst the three strains [77]. This

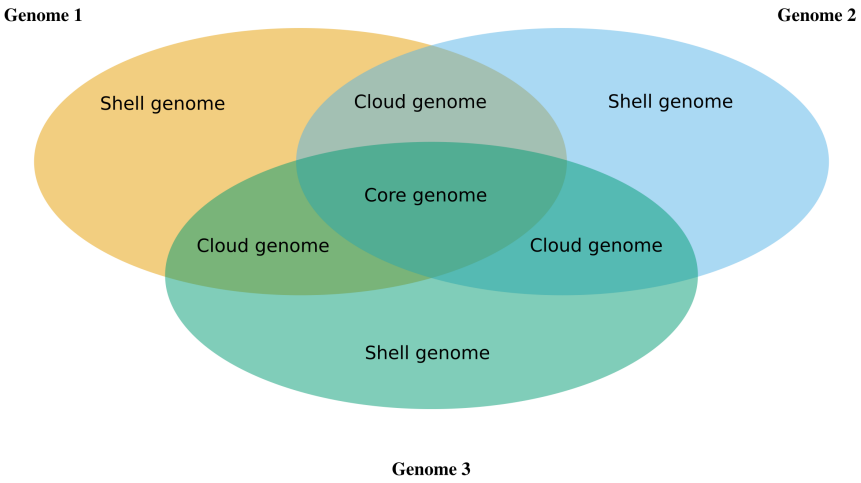

Figure 1.4 **Pan-genome and its components.**
A schematic representation of the pan-genome shared between three genomes, for instance three strains from the same species. The core genome represents the set of genes that are present in every genomes considered, the cloud genome represents genes that are shared by some genomes, but not all of them, while genes which are unique to one genome compose the shell genome. The cloud and the shell genomes are both sub-parts of the accessory genome.

shared proportion was then assessed to be even lower when more strains were studied together: a study gathering 61 strains of *E. coli* and *Shigella* strains found that only 6% of the gene families were shared between all strains, and that the proportion of the accessory genome could represent 80% of one individual's genome [78].

Open versus closed pan-genome

A taxon for which the re-sequencing of new individuals leads to the discovery of new variants with their new genes is said to have an *open* pan-genome. It means that the pan-genome of the taxon may greatly exceed the genome size of one organism, resulting in an extremely versatile gene content. On the contrary, a *closed* pan-genome is characterized by a more predominant core genome: as such, re-sequencing several genomes of the taxon might not identify a significant number of new genes.

Recently, several approaches have emerged to combine pan-genomics and metagenomics in

order to characterize strain-level variation of microbial communities [79–81]. Metagenomic data has also been used to retrieve pan-genomes, such as the strategy used in [28], which first identified gene clusters from isolated genomes or existing genomes in the database, and deduces the relatedness between genomes to construct pan-genomes. It then assesses the abundance and prevalence of the isolated microbial genes by performing reads mapping against metagenomic data. This strategy allows to identify subsidiary categories of environment-related genes, discriminating which sets of genes are or are not mandatorily related to one environment [28]. A *de novo* approach has also been developed, reconstructing pan-genomes based on co-abundances of genes found in metagenomics samples [29]. The pangenome of a species retrieved through both metagenomic and genomic data has sometimes been called a *meta-pangenome*, representing the complete set of genes expressed by a given species in an environment [30]. The collection of meta-pangenomes of all organisms detected in a specific environment has thus been named the *pan-metagenome*, which would represent the global genetic landscape of this given environment [30].

1.2.3 Redefining biological concepts

Another major realization revealed by the study of the biological dark matter was the differences of bacterias' biology with the biology of the multicellular organisms. An example is the determination of species: species has been historically determined based on morphological differences between individuals, or by reproductive isolation. However, at the microbial level, these determinants become irrelevant, as morphological differences are often shallow, and large-scale horizontal gene transfers (HGT) allow exchanges of wide portion of genomes between organisms belonging to different phyla. The impossibility for the multicellular organisms' species definition to be applied to microbes led to new definitions of this concept, based on molecular markers. Although several definitions of bacterial species have been proposed [82, 83], it became related to sequence identity. More notably, the whole-sequence *average nucleotidic identity* (ANI), has been widely accepted as a robust metric to discriminate through species [84, 85], with the threshold of 95% ANI being able to catch the majority of known species [86, 87].
The discovery of environmental microbial genomes missing considered-mandatory genes, enhanced the role of cooperation and communication in microbial ecology [88], and undermined the idea of completely independent individuals surviving on their own [89]. The fact that microbes rely heavily on other members of their community to survive may also be an answer to the problems of the great plate count anomaly and the low percentage

of bacterial species that develop in mono-specific culture [90].

Losing genes to increase fitness

Contrary to previous assumptions in which genomes lose genes only because of genetic drift, it has been proposed that the loss of genes within genomes of a microbial community would have been driven by a positive selection force. Carrying less genes may represent a benefit for bacteria, loosening the energy cost needed for its own management [91]. The then impotent bacteria would thus have been forced to cooperate with the other members of the community to maintain a complete metabolism [92]. This assumption is called the Black Queen Hypothesis [93, 94].

Studies focusing on gut microbiome brought this profound re-evaluation of the concept of individuality, to the multicellular eukaryotes. Microbiomes can then represent a considerable number of microbes, estimated for a typical human being to ten times the number of human cells. The important roles played by the microbiome in several pathways, as well as the intimately intertwined connections with host's cells, deeply impact the fitness of the host and its interactions with its environment. Thus, it became more relevant to consider the holobiont composed of the multicellular organism and its microbiome [6, 95] as the relevant evolutionary unit of interest.

1.2.4 Functional metagenomics

Metagenomic analysis targeting total DNA isolated from the environment may be performed using a different strategy, called functional metagenomics, which evaluates metabolic activities of interest [96]. This approach relies on cloning random fragments of community DNA in large insert vectors, similar to those used at the beginning of shotgun metagenomics, to generate an expression library. This library is then screened, looking for a target reaction with a specific substrate (fig. 1.5). Functional metagenomics allowed the identification of several functions, and also to identify genes related to these functions. This approach was for instance used to identify genes encoding for antibiotics resistance [97]. They were also applied to the characterization of genes encoding enzymes with a particular activity, which represents completely novel sequence types [98]. The limitations of this

approach come from the method itself, as the available screening systems only identify a limited range of functional activities. Besides, genes belonging to phylogenetically distant organisms with the host are susceptible to show a limited level of expression in the cloning host, a standard host being *E. coli* [99]. Lastly, the expression libraries have a limited size: for example a library containing 50,000 clones with an average insert size of 40 kbp, equivalent to 500 bacterial genomes with a size of 4 Mbp [100]. These size order of magnitude are far below metagenomics based on sequencing genomes, in which 1g of soil might contain in comparison several thousands of different microbial species [99]. To overcome the

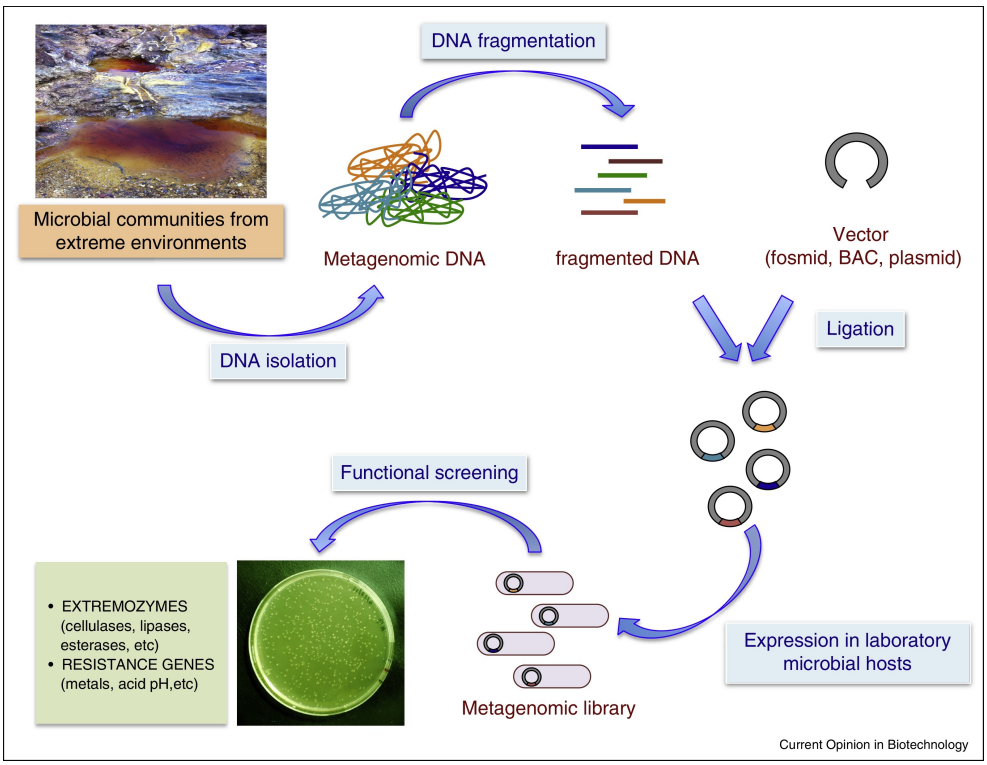

Figure 1.5 **Preparation of a library for functional metagenomics.**
Environmental DNA can be isolated from an environment using a DNA extraction method, obtaining sufficient DNA quantity and an appropriate average insert size. After isolation, the DNA molecule is fragmented, then the fragments are ligated into a linearised cloning vector, and the resulting library of recombinant vectors is inserted into a modified microbial host. Amplification of the metagenomic library will allow its use in different types of screenings. Source: [101]

size limits imposed by the expression library, and thus increasing the number of genes to consider through functional analyses, alternative approaches have been proposed, such as metatranscriptomics. Metatranscriptomics, named after metagenomics, involves the ran-

dom sequencing of microbial community mRNA [102]. It thus reflects the actual functions expressed by a community at a given time. This approach has been used to successfully identify antiobiotic resistance genes in gut microbiome [103] or interactions between gut microbiome and host's immune system [104]. Other approaches include the analysis of proteins expressed at a given time by the bacterial community, the metaproteomics [105]. The identification of the proteins is performed through a coupling of mass spectrometry, and the use of algorithms to precisely identify amino acids composing the peptidic chain. Metaproteomics has been proposed to be used as a complement of functional metagenomics [1]. The analysis of more complex metabolites with metabolomics also allowed to observe the metabolism of a drug applied to gut microbiome [106]. Lastly, systems biology represents an approach that aims to link several "-omics" methods, in order to modelize complete metabolic pathways and interactions occurring within a community [107, 108].

1.2.5 Towards genome-resolved metagenomics

After allowing a better description of the processes underlying the structure of bacterial communities, identifying the members of the community ("Who is there ?") and the function the community provides ("What are they doing ?"), another question was yet to be answered : "Who is doing what ?" [2]. Answering this question requires to link an organism to a function of the community. In 2004 already, two metagenomic studies successfully reconstructed individual bacterial genomes [47, 48], allowing to identify the *Leptospirum* genus as an essential taxon involved in iron oxidation process [47], and highlighting the need for a high sequence coverage to avoid a high fragmentation of the reconstructed genomes [48]. In 2013, Albertsen et al. developed an approach based on the differential abundance of genomic fragments to discriminate the sequences belonging to the same chromosome [109]. These individual genomes reconstructed from metagenomic data have been coined Metagenome-Assembled Genomes (MAGs), and are currently an important representation of uncultured microbial genomes. The reconstruction of MAGs represents a major achievement in metagenomics, allowing a better integration of taxonomic and functional information [5]. MAGs have also been used in several metagenomic studies to better identify genomes of numerous previously unknown species [9, 10, 110]. MAGs also gave the advantage to give an insight about gene sets present within a given organism or taxon, thus providing information to better identifying genetic mobility and potential metabolic interactions between taxa [5, 111].

1.2.6 Current limits

A current major challenge in metagenomics is to determine the phylogenetic origin of anonymous, unreferenced genome fragments. As metagenomics is still a recent discipline, the under-sampling of the microbial dark matter has caused a lack of reference genomes, which has hampered the ability to classify metagenomic fragments. It has been estimated that 40 to 60 percent of all the genes newly discovered from metagenomic studies still cannot be assigned to a known function [15]. This uncharacterised proportion of the genomes is generally not included in downstream analyses, constraining their results to conserved pathways and housekeeping functions [112]. This inability to describe such substantial proportion of the genomes still represents an impediment to the characterisation of species in microbiology.

1.3 Conclusion

Microbiology has seen a dramatic increase in its analytic power in the last half century, allowing an ever increasing characterization of environmental biodiversity, giving birth to metagenomics. Metagenomics really emerged and established with the development of high-throughput sequencing technologies, and benefited from the advances made in several sub-fields. Thus, metagenomic studies generally combine multiple approaches (fig. **??**) to obtain their data. Currently, metagenomics studies are routinely performed, and continue the description of communities whose the description remain uncomplete [9, 10]. They also help a better understanding of the deep complexity of bacterial ecology [87, 90, 113, 114], with several potential applications in the future. The perspectives metagenomics has opened in human health research allow to foreshadow optimistic future development for the field, notably regarding medical diagnosis [113, 115]. The task to perform is still significant: the gathered information in databases is still fragmented, and a large set of genes remain without taxonomic and functional annotation [15].
With the development of protocols enabling to reconstruct individual genomes from metagenomic data, a new facet of metagenomics these recent years. Genome-resolved metagenomics has allowed to isolate numerous genomes of previously unknown taxa [9, 10], filling the gap in the study of environmental bacterial communities. These individual genomes, or MAGs, also help to better understand the relationships between members of a community, identifying the genomes carrying specific genes and intervening in specific pathways [5] and used as a mean to reconstruct pangenomes [30]. The next chapter will

be dedicated to describe computational developments and methods applied to reconstruct MAGs.

Chapter 2

Genome-resolved metagenomics

Preamble

In this chapter we will focus on computational methods to reconstruct MAGs from metagenomes, and essential developments enabling genome-resolved metagenomics. The reconstruction process of MAGs from metagenomes can be divided into two main steps: the assembly, which allows to obtain sequences longer than reads, called contigs, and the binning of those contigs into putative genomes. The binning of contigs, which can be considered as the major part of the reconstruction of MAGs, exploit key features of genomes in order to cluster the assembled contigs into putative genomes. Several binning tools curretly exist, each differing slightly in their clustering algorithms, with different efficiency.

2.1 Assembly of reads

2.1.1 Too short to be good : a matter of size

Once a genome is sequenced into reads, depending on the technology used to sequence the DNA molecule, sequences obtained may be too short to predict structures, such as genes, gene clusters, or repeats regions, that span across regions longer than the reads themselves. This inability can cause difficulties to sort the reads in the right order to reconstruct the chromosomes. The inclusion of these structures being essential in the reconstruction of a genome, it is thus needed to obtain sequences longer than the output reads. It also helps to eliminate some sequencing errors present in reads.
The process to reconstruct sequences from reads is call an assembly, as the reads are assembled together to form longer sequences named *contigs*, short term for *contiguous*.

The assembly may be supervised thanks to the mapping of the reads through reference genomes, or performed *de novo* [116]. Supervised assembly output good quality contigs, its main advantages are its effectiveness in resolving repeats, its ability to exhibit better performances at low depths of coverage than *de novo* approaches [116]. However, it relies heavily on the availability of closely-related reference genomes. Because metagenomics aims at the exploration of hitherto ill-studied communities, for which there are limited available references, a supervised assembly may be limited in their applications [117]. A *de novo* assembly thus become more suitable, despite its increased computational cost.
First generation (Sanger) sequencing produced far fewer reads than the high-throughput sequencing technologies, with longer individual reads. The assembly of Sanger reads thus relied on overlap-layout consensus (OLC) approaches (fig. 2.1), in which overlaps are computed by performing pairwise alignments of all reads. The overlaps are then grouped together to construct a contiguous layout, and a consensus sequence is determined, by picking at each position the most likely nucleotides, such as in the Celera assembler [118]. The advent of the high-throughput sequencing technologies have both exponentially increased the number of reads, and dramatically shortened their average length. With the number of pairwise reads alignments growing with the number of reads following a polynomial factor, rendering the computational needs for an OLC-based assembly nearly impracticable. However, OLC assemblers have successfully been adapted for metagenomic data, like MAP, which improved the OLC strategy by preliminary filtering reads that would be used to compute the overlap graph [119] while also simplifying the graph assembly using pair-ended information. An other example is Omega [120] which lowers the computational costs of OLC assembly with a hash function built of the prefix and suffix of read. It then uses the hash function to compute overlaps, and simplify the graph assembly by trimming the reads, which are completely contained within a larger contiguous structure.
To overcome this computational impediment, de Bruijn Graph assemblers have been introduced [121] and have become widespread in the field. A de Bruijn Graph (dBg) is a directed graph where each node represents a $k$mer, i.e. a string of k nucleotides. Nodes in the graph are connected if the last $k-1$ nucleotides of one node correspond to the first $k-1$ nucleotides of the following node. The graph is built by decomposing each read into individual, overlapping $k$mers, and creating nodes for new $k$mers, updating coverage for existing $k$mers and adding vertices for new transitions (fig. 2.1). In an ideal case, a dBg would form a single line in which each node is connected to one other node in the forward direction, and to one in the reverse direction, except for the two nodes located

at the extremities of the graph. It would thus output a unique contig, spanning all the $k$mers retrieved from the reads. Real dBg are of course much more complex than this simple representation, and complex branching structures occur, as a result of structural variants, coverage differentials, heterozygosity, and sequencing errors. Much of the innovation in genome assembly algorithms has come from developing heuristics to simplify and navigate complex graphs, to output graphs. Amongst these advances, is the succinct dBg, which, thanks to a data structure enabling text compression [122], has helped to dramatically reduce the memory consumption to store the graph [123]. The dBg has thus

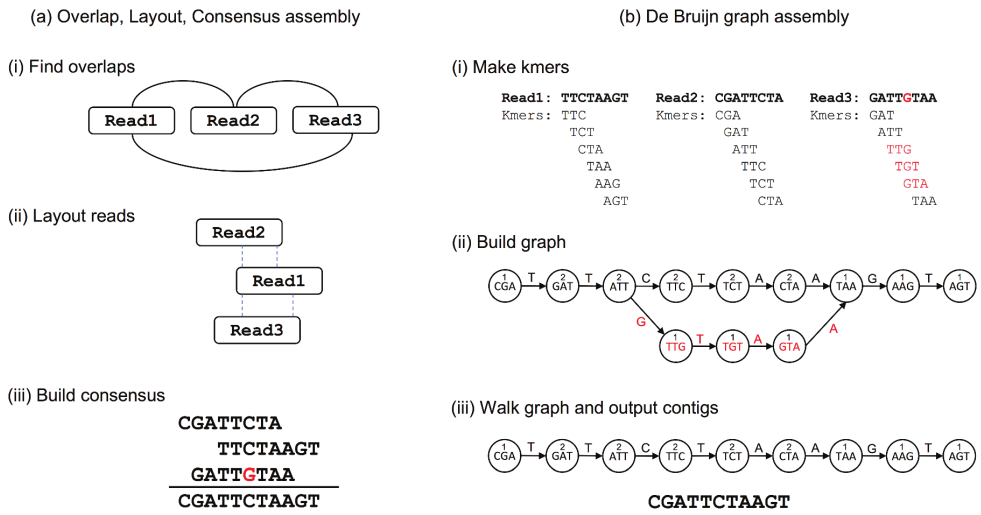

Figure 2.1 **A schematic representation of an assembly graph.**
In an OLC assembly (a), overlaps are found by performing pairwise reads alignments (i), then contiguous layouts are constructed based on these overlaps (ii), and a consensus contig is determined by the frequency of the nucleotides (iii). In a dBg assembly (b), first the $k$mers are retrieved from the reads (i), then the graph itself is constructed, linking $k$mers overlapping on $k-1$ nucleotides (ii), and contigs are constructed traveling through eulerian paths of the graph (iii). From [117]

two main advantages on OLC graph : its memory consumption is sensibly inferior, as well as the number of computational operations needed to construct it. It also may handle more easily with repetitions within the data, i.e. the presence of duplicated reads, than OLC assemblers. However, the segmentation of the reads into smaller fragments may lose context from the data, making dBg more prone to produce erroneous edges, between two $k$mers belonging to two different genomic regions for example.
Currently, the sequence-by-synthesis approach, such as Illumina, has an average error rate reported to be less than 0.5% per nucleotide [124], most of which being SNP. The presence

of erroneous reads may not represent a main source of errors assembly in OLC assembly: because of their rarity, errors in the sequence of the reads are easily removed during the construction of the consensus. However within a dBg assembly, each SNP will generate k erroneous $k$mers, resulting in the apparition of cycles with two k-length paths. Those cycles, also called "bubbles", may shorten the length of output contigs when they remain unsolved, while also increasing memory consumption [125]. A simple approach to remove erroneous reads from the input sequences is to rely on their abundance in the sequencing products. Because of the low error rate of the sequencers, the number of erroneous reads are expected to be consequently low, and thus, unique. As such, some assemblers only consider $k$mers with a number of occurences of at least 2 [125, 126].

Contents of a dBg

Generally, dBg are stored in a data structure relying on a hash table [127, 128]. k may be any particular integer, the number of possible $k$mers growing with the value of k (the number of possible $k$mers can theoretically reach 4^k). The hash table, and thus the hash function, then must be adapted to limit conflictual storage (*i. e.* giving to 2 different $k$mers the same hash value). However, the actual number of $k$mers contained within the whole set of reads is almost independent from the value of k, as it is equal to $C - k + 1$, with C the number of nucleotides composing the reads. As $C >> k$, the number of actual $k$mers can be considered as equivalent to C. As such, for values of k high enough, there is enough space in the hash table to limit conflicts (as $4^k >> C$), at the expense of an increased memory cost, with a high proportion of unused space (*i.e.* $k$mers that do not appear in the reads set). The choice of a relevant value for k thus being difficult, it has been recommended to construct several graphs assemblies using different values of k [129].

2.1.2 Specifities of the metagenomic assembly

The assembly of metagenomic reads, while relying on the same basis as the assembly of individual genomes, is a complex task. This is in part due to computational memory

constraints, but mainly as a result of biological complexity, including genetic diversity and mobile genetic elements present in the community. Long stretches of near-identical metagenomic sequences are especially hard to assemble with short reads, because such sequences might originate from multiple sources: repetitive DNA of a single genome, homologous regions of closely related strains, or conserved regions of different species that coexist in the community. The presence of closely-related species, or strains within species, may also create extensive overlaps in a *k*mer set, increasing the complexity of the assembly graph, as multiple genomes occupy much of the same *k*mer space [117]. Thus, the assembly process needs to resolve these regions to avoid generating erroneous rearrangements or to reconstruct chimeric contigs.

Namiki et al. [130] outlined the most important limitation of single-genome assembly software, which is their inability to cluster sequence reads with diverse origins and heterogeneous coverage. They designed a strategy to decompose the dBg of multiple species into subgraphs, each representing a cluster of reads from an individual species [130]. These individual subgraphs may represent population genotype bins. An enhanced version of this assembler, Metavelvet-SL, used a supervised machine learning approach to enhance the detection of chimeric nodes within the graph [131].

The memory boundary, which represents a major impediment to metagenomic assembly, was addressed by adapting memory-efficient data structures to store the dBg. Pell [127] thus first used a bloom filter to store a dBg. This probabilistic approach however generate false *k*mers, with a false-positive ratio inversely related to memory consumption, which could connect to actual *k*mers, thus deteriorating the quality of the contigs. This problem was resolved with Minia [128], which, by discriminating and eliminating these false *k*mers, succeeded to perform exact assembly. Other strategies to reduce high-peak of memory consumption was the use of parallel algorithms on distributed-memory machines, for example with HiPMer [132] and its later version adapted to metagenomic data, metaHiPMer [133], while Megahit [125] relies on a succinct dBg representation.

The abundance of the reads may represent a relevant metric to discriminate between two possible paths of a bubble within the graph, as *k*mers with close abundance may have a higher probability to belong to the same read, or to belong to reads from the same organism. A problem posed by the abundance of reads is that the handling of erroneous reads may cause the elimination of reads from actual, low-abundant organisms in the metagenome.

As the reads already undergo the filter of the sequencing step, they also may be more

under-represented within the sequencing products, at the point that reads from low-abundant organisms may be present at such low abundance that they do not pass the abundance threshold of the assembler, discarding them for the assembly. A strategy has been to adapt the abundance threshold, to the local depth level of the region of the assembly graph [134]. Megahit, a metagenomic assembler, answers this issue by introducing a *mercy kmers* strategy. Given two solid $k$mers (i.e. $k$mers with an abundance of at least 2) x and y from the same read, where x has zero outdegree and y has zero indegree. If all $(k+1)$-mers between x and y in that reads are not solid, they will be added to the dBg as *mercy kmers* to strengthen the contiguity of low-depth regions [125]. Another approach allowing a better integration of the variants present in the community is to rely on a colored dBg [135], which attributes colors to $k$mers to reflect their origin. This approach thus can help to identify the good path of a bubble, but is still impacted by sequence repetitions, and also eases the assembly of reads belonging to multiple samples [136].
An approach to ease the assembly of low-depth regions may be to perform a *co-assembly* of several metagenomes together. The idea behind this approach is that even if there is a rare organism living in several similar environments, the addition of metagenomic samples coming from these environments will increase the abundance of the rare organisms, exceeding the abundance threshold of the assembler. Metagenomic co-assembly was already performed in several studies with high-complexity environmental communities [5, 62]. However, co-assembly increases both the memory consumption, by adding $k$mers to be considered in the dBg, and the resolution time of the assembly step. Co-assembly also increases the probability to generate chimeric contigs, and may increase the abundance of erroneous $k$mers, resulting in shorter contigs as compared to single-sample assembly, leading to a more fragmented assembly [137].

2.1.3 Scaffolding

Although technological advances have made sequencing DNA much cheaper and faster, short-read, high-throughput sequencing exacerbates the central challenge in genome assembly: accurate assembly of genomes that are often highly repetitive. With the growing use of high-throughput sequencing, the fragmentation of new genome assemblies increased, affecting the results of comparative genomics analyses. The preparation of DNA libraries through bacterial cloning was a first reason to the diminishing contiguity of the assemblies. Plasmid libraries enabled generation of mate-pair reads, generating reads from both ends of the plasmid insert, for little additional cost relative to single-end sequencing. Many early

genome assemblies benefited from mate-pair sequencing, whose insert sizes were several kilobases long. Due to the nature of their assembly, contigs do not overlap, and there are gaps between them. The next step to the reconstruction of draft genomes would need thus to bridge the gaps between the contigs, to sort them. This next step is called *scaffolding*, and it can be done using either optical mapping or paired-end read linkage. Paired-end read linkage use the pair-mate sequencing characteristic, in which reads are sequenced by pairs, forward and reverse, separated by an arbitrary inserted sequence called the inner sequence or "insert". The final output of the scaffolding is composed of ordered contigs, and gaps (fig. 2.2). This step is however not mandatory in the reconstruction process of MAGs, as several binning tools rely equally on contigs or scaffolds as input data.

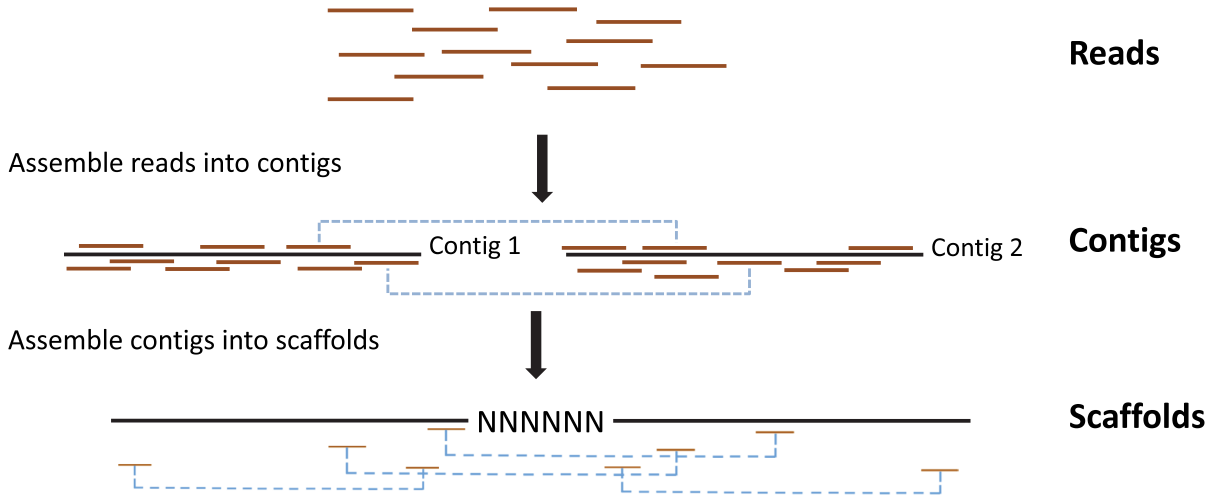

Figure 2.2 **Summary of the assembly step, from reads to scaffolds.**
Dot lines represent paired-end reads linkage between contigs. The position of the *forward* and *reverse* fragments of the paired-end reads help to order the contigs to reconstruct the genome. As the nucleotides separated two contigs are not known, they are noted as "N". From [138].

2.2 Binning

Metagenomic binning, the scond main step of MAGs reconstruction, consists in the clustering of reads [139, 140], contigs [24, 141–143], scaffolds [144] or genes [145] based on their genetic characteristics, including oligonucleotide frequency and/or coverage. This clustering step may be processed through a combination of different approaches, such as hierarchical clustering and neural networks. These clusters are then grouped with var-

ious data representation approaches into individual taxonomic bins. While performing binning of reads, those bins then generally aim to assess the metagenomic taxonomic diversity [139]. With contigs and scaffolds binning, the retrieved bins are considered as putative organisms' genomes [9, 10, 146]. However, contigs have been largely preferred over scaffolds as the main items to bin into putative genomes, with the only tool using scaffolds being converted to the use of contigs in its second version [144, 147]. Recently, a binning tool relying on scaffold exhibited higher performance at bin reconstruction than contigs-based binning tools [148]. Two main ways to perform contigs binning can be distinguished, the supervised binning, which relies on the assignment of contigs to a reference genome, and unsupervised binning, which perform the clustering of contigs *de novo*, relying only on intrinsic features of the sequences.

2.2.1 Supervised binning

Historically, it has not been possible to reconstruct the genomes of species belonging to complex communities due to insufficient sequence coverage, thus tool development has largely focused on classification algorithms that assign taxonomy to sequence fragments (including reads). Before 2007, this taxonomic assignation was mainly performed using ribosomal marker genes, but the sparse proportion of reads or contigs carrying these genes limited their efficiency [149]. A first method based on sequence composition [149] relied on a support vector machine classifier, inferring taxonomic assignation thanks to oligonucleotides composition of the genomic fragments, and was able to cluster fragments of at least 1 kb. MEGAN [150] relied on homology search, with a two-steps algorithm combining BLAST to compare sequences to reference genomes and finding the lowest common ancestor to assign taxonomy. Marker genes from specific lineages were also used as a post-processing to strengthen the analysis. This method was able to cluster reads as short as 35bp, which was the common output length for Illumina at this period. However it suffered from a high computation time to perform all sequences alignments. Phylogenetic affiliation based on Hidden Markov Models (HMM) has also been explored as a mean to perform reads taxonomic affiliation, using either Pfam proteins family [151], or genes considered universal in bacterial genomes [152], allowing classification of sequences as short as 80 bp, and drastically reducing the computation time compared to MEGAN. All these methods have however been criticised for their lack of performance, and also for their sensible decline of efficiency with sequence length. In order to gain performance, a combination of

these approaches has thus been proposed with PhymmBL [153], which used Interpolated Markov Models (IMM) trained on 539 complete, curated, bacterial genomes, coupled with a BLAST sequence comparison, to significantly enhance the clustering of reads with length of at least 100 bp. Other hybrid approaches have then been developed, coupling homology detection and bayesian phylogenetic affiliation [154], and sequence composition with homology detection [155]. The main hindrance underlying these supervised methods is the lack of reference genomes in databases for the major proportion of bacteria. In contrast, unsupervised clustering methods remove this impediment by performing self-comparison of the assembled contigs for genome binning. Recently, the taxonomic assignation of a contig to a reference genome was used within a tool relying on a semi-supervised spectral clustering approach [156], but its little gain in clustering efficiency came at the cost of a harsh increase of its high computation time when compared to recent unsupervised binning tools.

2.2.2 Unsupervised binning

Due to the lack of reference genomes that may hinder the efficiency of supervised binning tools, alternative binning approaches were developed based on the inherent characteristics of contigs. Based on the differences in sampling content (one sample or series of samples), clustering inputs (nucleotide composition-based or nucleotide composition-independent) and use of abundance information, current methods of recovering genome bins from metagenome assemblies can be divided into three types [157, 158] : sequence composition (SC)-based , differential abundance (DA)-based, and sequence composition and abundance (SCDA)-based. The major difference between the three methods is the starting point for the contig binning process.

SC methods rely on oligonucleotide frequency variations. Nucleotide composition has long been identified as being characteristic to a given species. The proportion of guanine and cytosine in the genome is notably influenced by the past evolution of organisms, having a strong impact on their competitiveness in their environment [159]. When considering constraining environments, it thus becomes difficult to discriminate between genomes from organisms belonging to the same community. Oligonucleotide frequency may also vary within the same genome, depending on the transcription activity of the genomic regions. To increase the quantity of information with a compositional metric, it was then considered to measure the proportion of small subsequences within the sequence. Oligonucleotides of 4 letters, called tetranucleotides, have been shown to overcome the limitations

of dinucleotides, and thus to represent an ideal metric to assign genomic fragments to a genome [160]. The incorporation of a combination of penta-mers and palindromic hexamers abundances has been recommended as the best compositional metric for datasets producing more than 50 bins [142]. However, this usage has not been established as a standard in the binning community, as it relies currently solely on tetranucleotides to assess compositional features of the contigs. Thus, the compositional metric in metagenomic binning is generally referred as TetraNucleotides Frequencies, or TNF.
While earlier binning attempts were conducted using metagenomic reads, current workflows assemble them into contigs first. The reasons are the sensible enhancement of metagenomic assembly these last years, and that both SC and DA signals are more robust on longer sequences.

Canonic representation of oligonucleotides

The total number of $k$mers contained in a string depends on the size N of its alphabet, with N^k number of elements. Thus within the DNA, there are $4^4 = 256$ different tetranucleotides. However, because of its double-strand structure and the complementarity of the bases, many tetra-mers have a reverse-complementary version, which can be deducted from the tetra-mer sequence itself. To reduce the number of tetra-mers to consider, they are represented into their *canonic* form, with the best lexicographically-ranked tetramers between the forward and reverse. This allows to reduce the number of tetra-mers to consider to compute TNF to 136.

Microbial communities surveys published in the beginning of the 2010s relied for their majority on SC, using mostly oligonucleotide frequency and %GC, while the latter become less and less considered, due to the greater resolution of the first. In their study, Iverson et al [161] first used a graph-based approach for assisting individual genome reassembly. They then construct a network graph where nodes represent the scaffold, and edges represent tetranucleotide Z-statistic correlation. Edges corresponding to a score below an empirically-determined correlation score are then removed from the graph, and connected components with a cumulative length above 950 kb are clustered into candidate genome bins.

SC-only approaches have mostly been applied to communities composed of genotypes that possess nucleotide composition pattern, such a low %GC, and consistent oligonucleotide frequency. It is likely, though not proven, that this technique may struggle with communities composed of genomes with high oligonucleotide compositional variance.
DA methods rely on similarity of coverage profiles to cluster contigs into putative genome bins. The contigs coverage may be estimated by mapping the reads belonging to one metagenome, or several metagenomes, to the contigs from the assembly. The term coverage used in metagenomics studies then generally referred to the mean vertical coverage, i.e., the mean times a read aligned to a position on the contigs. It has been shown that the mapping profile of genomic fragments represents a characteristics of a genome [162]. Thus, the rationale of DA binning tools is that contigs with similar coverage profiles have a higher probability to belong to similar genomes. It has been shown that DA is a more powerful metric to group contigs than SC alone, allowing to improve MAG reconstruction from the assembly [109, 163]. The resolution of the reconstruction of MAGs also increases with the number of metagenomic samples from which DA is estimated [24]. An example of DA binner is GroopM [164], which performs contigs binning through abundance profiles computed from several metagenomic samples. It also includes a refinement step, splitting, merging or deleting preliminary bins through detection of chimerism, using marker genes. However, it has limited capacity to separate contigs of closely related genotypes, which are placed in chimeric bins, and it requires at least three related samples to perform binning.

Infer the classes from the data

Emergent Self-Organizing Maps (ESOM) is an unsupervised clustering approach which relies on the inner structure of the data to perform an unsupervised clustering. One of its advantages over k-means, another popular clustering approach, is that the classes may be inferred from the map itself, and that it can leave a point unclassified. ESOM have greatly benefited from the development of deep neural networks, which have allowed them to scale, and their application to large datasets [165] make them relevant to be integrated in a binning tool [163].

Sharon et al. [163] reconstructed six complete and two near complete genomes from gas-

trointestinal microbiome, applying a DA approach on times series data. These genomes belonged to organisms representing a fraction as low as 0.05% of the total bacterial community, demonstrating the high resolution of their method. They first performed an iterative assembly, optimized thanks to preliminary defined coverage bins. Another example of genome binning relying on DA is the canopy algorithm [145]. The reconstruction strategy used a gene catalog established from preliminary assemblies of the metagenomes, to determine the co-abundance of each gene. Genes with similar abundance profiles, i.e., with a Pearson correlation coefficient above 0.9, are then clustered together in a gene group. Gene groups are then constructed as sets of genes whose abundance profiles exhibit a high correlation with the abundance profile of the central gene, that is chosen randomly. The gene groups are constructed iteratively, adding genes with an abundance profile which is similar to the mean abundance profile of the already-added genes in the group at each iteration. The formed gene groups, or canopies, which passed rejection criteria, *i.e.* containing more than two genes and originating from more than three metagenomic samples, are then considered as Co-Abundance gene Groups (CAGs), and a CAG containing more than 700 genes is considered as a Metagenomic Species (MGS). MGS are then used to realize an enhanced assembly of the reads, assembling the reads mapping to the same MGS together. The canopy algorithm's main advantage is its low complexity, with the reconstruction of wide CAGs performed with a high velocity, and the genes within the MGS showed consistent abundance profiles. The main limitation of DA-binning methods is the decline of their performance when the number of samples is low, rendering them not completely suitable for single-sample studies. As the dependency to a large volume of data became more or less fulfilled with the increasing capacity of production of data, the human supervision hindered the reproducibility and scalability of such approaches [24].
Binning approaches coupling both SC and DA metrics (fig. 2.3) have already been used for a decade now, with a first study reconstructing genome bins from cow rumen metagenome [166]. Mackelprang et al. [167] followed a similar approach, using a hierarchical agglomerative clustering method to process the tetranucleotide frequency matrix, then clustering metagenome contigs into genome bins adding differential abundance. Numerous tools have then followed this composite approach, such as CONCOCT [24], myCC [142], MaxBin [144] or Metabat [141]. CONCOCT integrates the SC profile of the contigs using a Principal Component Analysis (PCA), and the co-abundance profiles using Gaussian Mixture Models. Maxbin [144] classifies the scaffolds following the Expectation-Maximisation (EM) algorithm, which calculates the probability that a given scaffold belongs to any genome

at the same time. The EM algorithm performs two steps iteratively : first, it computes the probability for each point to belong to one cluster (expectation step), relying on maximum-likelihood estimation [168], then it updates new values of the clustering metrics for each cluster (maximisation step). It needs a preliminary determined number of clusters to perform, each with initial values of TNF and coverage, which are estimated *via* the detection of marker genes through the contigs. An enhanced version, allowing the estimation of contigs coverage profiles through several metagenomes, has then been developed in the software Maxbin2 [147].
Even though the rationale of the SCDA approaches are both to gain power resolution and limit the major drawback of the DA approaches, they still suffer from a sensible decline of their performances when only a limited number of metagenomes is available [157]. myCC [142] tries to alleviate this disadvantage by proposing an optional integration of DA profiles, with only the SC step being mandatory. The SC-binning step relies on a reduction of dimensionality of compositional genomic signatures, using Barnes-Hut stochastic neighbor embedding (Barnes-Hut SNE), which is a dimensionality reduction method. The DA profiles are then integrated to the SC metric if the user wishes to integrate them, and the contigs binning follows an Affinity Propagation (AP) algorithm.

