BlobTools

Interrogation of genome assemblies

Check for updates

SOFTWARE TOOL ARTICLE

BlobTools: Interrogation of genome assemblies [version 1; referees: 2 approved with reservations]

Dominik R. Laetsch [1,2], Mark L. Blaxter[1]

[1]Institute of Evolutionary Biology, University of Edinburgh, Edinburgh, EH9 3JT, UK
[2]The James Hutton Institute, Dundee, DD2 5DA, UK

v1 **First published:** 31 Jul 2017, **6**:1287 (doi: 10.12688/f1000research.12232.1)
Latest published: 31 Jul 2017, **6**:1287 (doi: 10.12688/f1000research.12232.1)

Abstract

The goal of many genome sequencing projects is to provide a complete representation of a target genome (or genomes) as underpinning data for further analyses. However, it can be problematic to identify which sequences in an assembly truly derive from the target genome(s) and which are derived from associated microbiome or contaminant organisms.
We present BlobTools, a modular command-line solution for visualisation, quality control and taxonomic partitioning of genome datasets. Using guanine+cytosine content of sequences, read coverage in sequencing libraries and taxonomy of sequence similarity matches, BlobTools can assist in primary partitioning of data, leading to improved assemblies, and screening of final assemblies for potential contaminants.
Through simulated paired-end read dataset,s containing a mixture of metazoan and bacterial taxa, we illustrate the main BlobTools workflow and suggest useful parameters for taxonomic partitioning of low-complexity metagenome assemblies.

Open Peer Review

Referee Status: ? ?

	Invited Referees	
	1	**2**
version 1 published 31 Jul 2017	? report	? report

1 **A. Murat Eren**, University of Chicago, USA
Marine Biological Laboratory , USA

2 **Richard M Leggett**, Earlham Institute, UK

Discuss this article

Comments (0)

Corresponding author: Dominik R. Laetsch (dominik.laetsch@gmail.com)

Author roles: Laetsch DR: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Project Administration, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Blaxter ML**: Conceptualization, Funding Acquisition, Methodology, Project Administration, Resources, Supervision, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

How to cite this article: Laetsch DR and Blaxter ML. **BlobTools: Interrogation of genome assemblies [version 1; referees: 2 approved with reservations]** *F1000Research* 2017, **6**:1287 (doi: 10.12688/f1000research.12232.1)

Grant information: DRL was supported by a James Hutton Institute/Edinburgh University School of Biological Sciences fellowship. MLB was supported by a BBSRC research grant (Project reference BB/P024238/1).
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

First published: 31 Jul 2017, **6**:1287 (doi: 10.12688/f1000research.12232.1)

Introduction

Advances in next generation sequencing technologies have generated vast amounts of data and knowledge (Goodwin *et al*., 2016). The decrease in cost per nucleotide lead to an increased application of these technologies to non-model organisms, life forms which have so far not been intensively studied by the research community. Genome-enabled science on these species can then illuminate novel processes and reveal the patterns of evolution. For non-model species, the luxury of large amounts of material from cultured isolates is often not possible, and research must progress from organisms sourced from the wild or from complex mixtures of species. DNA extracted from a sample may actually contain genomes from multiple organisms – food sources, host material, symbionts, pathogens, commensals and external contaminants – in addition to the target organism. In some cases, the associated genomes can be considered "contaminants", while in others, they can provide insights into the biology of the target organism. In all cases they should be identified, isolated and investigated with care.

Interrogation of genome assemblies to assure single-taxon origin is an elemental step in the genome sequencing process. Failure to identify non-target sequence can lead to false conclusions regarding the biology of the target organism, such as metabolic potential and events of horizontal gene transfer (HGT) between species. Several reports of HGTs into eukaryotic genomes have later been shown to have been based on undetected contamination in assemblies. Identification of contamination can radically change the conclusions of a study, as shown for the starlet sea anemone *Nematostella vectensis* (Artamonova & Mushegian, 2013) and the tardigrade *Hypsibius dujardini* (Koutsovoulos *et al*., 2016). Importantly, undetected non-target sequence contamination of published genomes will pollute public sequence databases and promote propagation of annotation errors.

Reliable assignment of a DNA sequence from a new assembly to its species-of-origin, *i. e.* the association of the sequence ID to an unique, numerical identifier (TaxID) of the National Centre for Biotechnology Information (NCBI) Taxonomy database (Federhen, 2012), is a non-trivial problem. Current contaminant screening pipelines are based on sequence similarity to sequences of known origin, sequence composition signatures such as *k*-mers, and/or shared coverage profiles across different datasets. Few are readily applicable to datasets of eukaryotic genomes of any size (Eren *et al*., 2015; Kumar *et al*., 2013; Mallet *et al*., 2017; Tennessen *et al*., 2016). Anvi'o (Eren *et al*., 2015) partitions assemblies by clustering sequences based on the output of CONCOCT (Alneberg *et al*., 2014). CONCOCT uses Gaussian mixture models to predict the cluster membership of sequences by considering sequence composition and coverage profiles. PhylOligo (Mallet *et al*., 2017) relies exclusively on sequence composition and performs iterative, partially supervised clustering of sequences based on sequence composition profiles. ProDeGe (Tennessen *et al*., 2016) uses a fully unsupervised method based on sequence similarity to databases and sequence composition to partition assemblies using principal component analysis (PCA). It should be noted that while taxonomic assignment based on higher order sequence composition (such as *k*-mers of length 4 or greater) is highly effective for bacterial sequences, its success has been limited for eukaryotic genomes, as the information content, represented by the number of coding bases, is lower, and sequence composition spectra often show multimodal distributions (Chor *et al*., 2009).

Existing contaminant screening pipelines also differ in the way results are presented. Anvi'o depicts assemblies through interactive plots with rich annotations of sequence composition features, coverages across datasets and taxonomic/binning results. PhylOligo offers heatmaps of hierarchical clusterings of sequences, tree visualisations, and t-SNE (t-Distributed Stochastic Neighbor Embedding) plots, where sequence composition clusterings have been reduced to two dimensions. ProDeGe displays sequences in an interactive, three-dimensional *k*-mer PCA plots.

BlobPlots, or taxon-annotated GC-coverage plots (Kumar *et al*., 2013) are another contamination detection and data partitioning methodology. BlobPlots are two-dimensional scatter plots, in which sequences are represented by dots and coloured by taxonomic affiliation based on sequence similarity search results. For each sequence, the position on the Y-axis is determined by the base coverage of the sequence in the coverage library, a proxy for molarity of input DNA. The position on the X-axis is determined by the GC content, the proportion of G and C bases in the sequence, which can differ substantially between genomes.

Here, we present BlobTools, a modular command-line solution for the visualisation of genome assemblies as BlobPlots, and taxonomic interrogation for purposes of quality control. BlobTools is a complete reimplementation of the Blobology pipeline (Kumar *et al*., 2013) focussed on usability, improved taxonomic assignment of sequences based on custom user input, and support for coverage information based on multiple formats and sequencing libraries. We demonstrate the features of BlobTools using synthetic datasets, and offer guidelines for efficient adoption of BlobTools into genome assembly programmes.

