iTaxoTools - tools for integrative taxonomy

Addendum to iTaxoTools manual: BlasTax

Version: August 2025

Previous executables of iTaxotools 0.1 are available at
https://github.com/iTaxoTools/iTaxoTools-Executables/releases

Both the previous and new tools can be downloaded from the project's website http://itaxotools.org which also features a section with useful links, news, and FAQs.

How to cite: When using one of the programs included in the iTaxoTools 0.1.1. release in your study, please cite the main paper as follows.

Vences, M., Miralles, A., Brouillet, S., Ducasse, J., Fedosov, A., Kharchev, V., Kumari, S, Patmanidis, S., Puillandre, N., Scherz, M. D., Kostadinov, I., Renner, S. S. (2021). iTaxoTools 0.1: Kickstarting a specimen-based software toolkit for taxonomists. *Megataxa* **6**: 77-92.

For MAFFT, CutAdapt, BLAST and BLAST+, cite also the original papers:

Katoh, K., Standley, D.M. (2013). MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Molecular Biology and Evolution 30: 772–780.

Martin, M. (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal, 1, 10–12.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. Journal of Molecular Biology, 215(3), 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., & Madden, T. L. (2009). BLAST+: Architecture and applications. BMC Bioinformatics, 10, 421. https://doi.org/10.1186/1471-2105-10-421

We welcome all suggestions, comments, questions and bug reports. Please use the GitHub platform to submit those: for this, you just need to sign up at GitHub (a quick and easy procedure), navigate to the respective repository (see table with links below), and create "New Issue". The team of developers regularly checks all the issues and will deal with them asap (bug reports will be treated as priority).

Example files provided:

nt_seqs_for_database.fas
aa_seqs_for_database.fas
nt_query_seqs.fas
aa_query_seqs.fas
nt_seqs_for_database_longnames.fas

museoscript_reference_database.fas
museoscript_query.FASTQ

blastappend_database.fas
blastappend_query.fas

codons_1stposstart.fas
codons_variableposstart.fas
codons_withstop.fas

All example files are nucleotide or protein sequences. They can be inspected in a text editor.

A comparatively large number of example files is provided to account for the diverse modes and functions of BlasTax. The majority of the example files contain nucleotide sequences, except for the two preceded with “aa” (amino acid = protein sequences). Sequences are in FASTA or FASTQ formats. They can be inspected in a text editor. To test BlasTax, use the appropriate example files for the respective program mode:

1. To make a nucleotide BLAST database, use nt_seqs_for_database.fas. Then perform a simple “blastn” search with nt_query_seqs.fas. For making a protein BLAST database and run a blastp search, use aa_seqs_for_database.fas and aa_query_seqs.fas, respectively.

2. To explore the problems and error messages that can arise when trying to make a database from an unsuitable FASTA file, try nt_seqs_for_database_longnames.fas. This file can be preprocessed in the program mode “FastPrepare” and afterwards, a database successfully made.

3. Use the files museoscript_reference_database.fas and museoscript_query.FASTQ to first make a BLAST database from the reference sequences (16S sequences of selected frogs) and then use the query file of raw FASTQ reads from an Illumina shotgun sequencing to retrieve matching reads.

4. Use blastappend_database.fas (a transcriptome assembly derived from Trinity) against blastappend_query.fas (an alignment of a protein-coding gene in related species of amphibians) to retrieve and add the matching sequence from the assembly.

5. To test protein translation and sequence trimming, try the example files with“codons” filename. The file codons_1stposstart.fas can be used to test regular protein translation and codon-aware alignment. Use codons_variableposstart.fas for the CodonTrim mode, which will adjust the sequences so that all sequences start with a first codon position. Use codons_withstop.fas to test removal of sequences or sequence stretches with stopcodons. You can also use the transcriptome asembly blastappend_database.fas under the “transcript” option to retrieve the longest open reading frame from each sequence.

The code of BLAST as well as command-line executables of the program are distributed open source on the servers of NCBI (https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/). For ease, BlasTax wraps several of the compiled executables and makes most of their options available via a dedicated GUI. Different from other implementations (see Table 1 of accompanying paper), BlasTax is a single executable and does not require previous installations of the original BLAST programs.

Upon starting the BlasTax executable, users are presented with a graphical user interface (GUI) similar to that of the tool TaxI2, also developed in the framework of iTaxoTools (see next page). The different program modes are accessible via a series of tiles on the starting window. By clicking the tiles, the respective options for this program mode appear. The upper symbols allow to “Open” input files, “Run” the program, and “Save” the results (where appropriate). The “Home” symbol allows to return to the starting window.

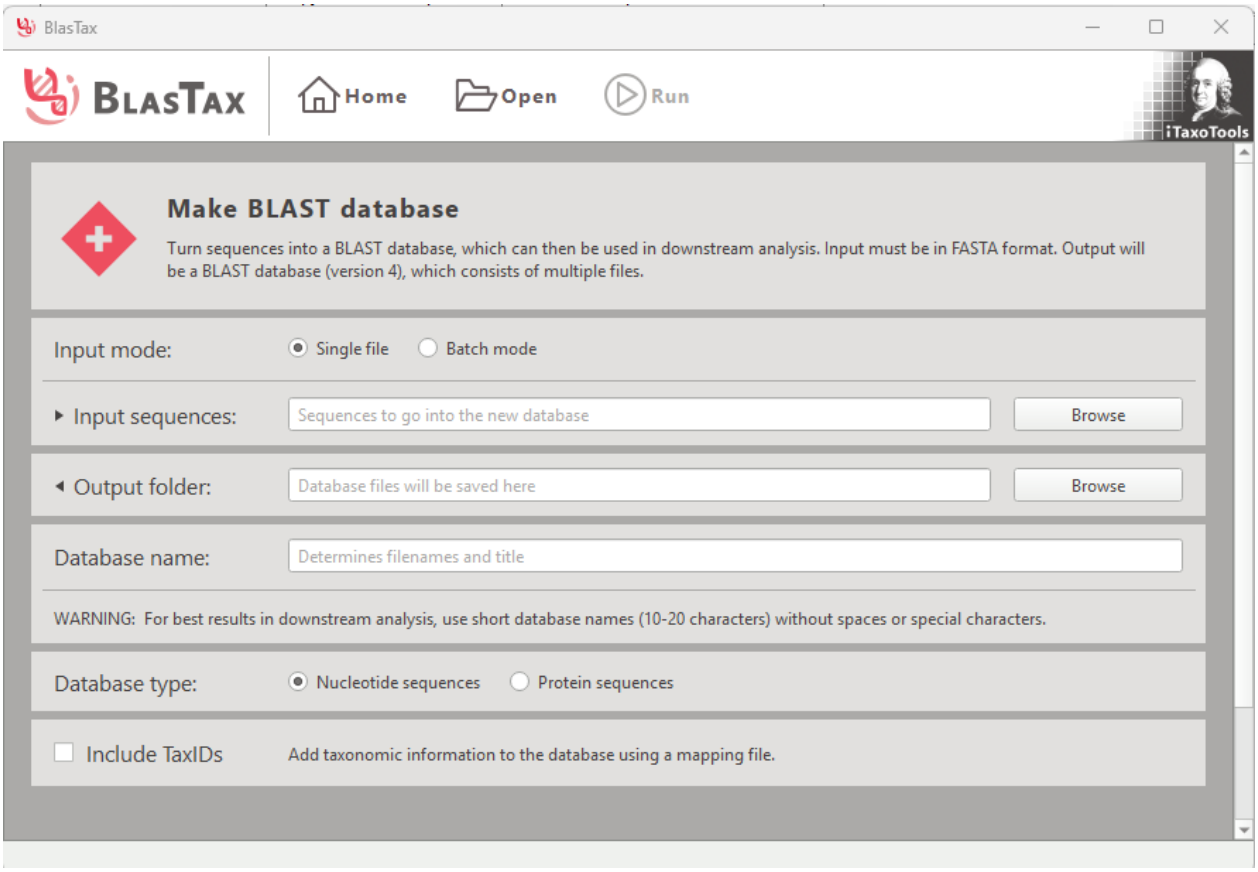

BlasTax
BlasTax
Home
Open
Run
iTaxoTools
› BLAST tools
Make BLAST database
Created from FASTA sequences
Regular BLAST
Find matching sequences
BLAST-Append
Append matching sequences
BLAST-Append-X
Append matching nucleotides
Decontaminate
Remove outlier sequences
Decontamination by taxonomy
Filter sequences by taxID
Assign taxonomy
Identify taxID and organism
Museoscript
Save matches as FASTA
Database operations
Export FASTA and taxID maps
Download NCBI databases
Get the latest taxDump and taxDB
› FASTA preparation
Fast prepare
Rename sequence identifiers
Fast split
Split sequence files
Fast merge
Merge FASTA files into one
Group merge
Merge FASTA files by filename
Removal of stop codons
Remove stop codons from a dataset
Codon trimming
Trim coding sequences to start codon
› Extras
SCaFoSpy
Create chimerical sequences for species
Protein translator
Translate nucleotides into proteins
MAFFT alignment
Multiple sequence alignment
Codon-aware alignment
Align via proteins without altering codons
Cutadapt
Remove adapters, quality trimming
About BlasTax
Information and citations

Basic BLAST-related program modes

► Make BLAST database. This wraps to the original *makeblastdb* executable. The user selects an input file in FASTA format and a name and output folder for the database, and chooses if the data are in nucleotide (DNA) or protein (amino acids) format. By pressing "Run" the database is created and the various files making up the database are written into the selected folder.

Warning 1: There are some limitations in the format of the FASTA files usable for a BLAST database. All sequence identifiers or headers (= sequence names) in the FASTA file must have less than 50 characters, and ideally consist of only standard alphanumeric characters. To make a FASTA file with long sequence identifiers or special characters in the sequence identifiers usable for making a BLAST database, use the program mode "FastPrepare" in BlasTax.
BlasTax also supports entering taxonomic identity of sequences based on a taxID mapping. file (https://www.ncbi.nlm.nih.gov/books/NBK569841/) which contains a list of sequence names (corresponding to those in the FASTA file) tabulator-separated from the respective taxID.

