Pellow *et al. Microbiome* (2021) 9:144
https://doi.org/10.1186/s40168-021-01068-z

Microbiome

SOFTWARE ARTICLE Open Access

SCAPP: an algorithm for improved plasmid assembly in metagenomes

David Pellow[1*], Alvah Zorea[2], Maraike Probst[3], Ori Furman[2], Arik Segal[4,5], Itzhak Mizrahi[2] and Ron Shamir[1]

Abstract

Background: Metagenomic sequencing has led to the identification and assembly of many new bacterial genome sequences. These bacteria often contain plasmids: usually small, circular double-stranded DNA molecules that may transfer across bacterial species and confer antibiotic resistance. These plasmids are generally less studied and understood than their bacterial hosts. Part of the reason for this is insufficient computational tools enabling the analysis of plasmids in metagenomic samples.

Results: We developed SCAPP (Sequence Contents-Aware Plasmid Peeler)—an algorithm and tool to assemble plasmid sequences from metagenomic sequencing. SCAPP builds on some key ideas from the Recycler algorithm while improving plasmid assemblies by integrating biological knowledge about plasmids.
We compared the performance of SCAPP to Recycler and metaplasmidSPAdes on simulated metagenomes, real human gut microbiome samples, and a human gut plasmidome dataset that we generated. We also created plasmidome and metagenome data from the same cow rumen sample and used the parallel sequencing data to create a novel assessment procedure. Overall, SCAPP outperformed Recycler and metaplasmidSPAdes across this wide range of datasets.

Conclusions: SCAPP is an easy to use Python package that enables the assembly of full plasmid sequences from metagenomic samples. It outperformed existing metagenomic plasmid assemblers in most cases and assembled novel and clinically relevant plasmids in samples we generated such as a human gut plasmidome. SCAPP is open-source software available from: https://github.com/Shamir-Lab/SCAPP.

Keywords: Plasmids, Assembly

Background

Plasmids play a critical role in microbial adaptation, such as antibiotic resistance or other metabolic capabilities, and genome diversification through horizontal gene transfer. However, plasmid evolution and ecology across different microbial environments and populations are poorly characterized and understood. Thousands of plasmids have been sequenced and assembled directly from isolated bacteria, but constructing complete plasmid sequences from short read data remains a hard challenge. The task of assembling plasmid sequences from shotgun metagenomic sequences, which is our goal here, is even more daunting.

There are several reasons for the difficulty of plasmid assembly. First, plasmids represent a very small fraction of the sample's DNA and thus may not be fully covered by the read data in high-throughput sequencing experiments. Second, they often share sequences with the bacterial genomes and with other plasmids, resulting in tangled assembly graphs. For these reasons, plasmids assembled from bacterial isolates are usually incomplete, fragmented into multiple contigs, and contaminated with sequences from other sources. The challenge is reflected in the title of a recent review on the topic: "On the (im)possibility of

*Correspondence: dpellow@post.tau.ac.il
[1]Blavatnik School of Computer Science, Tel Aviv University, 6997801 Tel Aviv, Israel
Full list of author information is available at the end of the article

reconstructing plasmids from whole-genome short-read sequencing data" [1]. In a metagenomic sample, these problems are amplified since the assembly graphs are much larger, more tangled, and fragmented.

There are a number of tools that can be used to detect plasmid sequences including PlasmidFinder [2], cBar [3], gPlas [4], PlasFlow [5], and others. There is also the plasmidSPAdes assembler for assembling plasmids in isolate samples [6]. However, there are currently only two tools that attempt to reconstruct complete plasmid sequences in metagenomic samples: Recycler [7] and metaplasmidSPAdes [8] (mpSpades). mpSpades iteratively generates smaller and smaller subgraphs of the assembly graph by removing contigs with coverage below a threshold that increases in each iteration. As lower coverage segments of the graph are removed, longer contigs may be constructed in the remaining subgraph. Cyclic contigs are considered as putative plasmids and then verified using the profile of their genetic contents. The main idea behind Recycler is that a single shortest circular path through each node in the assembly graph can be found efficiently. The circular paths that have uniform read coverage are iteratively "peeled" off the graph and reported as possible plasmids. The peeling process reduces the residual coverage of each involved node, or removes it altogether. We note that these tools, as well as our work, focus on circular plasmids and do not assemble linear plasmid sequences.

Here we present SCAPP (Sequence Contents-Aware Plasmid Peeler), a new algorithm that uses the peeling idea of Recycler and also leverages external biological knowledge about plasmid sequences. In SCAPP, the assembly graph is annotated with plasmid-specific genes (PSGs) and nodes are assigned weights reflecting the chance that they are plasmidic based on a plasmid sequence classifier [9]. In the annotated assembly graph, we prioritize peeling off circular paths that include plasmid genes and highly probable plasmid sequences. SCAPP also uses the PSGs and plasmid scores to filter out likely false positives from the set of potential plasmids.

Algorithm 1 SCAPP pipeline

Input: Assembly graph $G = (V, E)$ and read set R of the sample
Output: P: potential plasmids, O: confident plasmid predictions

Create annotated graph $G' = (V', E')$:
Initially $G' = G$
Map R to V'
$score(v) \leftarrow$ sequence plasmid probability $\forall v \in V'$
$w(v) = (1 - score(v))/(len(v) \cdot cov(v))\ \forall v \in V'$
$V^m = \{v \in V' | v$ contains a PSG$\}$, $w(v) = 0\ \forall v \in V^m$
$V' \leftarrow V' \setminus \{v \in V' |\, deg(v) = 0 \vee v$ is probable chromosome node
$\vee\ v$ is a non-compatible self-loop with $indeg(v) = outdeg(v) = 1\}$
$P \leftarrow \{v \in V' | v$ is a compatible self-loop$\}$
for each strongly connected component $CC \in G'$ **do**
    for $v \in V^m \cap CC$ in decreasing order by $len(v) \cdot cov(v)$ **do**
        Find lowest weight cycle C through v
        if C meets coverage and paired-end read criteria **then**
            $P \leftarrow P \cup \{C\}$, $G' \leftarrow peel(G', C)$
    for $v \in \{v \in CC |\, v$ is a probable plasmid node$\}$ in decreasing order by $len(v) \cdot cov(v)$ **do**
        Find lowest weight cycle C through v
        if C meets coverage and paired-end read criteria **then**
            $P \leftarrow P \cup \{C\}$, $G' \leftarrow peel(G', C)$
    while V' changes **do**
        $S \leftarrow \{\}$
        for $v \in V' \cap CC$ in decreasing order by $len(v) \cdot cov(v)$ **do**
            Find lowest weight cycle C through v
            $S \leftarrow S \cup C$
        for $C \in S$ in increasing order of coefficient of variation **do**
            if C meets coverage and paired-end read criteria **then**
                $P \leftarrow P \cup \{C\}$, $G' \leftarrow peel(G', C)$
$O \leftarrow \{C \in P | (C$ contains a PSG $\wedge$ *plasmid score*$(C) > 0.5)$
$\vee\ (C$ contains a PSG $\wedge\ C$ is self-loop $) \vee$ (*plasmid score*$(C) > 0.5 \wedge C$ is self-loop$)\}$

