BMC Bioinformatics

RESEARCH

Open Access

# Plaseval: a framework for comparing and evaluating plasmid detection tools

Aniket Mane[1*], Haley Sanderson[2], Aaron P. White[3], Rahat Zaheer[4], Robert Beiko[5,6] and Cédric Chauve[1*]

*Correspondence:
amane@sfu.ca; cedric.chauve@sfu.ca

[1] Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, Canada
[2] Agriculture and Agri-Food Canada, Saskatoon, Saskatchewan, Canada
[3] Department of Veterinary Microbiology, University of Saskatchewan, Saskatoon, Saskatchewan, Canada
[4] Agriculture and Agri-Food Canada, Lethbridge Research and Development Centre, Lethbridge, Alberta, Canada
[5] Department of Biology, Dalhousie University, Halifax, Nova Scotia, Canada
[6] Institute for Comparative Genomics, Halifax, Nova Scotia, Canada

## Abstract

**Background:** Plasmids play a major role in the transfer of antimicrobial resistance (AMR) genes among bacteria via horizontal gene transfer. The identification of plasmids in short-read assemblies is a challenging problem and a very active research area. *Plasmid binning* aims at detecting, in a draft genome assembly, groups (bins) of contigs likely to originate from the same plasmid. Several methods for plasmid binning have been developed recently, such as PlasBin-flow, HyAsP, gplas, MOB-suite, and plasmidSPAdes. This motivates the problem of evaluating the performances of plasmid binning methods, either against a given ground truth or between them.

**Results:** We describe PlasEval, a novel method aimed at comparing the results of plasmid binning tools. PlasEval computes a dissimilarity measure between two sets of plasmid bins, that can originate either from two plasmid binning tools, or from a plasmid binning tool and a ground truth set of plasmid bins. The PlasEval dissimilarity accounts for the contig content of plasmid bins, the length of contigs and is repeat-aware. Moreover, the dissimilarity score computed by PlasEval is broken down into several parts, that allows to understand qualitative differences between the compared sets of plasmid bins. We illustrate the use of PlasEval by benchmarking four recently developed plasmid binning tools—PlasBin-flow, HyAsP, gplas, and MOB-recon—on a data set of 53 *E. coli* bacterial genomes.

**Conclusion:** Analysis of the results of plasmid binning methods using PlasEval shows that their behaviour varies significantly. PlasEval can be used to decide which specific plasmid binning method should be used for a specific dataset. The disagreement between different methods also suggests that the problem of plasmid binning on short-read contigs requires further research. We believe that PlasEval can prove to be an effective tool in this regard. PlasEval is publicly available at https://github.com/acme92/PlasEval

**Keywords:** Plasmid prediction, Draft assembly, Comparative genomics, Benchmarking

## Introduction

Mobile Genetic Elements (MGEs) are DNA sequences within genomes that have the ability to move or be transferred within and between bacterial cells; MGEs can carry important genes, such as virulence factors or antimicrobial resistance (AMR) genes [1, 2]. Plasmids form an important family of MGEs due to their ability to transfer between

bacteria of different species through horizontal gene transfer, thus contributing to the spread of AMR genes. Due to their high mobility and the function of the genes they carry, detecting plasmids in bacterial isolates is important, whether it is for ecological studies or motivated by public health (e.g. tracking the spread of AMR genes in an infectious disease outbreak) [3]. This has motivated the development of bioinformatics tools to identify plasmids from draft bacterial genomes, a process called *plasmid binning*. Plasmid binning aims to detect, from a draft genome assembly, groups of contigs (called *plasmid bins*) that are assumed to originate from the same plasmid that was present in the sequenced isolate(s). Plasmid binning is known to be a challenging problem [4] and is an active research area [5–10].

With multiple plasmid binning methods developed to-date, choosing which one to apply in a given project for which plasmids are of interest is far from obvious. Suitability of a method can only be validated by applying various considered methods on a test data set, for which the true plasmids are known. Each true plasmid is represented by its corresponding plasmid bin harbouring the set of short read contigs that are mapped to it. These plasmid bins are hereafter referred to as the *ground truth*.

Such a data set can be built in several ways. It can be generated by analyzing isolates for which both closed genome assemblies, including plasmids, and sequencing data are publicly available, for example in public databases such as RefSeq and GenBank for closed assemblies and SRA for sequencing data. These testing samples can then be re-assembled into draft assemblies, after which ground truth plasmid bins can be obtained from mapping the resulting contigs against the true plasmids. Alternatively, isolates for which hybrid sequencing data, composed of both short reads (e.g. Illumina data) and long reads (e.g. Oxford Nanopore data), can be used. Hybrid assemblies tend to be more amenable to detecting plasmids than short-read or long-read assemblies individually [11–13]; once hybrid contigs identified as plasmids have been obtained, they can be used as in the previous case to define ground truth plasmid bins (see [14] for example). Nevertheless, given the plasmid bins predicted by a plasmid binning tool and ground truth plasmid bins, it is essential to assess the accuracy of the predicted plasmid bins through well-defined accuracy measures. Additionally, it is also of interest to compare the predictions of different plasmid-binning tools for the same assembly in order to identify plasmids that are consistently predicted across all the tools.

Accuracy measures developed to assess the quality of full genome assemblies [15] are not fully relevant for the plasmid binning problem. This has motivated the development of accuracy measures specific to the plasmid binning problem. The recent papers introducing gplas [8] and PlasBin-flow [10] introduced statistics inspired by the classical precision, recall and F1-score used to evaluate classification and clustering methods, that provide a high-level measure of accuracy. However, such high-level statistics do not immediately provide insight into the specific errors made by different methods; for example, before deciding which plasmid binning method to use, it might be of interest to know if incorrectly predicted plasmid bins result from true plasmid bins being split into several predicted bins or from the mixing of several true plasmid bins into several predicted plasmid bins. Moreover, comparing plasmid binning methods in absence of a ground truth, for example to understand the nature of the differences in the plasmid bins predicted by the considered tools could be valuable.

Here we introduce a novel measure of dissimilarity between a pair of sets of plasmid bins for a given draft genome assembly, implemented in the PlasEval software. The dissimilarity measure accounts for the difference in terms of contig content between two sets of plasmid bins, weighted by contig lengths. Mathematically, it is defined in terms of a parsimonious scenario of bin *splits* (contigs from a plasmid bin being split into two bins), followed by *joins* (two plasmid bins joined to form a single bin). The definition of the dissimilarity using splits and joins allows to separate the dissimilarity score between two sets of plasmid bins (either a set of predicted bins and a corresponding set of ground truth bins, or two sets of predicted bins) into different parts that provide both qualitative and quantitative insight into the nature of the differences between the two sets of bins. In Sect. , we introduce this dissimilarity measure together with an algorithm to compute it. We also describe practical cases in which the PlasEval tool is of interest. Section describes the results of our experiments, using PlasEval to evaluate four plasmid binning methods (MOB-recon [6], HyAsP [7], gplas [8] and PlasBin-flow [10]) on a data set of 53 *Escherichia coli* samples for which hybrid sequencing data is used to define ground truth plasmid bins. Our experiments illustrate the added value of PlasEval compared to high-level statistics such as precision and recall.

## Preliminaries

Plasmids are extra-chromosomal molecules present in a bacterial cell; they are generally much shorter than the chromosome(s), ranging from a few hundred nucleotides to less than a megabase, with larger plasmids being rare [16]. Most plasmids are circular molecules, and they can occur within a cell in multiple copies, up to hundreds of copies, although larger plasmids tend to be single-copy. Plasmids often harbor repeats (such as insertion sequences), that can be repeated within a plasmid, between plasmids or between plasmids and the chromosome.

***Plasmid binning.*** Given a contig assembly with contig set $\mathcal{C}$, a *plasmid bin $P$ is a set of contigs* (i.e. is an unordered subset of $\mathcal{C}$). The *plasmid binning* problem aims to compute, from the contig assembly of a bacterial isolate, a set of plasmid bins, ideally representative of the plasmid content of the isolate (i.e. where the contigs in each plasmid bin are the contigs of a plasmid of the isolate).

Due to the frequent presence of repeats in plasmid, a given contig $c \in \mathcal{C}$ can appear in several plasmid bins. A contig can also be repeated within a plasmid (which would make a plasmid bin a multiset of contigs), but most plasmid binning methods do not account for such features and we do not consider this case here.

Plasmid binning from a short-reads contigs assembly is a challenging problem [4]; advances in long-reads sequencing technologies allow to detect plasmids with greater accuracy [16] although they often miss shorter plasmids [17]. Nevertheless, in a practical context such as epidemiological surveillance, standard approaches still rely on short-reads sequencing [18], thus we focus this work on evaluating plasmid binning methods from a short-reads contigs assembly.

***Plasmid binning methods.*** Initial approaches to detect plasmids from an assembly focused on the problem of *contigs classification*. In the contig assembly of a bacterial isolate, a contig is *plasmidic* if the corresponding sequence occurs only in some plasmid(s) of the sequenced isolate, *chromosomal* if it belongs only to the chromosome

of the isolate, and *ambiguous* if it is a sequence that occurs both in some plasmid and in the chromosome. Contig classification aims to label contigs as either plasmidic, chromosomal or ambiguous, often using machine-learning approaches (see [14, 19]; [20–22]). The first plasmid binning methods included PlasmidSPAdes [5] and Recycler [23], both leveraging the information provided by the *assembly graph* (a graph whose vertices are contigs and edges indicate putative contiguity between contigs) to detect plasmid bins as cyclic groups of contigs that satisfy read-depth coverage consistency indicative of multi-copy plasmids. The idea of using the assembly graph, together with plasmid-specific features (presence of known plasmid genes, GC content, read-depth used as a proxy of copy number), is at the core of the most recent plasmid binning methods. Both HyAsP [7] and gplas [8] rely on heuristics to compute plasmid bins as walks in the assembly graphs, while PlasBin [9] and its recent extension PlasBin-flow [10] are based on Mixed Integer Linear Programming to compute plasmid bins as connected subgraphs of the assembly graph. MOB-recon [6] is the only method that does not make use of the assembly graph, relying instead on the comparison of the contigs against a carefully curated database of known plasmids and plasmid-specific genes.

***Measuring plasmid binning accuracy.*** Most papers on plasmid binning methods do measure the accuracy of a set of plasmid bins by comparing them to a *ground truth* set of plasmid bins using variations of the notions of precision, recall and F1-score. In our work, we compare the novel measure of accuracy that we introduce to the F1-score defined in [10], that we define formally below.

For a given contig set $\mathcal{C}$, let $\mathcal{A}$ be the set of plasmid bins predicted by a plasmid binning tool and $\mathcal{B}$ the set of true plasmid bins (ground truth). For a set of contigs $X$, we define the cumulative length of the contigs in $X$ as $L(X)$. For a predicted plasmid bin $P \in \mathcal{A}$ and a ground truth plasmid bin $T \in \mathcal{B}$, we define the overlap between $P$ and $T$ as the cumulative length of the contigs in the intersection of the two sets of contigs $P$, $T$: $\text{overlap}\,(P,T) = L(P \cap T)$. The precision and recall of the set $\mathcal{A}$ of predicted plasmid bins compared to the ground truth $\mathcal{B}$ are respectively defined as

$$p(\mathcal{A},\mathcal{B}) = \frac{\sum\limits_{P\in\mathcal{A}} \max\limits_{T\in\mathcal{B}} \text{overlap}\,(P,T)}{\sum\limits_{P\in\mathcal{A}} L(P)},\ r(\mathcal{A},\mathcal{B}) = \frac{\sum\limits_{T\in\mathcal{B}} \max\limits_{P\in\mathcal{A}} \text{overlap}\,(P,T)}{\sum\limits_{T\in\mathcal{B}} L(T)}$$

and the F1-score is the arithmetic mean of the precision and recall, $F_1(\mathcal{A},\mathcal{B}) = 2\frac{p(\mathcal{A},\mathcal{B})r(\mathcal{A},\mathcal{B})}{p(\mathcal{A},\mathcal{B})+r(\mathcal{A},\mathcal{B})}$.

## Methods

The framework we propose aims at computing a weighted set-theoretic dissimilarity measure between two inferred sets of plasmid bins. It is based on the idea of transforming one set of plasmid bins into the other one by a sequence of operations splitting plasmid bins, followed by a sequence of operations joining the resulting splitted bins, while accounting for the contig lengths, unequal contig content and the possible presence of repeated contigs.