Warning 2: BLAST+ requires folder and file names (paths) which are not overly complex and do not contain special characters (including spaces) - only standard ASCII alphanumerical characters and underscores are allowed. Ideally, make sure to choose a folder for the new database which has a relatively short folder path without any special character in the folder/subfolder names. BlasTax can usually deal with wrongly formed paths by moving all files into a temporary folder, but with very large datasets, this may slow down the program.



► Run regular BLAST. This wraps to the original executables *blastn*, *blastp*, *tblastn*, *blastx*, *tblastx*. The user selects a query FASTA file, a BLAST database (among the various files, it searches for ".nin" or ".pin" files by choosing one of those, the respective database is selected), an output folder, and the respective BLAST algorithm:

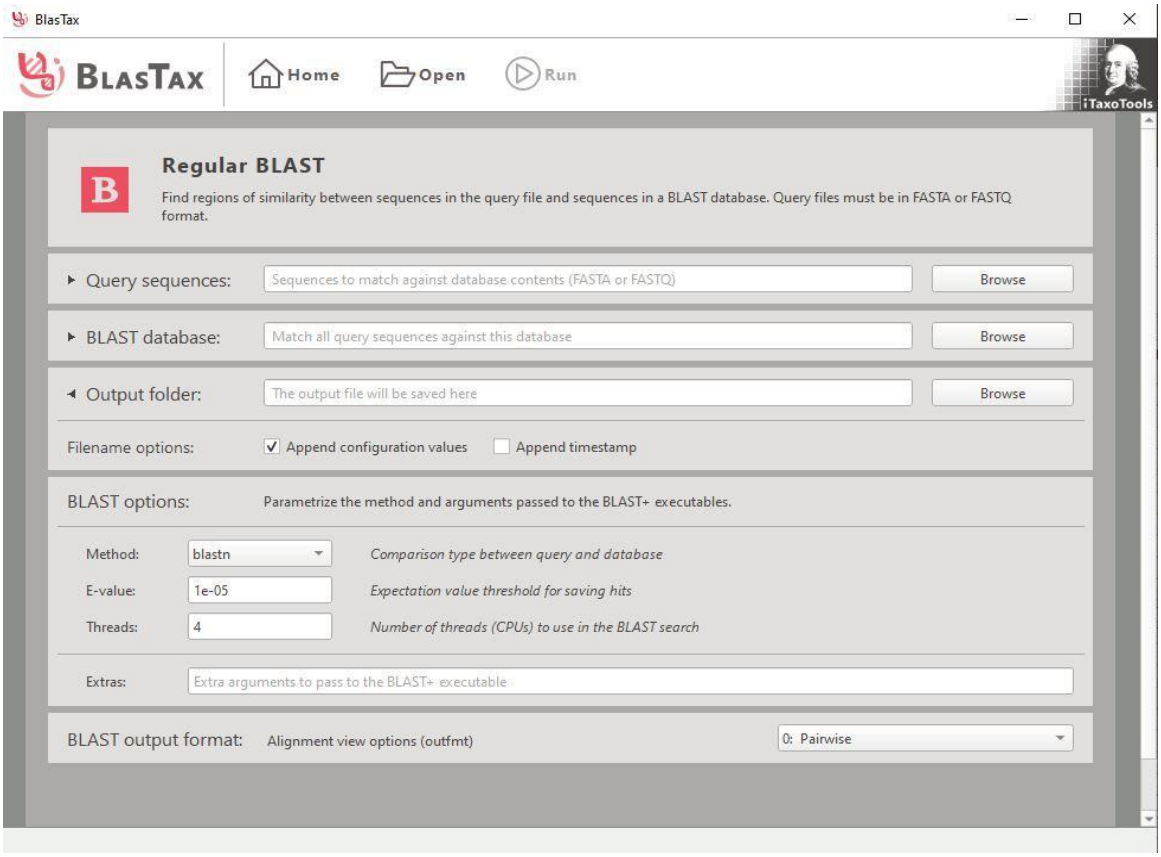
- Standard nucleotide-nucleotide BLAST (*blastn*) can be used to search for sequences in nucleotide sequence databases using nucleotide sequences as query, based on finding local regions of similarity.
- Standard protein-protein BLAST (*blastp*) can be used to search for sequences in amino acid sequence databases using amino acid (protein) sequences as query, based on finding local regions of similarity.

.

- Translated query vs protein database BLAST (*blastx*) searches sequences by comparing translated nucleotide (amino acid) query sequence (in all six reading frames) to a protein database.
- Protein query vs translated database BLAST (*tblastn*) searches sequences by comparing a protein sequence to the six-frame translations of sequences from a nucleotide database.
- Translated query vs translated database BLAST (*tblastx*) searches sequences by taking a nucleotide query sequence, translating it in all six frames, and comparing those translations to the six-frame translations of sequences from a nucleotide database.

After choosing the desired BLAST algorithm, several parameters for the BLAST search can be set, and the output format selected. Refer to the NCBI website for a detailed explanation of all parameters and output formats.

Press *Run* to execute the BLAST search.



Advanced program modes based on the BLAST algorithm

Besides making the general BLAST algorithms accessible for quick searches with a one-click executable with a user-friendly GUI, a second main goal of BlasTax is to leverage BLAST for a series of more specialized and complex analytical procedures. In general, these program modes are useful for purposes of phylogenomics and phylotranscriptomics, especially when analysing sequences of closely related species in a taxonomic context.

► MuseoScript is based on the MuseoScript code written by L. Rancilhac as a primarily Bash script (https://github.com/rancilhac/Museoscript), and published in Rancilhac et al. (2020). The implementation here is slightly different to "Regular BLAST" but serves the same purpose, that is, searching large raw sequence files from high-throughput shotgun sequencing (typically FASTQ files from Illumina sequencing) for reads that are matching a reference database.
In museomics –that is, the sequencing of archival DNA of historical specimens, often types, from museum collections– PCR-based approaches are often not applicable because the DNA is severely degraded and only consists of very small fragments. These however can be sequenced with high-throughput sequencing approaches, either directly from a shotgun library, or after target-enrichment based on taxon-specific baits. In either case, the DNA fragments corresponding to the target specimen and target genes are usually rare, sometimes only a few dozen among millions of reads. Such incomplete data are often sufficient, especially when the goal of the sequencing is not to obtain a sequence from a fundamentally unknown organism, but to assign a type to one of several genetic lineages identified from fresh material, i.e., by matching the obtained reads with the sequences of these lineages and identifying diagnostic positions.
For this purpose, the Museoscript algorithm uses BLAST to compare all reads from a FASTQ file with a database of reference sequences and writes all matches into a new FASTA file.

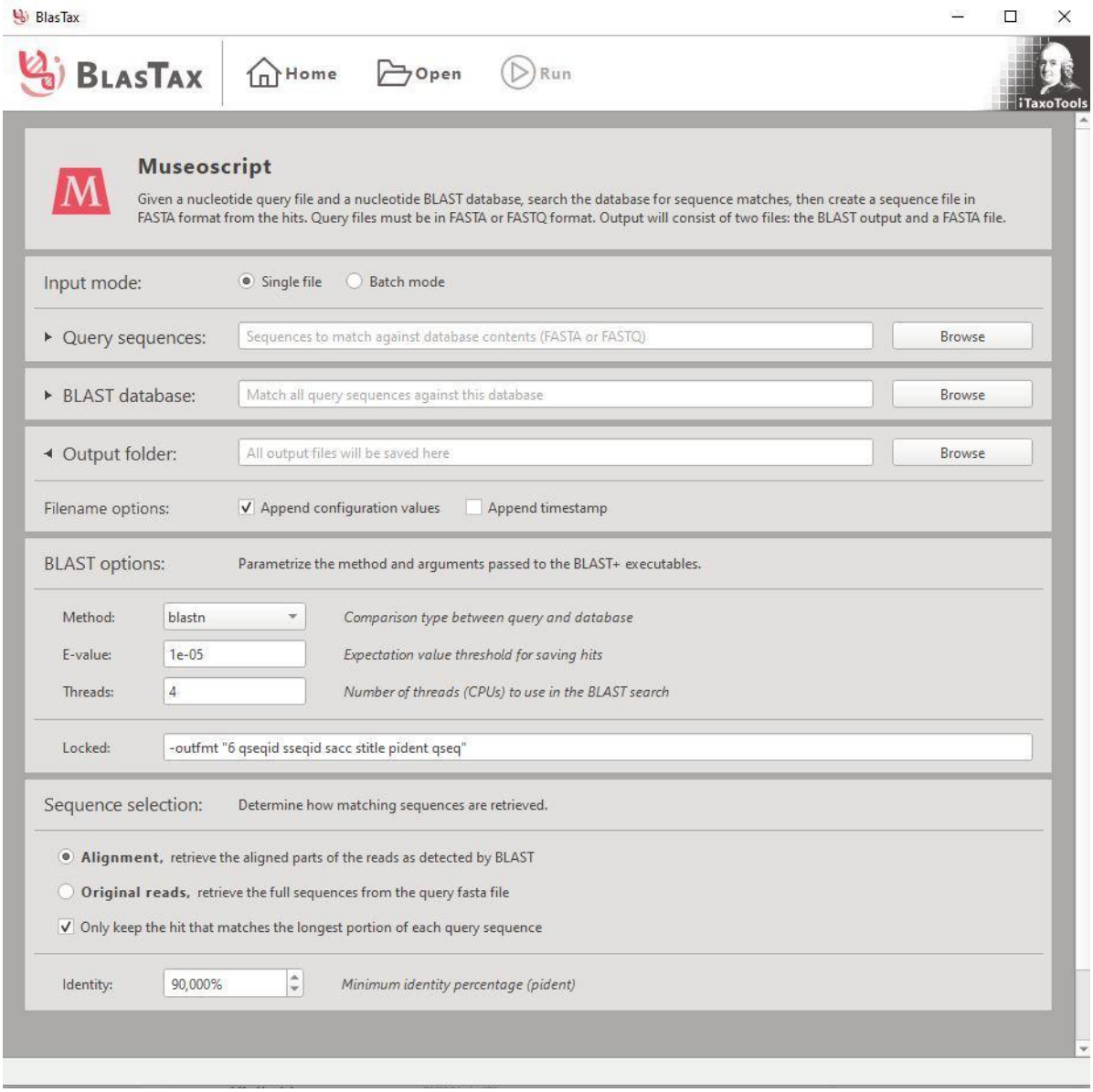
To run "Museoscript", first define as query sequence the FASTQ or FASTA file containing the archival DNA sequence reads. Then, define a BLAST database which should contain reference sequences of all genetic lineages to which the sequenced individual may belong, and the output folder where the output files should be saved. Then, specify the BLAST parameters and additionally, a percentage of sequence identity threshold (default: 90%) defining which matches should be retained and written into the output FASTA file.

