HyAsP, a hybrid assembler for plasmids

Robert Müller and Cedric Chauve

Supplement 1 – Technical details and additional results

This supplement provides technical details on HyAsP and its usage, information on the plasmid and gene databases as well as the experimental design, and additional analysis results.

1 Greedy algorithm

This section provides details on the key steps and usage of HyAsP. In addition, we provide an extensive list of its parameters in Section 6. We first provide a pseudocode that outlines the greedy algorithm.

Algorithm 1: Overview of HyAsP algorithm

```
Data: assembly graph AG, gene-contig mapping GCM
Result: collection of predicted plasmids PC

Determine seed contigs and sort them by their eligibility;
Initialise empty PC;
while ∃ eligible seed S do
    Initialise new plasmid P with seed S;
    while ∃ eligible extension of P do
        Choose extension and add the corresponding contig to P;
    Circularise the contig chain of P (if possible);
    Finalise the read-depth values of the contigs in P;
    foreach contig C in P do
        Decrease the read depth of C in AG by the depth of C in P;
    Add plasmid P to PC;
Split PC into putative plasmids and questionable plasmids;
Group the plasmids based on their characteristics (optional);
Output predicted plasmids;
```

1.1 Database generation

In order to identify the seed contigs and, thus, the plasmids in an assembly, the greedy algorithm relies on a database of plasmid genes. This database can be created by HyAsP (command create) from a collection of plasmids given through, e.g., accession numbers. The corresponding GenBank files are then obtained from NCBI and all features of type gene are extracted from them in FASTA format. The identifier of each gene contains its locus tag or, if that information is not available, the content of the gene-qualifier tag.

In order to avoid an unnecessarily inflated database, it can also be dereplicated based on the sequences of the genes. In addition, there are several filtering options to exclude plasmids from the database (see Section 6.1 for more information).

1.2 Eligibility criteria for seeds

A seed S has to satisfy the following *eligibility* criteria in order to be considered as a potential starting point:

density(S) $\geq$ min_seed_gene_density
: A gene density that is too low makes a seed ineligible, because a small number of gene hits (relative to the length of the contig) might indicate chance hits or chimeric contigs where a small plasmid portion is combined with a large chromosomal portion.

depth(S) $\geq$ min_read_depth
: A read depth below the threshold turns the seed ineligible in order to avoid starting a plasmid from

a sequencing fragment (with low read depth) or from some small residue remaining after using the seed in another plasmid.

length(S) $\leq$ max_length
: A seed is ineligible when its length alone already exceeds an upper bound on the length of plasmids.

The default values of the thresholds used in above criteria were derived from a large plasmid collection downloaded from the NCBI database (see Section 3 for more information). However, all thresholds are also user-definable.

The eligible seeds are managed by a seed enumerator, which sorts them based on their gene density, GC content and branching. The seed enumerator prefers seeds with at most one predecessor and one successor over those with more neighbours. Within both groups, seeds with a higher gene density and larger deviation from the overall GC content of all contigs are used first. A high gene density usually corresponds to a large number of plasmid genes found on the contig and, thus, is a strong, direct indicator for a plasmid origin. As has been previously observed Nishida (2012), plasmids and chromosomes of the same host tend to differ in GC content. Therefore, a contig is deemed the more likely to stem from a plasmid, the more it differs in GC content from the overall one, which is assumed to be dominated by the chromosomal contigs.

When a new plasmid is initialised, the seed enumerator is queryied for the next seed, returning the one with the highest score still eligible. After the construction of each plasmid, the read-depth values of the used seeds is reduced by the amount of depth with which they occur in that plasmid and those seeds that no longer satisfy the minimum read-depth requirement are removed from the enumerator, i.e. they are no longer eligible. Plasmids are constructed until the seed enumerator runs out of eligible seeds.

1.3 Eligibility criteria for extensions

During its construction, a plasmid is considered as a chain of oriented contigs. Orientation + implies that the contig is used as it is stated in the assembly graph, while orientation - indicates the use of its reverse complement. The contig chain initially consists only of the chosen seed (in orientation +). Subsequently, the plasmid is extended one contig at a time while there is an eligible extension at one of its endpoints. In each iteration, the endpoints are used to search for extensions to the left resp. right by examining the corresponding links in the assembly graph and considering the orientations of the endpoint contigs.

Consider an endpoint contig A of plasmid P and another contig B that is suitably connected to A in the assembly graph. The extension of P with B via A is *eligible* if the following criteria are satisfied:

novelty
: An extension is ineligible if the corresponding link between A and B has already been used in plasmid P.

depth(B) $\geq$ min_read_depth
: Eligible extensions involve only contigs with a sufficiently high read depth in order to avoid adding contigs that are more likely to be sequencing fragments or have only a small residue remaining after occurring in another plasmid.

|gc_content(P) - gc_content(B)| $\leq$ max_gc_diff
: An extension is only eligible if the difference in GC content between contig and the plasmid constructed so far is not too large.

gene proximity
: An eligible extension does not create overly large gene-free stretches in the plasmid. Thus, contig B is either a seed or the number of contigs between it and the last seed in P is at most max_intermediate_contigs or the number of nucleotides between it and the last seed in P is at most max_intermediate_nt.

length(B) + length(P) $\leq$ max_length
: An extension is ineligible if it would increase the length of the plasmid beyond the upper bound on the length of plasmids.

ext_score(P, B) $\leq$ max_score
: Extensions whose score exceeds max_score are ineligible as their quality is considered to be too poor. See below for the definition of ext_score.

The default values of above thresholds were derived from a large plasmid collection downloaded from the NCBI database (if applicable). However, all of them are also user-definable. The score of an extension is defined as follows:

$$\begin{aligned}\mathsf{ext\_score}(\mathsf{P},\mathsf{B}) \;=\; & \mathsf{weight.depth\_diff} * |1 - \mathsf{depth}(\mathsf{B}) \,/\, \mathsf{average\_depth}(\mathsf{P})| \\ & + \mathsf{weight.gene\_density} * \mathsf{density}(\mathsf{B}) \\ & + \mathsf{weight.gc\_diff} * |\mathsf{gc\_content}(\mathsf{P}) - \mathsf{gc\_content}(\mathsf{B})|\end{aligned}$$

with weight being a vector of weights of the different components of the scoring function. The weights (by default set to 1) are user-definable and can also be determined automatically from the standard deviation of the read depth, gene density and GC content, respectively, among the seed contigs. In each iteration, the eligible extensions with the lowest score is chosen.

1.4 Putative plasmids

A plasmid P is considered *putative* if it satisfies the following criteria:

$\mathsf{density}(\mathsf{P}) \geq \mathsf{min\_gene\_density}$
: If the gene density of the predicted plasmid is too low it might be (too) chimeric or even of chromosomal origin.

$\mathsf{depth}(\mathsf{P}) \geq \mathsf{min\_plasmid\_read\_depth}$
: A too low overall read depth can be an indicator that the plasmid consists mostly of sequencing fragments or residues.

$\mathsf{min\_length} \leq \mathsf{length}(\mathsf{P}) \leq \mathsf{max\_length}$
: Plasmids that are very short or long might be just fragments or rather parts of the chromosomes, respectively.

non-subplasmid
: A putative plasmid is not subsumed by another plasmid, i.e. the set of contigs underlying a putative plasmid is not a subset of another one.

1.5 Other modes of operation

In addition to the procedure described in the Methods section of the main manuscript, our greedy algorithm is equipped with several options to change its behaviour:

node-based extensions
: When the node-based mode is activated, the novelty criterion changes in a way that each contig can occur at most once per plasmid.

median read depth
: The average read depth of a plasmid is computed using the median instead of the mean.

probabilistic extensions
: Activating the probabilistic mode turns the extension step into a probabilistic choice. Instead of choosing the extension with the lowest score, each is assigned a probability based on the involved contig's share of the total read depth of all eligible extensions.

branching mode
: The greedy algorithm, by default, performs a single eligible extension per iteration at one of the two endpoints of the plasmid, creating a linear contig chain in which each contig has at most one left and one right extension. However, the number of eligible extensions performed per iteration, the *fanout*, can be increased, leading to branchings in the contig chains which rather turns into a contig network. For a fanout larger than one, a contig can have multiple extensions to the left resp. right in the contig network and there can be more than two endpoints. Thus, the branching mode allows to identify non-linear areas of the assembly graph consisting of contigs likely to be of plasmid origin.

1.6 From FASTQ reads to plasmids

In order to facilitate the simple predictions of plasmids from read data and a collection of genes, HyAsP was bundled with read preprocessing and assembly into a Python pipeline. First, the FASTQ reads are preprocessed with Trim Galore (Krueger (2016)) and sickle (Joshi and Fass (2011)) using their default thresholds for length and quality. Optionally, the reads are analysed before and after the preprocessing using FastQC (Andrews (2010)). The preprocessed read data is then assembly with Unicycler (Wick *et al.* (2017)) in normal mode (by default). Subsequently, the given genes are mapped to the contigs of the obtained assembly (using blastn (v2.6.0) from BLAST+ (Camacho *et al.* (2009))) and the hits are filtered by their quality and length. A hit has to show an identity of 95 % or higher and has to cover at least 95 % of the gene. Finally, the plasmids are determined with HyAsP based on the assembly and the filtered gene-contig hits.

1.7 Output files

Contig chains (only for fanout = 1)
Lists contigs and their orientation as they appear in the linear contig chain of all plasmids (both putative and questionable).
File name: contig_chains.csv
Format: <plasmid id>;<comma-separated list of contigs with orientation>
Example: plasmid_0;23+,25-,10+

Plasmids (only for fanout = 1)
Stores the plasmid sequences (concatenations of the (orientated) contigs) in FASTA format. The plasmid identifier is also used as the identifier of the FASTA entry. The additional information in the defline of each entry include seed_contig, length, median_read_depth or mean_read_depth, gene_density, num_cds, gc_content, circular.
File names: putative_plasmids.fasta, questionable_plasmids.fasta
Format: FASTA with additional information in defline (as tab-separated list of <property>=<value> pairs)

Contig collections (only for fanout > 1)
Lists name and orientation of all contigs for each putative resp. questionable plasmid.
File names: putative_plasmid_contigs_list.csv, questionable_plasmid_contigs_list.csv
Format: <plasmid id>;<comma-separated list of contigs with orientation>
Example: plasmid_0;23+,25-,10+

Contigs
Stores the contigs underlying plasmids in FASTA format. If a contig is used in negative orientation, the reverse complement of its sequence is stored in the output file. The identifier of a FASTA entry consists of the contig name and the contig identifier (separated by the —-symbol).
File names: putative_plasmid_contigs.fasta, questionable_plasmid_contigs.fasta
Format: FASTA

Tagged assembly graph
Stores a copy of the input assembly graph and adds colour and label information to contigs used in putative and questionable plasmids. Contigs occurring in at least one putative plasmids are blue, while those occurring one or more questionable plasmids (but no putative plasmid) are light blue. Each contig is labelled with the identifiers of the plasmids it occurs in and seed contigs also contain a * in their label.
File name: tagged_assembly.gfa
Format: GFA with additional (optional) tags for contigs

Plasmid bins (only for binning != NaN)
Lists the plasmid identifiers (of putative plasmids resp. all plasmids) grouped into the different bins.
File names: plasmid_bins_putative.csv, plasmid_bins_all.csv
Format: <comma-separated list of plasmid identifiers>
Example: plasmid_0,plasmid_10,plasmid_3

Variable	Minimum	Mean	SD	Q1	Median	Q3	Maximum
Length	537	87615	200522	7331	30288	87953	2657929
Number of genes	0	65	172	2	13	55	2504
Gene density	0.0000	0.4869	0.3227	0.1516	0.5704	0.7828	1.0000
Sequence GC content	0.1470	0.4427	0.1259	0.3354	0.4278	0.5388	0.7562
Genes GC content	0.1931	0.4511	0.1286	0.3417	0.4367	0.5588	0.7386

Supplement Table S1: Statistics on characteristics of the plasmids in the NCBI-database. Values related to the length and the number of genes were rounded to the nearest whole number.

Variable	Minimum	Mean	SD	Q1	Median	Q3	Maximum
Length	100	856	772	375	708	1095	51060
GC content	0.0808	0.5422	0.1322	0.4444	0.5836	0.6404	0.8511

Supplement Table S2: Statistics on characteristics of the genes in the NCBI-database. Values related to the length were rounded to the nearest whole number.

2 Plasmid databases

This section describes the characteristics of the plasmid and gene databases used in the experiments. Supplement 2 lists the number of plasmids per species in the different databases.

2.1 Characteristics of NCBI-database plasmids

The NCBI-database was built through the create command of HyAsP using the downloaded and reformatted (but not yet filtered) NCBI plasmid table from Section 3. Only GenBank plasmids released before the 19 December 2015 and at least 500 nt long were used (-t GenBank, -r 2015-12-19T00:00:00Z, -l 500) and, in addition, the plasmids LN868944.1, LN868945.1 and LN868946.1 were blacklisted (-b). Genes had to be at least 100 nt long to be included in the database (-m 100) and were dereplicated (-d). The plasmids were stored as well (-k) to be available as the database for MOB-recon.

