



iTaxoTools - tools for integrative taxonomy

iTaxoTools 0.1.1 alpha - Preliminary Manual

Disclaimer: The programs included in iTaxoTools are free software. All code specifically programmed for iTaxoTools can be redistributed and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**. See the GNU General Public License for more details.

All tools not specifically programmed for iTaxoTools but constituting a modification or extension of the original code are also free software and most of them licensed under GNU v3, but partly, other licences apply. Please check the original GitHub pages (see table at the end of Introduction below) for licensing of the original code of PTP, GMYC, tr2, DELINEATE, ABGD, ASAP, LINES 2.0 and MolD.

Content

0. Release notes

1. Introduction: The Concept of iTaxoTools

2. How to Cite

3. Distribution: Code, Stand-Alone Executables, Launcher, Webserver

4. Tools for Data Preparation

- 4.1. DNAconvert
- 4.2. latlonconverter
- 4.3. pyr8s
- 4.4. fastmerge and fastsplit
- 4.5. specimentablemerger
- 4.6. specimentablepruner
- 4.7. linebreaker
- 4.8. nodenamecorrector
- 4.9. unitconverter

5. Tools for Data Analysis

- 5.1. TaxI2
- 5.2. morphometricanalyzer
- 5.3. simplestatscalculator

6. Tools for Species Delimitation

- 6.1. ABGD
- 6.2. ASAP
- 6.3. DELINEATE
- 6.4. GMYC
- 6.5. PTP
- 6.6. tr2
- 6.7. LIMES
- 6.8. spartmapper

7. Tools for Species Diagnosis

- 7.1. DNADIAGNOSER
- 7.2. MOLD

8. References

0. Release Notes

Manual: This is a preliminary version of the manual (first published along with the pre-release of iTaxoTools, 26 March 2021) and will be continuously updated and expanded during April/May 2021.

General Notes:

► The current release iTaxoTools 0.1.1. alpha is to be considered a pre-release, along with submission of the respective manuscript to peer review, and publication of a preprint on BioRxiv. Improvement of the graphical user interfaces and exhaustive testing of all tools is in progress. The GUI versions of existing species delimitation programs (ABGD, ASAP, DELINEATE, GMYC, PTP, SODA, TR2) or molecular diagnosis programs (MOLD) should work reliably as the underlying original code has not been modified. We are also confident that the simple conversion or data transformation tools (DNAconvert, latlonconverter, fastsplit, fastmerge...) are largely bug-free as we have been using them extensively ourselves over several months. More extensive newly programmed tools (pyr8s, dnadiagnoser, morphometricanalyzer, TaxI2) may in some cases still be unstable and the accuracy of the produced results has not yet been comprehensively tested.

► In general, with all iTaxoTools programs on Windows, make sure all files (input and output) and the program itself are on the same logical drive, e.g., C:\ or D:\, otherwise the program may not work. It is of course no problem to specify different folders on the same logical drive for input/output files.

► Among future plans of iTaxoTools is the implementation of a Help Wiki

Specific release notes on the various tools:

All tools	<p>► When using the tools on Windows, make sure all files (input and output) and the program itself are on the same logical drive, e.g., C:\ or D:\, otherwise the programs may not work. It is of course no problem to specify different folders on the same logical drive for input/output files and have the program again in a different folder.</p>
specimentablepruner	<p>► This program comes with relatively extensive Python libraries and therefore takes a substantial time to load on slow systems.</p> <p>► This program has not yet been exhaustively tested and may contain bugs.</p> <p>► When using this program on Windows systems, as with all iTaxoTool programs, make sure all files (input and output) and the program itself are on the same logical drive, e.g., C:\ or D:\, otherwise the program may not work. It is of course no problem to specify different folders on the same logical drive for input/output files.</p>
specimentablemerger	<p>► This program comes with relatively extensive Python libraries and therefore takes a substantial time to load on slow systems.</p> <p>► This program has not yet been exhaustively tested and may contain bugs.</p> <p>► When using this program on Windows systems, as with all iTaxoTool programs, make sure all files (input and output) and the program itself are on the same logical drive, e.g., C:\ or D:\, otherwise the program may not work. It is of course no problem to specify different folders on the same logical drive for input/output files.</p>

linebreaker	► In the current pre-release there is an inconsistency in the name of the program (linebreaker vs. linebreak-replacer) which will be standardized in the next release.
simplestatscalculator	► In the current pre-release the help documentation is still incomplete.
spartmapper	► In the current pre-release the program can only read single-partition matricial SPART files (no XML-SPART and no multi-partition matricial SPART).
nodenamecorrector	► may not work properly with all variants of the Newick format
TaxI2	► The current pre-release includes a functional version of TaxI2 written in pure Python which can perform bug-free limited all-against all comparisons, as well as comparisons to reference database for files in tab-format. However, the current implementation still contains memory leaks that need to be fixed, and at present only can process relatively small files, i.e., ca. 300 unaligned or ca. 2000 pre-aligned sequences for the all-against all comparison mode, and a total of about 100,000 comparisons (e.g., testing 2000 sequences against a reference database of 50 sequences) for the reference dataset mode. These errors are being fixed, and future versions (especially web-based) will obviously include faster alignment options, possibly in other programming languages.
morphometricanalyzer	► In the current pre-release the program cannot deal with missing values. This will be fixed in the upcoming versions. ► As with all iTaxoTools, make sure the stand-alone program is on the same logical drive as the input and output files.
dnadiagnoser	► DNAdiagnoser includes a relatively heavy suite of libraries and therefore will take a substantial time to load. ► In the pre-release, the program currently includes the full <i>Homo sapiens</i> COI (or <i>cox1</i> gene) but additional references will be added and an option to specify/load additional reference sequences will be enabled.
PTP	► In the current version only runs with default settings. Options to set parameters will be included in forthcoming versions.
GMYC	► In the current version only runs with default settings. Options to set parameters will be included in forthcoming versions.
tr2	► In the current version only runs with default settings. Options to set parameters will be included in forthcoming versions.
DELINEATE	► In the current version only runs with default settings. Options to set parameters will be included in forthcoming versions.

1. Introduction: The Concept of iTaxoTools

Only few bioinformatic tools so far have been tailored to specifically fit the practical work of taxonomists. iTaxoTools is a collection of tools specifically developed to facilitate the taxonomic workflow: delimiting, diagnosing and describing species. This workflow requires taxonomists to examine voucher specimens and associated catalogues, field books and pictures; take, tabulate and statistically analyze morphometric measurements; define, tabulate and document phenotypic character states; estimate geographical ranges based on specimen provenances; align and analyze DNA sequences; and elaborate accurate specimen tables, species diagnoses and identification keys. Depending on the organism under study, it also may involve more specialized procedures such as comparing acoustic and visual signal repertoires of animals, or isolate and culture unicellular organisms. In addition, to fulfil standards of cybertaxonomy, data sets need to be archived in specialized repositories and new species names registered in online databases (Miralles et al. 2020).

The concept of iTaxoTools rests on four pillars: (1) **fully open source** code; (2) a **diversified** set of stand-alone programs ('modules') that in future versions will become increasingly interconnected; (3) a **specimen-centered** architecture, where at present tables (tab-delimited text files) with specimen identifier columns serve as main input format for many of the tools; and (4) a focus on **user-friendliness**, accessibility, and clear and transparent documentation.

Simplicity and user-friendliness are at the core of iTaxoTools. Because the majority of taxonomists is not familiar with programming languages, such as Python, all our tools are accessible via graphical user interfaces (GUI) – analyses can therefore be carried out with a few intuitive mouse clicks, under default or custom settings, without the need to enter commands in a command line.

As an important aspect, several of the newly developed tools include autocorrect routines to avoid the loss of time associated with the search for small misspellings or incorrect characters in input files that cause programs to fail.

We chose Python as the main programming language for our package, because it is characterized by its good readability and simple-to-learn syntax, and we documented newly written code extensively, to allow its re-use by other programmers. This comes at the cost of speed that would have been achieved by using the C programming language, but our toolkit in this early phase is not designed to cope with huge genomic datasets or analyses with tens of thousands of specimens. Currently iTaxoTools is designed to provide support for the most common taxonomic research projects that discover and name a limited number of species only (Miralles et al. 2020), but will be extended to large-scale projects in the future. This will require increasing memory usage and speed of the algorithms, possibly sometimes including C code, to be able to process very large data sets.

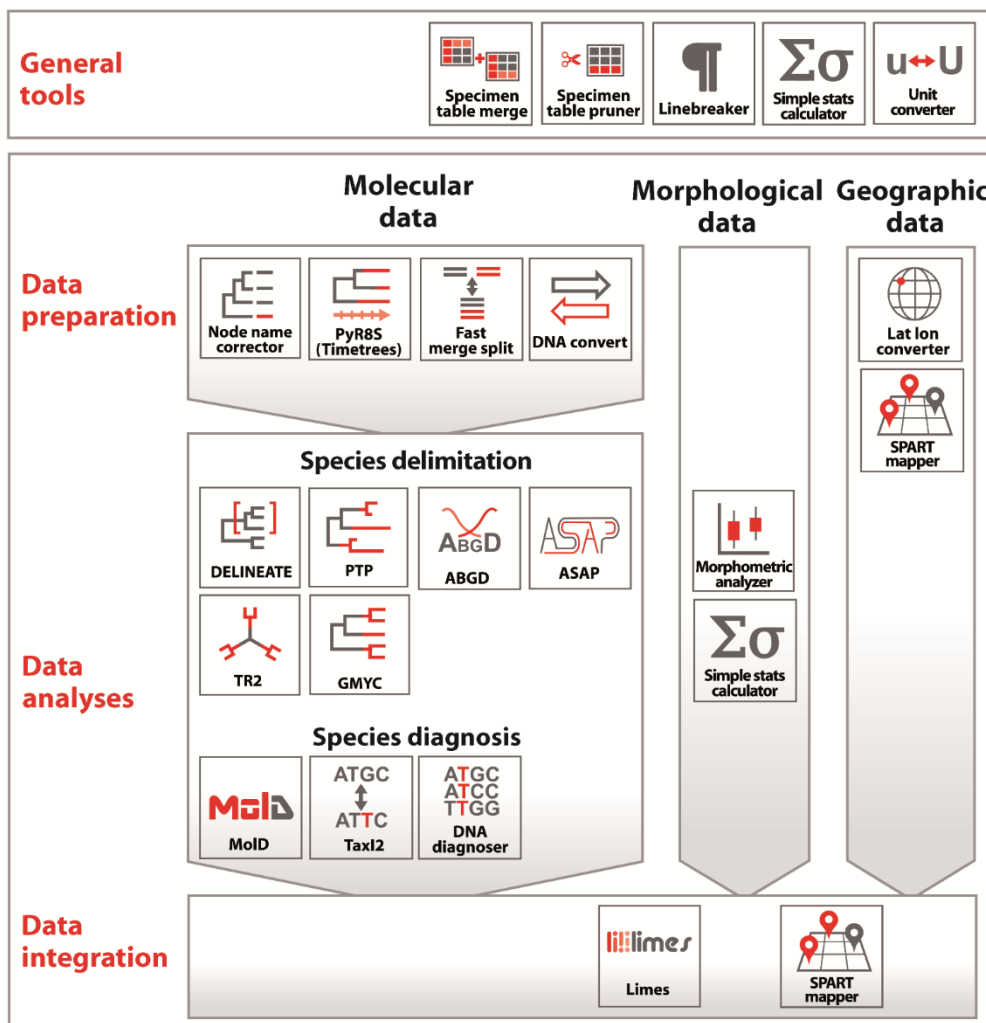
Considering that powerful programs exist for phylogenetic and phylogenomic analyses, and tasks such as multiple alignment of sequences or genome/transcriptome assembly, we did not attempt to include such functionalities in our toolkit. For many of these tasks, GUI-driven programs already exist, such as MEGA (Kumar et al. 2018) or raxmlGUI2 (Stamatakis 2014; Edler et al. 2020), and there is an active community both of commercial companies and academic research teams constantly extending these kinds of programs.

Regarding input and output files, the current version of iTaxoTools is not yet fully standardized. Those tools that were developed by other researchers and for which we here provide a GUI version have not been altered in their original code, which means, they require input files of the same format as the respective command-line driven or web-based original tools (ABGD, ASAP, DELINEATE,

GMYC, PTP, SODA, TR2, MOLD). See the respective sections on these tools below. See also below for notes on the available distributions of the tools.

However, iTaxoTools implements two innovations: The first is the SPART output format for species partition information (Miralles et al. 2021) that we have implemented in our versions of all the species delimitation programs, and that is read by LIMES and SPARTMAPPER. For now, only the matricial format has been implemented; it has been included in the native code of ABGD and ASAP (and LIMES and SPARTMAPPER), while our versions of the other delimitation programs for now provide spart output only as an add-on via the GUI code; in several of these, SPART output is being implemented in the native code as well by the original developers (e.g., already implemented in TR2).

As a second innovation, several programs of the iTaxoTools toolkit use tab-delimited text as standard input (and sometimes output) format. Usually, one column indicates the specimen identifier, reflecting the specimen-based workflow in alpha taxonomy. This will in subsequent versions allow the user to save the output of different tools for each specimen, and combine these results for further analysis. Especially, the tab-delimited format also allows easy editing of the data tables in spreadsheet editors.



The following table lists the repositories of the code of the tools included in this pre-release of iTaxoTools. The table also lists the main programmers involved in the development of each tool or its graphical user interface (GUI), and informs whether a tool was newly programmed for this project, adjusted from existing code (by adding a GUI plus sometimes additional functionalities), or included as original code and GUI without modification.

Tool	Github repository (original / modified)	Main programmers (original program) / GUI
dnacvert	https://github.com/iTaxoTools/DNAconvert	V. Kharchev
latlonconverter	https://github.com/iTaxoTools/latlon-converter	V. Kharchev
fastmerge	https://github.com/iTaxoTools/fastsplit-merge	V. Kharchev
fastsplit	https://github.com/iTaxoTools/fastsplit-merge	V. Kharchev
specimentablepruner	https://github.com/iTaxoTools/specimentablepruner	V. Kharchev
specimentablemerger	https://github.com/iTaxoTools/specimentablemerger	V. Kharchev
linebreaker	https://github.com/iTaxoTools/linebreaker	S. Kumari
simplestatscalculator	https://github.com/iTaxoTools/simple_stat	S. Kumari
unitconverter	https://github.com/iTaxoTools/UnitConverter	S. Kumari
spartmapper	https://github.com/iTaxoTools/linebreak_replacer	S. Kumari
nodenamecorrector	https://github.com/iTaxoTools/nodenamecorrector	V. Kharchev
pyr8s	https://github.com/iTaxoTools/pyr8s	S. Patmanidis
TaxI2	https://github.com/iTaxoTools/TaxI2	V. Kharchev
morphometricanalyzer	https://github.com/iTaxoTools/morphometricanalyzer	V. Kharchev
dnadiagnoser	https://github.com/iTaxoTools/dnadiagnoser	V. Kharchev
PTP	https://github.com/zhangjiajie/PTP https://github.com/iTaxoTools/PTP-PYQT5	(J. Zhang) GUI: S. Kumari
GMYC	https://github.com/zhangjiajie/pGMYC https://github.com/iTaxoTools/GMYC-PYQT5	(J. Zhang) GUI: S. Kumari
tr2	https://github.com/tfujisawa/tr2-delimitation-git https://github.com/iTaxoTools/pyqt5-tr2	(T. Fujisawa) GUI: S. Kumari
DELINEATE	https://github.com/iTaxoTools/pyqt5-delineate	(J. Sukumaran) GUI: S. Kumari
ABGD	https://bioinfo.mnhn.fr/abi/public/abgd/ https://github.com/iTaxoTools/ABGDpy	(S. Brouillet) GUI: S. Patmanidis
ASAP	https://bioinfo.mnhn.fr/abi/public/asap/ https://github.com/iTaxoTools/ASAPy	(S. Brouillet) GUI: S. Patmanidis
LIMES 2.0	https://github.com/iTaxoTools/LIMES	J. Ducasse
MolD	https://github.com/SashaFedosov/MolD https://github.com/iTaxoTools/MolD_pyqt5	(A. Fedosov) GUI: S. Kumari

2. How to Cite

When using one of the programs included in the iTaxoTools 0.1.1. release in your study, please cite the respective paper:

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. *BioRxiv*, DOI: [XXXXXX](#)

The source code of all tools can be found on GitHub: <https://github.com/iTaxoTools>

Furthermore, when using one of the tools originating from other teams, please make sure to primarily cite the respective original papers. This refers to all tools for species delimitation and one tool for molecular diagnosis:

ASAP: Puillandre, N., Brouillet, S. & Achaz, G. (2021) ASAP: assemble species by automatic partitioning. *Molecular Ecology Resources*, 21: 609-620.

ABGD: Puillandre, N., Lambert, A., Brouillet, S. & Achaz, G. (2012) ABGD, Automatic Barcode Gap Discovery for primary species delimitation. *Molecular Ecology*, 21, 1864–1877.

DELINEATE: Sukumaran, J., Holder, T.M. & Knowles, L.L. (2020) Incorporating the speciation process into species delimitation. <https://github.com/jeetsukumaran/delineate>.

GMYC: Pons, J., Barraclough, T.G., Gomez-Zurita, J., Cardoso, A., Duran, D.P., Hazell, S., Kamoun, S., Sumlin, W.D. & Vogler, A.P. (2006) Sequence-based species delimitation for the DNA taxonomy of undescribed insects. *Systematic Biology*, 55, 595–609.

PTP: Zhang J., Kapli P., Pavlidis P. & Stamatakis A. (2013) A general species delimitation method with applications to phylogenetic placements. *Bioinformatics*, 29, 2869–2876.

TR2: Fujisawa, T., Aswad, A. & Barraclough, T.G. (2016) A rapid and scalable method for multilocus species delimitation using Bayesian model comparison and rooted triplets. *Systematic Biology*, 65, 759–771

LIMES (indexes calculation): Ducasse, J., Ung, V., Lecointre, G. & Miralles, A. (2020). LIMES: a tool for comparing species partition. *Bioinformatics*, 36, 2282–2283.

LIMES (SPART files handling): Miralles, A., Ducasse, J., Brouillet, S., Flouri, T., Fujisawa, T., Kapli, P., Knowles, L.L., Kumari, S., Stamatakis, A., Sukumaran, J., Lutteropp, S., Vences, M. & Puillandre, N. (2021). SPART, a versatile and standardized data exchange format for species partition information. *bioRxiv* 2021.03.22.435428; doi: <https://doi.org/10.1101/2021.03.22.435428> (preprint, to be updated after publication)

MOLD: Fedosov, A., Achaz, G. & Puillandre, N. (2019) Revisiting use of DNA characters in taxonomy with MOLD - a tree independent algorithm to retrieve diagnostic nucleotide characters from monolocus datasets. *bioRxiv*, 838151; doi: <https://doi.org/10.1101/838151>

Furthermore, when using **PYR8S**, in many cases it will be appropriate to cite the original work by M.J. Sanderson on non-parametric rate smoothing and the r8s program:

PYR8S / r8s: Sanderson, M.J. (1997) A non-parametric approach to estimating divergence times in the absence of rate constancy. *Molecular Biology and Evolution*, 14, 1218–1231.

Sanderson, M.J. (2003) r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics*, 19, 301–302.

And for the general concept of comparing sequences against a reference database using pairwise alignments in **TaxI2**, it might be appropriate to cite the paper introducing the original TaxI program:

TaxI / TaxI2 : Steinke, D., Salzburger, W., Vences, M. & Meyer, A. (2005) TaxI - A software tool for DNA barcoding using distance methods. – *Philosophical Transactions of the Royal Society London, Ser. B*, 360, 1975–1980.

3. Distribution: Code, Stand-Alone Executables, Launcher, Webserver

All of the code developed by us is fully open source and available from dedicated GitHub repositories, all under <https://github.com/iTaxoTools>.

In the case of tools programmed by other researchers, the original references and links are specified in the GUI.