Methods

Implementation

BlobTools is written in Python and consists of a main executable that allows the user to interact with the implemented modules (see Table 1). It offers a simple, modular command line interface which can easily be adapted to process multiple datasets simultaneously using GNU parallel (Tange, 2011). Inputs for BlobTools are standard file formats commonly created during the course of genome assembly projects. The primary processing in BlobTools constructs a BlobDB data structure based on user input. From this data structure, BlobTools generates easily interpretable, two-dimensional visualisations ready for publication, in conjunction with tabular output, enabling the user to partition sequences and paired-end (PE) reads contributing to them, for separate downstream processing. We present two recommended workflows, one targeted at *de novo* genome assembly projects in the absence of a reference genome (Figure 1A) and another for projects where a reference genome is available (Figure 1B).

Table 1. Tasks performed by BlobTools module.

BlobTools module	Task
`create`	Parsing of input files and creation of BlobTools (JSON) data structure, *i. e.* BlobDB
`view`	Generation of tabular output for manual inspection and subsequent partitioning of sequences in the assembly, input files for CONCOCT, and/or COV files based on a BlobDB
`plot`	Plotting of BlobPlots based on a BlobDB
`covplot`	Plotting of CovPlots based on a BlobDB and a COV file
`seqfilter`	Partitioning of sequences from a FASTA file based on a list of sequence IDs
`bamfilter`	Partitioning of paired-end reads from a BAM file based on a list of sequence IDs and their mapping behaviour
`map2cov`	Generation of a COV file (containing base and read coverage) based on a BAM/CAS file
`taxify`	Annotation of tabular sequence similarity search output (*e. g.* BLAST/Diamond output) with TaxIDs from a mapping file or generation of a BlobTools hits file based on custom user input

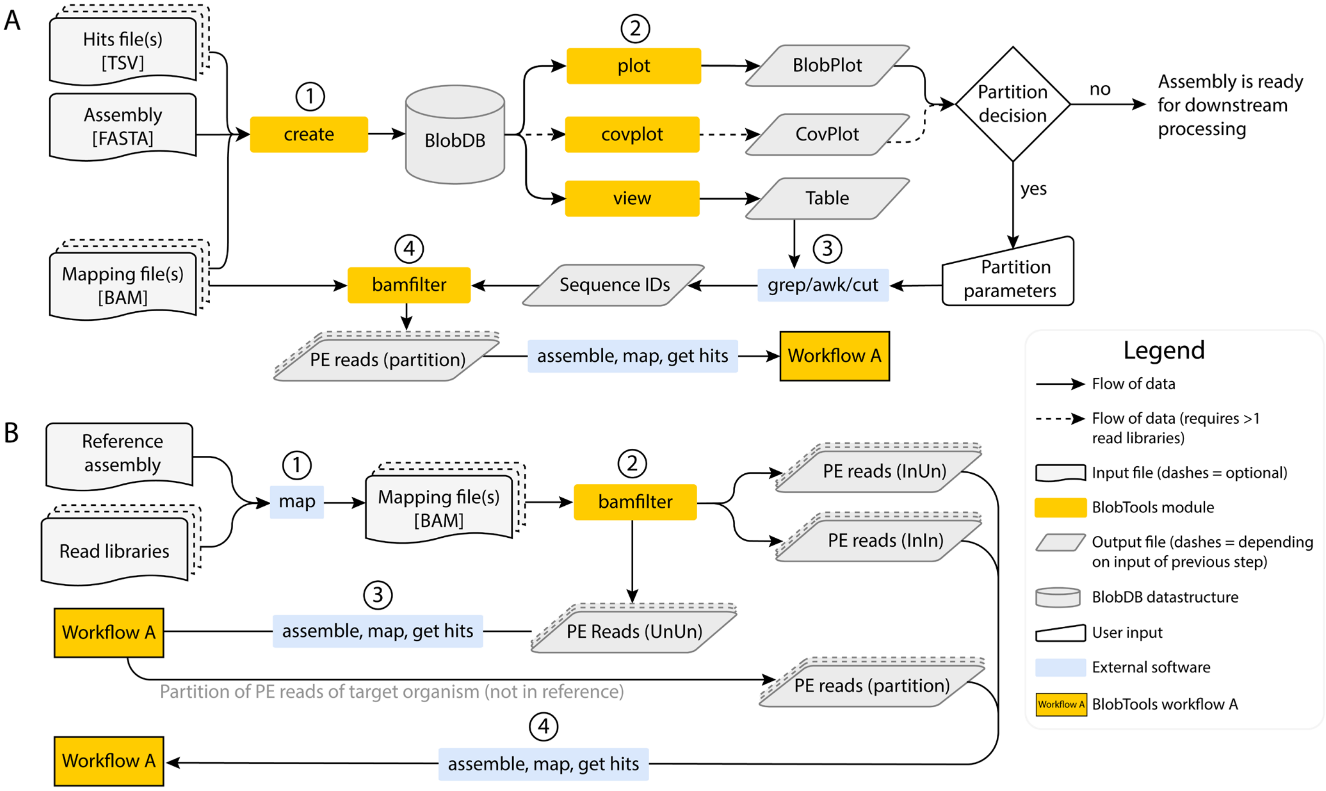

Figure 1. Two common BlobTools workflows for taxonomic interrogation of paired-end (PE) read datasets. (**A**) Workflow A. Targeted at *de novo* genome assembly projects in the absence of a reference genome. 1: Creation of a BlobDB data structure based on input files. 2: Visualisation of assembly and generation of tabular output. 3: Partitioning of sequence IDs in assembly, based on user-defined parameters informed by the visualisations. 4: Partitioning of PE reads based on sequence IDs. (**B**) Workflow B. Targeted at projects where a reference genome is available. 1: Reads are mapped against the reference genome. 2: BAM file is processed to generate FASTQ files based on read mapping behaviour. 3: FASTQ file of read pairs where neither read maps to the reference genome (UnUn) are assembled *de novo* and used in workflow A. 4: partition of read pairs of target taxon recovered from workflow A are assembled together with the other target taxon read pairs from step 2 and used in workflow A.

Taxonomy assignment

Taxonomy assignment in BlobTools is based on user-supplied, tab-separated-value (TSV) files composed of three columns: the input sequence ID, a NCBI TaxID, and a numerical score. We refer to these TSV files as 'hits' files below. They can be generated from the output of sequence similarity searches, such as BLAST (Camacho *et al.*, 2009) or Diamond blastx (Buchfink *et al.*, 2015) searches against public or reference databases, or the output of other contaminant identification tools. The BlobTools module `taxify` allows easy conversion of tabular file formats to BlobTools compatible input, in addition to annotation of similarity search results based on NCBI TaxID mapping files, as available from UniProt and NCBI.

Based on these inputs, BlobTools assigns a single NCBI taxonomy for each sequence in the assembly, based on the highest scoring NCBI TaxID at the following taxonomic ranks: species, genus, family, order, phylum, and superkingdom. Score calculation can be controlled by the user through a minimal score threshold (`--min_score`) and a minimal difference in scores (`--min_diff`) between the best and second-best scoring taxonomy. In addition, three non-canonical taxonomic annotations are possible: 'no-hit', the suffix '-undef' and 'unresolved'. Sequences not assigned to any taxonomic group, or not passing the `--min_score` threshold, are labelled 'no-hit'. If a NCBI TaxID has no explicit parent at a taxonomic rank, the suffix '-undef' is appended to the next upper taxonomic rank for which one does exist. In cases where the score difference between the best and second-best hits is smaller than `--min_diff`, sequences are labelled as 'unresolved'.