This program mode is fixed for nucleotide-nucleotide searches (*blastn*) and produces two output files:
(i) a FASTA file with all matching reads, optionally with details on the matching reference and percent of sequence identity added after the sequence identifier.
(ii) a BLAST output file in tab-delimited text format, specifying the details of all matches. This tab is produced following the output specifications -outfmt "6 qseqid sseqid sacc stitle pident qseq"

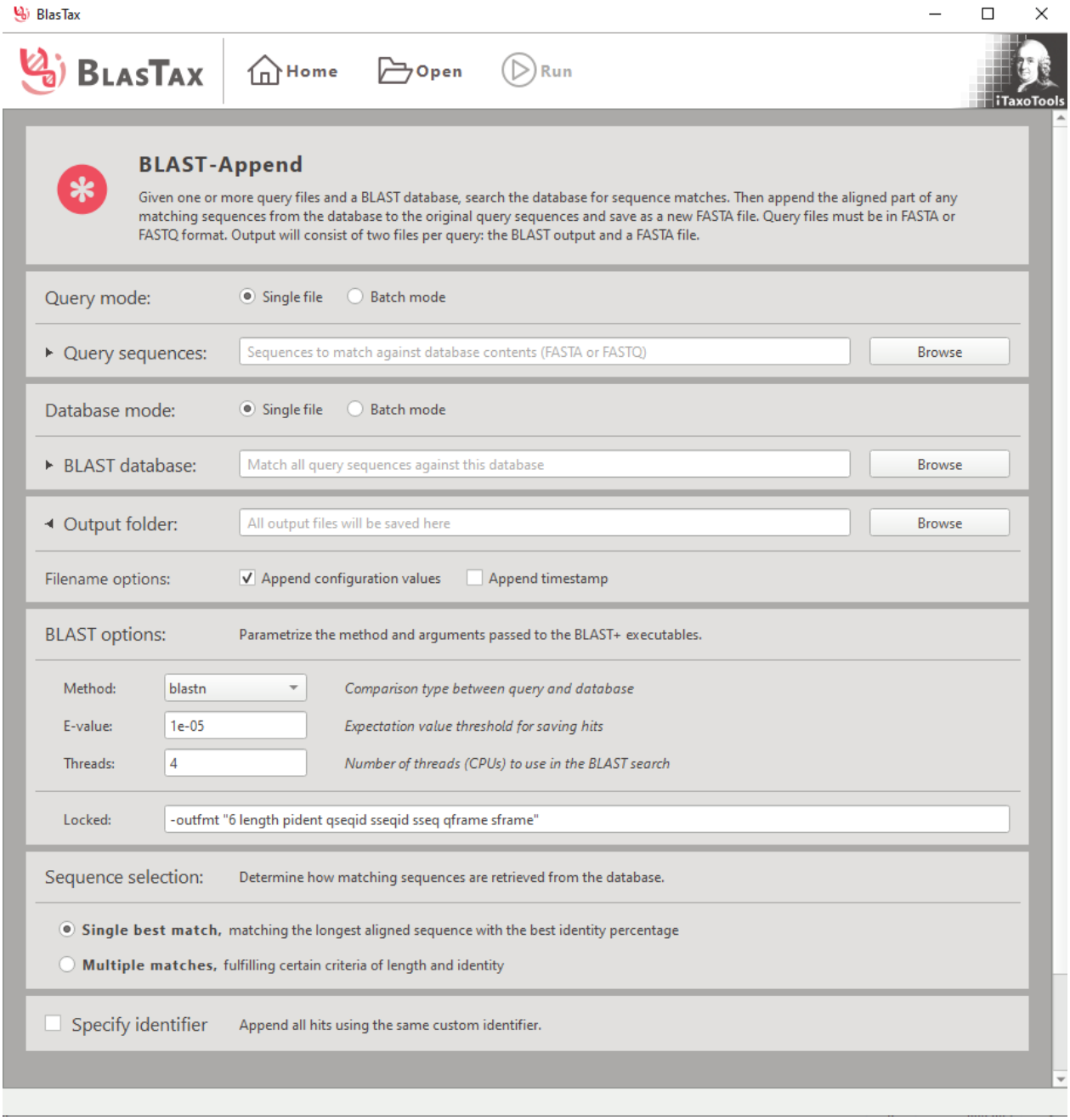
As a default, matches are deduplicated, that is, if various query sequences yielded matches of the same read in the database, only the longest of them is kept. It is possible to deactivate this option in a checkbox and add such multiple copies of the same read to the output file.

The output FASTA file should then be aligned or mapped to a reference, and it is good practice to attempt assembly against sequences from different related taxa and compare the obtained contigs in order to avoid reference bias.

Note that depending on the composition of the reference database, it is possible that the same reads can be found multiple times (e.g., retrieved by BLAST as matching against various reference sequences) and may then be written multiple times into the output FASTA file.
If you are uncertain whether the query file may contain contaminations, it can also be useful to include in the original reference database some sequences from unrelated organisms or from taxa that can be suspected to cause contamination (e.g., those that were processed in the same sequencing run). Inspecting the BLAST output file will then immediately give a rough overview which reference yielded the largest amounts of matches, and thus an indication how much the query sequences are affected by contamination.



► BlastAppend is tailored to complement phylogenomic alignments of long stretches of genes (usually protein-coding) for relatively similar (e.g., congeneric) organisms, by adding the homolog sequences for one or several new taxa. Batch mode is available for both query and reference, i.e., multiple FASTA files can be specified and processed consecutively, and the program can run consecutively on multiple databases.



As an example, imagine a set of 100 FASTA files containing ortholog sequences for five species of the genus *Salamandra*, to which the homologs of another two species of *Salamandra* (from a transcriptome assembly or an annotated genome) should be quickly added.
Strictly speaking, the sequences added will be homologs, but because the original alignments already consist of genes that were curated from potential paralogs or contaminations contaminants (and did not contain paralogs in this genus), it is not highly likely that paralogs of

these genes will exist in the genomes of additional species, if these are very closely related. Users may thus decide (at least for exploratory analyses) to add the best-matching sequences of the new species by simple BLAST searches. BlastAppend can run such a simple analysis as follows:

(i) make a BLAST database for each transcriptome assembly or ORFs from annotated genomes (CDS file) or other FASTA file containing the sequences to be added to the alignments.

(ii) specify in BlastAppend the BLAST database(s) to be used as source for the sequences of the additional taxa.

(iii) specify the query file(s), i.e., the FASTA alignments (aligned or unaligned) to which the new sequences should be added.

(iv) specify the output folder for the new FASTA files (with added BLAST matches)

(v) choose the BLAST algorithm (*blastn* for nucleotide sequences, *blastp* for protein sequences, or *tblastx* for translated query sequences, and adjust the BLAST parameters if required; usually, the default values will suffice). As default, the program runs a standard *blastn* search with a tabular output defined by the command -outfmt "6 length pident qseqid sseqid sseq qframe sframe"

(vi) make sure to select the option “single best matches” with the respective radio button (default option).

(vii) if desired, it is possible to define a sequence identifier (e.g., “Salamandra_mysteriosa”) which will be used in all FASTA files as identifier for the new sequence added (warning: this option only makes sense if a single database is used, not in database-batch mode).

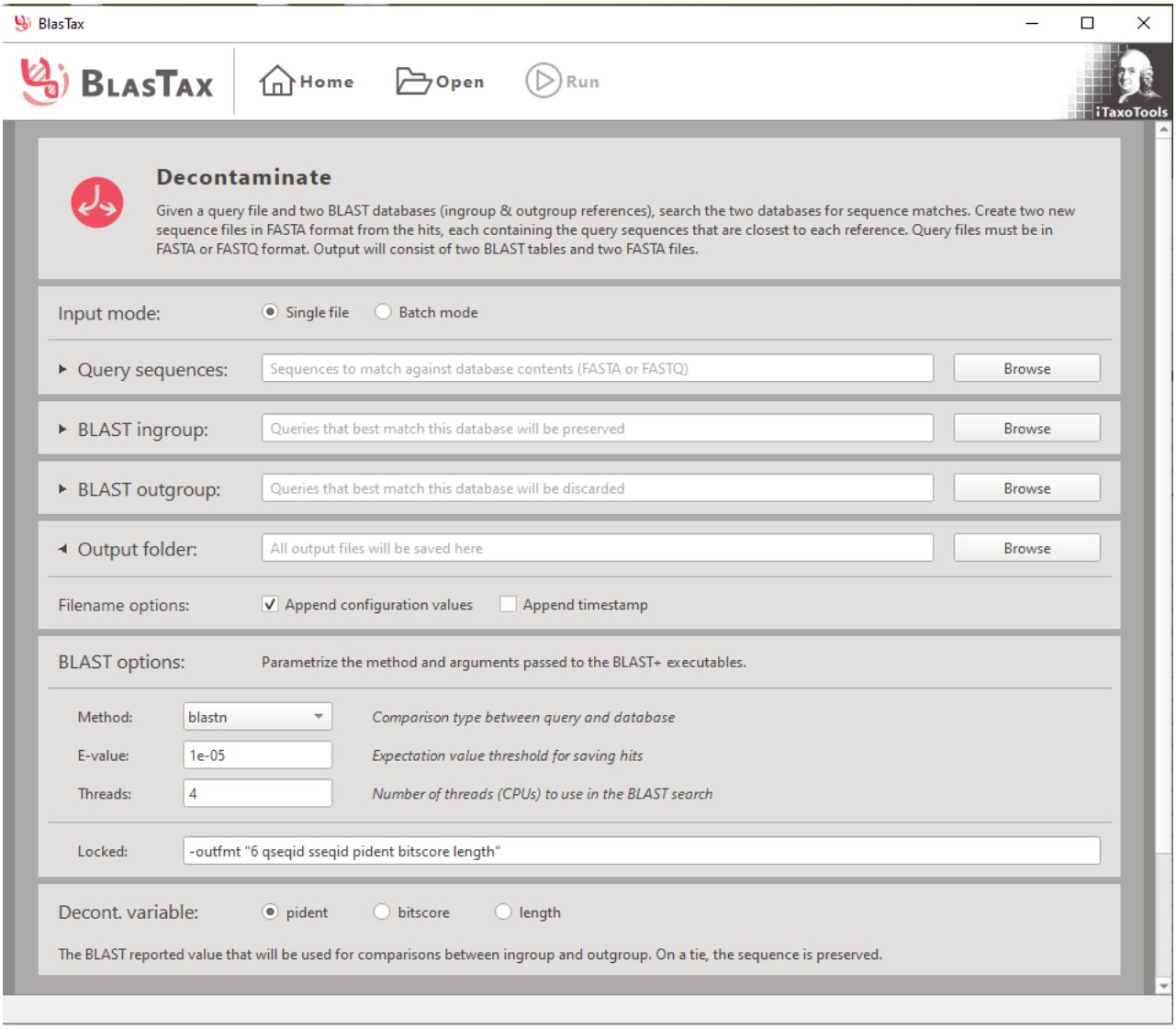
After executing, the program will iterate through all FASTA files (and with the database-batch mode, also through all specified databases), and write the FASTA files with the appended sequences in the specified folder. Note that only one matching sequence per database and FASTA file will be added (the best match to any of the sequences in the query FASTA file, and if several best matches are found, the longest of them with the highest identity percentage). The added sequences will be unaligned and may start at any position of a codon as only the matching part of the sequence is parsed. You can run “CodonTrimmer” and “Codon aware alignment” tools to align all sequences in the new FASTA files accounting for codon integrity.

