*Bioinformatics*
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Genome Analysis

# HyAsP, a hybrid assembler for plasmids

**Robert Müller [1] and Cedric Chauve [2,*]**

[1]Computational Methods for the Analysis of the Diversity and Dynamics of Genomes, Bielefeld University, 33615 Bielefeld, Germany;
[2]Department of Mathematics, Simon Fraser University, Vancouver BC V5A 1S6, Canada

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

## Abstract

**Motivation:** Plasmids are ubiquituous in bacterial genomes, and have been shown to be involved in important evolutionary processes, in particular the acquisition of antimicrobial resistance through horizontal gene transfer. However separating chromosomal contigs from plasmid contigs and assembling the later is still a challenging bioinformatices problem.
**Results:** We introduce HyAsP, a tool that identifies and assemble plasmid contigs following a hybrid approach based on a database of known plasmids and a greedy assembly algorithm. We test HyAsP on a large sample of bacterial data sets and observe that it generally outperforms other tools.
**Availability:** https://github.com/cchauve/HyAsP
**Contact:** cedric.chauve@sfu.ca
**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 Introduction

Plasmids are extra-chromosomal DNA molecules common – but not limited to – Bacteria and Archaea. Plasmids differ from chromosomes in various features, such as their length – they tend to be much shorter than chromosomes –, their copy number – plasmids can be present in multiple copies in a cell – and their GC content. Plasmids play an important part in horizontal gene transfer and, thus, in the transmission of virulence factors and antibiotic resistance (Dolejska and Papagiannitsis, 2018; Carattoli, 2013). Therefore, the effective identification of plasmids from bacterial samples is important to the development of mitigation strategies against the proliferation of drug-resistant bacteria.

Over the years, various approaches have been explored for the detection and classification of plasmids. Replicon and MOB typing schemes (Carattoli *et al.*, 2005; Garcillán-Barcia *et al.*, 2009) that rely on genes encoding replication resp. mobility functions (e.g. PlasmidFinder and pMLST (Carattoli *et al.*, 2014)) have been widely used but are also limited in scope and resolution (Fricke *et al.*, 2009). Other tools such as cBar (Zhou and Xu, 2010) and PlasFlow (Krawczyk *et al.*, 2018) aim to extract plasmids sequences from metagenomics data. Whole-genome sequencing (WGS) is becoming a standard approach in microbial genomics, including in a clinical context, but the identification of plasmids from assembly graphs remains a difficult task. Consequently, there is a high demand for *in silico* methods extracting plasmids from WGS assemblies and several tools have been recently developed. PLACNET (Lanza *et al.*, 2014) uses information from the assembly (e.g. scaffold links and read depth), reference sequences and plasmid-diagnostic sequence features, but also needs a subsequent expert analysis. Other tools such as Recycler (Rozov *et al.*, 2017) and plasmidSPAdes (Antipov *et al.*, 2016) do not depend on reference sequences and are unsupervised. Recycler predicts plasmids by repeatedly peeling off cycles of the assembly graph having a sufficiently low read depth variation and of minimum based on read-depth and length features. plasmidSPAdes assumes that the read depth of plasmids differs from the one of the chromosome, estimates the chromosomal read depth from the assembly graph, removes those contigs that are presumably of chromosomal origin and predicts plasmids from the connected components of this reduced assembly graph. The recent reference-based tool MOB-recon (Robertson and Nash, 2018) uses a database of reference plasmids and collections of known replicons and relaxases. Contigs of an assembly are mapped against the reference database and grouped into putative plasmid units, that are further refined by moving circular sequences into their own units and discarding those units without a replicon or relaxase. A broad review and benchmarks of existing methods are available in (Orlek *et al.*, 2017; Arredondo-Alonso *et al.*, 2017; Laczny *et al.*, 2017).

In this article, we present HyAsP, a novel tool for extracting plasmids from WGS assemblies in a fully automatic way. It combines ideas from both reference-based and depth-based methods to identify plasmids using information on the occurrences of known plasmid genes and considering characteristics of the contigs such as read depth and GC content. We compared the prediction quality of HyAsP with plasmidSPAdes and

 

MOB-recon on a data set comprising 147 plasmids and show that our new greedy algorithm generally outperformed the other tools.