Metabat [141] follows an original approach which modulates the weight of the co-abundance profiles in the binning operation on the number of samples [141]. The software performs a first clustering of contigs using only a distance probability estimated from TNF-euclidean distances, estimating the probability for two contigs to be binned together. Then, for each pair of contigs, a DA distance probability is estimated, computed as the unshared area under curve of the normal distributions of the DA profiles of the contigs. DA distance probabilities are then integrated to a global score, which is used to upgrade the first binning. In Metabat [141] the coverage profile of contigs is modelled as a normal distribution. The abundance distance between two contigs is thus set as the area under curve between the two contigs coverage distributions (fig 2.4). Both composition and abundance are then set as distance probabilities, and two contigs are clustered together when this distance reaches a threshold. This initial version of Metabat allows to increase the binning performance when compared to previously-published SCDA binning tools, both in terms of computational efficiency and accuracy. However, it is prone to inconsistent results when it is used on different datasets, and a fine tuning, requiring several runs with multiple parameters sets, may be needed to achieve an optimal performance [9].

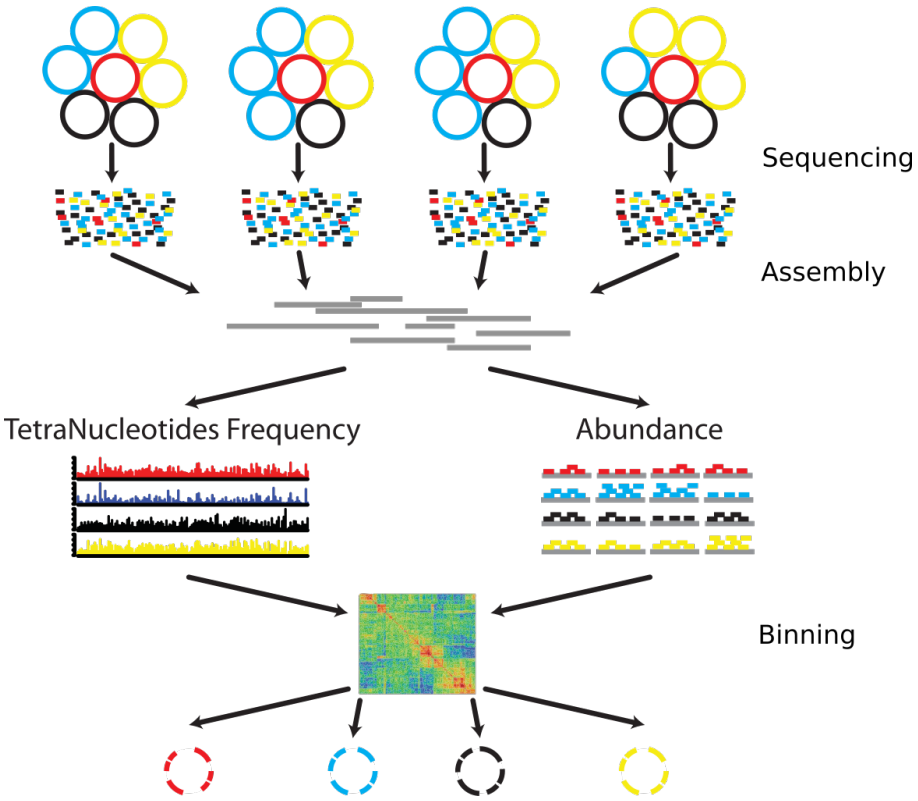

Figure 2.3 **Overview of a typical SCDA binning approach.**
After sequencing the genomes (coloured circles, top), the reads (coloured small lines) are assembled into contigs (grey lines). A compositional metric, generally the TNF, and coverage/abundance profiles are inferred from the contigs sequence, and both are used to cluster the contigs into putative genomes (dotted coloured circles, bottom). From [141]

Metabat2 [16] differentiates from its former version, with a preliminary designation of putative contigs to cluster using TNF, before adding the coverage information to finalize the binning, supported by a graph-based approach. This change of its main algorithm has helped Metabat2 to dramatically increase its precision when applied to high complexity communities. Besides, Metabat2 is currently the most rapid binning tool in the field of MAGs reconstruction, as well as the most frugal in terms of memory usage. Maxbin2 and Metabat2, both because of their performance and their ease of use, are the most used tools in metagenomics studies [158].

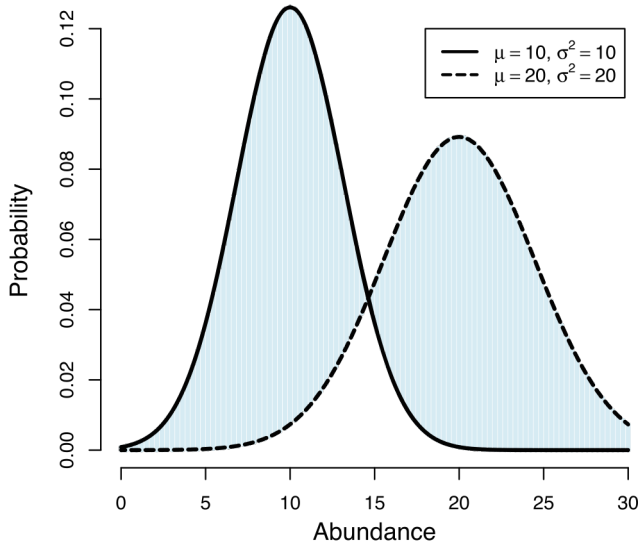

Figure 2.4 **Metabat/Metabat2 contig's coverage metric.**
The coverage of contigs is modelled as a normal distribution, of mean μ and standard deviation σ. The distance probability estimated through contigs' coverage is the unshared area under the curves of both contig's distribution (shaded area). From [141]

2.2.3 The need for a standardised benchmark

In the middle of the 2010s, an acknowledgement is made : even though the current metagenomic literature flourished with the development of assembly, binning and taxonomy profiling tools, their results became extremely difficult to compare. A couple of datasets already represented a relevant choice to perform the evaluation of new tools, such as the METAHIT dataset [174], or the Sharon dataset [163]. However, if these datasets

Tool Name	Access link	Type	Release	Last updated	Current citations	Ref.
ABAWACA	https://github.com/CK7/abawaca	SCDA	-	-	-	-
BMC3C	http://mlda.swu.edu.cn/codes.php?name=BMC3C	SCDA	2018	-	24	[169]
Canopy	http://git.dworzynski.eu/mgs-canopy-algorithm	DA	2014	-	560	[145]
COCACOLA	https://github.com/younglululu/COCACOLA	SCDA	2017	2017	66	[170]
COMET	https://github.com/damayanthiHerath/comet	SCDA	2017	2018	8	[171]
CONCOCT	https://github.com/BinPro/CONCOCT	SCDA	2013	2019	767	[24]
ESOM	https://github.com/MadsAlbertsen/multi-metagenome	DA	2013	-	789	[109]
GroopM	https://github.com/Ecogenomics/GroopM	DA	2014	2016	172	[164]
Maxbin/ Maxbin2	https://sourceforge.net/projects/maxbin	SCDA	2014	2020	321/672	[144, 147]
MetaBAT/ MetaBAT2	https://bitbucket.org/berkeleylab/metabat	SCDA	2015	2019	856/583	[16, 141]
myCC	https://sourceforge.net/projects/sb2nhri/files/MyCC	SCDA	2016	2017	126	[142]
SemiBin	https://github.com/BigDataBiology/SemiBin	SCDA	2022	2022	3	[172]
SolidBin	https://github.com/sufforest/SolidBin	SCDA	2019	2020	18	[156]
VAMB	https://github.com/RasmussenLab/vamb	SCDA	2018	2022	43	[173]

Table 2.1 Contigs binning tools.
Current numbers of citations are taken from Web of Science (october 2022).

were used for several tools evaluation studies [141, 156], they did not represent a standard. Thus, tools evaluation were still performed with different data sets, performance criteria, and evaluation strategies. Acknowledging this issue, a community-driven initiative, the Critical Assessment of Metagenome Interpretation (CAMI) aimed to establish standards in the design of benchmark datasets, procedures, choice of performance metrics and questions to focus on [175]. This study also led to the evaluation of several assemblers, binning tools (both supervised and unsupervised) and taxonomic profiling tools, and have pointed out the difficulty to recover taxonomic information below the family level, even though all binning tools have already shown good performance with communities without closely related strains [175]. The simulated datasets used for the CAMI challenge were then made publicly available, and are still currently standards for the evaluation of new binning tools, used in several studies presenting new tools [16, 176] or performing global evaluation of binning tools [177, 178]. A second version of the CAMI challenge started in 2019, even though the results have not been published yet.

2.2.4 Quality assessment : from bins to MAGs

The rapid development of genome binning tools led to the production of thousands of draft genomes, with expectations that the number of genome-resolved metagenomics studies continues to grow in number in the near future. This requires the availability of automated tools to assess the quality of MAGs, and to perform post-processing refinement or contaminated sequences removal. The main purpose of quality assessment and bins refinement is to avoid the submission of sub-optimal MAGs to public genome repositories, in order to maintain their quality. The Genome Standards Consortium (GSC) has developed two standards for reporting bacterial and archeal genomes [179]. These standards include the minimum information about a single-amplified genomes (MISAG) and a metagenomes-assembled genome (MIMAG). Due to the lack of genome references acting as ground truth, the authors advised the report of standard assembly statistics, including total assembly size, contig N50/L50, and maximum contig length. Information about the presence and completeness of the ribosomal and transfer RNA genes are also considered appropriate complement about MAG quality. Manual curation has also produced the best-quality MAGs, but this approach, which is time-consuming, may be prohibitive to be set as a standard, in regards with the amount of genomes to process, and also its lack of reproducibility.

No standards have been defined to evaluate the completeness and the contamination ra-

tio of a MAG. As often, the ideal approach might be the alignment of the MAG to a closely related reference genome. But the lack of reference for the majority of microbial lineages, and high levels of strain heterogeneity, exclude such a process. As an alternative, researchers have relied on the presence of "universal" marker genes to estimate completeness of MAGs. These genes have to exhibit several characteristics, including their presence in genomes of nearly all taxa, to be present in a single copy in the genome (enhancing their second name, SCGs - for Single Copy Genes) and to not be subject to horizontal transfer. The marker genes should exhibit specific characteristics, such as being essential for a wide range of microbial taxa, and be present in a single copy within the genome. Based on SCGs, the completeness of MAGs may be defined as the ratio of the number of SCGs detected within the genome to the total number of SCGs, while contamination may be defined as the ration of the observed SCGs in two or more copies to the total number of SCGs [180].

Several SCGs sets have been identified and validated [181], corresponding to bacterial and archaeal genomes, and are included in quality assessment software, such as CheckM [180], Anvi'o [182], mOTU [61] and BUSCO [183]. CheckM proposes a particular approach, and infers lineage-specific genes, based on the position of a query genome in a reference tree using a reduced set of multi-domain markers, finally allowing the use of an increased number of SCGs. It can also estimate the level of strain heterogeneity, which may represent an adequate complement of completeness and contamination, as it assesses whether the source of contamination are strains relatives or SCGs from unrelated taxa. Currently, as Anvi'o [182] and Vizbin [184] require human assistance during their workflows, CheckM remains the most widely used tool to assess MAGs quality, due to its ease-of-use, completely automated workflow, and high accuracy. However, its assessment of quality relying solely on SCGs has recently been questioned [185], due notably to their uneven localisation through a genome [186]. To overcome these limitations, an assessment of the quality of bins, using all genes detected in a bin, has been proposed with GUNC [14].

After quality assessment, MAGs may be classified as near-completed (single continuous sequence without gap or overall quality score equal to or above Q50), high-quality (completeness $\geq$ 90% and contamination $\leq$ 5%), medium-quality (completeness $\geq$ 50% and contamination $\leq$ 10%) and low-quality (completeness $\leq$ 50% and contamination $\leq$ 10%) [179]. A majority of downstream analysis tools recommend to discard low-quality MAGs to avoid false conclusions.

2.2.5 Dereplication

The reconstruction of MAGs from several independent metagenomes' assemblies belonging to similar environment leads almost inevitably to the recovery of highly similar MAGs across all the assemblies. In order to reduce the computation cost of downstream analyses, it is therefore often recommended to perform MAGs dereplication. The presence of multiple closely related genomes may also complicate the manual curation of MAGs [187]. Dereplication may be defined as the reduction of a given set of MAGs on the basis of sequence similarity among them. To this end, the estimation of Average Nucleotide Identity (ANI) between two genomic sequences has been set as a robust approach to compare prokaryotic genomes [188]. However, the computation of ANI requires pairwise alignments of MAGs, which may be computationally intensive if the number of genomes becomes high, as the number of alignments scales quadratically with the number of genomes. The utilisation of MUMmer [189] instead of BLAST as the alignment tool to perform these pairwise mappings has shown to be both faster and more robust to compute ANI between two closely-related genomes [188]. MUMmer has also been implemented as a Python package which is still maintained [190]. Another scalable algorithm, using identification of orthologous genes to compute genome-wide ANI (gANI) has been developed to refine taxonomic assignement of microbial genomes [191].
Alignment-free approaches to estimate genomes or metagenomes closeness has been developed, such as Mash [192]. Mash first creates sets of MAGs named *sketches* before computing the distance between two sketches, providing a similarity measure between the two MAGs (fig. 2.5).

This approach has revealed to be sensibly faster than alignment-based approaches to compare genomes, with the Mash distance being strongly correlated with ANI. However, it has been shown that the accuracy of Mash diminishes significantly with the completeness of MAGs [193]. A hybrid, bi-phasic approach, combining Mash and gANI has been proposed to both reduce the computational time required for genome dereplication, and ensure high accuracy, with the software dRep [193]. With this tool, the genome set is first divided into primary clusters using Mash, then each primary cluster is compared in a pairwise manner using gANI, constructing secondary clusters of near-identical genomes that can be dereplicated. Currently, dRep constitutes a routine tool to perform MAGs dereplication.
By removing a certain proportion of MAGs for the downstream analysis, the dereplication step constitutes a loss of information. In certain circumstances, such as the detection of

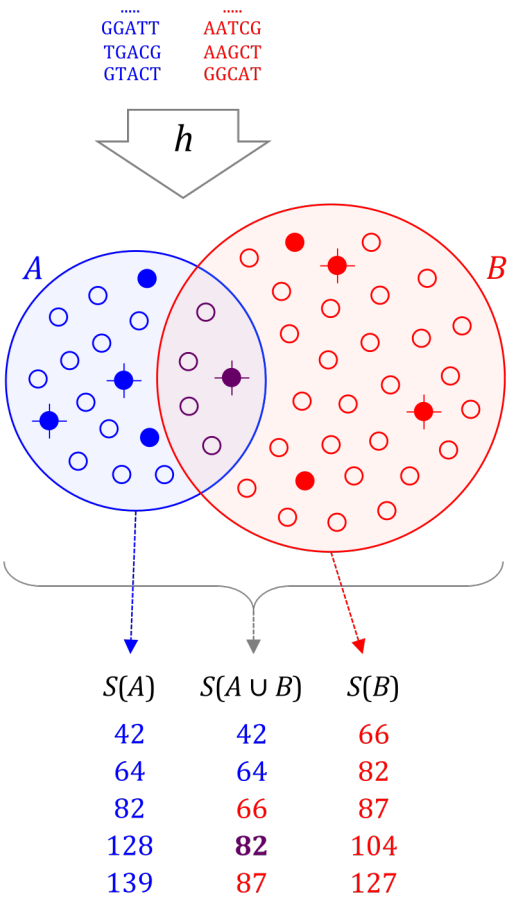

Figure 2.5 **Overview of the MinHash sketch strategy for estimating contigs similarity in Mash.**
First, the *k*mers of two objects to compare, *i. e.* contigs, or metagenomes (in red and blue, top) are decomposed into their constituent *k*mers, and each *k*mer is passed through a hash function h to obtain a hash (small circles). A and B thus represent the hash sets of the two elements to compare. The fraction of shared *k*mers between A and B (purple) is estimated by subsampling A and B: here, $S(A)$ and $S(B)$ both represent the 5 *k*mers with the smallest hash values (filled points) in A and B, respectively. Because $S(A \cup B)$ is a random sample of $A \cup B$, it is then an unbiased estimate of the fraction of shared *k*mers. From [192].

auxillary genes within bacterial strains, the dereplication may remove a significant and relevant information, hindering the detection of these genes [187]. The decision to dereplicate MAGs or not should thus be made in accordance with the purpose of the study.

2.2.6 MAG refinement

In order to increase the overall quality of MAGs before downstream analysis, different refinement approaches have been developed to increase their completeness and decrease their contamination. One of these approaches relies on using several binning tools to produce a set of optimized and non-redundant MAGs. Since numerous binning tools show differences in the metrics, integration of the metrics, and clustering algorithms they rely on, these tools do not catch the same part of information from the metagenomes into the MAGs they reconstruct. Pooling bins from several binning software has thus been thought as a process to gather a maximal proportion of the metagenome's information, thus compensating the limits of each method taken individually. Binning_refiner [194] performs pairwise BLAST alignments between sets of bins recovered from different binning tools, in order to identify shared contigs between them. These shared contigs are then output to produce new, refined bins, which exhibit both a decreased contamination level, a longer sequence without any contamination, and a higher precision. This refinement process reveales however to be very stringent, as it significantly diminishes the completeness of the refined bins. DAS Tool [27] also uses bins sets recovered from different binning tools, and detects shared contigs between two sets of bins relying on the presence of SCGs in the sequence of the contigs. Then, after designating the bin of best quality within a set of duplicated genomes, it retrieves the shared contigs found in the best bin from the duplicated versions of this bin. By the end of the process, DAS Tool outputs the complete, best bin from all the binners, coupled with exclusive contigs that have been binned to the equivalent of that bin by the others binning tools (fig 2.6). DAS Tool produces MAGs with higher completeness, but its aggregation procedure may also increases the contamination of the MAGs.

Another approach has been implemented in the METAWRAP [26] workflow, as a post-processing module which aims to refine the binning process. This refinement procedure outputs the best quality-score MAG from a set of copies obtained from the different binners, and their hybrid counterparts, constructed preliminary using the Binning_refiner tool which is more similar to the dereplication step performed by dRep. This procedure has shown to produce more MAGs of higher qualilty than both Binning_refiner alone and

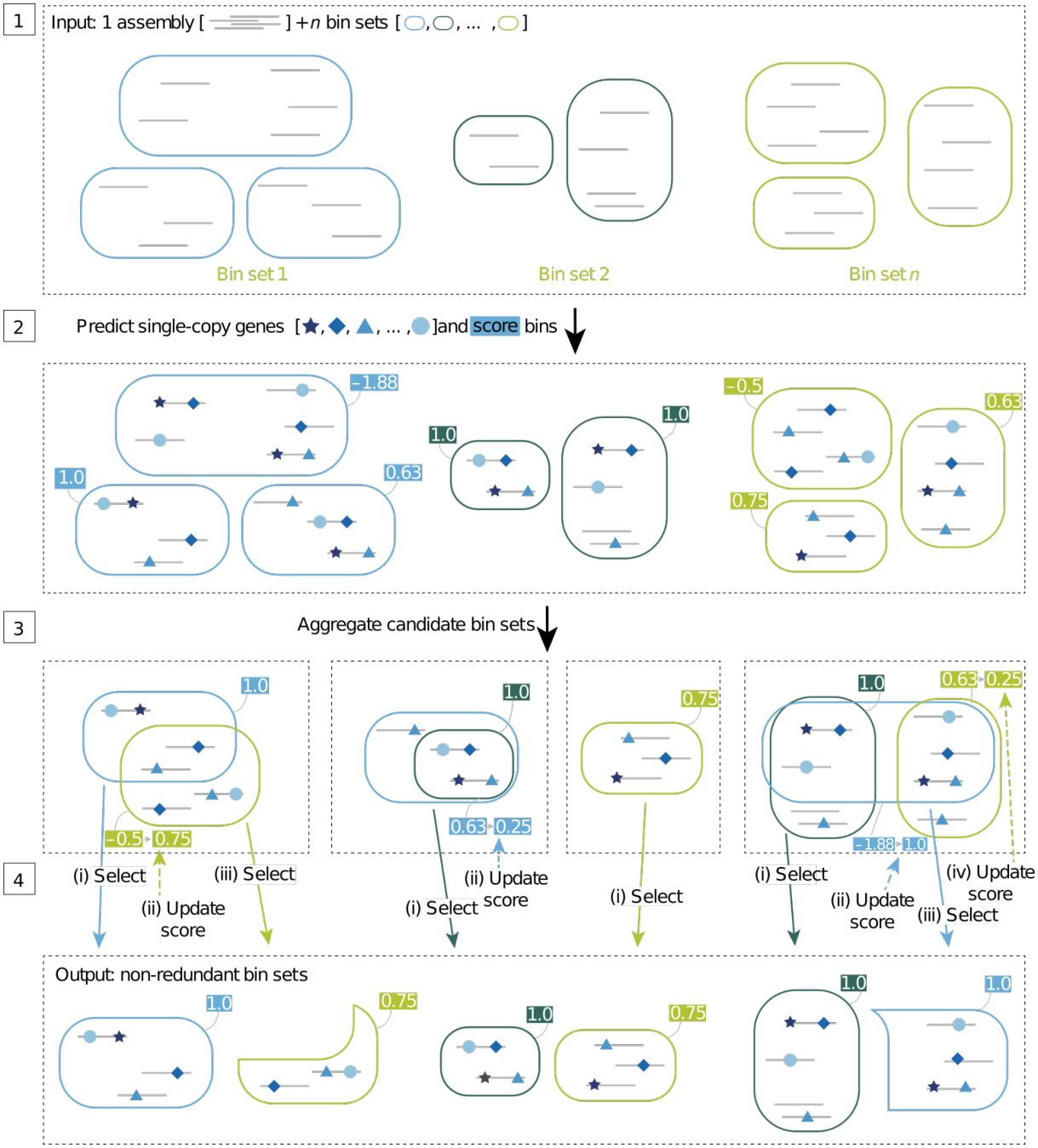

Figure 2.6 **Overview of the DAS Tool algorithm.**
Step 1: The input of DAS Tool comprises scaffolds of one assembly (grey lines) and a variable number of bin sets from different binning predictions (same-coloured rounded rectangles). Step 2: Single-copy genes (blue shapes) on scaffolds are predicted and scores (blue and green boxes) are assigned to bins. Step 3: Aggregation of redundant candidate bin sets from all binning predictions. Step 4: Iterative selection of high-scoring bins and updating of scores of remaining partial candidate bins. The output comprises non-redundant sets of high-scoring bins from different input predictions. From [27].

DAS Tool.

Another approach to improve the effectiveness of MAGs reconstruction is to perform a second assembly, posterior to the binning process. In the methods following this approach, reads are aligned to the bins and tagged with the bin they align on (fig. 2.7). The reads aligning to the same bin are then assembled together, following a supervised assembly step. Metawrap includes in its set of MAGs reconstruction modules a bin re-assembly step [26].

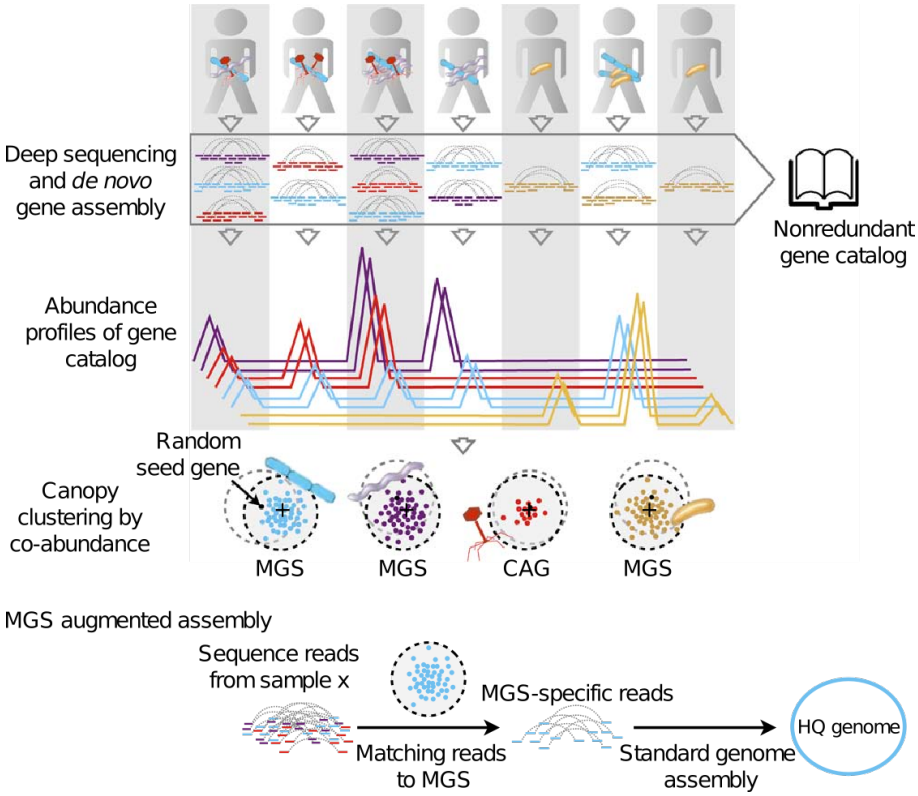

Figure 2.7 **Canopy clustering algorithm.**
The bins are reconstructed using correlation of co-abundance profiles of the contigs. Bins are then considered as CAGs (Co-abundance gene groups) or MetaGenomic Species (MGS) depending on their size. A post-processing step includes reassembly of the reads which have mapped to the same MGS, following a semi-supervised assembly, to reconstruct high-quality genomes. From [145].

2.3 Limits in MAGs reconstruction

2.3.1 Limits in SCGs-related approaches

SCGs have demonstrated their usefulness in the assessment of the quality of bins, or in the post-processing refinement step. However, if SCG-based quality estimators can detect redundant contamination with high sensitivity, they are less sensitive towards non-redundant contamination, since they only consider inventories of expected SCGs as a whole, ignoring potential disputed lineage assessments inferred from individual genes [180]. Especially, each set of SCGs has to be preliminary identified by scanning several reference genomes from a wide range of lineages, and as such cannot be considered exhaustive. This significantly limits their efficiency to assess quality of genomes assigned to taxa which are absent from their reference database [195]. Lineage-specific SCGs, particularly used in CheckM, are not evenly distributed across the genome, but locally clustered, further limiting their representation of the query genome. Recently, a new tool, GUNC [14] has been developed, estimating contamination of bins by the analysis of all the genes detected in their sequences. This method has shown a more precise assessment of non-redundant contamination of the bins, revealing higher levels of contamination of MAGs present in public databases [14]. The uniqueness of SCGs may also be questioned, as archaeal lineages belonging to the Asgard group have shown to borrow several copies of genes considered as SCGs in several bacterial taxa, notably genes coding for ribosomal proteins [196].

2.3.2 Relevance of binning metrics

A main limitation to all of the genome binning tools cited above is that they have to discard short contigs from the dataset, because both the composition and the coverage features become unreliable from short contigs (with the length threshold varying depending of the tool) reducing their recall values. In order to retrieve these short contigs and to integrate them in the binning process, several tools based on the exploration of the assembly graph have been developed in the recent years. These tools rely on the fact that contigs connected to each other in the assembly graph are more likely to belong to the same species [111]. GraphBin identifies the unbinned contigs connected to already-binned contigs in the assembly graph through a label propagation algorithm, and integrates these contigs to refine the bins [197]. RepBin follows a constraint-based representation of the graph assembly to perform genome binning, based on the presence of SCGs in the con-

tigs [176]. This tool both uses SCGs to infer the number of *labels* (*i.e.*, bins) to cluster the contigs into, and integrate the SCGs into pairwise cannot-link constraints, expecting that contigs carrying the same marker gene should not be binned together.

2.3.3 Difficulties to reconstruct the pangenome

The accessory genome represents the fraction of a genome that carries non-essential genes, *i.e.* genes that are not shared by all strains within the same species, or are exclusive to one unique strain [198]. It was defined as opposed to the *core genome*, which contains the genes coding for essential functions of the organism, such as the SCGs. The combination of the core genome of a species and the individual variants of its accessory genome represents the pangenome of this species [198]. A deep, even if not exhaustive, exploration of the pangenome of a species may imply to reconstruct numerous variants of this genome.

A MAG may be apparented to a consensus genome, as the binning process tends to eliminate numerous variants that may exist within a community. Globally, binning tools exhibit difficulties to recover genomes while closely related species or strains are present within a community [175]. Notably, the clustering of contigs into an exclusive bin completely hinders the possibility to efficiently reconstruct genome variants. SCGs-related methods may emphasize ths limitation, notably by considering the presence of variants genes as contamination [180], and thus are not relevant to assess the quality of accessory regions, within which there are no SCGs.

Different approaches have already attempted to reconstruct pangenomes. MSPMiner, instead of binning contigs, clusters co-abundant or partially co-abundant genes to reconstruct Metagenomic Species Pan-genomes (MSPs) [29]. Variation graphs represent an alternative representation of a genome that allows a better integration of existing variants [199]. This representation has been integrated to reconstruct bacterial strains from metagenomic samples [17].

2.4 Conclusion

During the recent years, several computational and algorithmic advances have helped a better integration of information from the increasing volume of genomic data. Devel-

opments in assembly algorithms and related methods have led to significant improvements in the accuracy and efficiency of genome assembly, and have well adapted to the specificities of metagenomic assembly. These successes are measured by more contiguous sequences, an increased numbers of predicted genes, a reduction of breakpoints and rearrangements within contigs, and error limits close to the expected sequencing substitution rate. Genome binning tools, performing the key step of the reconstruction of the MAGs, have explored through intrinsic features of contigs to perform their clustering with high confidence, without relying on reference genomes. Their clustering approaches have differed, and currently, tools integrating both compositional and coverage features represent the most efficient binning tools. The problem of genome binning still remains difficult, and all current approaches have their own limits. MAGs still remain more fragmented than genomes retrieved from classic genomic approaches, and their reconstruction still have difficulties to integrate regions belonging to the accessory genome. Based on this observation, a constraints programming (CP) approach may appear as a relevant answer, as it could allow the exploration of several equivalent *good* solutions to the problem. Some of previous binning approaches have integrated genomic features as constraints, in order to perform a constrained clustering approach. But there is still not a CP-based approach to perform genome binning. In the next chapter, we will describe a constrained-clustering in a CP framework, and how it would perform to bin contigs into genomes, and whether this approach answers the limits of currently existing binning software.

Chapter 3

MAGs reconstruction using constraint logic programming

Preamble

In this chapter, we will present our approach to perform contig binning in logic programming. The objective of this approach was to proposed a binning approach that would allow to better take into account of the possible solutions to a binning problem. We thought it would help to better catch the variability of variants inside a metagenome, and notably the strains diversity. The first part introduces the concepts of constraints programming and logic programming, and ASP. Secondly, we will detail the model we built in ASP, the constraints we implemented and the results we obtained, compared to another binning tool.

3.1 Introduction to declarative programming

The resolution of a problem through the description of the resolution process, generally by means of an algorithm, is called the *imperative* paradigm. For many programmers, the imperative paradigm may represent the classic approach to resolve a problem. The *declarative* programming takes the opposite point of view, in which the user has no interest in the description of *how* to solve the problem, but instead explains *what* is the problem. Therefore, the main purpose becomes the modeling of the problem, in order to proceed to its resolution. Declarative programming languages tend to eliminate side effects, and heavily rely on mathematical logic. The declarative programming paradigm itself represents a wide field in computer science, and gathers several subdomains such

as constraints programming (CP), logic programming, functional programming, or descriptive programming. Consequently, different languages have been developed in each of these domains: Prolog for constraint and logic programming, Haskell for functional programming, or HTML for descriptive programming, for example. This chapter will focus on logic programming and constraints programming.

Side effects

In computer science, an operation, function or expression is said to have a side effect if it modifies some state variable value(s) outside its local environment, *i.e.* if it has any observable interaction other than returning a value to the invoker of the operation. Imperative programming languages use side effects as a mean to update the state of the system, but they can impede the predictability of the behaviour of the program, or the reuse of functions. Declarative programming languages, by describing the system's state, tend to eliminate or at least minimize these effects. This feature makes them strongly similar to mathematical logic.

3.2 Constraint programming

3.2.1 Definitions

A constraint can be defined as an expression that discards solutions, which would be acceptable otherwise. It represents an efficient and straightforward means to verify that all solutions respect a property [200]. A constraint $c(x_1, ..., x_n)$ typically implies a finite number of decision variables $x_1, ..., x_n$. Each variable x_j may then take any value v_j from a finite set D_j, which is called the *domain* of the variable. The constraint then defines the relationship R_c, which is satisfied if all the observation variables have their values included in the relationship. A *constraint satisfaction problem*, or CSP, is then a finite constraints set $C = \{c_1, ..., c_n\}$ on a set of variables $\{x_1, .., x_n\}$. The CSP is said *satisfiable*, or feasible, if there exists a tuple $\{v_1, ..., v_n\}$ of values that simultaneously satisfied all the constraints in C. The tuple of values represents here a solution to the CSP. On the other hand, if such a tuple does not exist, *i.e.* if there is no assignment of values to variables from their respective domains for which all constraints are satisfied, then the problem is

unsatisfiable. Constraints may be categorized as instance-level constraints or cluster-level constraints. Instance-level constraints can be either *must-link* constraints, which formalise a relationship in which two items must be put in the same cluster if the relationship is verified, or *cannot-link* constraints, forbidding two items to be put in the same cluster if the relationship is verified [201]. On the other hand, cluster-level constraints apply on the characteristics of the group of items itself, such as its maximum size.
A language allowing constraint programming typically considers two steps. First, it models the space search to define which are the possible solutions to explore, and second, it applies the constraints to eliminate the undesired solutions from the search space.

Procedures in CP

For Bockmayr, CP has tried to weave imperative and declarative programming, although these two paradigms seem completely exclusive, as the declarative programming is static, while the imperative programming is dynamic. The constraints used to describe the solutions to be obtained may indeed be seen as an inclusion of a procedure within a more global declarative framework [200]. To each constraint is thus associated an algorithm to remove from the space search infeasible solutions.

3.2.2 Resolution of the CSP

Constraint satisfaction problems are combinatorial by definition. Therefore, for many categories of CSP, an efficient algorithm is unlikely to exist, as these problems are NP-complete. This means that, unless $P = NP$, an algorithm that guarantees to find a solution that satisfies all constraints would have a worst-case exponential time complexity, as it would be enumerative. A CSP can be seen as a generalisation of the Boolean satisfiability problem (SAT) [202]. SAT is a widely used modelling framework, and thus is the core of a large family of combinatorial problems. It consists in deciding whether a propositional logic formula can be satisfied, given suitable value assignments to the variables of the formula. Many CSP resolution works have been performed on binary CSP, leading to the development of efficient SAT solvers. CSP solvers use different techniques for CSP resolution, which can be categorised as backtracking methods, consistency techniques, and constraint propagation.

Backtracking

In practice, it may be sufficient to find a solution at a reasonable expense which satisfies not *all* the constraints, but *most* of the constraints. This is particularly the case if the problem contains soft constraints or an objective constraint. At the end, if all constraints have been satisfied, the solution is said to be *exact*, otherwise, the solution is *approximate*. Backtracking is a technique that uses approximate solutions as potential candidatesto possibly reach the exact solution. The first step is then to enumerate the potential candidates of the problem, which could be completed in various ways to give all the possible solutions. These potential candidates will then be placed as the nodes of a tree called *potential search tree*. In this tree, each candidate is the parent of candidates that differ from it by a single extension step, and the leaves of the tree are the candidates that cannot be extended any further. Backtracking tries to extend a potential candidate c that specifies consistent values for some of the variables, towards a complete assignment, by incrementally choosing a value for another variable, consistent with the values in c. When all the variables relevant to a constraint are instantiated, the validity of the constraint is checked. If c is a valid solution to the problem, *i.e.* if all constraints have been satisfied, then it is returned to the user, and all the sub-trees of c are enumerated as valid solutions. If c violates any of the constraints, then all the descendant candidates are ignored, leading to the pruning of the sub-tree rooted in c (see fig. 3.1 for an illustration of the 4-queens problem). Then, the algorithm backtracks to the most recently instantiated variable that has alternative variables available. This algorithm allows to remove numerous candidates from the search. Thus, backtracking is strictly better than a greedy search in which all potential solutions would be explored.
The Davis-Putnam-Logemann-Loveland (DPLL) algorithm [203], itself an enhancement of the Davis-Putnam algorithm, is the main backtracking algorithm used in CSP resolution, notably in SAT solvers. The DPLL alorithm is both complete and sound, *i.e.*, it will always find the solution of the problem if this solution exists, and all the returned solutions are guaranteed to be true. Its main limit is *thrashing*, which is caused by the occurrence of the same inconsistency several times within the search tree. This problem then causes repeated failure during the search, which is not avoidable, causing a waste of time [204].

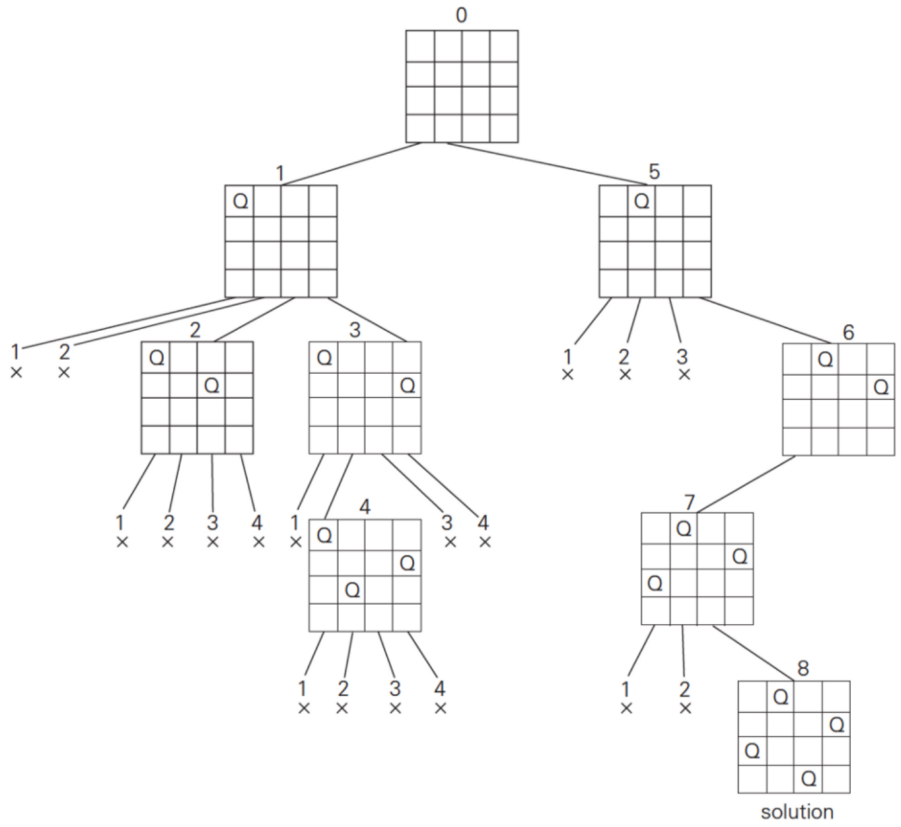

Figure 3.1 **Backtracking in the 4-queens problem.**
The n-queens problem is a well-known problem consisting of positioning n queens on a $n \times n$ chessboard, with no queen being placed on a square attacked by another queen. This problem is easily translated into a CSP. Here, backtracking is used to find the solution of a 4-queens problem. The number above each configuration is the order of visit of each configuration, while the numbers below represents the row assigned to the next queen, the n-th queen being position on the n-th row. A cross under a number represents an impossible assignment, resolving to a backtracking. Thus, starting from candidate 1 "queen 1 on column 1", it is thus impossible to assign "queen 2 on column 1", nor "queen 2 on column 2". The candidates descending from these candidates are then left unsearched, as they would all lead to unsatisfiable assignments. Next potential values assignment are then "queen 2 on column 3" and "queen 2 on column 4", in configurations 2 and 3, respectively. However, all assignments for queen 3 after configuration 2 are unsatisfiable: the search then needs to backtrack to the previous potential candidate, the configuration 3. From there, it continues to assign values to the next queens, backtracks whenever it reaches unsatisfiable assignments, until it finds a solution, the configuration 8.