Multiple 'hits' files can be provided as input. In this case, the behaviour of the taxonomy assignment process can be controlled further through 'taxrules'. The highest scoring taxonomy can either be inferred across all files ('bestsum') or successively ('bestsumorder') in the order they were supplied as input, allowing only sequence that received no hits from one file to be considered for taxonomic annotation in the next file, thereby leveraging reliability of scores of different input file sources.

The original blobology pipeline (Kumar *et al.*, 2013) recommended the use of a single, best BLAST hit per sequence for taxonomy assignment. However, taxonomically mis-annotated sequences in databases (derived from inclusion of un-screened genome assemblies) can lead to erroneous taxonomic annotation. BlobTools mitigates this issue by accepting multiple hits per sequence and allocating taxonomy based on the highest sum of scores.

It should be noted that a definitive taxonomic placement for every sequence in the assembly is not required for successful taxonomic partitioning of sequences, since differential coverage and sequence composition profiles between the genomes are often sufficient.

Visualisations

In BlobTools, sequences are depicted as circles in BlobPlots (as opposed to dots in the blobology pipeline), with diameters proportional to sequence length. The scatter-plot is decorated with coverage and GC histograms for each taxonomic group, which are weighted by the total span (cumulative length) of sequences occupying each bin. A legend reflects the taxonomic affiliation of sequences and lists count, total span and N50 by taxonomic group. Taxonomic groups can be plotted at any taxonomic rank and colours are selected dynamically from a colour map. The number of taxonomic groups to be plotted can be controlled (`--plotgroups`, default is '7') and remaining groups are binned into the category 'others'. An example is shown in Figure 2A.

The power of differential coverage profiles across different sequencing libraries for partitioning sequences in an assembly prompted the development of CovPlots (Figure 3) (Koutsovoulos *et al.*, 2016), which are analogous to BlobPlots, except that the GC-axis is substituted by the coverage-axis from another sequencing library. CovPlots can be used for the visualisation of patterns of differential coverage signatures between taxonomic groups in the assembly.

The modules for generating BlobPlots and CovPlots support additional input parameters controlling visualisation behaviour, including cumulative addition (`--cumulative`) or generation of separate plots for each taxonomic group (`--multiplot`), exclusion (`--exclude`) or relabelling (`--relabel`) of taxonomic groups, assignment of specific HEX colours to groups (`--colour`) or labelling sequences based on arbitrary, user defined categories (`--catcolour`). The latter could be, for instance, binned categories of RNAseq mappings to sequences in the assembly as shown in Koutsovoulos *et al.* (2016).

ReadCovPlots (Figure 2B and 2C) visualise the proportion of reads of a library that are unmapped or mapped, showing the percentage of mapped reads by taxonomic group, as barcharts. These can be of use for rapid taxonomic screening of multiple sequencing libraries within a single project. The underlying data of ReadCovPlots and additional metrics are written to tabular text files for custom analyses by the user.

Support of multiple coverage libraries

BlobTools supports coverage input (BAM/CAS format) from multiple sequencing libraries. As these data formats contain more information than needed, BlobTools parses coverage information of sequences (normalised base coverage and read coverage) into COV files in TSV format. These files can be generated through the module `map2cov` prior to construction of a BlobDB.

Within the BlobDB data structure, base and read coverage information is stored for each sequence in the assembly. If more than one coverage file is supplied, BlobTools constructs an additional coverage library ('cov_sum') internally, containing the sum of coverages for each sequence across all coverage files. This internal coverage library is considered when extracting views or plotting visualisations.

Operation

System requirements for BlobTools include a `UNIX` based operating system, Python 2.7, and pip. An installation script is provided, which installs Python dependencies, downloads and processes a copy of the NCBI TaxDump, and downloads and compiles a copy of samtools (Li *et al.*, 2009). Instructions for installation and execution of BlobTools can be found at https://github.com/DRL/blobtools.

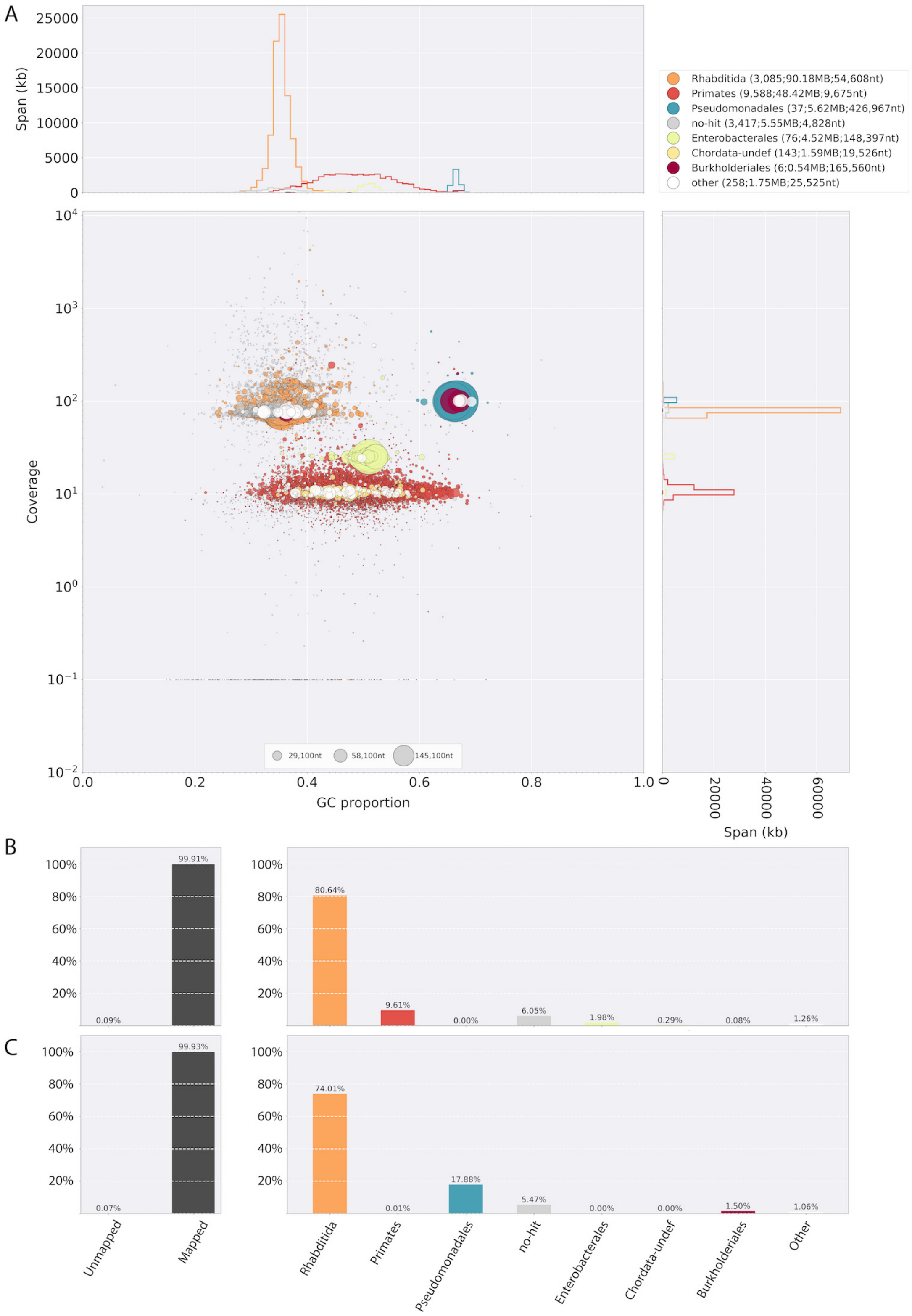

Figure 2. Visualisations of the combined assembly of simulated sequencing libraries. (**A**) BlobPlot of the assembly. Sequences in the assembly are depicted as circles, with diameter scaled proportional to sequence length and coloured by taxonomic annotation (at the rank of 'order') based on BLASTn and Diamond blastx similarity search results provided in this order and using taxrule 'bestsumorder'. Circles are positioned on the X-axis based on their GC proportion and on the Y-axis based on the sum of coverage across both library A and library B. (**B**) ReadCovPlot of library A. (**C**) ReadCovPlot of library B. In ReadCovPlots, mapped reads are shown by taxonomic group at the rank of 'order'.