“BlastAppend” also includes the option to add multiple matches by selecting the respective radio button in the GUI. In this case, if various matches are found in the database, they will all be added to the alignment, sometimes (e.g., in incomplete *de novo* transcriptome assemblies) these could be different non-contiguous fragments of the respective gene. To merge these stretches, discard poorly matching stretches, or decide in the case of overlapping and non-identical stretches; it is also possible to use “SCaFoS-Py” to merge overlapping sequences of the same taxon (see description below). An intermediate phylogenetic analysis between “BlastAppend” and “SCaFoS-Py” can be used to identify and remove possible paralogs among the newly added sequences.

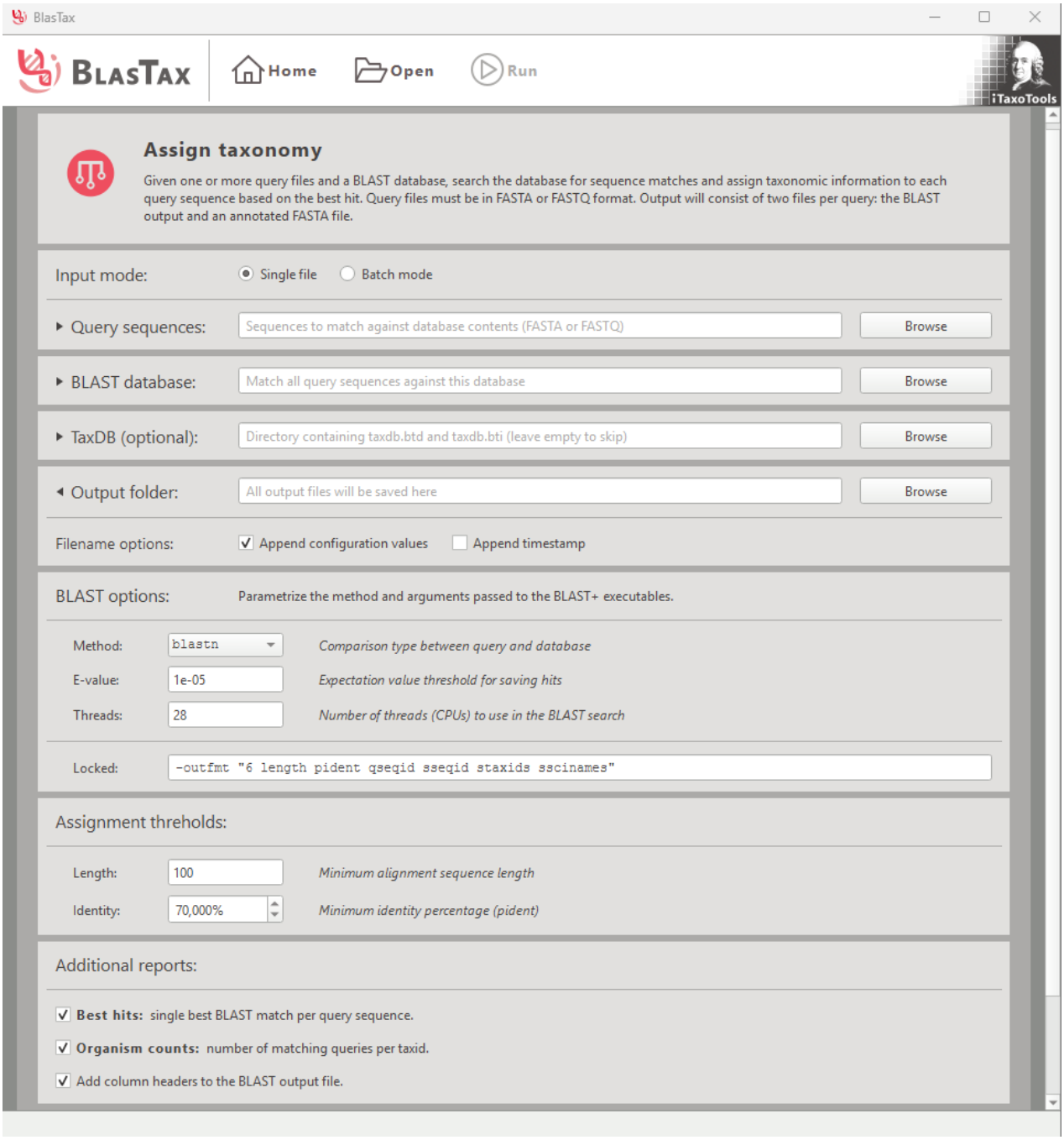
► Decontaminate is a program mode that makes use of BLAST to remove “outlier” sequences from a FASTA file, in particular if these arose by contamination. This program mode is based on a dual BLAST search, i.e., against two databases, one with ingroup sequences and one with outgroup sequences. For instance, in a FASTA query file containing sequences of the genus *Salamandra*, ingroup sequences could be a set of coding sequences (CDS) from high-quality reference genomes of amphibians, whereas outgroup sequences could consist of likely contaminants such as bacteria, protists, fungi, plants, invertebrates and mammals (*Homo sapiens*). The program will run two BLAST searches for every sequence from the query file, and determine if the highest similarity match is found in the ingroup or the outgroup database. The

sequences from the query FASTA file will then be written into two different output FASTA files: one for those matching more closely to the outgroup (likely contaminants) and those matching more closely to the ingroup (likely genuine sequences of the target salamander species). Batch mode is available, i.e., multiple FASTA query files can be specified and processed consecutively (BLASTing them against the same two databases of ingroup and outgroup sequences).

To run "Decontaminate", specify (i) the query FASTA file(s), (ii) the ingroup BLAST database, (iii) the outgroup BLAST database, (iv) the folder for the output FASTA files, (v) the BLAST algorithm and parameters, and (vi) the variable according to which the quality of the matches will be compared: percent identity (pident), alignment length (length) or bit-score (bitscore). The bit-score is a normalized and $\log_2$-scaled measure of the raw BLAST alignment score, which makes it comparable across BLAST searches with different reference datasets. That is, the bit-score remains constant for the same hit in databases of different sizes, and thus and can be used to compare matches in different databases or databases of increased size. This is unlike the e-value, which is calculated from the bit-score but it is normalized by the search space in the reference database at hand; e-values are not equivalent across BLAST searches.



