nature methods

BRIEF COMMUNICATION

https://doi.org/10.1038/s41592-019-0437-4

Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold

Martin Steinegger[1,2,3]*, Milot Mirdita[1] and Johannes Söding[1]*

The open-source de novo protein-level assembler, Plass (https://plass.mmseqs.com), assembles six-frame-translated sequencing reads into protein sequences. It recovers 2–10 times more protein sequences from complex metagenomes and can assemble huge datasets. We assembled two redundancy-filtered reference protein catalogs, 2 billion sequences from 640 soil samples (soil reference protein catalog) and 292 million sequences from 775 marine eukaryotic metatranscriptomes (marine eukaryotic reference catalog), the largest free collections of protein sequences.

A major limitation of shotgun metagenomics is that often a large fraction of short reads (80–90% in soil[1]) cannot be assembled into contiguous sequences (contigs) long enough to predict gene and protein sequences. Because low-abundance genomes are difficult to assemble, the unassembled reads contain a disproportionately large part of the genetic diversity and probably an even greater share of biological novelty, which is lost for subsequent analyses.

To decrease both this loss and the dependence on reference genomes, gene-centric approaches have been developed. Assemblies of hundreds of samples from one environment are pooled, and genes in the contigs are predicted and clustered into gene catalogs[2,3]. Gene abundances in each sample are found by mapping of reads to the reference gene clusters. Thereby, the functional and taxonomic composition of metagenomic samples and their dependence on environmental parameters can be studied. Also, since genes from the same genome should have the same abundance profiles across many samples, they can be binned together, thus enabling genome-based analyses[4].

State-of-the-art assemblers for metagenomic short reads[5–7] find contigs as paths through a de Bruijn graph. It has a node for each *k*-mer word in the reads and edges between *k*-mers occurring consecutively in a read. On metagenomic data, de Bruijn assemblers suffer from a limited sensitivity–selectivity trade-off: *k*-mers have to be long and specific to avoid the graph exploding with false edges. But long *k*-mers lack sensitivity when intra-population diversities are high and overlapping reads contain mismatches from single-nucleotide polymorphisms (SNPs). Whatever the *k*, *k*-mers will be too short to be specific enough in genomic regions conserved between species and too long to be sensitive enough in regions of high intra-population diversity. This dilemma leads to short, fragmented assemblies.

Assembling protein instead of nucleotide sequences has several advantages. (1) Most SNPs in microbial populations are silent or conservative in the encoded protein sequences (Supplementary Fig. 1). (2) Proteins have fewer and shorter repeats, a major challenge for assemblers. (3) Chimeric assemblies between similar protein sequences (≥97% sequence identity) are much less problematic in that they do not lead to false conclusions about which genes occur together in a genome. However, the two published protein assemblers, ORFome[8] and SFA–SPA[9], are too slow for large metagenomes and, as de Bruijn assemblers, they suffer from a limited specificity–sensitivity trade-off.

Plass uses a novel graph-free, greedy iterative assembly strategy (Fig. 1) that, together with its linear-time all-versus-all overlap computation (steps 2–4)[10], scales linearly in runtime and memory. This permits the overlap-based assembly of huge read sets on a single server. By computing full alignment overlaps instead of only *k*-mer matches, Plass overcomes the specificity–sensitivity limitation of de Bruijn assemblers, thus allowing it to recover several times more protein sequences from complex metagenomes.

Plass needs only 1 byte for every amino acid translated from the input reads, or ~500 GB of random-access memory (RAM) to assemble 2–3 billion 2 × 150 base pair (bp) reads. In contrast, memory and runtimes of overlap graph assemblers scale superlinearly. Plass combines their high specificity and sensitivity with the linear runtime and memory scaling of de Bruijn graph assemblers.

With our greedy assembly approach, the most critical aspect to analyze is the fraction of wrongly assembled sequences (precision). We therefore sought two challenging datasets containing many related genomes, increasing the risk of chimeras[11]. The first set consists of 96 single-cell assembled genomes of *Prochlorococcus*[12], cyanobacteria known for their high intra-species genetic diversity. The second set contains 738 marine single-cell assembled genomes: 489 *Prochlorococcus* and 249 genomes from a diverse range of prokaryotic and viral groups[13]. As ground truth, we predicted protein sequences on the genomes using Prodigal[14]. We simulated 2 × 150 bp reads with a mean coverage of 1 for each genome.

We assembled protein sequences using Plass and SFA–SPA[9]. We assembled nucleotide contigs with Velvet[5] and two of the top assemblers in recent benchmarks[11,15], Megahit[6] and metaSPAdes[7]. We predicted protein sequences in their contigs using Prodigal. We ignored unassembled reads by removing protein sequences with fewer than 100 residues.

The sensitivity is similar for the three nucleotide assemblers, whereas Plass assembles up to 56% more residues correctly than the next best tool (Fig. 2a). The Plass-assembled proteins cover over 80% of Megahit and metaSPAdes assemblies, whereas these cover only 40% of the Plass assembly (Supplementary Fig. 2).

