Introduction

What is batch_brb?

batch_brb is a command-line application for automated best reciprocal BLAST. Common ortholog identification methods can have organism availability limitations and difficulty identifying divergent sequences. These issues can be overcome with manual searching but this is a time consuming and tedious process making it unsuitable for moderate to larger scale analyses. batch_brb provides tools to automate the data collection process while maintaining flexibility in hit selection criteria, enabling the analysis of greater numbers of sequences.

batch_brb performs ortholog identification using best reciprocal BLAST. Sequences of interest are searched against a user created database. The top x hits per query per organism are extracted and filtered by y coverage of the query where x and y are specified by the user. Identical hits do not contribute to the hit count. These hits are searched against the organism of the original query sequences. The top x hits per query are filtered by y query coverage as above. Where the hits from the first and reverse BLAST match, the sequences are considered orthologs. batch_brb is designed to enable maximum coverage and requires user analysis of hits for the exclusion of mishits and paralogs.

Typical workflow

- Create BLAST database - batch_makeblastdb
- Create alias database - aliasdb_pipeline
- Retrieve accessions - accession_retrieve
- Ortholog identification - orthology_pipeline
- Genome walk - orthology_pipeline, merge_results
- Build phylogenetic trees - fasttree_pipeline

User support

To report issues please use the GitHub issue tracker, for general enquiries please contact ebutterfield@dundee.ac.uk

Citing batch_brb

A protocol publication describing batch_brb is currently in preparation. In the meantime please cite the GitHub repository. Please also cite the dependency publications.

Acknowledgements

This software was created by the Wellcome Centre for Anti-Infectives Research using Wellcome Trust funding.

Thank you to Tim Butterfield, Frederik Drost and Michele Tinti for comments on the code and to Ricardo Canavate del Pino and Ning Zhang for help with testing.

License

How to use batch_brb

Installation

batch_brb is available as a Bioconda package and requires Conda and Bioconda. On Linux/Unix-like operating system:

```
conda create -n batch_brb batch_brb
conda activate batch_brb
```

Change into the directory where batch_brb should be located and run:

```
batch_brb_setup
```

Dependencies

batch_brb has several dependencies, listed below. These are automatically installed during the batch_brb installation. If you use batch_brb please reference the dependencies.

- bash >= 5.0.018
- biopython >= 1.78
- blast >= 2.10.1
- fasttree >= 2.1.10
- muscle >= 3.8.1551
- numpy >= 1.19.2
- pandas >= 1.1.3
- python >= 3.8.6
- sed >= 4.8
- seqkit >= 0.13.2
- sqlite >= 3.33.0

- perl-fast

Command reference

batch_brb

This command is for the setup of batch_brb. This will create a batch_brb directory which will contain four directories: jobs, databases, templates and documentaion. jobs is the directory where jobs results will be located. databases is where organism fasta files (files provided by the user which are retrieved from repositories or sourced in-house) and the resulting databases the user generates will be located. templates contains the template files for the pipelines in both CSV and Excel formats. documentation contains the license and manual for user reference.

show

This command will show the GPL3 license.

batch_makeblastdb

```
This script will create a BLAST database from a fasta file and add the
accessions to a SQLite3 database.

USAGE: batch_makeblastdb [options]

options:
-h, --help          show brief help
-csv            csv file of parameters, required
                csv must be in format: INFILE, DB
                INFILE: Required, input fasta file
                DB: Optional, default = accession_db.db
```

This is to be used in conjunction with the 01_batch_makeblastdb_template file located in the templates folder. Templates are available as both Excel files and CSVs.

```
infile  Name of fasta file to convert to BLAST database
db      Name of SQLite3 database, OPTIONAL, default = accession_db.db
```

File must be saved as a csv, ensure no excess columns included

aliasdb_pipeline

```
This script will generate a BLAST alias database and input the details
into a SQLite3 database.

USAGE: aliasdb_pipeline [options]

options:
-h, --help    show brief help
-csv          csv file of parameters, required
              csv must be in format: DBLIST_FILE, DBTYPE, TITLE,
              OUTPUT, SQLITE3_DATABASE
              The following fields are optional, defualt values are shown,
              all other fields are required:
              SQLITE3_DB: default = accession_db.db
```

This is to be used in conjunction with the 02_make_aliasdb_template file located in the templates folder. Templates are available as both Excel files and CSVs.

```
dblist_file   Name of file containing list of databases to include
              in alias database (include file extension),
              include _database in the database names within the text file
              (dblist_file)
dbtype        prot or nucl depending on protein or nucleotide respectively
title         Descriptive title for the database - this is not the
              new database name
output        Name for the alias database
SQLite3_db    Name of SQLite3 database, OPTIONAL,
              default = accession_db.db
```

File must be saved as a csv, ensure no excess columns included

accession_retrieve

```
This script will retrieve matching accessions from the SQLite3 database,
if they are not found BLAST is performed to enable accession
identification by user.

USAGE: accession_retrieve [options]

options:
-h, --help          show brief help
-csv            csv file of parameters, required
                csv must be in format: FASTA_FILE, JOB_NAME,
                BLAST_DATABASE_NAME, SQLITE3_DATABASE, EVALUE,
                MAX, NUM_THREADS
                The following fields are optional, defualt values
                are shown, all other fields are required:
                SQLITE3_DB: SQLite3 database, default = accession_db.db
                EVALUE: Expect value for BLAST, default = 0.1
                MAX: int, maximum numnber of sequences to
                retrieve in BLAST, default = 5
                NUM_THREADS: int, number of threads to use, default = 4
```

