Contents

1 Citation 2

2 Availability 2

3 Authors 2

4 Overview 2

5 Requirements 2

6 Installation 3

- 6.1 Biopython 3
- 6.2 Exonerate 3
- 6.3 GeneMark-ES 4
- 6.4 TransDecoder 4
- 6.5 BLAST+ 4
- 6.6 BUSCO 4
- 6.7 MUSCLE 4
- 6.8 yn00 5
- 6.9 InterProScan 5
- 6.10 GOATools 5
- 6.11 ggplot, ggrepel, UpSetR, KaryoploteR 5

7 Usage 6

- 7.1 Workflow 6
- 7.2 Command-line arguments for gene prediction 6
  - 7.2.1 `--pred`, `--pred_only`, `--no_pred` (required) 6
  - 7.2.2 `--no_exonerate` 6
  - 7.2.3 `--qc` 6
  - 7.2.4 `--busco` 6
- 7.3 Command-line arguments for pangenome construction 7
  - 7.3.1 `--no_blast` 7
  - 7.3.2 `--no_panoct` 7
  - 7.3.3 `--refine` 7
- 7.4 Command-line arguments for characterization 7
  - 7.4.1 `--ips` 7
  - 7.4.2 `--go` 7
  - 7.4.3 `--yn00` 7
- 7.5 Command-line arguments for data visualization 8
  - 7.5.1 `--plots` 8
  - 7.5.2 `--karyo` 8
  - 7.5.3 `--size` 8
  - 7.5.4 `--upset` 8

8 References 9

1 Citation

Charley G. P. McCarthy & David A. Fitzpatrick (2019). “Pangloss: a tool for pan-genome analysis of microbial eukaryotes. *Genes*, 10(7):521. Link: https://doi.org/10.3390/genes10070521

2 Availability

Source code available for download at https://github.com/chmccarthy/Pangloss.

3 Authors

Charley McCarthy wrote this document.

4 Overview

Pangloss is a Python/R/Perl pipeline for pangenomic analysis of microbial eukaryotes. Pangloss consists of three major analytic components:

- A gene and gene location prediction pipeline using Exonerate (optional), GeneMark-ES and TransDecoder.
- Construction of a syntenic pangenome using the Perl software PanOCT, followed by an optional refinement of that pangenome using reciprocal homology between clusters of syntenic orthologs.
- Functional annotation and visualization of PanOCT-derived pangenome data.

Detailed information on the processes of each of these three components can be found in the **Usage** section below.

5 Requirements

Pangloss has been tested on macOS 10.13, Ubuntu 18.04.2 LTS and CentOS 7. Pangloss requires the latest versions of Python 2 (≥2.7.10), R (≥3.5.2) and Perl 5. Individual requirements for compontent analyses are detailed below:

Dependency	Purpose	Download
Biopython	FASTA processing, etc.	https://biopython.org/
Exonerate	Gene prediction using translated reference homologs.	https://github.com/nathanweeks/exonerate
GeneMark-ES	Gene prediction using Hidden Markov Models.	http://topaz.gatech.edu/GeneMark/license_download.cgi
TransDecoder	ORF detection in non-coding regions using position-weight matrices.	https://github.com/TransDecoder/TransDecoder
yn00	Pairwise selection analysis of syntenic clusters.	http://abacus.gene.ucl.ac.uk/software/paml.html#download
MUSCLE	Alignment of genes in syntenic clusters.	https://www.drive5.com/muscle/downloads.htm
BUSCO	Gene model set completeness analysis.	https://gitlab.com/ezlab/busco

6 Installation

Pangloss is available as executable code from https://github.com/chmccarthy/Pangloss and PanOCT is included in the repository. Links to installation instructions and other useful info for the various dependencies of Pangloss (assuming Python, R and Perl are installed) are given below.

6.1 Biopython

Tested version: 1.73

Installation instructions for Biopython are available from https://biopython.org/wiki/Download. For most Linux and macOS environments, Biopython can be installed via `pip`, e.g. `pip install biopython`. Pangloss requires Biopython 1.73 (released December 2018) or later, as previous versions contain bugs in the relevant packages for handling data from Exonerate. Biopython is imported within Pangloss (e.g. `from Bio import SeqIO`, etc.).

6.2 Exonerate

Tested version: 2.4

Exonerate is no longer officially supported (it appears), but a continuation of Exonerate is hosted at https://github.com/nathanweeks/exonerate. Installation instructions from source are provided at the same address. Exonerate can also be installed from apt-get on most Linux distributions or through Homebrew on macOS via brew install brewsci/bio/exonerate. Exonerate should be available in your PATH as exonerate or specified in your config file.