[1]Quantitative and Computational Biology Group, Max-Planck Institute for Biophysical Chemistry, Göttingen, Germany. [2]Department of Chemistry, Seoul National University, Seoul, Korea. [3]Center for Computational Biology, McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins School of Medicine, Baltimore, MD, USA. *e-mail: martin.steinegger@mpibpc.mpg.de; soeding@mpibpc.mpg.de

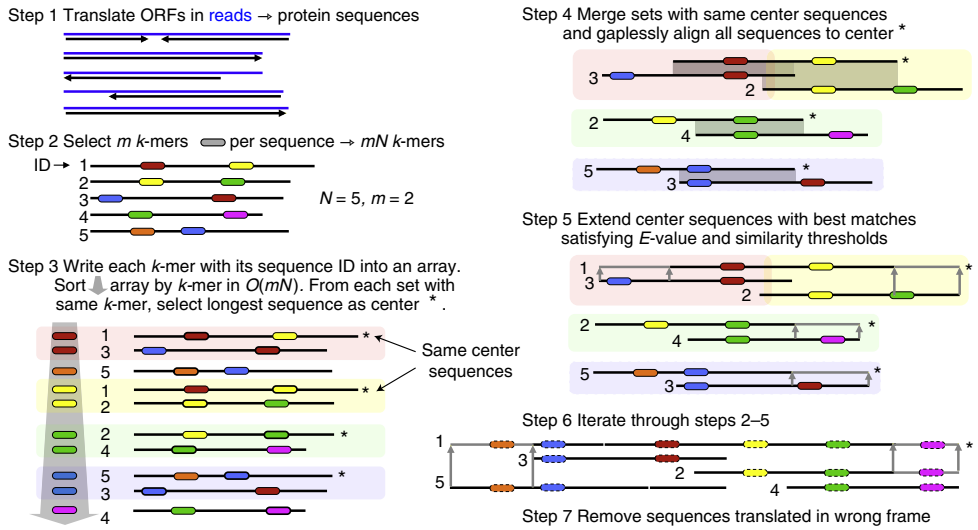

Fig. 1 | Plass workflow. Step 1: merge overlapping read pairs and translate all potential ORFs with ≥45 codons into protein sequences. Step 2: for each of these *N* sequences, select the *m* *k*-mers with the lowest hash values (default: $m=60$, $k=14$, reduced alphabet size 13). Write the *mN* *k*-mers with sequence identifiers into an array. ID, identification. Step 3: sort the array by *k*-mer to find for each *k*-mer the set of sequences containing it. Assign the longest sequence as the set's center. Step 4: resort the array by center sequence into groups and gaplessly align the center sequence to each group member (<*mN* alignments). Remove sequences with an *E* value > 10^{-5} from the group. Step 5: iteratively extend each center sequence by the group member with highest sequence identity (default minimum: 90%) until no further extensions are possible. Step 6: iterate steps 2–5 (default: 12×). Step 7: remove sequences translated in the wrong frame using a neural network.

Plass achieved the same precision below $X=97\%$ as the nucleotide assemblers (Fig. 2b). This is remarkable because the benchmark is biased against the protein assemblers: open reading frames (ORFs) are predicted on the nucleotide assemblies using the same tool used to define the reference protein sequences. The 2–7% of missing precision at $X=90\%$ is mainly caused by mispredicting ORFs in the assembled sequences or the single-cell genomes. Plass's neural network filter for suppressing proteins translated in the wrong frames (Methods) raised the precision at $X=90\%$ on the larger set by a few percent (Supplementary Fig. 3b).

Plass produced far fewer proteins at 99% sequence identity than the nucleotide assemblers, particularly on the larger dataset (Fig. 2b). Increasing the sequence identity threshold for merging sequence fragments from 90% to 97% (pink trace) markedly improved sensitivity and precision on both datasets, but still fewer proteins matched a reference protein with identity of ≥97%.

We therefore investigated whether the assembly of chimeric protein sequences impairs functional annotations. We annotated each assembled protein sequence with eggNOG[16] and compared its eggNOG annotation with the annotation of the best-matching sequence found in the reference protein set. We scored a true positive if annotations matched and a false positive if it did not or if assembled proteins could not be matched to any reference. Despite Plass's lower assembly precision, annotations achieved lower FDR values (=false positives/(true positives + false positives)) than those of the other assemblers (Fig. 2c). We believe this is due to (1) the high conservation of molecular and cellular functions at sequence identities above 70%[17], (2) the limited ability of homology-based annotation tools to predict the effects of point mutations and, most important, (3) the positive impact of more complete protein sequences on the prediction accuracy.

On real metagenomic datasets, no ground-truth reference sequences exist and precision cannot be measured. However, sensitivity in terms of the total number of assembled amino acids can be compared. We used four test sets: a single 11.3 gigabase pair sample from the human gut[18], 775 samples with 15 terabase pairs of eukaryotic metatranscriptome reads from TARA[19], a 31 gigabase pair sample from Hopland grass soil (Brodie et al., unpublished;

