Bioinformatics, 36(12), 2020, 3874–3876
doi: 10.1093/bioinformatics/btaa233
Advance Access Publication Date: 9 April 2020
Applications Note

Genome analysis

gplas: a comprehensive tool for plasmid analysis using short-read graphs

Sergio Arredondo-Alonso[1], Martin Bootsma[2,3], Yaïr Hein[3], Malbert R. C. Rogers[1], Jukka Corander[4,5,6], Rob J. L. Willems[1] and Anita C. Schürch[1,*]

[1]Department of Medical Microbiology, University Medical Center Utrecht, Utrecht University, 3584 CX Utrecht, The Netherlands, [2]Department of Epidemiology, Julius Center for Health Sciences and Primary Care of the UMC Utrecht, [3]Department of Mathematics, Faculty of Sciences, Utrecht University, 3584 CX Utrecht, The Netherlands, [4]Department of Parasites and Microbes, Wellcome Sanger Institute, Hinxton, Saffron Walden CB10 1RQ, UK, [5]Department of Biostatistics, University of Oslo, 0317 Oslo, Norway and [6]Department of Mathematics and Statistics, Helsinki Institute of Information Technology (HIIT), University of Helsinki, FI-00014 Helsinki, Finland

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on November 11, 2019; revised on January 27, 2020; editorial decision on March 27, 2020; accepted on April 2, 2020

Abstract

Summary: Plasmids can horizontally transmit genetic traits, enabling rapid bacterial adaptation to new environments and hosts. Short-read whole-genome sequencing data are often applied to large-scale bacterial comparative genomics projects but the reconstruction of plasmids from these data is facing severe limitations, such as the inability to distinguish plasmids from each other in a bacterial genome. We developed gplas, a new approach to reliably separate plasmid contigs into discrete components using sequence composition, coverage, assembly graph information and network partitioning based on a pruned network of plasmid unitigs. Gplas facilitates the analysis of large numbers of bacterial isolates and allows a detailed analysis of plasmid epidemiology based solely on short-read sequence data.

Availability and implementation: Gplas is written in R, Bash and uses a Snakemake pipeline as a workflow management system. Gplas is available under the GNU General Public License v3.0 at https://gitlab.com/sirarredondo/gplas.git.

Contact: a.c.schurch@umcutrecht.nl

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

A single bacterial cell can harbor several distinct plasmids; however, current plasmid prediction tools from short-read WGS often have a binary outcome (plasmid or chromosome). To bin predicted plasmids into discrete entities, we built a new method based on the following concepts: (i) contigs of the same plasmid have a uniform sequence coverage (Antipov *et al.*, 2016; Rozov *et al.*, 2016), (ii) plasmid paths in the assembly graph can be searched for using a greedy approach (Müller and Chauve, 2019) and (iii) removal of repeat units from the plasmid graphs disconnects the graph into independent components (Vielva *et al.*, 2017).

Here, we refined these ideas and introduce the concept of unitigs co-occurrence to create a pruned plasmidome network. Using an unsupervised approach, the network is queried to find highly connected nodes corresponding to sequences belonging to the same discrete plasmid unit, representing a single plasmid. We show that our approach outperforms other *de novo* and reference-based tools and fully automates the reconstruction of plasmids from short reads.

2 Materials and methods

2.1 Gplas algorithm

Given a short-read assembly graph (gfa format), segments (nodes) and edges (links) are extracted from the graph. Gplas uses mlplasmids (version 1.0.0, prediction threshold = 0.5) or plasflow (version 1.1, prediction threshold = 0.7) to classify segments as plasmid- or chromosome-derived and selects segments with an in- and out-degree of 1 (unitigs) (Arredondo-Alonso *et al.*, 2018; Krawczyk *et al.*, 2018). The *k*-mer coverage SD of the chromosome-derived unitigs is computed to quantify the fluctuation in the coverage of segments belonging to the same replicon unit. Plasmid-derived unitigs are considered to search for plasmid walks with a similar coverage and composition using a greedy approach (Supplementary Methods S1). Gplas creates a plasmidome network (undirected graph) in which nodes correspond to plasmid unitigs and edges are created and weighted based on the co-existence of the nodes in the solution space of the computed walks. Modularity values computed using a selection of partitioning algorithms (Blondel *et al.*, 2008; Newman, 2006; Pons and Latapy, 2005) are considered to perform

