metaSPAdes: a new versatile metagenomic assembler (Supplementary Material)

Sergey Nurk[1,*,**], Dmitry Meleshko[1,*], Anton Korobeynikov[1,2] and Pavel A Pevzner[1,3]

[1]Center for Algorithmic Biotechnology, Institute for Translational Biomedicine, St. Petersburg State University, St. Petersburg, Russia

[2]Department of Statistical Modelling, St. Petersburg State University, St. Petersburg, Russia

[3]Department of Computer Science and Engineering, University of California, San Diego, USA

*These authors contributed equally to this work
**corresponding author, sergeynurk@gmail.com

Supplemental Material: Data preprocessing

The SYNTH, HMP, MARINE, and SOIL datasets were pre-processed to remove adaptors and trim low-quality segments of the reads. We used cutadapt software v 1.9.1 (Martin 2011), trimming bases with PHRED quality < 20 from 3' end (parameter `-q 20`). Adaptor sequences were identified for each dataset individually, using FastQC v0.11.3 and manual reads inspection:

SYNTH: `-q 20 -a GAACTCCAGTCACTGACCAATCTCGTATGCCGTCTTCTGCTTG -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT`

HMP: `-q 20 -a AGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGATCTCGTATGCCGTCTTCTGCTTG –a CGGTTCAGCAGGAATGCCGAGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGATCTCGTATGCCGTCTTCTGCTTG -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT -A CGGCATTCCTGCTGAACCGAGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT`

MARINE: `-q 20 -a AGATCGGAAGAGCACACGTCT -A AGATCGGAAGAGCGTCGTGTA`

SOIL: `-q 20 -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCACACTTGAATCTCGTATGCCGTCTTCTGCTTG -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT`

Supplemental Material: Modifying the decision rule in exSPAnder for metagenomic data

The decision rule in exSPAnder uses a binary *support function, Support(e, e', D),* that reflects whether the read-pairs *connecting* edges *e* and *e'* support the conjecture that *e'* follows *e* at the distance *D* in the genome. As described in Prjibelski et al. (2014) and Vasilinetc et al. (2015), exSPAnder automatically adjusts its support function to the characteristics of a particular dataset (e.g. average depth of coverage in the case of isolate sequencing). However, since the support function was not adjusted to the *local* coverage depth, exSPAnder applied the same parameters to regions from both abundant and rare bacterial species, leading to suboptimal and error-prone metagenomic assemblies. metaSPAdes modifies the support function to take into account the read coverage *localCov* of the *specific* genomic region that is being reconstructed during the path extension process.

After *localCov* is computed (see section "Repeat resolution with exSPAnder" for details), metaSPAdes computes the following values based on the empirically estimated distribution of the insert sizes (see Prjibelski et al. (2014) and Vasilinetc et al. (2015) for details):

- $ExpectedReadPairs_{localCov}(e, e', D)$: the expected number of read-pairs connecting edges *e* and *e'* separated in the genome by distance *D*, under the assumption that the coverage is uniform with the average value *localCov*. Given the distribution of insert sizes and *localCov*, the value $ExpectedReadPairs_{localCov}(e, e', D)$ is defined by the lengths of edges *e* and *e'* and distance *D*.
- *ReadPairs(e, e' ,D)*: the total number of read-pairs that support the conjecture that *e'* follows *e* in the genome at distance *D*.
- $Support(e, e', D) = 1$ iff $ReadPairs(e,e',D)/ ExpectedReadPairs_{localCov}(e, e, D) > \alpha$ (the default value α=0.3).

Supplemental Material: Reducing running time and memory footprint of metaSPAdes

To scale metaSPAdes for large metagenomic datasets, we implemented various speed-ups and memory saving approaches, including:

- Parallelization of the graph simplification procedures for transforming the de Bruijn graph into the assembly graph, e.g. processing of bulges (Bankevich et al. 2012) and complex bulges (Nurk et al. 2013).
- Filtering of rare *k*-mers based on a *counting Bloom filter* (Fan et al. 2012) to optimize the BayesHammer error-correction module (Nikolenko et al. 2013) of SPAdes.
- Memory-efficient storage of cumulative information on paired-read alignments against the edges of the assembly graph.

Below we sketch a yet another novel approach for compact representation and efficient construction of the de Bruijn graphs implemented in the SPAdes toolkit, while algorithms for parallelization of the graph simplification procedures, filtering of rare *k*-mers, and memory-efficient storage of cumulative information on read-pairs will be described in Korobeynikov et al. (2017). Our goal was to develop a more memory-efficient approach for the assembly graph construction (as compared to the naive approach originally implemented in SPAdes (Bankevich et al. 2012)) that would efficiently support modifying operations necessary for assembly graph simplification, e.g., processing of tips and bulges.

First, sets of all distinct *k*-mers and (*k*+1)-mers in reads are efficiently computed with a "sort-and-compact" approach (Roy et al. 2014) and stored in an external memory. Only *canonical* (smallest in the reverse-complement pair) *k*-mers and (*k*+1)-mers are considered. Afterwards, a state-of-the-art perfect hash function (PHF) is constructed over the set V of all *k*-mers in reads (Botelho et al. 2013). Afterwards, a binary $|V|$x8 matrix (called *extensions matrix*) is constructed using the set of (*k*+1)-mers. Every row in the matrix corresponds to a *k*-mer with a specified hash value and every column corresponds to a potential single nucleotide extension of this *k*-mer either to the left or to the right. The PHF, the set of distinct *k*-mers (which is never loaded in RAM after the PHF is constructed), and the extension matrix serve as a representation of the de Bruijn graph. Since the constructed PHF data structure takes only 2.7 bits per *k*-

mer (Botelho et al. 2013), the entire representation takes only (8+2.7) bits per *k*-mer in RAM. The graph is further efficiently converted into its "condensed" form, which stores non-branching paths as sequences of nucleotides of arbitrary length.

With the perfect hashing of *k*-mers and extension matrix at its core, our approach resembles the one proposed in Iqbal et al. (2012), but significantly reduces its memory footprint by avoiding the need for storing all distinct *k*-mers in RAM. The important advantage of the described approach (compared to the original SPAdes implementation or to the approach based on the Bloom filters) is that it enables efficient deletion of edges in the de Bruijn graph during the graph simplification step, one of the most time consuming steps in the SPAdes pipeline.

The detailed comparison of algorithms for reducing running time and memory footprint in SPAdes with other recently proposed approaches for efficient de Bruijn graph construction based on:

- *Bloom filters* (Bloom 1970; Pell et al. 2012; Chikhi and Rizk 2013; Salikhov et al. 2014) implemented in the Minia assembler (Chikhi and Rizk 2013),
- *succinct de Bruijn graphs* (Bowe et al. 2012) implemented in the MEGAHIT assembler (Li et al. 2016),
- *perfect hashing* implemented in the Meraculous assembler (Chapman et al. 2011)

will be presented in Korobeynikov et al. (2017).

Supplemental Material: Bulge projection approach

A *bulge* is defined as two short alternative directed paths between the same vertices of the de Bruijn graph. Multiple bulges often aggregate into more complex subgraphs that we refer to as *complex bulges* (see Bankevich et al. (2012) and Nurk et al. (2013) for details). SPAdes detects subgraphs of the assembly graph containing short alternative paths and searches each subgraph for a certain subtree, subject to constrains specified in Nurk et al. 2013. Afterwards, all edges and vertices of a subgraph that do not belong to the identified subtree are removed. However, in contrast to other assemblers that discard information about the alternative paths deleted at the bulge removal step, SPAdes transfers auxiliary information associated with them (e.g., the coverage depth of the deleted paths) onto the retained tree and maintains a *projection index*: a mapping of discarded *k*-mers onto their remaining counterparts (Bankevich et al. 2012; Nurk et al. 2013). The projection index facilitates accurate reconstruction of strain-paths (paths in the consensus assembly graph corresponding to the strain-contigs) in metaSPAdes.