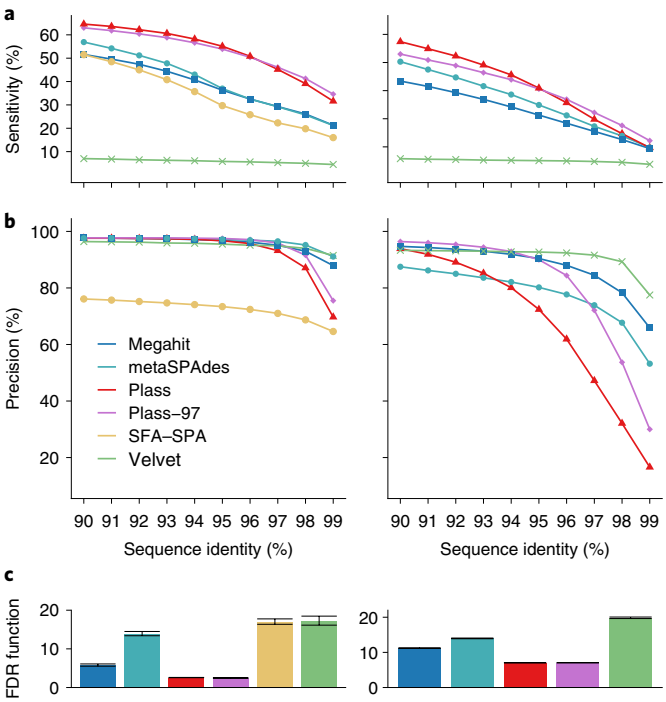

Fig. 2 | Sensitivity and precision of protein sequences assembled from synthetic reads and associated false discovery rate (FDR) of functional annotations. Reads were sampled from single-cell assembled genomes of 96 *Prochlorococcus* cells (left) and of 738 diverse marine prokaryotic cells and viruses (right). For the three nucleotide assemblers, we predicted protein sequences with Prodigal. **a**, Assembly sensitivity is the fraction of reference sequence amino acids that are aligned to an assembled protein sequence with a sequence identity of at least the value on the *x* axis. **b**, Assembly precision is the fraction of assembled amino acids that are aligned to a reference protein with sequence identity of at least the value on the *x* axis. **c**, FDR of eggNOG functional annotations for the assembled proteins. Color-coding as in **b**.

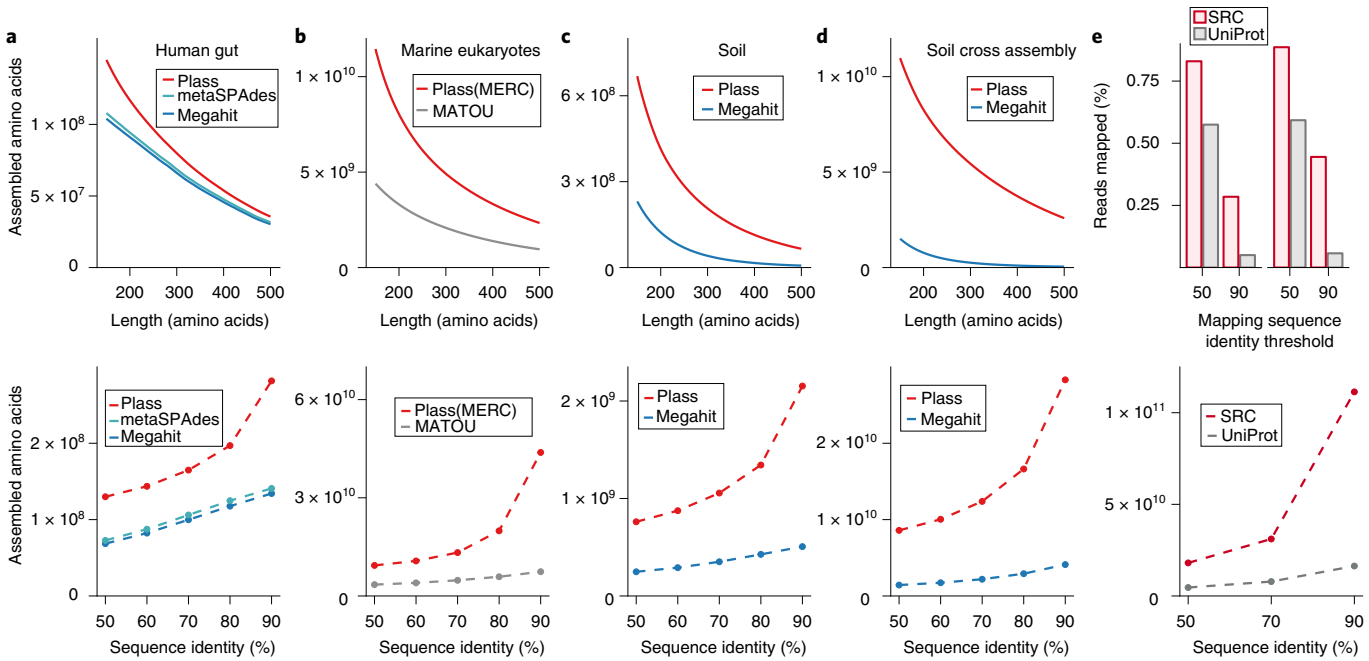

Fig. 3 | Plass assembles more protein sequences from various environments than the state of the art. a–d, Total number of amino acids in redundancy-filtered sets of protein sequences assembled by Plass (red traces) compared to the total number of amino acids of redundancy-filtered protein sequences predicted by Prodigal on contigs assembled by Megahit (**a**,**c**,**d**, blue) or on contigs in the eukaryotic metatranscriptome reference assembly (**b**, gray)[19]. Top half: dependence on the minimum protein sequence length using a redundancy-filtering with 80% maximum pairwise sequence identity. Bottom half: dependence on the strength of redundancy-filtering for a minimum sequence length of 100 amino acids. **e**, Top half: fraction of reads sampled from two soil metagenomes (not included in the SRC) that could be mapped with 90% and 50% minimum sequence identity to a sequence in the SRC or in UniProt. Bottom half: numbers of amino acids in SRC (red) and UniProt (gray) after redundancy-filtering with maximum pairwise sequence identity of 50%, 70% and 90%.

data publically available under project accession PRJNA330082) and 538 gigabase pairs of reads in 12 samples from the same project to test co-assembly (Fig. 3a–d).