Consistency checking

Consistency checking methods may discard many inconsistent candidates at a very early stage, and thus greatly shorten the search for consistent candidates. They were first introduced to improve the efficiency of picture recognition software, in which labeling all the lines of a picture in a consistent way is required. The number of possible combinations is very high, while very few are consistent. Consistency checking methods have been used on a wide variety of hard search problems. The idea behind consistency checking is to remove values from the domain of variables, if there is no consistent values satisfying a constraint in the other domains [204]. These methods are rarely used alone to resolve a CSP. By increasing the complexity of the applied consistency technique, more inconsistent values are discarded from the CSP. However, consistency techniques generally fail to eliminate all inconsistent values from the CSP, meaning that a search is still needed to complete the resolution [204]. The more complex consistency technique, the n-consistency, would ideally be able to find the solution of a CSP with n variables without performing any search. However, the complexity of the operation makes it more costly than a backtracking search. Then, consistency techniques are generally used to enhance backtracking search.

Consistency checking: example

Let $A < B$ be a constraint C between the variables A and B, with domains $D_A = \{3, .., 7\}$ and $D_B = \{1, .., 5\}$ respectively. For some values in D_A, no consistent values exist in D_B satisfying the constraint C. Such values can be removed from both domains, without risking the loss of any solution. At the end, we get reduced domains $D_A = \{3, 4\}$ and $D_B = \{4, 5\}$. However the reduction does not remove all inconsistent pairs ($A = 4, B = 4$ for instance, is still in domains). But for each value in D_A, it is possible to find at least one consistent value in D_B, and vice versa.

Constraint propagation

Constraint propagation helps to filter possible values from the domain of each variable, leading to the discarding of all the interpretations carrying the aforementioned values. Generally, as a constraint problem contains several constraints, reaching consistency for one constraint may cause some other constraints to become unfeasible, even if they were

feasible beforehand. Domain filtering has to be applied several times to constraints sharing common variables, until no further domain reduction is possible. Constraint propagation is used in CSP resolution as a means to embed backtracking and consistency checking, in order to improve the efficiency of the solvers. These hybrid approaches are usually called *look-ahead strategies* [204].

3.2.3 Optimization problems

A constraints optimization problem implies an objective function $f(x_1, ..., x_n)$ which needs to be maximized or minimized on the whole set of feasible solutions. Although algorithms for solving CSPs aim to simply reach a feasible solution, they can be adapted to find an optimal solution. This can be done through the addition of an objective variable, which would represent the objective function. Once an initial feasible solution is found, a new objective constraint is introduced to the problem, implying that the value of the objective variable must be better than in the initial solution. This process can be performed iteratively, until the problem becomes unsatisfiable: the last satisfiable solution is then an optimal solution [205]. The number of iterations needed to find the optimal solution is not fixed, and depends on the quality of the initial solution. This initial solution may generally be found by applying heuristic method.

3.3 Logic Programming

Logic programming represents a sub-paradigm of declarative programming, in which sentences express facts and rules following formal logic [206]. A sentence in predicate logic is defined as a finite set of clauses [206], which can be either rules or facts. Facts represent the knowledge base of the system, which can be queried. In a logic programming language, *rules* are written as logic clauses, such as:

H is true if B_1,.., B_n are true.

In such a statement, "H is true" is called the *head* of the rule, while "B1,..., Bn are true" is called the *body* of the rule. A fact is then a rule consisting of only a head: "H is true".

3.3.1 Representation of logic

In a logic programming framework, the relations between the different objects are materialized as *predicates*. For instance, if one wants to represent the color of a node in a graph, we can use a predicate `colornode(node, color)`. This representation is different from propositional logic, which relies on variables which may be true or false. Besides, in propositional logic, any fact is unique and therefore must be stored independently from the others. In predicate logic, facts are represented thanks to quantifiers: what is needed is the list of objects to which the predicate applies. Predicates are used to build atomic formulas, also named *atoms* which state true facts. Examples of predicates stating facts may be, following a Prolog-like syntax:

```
hero(heracles).
god(zeus).
enemies(magneto, xavier).
```

Which can be interpreted as *Heracles is a hero*, *Zeus is a god*, *Xavier and Magneto are enemies*. These facts are true, and are also *atoms*, which means they do not depend on any other logical connectives, *i.e.* they do not have strict subformulas. The two first facts contain only one item, while the latter contain two items (`magneto` and `xavier`). Thus, predicates `hero` and `god` have an *arity* of 1, and predicate `enemies` has an arity of 2. The arity of a relation thus represents the number of items concerned by the relation. Predicates can be alternatively noted following the form *<name_of_predicate>/n*, *n* being their arity degree. For the above examples, this notation would be *hero/1*, *god/1*, and *enemies/2*. Rules, on the other hand, contain a logical relationship, and are built following the form: `Head :- Body`. The interpretation of a rule is *If the body is true, so is the head.* Examples of rules may be:

```
immortal(zeus) :- god(zeus).
mortal(socrates) :- human(socrates).
```

Which represents the statements *If Zeus is a god, then Zeus is immortal*, and *If Socrates is a human, then Socrates is mortal.*
The whole set of items over which the variables of interest may range and that the language needs to represent is called the *universe of discourse*. Predicates establish relations

between combinations of objects from the universe of discourse. The entities of the universe of discourse may be represented through *terms*, which may be variables, noted as upper-case, or constants, noted as lower-case. Variables represent the whole set of possible interpretations of the language, which can be infinite. A first step before the resolution of the problem would be to find a subset of possible solutions composed only of constant terms deducted from the language, instead of all the elements from the universe of discourse [207].

3.3.2 Prolog and constraints in logic programming

CP aims to provides new approaches to solving discrete optimization problems, while at the same time being embedded into a high-level programming language. It first appeared in the form of constraints logic programming, with logic programming as the underlying programming language paradigm [200]. One of the main logical and declarative programming language is Prolog, or PROgramming in LOGics. Its development came to the end of a long history of research on theorem provers and automated deduction system, through the 1950s and 1960s. The first version of Prolog was developed by Colmerauer in 1971 [208]. Prolog is based on first-order predicate logic to describe the relations between items. It has been enhanced with various extensions since its first release, notably from the constraints logic programming community, to include constraint satisfaction concepts in the language. Search and backtracking are also built directly into the language, which greatly facilitates the development of search algorithms. Prolog may also support imperative features when the logical paradigm becomes inconvenient, which allows the language to exploit deliberate functions' side effects. It is currently able to process large amounts of data.

3.4 Answer Set Programming

Answer Set Programming (ASP) is a form of declarative programming oriented towards difficult, primarily NP-hard, search problems [209]. ASP is based on the stable models semantics of logic programming [210], which applies the ideas of nonmonotonic logic [211] and default logic to the analysis of negation as failure [212]. A difference with classic CP approaches, is that in ASP, search problems are reduced to computing stable models, also called *answer sets*, and answer set solvers, *i.e.* software generating stable

models, are used to perform search. An answer set is defined as a minimal set of atoms that satisfies the set of rules and facts of the problem [213]. This problem resolution is similar to a generate-and-test approach, coupling choice rules that describe potential solutions, a step called *grounding*, and constraints that eliminate unsuitable potential solutions (fig. 3.2).
The search algorithms used in the design of many answer set solvers are enhancements of the DPLL algorithm, and as such, process similarly as efficient SAT solvers [209]. The main advantage of these algorithms is that they are guaranteed to end, as opposed to the resolution in Prolog [209]. ASP language allows the modeling of a problem relying on logical predicates, and its syntax is heavily inspired by lparse [214]. lparse was originally created for answer set solver smodels [215], and its syntax shared strong similarities with Prolog.
The applications of ASP are numerous, for example in robotics [216], optimization in plannings and diagnosis, as well as in computational biology, to determine phylogenies [217], design of gene regulation graphs [218], or predict protein structure [219]. ASP also has industrial applications, notably in employee's management [220], or e-tourism and e-medicine [221]. Currently, there are several existing versions of ASP. The version chosen in this work was the implementation in POTASSCO (Potsdam Answer Set Solving COllection), which may represent the most-used version of ASP. The main software of interest for this study is clingo [222], which is a combination of two different software, gringo [223], which is the grounding software, and clasp [224], the solver.

3.4.1 Generation of the search space: the grounding

Theory

As predicate logic can lead to an infinite number of interpretations, it may seem an impossible task to reach the actual solution of a problem. However, there exists a subset of interpretations called *Herbrand interpretations*, in which all constants are assigned very simple meanings [225]. Under certain conditions, evaluating the Herbrand interpretations is enough to check the satisfiability of a set of sentences: this is the Herbrand's theorem. These interpretations rely on the concept of *ground* terms [226].

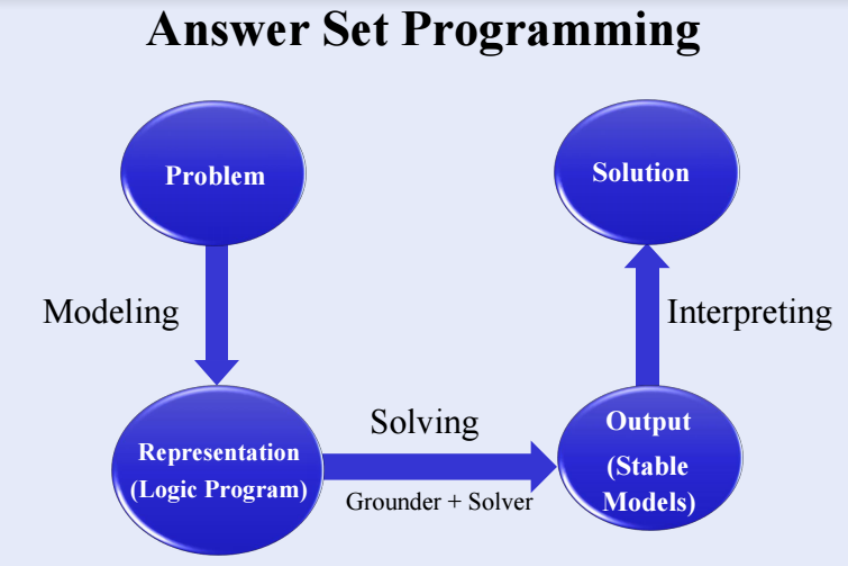

Figure 3.2 **Representation of problem solving in ASP.**
The problem is first modeled into a logic program, *i.e.*, is implemented to a computer program. Then, this logic program is interpreted by the grounder, which translates the program choice rules into a grounding program, composed of models carrying only constant values, and no variables. This grounding program is then processed by the solver, which eliminates the inconsistent models based on the constraints rules within the logic program. At the end, if the solver finds at least one stable model, *i.e.* a model whose values satisfy all the constraints, then it outputs all the stable models as solutions of the problem. From https://stackoverflow.com/tags/answer-set-programming/info

Herbrand's theorem and ground terms

A set of sentences in a predicates language without quantifier is satisfiable (verified) if and only if there is a Herbrand model, *i. e.* a model containing only ground terms, that satisfies these sentences.

A ground term is a term that does not contain any variables, but only constant symbols [226]. Similarly, an expression in predicate logic is said to be *ground* if, and only if, it contains only ground terms, *i.e.* the expression does not contains any variables. Ground terms may be used as a means to reduce the number of models/interpretations to test for consistency. First, the Herbrand Universe is a reduction of the universe of discourse. For a set of sentences in predicate logic, with at least one constant object, the Herbrand universe is the set of all ground terms that can be generated only with the constants of

the set of sentences [207]. If there is zero constant object, an arbitrary constant can be added. The Herbrand Universe is then used to obtain the Herbrand base. For a set of sentences, the Herbrand base is the set of all ground sentences that can be built solely with constants from the Herbrand Universe [227]. In other terms, it is the set of sentences $r(t_1, ..., t_n)$, where r is a n-arity constant, and $t_1, ..., t_n$ are ground terms belonging to the Herbrand Universe. The Herbrand base may thus been seen as the logic program using only constant terms. Finally, a Herbrand interpretation is an interpretation in which: i) the universe of discourse *is* the Herbrand Universe, ii) each constant object is interpreted as itself, and iii) every function symbols is interpreted as the function that applies to it [225].
The purpose of the grounding step is thus to generate the Herbrand interpretations of the problems, which would also be called more simply ground models. The generation of the ground models is performed through the application of choice rules of the logic program. Once the ground models have been generated, the next step is to check whether the interpretations satisfy the constraints. Generally, the grounding step is performed by a dedicated software. In the POTASSCO version of ASP, the grounding software is called gringo [223]. The set of ground models output by the grounder represents the *grounding program*, which is the input for the second software, the answer set solver.

3.4.2 Searching for answer sets

A Herbrand interpretation I would be a Herbrand model M if the elements belonging to the interpretation satisfy all the sentences of the language, *i.e.*, if for each sentence of the language, there is a subset of I that validates the interpretation. Thanks to the Herband theorem, one can also check the consistency and satisfiability of a set of sentences in a finite time: it would only need to test the Herbrand models. The resolution step is thus to check iteratively the Herbrand interpretations, and to discard any interpretation that does not satisfy the set of constraints rules.
The format to write constraints in ASP is a rule without a head, beginning directly with the operator `:-`. Thus, an ASP constraint is always seen as a restriction, as a no head rule always implies that the statement is False. As an example, to write the constraint $a \geq 3$, then the ASP syntax would be `:-` $a < 3$., "the value of a must not be strictly inferior to 3". It is however possible to use a negative statement (with the *not* word) to implement must-link constraints in ASP. An analogous version of the previous constraint would then be `:- not` $a >= 3$., "the value of a must not not be superior or equal to 3", thus "a

must be superior or equal to 3". ASP constraints are similar to SAT constraints, and they are used by the solver to remove models from the grounding program. The addition of constraints in an ASP program then leads to the reduction of the number of stable models obtained. However, even efficient constraints will not prevent a high computation time search, caused by a very large grounding program.
clasp [224] is the ASP solver developed by the POTASSCO lab. It was originally thought as a tool combining the high-level capacities of ASP with state-of-the art techniques from the Boolean constraints solving [224]. clasp relies on the Conflict-Driven Clause Learning (CDCL) algorithm [228], which has been employed for satisfiability checking in SAT solver. The CDCL algorithm is based, as its name suggests, on clause learning, which means that, when it meets an inconsistent value assignment, CDCL will keep that information. This information will then be added as an additional clause to resolve future inconsistency conflicts (fig. 3.3), allowing pruning of larger sub-trees from the search tree. The *backjumping* (instead of backtracking) in the search tree is also guided by the variable which was the source of inconsistency. Thus, it will not necessarily return back to the previous potential candidate: backjumping is thus said non-chronological.
clasp also supports parallel search using multi-threading with shared memory, parallel optimization, and includes declarative support for domain heuristics [229]. The optimization allowed by clasp is performed after having found a stable model for the problem. The optimization may thus be considered as a refinement of the satisfiability search, by adding new constraints that will eliminate further stable models, until it finds an optimal solution. With clasp, each suitable solution is associated to a score, and the purpose of the optimization step is to minimize this score [224].

3.5 Model used

3.5.1 Definition

The binning problem is formalized as a constrained clustering problem, coupled with a bi-objective optimization. The aim of this model is to cluster the contigs such that the clusters are the most well-defined. Then, the clustering has to be performed such that the dissimilarity within clusters is the smallest as possible, and the dissimilarity between two clusters as high as possible. Dao and colleagues have previously studied the formalization of a constrained clustering approach as a constraint programming problem [230]. The

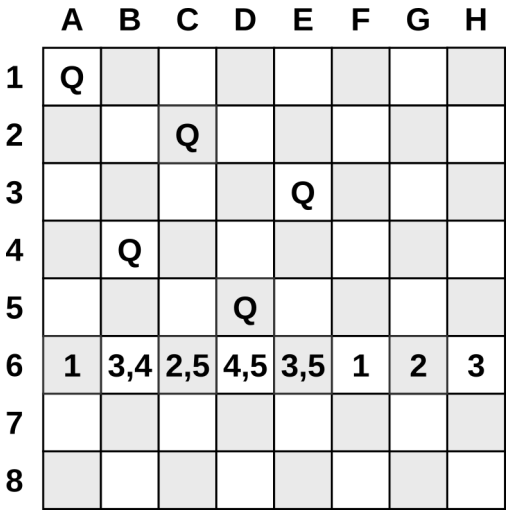

Figure 3.3 **Clause learning in the 8-queens problem.**
Queens are noted with the number of the row they are placed (*e.g.* queen 1 on row 1). The numbers in row 6 indicate the assigned queens that the corresponding squares are incompatible with. In this configuration, it is possible to realise that changing the position of queen 5 will not resolve the inconsistency, as the cases blocked by queen 5 are also blocked by other queens. Thus, backtracking to other assignments for the position of queen 5 will just end to a waste of time. Learning this clause will then allow the search to backjump to queen 4, which is the closest queen we can move to allow to position queen 6 on D6, to continue to search for further positions for the remaining queens. In this example, the queens represent the variables of the CSP, while the queens' positions represent the values. From [204].

problem was also an optimization problem, in which the objective was to both minimize the diameter of the clusters formed, and to maximise the distance separating two clusters, named margins.
Consider a clustering problem consisting in the clustering of n point in k clusters. The diameter D_c of a cluster c is defined as the maximum distance between two points o_i and o_j belonging to c (Equation 3.1).

$$D_c = max(d(o_i, o_j)); c \in [1, k]; O - i, o_j \in c \tag{3.1}$$

The margin $M_c c'$ between two dictinct clusters c and c' is defined as the minimum distance between two points o_i and o_j, belonging to c and c' respectively (Equation 3.2).

$$M_{cc'} = min(d(o_i, o_j)); c < c' \in [1, k], o_i \in c, o_j \in c' \tag{3.2}$$

3.5.2 Adaptation to the binning problem

In this work, we adapted the CP-constrained clustering approach to the binning of contigs, and implemented it in ASP. In order to compute distance between contigs, two metrics widely used by state-of-the-art binning tools were used: i) the compositional metric, *i.e.*, the TNF distance (Equation 3.3), was used to assess the clusters' diameters, and ii) the coverage distance (Equation 3.4) was used to compute the margins between the bins. Both distances were computed using metabat2 [16]. The definition of the TNF distance between two contigs p and q is a simple euclidean distance:

$$TNF_{p,q} = \sqrt{\sum_{i=1}^{136} (p_i - q_i)^2} \tag{3.3}$$

With p_i and q_i being the number of occurrences of the i-th tetranucleotide in contig p and q, respectively, and $i \in 1;136$. Because of the double strand nature of DNA molecule, each tetramer has both a *forward* and *reverse* version. We consider only the best lexicographically-ranked tetramer between the *forward* and *reverse* versions, which means that the frequencies of both *forward* and *reverse* are added to the same tetramer. For example, the tetramer $ACCT$ has the *reverse* version $AGGT$, so any occurrences of both $ACCT$ and $AGGT$ in a contig will be counted as $ACCT$, which is the best tetramer ranked in the couple. This allows to reduce the size of the TNF vector to compute TNF distance between contigs from 256 to 136. See also Chapter 2, section 2.2.2 "Unsupervised binning", p. 33.
The definition of the abundance distance bewteen two contigs p and q is:

$$abd_{p,q} = \frac{1}{2} \int |\phi_{\mu_p,\sigma_p^2} - \phi_{\mu_q,\sigma_q^2}| \tag{3.4}$$

Which is the unshared area under the curves of the distribution of abundances of the two contigs, each abundance being assimilated to a normal distribution ϕ of mean μ and standard deviation σ.
ASP, as several declarative programming languages, offers the advantage of enumerating several suitable solutions to the problem. The exploration of several equivalent solutions to a binning problem may represent a relevant answer to one of the main limitations of the current binning tools, which is that the output MAGs are consensus genomes. With ASP, we could in theory find every equivalent consensus genomes by finding all answer sets, which might then be relevant to capture pan-genomes and identify shared contigs

and/or genes between equivalent MAGs. Thanks to the search algorithm of clasp, it will also guarantee to find a stable model, if such a model exists.

3.5.3 Constraints

Single-Copy Genes (SCGs) are genes present in microbial genomes which are considered to be universal and present in the genome in a single copy (see also Chapter 2, section 2.2.4 "Quality assessment : from bins to MAGs", p.51). Usually, the presence or absence of these genes has been used to assess the quality of the reconstructed MAGs [163, 180, 231], *i. e.*, their completeness and their contamination. We integrated the presence of SCGs within contig's sequences as contraints of the ASP model. Because of their unicity within one genome, the presence of SCGs may be translated as a cannot-link constraint: two contigs must not be placed in the same cluster if they both carry copies of the same SCG.
Standard quality criteria have been proposed in order to define which bin can be considered a MAG [179]. Following these criteria, a bin must reach at least a "medium" quality, *i. e.*, it must reach a completeness of at least 50%, and a contamination level of at most 10% to be considered as a MAG. By adapting these quality criteria to the ASP model, we could implemented two cluster-level constraints: i) a bin must contain 50% of the total number of SCGs, and ii) a bin must not have copies of SCGs counting for more than 10% of the total of the SCGs set.
The estimation of completeness and contamination were formalized in the study of Parks and colleagues [180]. The estimation of completeness is then:

$$\frac{\sum_{s \in M} \frac{|s \cap G_M|}{|s|}}{|M|} \tag{3.5}$$

Where s is a set of collocated marker genes, M is the set of all collocated marker sets s, and G_M is the set of marker genes identified in a genome. Genome contamination is estimated from the number of multicopy marker genes identified in each marker set:

$$\frac{\sum_{s \in M} \frac{\sum_{g \in s} C_g}{|s|}}{|M|} \tag{3.6}$$

Where $C_g = N - 1$ for a gene g identified $N \geq 1$ times, and 0 for a missing gene. However these equations may be simplified, when estimating both the completeness and the con-

tamination without arranging SCGs into collocated sets. With this simplification, each marker gene is assigned to its own set, meaning that $|s| = 1$. That means that the completeness can simply be measured by counting the number of SCGs within a bin, and that each duplicated copy of a SCG equally increases the contamination of the bin.
In the binning problem, as the set of SCGs contains the 10 genes to determine the taxonomic affiliation of contigs with the tool mOTUs2 [232], the completeness threshold is 5, while the contamination threshold is 1. Following the simplification of formulas of completeness and contamination [180], the completeness constraint may then be formalized as:

$$\forall C, \forall M, \sum_{s \in M} S \cap G_M > 5 \tag{3.7}$$

Where C represents a cluster, M the whole set of SCGs sets, S a SCG, and G_M one set of SCG. "A set of SCG" in this context means a set of SCG assigned to one particular taxonomic lineage. While the contamination constraint may be formalized as:

$$\forall C, \forall M, \sum_{s \in M} \sum_{g \in s} C_g < 2 \tag{3.8}$$

Where C is a cluster, M is the whole set of sets of SCGs, S is a SCG, g is a copy of S, and C_g represents the SCGs found in the cluster C. It should be noted that all pairwise TNF and abundance distance have been computed beforehand, using metabat2, because of its fast computation. The distances values obtained were then rounded, as ASP works only with integers. Indeed, the computation of the distances through ASP would represent a tremendous computational effort, and maybe would not had even been possible. Even though it removed a sensible amount of work to the whole procedure, it was at the cost of two drawbacks. First, the predicates for the distances had to be written on disk, representing $2 * n^2$ predicates in total, with n the number of contigs. Thus, when the datasets began to reach a thousand points, both files containing the predicates distances weighted several megabytes. These values were still easily manageable, but one should remember that the metagenomic datasets are generally composed of tens to hundreds of thousands contigs. Second, the parsing and the integration of these files also took time, representing a hard limit under which the grounding time could not go below. Nevertheless, these drawbacks were still largely compensated by the gain of time allowed by preliminary computation of the distances.
The assignment rules limited a contig to be put into one unique cluster, in order to limit

the number of combinations. With this minimal model, the answer sets were composed of clusters containing a unique contig. Indeed, because of the objective functions of the model, the best solution would be to put each of the k contigs, k being the number of clusters, with the highest abundance distance with the others in one of the k bins. Each bin diameter would be null, and the margins could not grow further. Each contig added in a solution tends to increased the value of diameters, and tend to reduce the value of the margins, going against the objective function. To avoid to reach these irrelevant solutions, several constraints were added in the model, concerning the contents of the clusters. These constraints were: i) a cluster must contain at least two contigs, ii) the sum of the lengths of the contigs within any cluster must reach a minimal threshold, and iii) all the contigs have to be clustered. Of these three constraints, the third has the most powerful effect on the limitation of the number of combinations. Because the dataset used for testing the model contained very long contigs, the first and second constraints are less efficient to filter unsatisfiable solutions. Indeed, in case of a contig with a length already above the length limit, the assignation of this contig to a bin cannot avoid that this bin contains only two contigs. The first and second constraints are however not removed, because the presence of redundant constraints may still help to fasten the resolution.

3.6 Tests and results

3.6.1 Input data

As already stated above, both the pairwise TNF and abundance distances between contigs has been computed beforehands, through metabat2 software. The model is thus composed of a series of predicates facts representing these input data. The 2-ary predicate `contig/2` associates a contig identifier with the length of the contig, and was written in the ASP code as:

```
contig(C, L).
```

With C an integer ranking from 1 to N, N being the total number of contigs, and L the length of contig C. C is randomly chosen based on the order of the contig in the assembly fasta file. It thus does not represent any biologically relevant information, and is guaranteed to be unique. The distances are represented with 3-ary predicates `tnf/3` and `abd/3`:

```
tnf(C1, C2, T).
abd(C1, C2, A).
```

Where $C1$ and $C2$ representes two distinct contigs (*i.e.*, $C1 \neq C2$), and T and A the value of the TNF distance and abundance distance between the two contigs, respectively. The information of the presence of the SCGs is also represented with a simple 2-ary predicate `scg/2`:

```
scg(C, G).
```

With C being the identifier of the cluster, and G being an integer identifier number for the SCG.

3.6.2 First model

The main rule of the ASP model was the assignment rule, which assigns a contig to a bin. As the model clustered each contigs only once, the assignment rule was implemented in ASP as:

```
#const k.
bin(1..k).
1{att(B,C) : bin(B)}1 :- contig(C,_).
```

The first line sets the constant k, which represents the number of clusters, using to the statement `#const`. Constants help to increase code readability, and also can be managed without modifying the code, directly from the command line, *e.g.*, the gringo option `-const c=t` allows the user to set the value of the constant c to t. In this example, k has no value, so each run with gringo needs to set a value for k using the command line, otherwise, the run would simply fail. The line `1{att(B,C) : contig(C,_)}1` ensures that for each contig, the number of bins to be attributed could not go beyond or below 1, as this limit is set both in front of the rbacket of the line, which represents the minimum number of clusters to assign a contig to, and the value after the closing bracket representing the maximum number. The second line then generates k predicates `bin/1`, each with a different value between 1 and k. In the third line, the assignment rule generates the predicate `att/2`, which establishes the relationship between a contig C and its bin B. The underscore character simply means that the value present at this position is not considered in the choice rule.

The computation of the diameter of each cluster was translated into ASP program, using TNF distance:

```
distancetnf(C1,C2,T,B) :- att(B,C1), att(B,C2), C1<C2, bin(B),
tnf(C1,C2,T).
diam(B, D) :- bin(C1), bin(C2), D = #max{T : distancetnf(C1,C2,T,B)},
D!=#infimum.
```

The first line generates the predicate `distancetnf/4`, which establishes the relation between two contigs $C1$ and $C2$ belonging to the same bin B and having a TNF distance of T. The second line performs the search for the diameter of a bin B, *i.e.*, the maximum TNF distance between all contigs belonging to B, thanks to the `#max` statement. The statement `#infimum` is the built-in statement that computes the greatest lower bound of all D values. The computation of the margins follows a similar formulation:

```
distanceabd(C1,C2,D) :- att(C1, B1), att(C2, B2), B1!=B2, bin(B1),
bin(B2), C1<C2, abd(C1,C2,D).
margin(C1,C2,M) :- bin(C1), bin(C2), M = #min{D : distancediff(C1,C2,D)},
M!=#supremum, C1<C2.
```

The first line generates the predicate `distanceabd/3`, which extracts the value D of the abundance distance stored in predicates `abd/3` of two contigs $C1$ and $C2$, belonging to bins $B1$ and $B2$ respectively, $B1$ and $B2$ being strictly different bins. The second line then processes to search for the minimal value M among all values D in predicates `distanceadb/3` for the pair of bins $B1$ and $B2$. The search for minimal value relies on ASP included statement `#min`. The `#supremum` statement is the built-in statement that computes the least greater bound of values of M.

As stated previously (see section 3.4.2 "Searching for answer sets", p.72), clasp is also able to resolve optimization problems, and as such, optimization statements are already included in ASP. The implementation of the objectives of the problem are then:

```
#minimize{D@1 : diam(C, D)}.
#maximize{M@1 : margin(C1, C2, M)}.
```

The value after the `@` represents the priority of the objective function, the objectives being applied by priority ranked by ascending order. Because the model does not favour one or the other objective, the priorities stayed equal.

The constraint related to the minimum sum of contigs lengths was written in ASP as constraints rules:

```
#const lengthmin=200000.
attlengthmin(B,C,L) :- att(B,C), contig(C,L).
:- bin(B), not #sumL : attlengthmin(B,_,L)>lengthmin.
```

the first line created the constant `lengthmin`. Then, the second line creates a predicate `attlengthmin/3`, which stores the information of the bin B attributed to the contig C, and its length L. As these information were already present in the fact rules, this line may look useless. Its purpose is to simplify the writing of the third line, which represents the constraint itself. This constraints forbids a bin B if the sum of length of all contigs' length in it does not reach *lengthmin*.

Our first attempts may be resumed as a calibration step, as we did not have any insight on the time complexity of the resolution of the binning problem with ASP. We then just tested the most simplest model to randomly-generated toy datasets of growing size, from 200 contigs to 800 contigs. The number of clusters k was set to 5. The ASP model did not scale at all, with computation time quickly reaching hundreds of hours to obtain an optimal solution when the number of contigs grew above 200 (Table 3.1). The time to list all stable models *i.e.*, all solutions) grew even more, with an already unreachable time for a dataset containing as low as 400 points. If this was confirmed, this framework would not represent a relevant approach to the binning problem, as metagenomic datasets generally contain several hundreds of thousands contigs.

Contigs	T1	T2
200	7h28	37h28
400	128h	unknown
600	379h	unknown
800	975h	unknown

Table 3.1 Resolution time for binning randomly-generated toy datasets.
For each attempt, the contigs had to be binned in 5 clusters. Contigs = number of contigs within the toy dataset; T1 = Time to find one optimal stable model; T2 = Time to find all optimal stable models.

3.6.3 Improvements of the model

ASP lists *all* the possible solutions, not considering permutations of predicates in solutions. Thus, two solutions consisting of clustering all the contigs in the same clusters, but identifying the clusters with different ids, lead to the enumeration of two different solutions. A straightforward method to avoid parallel solutions is to attribute contigs to clusters depending on the rank of the contig identifier. Therefore, contig 1 will be clustered in cluster 1, then the first contig to not be clustered with contig 1 will be put in cluster 2, etc. Because the identifier of the first contig to put in cluster 1 will never change (it will *always* be contig 1), this specific case can even be written as a fact, to help to further fasten the generation of answer sets.
During the first attempts, the maximum TNF distance and the minimum abundance distance were computed with the `#max` and `#min` statements from ASP. Discussions with Flavio Everardo, a post-doctoral researcher from Schaub's group at Potsdam University (which is the main group involved ine the development of the POTASSCO software) have lead to a new formulation of the maximum and minimum formula.
Statements `#count` and `#sum` in ASP may become relatively problematic, because of the number of predicates generated. The first implementation of the ASP model, using these statements to enumerate predicates, has then a high computation time. The step to fasten in this case was the grounding, even though, because of the very high number of ground models generated, enhancing the implementation would also reduce the resolution time. Based on these remarks, the implementation of the avoidance of parallel solutions relied on the following lines:

```
aux_att(B,C) :- att(B,C).
aux_att(B,C-1) :- aux_att(B,C), C>0.
attmax(B,C) :- aux_att(B,C), not aux_att(B,C+1).
attmin(B,C) :- att(B,C-1), contig(C,_).
attmin(B,C+1) :- attmin(B,C), attmax(B,Cmax), C<Cmax.
attmin(B,C) :- attmin(B,C), not attmin(B,C-1).
:- attmin(B1,C1), attmin(B2,C2), B1<B2, C1>C2.
```

In the first three-lines block, rules looks for the maximum contig id present in a bin B. This maximum contig id is captured in the predicate `attmax/2`. This value $Cmax$ is then used as a limit to search for the minimum contig identifier present in B: at line 5 the contig identifier is captured in `attmin/2` predicate if, and only if, it is strictly less than

$Cmax$. If any high enough value (*i.e.*, any value greater than or equal to the number of contigs in the dataset) may be chosen instead of $Cmax$, $Cmax$ is ensured to be the lowest possible value to list all contigs identifier from a given bin. Without this limit, the line 6 would result in an infinite loop during the grounding, forbidding the resolution to continue further. The same remark applies to line 2, with the difference that a lower threshold value for contigs identifier is easier to set ($C > 0$). With this implementation, the research for the minimum contig identifier in each bin can be performed in linear time, depending only on the number of contigs in the dataset, as it only needs to list the contigs present in each bin. The last line represents the anti-parallel solution constraint itself, forbidding that the minimal contig identifier $C1$ from the bin $B1$ is greater than the minimum contig identifier $C2$ from bin $B2$, if $B1$ was less than to $B2$.
Rules determining the computation of diameter and margins were also modified. The implementation for the computation of the diameter then became:

```
diam(B,D) :- att(B,C1), att(B,C2), C1<C2, bin(B), tnf(C1,C2,D).
diam(B,D-1) :- diam(B,D), D>0.
maxDiameter(B,D) :- diam(B,D), not diam(B,D+1).
```

With the first line generating a predicate `diam/2`, extracting all distances D between pairs of contigs $C1$ and $C2$ belonging to the same bin B. The second line then processes to find the maximum value among all possible values of D in `diam/2`, and then stores that value D in the predicate `maxDiameter/2`. Besides including the enhanced search for the minimum value, these lines also have the advantage to not relying on an intermediate predicate like `distancetnf` in the previous implementation. The implementation for the margin is done in a similar fashion:

```
margin(M) :- att(B1,C1), att(B2,C2), bin(B1), bin(B2), B1<B2,
abd(C1,C2,M).
margin(M+1) :- margin(M), M<100.
minMargin(M) :- margin(M), not margin(M-1), M>-1.
```

With the first line generating the predicate `margin/3`, which extract all distances M between two contigs $C1$ and $C2$ belonging to two different bins $B1$ and $B2$, respectively. The second line then computes the search for the minimal values amongst all possible values of M, and the third line stores this minimum M in `minMargin/3`, once found. Similarly as the new implementation of the diameter, removing of intermediate predicate `distanceabd/3` allowes to reduce the number of predicates to produce to generate the grounding program, further fastening the grounding. A particularity with the two imple-

mentations is that the upper value of the abundance distance, and the lower value of the TNF distances, are already known. Thus, contrary to the search for the minimum (or maximum) contig identifier present in a bin, it is unnecessary to perform an intermediate search for these values. Thanks to these modifications in the implementation, our ASP model was able to perform the clustering of a toy dataset composed of a hundred contigs in a couple of seconds, and with a toy dataset of a thousand contigs in twenty minutes, far below the preliminary calibration tests.

3.6.4 Implementation of new constraints in ASP

The model was further enhanced with the implementation of SCGs-related constraints. At first, the SCGs have been introduced as a cannot-link constraint: two contigs must not be put in the same bin if they share a copy of the same SCG. One implementation in ASP is:

```
:- scg(C1, G) ; scg(C2, G) ; att(B, C1) ; att(B, C2).
```

At this stage, all the predicates present in the constraint had already been generated, so this line would not have any impact on the grounding. The line just stated that all models in which two contigs $C1$ and $C2$, binned in the same bin B and carrying a SCG with the same identifier G, would be unsatisfiable. This simple constraint was however highly stringent, and the standard characterisation of MAGs should allow some tolerance to the presence or absence of SCG in a putative MAG (see section 3.5.3 "Constraints" p.76). The implementation in the ASP program of the completeness constraint as a cluster-level constraint was:

```
#const threshold_completeness=5.
scg_bin(C,G) :- att(B,C), scg(C,G).
:- bin(B), not #countG : scg_bin(B,G)>threshold_completeness.
```

The second line is a rule, with the purpose to enumerate all the SCGs identifiers G belonging to the same cluster C. The third line is the constraint itself, and it forbids a cluster to not reach the `threshold_completeness`, which was fixed beforehands. The contamination constraint as a cluster-level constraint was implemented as follows:

```
#const threshold_contamination=1.
scg_bin_duplicate(B,G1) :- att(B,C1), att(B,C2), scg(C1,G1), scg(C2,G1),
C1<C2.
:- bin(B), #countG : scg_bin_duplicate(B,G)>threshold_contamination.
```

The second line is a rule that enumerates all the duplicated copies G_{ij} of the same SCG G_i, for all SCGs G_i carried by any contig j within the cluster C. The third line represents the constraint itself, and forbids a cluster to reach a level of contamination strictly superior to the threshold. In this case, a bin then must not contain more than 1 copy of any SCG.
As with the anti-parallel constraint (see p.82), the statement `#count` does not affect the computation time, because of the presence of the inequality sign.

3.7 Comparison with metabat2

Once both grounding and resolution times reached acceptable levels for medium-size toy datasets, our enhanced ASP model was compared to the results of metabat2. The selected dataset to perform this comparison was composed of 1183 contigs, belonging to 10 genomes from the CAMI [175] high complexity dataset. The CAMI dataset is a simulated metagenomic dataset, which has been used as a standard to perform comparison between MAGs reconstruction tools [146, 175]. The source genomes were selected in order to obtain all the 10 mOTUs marker genes in each expected genome.
To perform the comparison between metabat2 and our ASP model, we computed the recall, the precision and F1-Score of each approach. The recall is defined as:

$$Recall = \frac{number\ of\ correctly\ binned\ contigs}{number\ of\ contigs} \tag{3.9}$$

The precision is computed as:

$$Precision = \frac{number\ of\ correctly\ binned\ contigs}{number\ of\ binned\ contigs} \tag{3.10}$$

And the F1-Score is defined as:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.11}$$

Even though its computing performances had been dramatically fastened with the enhancement of its implementation, our ASP model still could not compete with the swiftness of metabat2. Indeed, our ASP model took more than 15 minutes to output an optimal solution, while the latter performed binning in a mere couple of seconds.

The precision of the binning was also in favor of metabat2. However, metabat2 found only

Bins	Ru	Co	Ba	Cl	Pa	Rh	Sh	SK340	SK678	Sy	Total bin
bin1	0	0	0	0	196	0	0	0	0	0	196
bin2	0	0	187	0	0	0	0	0	0	0	187
bin3	0	0	0	0	0	0	74	0	0	0	74
bin4	6	0	0	58	0	0	0	0	1	0	65
bin5	0	25	0	0	0	0	0	0	0	0	25
bin6	56	0	1	0	1	0	0	0	0	0	58
bin7	0	0	0	0	0	0	0	135	94	0	229
bin8	0	0	0	0	0	108	0	0	0	0	108
bin9	0	0	0	0	0	0	0	0	0	159	159
unbinned	11	8	18	0	11	3	15	1	3	11	81
Total genome	73	33	206	58	208	111	89	136	98	170	1182

Table 3.2 Metabat2 binning results.
Number of contigs binned per original genome. Lines: Metabat2 output bins, columns : Genomes of origin. Metabat2 only retrieved 9 bins out of 10 original genomes, as it merged SK340 and SK678, two closely related strains, into a single bin. Metabat2 also failed to bin 81 contigs, generating a trash "unbinned" bin. Recall: 0.84 ; Precision: 0.91 ; F1-Score: 0.84.