6.3 GeneMark-ES

Tested version: 4.3.8

macOS and Linux versions of GeneMark-ES executables (and the licence keys necessary to run GeneMark-ES) are available at http://topaz.gatech.edu/GeneMark/license_download.cgi. See INSTALL file in GeneMark-ES folder for instructions on how to "install" licence key. GeneMark-ES requires the YAML, Hash::Merge, Logger::Simple and Parallel::ForkManager Perl modules which are all available via cpanm. GeneMark-ES should be available in your PATH as gmes_petap.pl or specified in your config file.

6.4 TransDecoder

Tested version: 5.5

TransDecoder is available as executable code from https://github.com/TransDecoder/TransDecoder. Both executable programs TransDecoder.LongOrfs and TransDecoder.Predict should be in your PATH or specified in your config file.

6.5 BLAST+

Tested version: 2.9.0

BLAST+

6.6 BUSCO

Tested version: 3.1

Installation instructions for BUSCO are available at https://gitlab.com/ezlab/busco. For completedness analysis of protein sequence data HMMER must also be installed (available from http://hmmer.org/). Note that you need to specify a separate config.ini file for BUSCO analysis (generally located in BUSCOINSTALLPATH/scripts/../config/) and you need to change the location of HMMsearch to where you have installed the HMMER suite (e.g. /usr/local/bin) in that file. run_BUSCO.py must be in your PATH or otherwise specified in the config file for Pangloss.

6.7 MUSCLE

Tested version: 3.8.31

MUSCLE binaries can be found at https://www.drive5.com/muscle/downloads.htm. MUSCLE should be in your `PATH` as `MUSCLE` or as specified in your config file.

6.8 yn00

Tested version: 4.8 (PAML)

yn00 is part of the PAML package. PAML installation instructions are available from http://abacus.gene.ucl.ac.uk/software/paml.html#download - scroll down to the section entitled "UNIX/Linux and Mac OSX“. yn00 should be in your `PATH` as `yn00` or otherwise specified in your config file.

6.9 InterProScan

Tested version: 5.34

Installation instructions for InterProScan are available at https://github.com/ebi-pf-team/interproscan/wiki/HowToDownload. `interproscan.sh` should be in your PATH. InterProScan can only run on Linux distributions, due to its use of third-party binaries.

6.10 GOATools

Tested version: 0.8.12

See https://pypi.org/project/goatools/ for installation instructions, this in turn should make `map_to_slim.py` and `find_enrichment.py` available in your `PATH`. FET analysis in GOAtools uses either the `fisher` or `Scipy.stats.fisher` Python modules - generally the former is quicker. Both should be available via `pip`.

6.11 ggplot, ggrepel, UpSetR, KaryoploteR

Tested versions: 3.2, 0.81, 1.4, 1.10.3 respectively

ggplot, ggrepel and UpSetR can all be installed from `install.packages` within R. UpSetR plots are rendered using `Cairo`, which is available via `install.packages`, although plot visualization through `Cairo` is currently disabled for Linux operating systems.

7 Usage

7.1 Workflow

7.2 Command-line arguments for gene prediction

7.2.1 `--pred`, `--pred_only`, `--no_pred` (required)

These three arguments control how gene prediction analysis is carried out within the overal Pangloss pipeline. One of these arguments is required for Pangloss to run, and each argument is mutually exclusive of the other two. `--pred` runs gene prediction as part of the overall pipleine, `--pred_only` runs only the gene prediction part of the pipeline (as well as downstream analyses if `--qc` and `--busco` are enabled) and quits once all predictions are complete, and `--no_pred` does not run gene prediction. The latter two arguments are useful when gene sequence and gene location data is already available (either from previous Pangloss runs or from other sources).

7.2.2 `--no_exonerate`

This argument disables Exonerate-based gene prediction, which speeds up the overall gene prediction process but means you may miss potential gene models not detected by either GeneMark-ES or TransDecoder.

7.2.3 `--qc`

This argument enables a custom ”quality control“ assessment of gene prediction for a dataset by searching known ”dubious“ genes or pseudogenes against your dataset using BLASTp, and removing genes that have ≥70% similarity to known dubious genes. Works best for model organisms with knowns sets of dubious genes usually available from a genomic resource website, like that from the Saccharomyces Genome Database for *Saccharomyces cerevisiae*.

7.2.4 `--busco`

This argument enables BUSCO completeness analysis of predicted gene sets for each genome in an input dataset. Installation instructions for BUSCO are available at `https://gitlab.com/ezlab/busco`. For completeness analysis of protein sequence data HMMER must also be installed (available from `http://hmmer.org/`). **Note**: you need to specify a separate config.ini file for BUSCO analysis (generally located in `$BUSCO_INSTALL_PATH/scripts/../config/`) and you need to change the location of HMMsearch to where you have installed the HMMER suite (e.g. `/usr/local/bin`) in that file.