We tested SCAPP on both simulated and diverse real metagenomic data and compared its performance to Recycler and mpSpades. Overall, SCAPP performed better than the other tools across these datasets. SCAPP has higher precision than Recycler in all cases, meaning it more accurately constructs correct plasmids from the sequencing data. SCAPP also has higher recall than mpSpades in most cases, and higher precision in most of the real datasets. We developed and tested a novel strategy given parallel plasmidome and metagenome sequencing of the same sample. We show how to accurately assess the performance of the tools on metagenome data, even in the absence of known reference plasmids.

Implementation

SCAPP accepts as input a metagenomic assembly graph, with nodes representing the sequences of assembled contigs and edges representing k-long sequence overlaps between contigs, and the paired-end reads from which the graph was assembled. SCAPP processes each component of the assembly graph and iteratively assembles plasmids from them. The output of SCAPP is a set of cyclic sequences representing confident plasmid assemblies.

A high-level overview of SCAPP is provided in Table 1 and depicted graphically in Fig. 1; the full algorithmic details are presented below. For brevity, we describe only default parameters below; see Additional file 1, Section S1 for alternatives.

SCAPP is available from https://github.com/Shamir-Lab/SCAPP and fully documented there. It was written in Python3 and can be installed as a conda package, directly from Bioconda or from its sources.

The SCAPP algorithm

The full SCAPP algorithm is given in Algorithm 1. The peel function, which defines how cycles are peeled from the graph, is given in Algorithm 2.

Algorithm 2 $peel(G, C)$

Input: Assembly graph $G = (V, E)$ annotated with node coverage, cycle $C \subset G$

Output: Updated graph $G' = (V' \subseteq V, E' \subseteq E)$ with cycle C peeled

1: $G' = G$
2: $\mu_{cov'}(C) = \sum_{u \in C} f(u, C) cov'(u, C)$, the weighted mean of the discounted coverage of C in G
3: **for** $v \in C$ **do**
4: $\quad cov(v) \leftarrow max\{cov(v) - \mu_{cov'}(C), 0\}$
5: $\quad$ **if** $cov(v) = 0$ **then**
6: $\quad\quad V' \leftarrow V' \setminus v$
7: $\quad\quad E' \leftarrow E' \setminus \{e | e = (u, v) \cup e = (v, u)\ \forall u \in V\}$

Table 1

Overview of SCAPP

1: Annotate the assembly graph:
  a: Map reads to nodes of the assembly graph
  b: Find nodes with plasmid-specific gene matches
  c: Compute plasmid sequence scores of nodes
  d: Assign node weights
2: **for** each strongly connected component **do**
3: Iteratively peel uniform coverage cycles through plasmid gene nodes
4: Iteratively peel uniform coverage cycles through high scoring nodes
5: Iteratively peel shortest cycle through each remaining node if it meets plasmid criteria
6: Output the set of confident plasmid predictions

Read mapping

The first step in creating the annotated assembly graph (Table 1 step 1a) is to align the reads to the contigs in the graph. The links between paired-end reads aligning across contig junctions are used to evaluate potential plasmid paths in the graph. SCAPP performs read alignment using BWA [10] and the alignments are filtered to retain only primary read mappings, sorted, and indexed using SAMtools [11].

Plasmid-specific gene annotation

We created sets of PSGs by database mining and curation by plasmid microbiology experts from the Mizrahi Lab (Ben-Gurion University). Information about these PSG sets is found in Additional file 1, Section S2. The sequences themselves are available from https://github.com/Shamir-Lab/SCAPP/tree/master/scapp/data.

A node in the assembly graph is annotated as containing a PSG hit (Table 1 step 1b) if there is a BLAST match between one of the PSG sequences and the sequence corresponding to the node ($\geq$ 75% sequence identity along $\geq$ 75% of the length of the gene).

Plasmid sequence score annotation

We use PlasClass [9] to annotate each node in the assembly graph with a plasmid score (Table 1 step 1c). PlasClass uses a set of logistic regression classifiers for sequences of different lengths to assign a classification score reflecting the likelihood of each node to be of plasmid origin.

We re-weight the node scores according to the sequence length as follows. For a given sequence of length L and plasmid probability p assigned by the classifier, the re-weighted plasmid score is: $s = 0.5 + \frac{p - 0.5}{1 + e^{-0.001(L-2000)}}$. This tends to pull scores towards 0.5 for short sequences, for which there is lower confidence, while leaving scores of longer sequences practically unchanged.

Long nodes ($L > 10$ kbp) with low plasmid score ($s < 0.2$) are considered probable chromosomal sequences and are removed, simplifying the assembly graph. Similarly, long nodes ($L > 10$ kbp) with high plasmid score ($s > 0.9$) are considered probable plasmid nodes.