9 genomes out of the 10 expected genomes, because of the presence of two strains (SK340 and SK678) related to the same species in the dataset, which metabat2 was unable to discriminate (Table 3.2). Another major difference between the two approaches was that metabat2 was not able to cluster all the contigs, leaving 81 contigs put in an "unbinned" bin. Because our ASP model forces the clustering of all the contigs, and because the number of clusters on which to perform binning was fixed beforehands, it did not have these two disadvantages. This result was however more than just an artifact due to the configuration of the model, as the clustering of the contigs belonging to these two genomes was performed accurately, showing that the ASP model was able to discriminate the contigs belonging to these two strains. Considering this point, the overall binning performances of metabat2 were superior to the ASP model, exhibiting higher recall, precision and F1 Score than ASP binning. The ASP binning resulted in more mixing of contigs belonging to several expected genomes, and more scattering of contigs from the same genomes in

different bins (Table 3.3) than the binning performed by metabat2.

Bins	Ru	Co	Ba	Cl	Pa	Rh	Sh	SK340	SK678	Sy	Total bin
bin1	0	27	0	0	0	0	0	0	0	0	27
bin2	0	7	0	36	0	0	0	0	0	0	43
bin3	9	0	0	22	0	0	0	0	0	0	31
bin4	38	0	0	0	0	0	0	0	0	0	38
bin5	26	0	0	0	0	0	89	0	1	0	116
bin6	0	0	0	0	0	0	0	0	95	0	95
bin7	0	0	0	0	0	110	0	0	2	0	112
bin8	0	0	0	0	0	1	0	136	0	26	163
bin9	0	0	0	0	0	0	0	0	0	77	77
bin10	0	0	206	0	208	0	0	0	0	67	481
Total genome	73	34	206	58	208	111	89	136	98	170	1183

Table 3.3 ASP model binning results.
Number of contigs binned per genome of origin. Lines: ASP output bins, columns : Original genomes. The model was forced to cluster all the contigs, *i.e.* it did not have the capacity to discard any contig. The model was also given the number of clusters beforehands. Even with these advantages, the binning performances were inferior to the performances of metabat2. Recall: 0.71 ; Precision: 0.71; F1-Score: 0.71.

3.7.1 Estimating the number of bins beforehand

Instead of giving the exact number of contigs on which to perform binning to the ASP model, a different approach was tested. In this new approach, the number of expected genomes was not fixed to 10, but estimated using SCGs. We thus counted the number of contigs carrying each of the 10 SCGs in the dataset. An estimator of the possible number of bins was the sum of the number of contigs carrying one of the 5 lowest-abundant SCG within the dataset. This estimator, which was not precise, offered the advantage to compute the theoretically maximum number of bins one could expect to obtain considering the SCGs features. Following this approach, the number of clusters was estimated to be 50. These clusters were integrated in the model as putative bins, on which cluster constraints were not applied. These putative clusters were then considered as true bins if and only if contigs were assigned to them, and these true bins had to satisfy all the bins constraints. Thus, this new framework needed the addition of a new predicate `putative bin/1` and a

new rule to obtain `bin/1` predicates, which increased the complexity of both the grounding and the resolution steps:

```
#const k=50.
putative_bin(1..k).
1{att(B,C) : putative_bin(B)}1 :- contig(C,_).
bin(B) :- att(B,_).
```

The first three lines are just the same form than the previous assignment rule. But the fourth line, then, generated the predicate `bin/1` if a putative bin had at least one contig attributed to it, *i.e.*, a putative bin B was considered a true bin if there was at least one predicate `att/2` containing B. The complexity of the model thus grew sufficiently enough to fail the resolution, because of a too high number of grounded models.
In order to assess the effect of the number of bins on the resolution time, we used a reduced dataset composed of 350 contigs, to be clustered into 25 bins. The whole execution time of this framework was 4h, reduced to 25 minutes thanks to clasp multi-threading.

3.7.2 Unbinned contigs from metabat2

To reduce the number of contigs to cluster, an approach would be to use the logic programming framework as a post-processing step to enhance results obtained from an established binner. We thus started from the clustering results of metabat2, with the objective to cluster only the unbinned contigs. Because of the reduced size of the dataset, these unbinned contigs represent a minor proportion of the contigs set.
Even after diminishing the number of contigs to cluster with ASP, it still needed to process information from all existing contigs, notably the already-clustered contigs. As such, the grounding time remained barely affected by this change of methodology. Our ASP model integrated the whole binning result of metabat2 beforehand, and each metabat2 bin was assigned to an original genome, based on the maximum number of contigs belonging to that genome. For the contigs belonging to the two strains SK340 and SK678, which were merged into a single bin by metabat2, the bin was identified as SK. The results show that the model did not perform binning with high precision, as recall, precision and F1 score only reached 0.13. The binning thus resulted in a high number of ill-assigned contigs (Table 3.4). Because of the lower number of points to classify, the resolution time significantly plummeted. In order to look for contigs which would be assigned to different clusters, several binning solutions were explored. We identified the unbinned

contigs belonging to genomes Sh, SK678 and Sy to be assigned to different bins through several solutions (Table 3.4). A main limitation of the ASP model was that the unbinned contigs had a high TNF distance with all the other contigs, with no means to discriminate between contigs belonging to the same genome or to a different genome.

Bins	Ru	Co	Ba	Cl	Pa	Rh	Sh	SK340	SK678	Sy
Bin Ru	0	0	0	0	1	0	0	0	0	0
Bin Co	0	1	0	0	1	0	0	0	0	0
Bin Ba	0	0	0	0	1	0	[1,12,6.5,12]	0	[0,1,0.5,1]	0
Bin Cl	0	0	0	0	1	0	0	0	0	0
Bin Pa	0	0	18	0	3	0	[0,10,5,10]	0	0	[0,2,1,1]
Bin Rh	0	0	0	0	1	0	0	0	0	0
Bin Sh	11	7	0	0	1	3	[3,10,6.5,6]	1	[2,3,2.5,1]	[9,11,10,1]
Bin SK	0	0	0	0	1	0	0	0	0	0
Bin Sy	0	0	0	0	1	0	0	0	0	0
Total genome	11	7	18	0	11	3	15	1	3	11

Table 3.4 ASP model on metabat2 unbinned contigs.
Number of contigs that were left unbinned by metabat2, per expected genome. Lines: Metabat2 bins, assigned to expected genomes, with the expected genome assigned being the genome from which the maximum number of contigs in the metabat2 bin belong to; columns: Expected genomes. The results are summarized from several obtained solutions, with some contigs being assigned to different bins through several solutions. The different contigs assignments are shown in brackets, with minimum number of contigs, maximum number of contigs, average number of contigs, and standard deviation, respectively. Recall: 0.13; Precision: 0.13; F1-Score: 0.13. Recall and Precision shared the same value because the model was forced to bin all the contigs.

3.8 Discussion

The work presented in this chapter is a first attempt to formalize the binning process as a constraint logic programming problem. The purpose was the possible exploration of several solutions of the problem, in order to adress limits of previous binning approaches. Previous studies have already applied constraints to guide the binning of contigs [156] and have formalized constrained clustering as a CP problem [230]. Thus, we knew this framework could possibly be applied to the problem of binning of contigs into MAGs. We chose more specifically ASP to implement this binning model, for its possible application

to large combinatorial problems.
The application of the binning problem into ASP was not simple. First, problematic implementation heavily hindered the resolution of binning, even with very small toy datasets. Discussion with members of Torsten Schaub's group at Potsdam University, in charge of the development of POTASSCO software, helped to deepen our understanding of the language statements, dramatically fastening the problem resolution. Despite these improvements, its performance, both in terms of computation time and accuracy, remained far below the performances exhibited by a binning tool from the literature, metabat2.
The ASP binning model was applied to a small dataset, composed of contigs belonging to known original genomes, subsampled from a standard simulated dataset. We then compared the results of the ASP model to metabat2 [16], which is currently the binning tool providing the best computation performance, and the best binning efficiency. The comparison was much in favor of metabat2, notably in terms of computation time. However, computation time was not considered as a major point, at least as long as the problem can be solved in a humanly, feasible time; besides, the objectives of this work were to have a complete resolution, and to obtain several solutions to explore. However, using our current models, even the efficiency of the binning processed by the ASP model was below the efficiency of metabat2 binning (Tables 3.2, 3.3). An interesting point however was the inability of metabat2 to detect two genomes belonging to two closely related strains, this drawback being absent from the ASP binning results. Metabat2 merged the contigs belonging to the two genomes SK340 and SK678 into one unique genome, because of the small TNF distance existing between contigs of these two genomes. If the ASP model succeeded to discriminate between contigs belonging to one of these two genomes, it should be emphasized that the ASP model preliminarily knew the number of bins. It was also forced to bin all the contigs, and to not leave a bin empty. Even after acknowledging this bias, binning results showed that these two genomes were almost perfectly recovered into bins (Table 3.3). Contigs from the genome SK678 were the most perfectly binned contigs by the ASP model, with only 3 contigs dispatched in other bins, and with the bin containing the majority of the contigs not being contaminated by alien contigs. This may represent a clue of a strong discriminating effect of the use of SCGs constraints during the binning, as metabat2 does not display any SCG feature in its binning algorithm.
Removing the initial information of the number of bins to the ASP model had been interesting. In this new framework, the ASP model was instructed about a number of potential bins. These putative bins would not have to be filled, like in the previous formalization,

but instead, the model had to try to bin the contigs into a certain amount of bins, and then removed the unused putative bins. Because of the estimation of the maximum of putative bins to consider was too high, reaching 50, the resolution was not possible. In this case, the resolution generated a too high number of ground models to explore, reaching the physical upper limit of the number of combinations that clasp could possibly handle. Thus, even increasing the number of threads to perform resolution would not help. In fact, it allowed us to observe the strong effect of the number of bins on the resolution computation time. Thus, with a dataset composed of only 1183 contigs, even testing 25 putative bins resulted in a resolution failure, again because of a too high number of models to search. Attempts based on a smaller dataset composed of only 300 contigs and 25 putative bins successfully reach an optimal solution, but at the cost of a long resolution time. The model was also unable to keep only a number of bins equal to the number of original genomes, and kept all the 25 putative bins in the stable models output.
Application of the model as a post-processing step, in order to bin only the contigs left unbinned by metabat2, output imprecise results (Table 3.4). The main explanation for this decline in performance was the high TNF distances between each of the unbinned contigs with all the other contigs, even those belonging to the same expected genome. Another explanation may be the softening of the constraints of the model, in order to process with such reduced dataset, limiting their effectiveness.

3.9 Conclusion & Perspectives

There are many perspectives to this work. First, in order to reduce the grounding time, which would result in resolution time reduction, the input data might be filtered, before clingo processes them. A possible approach would be to preliminary remove some contigs from the dataset. This elimination would be based on a distance threshold, and would replace a constraint of the model. However, such a method might be dubious, as a distance constraint may help to find inconsistent models better. Besides, the application of the ASP model to the unbinned contigs could already be consider as a data filtering, as the binning had to be performed on only 8% of the contigs. Finally, the number of bins represented a harder barrier for the resolution that the number of contigs.
The addition of more constraints would definitely help the resolution step. A main limit of the ASP model was the small number of constraints, with only the SCG constraints and the bins size constraints. However, the addition of a constraint unrelated to composition

or abundance distance would result in the addition of more input data, resulting in an increased parsing time.
An interesting development would be to use the Constraint ASP (CASP) [19] paradigm in place of ASP. CASP, as it name suggests, is an adaptation to a CP resolution problem implemented in ASP: it would then follow a resolution step closer to classic CP resolution. CASP would notably allow to eliminate the grounding step [18], which would allow to gain a sensible computation time for a problem resolution.
While there are still paths to explore related to this work, a major drawback remains that the clustering of high amount of data represents a hard combinatorial problem, which would still exceed the scaling capacities of ASP. Besides, clustering literature in ASP seems nonexistent. We nevertheless hope that this introducing formalization of the genome binning problem into a logic programming framework would serve as a peliminary step for future development.

Chapter 4

MAGNETO: AN AUTOMATED WORKFLOW FOR GENOME-RESOLVED METAGENOMICS

Preamble

In this chapter, we present MAGNETO, a MAG reconstruction workflow which includes an unsupervised co-assembly module. First, we will present the context that led to the development of the pipeline. Secondly, we will compare the performance of our co-assembly module, which computes distance between metagenomes using their nucleotidic composition, against an approach relying on geographical distance. Last, we will compare four different assembly-binning strategies to reconstruct MAGs. This work was the subject of an article [31]

.

4.1 Context

Genomes are a valuable resource for characterizing and understanding the diversity, ecology and evolution of microbial organisms in the laboratory as well as in natural environments. As culture-based approaches have been historically used to recover genomes and enrich reference databases, current knowledge from most reference bacterial genomes come from axenic cultures. However, despite the improvement of culture-based approaches to cultivate novel microorganisms, the number of organisms that can be isolated and cultivated remain mainly constrained by specific growth conditions. Depending on the

considered environment, it is estimated that a proportion of only 0.1 to 1% of all microbial genomes could be cultivated [233, 234].

The rise of metagenomic studies, thanks to the rapid development of high-throughput shotgun sequencing, has allowed direct access to the diversity and functional potential of naturally occurring microorganisms, bypassing the cultivation bottleneck. For more than a decade, various studies have reconstructed genomes from metagenomes and contributed to describe thousands of novel microbial clades belonging to diverse environments, such as in the human gut [15], in soils and in aquatic environments [12, 13].

The reconstruction of these draft genomes, commonly called Metagenome-Assembled Genomes (MAGs), has now become a common approach, with numerous software developed during the last decade [16, 24, 109, 170]. As for the reconstruction of genomes from single organisms, MAG reconstruction can be split into two main steps: first, the *assembly* of the reads obtained from the sequencing into longer sequences called *contigs*; second, the *binning* of these contigs into MAGs, mainly using their compositional and/or abundance similarities. However, MAGs reconstruction can face several limitations including gaps, sequencing errors, local assembly errors, contigs chimeras and bin contamination, *i.e.* the inclusion of contigs belonging to different genomes in the same bin. The binning of contigs may also miss genomic regions in which nucleotidic composition significantly differs from the average genomic composition, such as ribosomic RNA regions, or mobile elements [235]. These limitations can be partially addressed by several quality checkpoints, misassemblies detection, and manual curation [182].

In addition, low abundance organisms are usually harder to recover, due to limited reads information during the assembly process [137]. When shallower sequencing is performed (*i.e.* when the number of bases output by the sequencer), reads from low-abundant genomes will be rare, and thus their assembly into contigs will be more difficult, as assemblers tend to consider these reads as erroneous and discard them. A common approach to increase the abundance of rare reads is to adapt the assembly strategy, that is not assembling a unique metagenomic sample (single-assembly), but *co-assembling* several samples together. Co-assembly will then tend to increase the number of occurrences of rare reads, and consequently incorporate them into resulting contigs, thereby capturing a higher fraction of the diversity within the samples. Co-assembly strategies have been instrumental for recovering higher numbers of MAGs [5, 110]; however, this approach increases the probability to generate fragmented assemblies [137, 175].

Sequencing effort & filtering effect

The volume of sequenced products output by a sequencer depends on the initial supplied quantity of nucleotidic products to perform the sequencing. Coupled with the aimed sequencing coverage, *i.e.* the number of times the same nucleotide is sequenced, these both reduce the fraction of the actual genomic material of the microbial community to be present in the sequencing products. Both these parameters may impact the downstream analysis, impacting notably individual genome reconstruction[47, 48] and gene detection [236].

The genome *binning* process consists in classifying contigs usually based on similarities of their sequence composition, their abundance, or their taxonomic affiliation. In most existing software, binning is performed using two main metrics, namely sequence composition [109] and contigs abundance [24]. Sequence composition is defined as the frequencies of all tetranucleotides within the contig sequence, called TNF (for TetraNucleotide Frequency). The abundance (or co-abundance) represents the mean vertical coverage of the contig in one (or several) sample(s) (see also: Chapter2, section 2.2.2 "Unsupervised binning", p.45). Other metrics, such as taxonomic affiliation of the contigs, may also be used to determine which contigs belong to the same bin [156]. The principal differences between existing binning softwares usually rely on the algorithm used to group contigs into genome bins. Most successful software have used density-based clustering [171], Gaussian mixture models [24], affinity propagation [237], or graph clustering [16]. Other methods can also perform binning on genes rather than contigs, relying on the presence of co-abundant genes within metagenomes, such as canopy clustering [145] and MSPminer [29]. The objects reconstructed by these methods may not be qualified as MAGs, and are commonly referred as MetaGenomic Species (MGS) and Co-Abundance gene Groups (CAGs).

Extracting knowledge from raw metagenomics data requires to handle several specific tasks, from assembly to gene calling and annotation, each of them often performed using dedicated software. Users commonly face choosing, configuring, and running different tools, which can be challenging and time-consuming. Recently, several metagenomics workflows have been developed [21, 22, 26, 238], often using specific default parameters for each integrated software. However, these workflows usually suffer from limits towards either the assembly step or the genome binning step. In workflows allowing to perform

co-assembly, sets of samples to co-assemble have to be determined and manually specified by the user, implying some *a priori* knowledge about the microbial ecosystem under study. Besides, only a single workflow [238] allows the user to compute co-abundances from metagenomes that have not been included in the assembly. As computing co-abundance profiles of contigs from multiple metagenomes may increase the precision of the metric [16, 239], the impossibility to compute large-scale co-abundance may be considered a limitation of these workflows.

In this chapter, we present MAGNETO, a fully automated workflow for genome-resolved metagenomics, implementing a co-assembly module that integrates a non-supervised method to define sets of samples to co-assemble without *a priori* knowledge. It also includes complementary strategies to compute abundance metrics from one to n metagenomes, even if they do not participate in the assembly process. In this study, we tested our co-assembly module on a set of marine metagenomes, against a co-assembly relying on existing knowledge. We also benchmarked four different assembly-binning strategies for MAGs reconstruction, on diverse datasets ranging in complexity, from a mock dataset representing a small bacterial community to human gut microbiome communities.

4.2 Design and implementation

MAGNETO is a Snakemake [240] workflow connecting open-source bioinformatics software, all available from Bioconda and conda-forge. Snakemake was chosen for its flexibility, its capacity to run both locally and on clusters, and its Conda management automating software installation. MAGNETO includes several tools designed for metagenomic studies. First, reads trimming is performed using fastp [241] and FastQ Screen [242]. The co-assembly module relies on Simka [243], which estimates metagenomic distances between samples based on their k-mers composition. MEGAHIT [125] then performs reads assembly/co-assembly. We use MetaBAT2 [16] to bin contigs, and we assess the quality of bins using CheckM [180]. The de-replication of bins into MAGs (bins of at least high or medium quality) is performed using dRep [193]. Notably, MAGNETO can also be used to establish gene catalogs, to better capture metagenomic gene diversity by producing a non-redundant set of genes through sequence clustering at a user-defined sequence identity cutoff (e.g. 95%) using Linclust [244]. GTDB-tk [245] is used to perform taxonomic annotation of dereplicated MAGs, and eggNOG-mapper [246] is used to perform the functional annotation of MAGs as well as the gene catalog (see Figure 4.1).

The four binning strategies are directly configurable by the user, and a quick configuration allows to perform from a single to all strategies for reconstructing MAGs. Notably, MAGNETO is currently the unique workflow providing an automated approach to define clusters of metagenomes for co-assembly. Importantly, MAGNETO and nf-core/mag are also the only workflows allowing users to perform a co-binning strategy. A synthetic comparison of functionalities provided by the workflows tested in this study is available in Table 4.1.

Table 4.1 Comparison of tasks performed by evaluated workflows.

Steps	ATLAS	METAWRAP	nf-core/mag	MAGNETO
Pre-processing				
Reads Trimming	✓	✓	✓	✓
Contamination	✓	✓	✓	✓
Assembly				
Co-assembly availability		✓	✓	✓
Compute sets to co-assemble				✓
Binning				
Co-binning availability		✓	✓	✓
Multiple binning software	✓	✓		
Bin refinement	✓	✓		
Bin reassembly	✓	✓		
Post-processing				
MAGs quality check	✓	✓	✓	✓
de-replication step	✓	✓	✓	✓
Genome annotation	✓	✓	✓	✓
Gene catalogue			✓	✓
Reproducibility				
Workflow management	✓		✓	✓
Packages management	✓		✓	✓

4.2.1 Reads pre-processing

Raw reads were filtered using fastp [241] and FastQ Screen [242]. fastp filters reads on their quality, length and complexity. FastQ Screen is a tool allowing to control contamination within metagenomic samples, by mapping their reads to reference genomes. These two tools provide results reports to the user that are useful to evaluate reads quality.

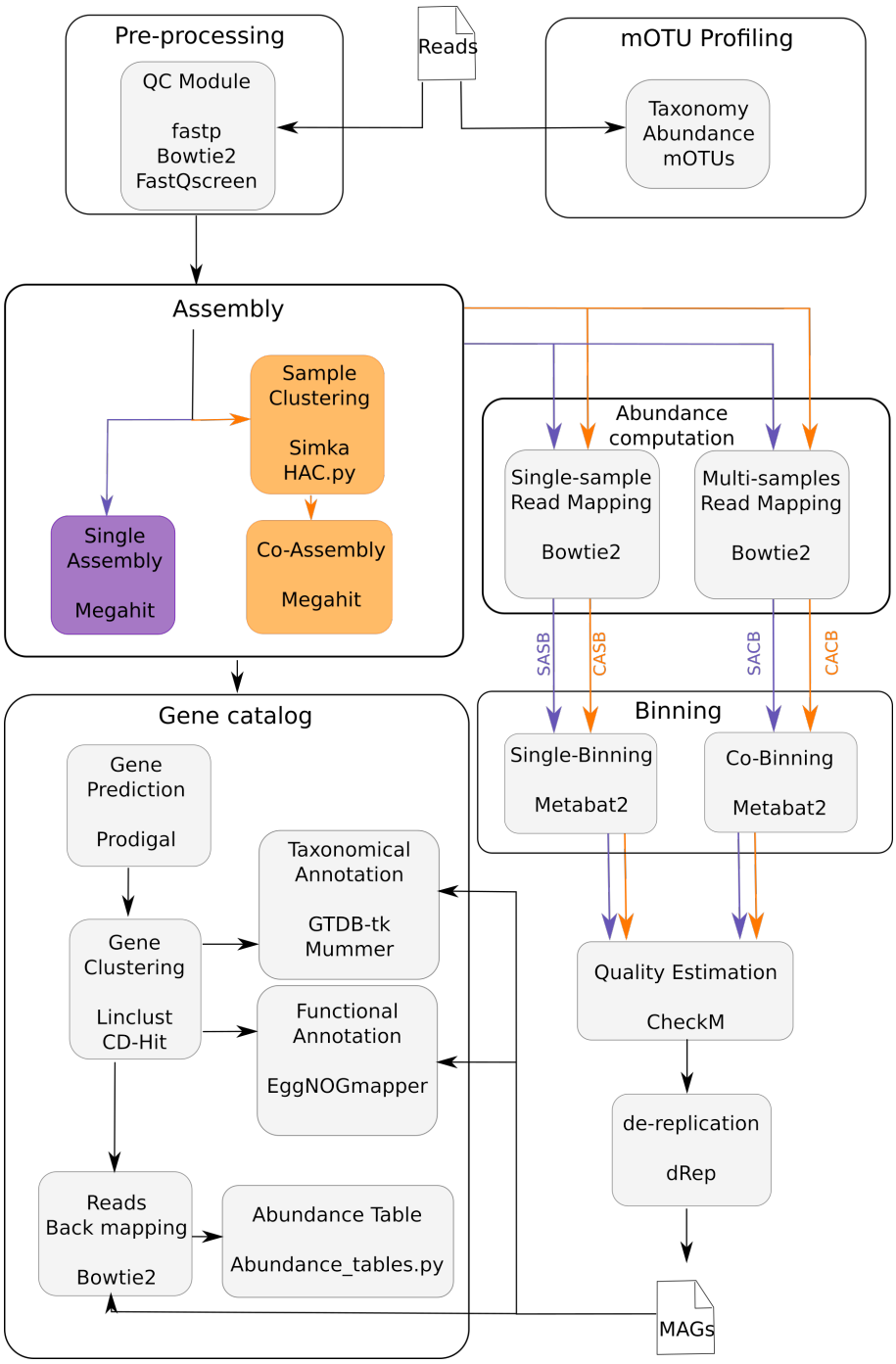

Figure 4.1 **Overview of the MAGNETO workflow.**
Summary view of modules implemented in the MAGNETO workflow, with the name of the software or script associated to each task. The workflow can be launched for a complete run, to process raw reads into a gene catalog and MAGs, but each module can also be run independently. In purple: path to perform single-assembly, corresponding to SASB (Single-Assembly-Single-Binning) and SACB (Single-Assembly-Co-Binning) strategies, and orange: path to perform co-assembly, corresponding to CASB (Co-Assembly-Single-Binning) and CACB (Co-Assembly-Co-Binning) strategies.

4.2.2 Assembly

We performed reads assembly using MEGAHIT [125], as this assembler provides an excellent trade-off between computational requirements and assembly quality [157]. metaSPAdes [126] could have been considered as it provides better performances than MEGAHIT in terms of overall percentage of the metagenome recruited in the assembly [247] and maximum length of scaffolds produced [247, 248]. However, this performance increase occurs at the cost of a greater consumption of computational resources [157] and a presence of a greater proportion of misassembled sequences in contigs than MEGAHIT [247]. More importantly, metaSPAdes was originally not designed to perform co-assembly, which constitutes a major drawback in our workflow. Moreover, MEGAHIT is able to capture micro-diversity from the metagenomes more efficiently than metaSPAdes, as it discards less low-abundant reads during assembly [157]. Co-assemblies of the marine metagenomes were performed using the *–presets meta-large* option, as these metagenomes revealed to be highly complex. All other assemblies were performed using the *–preset meta-sensitive* option.

4.2.3 Co-assembly strategy

In order to determine which samples to co-assemble, we used Simka, a de novo and scalable tool for comparative metagenomics [243]. Simka computes different distances based on k-mer counts, instead of species counts. In our case, we used their modified Jaccard (or AB-Jaccard) distance rather than the default Bray-Curtis distance, as the latter does not satisfy triangle inequality. Once the distance matrix from Simka was computed, samples were then clustered using a Ward-based hierarchical agglomerative clustering [249]. Then, we iteratively cut the dendrogram and assessed partitioning quality using the Silhouette method [250].

4.2.4 Genome binning strategies

Binning was performed using MetaBAT2 [16], as it is currently one of the fastest and best performing genome binners. We set the minimum length for contigs to be binned to 1500 nucleotides. As MetaBAT2 uses composition and abundance to perform binning, a preliminary step to map reads back to assembled contigs was performed to measure abundance. Reads mapping was achieved using Bowtie2 [251].

Instead of computing an abundance metric only from the metagenome assembled into contigs, MetaBAT2 may compute a co-abundance metric using contig coverage from several samples, even if these samples do not participate in the assembly. A co-abundance metric computed from several samples increases the quality of the genome bins produced [141]. Depending on the number of samples used to compute contigs abundance, the corresponding metric is either an abundance or a co-abundance metric. Thus, two strategies can be pursued in order to perform binning: (i) single-binning, which uses abundance of contigs measured from assembled metagenome(s); or (ii) co-binning, which uses co-abundance of contigs measured from all the metagenomes of a dataset. Combined with the decision of performing either single-assembly or co-assembly, we defined four binning strategies: Single Assembly of one metagenome with Single-Binning (SASB), Single Assembly of one metagenome with Co-Binning (SACB), Co-Assembly of one set of metagenomes with Single-Binning (CASB), and Co-Assembly of one set of metagenomes with Co-Binning (CACB).

4.2.5 Genome bins quality

Genome bins quality was defined by two metrics, namely completeness and contamination. Completeness measures the fraction of the initial genome captured, while completeness measures the fraction of alien sequences; both rely on the presence-absence patterns of universal Single-Copy marker Genes (SCGs). To assess genome bins quality, we used CheckM [180] and GUNC [14]. Based on contamination and completeness, we distinguished three standard quality levels for bins [179] : (i) high-quality bins (HQ) with completeness $> 90\%$ and contamination $< 5\%$, (ii) medium-quality bins with completeness $> 50\%$ and contamination $< 10\%$, while the (iii) low-quality bins (LQ) are bins that are neither HQ nor MQ. Only HQ and MQ bins were then considered to be MAGs. The comparisons of reconstructed MAGs quality from different strategies were performed using Mann-Withney U test using R [252]. As a MAG may be reconstructed independently either in two (or more) samples or two (or more) co-samples, MAGs are also de-replicated using dRep [193]. Two MAGs were considered to be duplicated if their pairwise ANI (Average Nucleotide Identity) score was above a given identity threshold t, (t being a percentage of sequence identity) on more than 60% of their bases [191]. We consider two different values for t: $t = 0.95$, which corresponds to a de-replication at species level, and $t = 0.99$, which corresponds to a de-replication at strain level [85].

4.2.6 Genome annotation module

Functional and taxonomic annotations were performed for the strain-level MAGs collection, which encompasses the species-level collection. To perform functional annotation of MAGs, we used eggNOG-mapper [246], and we used GTDB-tk [245] to perform taxonomic annotation. Finally, reads of each sample are mapped back onto both species- and strain-level MAGs collections using Bowtie2, and an abundance table is produced using an in-house Python script.

4.2.7 Gene annotation module

Coding DNA Sequences (CDS) are detected on assembled contigs per-sample (single-assembly) using Prodigal [253]. Genes from all samples are clustered at 95% identity using Linclust [244] in order to produce a non-redundant set of genes (gene95 collection). EggNOG [246] and MMSEQ2 [254] are used to annotate this gene collection, for functional and taxonomic information, respectively. Finally, reads from each sample are mapped back onto the gene95 collection using Bowtie2, and an abundance table is produced.

4.2.8 Future enhancements: adding modules

Thanks to the Snakemake implementation of MAGNETO, the addition of supplementary modules can be very straightforward. Future interesting features to deepen the metagenome analysis could be the inclusion of a module to measure the optimal maximum growth rate of reconstructed MAGs. The estimation of the optimal maximum growth rate temperature can give insight about the genome composition of a bacteria, as faster growth in environments with higher resource availability has been related to a greater genome size, and a higher number of ribosomal gene copies [255]. This module has already been developed, and used to assess optimal growth temperature of reconstructed MAGs from the Arctic ocean ; this work was our contribution to [25].
Estimations of minimum generation time and optimal growth temperature are performed using Growthpred2 [256]. Growthpred relies on codon usage biases in highly expressed genes identified in genomes. The identification of these highly expressed genes needed first, the extraction of the coding sequence of each MAG, using GffRead [257]. The highly expressed genes are then retrieved from the coding sequence using BLAST [258], and only the MAGs exhibiting at least 10 highly expressed genes are kept for optimal growth assessment. Growthpred v.1.08 was used and a Snakemake pipeline is available

at https://gitlab.univ-nantes.fr/combi-ls2n/growthsnake. Other enhancements will be the integration of other assembly software, and the addition of other binning tools, with an aim to incorporate cross-tools approaches in order to further improve the quality of binning, such as in ATLAS [22] or METAWRAP [26].

4.3 Determining co-assemblies using metagenomic distances

In [5], the authors studied the abundance of diazotrophic bacteria in oceanic surface metagenomes and showed that nitrogen fixation is an important feature of the prokaryotic communities living in ocean surface. As microbial genetic distances often co-vary with geographic distances in several habitats [259], co-assemblies were performed based on the geographic coordinates of the metagenomes, i.e. metagenomes belonging to the same oceanic region were co-assembled. In the euphotic zone, microbial community similarities between metagenomic samples from the same oceanic regions have been observed to be higher than similarity between metagenomes from distinct oceanic regions, although a separation by regional origin is unclear [8] as other environmental factors (e.g. ocean currents) can modulate genetic proximity between populations [74]. As a consequence, two geographically close metagenomes do not necessarily share the highest proportion of genomes, and two metagenomes belonging to the same ocean region may not be closer to metagenomes from other regions.

Given that the main goal of co-assembly is to increase the proportion of reads belonging to a given strain or species, we propose to identify sets of samples to co-assemble using metagenomic distances. To the best of our knowledge, very few studies have used sequence-based compositional distances to guide metagenomic co-assembly. Historically, metagenomic compositional distances have mainly been used to compare metagenomic samples [260] or MAGs [110], but not to actually guide the co-assembly process. However, a few recent studies have started to use metagenomic-based distances combined with clustering to guide the co-assembly process of metagenomes [261–263], while another study has used metagenomic distances to guide the co-binning (or co-mapping) process [264].
To perform co-assembly of large sets of oceanic metagenomes, we chose the Tara Ocean dataset. We considered the 176 metagenomes sampled at three different sea depths: the superficial water surface (SUR), the deep chlorophyll maximum (DCM), which corresponds to a depth of around 5 to 30m, and in which has been observed the maximal

planktonic photosynthetic activity, and the mesopelagic zone (MES) which corresponds to depth greater than 200m. We only consider the prokaryotic part of these metagenomes, which corresponds to water being filtered by filters of 0.22 to 1.6 µm and 0.2 to 3 µm. We first computed the distances between all the 176 Tara Ocean metagenomic samples using Simka [243]. In order to better represent the distribution of metagenomes, we then computed a dimensionality reduction on the distance matrix output, using a principal coordinates analysis (PCoA). The distribution of the metagenomes along the first two axis of the PCoA, allowed to observe that the metagenomes were spread into three clusters (Fig. 4.2). One of this cluster contained metagenomes originating from arctic ocean, southern ocean, with also two samples from the southernmost regions of south atlantic ocean, constituting the Polar cluster. The second cluster can be called the Mesopelagic cluster, as it contains almost only metagenomes sampled at the mesopelagic depth, except for one metagenome sampled at the DCM depth. The third cluster contained the rest of the non-polar, non-mesopelagic metagenomes. These three clusters can be well separated through the first axis of the PCoA, and the distribution of the metagenomes along the first axis may imply that this axis represents the temperature of the sea water. We then performed a k-medoids clustering on the metagenomic distance matrix, coupled with the measure of the Silhouette index [250], which is a method used to identify the optimal partition of a dataset. The Silhouette index was maximised for a number of 3 clusters, confirming the relevance of the three clusters previously observed on the PCoA projection (Fig. 4.3).

Then, in order to compare the metagenomic-distance approach against a previous approach to discriminate through groups of metagenomes to co-assemble, we focused on the marine metagenomes dataset which corresponds to the same 93 oceanic metagenomes as processed in Delmont et al. [5], which are available at the European Bioinformatics Institute (EBI) repository under project ID ERP001736. Here, we computed distances between metagenomes using Simka [243], and identified optimal clustering solutions using the Silhouette index [250] to delineate unsupervised sets of samples to co-assemble. Applying this approach on the same set of metagenomes (n=93) as in [5], we identified 24 optimal clusters. This number of clusters is significantly higher compared to the 12 clusters (Fig. 4.4A) based on the oceanic regions, which suggests that a different partition may be more relevant for co-assembly. As this optimal clustering generates smaller clusters, in order to insure a fair comparison between both approaches, we further identified a sub-optimal clustering (supp.fig 4.5) whose number of co-assembly sets is comparable

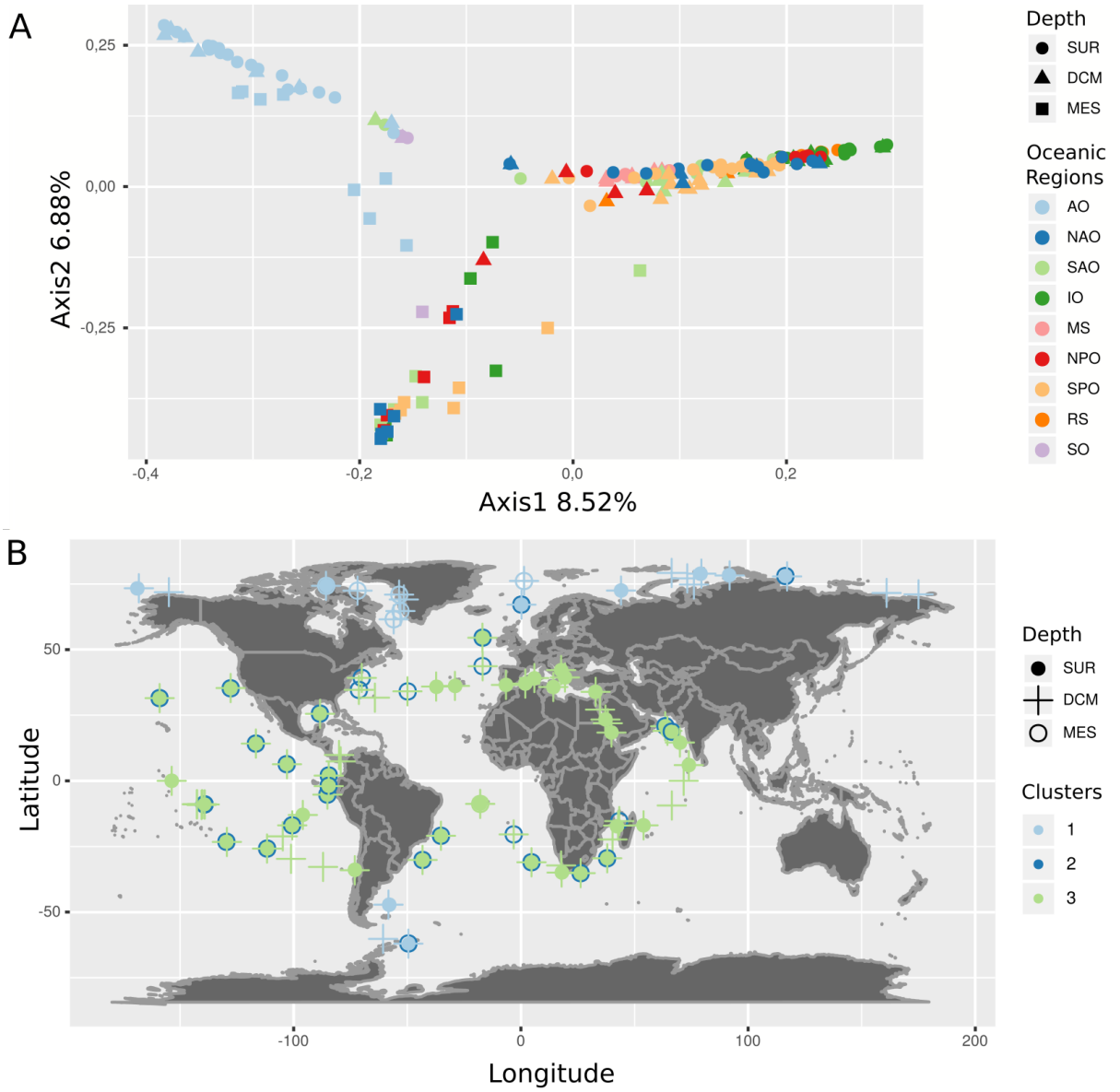

Figure 4.2 **Evaluating distribution of metagenomes using dimension-reduction on metagenomic-distance based matrix.**
(A) Projection of metagenomes along the first two axis of the PCoA computed on the matrix distance. Colours reflect the oceanic regions as defined by the Tara Oceans consortium, while points shapes represent the sea depth at which sampling was performed: surface layer (SUR), Deep Chlorophyll Maxima (DCM) or mesopelagic zone (MES). AO = Arctic Ocean, IO = Indian Ocean, MS = Mediterranean Sea, NAO = North Atlantic Ocean, SAO = South Atlantic Ocean, RS = Red Sea, NPO = North Pacific Ocean, SPO = South Pacific Ocean, SO = Southern Ocean. **(B)** Geographic distribution of the three clusters found with the PCoA. Each dot is a metagenome, the SUR depth is represented with plain dots, the DCM with crosses, and the MES with circles. X-axis and Y-axis are the values of the two first axis of the PCoA for each metagenome, with the percentage of explained variance by each axis.