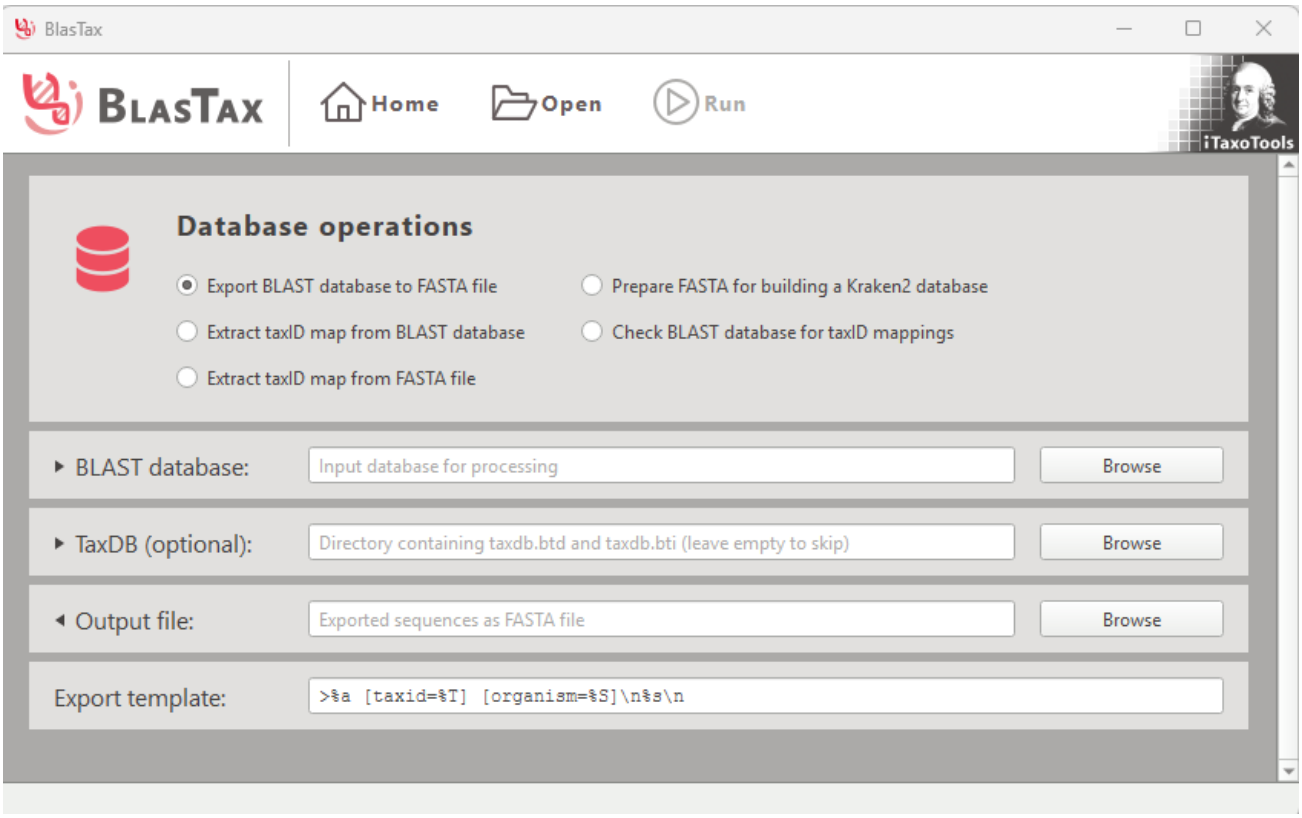
► Assign taxonomy is a program mode that makes use of NCBI taxonomy and the respective "taxID" identifiers to assign taxonomic information to sequences based on BLAST results. For this, it requires a BLAST database made with a taxID mapping file, i.e., including taxID information. In addition, the program can also retrieve actual taxon names (species names) for each hit in the database if the separate NCBI taxDB database is provided (see more information about obtaining downloading this database at https://www.ncbi.nlm.nih.gov/books/NBK569841/).



The regular output of this program mode is (1) a FASTA file with the assigned taxonomy (taxID) added in square brackets after each sequence name, based on the best match in the database, (2) an output table with all matches for each query sequence as well as with the taxID of the

matching sequence. Furthermore, based on the selected checkboxes, the user can also obtain (3) a tab-delimited table with only the best match in the database for each query sequence, along with taxID and species name (if available) of the match, and percent sequence identity and length of the matching stretch of sequence; and (4) a summary table which counts how many query sequences matched to the same reference sequence in the database. The latter table mirrors the basic functionality of specialized programs for DNA metabarcoding such as Kraken, QIIME 2 or OBItools, without the ambition to replace such tools which are optimized for large-scale analysis of organismal community structure via DNA metabarcoding. See also the experimental "Decontamination by taxonomy" mode below.

► Database operations provides GUI access to several operations related to BLAST databases and for this purpose wraps *blastdbcmd* of BLAST+. Specifically, it allows to (1) export a FASTA file from a BLAST database (optionally including taxID information in square brackets after sequence names, and also including species names if a taxDB is provided), and (2) extract a taxID mapping file from a BLAST database (if the database contains taxID information). It also is possible to (3) make a taxID mapping file from a FASTA file that includes taxID information in square brackets after the sequence name (format: `>sequencename1 [taxID=1234]`), (4) run a quick check on a BLAST database to verify it does contain taxID information, and (5) prepare a FASTA file for building a Kraken database (from a script by López Clinton et al. 2025: https://github.com/SamanthaLop/Small_Bugs_Big_Data)



Experimental program modes

BlasTax contains two program modes which implement advanced applications that are included experimentally and not yet exhaustively tested. They will be adjusted and more thoroughly documented in future versions of the program:

► BlastAppendX is similar to "BlastAppend" but uses translated nucleotide (amino acid) queries to retrieve nucleotide sequences for appending to existing nucleotide alignments.
"BlastAppendX" uses the *blastx* algorithm to test the translated query against a protein database, and then retrieves the sequences to be appended from the nucleotide FASTA file that corresponds to the protein database.
Compared to "BlastAppend", this mode is suitable for cases where the expected sequence identity is low between the nucleotide sequences in the query file(s) (i.e., the FASTA alignments to which new sequences should be appended) and the nucleotide sequences in the database (i.e, the transcriptome assembly or genome CDS of a new taxon). As an example, imagine a case where the query files consist of ortholog alignments from salamanders, and now sequences from one or several frog genomes should be added to this data set. The respective nucleotide sequences might be so divergent that a *blastn* search might not identify many homologous regions. Instead, a protein-level search (*tblastx*) will in most cases retrieve the matching sequences due to the higher conservation of amino acid sequences.

As a main requirement to use this program mode, users must prepare two files for the reference data set: (i) the nucleotide FASTA file from which sequences should be appended, and (ii) a protein FASTA file translated from the nucleotide file, with exactly the same sequence identifiers, which is then used to make a BLAST database.
As a further requirement, the query files must consist exclusively of protein coding sequences.
"BlastAppendX" is then executed similar to "BlastAppend", but the second reference file (the nucleotide file) needs to be specified as well. Because the *blastx* algorithm needs to translate the query sequences, the program will take substantially longer to complete compared to "BlastAppend".
This program mode is fully functional but has not been exhaustively tested yet.

► SCaFoS-Py is a simplified Python implementation of the program SCaFoS (Roure et al. 2007). Similar to SCaFoS, its main goal is to deal with alignments with multiple sequences of the same organism derived from a BLAST search, e.g. against a transcriptome assembly or genome CDS file. SCaFoS-Py recognizes sequences belonging to the same organism in a FASTA file by parts of the sequence identifier (header): either (i) looking up identical strings of characters before the first underscore, (ii) looking up identical strings of characters before the second underscore (assuming a naming scheme by "Genus_species_xxxx"), or (iii) looking up an identical strings of characters before a pipe character, following the MolD-FASTA convention (Fedosov et al. 2022).

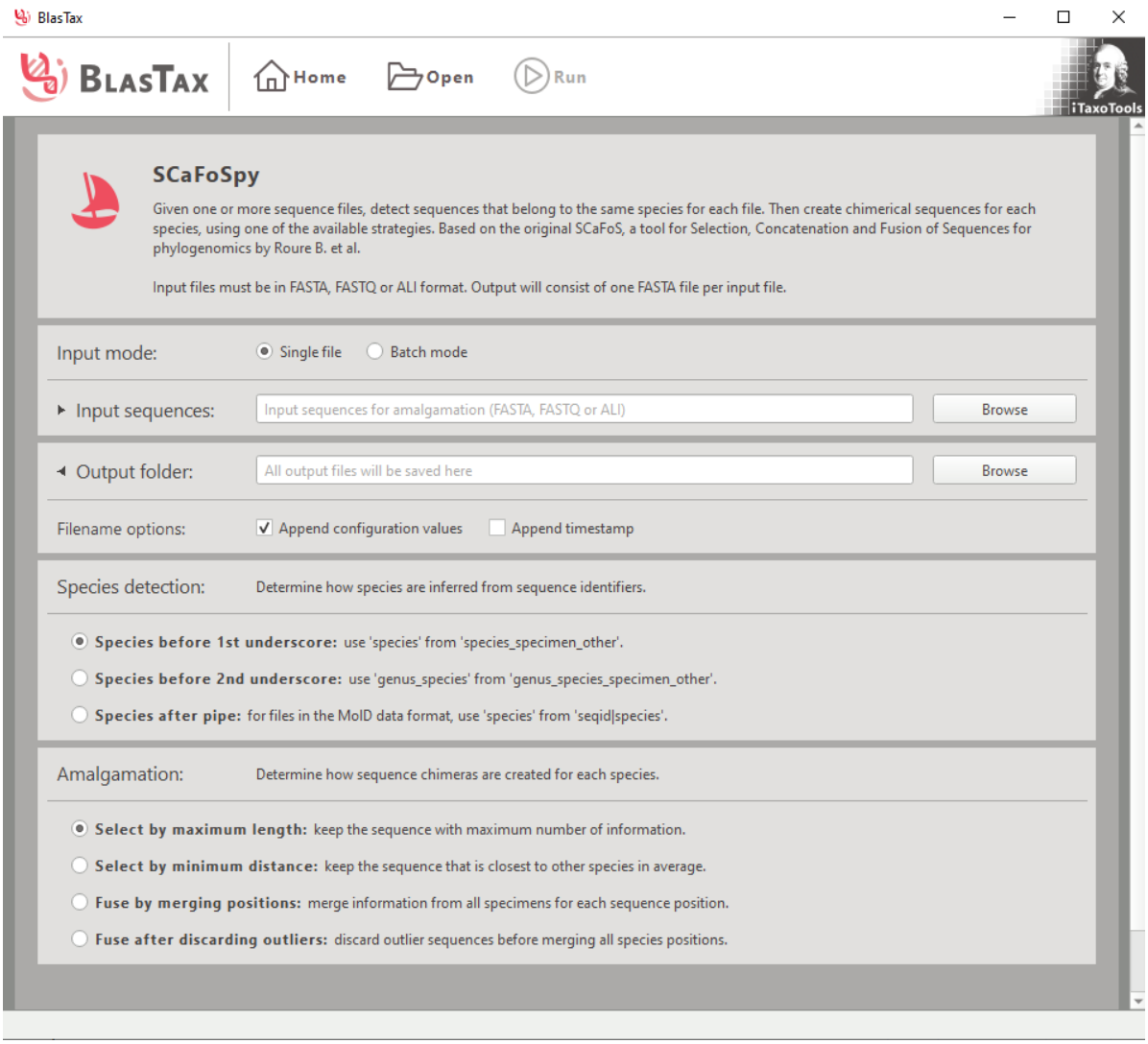
The program can be run under four different sets of rules for deciding how to select among and merge the sequences identified as belonging to the same taxa:
(i) select by maximum length: the longest sequence is chosen and parsed into the output FASTA file; all other sequences of the same species are discarded. This option has been tested and works reliably, both with aligned and unaligned sequences.