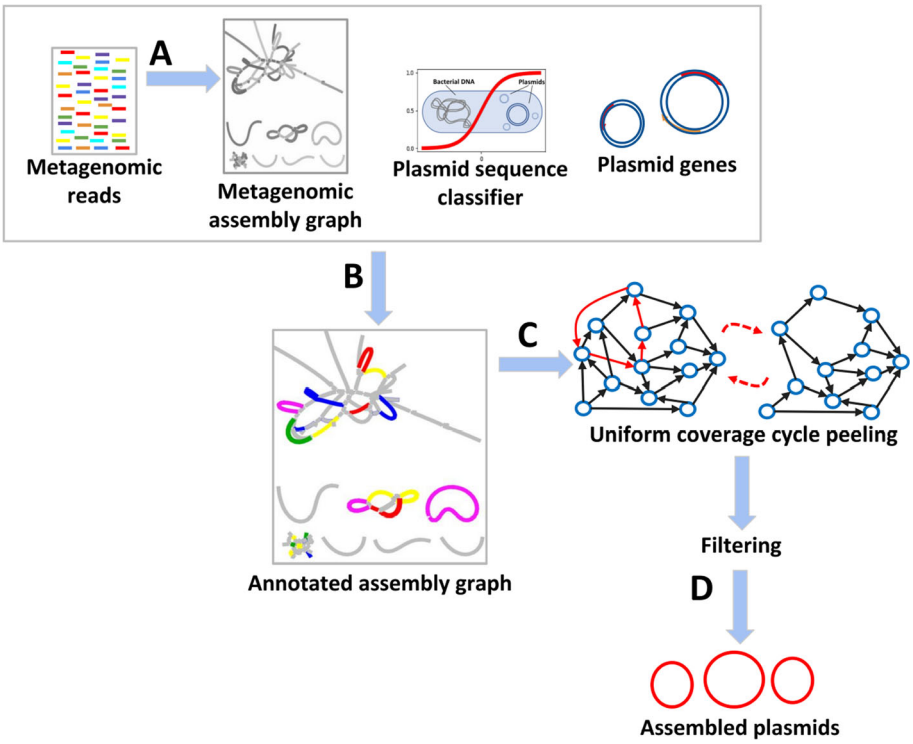

Fig. 1 Graphical overview of the SCAPP algorithm. (A) The metagenomic assembly graph is created from the sample reads. (B) The assembly graph is annotated with read mappings, presence of plasmid specific genes, and node weights based on sequence length, coverage, and plasmid classifier score. (C) Potential plasmids are iteratively peeled from the assembly graph. An efficient algorithm finds cyclic paths in the annotated assembly graph that have low weight and high chance of being plasmids. Cycles with uniform coverage are peeled. (D) Confident plasmid predictions are retained using plasmid sequence classification and plasmid-specific genes to remove likely false positive potential plasmids

Assigning node weights

In order to apply the peeling idea, nodes are assigned weights (Table 1 step 1d) so that lower weights correspond to higher likelihood to be assembled into a plasmid. Plasmid score and PSG annotations are incorporated into the node weights. A node with plasmid score s is assigned a weight $w(v) = (1-s)/(C \cdot L)$ where C is the depth of coverage of the node's sequence and L is the sequence length. This gives lower weight to nodes with higher coverage, longer sequence, and higher plasmid scores. Nodes with PSG hits are assigned a weight of zero, making them more likely to be integrated into any lowest-weight cycle in the graph that can pass through them.

Finding low-weight cycles in the graph

The core of the SCAPP algorithm is to iteratively find a lowest weight ("lightest") cycle going through each node in the graph for consideration as a potential plasmid. We use the bidirectional single-source, single-target shortest path implementation of the NetworkX Python package [12].

The order that nodes are considered matters since in each iteration potential plasmids are peeled from the graph, affecting the cycles that may be found in subsequent iterations. The plasmid annotations are used to decide the order that nodes are considered: first all nodes with PSGs, then all probable plasmid nodes, and then all other nodes in the graph (Table 1 step 2). If the lightest cycle going through a node meets certain criteria described below, it is peeled off, changing the coverage of nodes in the graph. Performing the search for light cycles in this order ensures that the cycles through more likely plasmid nodes will be considered before other cycles.

Assessing coverage uniformity

The lightest cyclic path, weighted as described above, going through each node is found and evaluated. Recycler sought a cycle with near uniform coverage, reasoning that all contigs that form a plasmid should have roughly the same coverage. However, this did not take into account the overlap of the cycle with other paths in the graph (see Fig. 2). To account for this, we instead compute a discounted coverage score for each node in the cycle based on its interaction with other paths as follows:

The *discounted coverage* of a node v in the cycle C is its coverage $cov(v)$ times the fraction of the coverage on all its

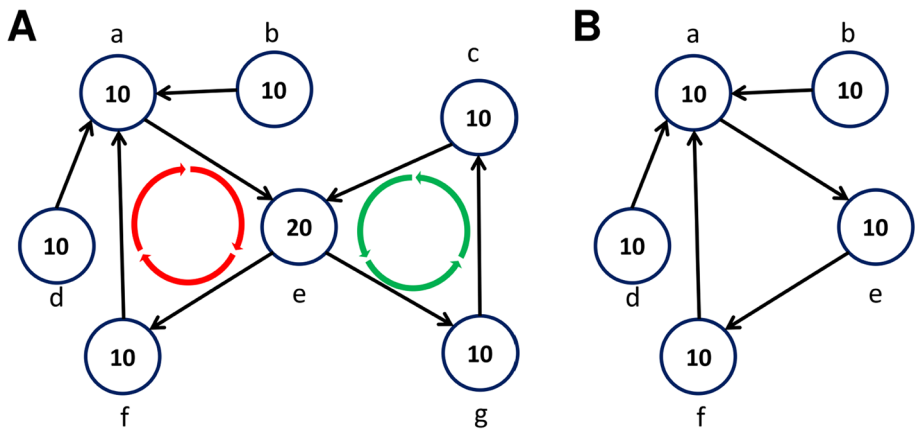

Fig. 2 Evaluating and peeling cycles. Numbers inside nodes indicate coverage. All nodes in the example have equal length. **A** Cycles (a, e, f) and (c, e, g) have the same average coverage (13.33) and coefficient of variation (CV, 0.35), but their discounted CV values differ: The discounted coverage of node a is 6, and the discounted coverage of node e is 10 in both cycles. The left cycle has discounted CV=0.22 and the right has discounted CV=0. By peeling off the mean discounted coverage of the right cycle (10) one gets the graph in **B**. Note that nodes g, c were removed from the graph since their coverage was reduced to 0, and the coverage of node e was reduced to 10

neighbors (both incoming and outgoing), $\mathcal{N}(v)$, that is on those neighbors that are in the cycle (see Fig. 2):

$$cov'(v, C) = cov(v) \cdot \left(\sum_{u \in C \wedge u \in \mathcal{N}(v)} cov(u) / \sum_{u \in \mathcal{N}(v)} cov(u) \right)$$

A node v in cycle C with contig length $len(v)$ is assigned a weight f corresponding to its fraction of the length of the cycle: $f(v, C) = len(v) / \sum_{u \in C} len(u)$. These weights are used to compute the weighted mean and standard deviation of the discounted coverage of the nodes in the cycle: $\mu_{cov'}(C) = \sum_{u \in C} f(u, C) cov'(u, C)$,

$$STD_{cov'}(C) = \sqrt{\sum_{u \in C} f(u, C)(cov'(u, C) - \mu_{cov'}(C))^2}$$

The coefficient of variation of C, which evaluates its coverage uniformity, is the ratio of the standard deviation to the mean:

$$CV(C) = \frac{STD_{cov'}(C)}{\mu_{cov'}(C)}$$

Finding potential plasmid cycles

After each lightest cycle has been generated, it is evaluated as a potential plasmid based on its structure in the assembly graph, the PSGs it contains, its plasmid score, paired-end read links, and coverage uniformity. The precise evaluation criteria are described in Additional file 1, Section S3. A cycle that passes them is defined as a potential plasmid (Table 1 steps 3–5). The potential plasmid cycles are peeled from the graph in each iteration as defined in Algorithm 2 (see also Fig. 2).