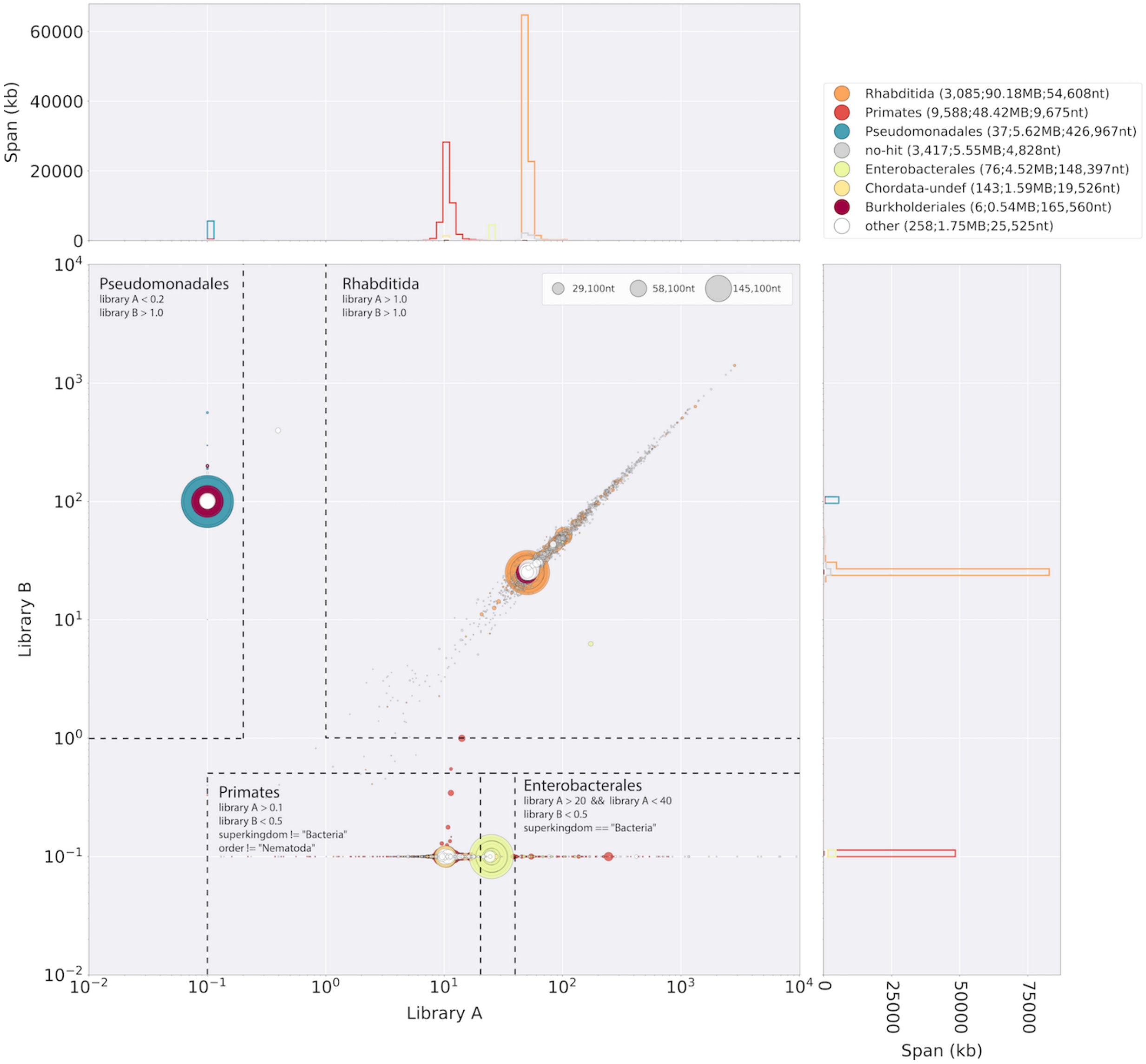

Figure 3. CovPlot of the combined assembly of simulated sequencing libraries. Sequences in the assembly are depicted as circles, with diameter scaled proportional to sequence length and coloured by taxonomic annotation (at the rank of 'order') based on BLASTn and Diamond blastx similarity search results provided in this order and using taxrule 'bestsumorder'. Circles are positioned on the X-axis based on coverage in library A and on the Y-axis based on coverage in library B. Parameters for partitioning the sequences in the assembly (which were applied to the tabular representation of the BlobDB) are indicated as dotted grey lines and text annotations in the scatter plot.

Two common BlobTools workflows for taxonomic interrogation of paired-end (PE) read datasets are depicted in the flowchart in Figure 1. Workflow A is targeted at *de novo* genome assembly projects where there is no preexisting reference genome. Workflow B should be followed where a reference genome is available.

Workflow A (Figure 1A) proceeds through construction a BlobDB data structure based on input files (step A1), visualisation of assembly and generation of tabular output (A2), partitioning of sequence IDs based on user-defined parameters informed by the visualisations (A3) and partitioning of PE reads based on sequence IDs (A4). It should be noted that while the BlobTools module `create` (step A1) supports multiple mapping formats, it is recommended that these are processed in advance using `map2cov`. Generation of tabular 'hits' files is simplified through the module `taxify`, which allows annotation of similarity search results based on TaxID mapping files or based on custom user input in tabular format.

BlobTools can process both PE and single-end read files. The module `bamfilter` in step A4 is only of relevance if PE read data is used, since single end read data can easily be partitioned using `GNU grep` or other tools. The module `bamfilter` can be controlled with a list of sequence IDs to include or to exclude. Use of an exclusion list causes all sequence IDs, except those specified, to be included. In both cases it will output up to four interleaved FASTQ files depending on the actual mapping behaviour of the read pairs and whether the parameter `--include_unmapped` is provided. Possible mapping behaviours of read pairs are: both reads mapping to included sequences (included-included: InIn), one read mapping to an included sequence and the other being unmapped (InUn), and one read mapping to an included sequence and the other mapping to an excluded sequence (ExIn). If the `--include_unmapped` parameter is specified, the module also writes read pairs where neither read maps to the assembly (UnUn). The latter case can occur if the assembler used for generating the sequences did not make use of all reads in the dataset. The resulting partitioned PE read files can then be assembled separately and the workflow is repeated. Decisions concerning which PE read files to use is left at the discretion of the user. However, as general rule, if target taxa have been sequenced at low coverages it might be preferable to be inclusive (using InIn, InEx, InUn and UnUn FASTQ files for assembly) and risking including non-target reads, than being exclusive (using only InIn and InUn for assembly) and risking losing significant proportions of reads from target genomes.