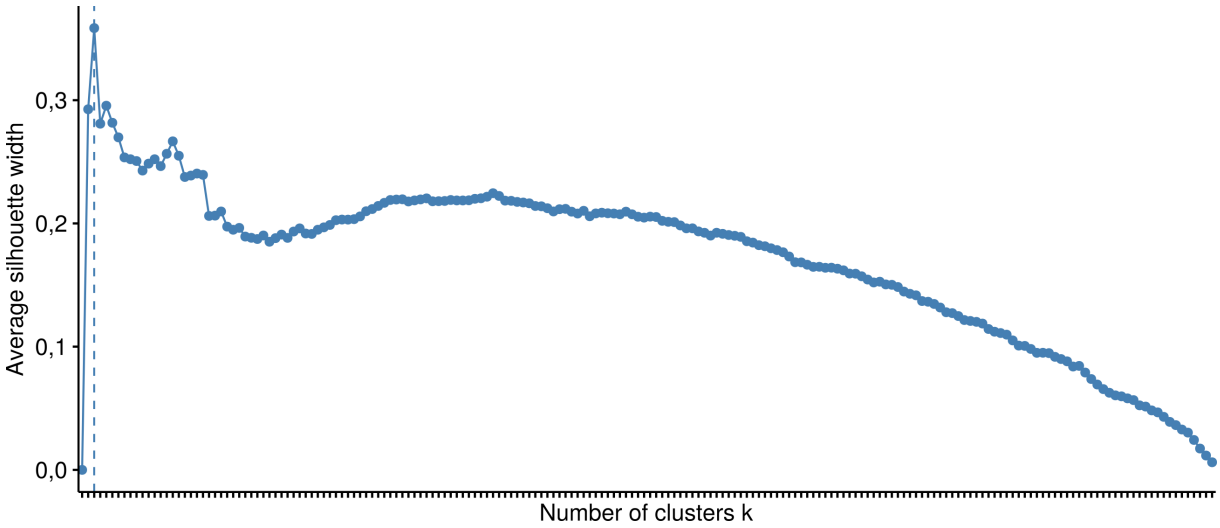

Figure 4.3 **Evaluation of the partition of the whole Tara metagenomes dataset.** Silhouette scores obtained when clustering the metagenomes into k groups. X-axis: values of k tested, k varying from 2 to $n-1$, with $n = 176$ (total number of samples). Y-axis: values of the Silhouette score estimated for value k tested. A high Silhouette score yields a better clustering. A maximum score is observed for $k = 3$, (dashed vertical line) which thus represents the optimal clustering.

to the number of oceanic regions used in [5]. This second clustering identified 11 clusters, which did not match the previously defined oceanic regions (Fig. 4.4A).

To evaluate the potential impact of co-assembly on assembly quality, we computed classical assembly quality metrics (N50 and L50) for both approaches. N50 represents the shortest contig length to cover at least 50% of the metagenome assembly [265], while L50 represents the smallest number of contigs whose added lengths cover 50% of the metagenome assembly [266]. The metagenomic distance-based (MD) and the oceanic regions (OR) approaches actually reconstructed contigs of similar quality. No significant differences were detected in either the number of misassemblies, or the N50 and L50 metrics (Fig. 4.6). When considering the total number of bins generated following both co-assembly strategies, we found that both approaches reconstructed very similar numbers of bins: 10,748 bins are generated using the MD approach, and 10,233 bins using the OR approach (Fig. 4.4B). To further compare both co-assembly strategies, as these bins may be very different in composition, we performed MAGs de-replication [193]. The MD approach systematically reconstructed more MAGs than the OR approach, at both species (95% ANI) and strain (99% ANI) levels (Fig. 4.7B). Considering MAGs quality, medium quality (MQ) MAGs reconstructed by the MD approach were significantly more complete

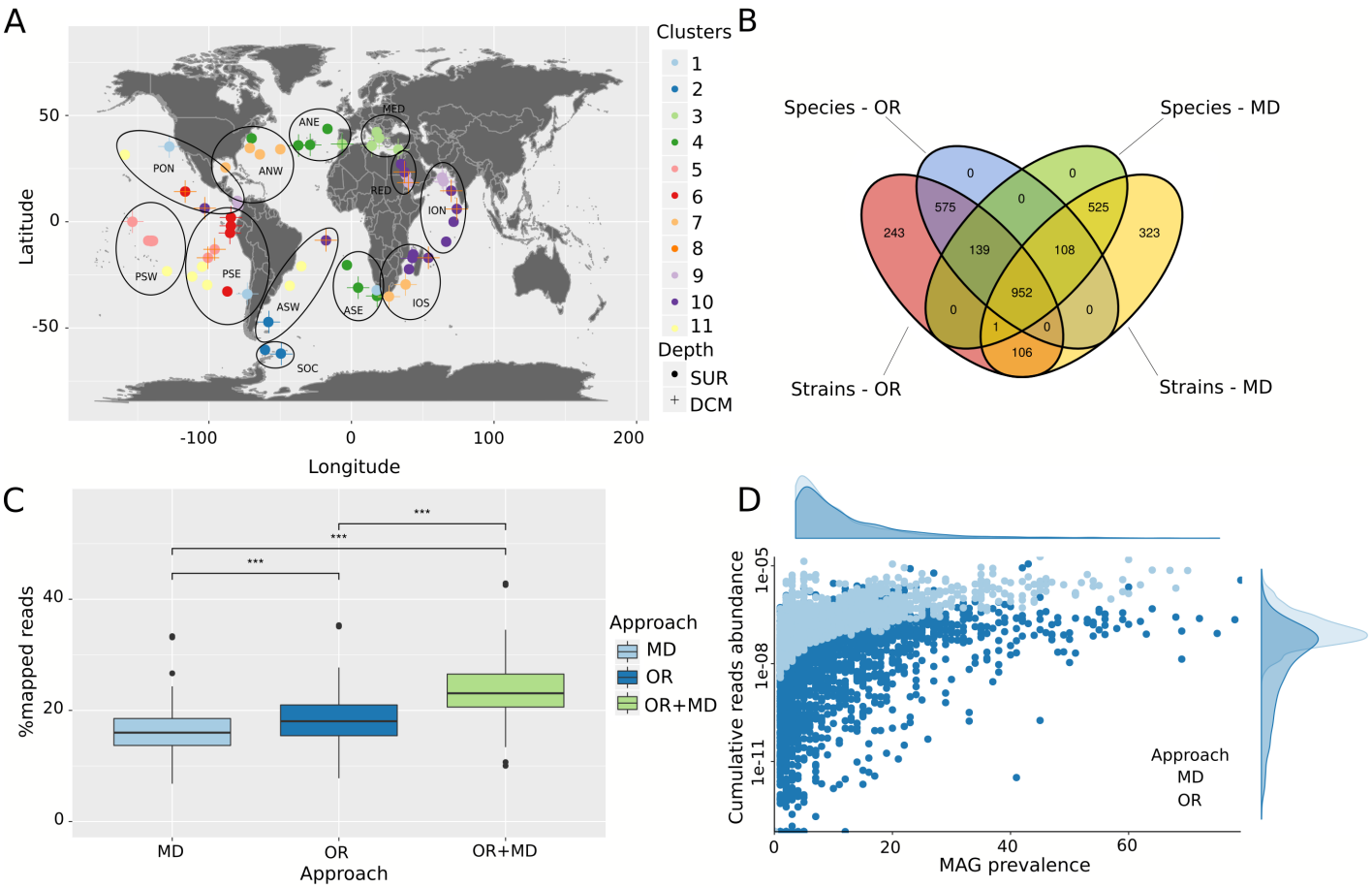

Figure 4.4 **Evaluating the metagenomic distance-based (MD) approach against the oceanic regions (OR) approach for delineating groups of samples to co-assemble.** **(A)** Repartition of clusters obtained with Simka. Each dot represents a metagenome obtained at a sampling station, with metagenomes located at Surface (SUR) represented as dots, and metagenomes situated at the Deep Chlorophyll Maxima (DCM) depth as crosses. Colours represent the cluster to which the metagenome belongs. Oceanic regions are represented as dark circles: ANE = Atlantic North-East, ANW = Atlantic North-West, ASE = Atlantic South-East, ASW = Atlantic South-West, ION = Indian Ocean North, IOS = Indian Ocean South, MED = MEDiterranean Sea, PON = Pacific Ocean North, PSE = Pacific South-East, PSW = Pacific South-West, RED = RED sea, SOC = Southern OCean. **(B)** Repartition of the common MAGs obtained after common de-replication between the two approaches. **(C)** Percentage of mapped reads on MAGs reconstructed by each approach, and on combined MAGs from both approaches, considering all mapping reads. MD+OR = MAGs from MD and OR approaches were combined together prior to reads mapping. **(D)** Prevalence-Abundance plot for MAGs reconstructed by both approaches (X-axis: MAG prevalence = number of metagenomic samples in which a MAG has a horizontal coverage above 0.3; Y-axis: MAG cumulative abundance. Percentage of mapped reads divided by the length of the MAG).

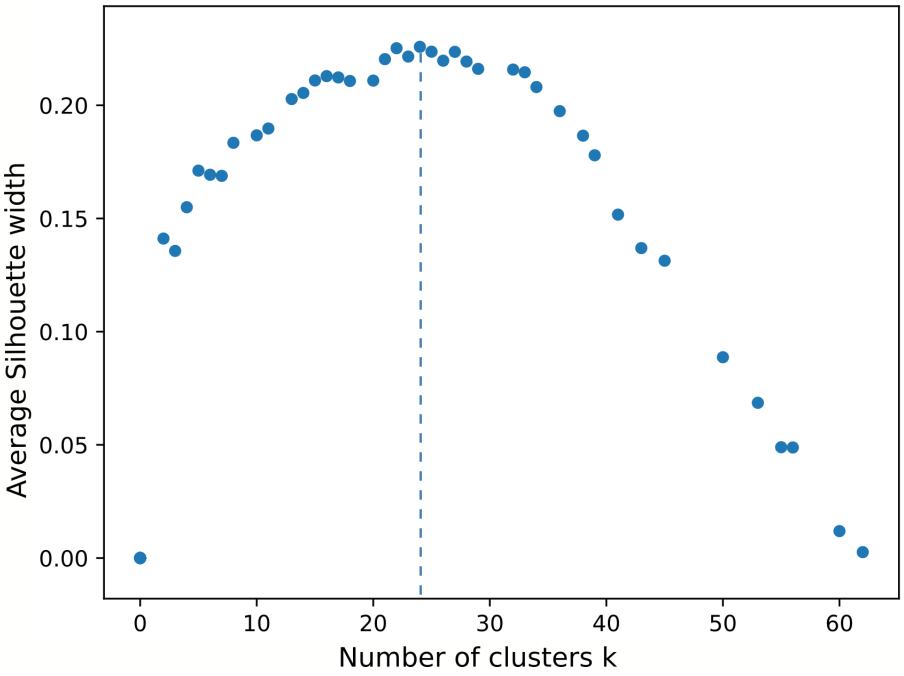

Figure 4.5 **Identifying metagenomic distance-based optimal clusters for Tara Ocean metagenomes.**
Silhouette scores obtained when clustering the metagenomes into k groups. X-axis: values of k tested, k varying from 2 to $n-1$, with $n = 93$ (the total number of samples used). Y-axis: Values of the Silhouette score estimated for value k tested. A high Silhouette score yields a better clustering. A maximum score is observed for $k = 24$ (dashed vertical line), which thus represents the optimal clustering.

(Mann–Whitney U test, $p = 0.01$, Fig. 4.8B), but evaluated as more contaminated (using checkM) than MQ MAGs reconstructed with the OR approach (Mann–Whitney U test, $p = 6.828 \cdot 10^{-05}$, Fig. 4.8D). However, when considering the GUNC contamination metric [14], contamination levels observed in MQ MAGs of the MD approach were significantly lower than MQ MAGs of the OR approach (Mann–Whitney U test, $p = 2.352 \cdot 10^{-07}$, Fig. 4.8F). Because GUNC assesses gene contamination based on all taxonomically annotated genes in a given genome, this latter approach may be considered as more robust than the checkM metric, and lead us to conclude that the MD approach actually reconstructed less contaminated MAGs. We found no significant differences in quality (completeness and contamination) for high quality (HQ) MAGs reconstructed by both approaches (Fig. 4.8). In addition, taxonomic annotations of strain-level de-replicated MAGs revealed a higher diversity recovered in the MD MAGs as compared to the OR MAGs (Fig. 4.9) in terms of number of distinct bacterial taxa, with a greater amount of annotated MAGs in MD

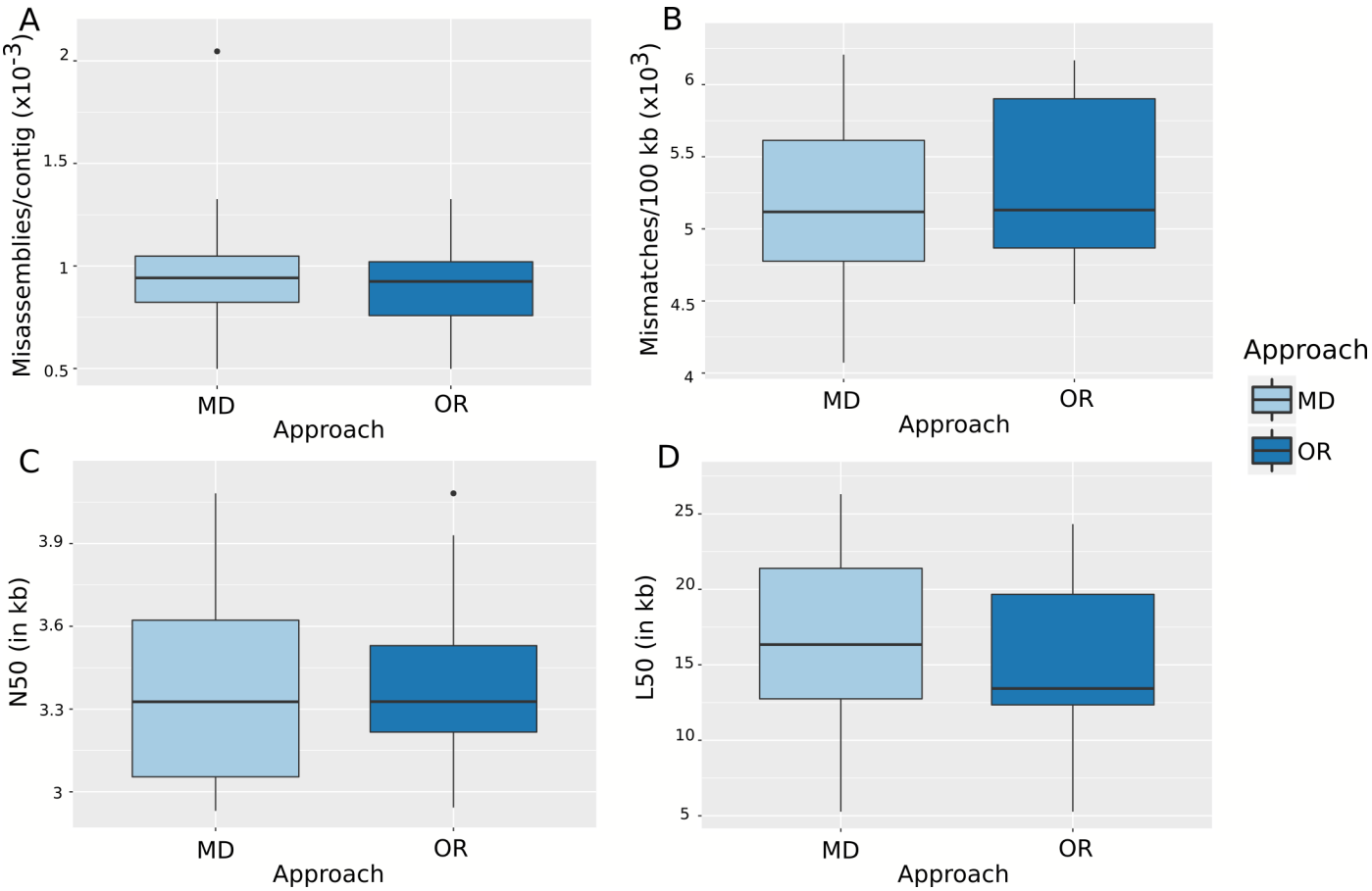

Figure 4.6 **Quality of the assembly of each Tara approach.**
A Comparison of the number of misassemblies normalized by the number of contigs per assembly of the metagenomic distance (MD) and the oceanic regions (OR) approaches; **B** Number of mismatches detected per 100kb of alignments in contigs per approach; **C** N50 values; **D** L50 values. No significant differences were found between the two approaches (Mann-Withney U test).

($n = 2006$) compared to OR ($n = 1869$) MAGs 4.9.

Next, we also performed a global de-replication of MAGs in order to compare sets of MAGs recovered by both approaches at species and strain levels. Remarkably, we observed that both approaches reconstructed a very high number of exclusive MAGs (Fig. 4.4B). The OR approach reconstructed 575 species-level and 243 strain-level MAGs that were not recovered by the MD approach, while the latter did reconstruct 525 species-level and 323 strain-level MAGs that were not recovered by the OR approach. This result strongly emphasizes the influence of the co-assembly step prior to genome binning, in particular re-

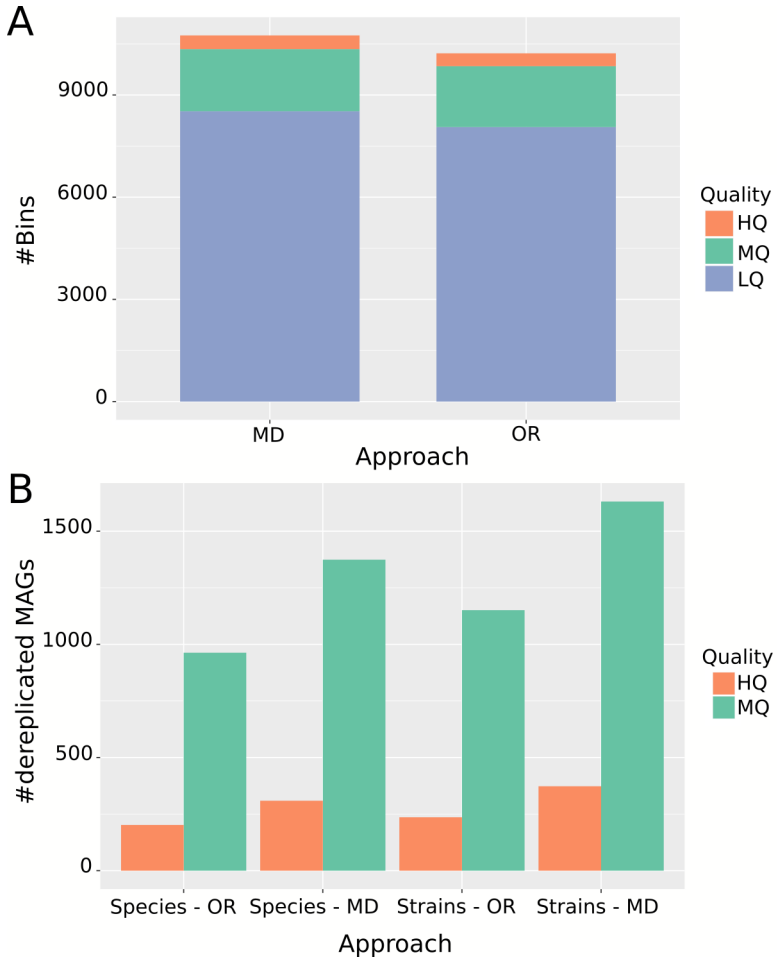

Figure 4.7 **Evaluating the metagenomic distance-based (MD) approach against the oceanic regions (OR) approach for delineating groups of samples to co-assemble.** **(A)** Total number of bins obtained after binning step. HQ = High Quality, MQ = Medium Quality, LQ = Low Quality. **(B)** Number of reconstructed MAGs after independent de-replication for each approach. Species resolution consists in a 95% ANI score de-replication, while Strains resolution consists in a 99% ANI score de-replication.

garding how metagenomes are grouped for co-assembly. Given this observation, we aimed at determining which approach could captured a greater proportion of metagenomic diversity by back-mapping reads on MAGs generated by both approaches. While we observed a lower proportion of reads mapping to MD MAGs, as compared to OR MAGs, this proportion significantly increased when mapping on combined MAGs from both approaches. This result confirms that distinct and complementary MAGs are reconstructed using each approach. However, when only considering reads mapping to MAGs detected in samples,

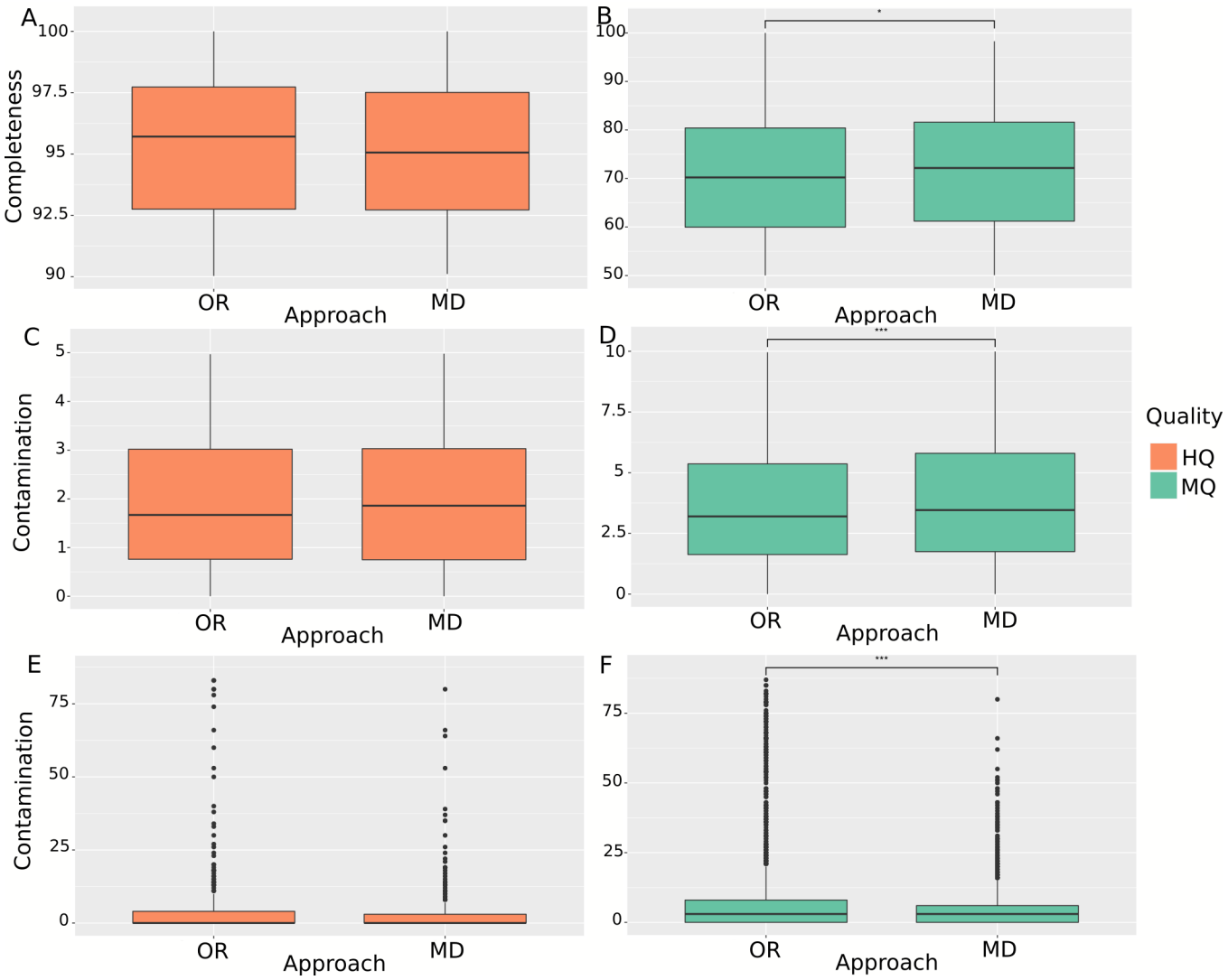

Figure 4.8 **Quality of reconstructed MAGs from the Tara Ocean metagenomes.** Comparison of the oceanic regions (OR) approach against the metagenomic distance approach (MD). HQ = High Quality, MQ = Medium Quality. **(A)** Completeness estimated for HQ MAGs; **(B)** Completeness estimated for MQ MAGs; **(C)** Contamination measured using SCGs for HQ MAGs; **(D)** Contamination measured using SCGs for MQ MAGs; **(E)** Contamination measured using all genes detected in the sequences of HQ MAGs; **(F)** Contamination measured using all genes detected in the sequences of MQ MAGs.

i.e. in which a given MAG has a minimum horizontal coverage (or breadth) of 30%, the MD approach recruited significantly more metagenomic reads as compared to the OR approach (Mann-Withney U test, $p < 2.2 \cdot 10^{-16}$, Fig. 4.4D). Thus, although the OR MAGs were detected in more samples as compared to MD MAGs (Mann-Withney U test,

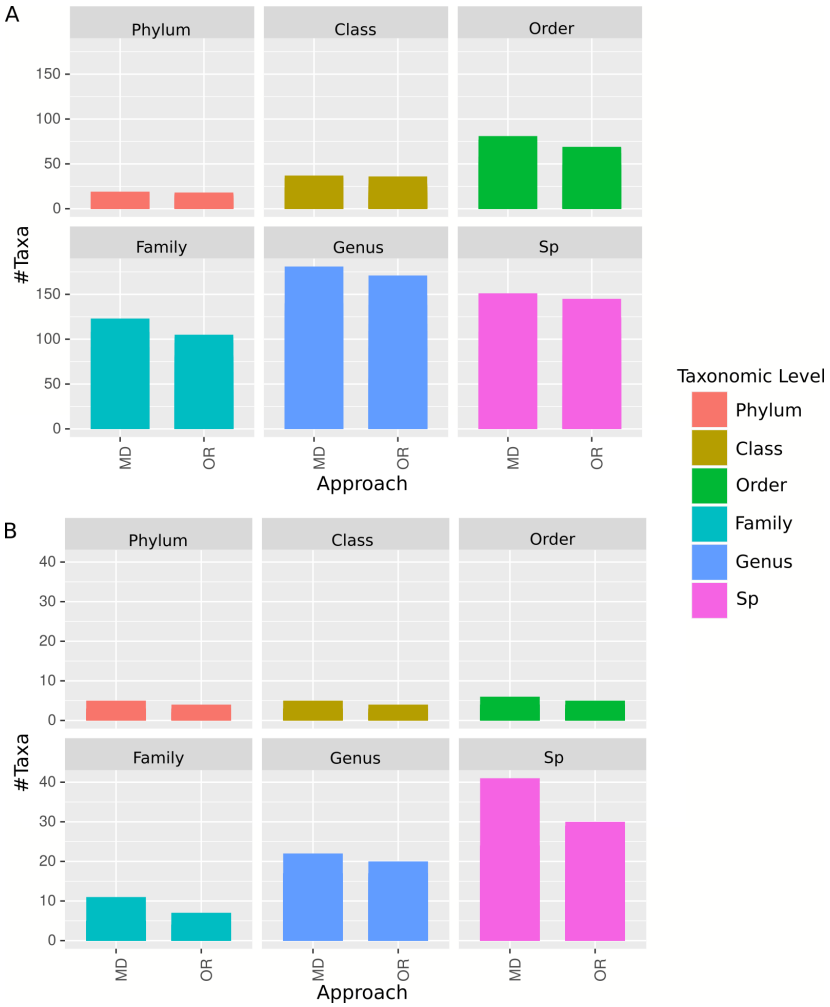

Figure 4.9 **Taxonomic diversity of reconstructed MAGs from the marine metagenomes.**
Comparison of the number of taxa retrieved from MAGs reconstructed following each approach. **(A)**: Number of bacterial taxa assigned to MAGs from both approaches, per taxonomic level ; **(B)**: Number of archaeal taxa assigned to MAGs from both approaches. Taxonomic annotation was performed using gtdb-tk on de-replicated MAGs at strain level (99% ANI score) using dRep. For each panel, the Y-axis represents the number of taxa found withing the complete set of MAGs of each approach. Number of MAGs assigned to a bacterial annotation is 1729 for MD and 1595 for OR, while there are 276 and 273 MAGs with an archeal annotation in MD and OR, respectively.

$p = 4.6 \cdot 10^{-10}$), the MD MAGs significantly improved the number and quality of reconstructed MAGs. In order to determine the prevalence of reconstructed MAGs in the set of MAGs of each reconstruction method (Fig. 4.4D), we performed a global de-replication

of the *TARA* Oceans MAGs, performed by combining sets of MAGs reconstructed with both approaches using dRep as described in the section 4.3 "Design and Implementation" below. Once this global de-replication was performed, the total number of de-replicated MAGs (dMAGs) related to one approach is thus:
$N_{ir} = n_{ir} + m_{ijr}$
with n_ir the number of dMAGs already reconstructed by the approach i at the resolution r, and m_{ij} the number of dMAGs reconstructed by approach j, but located in a de-replication cluster containing at least one MAG reconstructed by the approach i, at the de-replication resolution r. From the dRep output, we can identify the de-replication cluster each MAG belongs to, and the number of members located in the same de-replication cluster. We can then list, for a given de-replication resolution r, the set of dMAGs related to one approach i, searching for each non-unique dMAG (*i.e.* having at least one neighbour in its de-replication cluster) of approach j, if there is at least one MAG from approach i. Thus, to detect shared dMAGs between both approaches, we identified the common elements between the four sets of N_{ir} dMAGs.
To detect shared genomes between sets of genomes related to different de-replication resolutions (Fig. 4.4B), we should point out that the set of dMAGs at species-level for one approach is completely included in the set of dMAGs at strain-level from this same approach. Thus, we can non-ambiguously identify clustering relationships between two MAGs from different de-replication levels, or find exclusive MAGs reconstructed by one approach, but present at both species and strain levels.

4.3.1 Benchmarking assembly-binning strategies on simulated metagenomes

Different strategies for assembly and binning are currently used in the literature, each of them having its own advantages and disadvantages [110]. Thus, we defined four assembly-binning strategies representing the most currently used approaches to reconstruct MAGs. Namely, we considered single-assembly (SA, i.e. the assembly of a single metagenome) and co-assembly (CA, i.e. the joint assembly of n metagenomes) approaches, as well as single-binning (SB, i.e. genome binning solely using (co-)abundance information from metagenome(s) used to perform the (co-)assembly) and co-binning (CB, i.e. genome binning using co-abundance information from all metagenomes) approaches. We thus evaluated the following four strategies: Single-Assembly with Single-Binning (SASB),

Single-Assembly with Co-Binning (SACB), Co-Assembly with Single-Binning (CASB), and Co-Assembly with Co-Binning (CACB). We compared the performances of these four strategies on three different datasets, the CAMI [175] high complexity dataset, a lower complexity mockdataset generated using CAMISIM [267], and a human microbiome dataset from the Human Microbiome Project [11].

First, to evaluate and compare these four strategies on simulated metagenomes, we applied our MD clustering algorithm on the CAMI high-complexity dataset [175]. The CAMI high-complexity dataset is composed of five metagenomic samples simulated from a community of 596 known reference genomes and 478 circular elements. The optimal solution identified for the co-assembly regrouped all five metagenomes, probably due to the small number of metagenomes ($n = 5$) and the fact that they were simulated from the same pool of reference genomes. Therefore, only one co-assembly (of all 5 samples) was performed, and the CACB and CASB strategies were thus equivalent. Following genome binning using MetaBAT2 [16], the SACB strategy reconstructed the highest number of bins ($>$ 400 genome bins), while the CASB and SASB strategies reconstructed about 300 and 200 bins, respectively (Fig. 4.10A).

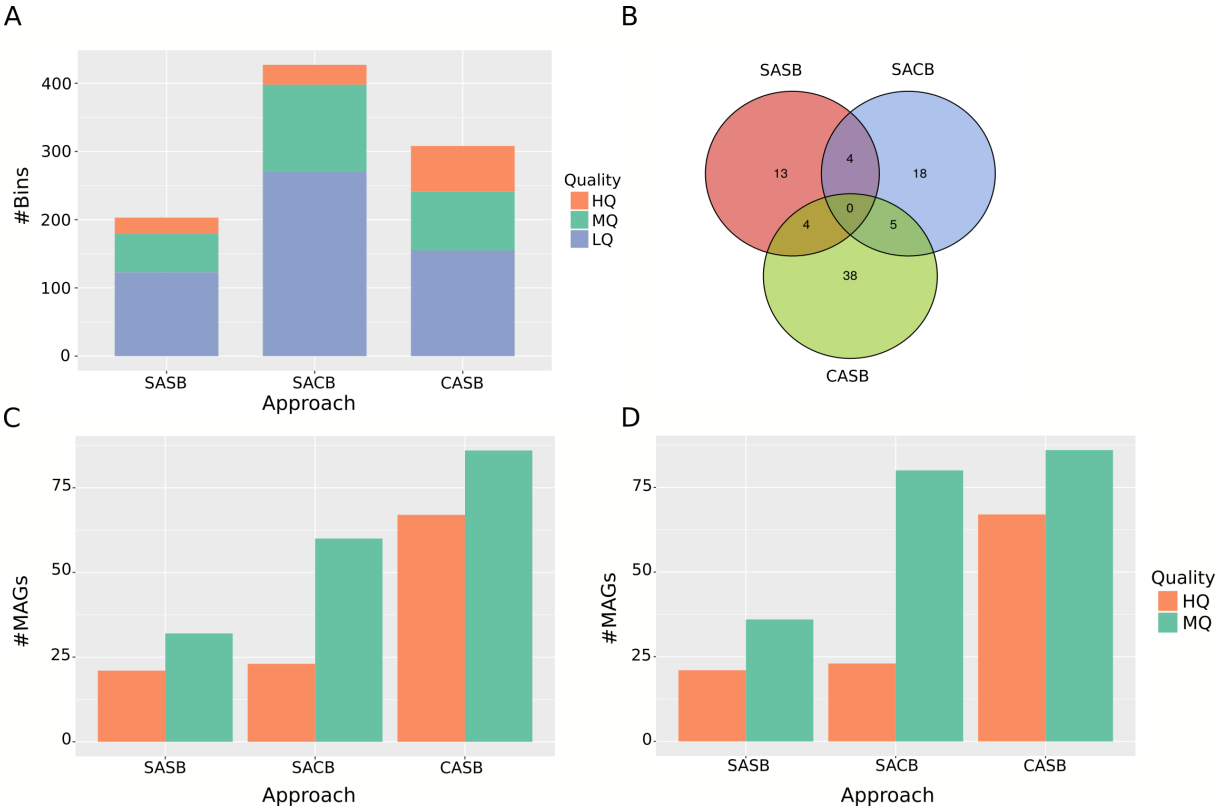

Figure 4.10 **Evaluating assembly-binning strategies on the CAMI dataset.** **(A)** Total number of bins obtained after binning step. Colours represent quality of genome bins estimated using CheckM: High Quality (HQ), Medium Quality (MQ), and Low Quality (LQ). **(B)** Number of MAGs mapping to a source genome within each strategy, corresponding to the number of expected genomes in the set of MAGs of each strategy. The diagram thus represents the common genomes found in each strategy. **(C,D)** Number of reconstructed MAGs after independent de-replication using dRep for each binning strategy, at **(C)** Species resolution, consisting in a 95% ANI score de-replication; and **(D)** Strains resolution, consisting in a 99% ANI score de-replication. SASB: Single-Assembly-Single-Binning, SACB: Single-Assembly-Co-Binning, CASB: Co-Assembly-Single-Binning.