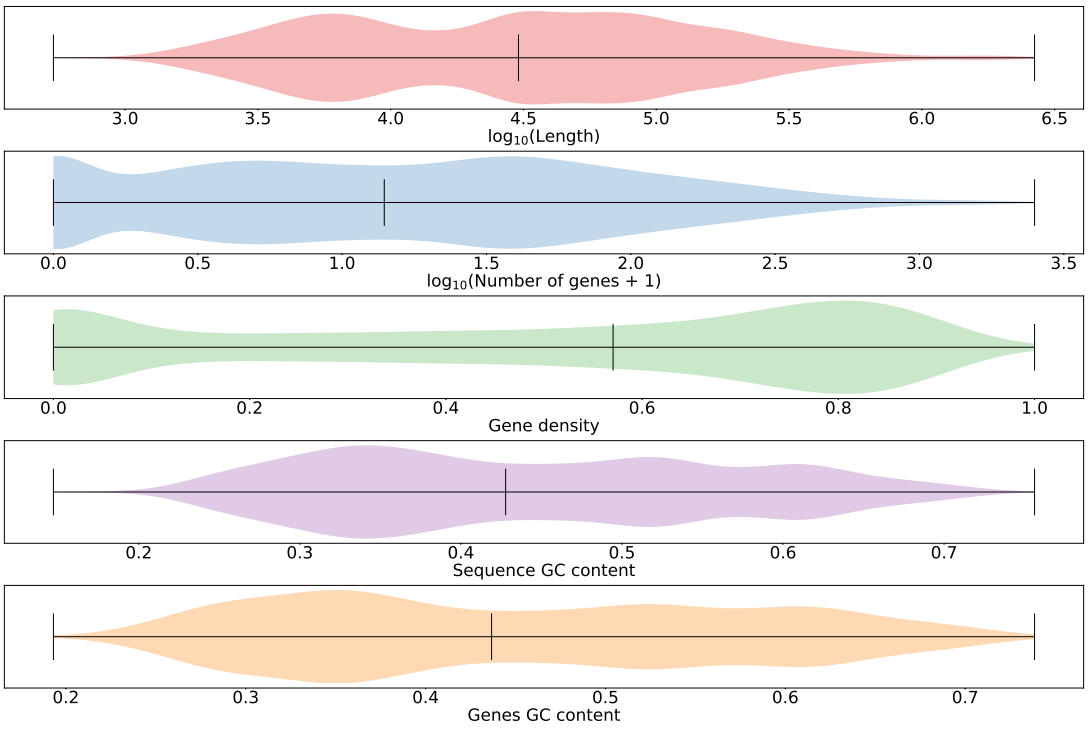
The three plasmids mentioned above were excluded from the database for their exceptional and strongly negative impact on the prediction quality. All three stem from the same assembly (GCF_001457675.1), are supposed to be plasmids of a *Salmonella enterica* subspecies and have the following characteristics:

Accession	Length (nt)	Number of genes
LN868944.1	727905	691
LN868945.1	147787	136
LN868946.1	141119	135

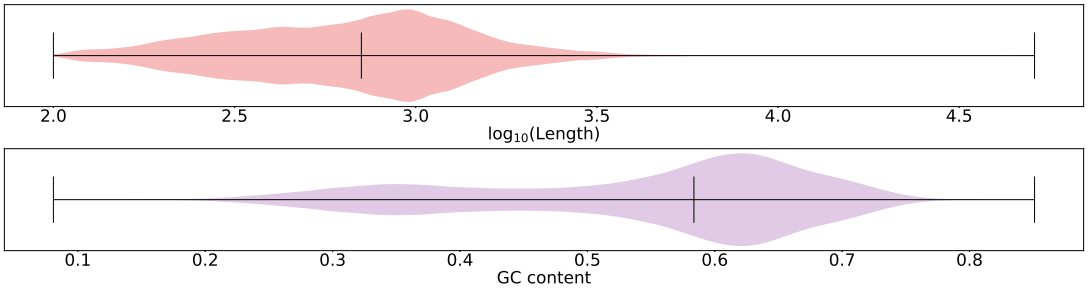
A first analysis (not shown here) with a NCBI-database still including these plasmids provided an overall F1 score of only 0.465295 (recall: 0.873862, precision: 0.317057) caused by very low precision scores on the 21 test samples from *Salmonella enterica*. The total precision on those 21 samples was only 0.114560, compared to 0.986203 after excluding the three plasmids, while the effect on the other test samples was very small. A closer inspection revealed that the genes of LN868944.1 make up between 9.2 and 12.3 % of the chromosomes in the 21 *Salmonella enterica* test samples. The genes of LN868945.1 and LN868946.1 cover between 1.6 and 2.6 % resp. 1.9 and 2.3 % of these chromosomes, while the genes of only 5 (of 157) other database plasmids from *Salmonella enterica* match at least one of the chromosomes. Their genes cover at most 0.6 % of a chromosome and usually way less. We then excluded the three plasmids of that single assembly from the database because they (or at least their gene collections) appeared to unusually chromosome-like and, thus, interfere with the capability of the greedy algorithm to distinguish between contigs of plasmid and chromosomal origin.

Supplement Figure S1 shows the distributions of the length, number of genes, gene density and GC content of all plasmids in the database. Furthermore, it displays the distribution of the overall GC content of all genes per gene-containing plasmids (4925 of 5822 contain at least one sufficiently long gene). Supplement Table S1 lists several statistics on the same characteristics.

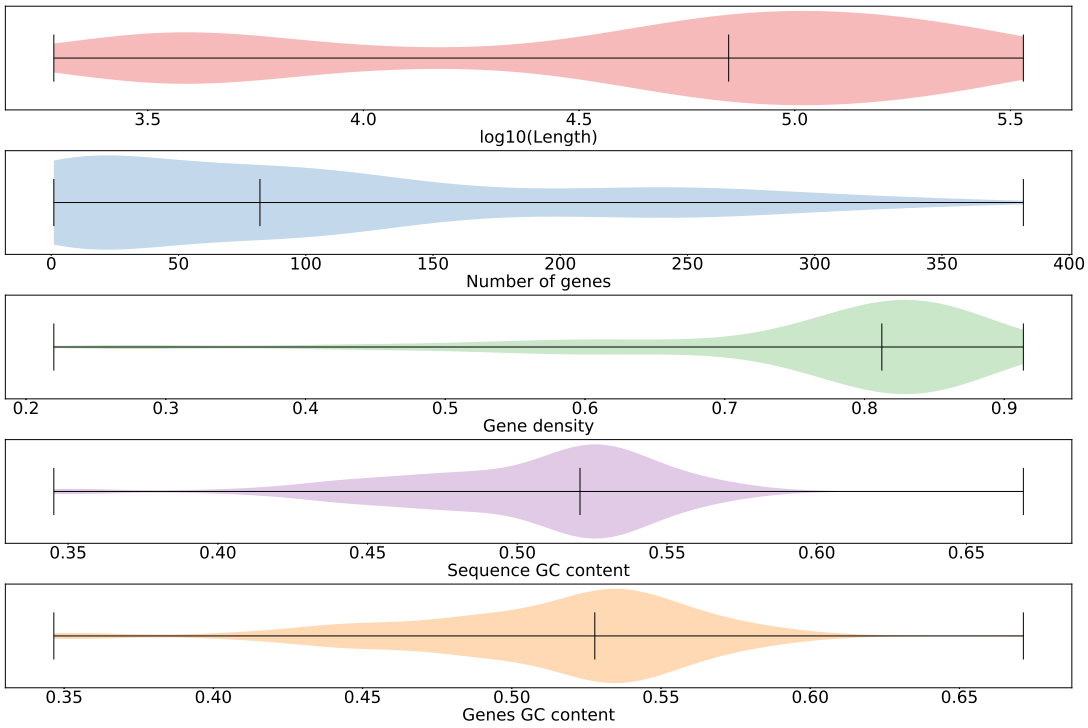
Similarly, Supplement Figure S2 and Supplement Table S2 provide information on the characteristics of the genes in the database.



Supplement Figure S1: Distributions of characteristics of the plasmids in the NCBI-database.



Supplement Figure S2: Distributions of characteristics of the genes in the NCBI-database.



Supplement Figure S3: Distributions of characteristics of the plasmids in the MOB-database.

Variable	Minimum	Mean	SD	Q1	Median	Q3	Maximum
Length	1916	90695	88663	6909	70277	133201	338850
Number of genes	1	103	97	9	82	155	382
Gene density	0.2199	0.7620	0.1373	0.7427	0.8124	0.8509	0.9137
Sequence GC content	0.3452	0.5050	0.0463	0.4810	0.5209	0.5324	0.6690
Genes GC content	0.3466	0.5128	0.0508	0.4871	0.5278	0.5428	0.6714

Supplement Table S3: Statistics on characteristics of the plasmids in the MOB-database. Values related to the length and the number of genes were rounded to the nearest whole number.

2.2 Characteristics of MOB-database plasmids

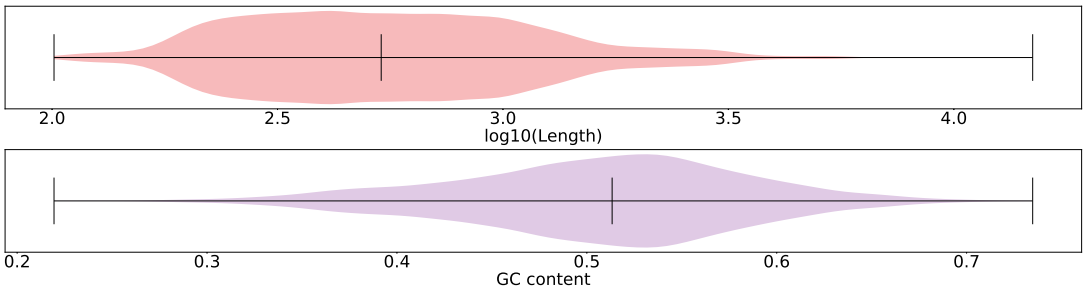
The MOB-database was built through the create command of HyAsP using an accession list (option -a) containing all 230 plasmids in the database samples of the MOB-suite benchmarking data set. All of them were longer than the required minimum length of 500 nt (-l 500). Again, the plasmids were stored (-k) and the genes were dereplicated and length-filtered (-d, -m 100).

Supplement Figure S3 shows the distributions of the length, number of genes, gene density and GC content of all plasmids in the database. Furthermore, it displays the distribution of the overall GC content of all genes per gene-containing plasmids (all 230 contain at least one sufficiently long gene). Supplement Table S3 lists several statistics on the same characteristics.

Similarly, Supplement Figure S4 and Supplement Table S4 provide information on the characteristics of the genes in the database.

Variable	Minimum	Mean	SD	Q1	Median	Q3	Maximum
Length	101	757	761	315	537	945	14958
GC content	0.2194	0.5065	0.0765	0.4596	0.5133	0.5575	0.7348

Supplement Table S4: Statistics on characteristics of the genes in the MOB-database. Values related to the length were rounded to the nearest whole number.



Supplement Figure S4: Distributions of characteristics of the genes in the MOB-database.

	Length	Number of genes	Gene density	Sequence-GC	Genes-GC
Minimum	830	1	0.041660	0.000106	0.000173
Mean	122903	126	0.788501	0.469267	0.476639
SD	234857	220	0.113209	0.117642	0.118312
0.005-quantile	1551	1	0.327631	0.238219	0.247857
0.025-quantile	2638	3	0.473261	0.264919	0.273834
0.25-quantile	20656	22	0.749200	0.365065	0.371116
Median	59334	66	0.819566	0.483765	0.491602
0.75-quantile	126385	140	0.862569	0.557292	0.568014
0.975-quantile	667190	627	0.920068	0.677752	0.684134
0.995-quantile	1777382	1633	0.937982	0.706170	0.709553
Maximum	2974672	2561	1.000000	0.874773	0.854296

Supplement Table S5: Statistics on basic characteristics of gene-containing plasmids based on the GenBank database. Values related to the length and the number of genes were rounded to the nearest whole number.