Workflow B (Figure 1B) should be applied when a reference genome is available. Reads are mapped against the reference genome (B1) and the resulting BAM file is processed with the module `bamfilter` (B2) using the parameter `--include_unmapped` and without providing a list of sequences. This will result in three FASTQ files: InIn, InUn and UnUn. Since taxonomic origin of the InIn and InUn reads has been established through the mapping step, only the UnUn reads are assembled *de novo* (B3) and processed via workflow A. This decreases computational requirements substantially. If workflow A yields a PE read partition of the target organism, which will consist of parts of the organism's genome not present in the reference, these reads are can be used together with the InIn and InUn reads from step 2 to generate a new assembly (B4), which should be screened again via Workflow A. This iterative procedure can easily be applied to projects studying highly variable species where segmental presence-absence is common and a reference genome is expanded (to form a pangenome) as new samples are sequenced, or holobiomes, where reference genomes of multiple taxa are expanded as new samples are added.

Use cases

A detailed description of the programs and commands used can be found in Supplementary File 1.

Data

To illustrate workflow A (Figure 1A), we simulated read libraries for the nematode *Caenorhabditis elegans* contaminated with other organisms (see Table 2). Library A contains *C. elegans* reads contaminated with reads from *Escherichia coli*, *Homo sapiens* chromosome 19 and *H. sapiens* mitochondrial (mtDNA) genome, mimicking a dataset where the target genome is contaminated with DNA from food (*E. coli*) and operator (*H. sapiens*). Library B is composed of *C. elegans* reads contaminated with *Pseudomonas aeruginosa*, mimicking a project where the metazoan target species is heavily colonised by a prokaryotic organism.

Taxonomic interrogation and partitioning of read pairs using BlobTools

We assembled both read datasets together and mapped each library individually against the assembly. We supplied the assembly to BlobTools, in addition to coverage information extracted from both BAM files and the results of sequence similarity searches.

To simulate cases where sequences of genomes in the assembly are not part of public sequence databases, we removed all sequences annotated under the taxonomic terms 'Caenorhabditis elegans', 'Hominids', 'Escherichia', 'Pseudomonas', and 'Other sequences' before conducting sequence similarity searches. The search results provided to BlobTools were BLASTn megablast search against NCBI nt (`-outfmt ′6 qseqid staxids bitscore std′ -max-target-seqs 1 -max_hsp 1 -evalue 1e-25`) and Diamond blastx searches against UniProt Reference Proteomes (`--outfmt 6 --sensitive --max-target-seqs 1 --evalue 1e-25`), supplied in this order and using taxrule 'bestsumorder'.

A BlobPlot (Figure 2A), ReadCovPlots (Figure 2B and C) and a CovPlot (Figure 3) were generated at the taxonomic rank of 'order'. A tabular view of the BlobDB was generated using the module `view` under the taxrule 'bestsumorder' and for the taxonomic ranks of 'superkingdom', 'phylum', and 'order'. We partitioned sequences based on differential coverage and taxonomy annotation (Figure 3) using the tabular view and the UNIX tools `GNU grep`, `GNU cut`, and `GNU awk`. Subsequently, read pairs were partitioned

Table 2. Simulated read libraries.

Dataset	Reference genome	INSDC Accession	Coverage (X)
Library A	*C. elegans* N2	GCA_000002985.3	50
	E. coli str. K-12 substr. MG1655	GCA_000801205	25
	H. sapiens chr19 GRCh38.p10	GCA_000001405.25	10
	H. sapiens mtDNA GRCh38.p10	GCA_000001405.25	250
Library B	*C. elegans* N2	GCA_000002985.3	25
	P. aeruginosa PAO1	GCA_000006765.1	100

based on mapping behaviour to these sequence partitions using the module `bamfilter` and read pairs where both reads mapped to included sequences (*i. e.* the InIn set) were assembled by taxonomic group.

We then generated BlobPlots for the four assemblies (named 'rhabditida-BT', 'primates-BT', 'pseudomonadales-BT' and 'enterobacterales-BT') (Figure 4). Coverage information was based on mapping of both simulated sequencing libraries against all four assemblies and sequences were coloured based on the genome-of-origin of the simulated reads mapping to them.

Evaluation of results

Cleaned assemblies were evaluated based on the count of simulated reads, by genome-of-origin, mapping to them (Table 3), and based on standard assembly metrics (Table 4).