In the remainder of this section, we denote by $\mathcal{C}$ the contig alphabet and consider two sets of plasmid bins denoted by $\mathcal{A}$ and $\mathcal{B}$ over $\mathcal{C}$. We denote by $D_\alpha(\mathcal{A},\mathcal{B})$ the dissimilarity between $\mathcal{A}$ and $\mathcal{B}$, where $\alpha \in [0,1]$ is a parameter used to weight how contig lengths contribute to the dissimilarity score. We first describe how to compute efficiently $D_\alpha(\mathcal{A},\mathcal{B})$ in the simpler context where no contig is repeated, then describe how we handle repeated contigs through a branch-and-bound algorithm. We conclude this section by a discussion on theoretical properties of the dissimilarity measure we introduce.

### Dissimilarity without repeated contigs

We first consider the case where each contig $c \in \mathcal{C}$ appears in at most one plasmid bin in $\mathcal{A}$ and at most one plasmid bin in $\mathcal{B}$. We denote by unique$(\mathcal{A},\mathcal{B})$ the set of *unique* contigs $c$ such that $c$ appears in $\mathcal{A}$ but not $\mathcal{B}$ or conversely, and by $\mathcal{A}'$ and $\mathcal{B}'$ the sets of plasmid bins obtained by deleting respectively from $\mathcal{A}$ and $\mathcal{B}$ these unique contigs. For a contig $c$, we denote by $\ell(c)$ its length. For a set of contigs $X$, we define $L_\alpha(X)$ as $L_\alpha(X) = \left(\sum_{c\in X} \ell(c)\right)^\alpha$, where $\alpha \in [0,1]$ is a fixed parameter to weight the contribution of contig lengths to the dissimilarity score.

We first define the operation of splitting and joining plasmid bins, together with the corresponding cost, defined in terms of the lengths of the involved contigs. Figure 1 illustrates these two operations.

**Definition 1** (Plasmid bin split and join) Let $P$ be a set of contigs (plasmid bin); a *split* of $P$ is composed of two non-empty sets $P', P''$ such that $P' \cap P'' = \emptyset$ and $P' \cup P'' = P$. The *cost of splitting* $P$ into $P', P''$ is $c_\alpha(P; P', P'') = \min(L_\alpha(P'), L_\alpha(P''))$.

A *join* is the symmetric operation: it takes two non-empty sets of contigs $P', P''$ and joins them into a single set $P$. The cost of joining $P', P''$ into $P$ is $c_\alpha(P', P''; P) = \min(L_\alpha(P'), L_\alpha(P''))$.

The similarity measure $D_\alpha(\mathcal{A},\mathcal{B})$ we introduce can be defined informally as being composed of two main parts (1) discarding from $\mathcal{A}$ and $\mathcal{B}$ their unique contigs, and accounting for the total length of these unique contigs, and (2) transforming $\mathcal{A}'$ into $\mathcal{B}'$ by first splitting the plasmid bins of $\mathcal{A}'$ and then joining the resulting bins to obtain $\mathcal{B}'$. Definition 2 below defines the set of bins obtained after the splitting phase.

**Definition 2** For a plasmid bin $P$ in $\mathcal{A}'$, we define its *refinement with respect to* $\mathcal{B}'$, denoted by ref$(P,\mathcal{B}')$, as the unique partition of $P$ into non-empty sets $P_1,\ldots,P_k$ of contigs such that for any $P_i$ there exists $Q \in \mathcal{B}'$ with $P_i = P \cap Q$.

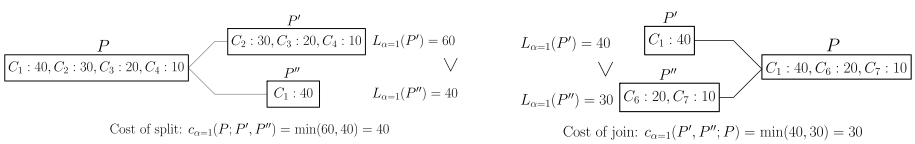


**Fig. 1** Cost of split and join operations with $\alpha = 1$

We denote by $\text{ref}(\mathcal{A}', \mathcal{B}')$ the set of plasmid bins obtained by refining all plasmid bins in $\mathcal{A}'$ with respect to $\mathcal{B}'$. It is straightforward to see that $\text{ref}(\mathcal{B}', \mathcal{A}') = \text{ref}(\mathcal{A}', \mathcal{B}')$. We also denote, for a plasmid bin $P$ of $\mathcal{A}'$ (resp. $Q$ from $\mathcal{B}'$) by $c_\alpha(P; P_1, ..., P_k)$ (resp. $c_\alpha(Q_1, ..., Q_k; Q)$) the cost of a most parsimonious sequence of splits (resp. joins) creating the bins $P_1, \ldots, P_k$ of $\text{ref}(\mathrm{P}, \mathcal{B}')$ (resp. $Q_1, ..., Q_k$ of $\text{ref}(\mathrm{Q}, \mathcal{A}')$).

**Definition 3** (Dissimilarity with no repeated contig)

$$D_\alpha(\mathcal{A}, \mathcal{B}) = \sum_{c \in \text{unique}(\mathcal{A}, \mathcal{B})} \ell(c)^\alpha + \sum_{P \in \mathcal{A}'} c_\alpha(P; P_1, ..., P_k) + \sum_{Q \in \mathcal{B}'} c_\alpha(Q_1, ..., Q_k; Q). \quad (1)$$

The first term in (1) accounts for unique contigs, while the second term records the cost of splitting $\mathcal{A}'$ into $\text{ref}(\mathcal{A}', \mathcal{B}')$, and the third term the cost of joining $\text{ref}(\mathcal{A}', \mathcal{B}')$ into $\mathcal{B}'$.

We now turn to the cost of refining a set of bins $P$ into $\{P_1, \ldots, P_k\}$ as defined in Definition 2. Lemma 1 below (proof provided in Appendix A) states that a most parsimonious way to refine a plasmid bin is to perform splits that create the sets $P_1, \ldots, P_k$ by increasing order of size: assuming $L(P_1) \leq L(P_2) \leq \cdots \leq L(P_k)$, creating $P_1$ out of $P$ by a single split, then $P_2$ out of $P - P_1$ by a split, and so on. By symmetry of splits and joins the same approach also applies to compute $c_\alpha(Q_1, ..., Q_k; Q)$.

**Lemma 1**

$$c_\alpha(P; P_1, ..., P_k) = \left( \sum_{P_i \in \text{ref}(\mathrm{P}, \mathcal{B}')} L_\alpha(P_i) \right) - \left( \max_{P_i \in \text{ref}(\mathrm{P}, \mathcal{B}')} L_\alpha(P_i) \right). \quad (2)$$

Thus, in the absence of repeated contigs, $D_\alpha(\mathcal{A}, \mathcal{B})$ can be computed easily by the following steps:

- First discarding unique contigs to create $\mathcal{A}', \mathcal{B}'$, which contributes $\sum_{c \in \text{unique}(\mathcal{A}, \mathcal{B})} \ell(c)^\alpha$ to $D_\alpha(\mathcal{A}, \mathcal{B})$,
- Then splitting $\mathcal{A}'$ into $\text{ref}(\mathcal{A}', \mathcal{B}')$, with cost as defined in Lemma 1,
- Finally joining $\text{ref}(\mathcal{A}', \mathcal{B}')$ into $\mathcal{B}'$, or equivalently splitting $\mathcal{B}'$ into $\text{ref}(\mathcal{A}', \mathcal{B}')$, with cost as defined in Lemma 1.

All these steps can be done in polynomial time, so in the absence of repeated contig, $D_\alpha(\mathcal{A}, \mathcal{B})$ can be computed in polynomial time.

#### Dissimilarity with repeated contigs

To handle the presence of repeated contigs, we follow the *maximal matching* approach used to compute genome rearrangement distances with repeated genes.

Let $c$ be a contig that appears in $k_1$ copies in $\mathcal{A}$ and $k_2$ copies in $\mathcal{B}$; we call the set of all copies of $c$ a *contig family* and we say this family is a *repeat family* if $k_1 + k_2 > 2$. We denote by $\text{repeats}(\mathcal{A}, \mathcal{B})$ the set of all repeat families in $\mathcal{A}, \mathcal{B}$.

A maximal matching $M_c$ for a contig family $c$, with respectively $k_1, k_2$ copies of $c$ in $\mathcal{A}, \mathcal{B}$, is composed of $k = \min(k_1, k_2)$ pairs of contigs $c$ (called edges), each pair containing one copy of $c$ in each of $\mathcal{A}, \mathcal{B}$. Given $M_c$, if we label each edge by a unique integer $i \in \{1, \ldots, k\}$ and rename with $c_i$ both extremities of the edge (contigs initially labelled $c$), while the remaining (if any) unmatched copies of $c$ that do not belong to any edge of $M_c$ are labeled arbitrarily $c_j$, $j \in \{k+1, \ldots, k_1+k_2\}$, we say that the contig family $c$ has been *resolved*, i.e. that all copies of $c$ have been renamed in such a way that they can be considered as distinguishable.

Let $M$ be a set of maximal matchings $M_c$, one for each contig family $c \in \text{repeats}(\mathcal{A}, \mathcal{B})$; we denote by $M(\mathcal{A}), M(\mathcal{B})$ the corresponding resolved sets of plasmid bins, where all repeated contigs have been renamed as described above, resulting in an instance with no repeated contig, for which $D_\alpha(M(\mathcal{A}), M(\mathcal{B}))$ can then be defined as in Definition 3 and computed in polynomial time.

**Definition 4** (Dissimilarity with repeated contig) Let $\mathcal{M}(\mathcal{A}, \mathcal{B})$ be the set of all sets of maximal matchings resolving all repeat families in $\mathcal{A}, \mathcal{B}$.

$$D_\alpha(\mathcal{A}, \mathcal{B}) = \min_{M \in \mathcal{M}(\mathcal{A}, \mathcal{B})} D_\alpha(M(\mathcal{A}), M(\mathcal{B})). \tag{3}$$

To search the space $\mathcal{M}(\mathcal{A}, \mathcal{B})$ of all matchings resolving repeat families in order to find one that minimizes the resulting dissimilarity, we implemented a branch-and-bound algorithm that we describe now at a high level, more details being provided in Appendix B.

The branch-and-bound algorithm is based on exploring a search tree that builds incrementally the sets of bins $\mathcal{A}, \mathcal{B}$ by adding contigs from repeat families, updating the distance as it goes on in a monotonic way.

- The root of the search tree (level 0) is defined as the instance $\mathcal{A}_0, \mathcal{B}_0$ obtained by keeping only non-repeat contig families; due to the absence of repeated contigs, $D_\alpha(\mathcal{A}_0, \mathcal{B}_0)$ can be computed in polynomial time.
- Each level $i$ of the search tree corresponds to adding the contigs of exactly one repeat family to $\mathcal{A}_{i-1}, \mathcal{B}_{i-1}$ in all possible ways, to define larger instances.
- Assume the considered repeat family is for contig $c$, having $k_1, k_2$ copies in $\mathcal{A}, \mathcal{B}$ respectively. There are $\binom{\max(k_1,k_2)}{\min(k_1,k_2)} \min(k_1, k_2)!$ possible maximal matchings between the copies of $c$ in $\mathcal{A}_i, \mathcal{B}_i$, and each matching defines a way to add all copies of $c$ to $\mathcal{A}_{i-1}, \mathcal{B}_{i-1}$ thus creating a larger instance $\mathcal{A}_i, \mathcal{B}_i$ with no repeated contig and for which the dissimilarity $D_\alpha(\mathcal{A}_i, \mathcal{B}_i)$ can be computed in polynomial time; the set of all these instances form the level $i$ of the search tree.
- Finally, a subtree of the search tree, say rooted at level $i$, is explored only if the dissimilarity value $D_\alpha(\mathcal{A}_i, \mathcal{B}_i)$ of the (partial) instance $\mathcal{A}_i, \mathcal{B}_i$ at its root is strictly lower than the best found dissimilarity value for a full instance; this approach is a proper bounding method as Lemma 2 (Appendix A) shows that adding contigs can never decrease the dissimilarity value.

In some cases, especially for sets of plasmid bins obtained from an assembly with a large number of short contigs, the branch-and-bound algorithm can be

time-consuming. To handle such cases, we allow users to set a *minimum length* parameter $\ell$ (default value $\ell = 0$) and all contigs of length below $\ell$ are discarded from the considered plasmid bins.