3 Establishing default values

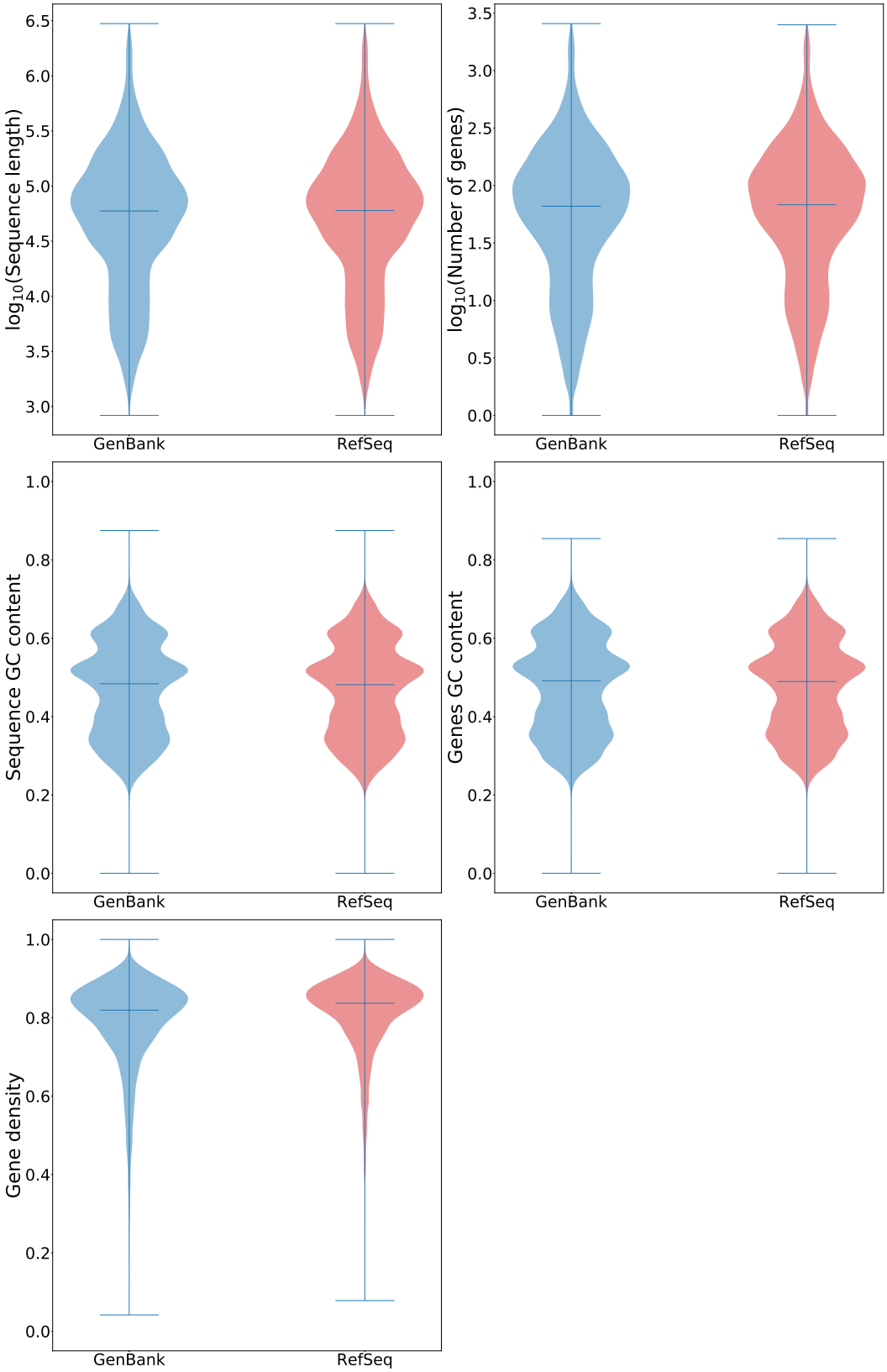
In order to obtain feasible default values for the parameters of HyAsP, we analysed plasmids from NCBI using table of available plasmids from `https://www.ncbi.nlm.nih.gov/genome/browse#!/plasmids/` (downloaded on 14 August 2018, 10:05 a.m.). Via the "Choose Columns" option on above website, we included all possible columns (especially the "Assembly" column). The downloaded file uses the comma as the separator but contains it also in the values. For the sake of an easier handling of the file, we changed the separator to tab and removed quotes surrounding values. Subsequently, we extracted the relevant information (organism name, RefSeq accession, GenBank accession, accession of associated assembly, release date) and kept only entries at assembly level "Complete Genome" and for which both a RefSeq and GenBank accession were provided. For each of the 8980 plasmid in the resulting, final table we then downloaded the reference sequences as well as the associated GenBank files and extracted the genes from the latter.

We analysed the plasmids based on the RefSeq resp. GenBank accession separately, since the plasmids are annotated differently in the two databases. While the observed differences were usually rather small, we provide the results of both analyses for the sake of completeness.

3.1 Basic characteristics of gene-containing plasmids

First, we analysed the basic characteristics of those plasmids with at least one known gene. Using the information available through the GenBank accession and RefSeq accession, 8563 resp. 8970 plasmids were retained. Besides their length, number of genes and gene density, we also computed the GC content of the whole sequence (sequence-GC) and the sequence parts representing genes (genes-GC) for each such plasmid. Tables S5 and S6 provide statistics on these characteristics and Figure S5 depicts their distributions.

Minimum and maximum plasmid length The shortest plasmid was 830 nt long and ignoring the shortest 0.5 resp. 2.5 % of the plasmids, the minimum changes to 1551 resp. 2664 nt. The length of the longest plasmid as 2,974,672 nt and ignoring the longest 0.5 resp. 2.5 % of the plasmids, the maximum changes to 1,754,877 resp. 659,748 nt.



Supplement Figure S5: Distributions of basic characteristics of gene-containing plasmids based on the GenBank database (upper) and RefSeq database (lower).

	Length	Number of genes	Gene density	Sequence-GC	Genes-GC
Minimum	830	1	0.078186	0.000106	0.000173
Mean	122364	128	0.810418	0.468863	0.476216
SD	231589	217	0.097872	0.116775	0.117088
0.005-quantile	1551	2	0.411738	0.237956	0.246834
0.025-quantile	2664	3	0.530248	0.265547	0.274146
0.25-quantile	21936	24	0.776131	0.365919	0.372421
Median	59926	68	0.837074	0.481501	0.489450
0.75-quantile	126426	143	0.873375	0.553694	0.563088
0.975-quantile	659749	624	0.923209	0.677505	0.682979
0.995-quantile	1754877	1635	0.940339	0.705774	0.709089
Maximum	2974672	2504	1.000000	0.874773	0.854296

Supplement Table S6: Statistics on basic characteristics of gene-containing plasmids based on the RefSeq database. Values related to the length and the number of genes were rounded to the nearest whole number.

Minimum gene density Using the GenBank database, the minimum observed gene density was approximately 4.6 %. Ignoring the 0.5 resp. 2.5 % of the plasmids with the lowest gene density, the minimum changes to 32.8 resp. 47.3 %. Using the RefSeq database, the minimum was higher at roughly 7.8 % and ignoring the 0.5 resp. 2.5 % of the plasmids with the lowest gene density, the minimum changes to 41.2 resp. 53.0 %.

The default values of the parameters min_length, max_length, min_gene_density were derived from these values. In order to increase the probability that seeds are indeed of plasmid origin, their gene-density threshold (min_seed_gene_density) is, by default, chosen to be higher (currently 50 %) than the one for the overall plasmid (min_gene_density).

3.2 Chromosomes versus plasmids

In addition, we compared the behaviour of the characteristics between chromosomes and plasmids. To that end, we first determined the accession numbers of all the different assemblies associated with the 8980 plasmids. For each such assembly, we then downloaded the assembly report from NCBI in order to obtain the accession numbers of the associated chromosomes and plasmids. Their GenBank files were temporarily downloaded as well and analysed to obtain information on the sequence length, number of genes, gene density, GC content and the number of intermediate nucleotides between non-overlapping genes. Tables S7 and S8 together with Figure S6 show statistics and the distributions of those characteristics (except the number of nucleotides, see below), respectively. An additional comparison between the chromosomes and plasmids per assembly is provided in Figures S7 and S8.

While there are plasmids which have a GC content very similar to the associated chromosomes, the vast majority of plasmids differs notably. Assuming that the overall GC content of the contigs in an assembly is dominated by the chromosome (due to their usually much higher length), the difference in GC content between the contig and the overall assembly might be a useful indicator of whether a contig should be considered of plasmid origin.

3.3 Variation of GC content within a plasmid

Next, we examined the variation of the GC content within a plasmid by computing the GC contents of 100 nt wide non-overlapping sliding windows (WGC). First, we determiend the minimum, mean, standard deviation, median and maximum for the windows of each plasmid. In a second round, we computed statistics on these per-plasmid statistics over all plasmids. Tables S9 and S10 provide these statistics and Figure S9 depicts the distributions of the per-sample statistics.

While the range of observed GC contents over all plasmids was quite large, the variability of WGC within a plasmid was relatively small. The standard deviation never exceeded 0.2 and for 99 % of the examined plasmids it fell between 0.045 and 0.123.

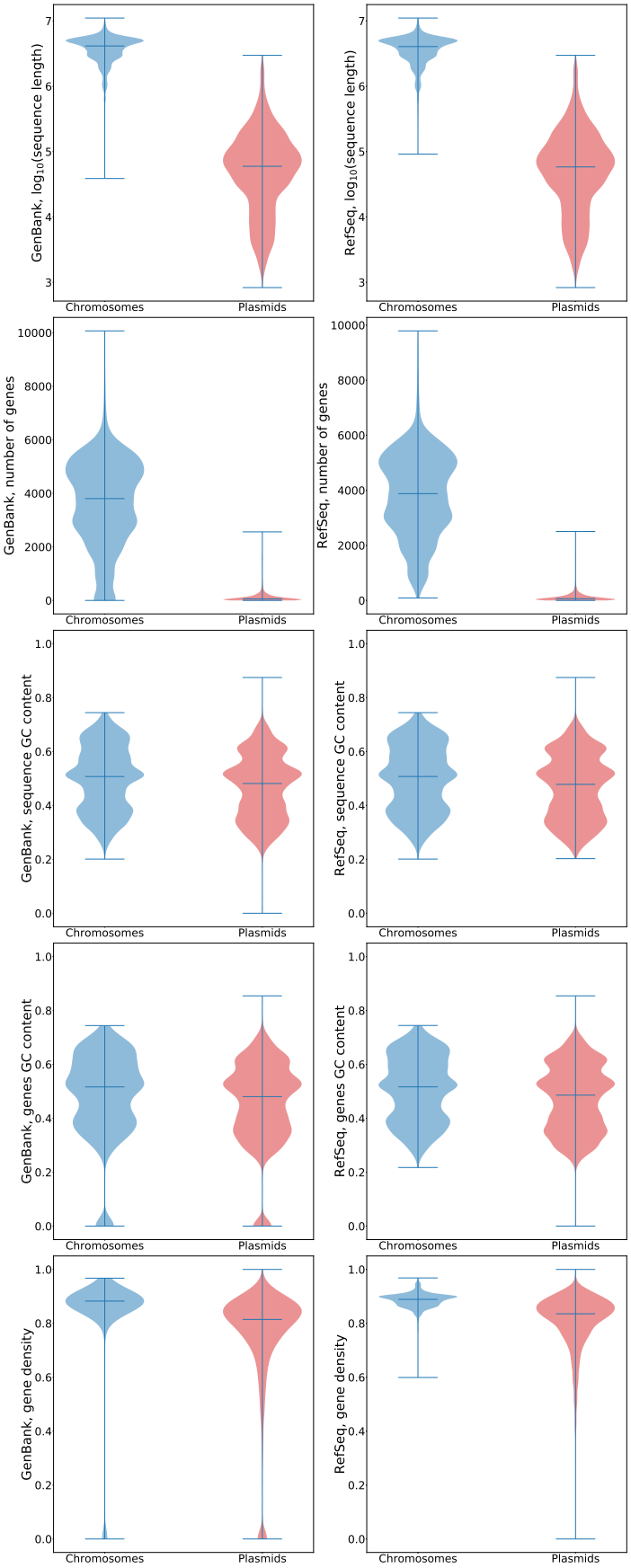
The parameter max_gc_diff was derived from these values.

		Length	Number of genes	Gene density	Sequence-GC	Genes-GC
Chromosomes	Minimum	38742	0	0.000000	0.200970	0.000000
	Mean	3987337	3630	0.834212	0.498654	0.482004
	SD	1538930	1677	0.197256	0.118686	0.160866
	0.005-quantile	477839	0	0.000000	0.254400	0.000000
	0.025-quantile	922298	0	0.000000	0.284794	0.000000
	0.25-quantile	2873127	2597	0.858477	0.390620	0.389918
	Median	4137160	3804	0.882626	0.507418	0.516504
	0.75-quantile	5102459	4915	0.897490	0.592867	0.596896
	0.975-quantile	6768600	6160	0.937481	0.694425	0.696463
	0.995-quantile	8863445	7920	0.950963	0.729432	0.730567
	Maximum	11064963	10068	0.967555	0.744350	0.744087
Plasmids	Minimum	830	0	0.000000	0.000106	0.000000
	Mean	122034	120	0.751627	0.468874	0.454343
	SD	231400	216	0.199873	0.116689	0.153119
	0.005-quantile	1550	0	0.000000	0.237700	0.000000
	0.025-quantile	2615	0	0.000000	0.265547	0.000000
	0.25-quantile	21597	16	0.734182	0.366035	0.359239
	Median	59633	61	0.814563	0.481415	0.480620
	0.75-quantile	126255	134	0.860672	0.553334	0.561269
	0.975-quantile	648452	601	0.919704	0.677518	0.682739
	0.995-quantile	1755368	1622	0.937889	0.704254	0.708983
	Maximum	2974672	2561	1.000000	0.874773	0.854296

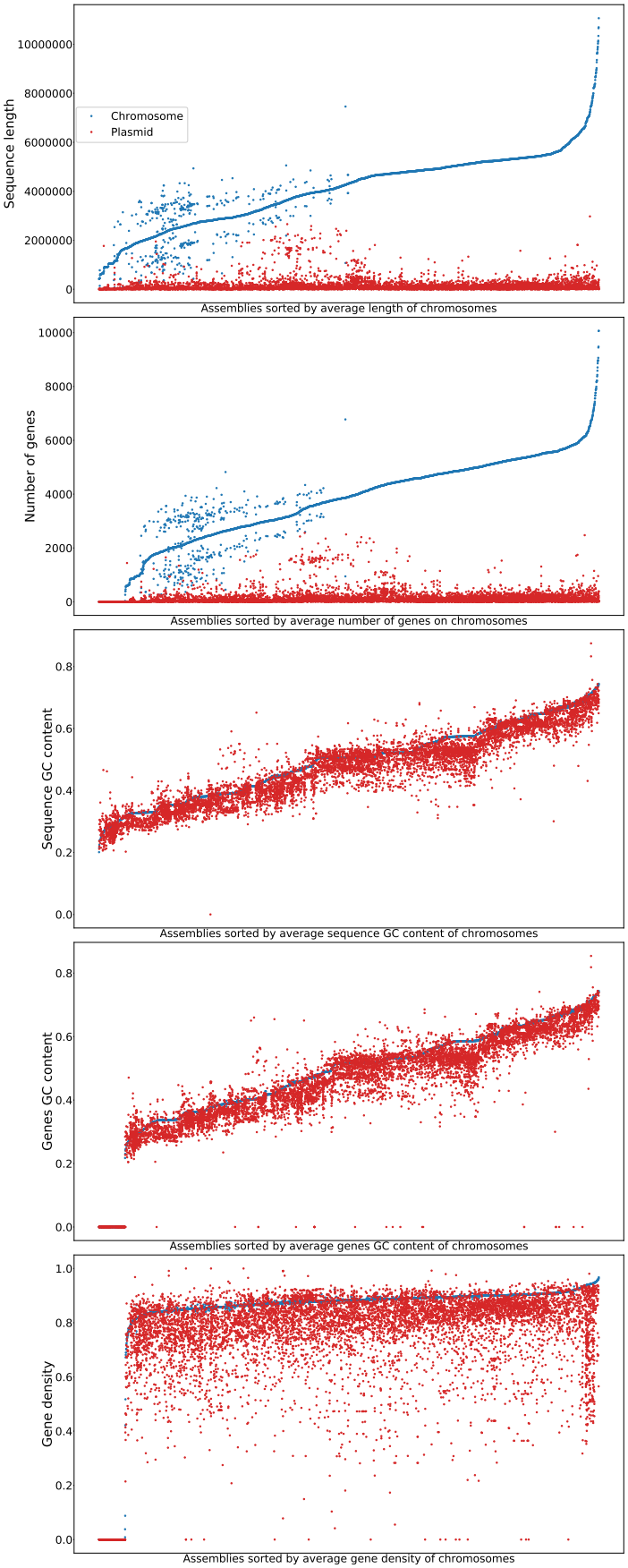
Supplement Table S7: Statistics on basic characteristics of chromosomes and plasmids based on the GenBank database. Values related to the length and the number of genes were rounded to the nearest whole number.