For human gut, Hopland soil and the soil co-assembly, Plass took 4 h 20 min, 6 h 20 min and 360 h, while Megahit took 3 h, 21 h 30 min and 200 h, respectively. On the gut sample, Plass assembled 32% more amino acids than Megahit+Prodigal. The marine eukaryotic reference catalog (MERC) assembled by Plass is 2.8-fold larger than the Marine Atlas of TARA Oceans Unigenes (MATOU) assembled by Velvet, and on the soil data Plass assembled 2.7 times more than Megahit. The 12 soil datasets could be not be co-assembled by Megahit owing to insufficient memory; therefore we assembled each sample separately and pooled the contigs. Plass co-assembled ten times more amino acids than Megahit.

The increase of the ratio with sequence length (Fig. 3c,d, top) indicates that the sequences assembled by Plass are considerably longer than those of Megahit+Prodigal. The gains in recovered protein sequences are similar at all levels of redundancy up to 80% sequence identity (bottom half of Fig. 2a–d).

The improved sensitivity of Plass over nucleotide assemblers also markedly affects the apparent taxonomic composition of metagenomics samples (Supplementary Fig. 4).

Finally, we assembled a soil reference protein catalog (SRC) from 18 Tbp of reads from all 640 soil samples sequenced between January 2016 and February 2018 using Illumina HiSeq/NovaSeq with 2×150 bp reads. The assembly took 6 weeks on 25 servers, each with 16 cores and 128 GB memory, and yielded 12 billion sequences. Redundancy reduction by clustering at 90% sequence identity using Linclust[10] resulted in 2 billion sequences containing 3.2×10^{11} amino acids. This dataset is 6.8, 4.0 and 3.9 times larger than UniProt after redundancy-filtering at 90%, 70% and 50%, respectively (Fig. 3e).

To assess how much the SRC represents soil metagenome diversity, we selected two soil samples not in the SRC, randomly sampled 10,000 read pairs, predicted protein sequences and searched with these through the 90%-redundancy-filtered versions of the SRC and UniProt. At 50% minimum sequence identity, 82.5% and 89.5% of the soil reads matched to the SRC in the two samples, while only 62% and 64% matched to UniProt (Fig. 3e).

Our chief limitation is that, unlike nucleotide assemblers, Plass cannot place the assembled protein sequences into genomic context. Furthermore, it cannot assemble intron-containing eukaryotic proteins, although, as shown, it can assemble eukaryotic proteins from transcriptome data. Another drawback is its inability to resolve homologous proteins from closely related strains or species with sequence identities above ~95%. However, the impact on the accuracy of predicted functions is low (Fig. 2) and bacterial phenotypes are determined more by the complement of horizontally acquired accessory genes than by minor variations in protein sequences.

With Plass, reference protein sequence catalogs for every environment can be generated, making the treasures of microbial evolutionary inventions accessible to biotechnology and pharmacology. By enriching multiple sequence alignments with diverse homologs, these catalogs will also improve homology detection, protein function and structure prediction[20].

Online content

Any methods, additional references, Nature Research reporting summaries, source data, statements of code and data availability and associated accession codes are available at https://doi.org/10.1038/s41592-019-0437-4.

Received: 1 August 2018; Accepted: 5 May 2019;
Published online: 24 June 2019

References

1. Howe, A. C. et al. *Proc. Natl. Acad. Sci. USA* **111**, 4904–4909 (2014).
2. Li, J. et al. *Nat. Biotechnol.* **32**, 834–841 (2014).
3. Sunagawa, S. et al. *Science* **348**, 1261359 (2015).
4. Nielsen, H. B. et al. *Nat. Biotechnol.* **32**, 822–828 (2014).
5. Zerbino, D. & Birney, E. *Genome Res.* **18**, 821–829 (2008).
6. Li, D. et al. *Bioinformatics* **31**, 1674–1676 (2015).
7. Nurk, S. et al. *Genome Res.* **27**, 824–834 (2017).
8. Ye, Y. & Tang, H. *J. Bioinform. Comput. Biol.* **7**, 455–471 (2009).
9. Yang, Y. et al. *Bioinformatics* **31**, 1833–1835 (2015).
10. Steinegger, M. & Söding, J. *Nat. Commun.* **9**, 2542 (2018).
11. Sczyrba, A. et al. *Nat. Methods* **14**, 1063–1071 (2017).
12. Kashtan, N. et al. *Science* **344**, 416–420 (2014).
13. Berube, P. M. et al. *Sci. Data* **5**, 180154 (2018).
14. Hyatt, D. et al. *BMC Bioinforma.* **11**, 119 (2010).
15. van der Walt, A. J. et al. *BMC Genom.* **18**, 521 (2017).
16. Huerta-Cepas, J. et al. *Mol. Biol. Evol.* **34**, 2115–2122 (2017).
17. Tian, W. & Skolnick, J. *J. Mol. Biol.* **333**, 863–882 (2003).
18. Lee, S. T. M. et al. *Microbiome* **5**, 50 (2017).
19. Carradec, Q. et al. *Nat. Commun.* **9**, 373 (2018).
20. Ovchinnikov, S. et al. *Science* **355**, 294–298 (2017).