Supplemental Material: Nx statistics

Nx is the length for which the collection of all scaffolds of that length or longer covers at least x percent of the total contig length in an assembly. For example, Nx for x=50 corresponds to the standard N50 statistics. Figure S1 presents the Nx statistics for SYNTH, HMP, MARINE, and SOIL datasets, while Table S4 contains N50 values for all assemblies.

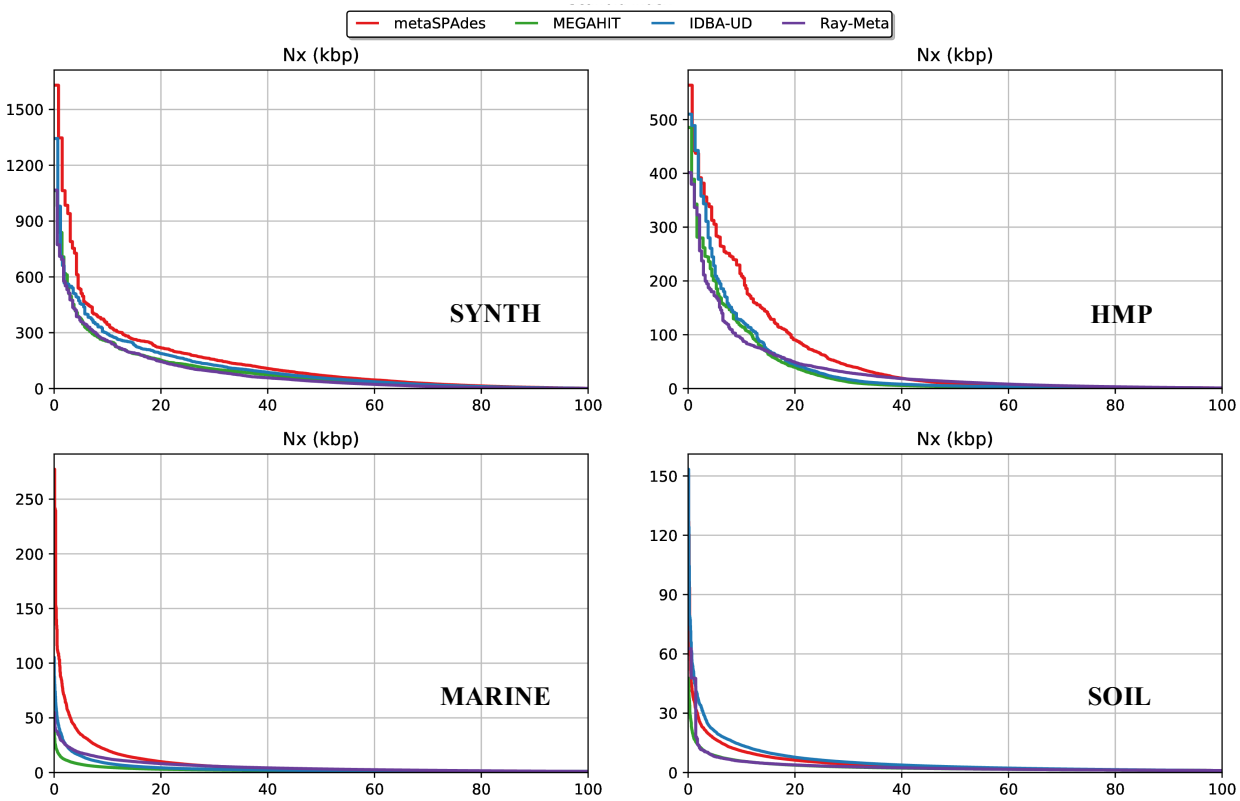

Figure S1. Nx statistics. Only scaffolds longer than 1 kb were considered for computing the Nx statistics.

dataset/assembler	**metaSPAdes**	**MEGAHIT**	**IDBA-UD**	**Ray-Meta**
SYNTH	70.9	50.4	56.7	37
HMP	9.3	3.6	4.9	12.6
MARINE	2.6	1.6	1.9	3.3
SOIL	2.5	1.9	2.9	1.9.

Table S4. N50 statistics (in kb). Only scaffolds longer than 1 kb were considered for computing the N50 statistics.

Supplemental Material: Analysis of the SYNTH dataset

Supplemental Table S2 presents the list of 64 reference genomes for the SYNTH dataset in the decreasing order of their coverage depths. Since metaQUAST was primarily designed to work with complete rather than draft reference genomes, we excluded four references (marked by red in the table) that were represented by multiple contigs. Removal of these references affected the list of "top 20" references (marked in green) that we used to generate Figure 2 in the main text. To avoid pseudo-misassemblies (correctly assembled regions mapped to an incorrect reference and thus classified as misassemblies) and other artifacts, we also excluded *Sulfurihydrogenibium sp. YO3AOP1* (SYO3AOP1) reference (marked in blue) that was closely related (95% average identity) to the more abundant *Sulfurihydrogenibium yellowstonense SS-5* (SyeSS-5) genome previously excluded from consideration because it is represented by a fragmented draft assembly. We further computed the fraction of bases covered by the sequencing reads for each reference and identified three references with relatively high depth of coverage (marked by yellow in Table S3) for which more than 5% of bases were not covered by reads. We decided to exclude these references from further analysis since they are likely to significantly differ from their counterparts in the sample (note that all these references have coverage exceeding 14X).

Supplemental Table S3 shows the per-reference metaQUAST report for the remaining (after removing red, blue and yellow entries in Supplemental Table S2) 56 reference genomes in the SYNTH dataset. metaSPAdes resulted in a significant improvement in NGA50 compared to other assemblers for many reference genomes. Specifically, it resulted in at least 20% improvement over MEGAHIT, IDBA-UD, and Ray-Meta for 33, 28, and 39 genomes, respectively. At the same time, the best assembly among MEGAHIT, IDBA-UD, and Ray-Meta improved over metaSPAdes by more than 20% in only 4 cases. With respect to the number of intragenomic misassemblies, metaSPAdes is on par with MEGAHIT, while IDBA-UD assemblies deteriorate when coverage drops below 20X (starting with row 53 in Table S2) and Ray-Meta falls out of competition. Across all considered references, metaSPAdes, MEGAHIT, IDBA-UD, and Ray-Meta resulted 21, 38, 237 and 27 intergenomic misassemblies, respectively. Note that

IDBA-UD resulted in an order of magnitude increase in the number of intergenomic misassemblies as compared to other assemblers.

The SYNTH dataset includes two highly similar (96 % Average Nucleotide Identity) bacteria from *Thermotoga* genus: a more abundant *Thermotoga sp. RQ2* (ThRQ2) with coverage 128X and less abundant *Thermotoga petrophila RKU-1* (TpeRKU-1) with coverage 48X.

The 4th row in Supplemental Table S3 illustrates that metaSPAdes significantly improves on other assemblers with respect to the contiguity of the abundant ThRQ2 genome reconstruction. Manual investigation using Icarus (Mikheenko et al. 2016) confirmed that MEGAHIT, IDBA-UD and Ray-Meta constructed over-fragmented assemblies with many contigs mapping to both related reference genomes. Consistent with the "consensus-assembly" focus of metaSPAdes, this improvement comes at the cost of the "genome fraction" statistics for rare TpeRKU-1 genome (row 29), since resulting long contigs unambiguously map to the abundant ThRQ2 genome.