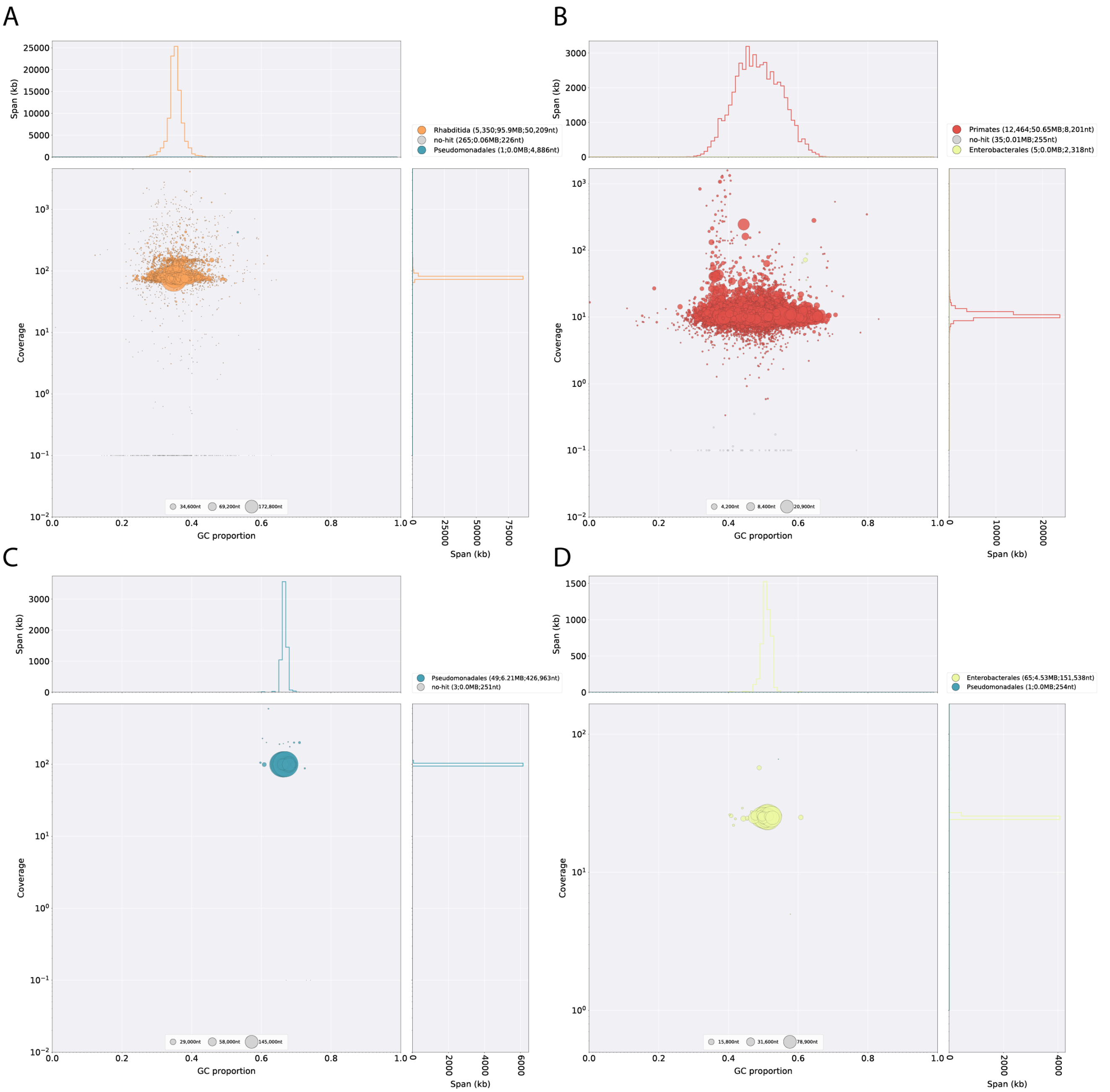

Figure 4. BlobPlots of assemblies by taxon after read partitioning using BlobTools. Coverage was obtained by mapping original reads to assemblies. Sequences are taxonomically annotated with 'true' taxonomy based on origin of simulated reads mapping to them. Sequences labelled as 'no-hit' did not receive any reads mapped to them. (**A**) Assembly of partition of Rhabditida reads ('rhabditida-BT'). One *P. aeruginosa* sequence (span 4,886 nt) remains. (**B**) Assembly of partition of Primates reads ('primates-BT'). Five *E. coli* sequences (total span 3,838 nt) remain. (**C**) Assembly of partition of Pseudomonadales reads ('pseudomonadales-BT'). (**D**) Assembly of partition of Enterobacterales reads ('enterobacterales-BT'). One sequence of *P. aeruginosa* (span 254 nt) remains.

Table 3. Percentages of reads (partitioned by taxonomic origin) mapped to sequences in each of the BlobTools-processed assemblies (suffix '-BT'). *: Reads that did not map to any sequence are listed under 'Not Mapped'. Bold: Zero reads mapped.

Taxonomic origin of simulated reads	Mapping to rhabditida-BT (%)	Mapping to primates-BT (%)	Mapping to pseudomonadales-BT (%)	Mapping to enterobacterales-BT (%)	Not mapped (%)
C. elegans	99.99	0.00	0.00	0.00	0.01
H. sapiens	0.02	99.33	0.00	0.00	0.66
P. aeruginosa	0.29	0.00	99.66	0.03	0.02
E. coli	0.72	0.22	0.06	98.64	0.35

To account for assembly and mapping biases, the original simulated read sets were also assembled separately by taxon, yielding the assemblies CELEG-SIM (reads simulated from the *C. elegans* genome), HSAPI-SIM (reads simulated from *H. sapiens* chromosome 19 and mtDNA), PAERU-SIM (reads simulated from *P. aeruginosa* genome), and ECOLI-SIM (reads simulated from *E. coli* genome).

We evaluated the effect of parameters of similarity searches against public databases on taxonomic annotation using BlobTools (see Supplementary File 2). Since exhaustive searches against large databases require time and computing power we focussed on parameters that limit resource usage and control the number of returned results. In both BLASTn and Diamond blastx, the options `-max-target-seq` and `-max-hsps` are implemented. The former is an early filter applied during primary search and excludes initial hits from later examination. The latter controls the number of high-scoring pairs (HSPs) reported between a query and a subject in the search. The BLAST specific parameter `-culling-limit` controls the number of hits that can be allocated to a given region on the query. For this dataset, the best trade-off between false positive and false negative taxonomic annotations was achieved by combining BLAST search `(-max-target-seqs 10 -evalue 1e-25)` against NCBI nt with Diamond blastx searches `(--evalue 1e-25 --max-target-seqs 1)` against UniProt Reference Proteomes, in this order, using BlobTools taxrule 'bestsumorder'. However, a much faster search with acceptable outcome was achieved by changing the BLASTn parameters to `-max-target-seqs 1 -max_hsps 1.`

Summary

We have presented the BlobTools pipeline and illustrated the main BlobTools workflow (Figure 1A) by successfully disentangling read pairs from two simulated datasets composed of metazoan and bacterial genomes. The small fraction of read pairs that received an erroneous taxonomic assignment or were left out during the partitioning step (Table 3) had little effect on the overall assembly success for each taxon (Table 4). The outcome could have been improved further by being more inclusive during the partitioning step of sequences (to decrease the number of unassigned read pairs), combined with a second round of BlobTools workflow A (to remove read pairs which were partitioned into the wrong taxonomic group).

The ease of interpretation of BlobPlots has favoured adoption by users, and the current implementation of BlobTools has been applied successfully to genome projects involving tardigrades (Koutsovoulos *et al.*, 2016; Yoshida *et al.*, 2017), mealybugs and their endosymbionts (Husnik & McCutcheon, 2016), ectoparasitic mites (Dong *et al.*, 2017), diptera (Dikow *et al.*, 2017), honeybees and their metagenomes (Gerth & Hurst, 2017), nematodes (Eves-van den Akker *et al.*, 2016; Gawryluk *et al.*, 2016; Slos *et al.*, 2017; Szitenberg *et al.*, 2017), bacteria (Fuller *et al.*, 2017; Mellbye *et al.*, 2017; Samad *et al.*, 2016; Wang & Chandler, 2016), butterflies (Nowell *et al.*, 2017), a fungal pathogen of barley (McGrann *et al.*, 2016), and fungi (Compant *et al.*, 2017).

BlobTools is a user-friendly and reliable solution for visualisation, quality control and taxonomic partitioning of genome datasets.

Table 4. **Metrics of reference genomes (suffix ‘-REF’), assemblies generated from simulated reads by taxon (suffix ‘-SIM’) and assemblies generated from reads partitioned using BlobTools pipeline (suffix ‘-BT’).**

Metric	CELEG-REF	CELEG-SIM	rhabditida-BT	HSAPI-REF	HSAPI-SIM	primates-BT	PAERU-REF	PAERU-SIM	pseudomonadales-BT	ECOLI-REF	ECOLI-SIM	enterobacterales-BT
Span (b)	100,286,401	95,970,640	95,964,660	58,634,185	50,765,888	50,660,776	6,264,404	6,221,846	6,215,193	4,636,831	4,561,104	4,534,517
count	7	5,536	5,616	2	12,700	12,504	1	58	52	1	87	66
N50 (b)	17,493,829	51,178	50,209	58,617,616	8,186	8,200	6,264,404	333,929	426,963	4,636,831	148,391	151,538
GC (%)	35.4	35.4	35.4	47.9	48.4	48.4	66.6	66.6	66.6	50.8	50.7	50.7
BUSCO (Complete, single copy in %)	97.8	92.7	92.8	3.1	1.5	1.6	98.2	98.2	98.2	99.5	99.5	99.2
BUSCO (Complete, duplicated in %)	0.6	0.4	0.4	0.1	0	0	0.2	0.4	0.4	0	0	0
BUSCO (Fragmented in %)	0.8	5.5	4.6	0.6	1	1	0	0	0	0.4	0.4	0.6
BUSCO (Missing in %)	0.8	1.4	2.2	96.2	97.5	97.4	1.6	1.4	1.4	0.1	0.1	0.2

Wider adoption of BlobTools screening by the research community will help control the influx of taxonomically mis-annotated sequences into public sequence databases and prevent inaccurate biological conclusions based on contaminated genome assemblies.