Algorithm 1 Gplas pseudocode

Data: Graph G from SPAdes or Unicycler
Result: Plasmidome network $G_{\mathcal{P}}$. Assignment of plasmid nodes $N_{\mathcal{P}}$ into different bins
Initialization;
Extract nodes N and links L from G;
Divide N as collection of plasmid-derived nodes $\mathcal{P}$ and chromosome-derived nodes $\mathcal{C}$ using mlplasmids or plasflow;
Discard $\mathcal{P}$ and $\mathcal{C}$ with an $d^i(v)$ and $d^o(v)$!= 1 and length < 1 kbp;
Determine the $s^2_{\mathcal{C}}$ of $\mathcal{C}$ based on the k-mer coverage;
for *each* $v_0 \in \mathcal{P}$ **do**
  Search through all the possible plasmid-like walks W starting from v_0;
  for *W in number of walks* **do**
    while $\exists$ *eligible extension E(W)* **do**
      Consider the last v in W
      Retrieve all candidate extensions $E(W)$
      Compute gplas scores g(W,v) of $E(W)$
      Filter $E(W)$ *with a g(W,v)* $< \xi$ *(default = 0.1, tunable by the user)*
      Sample a $E(W)$ *based on the vector g(W,v)*
      Extension of W using the selected v
    end
    Create a new set of links $L_{\mathcal{P}}$ *connecting* $N_{\mathcal{P}}$ *in W;*
    Reinitialize W considering again v_0 *as first element;*
  end
end
Compute the weights $H_{\mathcal{P}}$ of $L_{\mathcal{P}}$ based on their frequency in W;
Create a novel plasmidome network $G_{\mathcal{P}}(N_{\mathcal{P}}, L_{\mathcal{P}}, H_{\mathcal{P}})$;
Consider components (subgraphs) $G^i_{\mathcal{P}}$ from $G_{\mathcal{P}}$;
for *each* $G^i_{\mathcal{P}}$ *with* $N^i_{\mathcal{P}} > 1$ **do**
  Compute modularity values Q from $G^i_{\mathcal{P}}$ using three partitioning algorithms ;
  Consider all $Q > 0.2$ (tunable by the user) to split $G^i_{\mathcal{P}}$ and perform a voting decision ;
  Predict $N^i_{\mathcal{P}}$ as a single bin or classify $N^i_{\mathcal{P}}$ into bins based on the partitioning algorithm with a highest Q;
end
Classification of $N^i_{\mathcal{P}}$ in $G^i_{\mathcal{P}}$ with $N^i_{\mathcal{P}} = 1$ as singletons;
Plot $G_{\mathcal{P}}$ with colours according to bin classification;

Algorithm 1: Gplas pseudocode

a voting decision regarding the split of the components into different bins (subcomponents) in the undirected network (Supplementary Methods S1). These bins represent the set of plasmids present in the bacterial isolate and are plotted in the plasmidome network using igraph R package (Csardi *et al.*, 2006). The pseudocode and formalization of the algorithm are available in Algorithm 1 and Supplementary Methods S1, respectively.

2.2 Benchmarking dataset

Gplas was benchmarked against current existing tools to bin plasmid contigs from short-read WGS: (i) plasmidSPAdes (*de novo*-based approach, version 3.12) (Antipov *et al.*, 2016), (ii) mob-recon (reference-based approach, version 1.4.9.1) (Robertson and Nash, 2018) and (iii) hyasp (hybrid approach, version 1.0.0) (Müller and Chauve, 2019). To evaluate the binning tools, we selected a set of 28 genomes with short- and long-read WGS available including 106 plasmids from 9 different bacterial species, which were not present in the databases or training sets of the tools (Supplementary Methods S3 and Table S1) (Arredondo-Alonso *et al.*, 2020; De Maio *et al.*, 2019; Decano *et al.*, 2019; Wick *et al.*, 2017).

Let n_{bin} be the total number of nodes present in the predicted bin and define ref as the reference replicon sequence with a highest number of nodes in each bin. Let n_{ref} be the total number of nodes comprised in ref. We then define two metrics commonly used in metagenomics for binning evaluation: (i) precision and (ii) completeness (Supplementary Methods S4).

$$\text{precision} = \frac{n_{\text{bin}} \in n_{\text{ref}}}{n_{\text{bin}}}$$

$$\text{completeness} = \frac{n_{\text{bin}} \in n_{\text{ref}}}{n_{\text{ref}}}.$$

Table 1. Gplas benchmarking

Tool	Precision	Completeness	Bin size
gplas–mlplasmids	0.88/0.82[a]	0.79/0.72[a]	6.02/10.9[a]
gplas–plasflow	0.62/0.45[a]	0.52/0.32[a]	7.17/11.1[a]
hyasp	0.64/0.56[a]	0.36/0.30[a]	3.84/5.65[a]
mob-recon	0.79/0.71[a]	0.56/0.51[a]	3.4/7.22[a]
plasmidSPAdes	0.52/0.27[a]	0.56/0.38[a]	6.99/13.7[a]

[a]Components >1 node.