This is to be used in conjunction with the 03_accession_retrieve_template file located in the templates folder. Templates are available as both Excel files and CSVs.

```
Fasta_file            Name of the fasta file (include extension)
job_name              Name for the job
BLAST_database_name   Name of the organism database to retrieve sequences
                      from, do not include _database in the name, it will
                      be added automatically
SQLite3_db            Name of SQLite3 database, OPTIONAL,
                      default = accession_db.db
Evalue                Expect value, OPTIONAL, default = 0.1
max                   Maximum number of sequences to retrieve from
                      BLAST, OPTIONAL, default = 5
num_threads           Number of cores for BLAST, OPTIONAL, default = 4
```

File must be saved as a csv, ensure no excess columns included

orthology_pipeline

```
This script will calculate putative orthologs using best reciprocal BLAST
with the option to generate phylogenetic trees using FastTree.

USAGE: orthology_pipeline -csv [options]

options:
-h, --help          show brief help
-csv            csv file of parameters, required
                csv must be in format: JOB_NAME, ACCESSION_LIST,
                FB_DATABASE, RB_DATABASE, EVALUE, HITS, COVERAGE,
                SQLITE3_DB, NUM_THREADS, MAX, TREE, FREQUENCY, MODEL
                The following fields are optional, defualt values are
                shown, all other fields are required:
                SQLITE3_DB: SQLite3 database, default = accession_db.db
                EVALUE: expect value, default = 0.1
                NUM_THREADS: int, number of threads to use, default = 4
                MAX: int, maximum number of sequences to retrieve in
                BLAST, default = 150
                TREE: boolean (y/n), perform phylogenetic analysis,
                default = n
                FREQUENCY: float, frequency of gaps allowed per residue,
                if TREE selected default = 0.25
                MODEL: model to use for phylogenetic analysis,
                OPTIONS lg or wag for protein or
                gtr for nucleotide, if TREE selected default = JTT
                for protein and JC for nucleotide
```

This is to be used in conjunction with the 04_orthology_pipeline_form_template file located in the templates folder. Templates are available as both Excel files and CSVs.

```
Job_name          Name for the job
Accession_list    Name of the accession list file (include extension).
                  Accessions must be those retrieved with the
                  accession_retrieve pipeline
FB_database       Name of the first BLAST database, include _database in
                  the name if not an alias database,
                  do not include extension for either database type
RB_database       Name of the reverse BLAST database, MUST be a
                  single organism
                  database, do not include _database in the name
Evalue            Expect value, OPTIONAL, default = 0.1
Hits              Number of hits for orthology calculation
Coverage          Percentage coverage of query for orthology calculation
SQLite3_db        Name of SQLite3 database, OPTIONAL,
                  default = accession_db.db
Num_threads       Number of threads, OPTIONAL, default = 4
Max               Maximum number of hits to retrieve for BLAST,
                  OPTIONAL, default = 150
Tree (y/n)        y or n, include fasttree_pipeline, OPTIONAL, default = n
Frequency         float, frequency of gaps allowed per residue,
                  if TREE selected default = 0.25
Model             model to use for phylogenetic analysis, OPTIONS lg
                  or wag for protein or gtr for nucleotide,
                  if TREE selected default = JTT for protein and JC
                  for nucleotide
```

File must be saved as a csv, ensure no excess columns included

merge_results

```
usage: merge_results [-h] in1 in2 outfile

Combine csv results produced by the BLAST pipeline

positional arguments:
  in1         First csv file
  in2         Second csv file
  outfile     Name of output

optional arguments:
  -h, --help  show this help message and exit
```

fasttree_pipeline

```
This script will generate FastTree phylogenetic trees from input fasta or
accession lists specified in text files or orthology results in a CSV.

USAGE: fasttree_pipeline [options]

options:
-h, --help                show brief help
-db, --database         BLAST database to retrieve sequences from,
                      required for text or csv files
-f, --frequency           OPTIONAL, frequency of gaps allowed per residue,
                      default is 0.25
-csv                  OPTIONAL, CSV of ortholog results, the first column
                      must be query accessions with the heading Accession,
                      remaining columns must be the
                      results with one column per organism,
                      the first row should be organism names
-m                    OPTIONAL, model for phylogenetic analysis
                      (choice of lg or wag for protein or gtr
                      for nucleotide), default if not supplied is
                      JTT for protein and JC for nucleotide
```

delete_db

```
This script will delete a BLAST database and remove the corresponding
information from the SQLite3 database.

USAGE: delete_db [options]

options:
-h, --help          show brief help
-csv            csv file of parameters, required
                csv must be in format: BLAST_DB, SQLITE3_DB
                BLAST_DB: Required, name of BLAST database to delete
                SQLITE3_DB: Optional, SQLite3 database where data is
                stored, default = accession_db.db
```

This is to be used in conjunction with the delete_database_template file located in the templates folder. Templates are available as both Excel files and CSVs.

```
BLAST_db    REQUIRED, Name of BLAST database to delete,
            do not include _database in the name, do not include extension
SQLite3_DB  Name of SQLite3 database, OPTIONAL, default = accession_db.db
```

File must be saved as a csv, ensure no excess columns included