After de-replication, we compared the MAGs obtained for each strategy to the CAMI reference source genomes. When considering the distribution of expected genomes across all three strategies, we observed that the CASB strategy reconstructed more expected genomes than both single-assembly strategies (SASB and SACB). Surprisingly, we did not find expected genomes common to all strategies (Fig. 4.10B), which highlights the actual complementarity of these strategies. When considering only de-replicated genomes, CASB produced the highest number of MAGs. This difference was clear for HQ MAGs, for which CASB produced about 2.5 times more MAGs as compared to single-assembled strategies,

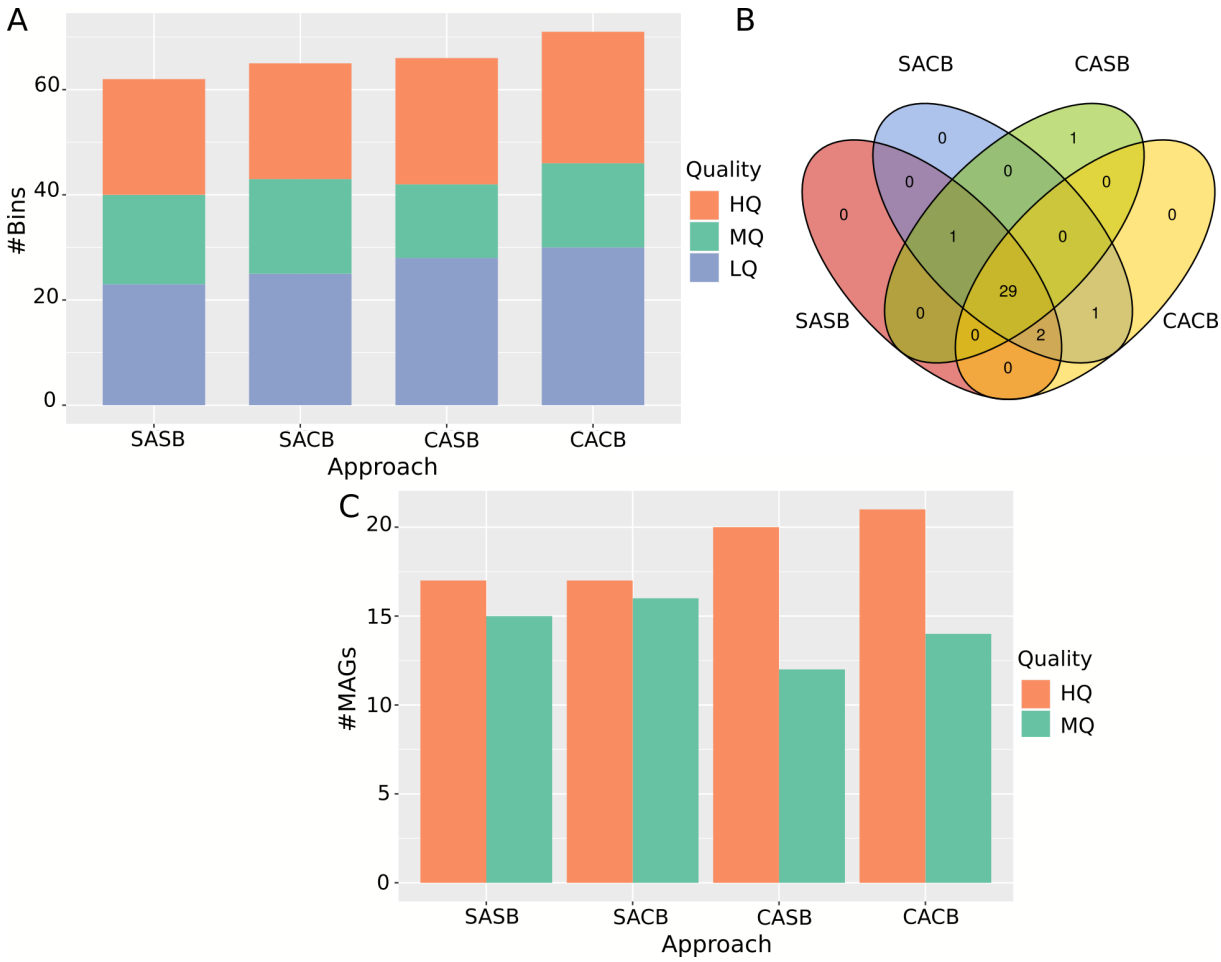

Figure 4.11 **Evaluating assembly-binning strategies on simulated metagenomes.** **(A)** Total number of bins obtained after the binning step. Colours represent quality of genome bins estimated using CheckM: High Quality (HQ), Medium Quality (MQ), and Low Quality (LQ). **(B)** Number of source genomes found in each strategy. Each number represents the number of times a MAG from a strategy maps against a source genome. Intersections represent common genomes between strategies. **(C)** Number of de-replicated MAGs obtained, after independent de-replication by dRep for each strategy. As the genomes are all represented with one single strain, de-replication at both species or strain resolution gives the same number of de-replicated genomes, so only one de-replication resolution is shown.

with both SACB and SASB generating a comparable number of HQ MAGs (Fig. 4.10CD). The number of reconstructed MAGs was also dependent of the de-replication level. At strain level, both single-assembly approaches reconstructed more non-redundant MAGs compared to species level, while CASB reconstructed the same number of MAGs at both species and strain levels. However, this increase only concerned the MQ MAGs, as the number of HQ MAGs remained unchanged (Fig. 4.10D). We did not find any significant

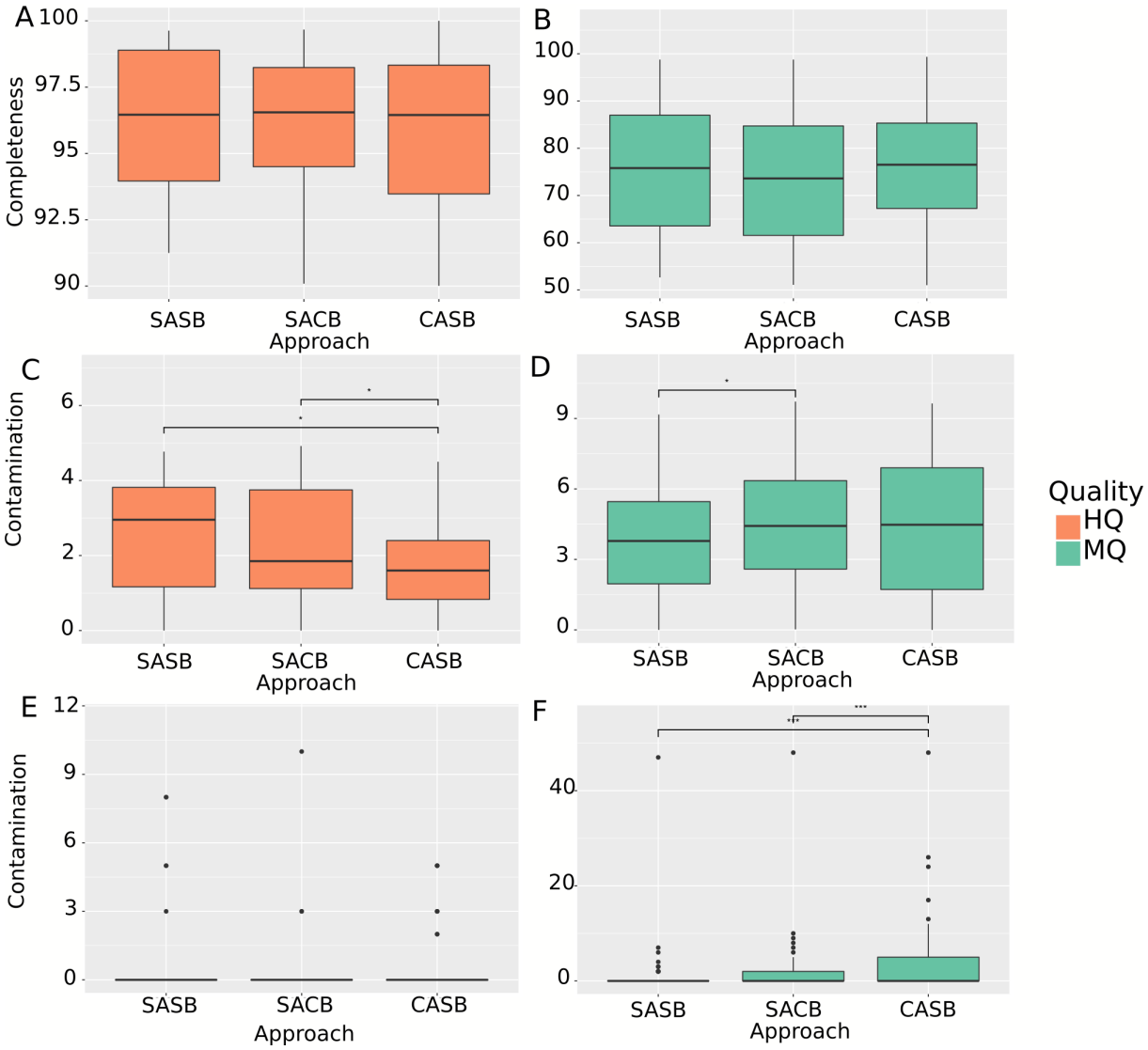

Figure 4.12 **Quality of reconstructed MAGs from the CAMI dataset.** Comparison of the quality of MAGs reconstructed with each strategy on the CAMI dataset. HQ = High Quality, MQ = Medium Quality. **(A)** Completeness estimated for HQ MAGs; **(B)** Completeness estimated for MQ MAGs; **(C)** Contamination measured using SCGs for HQ MAGs; **(D)** Contamination measured using SCGs for MQ MAGs; **(E)** Contamination measured using all genes detected in the sequences of HQ MAGs; **(F)** Contamination measured using all genes detected in the sequences of MQ MAGs.

differences in MAGs completeness between the different strategies, considering either HQ MAGs or MQ MAGs (Fig. 4.12 A&B). However, we did observe differences in contamination estimated from Single-Copy Genes (SCGs) using checkM. CASB HQ MAGs were less contaminated than SASB (Mann-Withney U test, p=0.01) and SACB (Mann-Withney U

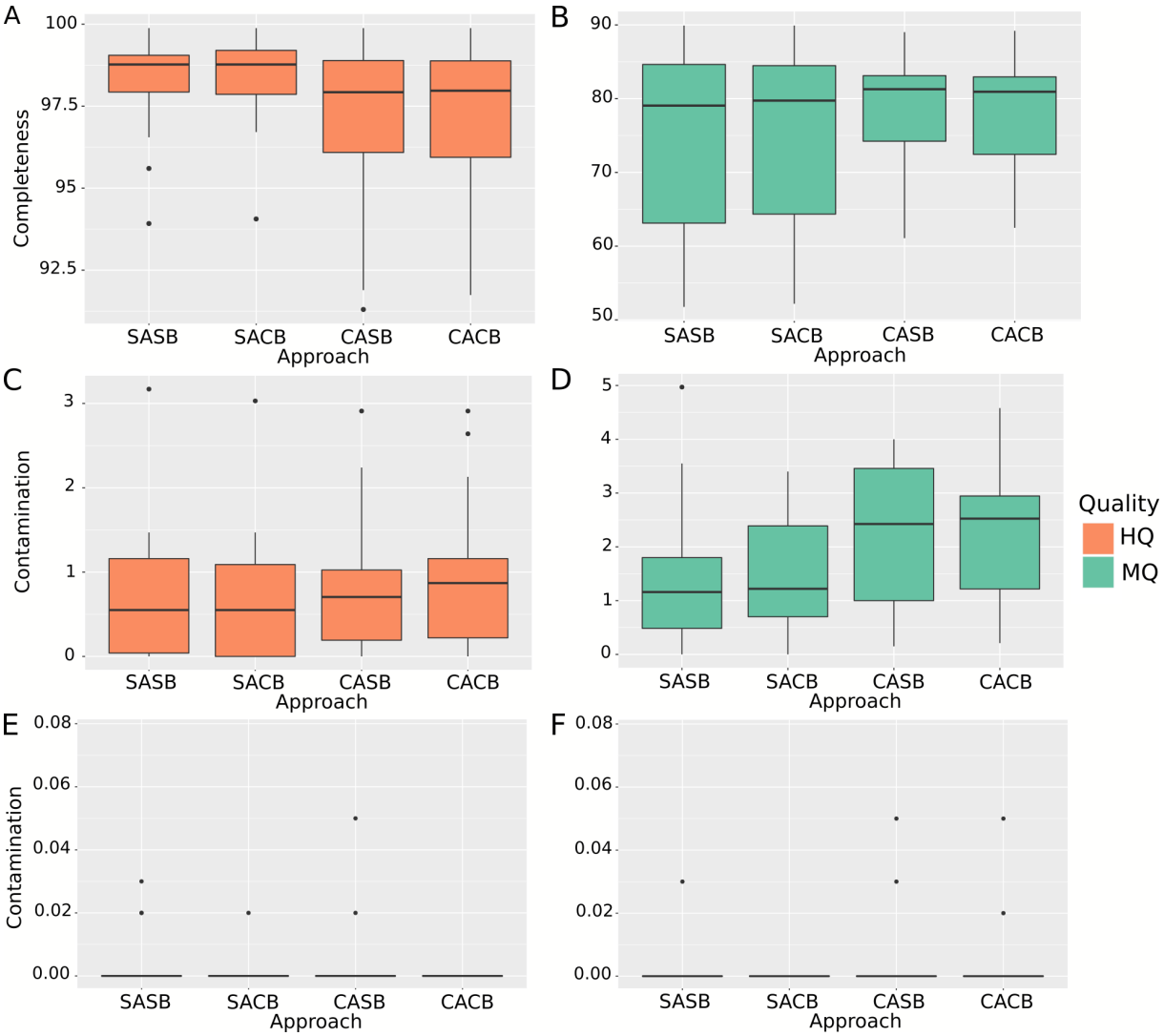

Figure 4.13 **Quality of reconstructed MAGs from the mockdataset.**
Comparison of the quality of MAGs reconstructed with each strategy on the mockdataset. HQ = High Quality, MQ = Medium Quality. **(A)** Completeness of HQ MAGs; **(B)** Completeness of MQ MAGs; **(C)** Contamination of HQ MAGs measured with CheckM; **(D)** Contamination of MQ MAGs measured with CheckM; **(E)** Contamination of HQ MAGs measured with GUNC; **(F)** Contamination of MQ MAGs measured with GUNC.

test, $p = 0.04$) HQ MAGs, while SACB MQ MAGs were less contaminated than SASB MQ MAGs (Mann-Withney U test, $p = 0.03$) (Fig. 4.12 C&D). When considering MAGs contamination estimated using taxonomically annotated genes with GUNC, CASB MAGs were predicted most contaminated (Fig. 4.12 E&F), with significant differences observed with either SASB (Mann-Withney U test, $p = 3 \cdot 10^{-4}$) and SACB (Mann-Withney U

test, $p = 2 \cdot 10^{-4}$) MAGs. We did not find any other differences in contamination levels between the four strategies.

Given that the MD clustering approach did not identify optimal clusters to co-assemble within the CAMI dataset, we used CAMISIM [267] to simulate an additional metagenomic dataset with a higher number of samples and a lower complexity. We thus simulated 20 metagenomes with a similar diversity of 100 reference genomes. On this simulated dataset, the MD clustering approach identified 8 optimal clusters to co-assemble. Here, the co-assembly-based strategies (CACB and CASB) reconstructed more bins than the single-assembly-based strategies (Fig. 4.11A), also when considering only HQ bins. After de-replication, we aimed to identify expected genomes among recovered MAGs by mapping them to reference genomes used for the metagenomes simulation. The majority ($n = 29$) of expected genomes we identified were reconstructed in all four strategies (Fig. 4.11B). The SACB strategy recovered a short majority of expected genomes ($n = 33$), as compared to CACB ans SASB ($n = 32$), and CASB ($n = 31$). However, the number of de-replicated MAGs was higher for both co-assembly strategies compared to single-assembly strategies (Fig.4.11C).

The drop in de-replicated MAGs from single-assembly strategies is likely a consequence of the higher number of assemblies performed in both SASB and SACB strategies. As single-assemblies are more numerous than co-assemblies, there is thus a higher probability to reconstruct, independently, several times the same MAG. Finally, using this simulated dataset, we did not detect any significant differences in the quality of MAGs reconstructed by the four strategies, neither in their completeness nor in their contamination levels (Fig. 4.13).

4.3.2 Comparing assembly-binning strategies on real metagenomes

To further compare the four genome reconstruction strategies, we applied them to a real metagenomic dataset, which is more complex in terms of species diversity and composition. Human gut microbiome studies represent a large fraction of publicly available metagenomes and are also good case studies as they represent metagenomes with intermediate complexity compared to soil or ocean metagenomes. Thus, we focused on analysing a selection of 150 metagenomes of human gut microbiomes from the Integrative Human Microbiome Project (HMP) [11]. Here, the MD-based clustering approach identified 64 metagenomic clusters to co-assemble. When comparing all four strategies before de-replication, both single-assembly strategies reconstructed more genome bins than both

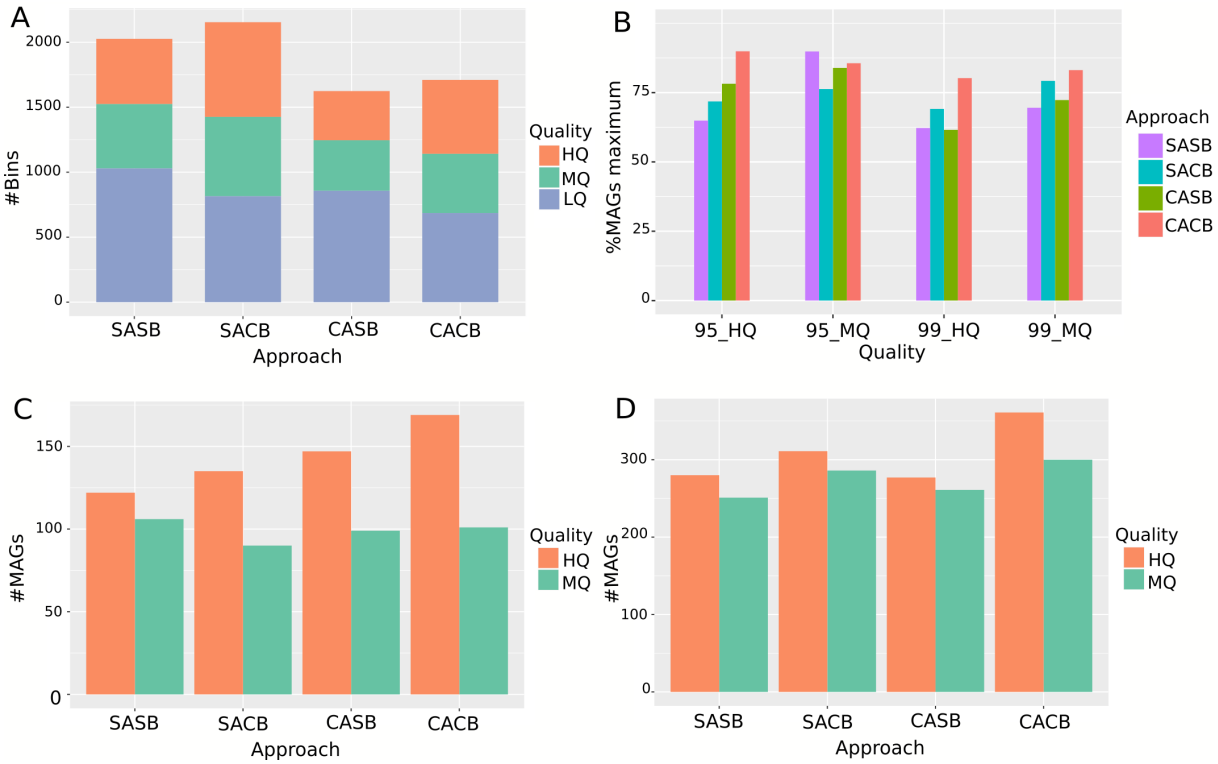

Figure 4.14 **Evaluating the binning strategies on the HMP dataset.**
(A) Total number of bins reconstructed per strategy. Colours represent the MAGs qualities, estimated with CheckM. **(B)** Proportion of MAGs reconstructed for each strategy, after common de-replication of the four strategies, at the species resolution (95% identity) or at the strain resolution (99% identity). Number of de-replicated MAGs from each strategy is compared to the number of maximum expected MAGs, which is the number of MAGs obtained after de-replication of all the four strategies together. **(C,D)**: Number of reconstructed MAGs after independent de-replication using dRep for each binning strategy, at **(C)** Species resolution, consisting in a 95% ANI score de-replication; and **(D)** Strains resolution, consisting in a 99% ANI score de-replication. SASB: Single-Assembly-Single-Binning, SACB: Single-Assembly-Co-Binning, CASB: Co-Assembly-Single-Binning, CACB: Co-Assembly-Co-Binning. HQ: High Quality, MQ: Medium Quality, LQ: Low Quality.

co-assembly strategies (Fig. 4.14A). Next, in order to determine how many MAGs we could expect to reconstruct at best by each strategy, we de-replicated altogether genome bins reconstructed by all strategies. The resulting number of de-replicated MAGs thus represents the highest number of MAGs we would be able to reconstruct with the HMP dataset combining all four strategies. We then compared each strategy by considering what proportion of the maximum number of MAGs it was able to reconstruct (Fig. 4.14B). After de-replication at the species level, despite the fact that single-assembly strategies recovered more bins, we observed that both co-assembly strategies reconstructed more MAGs

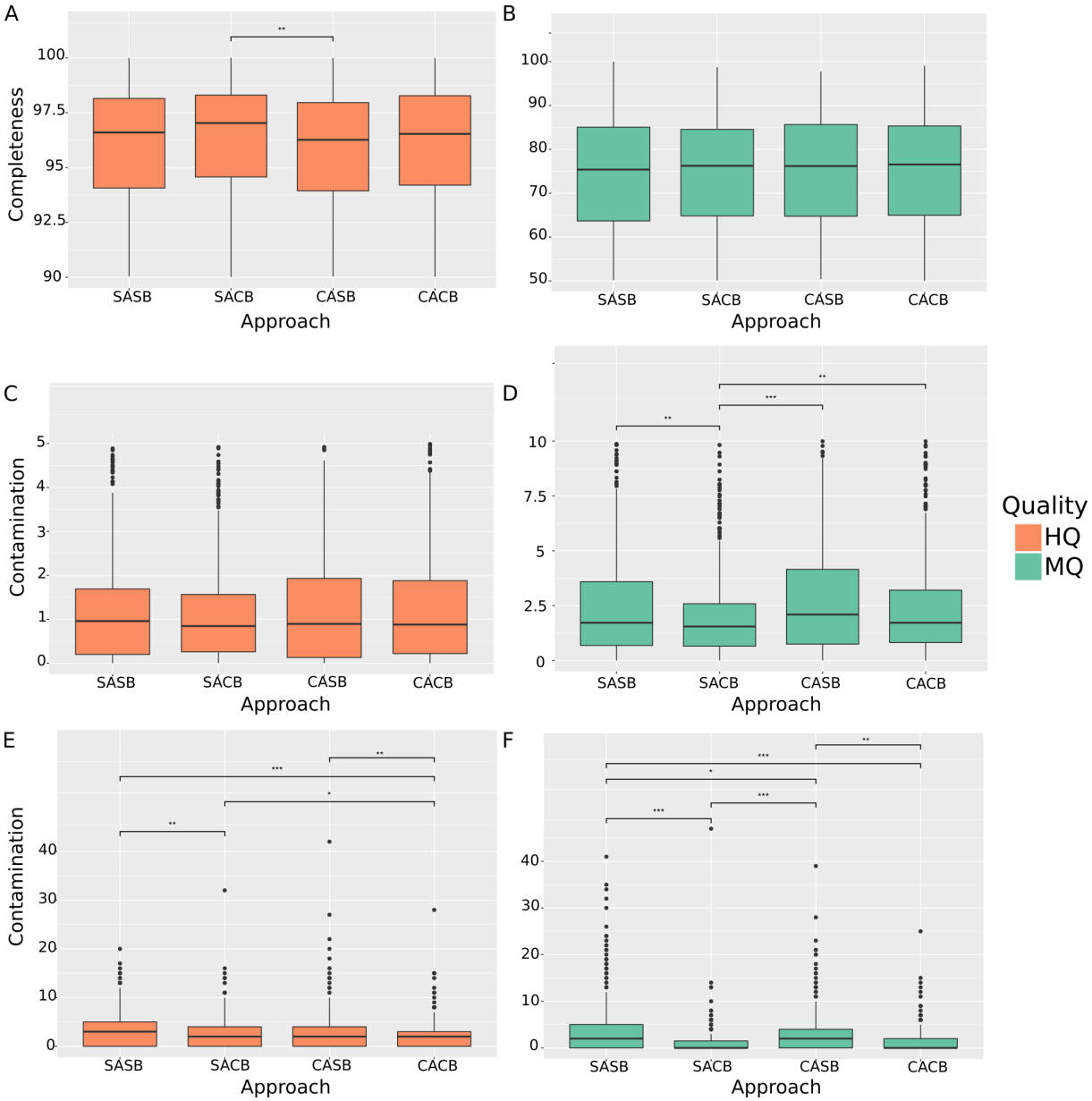

Figure 4.15 **Quality of reconstructed MAGs from the HMP dataset.** Comparison of the quality of MAGs reconstructed with each strategy on the HMP dataset. HQ = High Quality, MQ = Medium Quality. **(A)** Completeness estimated for HQ MAGs; **(B)** Completeness estimated for MQ MAGs; **(C)** Contamination measured using SCGs for HQ MAGs; **(D)** Contamination measured using SCGs for MQ MAGs; **(E)** Contamination measured using all genes detected in the sequences of HQ MAGs; **(F)** Contamination measured using all genes detected in the sequences of MQ MAGs.

than single-assembly strategies. Also, for both co-assembly and single-assembly strategies, the co-binning actually allowed to reconstruct more MAGs than the single-binning

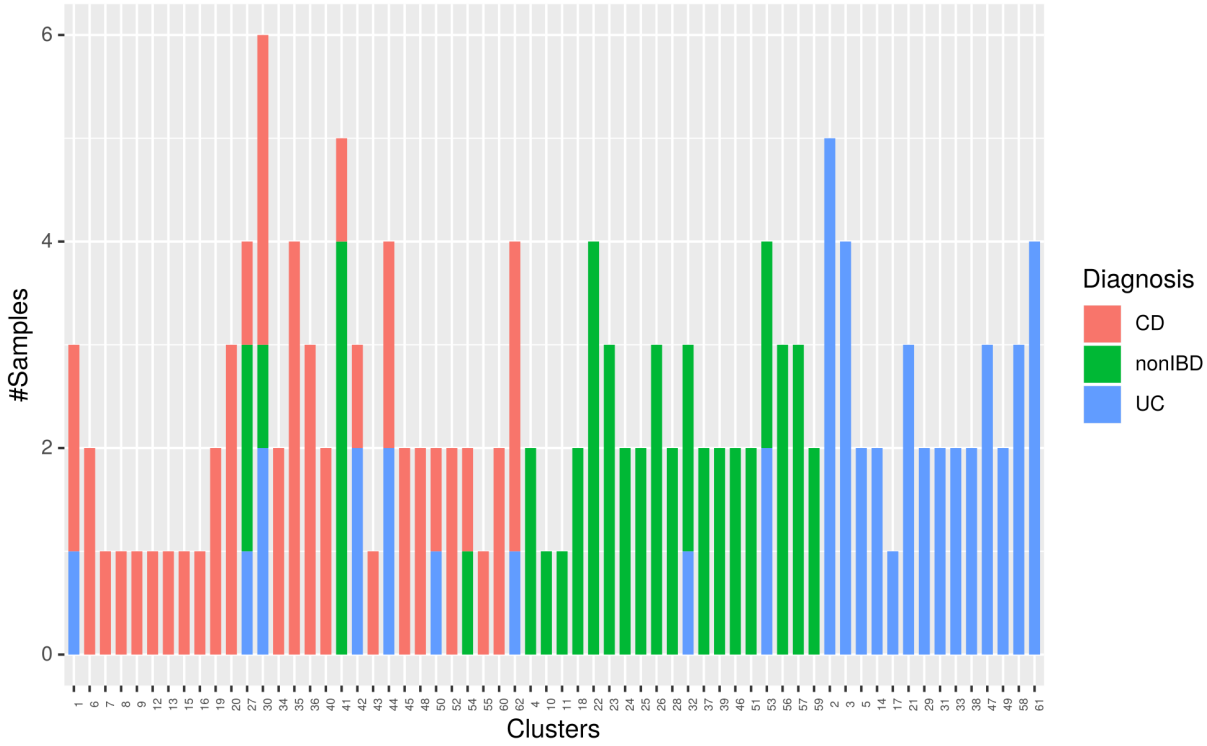

Figure 4.16 **Composition of the MD clusters obtained with the HMP dataset.** For each cluster, the amount of metagenomes is shown, with the IBD (Inflammatory Bowel Disease) diagnosis associated with each metagenome. Diagnoses: non-IBD = healthy; CD = Crohn's Disease; UC = Ulcerative Colitis

approach (Fig. 4.14B&C), which underlines the importance of integrating cross-samples information when binning genomes. However, after de-replication at strain level, we observed that the SACB strategy reconstructed more MAGs than CASB, while the SASB strategy reconstructed more HQ MAGs than the CASB strategy (Fig. 4.14B&D).

We also compared the MAGs quality (completeness and contamination) produced by each assembly-binning strategy. Differences in completeness were only observed between the SACB and CASB strategies, with SACB HQ MAGs being more complete than CASB HQ MAGs (Fig. 4.15A). Here, we also used both checkM (SCG-based) and GUNC (taxonomy-based) complementary approaches to estimate contamination. GUNC was able to detect more subtle differences in contamination between strategies than the checkM algorithm (Fig. 4.15). These observed differences demonstrate that co-binning strategies actually produce less contaminated MAGs than single-binning strategies, at all MAGs quality levels. Overall, these distinct results when de-replicating MAGs at species or strain level suggest that no single strategy can fit all needs. Therefore, the choice of

an assembly-binning strategy should be informed by a biological question and should consider the microbiome complexity under study.

4.3.3 Comparing MAGNETO to similar metagenomics workflows

Finally, we compared the performances of MAGNETO to metagenomics workflows dedicated to MAGs reconstruction, namely METAWRAP [20], ATLAS [22] and nf-core/mag [238]. We chose these three tools as they use similar software to perform assembly and binning, namely MEGAHIT [125] and MetaBAT2 [16]. The comparison of the workflows was performed using the HMP dataset. ATLAS is a workflow only permitting single-assembly of metagenomes, but integrates a binning refinement module using DAStool [27], which constitutes a good opportunity to evaluate whether single-assembly could perform better after binning refinement. METAWRAP also contains a binning refinement module, albeit less complex than the DAStool methodology. This refinement module performs pairwise alignment of MAGs to detect redundant genomes, to then only conserve MAGs showing the best quality amongst detected duplicated MAGs. nf-core/mag uses the exact same tools as our workflow to perform assembly and binning. As compared to ATLAS, we observed that MAGNETO systematically reconstructed more MAGs using any of the four assembly-binning strategies (Table 4.2). However, it also reconstructed less MAGs than METAWRAP. The higher number of MAGs produced by METAWRAP may be explained by its refinement module coupling several binners, as these binners may reconstruct more non-redundant MAGs, thus increasing their numbers. However, MAGNETO and nf/core-mag reconstructed the same number of MAGs for both CASB or CACB strategies. These similar results are most likely explained by the absence of a bins refinement module, and by the fact that in both workflows, the binning step used the exact same parameters.

4.4 Discussion

In this work, we present MAGNETO, a fully automated workflow enabling genome-resolved metagenomics. It implements a novel approach to compute clusters of metagenomes for co-assembly without *a priori* knowledge, as well as complementary assembly-binning strategies to maximize MAGs recovery towards specific goals. MAGNETO also provides key functionalities, from the construction and annotation of gene catalogs, to the gener-

Table 4.2 Number of reconstructed MAGs for the HMP dataset.
Comparison of the number of MAGs reconstructed with different workflows, and different strategies, after dereplication at strains resolution. MAGs: number of dereplicated MAGs; HQ: High Quality (Completeness > 90%, Contamination < 5%), MQ: Medium Quality (Completeness > 50%, Contamination < 10%).

Pipeline	Strategy	MAGs	
		HQ	MQ
ATLAS			
	SASB	253	120
METAWRAP			
	SASB	**302**	**295**
	SACB	**320**	242
	CASB	**377**	**320**
	CACB	**386**	**350**
nf-core/mag			
	CASB	277	261
	CACB	361	300
MAGNETO			
	SASB	280	251
	SACB	311	**286**
	CASB	277	261
	CACB	361	300

ation of genes and genomes abundance matrices.

4.4.1 An unsupervised approach to metagenomic co-assembly

We demonstrated the utility of a non-supervised metagenomic-distance based approach to guide metagenomics co-assembly on a large set of ocean metagenomes. Indeed, clusters of metagenomes identified by the MD-based approach did not overlap with oceanic regions previously used for guiding co-assembly of these metagenomes [5]. As anticipated, this implies that, in the ocean, geographic distances do not necessarily reflect compositional metagenomic distances between microbial communities. This observation can likely be explained by the fact that the composition of marine microbial communities are significantly structured through environmental filtering by key abiotic factors such as temperature [8] and ocean currents influencing species dispersal [268]. This allowed a

clear separation between metagenomes originating from polar regions, mesopelagic zone, and from superficial, temperate water. Interestingly, the MD-based clustering analysis grouped together in a single cluster (cluster #1, see Figure 4.4A, p.100) metagenomes from sampling stations facing upwelling currents. As upwelling regions are influenced by deep ocean currents raising cold nutrient-rich waters to the surface, they can significantly impact species diversity of marine microbial communities towards richer states [269, 270].

The rationale behind our metagenomic distance-based approach to perform co-assembly was to infer which metagenomes should be grouped together in an unsupervised fashion without *a priori* knowledge. The aim was to develop an approach that could guarantee the actual closeness of the metagenomes to co-assemble, thus emphasizing the increase in species-specific reads abundance for the assembler. Although the co-assembly of closely related metagenomes have been shown to erode contigs quality [137, 175], we could show that our approach did not increase fragmentation or misassemblies within contigs (Fig. 4.6, p.94). In fact, our MD approach reconstructs MAGs that are more complete, and less contaminated than the OR approach (Fig. 4.8, p.96). Although both metrics we used to estimate MAGs contamination reported contradictory results, we argue that GUNC [14] likely provides better estimates of contamination as it is based on a much larger set of genes as compared to CheckM [180], which assess contamination solely based on SCGs. As SCGs represent highly-conserved genes across all taxa, co-assembling similar metagenomes may actually increase the probability to assemble or bin core regions of closely related genomes. A higher fragmentation of the genomes was already observed following the co-assembly of metagenomes with closely related strains [175, 271], although it was also shown not to affect completeness nor the contamination of co-assembled genomes [193]. Accessory regions may thus be less affected by co-assembly, although they are also generally more difficult to bin [137].

We observed a very high number of exclusive MAGs between the OR and MD approaches, namely 525 for MD and 575 for OR, representing 31.2% and 33.3% of the MAGs reconstructed by each approach, respectively (Fig. 4.4B, p.92). This result indicates that, even if our approach performs better in terms of reconstructed MAGs quality, it nevertheless does not capture the same information from metagenomes as compared to the OR approach. This is confirmed by the increase in proportion of recruited reads when back-mapping to combined MAGs from both approaches (Fig.4.4C, p.92). Thus, combining the MD approach with a co-assembly based on *a priori* knowledge (when available) may represent a good opportunity to better capture the actual bacterial diversity in metagenomes.

However, the proportion of mapped reads was significantly higher on MD MAGs as compared to OR MAGs when considering only detected MAGs in samples (Fig.4.4D, p.92). Here, we could show that the OR approach reconstructed MAGs recruiting a higher proportion of reads, but that this higher proportion was mainly driven by MAGs displaying a very low horizontal coverage ($< 30\%$), suggesting these MAGs contained relatively small genomic regions recruiting a high proportion of reads. These observations, coupled with the smaller contamination observed in OR MAGs when estimated using SCGs, may imply that the OR approach allows a better reconstruction of core genomic regions, which are shared among a higher proportion of organisms.

Applying the MD-based co-assembly approach on the HMP dataset, we found that the identified clusters of metagenomes mostly corresponded to the IBD pathology affecting the patients (Fig. 4.16, p.106). Indeed, a majority of clusters containing metagenomes from healthy patients did not contain any metagenomes related to IBD (16 out of 23 clusters contain non-IBD metagenomes), and a majority of the clusters containing CD or UC patients are composed of metagenomes associated with only the same type of IBD (26 out of 34 clusters contain IBD metagenomes). This observation emphasizes the relevance of our method, as changes in the composition of the gut microbiota have been associated with IBD diagnosis [11, 272, 273].

4.4.2 A systematic comparison of assembly-binning strategies

When comparing the four different assembly-binning strategies we defined herein, we observed that *co-* strategies systematically reconstructed more MAGs than *single-* strategies. Notably, the CACB strategy was identified as the best performing in terms of number of recovered MAGs, across all (simulated and real) datasets we considered. This may be explained by i) the increase in (rare) reads abundance through the co-assembly, and ii) the higher amount of co-abundance information integrated into the co-binning process [24, 239]. On simulated datasets, co-assembly strategies systematically reconstructed more MAGs after de-replication, while applying single-binning or co-binning. However, this was not the case when analysing the HMP dataset, for which the SACB strategy reconstructed more strain-level MAGs than CASB. This may be due to an uneven distribution of strains across metagenomes. Indeed, human gut microbiomes tend to be personal and usually exhibit higher inter- than intra-individual community variations at strain level [274, 275]. Overall, if gut strains are individual-specific and thus only occur in a low number of metagenomes, co-assembly will be less effective to actually increase

strain-specific reads for improving their assembly. This result actually suggests that an MD-based approach integrating single-nucleotide polymorphism (SNP) information would be useful to improve the reconstruction of strain-level MAGs.

4.4.3 A multi-sample assembly-binning strategy maximizes genomes recovery

We showed that co-assembly approaches usually reconstructed higher numbers of (MQ) MAGs, albeit with a tendency to be more contaminated (HQ MAGs). As previously reported [137], this underlines the utility of co-assembly to recover rare or less-abundant genomes, and to maximise MAGs recovery from a limited number of metagenomes. Here, co-binning strategies (SACB & CACB) systematically reconstructed less contaminated MAGs than single-binning strategies (SASB & CASB) in datasets for which differences in MAGs quality could be detected between strategies. Thus, multi-sample co-abundance information computed across a minimum number of metagenomes appears particularly relevant to improve genome binning and to limit the erroneous grouping of contigs. However, the co-binning strategy may represent a severe limitation as it requires larger computational resources (CPU time and disk space), since it implies performing N^2 reads mapping operations, where N is the total number of metagenomes. For the CAMI dataset, differences in MQ MAGs quality between strategies were in contradiction with analyses of the other datasets, although the HQ MAGs comparison pointed towards similar conclusions as in the other datasets. This may be explained by the different number of MAGs reconstructed between each strategy. The 80 MAGs reconstructed by the SASB strategy may belong to abundant organisms, thus implying a smaller risk to increase contamination. However, as SACB and CASB reconstructed almost twice the number of MAGs compared to SASB, the MQ MAGs recovered by these strategies may belong to less abundant genomes, hence these MAGs may be harder to reconstruct with a few samples ($n = 5$), and thus may be more prone to contamination.

Interestingly, the effect of the co-assembly step on MAGs contamination is unclear. So far, only a few methods, including CheckM and GUNC, exist to estimate MAGs quality. When considering CheckM on the HMP dataset, single-assembly strategies reconstructed less contaminated MAGs than co-assembly strategies. However, when considering contamination estimated by GUNC, co-assembly strategies constructed less contaminated MAGs. These results underline the crucial need to develop more accurate methods to

properly estimate MAGs quality, and also highlight the utility to confront methods using complementary strategies to estimate genome quality.

Co-assembly constitutes a useful and affordable strategy for shallow sequenced metagenomes or when the number of metagenomes to co-assemble is limited. In such cases, the increase in complexity of the assembly is limited, thus removing the main computational limitation of co-assembly. Similar to co-assembly, co-binning is also impacted by metagenomic sequencing depth, as the computation time obviously increases with the number of reads. As demonstrated, the co-binning strategy represents a powerful and useful, though computer-intensive, strategy when numerous samples are available, as it helps to reconstruct more HQ MAGs. A potential perspective for improving the co-binning process would be to identify an optimal number of samples to compute co-abundances in order to optimize its cost-benefit ratio.

Discussion

In this thesis, we wanted to answer to two problems posed by genome-resolved metagenomics, i) the difficulty to reconstruct MAGs at the strains level and/or bacterial pan-genome, and ii) the unresolved question about how to perform co-assembly in MAGs reconstruction. We tried to answer to the first problem by developing a new approach to reconstruct MAGs, centered around the logic programming paradigm. The rationale was that the exploration of all equivalent solutions to the binning problems would help to retrieve relevant information to reconstruct strains genomes, and/or pan-genomes. The second problem was answered with the development of an unsupervised co-assembly approach, computing a nucletotidic distance between metagenomes as a relevant metric to group the metagenomes to co-assemble.

Logic programming for MAGs reconstruction

A first formalisation of contigs binning as a constraint programming problem

The work presented in Chapter 3 was, to the best of our knowledge, the first attempt to formalize the binning step as a logic programming problem. Previous attempts have already used constraints, but in the different framework of constraints clustering, such as [276]. The ambition of the logic programming approach was to explore several putative binning solutions, in order to better identify potential hitherto unreconstructed MAGs. This exploration of a high number of binning solutions are envisioned as a good answer to the limitations of classic binning tools, which can suffer during the reconstruction of genomes from rare organisms, or to reconstruct closely-related strains. The main result from our approach was its capacity to successfully reconstruct two closely-related genomes strains, while this discrimination was completely undetected by metabat2, which merged the strains into a single MAG. The supposedly better separation allowed by our model must be reconsidered, however, through the highly restrained framework of our model. Notably, when the ASP model was not giving any information about the number of

expected MAGs to reconstruct, its performance sharply dropped. Thus, a more precise pre-assessment of the putative number of MAGs to reconstruct would greatly benefit our ASP model, and would allow its better capacity for strains reconstruction to shine. If the determination of bacterial strains present in metagenomes still represents a difficult task in metagenomics, it also constitutes one of its major aims for the near future, with several recent studies focusing on strains determination [15, 17]. The possibility to explore several solutions to the binning problem to identify genomes strains and pan-genomes would then further emphasize the relevance of the logic programming approach to resolve the contigs binning problem.
However, the global efficiency of the ASP binning model to reconstruct MAGs revealed to be outperformed by the tool we compare with, metabat2. The addition of more constraints to our model could increase the precision of the clustering of contigs. Moreover, the capacity to discard some contigs from the dataset also represents a major advantage metabat2 had against our model. The inclusion of contigs dismissal in our model would then help to increase the precision, by removing contigs originating from genomic regions which differs significantly in their nucleotidic composition, and can be difficult to include into a genome bin.

Scalability problem of ASP

The main trouble remained however the too high number of solutions to explore, which exceeded the capacity of the clasp solver. The design of a more complete model, with the addition of more constraints, would help to further reduce the search space for the solver. The other main point from the development of the ASP framework was the significant reduction of computation time allowed by the improvement of the code implementation. There is without doubts still room for further technical enhancement, to fasten the problem resolution. Among the possible upgrades, the use of propagators, which are software written in imperative languages overlaying the ASP program, should be considered. They can handle more easily procedures which could be very costly in the resolution process performed by the ASP solver, easing the resolution of the problem. Their limited use in this work did not allow a significant enhancement of the resolution process. Other approaches, such as the development of constraint programming ASP (CASP), would include features which are more easily performed in full CP solving than in ASP. Notably, the CASP would remove the need to list all possible solutions [18], which would represent a major improvement of the ASP technique itself. The development of such approaches

constitutes however a rather recent discipline, with currently limited applications, but showed promising results [18, 19].