## 2 Methods

### 2.1 The HyAsP algorithm

HyAsP is a greedy algorithm for the reconstruction of plasmids from an assembly graph using information from known plasmid genes, read depth and GC content. It combines idea of reference-based methods such as MOB-recon and assembly-based methods such as plasmidSPAdes. We provide below a high level description of our algorithm and experiments, while all technical details are provided in Supplement Material.

*Identifying seeds.* Our algorithm starts from an assembly graph (obtained by Unicycler by default), and first identifies *seed contigs* (or simply *seeds*), defined as contigs which contain at least one plasmid gene (according to a given database of known plasmid genes). Seeds are used to identify parts of the assembly graph that serve as starting points to construct chains of contigs potentially defining plasmid fragments. However, contigs marked as seeds can also be of chromosomal origin as some genes occur both in plasmids and chromosomes. Such dubious seeds might be identified as, e.g., very long contigs with only a few gene hits, therefore, only seeds which satisfy some *eligibility* criteria – considering their gene density (proportion of a sequence which participates in at least one plasmid gene), read depth and length – are used to start a new plasmid.

*Greedy extension.* Next, seeds are enumerated based on their plasmid gene density and GC content, preferring those with a high gene density and a GC content differing from the average GC content of the assembly graph, and each seed is extended into a contig chain following a greedy approach. To extend the current contig chain, its two endpoints are searched for *eligible extensions*, using eligibility criteria aimed at averting likely errors by, e.g., limiting the length of the plasmid and avoiding overly long gene-free stretches as well as fluctuations in GC content that are too high. The process is guided by the structure of the assembly graph, similarities in read depth and GC content, and the plasmids gene density. The eligible extensions are scored based on these features and the extension with the best score is chosen.

*Updating read depth.* Once all eligible extensions have been performed, the resulting contig chain is considered as a potential plasmid fragment; it is then checked whether it can be circularised, if its first and last contigs are the same or overlap sufficiently. The construction of a plasmid fconcludes by determining a final read-depth value for each contig it contains. To this end, the average read depth (by default the mean) is computed and a contig is assigned the minimum of this average and its current read depth in the assembly graph. If a contig had a larger reads depth than the one it is assigned in the current plasmid, it is kept in the assembly graph for being potentially used by other plasmids, after having decreased its depth by the value assigned to it. This feature is motivated by the possible repetition of contigs in different plasmids.

*Postprocessing.* Once all plasmids are constructed, those which do not meet some quality requirements, determined from a large collection of known plasmid, are marked as *questionable* and separated from the other ones, called *putative* plasmids. Finally, the (putative) plasmids are grouped into bins based on their circularity, read depth and GC content. Each circular plasmid is put into its own bin. Non-circular plasmids are grouped together if both their read depth and GC content differ at most a user-specified number of standard deviations of the respective characteristic (calculated from the plasmids).

*Integrated pipeline.* Moreover, we incorporated the HyAsP algorithm into a pipeline accepting FASTQ reads and a collection of known plasmid genes as input. The reads are preprocessed (Trim Galore (Krueger, 2016), sickle (Joshi and Fass, 2011)), assembled using Unicycler (Wick *et al.*, 2017) and, subsequently, the genes are mapped to the assembly contigs using BLAST. Finally, the plasmids are predicted in the assembly graph using HyAsP as described above.

HyAsP is a single-threaded program written completely in Python (requiring Biopython, NumPy and pandas) with system calls to BLAST (blastn, makeblastdb). After downloading HyAsP, it can be used directly or installed as a package through pip. It is delivered with instructions how to construct gene databases, but other tools such as BLAST or Unicycler have to installed independently.

### 2.2 Experimental design

We evaluated HyAsP on a collection of real plasmids originally compiled for benchmarking MOB-suite (Robertson and Nash, 2018). We compared HyAsP with two other fully automatic tools for plasmid reconstruction, plasmidSPAdes and MOB-recon. They represent the two approaches that are combined in our tool: on the one hand, pure assembly guided by characteristics such as read depth and GC content and, on the other hand, identifying plasmid-like contigs using known plasmid sequences. Recycler was not included in the comparison as it performed notably worse than plasmidSPAdes in previous benchmarks and deals only with circular plasmids.