7.3 Command-line arguments for pangenome construction

7.3.1 `--no_blast`

This argument disables the all-vs.-all BLASTp search that Pangloss by default performs for the entire pangenome dataset. This is useful when BLASTp output has already been generated for a pangenome dataset (and in most cases, all-vs.-all BLASTp data is quicker to generate yourself using a HPC environment than it is in Pangloss). Just make sure that the name and location of your BLASTp output matches what's in the configuration file.

7.3.2 `--no_panoct`

This argument disables pangenome construction from a given dataset using PanOCT. This argument is here for debugging purposes, but may be useful in cases where PanOCT data has already been produced in a previous run.

7.3.3 `--refine`

This argument enables in-house "refinement" of a pangenome constructed by PanOCT. This is based on reciprocal strain top-hit homology between all member genes of two accessory syntenic clusters. More information will be available in McCarthy & Fitzpatrick (2019b), in review.

7.4 Command-line arguments for characterization

7.4.1 `--ips`

This argument enables InterProScan analysis of a pangenome dataset by Pangloss, and will run Pfam, InterPro and Gene Ontology annotation analysis. This analysis requires InterProScan to be installed and available in your `PATH`. **Note:** InterProScan can only run on Linux distributions, due to its use of third-party binaries (see https://github.com/ebi-pf-team/interproscan/wiki for more information). Pangloss will skip over InterProScan analysis if `--ips` is passed on non-Linux operating systems.

7.4.2 `--go`

This argument enables GO-slim enrichment analysis of core and accessory genomes using GOATools. Enrichment analysis in Pangloss (*via* GOATools) uses Fisher's exact test (FET) with parent term propagation and false discovery rate correction using 500 sampled p-values ($p \geq 0.05$).

7.4.3 `--yn00`

This argument enables Yang & Nielsen (2000) selection analysis using yn00, which is part of the PAML package of phylogenetic tools. Pangloss will run yn00 on each syntenic cluster in a pangenome, and generate a summary of the number of pairwise alignments in each cluster (if any) which exhibit traits of positive selection, i.e. their d_N/d_S ratio is ≥ 1.

7.5 Command-line arguments for data visualization

7.5.1 `--plots`

This argument enables all visualization analyses described below, and is equivalent to passing `--karyo --size --upset` to Pangloss.

7.5.2 `--karyo`

This argument enables karyotype plot generation (by `Karyotype.R`) of core and accessory genome content along all genomic sequences (contigs, chromosomes, &c.) within each genome in a pangenome dataset. `--karyo` produces two karyotype plots: one in which gene locations are coloured by their parent component (core: green, accessory: red), and one in which the same locations are coloured by the number of genes in their parent syntenic cluster (red-to-green gradient with red representing singleton genes and green representing core genes). This analysis requires the R packages `KaryoploteR`, `Hmisc` and `regioneR`. **Note:** Although karyotype plots are generated for each genome in a dataset, plots for genomes that are assembled to chromosome-level or are otherwise highly contiguous are generally the most informative/aesthetically pleasing. For more information, see http://bioconductor.org/packages/release/bioc/html/karyoploteR.html.

7.5.3 `--size`

This argument enables the generation of two simple size plots: a ring chart of the proportions of core and accessory syntenic clusters in a pangenome dataset (performed by `RingChart.R`) and a bar chart of the syntenic cluster size distribution within the same dataset (performed by `BarChart.R`). The latter also estimates the "true“ number of syntenic clusters (i.e. the "true“ pangenome size) within a dataset using the Chao lower bound method. The Chao estimate for pangenome size $\hat{N}$ is given by the equation

$$\hat{N} = N + \frac{y_1^2}{2y_2}$$

where N is the number of syntenic clusters in a pangenome, y_1 is the number of singleton clusters and y_2 is the number of doubleton (2-member) clusters. `RingChart.R` requires the R packages `ggplot2` and `ggrepel`. `BarChart.R` also requires `ggplot2`. The implementation of the Chao estimation method in `BarChart.R` is based on a previous implementation in the R package `micropan`.

7.5.4 `--upset`

This argument enables the generation of an UpSet plot representing the distribution of syntenic orthologs within the accessory genome component of a pangenome dataset. This is performed by `UpSet.R`, which requires the R packages `UpSetR` and on macOS `Cairo`. An UpSet plot is a way of visualizing the intersections between sets as an alternative to Venn or Euler diagrams, which are generally limited to a certain amount of input sets. For more information on UpSet plots, see this pretty thorough explanation of the concept: https://caleydo.org/tools/upset/.

8 References

To come.