Development of an automated workflow for genome-resolved metagenomics

An unsupervised approach to metagenomic co-assembly

Another contribution of this work was the development of an automatic, unsupervised method to determine sets of metagenome to co-assemble (see Chapter 4). The co-assembly in MAGs reconstruction has been indeed widely used, because of its advantage to increase the abundance of rare variants present in microbial communities. However, the use of co-assembly has revealed to be an "all or nothing" strategy, with many studies routinely co-assembling all the metagenomes in their datasets. Because of the increase in computational resource consumption, the co-assembly of all metagenomes at once may not be possible, especially when the metagenomes come from complex bacterial communities such as marine communities. Some studies thus choose to perform several co-assemblies of subsets of metagenomes of their dataset. This practice however rose the question of how the sets of metagenomes should be chosen. Studies performing co-assembly of subsets of marine metagenomes have to this end used geographic location [5, 10]. The *a priori* knowledge needed to determine sets to co-assemble may not always be available or relevant, constituting the rationale of our work. The sets of metagenomes retrieved from the metagenomic-distance (MD) matrix allowed to gather metagenomes following common environmental characteristics, such as the temperature. This observation emphasized the relevance of our MD approach, as the effect of temperature in shaping bacterial communities has already been shown [8]. This MD approach also reconstructed more MAGs and with higher quality than the approach relying on geographic location. Further clarification is however needed, notably because of the apparent contradiction between different two quality metrics. A possible explanation would be that co-assembling closely-related metagenomes eases the reconstruction of accessory parts of the pan-genome, while at the same time would worsen the reconstruction of core genome parts. The higher part of the strains contamination in the contamination metric relying on single-copy genes that we observed in the MD MAGs compared to the OR MAGs (see Fig. 4.8, p. 96) may represent evidence towards this hypothesis. However, further investigation, with a more complete

analysis of the genes found in the reconstructed MAGs, would be needed.

Comparison of MAGs reconstruction strategies

The comparison of the four strategies of MAGs reconstruction has revealed a strong positive effect of the computation of differential coverage amongst a high number of metagenomes. The effect of differential coverage has been measured as more efficient than the effect of the co-assembly, as single-assembly combined with co-binning could outperform co-assembly strategies. This observation may reflect that the deleterious effects of the co-assembly, namely the higher probability to produce fragmented MAGs, may only be overcome with a differential coverage computed on a high-enough number of metagenomes. This concerns essentially complex communities, as these observations were made on MAGs reconstructed from human gut microbiome datasets, which can contain several strains and variants.
The computation of differential coverage on all the available metagenomes may however not be necessary, and revealed to be costly, because of the quadratic number of operations to perform. An enhancement of this approach could be an assessment of an optimal number of samples on which to compute the differential coverage. Such an assessment has already been made in the past [24], but it would be interesting to know if that optimal number would depend on the complexity of the considered bacterial community.

Future improvements of MAGNETO

The implementation of MAGNETO allows the addition of more modules, some of which are already considered, in order to perform a more complete analysis of the MAGs. Notably, a module estimating the optimal growth of the MAGs based on their nucleotidic composition has been developed and had already contributed to the study of arctic MAGS [25]. Further additions would concern assembly software and binning tools, to increase the flexibility of the user. Besides, the binning protocol combining several tools to retrieve mosaic MAGs has exhibited promising results [26, 27], and might constitute an easy way to improve the efficiency of binning. Our workflow follows a MAGs reconstruction protocol that may be characterised as classic, summoning tools which have been widely used in recent metagenomic studies. In that respect, it suffers from the limitations of

these tools, and the hindrance with reconstruction of accessory regions of the pan-genome was left unanswered. In recent years, several studies aimed a better characterisation of pan-genomes in metagenomic studies [28–30]. Those works would be inspiring to allow a more complete analysis of bacterial communities in MAGNETO.

ANNEXES

ASP Model: second implementation, known number of bins

```
#const k=10.
#const lengthmin=200000.
#const rate=95.
%there are 10 marker genes
%so we want at least 5 marker genes within each cluster
#const threshold_completeness=5.
%%there are 10 marker genes, so we want a number of duplicates strictly below 2
#const threshold_conta=2.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%   Cluster attribution   %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%defining the bins
bin(1..k).

%assignation rule : a contig belongs to one bin and only one
1{att(B,C) : bin(B)}1 :- contig(B,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ANTIPARALLELIC CONSTRAINT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% First, look for maximum contig id in bin B%%
aux_att(B,C) :- att(B,C).
aux_att(B,C-1) :- aux_att(B,C), C>0.
```

```
attmax(B,C) :- aux_att(B,C), not aux_att(B,C+1).
%% Looking for minimum contig id in bin B %%
attmin(B,C) :- att(B,C-1), contig(C,_).
%use Cmax previously computed as a limit for attmin/2 predicate.
attmin(B,C+1) :- attmin(B,C), attmax(B,Cmax), C<Cmax.
attmin(B,C) :- attmin(B,C), not attmin(B,C-1).
:- attmin(B1,C1), attmin(B2,C2), B1<B2, C1>C2.

%%last, if cluster id B1 is smaller than cluster id B2,
%%then min contig id C1 must not be higher than min contig id C2
:- attmin(B1,C1), attmin(B2,C2), B1<B2, C1>C2.

%%%%%%%%%%%%%%%
% Cluster size %
%%%%%%%%%%%%%%%

%control cluster size using number of bases
%calculate total length in bins
attlengthmin(B,C,L) :- att(B,C), contig(C,L).
%bin must contain a certain amount of bases
:- bin(B), not #sumL : attlengthmin(B,_,L)>lengthmin.

%%%%%%%%%%%%%%%%%%%
%   Optimization   %
%%%%%%%%%%%%%%%%%%%

%% compute maximum diameter
diam(B,D) :- att(B,C1), att(B,C2), C1<C2, bin(B), tnf(C1,C2,D).
diam(B,D-1) :- diam(B,D), D>0.
maxDiameter(B,D) :- diam(B,D), not diam(B,D+1).

%% compute minimum margin
margin(M) :- att(B1,C1), att(B2,C2), bin(B1), bin(B2), B1<B2,
abd(C1,C2,M).
```

```
margin(M+1) :- margin(M), M<100.
minMargin(M) :- margin(M), not margin(M-1), M>-1.

%optimization
:~ maxDiameter(D). [1,D,d] %minimization of diameter
:~ minmargin(M). [-1,M,m] %maximization of margin

%%%%%%%%%%%%%%%%%%%

%addition of instance-level constraints (if needed)

%%%%%%%%%%%%%%%
% Completeness %
%%%%%%%%%%%%%%%

%completeness constraint : MAGs are binss in which there is more than 50% of marker
scg_bin(C,G) :- att(B,C), scg(C,G).
%there are 10 marker genes right now, so each bin needs 5 marker genes
:- bin(B), not #countG : scg_bin(B,G)>threshold_completeness.

%%%%%%%%%%%%%%%%
% Contamination %
%%%%%%%%%%%%%%%%
%%gene marker constraint : two contigs must not be in the same cluster
%%if they carry two copies of the same gene marker
:- att(C1,X1), att(C2,X2), mg(X1, M1), mg(X2, M2), X1!=X2, C1==C2, M1==M2.

%%contamination constraint : MAGs are clusters in which
%%there is less than 10% of contamination
scg_bin_duplicate(B,G1) :- att(B,C1), att(B,C2), scg(C1,G1), scg(C2,G1), C1<C2.
%%there are 10 marker genes, so we want a number of duplicates strictly below 2
:- bin(B), #countG : scg_bin_duplicate(B,G)>threshold_contamination.

%%Unused constraint : to belong to the same cluster, tnf distance must be below thr
```

```
%:- att(C1, X1), att(C2, X2), tnf(X1,X2,T), X1!=X2, C1==C2, T>70.

%%Unused constraint : if two contigs do not have a distance between them (distance too h
%%do not put them in the same cluster.
%:- att(C, X1), att(C, X2), not tnf(X1,X2,_), X1!=X2.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Show these predicates in answer %%%%
%#show tnf/3.
%#showdistancediff/3.
#show att/2.
%#show minmargin/1.
%#show maxDiameter/1.
#show cluster/1.
%#show mg_cluster/2.
%#show attlengthmin/3.
%#show margin/1.
%#show diam/1.
%#show mg/2.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ASP Model: unknown number of bins, 25 bins

```
#const k=25.
#const lengthmin=200000.

%there are 10 marker genes
%so we want at least 5 marker genes within each cluster
#const threshold_completeness=5.
%%there are 10 marker genes, so we want a number of duplicates strictly below 2
#const threshold_conta=2.
%#const thresholdtnf=30.

%define a set of putative bins
putative_bin(1..k).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%   Cluster attribution   %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% a contig C belong to one bin B and only one
1{att(B,C) : putative_bin(B)}1 :- contig(C,_).
%if there is at least one instance of predicate att/2 containing B
%so B is a true bin
bin(B) :- att(B,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ANTIPARALLELIC CONSTRAINT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% First, look for maximum contig id in bin B%%
aux_att(B,C) :- att(B,C).
aux_att(B,C-1) :- aux_att(B,C), C>0.
attmax(B,C) :- aux_att(B,C), not aux_att(B,C+1).
%% Looking for minimum contig id in bin B %%
attmin(B,C) :- att(B,C-1), contig(C,_).
%use Cmax previously computed as a limit for attmin/2 predicate.
```

```
attmin(B,C+1) :- attmin(B,C), attmax(B,Cmax), C<Cmax.
attmin(B,C) :- attmin(B,C), not attmin(B,C-1).
:- attmin(B1,C1), attmin(B2,C2), B1<B2, C1>C2.

%%last, if cluster id B1 is smaller than cluster id B2,
%%then min contig id C1 must not be higher than min contig id C2
:- attmin(B1,C1), attmin(B2,C2), B1<B2, C1>C2.

%%%%%%%%%%%%%%%
% Cluster size %
%%%%%%%%%%%%%%%

%control cluster size using number of bases
%calculate total length in bins
attlengthmin(B,C,L) :- att(B,C), contig(C,L).
%bin must contain a certain amount of bases
:- bin(B), not #sumL : attlengthmin(B,_,L)>lengthmin.

%%%%%%%%%%%%%%%%%%%
%   Optimization   %
%%%%%%%%%%%%%%%%%%%

%% compute maximum diameter
diam(B,D) :- att(B,C1), att(B,C2), C1<C2, bin(B), tnf(C1,C2,D).
diam(B,D-1) :- diam(B,D), D>0.
maxDiameter(B,D) :- diam(B,D), not diam(B,D+1).

%% compute minimum margin
margin(M) :- att(B1,C1), att(B2,C2), bin(B1), bin(B2), B1<B2,
abd(C1,C2,M).
margin(M+1) :- margin(M), M<100.
minMargin(M) :- margin(M), not margin(M-1), M>-1.

%optimization
```

```
:~ maxDiameter(D). [1,D,d] %minimization of diameter
:~ minmargin(M). [-1,M,m] %maximization of margin
%maximize number of points in clusters
%did not find way to implement it in other way
%#maximize{1 @1,C,X : att(C,X)}.

%%%%%%%%%%%%%%%%%%

%addition of instance-level constraints (if needed)

%%%%%%%%%%%%%%%
% Completeness %
%%%%%%%%%%%%%%%

%completeness constraint : MAGs are binss in which there is more than 50% of marker
scg_bin(C,G) :- att(B,C), scg(C,G).
%there are 10 marker genes right now, so each bin needs 5 marker genes
:- bin(B), not #countG : scg_bin(B,G)>threshold_completeness.

%%%%%%%%%%%%%%%%
% Contamination %
%%%%%%%%%%%%%%%%
%%gene marker constraint : two contigs must not be in the same cluster
%%if they carry two copies of the same gene marker
:- att(C1,X1), att(C2,X2), mg(X1, M1), mg(X2, M2), X1!=X2, C1==C2, M1==M2.

%%contamination constraint : MAGs are clusters in which
%%there is less than 10% of contamination
scg_bin_duplicate(B,G1) :- att(B,C1), att(B,C2), scg(C1,G1), scg(C2,G1), C1<C2.
%%there are 10 marker genes, so we want a number of duplicates strictly below 2
:- bin(B), #countG : scg_bin_duplicate(B,G)>threshold_contamination.

%%Unused constraint : to belong to the same cluster, tnf distance must be below thr
%:- att(C1, X1), att(C2, X2), tnf(X1,X2,T), X1!=X2, C1==C2, T>70.
```

```
%%Unused constraint : if two contigs do not have a distance between them (distance too h
%%do not put them in the same cluster.
%:- att(C, X1), att(C, X2), not tnf(X1,X2,_), X1!=X2.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Show these predicates in answer %%%%
%#show tnf/3.
%#showdistancediff/3.
#show att/2.
%#show minmargin/1.
%#show maxDiameter/1.
#show cluster/1.
%#show mg_cluster/2.
%#show attlengthmin/3.
%#show margin/1.
%#show diam/1.
%#show mg/2.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

BIBLIOGRAPHY

1. Fouhy, F., Stanton, C., Cotter, P. D., Hill, C. & Walsh, F., Proteomics as the final step in the functional metagenomics study of antimicrobial resistance, *Frontiers in Microbiology* **6,** ISSN: 1664-302X, https://www.frontiersin.org/articles/10.3389/fmicb.2015.00172 (2022) (2015).
2. Chistoserdova, L., Functional Metagenomics: Recent Advances and Future Challenges, en, *Biotechnology and Genetic Engineering Reviews* **26,** 335–352, ISSN: 0264-8725, 2046-5556 (Jan. 2009).
3. Sanger, F., Nicklen, S. & Coulson, A. R., DNA sequencing with chain-terminating inhibitors, en, *Proceedings of the National Academy of Sciences* **74,** 5463–5467, ISSN: 0027-8424, 1091-6490 (Dec. 1977).
4. Handelsman, J., Rondon, M. R., Brady, S. F., Clardy, J. & Goodman, R. M., Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products, en, *Chemistry & Biology* **5,** R245–R249, ISSN: 10745521 (Oct. 1998).
5. Delmont, T. O. *et al.,* Nitrogen-fixing populations of Planctomycetes and Proteobacteria are abundant in surface ocean metagenomes, en, *Nature Microbiology* **3,** 804–813, ISSN: 2058-5276 (July 2018).
6. Zilber-Rosenberg, I. & Rosenberg, E., Role of microorganisms in the evolution of animals and plants: the hologenome theory of evolution, eng, *FEMS microbiology reviews* **32,** 723–735, ISSN: 0168-6445 (Aug. 2008).
7. Robbins, R. J., Krishtalka, L. & Wooley, J. C., Advances in biodiversity: metagenomics and the unveiling of biological dark matter, en, *Standards in Genomic Sciences* **11,** 69, ISSN: 1944-3277 (Dec. 2016).
8. Sunagawa, S. *et al.,* Structure and function of the global ocean microbiome, *Science* **348,** _eprint: https://www.science.org/doi/pdf/10.1126/science.1261359, 1261359 (2015).

9. Parks, D. H. *et al.,* Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life, en, *Nature Microbiology* **2,** 1533–1542, ISSN: 2058-5276 (Nov. 2017).

10. Tully, B. J., Graham, E. D. & Heidelberg, J. F., The reconstruction of 2,631 draft metagenome-assembled genomes from the global oceans, en, *Scientific Data* **5,** 170203, ISSN: 2052-4463 (Dec. 2018).

11. Lloyd-Price, J. *et al.,* Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases, *Nature* **569,** Publisher: Nature Publishing Group, 655–662 (2019).

12. Hug, L. A. *et al.,* A new view of the tree of life, *Nature microbiology* **1,** Publisher: Nature Publishing Group, 1–6 (2016).

13. Nayfach, S. *et al.,* A genomic catalog of Earth's microbiomes, *Nature biotechnology* **39,** Publisher: Nature Publishing Group, 499–509 (2021).

14. Orakov, A. *et al.,* GUNC: detection of chimerism and contamination in prokaryotic genomes, *Genome biology* **22,** Publisher: Springer, 1–19 (2021).

15. Almeida, A. *et al.,* A unified catalog of 204,938 reference genomes from the human gut microbiome, *Nature biotechnology* **39,** Publisher: Nature Publishing Group, 105–114 (2021).

16. Kang, D. D. *et al.,* MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies, en, *PeerJ* **7,** Publisher: PeerJ Inc., e7359, ISSN: 2167-8359 (July 2019).

17. Da Silva, K. *et al.,* StrainFLAIR: strain-level profiling of metagenomic samples using variation graphs, en, *PeerJ* **9,** Publisher: PeerJ Inc., e11884, ISSN: 2167-8359 (Aug. 2021).

18. Arias, J., Carro, M., Salazar, E., Marple, K. & Gupta, G., *Constraint Answer Set Programming without Grounding* arXiv:1804.11162 [cs], May 2018, doi:10.48550/arXiv.1804.11162, http://arxiv.org/abs/1804.11162 (2022).

19. Lierler, Y., Constraint Answer Set Programming, en, 5 (2012).

20. Uritskiy, G. V., DiRuggiero, J. & Taylor, J., MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis, en, *Microbiome* **6,** ISSN: 2049-2618, doi:10.1186/s40168-018-0541-1, https://microbiomejournal.biomedcentral.com/articles/10.1186/s40168-018-0541-1 (2019) (Dec. 2018).

21. Clarke, E. L. *et al.,* Sunbeam: an extensible pipeline for analyzing metagenomic sequencing experiments, *Microbiome* **7,** Publisher: Springer, 46 (2019).

22. Kieser, S., Brown, J., Zdobnov, E. M., Trajkovski, M. & McCue, L. A., ATLAS: a Snakemake workflow for assembly, annotation, and genomic binning of metagenome sequence data, *BMC bioinformatics* **21,** Publisher: Springer, 1–8 (2020).

23. Krakau, S. *et al.,* nf-core/mag, doi:10.5281/zenodo.4529420, https://zenodo.org/record/4529420#.YKWAgeuxV7g (2021).

24. Alneberg, J. *et al.,* Binning metagenomic contigs by coverage and composition, en, *Nature Methods* **11,** 1144–1146, ISSN: 1548-7091, 1548-7105 (Nov. 2014).

25. Royo-Llonch, M. *et al.,* Compendium of 530 metagenome-assembled bacterial and archaeal genomes from the polar Arctic Ocean, en, *Nature Microbiology* **6,** Number: 12 Publisher: Nature Publishing Group, 1561–1574, ISSN: 2058-5276 (Dec. 2021).

26. Uritskiy, G. V., DiRuggiero, J. & Taylor, J., *MetaWRAP - a flexible pipeline for genome-resolved metagenomic data analysis* en, preprint (Microbiology, Mar. 2018), doi:10.1101/277442, http://biorxiv.org/lookup/doi/10.1101/277442 (2022).

27. Sieber, C. M. K. *et al.,* Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy, en, *Nature Microbiology* **3,** 836–843, ISSN: 2058-5276 (July 2018).

28. Delmont, T. O. & Eren, A. M., Linking pangenomes and metagenomes: the Prochlorococcus metapangenome, *PeerJ* **6,** e4320, ISSN: 2167-8359 (Jan. 2018).

29. Plaza Oñate, F. *et al.,* MSPminer: abundance-based reconstitution of microbial pangenomes from shotgun metagenomic data, *Bioinformatics* **35,** Publisher: Oxford University Press, 1544–1552 (2019).

30. Ma, B., France, M. & Ravel, J., eng, *in The Pangenome: Diversity, Dynamics and Evolution of Genomes* (eds Tettelin, H. & Medini, D.) (Springer, Cham (CH), 2020), ISBN: 978-3-030-38280-3 978-3-030-38281-0, http://www.ncbi.nlm.nih.gov/books/NBK558817/ (2022).

31. Churcheward, B., Millet, M., Bihouée, A., Fertin, G. & Chaffron, S., MAGNETO: An Automated Workflow for Genome-Resolved Metagenomics, *mSystems* **7,** Publisher: American Society for Microbiology, e00432–22 (June 2022).

32. Razumov, A., The direct method of calculation of bacteria in water: comparison with the Koch method, *Mikrobiologija* **1,** 131–146 (1932).

33. Staley, J. T. & Konopka, A., MEASUREMENT OF IN SITU ACTIVITIES OF NONPHOTOSYNTHETIC MICROORGANISMS IN AQUATIC AND TERRESTRIAL HABITATS, en, *Annual Review of Microbiology* **39,** 321–346, ISSN: 0066-4227, 1545-3251 (Oct. 1985).

34. Woese, C. R. & Fox, G. E., Phylogenetic structure of the prokaryotic domain: The primary kingdoms, en, *Proceedings of the National Academy of Sciences* **74,** Publisher: National Academy of Sciences Section: PNAS Classic Article, 5088–5090, ISSN: 0027-8424, 1091-6490 (Nov. 1977).

35. Tringe, S. G. & Hugenholtz, P., A renaissance for the pioneering 16S rRNA gene, en, *Current Opinion in Microbiology* **11,** 442–446, ISSN: 13695274 (Oct. 2008).

36. Britschgi, T. B. & Giovannoni, S. J., Phylogenetic analysis of a natural marine bacterioplankton population by rRNA gene cloning and sequencing, en, *Applied and Environmental Microbiology* **57,** 1707–1713, ISSN: 0099-2240, 1098-5336 (June 1991).

37. Schmidt, T. M., DeLong, E. F. & Pace, N. R., Analysis of a marine picoplankton community by 16S rRNA gene cloning and sequencing, EN, *Journal of Bacteriology,* doi:10.1128/jb.173.14.4371-4378.1991, https://journals.asm.org/doi/abs/10.1128/jb.173.14.4371-4378.1991 (2022) (July 1991).

38. DeLong, E. F., Archaea in coastal marine environments. en, *Proceedings of the National Academy of Sciences* **89,** 5685–5689, ISSN: 0027-8424, 1091-6490 (June 1992).

39. Ward, D. M., Weller, R. & Bateson, M. M., 16S rRNA sequences reveal uncultured inhabitants of a well-studied thermal community, *FEMS Microbiology Reviews* **6,** 105–115, ISSN: 0168-6445 (June 1990).

40. Ward, D. M., Weller, R. & Bateson, M. M., 16S rRNA sequences reveal numerous uncultured microorganisms in a natural community, en, *Nature* **345,** Bandiera_abtest: a Cg_type: Nature Research Journals Number: 6270 Primary_atype: Research Publisher: Nature Publishing Group, 63–65, ISSN: 1476-4687 (May 1990).

41. Stackebrandt, E., Liesack, W. & Goebel, B. M., Bacterial diversity in a soil sample from a subtropical Australian environment as determined by 16S rDNA analysis. en, *The FASEB Journal* **7,** 232–236, ISSN: 0892-6638, 1530-6860 (Jan. 1993).

42. Giovannoni, S. J., Britschgi, T. B., Moyer, C. L. & Field, K. G., Genetic diversity in Sargasso Sea bacterioplankton, en, *Nature* **345,** Number: 6270 Publisher: Nature Publishing Group, 60–63, ISSN: 1476-4687 (May 1990).

43. Tremblay, J. *et al.,* Primer and platform effects on 16S rRNA tag sequencing, en, *Frontiers in Microbiology* **6,** ISSN: 1664-302X, doi:10.3389/fmicb.2015.00771, http://journal.frontiersin.org/Article/10.3389/fmicb.2015.00771/abstract (2021) (Aug. 2015).

44. Shizuya, H. *et al.,* Cloning and stable maintenance of 300-kilobase-pair fragments of human DNA in Escherichia coli using an F-factor-based vector. en, *Proceedings of the National Academy of Sciences* **89,** 8794–8797, ISSN: 0027-8424, 1091-6490 (Sept. 1992).

45. Shizuya, H. & Kouros-Mehr, H., The development and applications of the bacterial chromosome cloning system, en, *Keio J Med,* 5 (2001).

46. Stein, J. L., Marsh, T. L., Wu, K. Y., Shizuya, H. & DeLong, E. F., Characterization of uncultivated prokaryotes: isolation and analysis of a 40-kilobase-pair genome fragment from a planktonic marine archaeon, EN, *Journal of Bacteriology,* doi:10.1128/jb.178.3.591-599.1996, https://journals.asm.org/doi/abs/10.1128/jb.178.3.591-599.1996 (2022) (Feb. 1996).

47. Tyson, G. W. *et al.,* Community structure and metabolism through reconstruction of microbial genomes from the environment, en, *Nature* **428,** 37–43, ISSN: 0028-0836, 1476-4687 (Mar. 2004).

48. Venter, J. C. *et al.,* Environmental Genome Shotgun Sequencing of the Sargasso Sea, en, *Science* **304,** 66–74, ISSN: 0036-8075, 1095-9203 (Apr. 2004).

49. Schloss, P. D. & Handelsman, J., Biotechnological prospects from metagenomics, en, *Current Opinion in Biotechnology* **14,** 303–310, ISSN: 0958-1669 (June 2003).

50. Ateto, A., *Bioinformatics for Beginners, Genes, Genome, Molecular Evolution, Databases and Analytical Tools.* ISBN: 978-0-12-410471-6 (Jan. 2014).

51. Schuster, S. C., Next-generation sequencing transforms today's biology, en, *Nature Methods* **5,** Number: 1 Publisher: Nature Publishing Group, 16–18, ISSN: 1548-7105 (Jan. 2008).

52. Shendure, J. *et al.,* DNA sequencing at 40: past, present and future, en, *Nature* **550,** 345–353, ISSN: 0028-0836, 1476-4687 (Oct. 2017).

53. Metzker, M. L., Sequencing technologies — the next generation, en, *Nature Reviews Genetics* **11,** Number: 1 Publisher: Nature Publishing Group, 31–46, ISSN: 1471-0064 (Jan. 2010).

54. Wu, D. *et al.,* A phylogeny-driven genomic encyclopaedia of Bacteria and Archaea, en, *Nature* **462,** Number: 7276 Publisher: Nature Publishing Group, 1056–1060, ISSN: 1476-4687 (Dec. 2009).

55. Rinke, C. *et al.,* Insights into the phylogeny and coding potential of microbial dark matter, en, *Nature* **499,** Number: 7459 Publisher: Nature Publishing Group, 431–437, ISSN: 1476-4687 (July 2013).

56. Whitman, W. B., Coleman, D. C. & Wiebe, W. J., Prokaryotes: The unseen majority, en, *Proceedings of the National Academy of Sciences* **95,** 6578–6583, ISSN: 0027-8424, 1091-6490 (June 1998).

57. Simmons, S. L. *et al.,* Population Genomic Analysis of Strain Variation in Leptospirillum Group II Bacteria Involved in Acid Mine Drainage Formation, en, *PLoS Biology* **6** (ed Eisen, J. A.) e177, ISSN: 1545-7885 (July 2008).

58. Sharp, C. E. *et al.,* Humboldt's spa: microbial diversity is controlled by temperature in geothermal environments, en, *The ISME Journal* **8,** 1166–1174, ISSN: 1751-7362, 1751-7370 (June 2014).

59. Walter, J. & Ley, R., The Human Gut Microbiome: Ecology and Recent Evolutionary Changes, en, *Annual Review of Microbiology* **65,** 411–429, ISSN: 0066-4227, 1545-3251 (Oct. 2011).

60. Philippot, L., Raaijmakers, J. M., Lemanceau, P. & van der Putten, W. H., Going back to the roots: the microbial ecology of the rhizosphere, en, *Nature Reviews Microbiology* **11,** 789–799, ISSN: 1740-1526, 1740-1534 (Nov. 2013).

61. Sunagawa, S. *et al.,* Metagenomic species profiling using universal phylogenetic marker genes, en, *Nature Methods* **10,** 1196–1199, ISSN: 1548-7091, 1548-7105 (Dec. 2013).

62. Tully, B. J., *Assembly Procedure Applied to TARA Oceans Data (Ex. North Pacific)* 2017, https://www.protocols.io/view/assembly-procedure-applied-to-tara-oceans-data-ex-hfqb3mw?.

63. Turnbaugh, P. J. *et al.,* A core gut microbiome in obese and lean twins, *Nature* **457,** 480–484, ISSN: 0028-0836 (Jan. 2009).

64. Qin, J. *et al.,* A metagenome-wide association study of gut microbiota in type 2 diabetes, eng, *Nature* **490,** 55–60, ISSN: 1476-4687 (Oct. 2012).

65. Subramanian, S. *et al.,* Persistent Gut Microbiota Immaturity in Malnourished Bangladeshi Children, *Nature* **510,** 417–421, ISSN: 0028-0836 (June 2014).

66. Pop, M. *et al.,* Diarrhea in young children from low-income countries leads to large-scale alterations in intestinal microbiota composition, *Genome Biology* **15,** R76, ISSN: 1465-6906 (2014).

67. Ramezani, A. & Raj, D. S., The Gut Microbiome, Kidney Disease, and Targeted Interventions, *Journal of the American Society of Nephrology : JASN* **25,** 657–670, ISSN: 1046-6673 (Apr. 2014).

68. Karlsson, F. H. *et al.,* Symptomatic atherosclerosis is associated with an altered gut metagenome, *Nature Communications* **3,** 1245, ISSN: 2041-1723 (Dec. 2012).

69. Hsiao, E. Y. *et al.,* The microbiota modulates gut physiology and behavioral abnormalities associated with autism, *Cell* **155,** 1451–1463, ISSN: 0092-8674 (Dec. 2013).

70. Francis, C. A., Roberts, K. J., Beman, J. M., Santoro, A. E. & Oakley, B. B., Ubiquity and diversity of ammonia-oxidizing archaea in water columns and sediments of the ocean, *Proceedings of the National Academy of Sciences of the United States of America* **102,** 14683–14688, ISSN: 0027-8424 (Oct. 2005).

71. Sullivan, M. B., Coleman, M. L., Weigele, P., Rohwer, F. & Chisholm, S. W., Three Prochlorococcus Cyanophage Genomes: Signature Features and Ecological Interpretations, en, *PLOS Biology* **3,** Publisher: Public Library of Science, e144, ISSN: 1545-7885 (Apr. 2005).

72. Baas Becking, L. G. M., *Geobiologie of inleiding tot de milieukunde* **18-19** (WP Van Stockum & Zoon, 1934).

73. Fondi, M. *et al.,* "Every Gene Is Everywhere but the Environment Selects": Global Geolocalization of Gene Sharing in Environmental Samples through Network Analysis, *Genome Biology and Evolution* **8,** 1388–1400, ISSN: 1759-6653 (Apr. 2016).

74. Richter, D. J. *et al.,* Genomic evidence for global ocean plankton biogeography shaped by large-scale current systems, *eLife* **11,** e78129, ISSN: 2050-084X (2022).

75. Medini, D., Donati, C., Tettelin, H., Masignani, V. & Rappuoli, R., The microbial pan-genome, en, *Current Opinion in Genetics & Development, Genomes and evolution* **15,** 589–594, ISSN: 0959-437X (Dec. 2005).

76. Mira, A., Martín-Cuadrado, A. B., D'Auria, G. & Rodríguez-Valera, F., The bacterial pan-genome: a new paradigm in microbiology, en, *International Microbiology* **13,** Number: 2, 45–57, ISSN: 1618-1905 (Sept. 2010).

77. Welch, R. A. *et al.,* Extensive mosaic structure revealed by the complete genome sequence of uropathogenic Escherichia coli, *Proceedings of the National Academy of Sciences* **99,** Publisher: Proceedings of the National Academy of Sciences, 17020–17024 (Dec. 2002).

78. Lukjancenko, O., Wassenaar, T. M. & Ussery, D. W., Comparison of 61 Sequenced Escherichia coli Genomes, *Microbial Ecology* **60,** 708–720, ISSN: 0095-3628 (2010).

79. Li, J. *et al.,* An integrated catalog of reference genes in the human gut microbiome, en, *Nature Biotechnology* **32,** Number: 8 Publisher: Nature Publishing Group, 834–841, ISSN: 1546-1696 (Aug. 2014).

80. Nayfach, S., Rodriguez-Mueller, B., Garud, N. & Pollard, K. S., An integrated metagenomics pipeline for strain profiling reveals novel patterns of bacterial transmission and biogeography, *Genome Research* **26,** 1612–1625, ISSN: 1088-9051 (Nov. 2016).

81. Scholz, M. *et al.,* Strain-level microbial epidemiology and population genomics from shotgun metagenomics, en, *Nature Methods* **13,** Number: 5 Publisher: Nature Publishing Group, 435–438, ISSN: 1548-7105 (May 2016).

82. Cohan, F. M., What are Bacterial Species?, en, *Annual Review of Microbiology* **56,** 457–487, ISSN: 0066-4227, 1545-3251 (Oct. 2002).

83. Fraser, C., Alm, E. J., Polz, M. F., Spratt, B. G. & Hanage, W. P., The Bacterial Species Challenge: Making Sense of Genetic and Ecological Diversity, *Science* **323,** Publisher: American Association for the Advancement of Science, 741–746 (Feb. 2009).

84. Ciufo, S. *et al.,* Using average nucleotide identity to improve taxonomic assignments in prokaryotic genomes at the NCBI, *International Journal of Systematic and Evolutionary Microbiology* **68,** Publisher: Microbiology Society, 2386–2392, ISSN: 1466-5034, (2018).

85. Olm, M. R. *et al.,* Consistent metagenome-derived metrics verify and delineate bacterial species boundaries, *Msystems* **5,** Publisher: Am Soc Microbiol, e00731–19 (2020).

86. Konstantinidis, K. T. & Tiedje, J. M., Genomic insights that advance the species definition for prokaryotes, *Proceedings of the National Academy of Sciences* **102,** Publisher: Proceedings of the National Academy of Sciences, 2567–2572 (Feb. 2005).

87. Jain, C., Rodriguez-R, L. M., Phillippy, A. M., Konstantinidis, K. T. & Aluru, S., High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries, en, *Nature Communications* **9,** Number: 1 Publisher: Nature Publishing Group, 5114, ISSN: 2041-1723 (Nov. 2018).

88. DeLong, E. F., Life on the Thermodynamic Edge, *Science* **317,** Publisher: American Association for the Advancement of Science, 327–328 (July 2007).

89. Black, A. J., Bourrat, P. & Rainey, P. B., Ecological scaffolding and the evolution of individuality, en, *Nature Ecology & Evolution* **4,** Number: 3 Publisher: Nature Publishing Group, 426–436, ISSN: 2397-334X (Mar. 2020).

90. Bernard, G., Pathmanathan, J. S., Lannes, R., Lopez, P. & Bapteste, E., Microbial Dark Matter Investigations: How Microbial Studies Transform Biological Knowledge and Empirically Sketch a Logic of Scientific Discovery, *Genome Biology and Evolution* **10,** 707–715, ISSN: 1759-6653 (Mar. 2018).

91. Giovannoni, S. J. *et al.,* Genome streamlining in a cosmopolitan oceanic bacterium, eng, *Science (New York, N.Y.)* **309,** 1242–1245, ISSN: 1095-9203 (Aug. 2005).

92. Pande, S. *et al.,* Fitness and stability of obligate cross-feeding interactions that emerge upon gene loss in bacteria, en, *The ISME Journal* **8,** Number: 5 Publisher: Nature Publishing Group, 953–962, ISSN: 1751-7370 (May 2014).

93. Morris, J. J., Lenski, R. E. & Zinser, E. R., The Black Queen Hypothesis: Evolution of Dependencies through Adaptive Gene Loss, en, *mBio* **3,** e00036–12, ISSN: 2161-2129, 2150-7511 (May 2012).

94. Mas, A., Jamshidi, S., Lagadeuc, Y., Eveillard, D. & Vandenkoornhuyse, P., Beyond the Black Queen Hypothesis, en, *The ISME Journal* **10,** Number: 9 Publisher: Nature Publishing Group, 2085–2091, ISSN: 1751-7370 (Sept. 2016).

95. Guerrero, R., Margulis, L. & Berlanga, M., Symbiogenesis: the holobiont as a unit of evolution, eng, *International Microbiology: The Official Journal of the Spanish Society for Microbiology* **16,** 133–143, ISSN: 1139-6709 (Sept. 2013).

96. Handelsman, J., Metagenomics: Application of Genomics to Uncultured Microorganisms, *Microbiology and Molecular Biology Reviews* **68,** 669–685, ISSN: 1092-2172 (Dec. 2004).

97. Mullany, P., Functional metagenomics for the investigation of antibiotic resistance, *Virulence* **5,** Publisher: Taylor & Francis _eprint: https://doi.org/10.4161/viru.28196, 443–447, ISSN: 2150-5594 (Apr. 2014).

98. Ferrer, M., Golyshina, O., Beloqui, A. & Golyshin, P. N., Mining enzymes from extreme environments, en, *Current Opinion in Microbiology, Ecology and Industrial Microbiology / RNA Techniques* **10,** 207–214, ISSN: 1369-5274 (June 2007).

99. Curtis, T. P., Sloan, W. T. & Scannell, J. W., Estimating prokaryotic diversity and its limits, *Proceedings of the National Academy of Sciences of the United States of America* **99,** 10494–10499, ISSN: 0027-8424 (Aug. 2002).

100. Mardanov, A. V., Kadnikov, V. V. & Ravin, N. V., en, *in Metagenomics* (ed Nagarajan, M.) 1–13 (Academic Press, Jan. 2018), ISBN: 978-0-08-102268-9, doi:10.1016/B978-0-08-102268-9.00001-X, https://www.sciencedirect.com/science/article/pii/B978008102268900001X (2022).

101. Mirete, S., Morgante, V. & González-Pastor, J. E., Functional metagenomics of extreme environments, en, *Current Opinion in Biotechnology, Energy biotechnology • Environmental biotechnology* **38,** 143–149, ISSN: 0958-1669 (Apr. 2016).

102. Moran, M. A., Metatranscriptomics: Eavesdropping on Complex Microbial Communities, en, *Microbe Magazine* **4,** 329–335, ISSN: 1558-7452, 1558-7460 (July 2009).

103. Korry, B. J., Cabral, D. J. & Belenky, P., Metatranscriptomics Reveals Antibiotic-Induced Resistance Gene Expression in the Murine Gut Microbiota, *Frontiers in Microbiology* **11,** ISSN: 1664-302X, https://www.frontiersin.org/articles/10.3389/fmicb.2020.00322 (2022) (2020).

104. Bashiardes, S., Zilberman-Schapira, G. & Elinav, E., Use of Metatranscriptomics in Microbiome Research, en, *Bioinformatics and Biology Insights* **10,** Publisher: SAGE Publications Ltd STM, BBI.S34610, ISSN: 1177-9322 (Jan. 2016).

105. Wilmes, P. & Bond, P. L., Metaproteomics: studying functional gene expression in microbial ecosystems, en, *Trends in Microbiology* **14,** 92–97, ISSN: 0966-842X (Feb. 2006).

106. Javdan, B. *et al.,* Personalized Mapping of Drug Metabolism by the Human Gut Microbiome, en, *Cell* **181,** 1661–1679.e22, ISSN: 0092-8674 (June 2020).

107. Breitling, R., What is systems biology?, *Frontiers in Physiology* **1,** ISSN: 1664-042X, https://www.frontiersin.org/articles/10.3389/fphys.2010.00009 (2022) (2010).

108. Pinu, F. R. *et al.,* Systems Biology and Multi-Omics Integration: Viewpoints from the Metabolomics Research Community, en, *Metabolites* **9,** Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, 76, ISSN: 2218-1989 (Apr. 2019).

109. Albertsen, M. *et al.,* Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes, en, *Nature Biotechnology* **31,** 533–538, ISSN: 1087-0156, 1546-1696 (June 2013).

110. Pasolli, E. *et al.,* Extensive unexplored human microbiome diversity revealed by over 150,000 genomes from metagenomes spanning age, geography, and lifestyle, *Cell* **176,** 649–662 (2019).

111. Barnum, T. P. *et al.,* Genome-resolved metagenomics identifies genetic mobility, metabolic interactions, and unexpected diversity in perchlorate-reducing communities, en, *The ISME Journal* **12,** Number: 6 Publisher: Nature Publishing Group, 1568–1581, ISSN: 1751-7370 (June 2018).

112. Quince, C. *et al.,* DESMAN: a new tool for de novo extraction of strains from metagenomes, en, *Genome Biology* **18,** ISSN: 1474-760X, doi:10.1186/s13059-017-1309-9, http://genomebiology.biomedcentral.com/articles/10.1186/s13059-017-1309-9 (2019) (Dec. 2017).