Supplemental Material: CAMI datasets

"Critical Assessment of Metagenome Interpretation" (CAMI) is a community initiative aimed at evaluating various approaches for analyzing metagenomes (http://www.cami-challenge.org/). Within this initiative, multiple synthetic datasets were simulated from reference genomes (including groups of closely related genomes) to facilitate benchmarking of metagenomic pipelines (available at https://data.cami-challenge.org/participate).

Supplemental Material "Analysis of CAMI datasets" presents benchmarking on a "medium complexity" dataset simulated from 225 genomes and containing 150 million paired-end reads ("Toy Test Dataset Medium Complexity Sample 1" referred to as $CAMI_{med}$) and on a "low complexity" dataset simulated from 30 genomes and containing 74 million reads ("Toy Test Dataset Low Complexity" referred to as $CAMI_{low}$). Paired-end reads had length of 100 bp and mean insert size of 180 bp (the errors were modelled after Illumina HiSeq reads). Unfortunately, it turned out that many reference genomes used for generating CAMI datasets were highly fragmented draft assemblies (rather than finished bacterial genomes). Such datasets are not ideal for assembly benchmarking studies since reads simulated from fragmented assemblies result in atypical assembly graphs with many missing edges (not to mention that metaQUAST was primarily designed to work with complete rather than draft reference genomes).

To bypass these problems, we excluded highly fragmented references (represented by draft assemblies with N50 below 200kbp) from benchmarking and used "`--fragmented`" option in metaQUAST. Table S5 presents information on all genomes comprising the $CAMI_{low}$ dataset along with the list of over-fragmented references. Table S6 presents information on 37 genomes comprising $CAMI_{med}$ dataset with coverage depth exceeding 20X (19 of them are over-fragmented).

No.	Taxonomic ID	Species	Genome size (Mbp)	Average coverage
1	434085	*Gamma proteobacterium IMCC2047*	2.2	873
2	247639	*Marine gamma proteobacterium HTCC2080*	3.6	53
3	1050222	*Paenibacillus sp. Aloe-11*	5.8	22

4	667138	*Thermoplasmatales archaeon I-plasma*	1.7	21
5	552396	*Erysipelotrichaceae bacterium 5_2_54FAA*	6.3	16
6	1007115	*Gamma proteobacterium SCGC AAA076-D13*	1.7	14
7	1122939	*Patulibacter americanus DSM 16676*	4.5	9
8	1111069	*Thermus sp. CCB_US3_UF1*	2.3	8
9	1131272	*Chloroflexi bacterium SCGC AB-629-P13*	0.8	8
10	1131273	*Marinimicrobia bacterium SCGC AB-629-J13*	1.9	8
11	1097667	*Patulibacter medicamentivorans*	5.1	7
12	1263001	*Firmicutes bacterium CAG:114*	2.3	4
13	1137281	*Formosa sp. AK20*	3.1	3
14	1345697	*Geobacillus sp. JF8*	3.5	2
15	1412874	*Uncultured archaeon A07HR60*	2.9	1.9
16	1224136	*Enterobacteriaceae bacterium LSJC7*	4.6	1.8
17	1229484	*Alpha proteobacterium LLX12A*	6.0	1.4
18	1229781	*Brevibacterium casei S18*	3.7	1.2
19	1235799	*Lachnospiraceae bacterium 3-2*	4.5	1.0
20	370895	*Burkholderia mallei 2002721280*	5.7	0.9
21	742723	*Lachnospiraceae bacterium 2_1_46FAA*	4.4	0.9
22	1045854	*Weissella koreensis KACC 15510*	1.4	0.7
23	1009708	*Alpha proteobacterium SCGC AAA536-G10*	2.2	0.6
24	1174684	*Sphingopyxis sp. MC1*	3.7	0.4
25	349101	*Rhodobacter sphaeroides ATCC 17029*	4.5	0.4
26	1230476	*Bradyrhizobium sp. DFCI-1*	7.7	0.3
27	245012	*Butyrate-producing bacterium SM4/1*	3.1	0.3
28	939301	*Alpha proteobacterium SCGC AAA015-O19*	1.7	0.2
29	1263006	*Firmicutes bacterium CAG:170*	2.5	0.2
30	1394711	*Candidatus Saccharibacteria bacterium RAAC3_TM7_1*	0.9	0.1

Table S5. The list of 30 reference genomes comprising the $CAMI_{low}$ dataset arranged in the decreasing order of their coverage depths. References that are not over-fragmented are marked in green.

No.	Taxonomic ID	Species	Genome size (Mbp)	Average coverage
1	1247738	*Campylobacter coli BIGS0015*	1.3	257
2	1097667	*Patulibacter medicamentivorans*	4.8	200
3	1399144	*Brevibacillus laterosporus PE36*	5.1	199
4	494419	*Arthrobacter sp. TB 23*	3.5	166
5	314254	*Oceanicaulis sp. HTCC2633*	3.2	140
6	290399	*Arthrobacter sp. FB24*	5.1	137
7	883112	*Facklamia ignava CCUG 37419*	1.8	133
8	434085	*gamma proteobacterium IMCC2047*	0.5	133
9	1131272	*Chloroflexi bacterium SCGC AB-629-P13*	0.8	108
10	1224136	*Enterobacteriaceae bacterium LSJC7*	4.6	96
11	1123317	*Streptococcus sobrinus DSM 20742 = ATCC 33478*	1.7	89
12	457393	*Bacteroides sp. 4_1_36*	4.6	88
13	1353530	*Bacteriovorax sp. DB6_IX*	2.5	82
14	1159204	*Mycoplasma gallisepticum NC08_2008.031-4-3P*	0.9	79
15	1209372	*Bacillus sp. WBUNB009*	5.6	77
16	1263006	*Firmicutes bacterium CAG:170*	2.3	77
17	1386080	*Bacillus sp. EGD-AK10*	4.3	76
18	1386078	*Pseudomonas sp. EGD-AK9*	3.9	70
19	322710	*Azotobacter vinelandii DJ*	5.4	68
20	766138	*Shigella boydii 965-58*	5.2	59
21	1283301	*Streptomyces afghaniensis 772*	9.1	53
22	1262788	*Clostridium sp. CAG:269*	1.5	48
23	1194208	*Streptococcus massiliensis 4401825*	1.7	45
24	392917	*Paenibacillus larvae subsp. larvae BRL-230010*	3.8	37
25	997884	*Bacteroides nordii CL02T12C05*	5.7	36
26	234826	*Anaplasma marginale str. St. Maries*	1.2	36
27	1203566	*Corynebacterium sp. KPL1859*	2.6	36
28	1198114	*Granulicella tundricola MP5ACTX9*	5.5	32
29	247639	*marine gamma proteobacterium HTCC2080*	3.6	30
30	98439	*Fischerella thermalis PCC 7521*	5.4	30
31	1042156	*Clostridium sp. SY8519*	2.8	28

32	1074065	*Streptococcus sobrinus TCI-98*	1.8	27
33	1218358	*Chlamydia psittaci WC*	1.2	26
34	522306	*Candidatus Accumulibacter phosphatis clade IIA str. UW-1*	5.3	23
35	95609	*Herbaspirillum sp. B39*	3.6	22
36	97137	*Lactobacillus sp. ASF360*	2.0	21
37	1130827	*Rickettsia sibirica subsp. sibirica BJ-90*	1.3	20

Table S6. The list of most abundant reference genomes (with coverage exceeding 20X) in the $CAMI_{med}$ dataset arranged in the in decreasing order of their coverage depths. References that are not over-fragmented are marked in green.