(ii) select by minimum distance: the program calculates pairwise genetic distances to all other sequences in the FASTA file; the sequence with the lowest average distance to all other sequences is chosen and parsed into the output FASTA file; all other sequences of the same taxa are discarded. This option is experimental and has not yet been exhaustively tested. It requires all sequences in the FASTA file to be aligned.

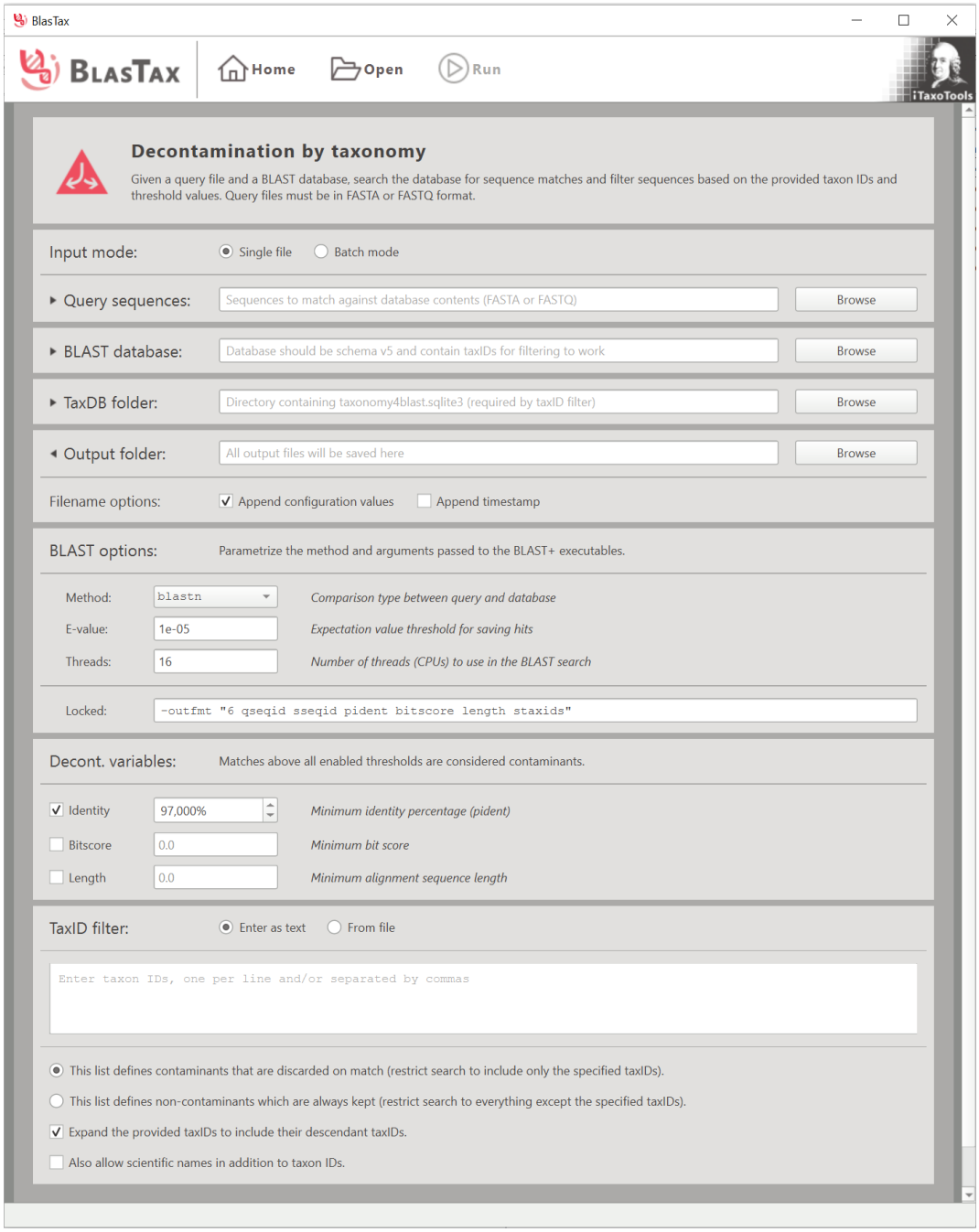
(iii) Fuse by merging positions: all sequences identified as belonging to one taxon are fused. Where disagreements are found in overlapping regions, these will be coded as IUPAC ambiguity codes. This option is functional and should work with fully aligned input FASTA files.

(iv) Fuse after discarding outliers: a combination of options ii and iii: first, sequences identified as outliers based on pairwise distances are removed and all remaining sequences identified as belonging to the same species are fused. This option works only with fully aligned sequences and is experimental; it has not been exhaustively tested.

SCaFoS-Py requires the user to specify the FASTA file(s) for processing, and an output folder where the new FASTA files are written (with only one sequence per species per FASTA file, after the selection / fusing process according to the option chosen). Batch mode is available, i.e., multiple FASTA query files can be specified and processed consecutively.



► Decontamination by taxonomy is a further experimental and not fully tested program mode which leverages NCBI taxonomy for filtering a set of sequencing based non their matches to a database.
Different from most other program modes, which are optimized for version 4 BLAST databases, this program mode requires a version 5 BLAST database that includes taxIDs. The user provides a list of NCBI taxIDs and/or taxon names according to NCBI taxonomy that should be removed from or kept in the output FASTA file. In addition, the user needs to provide the taxonomy4blast.sqlite database and the names.dmp file which can be downloaded from the NCBI FTP server (and BlasTax provides an option to automatically download these files). The program then uses BLAST and assigns taxonomy to each query sequence based on the best BLAST match, and uses this inferred identity for filtering.



Program modes for the preparation of sequence files for analysis

BlasTax includes several program modes that are not directly related to actual BLAST searches but are useful for preparing sequence files (in FASTA or FASTQ format) so that they can be processed by the various analysis modes of the program. Several of these modes, all available from dedicated tiles in the starting window, are adapted or modified from stand-alone tools previously developed in iTaxoTools but for convenience are here integrated in BlasTax.

► FastSplit and FastMerge serve to split large text files (typically FASTQ or FASTA files, including compressed files) into several files based on certain criteria, or merge several files into a single FASTQ or FASTA file. For more detailed explanations, see the regular iTaxoTools manual which includes chapters on these programs.

► GroupMerge is a new variant of FastMerge which merges FASTQ or FASTA files based on a part of their filename, and subsequently merges all sequences of each group into a single FASTA file. The grouping can be achieved

(i) by the first word in a filename, i.e., the series of characters until the first underscore: RAG1_Lissamphibia.fas and RAG1_Squamata.fas will be merged into one FASTA file.
(ii) by looking up a user specified number of characters at the start of filenames, e.g., the first three characters in a filename. Filenames agreeing in the first three characters of their name will be merged into one FASTA file.
(iii) by looking up a regular expression in the filename and grouping files with a matching first group of the specified regex

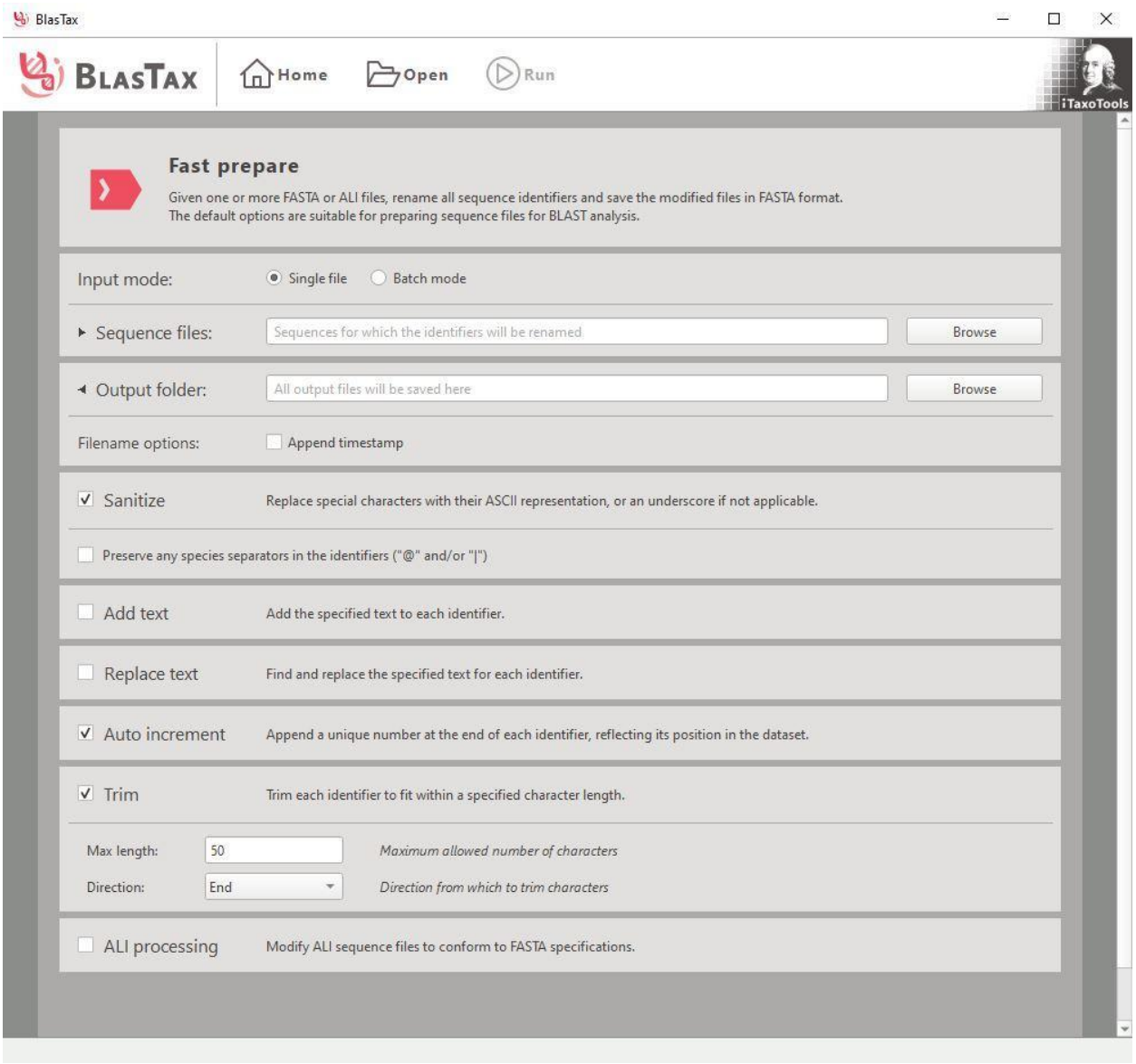
Furthermore, the program will check upon merging for duplicated sequence identifiers in the sequence files selected for merging, and the user can choose if all occurrences of duplicate identifiers, or only the first encountered occurrence should be kept.