113. Wang, W.-L. *et al.,* Application of metagenomics in the human gut microbiome, *World Journal of Gastroenterology : WJG* **21,** 803–814, ISSN: 1007-9327 (Jan. 2015).

114. Grossart, H.-P., Massana, R., McMahon, K. D. & Walsh, D. A., Linking metagenomics to aquatic microbial ecology and biogeochemical cycles, en, *Limnology and Oceanography* **65,** _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/lno.11382, S2–S20, ISSN: 1939-5590 (2020).

115. Hasin, Y., Seldin, M. & Lusis, A., Multi-omics approaches to disease, *Genome Biology* **18,** 83, ISSN: 1474-7596 (May 2017).

116. Ghurye, J. S., Metagenomic Assembly: Overview, Challenges and Applications, en, 10 (2016).

117. Ayling, M., Clark, M. D. & Leggett, R. M., New approaches for metagenome assembly with short reads, en, *Briefings in Bioinformatics* **21,** 584–594, ISSN: 1467-5463, 1477-4054 (Mar. 2020).

118. Myers, E. W. *et al.,* A Whole-Genome Assembly of *Drosophila*, en, *Science* **287,** 2196–2204, ISSN: 0036-8075, 1095-9203 (Mar. 2000).

119. Lai, B., Ding, R., Li, Y., Duan, L. & Zhu, H., A de novo metagenomic assembly program for shotgun DNA reads, *Bioinformatics* **28,** 1455–1462, ISSN: 1367-4803 (June 2012).

120. Haider, B. *et al.,* Omega: an Overlap-graph de novo Assembler for Metagenomics, *Bioinformatics* **30,** 2717–2722, ISSN: 1367-4803 (Oct. 2014).

121. Pevzner, P. A., Tang, H. & Waterman, M. S., An Eulerian path approach to DNA fragment assembly, *Proceedings of the National Academy of Sciences* **98,** Publisher: Proceedings of the National Academy of Sciences, 9748–9753 (Aug. 2001).

122. Ferragina, P., Luccio, F., Manzini, G. & Muthukrishnan, S., Compressing and indexing labeled trees, with applications, *Journal of the ACM* **57,** 4:1–4:33, ISSN: 0004-5411 (Nov. 2009).

123. Bowe, A., Onodera, T., Sadakane, K. & Shibuya, T., *Succinct de Bruijn Graphs* en, *in Algorithms in Bioinformatics* (eds Raphael, B. & Tang, J.) (Springer, Berlin, Heidelberg, 2012), 225–235, ISBN: 978-3-642-33122-0, doi:10.1007/978-3-642-33122-0_18.

124. Pfeiffer, F. *et al.,* Systematic evaluation of error rates and causes in short samples in next-generation sequencing, en, *Scientific Reports* **8,** Bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Data processing;Next-generation sequencing Subject_term_id: data-processing;next-generation-sequencing, 10950, ISSN: 2045-2322 (July 2018).

125. Li, D., Liu, C.-M., Luo, R., Sadakane, K. & Lam, T.-W., MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph, *Bioinformatics* **31,** 1674–1676 (2015).

126. Nurk, S., Meleshko, D., Korobeynikov, A. & Pevzner, P. A., metaSPAdes: a new versatile metagenomic assembler, en, *Genome Research* **27,** 824–834, ISSN: 1088-9051, 1549-5469 (May 2017).

127. Pell, J. *et al.,* Scaling metagenome sequence assembly with probabilistic de Bruijn graphs, *Proceedings of the National Academy of Sciences* **109,** Publisher: Proceedings of the National Academy of Sciences, 13272–13277 (Aug. 2012).

128. Chikhi, R. & Rizk, G., *Space-Efficient and Exact de Bruijn Graph Representation Based on a Bloom Filter* en, *in Algorithms in Bioinformatics* (eds Raphael, B. & Tang, J.) (Springer, Berlin, Heidelberg, 2012), 236–248, ISBN: 978-3-642-33122-0, doi:10.1007/978-3-642-33122-0_19.

129. Chikhi, R. & Medvedev, P., Informed and automated k-mer size selection for genome assembly, en, *Bioinformatics* **30,** 31–37, ISSN: 1367-4803, 1460-2059 (Jan. 2014).

130. Namiki, T., Hachiya, T., Tanaka, H. & Sakakibara, Y., MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads, *Nucleic Acids Research* **40,** e155, ISSN: 0305-1048 (Nov. 2012).

131. Afiahayati, Sato, K. & Sakakibara, Y., MetaVelvet-SL: an extension of the Velvet assembler to a de novo metagenomic assembler utilizing supervised learning, *DNA Research: An International Journal for Rapid Publication of Reports on Genes and Genomes* **22,** 69–77, ISSN: 1340-2838 (Feb. 2015).

132. Georganas, E. *et al., HipMer: an extreme-scale de novo genome assembler in SC '15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* ISSN: 2167-4337 (Nov. 2015), 1–11, doi:10.1145/2807591.2807664.

133. Georganas, E. *et al.,* Extreme Scale De Novo Metagenome Assembly, en, *arXiv:1809.07014 [cs, q-bio],* arXiv: 1809.07014, http://arxiv.org/abs/1809.07014 (2020) (Sept. 2018).

134. Peng, Y., Leung, H. C. M., Yiu, S. M. & Chin, F. Y. L., IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth, *Bioinformatics* **28,** 1420–1428, ISSN: 1367-4803 (June 2012).

135. Boisvert, S., Raymond, F., Godzaridis, É., Laviolette, F. & Corbeil, J., Ray Meta: scalable de novo metagenome assembly and profiling, en, *Genome Biology* **13,** R122, ISSN: 1465-6906 (2012).

136. Iqbal, Z., Caccamo, M., Turner, I., Flicek, P. & McVean, G., De novo assembly and genotyping of variants using colored de Bruijn graphs, *Nature genetics* **44,** 226–232, ISSN: 1061-4036 (Jan. 2012).

137. Coleman, I. & Korem, T., Embracing Metagenomic Complexity with a Genome-Free Approach, *mSystems* **6,** Publisher: American Society for Microbiology, e00816–21 (Aug. 2021).

138. Johnson, M. *et al.,* Evaluating Methods for Isolating Total RNA and Predicting the Success of Sequencing Phylogenetically Diverse Plant Transcriptomes, *PloS one* **7,** e50226 (Nov. 2012).

139. Wang, Y., Hu, H. & Li, X., MBBC: an efficient approach for metagenomic binning based on clustering, *BMC Bioinformatics* **16,** 36, ISSN: 1471-2105 (Feb. 2015).

140. Girotto, S., Pizzi, C. & Comin, M., MetaProb: accurate metagenomic reads binning based on probabilistic sequence signatures, *Bioinformatics* **32,** i567–i575, ISSN: 1367-4803 (Sept. 2016).

141. Kang, D. D., Froula, J., Egan, R. & Wang, Z., MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities, en, *PeerJ* **3,** e1165, ISSN: 2167-8359 (Aug. 2015).

142. Lin, H.-H. & Liao, Y.-C., Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes, en, *Scientific Reports* **6,** Number: 1 Publisher: Nature Publishing Group, 1–8, ISSN: 2045-2322 (Apr. 2016).

143. Herath, D., Tang, S.-L., Tandon, K., Ackland, D. & Halgamuge, S. K., CoMet: a workflow using contig coverage and composition for binning a metagenomic sample with high precision, en, *BMC Bioinformatics* **18,** ISSN: 1471-2105, doi:10.1186/s12859-017-1967-3, https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1967-3 (2019) (Dec. 2017).

144. Wu, Y.-W., Tang, Y.-H., Tringe, S. G., Simmons, B. A. & Singer, S. W., MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm, en, *Microbiome* **2,** 26, ISSN: 2049-2618 (Dec. 2014).

145. Nielsen, H. B. *et al.,* Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes, en, *Nature Biotechnology* **32,** Number: 8 Publisher: Nature Publishing Group, 822–828, ISSN: 1546-1696 (Aug. 2014).

146. Kang, D., *MetaBat2 ReadMe* 2019, https://bitbucket.org/berkeleylab/metabat/src/master/README.md.

147. Wu, Y.-W., Simmons, B. A. & Singer, S. W., MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets, *Bioinformatics* **32,** 605–607 (2015).

148. Muralidharan, H. S., Shah, N., Meisel, J. S. & Pop, M., Binnacle: Using Scaffolds to Improve the Contiguity and Quality of Metagenomic Bins, *Frontiers in Microbiology* **12,** ISSN: 1664-302X, https://www.frontiersin.org/articles/10.3389/fmicb.2021.638561 (2022) (2021).

149. McHardy, A. C., Martín, H. G., Tsirigos, A., Hugenholtz, P. & Rigoutsos, I., Accurate phylogenetic classification of variable-length DNA fragments, en, *Nature Methods* **4,** Number: 1 Publisher: Nature Publishing Group, 63–72, ISSN: 1548-7105 (Jan. 2007).

150. Huson, D. H., Auch, A. F., Qi, J. & Schuster, S. C., MEGAN analysis of metagenomic data, en, *Genome Research* **17,** Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab, 377–386, ISSN: 1088-9051, 1549-5469 (Mar. 2007).

151. Krause, L. *et al.,* Phylogenetic classification of short environmental DNA fragments, *Nucleic Acids Research* **36,** 2230–2239, ISSN: 0305-1048 (Apr. 2008).

152. Wu, M. & Eisen, J. A., A simple, fast, and accurate method of phylogenomic inference, *Genome Biology* **9,** R151, ISSN: 1474-760X (Oct. 2008).

153. Brady, A. & Salzberg, S. L., Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models, en, *Nature Methods* **6,** Number: 9 Publisher: Nature Publishing Group, 673–676, ISSN: 1548-7105 (Sept. 2009).

154. Parks, D. H., MacDonald, N. J. & Beiko, R. G., Classifying short genomic fragments from novel lineages using composition and homology, *BMC Bioinformatics* **12,** 328, ISSN: 1471-2105 (Aug. 2011).

155. MacDonald, N. J., Parks, D. H. & Beiko, R. G., Rapid identification of high-confidence taxonomic assignments for metagenomic data, *Nucleic Acids Research* **40,** e111, ISSN: 0305-1048 (Aug. 2012).

156. Wang, Z., Wang, Z., Lu, Y. Y., Sun, F. & Zhu, S., SolidBin: improving metagenome binning with semi-supervised normalized cut, en, *Bioinformatics* (ed Hancock, J.) ISSN: 1367-4803, 1460-2059, doi:10.1093/bioinformatics/btz253, https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btz253/5448864 (2019) (Apr. 2019).

157. Sangwan, N., Xia, F. & Gilbert, J. A., Recovering complete and draft population genomes from metagenome datasets, en, *Microbiome* **4,** ISSN: 2049-2618, doi:10.1186/s40168-016-0154-5, http://www.microbiomejournal.com/content/4/1/8 (2019) (Dec. 2016).

158. Kayani, M. u. R., Huang, W., Feng, R. & Chen, L., Genome-resolved metagenomics using environmental and clinical samples, *Briefings in Bioinformatics* **22,** bbab030, ISSN: 1477-4054 (Sept. 2021).

159. Hildebrand, F., Meyer, A. & Eyre-Walker, A., Evidence of Selection upon Genomic GC-Content in Bacteria, en, *PLOS Genetics* **6,** Publisher: Public Library of Science, e1001107, ISSN: 1553-7404 (Sept. 2010).

160. Teeling, H., Meyerdierks, A., Bauer, M., Amann, R. & Glöckner, F. O., Application of tetranucleotide frequencies for the assignment of genomic fragments, en, *Environmental Microbiology* **6,** _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1462-2920.2004.00624.x, 938–947, ISSN: 1462-2920 (2004).

161. Iverson, V. *et al.,* Untangling Genomes from Metagenomes: Revealing an Uncultured Class of Marine Euryarchaeota, en, *Science* **335,** 587–590, ISSN: 0036-8075, 1095-9203 (Feb. 2012).

162. Lander, E. S. & Waterman, M. S., Genomic mapping by fingerprinting random clones: A mathematical analysis, en, *Genomics* **2,** 231–239, ISSN: 0888-7543 (Apr. 1988).

163. Sharon, I. *et al.,* Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization, en, *Genome Research* **23,** Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab, 111–120, ISSN: 1088-9051, 1549-5469 (Jan. 2013).

164. Imelfort, M. *et al.,* GroopM: an automated tool for the recovery of population genomes from related metagenomes, en, *PeerJ* **2,** e603, ISSN: 2167-8359 (Sept. 2014).

165. Ultsch, A. & Morchen, F., ESOM-Maps: tools for clustering, visualization, and classi cation with Emergent SOM, en, 7 (2005).

166. Hess, M. *et al.,* Metagenomic Discovery of Biomass-Degrading Genes and Genomes from Cow Rumen, *Science* **331,** Publisher: American Association for the Advancement of Science, 463–467 (Jan. 2011).

167. Mackelprang, R. *et al.,* Metagenomic analysis of a permafrost microbial community reveals a rapid response to thaw, en, *Nature* **480,** Number: 7377 Publisher: Nature Publishing Group, 368–371, ISSN: 1476-4687 (Dec. 2011).

168. Do, C. B. & Batzoglou, S., What is the expectation maximization algorithm?, en, *Nature Biotechnology* **26,** Number: 8 Publisher: Nature Publishing Group, 897–899, ISSN: 1546-1696 (Aug. 2008).

169. Yu, G., Jiang, Y., Wang, J., Zhang, H. & Luo, H., BMC3C: binning metagenomic contigs using codon usage, sequence composition and read coverage, en, *Bioinformatics* (ed Berger, B.) ISSN: 1367-4803, 1460-2059, doi:10.1093/bioinformatics/bty519, https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/bty519/5045915 (2019) (June 2018).

170. Lu, Y. Y., Chen, T., Fuhrman, J. A. & Sun, F., COCACOLA: binning metagenomic contigs using sequence COmposition, read CoverAge, CO-alignment and paired-end read LinkAge, en, *Bioinformatics,* btw290, ISSN: 1367-4803, 1460-2059 (June 2016).

171. Herath, D., Tang, S.-L., Tandon, K., Ackland, D. & Halgamuge, S. K., CoMet: a workflow using contig coverage and composition for binning a metagenomic sample with high precision, en, *BMC Bioinformatics* **18,** ISSN: 1471-2105, doi:10.1186/s12859-017-1967-3, https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1967-3 (2019) (Dec. 2017).

172. Pan, S., Zhu, C., Zhao, X.-M. & Coelho, L. P., A deep siamese neural network improves metagenome-assembled genomes in microbiome datasets across different environments, en, *Nature Communications* **13,** Number: 1 Publisher: Nature Publishing Group, 2326, ISSN: 2041-1723 (Apr. 2022).

173. Nissen, J. N. *et al.,* Binning microbial genomes using deep learning, en, *bioRxiv,* doi:10.1101/490078, http://biorxiv.org/lookup/doi/10.1101/490078 (2019) (2018).

174. Qin, J. *et al.,* A human gut microbial gene catalogue established by metagenomic sequencing, en, *Nature* **464,** Number: 7285 Publisher: Nature Publishing Group, 59–65, ISSN: 1476-4687 (Mar. 2010).

175. Sczyrba, A. *et al.,* Critical assessment of metagenome interpretation—a benchmark of metagenomics software, *Nature methods* **14,** 1063 (2017).

176. Xue, H., Mallawaarachchi, V., Zhang, Y., Rajan, V. & Lin, Y., *RepBin: Constraint-based Graph Representation Learning for Metagenomic Binning* tech. rep. arXiv:2112.11696, arXiv:2112.11696 [cs, q-bio] type: article (arXiv, Dec. 2021), doi:10.48550/arXiv.2112.11696, http://arxiv.org/abs/2112.11696 (2022).

177. Meyer, F. *et al.,* AMBER: Assessment of Metagenome BinnERs, *GigaScience* **7,** giy069, ISSN: 2047-217X (June 2018).

178. Yue, Y. *et al.,* Evaluating metagenomics tools for genome binning with real metagenomic datasets and CAMI datasets, *BMC Bioinformatics* **21,** 334, ISSN: 1471-2105 (July 2020).

179. Bowers, R. M. *et al.,* Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea, en, *Nature Biotechnology* **35,** 725–731, ISSN: 1087-0156, 1546-1696 (Aug. 2017).

180. Parks, D. H., Imelfort, M., Skennerton, C. T., Hugenholtz, P. & Tyson, G. W., CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes, *Genome research* **25,** 1043–1055 (2015).

181. Swan, B. K. *et al.,* Prevalent genome streamlining and latitudinal divergence of planktonic bacteria in the surface ocean, *Proceedings of the National Academy of Sciences* **110,** Publisher: Proceedings of the National Academy of Sciences, 11463–11468 (July 2013).

182. Eren, A. M. *et al.,* Anvi'o: an advanced analysis and visualization platform for 'omics data, *PeerJ* **3,** Publisher: PeerJ Inc., e1319 (2015).

183. Simão, F. A., Waterhouse, R. M., Ioannidis, P., Kriventseva, E. V. & Zdobnov, E. M., BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs, *Bioinformatics* **31,** 3210–3212, ISSN: 1367-4803 (Oct. 2015).

184. Laczny, C. C. *et al.,* VizBin - an application for reference-independent visualization and human-augmented binning of metagenomic data, en, *Microbiome* **3,** 1, ISSN: 2049-2618 (Jan. 2015).

185. Chen, L.-X., Anantharaman, K., Shaiber, A., Eren, A. M. & Banfield, J. F., Accurate and complete genomes from metagenomes, *Genome research* **30,** Publisher: Cold Spring Harbor Lab, 315–333 (2020).

186. Kang, Y. *et al.,* Flexibility and Symmetry of Prokaryotic Genome Rearrangement Reveal Lineage-Associated Core-Gene-Defined Genome Organizational Frameworks, *mBio* **5,** Publisher: American Society for Microbiology, e01867–14 (Dec. 2014).

187. Evans, J. T. & Denef, V. J., To Dereplicate or Not To Dereplicate?, *mSphere* **5,** Publisher: American Society for Microbiology, e00971–19 (June 2020).

188. Richter, M. & Rosselló-Móra, R., Shifting the genomic gold standard for the prokaryotic species definition, en, *Proceedings of the National Academy of Sciences* **106,** 19126–19131, ISSN: 0027-8424, 1091-6490 (Nov. 2009).

189. Marçais, G. *et al.,* MUMmer4: A fast and versatile genome alignment system, en, *PLOS Computational Biology* **14,** Publisher: Public Library of Science, e1005944, ISSN: 1553-7358 (Jan. 2018).

190. Pritchard, L., H. Glover, R., Humphris, S., G. Elphinstone, J. & K. Toth, I., Genomics and taxonomy in diagnostics for food security: soft-rotting enterobacterial plant pathogens, en, *Analytical Methods* **8,** Publisher: Royal Society of Chemistry, 12–24 (2016).

191. Varghese, N. J. *et al.,* Microbial species delineation using whole genome sequences, *Nucleic acids research* **43,** Publisher: Oxford University Press, 6761–6771 (2015).

192. Ondov, B. D. *et al.,* Mash: fast genome and metagenome distance estimation using MinHash, en, *Genome Biology* **17,** ISSN: 1474-760X, doi:10.1186/s13059-016-0997-x, http://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0997-x (2019) (Dec. 2016).

193. Olm, M. R., Brown, C. T., Brooks, B. & Banfield, J. F., dRep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication, *The ISME journal* **11,** 2864 (2017).

194. Song, W.-Z. & Thomas, T., Binning_refiner: improving genome bins through the combination of different binning programs, *Bioinformatics* **33,** 1873–1875, ISSN: 1367-4803 (June 2017).

195. Becraft, E. D. *et al.,* Rokubacteria: Genomic Giants among the Uncultured Bacterial Phyla, *Frontiers in Microbiology* **8,** ISSN: 1664-302X, https://www.frontiersin.org/article/10.3389/fmicb.2017.02264 (2022) (2017).

196. Garg, S. G. *et al.,* Anomalous Phylogenetic Behavior of Ribosomal Proteins in Metagenome-Assembled Asgard Archaea, *Genome Biology and Evolution* **13,** evaa238, ISSN: 1759-6653 (Jan. 2021).

197. Mallawaarachchi, V., Wickramarachchi, A. & Lin, Y., GraphBin: refined binning of metagenomic contigs using assembly graphs, *Bioinformatics* **36,** 3307–3313, ISSN: 1367-4803 (June 2020).

198. Tettelin, H. *et al.,* Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: Implications for the microbial "pan-genome", *Proceedings of the National Academy of Sciences* **102,** Publisher: Proceedings of the National Academy of Sciences, 13950–13955 (Sept. 2005).

199. Garrison, E. *et al.,* Variation graph toolkit improves read mapping by representing genetic variation in the reference, en, *Nature Biotechnology* **36,** Number: 9 Publisher: Nature Publishing Group, 875–879, ISSN: 1546-1696 (Oct. 2018).

200. Bockmayr, A. & Hooker, J. N., en, *in Handbooks in Operations Research and Management Science* (eds Aardal, K., Nemhauser, G. L. & Weismantel, R.) 559–600 (Elsevier, Jan. 2005), doi:10.1016/S0927-0507(05)12010-6, https://www.sciencedirect.com/science/article/pii/S0927050705120106 (2022).

201. Wagstaff, K. & Cardie, C., Clustering with Instance-Level Constraints, en, 1 (2000).

202. Hoos, H. H. & Stützle, T., *Stochastic local search: Foundations and applications* (Elsevier, 2004).

203. Davis, M., Logemann, G. & Loveland, D., A machine program for theorem-proving, *Communications of the ACM* **5,** 394–397, ISSN: 0001-0782 (July 1962).

204. Barták, R., Constraint Propagation and Backtracking-based Search, en, 43 (2005).

205. Brailsford, S. C., Potts, C. N. & Smith, B. M., Constraint satisfaction problems: Algorithms and applications, en, *European Journal of Operational Research* **119,** 557–581, ISSN: 0377-2217 (Dec. 1999).

206. Van Emden, M. H. & Kowalski, R. A., The Semantics of Predicate Logic as a Programming Language, en, *Journal of the ACM* **23,** 733–742, ISSN: 0004-5411, 1557-735X (Oct. 1976).

207. Enderton, H. B., *A mathematical introduction to logic* 2nd ed, en, ISBN: 978-0-12-238452-3 (Harcourt/Academic Press, San Diego, 2001).

208. Colmerauer, A. & Roussel, P., *in History of programming languages—II* 331–367 (Association for Computing Machinery, New York, NY, USA, Jan. 1996), ISBN: 978-0-201-89502-5, https://doi.org/10.1145/234286.1057820 (2022).

209. LIFSCHITZ, V., *What is answer set programming? in Proc. 23rd AAAI Conf. on Artificial Intelligence, 2008* (2008), 1594–1597.

210. Gelfond, M. & Lifschitz, V., *The stable model semantics for logic programming. ICSLP, 1988* 1988.

211. Moore, R. C., Semantical considerations on nonmonotonic logic, en, *Artificial Intelligence* **25,** 75–94, ISSN: 0004-3702 (Jan. 1985).

212. Clark, K. L., en, *in Logic and Data Bases* (eds Gallaire, H. & Minker, J.) 293–322 (Springer US, Boston, MA, 1978), ISBN: 978-1-4684-3384-5, doi:10.1007/978-1-4684-3384-5_11, https://doi.org/10.1007/978-1-4684-3384-5_11 (2022).

213. Ferraris, P., Lifschitz, V. & Todorova, Y. M., Mathematical Foundations of Answer Set Programming, en, 126 (2005).

214. Syrjänen, T., Lparse 1.0 user's manual, Publisher: Citeseer (2000).

215. Niemelä, I. & Simons, P., *Smodels — an implementation of the stable model and well-founded semantics for normal logic programs* en, *in Logic Programming And Nonmonotonic Reasoning* (eds Dix, J., Furbach, U. & Nerode, A.) (Springer, Berlin, Heidelberg, 1997), 420–429, ISBN: 978-3-540-69249-2, doi:10.1007/3-540-63255-7_32.

216. Erdem, E. & Patoglu, V., Applications of ASP in Robotics, en, *KI - Künstliche Intelligenz* **32,** 143–149, ISSN: 1610-1987 (Aug. 2018).

217. Le, T., Nguyen, H., Pontelli, E. & Son, T. C., *ASP at Work: An ASP Implementation of PhyloWS in Technical Communications of the 28th International Conference on Logic Programming (ICLP'12)* (eds Dovier, A. & Costa, V. S.) **17,** ISSN: 1868-8969 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2012), 359–369, ISBN: 978-3-939897-43-9, doi:10.4230/LIPIcs.ICLP.2012.359, http://drops.dagstuhl.de/opus/volltexte/2012/3636 (2022).

218. Gebser, M. *et al.,* Repair and Prediction (under Inconsistency) in Large Biological Networks with Answer Set Programming, en, 11 (2011).

219. Dovier, A., Formisano, A. & Pontelli, E., An empirical study of constraint logic programming and answer set programming solutions of combinatorial problems, *Journal of Experimental & Theoretical Artificial Intelligence* **21,** Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/09528130701538174, 79–121, ISSN: 0952-813X (June 2009).

220. Ricca, F. *et al.,* Team-building with answer set programming in the Gioia-Tauro seaport, en, *Theory and Practice of Logic Programming* **12,** Publisher: Cambridge University Press, 361–381, ISSN: 1475-3081, 1471-0684 (May 2012).

221. Erdem, E., Gelfond, M. & Leone, N., Applications of Answer Set Programming, en, *AI Magazine* **37,** 53–68, ISSN: 2371-9621, 0738-4602 (Oct. 2016).

222. Gebser, M., Kaminski, R., Kaufmann, B. & Schaub, T., Clingo = ASP + Control: Preliminary Report, *CoRR* **https://arxiv.org/abs/1405.3694** (2014).

223. Gebser, M., Schaub, T. & Thiele, S., en, *in Logic Programming and Nonmonotonic Reasoning* Series Title: Lecture Notes in Computer Science, 266–271 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007), ISBN: 978-3-540-72199-4 978-3-540-72200-7, doi:10.1007/978-3-540-72200-7_24, http://link.springer.com/10.1007/978-3-540-72200-7_24 (2022).

224. Gebser, M., Kaufmann, B., Neumann, A. & Schaub, T., *clasp: A Conflict-Driven Answer Set Solver | SpringerLink* 2007, https://link.springer.com/chapter/10.1007/978-3-540-72200-7_23 (2022).

225. Coppin, B., *Artificial Intelligence Illuminated* Computer ed. edition, English, ISBN: 978-0-7637-3230-1 (Jones & Bartlett Learning, Boston, Apr. 2004).

226. Rosen, K. H., *Handbook of Discrete and Combinatorial Mathematics* en, ISBN: 978-1-58488-781-2 (CRC Press, Oct. 2017).

227. Genesereth, M. & Kao, E., *Herbrand semantics* 2015.

228. Marques Silva, J. & Sakallah, K., *GRASP-A new search algorithm for satisfiability in Proceedings of International Conference on Computer Aided Design* (Nov. 1996), 220–227, doi:10.1109/ICCAD.1996.569607.

229. Gebser, M., Kaminski, R., Kaufmann, B., Romero, J. & Schaub, T., en, *in Logic Programming and Nonmonotonic Reasoning* Series Title: Lecture Notes in Computer Science, 368–383 (Springer International Publishing, Cham, 2015), ISBN: 978-3-319-23263-8 978-3-319-23264-5, doi:10.1007/978-3-319-23264-5_31, http://link.springer.com/10.1007/978-3-319-23264-5_31 (2022).

230. Dao, T.-B.-H., Duong, K.-C. & Vrain, C., Constrained clustering by constraint programming, en, *Artificial Intelligence, Combining Constraint Solving with Mining and Learning* **244,** 70–94, ISSN: 0004-3702 (Mar. 2017).

231. Wrighton, K. C. *et al.,* Fermentation, Hydrogen, and Sulfur Metabolism in Multiple Uncultivated Bacterial Phyla, en, *Science* **337,** 1661–1665, ISSN: 0036-8075, 1095-9203 (Sept. 2012).

232. Milanese, A. *et al.,* Microbial abundance, activity and population genomic profiling with mOTUs2, en, *Nature Communications* **10,** ISSN: 2041-1723, doi:10.1038/s41467-019-08844-4, http://www.nature.com/articles/s41467-019-08844-4 (2019) (Dec. 2019).

233. Kaeberlein, T., Lewis, K. & Epstein, S. S., Isolating "uncultivable" microorganisms in pure culture in a simulated natural environment, eng, *Science (New York, N.Y.)* **296,** 1127–1129, ISSN: 1095-9203 (May 2002).

234. Garza, D. R. & Dutilh, B. E., From cultured to uncultured genome sequences: metagenomics and modeling microbial ecosystems, *Cellular and Molecular Life Sciences* **72,** 4287–4308, ISSN: 1420-682X (2015).

235. Nelson, W. C., Tully, B. J. & Mobberley, J. M., Biases in genome reconstruction from metagenomic data, *PeerJ* **8,** Publisher: PeerJ Inc., e10119 (2020).

236. Duarte, C. M. *et al.,* Sequencing effort dictates gene discovery in marine microbial metagenomes, *Environmental Microbiology* **22,** 4589–4603, ISSN: 1462-2912 (Nov. 2020).

237. Graham, E. D., Heidelberg, J. F. & Tully, B. J., BinSanity: unsupervised clustering of environmental microbial assemblies using coverage and affinity propagation, en, *PeerJ* **5,** e3035, ISSN: 2167-8359 (Mar. 2017).

238. Krakau, S., Straub, D., Gourlé, H., Gabernet, G. & Nahnsen, S., nf-core/mag: a best-practice pipeline for metagenome hybrid assembly and binning, en, 8.

239. Paoli, L. *et al.,* Uncharted biosynthetic potential of the ocean microbiome, *bioRxiv,* Publisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2021/03 doi:10.1101/2021.03.24.436479, https://www.biorxiv.org/content/early/2021/03/24/2021.03.24.436479 (2021).

240. Köster, J. & Rahmann, S., Snakemake—a scalable bioinformatics workflow engine, *Bioinformatics* **28,** Publisher: Oxford University Press, 2520–2522 (2012).

241. Chen, S., Zhou, Y., Chen, Y. & Gu, J., fastp: an ultra-fast all-in-one FASTQ preprocessor, *Bioinformatics* **34,** _eprint: https://academic.oup.com/bioinformatics/article-pdf/34/17/i884/25702346/bty560.pdf, i884–i890, ISSN: 1367-4803 (Sept. 2018).

242. Wingett, S. W. & Andrews, S., FastQ Screen: A tool for multi-genome mapping and quality control, *F1000Research* **7,** Publisher: Faculty of 1000 Ltd (2018).

243. Benoit, G. *et al.,* Multiple comparative metagenomics using multiset k -mer counting, en, *PeerJ Computer Science* **2,** e94, ISSN: 2376-5992 (Nov. 2016).

244. Steinegger, M. & Söding, J., Clustering huge protein sequence sets in linear time, en, *Nature Communications* **9,** ISSN: 2041-1723, doi:10.1038/s41467-018-04964-5, http://www.nature.com/articles/s41467-018-04964-5 (2019) (Dec. 2018).

245. Chaumeil, P.-A., Mussig, A. J., Hugenholtz, P. & Parks, D. H., *GTDB-tk: a toolkit to classify genomes with the Genome Taxonomy Database* 2020.

246. Huerta-Cepas, J. *et al.,* eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses, *Nucleic acids research* **47,** Publisher: Oxford University Press, D309–D314 (2019).

247. Forouzan, E., Shariati, P., Maleki, M. S. M., Karkhane, A. A. & Yakhchali, B., Practical evaluation of 11 de novo assemblers in metagenome assembly, *Journal of microbiological methods* **151,** Publisher: Elsevier, 99–105 (2018).

248. Vollmers, J., Wiegand, S. & Kaster, A.-K., Comparing and Evaluating Metagenome Assembly Tools from a Microbiologist's Perspective - Not Only Size Matters!, en, *PLOS ONE* **12** (ed Rodriguez-Valera, F.) e0169662, ISSN: 1932-6203 (Jan. 2017).

249. Ward Jr, J. H., Hierarchical grouping to optimize an objective function, *Journal of the American statistical association* **58,** Publisher: Taylor & Francis, 236–244 (1963).

250. Rousseeuw, P. J., Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of computational and applied mathematics* **20,** 53–65 (1987).

251. Langmead, B. & Salzberg, S. L., Fast gapped-read alignment with Bowtie 2, *Nature methods* **9,** 357 (2012).

252. R Core Team, *R: A Language and Environment for Statistical Computing* https://www.R-project.org/ (R Foundation for Statistical Computing, Vienna, Austria, 2018).

253. Hyatt, D. *et al.,* Prodigal: prokaryotic gene recognition and translation initiation site identification, *BMC bioinformatics* **11,** 119 (2010).

254. Steinegger, M. & Söding, J., MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets, *Nature biotechnology* **35,** Publisher: Nature Publishing Group, 1026–1028 (2017).

255. Roller, B. R. K., Stoddard, S. F. & Schmidt, T. M., Exploiting rRNA operon copy number to investigate bacterial reproductive strategies, en, *Nature Microbiology* **1,** Number: 11 Publisher: Nature Publishing Group, 1–7, ISSN: 2058-5276 (Sept. 2016).

256. Vieira-Silva, S. & Rocha, E. P. C., The Systemic Imprint of Growth and Its Uses in Ecological (Meta)Genomics, *PLoS Genetics* **6,** e1000808, ISSN: 1553-7390 (Jan. 2010).

257. Pertea, G. & Pertea, M., GFF Utilities: GffRead and GffCompare, *F1000Research* **9,** ISCB Comm J–304, ISSN: 2046-1402 (Sept. 2020).

258. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J., Basic local alignment search tool, en, *Journal of Molecular Biology* **215,** 403–410, ISSN: 0022-2836 (Oct. 1990).

259. Clark, D. R., Underwood, G. J., McGenity, T. J. & Dumbrell, A. J., What drives study-dependent differences in distance–decay relationships of microbial communities?, *Global Ecology and Biogeography* **30,** _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111 811–825 (2021).

260. Cabello-Yeves, P. J. *et al.,* The microbiome of the Black Sea water column analyzed by shotgun and genome centric metagenomics, *Environmental microbiome* **16,** Publisher: Springer, 1–15 (2021).

261. Karthikeyan, S. *et al.,* Genome repository of oil systems: an interactive and searchable database that expands the catalogued diversity of crude oil-associated microbes, *Environmental Microbiology* **22,** Publisher: Wiley Online Library, 2094–2106 (2020).

262. Jégousse, C., Vannier, P., Groben, R., Glöckner, F. O. & Marteinsson, V., A total of 219 metagenome-assembled genomes of microorganisms from Icelandic marine waters, *PeerJ* **9,** Publisher: PeerJ Inc., e11112 (2021).

263. Vosloo, S. *et al.,* Evaluating de Novo Assembly and Binning Strategies for Time Series Drinking Water Metagenomes, *Microbiology spectrum* **9,** Publisher: Am Soc Microbiol, e01434–21 (2021).

264. Merrill, B. D. *et al.,* Ultra-deep Sequencing of Hadza Hunter-Gatherers Recovers Vanishing Microbes, *bioRxiv,* Publisher: Cold Spring Harbor Laboratory (2022).

265. Miller, J. R., Koren, S. & Sutton, G., Assembly Algorithms for Next-Generation Sequencing Data, *Genomics* **95,** 315–327, ISSN: 0888-7543 (June 2010).

266. Lang, D., Zimmer, A. D., Rensing, S. A. & Reski, R., Exploring plant biodiversity: the Physcomitrella genome and beyond, en, *Trends in Plant Science* **13,** 542–549, ISSN: 1360-1385 (Oct. 2008).

267. Fritz, A. *et al.,* CAMISIM: simulating metagenomes and microbial communities, *Microbiome* **7,** Publisher: BioMed Central, 1–12 (2019).

268. Logares, R. *et al.,* Disentangling the mechanisms shaping the surface ocean microbiota, *Microbiome* **8,** Publisher: Springer, 1–17 (2020).

269. Kerkhof, L., Voytek, M., Sherrell, R. M., Millie, D. & Schofield, O., Variability in bacterial community structure during upwelling in the coastal ocean, *Hydrobiologia* **401,** Publisher: Springer, 139–148 (1999).

270. Allen, L. Z. *et al.,* Influence of nutrients and currents on the genomic composition of microbes across an upwelling mosaic, *The ISME journal* **6,** Publisher: Nature Publishing Group, 1403–1414 (2012).

271. Charuvaka, A. & Rangwala, H., Evaluation of short read metagenomic assembly, *BMC Genomics* **12,** S8 (2011).

272. Le Gall, G. *et al.,* Metabolomics of fecal extracts detects altered metabolic activity of gut microbiota in ulcerative colitis and irritable bowel syndrome, *Journal of proteome research* **10,** Publisher: ACS Publications, 4208–4218 (2011).

273. Zheng, D., Liwinski, T. & Elinav, E., Interaction between microbiota and immunity in health and disease, *Cell research* **30,** Publisher: Nature Publishing Group, 492–506 (2020).

274. Flores, G. E. *et al.,* Temporal variability is a personalized feature of the human microbiome, *Genome biology* **15,** Publisher: Springer, 1–13 (2014).

275. Gilbert, J. A. *et al.,* Current understanding of the human microbiome, *Nature medicine* **24,** Publisher: Nature Publishing Group, 392–400 (2018).

276. Wang, Z., Wang, Z., Lu, Y. Y., Sun, F. & Zhu, S., SolidBin: improving metagenome binning with semi-supervised normalized cut, *Bioinformatics* **35,** Publisher: Oxford University Press, 4229–4238 (2019).

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Nantes Université

Titre : Modèles et méthodes pour la métagénomique à résolution génomique

Mot clés : Binning, assemblage, métagénomique, ASP

Résumé : La reconstruction de génomes à partir de données métagénomiques, aussi appelés MAG) représente une étape majeure dans l'étude des communautés microbiennes. La reconstruction de MAGs souffre néanmoins de limitations, telles que la nature fragmentée de ces MAGs, les difficultés inhérentes à la reconstruction du pangénome, ou la capture des variations entre souches d'une même espèce. Dans cette thèse, le problème du binning a été appréhendé à travers un modèle de clustering suivant le paradigme de la logique déclarative, l'objectif étant de maximiser l'information sur les génomes présents grâce à l'exploration de l'ensemble des solutions de binning possible. Ce modèle de binning incluant métrique compositionnelle, mesure d'abondance et occurrence de gènes marqueurs a été implémenté en langage ASP. Nous nous sommes ensuite concentrés sur l'optimisation du processus d'assemblage, étape préliminaire clé de la classification de contigs, avec pour objectif d'encore améliorer la reconstruction de MAGs. Nous avons développé une approche automatique pour guider le processus de co-assemblage, couplant des distances métagénomiques avec une méthode d'optimisation du clustering. Cette approche a été intégrée dans un nouveau workflow de reconstruction de MAGs, MAGNETO, qui intègre également des stratégies assemblage-binning complémentaires.

Title: Models and methods for genome-resolved metagenomics

Keywords: Binning, co-assembly, metagenomics, ASP

Abstract: The reconstruction of individual genomes from metagenomic data, also called MAGs has constituted a major milestone in the study of microbial communities. However, the recovery of MAGs still suffers several limitations, including the mosaic and population nature of these MAGs, the inherent difficulties to assemble pangenomes, and the recovery of strain-level variations for a given species. In this thesis, a declarative programming framework was designed and used to resolve the genome binning problem through a constrained clustering approach, with the goal to explore several optimal binning solutions, informing us about the organization dynamics of naturally occurring genomes. A novel genome binning model integrating compositional and abundance information as well as constraints on single-copy core genes was designed and implemented using the ASP language. With the goal to further enhance the recovery of MAGs, we focused on optimizing the assembly process, a key genome binning preprocessing step. We developed an automated approach to guide the co-assembly process, combining metagenomic compositional distances with an optimal clustering method. These developments were implemented into a novel genome-resolved metagenomics workflow called MAGNETO, integrating complementary assembly-binning strategies.