3 Results

Gplas in combination with mlplasmids obtained the highest average precision (0.88) indicating that the predicted components were mostly formed by nodes belonging to the same discrete plasmid unit (Table 1 and Supplementary Fig. S1). The reported average completeness value (0.79) showed that most of the nodes from a single plasmid were recovered as a discrete plasmid bin by gplas (Table 1 and Supplementary Fig. S2). We observed a decline in the performance of gplas in combination with mlplasmids (precision = 0.82, completeness = 0.72) when considering uniquely bins with a size larger than one which indicated merging problems of large plasmids with a similar *k*-mer coverage (Supplementary Fig. S3 and Results S2). However, in all cases, the performance of gplas in combination with mlplasmids performed better than other *de-novo* and reference-based tools tested here (Table 1). To show the potential of gplas in combination with mlplasmids, we showcase the performance of our approach in two distinct bacterial isolates (Supplementary Results S1 and S2).

Mlplasmids only contains a limited range of species models (Supplementary Methods). For other bacterial species, we observed that plasflow probabilities in combination with gplas performed similar than the other *de-novo* approaches but also introduced bias when wrongly predicting chromosome contigs as plasmid nodes (Table 1 and Supplementary Fig. S1), thereby creating bins corresponding to chromosome and plasmid chimeras (precision = 0.62).

4 Discussion

We present a new tool called gplas, which enables the binning and a detailed analysis workflow of binary classified plasmid contigs into discrete plasmid units by relying on the structure of the assembly graph, *k*-mer information and partitioning of a pruned plasmidome network. A limitation of the presented approach is the generation of chimeras resulting from plasmids with similar *k*-mer profiles, *k*-mer coverage and sharing repeat unit(s), such as a transposase or an IS element. These cases cannot be unambiguously solved. Here, we integrated and extended upon features to predict plasmid sequences

and exploit the information present in short-read graphs to automate the reconstruction of plasmids.

Acknowledgements

We would like to thank Dr Bryan Wee for testing and contributing to the development of gplas.

Funding

This work was supported by the Joint Programming Initiative in Antimicrobial Resistance [JPIAMR Third call, STARCS, JPIAMR2016-AC16/00039 to S.A.-A. and R.J.L.W.]. It was also funded by the European Research Council [grant number 742158 to J.C.].

Conflict of Interest: none declared.

References

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

Arredondo-Alonso,S. *et al*. (2018) mlplasmids: a user-friendly tool to predict plasmid- and chromosome-derived sequences for single species. *Microb. Genom.*, **4**, **e000224**.

Arredondo-Alonso,S. *et al*. (2020) Plasmids shaped the recent emergence of the major nosocomial pathogen enterococcus faecium. *mBio* **11**, e03284-19.

Blondel,V.D. *et al*. (2008) Fast unfolding of communities in large networks. *J. Stat. Mech.*, **2008**, P10008.

Csardi,G. *et al*. (2006) The igraph software package for complex network research. *InterJ. Complex Syst.*, **1695**, 1–9.

De Maio,N. *et al*.; on behalf of the REHAB Consortium. (2019) Comparison of long-read sequencing technologies in the hybrid assembly of complex bacterial genomes. *Microb. Genom.*, **5**, **e000294**.

Decano,A.G. *et al*. (2019) Complete assembly of *Escherichia coli* sequence type 131 genomes using long reads demonstrates antibiotic resistance gene variation within diverse plasmid and chromosomal contexts. *mSphere*, **4**, e00130.

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

Müller,R. and Chauve,C. (2019) HyAsP, a greedy tool for plasmids identification. *Bioinformatics*, **35**, 4436–4439.

Newman,M.E.J. (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, **74**, 036104.

Pons,P. and Latapy,M. (2005) Computing communities in large networks using random walks. *Computer and Information Sciences – ISCIS* **3733**, 284–293.

Robertson,J. and Nash,J.H.E. (2018) MOB-suite: software tools for clustering, reconstruction and typing of plasmids from draft assemblies. *Microb. Genom.*, **4**, e000206.

Rozov,R. *et al*. (2016) Recycler: an algorithm for detecting plasmids from de novo assembly graphs. *Bioinformatics*, **33**, 475–482.

Vielva,L. *et al*. (2017) PLACNETw: a web-based tool for plasmid reconstruction from bacterial genomes. *Bioinformatics*, **33**, 3796–3798.

Wick,R.R. *et al*. (2017) Completing bacterial genome assemblies with multiplex MinION sequencing. *Microb. Genom.*, **3**, e000132.