Filtering confident plasmid assemblies

In the final stage of SCAPP, PSGs and plasmid scores are used to filter out likely false-positive plasmids from the output and create a set of confident plasmid assemblies (Table 1 step 6). All potential plasmids are assigned a length-weighted plasmid score and are annotated with PSGs as was done for the contigs during graph annotation. Those that belong to at least two of the following sets are reported as confident plasmids: (a) potential plasmids containing a match to a PSG, (b) potential plasmids with plasmid score > 0.5, (c) self-loop nodes.

Results

We tested SCAPP on simulated metagenomes, human gut metagenomes, a human gut plasmidome dataset that we generated and also on parallel metagenome and plasmidome datasets from the same cow rumen microbiome specimen that we generated. The test settings and evaluation methods are described in Additional file 1, Section S5.

Simulated metagenomes

We created seven read datasets simulating metagenomic communities of bacteria and plasmids and assembled them. Datasets of increasing complexity were created as shown in Table 2. We randomly selected bacterial genomes along with their associated plasmids and used realistic distributions for genome abundance and plasmid copy number. Further details of the simulation can be found in Additional file 1, Section S4, and in Additional file 2. 5M paired-end reads were generated for Sim1 and

Table 2 Performance on simulated metagenome datasets. The number of covered plasmids (# covered) reports the number of the simulation plasmids that were covered by reads along at least 95% of their length. The set of covered plasmids is used as the gold standard in calculating the performance metrics. The numbers in parentheses are the median plasmid lengths (in kbp). F1 score is presented as a percent

				Recycler		mpSpades		SCAPP	
Sample	**# genomes**	**# plasmids**	**# covered**	**# plasmids**	**F1**	**# plasmids**	**F1**	**# plasmids**	**F1**
Sim1	10	9 (5.9)	9 (5.9)	7 (5.6)	50.0	1 (4.3)	20.0	5 (5.6)	57.1
Sim2	50	47 (19.3)	37 (13.5)	20 (3.8)	40.1	9 (5.0)	39.1	23 (5.5)	43.3
Sim3	200	210 (22.4)	136 (9.6)	61 (3.6)	32.8	27 (7.0)	32.3	48 (5.8)	42.9
Sim4	200	177 (25.4)	132 (12.7)	62 (4.1)	40.8	29 (6.0)	36.5	51 (6.2)	48.9
Sim5	300	318 (23.9)	253 (9.6)	115 (3.6)	35.2	53 (5.1)	33.8	100 (6.5)	47.5
Sim6	400	480 (13.5)	368 (9.1)	138 (3.0)	28.5	59 (5.5)	27.1	118 (5.5)	36.5
Sim7	500	571 (17.3)	410 (8.7)	132 (3.5)	31.1	69 (5.3)	28.1	141 (5.2)	40.5

Sim2, 10M for Sim3 and Sim4, and 20M for Sim5, Sim6, and Sim7.

Table 2 presents features of the simulated datasets and reports the performance of Recycler, mpSpades, and SCAPP on them. For brevity we report only F1 scores; precision and recall scores are reported in Supplementary Table 1, Additional file 1 (Section S6). Here, and throughout, all scores are adjusted to percent. SCAPP had the highest F1 score in all cases, followed by Recycler. SCAPP consistently achieved higher precision than Recycler, allowing it to perform better overall. mpSpades had the highest precision, but assembled far fewer plasmids than the other tools and gained lower recall and F1 scores. In fact, most of the plasmids assembled by mpSpades were also assembled by the other tools (see Figure S1 in Additional file 1), suggesting that these plasmids were easier to capture.

All of the tools assembled mostly shorter plasmids as reflected in the median plasmid lengths. This is likely due to the higher coverage and simplicity in the assembly graph of these plasmids, as also evidenced by the shorter lengths of the covered plasmids. SCAPP assembled many more long plasmids (> 10 kbp) than the other tools, achieving much higher recall and higher F1 score for these longer plasmids than the other tools, at the cost of some precision (see Supplementary Table 2 in Additional file 1, Section S6 for results broken down by short and long plasmids).

Human gut microbiomes

We tested the plasmid assembly algorithms on data of twenty publicly available human gut microbiome samples selected from the study of Vrieze et al. [13]. The true set of plasmids in these samples is unknown. Instead, we matched all assembled contigs to PLSDB [14] and considered the set of the database plasmids that were covered by the contigs as the gold standard (see Additional file 1, Section S5 for details). All tools were evaluated according to the same gold standard. We note that this limits the evaluation to known plasmids, potentially overcounting the number of false positive plasmids. We chose the human gut microbiome in this experiment and the next, as it is one of the most widely studied microbiome environments so plasmids in gut microbiome samples are most likely to be represented in the database.

Table 3 presents the results of the three algorithms averaged across all twenty samples. The detailed results on each of the samples are presented in Supplementary Table 2 and Figure S2, Additional file 1 (Section S7). SCAPP performed best in more cases, with mpSpades failing to assemble any gold standard plasmid in over half the samples. We note that all of the cases where SCAPP had recall of 0 occurred when the number of gold standard plasmids was very small and the other tools also failed to assemble them. On the largest samples with the most gold standard plasmids SCAPP performed best, highlighting its superior performance on the types of samples most likely to be of interest in experiments aimed at plasmid assembly. SCAPP consistently outperformed Recycler by achieving higher precision, a result that is consistent with the other experiments.

Human gut plasmidome

The protocol developed in Brown Kav et al. [15] enables extraction of DNA from isolate or metagenomic samples

Table 3 Performance on the human gut metagenomes. Number of plasmids, the median plasmid length (in kbp), and performance measures for all tools are averaged across the twenty samples. The average number of plasmids and median length of the gold standard sets of plasmids were 4.8 and 12.4 respectively

Tool	**# plasmids**	**Median length**	**Precision**	**Recall**	**F1**
Recycler	15.9	3.6	7.1	36.4	10.9
mpSpades	6.5	5.0	7.9	17.4	10.3
SCAPP	9.8	4.4	11.5	36.4	16.1

with the plasmid content highly enriched. The sequence contents of such a sample is called the *plasmidome* of the sample. This enrichment for plasmid sequences increases the chance of revealing the plasmids in the sample. The protocol was assessed to achieve samples with at least 65% plasmid contents by Krawczyk et al. [5]. We sequenced the plasmidome of the human gut microbiome from a healthy adult male according to the plasmid enrichment protocol. 18,616,649 paired-end reads were sequenced with the Illumina HiSeq2000 platform, read length 150bp and insert size 1000.