Supplemental Material: Analysis of the CAMI datasets

Figure S2 shows cumulative scaffold length and Nx statistics for the CAMI datasets. Figure S3 and S4 present detailed per-reference benchmarking results for the $CAMI_{low}$ and $CAMI_{med}$ datasets, respectively. These figures are based on non-over-fragmented reference genomes listed in Tables S5 and S6 (see Supplemental Material "CAMI datasets" for details). Note that since only a fraction of references have been provided to metaQUAST, the number of intergenomic misassemblies is likely to be underestimated.

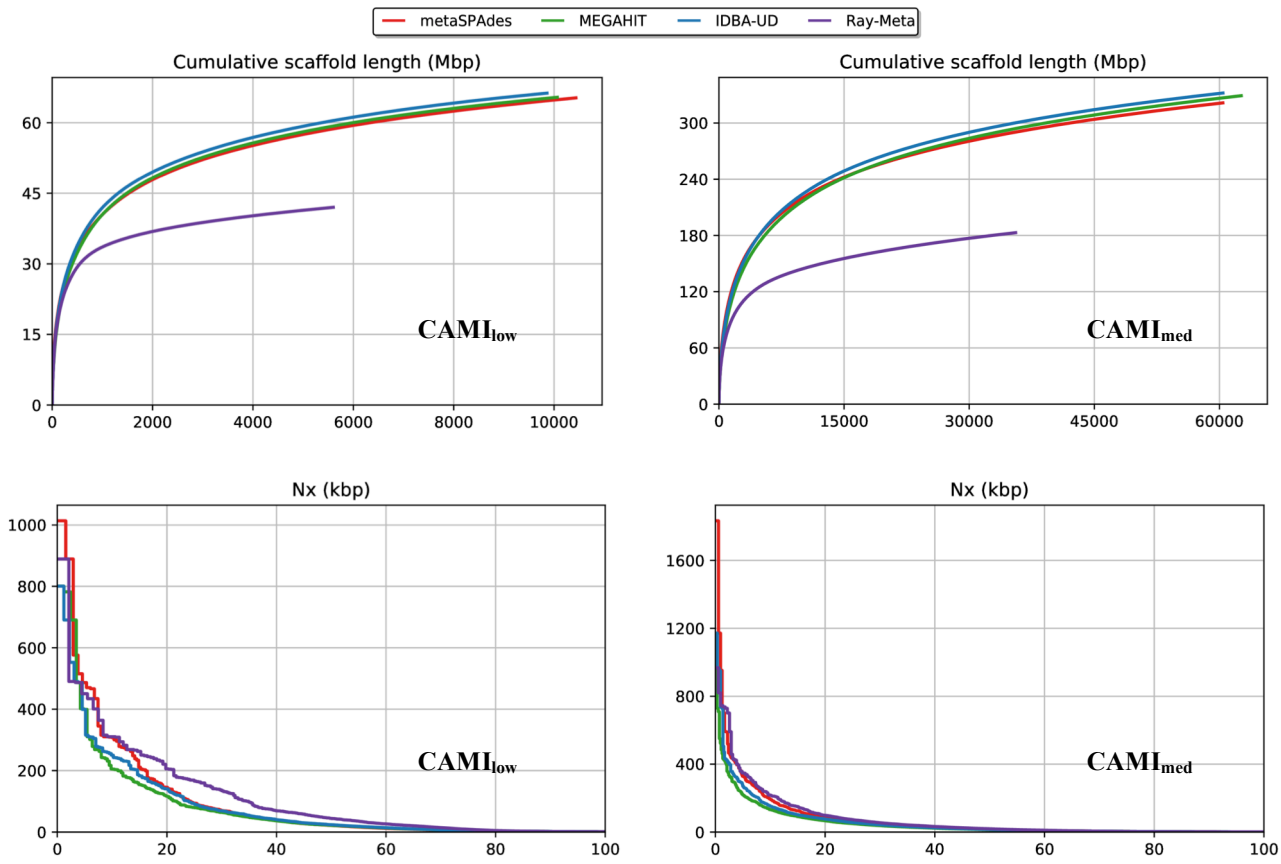

Figure S2. Cumulative scaffold length and Nx plots for $CAMI_{low}$ and $CAMI_{med}$ datasets.

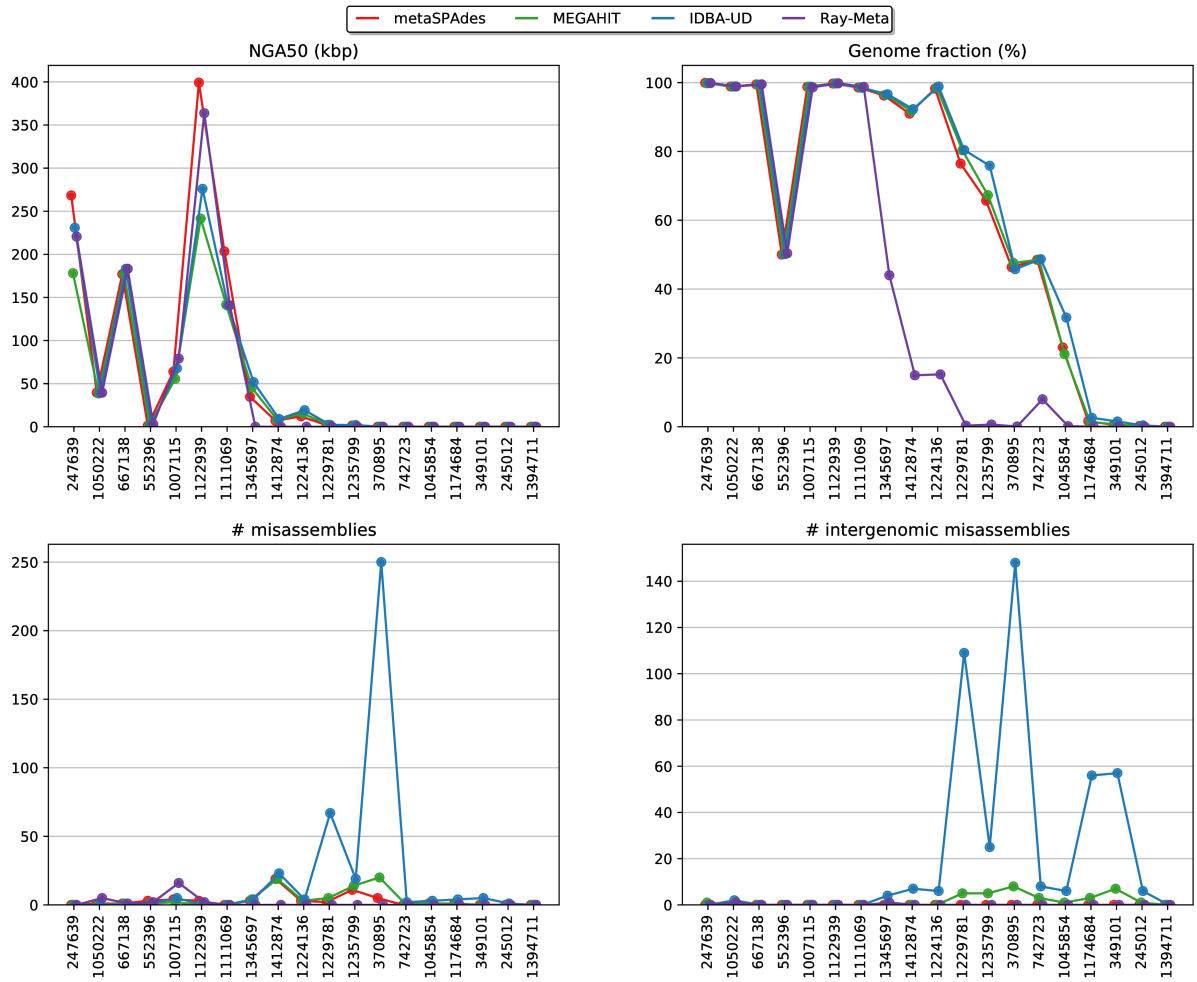

Figure S3. The NGA50 statistics (top left), the fraction of the reconstructed genome (top right), the number of intragenomic misassemblies (bottom left) and the number of intergenomic misassemblies (bottom right) for non over-fragmented references of CAMI$_{low}$ dataset. References are specified by their Taxonomic IDs (see Table S5) and arranged in the decreasing order of their coverage depths.