Software and data availability

BlobTools source code: https://github.com/DRL/blobtools

Archived source code as at time of publication: http://doi.org/10.5281/zenodo.833879 (Laetsch *et al.*, 2017)

License: GNU-GPL

A walk through for all analyses in this study is deposited at https://github.com/DRL/blobtools_manuscript, together with additional code and resulting output files.

Competing interests

No competing interests were disclosed.

Grant information

DRL was supported by a James Hutton Institute/Edinburgh University School of Biological Sciences fellowship. MLB was supported by a BBSRC research grant (Project reference BB/P024238/1).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgements

We thank members of the Blaxter Nematode and Neglected Genomics lab in Edinburgh for support, criticism and suggestions. We thank Georgios Koutsovoulos, Sujai Kumar, Tim Booth, and Jason Stajich for contributions to the BlobTools code base. We thank Carlos Caurcel and Sujai Kumar for comments on the manuscript. We thank Judith Risse and the team of Edinburgh Genomics for feature requests and implementing BlobTools in their quality control pipeline. We thank all GitHub users who have raised questions, issues and submitted feature requests.

Supplementary material

Supplementary File 1: Supplementary methods.

Click here to access the data.

Supplementary File 2: Supplementary results, including:

- **Table S1:** F-scores for evaluation of influence of parameters of BLASTn searches against NCBI nt and Diamond blastx searches against UniProt Reference Proteomes on taxonomic assignment by BlobTools.
- **Table S2:** Precision and recall for evaluation of influence of parameters of BLASTn searches against NCBI nt and Diamond blastx searches against UniProt Reference Proteomes on taxonomic assignment by BlobTools.
- **Table S3:** Number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) bases for evaluation of influence of parameters of BLASTn searches against NCBI nt and Diamond blastx searches against UniProt Reference Proteomes on taxonomic assignment by BlobTools.

Click here to access the data.

References

Alneberg J, Bjarnason BS, de Bruijn I, *et al.*: **Binning metagenomic contigs by coverage and composition.** *Nat Methods.* 2014; **11**(11): 1144–1146.
PubMed Abstract | Publisher Full Text

Artamonova II, Mushegian AR: **Genome sequence analysis indicates that the model eukaryote *Nematostella vectensis* harbors bacterial consorts.** *Appl Environ Microbiol.* 2013; **79**(22): 6868–6873.
PubMed Abstract | Publisher Full Text | Free Full Text

Buchfink B, Xie C, Huson DH: **Fast and sensitive protein alignment using diamond.** *Nat Methods.* 2015; **12**(1): 59–60.
PubMed Abstract | Publisher Full Text

Camacho C, Coulouris G, Avagyan V, *et al.*: **Blast+: architecture and applications.** *BMC Bioinformatics.* 2009; **10**: 421.
PubMed Abstract | Publisher Full Text | Free Full Text

Chor B, Horn D, Goldman N, *et al.*: **Genomic DNA *k*-mer spectra: models and modalities.** *Genome Biol.* 2009; **10**(10): R108.
PubMed Abstract | Publisher Full Text | Free Full Text

Compant S, Gerbore J, Antonielli L, *et al.*: **Draft Genome Sequence of the Root-Colonizing Fungus *Trichoderma harzianum* B97.** *Genome Announc.* 2017; **5**(13): pii: e00137–17.
PubMed Abstract | Publisher Full Text | Free Full Text

Dikow RB, Frandsen PB, Turcatel M, *et al.*: **Genomic and transcriptomic resources for assassin flies including the complete genome sequence of *Proctacanthus coquilletti* (Insecta: Diptera: Asilidae) and 16 representative transcriptomes.** *PeerJ.* 2017; **5**: e2951.
PubMed Abstract | Publisher Full Text | Free Full Text

Dong X, Armstrong SD, Xia D, *et al.*: **Draft genome of the honey bee**

ectoparasitic mite, *Tropilaelaps mercedesae*, is shaped by the parasitic life history. *Gigascience.* 2017; **6**(3): 1–17.
PubMed Abstract | Publisher Full Text | Free Full Text

Eren AM, Esen ÖC, Quince C, *et al.*: **Anvi'o: an advanced analysis and visualization platform for 'omics data.** *PeerJ.* 2015; **3**: e1319.
PubMed Abstract | Publisher Full Text | Free Full Text

Eves-van den Akker S, Laetsch DR, Thorpe P, *et al.*: **The genome of the yellow potato cyst nematode, *Globodera rostochiensis*, reveals insights into the basis of parasitism and virulence.** *Genome Biol.* 2016; **17**(1): 124.
PubMed Abstract | Publisher Full Text | Free Full Text

Federhen S: **The NCBI Taxonomy database.** *Nucleic Acids Res.* 2012; **40**(Database issue): D136–43.
PubMed Abstract | Publisher Full Text | Free Full Text

Fuller SL, Savory E, Weisberg AJ, *et al.*: **Isothermal amplification and lateral flow assay for detecting crown gall-causing *Agrobacterium* spp.** *Phytopathology.* 2017.
PubMed Abstract | Publisher Full Text

Gawryluk RM, Del Campo J, Okamoto N, *et al.*: **Morphological Identification and Single-Cell Genomics of Marine Diplonemids.** *Curr Biol.* 2016; **26**(22): 3053–3059.
PubMed Abstract | Publisher Full Text

Gerth M, Hurst GDD: **Short reads from honey bee (*Apis* sp.) sequencing projects reflect microbial associate diversity.** *PeerJ.* 2017; **5**: e3529.
PubMed Abstract | Publisher Full Text | Free Full Text

Goodwin S, McPherson JD, McCombie WR: **Coming of age: ten years of next-generation sequencing technologies.** *Nat Rev Genet.* 2016; **17**(6): 333–351.
PubMed Abstract | Publisher Full Text

Husnik F, McCutcheon JP: **Repeated replacement of an intrabacterial symbiont in the tripartite nested mealybug symbiosis.** *Proc Natl Acad Sci U S A.* 2016; **113**(37): E5416–24.
PubMed Abstract | Publisher Full Text | Free Full Text

Koutsovoulos G, Kumar S, Laetsch DR, *et al.*: **No evidence for extensive horizontal gene transfer in the genome of the tardigrade *Hypsibius dujardini*.** *Proc Natl Acad Sci U S A.* 2016; **113**(18): 5053–5058.
PubMed Abstract | Publisher Full Text | Free Full Text

Kumar S, Jones M, Koutsovoulos G, *et al.*: **Blobology: exploring raw genome data for contaminants, symbionts and parasites using taxon-annotated GC-coverage plots.** *Front Genet.* 2013; **4**: 237.
PubMed Abstract | Publisher Full Text | Free Full Text

Laetsch DR, Koutsovoulos G, Booth T, *et al.*: **DRL/blobtools: BlobTools v1.0.** *Zenodo.* 2017.
Data Source

Li H, Handsaker B, Wysoker A, *et al.*: **The sequence alignment/map format and samtools.** *Bioinformatics.* 2009; **25**(16): 2078–2079.
PubMed Abstract | Publisher Full Text | Free Full Text

Mallet L, Bitard-Feildel T, Cerutti F, *et al.*: **PhylOligo: a package to identify contaminant or untargeted organism sequences in genome assemblies.** *Bioinformatics.* 2017.
PubMed Abstract | Publisher Full Text