The gold standard set of plasmids, determined as for the gut metagenome samples, consisted of 74 plasmids (median length = 2.1 kbp). Note that the plasmidome extraction process over-amplifies shorter plasmids, as reflected in the shorter median plasmid length. Performance was computed as in the metagenomic samples and is shown in Table 4. SCAPP achieved best overall performance, while mpSpades had lower precision and much lower recall than the other tools.

Notably, although the sample was obtained from a healthy donor, some of the plasmids reconstructed by SCAPP matched reference plasmids found in potentially pathogenic hosts such as *Klebsiella pneumoniae*, pathogenic serovars of *Salmonella enterica*, and *Shigella sonnei*. The detection of plasmids previously isolated from pathogenic hosts in the healthy gut indicates potential pathways for transfer of virulence genes.

We used MetaGeneMark [16] to find potential genes in the plasmids assembled by SCAPP. Two hundred ninety-four genes were found, and we annotated them with the NCBI non-redundant (nr) protein database using BLAST. Forty-six of the plasmids contained 170 (58%) genes with matches in the database (> 90% sequence identity along > 90% of the gene length), of which 77 (45%) had known functional annotations, which we grouped manually in Fig. 3A. There were six antibiotic and toxin (such as heavy metal) resistance genes, all on plasmids that were not in the gold standard set, highlighting SCAPP's ability to find novel resistance carrying plasmids. Sixty of the 77 genes (78%) with functional annotations had plasmid-associated functions: replication, mobilization, recombination, resistance, and toxin-antitoxin systems. Twenty-nine out of the 33 plasmids that contained functionally annotated genes (88%) contained at least one of these plasmid associated functions. This provides a strong indication that SCAPP succeeded in assembling true plasmids of the human gut plasmidome.

We also examined the hosts that were annotated for the plasmid genes and found that almost all of the plasmids with annotated genes contained genes with annotations from a variety of hosts, which we refer to here as "broad-range" (see Fig. 3B). Of the 40 plasmids with genes from annotated hosts, only 10 (25%) had genes with annotated hosts all within a single phylum. This demonstrates that these plasmids assembled and identified by SCAPP may be involved in one stage of transferring genes, such as the antibiotic resistance genes we detected, across a range of bacteria.

Table 4 Performance on the human gut plasmidome. Number of plasmids, the median plasmid length (in kbp), and performance measures for all tools

Tool	# plasmids	Median length	Precision	Recall	F1
Recycler	93	2.1	15.1	37.8	21.5
mpSpades	53	3.0	11.3	9.4	10.3
SCAPP	82	2.4	17.1	35.9	23.1

Parallel metagenomic and plasmidome samples

We performed two sequencing assays on the same cow rumen microbiome sample of a four-month old calf. In one subsample total DNA was sequenced. In the other, plasmid-enriched DNA was extracted as described in Brown Kav et al. [15] and sequenced (see Fig. 4). 27,127,784 paired-end reads were sequenced in the plasmidome, and 54,292,256 in the metagenome. Both were sequenced on the Illumina HiSeq2000 platform with read length 150bp and insert size 1000.

This parallel data enabled us to assess the plasmids assembled on the metagenome using the plasmidome, without resorting to PLSDB matches as the gold standard. Such assessment is especially useful for samples from non-clinical environments such as the cow rumen, as PLSDB likely under-represents plasmids in them.

Table 5 summarizes the results of the three plasmid discovery algorithms on both subsamples. mpSpades made the fewest predictions and Recycler made the most. To compare the plasmids identified by the different tools, we considered two plasmids to be the same if their sequences matched at > 80% identity across > 90% of their length. The comparison is shown in Figure S3, Additional file 1 (Section S8). In the plasmidome subsample, 50 plasmids were identified by all three methods. Seventeen were common to the three methods in the metagenome. In both subsamples, the Recycler plasmids included all or almost all of those identified by the other methods plus a large number of additional plasmids. In the plasmidome, SCAPP and Recycler shared many more plasmids than mpSpades and Recycler.

We also evaluated the results of the plasmidome and metagenome assemblies by comparison to PLSDB as was done for the human gut samples. The metagenome contained only one matching PLSDB reference plasmid, and none of the tools assembled it. The plasmidome had only seven PLSDB matches, and mpSpades, Recycler, and SCAPP had F1 scores of 2.86, 2.67, and 1.74, respectively. The low fraction of PLSDB matches out of the assem-

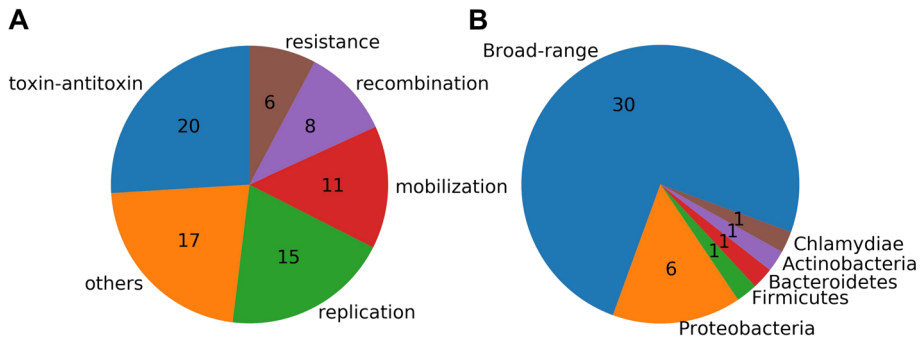

Fig. 3 Annotation of genes on the plasmids identified by SCAPP in the human gut plasmidome sample. **A** Functional annotations of the plasmid genes. **B** Host annotations of the plasmid genes. "Broad-range" plasmids had genes annotated with hosts from more than one phylum

bled plasmids suggests that the tools can identify novel plasmids that are not in the database.

In order to fully leverage the power of parallel samples, we computed the performance of each tool on the metagenomic sample using the reads of the plasmidomic sample, without doing any contig and plasmid assembly on the latter. The rationale was that the reads of the plasmidome represent the full richness of plasmids in the sample in a way that is not biased by a computational procedure or prior biological knowledge.