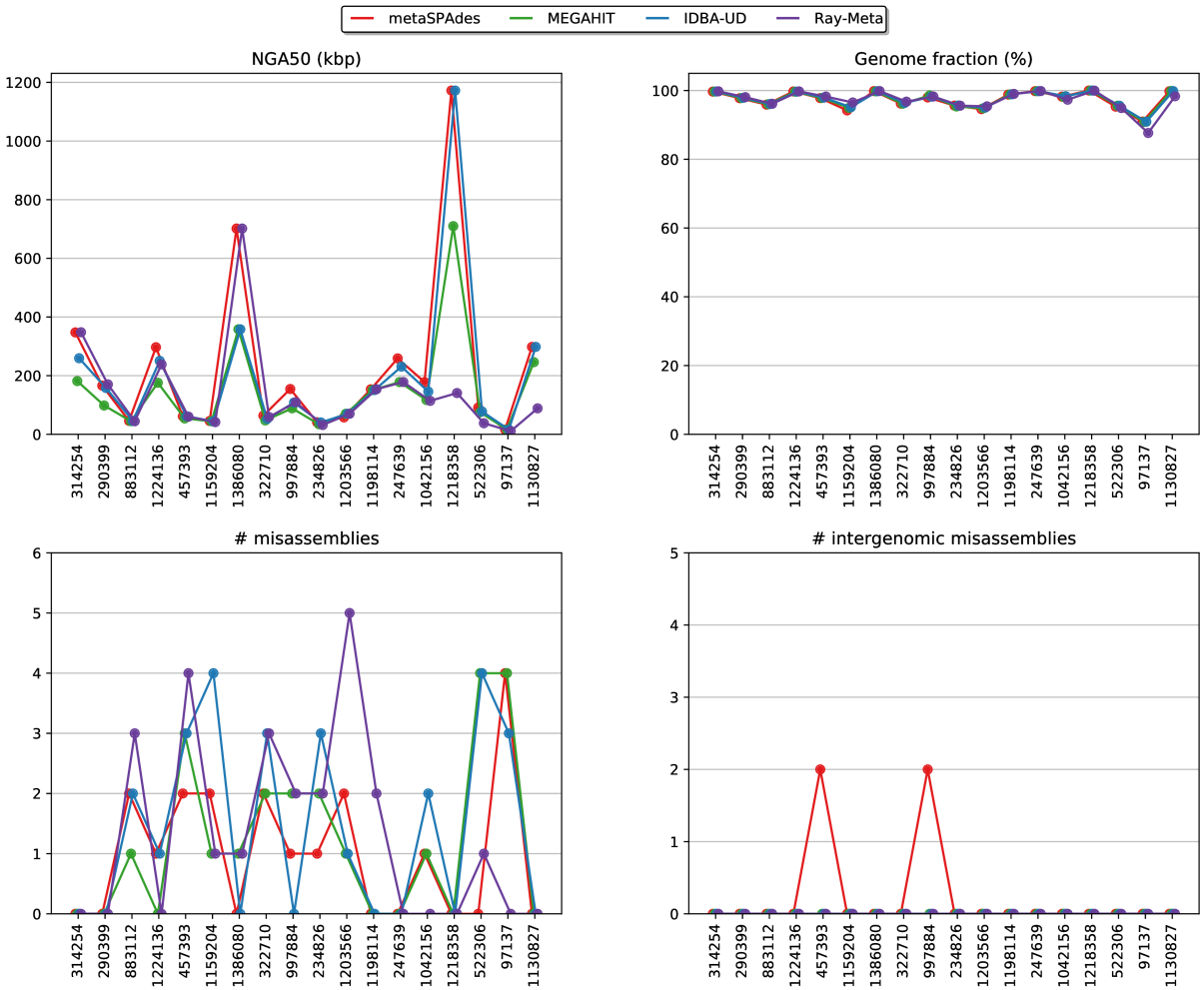

Figure S4. The NGA50 statistics (top left), the fraction of the reconstructed genome (top right) the number of intragenomic misassemblies (bottom left) and the number of intergenomic misassemblies (bottom right) for 18 most abundant non over-fragmented references species from the CAMI$_{med}$ dataset. References are specified by their Taxonomic IDs (see Table S6) and arranged in the decreasing order of their coverage depths. The genomes are arranged in the decreasing order of their coverage depths.

Supplemental Material: Benchmarking SPAdes against metaSPAdes

Various changes in metaSPAdes contribute to improved metagenomic assemblies as compared to SPAdes. In particular, to clear the assembly graph from erroneous connections, SPAdes removes edges with coverage depth below an automatically selected *coverage threshold* (Bankevich et al. 2012). Computing the threshold based on the entire assembly graph of a metagenomic dataset proved to be highly unstable and often forces SPAdes to remove many genomic edges, thus increasing the number of misassemblies in the contigs originating from rare species.

In contrast to SPAdes, metaSPAdes uses a fixed coverage threshold set to a low value of 2.5 and primarily relies on analyzing *relative* local coverage of incident edges, thus accurately processing low-covered regions of the assembly graph (note that the coverage of an edge in the SPAdes condensed assembly graph is measured as an average coverage of (k+1)-mers contributing to this edge). In an attempt to provide a fair benchmarking, we additionally tested unusual SPAdes FIX and SPAdes SC FIX configurations of SPAdes with a coverage threshold set to 2.5 (see below).

We use the SYNTH dataset to illustrate metaSPAdes improvement over SPAdes in metagenomics setting. SPAdes was launched in four different modes to ensure comprehensive benchmarking:

- SPAdes in the "isolate" mode (referred simply as SPAdes),
- SPAdes in the "single-cell" (--sc) mode (referred as SPAdes SC),
- SPAdes with the fixed value of the coverage threshold set at 2.5X as in metaSPAdes (referred as SPAdes FIX),
- SPAdes SC with the fixed value of the coverage threshold set at 2.5X as in metaSPAdes (referred as SPAdes SC FIX).

Table S7 presents the benchmarking results and illustrates that metaSPAdes generates substantially more accurate assemblies than all variants of SPAdes, including variants with the fixed low coverage thresholds: metaSPAdes, SPAdes, SPAdes FIX, SPAdes SC and SPAdes SC FIX resulted in102, 353, 163, 159 and 142 intragenomic misassemblies and 21, 239, 119, 75 and 51 intergenomic

misassemblies, respectively. We note that the SYNTH dataset contains very few references with low depth of coverage. Our analysis suggests that taking a subset of the reads for the SYNTH dataset makes the observed difference even more pronounced. With respect to NGA50, metaSPAdes either improves or remains on par with all SPAdes configurations (only for eight genomes, one of the SPAdes configurations improved on metaSPAdes by more than 15%).