		Length	Number of genes	Gene density	Sequence-GC	Genes-GC
Chromosomes	Minimum	91776	90	0.599156	0.200970	0.217553
	Mean	3941123	3835	0.885656	0.499401	0.507697
	SD	1564820	1515	0.026939	0.120824	0.119642
	0.005-quantile	447657	416	0.787493	0.254420	0.265699
	0.025-quantile	910664	858	0.830633	0.283788	0.293554
	0.25-quantile	2834059	2818	0.869221	0.390553	0.401573
	Median	4050034	3880	0.889050	0.507452	0.516777
	0.75-quantile	5072582	5007	0.901655	0.599533	0.607748
	0.975-quantile	6825306	6256	0.941366	0.693353	0.696654
	0.995-quantile	8948782	8023	0.954070	0.729398	0.730100
	Maximum	11064963	9791	0.968535	0.744350	0.744461
Plasmids	Minimum	830	0	0.000000	0.202597	0.000000
	Mean	124814	129	0.807570	0.468138	0.474875
	SD	242432	227	0.104259	0.119656	0.121031
	0.005-quantile	1549	1	0.401569	0.235895	0.241007
	0.025-quantile	2619	3	0.515296	0.263095	0.271034
	0.25-quantile	20371	22	0.773161	0.361519	0.366787
	Median	58299	65	0.835421	0.478330	0.485923
	0.75-quantile	126483	142	0.873947	0.564364	0.573646
	0.975-quantile	707687	654	0.923876	0.676454	0.682358
	0.995-quantile	1810943	1639	0.940694	0.702445	0.708127
	Maximum	2974672	2504	1.000000	0.874773	0.854296

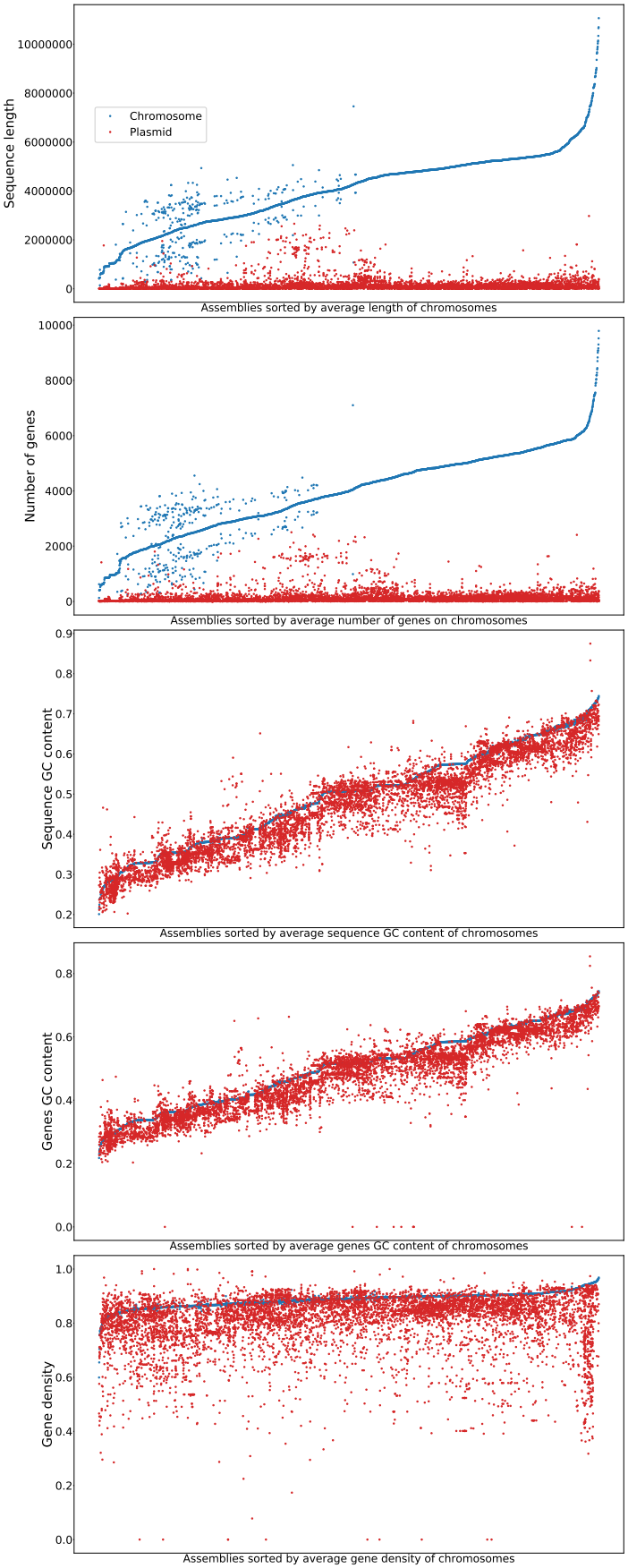
Supplement Table S8: Statistics on basic characteristics of chromosomes and plasmids based on the RefSeq database. Values related to the length and the number of genes were rounded to the nearest whole number.



Supplement Figure S6: Distributions of basic characteristics of plasmids and chromosomes based on the GenBank database (left) and RefSeq database (right). The y-axes are shared per row in above plot. The accumulation of very small values for genes-GC and the gene gensity is caused by chromosomes and plasmids without any annotated gene.



Supplement Figure S7: Basic characteristics of plasmids and chromosomes per assembly based on the GenBank database. The assemblies are sorted by the average value of the respective characteristic of the chromosomes in each assembly.



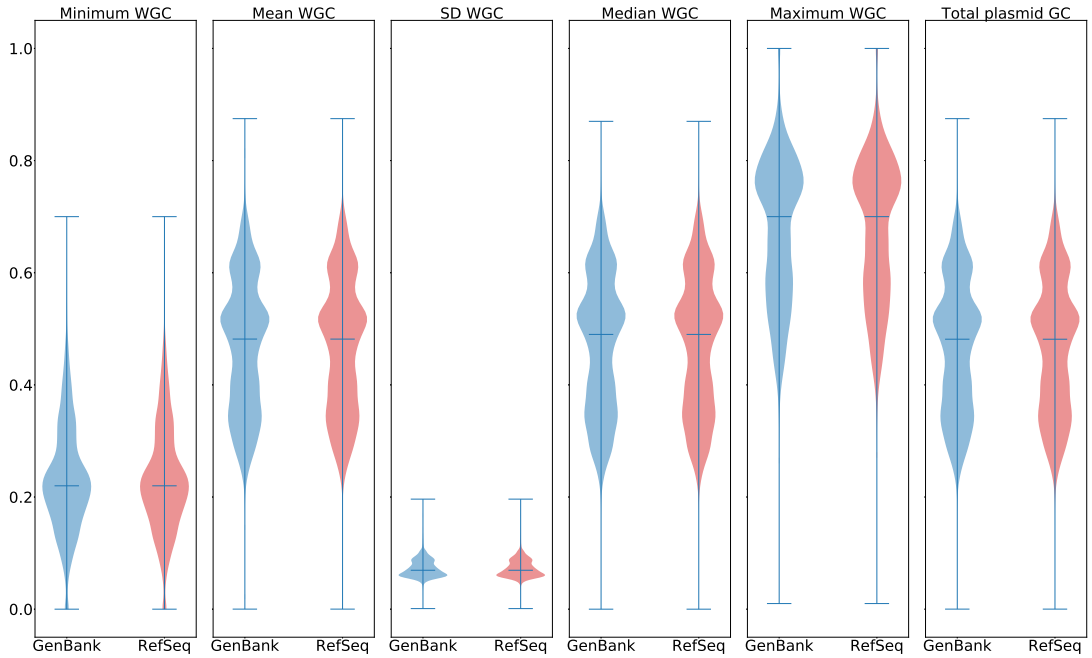
Supplement Figure S8: Basic characteristics of plasmids and chromosomes per assembly based on the RefSeq database. The assemblies are sorted by the average value of the respective characteristic of the chromosomes in each assembly.

	Minimum WGC	Mean WGC	SD WGC	Median WGC	Maximum WGC
Minimum	0.000000	0.000106	0.001026	0.000000	0.010000
Mean	0.231942	0.468866	0.072729	0.471610	0.674040
SD	0.096268	0.116774	0.015509	0.119791	0.127478
0.005-quantile	0.000000	0.237269	0.045463	0.230000	0.380000
0.025-quantile	0.060000	0.265603	0.051213	0.260000	0.430000
0.25-quantile	0.170000	0.366493	0.060631	0.365000	0.570000
Median	0.220000	0.481635	0.069405	0.490000	0.700000
0.75-quantile	0.290000	0.553485	0.083947	0.560000	0.770000
0.975-quantile	0.440000	0.677580	0.105043	0.680000	0.870000
0.995-quantile	0.500000	0.705752	0.122506	0.710000	1.000000
Maximum	0.700000	0.874782	0.196249	0.870000	1.000000

Supplement Table S9: Statistics on per-plasmid WGC statistics based on the GenBank database.

	Minimum WGC	Mean WGC	SD WGC	Median WGC	Maximum WGC
Minimum	0.000000	0.000106	0.001026	0.000000	0.010000
Mean	0.231960	0.468881	0.072728	0.471624	0.674056
SD	0.096279	0.116776	0.015509	0.119792	0.127480
0.005-quantile	0.000000	0.237271	0.045463	0.230000	0.380000
0.025-quantile	0.060000	0.265605	0.051214	0.260000	0.430000
0.25-quantile	0.170000	0.366512	0.060627	0.365000	0.570000
Median	0.220000	0.481701	0.069401	0.490000	0.700000
0.75-quantile	0.290000	0.553522	0.083946	0.560000	0.770000
0.975-quantile	0.440000	0.677579	0.105042	0.680000	0.870000
0.995-quantile	0.500000	0.705749	0.122505	0.710000	1.000000
Maximum	0.700000	0.874782	0.196249	0.870000	1.000000

Supplement Table S10: Statistics on per-plasmid WGC statistics based on the RefSeq database.



Supplement Figure S9: Distributions of per-plasmid WGC statistics based on the GenkBank and RefSeq database.

	All	Mean per plasmid	Median per plasmid
Minimum	0	1	0
Mean	191	226	146
SD	268	171	166
0.005-quantile	0	44	15
0.025-quantile	1	81	32
0.25-quantile	35	147	81
Median	104	192	111
0.75-quantile	243	258	163
0.975-quantile	876	594	480
0.995-quantile	1537	971	929
Maximum	20928	6132	6118

Supplement Table S11: Statistics on the number of intermediate nucleotides between neighbouring, non-overlapping genes in plasmids based on the GenBank database.

	All	Mean per plasmid	Median per plasmid
Minimum	0	3	0
Mean	175	200	131
SD	230	119	116
0.005-quantile	0	46	16
0.025-quantile	1	81	33
0.25-quantile	34	137	77
Median	100	172	103
0.75-quantile	229	228	148
0.975-quantile	772	475	389
0.995-quantile	1298	808	748
Maximum	10990	4022	4854

Supplement Table S12: Statistics on the number of intermediate nucleotides between neighbouring, non-overlapping genes in plasmids based on the GenBank database.

3.4 Number of intermediate nucleotides between genes

Finally, we examined the number of nucleotides between neighbouring, non-overlapping genes in plasmids. For both databases, we computed the statistics over the pooled distances from all plasmids and over the means resp. medians of the distances per plasmid.