We calculated the *plasmidome read-based precision* by mapping the plasmidomic reads to the plasmids assembled from the metagenomic sample (Fig. 4). A plasmid with $>$ 90% of its length covered by more than one plasmidomic read was considered to be a true positive. The precision of an algorithm was defined as the fraction of true positive plasmids out of all reported plasmids. The *plasmidome read-based recall* was computed by mapping the plasmidomic reads to the contigs of the metagenomic assembly. Contigs with $>$ 90% of their length covered by plasmidomic reads at depth $>$ 1 were called *plasmidic contigs*. Plasmidic contigs that were part of the assembled plasmids were counted as true positives, and those that were not were considered false negatives. The recall was defined as the fraction of the plasmidic contigs' length that was integrated in the assembled plasmids. Note that

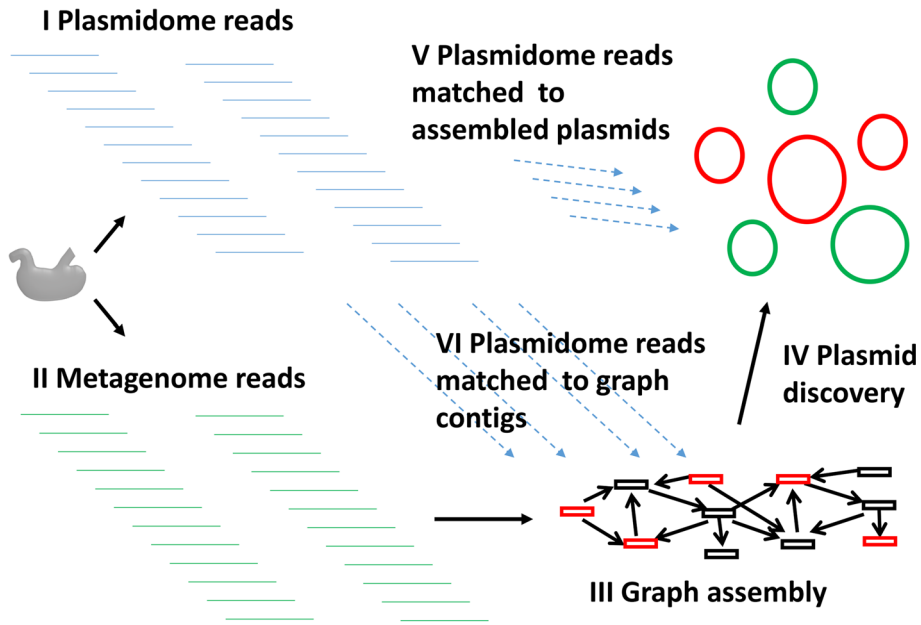

Fig. 4 Outline of the read-based performance assessment. Plasmidome (I) and metagenome reads (II) are obtained from subsamples of the same sample. (III) The metagenome reads are assembled into a graph. (IV) The graph is used to detect and report plasmids by the algorithm of choice. (V) The plasmidome reads are matched to assembled plasmids. Matched plasmids (red) are used to calculate plasmid read-based precision. (VI) The plasmidome reads are matched to the assembly graph contigs. Covered contigs (red) are considered plasmidic. The fraction of total length of plasmidic contigs included in the detected plasmids gives the plasmidome read-based recall

Table 5 Number of plasmids assembled by each tool and their median lengths (in kbp) for the parallel metagenome and plasmidome samples

Tool	metagenome		plasmidome	
	# plasmids	median length	# plasmids	median length
Recycler	60	4.3	147	1.7
SCAPP	25	5.8	110	1.8
mpSpades	26	6.2	65	2.0

the precision and recall here are measured using different units (plasmids and base pairs, respectively) so they are not directly related. For mpSpades, which does not output a metagenomic assembly, we mapped the contigs from the metaSPAdes assembly to the mpSpades plasmids using BLAST (> 80% sequence identity matches along > 90% of the length of the contigs).

There were 293 plasmidic contigs in the metagenome assembly graph, with a total length of 146.6 kbp. The plasmidome read-based performance is presented in Fig. 5A. All tools achieved a similar recall of around 12. SCAPP and mpSpades performed similarly, with SCAPP having slightly higher precision (24.0 vs 23.1) but slightly lower recall (11.9 vs 12.2). Recycler had a bit higher recall (13.1), but at the cost of far lower precision (11.7). Hence, a much lower fraction of the plasmids assembled by Recycler in the metagenome were actually supported by the parallel plasmidome sample, adding to the other evidence that the false positive rate of Recycler exceeds that of the other tools.

We also compared the plasmids assembled by each tool in the two subsamples. For each tool, we considered the plasmids it assembled from the plasmidome to be the gold standard set, and used it to score the plasmids it assembled in the metagenome. The results are shown in Fig. 5B. SCAPP had the highest precision. Since mpSpades had a much smaller gold standard set, it achieved higher recall and F1. Recycler output many more plasmids than the other tools in both samples, but had much lower precision, suggesting that many of its plasmid predictions may be spurious.

Next, we considered the union of the plasmids assembled across all tools as the gold standard set and recomputed the scores. We refer to them as "overall" scores. Figure 5C shows that overall precision scores were the same as in Fig. 5B, while overall recall was lower for all the tools, as expected. mpSpades underperformed because of its smaller set of plasmids, and SCAPP had the highest overall F1 score. Recycler performed relatively better on recall than the other tools as expected, as it reports many plasmids and has significant overlap with the plasmids reported by the other tools.

We detected potential genes in the plasmids assembled by SCAPP in the plasmidome sample and annotated them as we did for the human gut plasmidome. The gene function and host annotations are shown in Figure S4, Additional file 1 (Section S8). Out of 242 genes, only 34 genes from 17 of the plasmids had annotations, and only 18 of these had known functions, highlighting that many of the plasmids in the cow rumen plasmidome are as yet unknown. The high percentage of genes of plasmid function (15/18) indicates that SCAPP succeeded in assembling novel plasmids. Unlike in the human gut plasmidome, most of the plasmids with known host annotations had hosts from a single phylum.

Performance summary

We summarize the performance of the tools across all the test datasets in Table 6. The performance of two tools was