The MOB-suite benchmarking data set consists of 133 bacterial samples comprising the same number of closed genomes and 377 plasmids. In order to simulate the use of plasmids identification tools on a newly WGS dataset, using only knowledge from previously existing plasmid data, the data set was split in half based on the release date of the corresponding read data. The 66 samples released on 19 December 2015 or later formed the *test samples*, spanning 147 plasmids from 8 different species, that we call the emphground truth plasmids. The other 67 samples were used as the first set of *database samples* and we refer to the 230 plasmids and 10685 genes from them also as the *MOB-database*. Similarly, we created a second set of references by collecting all plasmids available from NCBI that were released before 19 December 2015 (5826 plasmids and 382881 genes) that we call the *NCBI-database*.

We assessed the predictions of all tools on each test sample in terms of their *precision* (proportion of predicted plasmids corresponding to expected references) and *recall* (proportion of expected reference plasmids corresponding to predictions) by mapping the putative plasmids (HyAsP) resp. plasmid contigs (MOB-recon, plasmidSPAdes) against the ground truth plasmids using BLAST. Precision and recall were then summarized with the *F1 score* per test sample. In addition, the total precision, recall and F1 score over all test samples (per species) were determined by summing over the predicted plasmids of all test samples (per species).

## 3 Results

In the following, we show overall and aggregated metrics assessing the predictions based on the NCBI-database. More results, including an inspection of the predictions per test sample and the analysis based on the MOB-database, are provided in Supplementary Material.

As shown in Table 1, HyAsP outperformed plasmidSPAdes and MOB-recon in both total precision and recall. Consequently, the F1 score of HyAsP was notably higher as well. In addition, plasmidSPAdes and MOB-recon showed a notable trade-off between precision and recall.

We also observed a trade-off in precision and recall between the MOB-database and the NCBI-database: HyAsP attained a higher precision

| | Tool | Precision | Recall | F1 score |
|---|---|---|---|---|
| NCBI | HyAsP | 0.871445 | 0.898822 | 0.884922 |
| | plasmidSPAdes | 0.659211 | 0.741983 | 0.698152 |
| | MOB-recon | 0.760241 | 0.583909 | 0.660509 |
| MOB | HyAsP | 0.934515 | 0.775169 | 0.847416 |
| | plasmidSPAdes | 0.659211 | 0.741983 | 0.698152 |
| | MOB-recon | 0.760241 | 0.583909 | 0.660509 |

Table 1. Aggregated precision, recall and F1 score of HyAsP, plasmidSPAdes and MOB-recon, over the 66 test samples using the NCBI-database and MOB-database, respectively.

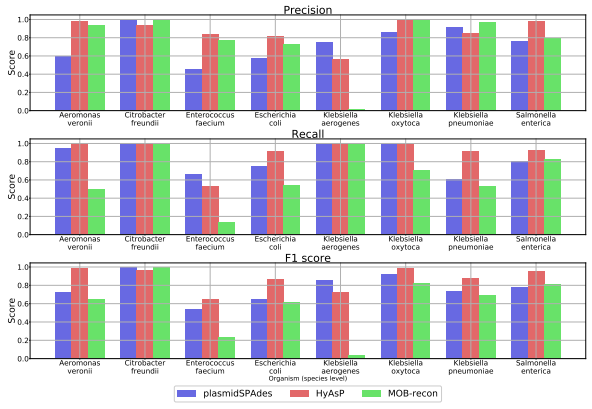


**Fig. 1.** Precision, recall and F1 score per species of HyAsP, plasmidSPAdes and MOB-recon on the test samples using the NCBI-database.

(0.934515) and lower recall (0.775169) using the less extensive MOB-database, which led to a similar but slightly lower F1 score (0.847416).

However, note that we excluded from the reference sequences three *Salmonella enterica* plasmids from the NCBI-database, which exhibited unusually chromosome-like characteristics. These plasmids (resp. their genes) caused a severe drop in precision on test samples from *Salmonella enterica* but had hardly any effect on other test samples. We refer to Section 2.1 of Supplementary Material for more information on this issue.

Subsequently, we analyzed the predictions after grouping the samples based on the associated organism at the species level. Fig. 1 shows the results for the 8 different species included in the test samples. HyAsP performed similarly well or better than plasmidSPAdes and MOB-recon for the majority of species. Only for *Klebsiella aerogenes*, did it fall notably behind plasmidSPAdes but still outperformed MOB-recon. While there are species, for which the other tools achieved a higher recall or precision, the trade-off between both is usually smaller for HyAsP leading to a better overall prediction quality (in terms of the F1 score).