McGrann GR, Andongabo A, Sjökvist E, *et al.*: **The genome of the emerging barley pathogen *Ramularia collo-cygni*.** *BMC Genomics.* 2016; **17**: 584.
PubMed Abstract | Publisher Full Text | Free Full Text

Mellbye BL, Davis EW 2nd, Spieck E, *et al.*: **Draft Genome Sequence of *Nitrobacter vulgaris* Strain Ab$_1$, a Nitrite-Oxidizing Bacterium.** *Genome Announc.* 2017; **5**(18): pii: e00290-17.
PubMed Abstract | Publisher Full Text | Free Full Text

Nowell RW, Elsworth B, Oostra V, *et al.*: **A high-coverage draft genome of the mycalesine butterfly *Bicyclus anynana*.** *Gigascience.* 2017; **6**(7): 1–7.
PubMed Abstract | Publisher Full Text | Free Full Text

Samad A, Trognitz F, Antonielli L, *et al.*: **High-Quality Draft Genome Sequence of an Endophytic *Pseudomonas viridiflava* Strain with Herbicidal Properties against Its Host, the Weed *Lepidium draba* L.** *Genome Announc.* 2016; **4**(5): pii: e01170–16.
PubMed Abstract | Publisher Full Text | Free Full Text

Slos D, Sudhaus W, Stevens L, *et al.*: ***Caenorhabditis monodelphis* sp. n.: defining the stem morphology and genomics of the genus *caenorhabditis*.** *BMC Zool.* 2017; **2**(1): 4.
Publisher Full Text

Szitenberg A, Salazar-Jaramillo L, Blok VC, *et al.*: **Comparative genomics of apomictic root-knot nematodes: Hybridization, ploidy, and dynamic genome change.** *BioRxiv.* 2017.
Publisher Full Text .

Tange O: **Gnu parallel - the command-line power tool**. *login: The USENIX Magazine.* 2011; **36**(1): 42–47.
Reference Source

Tennessen K, Andersen E, Clingenpeel S, *et al.*: **ProDeGe: a computational protocol for fully automated decontamination of genomes.** *ISME J.* 2016; **10**(1): 269–272.
PubMed Abstract | Publisher Full Text | Free Full Text

Wang Y, Chandler C: **Candidate pathogenicity islands in the genome of '*Candidatus* rickettsiella isopodorum', an intracellular bacterium infecting terrestrial isopod crustaceans.** *PeerJ.* 2016; **4**: e2806.
PubMed Abstract | Publisher Full Text | Free Full Text

Yoshida Y, Koutsovoulos G, Laetsch DR, *et al.*: **Comparative genomics of the tardigrades hypsibius dujardini and ramazzottius varieornatus.** *BioRxiv.* 2017.
Publisher Full Text

Open Peer Review

Current Referee Status: ? ?

Version 1

Referee Report 27 September 2017

doi:10.5256/f1000research.13242.r25294

?

Richard M Leggett

Earlham Institute, Norwich, UK

This paper describes BlobTools, an open source software package for partitioning of genomic data, principally for contamination control. It is a reimplementation of the Blobology pipeline previously described by one of the authors.

The paper makes a compelling case for the usefulness of blob plots, by citing a large number of previous works that have adopted the approach. The operation of the tool and the use cases look well thought out.

The manuscript states that the software should work on a UNIX-based operating system, but I had some difficulties with Mac OS. I found I needed to install wget, but then encountered issues with the python installation and pip that I was unable to overcome. Some guidance for Mac users in the instructions would be appreciated, as these do make up a significant number of users of bioinformatics software. I was, however, able to install very easily on a Linux machine.

Though the simulated dataset examples are useful, I would have liked to see a use case involving a real dataset, showing the real impact that BlobTools had. It would also be useful if the authors could provide a brief tutorial based around a small dataset (real or simulated).

A few minor comments:
In Abstract, a typo in final paragraph “dataset,s”.
In Introduction paragraph, “The decrease in cost per nucleotide lead” should be “has led”.
A little bit the introduction paragraph feels like it was written a few years ago - ie. non-model organisms have been sequenced for many years.
Second paragraph: interrogation of genome assemblies… is an elemental step in the genome sequencing process. More a part of genome assembly than sequencing?
Second paragraph: “Several reports of HGTs… have been shown…” - provide refererences.

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Referee Report 11 August 2017

doi:10.5256/f1000research.13242.r24671

? **A. Murat Eren** [1,2]

[1] Department of Medicine, University of Chicago, Chicago, IL, USA

[2] Marine Biological Laboratory , Woods Hole, MA, USA

The study by Laetsch and Blaxter describes the workflow of BlobTools, an open source software package for the curation of low-complexity metagenomic assemblies. The work is well-written and clear, and the efficacy of the tool have already been demonstrated by many previous studies. Operational procedures and use cases laid out in the current work will likely be very useful to researchers who wish to rapidly screen their assemblies.

I have two minor suggestions. The first one is about the following sentence:

Anvi'o (Eren et al., 2015) partitions assemblies by clustering sequences based on the output of CONCOCT (Alneberg et al., 2014).

This is not quite accurate. Anvi'o *can* employ CONCOCT to automatically partition contigs into genome bins, however, it is only optional. The default mode of anvi'o uses multiple aspects of data (including the differential normalized coverage of contigs across libraries --if multiple samples are available, GC-content, and/or tetranucleotide frequencies) to generate a hierarchical clustering dendrogram that can be used for the identification of distinct genome bins.

My second suggestion is to include a citation to the study by Delmont and Eren, "*Identifying contamination with advanced visualization and analysis practices: metagenomic approaches for eukaryotic genome assemblies*"[1] as I believe it would make an appropriate addition to the introduction.

The readers could definitely benefit from an appropriate discussion of the limitations and advantages of the 2D approach BlobTools promote in contrast to other ways to do it. 2D plots are inherently limited with respect to the number of layers of data they can display. After adding coverage and GC-content as axes to organize data points on an ordination, these displays are enriched with the use of colors (i.e. for taxonomy or any other single categorical data) and dot sizes (i.e. for sequence length or any other single continuous data). Besides the simpler attributes of data, the use of anvi'o in doi:10.7717/peerj.1839[1] brings into a single interactive display many additional perspectives, including the abundance of transcripts matching to contigs, the occurrence of contigs in different sequencing libraries, and horizontally transferred genes as claimed by others, that can benefit expert investigations of assemblies. That being said, it is important to note that the visualization strategy anvi'o relies on has disadvantages: it requires the computation of a hierarchical clustering dendrogram, and the computational complexity of this step limits the number of contigs that can be processed and displayed in reasonable amount of resources to about 25,000. This creates a need for efficient and intuitive tools like BlobTools to rapidly process large metagenomic assembly datasets of low-complexity.

References

1. Delmont TO, Eren AM: Identifying contamination with advanced visualization and analysis practices: metagenomic approaches for eukaryotic genome assemblies.*PeerJ*. 2016; **4**: e1839 PubMed Abstract | Publisher Full Text

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: I am one of the authors of anvi'o.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 18 Aug 2017

Dom Laetsch, University of Edinburgh, UK

Dear Murat,

Let me first thank you for reviewing our manuscript.

We completely agree with your comments and suggestions and will:

- Expand on our description of the Anvi’o pipeline
- Add the suggested citation in the introduction
- Elaborate on the limitations of the visualisations generated by BlobTools.

We will upload the corrections as soon as possible.

All the best,

Dom

Competing Interests: No competing interests were disclosed.