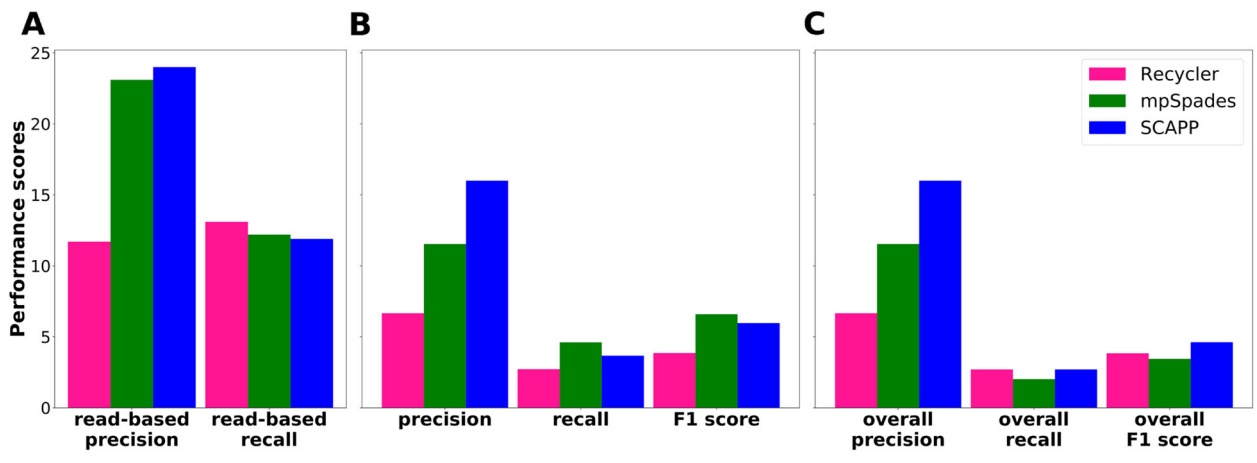

Fig. 5 Performance on the parallel datasets. **A** Plasmidome read-based performance. **B** Performance of each tool on the plasmids assembled from the metagenome using as gold standard the plasmids assembled from the plasmidome by the same tool. **C** Overall performance on the plasmids assembled from the metagenome compared to the union of all plasmids assembled by all tools in the plasmidome

Table 6 Summary of performance. Comparison of the performance of the tools on each of the datasets. When multiple samples were tested, the number of samples appears in parentheses, and average performance is reported. For the parallel samples results are for the evaluation of the metagenome based on the plasmidome, and precision and recall are plasmidome read-based. Unless otherwise stated, F1 score is used. Note that in the simulations, SCAPP ≫ mpSpades

Test	Ranking
Simulations (7)	SCAPP > Recycler > mpSpades
Human gut metagenomes (20)	SCAPP ≫ mpSpades > Recycler
Plasmidome	SCAPP > Recycler ≫ mpSpades
Parallel: within tool	mpSpades > SCAPP ≫ Recycler
Parallel: "overall", across tools	SCAPP > Recycler > mpSpades
Parallel: precision	SCAPP ≈ mpSpades ≫ Recycler
P arallel: recall	Recycler > mpSpades ≈ SCAPP

considered similar (denoted ≈) if their scores were within 5% of each other. Performance of one tool was considered to be much higher than the other (≫) if its score was > 30% higher (an increase of $5-30\%$ is denoted by >).

We see that in most cases SCAPP was the highest performer. Furthermore, in all other cases SCAPP performed close to the top performing tool.

Resource usage

The runtime and memory usage of the three tools are presented in Table 7. Recycler and SCAPP require assembly by metaSPAdes and pre-processing of the reads and the resulting assembly graph. SCAPP also requires post-processing of the assembled plasmids. mpSpades requires post-processing of the assembled plasmids with the plasmidVerify tool. The reported runtimes are for the full pipelines necessary to run each tool – from reads to assembled plasmids.

In almost all cases assembly was the most memory intensive step, and so all tools achieved very similar peak memory usage (within 0.01 GB). Therefore, we report the RAM usage for this step.

Table 7 Resource usage of the three methods. Peak RAM of the assembly step (metaSPAdes for Recycler and SCAPP, metaplasmidSPAdes for mpSpades) in GB. Runtime (wall clock time, in minutes) is reported for the entire pipeline including assembly and any pre-processing and post-processing required. Human metagenome results are an average across the 20 samples

		Runtime (minutes)		
Dataset	RAM (GB)	Recycler	mpSpades	SCAPP
Human metagenomes	21	115	103	130
Plasmidome	30	907	548	909
Parallel metagenome	148	2118	2132	2230
Parallel plasmidome	26	881	684	884

The assembly step was also the longest step in all cases. SCAPP was slightly slower than Recycler as a result of the additional annotation steps, and mpSpades was 5–40% faster. However, note that mpSpades does not output a metagenomic assembly graph, so users interested in both the plasmid and non-plasmid sequences in a sample would need to run metaSPAdes as well, practically doubling the runtime.

Performance measurements were made on a 44-core, 2.2 GHz server with 792 GB of RAM. Sixteen processes were used where possible. Recycler is single-threaded, so only one process was used for it.

Discussion

Plasmid assembly from metagenomic sequencing is a very difficult task, akin to finding needles in a haystack. This difficulty is demonstrated by the low numbers of plasmids found in real samples. Even in samples of the human gut microbiome, which is widely studied, relatively few plasmids that have matches in the extensive plasmid database PLSDB were recovered. Despite the challenges, SCAPP was able to assemble plasmids across a number of clinically relevant samples. SCAPP significantly outperformed mpSpades in simulation and on a range of human gut metagenome and plasmidome samples. In simulation mpSpades achieved very high precision at the expense of low recall, and SCAPP had higher combined F1 score. The high precision was not observed in real data, which is more difficult than the simulations. SCAPP was also consistently better than Recycler across almost all tests. Though SCAPP and Recycler share the idea of cycle peeling, SCAPP was shown to have higher precision, due to incorporating additional biological information and better edge weighting.

Another contribution of this study is the joint analysis of the parallel metagenome and plasmidome from the same sample. We show that this enables a novel way to evaluate plasmid assembly algorithms on the metagenome data, by using the coverage information from the plasmidome. This novel approach bypasses the need to rely on known plasmids for evaluation, which is biased due to research focus. We developed several evaluation metrics for such data, and think they can be useful for future plasmid studies, especially in non-clinical and non-human samples where plasmid knowledge is scarce.

A key difficulty in evaluation of performance of plasmid discovery algorithms is the lack of gold standard. The verification of reported plasmids is done either based on prior biological knowledge, which is biased, or by experimental verification, which is slow and expensive. Moreover, such verification evaluates precision but does not give information on the extent of missed plasmids, or recall. While

simulations can evaluate both parameters accurately, they are inherently artificial, and necessitate many modeling assumptions that are not fully supported by experimental data. For that reason we chose here to focus primarily on real data, and preferred diversity in the real data types over extensive but artificial simulations. The parallel samples strategy is another partial answer to this problem.

SCAPP has several limitations. Like the other de Bruijn graph-based plasmid assemblers, it may split a cycle into two when a shorter cycle is a sub-path of a longer cycle. It also has difficulties in finding very long plasmids, as these tend to not be completely covered and fragmented into many contigs in the graph. Note however that it produced longer cycles than Recycler. Compared to mpSpades, each algorithm produced longer cycles in different tests. Another limitation is the inherent bias in relying on known plasmid genes and plasmid databases, which tend to under-represent non-clinical samples. With further use of tools like SCAPP, perhaps with databases tailored to specific environments, further improvement is possible.