The Github repositories also include command-line versions of most tools (except for some where we wrapped a Python GUI around an original C code; here, the command line versions are available from the original distribution of these programs).

We distribute pre-compiled executables of all tools for Windows (tested on Windows 7 and Windows 10) and Linux, as well as a selection of tools for MacOS. More Mac executables will be compiled and added in future releases. Each tool is a single, easily portable standalone executable and can be downloaded and run without any of the other tools. This makes it possible for users to only download or use a portion of the software, adapted to their needs. However, keep in mind that these standalone executables come with all libraries needed for execution of the respective programs, and these need to be unpacked into a temporary folder when the program starts. Some of these tools therefore will take some time to start.

Furthermore, we distribute one "launcher" which is a standalone executable as well, containing the majority of tools that can be launched from the respective program symbols.

Several of the previously existing programs already run from different webserver; we will establish a dedicated webserver where all tools can be run online in the near future.

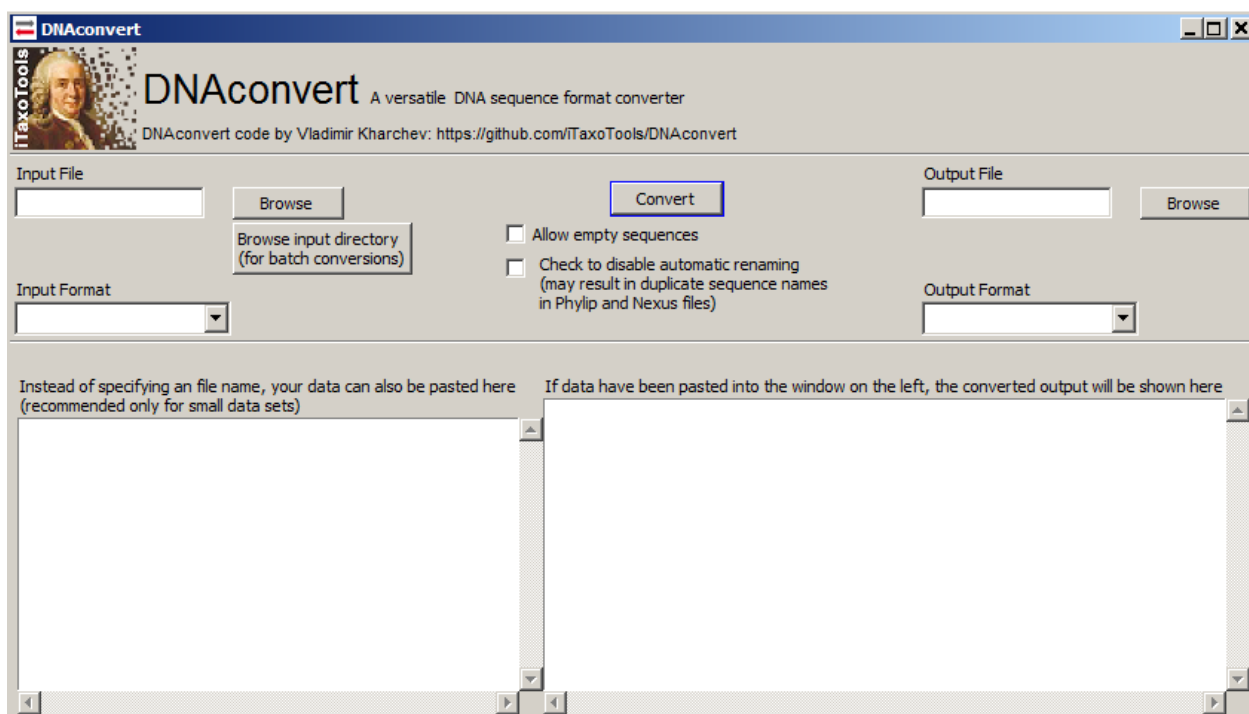
4. Tools for Data Preparation

4.1. DNAconvert

DNAconvert is a versatile tool converting among different sequence formats. It converts molecular sequence files sequentially and therefore can deal with very large files, such as fastq files of several GB. The program has mostly been written for DNA sequences but at least some of the conversions should also work for protein (amino acid) sequence files.

The tool can be run specifying an input file and a name for the output file. In this case, if the file extension is unambiguous (e.g., .fas, .nex) there is no need to specify the format which is automatically interpreted by the program. However, with ambiguous extensions (e.g., .txt or no extension), or to override an equivocal extension, input and output formats need to be specified with the pull-down menus

Alternatively the input data can be pasted into the field in the lower left; in this case, the input and output formats must be specified, and the output will be displayed in the field at the lower right from where it can be examined and copy-pasted into any other text editor.



The program is executed by the "Convert" button.

Two options can be selected with checkboxes:

Firstly, the option "Allow empty sequences" will also translate empty sequences, that is, sequence names followed by no sequence will be included in the converted file. By default such empty sequences will be ignored and not included in the converted file.

Secondly, by default the program will apply an automated renaming of sequences when these are converted into formats such as Philip or Nexus where a limit to the numbers of characters in sequence names are applied. Specifically, long sequence names will be cut to a certain length, and the last characters will be replaced by a continuous numbering. This guarantees unique sequence names which are required by many analysis programs, but may lead to "mutilation" of sequence names that are to be conserved. The respective checkbox disables the automated sequence renaming.

The following sequence formats are implemented in the program:

A central format is a **tab-delimited file**. This is an unusual sequence format but has the advantage that it allows to store and organize sequences in spreadsheet editors such as Microsoft Excel. Tab-delimited files can also easily be exported from databases, and have the advantage over CSV files that separator use is unambiguous (CSV files as a standard are comma separated, but in German, French, Spanish and other languages are often semicolon-separated which can cause parsing errors).

seqid	specimen_voucher	organism	sequence
1234	ZCMV001	Mantella aurantiaca	ATTGGAAATAATCTTGACAATGAATCTGAGGGG
ABCDEF	ZCMV002	Mantella aurantiaca	ATTGGAAATAATCTTGACAATGAATCTGAGGGG
specimen3	FGZC 30142	Mantella aurantiaca	ATTGGAAATAATCTTGACAATGAATCTGAGGGG
specimen4	FGZC 30143	Mantella aurantiaca	ATTGGAAATAATCTTGACAATGAATCTGAGGGG
1 2 3 4	ZSM 322/2007	Mantella aurantiaca	ATTGGAAATAATCTTGACAATGAATCTGAGGGG
specimen6	ZSM 323/2007	Mantella aurantiaca	ATTGGAAATAATCTTGACAATGAATCTGAGGGG
specimen7	ZSM 324/2007	Mantella crocea	ATTGGAAATAATCTTGACAATGAATCTGAGGGG
specimen8	MNHN 1991.344	Mantella crocea	ATTGGGAGCAATCTTGACAATGAATCTGAGGGG
specimen9	MNHN 1991.345	Mantella cowani	ATTGGAAATAATCTTGACAATGAATCTGAGGGG

Each column (field) must contain a header in the uppermost row. The vocabulary used for the headers in tab-delimited files in iTaxoTools follows as much as possible the syntax used in DarwinCore, ABGD and/or NCBI Genbank. Headers are largely case-insensitive and robust against common misspellings.

The tab-delimited format requires a field with unique sequence identifier called "seqid". This field will not be used further but will be crucial in future (database) extensions of iTaxoTools. If seqid is missing or values are repeated, DNAconvert will issue an error message but will proceed with conversion.

The sequence is in a further column, with the header "sequence". If a column "sequence" is missing, the program will interpret any other column containing "sequence" in the title (such as "COI sequence") as equivalent but will issue a warning.

All fields inbetween seqid and sequence will be considered as part of the sequence name, and will be concatenated to form the sequence name in other formats. If no such fields are present, then seqid will be used as sequence name in the converted file.

Because many analysis programs do not run with sequence names containing special characters, during the conversion process all characters other than the following will be converted to underscores: 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ_abcdefghijklmnopqrstuvwxyz

When converting the tab-delimited file above into fasta format, the result will thus be as follows:

```
>ZCMV001_Mantella_aurantiaca↓
ATTGGAAATAATCTTGTAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACTCACC CGATT TTTTACATTCCACTTTA
>ZCMV002_Mantella_aurantiaca↓
ATTGGAAATAATCTTGTAATGAATCTGAGGGGGATTCTggCGGTAGACAACGCAACTCACC CGATT TTTTACATTCCACTT
>FGZC_30142_Mantella_aurantiaca↓
ATAATCTTGTAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACTCACC CGATT TTTTACATTCCACTTTATCTTACC
>FGZC_30143_Mantella_aurantiaca↓
ATTGGAAATAATCTTGTAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACTCACC CGATT TTTTACATTCCACTTTA
>ZSM_322_2007_Mantella_aurantiaca↓
ATTGGAAATAATCTTGTAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACTCACC CGATT TTTTACATTCCACTTTA
>ZSM_323_2007_Mantella_aurantiaca↓
GGTTTCGTATTAAATTAGGAGCCCTCGCTGCTTCTACCTTCTCCCTAATCTTCTAGGAGATCCAGACAATTTTACCCAG
>ZSM_324_2007_Mantella_crocea↓
ATTGGAAATAATCTTGTAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACTCACC CGATT TTTTACATTCCACTTTA
>MNHN_1991_344_Mantella_crocea↓
ATTGGGAGCAATCTTGTAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACTTACCCGATT TTTTACATTCCATT TTA
>MNHN_1991_345_Mantella_cowani↓
ATTGGAAATAATCTTGTAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACTCACC CGATT TTTTACATTCCACTTTA
```

Additional standard sequence formats implemented are: fasta, fastq, phylip, relaxed phylip, genbank, and genbank-export:

fasta is a very simple format that does not require further explanation. When converting fasta into tab, the sequence is placed in the field "seqid".

fastq is a derivative of the fasta format that contains also information on the quality/reliability of the sequence as determined by specific instruments. DNAconvert only transform from fastq into other formats. For this, the first line of the fastq format (starting with @) will be considered as seqid and used as sequence name. DNAconvert can deal with very large fastq files and transform them into fasta, even if the process may take very long.

fasta_gbexport is a customized derivative of fasta that contains some specific information needed when uploading newly obtained sequences to the Genbank repository (GB). For this format, only conversion from and to tab files is reliably implemented. The standard source modifiers used by Genbank (<https://www.ncbi.nlm.nih.gov/WebSub/html/help/genbank-source-table.html>) are recognized and if present in the original tab-delimited file, they are transformed into a square-bracket format that will be recognized in the gb-sub or BankIt submission system of Genbank when uploading sequences. Recognized terms are: organism, mol_type, altitude, bio_material, cell_line, cell_type, chromosome, citation, clone, clone_lib, collected_by, collection_date, country, cultivar, culture_collectiondb_xref, dev_stage, ecotype, environmental_samplefocus, germlinehaplogroup, haplotype, host, identified_by, isolate, isolation_source, lab_host, lat_lon, macronuclearmap, mating_type, metagenome_source, note, organelle, PCR_primersplasmid, pop_variant, proviralrearrangedsegment, serotype, serovar, sex, specimen_voucherstrain, sub_clone, submitter_seqid, sub_species, sub_strain, tissue_lib, tissue_type, transgenictype_material, variety.

For instance the tab-delimited table

seqid	organism	cataloguenumber	mol_type	specimen-voucher	sequence
gehringi_MSZC128	Calumma gehringi	ZCMV3456	Genomic DNA	MSZC 128	AGGAAGCATTAAACCAACACAA
gehringi_MSZC129	Calumma gehringi	ZCMV3457	Genomic DNA	MSZC 129	AGGTTGCATTAAACCAACACAA

becomes

```
>gehringi_MSZC128 [organism=Calumma gehringi] [mol_type=Genomic DNA] [specimen-voucher=MSZC 128]  
AGGAAGCATTAAACCAACACAAC  
>gehringi_MSZC129 [organism=Calumma gehringi] [mol_type=Genomic DNA] [specimen-voucher=MSZC 129]  
AGGTTGCATTAAACCAACACAAC
```

This file can be uploaded during the sequence submission process.

The program also performs a number of additional autocorrections and checks. It corrects spelling mistakes such as `specimen_voucher` and `specimenvoucher` (corrected to `specimen-voucher` as required by Genbank), checks if sequences are <200 bp (won't be accepted by Genbank), shortens seqid to 25 characters at most, cuts terminal gaps and missing data symbols from sequences, checks if required minimum sequence information is provided, and others.

nexus is a well-established and very complex syntax used for encoding phylogenetic information for analysis. DNAconvert implements an own nexus parser and alternatively (currently still disabled) can also make use of the <https://github.com/dlce-eva/python-nexus> library. DNAconvert is able to read interleaved and non-interlaved nexus files, but writes only non-interleaved files for now.

The nexus format requires all sequences being of similar length (aligned). The program checks this before conversion; if sequences are of unequal length, they are filled with terminal dashes to make them equal, and a warning is issued. The program limits sequence names to 100 characters; longer sequence names are automatically shortened.

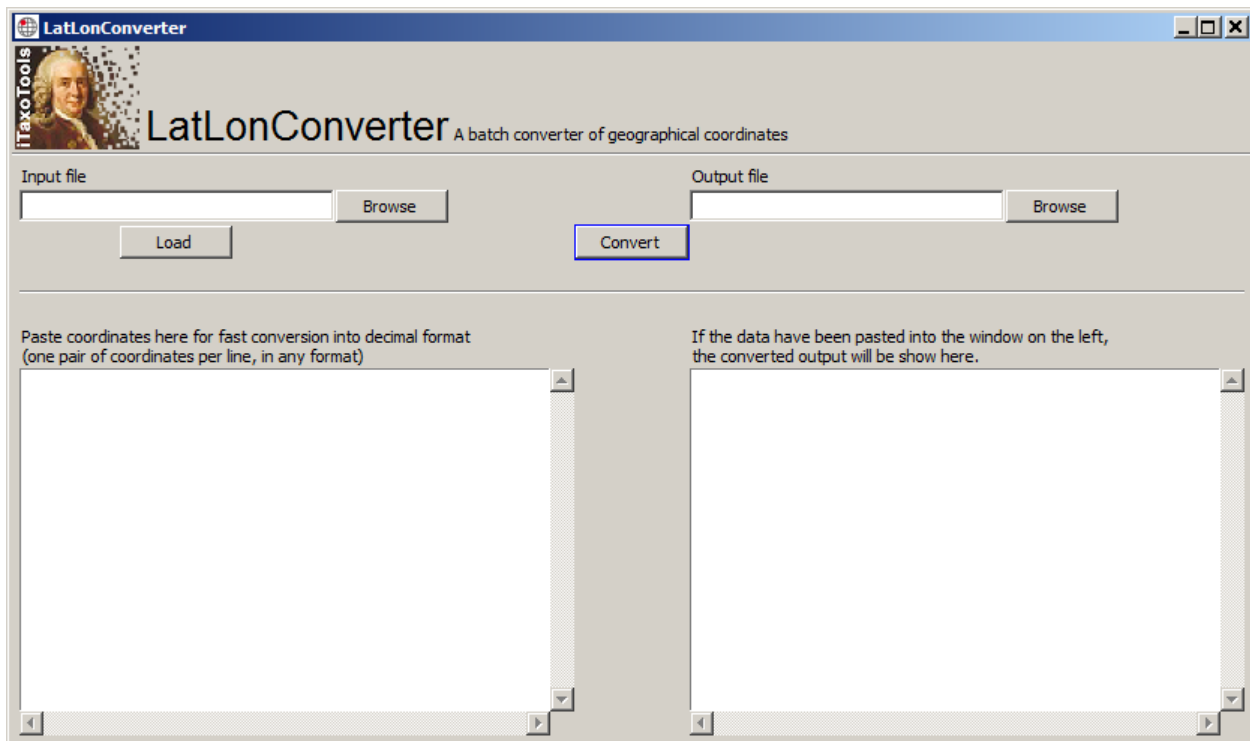
phylip is another standard format for phylogenetic analysis. In genuine phylip, sequence names must be exactly 10 characters long. DNAconvert automatically abbreviates longer sequence names and makes them unique by replacing the last characters by numbers. Sequences in phylip needs to be aligned; if unaligned sequences (sequences of different length) are detected, equal length is enforced by adding terminal dashes and a warning is issued.

relaxed phylip is identical to phylip but without a specific length restriction for species names.

genbank is the format of sequence flatfiles that can be downloaded from NCBI Genbank. Besides the sequence itself, these files contain information on source modifiers, gene identity and submitters. DNAconvert converts Genbank files into tab files, parsing all source modifiers into separate fields (columns) and also parsing information on the gene, authors, publication and others. DNAconvert is also able to convert genbank files into other formats such as fasta, but we recommend first converting into tab format, then editing the file in a spreadsheet editor, and then converting into other sequences formats. Note that some Genbank flatfiles have variation in their format and are not parsed correctly, but this does usually not apply to many sequences.

4.2 latlonconverter

This tool has the goal to provide a versatile conversion of different formats of geographical coordinates into each other, in particular, into the decimal format which is used by most analytical algorithms and geographical information systems.



Overall, there are different ways to write geographical coordinates, and in addition, for each of them a lot of small variants and erroneous ways of writing them are used by non-experts. This makes it very time-consuming to convert them, often with a lot of manual work.

latlonconverter works in two steps: First the program identifies the most common sources of misspelling or variants of coordinate formats, such as different separators between latitude and longitude (space, tabulator, comma, semicolon), different characters used for degrees, minutes and seconds, or different characters to indicate east, west, north and south (e.g., in German, east is abbreviated "O" from Ost, in Spanish west is abbreviated "O" from "oeste" - with equivocal use of these two characters, for instance, the program returns a warning message).

Second, the program then converts the coordinates into two commonly used formats and outputs the variants in a separate output file.

Input and output files are in the tab-separated text format. A typical input file provides the data as in the following example. The fields "country", "realm", "locality", "species" are not required and will not be used by the converter, but will be included unmodified in the output file. The program will read the fields "lat", "lon" and "latlon".

specimenid	species	realm	country	locality	lat	lon	latlon
ZCMV1234	Mantella aurantiaca	marine	Andasibe	Andasibe	52.28030	10.54879	
ZCMV1236	Mantella aurantiaca	terrestrial	Andasibe	Andasibe	52.36335°N	10.21979°E	
ZCMV1235	Mantella aurantiaca	marine	Andasibe	Andasibe			52,28130N 10,78873E
ZCMV1237	Mantella aurantiaca	terrestrial	Andasibe	Andasibe			52.28030, 10.54879
ZCMV1238	Mantella aurantiaca	terrestrial	Andasibe	Andasibe	52°16'49.1"N	10°32'55.3"E	
ZCMV2345	Mantella crocea	terrestrial	Fierenana	Fierenana			52°16'49.1"N; 10°32'55.3"E
ZCMV3457	Mantella crocea	terrestrial	Fierenana	Fierenana			52°24.35'N, 10°38.9678'E
ZCMV3456	Mantella crocea	terrestrial	Fierenana	Fierenana	52°16'49.375"N	10°34'58.1"E	

The output file will include the original coordinate information, as well as latitude and longitude both in decimal and standardized sexagesimal format.