No.	Abbreviation	NGA50					Intragenomic misassemblies				
		meta SPAdes	SPAdes	SPAdes FIX	SPAdes SC	SPAdes SC FIX	meta SPAdes	SPAdes	SPAdes FIX	SPAdes SC	SPAdes SC FIX
1	Neq	381775	320953	141036	397028	397028	0	1	1	1	1
2	Pho	293747	230474	187452	230525	159269	2	1	1	1	1
3	Rba	183456	213265	209811	209811	201446	5	3	3	3	3
4	ThRQ2	45835	5372	5180	7661	7634	0	1	1	0	0
5	Afu	183804	88770	88770	88770	88770	2	3	3	3	3
6	Neu	46392	46505	45758	46286	46286	2	3	3	3	3
7	ThDSM 4359	57139	47814	47814	55811	55811	0	0	0	0	0
8	Sto	76823	63432	58674	63417	63417	0	0	1	0	0
9	HY04AAS1	258515	393877	393877	192792	192792	0	0	0	0	0
10	Gau	1347186	729009	508460	728869	728869	0	0	0	0	0
11	PaeIM2	91533	81696	76953	81696	81696	1	0	0	0	0
12	Pfu	59599	55107	54575	57757	54785	0	3	3	0	0
13	Cvi	177554	154672	154672	154672	154672	1	2	2	1	1
14	Aca	301792	222496	222496	222496	222496	0	0	0	0	0
15	Pca	276318	221867	221867	221867	221867	0	0	0	0	0
16	Abo	86325	186114	27374	185944	107733	0	0	0	0	0
17	GsuPCA	195995	195995	191745	195995	195995	4	4	4	4	4
18	PmaEX-H1	1063451	1063860	1063860	1063622	1063622	0	0	0	0	0
21	Mja	121804	109197	93232	109195	109195	1	5	2	1	1
22	Tde	168453	121355	121355	120263	120263	0	0	0	0	1
23	Mka	984916	331221	331221	267844	267844	0	0	0	0	0
24	Pas	154812	120505	120505	112944	112944	0	1	0	0	0
26	Cte	169383	169718	169718	141095	141095	0	1	1	1	1
27	MmaC5	169568	22956	22956	11853	11853	0	0	0	0	0
28	Dtu	939665	226110	212115	452770	452770	0	0	0	0	0
29	TpeRKU-1	0	4833	4677	3831	3831	0	2	2	2	2
30	TthHB8	60940	58802	56702	60940	60940	1	0	0	0	0
31	Cli	102105	101378	100963	69167	69167	2	3	2	3	3
33	Csa	39297	25682	25682	19887	19844	5	8	2	5	4
34	Wsu	156243	156698	156698	156698	156698	0	0	0	0	0
35	MmaS2	109948	24359	24347	10096	9821	1	0	1	0	0
36	Cph	40887	37259	37122	35262	32264	5	3	3	4	4
37	Pph	89343	68294	68413	60304	66096	1	2	2	5	5
38	CauJ-10-fl	84420	68621	85381	54705	49548	7	14	6	11	9

39	Amu	189693	165681	150334	165198	165198	2	3	2	3	3
40	Cth	74152	63141	63141	56929	56929	3	3	3	7	4
41	Pgi	30817	31092	30693	23712	23770	3	7	4	3	3
43	Cbe	40725	30509	30509	16594	16594	0	1	2	3	3
44	Tps	54527	54499	70875	39066	37582	0	1	0	3	2
45	Lch	15689	14758	15424	15178	15417	1	11	3	0	1
46	NPCC7120	137846	114085	141094	46894	46894	3	7	6	6	4
48	Iho	212262	158711	158711	78403	78403	0	2	1	0	0
49	Bth	132888	88148	125434	64200	62811	7	10	3	8	8
50	Hvo	26447	26367	26549	23825	24089	0	6	3	3	3
51	Hau	112847	139642	163766	65076	67838	3	11	4	6	7
52	Sar	10460	7604	9046	4620	4614	2	17	4	4	3
53	Str	9743	6695	7816	3925	3861	1	10	7	7	8
54	Bvu	88327	81478	87246	48938	48938	1	5	2	2	1
55	DraR1	15750	16608	17132	13499	13499	0	4	2	0	0
56	MacC2A	36848	31194	33235	20788	19289	9	26	3	7	5
58	Bbr	5650	5555	5826	5475	5485	3	43	27	19	16
60	Rpo	12719	13684	14150	13768	14150	0	11	3	1	0
61	Zmo	43038	52284	52284	49999	50576	1	5	2	2	2
62	BxeLB400	4821	4924	5016	4918	4939	12	52	27	20	18
63	SbaOS185	6015	30454	6924	6450	6348	5	22	6	4	4
64	SbaOS223	0	4934	1037	1091	1104	6	36	6	3	1

Table S7. Benchmarking of metaSPAdes against SPAdes on a SYNTH dataset. NGA50 statistics and the number of intragenomic misassemblies for 56 reference genomes shown in Table S3. See "Supplemental Material: Analysis of the SYNTH dataset" for motivation on excluding 8 out of 64 reference genomes from the SYNTH dataset. The reference genomes are arranged in the decreasing order of their coverage depths. The colors of the cells reflect how much the results of various assemblers differ from the median value (blue/red cells indicate that the results improve/deteriorate as compared to the median value).

Supplemental Material "Effect of novel algorithmic approaches in metaSPAdes on assembly quality"

illustrates how other algorithmic changes in metaSPAdes contributed to improved assembly quality.

Supplemental Material: Effect of novel algorithmic approaches in metaSPAdes on assembly quality

To investigate how various algorithmic changes contribute to the improved assemblies, we considered the following features distinguishing metaSPAdes from SPAdes SC FIX that performed the best among four modes of SPAdes (see benchmarking results in Supplemental Material: "Benchmarking SPAdes against metaSPAdes"):

a. novel assembly graph simplification procedures and exSPAnder modifications (see sections "Detecting and masking strain variations" and "Analyzing filigree edges in the assembly graph" in the main text as well as the Supplemental Material "Modifying the decision rule in exSPAnder for metagenomic data");

b. novel algorithm that exploits differences between strains to improve the consensus assembly (see section "Utilizing strain differences for repeat resolution in metaSPAdes" in the main text);

c. novel coverage-based decision rule in exSPAnder (see section "A new metagenomic decision rule in metaSPAdes" in the main text).

Table S8 presents the SYNTH dataset assembly statistics for the following configurations of metaSPAdes and SPAdes:

- metaSPAdes,
- metaSPAdes with option (c) disabled (referred to as metaSPAdes –c),
- metaSPAdes with options (c) and (b) disabled (referred to as metaSPAdes –c –b),
- SPAdes SC FIX.

Note that configuration metaSPAdes –c –b corresponds to enabling the option (a) in SPAdes SC FIX.