#### Normalized dissimilarity

In order to compare the dissimilarity we introduced to the notions of precision, recall and F1-score, we normalize it between 0 and 1, by dividing it by the sum of the weighted lengths of the contigs appearing in $\mathcal{A}$ and$\mathcal{B}$:

$$d_\alpha(\mathcal{A},\mathcal{B}) = \frac{D_\alpha(\mathcal{A},\mathcal{B})}{\sum\limits_{P\in\mathcal{A}}\sum\limits_{c\in P}\ell(c)^\alpha + \sum\limits_{Q\in\mathcal{B}}\sum\limits_{c\in Q}\ell(c)^\alpha}. \tag{4}$$

Note that this normalization accounts for the presence of repeats as a contig $c$ appearing in several bins will contribute by $\ell(c)^\alpha$ according to it copy number in both $\mathcal{A}$ and $\mathcal{B}$. It is straightforward to prove that (1) $d_\alpha(\mathcal{A},\mathcal{B}) = 0$ if and only if $\mathcal{A} = \mathcal{B}$, and (2) $d_\alpha(\mathcal{A},\mathcal{B}) = 1$ if and only if $\mathcal{A}$ and $\mathcal{B}$ have no common contig, i.e. every contig appears only in either $\mathcal{A}$ or $\mathcal{B}$.

#### Discussion

If one considers $\mathcal{A}$ as a set of plasmid bins predicted by a plasmid binning tool and $\mathcal{B}$ as the ground truth for the same sample, $D_\alpha(\mathcal{A},\mathcal{B})$ accounts for four kinds of errors in the predicted bins: contigs that are erroneously predicted as plasmidic (*extra contigs*), plasmidic contigs that are not included in any predicted bin (*missed contigs*), contigs from different true bins that are placed into the same predicted bin (accounted for by splits) and contigs of a true bin that are separated into several predicted bins (accounted for by joins).

The parameter $\alpha$ allows to control the contribution of contig lengths in the dissimilarity score. With $\alpha = 0$, the dissimilarity is purely set-theoretic and counts only splits and joins, augmented by 1 for the presence of extra contigs and 1 for missing contigs, and thus does not account at all for contig length. With $\alpha = 1$, the contribution of splits (resp. joins) to the dissimilarity score is exactly the *precision* (resp. *recall*) as defined in [10] (see Sect. ). Using $\alpha = 1$ leads to the issue that the fragmentation of a true plasmid bin into several predicted bins is not accounted for. To illustrate this, consider a true plasmid bin $P$ with $2k$ contigs, all of the same length, and two sets of predicted bins, $\mathcal{P}_1$ composed of two bins each containing $k$ contigs, and $\mathcal{P}_2$ composed of one bin of $k$ contigs and $k$ bins of one contig each. We then have $d_1(\mathcal{P}_1,\{P\}) = d_1(\mathcal{P}_2,\{P\}) = 0.5$, which is the value of both the precision and the recall, thus not accounting for the higher fragmentation of $\mathcal{P}_2$, while for any $\alpha < 1$, $d_1(\mathcal{P}_1,\{P\}) < d_1(\mathcal{P}_2,\{P\})$. In our experiments, we use $\alpha = 0.5$.

The core of the definition of $D_\alpha(\mathcal{A},\mathcal{B})$ is to transform the set of plasmid bins $\mathcal{A}'$ into $\mathcal{B}'$, where both have equal contig content. This relates to the *syntenic distance* introduced in [24] as a genome rearrangement distance for genomes represented as unordered sets of genes, whose computation is NP-complete [25]. It differs in two major points: (1) while splits and joins correspond to fissions and fusions, our approach

does not consider the operation *translocations* considered in the syntenic distance, that would correspond to exchanging sets of contigs between two plasmid bins, and (2) we consider only sequences of splits followed by joins and do not consider mixing both operations. Including translocation in measuring the dissimilarity between sets of plasmid bins would potentially capture more complex errors, such as the mixing of two true plasmid bins into two predicted bins, at the expense of tractability. Considering arbitrary sequences of splits and joins would result in a lower dissimilarity score, although we motivate our choice of considering only sequences of splits followed by joins by the fact that the structure of the intermediate set of bins obtained after all splits, $\mathrm{ref}(\mathcal{A}', \mathcal{B}')$, is indicative of the structural differences between the two considered sets of plasmid bins. Moreover, this approach leads to an efficient way to compute the dissimilarity with no repeated contigs, which is crucial in ensuring that the branch-and-bound algorithm that considers repeats does finish in a reasonable computational time.

#### Practical use of PlasEval

PlasEval is designed to provide computational biologists and bioinformaticians with a precise way to assess the performances of methods aimed at detecting plasmids from short read draft assemblies.

The main application of PlasEval is to compare a set of plasmid binning tools. It compares two sets of plasmid bins, obtained using an identical set of contigs; meaning the same assembly graph has been processed by multiple plasmid binning tools. Given a test dataset of samples for which both short read assemblies and ground truth plasmid bins are available, PlasEval allows to determine the strengths and weaknesses of all considered tools, with high-level statistics (the dissimilarity score) and more refined statistics in terms of the four components of the dissimilarity score (missing and extra contigs, plasmid bins splits and joins). This can be motivated by several applications:

- Comparing a novel plasmid binning tool against a set of state-of-the-art existing plasmid binning tools.
- Investigating a set of plasmid binning tools with the aim to chose one (or several) to apply on a specific dataset.
- Benchmarking a set of plasmid binning tools either against a dataset for which the ground truth plasmid bins are known.

In the first two cases, using PlasEval requires the availability of samples for which the true plasmid bins are known. This can be obtained from samples for which short read assemblies are available or can be computed from sequencing data and either (1) annotated closed genomes are available or (2) hybrid (short and long read) assemblies are available. In Sect.  we describe a simple protocol to determine ground truth plasmid bins from short read assemblies and hybrid assemblies.

## Results and discussion

In this section, we use PlasEval to evaluate four plasmid binning tools on a set of 53 samples from a collection of *Escherichia coli* genomes sequenced using both short and long reads by Sanderson et al [26].

### Data

The isolates in this analysis belong to a collection of *E. coli* strains collected from Saskatchewan broiler farms by Sanderson et al [26]. The hybrid assemblies, combining short and long reads, were obtained using Unicycler (v0.5.0) [27] in hybrid mode, and were used to define the ground truth for each sample. The hybrid assemblies are available under NCBI BioProject accession number PRJNA912639. The short-read assemblies were generated as follows: Illumina short-read quality was checked using FastQC (v0.11.9) [28], then short reads were trimmed and the adaptors were removed using Fastp (v0.23.4) [29], then the reads were assembled using Unicycler (v0.5.0) [27]. The quality of the short-read assemblies was assessed with Quast (v5.0.2) [30].

For each hybrid assembly, generated hybrid contigs were labelled as chromosome, plasmid or ambiguous based on contig length and circularity. Circular contigs that were longer than 500, 000 bp were labelled as chromosomes. Circular contigs shorter than 500, 000 bp were labelled as plasmids, while the remaining contigs were labelled as ambiguous. In order to have high-quality ground truth, we selected 53 hybrid assemblies with no ambiguous contigs collectively containing a total of 190 plasmidic contigs, called plasmids from now on. The plasmids had an average length of 48, 365 bp, with the largest plasmid in the dataset being of length 206, 436 bp. Figure 5 shows the length distribution of chromosomal contigs.

The plasmid binning tools we evaluate are designed to predict plasmid bins from short-read assemblies. We assembled the short reads for the 53 samples using Unicycler, that generates, together with a set of contigs, an *assembly graph* that is used by three of the considered plasmid binning tools. The short-read assemblies of the 53 samples consists of 18, 078 contigs in total: 872 of these contigs are longer than $10^5$ bp, 1, 830 contigs are between $10^4$ bp and $10^5$ bp, 1, 793 are between $10^3$ bp and $10^4$ bp, 7, 798 are between 100 bp and 1, 000 bp, while the remaining 5, 785 contigs are shorter than 100 bp. The average length of the short-read contigs is 14, 983 bp, with the longest contig being of length 790, 743 bp.

The hybrid assemblies and short-read assemblies are available at https://zenodo.org/records/10785150, and at NCBI in the BioProject PRJNA912639. We refer to these genomes here by their NCBI BioSample accessions; for example, "SAMN32247327" is the accession for the genome with the assigned isolate name "EC_B1_9226_C5_H_CuN_CeP" that is accessible at https://www.ncbi.nlm.nih.gov/biosample/SAMN32247327/ in the BioProject.

### Experimental setup

For all samples, the plasmid binning tools MOB-recon [6], HyAsP [7], gplas [8] and PlasBin-flow [10] were used on the short-read assemblies to predict plasmid bins. MOB-recon, being a homology-based method, selects contigs potentially belonging to

plasmids by mapping them against a plasmid gene database of plasmid replicons and relaxase genes. It also assigns the contig a cluster label. Contigs with the same cluster label are then grouped together to form a plasmid bin. Gplas uses a plasmid identification tool to assign probabilities for each contig belonging to a plasmid or chromosome. It then forms a plasmidome network by identifying plasmid-like walks in the assembly graph. This network is then partitioned into components that represent plasmid bins. HyAsP identifies seed contigs in the assembly graph; these contigs have a high proportion of genes from known plasmid gene databases. It tries to identify circuits or trails in the assembly graph by a greedy approach, aiming to optimize an objective function that accounts for a prior plasmid score for each contig, GC content and depth of sequencing used as a proxy for copy number; the computed circuit or trail define each a plasmid bin. PlasBin-flow follows the general principle of HyAsP but defines plasmid bins as connected subgraphs of the assembly graphs identified using an exact Mixed Integer Linear Programming approach where the copy number of each plasmid bin is approximated using a network flow. For running gplas, we used the *E. coli* model of mlplasmids [19] to classify contigs as plasmidic or chromosomal prior to the binning step. We also used the classification probabilities computed by mlplasmids as the plasmid score for PlasBin-flow, keeping all other parameters with their default value. MOB-recon and HyAsP were run using their default parameters.

Ground truth plasmid bins were obtained for the 53 selected samples by mapping the short-read contigs to the hybrid contigs using BLAST+ [31], discarding BLAST hits with identity below 95% or covering less than 80% of the short-read contig. For a given hybrid assembly contig (considered as a full plasmid), the corresponding ground truth plasmid bin is defined as the set of all short-read contigs belonging to hits to this hybrid contig.

The ground truth plasmid bins and the results from the four plasmid binning tools are available at https://zenodo.org/records/10785150.

In a practical context, the design of our experiments illustrate the use of PlasEval in a project where one would like to decide which plasmid binning tool to use in a specific project where both short read and long rad sequencing data is available for a set of isolates. A subset of samples for which hybrid sequencing data allow to recover a reasonable ground truth are used to benchmark a set of plasmid binning tools, and the PlasEval results allows to understand how these tools behave on this data and to select a tool to use on the whole dataset.

#### Evaluating accuracy against the ground truth

We first evaluated the results of all four binning methods using the notions of precision, recall and F1 score defined in Sect. . We then used PlasEval to compute the dissimilarity scores between the predicted plasmid bins of each plasmid binning tool and the ground truth. Figure 2 shows the correlation between the F1-score and the dissimilarity measure for all 53 samples.

Despite the difference between the two accuracy measures, the dissimilarity score $d_{0.5}$ is mostly inversely correlated to the F1 score: PlasBin-flow and HyAsP showed Pearson correlation coefficients of $-0.85$ and $-0.76$ respectively, while we observe a stronger correlation coefficient of $-0.91$ and $-0.92$ for the results of gplas and MOB-recon, respectively.