Acknowledgements

We are grateful to C. Notredame and C. Seok for hosting M.S. at the Centre for Genomic Regulation and Seoul National University for 12 and 30 months, respectively. We thank S. Sunagawa, F. Meyer and A. Sczyrba for helpful discussions, and T. Brown for his early analysis and detailed feedback on Plass results. We thank all who contributed metagenomic datasets used to build SRC and MERC, in particular contributors to the TARA ocean project and the US Department of Energy Joint Genome Institute (http://www.jgi.doe.gov). This work was supported by the EU's Horizon 2020 Framework Programme (Virus-X, grant no. 685778).

Author contributions

M.S. and J.S. designed the research study. M.S. and M.M. developed code and performed the analyses. M.S. and J.S. wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41592-019-0437-4.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to M.S. or J.S.

Peer review information: Lei Tang was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Methods

Plass proceeds in seven steps, summarized in Fig. 1.

Merging paired-end reads and six-frame translation. In step 1, Plass merges overlapping paired-end reads into longer sequences using code from the open-source FLASH tool[21]. Plass then extracts all ORFs with at least 45 codons and translates them into protein sequences. (Alternative codon tables can be specified with option -translation-table.)

Finding overlaps in linear time. The identification of all overlapping alignments (Fig. 1, steps 2–4) is critical for the performance of overlap assemblers. Previously proposed protein-level assemblers have a runtime complexity that scales quadratically with the input set size[8,9]. A typical metagenomic read set with 100 million reads requires 10^{16} comparisons with a quadratic method. To speed up the computation, we adapted our linear-time clustering algorithm Linclust[10] for assembly.

In step 2, Plass transforms each protein sequence into a reduced amino acid alphabet, whose 13 letters represent the following groups of amino acids: (L, M), (I, V), (K, R), (E, Q), (A, S, T), (N, D) and (F, Y). From each reduced sequence it selects the m (default: $m=60$) k-mers with lowest hash values (or all k-mers if the sequence contains fewer than m). Our rolling hash function[10] maps each k-mer onto a range of $0,2^{16}$ such that even single residue changes result in quasi-random, unrelated hash values. For each of the $\sim mN$ selected k-mers, Plass stores in an array the k-mer index (8 bytes), the sequence identifier (4 bytes), the k-mer position in the sequence (2 bytes) and the length of the sequence (2 bytes).

In step 3, Plass sorts the array by k-mer index and sequence length to find the sets of sequences containing the same k-mer. For each set, it picks the longest sequence as the center sequence. For each member of the k-mer set, it overwrites the k-mer index with the center sequence identifier and computes the diagonal $i-j$ on which the shared k-mer match occurs, where i is the k-mer position in the center sequence and j is the position in the member sequence. The array now contains the center sequence identifier, the member sequence identifier, the k-mer match diagonal and the length of the member sequence. It sorts the array again, this time by center sequence identifiers, and removes duplicate center-member pairs. If more than one diagonal match between a center and member sequence is found, only the match with the lowest diagonal is kept.

In step 4, Plass computes an ungapped local alignment between each center sequence and each group member, using one-dimensional dynamic programming on the diagonal $i-j$ of the k-mer match. It computes E values using ALP[22] and, by default, the Blosum62 substitution matrix. Alignments with an $E > 10^{-5}$ (default) and a sequence identity < 90% (default) are rejected.

Extending protein reads. In step 5, Plass extends the center sequence by concatenating the non-overlapping residues of the member sequence with highest similarity in the overlap. More precisely, it processes the list of alignments with the member sequences in order of descending overlap sequence identity, until one side of the center sequence has been extended and the other side has either been extended as well or has no extending alignments left in the list. Then it realigns the extended center sequence with all yet unprocessed member sequences and iterates the extension until the entire list of alignments has been processed.

Iterative assembly. Plass iterates through steps 2–5 12 times (default), each time updating the original version of the center sequences with their extended versions and keeping all other sequences unchanged (step 6). To extract different k-mers in each new iteration, we increment the step size of the circular shift inside our rolling hash function[10].

Removing proteins translated in wrong frames. In step 7, Plass removes sequences translated from wrong ORFs or assembled from such sequences. ORFs translated in the wrong frame contain a stop codon approximately every 64/3 residues, and so only a fraction of around $\exp(-45/21.3) \approx 12\%$ contain at least 45 codons.

Plass filters out sequences translated from wrong ORFs based on their amino acid and dipeptide composition, which differs from correctly translated, real protein sequences. Here, we trained a neural network using as features the 20 length-normalized amino acid frequencies of the sequence and the 6^2 length-normalized dipeptide frequencies in a reduced alphabet with a size of six. Our fully connected network has 56 input nodes, a hidden layer of 96 nodes and a single output node.

We trained the network using the Keras deep-learning framework using the Adam optimizer with a 10% drop-out probability and the binary cross-entropy loss function. We left 10% of the data out for cross validation. The network was integrated into Plass using Kerasify.

For training, we created a positive set of known coding sequences and a negative set of sequences translated in a wrong reading frame. The positive set contained 2.4 million proteins sampled from the prokaryotic subset of the Uniclust30 representative sequences[23]. For the negative set, we extracted ORFs from all frames of 757 prokaryotic genomes contained in the KEGG database[24] and clustered them using MMseqs2 (ref. [25]) with a maximum sequence identity of 30% and a minimum coverage of 80%. Clusters without any member with coding sequence annotation in KEGG or homology to entries in Uniclust30 (requiring an E value of $<10^{-3}$) were extracted. From these, we sampled 2.4 million sequences.