Conclusions

We introduced SCAPP, a new plasmid discovery tool based on combination of graph theoretical and biological considerations. Overall, SCAPP demonstrated better performance than Recycler and metaplasmidSpades in a wide range of real samples from diverse contexts. By applying SCAPP across large sets of samples, many new plasmid reference sequences can be assembled, enhancing our understanding of plasmid biology and ecology.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40168-021-01068-z.

Additional file 1 — Supplementary information: Supplementary methods, experimental settings information, and results supporting the main text of this paper, including Figures S1-S4, Supplementary Tables 1 and 2. PDF file.

Additional file 2 — Simulation reference genomes: Tab-separated list of the human gut-specific reference genomes used in the simulations.

Acknowledgements

We thank members of the Shamir Lab for their help and advice—Roye Rozov, Lianrong Pu, Hagai Levi, and Nimrod Rappoport.

Authors' contributions

DP developed and implemented the SCAPP algorithm and benchmark experiments, performed analysis, and wrote the manuscript. AZ assisted with analysis and plasmid annotations and wrote the manuscript. MP curated plasmid-specific genes, assisted with gene annotations and wrote the manuscript. OF oversaw the parallel cow rumen metagenome-plasmidome experiment. AS oversaw the human gut plasmidome experiment. IM oversaw experimental and analysis aspects of the project and edited the manuscript. RS oversaw the computational and analysis aspects of project and edited the manuscript. All authors edited and approved the final manuscript.

Funding

PhD fellowships from the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University and Israel Ministry of Immigrant Absorption (to DP). Israel Science Foundation (ISF) grant 1339/18, US - Israel Binational Science Foundation (BSF) and US National Science Foundation (NSF) grant 2016694 (to RS), ISF grant 1947/19 and ERC Horizon 2020 research and innovation program grant 640384 (to IM).

Availability of data and materials

The datasets supporting the conclusions of this article are available in the sequence read archive (SRA), accession numbers: ERR1297645, ERR1297651, ERR1297671, ERR1297685, ERR1297697, ERR1297700, ERR1297720, ERR1297738, ERR1297751, ERR1297770, ERR1297785, ERR1297796, ERR1297798, ERR1297810, ERR1297822, ERR1297824, ERR1297834, ERR1297838, ERR1297845, ERR1297852 (for the human gut metagenomes); accession SRR11038083 (for the human gut plasmidome); and accessions SRR11038085 and SRR11038085 (for the cow rumen metagenome and plasmidome samples, respectively). Project name: SCAPP
Project homepage: https://github.com/Shamir-Lab/SCAPP
Operating system: Platform independent (tested on Linux)
Programming language (Python3)
License: MIT

Declarations

Ethics approval and consent to participate

Sequencing of the human gut plasmidome was approved by the local ethics committee of Clalit HMO, approval number 0266-15-SOR. Extraction and sequencing of the cow rumen microbiome was approved by the local ethics committee of the Volcani Center, approval numbers 412/12IL and 566/15IL.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Blavatnik School of Computer Science, Tel Aviv University, 6997801 Tel Aviv, Israel. [2]Department of Life Sciences, Ben-Gurion University of the Negev and the National Institute for Biotechnology in the Negev, 8410501 Beer-Sheva, Israel. [3]Institute of Microbiology, University of Innsbruck, A-6020 Innsbruck, Austria. [4]Health Sciences, Ben-Gurion University of the Negev, 8410501 Beer-Sheva, Israel. [5]Soroka University Medical Center, 8410501 Beer-Sheva, Israel.

Received: 14 August 2020 Accepted: 1 April 2021
Published online: 25 June 2021

References

1. Arredondo-Alonso S, Willems R, van Schaik W, Schürch A. On the (im)possibility of reconstructing plasmids from whole-genome short-read sequencing data. Microb Genomics. 2017;3(10):000128.
2. Carattoli A, Zankari E, García-Fernández A, Larsen M, Lund O, Villa L, Aarestrup F, Hasman H. In silico detection and typing of plasmids using PlasmidFinder and plasmid multilocus sequence typing. Antimicrob Agents Chemother. 2014;58(7):3895–903.
3. Zhou F, Xu Y. cBar: a computer program to distinguish plasmid-derived from chromosome-derived sequence fragments in metagenomics data. Bioinforma. 2010;26(16):2051–2.
4. Arredondo-Alonso S, Bootsma M, Hein Y, Rogers MR, Corander J, Willems RJ, Schürch AC. gplas: a comprehensive tool for plasmid analysis using short-read graphs. Bioinformatics. 2020;36(12):3874–6.
5. Krawczyk P, Lipinski L, Dziembowski A. PlasFlow: predicting plasmid sequences in metagenomic data using genome signatures. Nucleic Acids Res. 2018;46(6):35.
6. Antipov D, Hartwick N, Shen M, Raiko M, Lapidus A, Pevzner P. plasmidSPAdes: assembling plasmids from whole genome sequencing data. Bioinforma. 2016;32(22):3380–7.

7. 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. Bioinforma. 2017;33(4):475–82.
8. Antipov D, Raiko M, Lapidus A, Pevzner P. Plasmid detection and assembly in genomic and metagenomic data sets. Genome Res. 2019;29(6):961–8.
9. Pellow D, Mizrahi I, Shamir R. PlasClass improves plasmid sequence classification. PLoS Comput Biol. 2020;16(4):1007781.
10. Li H. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. arXiv preprint arXiv:1303.3997. 2013.
11. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. The sequence alignment/map format and samtools. Bioinforma. 2009;25(16):2078–9.
12. Hagberg A, Schult D, Swart P. Exploring network structure, dynamics, and function using NetworkX. In: Varoquaux G, Vaught T, Millman J, editors. Proceedings of the 7th Python in Science Conference (SciPy). Pasadena: Los Alamos National Lab (LANL); 2008. p. 11–5.
13. Vrieze A, Van Nood E, Holleman F, Salojärvi J, Kootte R, Bartelsman J, Dallinga–Thie G, Ackermans M, Serlie M, Oozeer R, et al. Transfer of intestinal microbiota from lean donors increases insulin sensitivity in individuals with metabolic syndrome. Gastroenterol. 2012;143(4):913–6.
14. Galata V, Fehlmann T, Backes C, Keller A. PLSDB: a resource of complete bacterial plasmids. Nucleic Acids Res. 2018;47(D1):195–202.
15. Brown Kav A, Benhar I, Mizrahi I. A method for purifying high quality and high yield plasmid dna for metagenomic and deep sequencing approaches. J Microbiol Meth. 2013;95(2):272–9.
16. Zhu W, Lomsadze A, Borodovsky M. Ab initio gene identification in metagenomic sequences. Nucleic Acids Res. 2010;38(12):132.

Publisher's Note