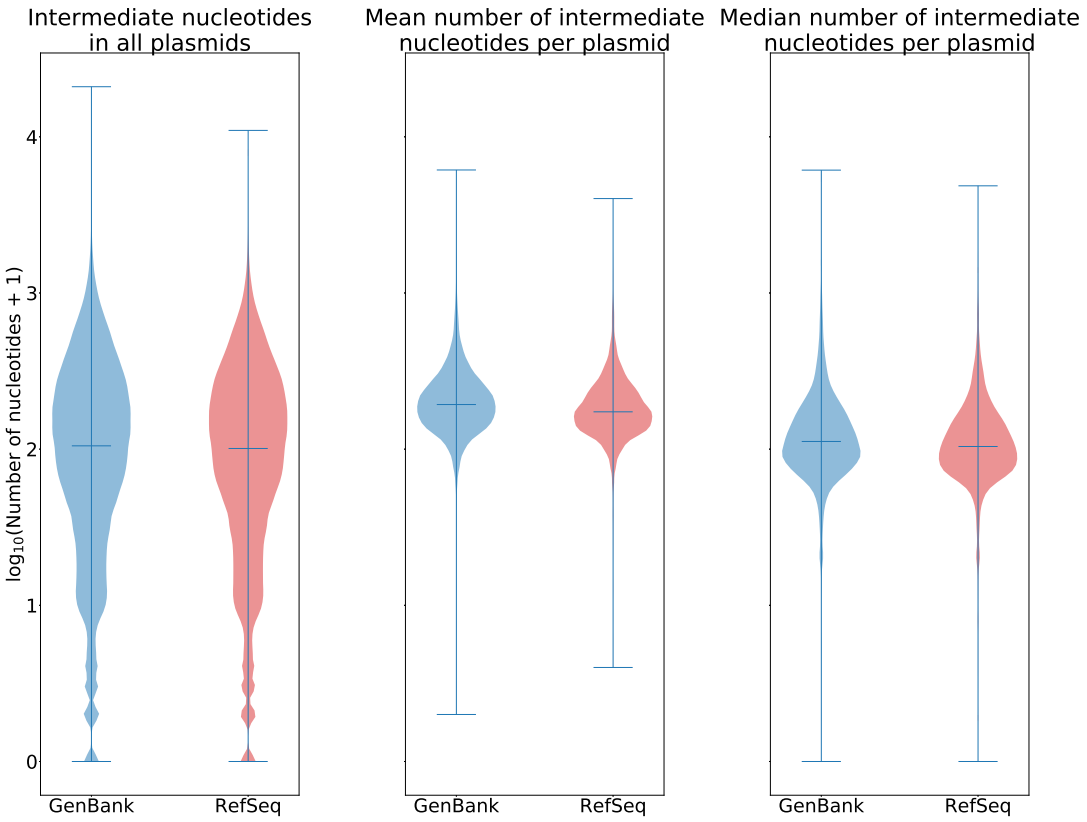
The number of nucleotides between genes is on average below 200 (GenBank: 190, RefSeq: 174) and rarely exceeds 1600. Based on the GenBank database, the maximum number of nucleotides between genes is 20928 nt and ignoring the largest 0.5 resp. 2.5 % of the gaps, the maximum changes to 1537 resp. 876 nt. Using the RefSeq database, the maximum was 10990 nt. Ignoring the largest 0.5 resp. 2.5 % of the gaps, the maximum changes to 1298 resp. 772 nt.

The parameter max_intermediate_nt was derived from these values. The median contig length varied between a few hundred and a few thousand nucleotides in the assembly graphs from preliminary tests (not shown here). Hence, parameter max_intermediate_contigs was chosen to be a small integer.

3.5 Other parameters

HyAsP has been developed using assemblies obtained from Unicycler, which outputs normalised read-coverage values. Thus, the read coverage can be smaller than 1 and we observed in preliminary tests (not shown here) that the read coverage of plasmids (correctly predicted with high probability) can be below 1 as well. Consequently, min_plasmid_read_depth was chosen as some lower proportion of the median read depth of the assembly graph. In order to account for possible (downward) variations, the default value of min_read_depth is even a bit lower than but related to min_plasmid_read_depth.

So far, there is no clear indication for a good threshold on the maximum score of an extension (max_score) and the minimum length of the overlap between the ends of a plasmid to confidently circularise it (overlap_ends). Therefore, both features are deactivated by setting their default values to infinity.



Supplement Figure S10: Distributions of the number of intermediate nucleotides between neighbouring, non-overlapping genes in plasmids.

Similarly, preliminary tests led to similar results and, thus, provided no clear conclusion on several other options. For this reason, the relative size of the scoring-function weights (score_weights) was chosen to be balanced by default. Furthermore, the mean instead of the median (use_median) is used to compute the average read depth of a plasmid. The mean was given the preference to the median because a change of newly added contigs in this characteristic becomes visible more quickly during the plasmid construction. Also, HyAsP performs link-based instead of node-based extensions as this allows repeats (based on the same contig) to occur in the predicted plasmids.

Organism	Test samples	MOB-database samples	NCBI-database samples
Aeromonas veronii	1	0	0
Citrobacter freundii	1	3	27
Enterococcus faecium	2	0	34
Escherichia coli	30	5	261
Klebsiella aerogenes	1	2	4
Klebsiella oxytoca	1	5	27
Klebsiella pneumoniae	9	28	187
Salmonella enterica	21	12	157

Supplement Table S13: Number of plasmids per species in the test and database samples. The organisms of the same species can be from different strains or subspecies.

4 Experimental design

This sections provides the more (technical) details on the comparison of HyAsP with plasmidSPAdes (Antipov *et al.* (2016)) and MOB-recon (Robertson and Nash (2018)).

4.1 Data preparation

The plasmid databases (see Section 2 for the details) were generated using the create command HyAsP. The plasmids from the database samples formed the plasmid database needed by MOB-recon (after clustering them using MOB-cluster, another tool from MOB-suite, and creating Mash (Ondov *et al.* (2016)) sketches for them), while the gene collection was used as the existing knowledge for the greedy algorithm. Table S13 shows how often different species feature on the database and test samples.

4.2 Metrics

The predictions of all tools were assessed in terms of their *precision* (proportion of predicted plasmids corresponding to references) and *recall* (proportion of reference plasmids corresponding to predictions) by matching the putative plasmids (greedy algorithm) resp. plasmid contigs (MOB-recon, plasmidSPAdes) against the reference plasmids through BLAST – more precisely, BLAST+ (blastn v2.6.0, Camacho *et al.* (2009)). In order to compute these metrics, we relied on the positions of the BLAST matches between predicted plasmid contigs (as query, from qstart to qend) and reference plasmids (as subject, from sstart to send). For the greedy algorithm, we used the predicted plasmid as a contig collection of size 1. For a BLAST match between a contig A and a plasmid B, we defined the interval of matching positions on the query and the subject as $\mathsf{M}^{(Q)}_{\mathsf{A,B}} = [\mathsf{qstart..qend}]$ and $\mathsf{M}^{(S)}_{\mathsf{A,B}} = [\mathsf{sstart..send}]$, respectively. For each tool and sample, we subsequently assessed set of predicted plasmids $\mathcal{P}$ and set of reference plasmids $\mathcal{R}$ by computing precision and recall per plasmid:

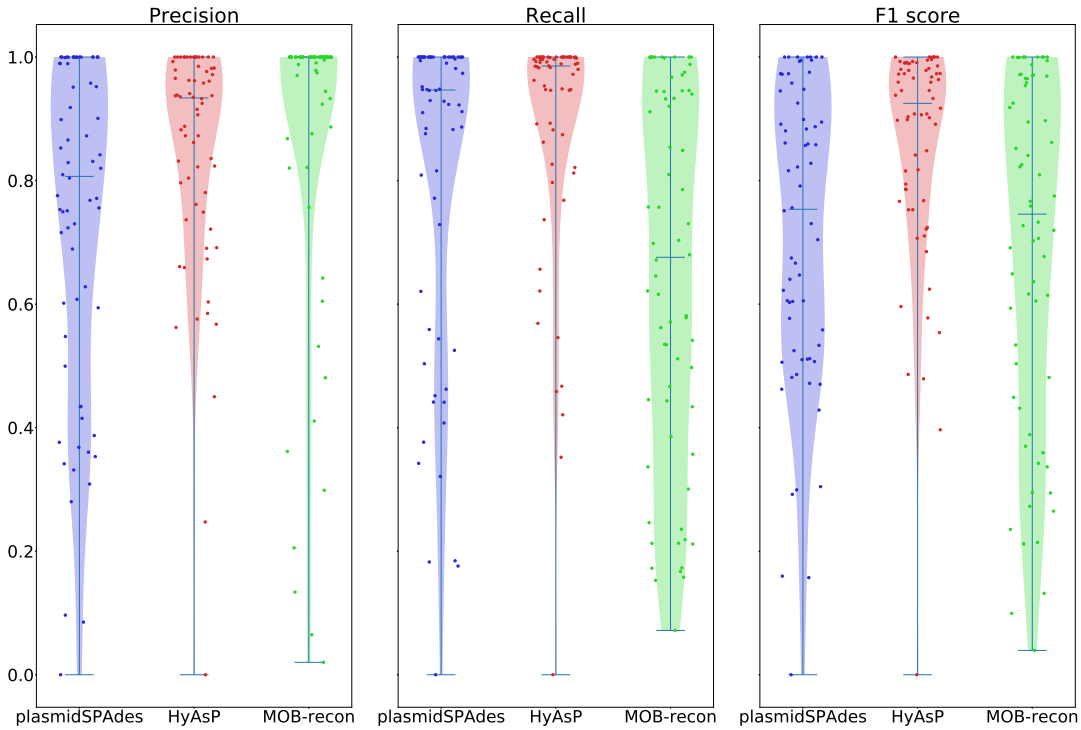
$$\mathsf{recall}(\mathsf{R}) = \left| \bigcup_{\substack{contig\ \mathsf{C}\ in\ \mathsf{P},\\ \mathsf{P}\in\mathcal{P}}} \mathsf{M}^{(S)}_{\mathsf{C,R}} \right| \Big/ \mathsf{length}(\mathsf{R}), \quad \forall\, \mathsf{R} \in \mathcal{R}$$

$$\mathsf{precision}(\mathsf{P}) = \left(\sum_{contig\ \mathsf{C}\ in\ \mathsf{P}} \left| \bigcup_{\mathsf{R}\in\mathcal{R}} \mathsf{M}^{(Q)}_{\mathsf{C,R}} \right| \right) \Big/ \mathsf{length}(\mathsf{P}), \quad \forall\, \mathsf{P} \in \mathcal{P}$$

Note that above unions of the position intervals do not create multisets, i.e. positions covered by multiple intervals still contribute only once. The precision and recall values of the predicted resp. reference plasmids were then summarised per sample S:

$$\mathsf{sample\_recall}(\mathsf{S}) = \sum_{\mathsf{R}\in\mathcal{R}} (\mathsf{length}(\mathsf{R}) * \mathsf{recall}(\mathsf{R})) \Big/ \sum_{\mathsf{R}\in\mathcal{R}} \mathsf{length}(\mathsf{R})$$

$$\mathsf{sample\_precision}(\mathsf{S}) = \sum_{\mathsf{P}\in\mathcal{P}} (\mathsf{length}(\mathsf{P}) * \mathsf{precision}(\mathsf{P})) \Big/ \sum_{\mathsf{P}\in\mathcal{P}} \mathsf{length}(\mathsf{P})$$



Supplement Figure S11: Distributions of precision, recall and F1 score of HyAsP, plasmidSPAdes and MOB-recon on the test samples using the NCBI-database. The horizontal lines in each violin plot represent the minimum, median and maximum value of the respective metric-tool combination.

4.3 Tools

In our evaluation, we ran all algorithms with default parameters if not stated otherwise. SPAdes (Bankevich *et al.* (2012), v3.12.0) was used with –plasmid to actually use plasmidSPAdes and –careful to reduce the number of mismatches and indels, while MOB-recon (v1.4.1) needed –plasmid_db and –plasmid_mash_db to use non-default plasmid databases. HyAsP was run as part of pipeline described in Section 1.6. For the sake of a consistent comparison, the resulting preprocessed FASTQ reads and Unicycler assembly were used as inputs for plasmidSPAdes and MOB-recon, respectively.

5 Results

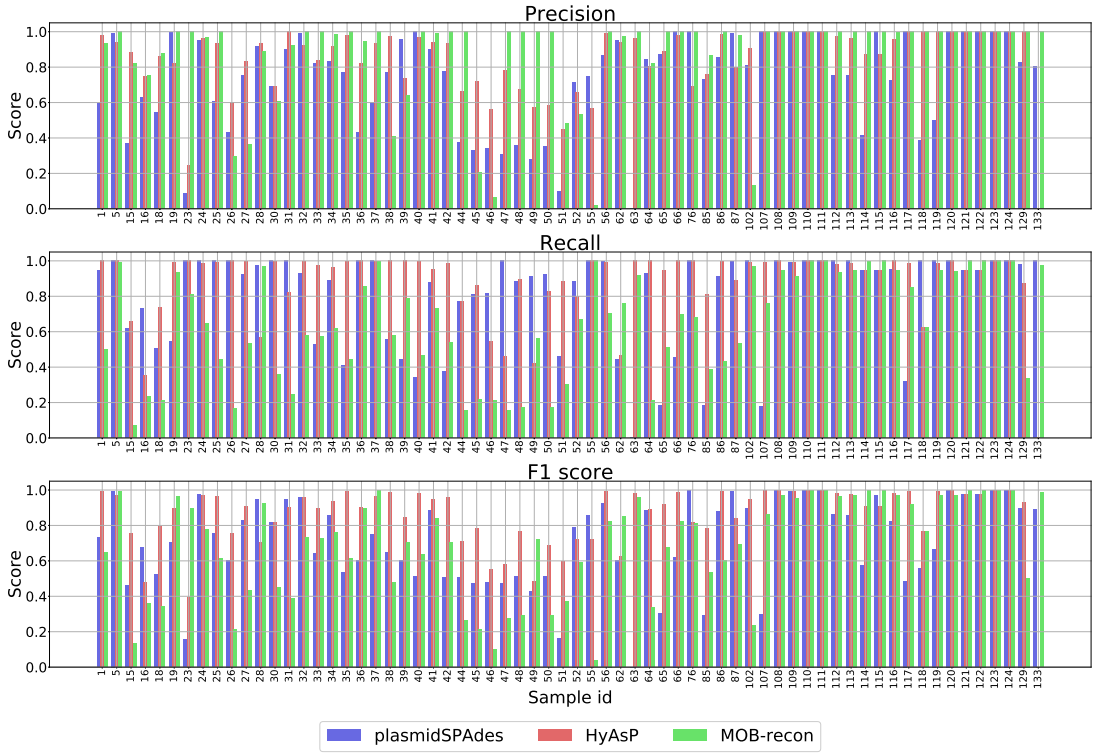
This section shows the remaining results of the experiments using the NCBI-database and the MOB-database, respectively.