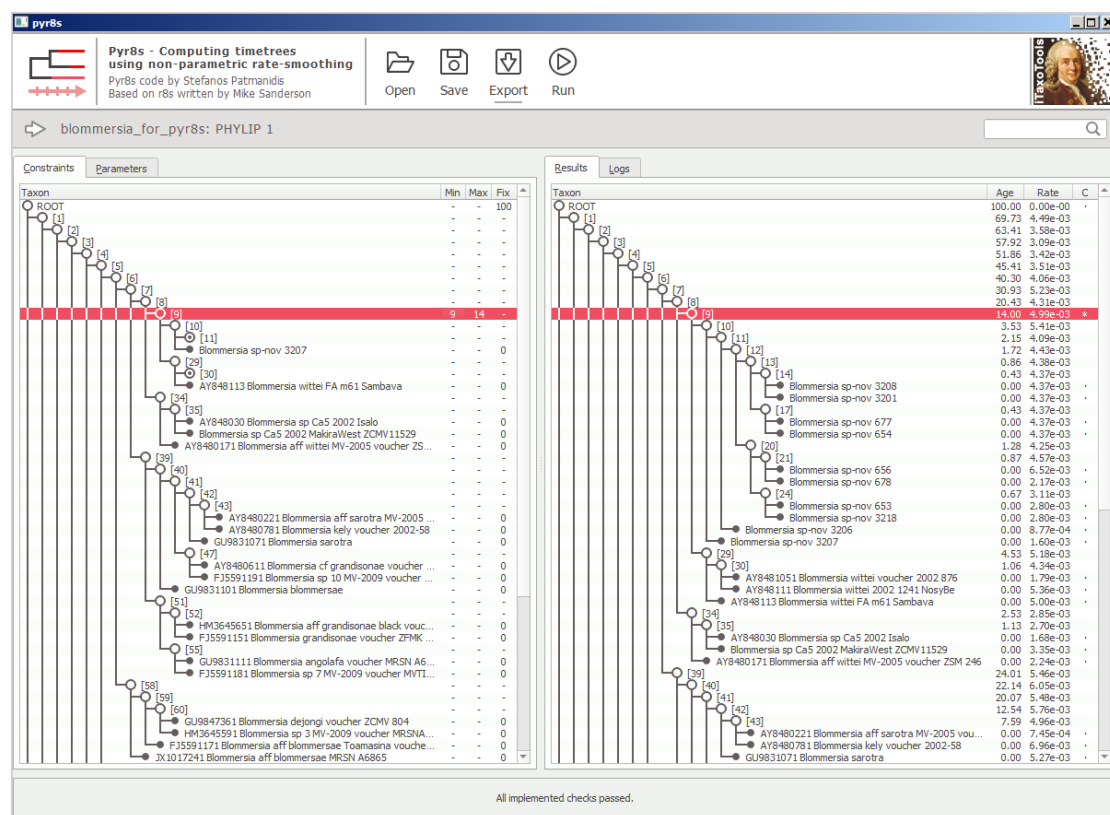
As with DNAconvert, coordinates can also be pasted in the respective boxes and will automatically converted into decimal coordinates that can be then become available in the second box. In this case, only latitude and longitude (one value per row) should be used as input, without any metadata.

4.3. pyr8s

pyr8s is a tool to transform phylogenetic trees with branch lengths being proportional to the number of substitutions (phylograms) into ultrametric trees (timetrees), with the option of applying time calibrations to internal nodes. The implemented algorithm (non-parametric rate smoothing) achieves this transformation by adjusting branch lengths using only the phylogram as input, without referring back to the original (sequence) data used to construct the tree as is required by other programs such as MCMCtree, BEAST, or MEGAX (Yang & Rannala 2006; Bouckaert et al. 2014; Kumar et al. 2018). The resulting ultrametric trees are useful for many evolutionary analyses as well as for species delimitation using GMYC.

The non-parametric rate smoothing (NPRS) approach was developed by Sanderson (1997) and later implemented as part of the program r8s (Sanderson 2003). It has previously been implemented in the R package ape (Paradis et al. 2004), but was removed from the latter and from the newest releases of r8s due to licensing issues. The original version of r8s relied on a modified implementation of Powell's conjugate direction method which was incompatible with open-source licensing (Powell 1964; Gill et al. 1981; Press et al. 1992).

In the GUI-driven tool pyr8s, the NPRS algorithm has been newly coded, making use of the open-source libraries DendroPy (Sukumaran & Holder 2010) and SciPy (Virtanen et al. 2020), thus resolving the previous licensing issues. This new version provides a GUI for user-friendly setting of time constraints, exposes a Python interface for lower-level analysis and maintains support for r8s-formatted input files.



The design of the pyr8s GUI is similar to various other tools in the iTaxoTools distribution. The upper bar of the GUI features four buttons: Open, to open the input file (a Phylip/Newick-formatted

phylogram); Run, to run the program once all parameters have been set. Save, to save the Results session of the analyses; and Export, to export the ultrametric tree (chronogram), a "ratogram" or a table with results.

The left box of the GUI allows to set interactively upper and/or lower time constraints for all nodes in the tree, and in a second tab, adjust multiple parameters.

The right box displays the progress of the analysis which remains available on the Log tab, and shows the resulting tree with age estimates for node after completion of the analysis.

4.4. fastmerge and fastsplit

fastmerge and fastsplit are two very simple tools to handle large sequence files. The tools have been written specifically for fasta and fastq files but in principle can also work with other sequence formats, and indeed with any other kind of text file, although several functionalities will then not be applicable.

The purpose of fastsplit is to split large sequence files into smaller portions. This can be useful when handling high-throughput sequencing data which often are provided in files of several Gigabyte (GB) and thus can be difficult to handle. For instance, some storage media can only store files up to 2 GB, and with slow internet connection it can also be advisable to transfer large files in smaller portions.

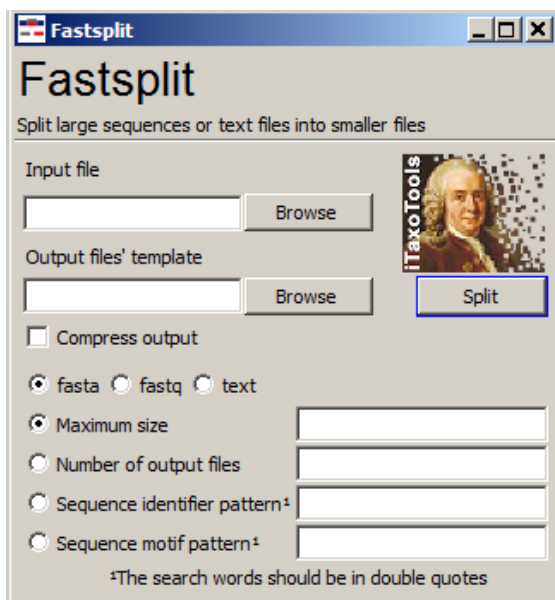
To use fastsplit, the user has to specify the input file and a template for the desired output files - the program will then produce several files with this name, and an automatic numbering included in the file names.

Input and output files can be (gz) compressed.

Splitting can be guided by either specifying a maximum size for each out file, or specifying the total number of output files (which then will be of almost equal size).

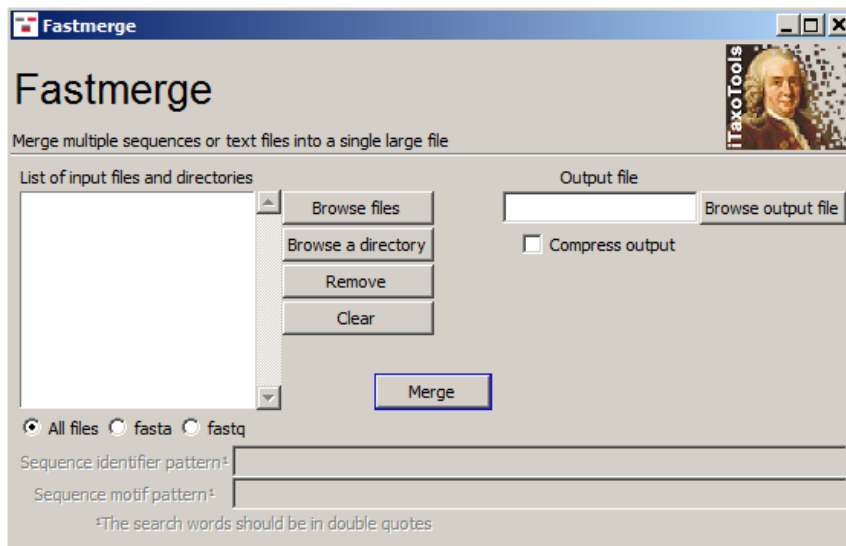
If fasta or fastq file types are specified, fastsplit will make sure to avoid splitting inbetween sequences, i.e., each file will always start with the sequence name (which is preceded by > in fasta, and @ in fastq).

Lastly, the program also allows to specify either a string of characters in the sequence name, or a motif of base pairs in the sequences. These need to be specified in double quotes, such as "sapiens" or "ACAAAGT". In this case, fastsplit will only include in the output files those sequences containing the requested motif - however, this function can take a long time with very large sequence files, and is not guaranteed to work, especially with complex fastq formats which can differ depending on the sequence platform generating them.



The function of fastmerge is basically the reverse of fastsplit. The program allows specifying a series of input files, and these will be merged into one single file for which name and folder need to be specified. The output can be (gz) compressed.

As with fastsplit, fastmerge allows specifying a string of characters in the sequence names, or a sequence motif, and will only add sequences complying with the request into the merge files - but also here, this can be a long-lasting process which might not work perfectly with idiosyncratic variations of the input files.

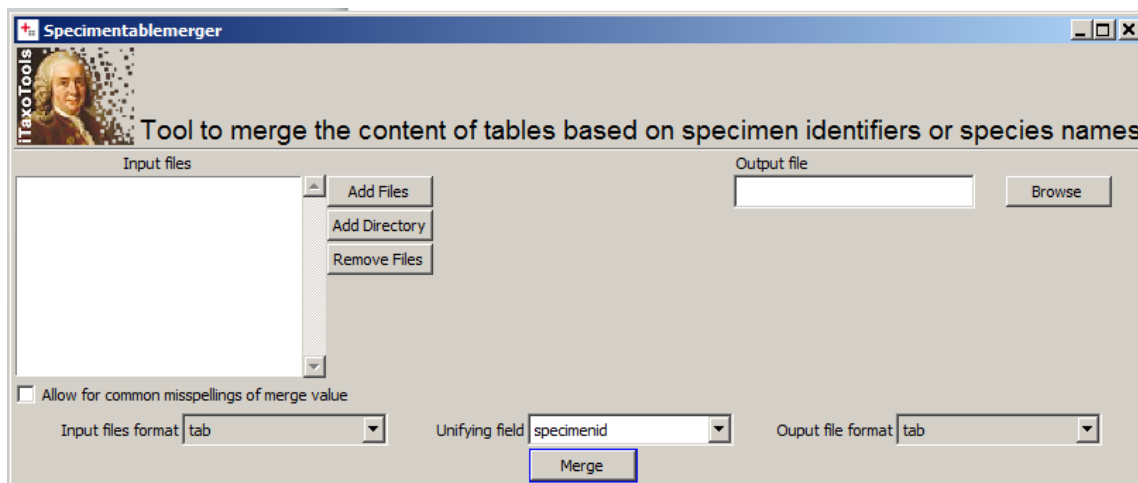


4.5. specimentablemerger

Specimens are at the center of alpha-taxonomic research, and in our experience, many taxonomists manage specimen data for research purposes in spreadsheet editors such as Excel. Obviously, large-scale specimen data should be managed in databases, but for typical research purposes, subsets of data will be organized, managed, summarized, and explored using spreadsheets. Different kinds of data will however be present in different spreadsheets, for instance, one spreadsheet with DNA sequences, one with morphometric data, and one with ecological data recorded for each specimen in the field. While it is possible to merge such data sets, again, using database programs if the specimen identifiers used are the same in each spreadsheet, for many applications and especially, for exploratory and preliminary analyses, it will be useful to apply such merging to spreadsheets themselves.

The tool `specimentablemerger` implements this function. It works with tables where each row corresponds to a set of values of one specimen, by selecting as input various input files (tab-delimited text, or alternatively, comma or semicolon delimited CSV files), specifying one of several pre-defined fields for merging (mostly "specimenid"), and merging all of these into a new spreadsheet where from all original tables, values with the same specimenid will be placed in one row.

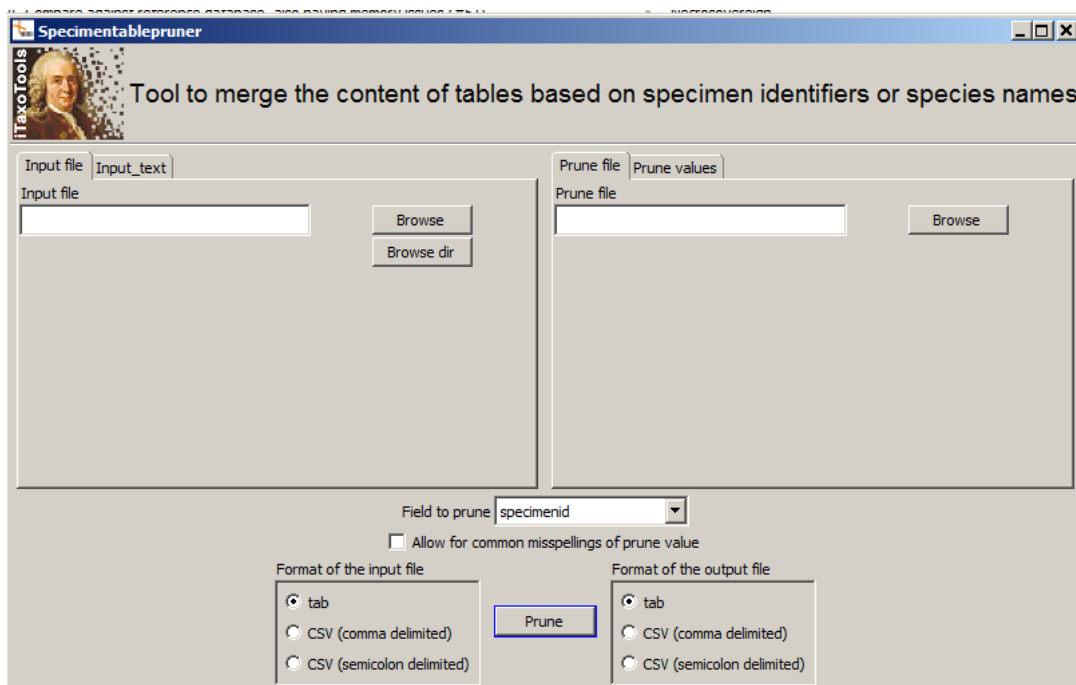
The program also issues in a separate field (column) warnings if non-coinciding values for the same field and the same specimenid are included in different tables, and includes one option (checkbox at the lower left) to autocorrect common misspellings of specimen identifiers. For instance, if the catalogue number MNHN-IM-2013-16138 is used as identifier for one specimen, this option will consider as identical specimenids MNHNIM201316138, MNHN_IM_2013-16138, MNHN IM-2013-16138 and similar variants.



4.6. specimentablepruner

Similar to the previous one, this tool performs important modifications on tables that contain values of different variables per specimen (where every specimen represents one row). The tool takes as input one tab-delimited table (or comma/semicolon delimited CSV), plus a series of specimen identifier values.

Also in this tool, an autocorrect option to correct common misspellings of specimen identifiers can be activated.

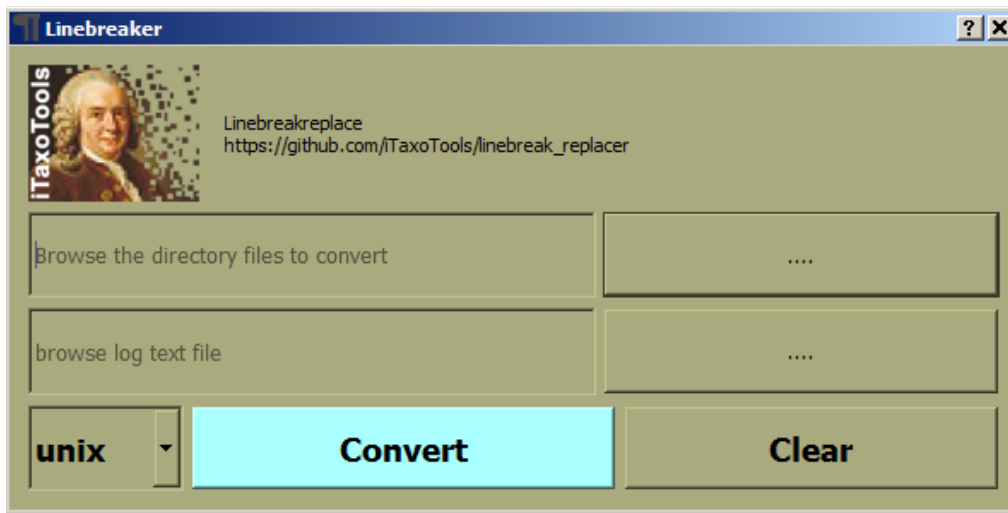


The screenshot shows the 'Specimentablepruner' application window. The title bar reads 'Specimentablepruner'. The main window has a header with the iTaxoTools logo and the text 'Tool to merge the content of tables based on specimen identifiers or species names'. Below the header, there are two main sections: 'Input file' and 'Prune file'. The 'Input file' section has tabs for 'Input file' and 'Input_text', with the 'Input file' tab selected. It contains a text box for the input file, a 'Browse' button, and a 'Browse dir' button. The 'Prune file' section has tabs for 'Prune file' and 'Prune values', with the 'Prune file' tab selected. It contains a text box for the prune file and a 'Browse' button. Below these sections, there is a 'Field to prune' dropdown menu set to 'specimenid'. A checkbox labeled 'Allow for common misspellings of prune value' is present. At the bottom, there are two groups of radio buttons for 'Format of the input file' and 'Format of the output file', both with 'tab' selected. A 'Prune' button is located between these two groups.

4.7. linebreaker

Often, when software tools do not accept particular input files, this is not due to errors in the syntax or structure of the files, but is caused by the wrong kind of line break. Windows and old DOS terminate lines a combination of a CR and a LF character whereas UNIX (Including Linux) uses a LF character only. Current Mac operating systems (OS X) also use a single LF character, but the classic Mac OS used a single CR character for line breaks. While many recent text and code editors, and phylogenetic programs, are robust against variation in line breaks, this is not the case in many vintage and exotic programs.

To deal with such issues, Linebreaker is a small and very simple utility that takes as input a text file and converts within the file all line breaks into one of the aforementioned formats.



4.8. nodenamecorrector

Some tree editors (programs to open and visualize phylogenetic trees in the Newick/Phylip format) have problems if leaf names (taxon names) contain special characters. The very small and simple tool nodenamecorrector in some cases can alleviate this problem by "repairing" tree nodes. The program takes as input a Newick treefile and replaces all special characters in the leaf names by underscores. However, given the high variation in details of the Newick-format used by different programs, the program may not in all cases be successful in this purpose.

4.9. unitconverter

This is a simple tool that does not require much explanation. It features various tabs for molarity, distance, volume, weight and time. Each tab shows fields for a large number of different units of the respective category. The user enters a value in one of the fields, and all other fields will automatically and in real time display the converted value.

If values should be consistently displayed as scientific numbers, the respective checkbox can be selected (e.g., 1.00e-03 instead of 0.001). After clicking the checkbox, the value needs to be entered newly for changes to take place.