► Fast prepare is an essential tool to modify FASTA files so that they become suitable for making a BLAST database (the program *makeblastdb* has some limitations on the length and format of FASTA headers). Importantly, this program mode has the following options which can be selected in the GUI with the respective checkboxes:

(i) Automatically sanitize (but not trim) sequence identifiers in a FASTA file (all characters not being standard alphanumerical, including spaces, will be converted to underscores)
(ii) Trim sequence identifiers (for making a BLAST database, a maximum of 50 characters are allowed in sequence identifiers). Trimming can take place (user selection) at the end or the beginning of the identifiers, i.e., characters are cut from the beginning or end of the sequence names (=identifiers or headers) when their length surpasses the maximum number of characters specified.
(iii) To make sure there are no identical sequence names (unique sequence identifiers are required in a BLAST database), the option “Auto Increment” will append a unique number at the end of each sequence identifiers. If this option is checked in combination with the “Trim” option, the trimmed sequence identifiers will have the maximum number of characters specified while having a unique number at the end.

(iv) In addition, FastPrepare also allows modifying sequence identifiers by a simple search-replace function, or by adding a specified set of characters at the end or beginning of each identifier.

Batch mode is available in FastPrepare, i.e., multiple FASTA files can be specified and processed consecutively.



► Removal of stop codons is a tool to identify and remove stop codons from stretches of protein-coding genes, which frequently appear due to errors in genome sequencing, assembly and annotation in genome and transcriptome data. Sequencing errors, low-coverage regions, or the erroneous inclusion of introns in genome annotation might lead to the presence of erroneous stop codons or open reading frame shifts. This tool assumes that such stop codons are likely erroneous and are thus removed together with the upcoming downstream sequence, or the entire sequence might also be filtered out. It also generates modified FASTA file(s) as output into a user-selected folder.

The appropriate genetic code (translation table) and reading frame can be selected by the user and must be the same for all sequences processed in one run. Gaps should be removed before the analysis (i.e., unaligned sequences should be used as input). Batch mode is available, i.e., multiple FASTA files can be specified and processed consecutively.

The program searches for stop codons in each sequence and based on the user selection (clicking one of the radio buttons) performs one of the following action:

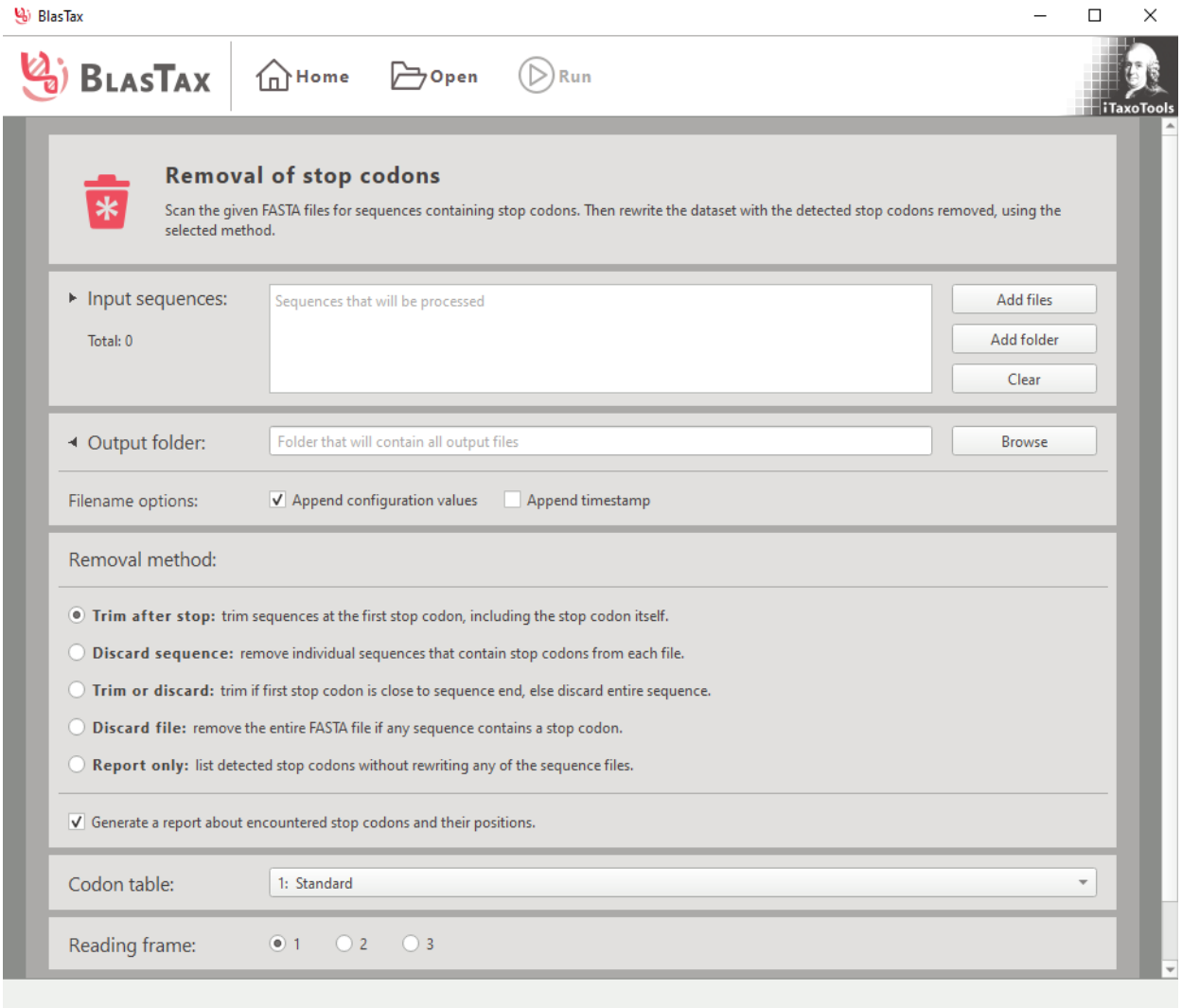
(i) For each sequence where a stop codon is encountered, remove the entire stretch of nucleotides after the stop codon (and including the stop codon).

(ii) For each sequence where a stop codon is encountered, remove the entire sequence

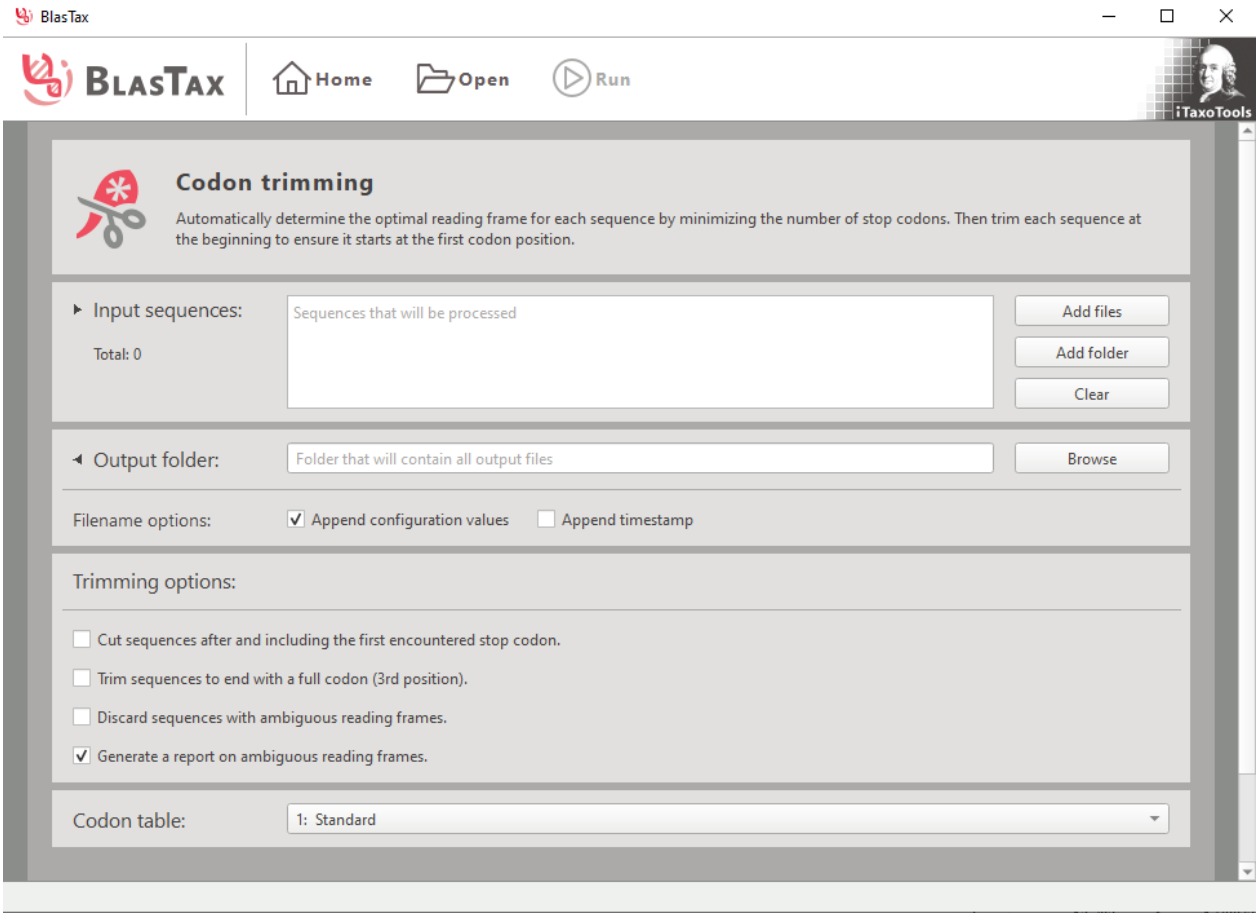
(iii) Trim sequences for stop codons found in the (user-defined) terminal part of the sequences, remove entire sequences if stop codons are encountered elsewhere.