5.1 Analysis with NCBI-database

The following table restates the total precision, recall and F1 score of our greedy algorithm, plasmidSPAdes and MOB-recon over all 66 test samples:

Tool	Precision	Recall	F1 score
HyAsP	0.871445	0.898822	0.884922
plasmidSPAdes	0.659211	0.741983	0.698152
MOB-recon	0.760241	0.583909	0.660509

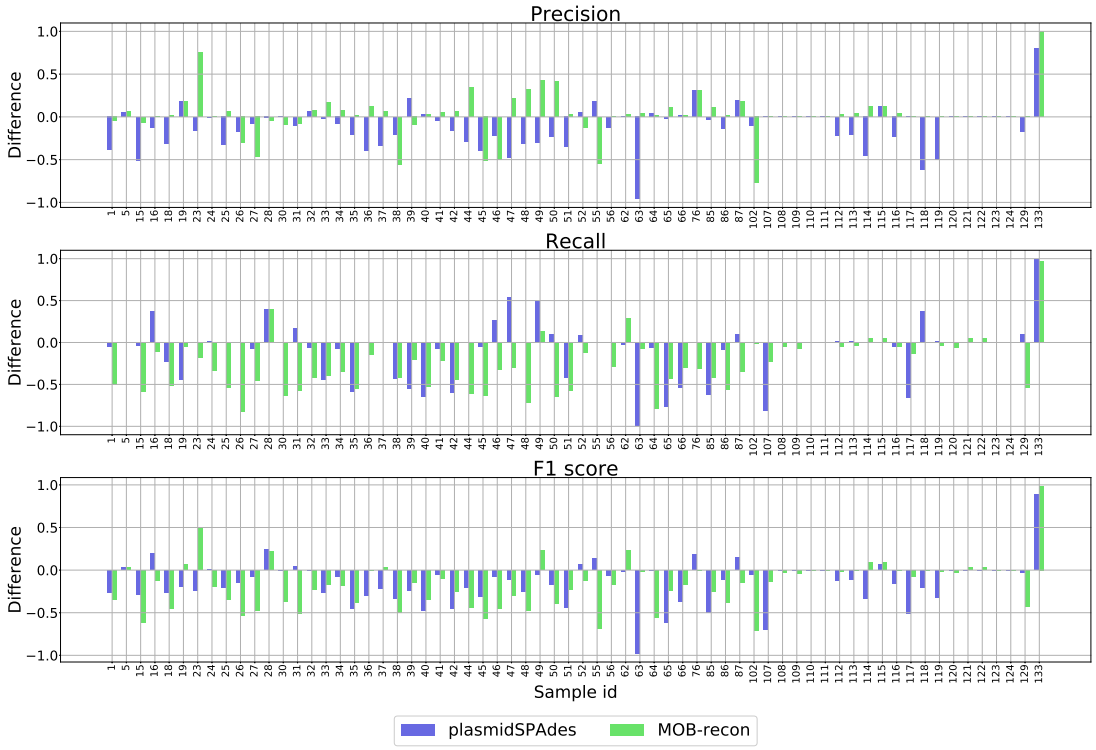
Figure S12 shows that all three tools varied vastly in the metrics across the collection of test samples. Each tool attained a precision, recall and F1 score of 1.0 on several samples (sometimes even at the same time), but all of them also performed weakly on some samples with scores equal to or close 0. While there were samples with high scores for all three tools (e.g. sample 124), other samples differed notably in difficulty for the tools. For example, plasmidSPAdes and MOB-recon performed very well on sample 133, while the greedy algorithm did not even find part of the expected plasmid. On sample 63 (resp. 55), in turn, only the greedy algorithm and MOB-recon (resp. plasmidSPAdes) attained high scores. Additional information on how the scores were distributed for the different tools are provided in Table S14 and Figure S11.



Supplement Figure S12: Precision, recall and F1 score per test sample of HyAsP, plasmidSPAdes and MOB-recon using the NCBI-database.

Metric	Tool	Minimum	Mean	SD	Q1	median	Q3	Maximum
	HyAsP	0.0000	0.8483	0.1929	0.7662	0.9338	0.9843	1.0000
Precision	plasmidSPAdes	0.0000	0.7319	0.2728	0.5592	0.8068	0.9897	1.0000
	MOB-recon	0.0201	0.8709	0.2579	0.8960	1.0000	1.0000	1.0000
	HyAsP	0.0000	0.8894	0.1974	0.8763	0.9855	1.0000	1.0000
Recall	plasmidSPAdes	0.0000	0.8082	0.2705	0.6477	0.9467	1.0000	1.0000
	MOB-recon	0.0717	0.6486	0.2987	0.4362	0.6758	0.9446	1.0000
	HyAsP	0.0000	0.8549	0.1842	0.7721	0.9250	0.9812	1.0000
F1 score	plasmidSPAdes	0.0000	0.7128	0.2483	0.5149	0.7535	0.9402	1.0000
	MOB-recon	0.0394	0.6901	0.2855	0.4571	0.7457	0.9653	1.0000

Supplement Table S14: Statistics on precision, recall and F1 score of HyAsP, plasmidSPAdes and MOB-recon across all test samples using the NCBI-database.



Supplement Figure S13: Difference in precision, recall and F1 score per test sample between HyAsP and plasmidSPAdes and MOB-recon using the NCBI-database. The zero line represents the respective score of HyAsP. Positive (negative) values indicate that the other tool performed better (worse) than the greedy algorithm.

Figure S13 details how much plasmidSPAdes and MOB-recon performed better or worse than HyAsP on the different samples. The plots show that the greedy algorithm attained higher F1 scores on a broad range of samples, while MOB-recon (plasmidSPAdes) showed a better precision (recall) on several samples.

Figure S14 shows the quality of the predictions after grouping the samples based on the associated organism at the species level. HyAsP performed similarly well or better than plasmidSPAdes and MOB-recon for all observed species except *Klebsiella aerogenes*, for which it fell 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).

Since HyAsP provides information on the order and orientation of contigs by predicting entire plasmid sequences, we also briefly analysed the occurrence of misassembly events using QUAST (Gurevich *et al.* (2013), v4.6.3). Figure S15 shows that misassemblies occurred in the majority of samples and that relocations were the dominant misassembly type.

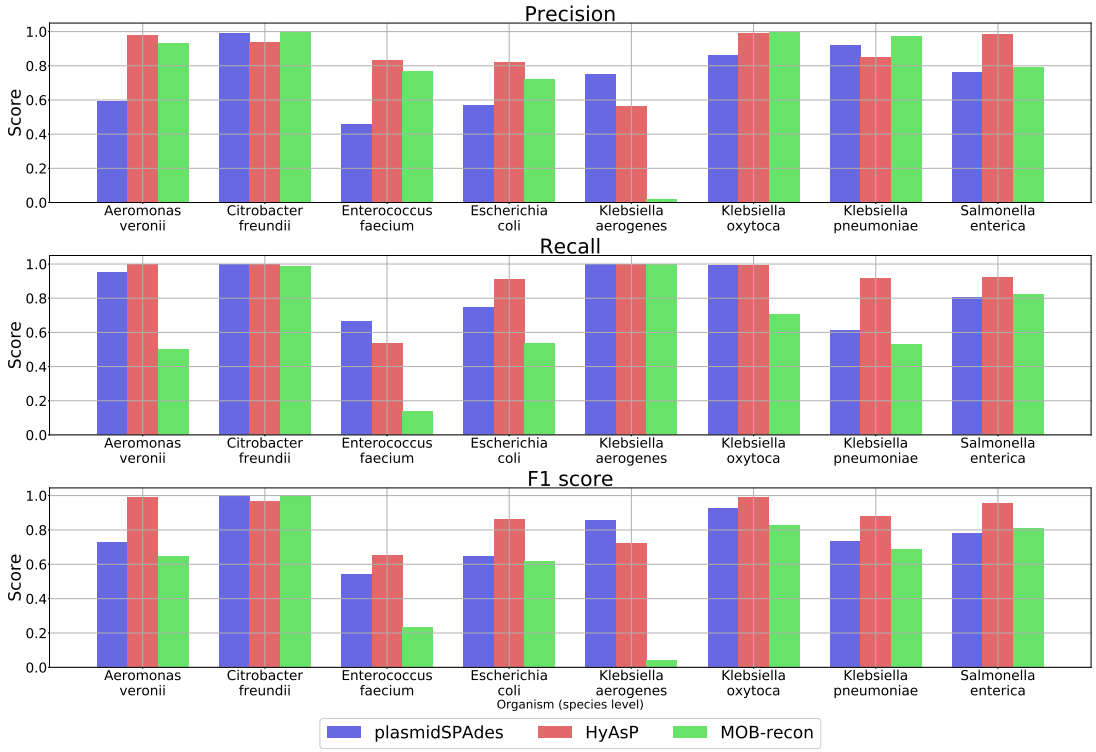
5.2 Analysis with MOB-database

First, we again summarised the analysis by computing the total precision, recall and F1 score of our greedy algorithm, plasmidSPAdes and MOB-recon over all 66 test samples:

Tool	Precision	Recall	F1 score
HyAsP	0.934515	0.775169	0.847416
plasmidSPAdes	0.659211	0.741983	0.698152
MOB-recon	0.760241	0.583909	0.660509

HyAsP outperformed plasmidSPAdes and MOB-recon in both total precision and recall. While plasmidSPAdes showed a better recall than MOB-recon, its precision was lower, overall leading to a similar F1 score which was notably lower than the F1 score of HyAsP.

Figure S17 shows that all three tools again varied vastly in the metrics across the collection of test samples. As before, each tool attained a precision, recall and F1 score of 1.0 on several samples (sometimes



Supplement Figure S14: Precision, recall and F1 score per species of HyAsP, plasmidSPAdes and MOB-recon on the test samples using the NCBI-database.

Metric	Tool	Minimum	Mean	SD	Q1	median	Q3	Maximum
	HyAsP	0.0000	0.8841	0.2446	0.8875	0.9985	1.0000	1.0000
Precision	plasmidSPAdes	0.0000	0.7319	0.2728	0.5592	0.8068	0.9897	1.0000
	MOB-recon	0.0201	0.8709	0.2579	0.8960	1.0000	1.0000	1.0000
	HyAsP	0.0000	0.8079	0.2628	0.7017	0.9375	0.9962	1.0000
Recall	plasmidSPAdes	0.0000	0.8082	0.2705	0.6477	0.9467	1.0000	1.0000
	MOB-recon	0.0717	0.6486	0.2987	0.4362	0.6758	0.9446	1.0000
	HyAsP	0.0000	0.8305	0.2441	0.7792	0.9146	0.9919	1.0000
F1 score	plasmidSPAdes	0.0000	0.7128	0.2483	0.5149	0.7535	0.9402	1.0000
	MOB-recon	0.0394	0.6901	0.2855	0.4571	0.7457	0.9653	1.0000