No.	Abbreviation	NGA50				Intragenomic misassemblies			
		metaSPAdes	metaSPAdes -c	metaSPAdes -c-b	SPAdes SC FIX	metaSPAdes	metaSPAdes -c	metaSPAdes -c-b	SPAdes SC FIX
1	Neq	381775	381775	381775	397028	0	0	0	1
2	Pho	293747	293747	144236	159269	2	2	1	1
3	Rba	183456	197615	197622	201446	5	5	5	3
4	ThRQ2	45835	26177	20523	7634	0	0	0	0
5	Afu	183804	183804	87452	88770	2	2	3	3
6	Neu	46392	46392	46392	46286	2	2	2	3

7	ThDSM4359	57139	57139	41488	55811	0	0	0	0
8	Sto	76823	76823	58525	63417	0	0	0	0
9	HY04AAS1	258515	258515	192792	192792	0	0	0	0
10	Gau	1347186	1347186	728458	728869	0	0	0	0
11	PaeIM2	91533	91533	79833	81696	1	1	1	0
12	Pfu	59599	59599	54785	54785	0	0	1	0
13	Cvi	177554	177554	153627	154672	1	1	1	1
14	Aca	301792	301792	222725	222496	0	0	0	0
15	Pca	276318	276318	221867	221867	0	0	0	0
16	Abo	86325	86325	67173	107733	0	0	2	0
17	GsuPCA	195995	195995	195995	195995	4	4	4	4
18	PmaEX-H1	1063451	1063451	1063230	1063622	0	0	0	0
21	Mja	121804	121804	109197	109195	1	1	1	1
22	Tde	168453	168453	95496	120263	0	0	0	1
23	Mka	984916	984916	330962	267844	0	0	0	0
24	Pas	154812	154812	120505	112944	0	0	0	0
26	Cte	169383	169383	140692	141095	0	0	1	1
27	MmaC5	169568	169568	14711	11853	0	0	0	0
28	Dtu	939665	939665	452770	452770	0	0	0	0
29	TpeRKU-1	0	0	0	3831	0	0	0	2
30	TthHB8	60940	60940	60940	60940	1	1	1	0
31	Cli	102105	101377	94407	69167	2	2	2	3
33	Csa	39297	39297	21211	19844	5	4	4	4
34	Wsu	156243	156243	156243	156698	0	0	0	0
35	MmaS2	109948	109948	15609	9821	1	1	1	0
36	Cph	40887	40887	39734	32264	5	4	2	4
37	Pph	89343	89343	75358	66096	1	1	1	5
38	CauJ-10-fl	84420	84420	73383	49548	7	6	8	9
39	Amu	189693	189693	188054	165198	2	2	3	3
40	Cth	74152	74152	64874	56929	3	3	5	4
41	Pgi	30817	30817	28143	23770	3	3	2	3
43	Cbe	40725	40725	24143	16594	0	0	0	3
44	Tps	54527	54527	54527	37582	0	0	0	2
45	Lch	15689	15765	15178	15417	1	0	0	1
46	NPCC7120	137846	137846	141091	46894	3	3	3	4
48	Iho	212262	212262	158711	78403	0	0	0	0
49	Bth	132888	132888	112266	62811	7	7	4	8
50	Hvo	26447	25613	26447	24089	0	0	0	3
51	Hau	112847	107560	107560	67838	3	3	5	7

52	Sar	10460	10544	7564	4614	2	1	1	3
53	Str	9743	9571	6680	3861	1	1	1	8
54	Bvu	88327	88327	79291	48938	1	1	0	1
55	DraR1	15750	15750	15737	13499	0	0	0	0
56	MacC2A	36848	36848	31044	19289	9	9	7	5
58	Bbr	5650	5650	5555	5485	3	3	3	16
60	Rpo	12719	12719	12591	14150	0	0	0	0
61	Zmo	43038	43038	43038	50576	1	1	1	2
62	BxeLB400	4821	4821	4802	4939	12	12	11	18
63	SbaOS185	6015	6014	5924	6348	5	5	5	4
64	SbaOS223	0	0	0	1104	6	6	5	1

Table S8. Effects of various improvements in metaSPAdes on assembly of the SYNTH dataset. NGA50 statistics and the number of intragenomic misassemblies reported for 56 reference genomes from the SYNTH dataset. The reference genomes are arranged in the decreasing order of their coverage depths. The colors of the cells reflect how much the results of various assemblers differ from the median value (blue/red cells indicate that the results improve/deteriorate as compared to the median value).

As discussed in the Supplemental Material "Benchmarking SPAdes against metaSPAdes", feature (a) contributes to the increased stability and accuracy of the metagenomic assemblies. For example, the total number of intragenomic (intergenomic) misassembles decreased from 142 (51) for SPAdes SC FIX (which was the most accurate of all tested SPAdes configurations) to 97 (15) for metaSPAdes –c –b (Table S8). The coverage-based decision rule (c) plays an important role in the improved reconstruction of the most abundant across closely related genomes. As NGA50 statistics in Table S8 illustrates, it contributed to the improved contiguity of ThRQ2 assembly, discussed in section "Analysis of the SYNTH dataset". Table S8 also illustrates that feature (b) resulted in a substantial contiguity gain even though the SYNTH dataset has few closely related organisms. This is likely explained by the positive effect of the two-step strategy on the resolution of imperfect repeats, which were collapsed (into long perfect repeats) during graph simplification. Altogether, various algorithmic improvements resulted in more than 30% gain in metaSPAdes NGA50 statistics compared to SPAdes SC FIX for 27 out of 56 considered reference genomes. In contrast, more than 10% decrease in metaSPAdes NGA50 statistics compared to SPAdes SC FIX was observed for only 4 reference genomes.

Figure S5 illustrates the effects of various algorithmic improvements in metaSPAdes on the

assembly of the HMP dataset. Note that SPAdes SC FIX show very similar results to IDBA-UD, while various improvements in metaSPAdes contribute to the gradual improvements in the assembly contiguity (see Figures 1 and S1 for MEGAHIT and Ray-Meta results).

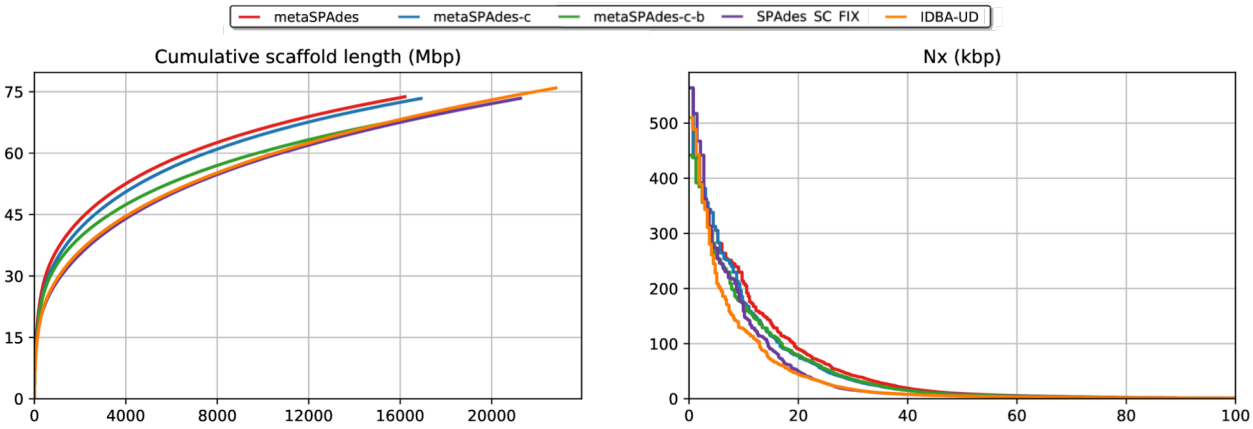

Figure S5. Effects of individual metaSPAdes features on the assembly of HMP dataset.

Supplemental Material: Analysis of the HMP dataset

As described in the main text, we identified only three references that were at least 70% covered by contigs generated by at least one of four assemblers analyzed in this study (*Streptococcus salivarius SK126, Neisseria subflava NJ9703*, and *Prevotella melaninogenica ATCC 25845* abbreviated as *Ssa, Nsu, and Pme,* respectively). Figure S6 presents benchmarking results for these three genomes.

Note that the number of reported errors in the HMP assembly significantly exceeds the number of errors for the SYNTH and CAMI datasets or the number of errors in typical assemblies of isolates. We believe that most of these errors represent metaQUAST artifacts (rather than true assembly errors) caused by the significant differences between the three recruited references and the related genomes in the sample. Poor coverage of the two out of three references genomes by the assembly contigs further confirms this conclusion.