The screenshot shows a web browser window titled "Unit Converter". The main heading is "Unit Converter Tools". In the top right corner, there is a small portrait of a historical figure and the text "iTaxoTools". Below the heading, there is a checkbox labeled "Click here for Scientific numbers". A horizontal tab bar contains five tabs: "Molarity", "distance", "volume", "weight", and "time". The "distance" tab is currently selected. The interface displays two columns of units with corresponding input fields. The left column lists metric units, and the right column lists imperial and nautical units. The values shown in the fields are as follows:

Unit	Value
meter	1
kilometer	0.001
centimeter	100.0
millimeter	1000.0
micrometer	1000000.0
micron	1000000.0
nanometer	1000000000.0
picometer	1000000000000.0
decimeter	10.0
nautical league	0.0001799856
nautical mile	0.0005399568
inch	39.370079
yard	1.093613
foot	3.28084
league	0.0002071237
mile	0.0006213712
light year	1.06e-14

5. Tools for Data Analysis

5.1. TaxI2

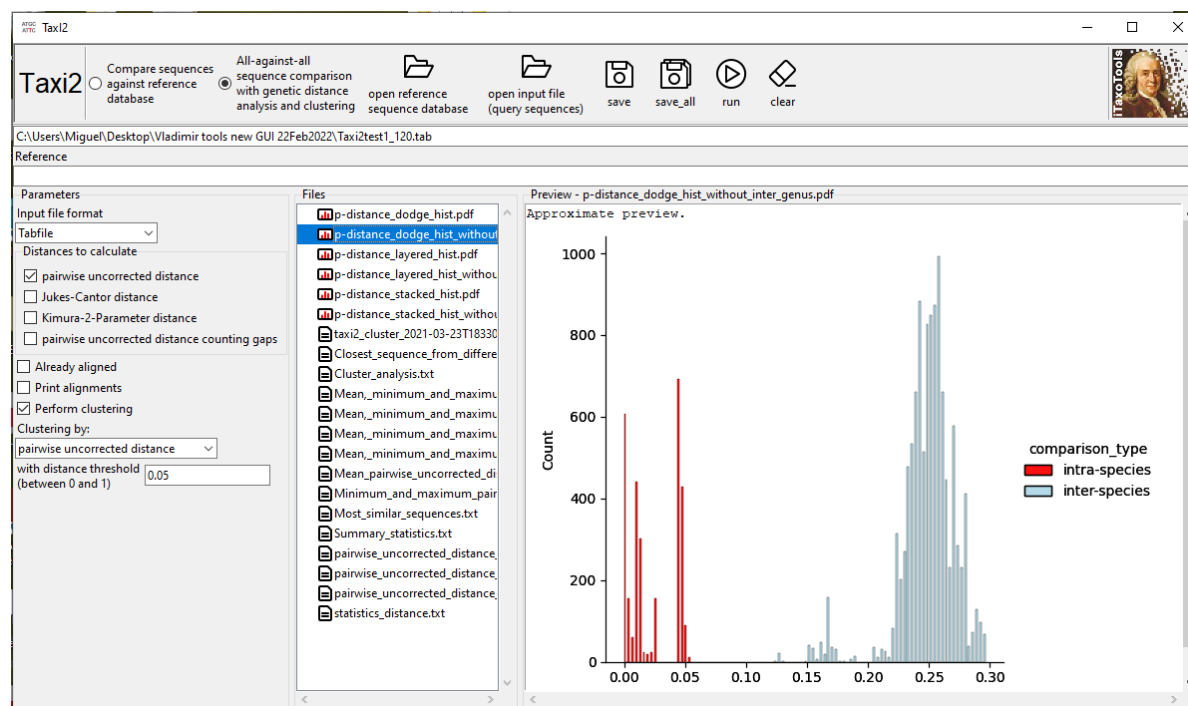
TaxI2 is a tool for pairwise sequence comparison, usually to be used in the framework of DNA barcoding. To analyze DNA barcoding data sets, Steinke et al. (2005) proposed the program TaxI (from Taxon Identification), which performs pairwise alignments between query sequences and a reference sequence database, and calculates pairwise distances based on these alignments. The authors argued that compared to a multiple sequence alignment (MSA) these distance calculations may be more accurate in the case of highly divergent markers including multiple insertions and deletions, such as stretches of mitochondrial ribosomal RNA genes.

In TaxI2 we have implemented two different approaches: (i) an all-against-all comparison approach for either unaligned sequences, or pre-MSA aligned data sets, and (ii) a reference dataset approach.

Input data can be in fasta format, but can also be in tab-delimited format with added metadata column such as species, or as Genbank flatfiles. Tab-delimited text or GB file input then allow computing various metrics, genetic distances between and within species and genera, and pairwise distributions displaying the barcode gap. These calculations can be done for different distance metrics such as uncorrected, Jukes-Cantor or Kimura-2-Parameter.

The all-against-all approach also implements a simple threshold clustering approach where samples can be clustered based on a predefined genetic distance. The approach chosen is a simple one where the full set of sequences connected to each other by any distance below the threshold are clustered (thus not depending on input order), even if violating the threshold in other comparisons within the cluster (i.e., clusters can contain sequences that are connected by distances above the threshold; cf. Meier et al. 2006). The program reports and quantifies these threshold violations. Results are provided as summary in simple text, and as matricial SPART file.

Graphs can be previewed, and will be saved as vector graphs in PDF format. They can be imported in vector-based graphic programs such as Adobe Illustrator or CorelDraw without loss of quality, and keeping all text editable.

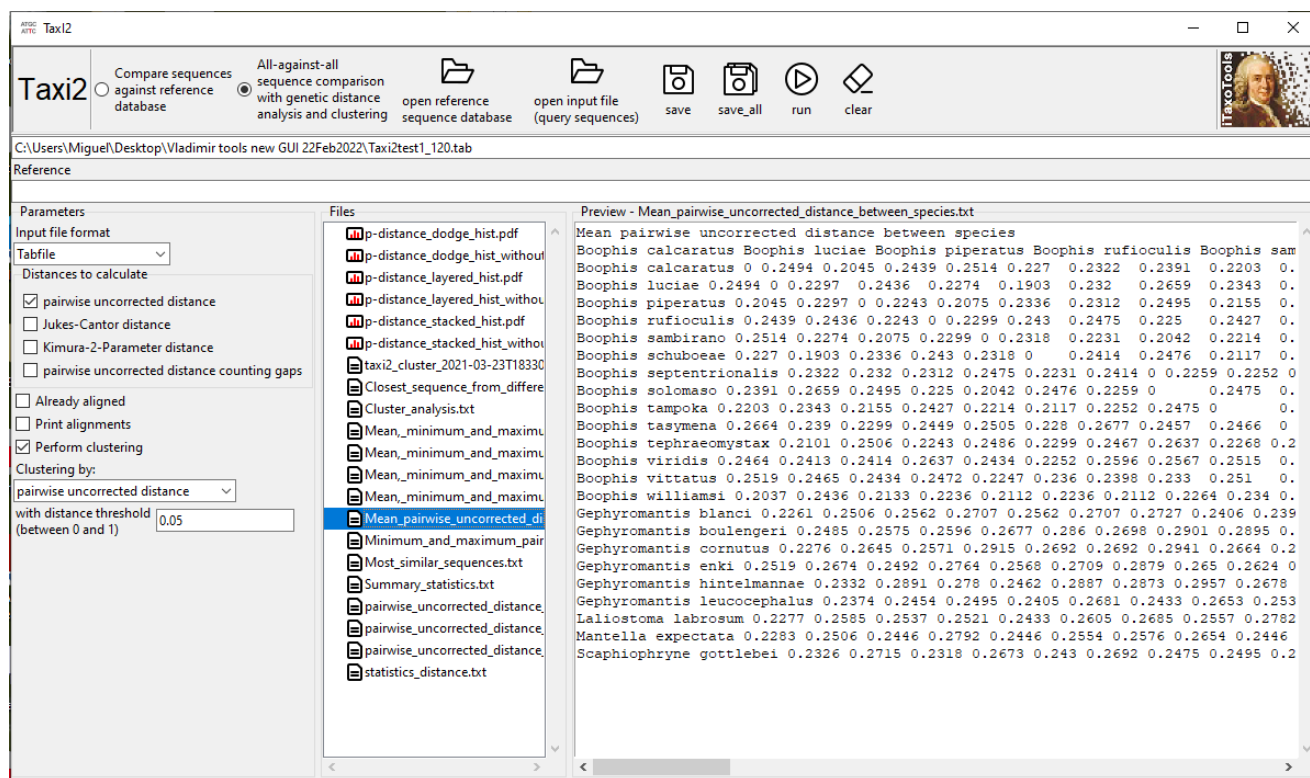


If input is given with metadata as tab-delimited file, it should contain a series of fields as in the following example:

seqid	specimen_voucher	organism	sequence
specimen1	ZCMV001	Boophis luteus	CCCTCTAAACTCTTC
specimen2	ZCMV1234	Boophis luteus	CCCTCTAAACTCATC
specimen3	FGZC7384	Boophis sandrae	CCACTAAGCTCTTC
specimen4	FGZC7385	Guibemantis liber	CCGTCAACCACTC

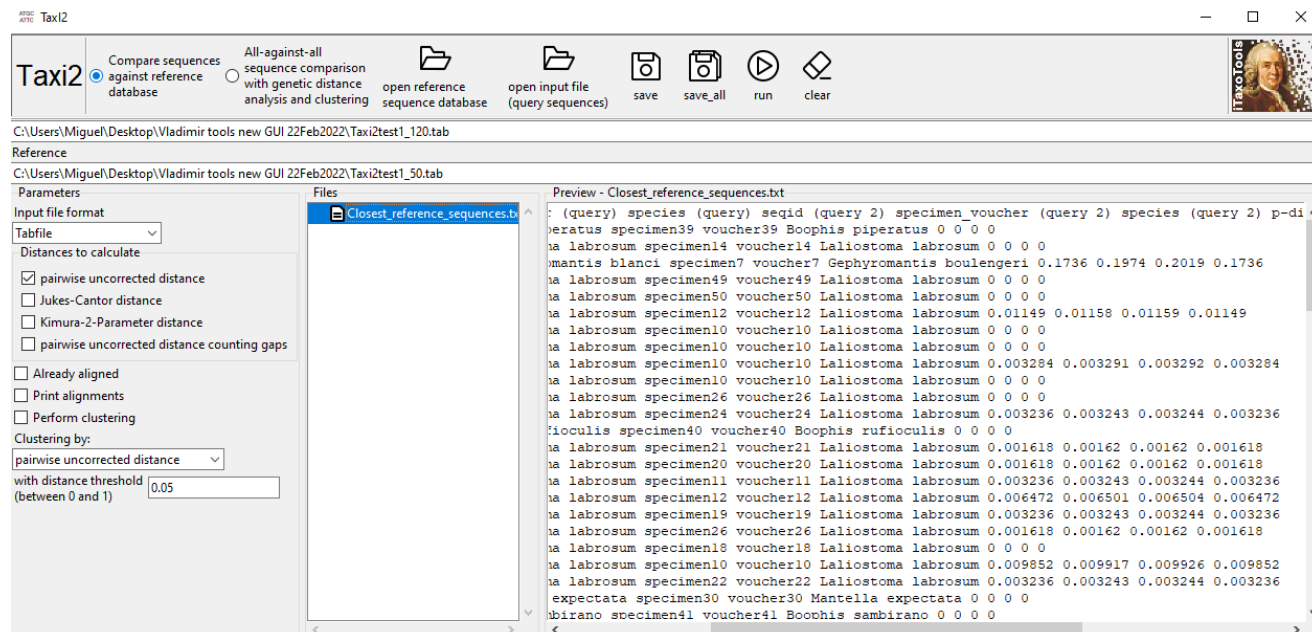
The program will use as synonyms the field name "organism" (compliant to Genbank syntax) as in the example or "species".

All tables in the output will be provided as tab-delimited text as well. They can either be copy-pasted from the preview window directly into a spreadsheet editor, or the "Save" or "Save All" buttons in the upper bar can be used to save one or all output files to a selected directory.



When comparing a file with query sequences against a reference database (the initial implementation of TaxI; Steinke et al. 2005), only one output file is produced which provides for each query sequence the most similar sequence in the reference sequence dataset, and the genetic distances to it.

For this approach, two files need to be specified - a query and a reference file with sequences.



5.2. morphometricanalyzer

This is a tool for exploratory analysis of morphometric datasets, although it would also be possible to use it to analyze other datasets of continuous variables (e.g. bioacoustic).

The program takes as input tab-delimited text files with species hypotheses (i.e., a field/column named "species" or "taxon" or "organism") and some other optional categories, as in the following example:

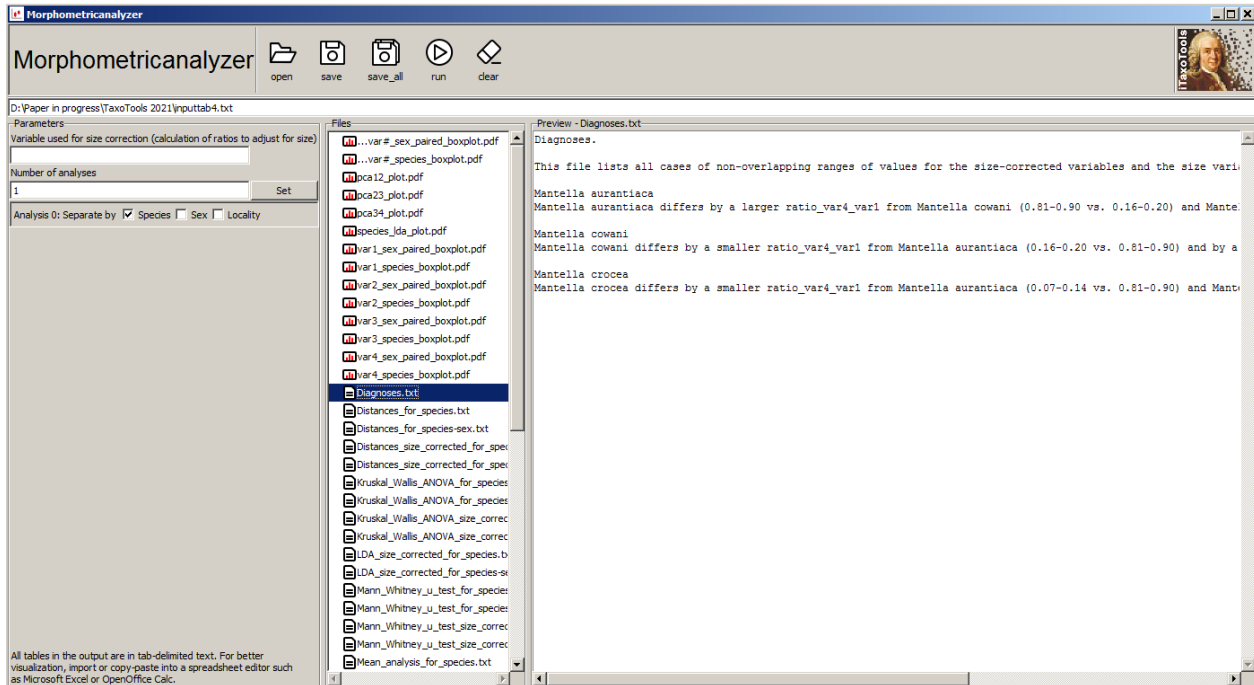
specimenid	TAXON	sex	locality	Var1	Var2	Var3	Var4
ZCMV1234	Mantella aurantiaca	M	Andasibe	203.1	34	75	182
ZCMV1235	Mantella aurantiaca	M	Andasibe	207.2	32	74	178
ZCMV1236	Mantella aurantiaca	Female	Andasibe	205.5	33	73	181
ZCMV1237	Mantella aurantiaca	M	Andasibe	206.1	3	74	182
ZCMV1238	Mantella aurantiaca	M	Andasibe	208	31	74	184
ZCMV2345	Mantella crocea	F	Fierenana	223.5	35	73	18
ZCMV3456	Mantella crocea	male	Fierenana	204.2	31	76	17
ZCMV3457	Mantella crocea	M	Fierenana	201.2	32	79	14
ZCMV3458	Mantella crocea	M	Fierenana	202.1	33	72	19
ZCMV3459	Mantella crocea	M	Fierenana	207.4	35	75	30
ZCMV3460	Mantella crocea	Male	Fierenana	204	33	76	21
FGZC9877	Mantella aurantiaca	F	Beparasy	226.9	31	41	184
ZCMV1	Mantella cowani	M	Antoetra	203.7	30	77	40
ZCMV21	Mantella cowani	M	Antoetra	210.9	31	76	41
ZCMV3	Mantella cowani	M	Antoetra	220.3	34	79	45
ZCMV4	Mantella cowani	M	Antoetra	240	29	80	39
ZCMV5	Mantella cowani	M	Antoetra	216	30	73	44
ZCMV6	Mantella cowani	M	Antoetra	221	31	72	43

As most other tools in iTaxoTools, field names are case-insensitive, and several synonyms are accepted.

The program then performs automatically a series of statistical comparisons between species (and between other categories such as sex or stage). These include:

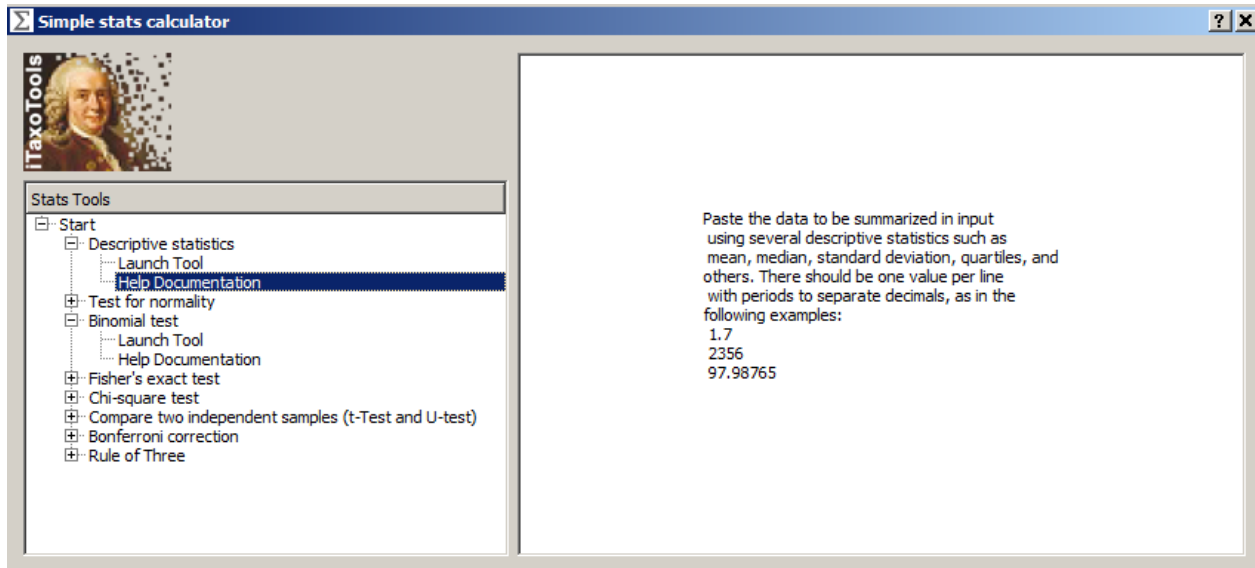
- Descriptive statistics: means, medians, standard deviation, minimum and maximum values per species or sex (or species/sex).
- Boxplots for each variable by species (or species/sex, etc) which can be saved as PDF.
- Pairwise comparisons: Mann-Whitney U-tests and Student's t-tests between all pairs of species (including Bonferroni correction for multiple comparisons).
- Comparisons among all species with ANOVA and non-parametric Kruskal-Wallis AANOVA
- A simple Principal Component analysis, including PDF-format scatterplots among the first principal components, with different symbols and colors per species.
- An exploratory Discriminant Analysis plot (PDF-format scatterplot with different symbols and colors per species)

As a final feature, the program also outputs pre-formulated taxonomic diagnoses, with full-text sentences specifying by which morphometric value or ratio a species/population differs from other species/populations with statistical significance, or without value overlap.

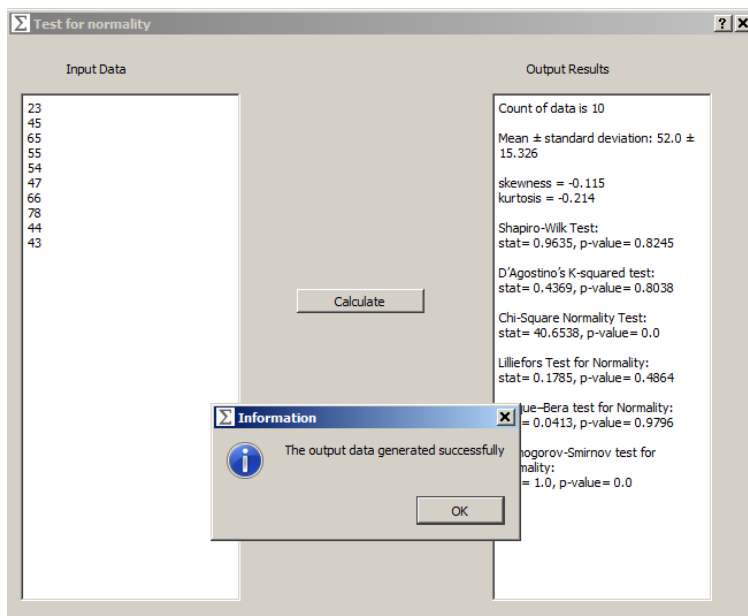


5.3. simplestatscalculator

This program features a collection of simple tools for performing ad-hoc statistical tests. The starting window has in the left box a tree-like menu of the various included tools whereas the right box will feature precise instructions and help files explaining each of the tests.



After double-clicking the "Launch Tool" option for each test, a new window opens where the values for the respective test can be manually entered or pasted.



6. Tools for Species Delimitation

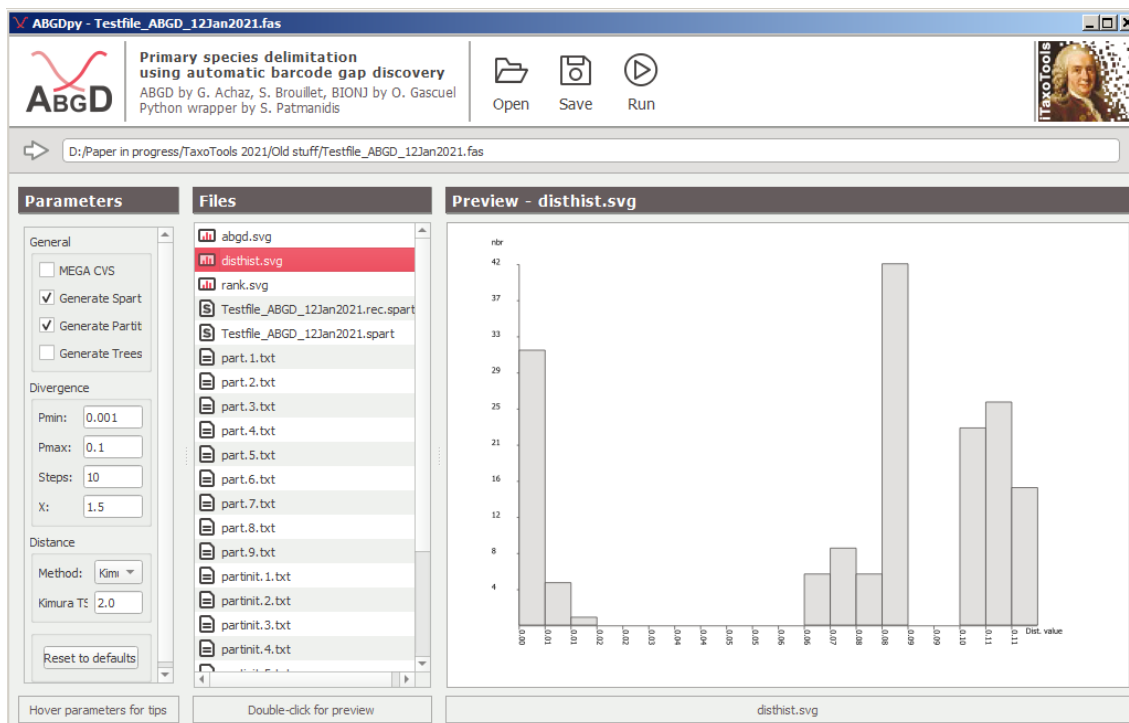
A special emphasis in the first development phase of iTaxoTools is species delimitation. Although many of the existing algorithms for species delimitation from molecular data appear to delimit populations rather than species (Sukumaran & Knowles 2017) and thus overestimate species numbers, they can serve to formulate initial species hypotheses - and these can then be tested in an integrative taxonomy pipeline.

In the field of species delimitation, we have focused on tools already available in Python programming language. For these tools, we added user-friendly GUIs and slightly extended the functionality, for example by enabling them to output species partition information in the standardized "spart" format proposed by Miralles et al. (2021).

6.1. ABGD

The Automatic Barcode Gap Discovery (ABGD) approach for primary species delimitation was developed by Puillandre et al. (2012). ABGD has been written in C language, provided as command line tool, and deployed on a webserver (<https://bioinfo.mnhn.fr/abi/public/abgd/abgdweb.html>). For iTaxoTools, we programmed a GUI in Python (ABGDpy), wrapped around the original ABGD code, to make the program more easily accessible to users, also in offline situations.

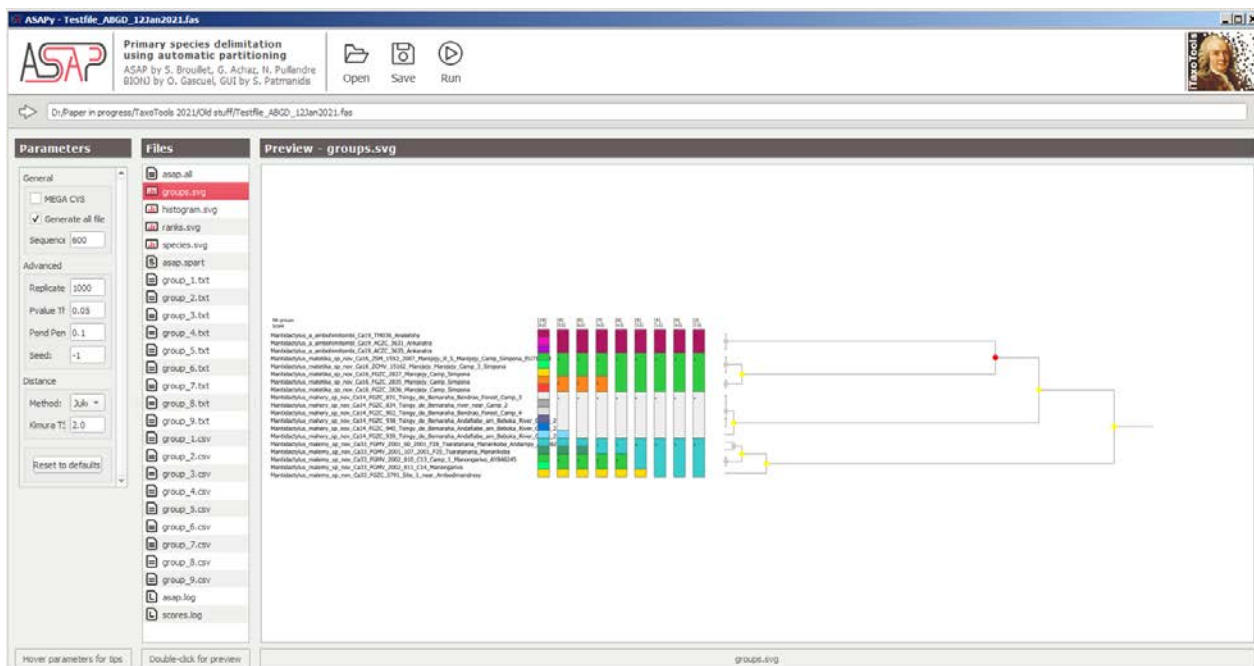
ABGD takes as input a set of aligned sequences (e.g. in fasta format) and calculates pairwise distances (or takes as input directly a matrix of pairwise genetic distances), then uses a coalescent model to identify the position of the most likely barcode gap, based on maximal genetic intraspecific distances defined a priori by the user, and uses the DNA barcoding gap to propose species partitions. Extensive documentation on the functioning of ABGD is found in the original publication (Puillandre et al. 2021) and further information on its website (see above).



6.2. ASAP

The Assemble Species by Automatic Partitioning (ASAP) approach to species delimitation has been introduced by Puillandre et al. (2021). Similar to ABGD, the original code has been written in C, and we have complemented the original code with Python-based GUI (ASAPy) to be included in iTaxoTools. The ASAP approach performs species delimitation from single-locus sequence data, proposing species partitions ranked by a new scoring system that uses no biological prior insight of intraspecific diversity.

Details of the approach are explained in Puillandre et al. (2020) and on the ASAP webserver (<https://bioinfo.mnhn.fr/abi/public/asap/>) where analyses can also be run online. Different from many other species delimitation tools, ASAP also provides a series of visualization tools such as displaying alternative species partitions on a tree built from the original sequences.



The following are some FAQ answers from ASAP's webpage:

ASAP is a tool designed to propose partitions of species hypotheses using genetic distances calculated between DNA sequences. ASAP can handle more than 10 000 sequences, but the computation time can be quite important in this case (several hours).

For input files, the fasta format is the most convenient format. If you provide a distance matrix (phylip dnadist or MEGA CSV) you must provide the length of the alignment used to compute the matrix. Please rename the ".csv" extension into ".txt" as CSV can be interpreted as special objects by some browsers and will produce unexpected results

Available options:

- **Probability:** At each step of the process, ASAP clusters objects within a same distance range into a node. An object is either a node or a specimen. A probability (*) is calculated for each node at each step of the process. If the probability of a node is below the value indicated here, then ASAP will readjust the number of putative species, splitting each node which probability value is below. The default value is 0.01.
- **Scores kept:** Number of results with the highest scores to be displayed in the table and on the curve.
- **Seed value:** ASAP makes simulations which are based on a random seed generator. If you change the seed, the probability may be slightly different at each run. (leave -1 if you don't want to use a fixed seed value).
- **Highlighted results:** You may want to visualize the partitions included in a given range of genetic distances (e.g. in the vicinity of the barcode gap). This has no impact on the ASAP results. A star will be added to the best scores belonging to this interval and a blue rectangle will be drawn on the graphic, helping you to spot these partitions.

(*) See manuscript for details

When interpreting the results:

- **Nbgroups** is the number of species as identified by ASAP in the corresponding partition. ASAP identifies different partitions, and the score is an indicator of which partition you have to look at. It is a combination between the two following parameters (probability and slope)
 - **Proba** is the probability that the partition at the step n is different from the partition at the step $n-1$. Please, refer to the publication for more details.
 - **W** is the slope of the curve shown on the right ("Ranked distances") at a given genetic distance value (see below). A high value means that the next distances (bigger and smaller) values are far.
 - **Dc** is the value of the "jump" distance used to calculate the slope.

You can tune some parameters. All the partitions within the range of genetic distances you provided will be preceded by a star and a blue area corresponding to this range will be drawn on the curve. Default values are 0.005 and 0.05.

For each partition and for each node for which a probability has been calculated, the darker the color of the dots and squares, the higher the probability. When the probability was not computed, the square is grey. We choose to use different symbols (dots and squares) for the table and the curve in one hand and for the dendrogram in the other hand, because the probability is not calculated the same way. For the table and the curve, it corresponds to the probability of the partition. For the dendrogram, it corresponds to the probability that merging the groups within the node is compatible with the known distances inside each of these groups. A very low probability (dark color) indicates that this group is unlikely, i.e. that the groups within this node probably correspond to different species. . Please refer to the publication for more details.

ASAP uses a seed to generate random partitions in order to estimate the probability of a partition. A new seed can slightly change the probabilities.

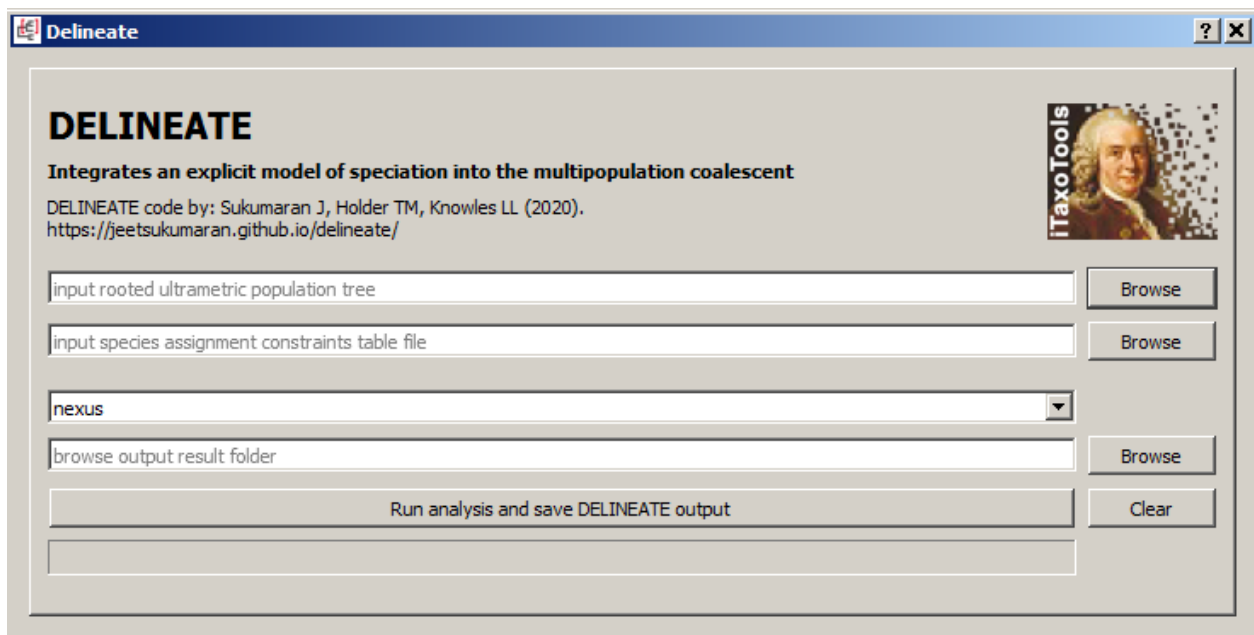
Remember that ASAP is an exploratory tool designed to identify the best partitions of species, given the criteria used by ASAP (in particular the genetic distances). Your own species hypotheses might be based on other data, methods or criteria of species delimitation, and might thus be different from the best ASAP partitions. Combining all these results in an integrative taxonomy approach is generally a good idea.

6.3. DELINEATE

DELINEATE, written by Sukumaran et al. (2020), is a tool for species delimitation that integrates an explicit model of speciation into the multipopulation coalescent. It takes as input a rooted ultrametric tree from a multispecies coalescent analysis, in Nexus or Newick format; and a second input file with a table assigning specimens to species, or flagging their species identity as unknown. The program then outputs various alternative species partitions, ranked by a probability score, in a JSON output format.

The approach of DELINEATE, along with example files, is described in great depth and detail by Jeet Sukumaran on the program's website: <https://jeetsukumaran.github.io/delineate/>

iTaxoTools added to the original DELINEATE code a GUI, plus an output in the SPART format of the most probable partition found.



6.4. GMYC

This program, introduced by Pons et al. (2006) and described in more depth by Fujisawa and Barraclough (2013) was one of the first species delimitation approaches from molecular data as an explicit bioinformatic algorithm. It is based on the Generalized Mixed Yule Coalescent and uses as input an ultrametric tree in Newick or Nexus format that should be derived from single-locus data. It then uses a likelihood approach to analyse the timing of branching events, seeking for the most likely switch between a Yule (interspecific) and a coalescent (intraspecific) branching structure.

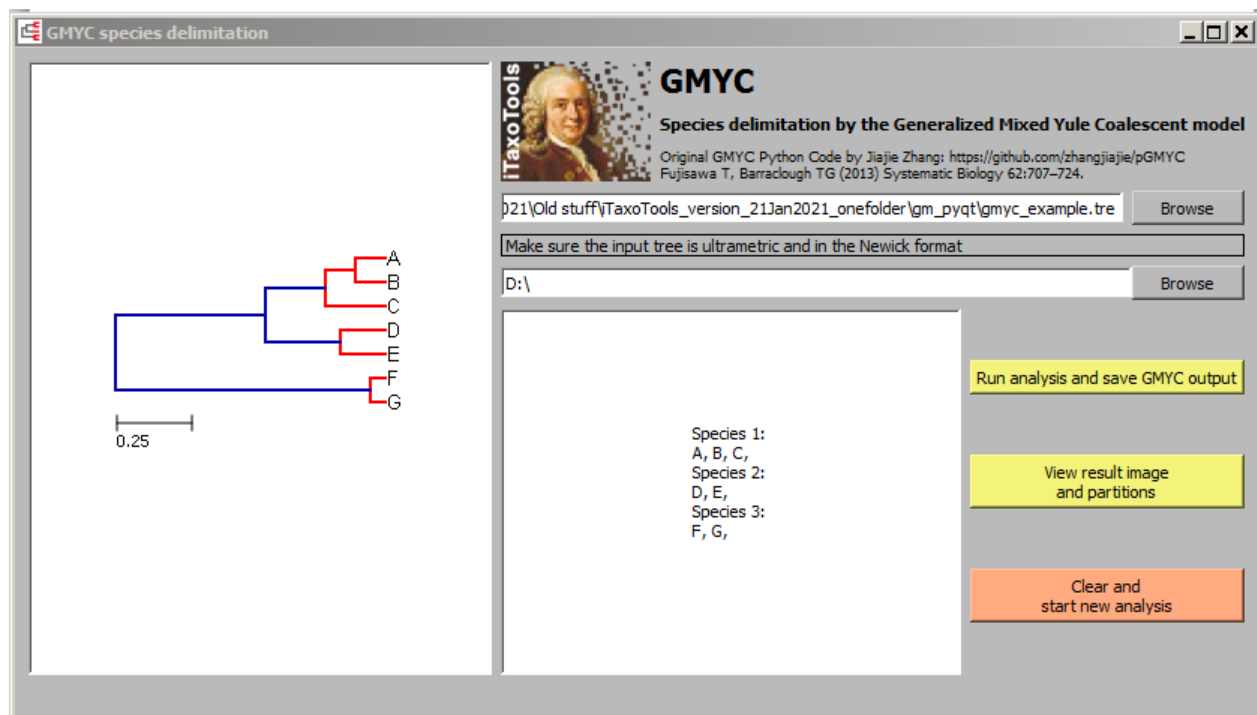
GMYC has been deployed on a webserver thanks to Jiajie Zhang at Heidelberg Institute for Theoretical Studies: <https://species.h-its.org/gmyc/>

A tutorial for GMYC has been published by Francois Michonneau: <https://francoismichonneau.net/gmyc-tutorial/>

Further information on GMYC can also be found on the website of Tomochika Fujisawa: <https://tmfujis.wordpress.com/2013/11/07/comparing-gmyc-species-with-other-delimitations/>

In order to prepare an ultrametric tree, in cases that only a regular phylogram is available, users can consider using pyr8s (part of iTaxoTools).

For iTaxoTools, we used the original code of the Python version of GMYC (written by Jiajie Zhang) and added a GUI to it. This GMYC version also outputs the resulting species partition in SPART format.

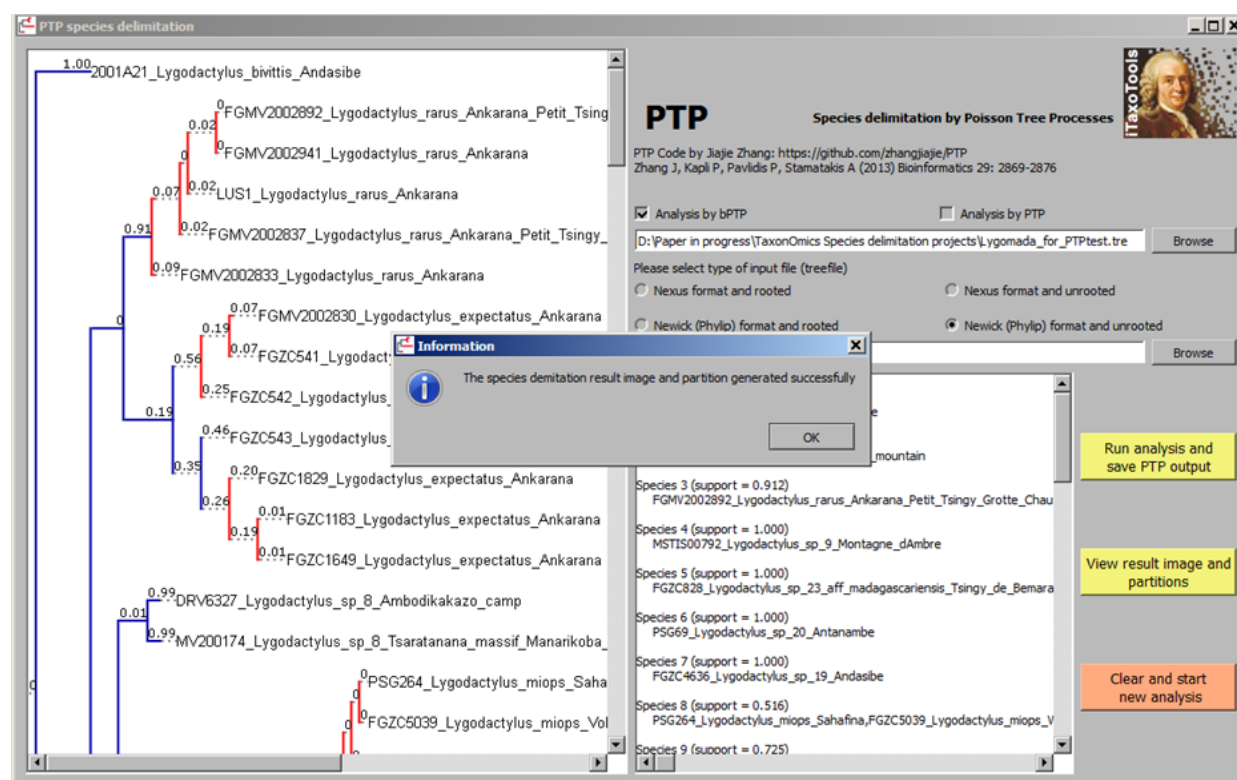


6.5. PTP

Species delimitation based on Poisson Tree Processes (PTP) was introduced by Zhang et al. (2013). Similar to GMYC, it uses a single-locus tree as input, but for PTP the tree should be non-ultrametric (a regular phylogram), in Newick or Nexus format. The approach models speciation on branching events in terms of number of mutations (inferred from branch lengths), and the Python code by Jiajie Zhang implements a Bayesian and an ML version.

A PTP webserver is implemented at <https://species.h-its.org/> and on this website, additional information on PTP can be found.

For iTaxoTools, we used the original code of the Python version of PTP and added a GUI to it. This PTP version also outputs the resulting species partition in SPART format.



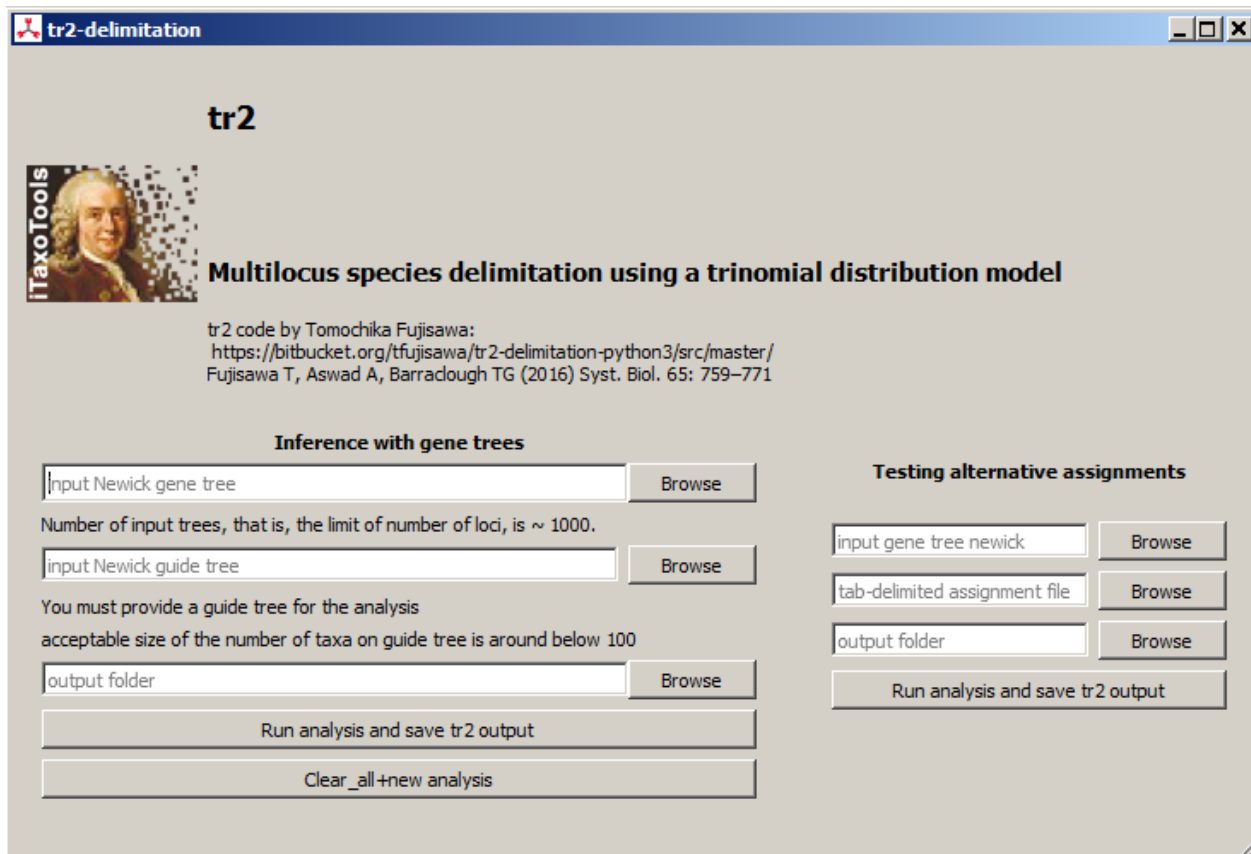
6.6. tr2

Species delimitation using Bayesian model comparison and rooted triplets (tr2) has been proposed by Fujisawa et al. (2016). This program takes as input a set of gene trees, and optionally a guide species tree, then calculates posterior probability scores for user-specified delimitation hypotheses. Alternatively, it can find the best delimitation under a guide tree specifying a hierarchical structure of species grouping.

The original tr2 implementation uses the program triplec (<http://www.cibiv.at/software/triplec/>) to construct the guide tree; however, the guide tree (species tree) can also be constructed using other approaches.

Tomochika Fujisawa provides important information on tr2 on his website (<https://tmfujis.wordpress.com/2016/10/17/multilocus-delimitation-with-tr2-guide-tree-approach/>) and also provides a tutorial with example files on Github (https://github.com/tfujisawa/tr2_tutorial)

For iTaxoTools, we added to tr2 a GUI and SPART output; however, the new version of tr2, updated by T. Fujisawa in 2021, also produces SPART output natively.



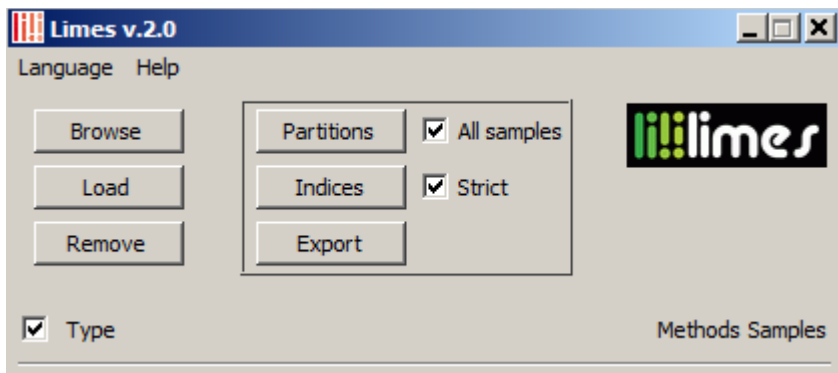
6.7. LIMES v2.0

Ducasse et al. (2020) introduced LIMES, a program not to delimit species but to compare species partitions using various statistical indexes. The original distribution of LIMES is explained and introduced on its website, <https://limes.prod.lamp.cnrs.fr/>

The new version of LIMES included in iTaxoTools presents two distinct functionalities:

► **Species partitions comparison:** LIMES makes objective comparisons of species partitions resulting from different species delimitation approaches, regardless of the methods or type of dataset initially used to infer them. It is specifically adapted to compare datasets composed of a significant number of species (typically at a generic or suprageneric level). It calculates four different indexes: *Ctax*, *mCtax*, *Rtax* (Miralles & Vences 2013) and the *Match Ratio* (Ahrens et al. 2016).

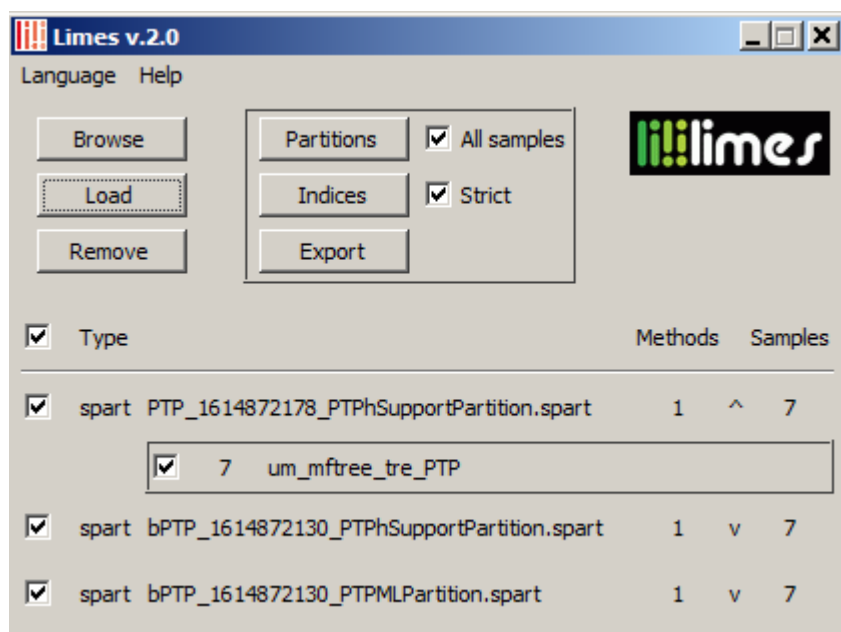
► **Species partitions handling:** In the context of the development of the species partition format SPART (Miralles et al. 2021), this new version of LIMES allows users to easily handle species partitions, i.e. to merge or to extract a selected sets of species partitions.



Starting: The program first requires the user to browse ("Browse") for one or several species partition (SPART) file and to select it (them). Next, the file(s) needs to be loaded ("Load"). During the loading process, the program automatically checks the SPART file(s) for integrity, i.e., for a correct syntax as proposed by Miralles et al. (2021). In the current version, LIMES only accepts the matricial SPART format (not yet the XML-SPART format).

If correctly loaded, the program lists the SPART files with the number of methods included, and the number of samples (individuals) present in each partitions. The different partitions present in a given file can also be shown or masked (by default) by clicking on the arrows ("^", "v"). The following samples has three SPART files loaded, produced with different variants of PTP program implemented in iTaxoTools.

Partitions handling: From this list of files (and partitions they contains), users can recompose a new tailored partition file (i.e. merging or excluding a selection of partitions) by clicking on those of interest, and then export the corresponding spart file ("Export") for other applications.



Partitions comparison : LIMES then compares the loaded partitions, providing a comparison of assignment for each individual ("Partitions") and can calculates various taxonomic indices ("Indices") among the loaded partitions (note that partitions can be selected or deselected using the respective checkboxes, in order to include only a subset of partitions in these comparisons).

The screenshot shows the Limes v.2.0 application window displaying a comparison table. The table has five columns: 'select ->', 'um_mftree_tre_PTP', 'ptp_example_tre_PTP_1', 'ptp_example_tre_PTP_2', and 'Species'. The table contains 15 rows of data, each starting with a letter or code in the 'select ->' column.

select ->	um_mftree_tre_PTP	ptp_example_tre_PTP_1	ptp_example_tre_PTP_2	Species
A	1	-	-	
B	1	-	-	
C	1	1	1	
D	1	-	-	
E	1	-	-	
F	1	-	-	
G	1	-	-	
d1	-	2	2	
d2	-	2	2	
e1	-	3	3	
e2	-	3	3	
f1	-	4	4	
f2	-	4	4	

Limes v.2.0

☒ Ctax
 ☐ Match ratio
 ☐ collimator
 Save

Source(s) :

```

PTP_1614872178_PTPSupportPartition.spart - 10/03/2021 13:56:52
bPTP_1614872130_PTPMLPartition.spart - 10/03/2021 13:56:52
bPTP_1614872130_PTPSupportPartition.spart - 10/03/2021 13:56:52
  
```

Discarded samples:

```

F
D
G
f2
A
f1
B
d2
e1
e2
E
d1
  
```

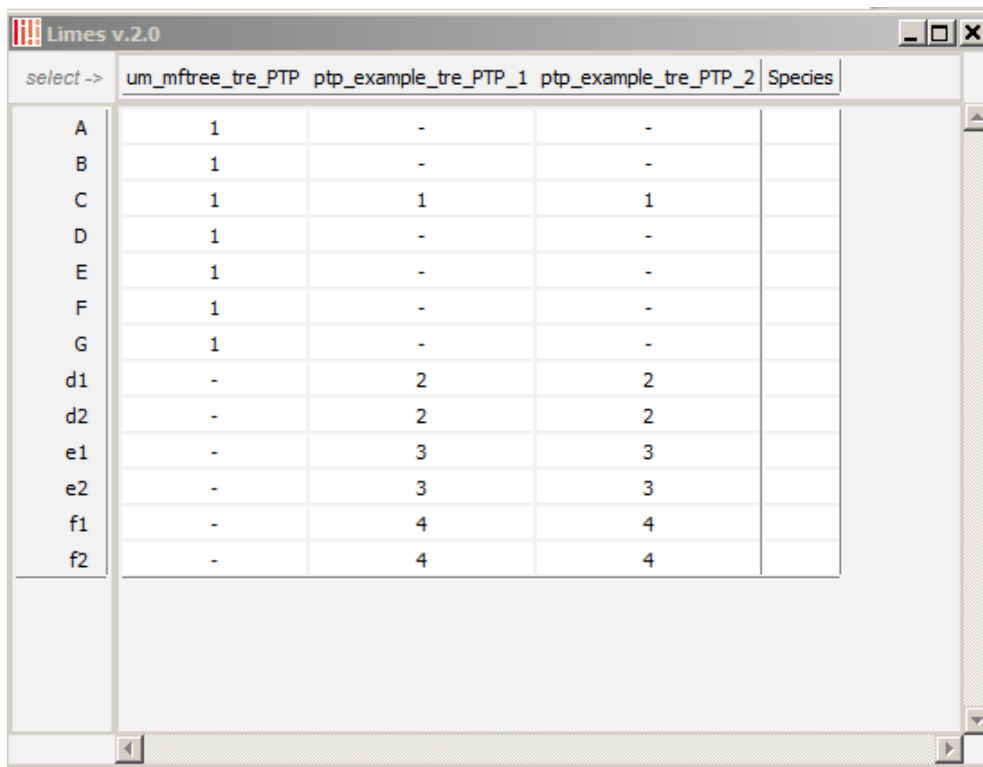
```

M1 = um_mftree_tre_PTP
M2 = ptp_example_tre_PTP_1
M3 = ptp_example_tre_PTP_2
  
```

	Esp	Rtax	Ctax -->		
			M1	M2	M3
M1	1	N/A		N/A	N/A
M2	1	N/A			N/A
M3	1	N/A			

	mCtax	Rtax	Ctax -->		
			M1	M2	M3
M1	N/A	N/A		N/A	N/A
M2	N/A	N/A			N/A
M3	N/A	N/A			

3 methods, 1 samples. Warning: 12 samples had to be discarded



The screenshot shows the Limes v.2.0 application window. At the top, there is a title bar with the text "Limes v.2.0" and standard window controls. Below the title bar is a menu bar with the text "select ->" followed by a list of options: "um_mftree_tre_PTP", "ptp_example_tre_PTP_1", "ptp_example_tre_PTP_2", and "Species". The main area of the window contains a table with 5 columns and 15 rows. The first column lists labels (A, B, C, D, E, F, G, d1, d2, e1, e2, f1, f2). The next three columns contain numerical values (1, -1, 1, -1, -1, -1, -1, 2, 2, 3, 3, 4, 4). The fifth column is empty. The table is surrounded by a light gray border, and there are scroll bars on the right and bottom.

	um_mftree_tre_PTP	ptp_example_tre_PTP_1	ptp_example_tre_PTP_2	Species
A	1	-	-	
B	1	-	-	
C	1	1	1	
D	1	-	-	
E	1	-	-	
F	1	-	-	
G	1	-	-	
d1	-	2	2	
d2	-	2	2	
e1	-	3	3	
e2	-	3	3	
f1	-	4	4	
f2	-	4	4	

Limes v.2.0

☒ Ctax
 ☐ Match ratio
 ☐ collimator
 Save

Source(s) :

```

PTP_1614872178_PTPSupportPartition.spart - 10/03/2021 13:56:52
bPTP_1614872130_PTPMLPartition.spart - 10/03/2021 13:56:52
bPTP_1614872130_PTPSupportPartition.spart - 10/03/2021 13:56:52
  
```

Discarded samples:

```

F
D
G
f2
A
f1
B
d2
e1
e2
E
d1
  
```

```

M1 = um_mftree_tre_PTP
M2 = ptp_example_tre_PTP_1
M3 = ptp_example_tre_PTP_2
  
```

	Esp	Rtax	Ctax -->		
			M1	M2	M3
M1	1	N/A		N/A	N/A
M2	1	N/A			N/A
M3	1	N/A			

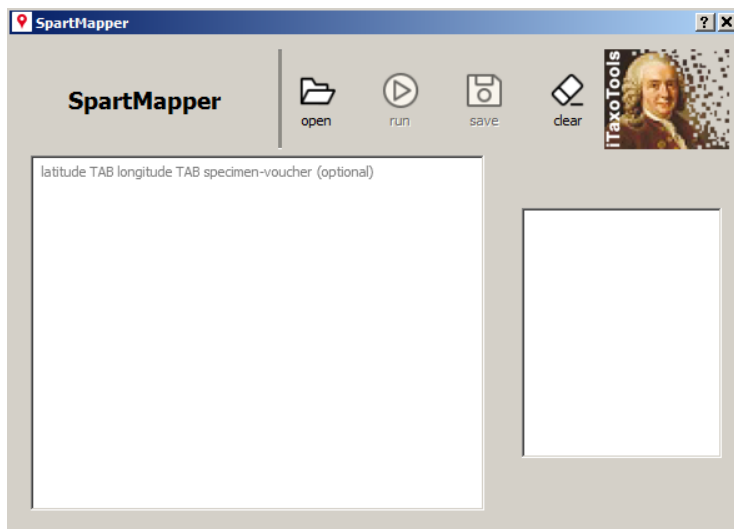
	mCtax	Rtax	Ctax -->		
			M1	M2	M3
M1	N/A	N/A		N/A	N/A
M2	N/A	N/A			N/A
M3	N/A	N/A			

3 methods, 1 samples. Warning: 12 samples had to be discarded

6.8. spartmapper

As a second tool besides LIMES making use of SPART files, iTaxoTools includes spartmapper. The purpose of this program is to visualize the geographic placement of a set of latitude/longitude coordinates and, if combined with a SPART file, place these locations in different color depending on the assignment of samples in a species partition. It includes a live viewer (only with internet connection) and an export of the data as HTML and KML (for visualization in Google Earth and Google Maps).

The program features a single "Open" button. When pressed, the user is asked sequentially to specify a file with geographical coordinates, and a SPART file (the latter being non-compulsory).



Geographical coordinates must be provided in a tab-delimited format, along with specimen identifier information, in the form of a field "specimen-voucher" (case insensitive; also accepted as "specimen_voucher"), as in the following example file.

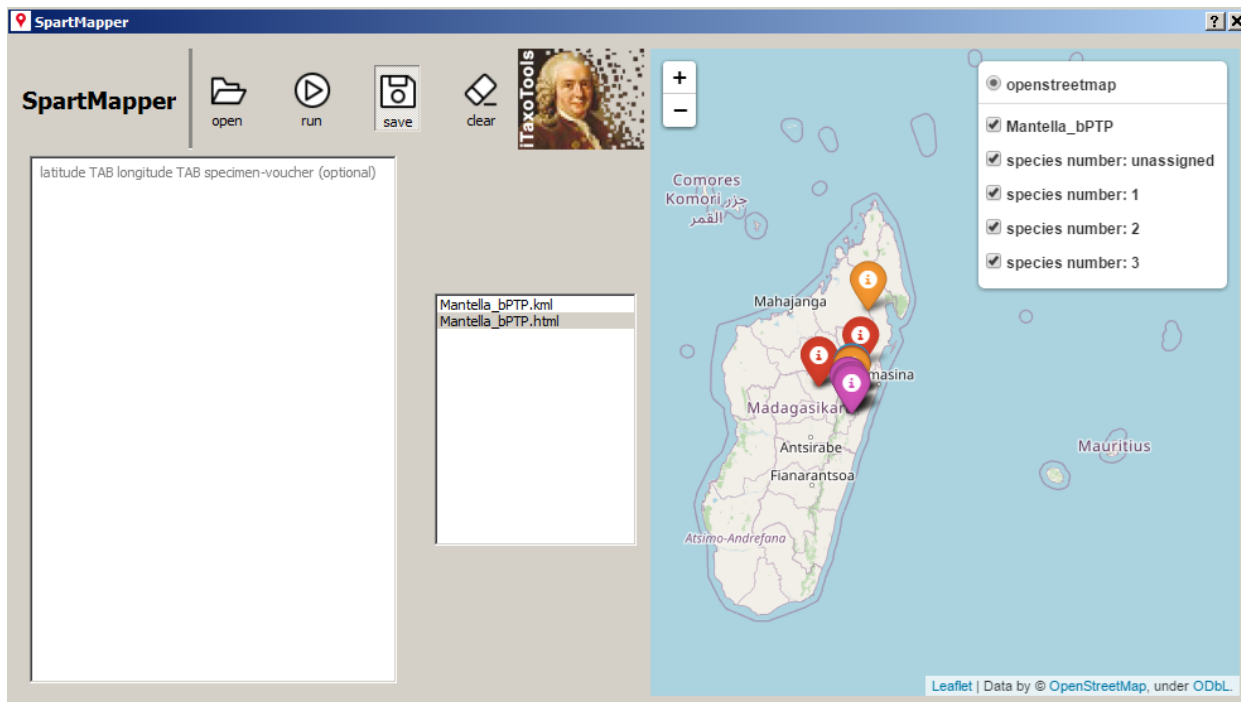
specimen_voucher	latitude	longitude
aura_ZCMV1234	-18.840597	48.29446319
aura_ZCMV1235	-18.840597	48.29446319
aura_ZCMV1236	-18.840597	48.29446319
aura_ZCMV1238	-18.86239067	48.36801703
aura_ZCMV1239	-19.03095015	48.37516196
aura_FGZC987	-19.0903783	48.43354627
aura_FGZC986	-19.00315826	48.43232143
crocea_ZCMV234	-18.20674021	47.28694011
crocea_ZCMV235	-18.2067	47.28694
miloty_ACZC324	-18.48933842	48.41599015
miloty_ACZC329	-18.49979271	48.41435702
crocea_ZCMV236	-17.50672593	48.73102035
crocea_ZCMV237	-15.6480221	48.98115884
miloty_ACZV679	-18.45332435	48.41599015
miloty_ZCMV479	-18.37546169	48.44293675
miloty_ZCMV480	-18.52611918	48.42048125
miloty_ZCMV481	-18.52611918	48.42048125

The values in the "specimen-voucher" field must match the designation of samples (individuals) in the SPART file.

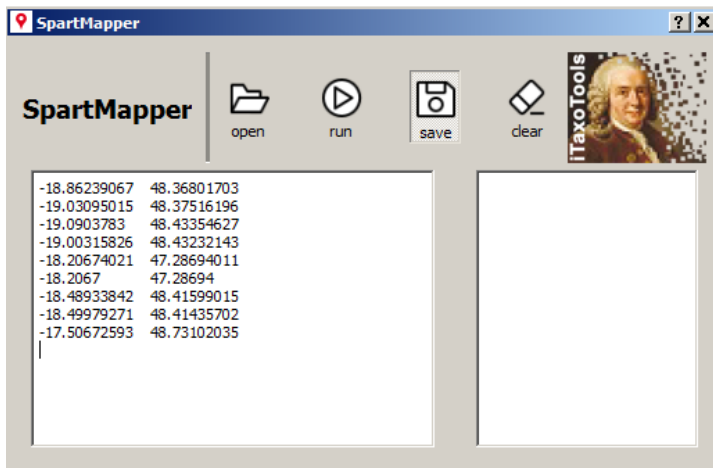
After specifying the input file(s), the "Run" button executes the program.

A double click on the produced KML file will then visualize the file content in the preview box, whereas a double click on the HTML file will open the live viewer using the OpenStreetMap visualization. The button "Save" serves to save both files to a specified folder.

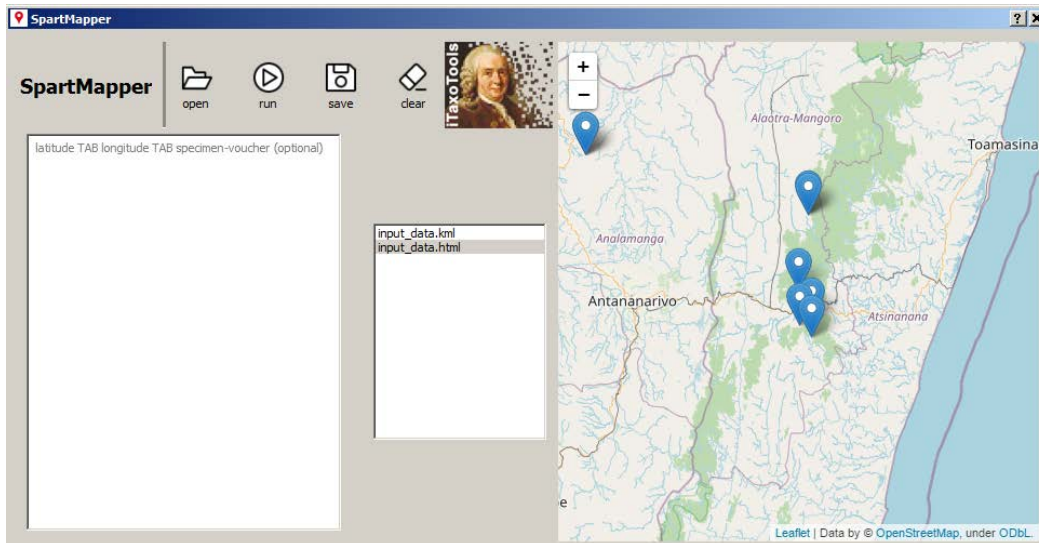
If no SPART file is specified, all records will show in the same color under "unassigned".



For quick visualization of a set of coordinates without specimen numbers, it is also possible to paste these (latitude, longitude: tab delimited) into the box on the left.



Also in this case, maps will be produced with all records under "unassigned".



7. Tools for Species Diagnosis

The diagnosis of new species – rather than its lengthy description – represents the most important part of the alpha-taxonomic process. Several software tools have been proposed to extract diagnostic nucleotide positions of clades and species, of which iTaxoTools currently implements two.

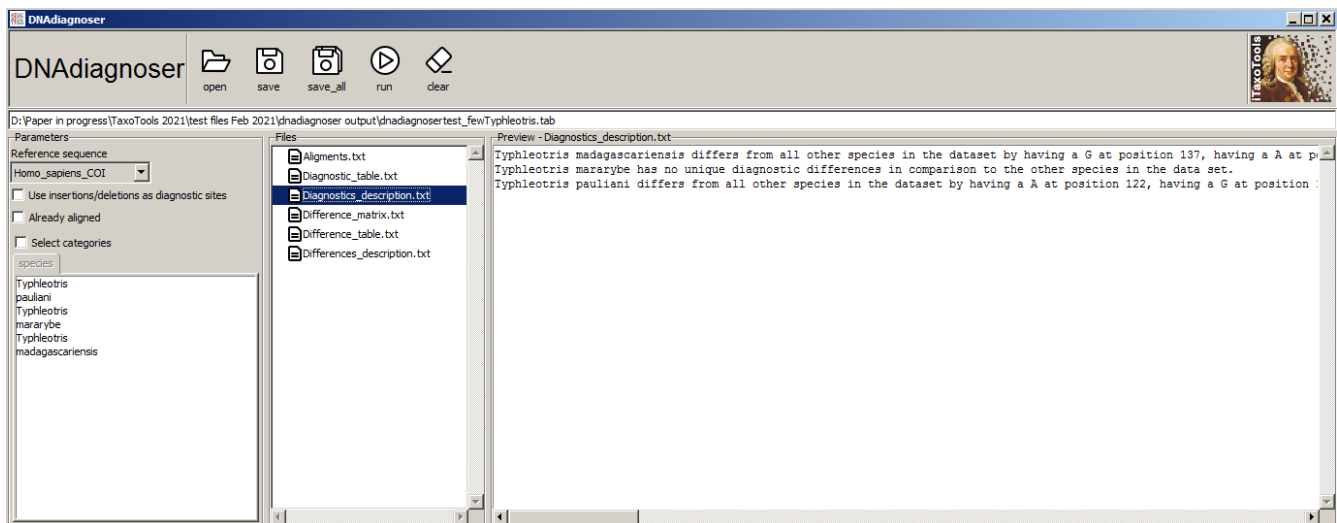
7.1. dnadiagnoser

In order to facilitate the use of such DNA characters in differential diagnoses of new species, we implemented a crucial new tool for DNA taxonomy named dnadiagnoser. This tool takes as input tab-delimited text files in which one column specifies the unit for analysis (typically the species). The files should specify a specimen identifier (specimen-voucher), the species, and the sequence, using these (case-insensitive) column headers, as in the following example.

specimen_voucher	species	sequence
VIN43	Typhleotris pauliani	ccagcccggcgccactattggggagacg
VIN20	Typhleotris pauliani	ccagcccggcgccactattgggggacg
SAF6	Typhleotris pauliani	gccagcccggcgccactattgggggac
SAF5	Typhleotris pauliani	ccagcccggcgccactattgggggacg
SAF4	Typhleotris pauliani	gccagcccggcgccactattgggggac
SAF2	Typhleotris pauliani	gccagcccggcgccactattgggggac
SAF19	Typhleotris pauliani	accaaactacaatgtcgtcgtcacag
AND205	Typhleotris pauliani	ccctgagccttctattcgcgcggagc
TE94	Typhleotris mararybe	cgccggagctgagccaacccggcgca
TE92	Typhleotris mararybe	tgagccaacccggcgccactactgggg
TE83	Typhleotris mararybe	cgccggagctgagccaacccggcgca
TE62	Typhleotris mararybe	cgccggagctgagccaacccggcgca
TE61	Typhleotris mararybe	tgagccaacccggcgccactactgggg
LA148	Typhleotris mararybe	tgagccaacccggcgccactactgggg
LA147	Typhleotris mararybe	tgagccaacccggcgccactactgggg
VIN9	Typhleotris madagascariensis	ccaacccggcgccgctactgggggatg
VIN65	Typhleotris madagascariensis	ccaacccggcgccgctactgggggatg
VIN5	Typhleotris madagascariensis	ccaacccggcgccgctactgggggatg
VIN4	Typhleotris madagascariensis	ccaacccggcgccgctactgggggatg
VIN3	Typhleotris madagascariensis	ccaacccggcgccgctactgggggatg
VIN2	Typhleotris madagascariensis	ccaacccggcgccgctactgggggatg
VIN188	Typhleotris madagascariensis	ccctgagccttctattcgcgcggagc
VIN179	Typhleotris madagascariensis	cgccggagctgagccaacccggcgca
VIN178	Typhleotris madagascariensis	cgccggagctgagccaacccggcgca

The program then provides as output pre-formulated text sentences which specify (i) in a pairwise fashion, all the diagnostic sites of one species against all other species, and (ii) the unique diagnostic sites (if any) that differentiate a species against all other species. These text sentences can then directly be used in species diagnoses.

As a further innovation dnadiagnoser interprets one of the sequences in the input alignment as reference sequence and outputs the diagnostic sites relative to this sequence. Unaligned sequences are then pairwise aligned against the reference sequence to identify diagnostic positions, and labelled according to their position in the reference sequence, a procedure that works reliably in sets of sequences with no or only few insertions or deletions such as COI.



Alternatively, dnadiagnoser can also run a file of pre-aligned sequences, in which case positions are calculated relative to the alignment.

Several additional features in dnadiagnoser are:

- ▶ The possibility to use or not gaps (insertions/deletions) as diagnostic sites
- ▶ Selecting particular species and restricting the comparisons to those

7.2. MOLD

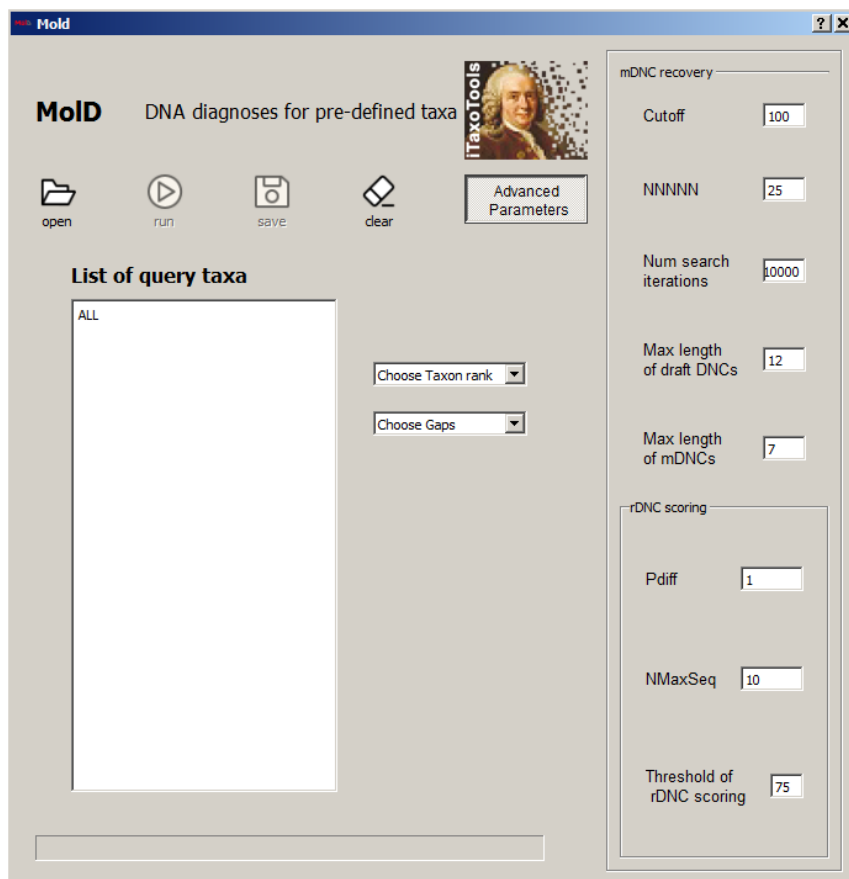
As a second program for species diagnosis, iTaxoTools implements a GUI version of MoID (Fedosov et al. 2020). This program is tailored for recovering DNA-based diagnoses in large DNA dataset, and is capable of identifying diagnostic combinations of nucleotides (DNCs) in addition to single (pure) diagnostic sites. The crucial and unique functionality of MoID allows assembling DNA diagnoses that fulfil pre-defined criteria of reliability, which is achieved by repeatedly scoring diagnostic nucleotide combinations against datasets of in-silico mutated sequences.

A webserver for MoID has been implemented by the developers of the program:

<https://mold.testapi.me/>

At this same website, a dedicated manual can be downloaded which is reproduced on the following pages.

The GUI version of MoID implements the same options as the web version. Different from other GUIs in iTaxoTools, information on the particular option in the menu is provided when cursor-hovering over the field in the GUI.



The following instructions are verbatim copied from the original MolD manual v. 1.3 (by A. Fedosov, 4.12.2020)

Input: data file

The input file is in fasta format: each entry starts with the identifier line, and one or more lines of nucleotide sequence. Identifier line starts with '>' and must contain two parts, separated by a pipe ('|') symbol. The first part is a free-style **sequence identifier**, the second is the **taxon identifier** of the query level. The names of the taxa to be diagnosed correspond to the **second** element.

1. query ID example:

>GBXXXXXXX | query

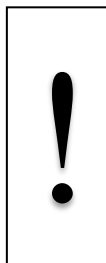
2. reference ID example:

>GBXXXXXXX | ref1

EXAMPLE

[Species of the cone-snail genus *Conasprella* (Gastropoda) – Puillandre et al. 2014]

```
#####
>Conasprella_alisi1|alisi
TATAAGATTTTGGCTTTTACCTCCTGCCCTTCTTTACTCCTTTCTTCAGCT
>Conasprella_alisi2|alisi
TATAAGATTTTGGCTTTTACCTCCTGCCCTTCTTTACTCCTTTCTTCAGCT
>Conasprella_alisi3|alisi
TATAAGATTTTGGCTTTTACCTCCTGCTCTTCTTTACTCCTTTCTTCAGCT
>Conasprella_baileyi1|baileyi
TATAAGATTTTGACTTTTGCCCTCCGCCCTTCTTTACTTCTTTCTTCAGCC
>Conasprella_baileyi2|baileyi
TATAAGATTTTGACTTTTGCCCCGCCCTTCTTTACTTCTTTCTTCAGCC
>Conasprella_boholensis|boholensis
TATAAGATTTTGACTTTTACCTCCTGCGCTTCTTTACTTCTTTCTTCAGCT
>Conasprella_boucheti|boucheti
TATAAGATTTTGACTTTTACCTCCCGCACTTCTTTACTTCTTTCTTCAGCT
>Conasprella_comatosa|comatosa
TATAAGATTTTGACTTTTACCTCCTGCGTTGCTTCTACTCTTATCTTCAGCT
>Conasprella_coriolisi|coriolisi
TATAAGATTTTGACTTTTACCCCTGCGTTGCTTCTACTCCTATCTTCAGCT
#####
```



Please, check that:

1. Each identifier line has only one pipe symbol.
2. No spaces are present in the sequence and taxon identifiers.
3. All taxa identifiers are provided and correct.
4. Sequence lines only contain valid nucleotides ('A', 'C', 'G', 'T'), gaps ('-'), and ambiguous nucleotides ('N', 'K', 'M', 'R', 'S', 'W', 'Y')

Please, note that:

1. Any data file extension (.fas / .fa / .fasta / .txt etc.) will do the job.

Input: Parameters

Either entered in the interface, or (if a command-line implementation is used) provided in the parameter file after '='.

1. INPUT / OUTPUT FILES

- INPUT_FILE – input alignment file with complete path.
- OUTPUT_FILE – output file with complete path (only command-line version)

2. INPUT PARAMETERS (NO DEFAULTS - no parameters entered will lead to an error).

Q TAXA (Query taxa)

ALL	if all taxa in the dataset are to be diagnosed.
>N	if all taxa with more than N sequences available (where N is a natural number) to be diagnosed.
Taxon1, Taxon2 . . .	a comma separated list of taxa to be diagnosed. Please check that all taxa identifiers are provided as in the input alignment.

Taxon_rank

1	for species
2	for supraspecific taxa

Code gaps as characters (whether alignment gaps are used as a character or not)

Yes	dashes ('-') in the alignment are transformed into 'D', which is treated as an independent characters
No	dashes are treated as missing data ('N')

3. ADVANCED PARAMETERS FOR mDNC RECOVERY

[For explanation see 'Review of MolD' below or Fedosov *et al.* 2019. If you do not want to set them leave respective fields blank, and the defaults will be used.]

Cutoff

-integer (default 100)	denotes the number of informative sites to be considered for inclusion into a mDNC;
-integer prepended by ('>')	Informative sites are ranked based on how many sequences of reference taxa differ from the query in the nucleotide at each site (The cut-off value). If this option selected, all informative sites with cut-off value above specified after ('>') will be considered. If '>0' is used all informative sites will be considered for inclusion into mDNC.

NumberN

Number of ambiguously called nucleotides allowed, integer (default **5**).

Number_of_iterations

Number of iterations of MolD, integer (default **10000**).

MaxLen1

Maximum length of draft DNCs, integer (default **12**).

MaxLen2

Maximum length of refined mDNCs, integer (default **7**).

4. PARAMETERS OF ARTIFICIAL DATASETS (only rDNSs).

Pdiff	Percent difference between original and modified sequences, integer (default 1 for species-level taxa, 3 for for supraspecific taxa).								
NmaxSeq	Max number of sequences per taxon to modify, integer (default 10).								
Scoring	<p>To score each candidate rDNC, 100 simulated test datasets are created. If rDNC remains valid in a test dataset, it adds 1 to the score, so lowest possible score is 0 and highest is 100. If two consecutive scores are above the threshold value defined by a keyword argument here (default is moderate) the rDNC is output. Arguments:</p> <table><tr><td>lousy</td><td>66</td></tr><tr><td>moderate</td><td>75</td></tr><tr><td>stringent</td><td>90</td></tr><tr><td>very_stringent</td><td>95</td></tr></table>	lousy	66	moderate	75	stringent	90	very_stringent	95
lousy	66								
moderate	75								
stringent	90								
very_stringent	95								

Review of the MolD algorithm

[For term definition and theoretical background see: Fedosov A.E., Achaz G., Puillandre N. 2019. Revisiting use of DNA characters in taxonomy with MolD - a tree independent algorithm to retrieve diagnostic nucleotide characters from monolocus datasets. *BioRxiv*. DOI: 10.1101/838151]

The MolD algorithm is divided into five consecutive steps. At **first** step sequences are sorted by taxon (as defined by the taxon identifier of the input) and the sites conserved within each taxon are identified.

At the **second** step, each of the sites shared by all query taxon sequences is assigned a **cut-off** value, which corresponds to the number of reference taxa sequences in the alignment with different nucleotide at this site. The sites that are conserved across the entire data set have a minimum cut-off value of 0 (i.e. non-informative). The sites that correspond to Type 1 characters (see Fedosov et al. 2019, Fig. 1) immediately differentiate the query, and have a maximum cut-off value. In this case the cut-off value equals to the total number of reference taxa sequences in the data set. Either the desired size of this subset (parameter **cutoff**, by default set to 100), or the threshold cut-off value ($>N$) can be set by user.

The **third** step contains main functionality of the MolD algorithm implemented in two piped core functions. The `step_reduction_complist` function initiates a draft DNC, and extends it by picking up informative sites one-by-one in random order and appending to the draft DNC. For each picked site the reference taxa sequences that differ at this site from the focus taxon sequences are identified and excluded from further comparisons. The list of reference taxa sequences that share a draft DNC with the query is thus reduced step-by-step until its length equals zero. This is a condition at which the function terminates, and the draft DNC is output, if it comprises no more than a predefined number of sites (parameter **Maxlen1**, default 12). The draft DNC allows unambiguous differentiation of the query taxon members in the analyzed data set, but it usually includes more sites than necessary. So, the draft DNC, is refined by the `RemoveRedundantPositions` function. This function removes redundant sites from the draft DNCs by picking and discarding sites successively one-by-one, and each time checking whether the thus shortened combination remains diagnostic for the query or not. Once the draft DNC cannot be further refined, it constitutes an mDNC, and is sent to output, if its length is equal to or less than a pre-defined (parameter **Maxlen2**, default 7). Each of the mDNCs defines a minimal and sufficient condition for a nucleotide sequence (and a corresponding specimen) to belong to the query taxon. Single execution of the two core functions is termed a **search iteration**; each search iteration supplies one mDNC in the case that length criteria are met. By default, MolD run runs 10,000 search iterations, but their number can be set by a user (parameter **Number_of_iterations**) to generate a pool of mDNCs. The list of non-identical mDNCs sorted by length constitutes the output of the third step.

Two mDNCs may overlap by one or several sites, or share no sites; in the latter case the two mDNCs are termed independent mDNCs (see Fedosov et al. 2019). In the case that all identified mDNCs share one or more sites (i.e. no independent combinations are identified), such site(s) present in all mDNCs are termed **key positions**. The key position(s) are crucial for diagnosing a taxon, because a substitution at this site even in one sequence attributed to a query would immediately make the query-taxon impossible to diagnose with the selected genetic marker. On the contrary, when n independent mDNCs are recovered, n substitutions would be needed to make the query taxon undiagnosable; the

likelihood of the latter scenario is obviously much lower. At the **fourth** step the set of mDNCs is analyzed to identify independent mDNCs, or (if present), key position(s). In the case that no mDNCs were recovered for a pre-defined set of DNA sequences, an exception is raised.

At the **fifth** step the set of mDNCs is converted into rDNC that fulfills pre-defined requirements of robustness. An rDNC is constructed from the list of mDNCs produced by MOLD in the step 3. First, mDNCs are sorted by increasing lengths, and mDNCs of the same length are 'binned'. In each bin, a given site can be shared by several mDNCs. MOLD computes for each site in each bin, its frequency of occurrence. Sites with frequency 1 are present in all mDNCs of the bin. Sites are thus double sorted, first by the mDNC length and then by frequencies. The top sites of this ranking have the highest frequency among the shortest mDNCs. If Type 1 characters exist for a query, they make the top of ranking, as they are considered as the DNCs of the length 1.

A new rDNC is seeded using one random mDNC among the shortest ones (i.e. Type 1 characters when they exist in the list). Then, extra sites are picked from the top of the double-sorted list of sites and are added to the rDNC one-by-one. After each addition of a site, the rDNC is scored for reliability (see below), and the score is recorded. The rDNC extension process stops, either when two successive scores exceed the user-defined reliability threshold (parameter **Scoring**) - then the best-scoring rDNC is sent to output, or when the rDNC reaches 10 nucleotide sites. In the latter case, if at any step an rDNC has scored above the reliability threshold, it is output with a warning. If the scores remain consistently below the reliability threshold, a message is output to inform the user that no sufficiently reliable rDNC could be constructed.

To evaluate an rDNC after each step of elongation, MOLD uses simulated datasets that are generated by altering the original alignment with artificial mutations. MOLD repeatedly creates **test datasets** with artificial sequences derived from the real ones. Each artificial sequence is generated by introducing p nucleotide substitutions into an existing sequence, where p is a random uniform integer in $[1, xL/100]$, where x is the parameter **Pdiff** (default **1**), and L is the alignment length. Mutations are introduced only at polymorphic sites by substituting the original nucleotide into one of the three others, selected randomly with respect to their observed frequencies at this site in the original alignment. For each species of the original alignment, k randomly sampled sequences (parameter **NmaxSeq**, default **10**) are artificially created by mutation. For species with more than 10 sequences in the original alignment, randomly sampled sequences with no mutation are added to the test dataset to match the original number of sequences for this species. Therefore, a test datasets includes at least 10 sequences per species.

For each rDNC evaluation step, MOLD generates 100 new test datasets. For each of them, the rDNC under evaluation scores 1 if it unambiguously delimits the query taxon (unique combination defining the query taxon) or 0 otherwise. So, DNC score ranges from 0 (if rDNC failed in all 100 test datasets) and 100. Importantly, MOLD tolerates one discordant site when evaluating whether the query taxon is correctly diagnosed in a test dataset - if all but one sites delineate the query unambiguously, it scores 1.

Thus the rDNC is output as a final DNA diagnosis if:

- rDNC scores **66+** in two consecutive runs, and the Scoring is set as **lousy**, or
- rDNC scores **75+** in two consecutive runs, and the Scoring is set as **moderate**, or
- rDNC scores **90+** in two consecutive runs, and the Scoring is set as **stringent**, or
- rDNC scores **95+** in two consecutive runs, and the Scoring is set as **very stringent**.

Quick how to... [some advices to help setting the MOLD run]

First it makes sense to run MOLD with all default settings and check whether all queries were successfully diagnosed or not. If not, one of the following issues might happen:

Issue	Reason / Troubleshooting
No mDNCs identified for a query or Number of identified mDNC is too small (< 10)	<p>There is a problem with sequences attribution to taxa/ Please, check carefully taxon identifiers in query and reference records. It is strongly recommended to make sure that taxon identifiers correspond to clades in the phylogenetic tree.</p> <p>Lack or paucity of signature DNA characters/ More thorough search for mDNCs may help. Set up 'Cutoff' as '>0' to include all informative sites in the mDNCs. If it doesn't help, try excluding sequences containing 'N's at the alignment polymorphic sites.</p> <p>If it doesn't help, at last resort:</p> <ul style="list-style-type: none"> -If the query is a superspecific taxon, try splitting it into distinctive phylogenetic clusters, and providing a separate diagnosis to each of them. -If the query is a species, it looks like you may need to look for an alternative locus, or consider deeper genomic sampling.
No sufficiently reliable rDNC could be identified for a query (while multiple mDNC are recovered)	<p>There is a problem with sequences attribution to taxa/ Please, check carefully taxon identifiers in query and reference records. It is strongly recommended to make sure that taxon identifiers correspond to clades in the phylogenetic tree.</p> <p>Too strict parameters for scoring rDNC/ Try relaxing (using lower values) each of the following parameters* in the following order :</p> <p>NMaxSeq – setting it at 5 is acceptable, below is not recommen.</p> <p>Pdiff – for species-level it should be either 1 or 2.</p> <p>Scoring – 'lousy' should only be used at last resort</p> <p>*note that by relaxing each or all parameters you compromise reliability of the resulting diagnosis.</p>

8. References

- Ahrens, D., Fujisawa, T., Krammer, H.J., Eberle, J., Fabrizi, S. & Vogler, A.P. (2016) Rarity and incomplete sampling in DNA-based species delimitation. *Systematic Biology*, 65, 478–494.
- Coleman, C.O., Lowry, J.K. & Macfarlane, T. (2010) DELTA for Beginners: An introduction into the taxonomy software package DELTA. *ZooKeys*, 45, 1–75.
- Dayrat, B. (2005) Toward integrative taxonomy. *Biological Journal of the Linnean Society*, 85, 407–415.
- De Queiroz, K. (2007) Species concepts and species delimitation. *Systematic Biology*, 56, 879–886.
- Ducasse, J., Ung, V., Lecointre, G. & Miralles, A. (2020). LIMES: a tool for comparing species partition. *Bioinformatics*, 36, 2282–2283.
- Edler, D., Klein, J., Antonelli, A., Silvestro, D. (2020) raxmlGUI 2.0: A graphical interface and toolkit for phylogenetic analyses using RAxML. *Methods in Ecology and Evolution*, doi: <http://dx.doi.org/10.1111/2041-210X.13512>
- Fedosov, A., Achaz, G. & Puillandre, N. (2019) Revisiting use of DNA characters in taxonomy with MolD - a tree independent algorithm to retrieve diagnostic nucleotide characters from monolocus datasets. *bioRxiv*, 838151; doi: <https://doi.org/10.1101/838151>
- Fujisawa, T., Aswad, A. & Barraclough, T.G. (2016) A rapid and scalable method for multilocus species delimitation using Bayesian model comparison and rooted triplets. *Systematic Biology*, 65, 759–771
- Fujisawa, T. & Barraclough, T.G. (2013) Delimiting species using single-locus data and the Generalized Mixed Yule Coalescent approach: a revised method and evaluation on simulated data sets. *Systematic Biology*, 62, 707–724.
- Güntsch, A., Groom, Q., Hyam, R., Chagnoux, S., Röpert, D., Berendsohn, W., Casino, A., Droège, G., Gerritsen, W., Holetschek, J., Marhold, K., Mergen, P., Rainer, H., Smith, V. & Triebel, D. (2018) Standardised globally unique specimen identifiers. *Biodiversity Information Standards*, 2, e26658.
- Hütter, T., Ganser, M.H., Kocher, M., Halkic, M., Agatha, S., Augsten, N. (2020) DeSignate: detecting signature characters in gene sequence alignments for taxon diagnoses. *BMC Bioinformatics*, 21, 151.
- 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.
- Kumar, S., Stecher, G., Li, M., Knyaz, C. & Tamura, K. (2018) MEGA X: Molecular Evolutionary Genetics Analysis across computing platforms. *Molecular Biology and Evolution*, 35, 1547–1549.
- Lanfear, R., Frandsen, P.B., Wright, A.M., Senfeld, T. & Calcott, B. (2016) PartitionFinder 2: new methods for selecting partitioned models of evolution for molecular and morphological phylogenetic analyses. *Molecular Biology and Evolution*, 34, 772–773.
- Lendemer, J., Thiers, B., Monfils, A.K., Zaspel, J., Ellwood, E.R., Bentley, A., LeVan, K., Bates, J., Jennings, D., Contreras, D., Lagomarsino, L., Mabey, P., Ford, L.S., Guralnick, R., Gropp, R.E., Revelez, M., Cobb, N., Selmann, K. & Aime, M.C. (2020) The extended specimen network: a strategy to enhance US biodiversity collections, promote research and education. *BioScience*, 70, 23–30.
- Merckelbach, L.M. & Borges, L.M.S. (2020) Make every species count: fastachar software for rapid determination of molecular diagnostic characters to describe species. *Molecular Ecology Resources*, 20, 1761–1768.
- Meier, R., Kwong, S., Vaidya, G. & Ng, P.K.L. (2006) DNA Barcoding and taxonomy in Diptera: a tale of high intraspecific variability and low identification success. *Systematic Biology*, 55, 715–728.
- Miralles, A., Bruy, T., Wolcott, K., Scherz, M.D., Begerow, D., Beszteri, B., Bonkowski, B., Felden, J., Gemeinholzer, B., Glaw, F., Glöckner, F.O., Hawlitschek, O., Kostadinov, I., Nattkemper, T.W., Printzen, C., Renz, J., Rybalka, N., Stadler, M., Weibulat, T., Wilke, T., Renner, S.S., Vences, M. (2020) Repositories for taxonomic data: Where we are and what is missing. *Systematic Biology*, 69, 1231–1253.
- Padial, J.M., Miralles, A., De la Riva, I. & Vences, M. (2010) The integrative future of taxonomy. *Frontiers in Zoology*, 7, e16.
- Miralles, A. & Vences, M. (2013) New metrics for comparison of taxonomies reveal striking discrepancies among species delimitation methods in *Madascincus* lizards. *PLoS ONE*, 8, e68242.
- Miralles, A., Ducasse, J., Brouillet, S., Flouri, T., Fujisawa, T., Kapli, P., Knowles, L.L., Kumari, S., Stamatakis, A., Sukumaran, J., Lutteropp, S., Vences, M. & Puillandre, N. (2021) SPART, a versatile and standardized data exchange format for species partition information. *BioRxiv*, doi: <https://doi.org/10.1101/2021.03.22.435428>
- Mirarab, S., Reaz, R., Bayzid, Md. S., Zimmermann, T., Swenson, M.S., Warnow, T. (2014) ASTRAL: Genome-scale coalescent-based species tree estimation. *Bioinformatics*, 30, i541–i548.
- Paradis, E., Claude, J. & Strimmer, K. (2004) APE: Analyses of Phylogenetics and Evolution in R language, *Bioinformatics*, 20, 289–290.
- Patterson, D.J., Cooper, J., Kirk, P.M., Pyle, R.L. & Remsen, D.P. (2010) Names are key to the big new biology. *Trends in Ecology and Evolution*, 25, 686–691.
- Pons, J., Barraclough, T.G., Gomez-Zurita, J., Cardoso, A., Duran, D.P., Hazell, S., Kamoun, S., Sumlin, W.D. & Vogler, A.P. (2006) Sequence-based species delimitation for the DNA taxonomy of undescribed insects. *Systematic Biology*, 55, 595–609.

- Powell, M.J.D. (1964) An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7, 155–162.
- Press, W.H., Flannery, B.P., Teukolsky, S.A. & Vetterling, W.T. (1992) *Numerical Recipes in C*. Cambridge University Press, New York. 2nd ed.
- Puillandre, N., Brouillet, S. & Achaz, G. (2020) ASAP: assemble species by automatic partitioning. *Molecular Ecology Resources*, 21: 609–620.
- Puillandre, N., Lambert, A., Brouillet, S. & Achaz, G. (2012) ABGD, Automatic Barcode Gap Discovery for primary species delimitation. *Molecular Ecology*, 21, 1864–1877.
- Rabiee, M. & Mirarab, S. (2021) SODA: Multi-locus species delimitation using quartet frequencies, *Bioinformatics*. btaa1010, <https://doi.org/10.1093/bioinformatics/btaa1010>
- Renner, S.S. (2016) A return to Linnaeus's focus on diagnosis, not description: The use of DNA characters in the formal naming of species. *Systematic Biology*, 65, 1085–1095.
- Riedel, A., Sagata, K., Surbakti, S., Tänzler, R. & Balke, M. (2013) One hundred and one new species of *Trigonopterus* weevils from New Guinea. *Zookeys*, 280, 1–150.
- Ratnasingham, S. & Hebert, P.D. (2013) A DNA-based registry for all animal species: the barcode index number (BIN) system. *PLoS ONE*, 8, e66213.
- Sanderson, M.J. (1997) A non-parametric approach to estimating divergence times in the absence of rate constancy. *Molecular Biology and Evolution*, 14, 1218–1231.
- Sanderson, M.J. (2003) r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics*, 19, 301–302.
- Sarkar, I.N., Planet, P.J., Desalle, R. (2008) caos software for use in character-based DNA barcoding. *Molecular Ecology Resources*, 8, 1256–1259.
- Solís-Lemus, C., Knowles, L.L. & Ané, C. (2015) Bayesian species delimitation combining multiple genes and traits in a unified framework. *Evolution*, 69, 492–507.
- Stamatakis, A. (2014) RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* 30, 1312–1313. <https://doi.org/10.1093/bioinformatics/btu033>.
- Steinke, D., Salzburger, W., Vences, M. & Meyer, A. (2005) TaxI - A software tool for DNA barcoding using distance methods. – *Philosophical Transactions of the Royal Society London, Ser. B*, 360, 1975–1980.
- Sukumaran, J. & Knowles, L.L. (2017) Multispecies coalescent delimits structure, not species. *Proceedings of the National Academy of the U.S.A.*, 114, 1607–1612.
- Sukumaran, J., Holder, T.M. & Knowles, L.L. (2020) Incorporating the speciation process into species delimitation. <https://github.com/jeetsukumaran/delineate>.
- Sukumaran, J. & Holder, M.T. (2010) DendroPy: A Python library for phylogenetic computing. *Bioinformatics*, 26, 1569–1571.
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17, 261–272.
- Wheeler, Q.D., Knapp, S., Stevenson, D.W., Stevenson, J., Blum, S.D., Boom, B.M., Borisy, G.G., Buizer, J.L., De Carvalho, M.R., Cibrián, A., Donoghue, M.J., Doyle, V., Gerson, E.M., Graham, C.H., Graves, P., Graves, S.J., Guralnick, R.P., Hamilton, A.L., Hanken, J., Law, W., Lipscomb, D.L., Lovejoy, T.E., Miller, H., Miller, J.S., Naeem, S., Novacek, M.J., Page, L.M., Platnick, N.I., Porter-Morgan, H., Raven, P.H., Solis, M.A., Valdecasas, A.G., Van Der Leeuw, S., Vasco, A., Vermeulen, N., Vogel, J., Walls, R.L., Wilson, E.O. & Woolley, J.B. (2012) Mapping the biosphere: exploring species to understand the origin, organization and sustainability of biodiversity. *Systematics and Biodiversity*, 10, 1–20.
- Yang, Z. & Rannala, B. (2006) Bayesian estimation of species divergence times under a molecular clock using multiple fossil calibrations with soft bounds. *Molecular Biology and Evolution*, 23, 212–226.
- Zhang J., Kapli P., Pavlidis P. & Stamatakis A. (2013) A general species delimitation method with applications to phylogenetic placements. *Bioinformatics*, 29, 2869–2876.