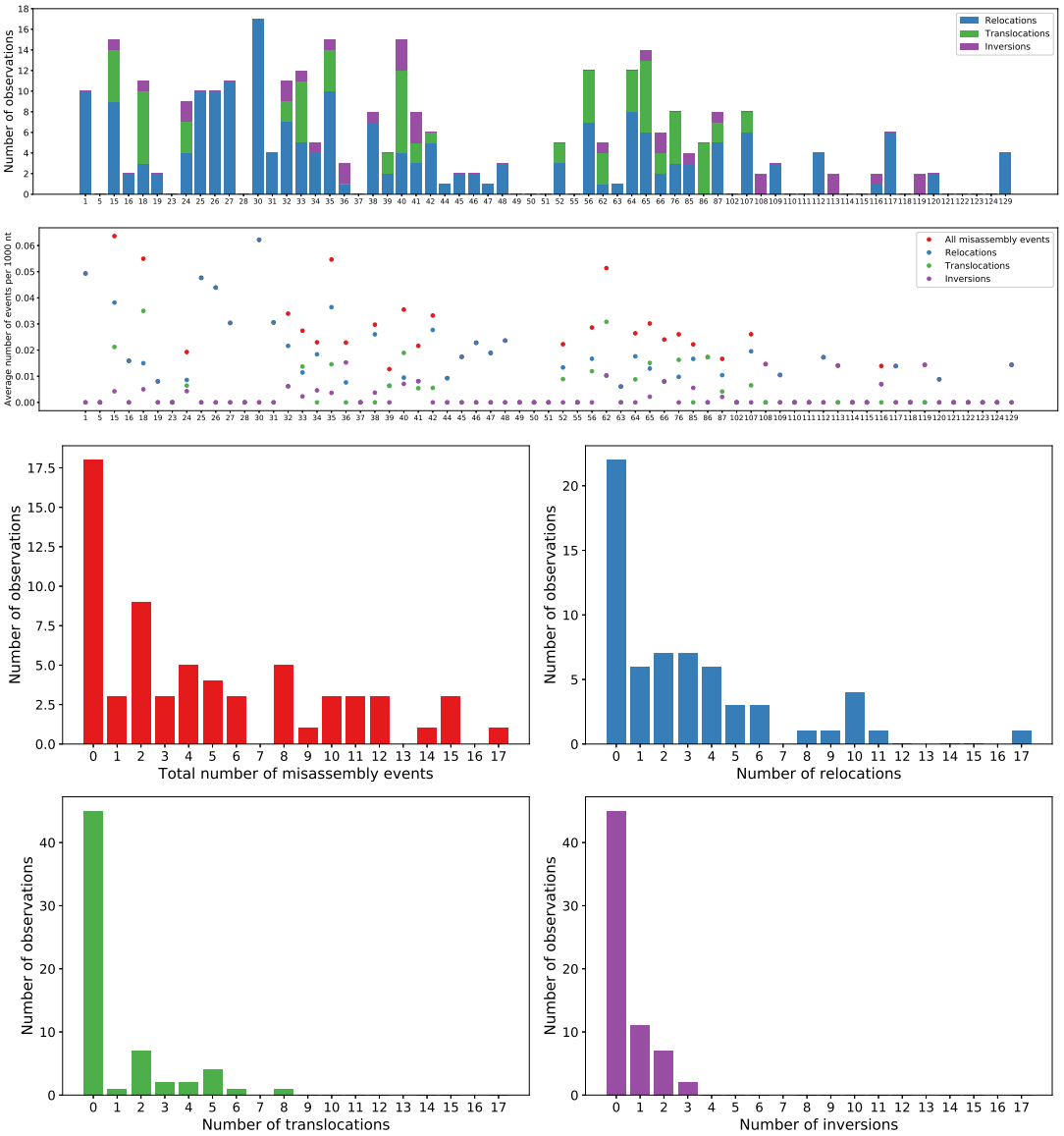
Supplement Table S15: Statistics on precision, recall and F1 score of HyAsP, plasmidSPAdes and MOB-recon across all test samples using the MOB-database.

even at the same time), but all of them also performed weakly on some samples with scores equal to or close 0. The differences in difficulty of the individual samples for the tools did not differ much from the analysis with the NCBI-database. Additional information on how the scores were distributed for the different tools are provided in Table S15 and Figure S16.

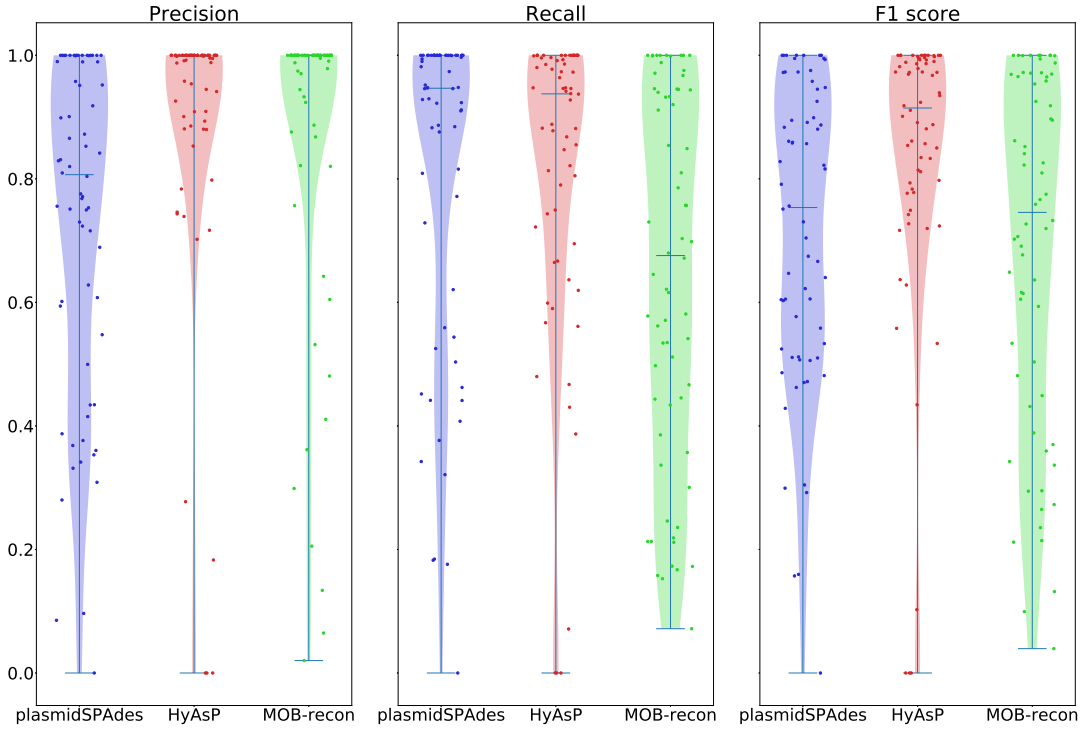
Figure S18 details how much plasmidSPAdes and MOB-recon performed better or worse than HyAsP on the different samples and, as before, the greedy algorithm attained higher F1 scores on a broad range of samples, while MOB-recon (plasmidSPAdes) showed a better precision (recall) on several samples.

Subsequently, we analysed the predictions after grouping the samples based on the associated organism on the species level. Figure S19 shows the results for the 8 different species listed for the test samples. Except for *Enterococcus faecium*, HyAsP outperformed plasmidSPAdes and MOB-recon in terms of F1 score for all species in the analysis. Similarly, the greedy algorithm performed as good as or better than the other two tools in terms of precision and recall for almost all species. plasmidSPAdesachieved a higher recall and F1 score than MOB-recon for most species. MOB-recon, in turn, attained a better precision for all species except *Klebsiella aerogenes*, for which it showed an unually low recall of almost 0.

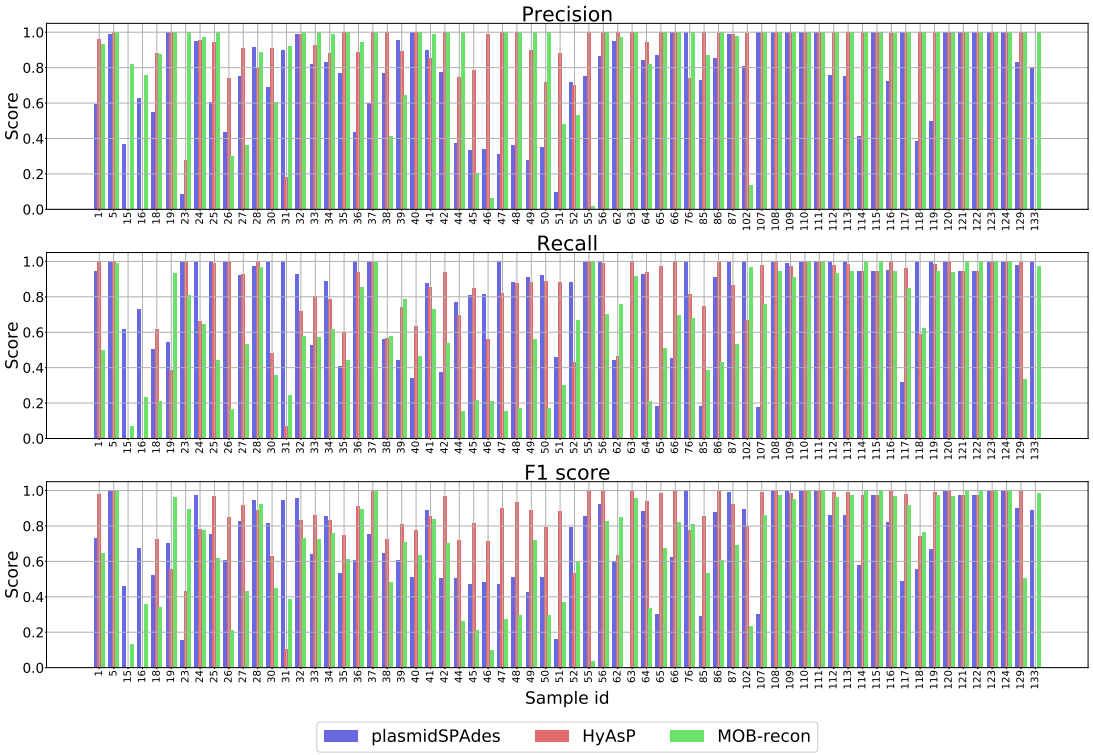
As before, we briefly analysed the occurrence of misassembly events using QUAST (v4.6.3). Again, misassemblies occurred in the majority of samples and relocations were the dominant misassembly type



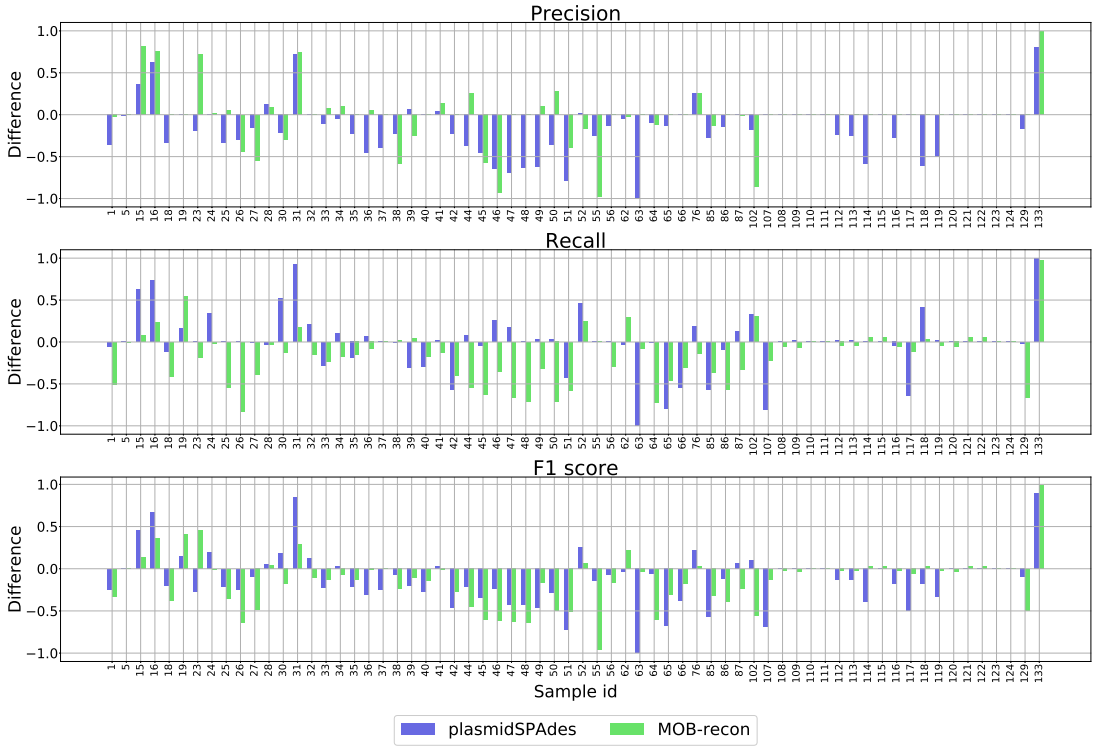
Supplement Figure S15: Misassembly events detected by QUAST in the plasmids predicted by HyAsP using the NCBI-database. The counts of the different types of misassembly events are stated per test sample (*top*) and as histograms (*bottom*). To relate the number of misassembly events to the length of the predictions, the rates of misassembly events per 1000 nt are provided (*middle*).