Predicting start codons. To determine the correct start codon and minimize overextension at the N terminus in the ORF translation step 1, Plass (in step 1) extracts and translates all ORFs with at least 20 codons starting with a putative ATG start codon that is the first ATG codon after a stop codon in the same frame. These ATGs are marked by an asterisk (*) prepended before the methionine residue. These sequences help Plass to predict start codons, because the coding sequence cannot start before such an ATG (Supplementary Fig. 5).

After the alignment and extension step 4 in the first iteration, Plass reconstructs the multiple sequence alignment of all merged sequences. Where at least 20% of all methionines in a column are marked by a prepended asterisk, it removes the preceding residues from all other sequences and prepends an asterisk to all sequences to mark the start. If several columns fulfill the 20% criterion, it trims the sequences at the most downstream of these columns. The start codon prediction is only done in the first iteration to save time and disk space.

Suppressing repetitive sequences. Protein repeats can lead to unwanted extensions during assembly. We therefore detect sequences with repeat regions during step 2 as those containing at least eight (default) identical k-mers (in the 13-letter alphabet). These sequences are ignored during all steps.

Memory-efficient processing of huge input sets. Plass needs 1 byte per residue of translated protein sequence generated in step 1 to keep these sequences in memory and avoid random disk accesses in the alignment step 4. The k-mer array in steps 2 and 3 occupies $m \times 16$ bytes $= 720$ bytes of memory per sequence, which is around 16 times more than 1 byte per residue. To be able to assemble datasets whose k-mer array would not fit into main memory, Plass automatically splits the k-mer array into chunks that fit into main memory of size M and processes them sequentially, with little loss in speed. Plass sets the maximum number of chunks to $S = (mN \times 16\text{byte}/M)$. For each chunk c out of S, we proceed with steps 2 and 3 exactly as described before, except that we only extract k-mers whose index R satisfies $(R \bmod S) = c$ and that we store the chunks of k-mer arrays on hard disk. After all splits have been computed, we merge them into a single k-mer array.

Assembly quality benchmark. We could not use the standard benchmark developed by the Critical Assessment of Metagenomic Interpretation (CAMI)[11], because the mutation model of the sgEvolver tool used for simulating population microdiversity (strain diversity) does not penalize frame-disrupting indels and non-conservative substitutions within coding regions. This leads to very low and unrealistic conservation of coding regions. The synthetic reads generated with such a model are certainly realistic enough to test nucleotide assemblers but would render protein-level assembly absurdly unsuitable.

We sought to construct a genomic benchmarking set that would contain a high degree of natural variation, in which the genomic sequences reflect the actual evolutionary pressures on them. We downloaded from the sequence read archive (SRA) two sets of genomes assembled from single-cell sequencing libraries. The first set contains genomes of 96 *Prochlorococcus* genomes[12]. These cells were taken from the same ocean water sample and represent a population of the cyanobacteria *Prochlorococcus*, the most abundant marine photosynthetic organism on earth noted for high intra-species diversities. Sequence identities of 16S ribosomal RNA ITS sequences in a matched sample are between 50% and 100%. The second set contains 738 single-cell genome assemblies (NCBI project PRJNA445865) consisting of 489 *Prochlorococcus*, 50 *Synechococcus*, 82 SAR11, 17 SAR116, 16 SAR86, nine extracellular virus particles and 75 additional sympatric microorganisms, sampled at 22 locations in the Atlantic and Pacific Oceans[13].

As a ground-truth reference, we predicted protein sequences on the genomes using Prodigal[14] and removed sequences shorter than 100 residues, resulting in a redundant reference set of 109,014 protein sequences for set 1 and 829,899 for set 2. We reduced the redundancy by clustering with Linclust at 95% sequence identity and 99% minimum coverage of the shorter sequence (options --cov-mode 1 -c 0.99 --min-seq-id 0.95), resulting in the non-redundant reference set with 14,943 (set 1) and 460,653 (set 2) sequences.

We created two synthetic read datasets from the two sets of single-cell genomes, setting the mean coverage to 1 for each genome, which yielded 392,790 reads for set 1 and 4,994,546 for set 2. We used randomreads.sh from the BBmap software suite with options paired snprate=0.005 adderrors coverage=1 len=150 mininsert=150 maxinsert=350 gaussian=true to simulate 2×150 bp paired-end overlapping reads with sequencing errors.

We then assembled the synthetic paired-end read dataset with Megahit, metaSPAdes, Plass, SFA–SPA and Velvet, using the default parameters of each tool. We also tested Plass with with a stricter minimum sequence identity for merging sequences (option --min-seq-id 0.97), 'Plass-97' in Fig. 2. For the nucleotide assemblers, we called proteins from the assembled contigs using Prodigal in metagenomics mode. We ignored all proteins shorter than 100 residues.

We calculated the precision by searching with the assembled proteins through the redundant reference set, using MMseqs2 with options -a -s 5 --max-seqs 5000

--min-seq-id 0.89. We filtered the aligned set by minimum sequence identity thresholds between 90% and 99%. For each search result, we only considered the longest alignment that fulfilled the minimum sequence identity criterion. We computed the precision for each sequence identity threshold as the ratio of the total count of aligned residues divided by the total length of the assembled proteins. Precision of 100% was reached when all assembled protein residues could be aligned to a reference protein sequence.