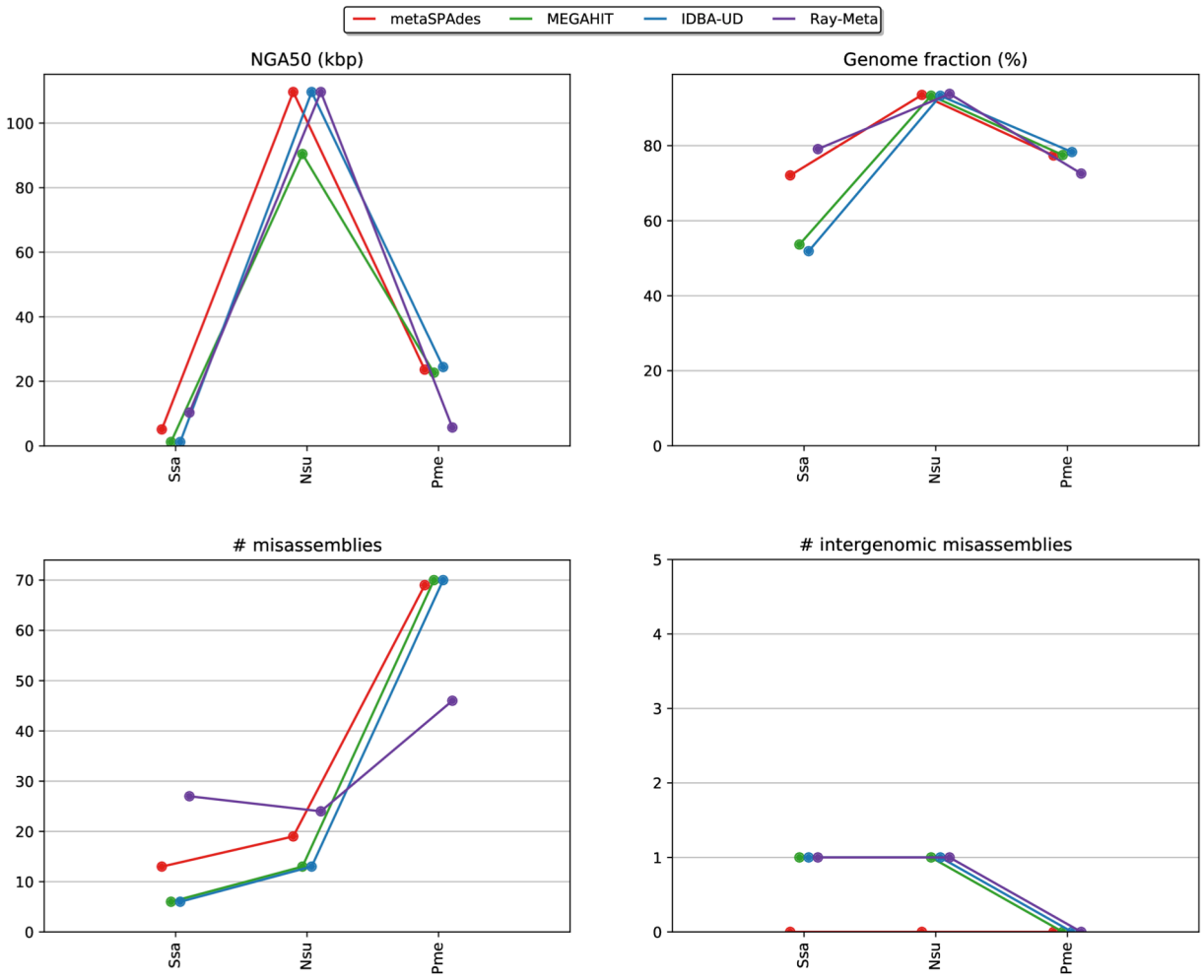

Figure S6. The NGA50 statistics (top left), the fraction of the reconstructed genome (top right), the number of intragenomic misassemblies (bottom left) and the number of intergenomic misassemblies (bottom right) for three reference genomes identified for the HMP dataset. References are arranged in the decreasing order of their average coverage-depths (183X, 118X, and 15X for *Ssa, Nsu*, and *Pme,* respectively).

References

Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, Lesin VM, Nikolenko SI, Pham S, Prjibelski AD, et al. 2012. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. *J Comput Biol* **19**: 455–477.

Bloom BH. 1970. Space/time trade-offs in hash coding with allowable errors. *Commun ACM* **13**: 422–426.

Botelho FC, Pagh R, Ziviani N. 2013. Practical perfect hashing in nearly optimal space. *Inf Syst* **38**: 108–131.

Bowe A, Onodera T, Sadakane K, Shibuya T. 2012. Succinct de Bruijn Graphs. In *Algorithms in Bioinformatics*, pp. 225–235.

Chapman JA, Ho I, Sunkara S, Luo S, Schroth GP, Rokhsar DS. 2011. Meraculous: de novo genome assembly with short paired-end reads. *PLoS One* **6**: e23501.

Chikhi R, Rizk G. 2013. Space-efficient and exact de Bruijn graph representation based on a Bloom filter. *Algorithms Mol Biol* **8**: 22.

Fan L, McElroy K, Thomas T. 2012. Reconstruction of ribosomal RNA genes from metagenomic data. *PLoS One* **7**: e39948.

Iqbal Z, Caccamo M, Turner I, Flicek P, McVean G. 2012. De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nat Genet* **44**: 226–232.

Korobeynikov A, Gorshkov Y, Nurk S, Bankevich A. 2017. LargeSPAdes: scaling SPAdes assembler for large genomes. *(in preparation)*.

Li D, Luo R, Liu C-M, Leung C-M, Ting H-F, Sadakane K, Yamashita H, Lam T-W. 2016. MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices. *Methods* **102**: 3–11.

Martin M. 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal* **17**: 10.

Mikheenko A, Valin G, Prjibelski A, Saveliev V, Gurevich A. 2016. Icarus: visualizer for de novo assembly evaluation. *Bioinformatics* **32**: 3321–3323.

Nikolenko SI, Korobeynikov AI, Alekseyev MA. 2013. BayesHammer: Bayesian clustering for error correction in single-cell sequencing. *BMC Genomics* **14**: S7.

Nurk S, Bankevich A, Antipov D, Gurevich A a, Korobeynikov A, Lapidus A, Prjibelski AD, Pyshkin A, Sirotkin A, Sirotkin Y, et al. 2013. Assembling Single-Cell Genomes and Mini-Metagenomes From Chimeric MDA Products. *J Comput Biol* **20**: 714–737.

Pell J, Hintze A, Canino-Koning R, Howe A, Tiedje JM, Brown CT. 2012. Scaling metagenome sequence assembly with probabilistic de Bruijn graphs. *PNAS* **109**: 13272–13277.

Prjibelski AD, Vasilinetc I, Bankevich A, Gurevich A, Krivosheeva T, Nurk S, Pham S, Korobeynikov A, Lapidus A, Pevzner P a. 2014. ExSPAnder: A universal repeat resolver for DNA fragment assembly. *Bioinformatics* **30**: 293–301.

Roy RS, Bhattacharya D, Schliep A. 2014. Turtle: Identifying frequent k-mers with cache-efficient algorithms. *Bioinformatics* **30**: 1950–1957.

Salikhov K, Sacomoto G, Kucherov G. 2014. Using cascading Bloom filters to improve the memory usage for de Brujin graphs. *Algorithms Mol Biol* **9**: 2.

Vasilinetc I, Prjibelski AD, Gurevich A, Korobeynikov A, Pevzner P. 2015. Assembling short reads from jumping libraries with large insert sizes. *Bioinformatics* **31**: 3262–3268.