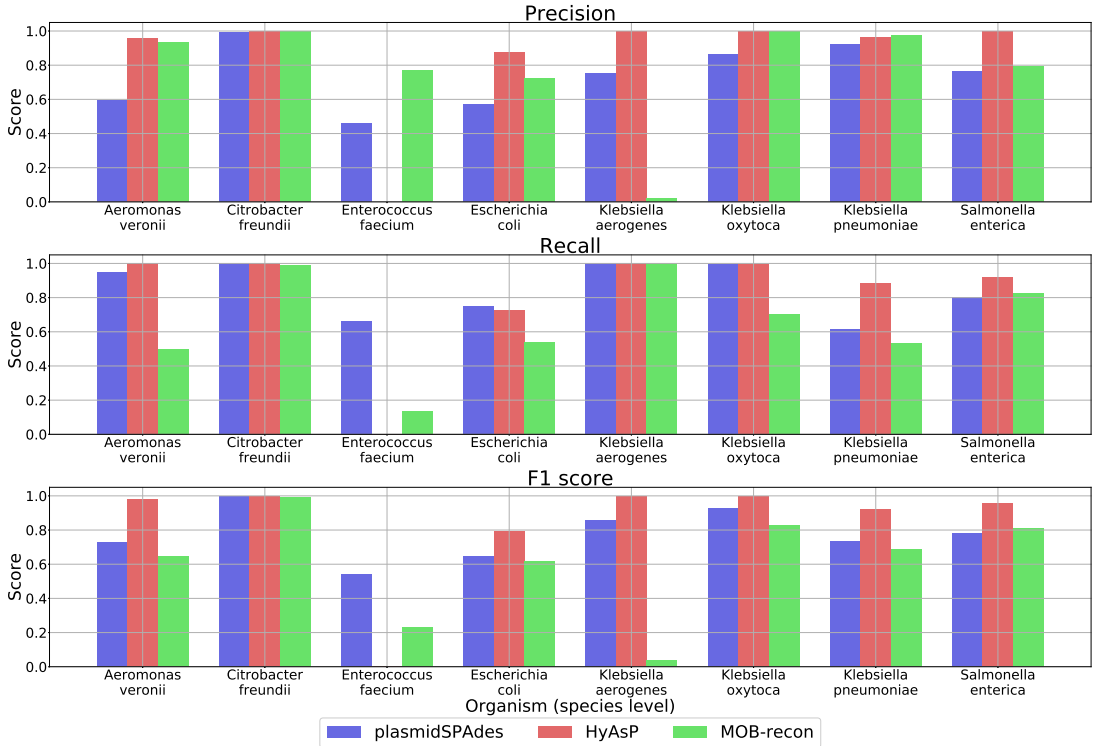
Supplement Figure S16: Distributions of precision, recall and F1 score of HyAsP, plasmidSPAdes and MOB-recon on the test samples using the MOB-database. The horizontal lines in each violin plot represent the minimum, median and maximum value of the respective metric-tool combination.



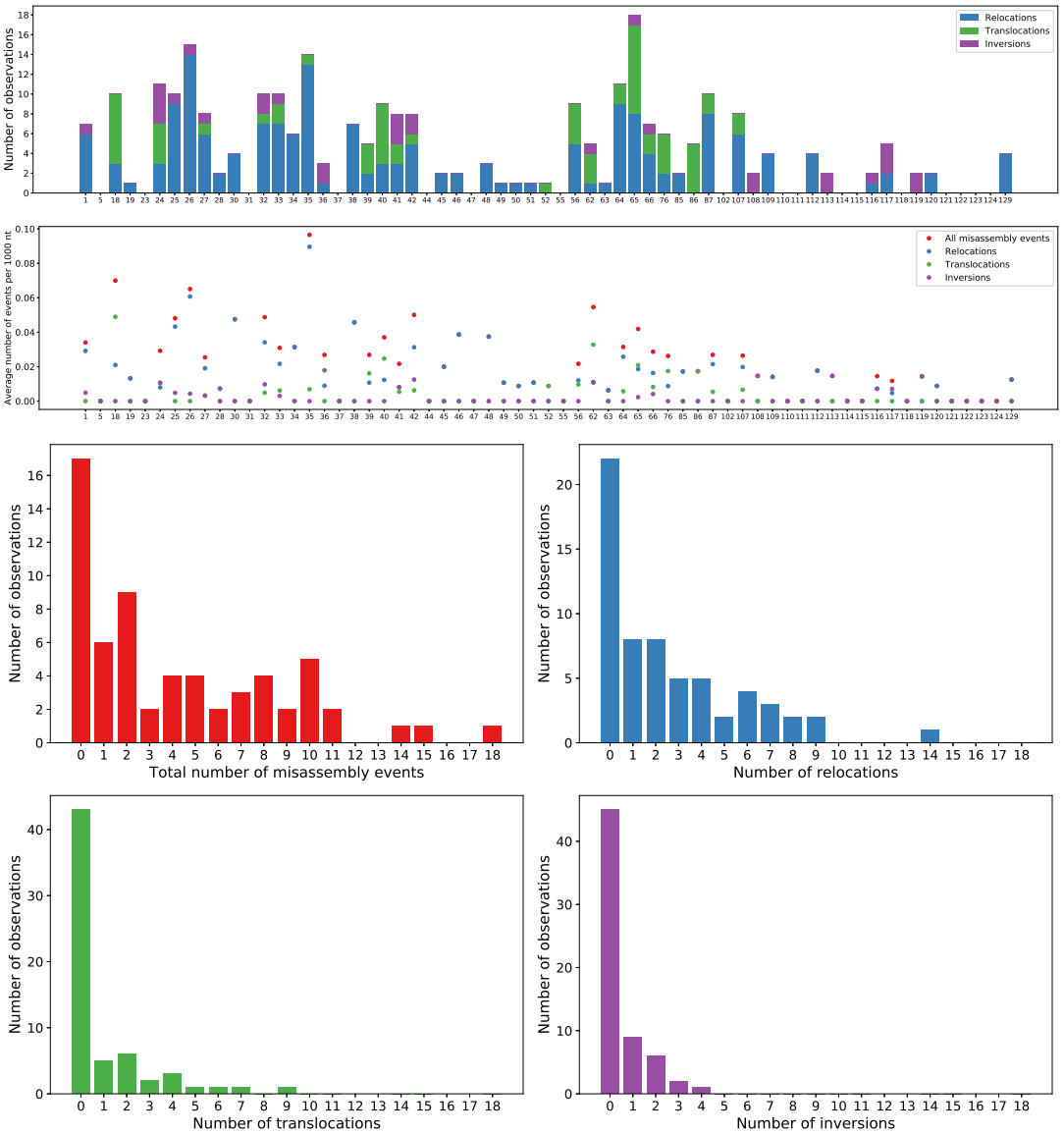
Supplement Figure S17: Precision, recall and F1 score per test sample of HyAsP, plasmidSPAdes and MOB-recon using the MOB-database.



Supplement Figure S18: Difference in precision, recall and F1 score per test sample between HyAsP and plasmidSPAdes and MOB-recon using the MOB-database. The zero line represents the respective score of HyAsP. Positive (negative) values indicate that the other tool performed better (worse) than the greedy algorithm.



Supplement Figure S19: Precision, recall and F1 score per species of HyAsP, plasmidSPAdes and MOB-recon on the test samples using the MOB-database.



Supplement Figure S20: Misassembly events detected by QUAST in the plasmids predicted by HyAsP using the MOB-database. The counts of the different types of misassembly events are stated per test sample (*top*) and as histograms (*bottom*). To relate the number of misassembly events to the length of the predictions, the rates of misassembly events per 1000 nt are provided (*middle*).

(Figure S20).

6 Parameters of HyAsP

A range of parameters allows adjusting the creation of the database (command create), the generating of a gene-contig mapping (commands map and filter) and the greedy algorithm (command find).

6.1 Creating the gene database

Command greefin create <genes file> can be combined with:

--from_accession, -a (default: (empty string), i.e. not used)
: Path to file containing one (plasmid) accession number per line OR list of accession numbers (see --from_command_line).

--from_genbank, -g (default: (empty string), i.e. not used)
: Path to file containing path to a GenBank file per line OR list of paths (see --from_command_line).

--from_plasmid_table, -p (default: (empty string), i.e. not used)
: Path to plasmid table downloaded from NCBI (`https://www.ncbi.nlm.nih.gov/genome/browse#!/plasmids/`) with all possible columns.

--keep_plasmids, -k (default: (empty string), i.e. deactivated)
: Stores the plasmids underlying the gene database in FASTA format if a file is specified.

--dereplicate, -d (default: False)
: Removes duplicate genes from database if activated.

--from_command_line, -c (default: False)
: Instead of a file containing the accession numbers (file paths), the options -a (-g) expect a comma-separated list of accession numbers (file paths). Cannot be combined from -p.

--extend, -e (default: False)
: Genes (and plasmids) are added to an existing database instead of overwriting it.

--released_before, -r (default: (empty string), i.e. deactivated)
: Consider only plasmids released before the specified date. Date format: YYYY-MM-DDTHH:MM:SSZ, e.g. 2005-07-31T00:00:00Z. Can only be combined with -p.

--type, -t (default: both)
: Build the databases from the RefSeq accession numbers (RefSeq), GenBank accession numbers (GenBank) or both (both). Affects only option -p.

--blacklist, -b (default: (empty string), i.e. deactivated)
: Comma-separated list of accession numbers of plasmids not to be included in the databases. Cannot be combined with -e.

--min_length, -l (default: 0)
: Minimum length of plasmids to be considered for the database.

--max_length, -L (default: ∞)
: Maximum length of plasmids to be considered for the database.

--min_gene_length, -m (default: 0)
: Minimum length of genes to be considered for the database.

The database is created from either accession numbers or (already downloaded) GenBank files or the NCBI plasmid table, i.e. the options -a, -g and -p cannot be combined.

6.2 Mapping a collection of genes to the contigs of an assembly

Command greefin map <genes file> <mapping file> can be combined with:

--from_fasta, -f (default: (empty string), i.e. not used)
: Path to file containing the contigs (in FASTA format) to which the genes should be matched.

--from_gfa, -g (default: (empty string), i.e. not used)
: Path to file containing the contigs (as part of assembly graph in GFA format) to which the genes should be matched.

--clean, -c (default: False)
Remove temporary files after the mapping has been created.

The mapping is created from either a FASTA file or a GFA file, i.e. the options -f and -g cannot be combined.

6.3 Filtering a gene-contig mapping

Command greefin filter <genes file> <mapping file> <filtered mapping file> can be combined with:

--identity_threshold, -i (default: 0.95)
Minimum identity of hits retained in the mapping.

--length_threshold, -l (default: 0.95)
Minimum fraction of query (gene) that has be matched to keep a hit.

--find_fragmented, -f (default: False)
Search for fragmented hits, i.e. several short high-identity hits that together satisfy the length threshold.

6.4 Finding plasmids in an assembly

Command greefin find <assembly graph> <genes file> <mapping file> <output directory> can be combined with:

--min_gene_density, -g (default: 0.3)
Minimum gene density of a putative plasmid. Plasmids with a lower gene density are marked as questionable.

--min_seed_gene_density, -k (default: 1.5 × min_gene_density)
Minimum gene density necessary for a contig to be considered as a seed.

--min_length, -l (default: 1500)
Minimum length of a putative plasmid. Shorter plasmids are marked as questionable.

--max_length, -L (default: 1750000)
Maximum length of a putative plasmid. Gene-containing contigs longer than max_length are not used as seeds. A contig is excluded from list of potential extensions, if the combined length of the contig and the plasmid is larger than max_length.

--min_read_depth, -r (default: 0.75 × min_plasmid_read_depth)
Minimum read depth of a contig to be able to participate in a plasmid.

--min_plasmid_read_depth, -d (default: 0.4 × (median read depth of input assembly graph))
Minimum average read depth of a putative plasmid. Plasmids with a lower average read depth are marked as questionable.

--max_gc_diff, -G (default: 0.15)
Maximum difference in GC content between a plasmid and a potentially added contig.

--max_intermediate_contigs, -c (default: 2)
Maximum number of gene-free contigs between two gene-containing contigs in a plasmid. A contig is excluded from the list of potential extensions, if its addition would violate this threshold.

--max_intermediate_nt, -n (default: 2000)
Maximum total length of any consecutive sequence of gene-free contigs in a plasmid. A contig is excluded from the list of potential extensions, if its addition would violate this threshold.

--max_score, -s (default: ∞)
Maximum score of a potential extension. Possible extensions with a higher score are discarded.

--score_weights, -w (default: depth_diff=1,gene_density=1,gc_diff=1)
Weights of the different components of the function used to score extensions. Comma-separated list of entries of the form <name>=<value>. The weight of a component is determined automatically if the question mark (?) is used as <value>.

--keep_subplasmids, -q (default: False)
: Do not mark plasmids whose underlying set of contigs is contained by others as questionable. If several plasmids have the same underlying set of contigs, one of them will remain in the collection of putative plasmids.

--overlap_ends, -o (default: ∞)
: Minimum overlap between the two ends of a plasmid in order to mark it as circular.

--binning, -b (default: NaN, i.e. deactivated)
: Factor determining how many standard deviations the read depth and GC content of plasmids are allowed to differ from the 'centre' of their bin. Binning is activated by setting the parameter to any value different from NaN.

--fanout, -f (default: 1)
: Maximum number of predecessors / successors of any contig in a 'plasmid' (or rather contig collection). Setting this parameter to any value > 1 leads to non-linear / branching contig chains. Changes which files are generated as output. Cannot be used together with probabilistic.

--probabilistic, -p (default: False)
: Flag changing the behaviour of the extension step to a probabilistic choice. The probability of an extension is the share of involved contig of the total read depth of all extensions. Cannot be used together with fanout.

--use_node_based, -N (default: False)
: Flag changing the behaviour of the extension step to a node-based loop avoidance (instead of link-based).

--use_median, -u (default: False)
: Flag activating the use of median (instead of mean) in order to compute the average read depth of a plasmid.

See Section 3 for information on how the default values were derived.

References

Andrews, S. (2010). FastQC: A quality control tool for high throughput sequence data. Software available at https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

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

Bankevich, A. *et al.* (2012). SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of computational biology : a journal of computational molecular cell biology*, **19**(5), 455477.

Camacho, C. *et al.* (2009). BLAST+: architecture and applications. *BMC Bioinformatics*, **10**(1), 421.

Gurevich, A. *et al.* (2013). QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, **29**(8), 1072–1075.

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`.

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

Nishida, H. (2012). Comparative Analyses of Base Compositions, DNA Sizes, and Dinucleotide Frequency Profiles in Archaeal and Bacterial Chromosomes and Plasmids. *International Journal of Evolutionary Biology*, **Volume 2012**, Article ID 342482, 5 pages.

Ondov, B. D. *et al.* (2016). Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biology*, **17**(1), 132.

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

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