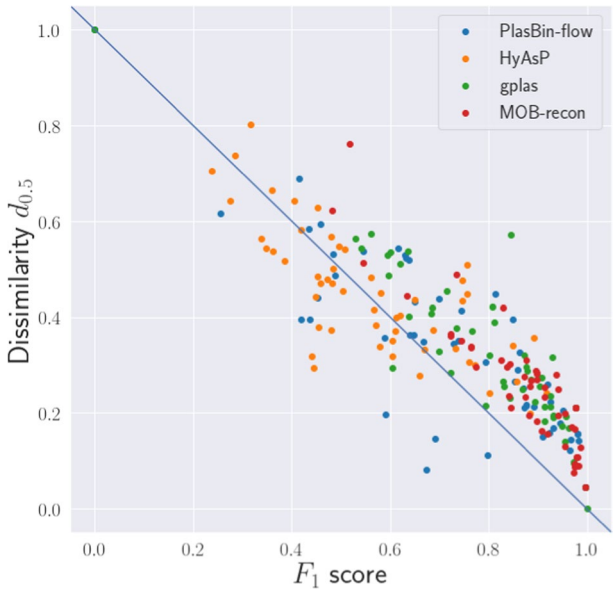


**Fig. 2** Correlation between F1 score and dissimilarity *D*

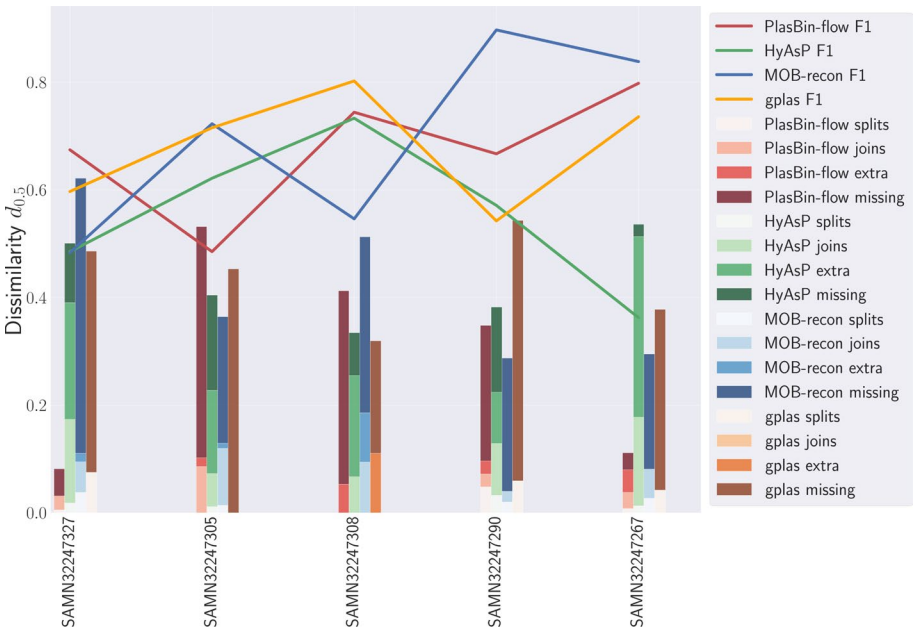


**Fig. 3** Dissimilarity comparison between all methods on five reference genome assemblies. The dissimilarity value is broken into its four components accounting for extra and missed contigs, splits and joins

Overall this shows that the novel measure that we introduce is consistent with the F1 score. However, one of the interests of the dissimilarity measure is that it can be broken down into four components representing missing and extra contigs in predicted plasmid bins as well as the cost of splits and joins, thus providing a refined view of the errors in predicted plasmid bins. We illustrate this in Fig. 3 where we show the respective contribution of all four components of the dissimilarity measure $d_{0.5}$ for five randomly chosen samples.

**Table 1** Mean and standard deviation of the PlasEval dissimilarity score and $F_1$-score for the predictions of all four methods on 53 *E. coli* samples. The contribution of splits, joins, extra contigs and missing contigs to the dissimilarity is also shown

| Method | Splits | Joins | Extra | Missing | $d_{0.5}$ | $F_1$ |
|---|---|---|---|---|---|---|
| PlasBin-flow | 0.0018 | 0.0476 | 0.0458 | 0.2372 | 0.3324 | 0.7275 |
| | ± 0.0077 | ± 0.0547 | ± 0.0486 | ± 0.1792 | ± 0.1878 | ± 0.2172 |
| HyAsP | 0.0052 | 0.0795 | 0.3019 | 0.0599 | 0.4464 | 0.5631 |
| | ± 0.0114 | ± 0.0793 | ± 0.1832 | ± 0.0587 | ± 0.1341 | ± 0.1747 |
| gplas | 0.0151 | 0.0024 | 0.0455 | 0.2501 | 0.3129 | 0.8068 |
| | ± 0.0312 | ± 0.0079 | ± 0.0557 | ± 0.1753 | ± 0.1741 | ± 0.1798 |
| MOB-recon | 0.0039 | 0.0186 | 0.0518 | 0.1768 | 0.2511 | 0.8730 |
| | ± 0.0084 | ± 0.0347 | ± 0.0711 | ± 0.1229 | ± 0.1400 | ± 0.1230 |

For instance, for sample SAMN32247327, we can see that MOB-recon and HyAsP have the same F1 scores but they have different dissimilarity $d_{0.5}$ values and error types; the error in the plasmid bins predicted by gplas stem mostly from missing (plasmidic) contigs, while HyAsP errors are dominated by including in plasmid bins contigs not present in the ground truth (i.e. chromosomal contigs).

Table 1 below shows the mean and standard deviation of the F1-score and the PlasEval dissimilarity score for the four tools and over all samples. Figure 6 provides a detailed illustration of the contribution of the four components of the PlasEval score over all samples and all considered methods.

Table 1 shows that overall, HyAsP is the least accurate of the three tools, and that the main errors in plasmid bins computed by HyAsP result from the inclusion in plasmid bins of chromosomal contigs. Interestingly, PlasBin-flow, a method based on a similar objective than HyAsP but using an exact optimization method instead of a greedy heuristic, is much more accurate than HyAsP; one can observe that its errors are mostly due to the omission of plasmidic contigs in the plasmid bins, while it is much more accurate than HyAsP for the other three components of the PlasEval score. This comparison illustrates the trade-off between an aggressive greedy heuristic (HyAsP) and an exact optimization method. Overall, the most accurate method on these samples is MOB-recon, followed by gplas. Similarly to PlasBin-flow, for both MOB-recon and gplas, the errors, as measured by the PlasEval score, stem from the omission of plasmidic contigs in the plasmid bins. Overall, more than 80% of the dissimilarity can be attributed to discrepancies in identifying plasmid contigs, suggesting that an avenue to improve plasmid binning is to improve true plasmidic contigs identification (contigs classification). Moreover, the patterns observed in terms of the contributions of splits and joins are different in the three most accurate methods PlasBin-flow, MOB-recon and gplas. For PlasBin-flow the contribution of splits is very low, while joins contribute more, suggesting it is a conservative method, that tends to not mix into a single plasmid bin contigs from different plasmids, at the expense of splitting true plasmids into several bins. The contribution of splits for gplas is somewhat high, suggesting a higher rate of true plasmids mixed into a single plasmid bin. Last, MOB-recon sits in-between with a low contribution of splits and a moderate contribution of joins.

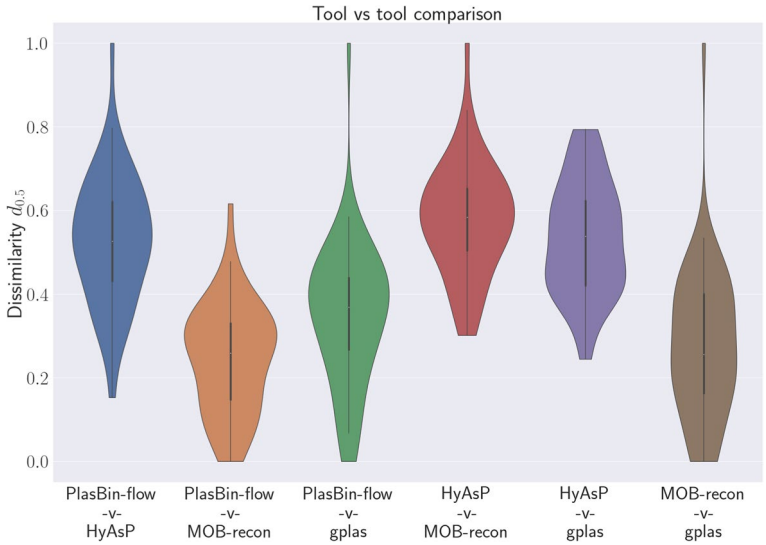


**Fig. 4** Distribution of the dissimilarity measure $d_{0.5}$ between pairs of plasmid binning tools over all 53 genomes in the data set

**Table 2** Mean and standard deviation for the different components of the dissimilarity score between results of different plasmid binning predictions

| Method pairs | Splits | Joins | Extra | Missing | $d_{0.5}$ |
|---|---|---|---|---|---|
| PlasBin-flow | 0.0358 | 0.0200 | 0.0363 | 0.4369 | 0.5289 |
| -vs- HyAsP | ± 0.0436 | ± 0.0318 | ± 0.0440 | ± 0.1735 | ± 0.1536 |
| PlasBin-flow | 0.0017 | 0.0924 | 0.0773 | 0.0756 | 0.2470 |
| -vs- gplas | ± 0.0076 | ± 0.0887 | ± 0.0705 | ± 0.0961 | ± 0.1311 |
| PlasBin-flow | 0.0106 | 0.0481 | 0.1101 | 0.1813 | 0.3501 |
| -vs- MOB-recon | ± 0.0295 | ± 0.0668 | ± 0.0836 | ± 0.1720 | ± 0.1718 |
| HyAsP | 0.0007 | 0.0663 | 0.4694 | 0.0423 | 0.5787 |
| -vs- gplas | ± 0.0038 | ± 0.0666 | ± 0.1502 | ± 0.0467 | ± 0.1354 |
| HyAsP | 0.0115 | 0.0715 | 0.4112 | 0.0383 | 0.5325 |
| -vs- MOB-recon | ± 0.0218 | ± 0.0792 | ± 0.1567 | ± 0.0505 | ± 0.1349 |
| gplas | 0.0371 | 0.0030 | 0.0784 | 0.1578 | 0.2763 |
| -vs- MOB-recon | ± 0.0510 | ± 0.0100 | ± 0.0836 | ± 0.1581 | ± 0.1788 |

The comparison of PlasBin-flow and gplas is interesting and illustrates well the use of PlasEval. Indeed, one can observe a similar overall score, similar contributions of missing and extra contigs, but a different pattern in the contributions of splits and joins. It follows that, if one is interested in choosing a plasmid binning tool that results in predicted plasmid bins that do not mix several true plasmids into a single bin, then PlasBin-flow is a more appropriate tool than gplas.

### Comparing plasmid-binning tools

Alternatively to providing a refined measure of accuracy against a given ground truth, the PlasEval dissimilarity measure is useful to compare plasmid binning tools. Indeed, given two sets of predicted plasmid bins obtained with two different plasmid binning

tools, an analysis similar to the one we conducted in the previous section can quantify the differences in the results of both tools.

Table 2 shows that with the exception of HyAsP, the dissimilarity between the other tools in terms of the predicted plasmid bins is generally found to be below 0.5. Most dissimilarity scores between the predictions of any two of PlasBin-flow, MOB-recon and gplas tend be between 0.1 and 0.5. When comparing the results of these tools against HyAsP, the dissimilarity scores generally range between 0.4 and 0.7. Figure 4 also shows that a small tail of high dissimilarity scores can be observed for almost all the pairs of tools.

The two tools showing the largest agreement are PlasBin-flow and gplas. Joins (from PlasBin-flow to gplas) are the main source of dissimilarity between the two tools, indicating that plasmid bins from gplas tend to result from joining plasmid bins from PlasBin-flow. On the other hand, HyAsP predictions are highly dissimilar to those of other tools. This can be expected since PlasBin-flow, gplas and MOB-recon seem to have a relatively conservative approach to selecting plasmid contigs as compared to HyAsP. Compared against MOB-recon and HyAsP, PlasBin-flow excludes in its plasmid bins a large number of contigs that the other methods contain in their predictions. More interestingly, we observe that differences in plasmid bins are mostly due to plasmid bins of PlasBin-flow that are joined in MOB-recon. Last, when comparing MOB-recon and gplas, we observe that both methods differ in terms of contigs that appear only in bins of one method, followed by the fact that a significant number of plasmid bins from MOB-recon occur in a single gplas bin, while to the contrary, gplas bins do not exhibit such a feature. Figures 7 and 8 provide a refined illustration of the differences discussed above.

## Conclusion

In this work, we introduce a framework that enables to compare two sets of predicted plasmid bins. We provide a measure to quantify the disagreement between the two sets through a dissimilarity score, whose individual components (missing and extra contigs, splits, joins) provide both a quantitative and qualitative assessment of the dissimilarity between the two considered sets of plasmid bins. Moreover, the introduced dissimilarity measure accounts for important features such as contig length and copy number. This is especially important as common repeats, such as Insertion Sequences (IS) are known to be widely present in plasmids.