(iv) For each FASTA file containing a sequence with a stop codon, remove the entire FASTA file (i.e., do not parse this FASTA file in the output folder).

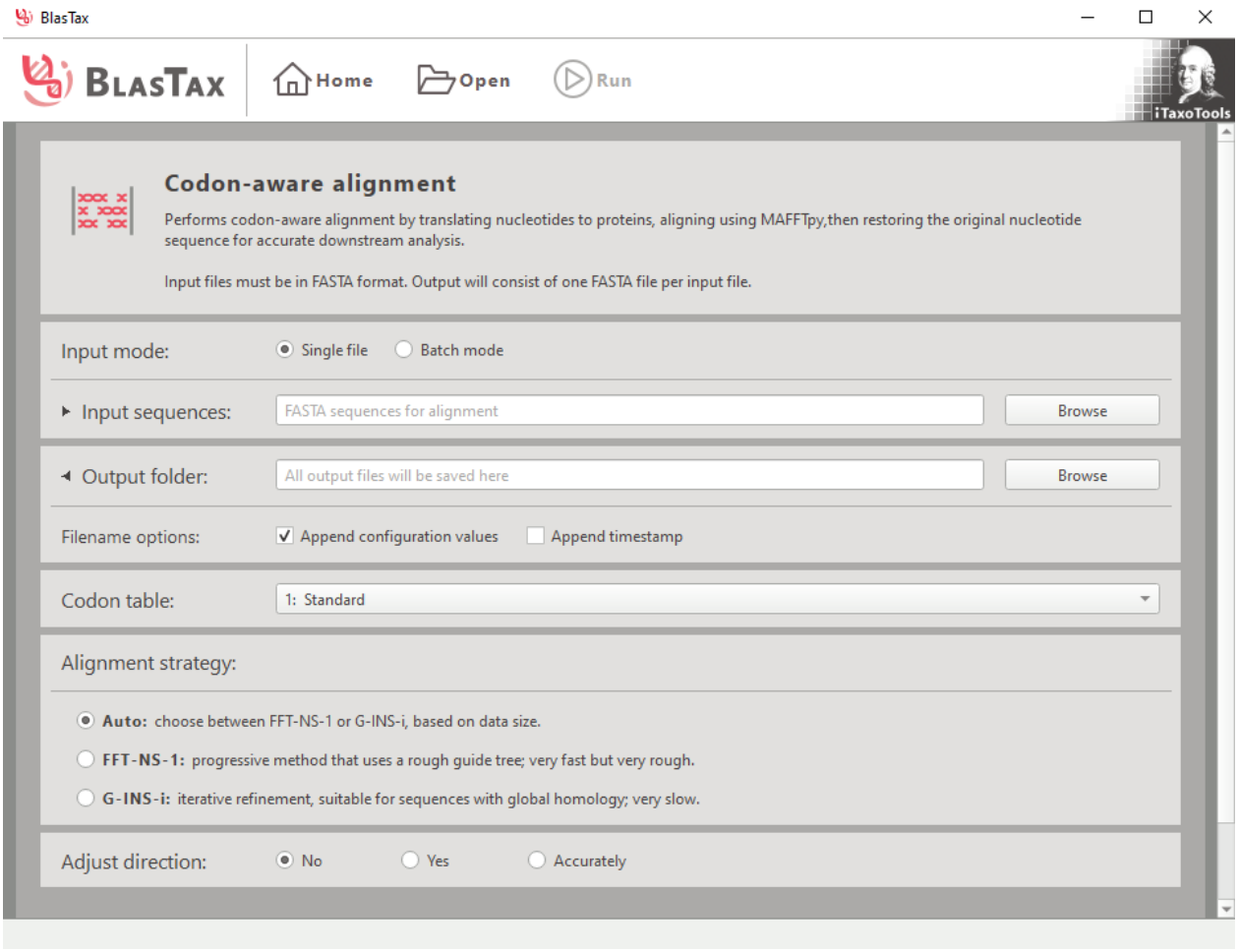
(v) only a report with a list of encountered stop codons is produced, without modification of any FASTA file.



► Codon trimming can be used for FASTA files consisting only of coding sequences that do not all start in the same reading frame (i.e., some sequences may start at the first codon position, whereas others start at the second and third position. Such sequences cannot be easily used for codon-aware multiple sequence alignment or for single-reading frame protein translation. “CodonTrimming” therefore looks up the most likely (stop codon free) translation per sequence (i.e., the longest open reading frame for forward open reading frames 1, 2 and 3), and trims the respective sequence at the beginning so that it starts at a the first codon position. Batch mode is available, i.e., multiple FASTA files can be specified and processed consecutively.



► Codon-aware alignment aligns nucleotide coding sequences, based on the respective protein alignment. This approach preserves codons as the unit of evolution, avoiding the inclusion of gaps within codons, which will happen if protein-coding genes are aligned at the nucleotide level. “Codon-aware alignment” is not directly related to BLAST searches but can be used to re-align nucleotide sequences after adding new sequences to FASTA files via the “BLAST-Append mode”. To run “Codon-aware alignment”, make sure that every sequence in your FASTA file(s) starts with a first codon position (if necessary, trim with “CodonTrimming” and check the translation with “ProteinTranslator”). Batch mode is available, i.e., multiple FASTA files can be specified and processed consecutively.



► CutAdapt is the implementation of the original Cutadapt code from Martin (2011) as available from GitHub (https://github.com/marcelm/cutadapt). It makes the basic functions of adapter removal from FASTQ and FASTA files, and quality trimming of FASTQ files, accessible via GUI. Explanations of the various options are given in the GUI and can also be found in more detail on the Cutadapt website (https://cutadapt.readthedocs.io/en/stable/).

► ProteinTranslator is a versatile tool to translate nucleotide sequences into protein (amino acid) sequences. The user specifies the CDS nucleotide FASTA file(s) to be translated, the output folder, the filename (in case of translation of a single nucleotide FASTA file), the appropriate genetic code (codon table) and the reading frame. An "autodetect" option for the reading frame is also available. Batch mode is available, i.e., multiple FASTA files can be specified and processed consecutively. The tool can perform the translation in four different flavors:

(i) Coding sequence: Here, the program will search for the reading frame with the lowest number of stop codons (usually one frame without stop codons should exist if the sequences are long enough), or use for all sequences the specified reading frame.

(ii) Coding sequence with stop: Here the program will perform as in the first option, but assuming that the sequences include a terminal stop codon.

(iii) Transcript: Here the program will look for the longest open reading frame, i.e., the longest coding sequence without stops. Among other use cases, this is for transcriptome assemblies where the sequences may contain non-coding parts at the end and beginning, and some sequences may be reverse-complemented.

(iv): All: Here the program will simply translate the sequences using all six possible reading frames, irrespective of stop codons.

References cited

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. Journal of Molecular Biology, 215(3), 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2

Bolyen, E., Rideout, J. R., Dillon, M. R., Bokulich, N. A., ... Knight, R. & Caporaso, J. G. (2019). Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. Nature Biotechnology, 37, 852–857. https://doi.org/10.1038/s41587-019-0209-9

Boyer, F., Mercier, C., Bonin, A., Le Bras, Y., Taberlet, P. & Coissac, E. (2016). OBITools: A UNIX-inspired software package for DNA metabarcoding. Molecular Ecology Resources, 16, 176–182. https://doi.org/10.1111/1755-0998.12428Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., & Madden, T. L. (2009). BLAST+: Architecture and applications. BMC Bioinformatics, 10, 421. https://doi.org/10.1186/1471-2105-10-421

Fedosov, A., Achaz, G., Gontchar, A. & Puillandre, N. (2022) Mold, a novel software to compile accurate and reliable DNA diagnoses for taxonomic descriptions. Molecular Ecology Resources, 22, 2038–2053. https://doi.org/10.1111/1755-0998.13590

Katoh, K., Standley, D.M. (2013). MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Molecular Biology and Evolution 30: 772–780.

López Clinton, S., Iwaszkiewicz-Eggebrecht, E., Miraldo, A., Goodsell, R., Webster, M.T., Ronquist, F., van der Valk, T. (2025) Small bugs, big data: Metagenomics for arthropod biodiversity monitoring. Ecology and Evolution, 15,e72163.

Martin, M. (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal, 1, 10–12.

Rancilhac L., T. Bruy, M.D. Scherz, E. Almeida Pereira, M. Preick, N. Straube, M.L. Lyra, A. Ohler, J.W. Streicher, F. Andreone, A. Crottini, C.R. Hutter, J.C. Randrianantoandro, A. Rakotoarison, F. Glaw, M. Hofreiter & M. Vences (2020): Target-enriched DNA sequencing from historical type material enables a partial revision of the Madagascar giant stream frogs (genus *Mantidactylus*). – *Journal of Natural History* 54: 87-118.

Roure, B., Rodriguez-Ezpeleta, N., & Philippe, H. (2007). SCaFoS: A tool for selection, concatenation and fusion of sequences for phylogenomics. BMC Evolutionary Biology, 7(Suppl 1), S2. https://doi.org/10.1186/1471-2148-7-S1-S2

Vences, M., Miralles, A., Brouillet, S., Ducasse, J., Fedosov, A., Kharchev, V., Kumari, S, Patmanidis, S., Puillandre, N., Scherz, M. D., Kostadinov, I., Renner, S. S. (2021). iTaxoTools 0.1: Kickstarting a specimen-based software toolkit for taxonomists. Megataxa 6: 77-92.

Wood, D. E. & Salzberg, S. L. (2014). Kraken: Ultrafast metagenomic sequence classification using exact alignments. Genome Biology, 15, R46. https://doi.org/10.1186/gb-2014-15-3-r46