To avoid giving too much weight to highly conserved proteins, we redundancy-filtered the reference proteins for the sensitivity analysis as described. We calculated the sensitivity by searching with the non-redundant reference set through the assembled proteins, using MMseqs2 with options -a -s 5 --max-seqs 500000 --min-seq-id 0.89. We filtered the aligned set by minimum sequence identity thresholds between 90% and 99%. For each search result, we only considered the longest alignment that fulfilled the minimum sequence identity criterion. We computed the sensitivity for each sequence identity threshold as the ratio of the total count of aligned residues divided by the total length of the proteins in the non-redundant set. Sensitivity of 100% was reached when all reference protein residues could be aligned to an assembled protein sequence.

Accuracy of functional annotation of assembled proteins. To test the impact of chimeric assemblies and assembly quality on functional annotation quality, we measured the accuracy of functional annotations on the proteins assembled from the reads simulated from the single-cell marine genomes (Fig. 2). We functionally annotated each protein sequence assembled from the simulated reads and each reference protein from the single-cell genomes with an Orthologous Group using the eggNOG mapper[16] and the eggNOG database (v.4.5.1)[26] (options -d bact -m diamond -override -cpu 16). We compared the eggNOG annotation of each assembled protein sequence with the annotation of the best-matching sequence found in the reference protein set using MMseqs2. If the Orthologous Groups differed, the annotation was false positive or otherwise true positive. Assembled proteins that could be matched to any reference protein were considered to be false positives. The error bars in Fig. 2c assume Poisson errors.

Protein sequence recovery on metagenomic datasets. For the benchmark test on real metagenomic data (Fig. 3a–d) we used the following datasets: (a) a single human gut sample from the SRA (SRR5024285)[18], (b) 775 samples from TARA eukaryotic metatranscriptomes downloaded from the European Nucleotide Archive (ENA) (PRJEB6609)[19], (c) a soil sample from Integrated Microbial Genomes & Microbiomes (IMG/M) project 1003784 (sample: 6398.7.44014) and (d) 12 samples from the same project (samples: 6679.7.51457 6478.6.45123, 6679.6.51456, 6398.7.44014, 6478.7.45124, 6674.6.51288, 6679.5.51455, 6674.4.51285, 6478.5.45122, 6478.4.45121, 6674.3.51284, 6674.5.51286). The soil data are also available at SRA project PRJNA330082. All samples used in Fig. 3a,c,d consisted of paired-end reads of 2 × 150 bp length, while dataset b consisted of reads with 2 × 102 bp length.

We assembled paired-end reads in datasets a, c and d using Megahit and Plass with default parameters. The benchmarks for sets in Fig. 3a–c were carried out on a single computer with two eight-core Intel Xeon E5-2640v3 central processing units (CPUs) with 128 GB RAM. The co-assembly in Fig. 3d was performed on a server with two 14-core Intel Xeon E5-2680v4 CPUs with 768 GB RAM. During the co-assembly, Megahit aborted with a segmentation fault on the 768 GB server. We therefore performed 12 separate assemblies and pooled the results.

We could compare Plass only to Megahit on datasets c and d, since Velvet terminated with segmentation faults, metaSPAdes terminated with messages specifying a required amount of RAM in excess of the available 128 GB, and SFA–SPA did not finish execution within 3 d.

For Fig. 3b, we assembled the 775 TARA metatranscriptomes using Plass and compared the results with the MATOU catalog[19], assembled using Velvet. For that purpose, we called protein sequences using Prodigal in metagenomics mode on all MATOU contigs, since these often do not contain full-length protein sequences. Eukaryotic protein sequences contain repeats more frequently than viral or prokayotic ones. We therefore masked low complexity regions of the assemblies created by Plass using tantan[27] and removed all assembled proteins with more than 50% masked residues.

To analyze the diversity of the obtained sets at various redundancy levels, we clustered all assembled protein sequence sets with Linclust using the parameters --kmer-per-seq 80 --cluster-mode 2 --cov-mode 1 -c 0.9 at sequence identity thresholds --min-seq-id from 50% to 90%.

Taxonomic classification and quantification. We investigated the influence of the assembly method on the taxonomic composition (Supplementary Fig. 6). Instead of matching nucleotide reads to reference genomes, we performed the taxonomic matching on the protein level because, first, many species sampled with metagenomics do not contain a close homolog in the reference databases and, second, protein-level comparison affords a much higher sensitivity.

Our strategy was (1) to map reads via the translated ORFs they contain to assembled protein sequences and (2) to map the assembled protein sequences to taxonomic nodes in the NCBI taxonomic tree. We thereby mapped each read transitively to one taxonomic node.

To map the assembled protein sequences to taxonomic nodes (step 2 above), we implemented the 2bLCA protocol[28] as new MMseqs2 module mmseqs taxonomy (Supplementary Fig. 4) and assigned the assembled protein sequences to the 90% redundancy-filtered UniProt database (Uniclust90 2017_07)[23], which contains taxonomic assignments to the NCBI tree for each sequence.

Using the two-step transitive mapping, we computed read counts for all taxonomic nodes. We then pooled the counts for each phylum in the tree and in addition recorded counts of reads assigned by 2bLCA to taxa above the phylum level. Only the eight most abundant taxa were then kept and counts of all others were pooled into a category 'Others'.

In Supplementary Fig. 6a, we show the results for the soil sample assemblies from Fig. 3c (blue, Megahit; red, Plass) and the assemblies of the 12 soil samples from Fig. 3d (light blue, Megahit; light red, Plass), together with the ratios on top. The inset gives the fraction of reads in the single and the 12 soil samples that could be mapped to an assembled protein sequence with a minimum sequence identity of 90% (step 1 above).