We applied this method for evaluating the predicted plasmids from four methods (PlasBin-flow, HyAsP, gplas and MOB-recon) on a data set of 53 *E. coli* samples for which the ground truth plasmid bins were obtained from hybrid assemblies. Our analysis shows that PlasEval provides a useful way to compare predicted plasmid bins either against a ground truth or between pairs of plasmid binning tools, that can inform either developers of novel plasmid binning methods or potential users having to decide on using a specific plasmid binning tool on a specific dataset. The results we obtained when analyzing the predictions of gplas, HyAsP, MOB-recon and PlasBin-flow show that the methods do behave significantly differently. This suggests that the problem of plasmid binning from short-reads data is still in need of further developments and we believe that PlasEval will be a useful tool toward this effort.

## Appendix A

### Proofs and additional lemma

***Proof of Lemma 1***

$$D_\alpha(\{P\}, \text{ref}(\mathrm{P}, \mathcal{B}')) = \left(\sum_{P_i \in \text{ref}(\mathrm{P}, \mathcal{B}')} L_\alpha(P_i)\right) - \left(\max_{P_i \in \text{ref}(\mathrm{P}, \mathcal{B}')} L_\alpha(P_i)\right)$$

It can be easily shown that $\text{ref}(\mathrm{P}, \mathcal{B}')$ is the optimal way to partition $P$. There no unique contigs between $P$ and $\text{ref}(\mathrm{P}, \mathcal{B}')$.

Say $P = P_1 \cup P_2 \cup ... \cup P_k$ such that $L_\alpha(P_1) \leq L_\alpha(P_2) \leq ... \leq L_\alpha(P_k)$.

While partitioning any set into two sets, the size of the smaller set counts towards the cost of the split. Thus, we repeatedly partition $P$ to obtain $\text{ref}(\mathrm{P}, \mathcal{B}')$ by choosing the smallest set in $\text{ref}(\mathrm{P}, \mathcal{B}')$ to split first.

$$\begin{aligned}
D_\alpha(\{P\}, \text{ref}(\mathrm{P}, \mathcal{B}')) &= \operatorname*{argmin}_{S \in \mathcal{S}(P, \text{ref}(\mathrm{P}, \mathcal{B}'))} c_\alpha(S) \\
D_\alpha(\{P\}, \text{ref}(\mathrm{P}, \mathcal{B}')) &= c_\alpha(P; P_1 \cup ... \cup P_{k-1}, P_k) \\
D_\alpha(\{P\}, \text{ref}(\mathrm{P}, \mathcal{B}')) &= c_\alpha(P; P_1 \cup P_2 ... \cup P_{k-1}) + L_\alpha(P_k)
\end{aligned}$$

Repeatedly removing the smallest set of $\text{ref}(\mathrm{P}, \mathcal{B}')$ available, we get

$$\begin{aligned}
D_\alpha(\{P\}, \text{ref}(\mathrm{P}, \mathcal{B}')) &= \sum_{2 \leq j \leq k} L_\alpha(P_j) \\
D_\alpha(\{P\}, \text{ref}(\mathrm{P}, \mathcal{B}')) &= \left(\sum_{P_i \in \text{ref}(\mathrm{P}, \mathcal{B}')} L_\alpha(P_i)\right) - L_\alpha(P_1)
\end{aligned}$$

Since, $P_1$ was the set in $\text{ref}(\mathrm{P}, \mathcal{B}')$ with the largest size, this proves the lemma. □

We now introduce a lemma that implies that, when comparing two sets of bins $\mathcal{A}, \mathcal{B}$, adding a contig $c$ of length $\ell$ to a bin $P \in \mathcal{A}$ and $Q \in \mathcal{B}$, will not result in either a decrease of the most parsimonious cost of the splits transforming $\mathcal{A}$ into $\text{ref}(\mathcal{A}', \mathcal{B}')$ (and, by symmetry between splits and joins, of the joins transforming $\mathcal{B}'$ $\text{ref}(\mathcal{A}', \mathcal{B}')$) or an increase of the cost of these splits (resp. joins) of more than $\ell^\alpha$. We will use this Lemma in the bounding function for the branch-and-bound algorithm we define to handle the case of repeated contigs.

**Lemma 2** *Let $P$ be a plasmid bin, $P_0, P_1, \ldots, P_k$ be such that $P_0 = \emptyset$ and $\cup_{i=0}^k P_i = P$. Let $c$ be a contig not in $P$, of length $\ell$. Let $P' = P \cup \{c\}$ and $P'_0, P'_1, \ldots, P'_k$ such that* (1) *there exists $j$ with $P'_j = P_j \cup \{c\}$ and* (2) *for any $i \neq j$ $P'_i = P_i$. Then*

$$D_\alpha(\{P\}, \{P_1, \ldots, P_k\}) \leq D_\alpha(\{P'\}, \{P'_0, \ldots, P'_k\}), \tag{A1}$$

$$D_\alpha(\{P'\}, \{P'_0, \ldots, P'_k\}) \leq D_\alpha(\{P\}, \{P_1, \ldots, P_k\}) + \ell^\alpha. \tag{A2}$$

***Proof*** First, as $P_0 = \emptyset$, it does not contribute anything to $D_\alpha(\{P\}, \{P_0, \ldots, P_k\})$, as $L(P_0) = 0$, which implies that $D_\alpha(\{P\}, \{P_1, \ldots, P_k\}) = D_\alpha(\{P\}, \{P_0, \ldots, P_k\})$.

For similar reasons, if $j \neq 0$, then $P'_0 = P_0 = \emptyset$ so $D_\alpha(\{P'\}, \{P'_1, \ldots, P'_k\}) = D_\alpha(\{P'\}, \{P'_0, \ldots, P'_k\})$; the only case where this does not hold is when $P'_0 = \{c\}$ (so $j = 0$).

Next, without loss of generality, assume that for any $i$, $L(P_i) \leq L(P_k)$, i.e. $P_k$ is the subset of $P$ of maximal cumulative length. It follows from Lemma 1 that

$$D_\alpha(\{P\}, \{P_0, \ldots, P_k\}) = \sum_{i=0}^{k-1} L(P_i)^\alpha.$$

We first assume that $L(P'_j) > L(P_k)$. Lemma 1 implies that

$$D_\alpha(\{P'\}, \{P'_0, \ldots, P'_k\}) = D_\alpha(\{P\}, \{P_0, \ldots, P_k\}) - L(P_j)^\alpha + L(P_k)^\alpha. \tag{A3}$$

As $L(P_k) < L(P_j) + \ell$ and $\alpha \geq 0$, we have that $L(P_k)^\alpha \leq (L(P_j) + \ell)^\alpha$. Moreover, as $\alpha \leq 1$, $(L(P_j) + \ell)^\alpha \leq L(P_j)^\alpha + \ell^\alpha$. This implies that $L(P_k)^\alpha < L(P_j)^\alpha + \ell^\alpha$, which, combined with (A3), implies (A2). Moreover, as $L(P_k) \geq L(P_j)$, $L(P_k)^\alpha \geq L(P_j)^\alpha$, which, combined with (A3), implies (A1).

We now assume that $L(P'_j) \leq L(P_k)$. Lemma 1 implies that

$$D_\alpha(\{P'\}, \{P'_0, \ldots, P'_k\}) = D_\alpha(\{P\}, \{P_0, \ldots, P_k\}) - L(P_j)^\alpha + L(P'_j)^\alpha. \tag{A4}$$

As $L(P'_j) = L(P_j) + \ell$, we have $L(P_j)^\alpha \leq L(P'_j)^\alpha \leq L(P_j)^\alpha + \ell^\alpha$ which, combined with (A4), implies (A2). Moreover, as $L(P'_j) > L(P_j)$, $L(P_k)^\alpha > L(P_j)^\alpha$, which, combined with (A4), implies (A1). □

## Appendix B

### Branch-and-bound algorithm

The branch-and-bound algorithm searches the space $\mathcal{M}(\mathcal{A}, \mathcal{B})$ of all matchings between the copies of contigs of $\mathcal{A}, \mathcal{B}$ to find a matching with the optimal dissimilarity score. It explores a search tree that is constructed by introducing a contig family containing repeats per level.

The root is at level 0, where we have $\mathcal{A}_0, \mathcal{B}_0$ consisting of only non-repeat contig families. As a result, the matching at this level is unambiguous. Each subsequent level introduces a contig family containing repeated contigs. For the family of a particular contig $c$, if $\mathcal{A}, \mathcal{B}$ have $k_1, k_2$ copies respectively, the total number of possible matchings for this family is $\binom{\max(k_1,k_2)}{\min(k_1,k_2)} \min(k_1, k_2)!$. Each edge from a search tree node at level $i-1$ to one at level $i$ represents a choice of a specific matching between the contig copies for the family at level $i$. Once a matching is chosen, we resolve the matching for the contig family $c$, while the remaining $|k_1 - k_2|$ copies of $c$ are removed as unique copies.

Note that before resolving the matching for a contig family at level $i$, the matchings for contig families at levels 1 to $i-1$ have already been resolved. As a result, each node of the

search tree depicts a unique matching of copies of contigs from families at levels 1 to $i-1$ as well as the contigs from non-repeat families from level 0. At each tree search node, after resolving the matching for all contig families at the node, we compute the $\mathrm{ref}(\mathrm{P}, \mathcal{B}')_{\mathrm{i}}$ for each $P \in \mathcal{A}'$. This is the refinement of $P$ according to bins in $\mathcal{B}'$ while only accounting for contig families encountered till level $i$. This helps in computing the splits to be made from $A'_i$. Similarly, we can compute the joins to be made from $\mathrm{ref}(\mathcal{A}', \mathcal{B}')_{\mathrm{i}}$ to $\mathcal{B}_i$.

**Algorithm 1** Branch and bound algorithm for transforming $\mathcal{A}$ to $\mathcal{B}$

```
function RECURSIVE_COMPARE(cost, i)
    if i < n then
        Generate M(A, B)_i for contig family c at level i
        for m ∈ M(A, B)_i do
            Resolve contig family according to matching m
            Remove extra copies of c, if any
            Compute cost of splits and joins for m
            if cost < mincost then
                Increase level by 1
                RECURSIVE_COMPARE(cost, i)
            else
                Decrease level by 1
            end if
        end for
    else
        Update mincost
    end if
end function
```

## Appendix C

### Data

See Fig. 5, and Table 3.

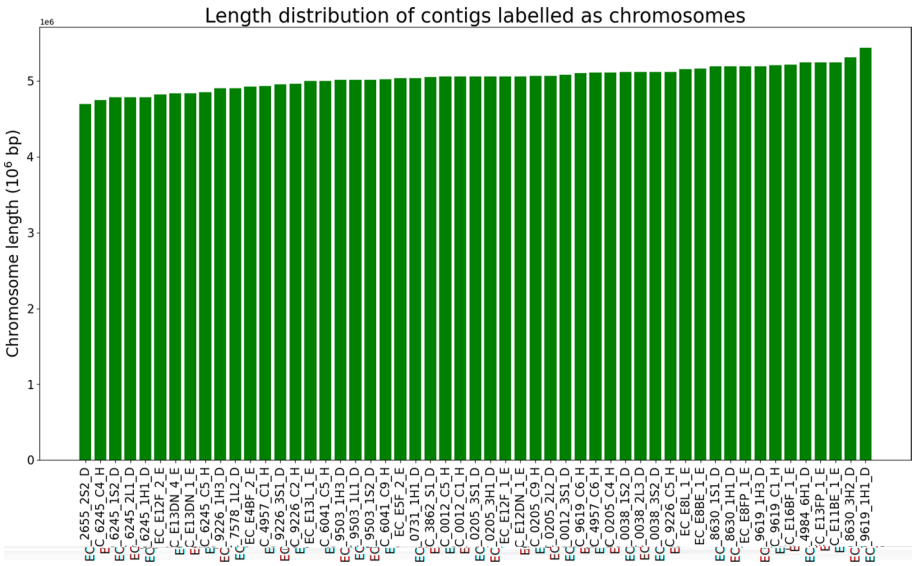


**Fig. 5** Lengths of hybrid contigs classified as chromosomes for 53 genomes in the data set; the lengths of these contigs are over 4,600,000 bp as is expected for the *E. coli* chromosome

**Table 3** Statistics on input assemblies: number of contigs of the short-read assembly, number of plasmids from the hybrid assembly, number of plasmids by length (4 columns), number of short-read contigs in ground truth plasmid bins, total length of plasmids

| Sample | Strain name | Contigs | Plasmids | < 5000bp | < 10000bp | < 50000bp | < 500000bp | Plasmid contigs | Total length |
|---|---|---|---|---|---|---|---|---|---|
| SAMN32247267 | EC_3862_S1_D | 291 | 4 | 1 | 1 | 0 | 2 | 51 | 202366 |
| SAMN32247268 | EC_8630_1H1_D | 303 | 4 | 2 | 1 | 0 | 1 | 58 | 203907 |
| SAMN32247272 | EC_6245_2L1_D | 216 | 4 | 1 | 1 | 0 | 2 | 61 | 224822 |
| SAMN32247290 | EC_6245_1S2_D | 233 | 4 | 1 | 1 | 0 | 2 | 64 | 225727 |
| SAMN32247291 | EC_0012_3S1_D | 450 | 9 | 6 | 2 | 0 | 1 | 35 | 146511 |
| SAMN32247292 | EC_6041_C9_H | 313 | 3 | 0 | 0 | 1 | 2 | 14 | 208398 |
| SAMN32247302 | EC_E13L_1_E | 312 | 2 | 1 | 0 | 0 | 1 | 26 | 205401 |
| SAMN32247303 | EC_9503_1S2_D | 223 | 1 | 0 | 0 | 0 | 1 | 36 | 194207 |
| SAMN32247305 | EC_E13DN_1_E | 262 | 2 | 0 | 0 | 0 | 2 | 50 | 194364 |
| SAMN32247308 | EC_9619_C6_H | 514 | 4 | 2 | 1 | 0 | 1 | 32 | 123799 |
| SAMN32247314 | EC_7578_1L2_D | 276 | 2 | 0 | 1 | 0 | 1 | 15 | 120057 |
| SAMN32247319 | EC_E8FP_1_E | 360 | 3 | 2 | 0 | 0 | 1 | 5 | 66076 |
| SAMN32247327 | EC_9226_C5_H | 478 | 6 | 0 | 3 | 1 | 2 | 105 | 273835 |
| SAMN32247333 | EC_E11BE_1_E | 511 | 3 | 2 | 0 | 0 | 1 | 6 | 61764 |
| SAMN32247335 | EC_0205_C9_H | 510 | 6 | 1 | 2 | 1 | 2 | 33 | 196797 |
| SAMN32247345 | EC_4957_C1_H | 198 | 2 | 0 | 1 | 0 | 1 | 10 | 127370 |
| SAMN32247347 | EC_0205_3H1_D | 333 | 5 | 1 | 1 | 1 | 2 | 29 | 294670 |
| SAMN32247348 | EC_0205_C4_H | 327 | 4 | 1 | 1 | 0 | 2 | 18 | 174136 |
| SAMN32247353 | EC_0012_C5_H | 402 | 5 | 2 | 2 | 0 | 1 | 36 | 108981 |
| SAMN32247354 | EC_4984_6H1_D | 484 | 7 | 5 | 1 | 0 | 1 | 30 | 144761 |
| SAMN32247363 | EC_E4BF_2_E | 257 | 2 | 0 | 0 | 1 | 1 | 18 | 154263 |
| SAMN32247365 | EC_9226_C2_H | 346 | 4 | 0 | 2 | 0 | 2 | 54 | 306919 |
| SAMN32247396 | EC_0205_3S1_D | 250 | 5 | 1 | 1 | 1 | 2 | 18 | 294670 |
| SAMN32247397 | EC_6245_C4_H | 314 | 3 | 0 | 1 | 0 | 2 | 50 | 304555 |
| SAMN32247399 | EC_0038_2L3_D | 227 | 2 | 0 | 1 | 0 | 1 | 22 | 139664 |
| SAMN32247400 | EC_9226_1H3_D | 181 | 3 | 2 | 0 | 0 | 1 | 30 | 139815 |

**Table 3** (continued)

| Sample | Strain name | Contigs | Plasmids | < 5000bp | < 10000bp | < 50000bp | < 500000bp | Plasmid contigs | Total length |
|---|---|---|---|---|---|---|---|---|---|
| SAMN32247404 | EC_E13FP_1_E | 587 | 5 | 3 | 0 | 0 | 2 | 39 | 200739 |
| SAMN32247411 | EC_8630_3H2_D | 373 | 1 | 0 | 0 | 0 | 1 | 11 | 117668 |
| SAMN32247425 | EC_9619_1H1_D | 990 | 1 | 0 | 0 | 0 | 1 | 95 | 119252 |
| SAMN32247430 | EC_0012_C1_H | 406 | 5 | 2 | 2 | 0 | 1 | 36 | 108981 |
| SAMN32247441 | EC_0205_2L2_D | 511 | 9 | 5 | 3 | 0 | 1 | 32 | 173761 |
| SAMN32247444 | EC_9226_3S1_D | 244 | 5 | 3 | 0 | 0 | 2 | 61 | 254617 |
| SAMN32247449 | EC_E12F_1_E | 227 | 2 | 0 | 1 | 0 | 1 | 4 | 76380 |
| SAMN32247466 | EC_0038_1S2_D | 213 | 2 | 0 | 1 | 0 | 1 | 22 | 139707 |
| SAMN32247471 | EC_E5F_2_E | 294 | 4 | 1 | 1 | 1 | 1 | 28 | 131642 |
| SAMN32247475 | EC_E12DN_1_E | 228 | 2 | 0 | 1 | 0 | 1 | 4 | 76371 |
| SAMN32247476 | EC_6245_C5_H | 208 | 2 | 0 | 1 | 0 | 1 | 4 | 104630 |
| SAMN32247480 | EC_8630_1S1_D | 316 | 4 | 2 | 1 | 0 | 1 | 60 | 203907 |
| SAMN32247483 | EC_E8L_1_E | 334 | 3 | 1 | 0 | 0 | 2 | 10 | 124500 |
| SAMN32247488 | EC_E12F_2_E | 264 | 1 | 0 | 0 | 0 | 1 | 1 | 100424 |
| SAMN32247493 | EC_9503_1H3_D | 198 | 1 | 0 | 0 | 0 | 1 | 27 | 194207 |
| SAMN32247494 | EC_6041_C5_H | 289 | 3 | 1 | 0 | 1 | 1 | 7 | 105863 |
| SAMN32247496 | EC_0731_1H1_D | 443 | 8 | 3 | 2 | 0 | 3 | 38 | 386853 |
| SAMN32247505 | EC_9503_1L1_D | 242 | 1 | 0 | 0 | 0 | 1 | 36 | 194207 |
| SAMN32247508 | EC_0038_3S2_D | 231 | 2 | 0 | 1 | 0 | 1 | 22 | 139707 |
| SAMN32247512 | EC_E13DN_4_E | 274 | 2 | 0 | 0 | 0 | 2 | 48 | 194364 |
| SAMN32247515 | EC_E16BF_1_E | 440 | 2 | 1 | 0 | 0 | 1 | 64 | 198070 |
| SAMN32247518 | EC_6245_1H1_D | 248 | 4 | 1 | 1 | 0 | 2 | 66 | 224822 |
| SAMN32247519 | EC_2655_2S2_D | 214 | 5 | 4 | 0 | 0 | 1 | 31 | 219001 |
| SAMN32247520 | EC_E8BE_1_E | 401 | 4 | 2 | 0 | 0 | 2 | 9 | 126082 |
| SAMN32247522 | EC_4957_C6_H | 354 | 2 | 0 | 1 | 0 | 1 | 24 | 122870 |
| SAMN32247523 | EC_9619_1H3_D | 389 | 4 | 2 | 1 | 0 | 1 | 10 | 73336 |
| SAMN32247534 | EC_9619_C1_H | 591 | 4 | 1 | 1 | 0 | 2 | 17 | 173547 |

## Appendix D

### Tool evaluations using PlasEval

See Fig. 6, and Table 4.

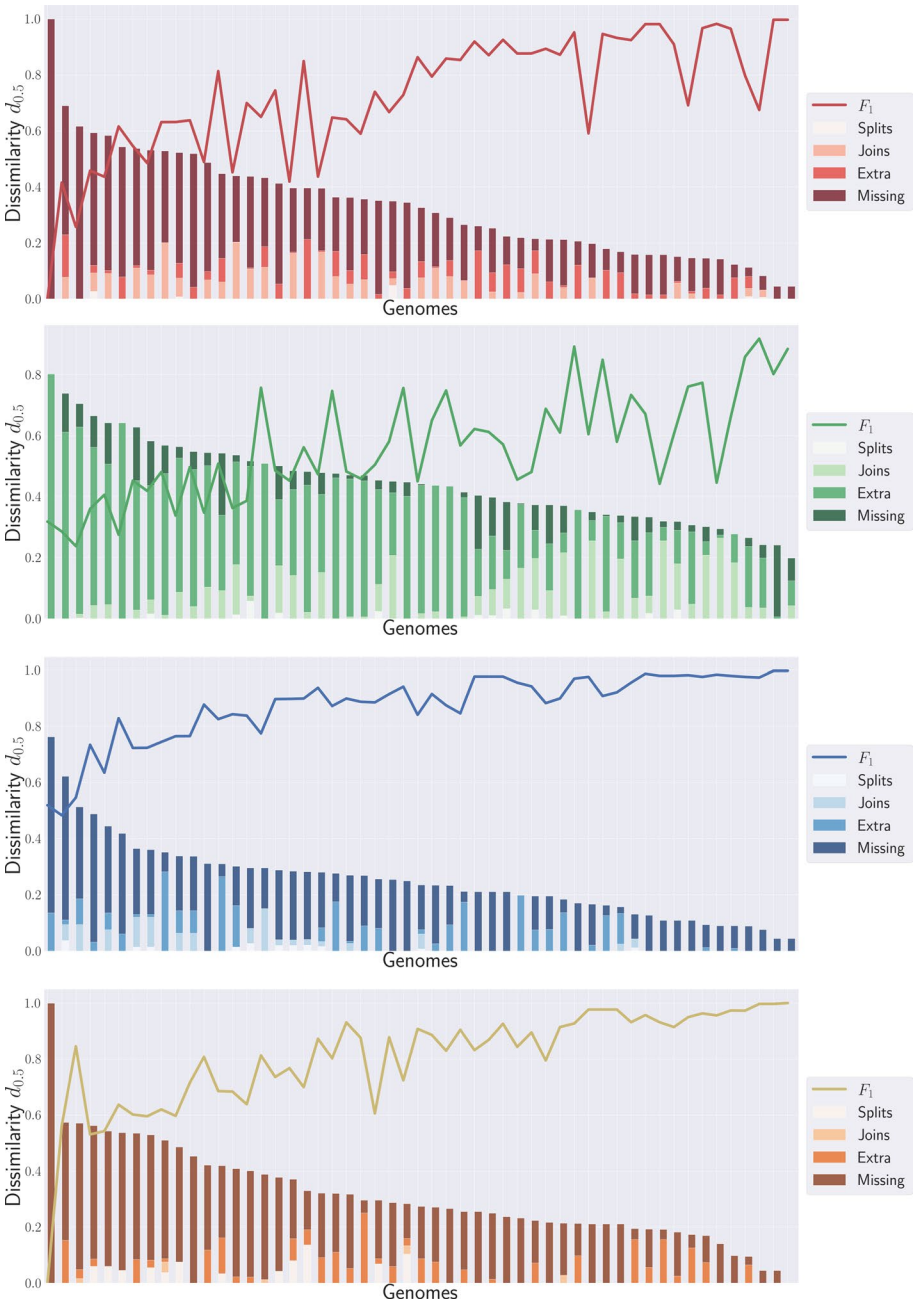


**Fig. 6** PlasEval score and F1 score for all samples and all tools. Genomes on the x-axis are sorted in decreasing order of dissimilarity with respect to the reference. From top to bottom: PlasBin-flow, HyAsP, MOB-recon, gplas

**Table 4** The $F_1$-score and the dissimilarity score subtracted from 1, computed for the predictions of the 53 E. coli samples considered in this analysis

| | PlasBin-flow | | HyAsP | | gplas | | MOB-recon | |
|---|---|---|---|---|---|---|---|---|
| **Sample** | $1 - d_{0.5}$ | $F_1$ | $1 - d_{0.5}$ | $F_1$ | $1 - d_{0.5}$ | $F_1$ | $1 - d_{0.5}$ | $F_1$ |
| SAMN32247267 | 0.8884 | 0.7977 | 0.4637 | 0.3624 | 0.622 | 0.7355 | 0.7047 | 0.838 |
| SAMN32247268 | 0.4569 | 0.6154 | 0.5153 | 0.452 | 0.4654 | 0.6015 | 0.7441 | 0.885 |
| SAMN32247272 | 0.4631 | 0.5456 | 0.6819 | 0.6043 | 0.4378 | 0.5305 | 0.7178 | 0.8987 |
| SAMN32247290 | 0.6516 | 0.6667 | 0.6176 | 0.5709 | 0.4570 | 0.5421 | 0.7125 | 0.8968 |
| SAMN32247291 | 0.782 | 0.8763 | 0.4581 | 0.5077 | 0.7506 | 0.8684 | 0.6983 | 0.8429 |
| SAMN32247292 | 0.8551 | 0.9667 | 0.5634 | 0.649 | 0.8312 | 0.963 | 0.8046 | 0.942 |
| SAMN32247302 | 0.6495 | 0.7391 | 0.65 | 0.604 | 0.7347 | 0.8295 | 0.9101 | 0.9833 |
| SAMN32247303 | 0.5606 | 0.4518 | 0.7058 | 0.4454 | 0.7687 | 0.8434 | 0.8912 | 0.9792 |
| SAMN32247305 | 0.4683 | 0.485 | 0.5957 | 0.6213 | 0.5467 | 0.715 | 0.6354 | 0.7225 |
| SAMN32247308 | 0.5873 | 0.7439 | 0.6655 | 0.7328 | 0.6806 | 0.802 | 0.4874 | 0.5459 |
| SAMN32247314 | 0.8416 | 0.9242 | 0.5497 | 0.5803 | 0.7451 | 0.9046 | 0.7663 | 0.9148 |
| SAMN32247319 | 0.7405 | 0.919 | 0.3589 | 0.2746 | 0.8057 | 0.9315 | 0.7883 | 0.846 |
| SAMN32247327 | 0.9179 | 0.6742 | 0.4994 | 0.4849 | 0.5138 | 0.5966 | 0.3782 | 0.4829 |
| SAMN32247333 | 0.6043 | 0.8492 | 0.5658 | 0.7473 | 0.7043 | 0.876 | 0.6902 | 0.8254 |
| SAMN32247335 | 0.7475 | 0.8702 | 0.4522 | 0.4969 | 0.6795 | 0.8724 | 0.7459 | 0.9144 |
| SAMN32247345 | 0.8775 | 0.9647 | 0.6992 | 0.773 | 0.8076 | 0.9569 | 0.7049 | 0.7741 |
| SAMN32247347 | 0.6932 | 0.7933 | 0.6937 | 0.7604 | 0.8599 | 0.9557 | 0.9117 | 0.9755 |
| SAMN32247348 | 0.8211 | 0.9461 | 0.2961 | 0.2372 | 0.6709 | 0.6992 | 0.8332 | 0.9756 |
| SAMN32247353 | 0.5674 | 0.6495 | 0.4561 | 0.3474 | 0.7130 | 0.8778 | 0.6626 | 0.7646 |
| SAMN32247354 | 0.656 | 0.7282 | 0.518 | 0.5612 | 0.5805 | 0.6852 | 0.7239 | 0.8721 |
| SAMN32247363 | 0.7352 | 0.8533 | 0.802 | 0.8838 | 0.7865 | 0.9142 | 0.6890 | 0.8772 |
| SAMN32247365 | 0.5134 | 0.4885 | 0.6268 | 0.6874 | 0.5993 | 0.6381 | 0.8292 | 0.9693 |
| | **PlasBin-flow** | | **HyAsP** | | **gplas** | | **MOB-recon** | |
| **Sample** | $1 - d_{0.5}$ | $F_1$ | $1 - d_{0.5}$ | $F_1$ | $1 - d_{0.5}$ | $F_1$ | $1 - d_{0.5}$ | $F_1$ |
| SAMN32247396 | 0.8495 | 0.9094 | 0.735 | 0.8575 | 0.9022 | 0.9733 | 0.9069 | 0.9755 |
| SAMN32247397 | 0.4072 | 0.4577 | 0.6294 | 0.6096 | 0.4901 | 0.6197 | 0.7651 | 0.841 |
| SAMN32247399 | 0.8429 | 0.9811 | 0.5292 | 0.482 | 0.7894 | 0.9767 | 0.7894 | 0.9767 |
| SAMN32247400 | 0.4716 | 0.6309 | 0.758 | 0.9174 | 0.7272 | 0.9077 | 0.7505 | 0.9409 |
| SAMN32247404 | 0.6742 | 0.8631 | 0.4179 | 0.4187 | 0.6835 | 0.9307 | 0.5812 | 0.8287 |
| SAMN32247411 | 0.7876 | 0.8932 | 0.3353 | 0.3598 | 0.7773 | 0.895 | 0.6485 | 0.7439 |
| SAMN32247425 | 0.4815 | 0.6373 | 0.3725 | 0.4521 | 0.4295 | 0.8454 | 0.2379 | 0.5187 |
| SAMN32247430 | 0.637 | 0.6473 | 0.4365 | 0.3377 | 0.7297 | 0.8862 | 0.6622 | 0.7643 |
| SAMN32247441 | 0.7884 | 0.8718 | 0.5467 | 0.5039 | 0.8182 | 0.9142 | 0.8046 | 0.8824 |
| SAMN32247444 | 0.8034 | 0.5908 | 0.7229 | 0.6599 | 0.6118 | 0.8124 | 0.73 | 0.8988 |
| SAMN32247449 | 0.9558 | 0.9967 | 0.4919 | 0.7567 | 0.9558 | 0.9967 | 0.9558 | 0.9967 |
| SAMN32247466 | 0.8429 | 0.9811 | 0.5852 | 0.5673 | 0.7893 | 0.9767 | 0.7893 | 0.9767 |
| SAMN32247471 | 0.5525 | 0.8128 | 0.4325 | 0.4806 | 0.6293 | 0.767 | 0.7195 | 0.9369 |
| SAMN32247475 | 0.9558 | 0.9967 | 0.6426 | 0.8914 | 0.9558 | 0.9967 | 0.9558 | 0.9967 |
| SAMN32247476 | 0.3841 | 0.2565 | 0.5527 | 0.7559 | 1.0 | 1.0 | 0.9238 | 0.9732 |
| SAMN32247480 | 0.3108 | 0.4152 | 0.522 | 0.4727 | 0.4267 | 0.5608 | 0.7317 | 0.8873 |
| SAMN32247483 | 0.7858 | 0.8765 | 0.5591 | 0.4498 | 0.827 | 0.9494 | 0.8164 | 0.899 |
| SAMN32247488 | 0.0 | 0.0 | 0.1979 | 0.3176 | 0.0 | 0.0 | 0.8016 | 0.9551 |
| SAMN32247493 | 0.6047 | 0.4363 | 0.6613 | 0.5793 | 0.7841 | 0.7945 | 0.8912 | 0.9814 |
| SAMN32247494 | 0.7941 | 0.9517 | 0.2624 | 0.2852 | 0.905 | 0.9726 | 0.8432 | 0.9206 |
| SAMN32247496 | 0.8542 | 0.6909 | 0.6672 | 0.6704 | 0.7050 | 0.6054 | 0.8696 | 0.9551 |
| SAMN32247505 | 0.6038 | 0.4187 | 0.6814 | 0.4417 | 0.7454 | 0.8318 | 0.8912 | 0.9792 |

**Table 4** (continued)

| Sample | PlasBin-flow | | HyAsP | | gplas | | MOB-recon | |
|---|---|---|---|---|---|---|---|---|
| | $1 - d_{0.5}$ | $F_1$ | $1 - d_{0.5}$ | $F_1$ | $1 - d_{0.5}$ | $F_1$ | $1 - d_{0.5}$ | $F_1$ |
| SAMN32247508 | 0.8583 | 0.9822 | 0.5306 | 0.4588 | 0.7894 | 0.9767 | 0.7894 | 0.9767 |
| SAMN32247512 | 0.4170 | 0.4358 | 0.6017 | 0.6115 | 0.4630 | 0.6366 | 0.6386 | 0.7229 |
| SAMN32247515 | 0.6432 | 0.5894 | 0.3585 | 0.4058 | 0.5789 | 0.8073 | 0.5555 | 0.6345 |
| SAMN32247518 | 0.478 | 0.6312 | 0.6267 | 0.4804 | 0.4709 | 0.5952 | 0.716 | 0.8974 |
| SAMN32247519 | 0.6376 | 0.641 | 0.6215 | 0.4553 | 0.5919 | 0.6834 | 0.911 | 0.979 |
| SAMN32247520 | 0.7768 | 0.9252 | 0.6586 | 0.8484 | 0.8090 | 0.9314 | 0.8375 | 0.9076 |
| SAMN32247522 | 0.5629 | 0.6989 | 0.7587 | 0.8014 | 0.7639 | 0.9261 | 0.5118 | 0.7337 |
| SAMN32247523 | 0.8322 | 0.932 | 0.524 | 0.7459 | 0.7876 | 0.9268 | 0.8729 | 0.9867 |
| SAMN32247534 | 0.7105 | 0.8582 | 0.4834 | 0.386 | 0.7167 | 0.7237 | 0.7667 | 0.8746 |

## Appendix E

### Tool versus tool comparisons

Figures 7 and 8 provide further details on the differences between the binning predictions of PlasBin-flow, MOB-recon, gplas and HyAsP.

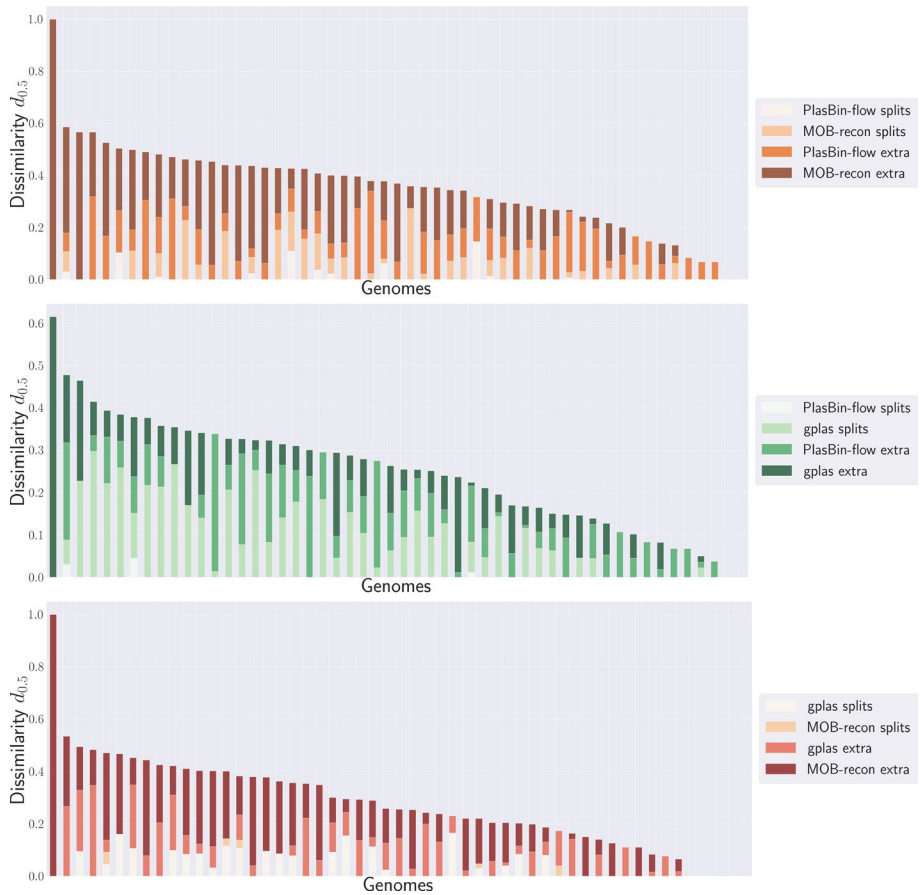


**Fig. 7** Pairwise comparisons between PlasBin-flow, MOB-recon and gplas

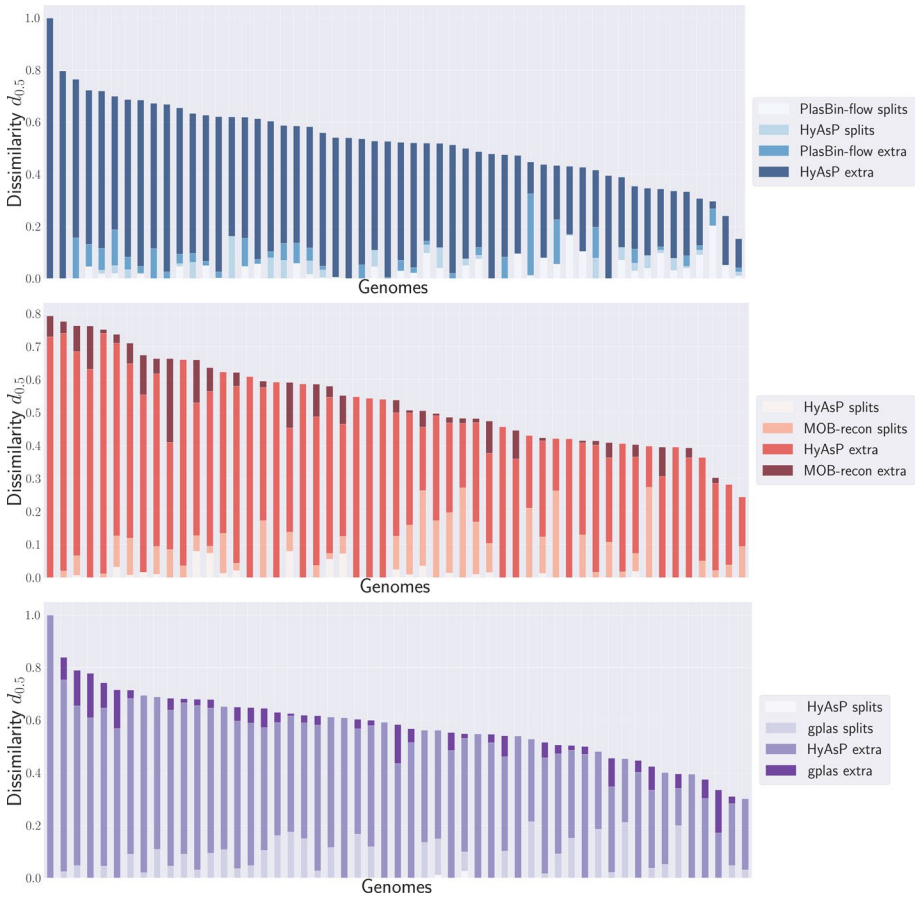


**Fig. 8** Pairwise comparisons between HyAsP and PlasBin-flow, MOB-recon and gplas

## Appendix F

### Running time and function calls

| Comparison | Mean time (s) | Max time (s) | Mean # calls | Max # calls |
|---|---|---|---|---|
| PlasBin-flow - Ground truth | 0.18 | 6.81 | 545 | 19673 |
| HyAsP - Ground truth | 1.14 | 52.63 | 980 | 40105 |
| Gplas - Ground truth | 0.002 | 0.02 | 9 | 24 |
| MOB-recon - Ground truth | 0.003 | 0.02 | 11 | 28 |
| PlasBin-flow - HyAsP | 0.43 | 15.28 | 658 | 24459 |
| PlasBin-flow - Gplas | 0.005 | 0.1 | 22 | 400 |
| PlasBin-flow - MOB-recon | 0.27 | 1.28 | 95 | 4368 |
| HyAsP - Gplas | 0.003 | 0.009 | 10 | 23 |
| HyAsP - MOB-recon | 0.004 | 0.024 | 11 | 28 |
| Gplas - MOB-recon | 0.001 | 0.01 | 8 | 16 |

**Author contributions**
AM and CC designed the method and experiments. AM implemented the method and performed the experiments. HS, AW and RZ provided the ***E. coli*** data used in this paper. AM and CC analyzed the data. AM, HS, RZ, RB and CC contributed to the manuscript.

**Funding**
This research was funded by Genome Canada grant ARETE (Antimicrobial Resistance: Emergence, Transmission, and Ecology) to RB, and an NSERC Discovery Grant to CC.

**Availability of data and materials**
The hybrid assemblies and short-read assemblies are available at https://zenodo.org/records/10785150, and at NCBI in the BioProject PRJNA912639. The ground truth plasmid bins and the results from the four plasmid binning tools are also available at https://zenodo.org/records/10785150.
PlasEval is publicly available at https://github.com/acme92/PlasEval and has been programmed in Python.

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

**Competing interests**
The authors declare that they have no competing interests.

Received: 16 May 2024 Accepted: 19 September 2024
Published online: 26 November 2024

**References**

1. Partridge SR, Kwong SM, Firth N, Jensen SO. Mobile genetic elements associated with antimicrobial resistance. Clin Microbiol Rev. 2018. https://doi.org/10.1128/cmr.00088-17.
2. De Oliveira DMP, Forde BM, Kidd TJ, Harris PNA, Schembri MA, Beatson SA, Paterson DL, Walker MJ. Antimicrobial resistance in ESKAPE pathogens. Clin Microbiol Rev. 2020. https://doi.org/10.1128/cmr.00181-19.
3. ...Sanderson H, Gray KL, Manuele A, Maguire F, Khan A, Liu C, Navanekere Rudrappa C, Nash JHE, Robertson J, Bessonov K, Oloni M, Alcock BP, Raphenya AR, McAllister TA, Peacock SJ, Raven KE, Gouliouris T, McArthur AG, Brinkman FSL, Fink RC, Zaheer R, Beiko RG. Exploring the mobilome and resistome of Enterococcus faecium in a One Health context across two continents. Microb Genom. 2022. https://doi.org/10.1099/mgen.0.000880.
4. Arredondo-Alonso S, Willems RJ, Schaik W, Schürch AC. On the (im)possibility of reconstructing plasmids from whole-genome short-read sequencing data. Microb Genom. 2017. https://doi.org/10.1099/mgen.0.000128.
5. Antipov D, Hartwick N, Shen M, Raiko M, Lapidus A, Pevzner PA. plasmidSPAdes: assembling plasmids from whole genome sequencing data. Bioinformatics. 2016;32(22):3380–7. https://doi.org/10.1093/bioinformatics/btw493.
6. Robertson J, Nash J. MOB-suite: software tools for clustering, reconstruction and typing of plasmids from draft assemblies. Microb Genom. 2018. https://doi.org/10.1099/mgen.0.000206.
7. Müller R, Chauve C. HyAsP, a greedy tool for plasmids identification. Bioinformatics. 2019;35(21):4436–9. https://doi.org/10.1093/bioinformatics/btz413.
8. Arredondo-Alonso S, Bootsma M, Hein Y, Rogers MRC, Corander J, Willems RJL, Schürch AC. gplas: a comprehensive tool for plasmid analysis using short-read graphs. Bioinformatics. 2020;36(12):3874–6. https://doi.org/10.1093/bioinformatics/btaa233.
9. Mane A, Faizrahnemoon M, Chauve C (2022) A mixed integer linear programming algorithm for plasmid binning. In: Comparative Genomics: 19th International Conference, RECOMB-CG 2022, La Jolla, CA, USA, May 20-21, 2022, Proceedings. pp 279–292. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-031-06220-9_16
10. Mane A, Faizrahnemoon M, Vinař T, Brejová B, Chauve C. PlasBin-flow: a flow-based MILP algorithm for plasmid contigs binning. Bioinformatics. 2023;39(Supplement–1):288–96. https://doi.org/10.1093/bioinformatics/btad250.
11. Chen Z, Erickson DL, Meng J. Benchmarking hybrid assembly approaches for genomic analyses of bacterial pathogens using Illumina and Oxford Nanopore sequencing. BMC Genom. 2020. https://doi.org/10.1186/s12864-020-07041-8.
12. Khezri A, Avershina E, Ahmad R. Hybrid assembly provides improved resolution of plasmids, antimicrobial resistance genes, and virulence factors in escherichia coli and klebsiella pneumoniae clinical isolates. Microorganisms. 2021;9(12):2560. https://doi.org/10.3390/microorganisms9122560.
13. Bouras G, Sheppard AE, Mallawaarachchi V, Vreugde S. Plassembler: an automated bacterial plasmid assembly tool. Bioinformatics. 2023. https://doi.org/10.1093/bioinformatics/btad409.
14. Sielemann J, Sielemann K, Brejová B, Vinař T, Chauve C. plASgraph2: using graph neural networks to detect plasmid contigs from an assembly graph. Front Microbiol. 2023. https://doi.org/10.3389/fmicb.2023.1267695.
15. Mikheenko A, Prjibelski A, Saveliev V, Antipov D, Gurevich A. Versatile genome assembly evaluation with QUAST-LG. Bioinformatics. 2018;34(13):142–50. https://doi.org/10.1093/bioinformatics/bty266.

16. Shintani M, Sanchez ZK, Kimbara K. Genomics of microbial plasmids: classification and identification based on replication and transfer systems and host taxonomy. Front Microbiol. 2015. https://doi.org/10.3389/fmicb.2015.00242.
17. Johnson J, Soehnlen M, Blankenship HM. Long read genome assemblers struggle with small plasmids. Microb Genom. 2023. https://doi.org/10.1099/mgen.0.001024.
18. Dusadeepong R, Delvallez G, Cheng S, Meng S, Sreng N, Letchford J, Choun K, Teav S, Hardy L, Jacobs J, Hoang T, Seemann T, Howden BP, Glaser P, Stinear TP, Vandelannoote K. Phylogenomic investigation of an outbreak of fluoroquinolone-resistant salmonella enterica subsp. enterica serovar paratyphi a in phnom penh, cambodia. Microb Genom. 2023. https://doi.org/10.1099/mgen.0.000972.
19. Arredondo-Alonso S, Rogers MRC, Braat JC, Verschuuren TD, Top J, Corander J, Willems RJL, Schürch AC. mlplasmids: a user-friendly tool to predict plasmid- and chromosome-derived sequences for single species. Microb Genom. 2018;4(11): 000224. https://doi.org/10.1099/mgen.0.000224.
20. Andreopoulos WB, Geller AM, Lucke M, Balewski J, Clum A, Ivanova NN, Levy A. Deeplasmid: deep learning accurately separates plasmids from bacterial chromosomes. Nucleic Acids Res. 2021;50(3):17–17. https://doi.org/10.1093/nar/gkab1115.
21. Graaf-van Bloois L, Wagenaar JA, Zomer AL. Rfplasmid: predicting plasmid sequences from short-read assembly data using machine learning. Microb Genom. 2021. https://doi.org/10.1099/mgen.0.000683.
22. Pradier L, Tissot T, Fiston-Lavier A, Bedhomme S. Plasforest: a homology-based random forest classifier for plasmid detection in genomic datasets. BMC Bioinform. 2021;22(1):349. https://doi.org/10.1186/S12859-021-04270-W.
23. Rozov R, Brown Kav A, Bogumil D, Shterzer N, Halperin E, Mizrahi I, Shamir R. Recycler: an algorithm for detecting plasmids from de novo assembly graphs. Bioinformatics. 2016;33(4):475–82. https://doi.org/10.1093/bioinformatics/btw651.
24. Ferretti V, Nadeau JH, Sankoff D (1996) Original synteny. In: Combinatorial Pattern Matching. Lecture Notes in Computer Science, pp 159–167. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-61258-0_13
25. Dasgupta B, Jiang T, Kannan S, Li M, Sweedyk E. On the complexity and approximation of syntenic distance. Discret Appl Math. 1998;88(1–3):59–82. https://doi.org/10.1016/s0166-218x(98)00066-3.
26. Sanderson H, Nnajide C, McCarthy M, Singh R, Rubin J, Dillon J-A, White A. Hybrid genome assemblies of 245 avian and broiler barn environment-associated escherichia coli strains isolated from saskatchewan broiler farms. Microbiol Resour Announc. 2023;12:0011023. https://doi.org/10.1128/mra.00110-23.
27. Wick RR, Judd LM, Gorrie CL, Holt KE. Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. PLoS Comput Biol. 2017;13(6):1–22. https://doi.org/10.1371/journal.pcbi.1005595.
28. Andrews S (2010) FastQC: a quality control tool for high throughput sequence data; http://www.bioinformatics.babraham.ac.uk/projects/fastqc
29. Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one fastq preprocessor. Bioinfotmatics. 2018;34(17):884–90. https://doi.org/10.1093/bioinformatics/bty560.
30. Gurevich A, Saveliev V, Vyahhi G. Nikolayand Tesler: Quast: quality assessment tool for genome assemblies. Bioinfotmatics. 2013;29(8):1072–5. https://doi.org/10.1093/bioinformatics/btt086.
31. Camacho C, Couluris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL. BLAST+: architecture and applications. BMC Bioinf. 2009;10:421. https://doi.org/10.1186/1471-2105-10-421.

## Publisher's Note