## 4 Conclusion

We introduced a new method, HyAsP, for the reconstruction of plasmids using an hybrid approach combining reference plasmids and an assembly graph. Our results showed that this combination can improve the quality of plasmid prediction. Future developments could focus on the careful design of the references database – especially related to genes shared by plasmids and chromosomes in specific species –, alternatives to the greedy paradigm to construct plasmid contigs chains and the inclusion of long reads data.

## Funding

This work is funded by the International DFG Research Training Group GRK 1906/1. CC acknowledges the support of NSERC (Discovery Grant RGPIN-2017-03986) and of ComputeCanada.

## References

Antipov, D. *et al.* (2016). plasmidSPAdes: assembling plasmids from whole genome sequencing data. *Bioinformatics*, **32**(22), 3380–3387.

Arredondo-Alonso, S. *et al.* (2017). On the (im)possibility of reconstructing plasmids from whole-genome short-read sequencing data. *Microbial Genomics*, **3**(10). doi: 10.1099/mgen.0.000128.

Carattoli, A. (2013). Plasmids and the spread of resistance. *International Journal of Medical Microbiology*, **303**(6-7), 298–304.

Carattoli, A. *et al.* (2005). Identification of plasmids by PCR-based replicon typing. *Journal of Microbiological Methods*, **63**(3), 219 – 228.

Carattoli, A. *et al.* (2014). In silico detection and typing of plasmids using plasmidfinder and plasmid multilocus sequence typing. *Antimicrobial Agents and Chemotherapy*, **58**(7), 3895–3903.

Dolejska, M. and Papagiannitsis, C. C. (2018). Plasmid-mediated resistance is going wild. *Plasmids*, **In press**. doi: j.plasmid.2018.09.010.

Fricke, W. F. *et al.* (2009). Comparative Genomics of the IncA/C Multidrug Resistance Plasmid Family. *Journal of Bacteriology*, **191**(15), 4750–4757.

Garcillán-Barcia, M. P. *et al.* (2009). The diversity of conjugative relaxases and its application in plasmid classification. *FEMS Microbiology Reviews*, **33**(3), 657–687.

Joshi, N. and Fass, J. (2011). sickle: A sliding-window, adaptive, quality-based trimming tool for fastq files. Software available at https://github.com/najoshi/sickle.

Krawczyk, P. S. *et al.* (2018). PlasFlow: predicting plasmid sequences in metagenomic data using genome signatures. *Nucleic Acids Research*, **46**(6), e35.

Krueger, F. (2016). Trim Galore. Software available at https://github.com/FelixKrueger/TrimGalore.

Laczny, C. C. *et al.* (2017). Assessing the heterogeneity of in silico plasmid predictions based on whole-genome-sequenced clinical isolates. *Briefings in Bioinformatics*, page bbx162.

Lanza, V. F. *et al.* (2014). Plasmid Flux in Escherichia coli ST131 Sublineages, Analyzed by Plasmid Constellation Network (PLACNET), a New Method for Plasmid Reconstruction from Whole Genome Sequences. *PLOS Genetics*, **10**(12), 1–21.

Orlek, A. *et al.* (2017). Plasmid Classification in an Era of Whole-Genome Sequencing: Application in Studies of Antibiotic Resistance Epidemiology. *Frontiers in Microbiology*, **8**, 182.

Robertson, J. and Nash, J. H. E. (2018). MOB-suite: software tools for clustering, reconstruction and typing of plasmids from draft assemblies. *Microbial Genomics*, **4**(8). doi: 10.1099/mgen.0.000206.

Rozov, R. *et al.* (2017). Recycler: an algorithm for detecting plasmids from de novo assembly graphs. *Bioinformatics*, **33**(4), 475–482.

Wick, R. R. *et al.* (2017). Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. *PLOS Computational Biology*, **13**(6), 1–22.

Zhou, F. and Xu, Y. (2010). cBar: a computer program to distinguish plasmid-derived from chromosome-derived sequence fragments in metagenomics data. *Bioinformatics*, **26**(16), 2051–2052.