In Supplementary Fig. 6b, we show the count of assembled amino acids within various coverage ranges. Coverage of an assembled protein sequence is the sum of the number of residues aligned to that sequence during mapping divided by the length of the assembled protein sequence.

Around five to ten times more reads can be mapped to the set of protein sequences assembled by Plass (red) than to the set predicted by Prodigal on the Megahit assembly (blue). The gains are particularly high for high coverages.

SRC and analysis. We downloaded from the SRA all 640 metagenomic datasets that (1) had the 'soil metagenome' taxon identifier, (2) had dates between January 2014 and February 2018, (3) were sequenced on Illumina HiSeq or NovaSeq machines and (4) had paired-end reads of at least 2 × 150 bp length. Sample identifiers are contained in a file SRC_sample_ids.txt at https://github.com/martin-steinegger/plass-analysis.

Plass assembled the 18 Tbp of raw reads on 25 servers with 2 × 8-core Intel Xeon E5-2640v3 CPUs and 128 GB of RAM. We removed protein sequences shorter than 100 residues and redundancy-filtered the protein sequences from each sample using Linclust with options --min-seq-id 0.95 --alignment-mode 3 -c 0.99 --cov-mode 1 --cluster-mode 2). We pooled these 12 billion protein sequences and further reduced their redundancy by clustering with Linclust (--cov-mode 1 -c 0.9 --min-seq-id 0.9). The clustering was done hierarchically, since Linclust can only process $2^{32} - 1$ sequences at once. The final set contains 2,022,891,389 sequences. At least 52.3 million of these sequences are complete, because Plass found the stop codon and the earliest possible start codon.

We chose two metagenomic soil sets (SRR5919294 and SRR6201924) that were not part of the 640 datasets used for building the SRC. We sampled 100,000 paired reads per sample, merged the overlapping read pairs using FLASH[21], predicted protein sequence fragments using Prodigal[14], and searched through the 90%-redundancy-filtered versions of SRC and UniProt[23] using the mmseqs map workflow (below). We computed the fraction of mapped reads out of the total read count, demanding a minimum sequence identity of 50% or 90% (option --min-seq-id).

Read mapping. We used the novel mmseqs map workflow from the MMseqs2 package to find very similar matches in a protein sequence database. It first calls the mmseqs prefilter module with a low sensitivity setting of -s 2 to detect high scoring diagonals and then computes an ungapped alignment using the mmseqs rescorediagonal module. To achieve maximum speed, no gapped alignment is computed, query sequences are not masked for low complexity regions (--mask-mode 0) and no compositional bias correction is applied (--comp-bias-corr 0). By default, the mapping workflow requires that 90% of query sequence residues are aligned to a database sequence (--cov-mode 2 -c 0.9).

Software versions used. Prodigal v.2.6.3, FLASH v.1.2.11, Velvet v.1.2.10, SFA–SPA v.0.2.1, metaSPAdes v.3.10.1, Megahit v.1.1.1-2-g02102e1 and eggNOG mapper v.1.0.3 were all used.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The assembled protein sequence sets are available in FASTA format under a Creative Commons Attribution CC-BY 4.0 License at https://plass.mmseqs.com. All scripts and benchmark data including command-line parameters necessary to reproduce the benchmark and analysis results presented are available at https://github.com/martin-steinegger/plass-analysis.

Code availability

Plass is GPLv3-licensed open-source software. The source code and binaries for Plass can be downloaded at https://github.com/soedinglab/plass.

References

21. Magoc, T. & Salzberg, S. L. *Bioinformatics* **27**, 2957–2963 (2011).
22. Sheetlin, S. et al. *Bioinformatics* **32**, 304–305 (2016).
23. Mirdita, M. et al. *Nucleic Acids Res.* **45**, D170–D176 (2017).
24. Kanehisa, M. et al. *Nucleic Acids Res.* **45**, D353–D361 (2016).
25. Steinegger, M. & Söding, J. *Nat. Biotechnol.* **35**, 1026–1028 (2017).
26. Huerta-Cepas, J. et al. *Nucleic Acids Res.* **44**, D286–D293 (2016).
27. Frith, M. *Nucleic Acids Res.* **39**, E23 (2011).
28. Hingamp, P. *ISME J.* **7**, 1678–1695 (2013).

nature research

Corresponding author(s): Johannes Soeding, Martin Steinegger

Last updated by author(s): 2019/03/15

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☒	☐	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☒	☐	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Plass is GPLv3-licensed open source software. The source code and binaries for Plass can be downloaded at https://github.com/soedinglab/plass. The assembled protein sequence sets are available as FASTA formatted files at https://plass.mmseqs.org.
Data analysis	All scripts and benchmark data including command-line parameters necessary to reproduce the benchmark and analysis results presented are available at https://github.com/martin-steinegger/plass-analysis.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

We link the ressources in the Code availability and Data availability. Our software is GPLv3-licensed and the data is available under Creative Commons Attribution CC-BY 4.0 License

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We took the largest available set of single cell related Prochlorococcus genomes (> 500 genomes) for the benchmark
Data exclusions	No data was excluded
Replication	We replicated our benchmark on two different synthetic benchmarks with 96 genomes and >500 genomes. We also tested our method on four real metagenomes (Human Gut, Marine, Single Soil, multiple Soil samples)
Randomization	We did not apply randomization in this in-silico study.
Blinding	Data were not partitioned into groups

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging