iTaxoTools - tools for integrative taxonomy

iTaxoTools 0.1 - Manual

Version: June 11, 2021

Excecutables of iTaxotools 0.1 are available at
https://github.com/iTaxoTools/iTaxoTools-Executables/releases
(this link also contains a "launcher" executable which is a single tool containing all of the tools; however, each tool is also available as single standalone executable)

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

How to cite: When using one of the programs included in the iTaxoTools 0.1 release in your study, please cite the main paper as follows, and for several programs cite also the original paper (see the "How to cite" section below).

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, https://doi.org/10.1101/2021.03.26.435825
(in press in Megataxa)

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

Content

0. Release Notes

Manual: This is a first version of the manual (published along with the release of iTaxoTools 0.1 in 2021). The manual will be continuously updated and expanded.

General Notes:

► The current release iTaxoTools 0.1 in several aspects is still preliminary. Improvement of the graphical user interfaces and exhaustive testing of all tools is in continuous 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

► Please use the GitHub platform to submit suggestions, comments, questions and bug reports, by creating a "New Issue" under the respective repository. The team of developers regularly checks all the issues and will deal with them asap (bug reports will be treated as priority).

Specific release notes on the various tools:

All tools	►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.
specimentablepruner	►This program comes with relatively extensive Python libraries and therefore takes a substantial time to load on slow systems. ►This program has not yet been exhaustively tested and may contain bugs. ►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.
specimentablemerger	►This program comes with relatively extensive Python libraries and therefore takes a substantial time to load on slow systems. ►This program has not yet been exhaustively tested and may contain bugs. ►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.
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 matricial SPART files (no XML-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. ►The pairwise alignment algorithm has been set so that it should perform well with standard sequences (e.g., COI). However we recommend to check the alignments, and if alignment probalems are apparent, either use the original Python code where alignment settingsa can be adjusted, or use the program with pre-aligned input files.
PTP	►We are providing both a GUI version that only runs with default settings, and one with a re-designed GUI that also includes the option to set advanced parameters.
GMYC	
tr2	
DELINEATE	

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.

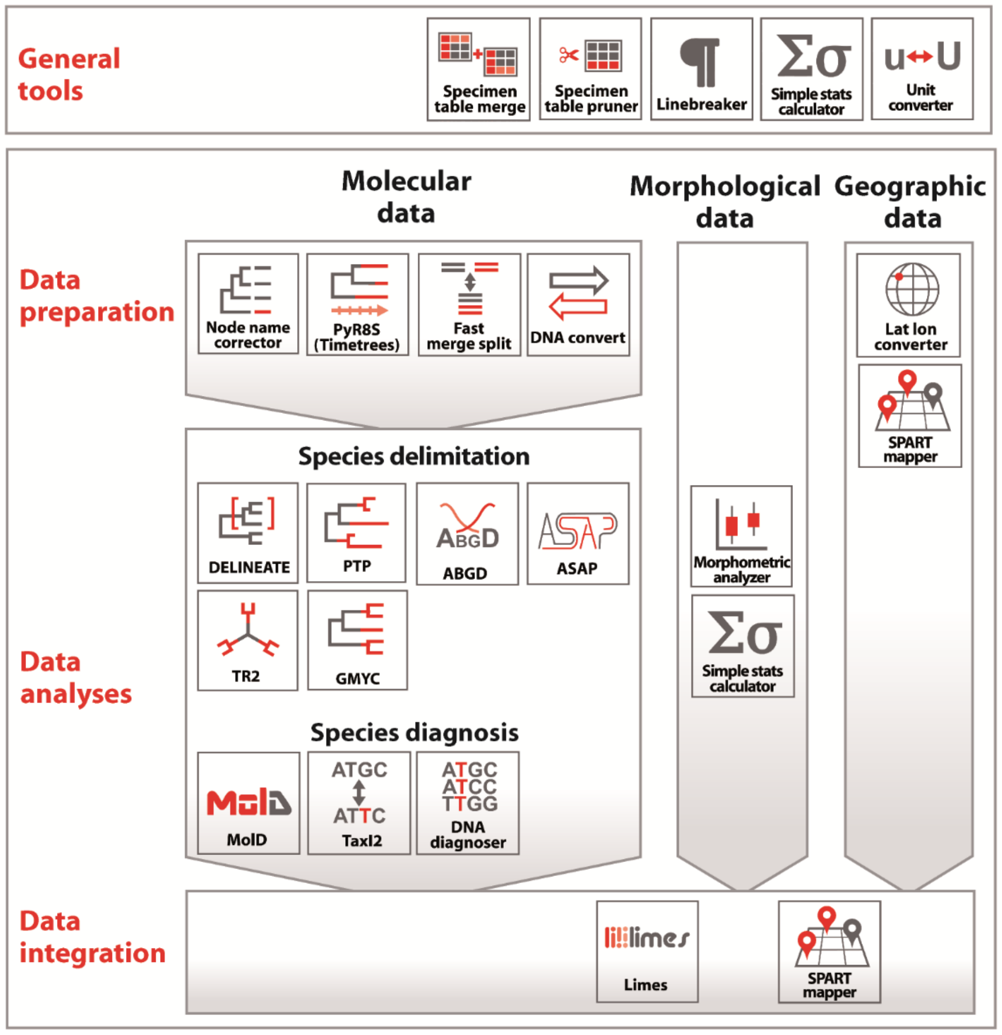
We chose Python as the main programming language for our package, because it is characterized by its good readability and simple-to-learn syntax, and we documented newly written code extensively, to allow its re-use by other programmers. This comes at the cost of speed that would have been achieved by 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. 20202), 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)**
iTaxoTools executables (repository for the binaries in Windows, Linux and Mac format)	***https://github.com/iTaxoTools/iTaxoTools-Executables***	***NA***
dnaconvert	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/Spartmapper	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/jeetsukumaran/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. 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, https://doi.org/10.1101/2021.03.26.435825
(in press in Megataxa)

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

In addition, 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

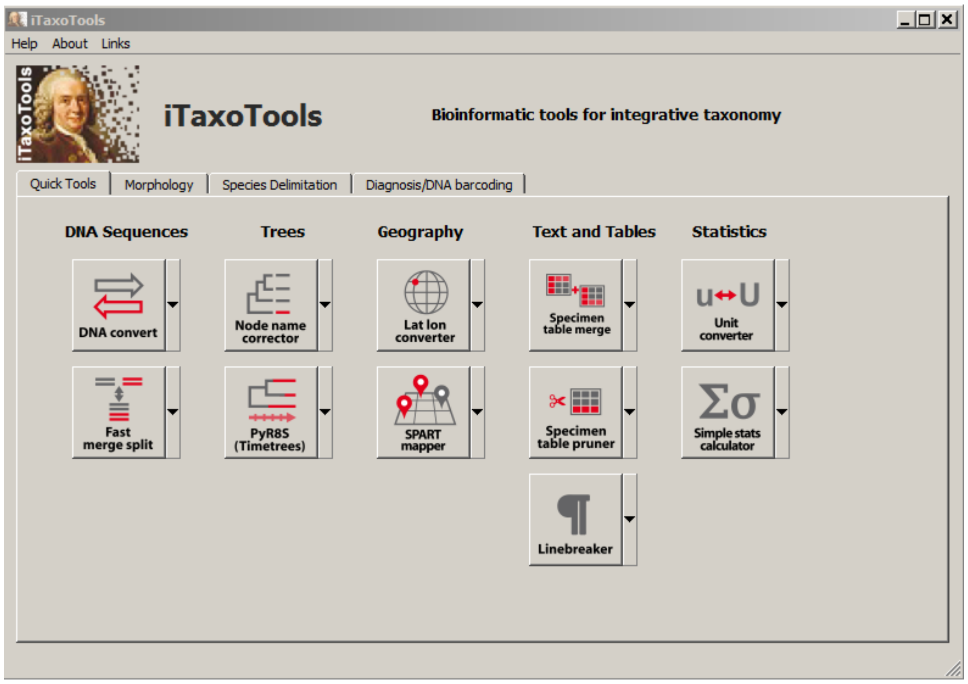
All of the code developed by us is fully open source and available from dedicated GitHub repositories, all under https://github.com/iTaxoTools.

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. Press on the respective button to start the tool (in a new window), or use the drop-down menu at the edge of the button to see a help file /tutorial or to download an example file (a full set of example files is available from http://itaxotools.org). We do not recommend opening and running simultaneously multiple tools using the launcher; if you encounter problems, try closing all tools and opening only one at a time.



Several of the previously existing programs already run from different webservers; we plan to establish a dedicated webserver where all tools can be run online in the near future. See links and updates on http://itaxotools.org

4. Tools for Data Preparation

4.1. DNAconvert

Example files provided:

DNAconvert_examplefile1_iTaxoTools_0_1.tab
DNAconvert_examplefile2_iTaxoTools_0_1.fas
DNAconvert_examplefile3_iTaxoTools_0_1.gb

Three example files are provided, in three of the main formats: tab-delimited text (.tab), fasta (.fas), Genbank flatfile (.gb).

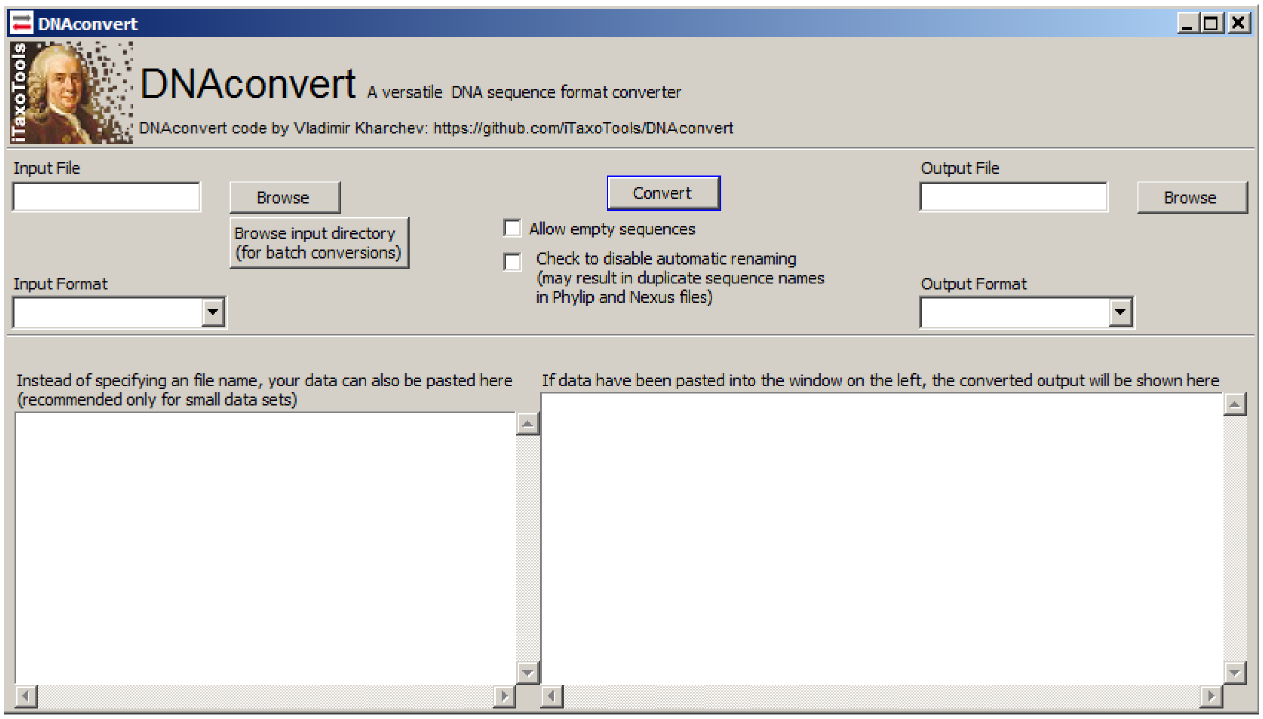
Try the program by using either of these files as input. There is no need to specify the input format if the extension (.tab, .fas, .gb) is correct. Specify the required output format and the output file name, and run the program. [*Note that you can also run the tool by directly pasting input into the box, in which case you must specify the format.*]

To open, inspect and edit the example files, use a simple text editor. The .tab file can also be directly opened in a spreadsheet editor such as Excel

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.


DNAconvert
iTaxoTools
DNAconvert A versatile DNA sequence format converter
DNAconvert code by Vladimir Kharchev: https://github.com/iTaxoTools/DNAconvert
Input File
Browse
Browse input directory (for batch conversions)
Convert
Allow empty sequences
Check to disable automatic renaming (may result in duplicate sequence names in Phylip and Nexus files)
Output File
Browse
Input Format
Output Format
Instead of specifying an file name, your data can also be pasted here (recommended only for small data sets)
If data have been pasted into the window on the left, the converted output will be shown here

The program is executed by the "Convert" button.

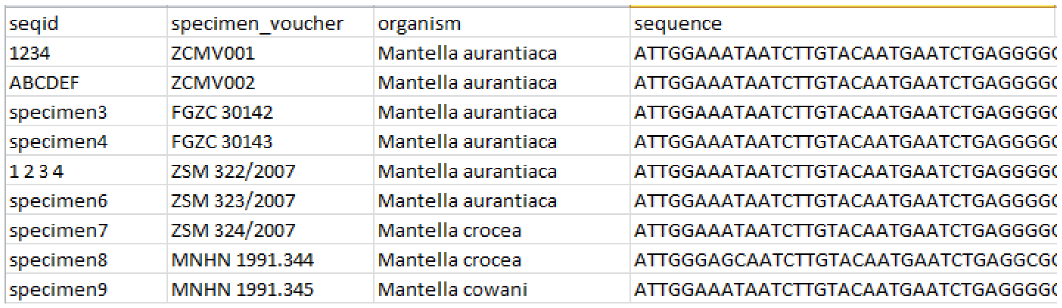
Two options can be selected with checkboxes:

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).

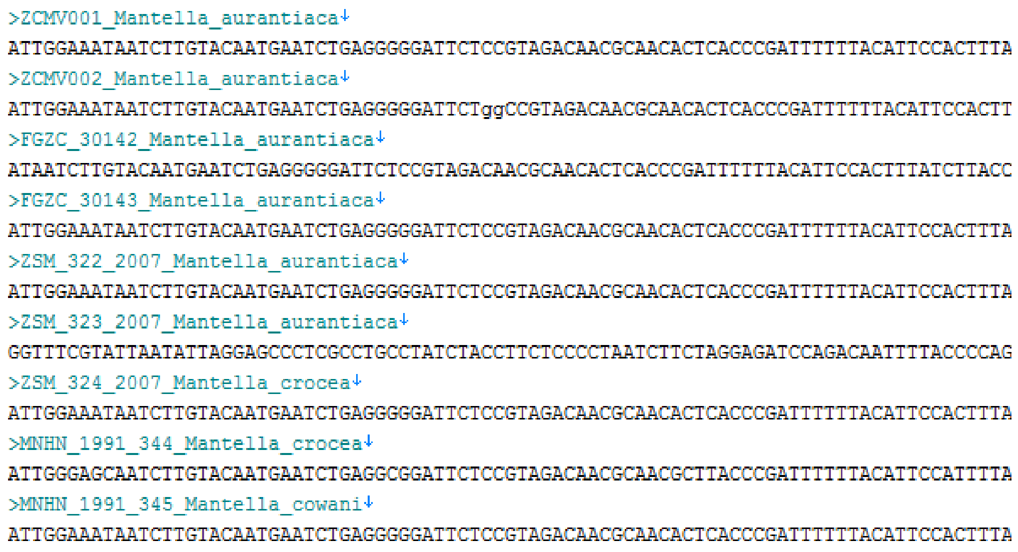


seqid	specimen_voucher	organism	sequence
1234	ZCMV001	Mantella aurantiaca	ATTGGAAATAATCTTGTACAATGAATCTGAGGGG
ABCDEF	ZCMV002	Mantella aurantiaca	ATTGGAAATAATCTTGTACAATGAATCTGAGGGG
specimen3	FGZC 30142	Mantella aurantiaca	ATTGGAAATAATCTTGTACAATGAATCTGAGGGG
specimen4	FGZC 30143	Mantella aurantiaca	ATTGGAAATAATCTTGTACAATGAATCTGAGGGG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

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
ATTGGAAATAATCTTGTACAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACACTCACCCGATTTTTTACATTCCACTTTA
>ZCMV002_Mantella_aurantiaca
ATTGGAAATAATCTTGTACAATGAATCTGAGGGGGATTCTggCCGTAGACAACGCAACACTCACCCGATTTTTTACATTCCACTT
>FGZC_30142_Mantella_aurantiaca
ATAATCTTGTACAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACACTCACCCGATTTTTTACATTCCACTTTATCTTACC
>FGZC_30143_Mantella_aurantiaca
ATTGGAAATAATCTTGTACAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACACTCACCCGATTTTTTACATTCCACTTTA
>ZSM_322_2007_Mantella_aurantiaca
ATTGGAAATAATCTTGTACAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACACTCACCCGATTTTTTACATTCCACTTTA
>ZSM_323_2007_Mantella_aurantiaca
GGTTTCGTATTAATATTAGGAGCCCTCGCCTGCCTATCTACCTTCTCCCCTAATCTTCTAGGAGATCCAGACAATTTTACCCCAG
>ZSM_324_2007_Mantella_crocea
ATTGGAAATAATCTTGTACAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACACTCACCCGATTTTTTACATTCCACTTTA
>MNHN_1991_344_Mantella_crocea
ATTGGGAGCAATCTTGTACAATGAATCTGAGGCGGATTCTCCGTAGACAACGCAACGCTTACCCGATTTTTTACATTCCATTTTA
>MNHN_1991_345_Mantella_cowani
ATTGGAAATAATCTTGTACAATGAATCTGAGGGGGATTCTCCGTAGACAACGCAACACTCACCCGATTTTTTACATTCCACTTTA
```

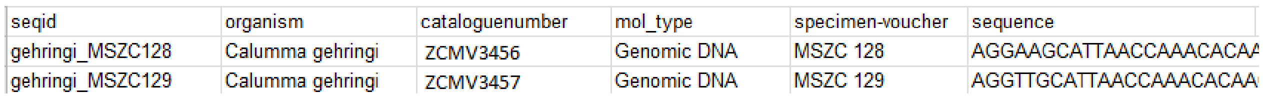
Additional standard sequence formats implemented are: fasta, fastq, phylip, relaxed phylip, genbank, and genbank-export:

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	AGGAAGCATTAACCAAACACA
gehringi_MSZC129	Calumma gehringi	ZCMV3457	Genomic DNA	MSZC 129	AGGTTGCATTAACCAAACACAA

becomes

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

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.

mold-fasta **[erroneously moid-fasta in the GUI - to be fixed in the next release]** is a fasta variant where the sequence names (when transformed from a gb or tab file) has the species name and sequence ID arranged in the way needed as input for the program MolD.

Additional information on the code and command-line version of the tool

You can also run DNAconvert directly in Python, either making use of the GUI or as command line. The following is a summary of the information on the code and the command-line version.

Dependencies

- python-nexus

Installation

Installation is currently not intended. Downloading should be enough

Generating an executable

Using PyInstaller is recommended. After the following instruction a directory dist will be created (among other) and the executable will be inside it.

Linux
Install PyInstaller from PyPI:
pip install pyinstaller
Then run

```
pyinstaller --onefile DNAconvert.py
```

Windows
Install PyInstaller:
Installing on Windows
Then run

```
pyinstaller --onefile --windowed DNAconvert.py
```

Usage

```
usage: DNAconvert.py [-h] [--cmd] [--allow_empty_sequences]
                [--informat INFORMAT] [--outformat OUTFORMAT]
                [infile] [outfile]
    DNAconvert.py

positional arguments:
  infile               the input file
  outfile              the output file

optional arguments:
  -h, --help           show this help message and exit
  --cmd                activates the command-line interface
  --allow_empty_sequences
                       set this to keep the empty sequences in the output
                       file
  --disable_automatic_renaming
                       disables automatic renaming, may result in duplicate
                       sequence names in Phylip and Nexus files
  --informat INFORMAT   format of the input file
  --outformat OUTFORMAT
                       format of the output file
```

Batch processing
If infile is a directory, all files in it will be converted. In this case informat and outformat arguments are required.
Specifying names of the output files:

- outfile contains a '#' character: '#' will be replaced with the base names of input files.
- outfile is a directory: the output files will be written in it, with the same names as input files.

Supported formats

- tab: Internal tab format
- tab_noheaders: Internal tab format without headers
- fasta: FASTA format
- relaxed_phylip: relaxed Phylip format
- fasta_hapview: FASTA format for Haplotype Viewer
- phylip: Phylip format
- fastq: FASTQ format
- fasta_gbexport: FASTA format for export into Genbank repository
- nexus: NEXUS format

- genbank: Genbank flat file format
- moid_fas: FASTA format with sequence name matching requirements for the tool MolD

Recognised extension

If format is not provided, the program can infer it from the file extension
Currently recognised:

- .tab, .txt, .tsv: Internal tab format
- .fas, .fasta, .fna: FASTA format
- .rel.phy: relaxed Phylip format
- .hapv.fas: FASTA format for Haplotype Viewer
- .phy: Phylip format
- .fastq, .fq: FASTQ format
- .fastq.gz, .fq.gz: FASTQ format compressed with Gzip
- .gb.fas: FASTA format for export into Genbank repository
- .nex: NEXUS format
- .gb: Genbank flat file format

Files with extension .gz are uncompressed automatically

Options

DNAconvert uses two parsers for NEXUS format: internal and the one from python-nexus package.
In the file data/cfg.tab the string of form
nexus_parser<Tab>(method)
determines the parser. (method) is either internal or python-nexus.

4.2 latlonconverter

Example file provided:

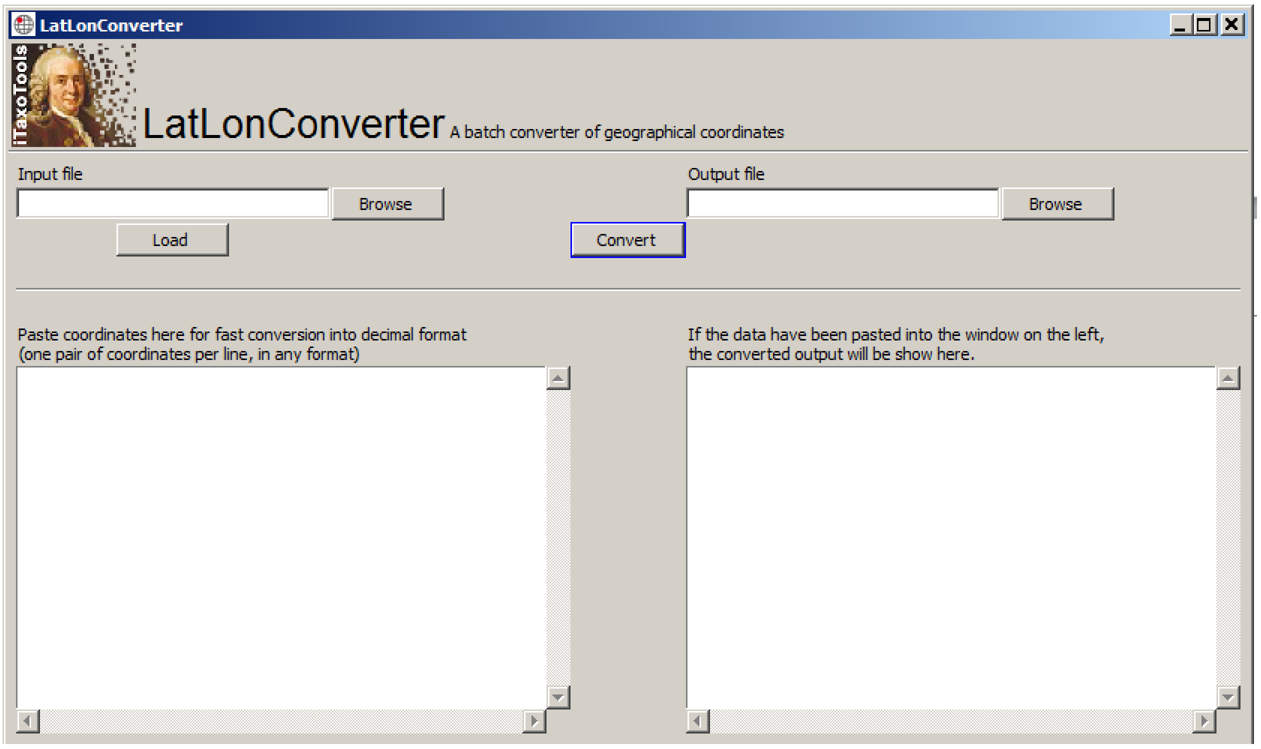
Latlonconverter_examplefile_iTaxoTools_0_1.tab

The example file is a tab-separated tab file with geographical coordinates in different formats.

Try the program by using this file as input. [*Note that you can also run the tool by directly pasting input into the box, in which case the required format is simpler: latitude and longitude in any format and with any separator, and without heading.*]

To open, inspect and edit the example file and the output, you can use use a simple text editor, but for better visualization we recommend to open both files in a spreadsheet editor such as Excel

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.


LatLonConverter
iTaxoTools
LatLonConverter A batch converter of geographical coordinates
Input file
Browse
Load
Convert
Output file
Browse
Paste coordinates here for fast conversion into decimal format
(one pair of coordinates per line, in any format)
If the data have been pasted into the window on the left,
the converted output will be show here.

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.

Additional information on the code and command-line version of the tool

latlon_conv.py --cmd < input_file > output_file latlon_conv.py

The `--cmd` option launches the command-line version, otherwise the GUI is launched.

The input file should contain coordinates in maximum two (tab-separated) columns. First line can be a heading with column headings being one of: 'lat', 'latitude', 'lon', 'longitude', 'latlon', 'lonlat', 'lat-lon', 'lon-lat'.

The output file is a tab-separated table with original coordinates and standard formattings of them

4.3. pyr8s

Example files provided:

pyr8s_examplefile1_Blommersia_tree_iTaxoTools_0_1.tre
pyr8s_examplefile2_Blommersia_tree_iTaxoTools_0_1.nex

The two example files are based on a phylogenetic tree of Madagascar frogs of the genus *Blommersia*, including some specimens of *Blommersia transmarina* from Mayotte, Comoro islands. The trees are maximum likelihood trees with branch lengths representing molecular substitutions.

The first file only includes a tree in plain Newick format. It can be imported in pyr8s and run as is, or after setting custom calibrations on the nodes.

The second example is in Nexus format. It includes the same treefile but also a separate "rates" block. In this example, the age between the *Boophis* outgroup and *Blommersia* is fixed at 64 million years ago (mya) according to the estimate in timetree.org, and the age of the colonization of Mayotte (split between *B. transmarina* and *B. wittei*) is constrained by the geological age of the island at 8-12 mya. These settings are automatically applied when the file is loaded. For the syntax of the rates block, please read the section below.

More examples are available in the GitHub repository (https://github.com/iTaxoTools/pyr8s).

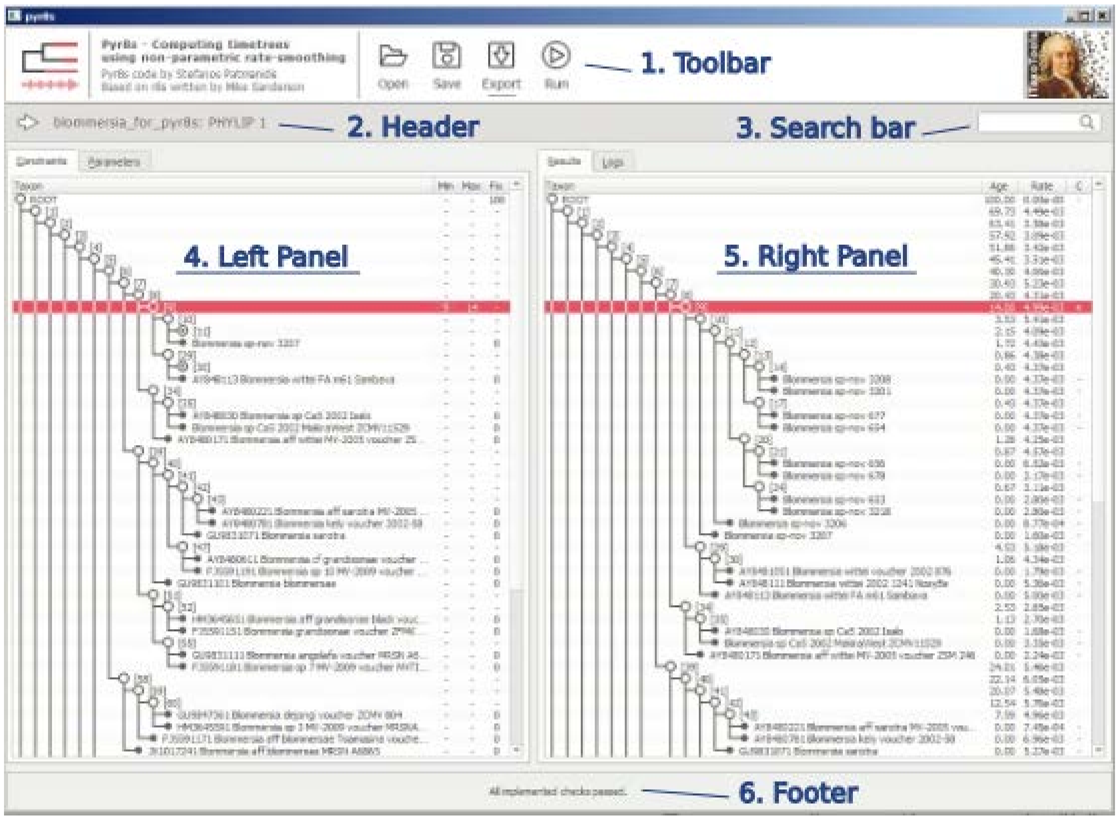
pyr8s is a tool for estimating divergence times and absolute rates ("r8s") of molecular evolution. It can be used to transform phylogenetic trees where the branch lengths are proportional to the number of substitutions (phylograms) into ultrametric timetrees (chronograms). One or more calibration points can be added to permit scaling of times and rates to real units. These calibrations can take one of two forms: assignment of a fixed age to a node, or enforcement of a minimum or maximum age constraint on a node, which is generally a better reflection of the information content of fossil evidence. Terminal nodes are permitted to occur at any point in time, allowing investigation of rate variation in phylogenies such as those obtained from "serial" samples of viral lineages through time.

The program implements nonparametric rate smoothing (NPRS, Sanderson 1997) for reconstructing divergence times. This method uses 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). NPRS relaxes the assumption of a molecular clock by using a least squares smoothing of local estimates of substitution rates. The optimality criterion is a sum of squared differences in local rate estimates compared from branch to neighboring branch. The resulting ultrametric trees are useful for many evolutionary analyses. They can also be used for species delimitation using GMYC; however, pyr8s at present may have limitations transforming trees containing many zero-length branches (e.g., from samples with identical sequences), and therefore potentially could lead to artifacts when such transformed trees are then used as input for GMYC.

pyr8s provides a Python implementation of techniques used in the program "r8s" (Sanderson 2003). This documentation is largely inspired by the original user‘s manual, which is made available on GitHub for further reference.

Graphical Interface:

The GUI exposes all of the module functionalities, explained in order of appearance:


pyr8s
Pyr8s - Computing timetrees using non-parametric rate-smoothing
Pyr8s code by Stefanos Patmanidis
Based on r8s written by Mike Sanderson
Open
Save
Export
Run
1. Toolbar
blommersia_for_pyr8s: PHYLIP 1
2. Header
3. Search bar
Constraints
Parameters
Taxon
Min
Max
Fix
4. Left Panel
Results
Logs
Age
Rate
5. Right Panel
All implemented checks passed.
6. Footer

1. Toolbar: Contains buttons for the following actions:

A. Open: The following input formats are accepted:

- Phylip/Newick (default):
  Open a plain phylogram with no parameters. Branch lengths must correspond to the number of substitutions along branches (or the numbers of substitutions per site along that branch). If many trees are listed, only the first is imported.

- Nexus (.nex extension):
  Must contain a phylogram (as above). If a rates/r8s block is present, import all constraints and parameters. If many trees are listed, only the first is imported.

- Pyr8s session (.r8s extension):
  Restore a previous session, saved in the form of a Python binary file.
  WARNING: Binary files are not secure. Only open files you trust.

B. Save: Store the current session, including tree, constraints, parameters and results (if any) in the form of a Python binary file (.r8s extension).

C. Export: Store analysis results in any of the available formats:

- Chronogram: An ultrametric Newick tree. Branch lengths correspond to time durations along branches.
- Ratogram: A Newick tree. Branch lengths correspond to absolute **rates** of substitutions along branches.
- Table: A tab-separated file with three columns for each node: name, age and rate of substitutions.

D. Run/Cancel: Begin an analysis on the open tree with the given contraints and parameters. Progress will be reported at the "Logs" tab in real time. You may interrupt an ongoing analysis (progress will be lost).

2. Header: The currently open filename and tree are listed here.

3. Search bar: Highlight taxa whose names contain your text.

4. Left Panel: Contains two tabs with the program input:

A. Constraints: One or more nodes can have their age fixed or constrained prior to estimating divergence times. Time is measured backwards from the most recent tip of the tree, which is assumed to have a time of 0 by default. In other words, times might best be thought of as "ages" with respect to the present. You may interactively set time constraints or edit the label of each node by double-clicking any field. The following columns are shown:

- Taxon: The name of the taxon. Nodes with no name show as "[X]", where X is the node index (starting in order from 0 at the root). Empty the field to clear the name.
- Fix: Any node in the tree, terminal or internal, can have its age fixed or free. A node that is fixed has an age that is set to a specific value by the user. A node that is free has an age that must be estimated by the program. By default, all internal nodes are assumed to be free, and all terminal nodes are assumed to be fixed, but any node can have its status changed.
- Min/Max: Any free node can be constrained by either a minimum or a maximum age. This is different from fixing a node’s age to a specific value. Computationally, it changes the optimization problem into one with bound constraints, which is much more difficult to solve.

Fixing the age of nodes generally makes it easier to estimate other ages or rates in the tree - that is, it reduces computation time. Roughly speaking, the divergence time algorithms require that at least one internal node in the tree be fixed, or that several nodes be constrained. If this is awkward, it is always possible to fix the root (or some other node) to have an arbitrary age of 100, and then all rates and times will be in units scaled by that age (this can be quickly achieved by setting the "Scalar" parameter). Under certain conditions, the age of unfixed nodes cannot be estimated uniquely. See further discussion under “Uniqueness of solutions” in the original r8s manual.

B. Parameters: You may fine-tune the underlying algorithms by adjusting the listed values. A new analysis must be run after changing any parameter in order to update the results. See the section below for a detailed explanation of each option.

5. Right Panel: Contains two tabs with the program output:

A. Results: Displays the final tree after successfully running an analysis. Selecting any row will highlight the corresponding taxon in the Constraints tab and vice versa. The results may be saved or exported, but not edited individually. Contains the following columns:

- Taxon: The name of the node as specified in the input.
- Age: The estimated age of the node.
- Rate: The estimated rate of substitution of the branch the node subtends.
- C: Indicates node mode: empty if free, dot if fixed, asterisk if constrained by min/max.

You may notice that there are less nodes in the results than in the starting tree. This is because all zero-length branches are removed and converted to hard polytomies. Terminal zero-length branches are ignored.

B. Logs: Collects all text output produced by the module and displays the progress of any ongoing analysis in real time.

6. Footer: Reports the program status. Displays a warning if the iteration limit was reached during analysis, in which case the results are unreliable. More quality checks from the original program are meant to be implemented in the future (notably missing is the gradient check of the results).

Parameters

The following options are split in categories and exposed to modification by the user:

Category	**Parameter**	**Default**	**Description**
General	Number of guesses	1	Number of initial starts using different initial times. Keep the best solution.
	Perturbation factor	0.01	Fractional perturbation of ages between each barrier iteration.
	Large value	1e+30	An infinitely large number for clamping.
	Seed	0	For the random number generator. Use system time if set to zero.
	Scalar	No	Force root age to 100 and ignore all other constraints.
Branch length	Format	Guess	One of three options: - Total: lengths have units of total numbers of substitutions. - Persite: lengths are in units of numbers of substitutions per site. - Guess: like Persite, but use rough estimate for "Subs per Site" based on maximum branch length.
	Subs per site	1	Number of sites in sequences that branch lengths on input trees were calculated from.
	Round	Yes	Round branch lengths to nearest integer.
Barrier	Manual	Yes	Use relaxed constrained optimization as described in the paper. Otherwise use scipy (problematic).
	Max iterations	10	Maximum allowed number of iterations for the relaxed constrained optimization.
	Initial factor	0.25	Initial barrier factor of the penalty function.
	Multiplier	0.1	Fractional decrease in barrier function on each barrier iteration.
	Tolerance	0.0001	Fractional tolerance in stopping criterion between barrier replicates.
Method	Method	NPRS	Objective function for optimization. Only Non-parametric rate smoothing implemented so far.
	Exponent	2	Exponent used for NPRS.
	Logarithmic	No	Logarithmic rate differences for NPRS.
Algorithm	Algorithm	Powell	Algorithm used for the optimization of the objective function.
	Variable tolerance	1e-08	Powell internal parameter.
	Function tolerance	1e-08	Powell internal parameter.

Nexus rates block

Pyr8s supports a subset of the nexus file commands specified in the r8s manual. This can be used to fix ages, set age constraints, set parameters, name clades and display text output. Specifically:

- All r8s commands must be found within a block labeled R8S or RATES.
- All commands are case-insensitive and follow the syntax: ***COMMAND*** {*OPTION*=VALUE};
- The following ***commands*** and *options* are recognised:

> ***BLFORMAT*** *NSITES*=<int> *LENGTHS*=TOTAL|PERSITE|GUESS *ROUND*=YES|NO *ULTRAMETRIC*=NO;
> ***MRCA*** <cladename> <nodename1> <nodename2>;
> ***FIXAGE*** *TAXON*=<nodename> *AGE*=<real>;
> ***UNFIXAGE*** *TAXON*=< nodename >;
> ***CONSTRAIN*** *TAXON*=< nodename > *MAXAGE|MAX_AGE* =<real> *MINAGE|MIN_AGE*=<real>;
> ***DIVTIME*** *METHOD*=NPRS|NP *ALGORITHM*=POWELL|PL;
> ***SET*** *NUM_TIME_GUESSES*=<int> *NPEXP*=<exponent> *PENALTY*=ADD|LOG *PERTURB_FACTOR*=<real> *MAXBARRIERITER*=<int> *BARRIERMULTIPLIER*=<real> *INITBARRIERFACTOR*=<real>;
> ***DESCRIBE*** *PLOT*= CHRONOGRAM|TREE_DESCRIPTION;
> ***SHOWAGE;***
> ***COLLAPSE***; (always executed)
> ***PRUNE***; (always executed)
> ***SCALAR***; (new, activates the "scalar" parameter)

Please refer to the provided example files and the r8s manual for more details.

Additional information

The pyr8s module can alternatively be called as a command-line tool or from within a Python script/interpreter. Please refer to the GitHub README file for more details.

4.4. fastmerge and fastsplit

Example files provided:

fastsplit_examplefile1_16SMantellidae_iTaxoTools_0_1.fas
fastsplit_examplefile2_BufoMicrobiome_iTaxoTools_0_1.fastq

We provide two relatively large files which can be used to test the splitting functions of fastsplit.
Example file 1 is a fasta file with sequences of the 16S rRNA gene of mantellid frogs.
Example file 2 is a fastq file from Illumina amplicon sequencing (bacterial 16S rRNA) of the microbiome of a toad, *Bufo bufo.*

You can examine these files and their syntax in a text editor.

For testing the fastmerge program, you can use example file 1 together with other example files in the fasta format, such as those for the MolD tool.

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.

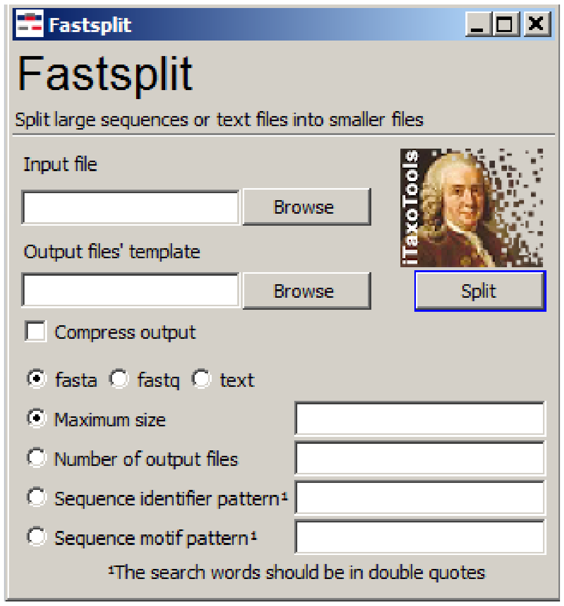
To use fastsplit, the user has to specify the input file and a template for the desired output files - the program will then produce several files with this name, and an automatic numbering included in the file names.

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).

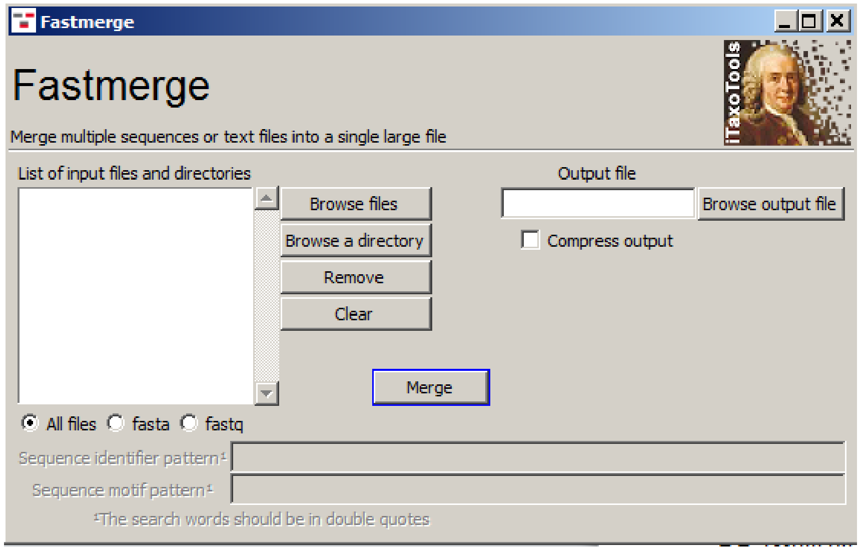
If fasta or fastq file types are specified, fastsplit will make sure to avoid splitting inbetween sequences, i.e., each file will always start with the sequence name (which is preceded by > in fasta, and @ in fastq).

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 specifed 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 me 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.


Fastmerge
Fastmerge
Merge multiple sequences or text files into a single large file
List of input files and directories
Browse files
Browse a directory
Remove
Clear
Merge
Output file
Browse output file
Compress output
All files
fasta
fastq
Sequence identifier pattern¹
Sequence motif pattern¹
¹The search words should be in double quotes
iTaxoTools

Additional information on the code and command-line version of the tools

Generating an executable

Using PyInstaller is recommended. After the following instruction a directory dist will be created (among other) and the executable will be inside it.

Linux
Install PyInstaller from PyPI:
```
pip install pyinstaller
```
Then run
```
pyinstaller --onefile fastmerge.py
```

Windows
Install PyInstaller:
Then run
```
pyinstaller --onefile --windowed fastmerge.py
```

Fastmerge, usage

```
usage: fastmerge.py [-h] [--cmd] [--fasta | --fastq] [--seqid PATTERN]
                    [--sequence PATTERN]
       fastmerge.py

optional arguments:
  -h, --help          show this help message and exit
  --cmd               Launches in the command-line mode
  --fasta             Process only .fas and .fas.gz files
  --fastq             Process only .fq, .fq.gz, .fastq and .fastq.gz files
  --seqid PATTERN     Filter pattern for sequence names
  --sequence PATTERN  Filter pattern for sequences
```

Command-line interface

Fastmerge reads the list of files and directories from the standard input and merges them into one file. For each directory, it merges the files inside it. It uncompresses the gzip archives if necessary. The output is written to the standard output.
When --seqid or --sequence and either --fasta or --fastq options are given, only the sequence records matching the patterns are written to the output.

Fastsplit, usage

```
usage: fastsplit.py [-h] [--fasta | --fastq | --text] [--split_n SPLIT_N | --maxsize MAXSIZE | --seqid
PATTERN | --sequence PATTERN] [--compressed]
                    [infile] [outfile]

positional arguments:
  infile              Input file name
  outfile             Output file template

optional arguments:
  -h, --help          show this help message and exit
  --fasta             Input file is a fasta file
  --fastq             Input file is a fastq file
  --text              Input file is a text file
  --split_n SPLIT_N   number of files to split into
  --maxsize MAXSIZE   Maximum size of output file
  --seqid PATTERN     split the records that match the sequence identifier pattern
  --sequence PATTERN  split the records that match the sequence motif pattern
  --compressed        Compress output files with gzip
```

Command-line interface

Fastsplit reads the input file and splits into files according to the options. Currently supported formats are FASTA and FastQ. Arbitrary text files can also be processed, but seqid and sequence options are not supportd for them. The spliting criteria are:

- Number of output parts (split_n)
- Limit on maximum size of parts (maxsize). The size is specified in in the format {number}{unit} where unit is one of 'bBkKmMgG'. Examples: 165b (165 bytes), 56K (56 kilobytes), 34M (34 Megabytes), 1.5g (1.5 Gigabytes)
- Split into two files: one with records matching a pattern and one with the remaining records:
    - Match the sequence identifier (seqid)
    - Match the motifs in the sequence (sequence)

Maximum size and the number of output files are only enforced approximately.

Filtering

A pattern consists of strings in double quotes, operators 'and', 'or' and 'not' (unquoted) and parentheses. It should be given in single quotes for the command-line interface and unquoted for the GUI.
Examples:

- "Boophis"
- not "Boophis"
- "Boophis" and "Madagascar"
- "Boophis" or "Madagascar"
- ("Boophis" or "Madagascar") and "Ranomafana"

4.5. specimentablemerger

Example files provided:

specimentablemerger_examplefile1_Uroplatus_Morphology_iTaxoTools_0_1.tab
specimentablemerger_examplefile2_Uroplatus_cox1_iTaxoTools_0_1.tab

The example files are tab-delimited text files containing morphological and genetic (COI sequence) data for geckos of the genus *Uroplatus*. You can open these files in a text editor, but we recommend inspecting them in a spreadsheet editor such as Excel.

Note that for three specimens in example file 1, the specimenid identifier is slightly misspelled:

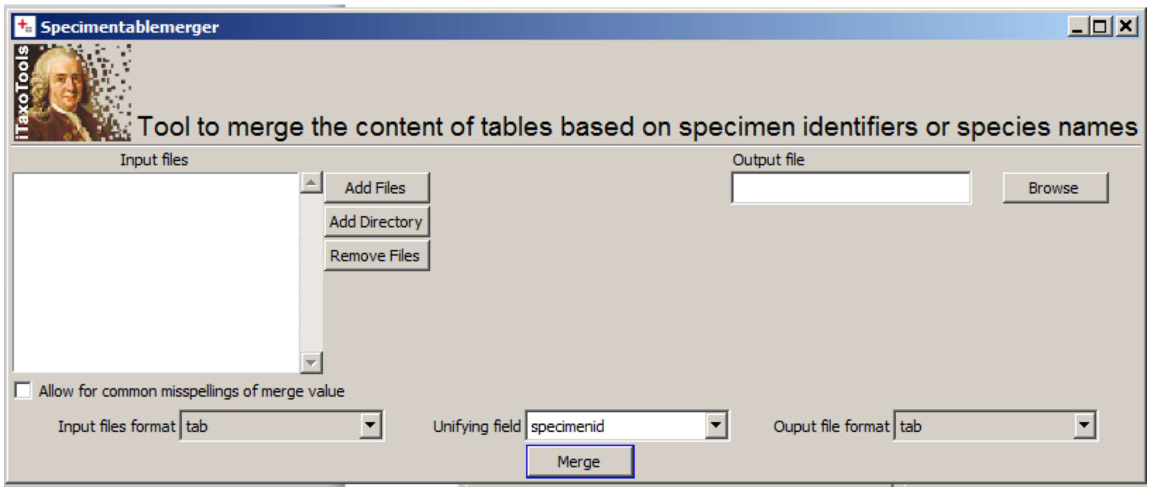
FGMV2002.2387 instead of FGMV 2002.2387 [space missing]
FGMV 2002-2388 instead of FGMV 2002.2388 [dash instead of period]
FGMV 2002-2390 instead of FGMV 2002.2390 [dash instead of period]

The rows of these specimens will not merged under the default settings of the program, but activating the checkbox "allow for common misspellings of merged value" will recognize these values as referring to the same specimenid.

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 are should be managed in dabases, 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 spreadsheets 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.



Specimentablemerger

Tool to merge the content of tables based on specimen identifiers or species names

Input files

Output file

☐ Allow for common misspellings of merge value

Input files format: tab

Unifying field: specimenid

Ouput file format: tab

4.6. specimentablepruner

Example files provided:

specimentablepruner_examplefile1_Gephyromantis_morphometry_iTaxoTools_0_1.tab
specimentablepruner_examplefile2_specimenids_for_pruning_iTaxoTools_0_1.txt

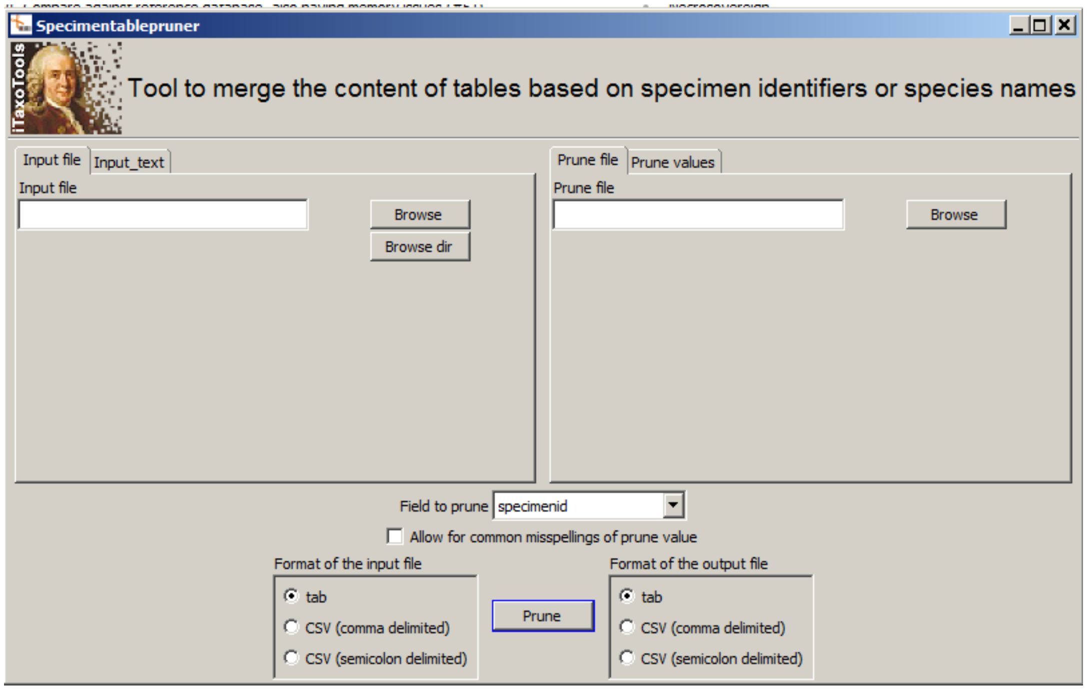
The first example file is a table with morphometric measurements of frogs of the genus *Gephyromantis*, in tab-delimited text format.
The second example file is a simple list with specimenid values for those specimens that do not have a reliable species identification and that we therefore maybe want to remove from the original table.

Run the program specifying the first example file as input file and the second example file as prune file.

You can examine these files in a text editor. For inspecting the first example file, we recommend using a spreadsheet editor such as Excel.

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.

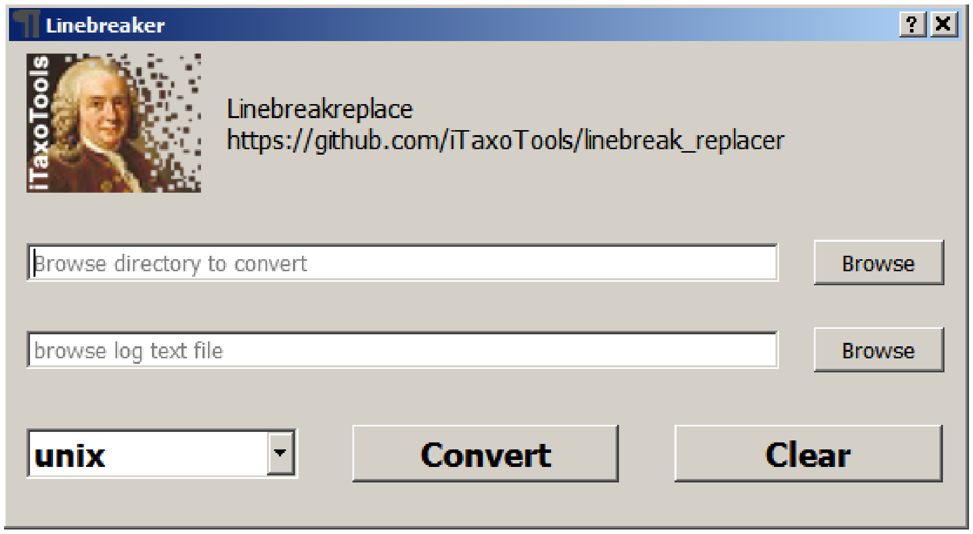
Also in this tool, an autocorrect option to correct common misspellings of specimen identifiers can be activated.



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.



Linebreakreplace
https://github.com/iTaxoTools/linebreak_replacer

Browse directory to convert

browse log text file

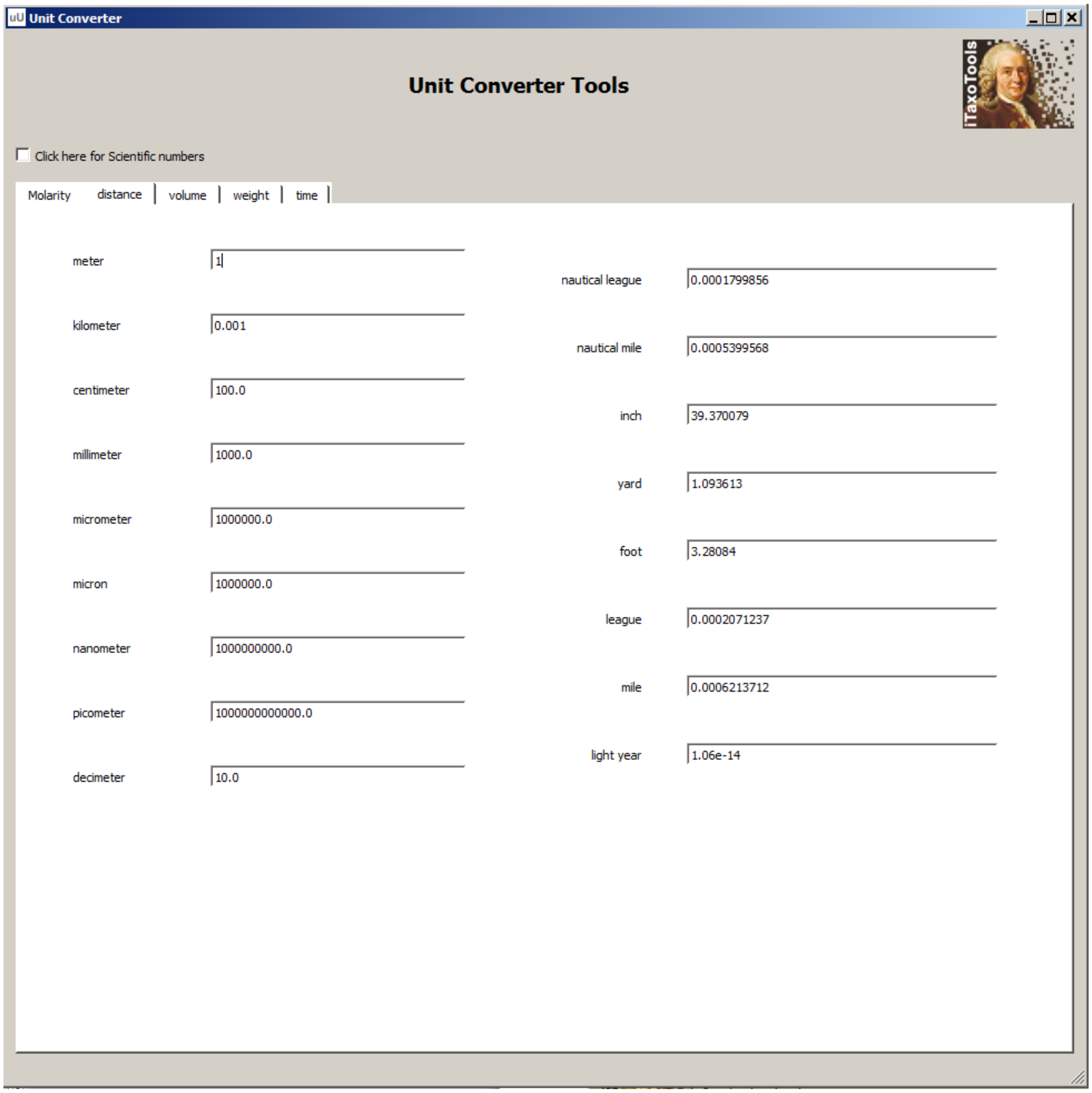
unix

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.



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

Example files provided:

Taxi2_examplefile1_50samples_iTaxoTools_0_1.tab
Taxi2_examplefile2_500samples_aligned_iTaxoTools_0_1.tab
Taxi2_examplefile3_MadaFrogs_cytb_references_iTaxoTools_0_1.tab

All example files are in tab-delimited text format. They can be inspected in a text editor but are best understood if opened in a spreadsheet editor such as Excel.

All example files represent data sets of cytochrome b sequences of frogs from Madagascar (family Mantellidae).

The first example file is a small set of unaligned sequences, including several replicates of the same species, that can be used as input file to run the program in the all-against all mode and explore the output (e.g., the histograms of genetic distances and summary statistics). Running this file should take at most one minute.

The second example file is a larger set of 500 sequences that are already aligned, so it can be run in the all-against-all mode with the "already aligned" option checked. Since in this mode, no pairwise algnments for all comparisons are needed, the large number of comparisons can also be computed in a reasonable amount of time of a few minutes. *Note that while running, the program may not be responsive – just be patient and wait for completion of the calculation. Also note that this dataset is basically composed of 10 concatenated copies of the first example file, so the overall output will not substantially differ.*

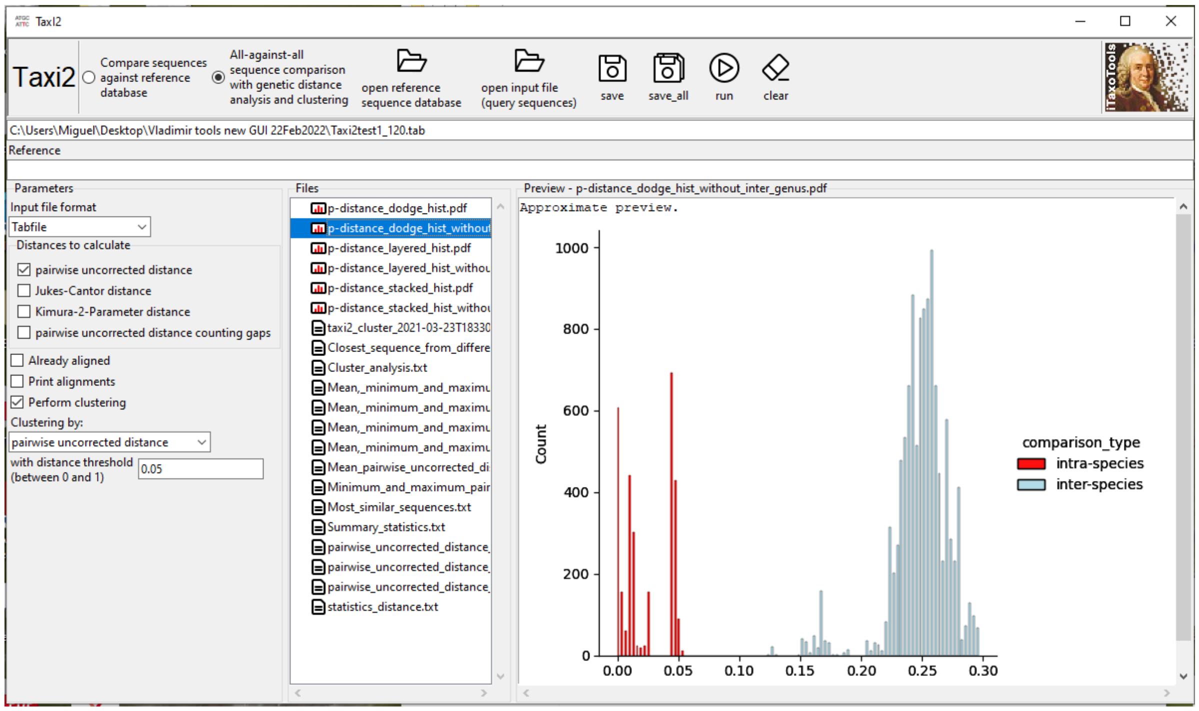
The third example file contains one cytochrome b sequence per species, for almost 300 species and candidate species of mantellid frogs. Therefore this file makes up a typical reference database. To run the program in the "compare against reference sequence database" mode, select that option with the respective radiobutton, define example file 3 as reference and example file 1 as query. The program will output information on the closest reference sequence for each sequence in the query set, based on pairwise alignments.
[Note: When switching between all-against-all comparisons and reference-comparison modes, it is suggested to close and reopen the program, to make sure the right mode is executed. Future versions of the program will include an overhauled GUI to better separate these two modes.]

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.



( ) Compare sequences against reference database [● All-against-all sequence comparison with genetic distance analysis and clustering]

C:\Users\Miguel\Desktop\Vladimir tools new GUI 22Feb2022\Taxi2test1_120.tab

Reference

Parameters

Input file format: Tabfile

Distances to calculate
- [✓] pairwise uncorrected distance
- [ ] Jukes-Cantor distance
- [ ] Kimura-2-Parameter distance
- [ ] pairwise uncorrected distance counting gaps

- [ ] Already aligned
- [ ] Print alignments
- [✓] Perform clustering

Clustering by: pairwise uncorrected distance

with distance threshold (between 0 and 1): 0.05

Files
- p-distance_dodge_hist.pdf
- p-distance_dodge_hist_without
- p-distance_layered_hist.pdf
- p-distance_layered_hist_withou
- p-distance_stacked_hist.pdf
- p-distance_stacked_hist_withou
- taxi2_cluster_2021-03-23T18330
- Closest_sequence_from_differe
- Cluster_analysis.txt
- Mean,_minimum_and_maximu
- Mean,_minimum_and_maximu
- Mean,_minimum_and_maximu
- Mean,_minimum_and_maximu
- Mean_pairwise_uncorrected_di
- Minimum_and_maximum_pair
- Most_similar_sequences.txt
- Summary_statistics.txt
- pairwise_uncorrected_distance_
- pairwise_uncorrected_distance_
- pairwise_uncorrected_distance_
- statistics_distance.txt

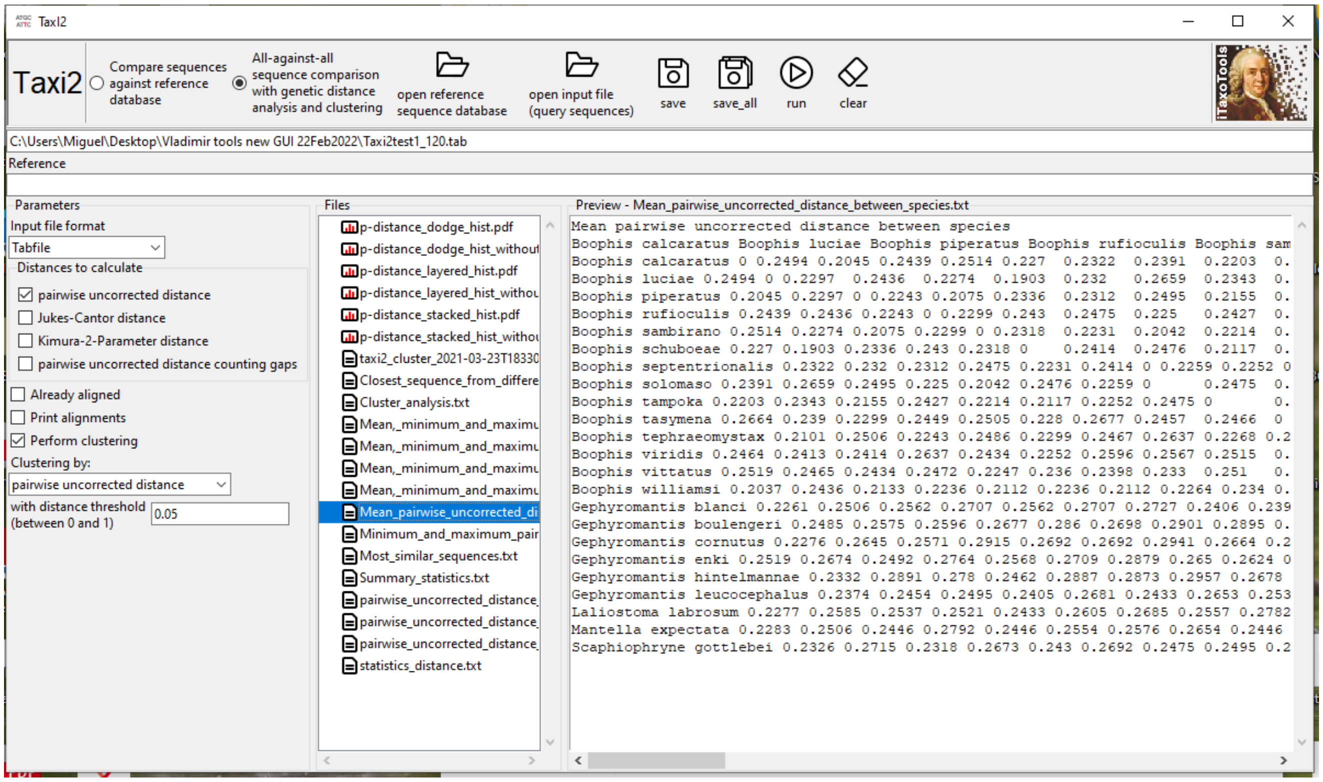
Preview - p-distance_dodge_hist_without_inter_genus.pdf

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.


TaxI2
Taxi2
Compare sequences against reference database
All-against-all sequence comparison with genetic distance analysis and clustering
open reference sequence database
open input file (query sequences)
save
save_all
run
clear
iTaxoTools

C:\Users\Miguel\Desktop\Vladimir tools new GUI 22Feb2022\Taxi2test1_120.tab

Reference

Parameters

Input file format: Tabfile

Distances to calculate
- [✓] pairwise uncorrected distance
- [ ] Jukes-Cantor distance
- [ ] Kimura-2-Parameter distance
- [ ] pairwise uncorrected distance counting gaps

- [ ] Already aligned
- [ ] Print alignments
- [✓] Perform clustering

Clustering by: pairwise uncorrected distance

with distance threshold (between 0 and 1): 0.05

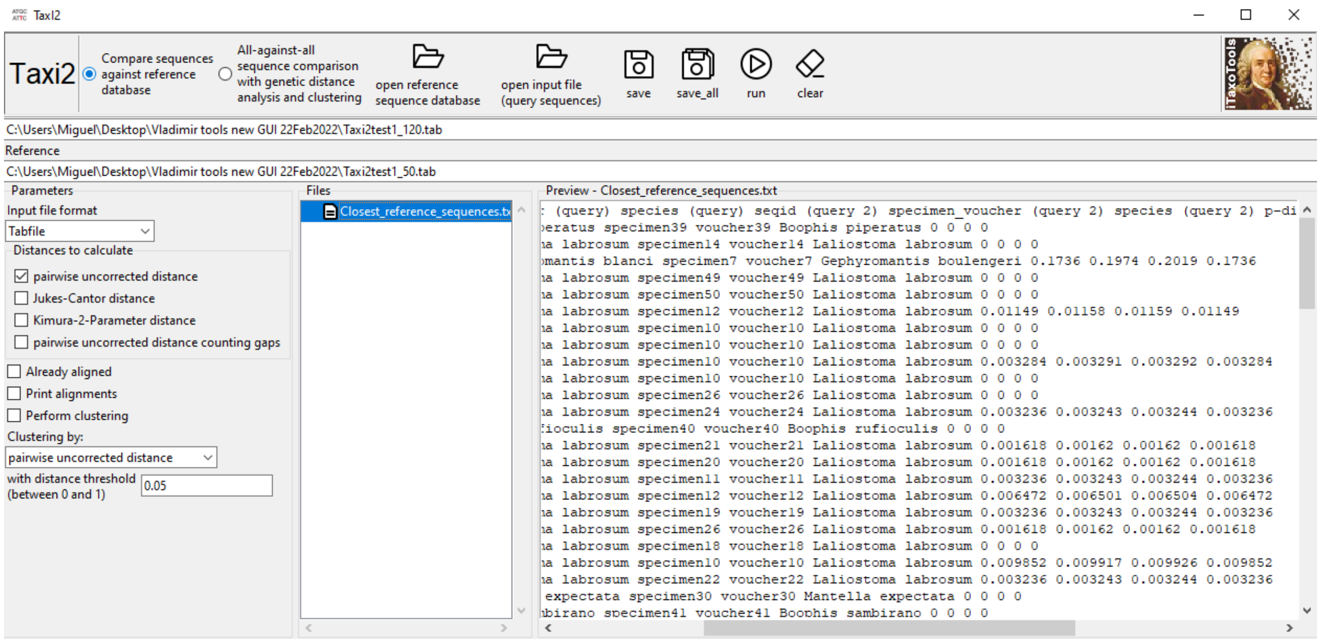
Files
- p-distance_dodge_hist.pdf
- p-distance_dodge_hist_without
- p-distance_layered_hist.pdf
- p-distance_layered_hist_withou
- p-distance_stacked_hist.pdf
- p-distance_stacked_hist_withou
- taxi2_cluster_2021-03-23T18330
- Closest_sequence_from_differe
- Cluster_analysis.txt
- Mean,_minimum_and_maximu
- Mean,_minimum_and_maximu
- Mean,_minimum_and_maximu
- Mean,_minimum_and_maximu
- Mean_pairwise_uncorrected_di
- Minimum_and_maximum_pair
- Most_similar_sequences.txt
- Summary_statistics.txt
- pairwise_uncorrected_distance_
- pairwise_uncorrected_distance_
- pairwise_uncorrected_distance_
- statistics_distance.txt

Preview - Mean_pairwise_uncorrected_distance_between_species.txt

```
Mean pairwise uncorrected distance between species
Boophis calcaratus Boophis luciae Boophis piperatus Boophis rufioculis Boophis sam
Boophis calcaratus 0 0.2494 0.2045 0.2439 0.2514 0.227  0.2322  0.2391  0.2203  0.
Boophis luciae 0.2494 0 0.2297  0.2436  0.2274  0.1903  0.232   0.2659  0.2343  0.
Boophis piperatus 0.2045 0.2297 0 0.2243 0.2075 0.2336  0.2312  0.2495  0.2155  0.
Boophis rufioculis 0.2439 0.2436 0.2243 0 0.2299 0.243  0.2475  0.225   0.2427  0.
Boophis sambirano 0.2514 0.2274 0.2075 0.2299 0 0.2318  0.2231  0.2042  0.2214  0.
Boophis schuboeae 0.227 0.1903 0.2336 0.243 0.2318 0    0.2414  0.2476  0.2117  0.
Boophis septentrionalis 0.2322 0.232 0.2312 0.2475 0.2231 0.2414 0 0.2259 0.2252 0
Boophis solomaso 0.2391 0.2659 0.2495 0.225 0.2042 0.2476 0.2259 0      0.2475  0.
Boophis tampoka 0.2203 0.2343 0.2155 0.2427 0.2214 0.2117 0.2252 0.2475 0       0.
Boophis tasymena 0.2664 0.239 0.2299 0.2449 0.2505 0.228 0.2677 0.2457  0.2466  0
Boophis tephraeomystax 0.2101 0.2506 0.2243 0.2486 0.2299 0.2467 0.2637 0.2268 0.2
Boophis viridis 0.2464 0.2413 0.2414 0.2637 0.2434 0.2252 0.2596 0.2567 0.2515  0.
Boophis vittatus 0.2519 0.2465 0.2434 0.2472 0.2247 0.236 0.2398 0.233  0.251   0.
Boophis williamsi 0.2037 0.2436 0.2133 0.2236 0.2112 0.2236 0.2112 0.2264 0.234 0.
Gephyromantis blanci 0.2261 0.2506 0.2562 0.2707 0.2562 0.2707 0.2727 0.2406 0.239
Gephyromantis boulengeri 0.2485 0.2575 0.2596 0.2677 0.286 0.2698 0.2901 0.2895 0.
Gephyromantis cornutus 0.2276 0.2645 0.2571 0.2915 0.2692 0.2692 0.2941 0.2664 0.2
Gephyromantis enki 0.2519 0.2674 0.2492 0.2764 0.2568 0.2709 0.2879 0.265 0.2624 0
Gephyromantis hintelmannae 0.2332 0.2891 0.278 0.2462 0.2887 0.2873 0.2957 0.2678
Gephyromantis leucocephalus 0.2374 0.2454 0.2495 0.2405 0.2681 0.2433 0.2653 0.253
Laliostoma labrosum 0.2277 0.2585 0.2537 0.2521 0.2433 0.2605 0.2685 0.2557 0.2782
Mantella expectata 0.2283 0.2506 0.2446 0.2792 0.2446 0.2554 0.2576 0.2654 0.2446
Scaphiophryne gottlebei 0.2326 0.2715 0.2318 0.2673 0.243 0.2692 0.2475 0.2495 0.2
```

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.


TaxI2
Taxi2
Compare sequences against reference database
All-against-all sequence comparison with genetic distance analysis and clustering
open reference sequence database
open input file (query sequences)
save
save_all
run
clear
iTaxoTools

C:\Users\Miguel\Desktop\Vladimir tools new GUI 22Feb2022\Taxi2test1_120.tab

Reference

C:\Users\Miguel\Desktop\Vladimir tools new GUI 22Feb2022\Taxi2test1_50.tab

Parameters

Input file format: Tabfile

Distances to calculate

- [✓] pairwise uncorrected distance
- [ ] Jukes-Cantor distance
- [ ] Kimura-2-Parameter distance
- [ ] pairwise uncorrected distance counting gaps

- [ ] Already aligned
- [ ] Print alignments
- [ ] Perform clustering

Clustering by: pairwise uncorrected distance

with distance threshold (between 0 and 1): 0.05

Files

Closest_reference_sequences.txt

Preview - Closest_reference_sequences.txt

```
: (query) species (query) seqid (query 2) specimen_voucher (query 2) species (query 2) p-di
peratus specimen39 voucher39 Boophis piperatus 0 0 0 0
na labrosum specimen14 voucher14 Laliostoma labrosum 0 0 0 0
pmantis blanci specimen7 voucher7 Gephyromantis boulengeri 0.1736 0.1974 0.2019 0.1736
na labrosum specimen49 voucher49 Laliostoma labrosum 0 0 0 0
na labrosum specimen50 voucher50 Laliostoma labrosum 0 0 0 0
na labrosum specimen12 voucher12 Laliostoma labrosum 0.01149 0.01158 0.01159 0.01149
na labrosum specimen10 voucher10 Laliostoma labrosum 0 0 0 0
na labrosum specimen10 voucher10 Laliostoma labrosum 0 0 0 0
na labrosum specimen10 voucher10 Laliostoma labrosum 0.003284 0.003291 0.003292 0.003284
na labrosum specimen10 voucher10 Laliostoma labrosum 0 0 0 0
na labrosum specimen26 voucher26 Laliostoma labrosum 0 0 0 0
na labrosum specimen24 voucher24 Laliostoma labrosum 0.003236 0.003243 0.003244 0.003236
fioculis specimen40 voucher40 Boophis rufioculis 0 0 0 0
na labrosum specimen21 voucher21 Laliostoma labrosum 0.001618 0.00162 0.00162 0.001618
na labrosum specimen20 voucher20 Laliostoma labrosum 0.001618 0.00162 0.00162 0.001618
na labrosum specimen11 voucher11 Laliostoma labrosum 0.003236 0.003243 0.003244 0.003236
na labrosum specimen12 voucher12 Laliostoma labrosum 0.006472 0.006501 0.006504 0.006472
na labrosum specimen19 voucher19 Laliostoma labrosum 0.003236 0.003243 0.003244 0.003236
na labrosum specimen26 voucher26 Laliostoma labrosum 0.001618 0.00162 0.00162 0.001618
na labrosum specimen18 voucher18 Laliostoma labrosum 0 0 0 0
na labrosum specimen10 voucher10 Laliostoma labrosum 0.009852 0.009917 0.009926 0.009852
na labrosum specimen22 voucher22 Laliostoma labrosum 0.003236 0.003243 0.003244 0.003236
 expectata specimen30 voucher30 Mantella expectata 0 0 0 0
nbirano specimen41 voucher41 Boophis sambirano 0 0 0 0
```

<u>Alignment settings:</u>

TaxI2 can either work with pre-aligned sequences or performs pairwise alignments for all comparisons. For the pairwise alignments, default settings are as follows:
gap penalty (score to open a gap in the middle of a sequence) = -3
gap extend penalty (score to extend an existing gap in the middle of a sequence) = -1
end gap penalty (score to create a gap at an end of a sequence) = -1
end gap extend penalty (score to extend a gap at an end of a sequence) = -1
match score (score for matching nucleotides) = 1
mismatch score (score for non-matching nucleotides) = -1

In the Python version, these values can be changed in the text file `data/scores.tab`
For the compiled standalone tools, at present the user has no option to change the default values. The alignment values have been set based on our experience with relatively short (ca. 300-500 bp) data sets of both protein-coding (mitochondrial cox1) and ribosomal RNA (mitochondrial 16S rRNA) genes, and should work appropriately with both kinds of data. The next version of the program will include advanced options to change these alignment parameters and choose various further alignment alternatives. If you have doubts on the performance of the pairwise aligner, we suggest to use a pre-aligned input file (e.g., aligned with MAFFT).

5.2. morphometricanalyzer

Example files provided:

Morphometricanalyzer_examplefile1_iTaxoTools_0_1.tab
Morphometricanalyzer_examplefile2_iTaxoTools_0_1.tab

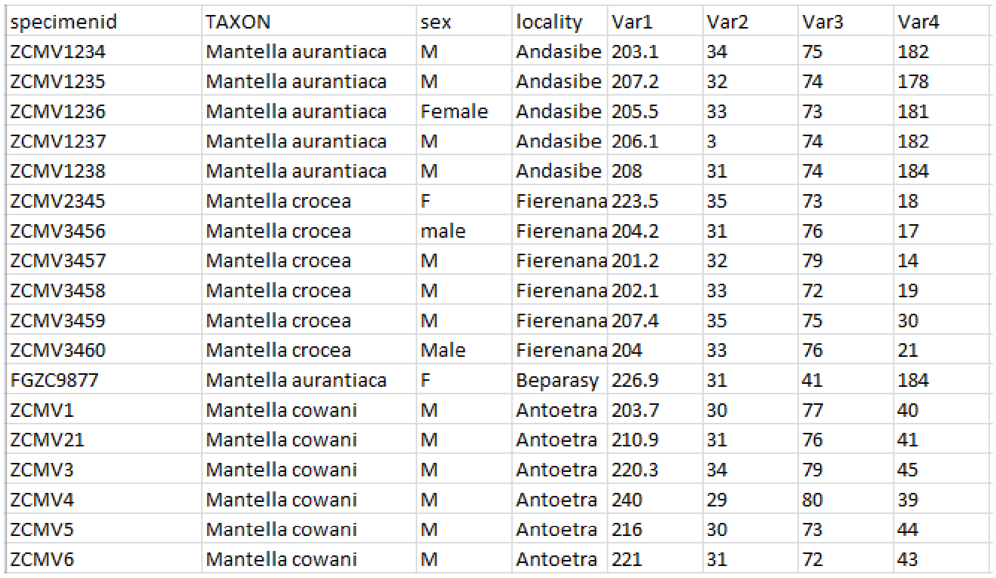
The two example files provided are tab-delimited text files presenting data that can be analyzed by the program. The files can be opened in a text editor but are best inspected and edited in a spreadsheet editor such as Excel.

Example file 1 is a very small mock-up file with a series of presumed measurements in three species of frogs of the genus Mantella. Run this file to get a general understanding of the output of the program.

Example file 2 is a real data set of morphometric measurements (all in mm) in a larger series of frogs of the genus Gephyromantis. This file can serve as a model of how input data for the program should be prepared. Run this file to explore the performance of the program more in depth.
[Note: the second example file also serves to illustrate some limitations of the current program output, such as the plots where the species label text will be overlapping; this however can easily be modified by importing the PDF output in a vector-based graphics program such as Illustrator or CorelDraw.

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)

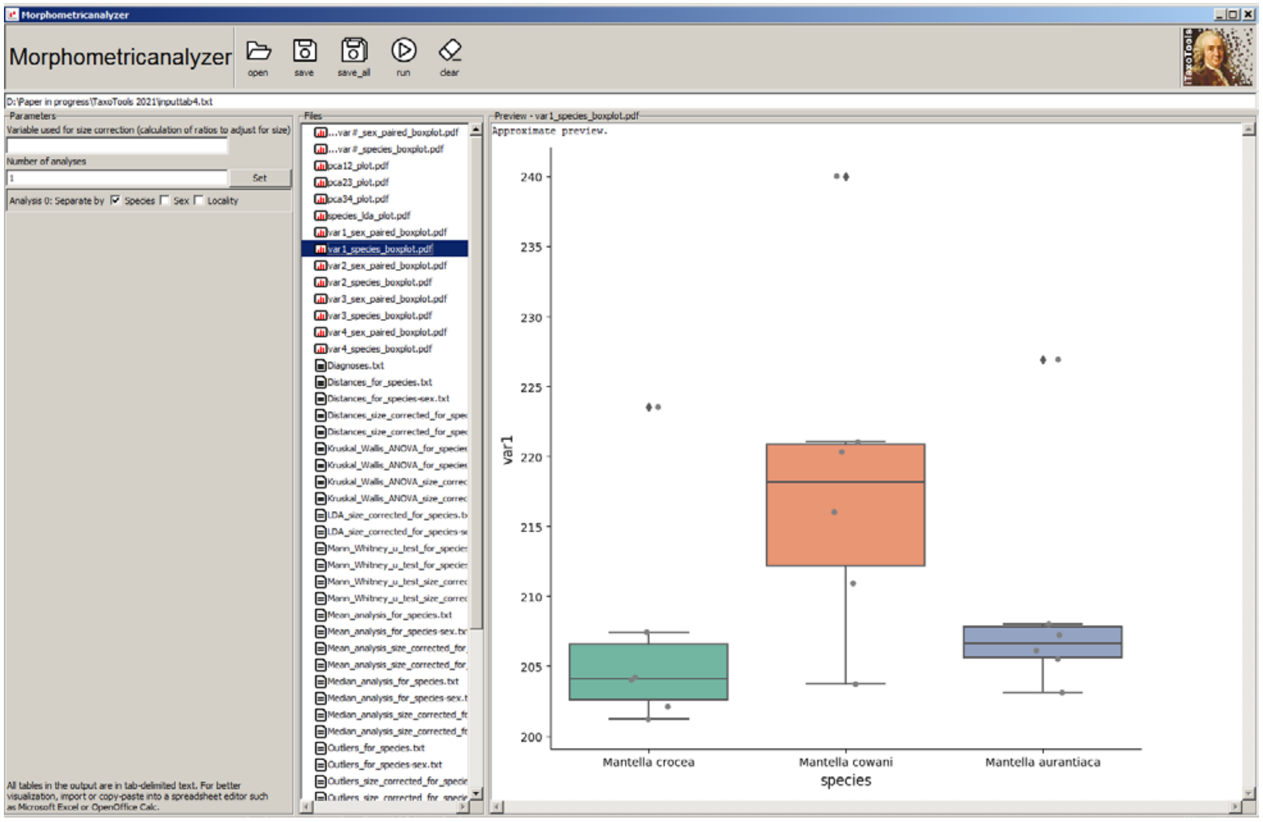
►Among-species and among-specimen matrices of morphometric distances (e.g., Euclidian).

All table output will be tab-delimited for easy copy-pasting into a spreadsheet editor. Furthermore, text output is also provided explaining in words the significant differences found between species.

The program will also take one variable (either the first one, or one specified by the user) as standard size variable, and will size-correct all others by calculating simple ratios to the size variable. Then, most of the statistical analyses above will be repeated with the size-corrected variables.

As several other tools in this collection, the program features an upper bar with buttons to open the input file, run the analyses, clear the results to start a new analysis, and save one or all output files after completion of the analysis.

After the analysis is completed, all out files will be listed in the Preview box, and by clicking on them they can be previewed in the right box.



D:\Paper in progress\TaxoTools 2021\inputtab4.txt

Variable used for size correction (calculation of ratios to adjust for size)

Number of analyses

1

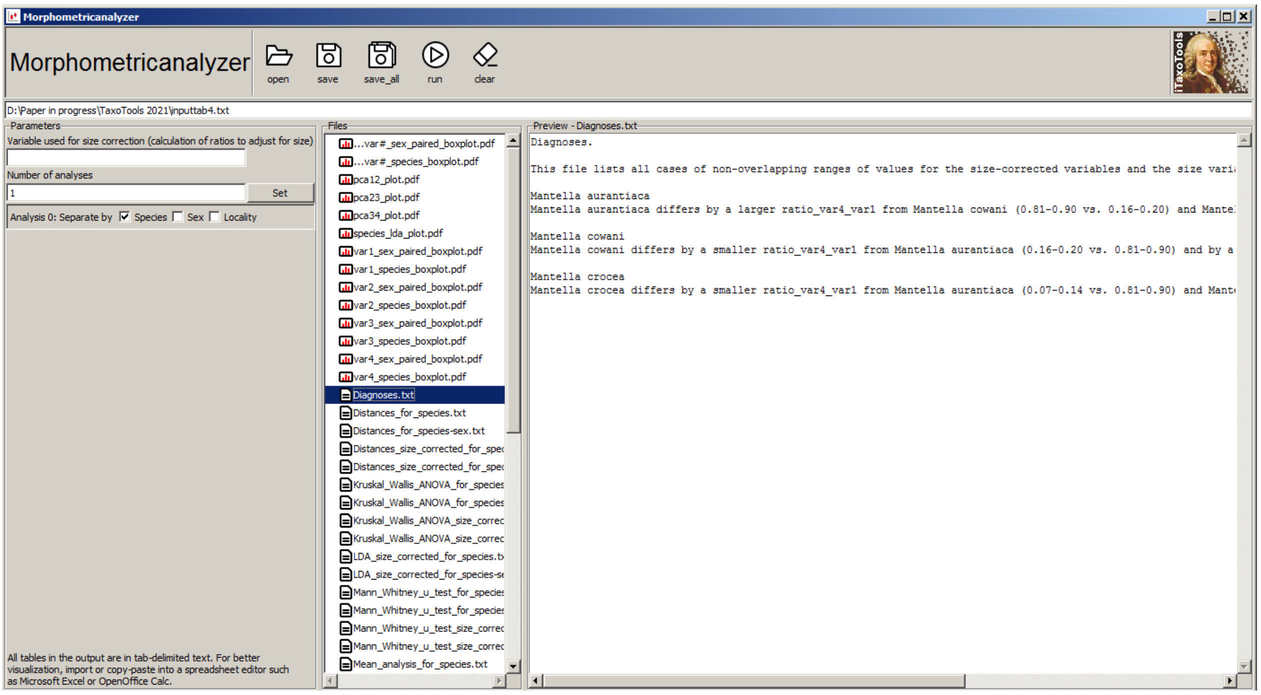
Analysis 0: Separate by [✓ Species] Sex Locality

All tables in the output are in tab-delimited text. For better visualization, import or copy-paste into a spreadsheet editor such as Microsoft Excel or OpenOffice Calc.

```
...var#_sex_paired_boxplot.pdf
...var#_species_boxplot.pdf
pca12_plot.pdf
pca23_plot.pdf
pca34_plot.pdf
species_lda_plot.pdf
var1_sex_paired_boxplot.pdf
var1_species_boxplot.pdf
var2_sex_paired_boxplot.pdf
var2_species_boxplot.pdf
var3_sex_paired_boxplot.pdf
var3_species_boxplot.pdf
var4_sex_paired_boxplot.pdf
var4_species_boxplot.pdf
Diagnoses.txt
Distances_for_species.txt
Distances_for_species-sex.txt
Distances_size_corrected_for_spec
Distances_size_corrected_for_spec
Kruskal_Wallis_ANOVA_for_species
Kruskal_Wallis_ANOVA_for_species
Kruskal_Wallis_ANOVA_size_correc
Kruskal_Wallis_ANOVA_size_correc
LDA_size_corrected_for_species.tx
LDA_size_corrected_for_species-se
Mann_Whitney_u_test_for_species
Mann_Whitney_u_test_for_species
Mann_Whitney_u_test_size_correc
Mann_Whitney_u_test_size_correc
Mean_analysis_for_species.txt
Mean_analysis_for_species-sex.tx
Mean_analysis_size_corrected_for
Mean_analysis_size_corrected_for
Median_analysis_for_species.txt
Median_analysis_for_species-sex.t
Median_analysis_size_corrected_fo
Median_analysis_size_corrected_fo
Outliers_for_species.txt
Outliers_for_species-sex.txt
Outliers_size_corrected_for_specie
Outliers_size_corrected_for_specie
```

```
Approximate preview.
```

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.



D:\Paper in progress\TaxoTools 2021\inputtab4.txt

Parameters

Variable used for size correction (calculation of ratios to adjust for size)

Number of analyses: 1 Set

Analysis 0: Separate by [✓ Species] Sex Locality

All tables in the output are in tab-delimited text. For better visualization, import or copy-paste into a spreadsheet editor such as Microsoft Excel or OpenOffice Calc.

Files

- ...var#_sex_paired_boxplot.pdf
- ...var#_species_boxplot.pdf
- pca12_plot.pdf
- pca23_plot.pdf
- pca34_plot.pdf
- species_lda_plot.pdf
- var1_sex_paired_boxplot.pdf
- var1_species_boxplot.pdf
- var2_sex_paired_boxplot.pdf
- var2_species_boxplot.pdf
- var3_sex_paired_boxplot.pdf
- var3_species_boxplot.pdf
- var4_sex_paired_boxplot.pdf
- var4_species_boxplot.pdf
- Diagnoses.txt
- Distances_for_species.txt
- Distances_for_species-sex.txt
- Distances_size_corrected_for_spe
- Distances_size_corrected_for_spe
- Kruskal_Wallis_ANOVA_for_species
- Kruskal_Wallis_ANOVA_for_species
- Kruskal_Wallis_ANOVA_size_correc
- Kruskal_Wallis_ANOVA_size_correc
- LDA_size_corrected_for_species.t
- LDA_size_corrected_for_species-s
- Mann_Whitney_u_test_for_species
- Mann_Whitney_u_test_for_species
- Mann_Whitney_u_test_size_correc
- Mann_Whitney_u_test_size_correc
- Mean_analysis_for_species.txt

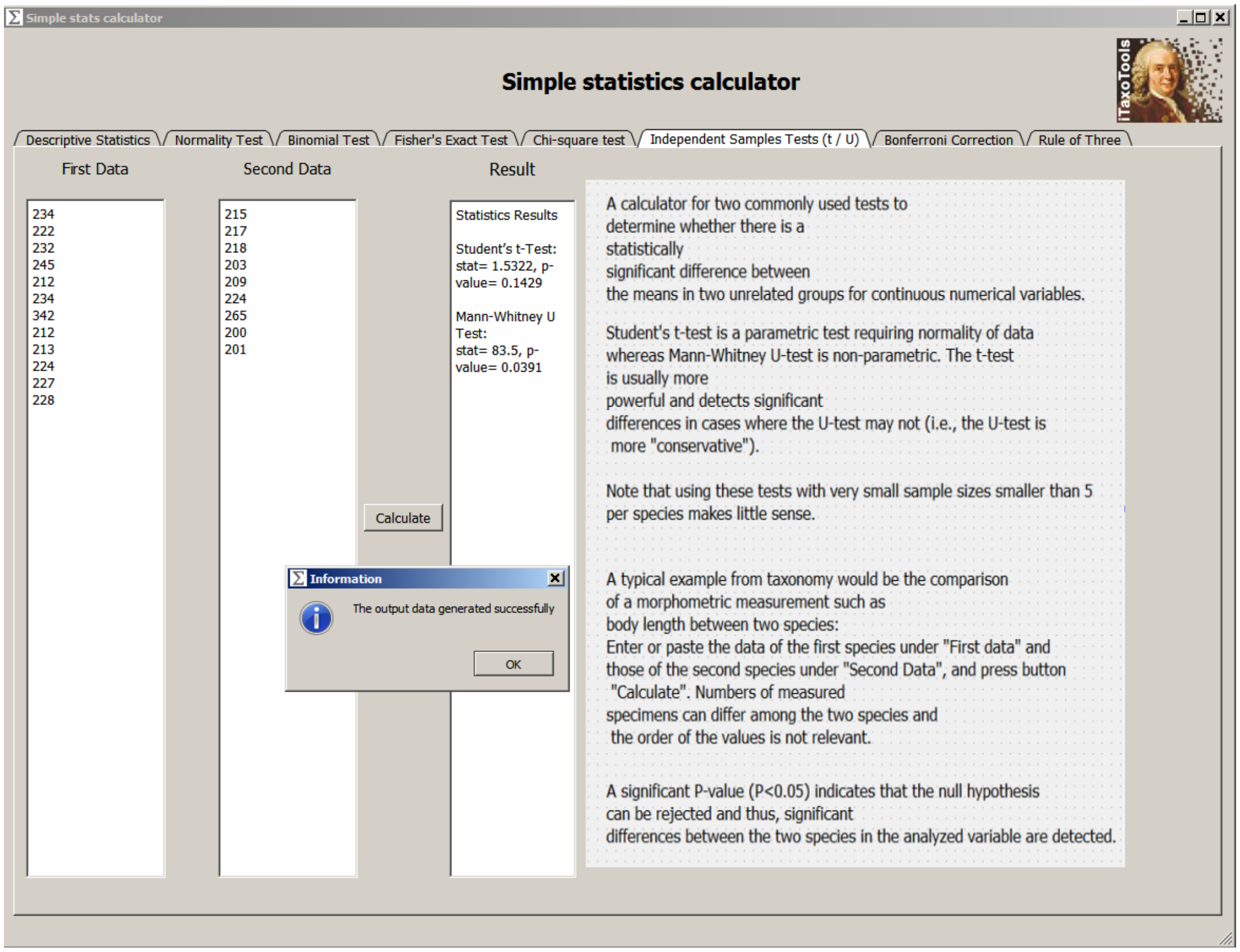
Preview - Diagnoses.txt

```
Diagnoses.

This file lists all cases of non-overlapping ranges of values for the size-corrected variables and the size vari

Mantella aurantiaca
Mantella aurantiaca differs by a larger ratio_var4_var1 from Mantella cowani (0.81-0.90 vs. 0.16-0.20) and Mante

Mantella cowani
Mantella cowani differs by a smaller ratio_var4_var1 from Mantella aurantiaca (0.16-0.20 vs. 0.81-0.90) and by a

Mantella crocea
Mantella crocea differs by a smaller ratio_var4_var1 from Mantella aurantiaca (0.07-0.14 vs. 0.81-0.90) and Mant
```

5.3. simplestatscalculator

This program features a collection of simple tools for performing ad-hoc statistical tests. Each test is shown on a different tab, along with some explanation of the test, formulated for "absolute beginners".

The values to be tested can be written or pasted into the respective box or boxes, and the test result is shown after pushing the "Calculate" button.



First Data	Second Data
234	215
222	217
232	218
245	203
212	209
234	224
342	265
212	200
213	201
224	
227	
228	

```
Statistics Results

Student's t-Test:
stat= 1.5322, p-
value= 0.1429

Mann-Whitney U
Test:
stat= 83.5, p-
value= 0.0391
```

The output data generated successfully

A calculator for two commonly used tests to determine whether there is a statistically significant difference between the means in two unrelated groups for continuous numerical variables.

Student's t-test is a parametric test requiring normality of data whereas Mann-Whitney U-test is non-parametric. The t-test is usually more powerful and detects significant differences in cases where the U-test may not (i.e., the U-test is more "conservative").

Note that using these tests with very small sample sizes smaller than 5 per species makes little sense.

A typical example from taxonomy would be the comparison of a morphometric measurement such as body length between two species:
Enter or paste the data of the first species under "First data" and those of the second species under "Second Data", and press button "Calculate". Numbers of measured specimens can differ among the two species and the order of the values is not relevant.

A significant P-value (P<0.05) indicates that the null hypothesis can be rejected and thus, significant differences between the two species in the analyzed variable are detected.

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).

Warning: Species delimitation tools are intended to explore the most likely species partitions within a given dataset and under very specific assumptions. They implement various species boundary detection criteria, such as barcode gap, coalescence, or monophyly, and will therefore very likely provide disparate results.

We thus emphasize once more that the results inferred by a given method should therefore be considered with caution, and are to be regarded as an objective decision aid for partitioning individuals into **subsets which in many cases will correspond to populations rather than species (see: Sukumaran & Knowles 2017).** For drawing reliable taxonomic conclusions, they must always be interpreted along with complementary investigative approaches. Combining these results from different datasets and methods, and to interpret them in an integrative taxonomic approach is generally a good idea.

Reference:

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.

6.1. ABGD

Example file provided:

ABGD_examplefile1_LophiotomaCOI_iTaxoTools_0_1.fas

[*Note: in the screenshots below, the same file is used with the simple name "Lophiotoma.fas"*]

This dataset includes 276 COI sequences of marine gastropods (Conoidea, Turridae, *Lophiotoma*) published in Puillandre et al (Puillandre N, Fedosov AE, Zaharias P, Aznar-Cormano L, & Kantor YI. 2017. A quest for the lost types of *Lophiotoma* (Gastropoda: Conoidea: Turridae): integrative taxonomy in a nomenclatural mess. Zoological Journal of the Linnean Society 181: 243‑71). Each sequence is labelled with a unique voucher number (from the MNHN, Paris), followed by the species name, as delimited in the corresponding article.

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) and wrapped around the original ABGD code, to make the program more easily accessible to users, also in offline situations.

Quick start

ABGD takes as input a set of aligned sequences (e.g. in fasta format) (Open) and then calculates pairwise distances, or directly a matrix of pairwise genetic distances. ABGD uses a coalescent model to identify the most likely position of the barcode gap, based on maximal genetic intraspecific distances defined a priori by the user, and uses the DNA barcoding gap to propose species partitions (Run). 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).

The obtained results can be exported in various formats (Save) according to the user needs (ex. .spart or .txt to export species partition (.spart is recommended), .svg for the diffferent vectorial graphics, .log for the statistics, or .tree for the tree topologies.

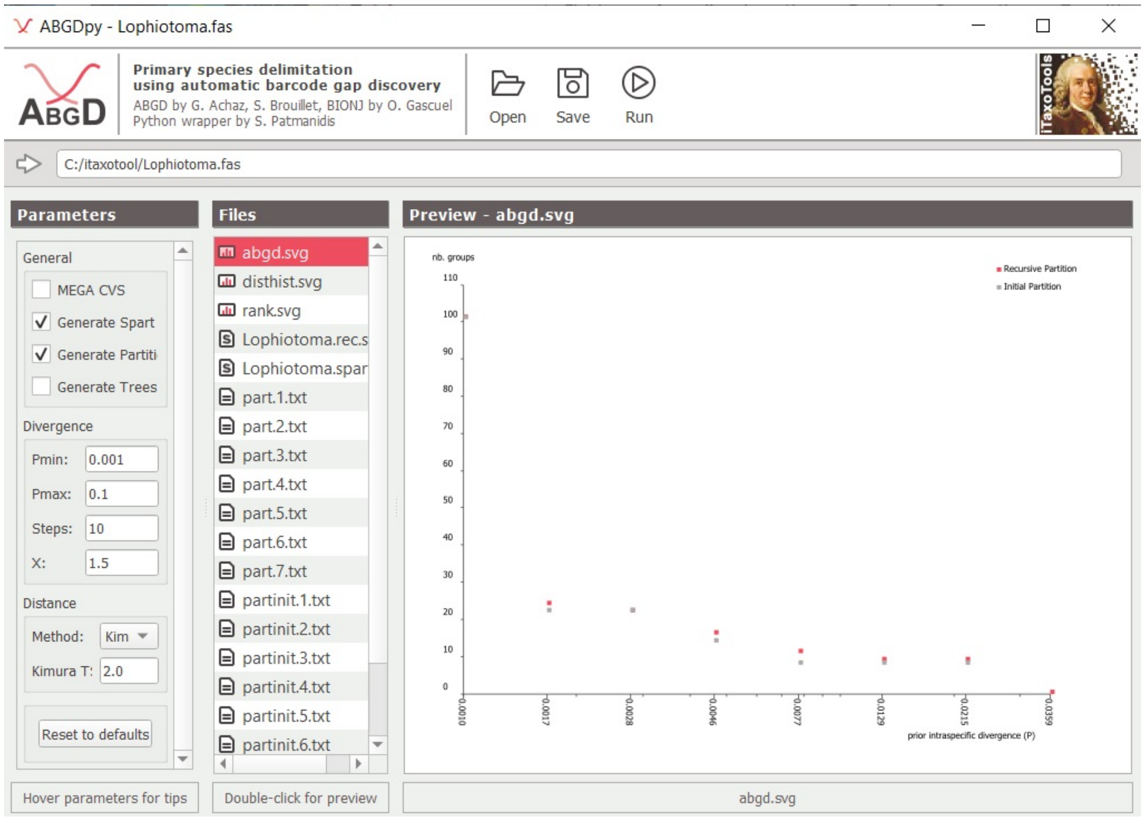
Parameters:

- Pmin and Pmax correspond to the minimum and maximum values of the parameter P, which corresponds to the genetic distance a priori defined by the user, and that ABGD will use to estimate the intraspecific variability and then the position of the barcode gap. By default, these values are 0.001 and 0.1, respectively, which means that the barcode gap is expected to be found within this range (thus corresponding to 0.1–10% pairwise sequence divergence).

- Steps corresponds to the number of values of P ABGD will test, within the range of values defined by Pmin and Pmax. By default, the value is 10, which means that ABGD will identify a barcode gap and provide a partition of species hypotheses for 10 values of P, regularly spaced within the range defined by Pmin and Pmax.

- X is a proxy for the minimum gap width. By default, it is set to 1.5, which means that ABGD will consider as the barcode gap the first local maximum slope of the ranked distances not included in the intraspecific distribution and 1.5 larger than any gap in the intraspecific distribution. In some datasets, X will need to be decreased so that ABGD will be more sensitive and able to detect a less important/pronounced barcode gap.

- distance refers to the substitution model used to calculate the genetic distances.



C:/itaxotool/Lophiotoma.fas

General
- [ ] MEGA CVS
- [✓] Generate Spart
- [✓] Generate Partiti
- [ ] Generate Trees

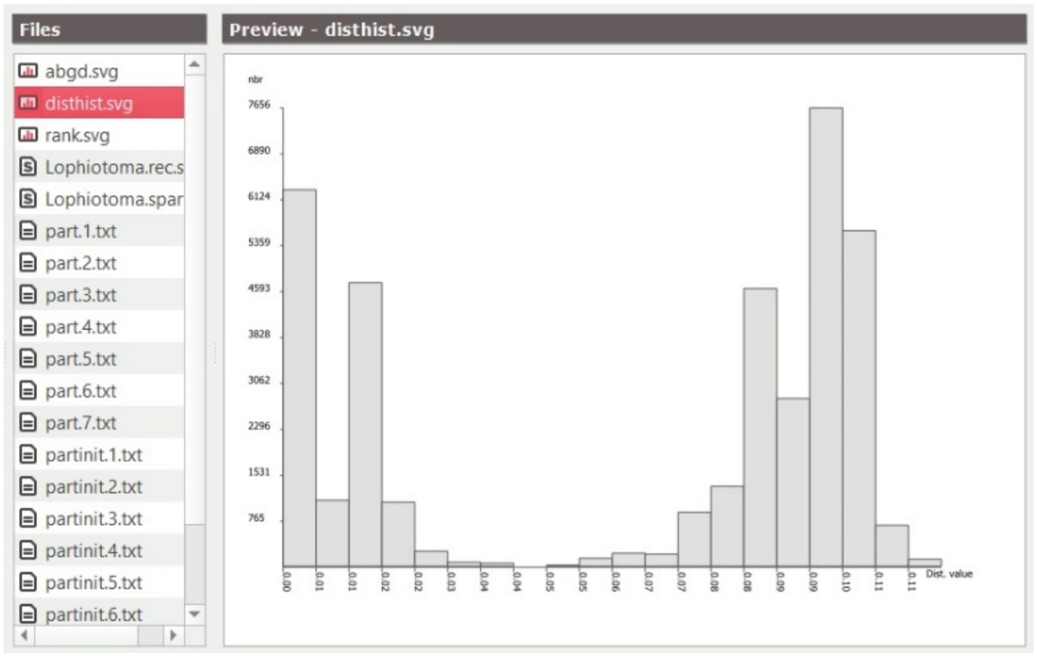
Divergence
Pmin: 0.001
Pmax: 0.1
Steps: 10
X: 1.5

Distance
Method: Kim
Kimura T: 2.0

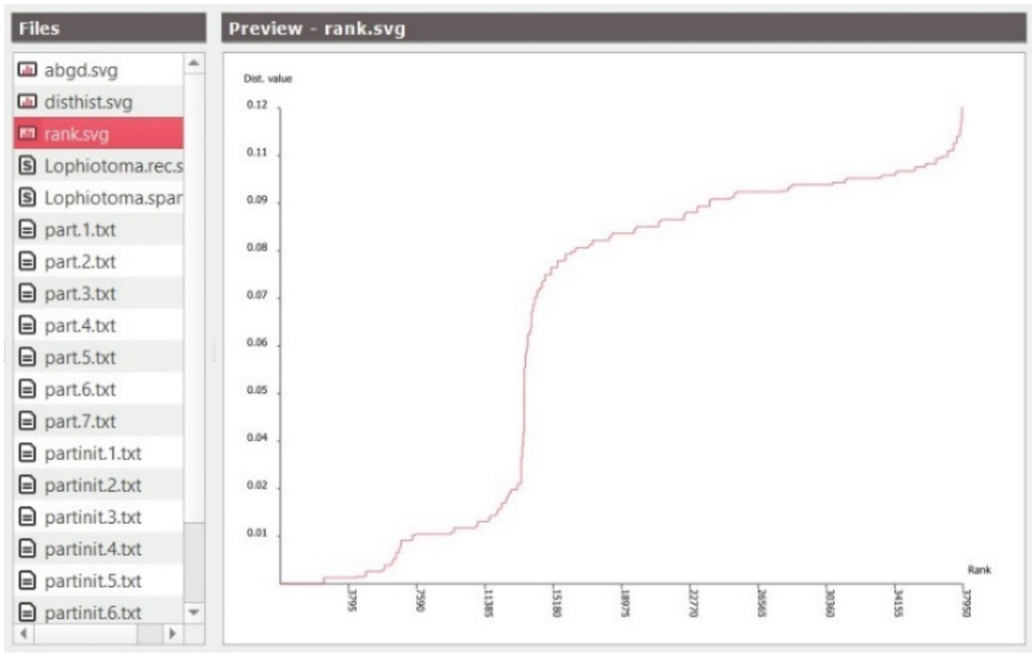
Files: abgd.svg, disthist.svg, rank.svg, Lophiotoma.rec.s, Lophiotoma.spar, part.1.txt, part.2.txt, part.3.txt, part.4.txt, part.5.txt, part.6.txt, part.7.txt, partinit.1.txt, partinit.2.txt, partinit.3.txt, partinit.4.txt, partinit.5.txt, partinit.6.txt

Graph "abgd" (shown above): on the X axis are reported the P values used by ABGD to identify the species partitions. For each P value, the number of species (Y axis) is reported, with the grey dots corresponding to the initial partition (ABGD run only once) and the red dots corresponding to the recursive partitions (ABGD is applied recursively on each species hypotheses defined in the initial partition, until no barcode gap is detected).

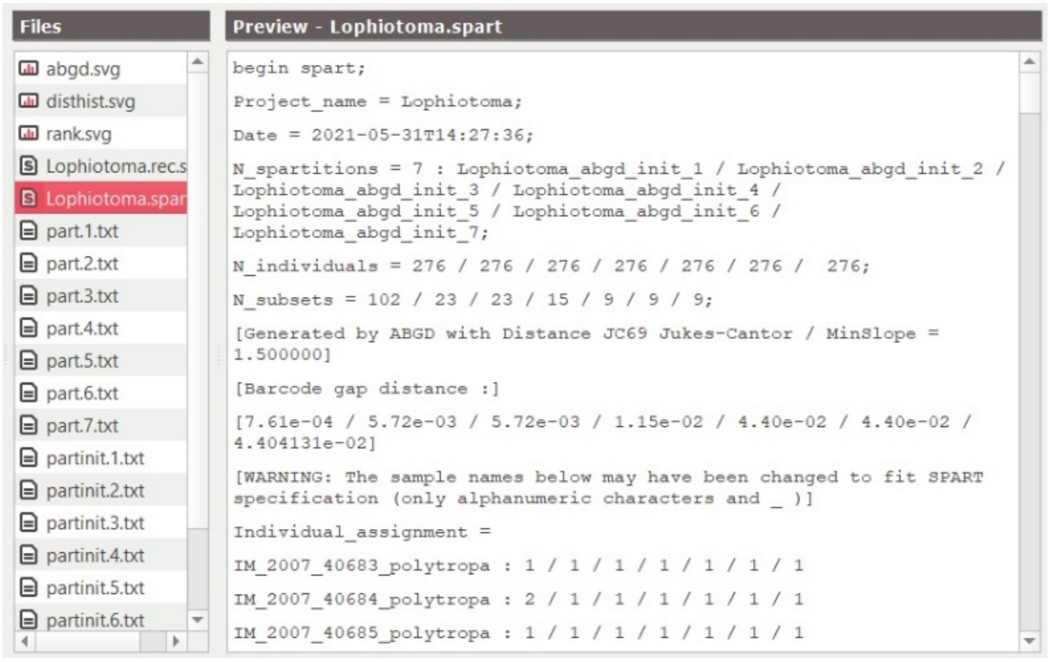
Furthermore, the program outputs a series of additional graphs and tables, as shown and explained on the next pages.



Graph "disthist": this histogram represents the distribution of the pairwise genetic distances, in which each bar represents the number of genetic distances (Y axis) found in the corresponding class of genetic distances (X axis).



Graph "rank": the genetic distances are ranked from the smallest to the largest. It typically results in a sigmoid curve, in which the inflexion point corresponds to the barcode gap in the distribution of genetic distances (cf. graph "disthist").



```
begin spart;
Project_name = Lophiotoma;
Date = 2021-05-31T14:27:36;
N_spartitions = 7 : Lophiotoma_abgd_init_1 / Lophiotoma_abgd_init_2 /
Lophiotoma_abgd_init_3 / Lophiotoma_abgd_init_4 /
Lophiotoma_abgd_init_5 / Lophiotoma_abgd_init_6 /
Lophiotoma_abgd_init_7;
N_individuals = 276 / 276 / 276 / 276 / 276 / 276 /  276;
N_subsets = 102 / 23 / 23 / 15 / 9 / 9 / 9;
[Generated by ABGD with Distance JC69 Jukes-Cantor / MinSlope =
1.500000]
[Barcode gap distance :]
[7.61e-04 / 5.72e-03 / 5.72e-03 / 1.15e-02 / 4.40e-02 / 4.40e-02 /
4.404131e-02]
[WARNING: The sample names below may have been changed to fit SPART
specification (only alphanumeric characters and _ )]
Individual_assignment =
IM_2007_40683_polytropa : 1 / 1 / 1 / 1 / 1 / 1 / 1
IM_2007_40684_polytropa : 2 / 1 / 1 / 1 / 1 / 1 / 1
IM_2007_40685_polytropa : 1 / 1 / 1 / 1 / 1 / 1 / 1
```

Lophiotoma.spart and Lophiotoma.rec.spart contains all the species partitions proposed by ABGD obtained with the initial and recursive approach, respectively, in the spart format.

The part.X.txt and partinit.X.txt files (not shown here) contains each one species partition, obtained with the Xth P value, with the recursive and initial approach, respectively.

The number of species suggested by ABGD for the *Lophiotoma* dataset varies from 9 to 102. The variance is of course huge, and deciding which partition is the correct one does not seem straightforward. As stated above, ABGD, as the other monolocus species delimitation methods, is not intended to directly provide to the user the final partition. Instead, ABGD typically reports a range of likely species partitions given the distribution of genetic distances. Then, only in the light of independent data the user will be able to evaluate these partitions.
In the corresponding article, the *Lophiotoma* dataset of *cox1* sequences has been jointly analyzed with a dataset of nuclear-encoded 28S sequences, and with morphological, bathymetrical and geographical data. It soon appeared that the partition with 9 species was the most likely, because the species defined in the other partitions (with 15, 23 or 102 species) were not recovered as monophyletic with the 28S gene, and were splitting into different species specimens that were homogeneous in terms of morphology and geographical and bathymetrical distributions.
However, one species defined by ABGD in the 9-species partition included two morphologically very distinct groups of specimens, furthermore corresponding to 2 monophyletic groups, albeit very closely related, in a phylogenetic tree obtained by concatenating the COI and 28S dataset. These two groups were furthermore sympatric, and we decided to consider them as two distinct species, even if ABGD did not recover them. We thus ended up with 10 different species.

6.2. ASAP

Example file provided:

ASAP_examplefile1_LophiotomaCOI_iTaxoTools_0_1.fas

[*Note: Same file as used as example for ABGD but renamed to make it easier to find in the collection of example files. In the screenshots below, the same file is used with the simple name "Lophiotoma.fas"*]

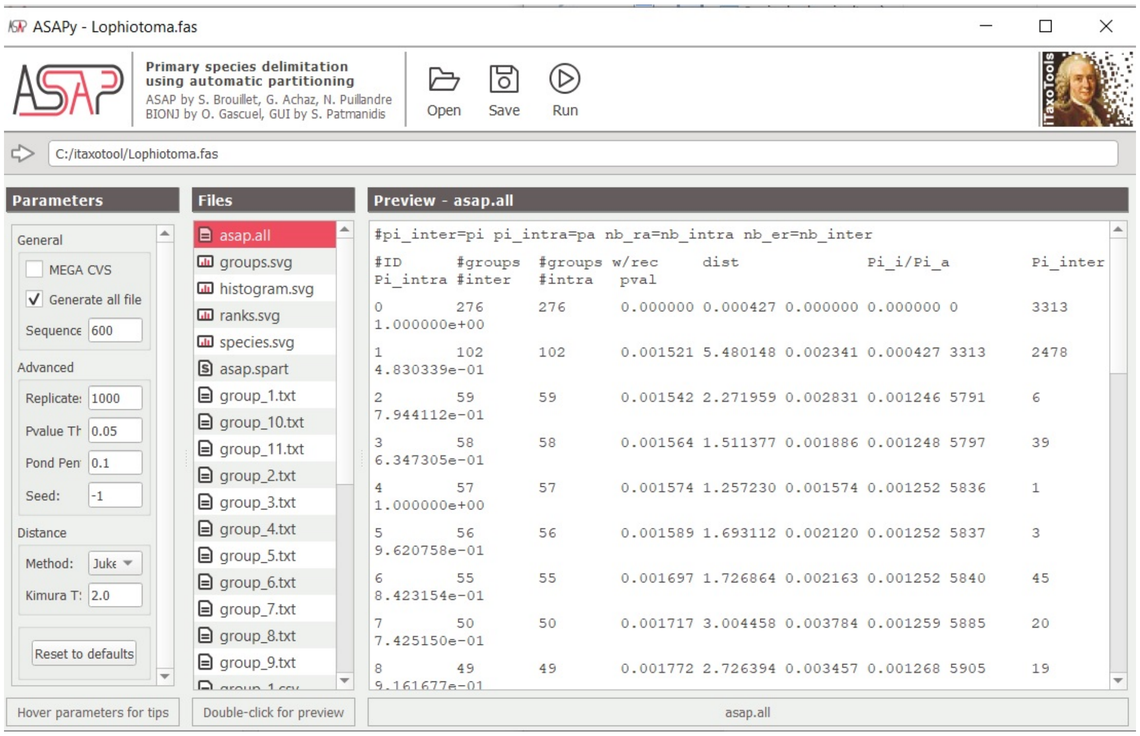
The Assemble Species by Automatic Partitioning (ASAP) approach to species delimitation has been introduced by Puillandre et al. (2021). ASAP is a tool designed to propose partitions of species hypotheses using genetic distances calculated between DNA sequences.

Similar to ABGD, the original code has been written in C, and we have complemented it 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. A FAQ webpage is also available at https://bioinfo.mnhn.fr/abi/public/asap/FAQ_asap.html

Contrary to 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.

ASAP can handle more than 10 000 sequences, but the computation time can be quite important in this case (several hours).

Quick start

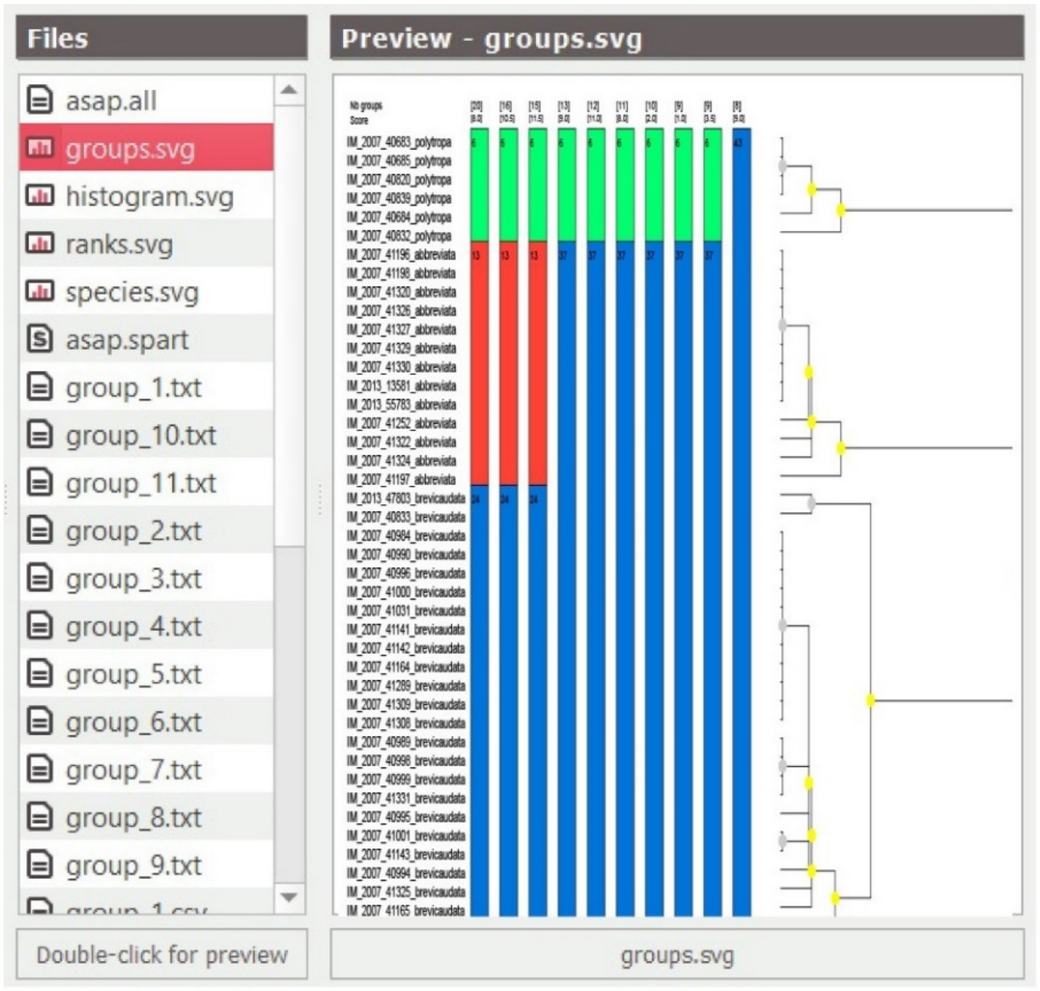


```
#pi_inter=pi pi_intra=pa nb_ra=nb_intra nb_er=nb_inter
#ID      #groups  #groups w/rec    dist               Pi_i/Pi_a          Pi_inter
Pi_intra #inter   #intra   pval
0        276      276      0.000000 0.000427 0.000000 0.000000 0        3313
1.000000e+00
1        102      102      0.001521 5.480148 0.002341 0.000427 3313     2478
4.830339e-01
2        59       59       0.001542 2.271959 0.002831 0.001246 5791     6
7.944112e-01
3        58       58       0.001564 1.511377 0.001886 0.001248 5797     39
6.347305e-01
4        57       57       0.001574 1.257230 0.001574 0.001252 5836     1
1.000000e+00
5        56       56       0.001589 1.693112 0.002120 0.001252 5837     3
9.620758e-01
6        55       55       0.001697 1.726864 0.002163 0.001252 5840     45
8.423154e-01
7        50       50       0.001717 3.004458 0.003784 0.001259 5885     20
7.425150e-01
8        49       49       0.001772 2.726394 0.003457 0.001268 5905     19
9.161677e-01
```

The fasta format is the most convenient format tob e used as input (Open). 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.

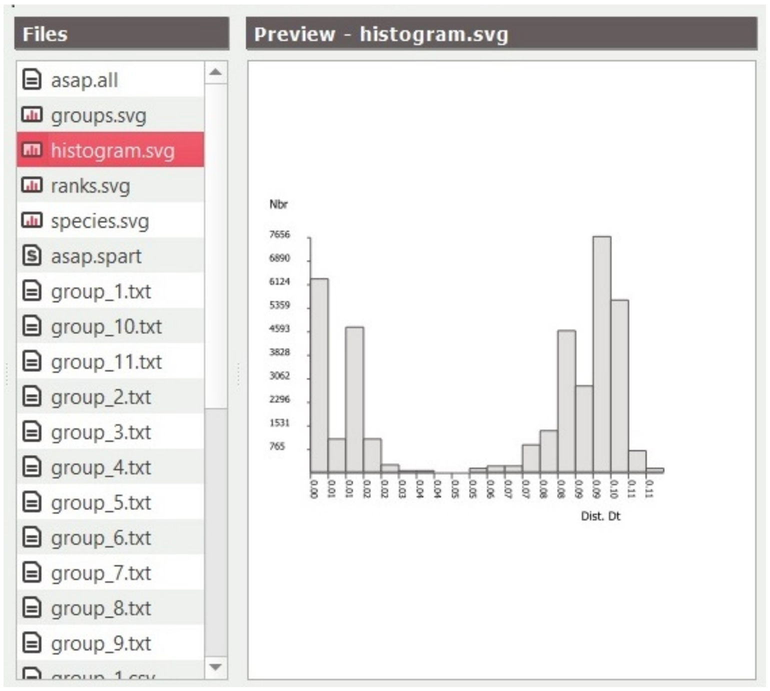
ASAP will then calculates pairwise distances (or takes as input directly a matrix of pairwise genetic distances). ASAP will then merge sequences into “groups” that are successively further merged until all sequences form a single group (Run). At each merging step, the newly created partition is characterized first by a *probability that quantifies the chances that each of its new groups is a single species;* and second by the difference between the barcode widths calculated for the previous and for this new partition. These two metrics (probability and barcode gap width) are then combined into a single *asap-score* that is used to rank the partitions: the lower the asap-score, the better the partition.

The obtained results can be exported in various formats (Save) according to the user needs: .spart or .txt to export species partition (.spart recommended), .svg for the diffferent vectorial graphics, .log for the statistics, or .tree for the tree topologies.

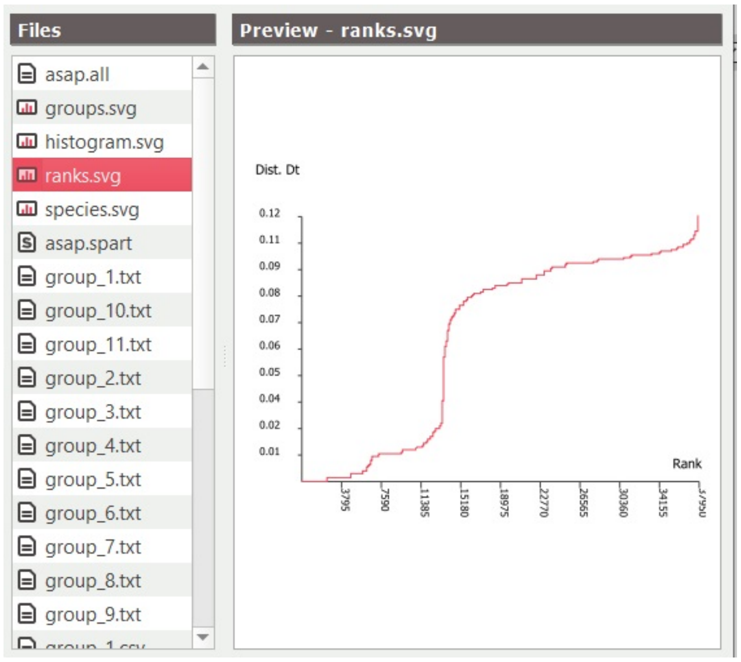


The above screenshot shows the graph ("groups") representing all the inferred partitions. The first column lists the sample ID. The N columns that follows correspond to the N best partitions proposed by ASAP, in which each colored box represent one species hypothesis. The N partitions are ranked from left to right from the less to the most inclusive, thus facilitating the comparison between the partitions. Furthermore, on the top of each column, the number of inferred species and the asap-score is reported.

On the right part of the graph, the NJ tree obtained by ASAP, illustrating the relationships between all the samples of the dataset. The colored dots on the NJ tree correspond to the probability that merging the groups within the node is compatible with the known distances inside each of these groups. This probability is calculated for each node; the darker the color of the dots, the lower the probability. 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. When the probability was not computed, the dot is grey.



Graph "histogram": this histogram represents the pairwise distribution of the genetic distances, in which each bar represents the number of genetic distances (Y axis) found in the corresponding class of genetic distances (X axis).



Graph "ranks":the genetic distances are ranked from the smallest to the largest. It typically results in a sigmoid curve, in which the inflexion point corresponds to the barcode gap in the distribution of genetic distances cf. (graph "disthist").

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 beetween 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.

As for ABGD, the number of species was not the same in the different partitions provided by ASAP. But contrary to ABGD, ASAP provides a score to each partition, and the two most likely partitions, as defined by the asap-score, are the partitions with 7 and 9 species. Once again, the independent data were mandatory to identify the correct partitions, and we ended up with 10 species (see the ABGD section for more details).

One advantage of ASAP over ABGD is also the graphical representation of the results: the vertical boxes in front oft he tree represent the species hypotheses provided in the best partitions, and the user can easily compare them and visualize e.g. which species are split or lump in different partitions. Note that the online version of the tool also offers an interactive visual interface to explore the different species partitions even better.

6.3. DELINEATE

Example files provided:

DELINEATE_examplefile1_lionepha_assignment_iTaxoTools_0_1.tsv
DELINEATE_examplefile2_lionepha_tree_iTaxoTools_0_1.nex

The two example files for the DELINEATE program constitute the two input files for the final step of the DELINEATE approach in the workflow tutorial of J. Sukumaran on the DELINEATE webpage (https://jeetsukumaran.github.io/delineate/workflow1.html). File 1 is a table where species assignment for some individuals in the analysis is fixed while others are left unassigned (to be assigned by the program). File 2 is a population-level tree in Nexus format.

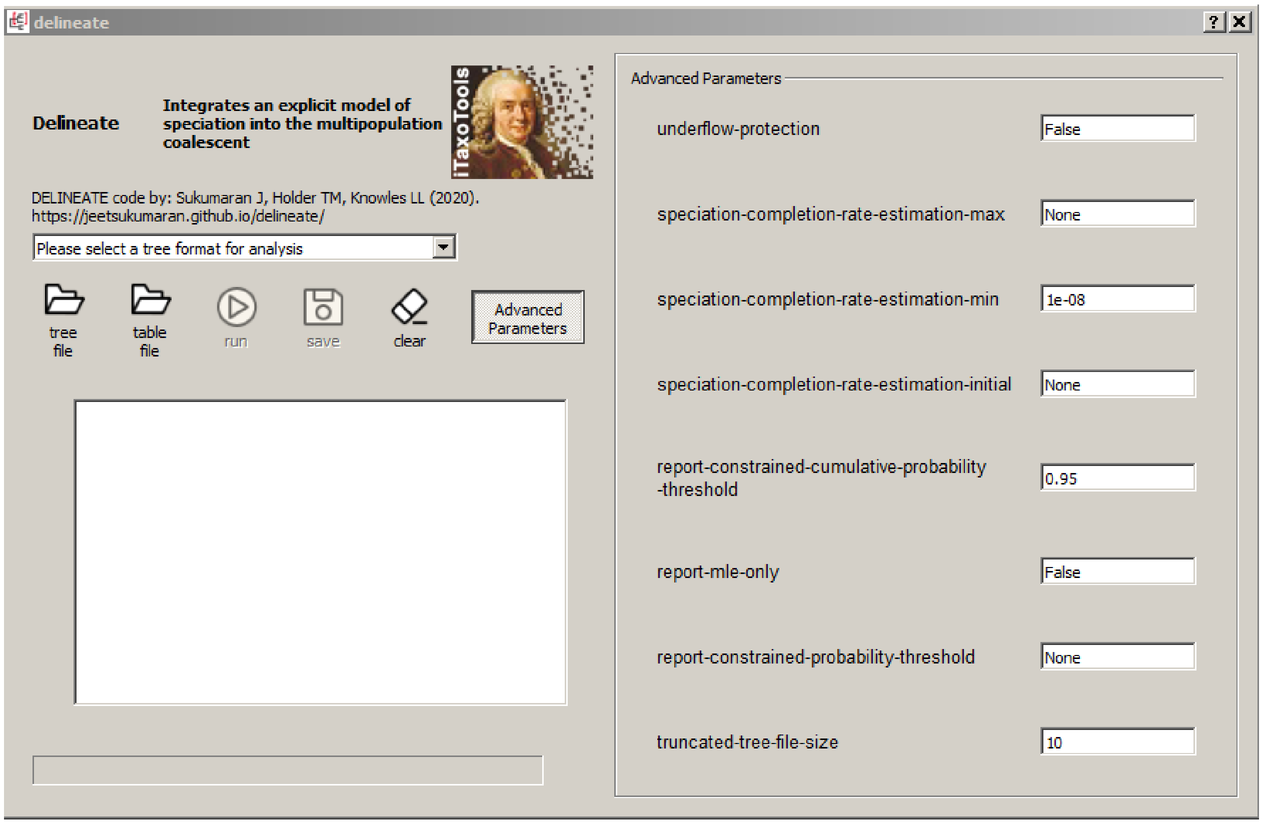
[Note that these two input files and the analysis provided by the DELINEATE tool only correspond to the last step of the workflow. To obtain these files, analyses with BP&P and StarBEAST are part of the overall workflow, described on the DELINEATE website. Also note that some further functions of the command-line driven DELINEATE Python tool such as the delineate-bppsum script are not available from the current GUI version.]

[Also note that these examples have been modified to limit the number of unconstrained individuals to 7 (there are 13 unconstrained samples in the original example files), by just randomly constraining some of them. With up to 7 unconstrained individuals (= samples) the program runs rather quickly on a normal PC, while it will take very long, potentially using up all available memory, with a higher number of unconstrained individuals.]

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.



Delineate Integrates an explicit model of speciation into the multipopulation coalescent

DELINEATE code by: Sukumaran J, Holder TM, Knowles LL (2020).
https://jeetsukumaran.github.io/delineate/

Please select a tree format for analysis

Advanced Parameters

underflow-protection	False
speciation-completion-rate-estimation-max	None
speciation-completion-rate-estimation-min	1e-08
speciation-completion-rate-estimation-initial	None
report-constrained-cumulative-probability-threshold	0.95
report-mle-only	False
report-constrained-probability-threshold	None
truncated-tree-file-size	10

6.4. GMYC

Example files provided:

GMYC_examplefile1_ultrametric_tree_iTaxoTools_0_1.tre
GMYC_examplefile2_ultrametric_tree_relaxed_clock_iTaxoTools_0_1.tre
GMYC_examplefile3_ultrametric_tree_iTaxoTools_0_1.tre

The first file is a very simple ultrametric tree distributed with the Python version of GMYC.
The second example file is an ultrametric tree from a relaxed clock analysis in BEAST, from the GMYC tutorial of F. Michonneau (see below).
The third example file is an ultrametric tree in Newick format showing relationships among different individuals and species of geckos of the genus *Lygodactylus*.

[note that the third tree file will not lead to a useful species delimitation, for unexplored reasons; it is provided as a cautionary example to show that care needs to be applied when calculating the ultrametric trees for input in GMYC]

Try the program by using these trees. The output files will be saved into the folder that you specify. The output tree will show clades of samples grouped into one species in red, and their connecting branches in blue. The program will also output a "spart" file that can be used as input for comparing partitions in LIMES.

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.

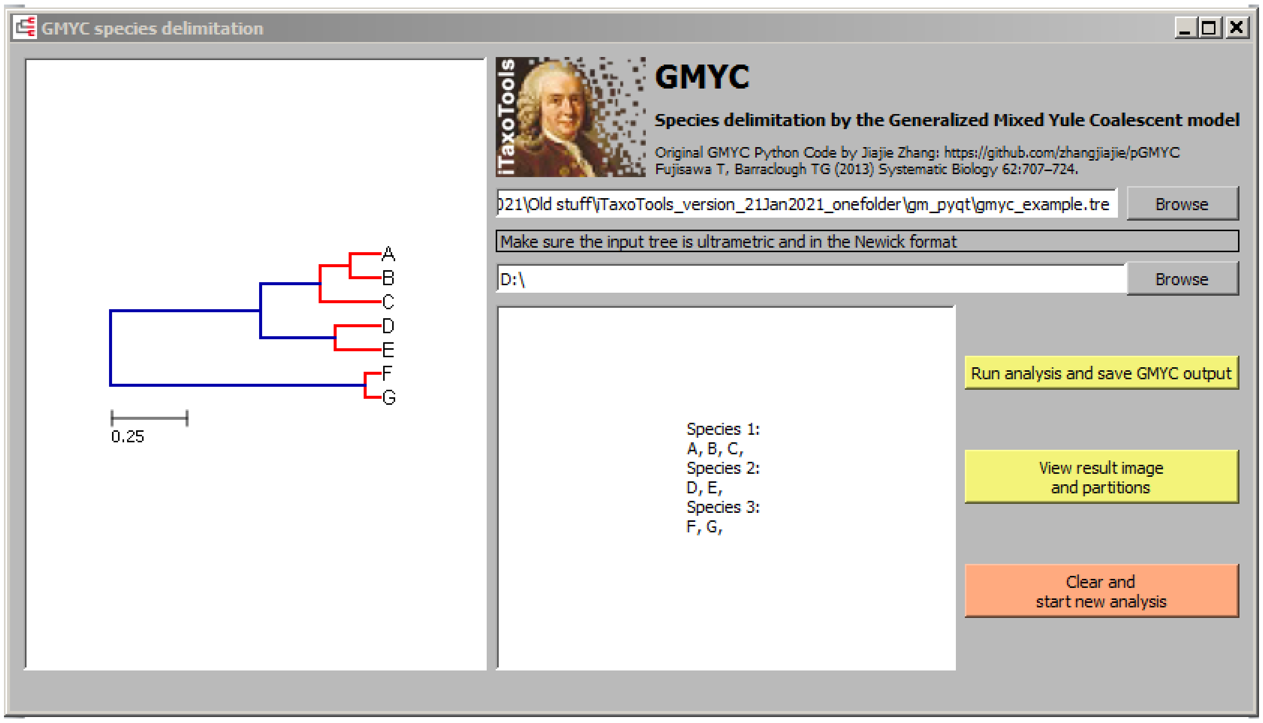
GMYC has been deployed on a webserver thanks to Jiajie Zhang at Heidelberg Institute for Theoretical Studies: https://species.h-its.org/gmyc/

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.


GMYC species delimitation
iTaxoTools
A
B
C
D
E
F
G
0.25
Browse
Browse
Run analysis and save GMYC output
View result image and partitions
Clear and start new analysis

GMYC

Species delimitation by the Generalized Mixed Yule Coalescent model

Original GMYC Python Code by Jiajie Zhang: https://github.com/zhangjiajie/pGMYC
Fujisawa T, Barraclough TG (2013) Systematic Biology 62:707–724.

)21\Old stuff\iTaxoTools_version_21Jan2021_onefolder\gm_pyqt\gmyc_example.tre

Make sure the input tree is ultrametric and in the Newick format

D:\

```
Species 1:
A, B, C,
Species 2:
D, E,
Species 3:
F, G,
```

6.5. PTP

Example file provided:

PTP_examplefile_tree_iTaxoTools_0_1.tre

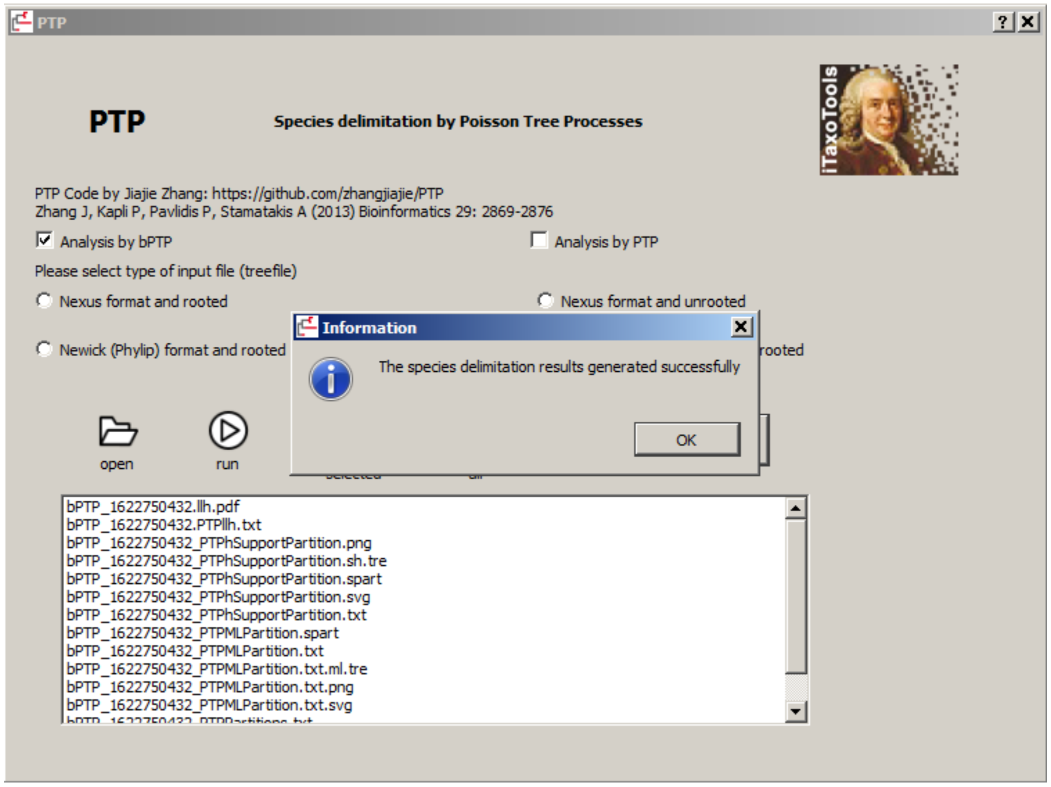
The example file is an tree in Newick format (with branch lengths, obtained by Maximum Likelihood analysis in MEGA) showing relationships among different individuals and species of geckos of the genus *Lygodactylus*.

Try the program by using this tree. The output files will be saved into the folder that you specify. The output tree will show clades of samples grouped into one species in red, and their connecting branches in blue. The program will also output a "spart" file that can be used as input for comparing partitions in LIMES.

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.


PTP

PTP Species delimitation by Poisson Tree Processes

PTP Code by Jiajie Zhang: https://github.com/zhangjiajie/PTP
Zhang J, Kapli P, Pavlidis P, Stamatakis A (2013) Bioinformatics 29: 2869-2876

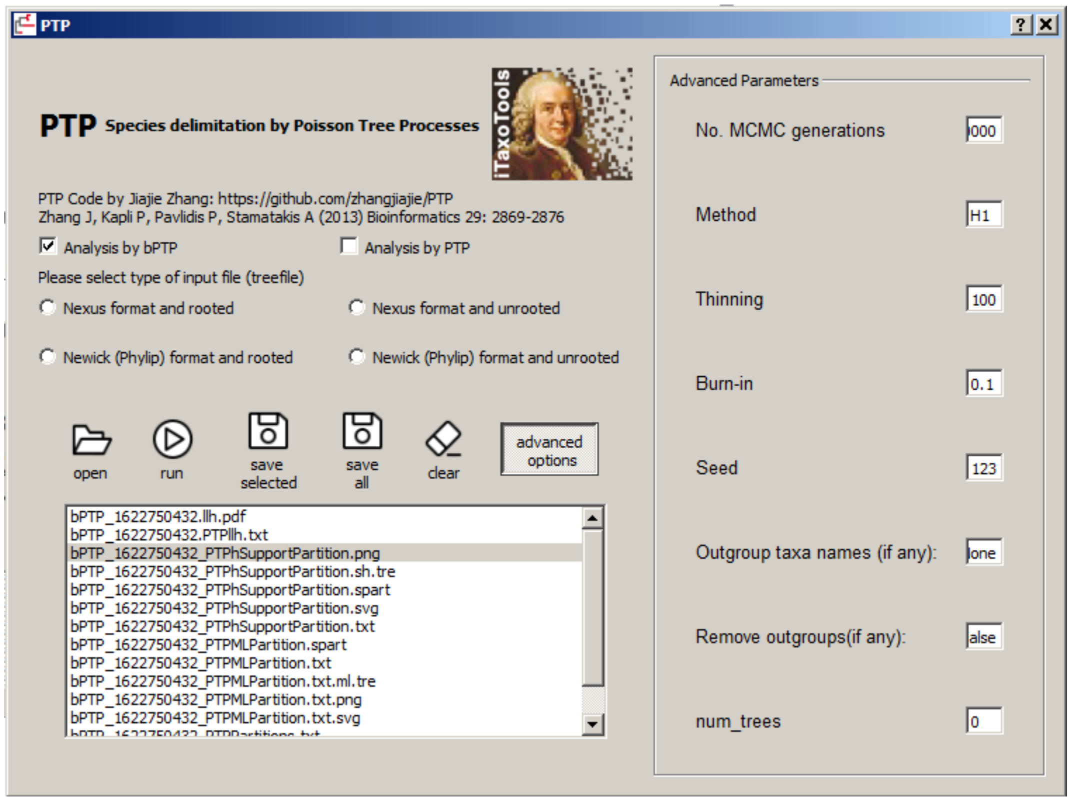
[✓ Analysis by bPTP] Analysis by PTP

Please select type of input file (treefile)

Nexus format and rooted / Nexus format and unrooted / Newick (Phylip) format and rooted / ...rooted

Information
The species delimitation results generated successfully
OK

```
bPTP_1622750432.llh.pdf
bPTP_1622750432.PTPllh.txt
bPTP_1622750432_PTPhSupportPartition.png
bPTP_1622750432_PTPhSupportPartition.sh.tre
bPTP_1622750432_PTPhSupportPartition.spart
bPTP_1622750432_PTPhSupportPartition.svg
bPTP_1622750432_PTPhSupportPartition.txt
bPTP_1622750432_PTPMLPartition.spart
bPTP_1622750432_PTPMLPartition.txt
bPTP_1622750432_PTPMLPartition.txt.ml.tre
bPTP_1622750432_PTPMLPartition.txt.png
bPTP_1622750432_PTPMLPartition.txt.svg
```


PTP

PTP Species delimitation by Poisson Tree Processes

PTP Code by Jiajie Zhang: https://github.com/zhangjiajie/PTP
Zhang J, Kapli P, Pavlidis P, Stamatakis A (2013) Bioinformatics 29: 2869-2876

[✓ Analysis by bPTP] Analysis by PTP

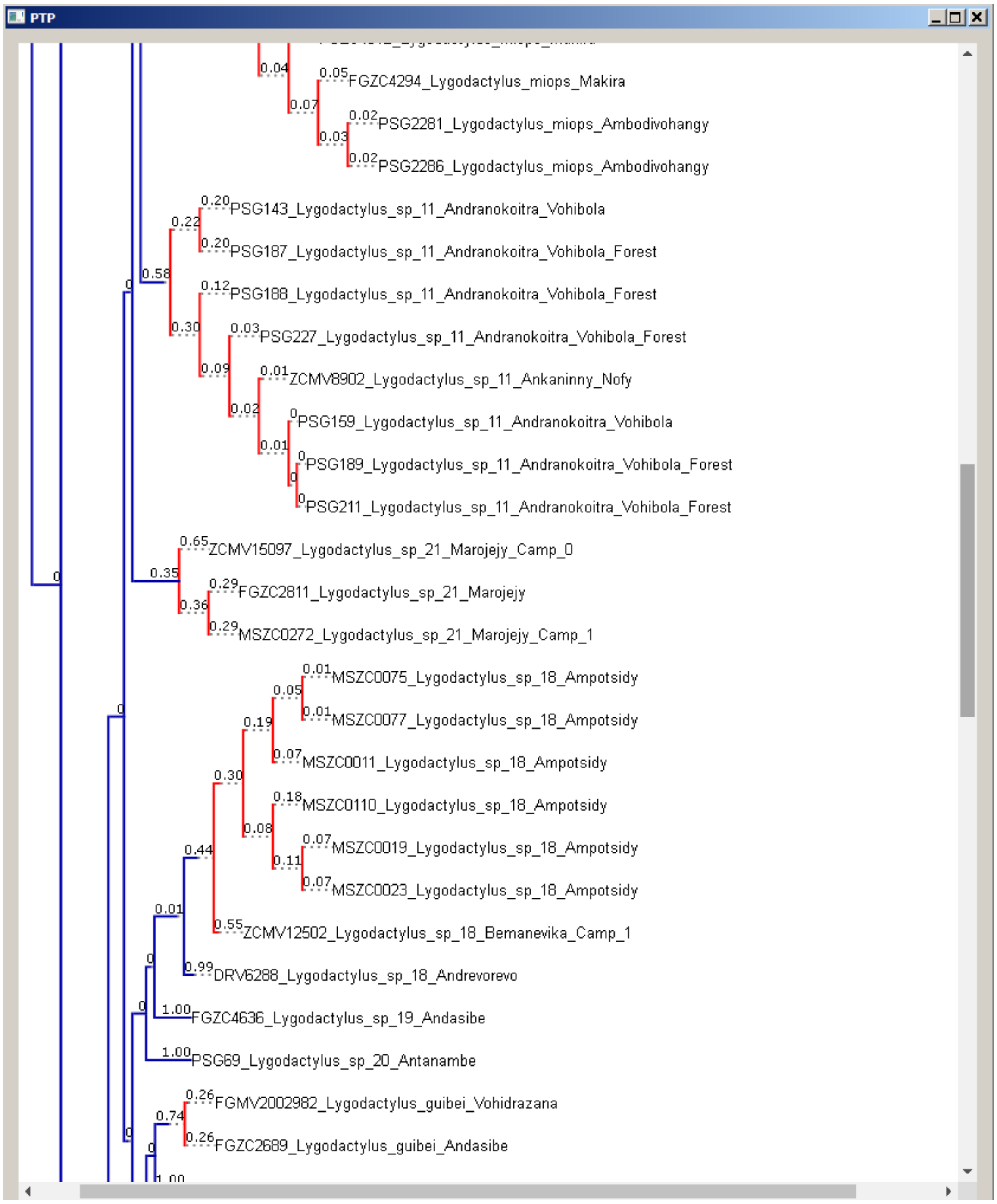
Please select type of input file (treefile)

Nexus format and rooted / Nexus format and unrooted / Newick (Phylip) format and rooted / Newick (Phylip) format and unrooted

```
bPTP_1622750432.llh.pdf
bPTP_1622750432.PTPllh.txt
bPTP_1622750432_PTPhSupportPartition.png
bPTP_1622750432_PTPhSupportPartition.sh.tre
bPTP_1622750432_PTPhSupportPartition.spart
bPTP_1622750432_PTPhSupportPartition.svg
bPTP_1622750432_PTPhSupportPartition.txt
bPTP_1622750432_PTPMLPartition.spart
bPTP_1622750432_PTPMLPartition.txt
bPTP_1622750432_PTPMLPartition.txt.ml.tre
bPTP_1622750432_PTPMLPartition.txt.png
bPTP_1622750432_PTPMLPartition.txt.svg
```

Advanced Parameters

No. MCMC generations	000
Method	H1
Thinning	100
Burn-in	0.1
Seed	123
Outgroup taxa names (if any):	one
Remove outgroups(if any):	alse
num_trees	0


PTP
0.04
0.05 FGZC4294_Lygodactylus_miops_Makira
0.07
0.02 PSG2281_Lygodactylus_miops_Ambodivohangy
0.03
0.02 PSG2286_Lygodactylus_miops_Ambodivohangy
0.20 PSG143_Lygodactylus_sp_11_Andranokoitra_Vohibola
0.22
0.20 PSG187_Lygodactylus_sp_11_Andranokoitra_Vohibola_Forest
0.58
0
0.12 PSG188_Lygodactylus_sp_11_Andranokoitra_Vohibola_Forest
0.30
0.03 PSG227_Lygodactylus_sp_11_Andranokoitra_Vohibola_Forest
0.09
0.01 ZCMV8902_Lygodactylus_sp_11_Ankaninny_Nofy
0.02
0 PSG159_Lygodactylus_sp_11_Andranokoitra_Vohibola
0.01
0 PSG189_Lygodactylus_sp_11_Andranokoitra_Vohibola_Forest
0
0 PSG211_Lygodactylus_sp_11_Andranokoitra_Vohibola_Forest
0.65 ZCMV15097_Lygodactylus_sp_21_Marojejy_Camp_0
0
0.35
0.29 FGZC2811_Lygodactylus_sp_21_Marojejy
0.36
0.29 MSZC0272_Lygodactylus_sp_21_Marojejy_Camp_1
0.01 MSZC0075_Lygodactylus_sp_18_Ampotsidy
0.05
0
0.01 MSZC0077_Lygodactylus_sp_18_Ampotsidy
0.19
0.07 MSZC0011_Lygodactylus_sp_18_Ampotsidy
0.30
0.18 MSZC0110_Lygodactylus_sp_18_Ampotsidy
0.08
0.44
0.07 MSZC0019_Lygodactylus_sp_18_Ampotsidy
0.11
0.07 MSZC0023_Lygodactylus_sp_18_Ampotsidy
0.01
0.55 ZCMV12502_Lygodactylus_sp_18_Bemanevika_Camp_1
0
0.99 DRV6288_Lygodactylus_sp_18_Andrevorevo
0
1.00 FGZC4636_Lygodactylus_sp_19_Andasibe
1.00 PSG69_Lygodactylus_sp_20_Antanambe
0.26 FGMV2002982_Lygodactylus_guibei_Vohidrazana
0.74
0
0.26 FGZC2689_Lygodactylus_guibei_Andasibe
0

6.6. tr2

Example files provided:

tr2_examplefile2_allgenetrees_iTaxoTools_0_1.tre
tr2_examplefile3_guidetree_iTaxoTools_0_1.tre
tr2_examplefile1_hypothesis_iTaxoTools_0_1.txt

The example files illustrate tr2 species delimitation using an example from *Sistrurus* rattlesnakes, originally used by T. Fujisawa to introduce the program. The files consist of a set of gene trees (in Newick format) for various *Sistrurus* individuals, a guide tree (in Newick format; to be used for the guide tree approach) and a hypothesis table (to be used for the hypothesis testing approach).
[note that the guide tree used here is a simple topology without branch length; most programs will provide a species tree (guide tree) with branch length information]

Try the program by using either the gene trees with the guide tree (for the guide tree approach), or the gene trees with the hypothesis file (for the hypothesis testing approach). The output files will be saved into the folder that you specify.

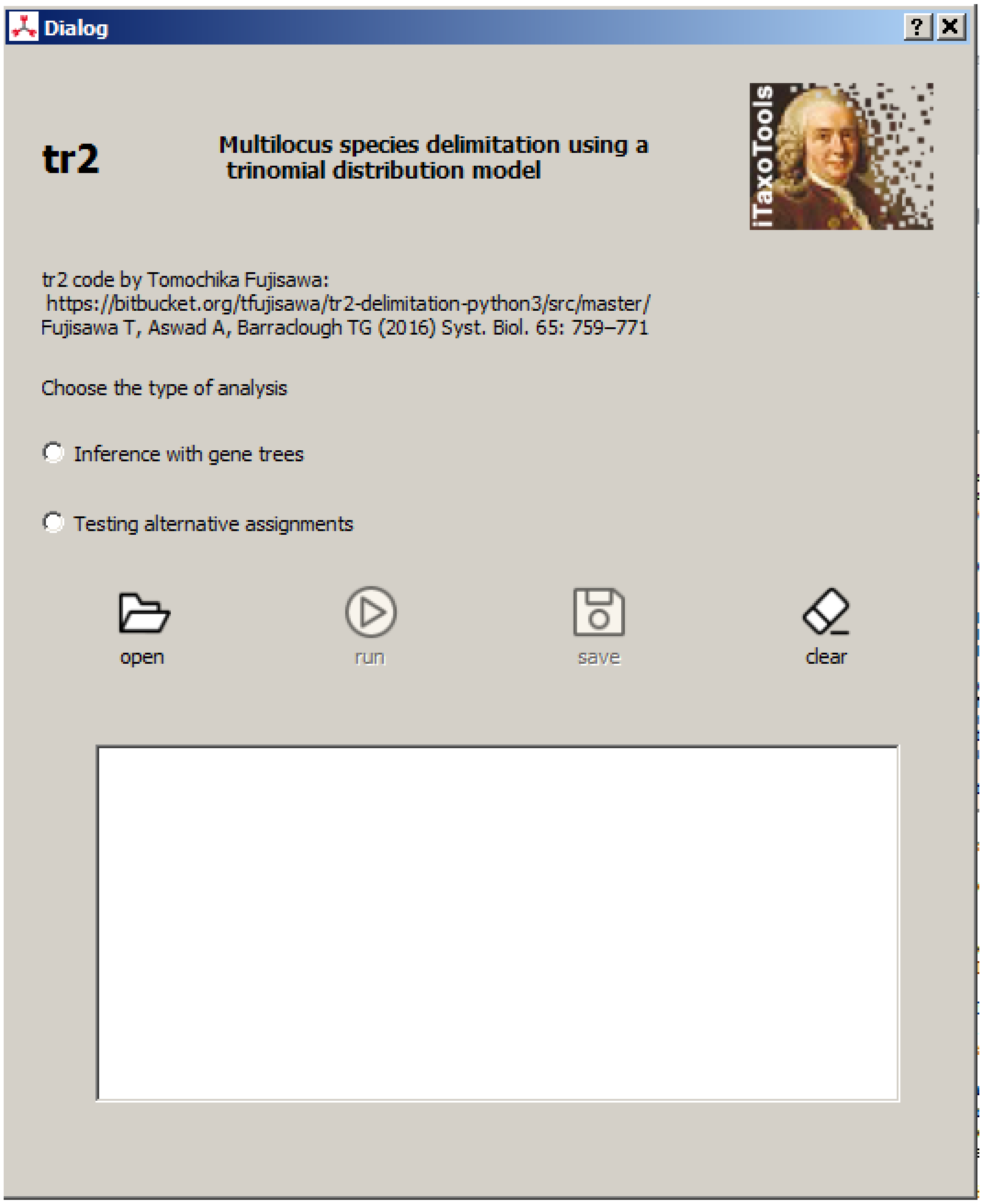
To open, inspect and edit the example files, use a simple text editor. The .tre files should best be visualized with a tree editing program; they will for instance open in MEGA.

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 the same example files, but without guide tree) 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.



Multilocus species delimitation using a trinomial distribution model

tr2 code by Tomochika Fujisawa:
https://bitbucket.org/tfujisawa/tr2-delimitation-python3/src/master/
Fujisawa T, Aswad A, Barraclough TG (2016) Syst. Biol. 65: 759–771

Choose the type of analysis

Inference with gene trees

Testing alternative assignments

6.7. LIMES v2.0

Example files provided:

LIMES_examplefile1_mtDNA-withABGD_iTaxoTools_0_1.spart
LIMES_examplefile2_mtDNA-withGMYC_iTaxoTools_0_1.spart
LIMES_examplefile3_PCA-Morpho_iTaxoTools_0_1.csv

These three fictituous example files present different formats that can all be read by Limes and merged together into a single multiple partition spart file (as exemplified in the screenshots below).

LIMES_examplefile4_Mantella_PTPhsupport_iTaxoTools_0_1.spart
LIMES_examplefile5_Mantella_PTPML_iTaxoTools_0_1.spart
LIMES_examplefile6_Mantella_ABGD_iTaxoTools_0_1.spart

These are additional example files in the matricial SPART format, obtained from analysis of a dataset of cytochrome b sequences of Mantella frogs, with ASAP and PTP (for PTP, first a maximum likelihood tree was built in MEGA). The SPART show different ways the Mantella individuals can be distributed in species partitions. Note that example file 1 shows a partition that obviously is based on a flawed analysis, with every individual assigned to a separate partition.

You can visualize and edit the SPART files in any text editor.

Description: This new version presents two distinct functionalities :

(1) **Species partitions comparison:** Ducasse et al. (2020) introduced LIMES v1, a program not to delimit species but to compare species partitions (inferred by other tools) using various statistical indexes. This functionality, also present in LIMES v.2.0, 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). The original distribution of LIMES is explained and introduced on its website, https://limes.prod.lamp.cnrs.fr/

(2) **Species partitions handling (new functionality):** In the context of the development of the species partition format SPART (Miralles et al. 2021), the present new version of LIMES (v2.0) has been programmed and integrated in iTaxoTools, in order to allow users to easily handle species partitions, i.e. to merge or to extract a selected sets of species partitions.

Repositories: Besides the iTaxoTools Github link for LIMES v2.0, both versions of Limes (v1 and v2.0) are also available (including the pure-python package) at https://www.limes.cnrs.fr/

Formats accepted: The current version, LIMES only accepts the matricial species partition (SPART) format, already implemented by several species delimitation tools (not yet the XML-SPART format). If needed, species partition files can also be manually edited with EXCEL (.csv, .xls and .xlsx format).

Moreover, LIMES also accepts species partitions results copy-pasted from the original websites of ABGD, GMYC et PTP (see Ducasse et al 2020). The example below present two equivalent versions – both accepted by LIMES 2.0 – of a same species partition dataset with three specimens (001 to 003) and two distinct species partitions (infered from methods A and B). When possible, we recommend to preferably use SPART format.

SPART format (.spart)	Manually written formats (.csv, .xls)

begin spart;
Project_name = myexemple;
Date = 2021-05-11T10:40:08;
N_spartitions = 2: methodA / methodB;
N_individuals = 3 / 3;
N_subsets = 3 / 2;
Individual_assignment =
Sp001: 1 / 1
Sp002: 2 / 1
Sp003: 3 / 2;
end;
	specimen,methodA,methodB
	sp001,1,1
	sp002,2,1
	sp003,3,2
	
	Note: different type of separators (comma, semi-colon, tab, space, can also be used)

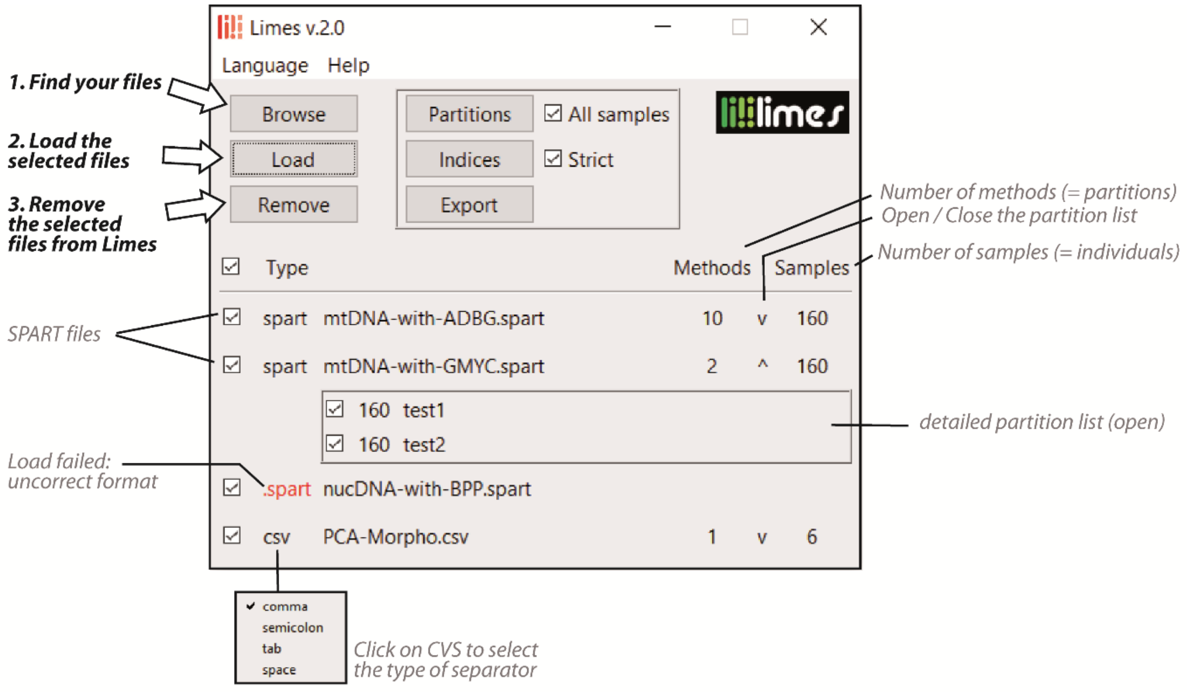
General comments about the stucture of the input data files: each line represents a specimen and each column a species delimitation method. Inside a given column, all the samples which are assigned to the same species share the same number. The actual value of the numbers does not matter (it has to be regarded as a species identifier within a given column); in the same way, there is no relation between the numbers of a given sample through the several methods. If a sample should not be considered for a method (missing data), it is noted "–" or "?".

Comment about the CSV format: The table can be provided in a CSV formated file (.csv), with the comma (or semicolon, tab or space) as delimiter.

Comments about the Excel formats (.xls or .xlsx) : The sheet including the table has to be indicated in LIMES v2.0 (click on ".xls" or ".xlsx"); by default, LIMES will read the table of the first sheet. The first non empty line in the sheet is the title line of the methods, each cell giving the name of a method. Then, the first non empty column, on the left of the first method column, is the title column of the samples, each cell giving the name of a sample. The columns more on the left are ignored. Alternatively, there is another method to locate the table in the sheet: If a cell anywhere in the sheet contains the word "LIMES", this cell defines the intersection between the title line of the methods and the title column of the samples. All the lines above and all the columns on the left are ignored. This method allows more flexibility to include other data in the sheet.
In all cases: (1) If the title cell of a column is empty, all the column is ignored; the same for the lines where the title cell is empty (this allows to let empty columns and lines, and even to insert comments around the cells containing the numbers). (2) All the method names and all the sample names must be distinct (redundant names or methods are not permitted).

Starting : The program first requires the user to browse ("Browse" [1]) for one or several species partition file (preferably SPART, but also cvs, xls and xlsx) and to select it (them). Next, the file(s) needs to be loaded ("Load" [2]). During the loading process, the program checks the file(s) for integrity, i.e., for a correct syntax as proposed by Miralles et al. (2021) for SPART files.

If correctly loaded, the program lists the files with the number of methods included, and the number of samples (individuals) present in at least one partition. The different partitions present in a given file can also be shown or masked (by default) by clicking on the arrows ("∧", "∨"); i.e. to present the name and the number of samples they contain. The following example has two SPART files successfully loaded, produced with different variants of PTP program implemented in iTaxoTools, plus one CSV file. If necessary, some files can be removed from the ongoing analysis [3].

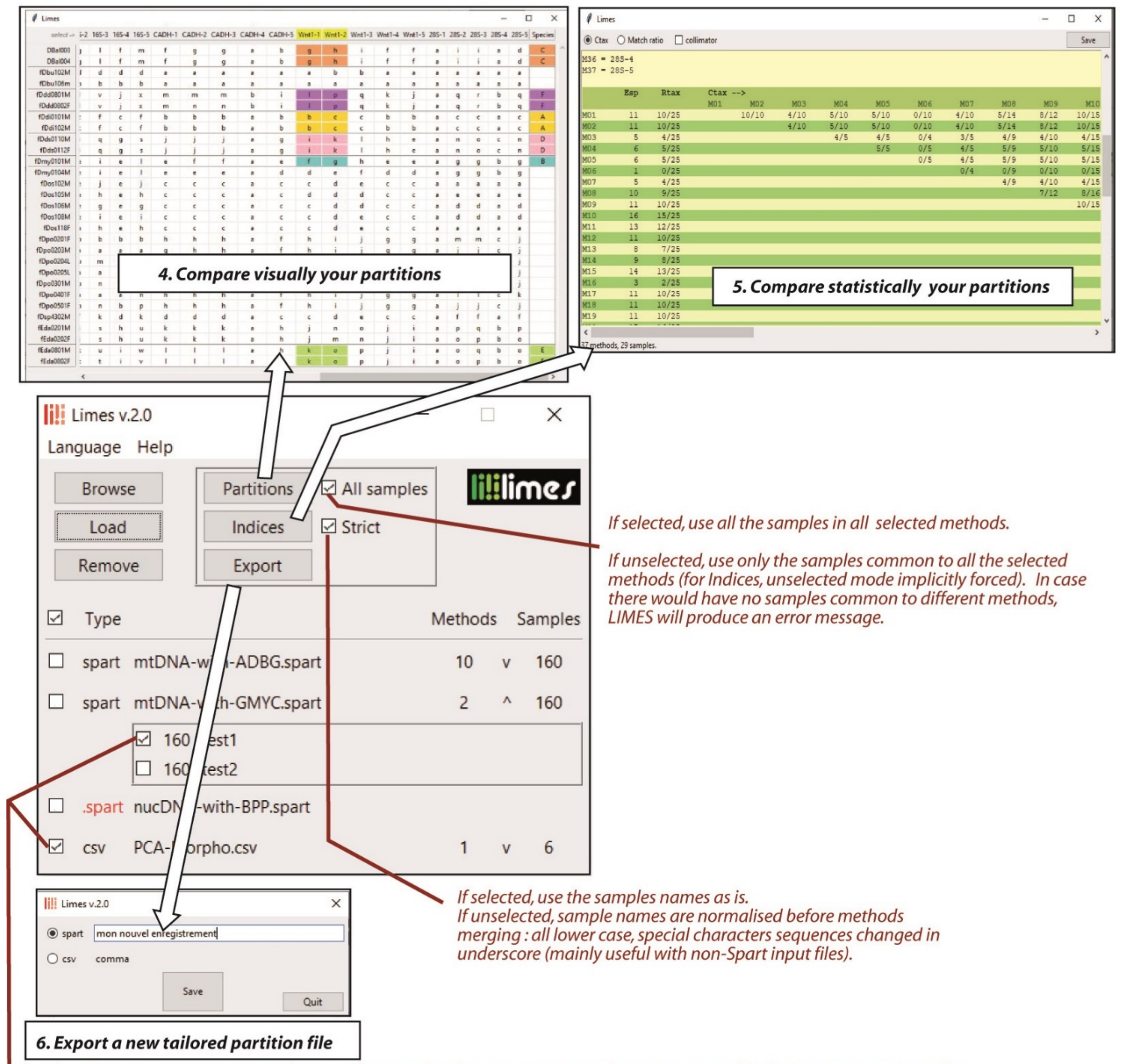

Limes v.2.0
Language Help
1. Find your files
2. Load the selected files
3. Remove the selected files from Limes
Browse
Load
Remove
Partitions
Indices
Export
All samples
Strict
limes
Number of methods (= partitions)
Open / Close the partition list
Number of samples (= individuals)
SPART files
detailed partition list (open)
Load failed: uncorrect format
comma
semicolon
tab
space
Click on CVS to select the type of separator

	Type		Methods		Samples
☑	spart	mtDNA-with-ADBG.spart	10	v	160
☑	spart	mtDNA-with-GMYC.spart	2	^	160
		☑ 160 test1			
		☑ 160 test2			
☑	.spart	nucDNA-with-BPP.spart			
☑	csv	PCA-Morpho.csv	1	v	6

Comparison of species partitions: LIMES then compares the loaded species partitions, providing a comparison of assignment for each individual ("Partitions"). Partitions can be compared visually [4] and LIMES can calculates various taxonomic indices ("Indices") among the loaded partitions [5] (note that partitions can be selected or deselected using the respective checkboxes, in order to include only a subset of partitions in these comparisons).

Important: This version of LIMES cannot yet manage missing data to calculate indices (i.e. index calculations exclude samples if they have at least one missing data (unassigned individual) in a partition involved in a given calculation)

Handling of species partitions: From this list of files (and partitions they contains), users can recompose a new tailored species partition file (i.e. merging or excluding a selection of partitions) by clicking on those of interest, and then export the corresponding SPART file or CSV file ("Export" [6]) for other applications.



4. Compare visually your partitions

select ->	i-2	16S-3	16S-4	16S-5	CADH-1	CADH-2	CADH-3	CADH-4	CADH-5	Wnt1-1	Wnt1-2	Wnt1-3	Wnt1-4	Wnt1-5	28S-1	28S-2	28S-3	28S-4	28S-5	Species
DBal003	[illegible]	l	f	m	f	g	g	a	b	g	h	i	f	f	a	i	i	a	d	C
DBal004	[illegible]	l	f	m	f	g	g	a	b	g	h	i	f	f	a	i	i	a	d	C
fDbu102M	[illegible]	d	d	d	a	a	a	a	a	a	b	b	a	a	a	a	a	a	a	
fDbu106m	[illegible]	b	b	b	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	
fDdd0801M	[illegible]	v	j	x	m	m	m	b	i	l	p	q	k	j	a	q	r	b	q	F
fDdd0802F	[illegible]	v	j	x	m	n	n	b	i	l	p	q	k	j	a	q	r	b	q	F
fDdi0101M	[illegible]	f	c	f	b	b	b	a	b	b	c	c	b	b	a	c	c	a	c	A
fDdi102M	[illegible]	f	c	f	b	b	b	a	b	b	c	c	b	b	a	c	c	a	c	A
fDds0110M	[illegible]	q	g	s	j	j	j	a	g	i	k	l	h	e	a	n	o	c	n	D
fDds0112F	[illegible]	q	g	s	j	j	j	a	g	i	k	l	h	e	a	n	o	c	n	D
fDmy0101M	[illegible]	i	e	l	e	f	f	a	e	f	g	h	e	e	a	g	g	b	g	B
fDmy0104M	[illegible]	i	e	l	e	e	e	a	d	d	e	f	d	d	a	g	g	b	g	
fDos102M	[illegible]	j	e	j	c	c	c	a	c	c	d	e	c	c	a	a	a	a	a	
fDos105M	[illegible]	h	e	h	c	c	c	a	c	d	d	d	c	c	a	e	e	a	e	
fDos106M	[illegible]	g	e	g	c	c	c	a	c	c	d	d	c	c	a	d	d	a	d	
fDos108M	[illegible]	i	e	i	c	c	c	a	c	c	d	e	c	c	a	d	d	a	d	
fDos118F	[illegible]	h	e	h	c	c	c	a	c	c	d	e	c	c	a	a	a	a	a	
fDpo0201F	[illegible]	b	b	b	h	h	h	a	f	h	i	j	g	g	a	m	m	c	j	
fDpo0203M	[illegible]	a	a	a	g	h	h	a	f	h	i	j	g	g	a	j	j	c	j	
fDpo0204L	[illegible]	m	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	j	
fDpo0205L	[illegible]	a	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	j	
fDpo0301M	[illegible]	n	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	j	
fDpo0401F	[illegible]	a	a	n	h	h	h	a	f	h	i	j	g	g	a	i	i	c	k	
fDpo0501F	[illegible]	n	b	p	h	h	h	a	f	h	i	j	g	g	a	j	j	c	j	
fDsp4302M	[illegible]	k	d	k	d	d	d	a	c	c	d	e	c	c	a	f	f	a	f	
fEda0201M	[illegible]	s	h	u	k	k	k	a	h	j	n	o	j	i	a	p	q	b	p	
fEda0202F	[illegible]	s	h	u	k	k	k	a	h	j	m	n	j	i	a	o	p	b	o	
fEda0801M	[illegible]	u	i	w	l	l	l	a	h	k	o	p	j	i	a	o	q	b	o	E
fEda0802F	[illegible]	t	i	v	l	l	l	a	h	k	o	p	j	i	a	o	p	b	o	[illegible]

5. Compare statistically your partitions

● Ctax ○ Match ratio ☐ collimator

```
M36 = 28S-4
M37 = 28S-5

        Esp     Rtax      Ctax -->
                          M01     M02     M03     M04     M05     M06     M07     M08     M09     M10
M01      11    10/25             10/10    4/10    5/10    5/10    0/10    4/10    5/14    8/12   10/15
M02      11    10/25                      4/10    5/10    5/10    0/10    4/10    5/14    8/12   10/15
M03       5     4/25                               4/5     4/5     0/4     3/5     4/9    4/10    4/15
M04       6     5/25                                       5/5     0/5     4/5     5/9    5/10    5/15
M05       6     5/25                                               0/5     4/5     5/9    5/10    5/15
M06       1     0/25                                                       0/4     0/9    0/10    0/15
M07       5     4/25                                                               4/9    4/10    4/15
M08      10     9/25                                                                      7/12    8/16
M09      11    10/25                                                                             10/15
M10      16    15/25
M11      13    12/25
M12      11    10/25
M13       8     7/25
M14       9     8/25
M15      14    13/25
M16       3     2/25
M17      11    10/25
M18      11    10/25
M19      11    10/25
```

37 methods, 29 samples.

☑ All samples

☑ Strict

☑	Type		Methods		Samples
☐	spart	mtDNA-w -ADBG.spart	10	v	160
☐	spart	mtDNA-v h-GMYC.spart	2	^	160
☑	160	est1			
☐	160	test2			
☐	.spart	nucDN -with-BPP.spart			
☑	csv	PCA- orpho.csv	1	v	6

● spart mon nouvel enregistrement

○ csv comma

If selected, use all the samples in all selected methods.

If unselected, use only the samples common to all the selected methods (for Indices, unselected mode implicitly forced). In case there would have no samples common to different methods, LIMES will produce an error message.

If selected, use the samples names as is.
If unselected, sample names are normalised before methods merging : all lower case, special characters sequences changed in underscore (mainly useful with non-Spart input files).

6. Export a new tailored partition file

Select the partitions you want to focuse on (to visualise them, to compute indexes, or to assemble them into a new SPART file)

Command line mode :

Synopsis :

```
% limes -I -m -n file...
% limes -O -n -c [-s sep] fmt [titre] file...
% limes -C file
% limes
% limes -hh
```

In the first form (-I), load all the files `file` and display the indices calculated on all the methods.

In the second form (-O), load all the files `file` and product the merged file in `fmt` format. `fmt` has one of the following values:

`Spart`	Spart format. In this case, the first argument *title* is the title which will be stored in the Project_name block.
`Csv`	CSV format.

In the third form (-C), just check the file `file`, and displays an error message if the file is invalid.

Without argument (fourth form), launches the graphical interface.

By default, if none of the -IOC options are supplied and arguments are present, the -C option is taken into account.

In all cases, the type of the file is identified by the extension:

`.spart`	Spart format
`.xls, .xlsx`	Excel format
`.csv`	CSV format
other	Limes tries to identify one of the single-partition format ABGD, GMYC or PTP.

In the case of CSV and Excel files, the extension can be followed by a complement after a colon « : »

For a CSV file, this is the separator. Example: « `myfile.csv:;` ». The default separator is the comma « `,` ».

For an Excel file, this is the name or number (from 1) of the sheet. Example: « `myfile.xls:sheet2` » or « `myfile.xls:2` ». By default, the first sheet is taken into account.

Options:

I	Calculate the indices (see text).
O	Merge the files and produce a file in the specified format (see text).
C	Simply perform the syntax check of the file (see text).
m	Calculates and displays match ratios rather than cTax (with -I only).
n	Standardizes the names of the samples before merging.

c	Only takes into account the samples common to all the methods (possibly after normalization if the -n option is supplied). Option forced implicitly with -I.
s	Specify the separator *sep*. Only if *fmt* is csv. Comma by default.
h	Display this help in French, or in English if duplicated

References.

Ahrens, D. *et al.* (2016) Rarity and Incomplete Sampling in DNA-Based Species Delimitation. *Syst. Biol.,* **65**, 478–494.

Ducasse J, Ung V, Lecointre G, **Miralles A** (2020). LIMES, a tool for comparing species partition. *Bioinformatics*, 2282-2283.

Miralles, A., Vences M. (2013) New Metrics for Comparison of Taxonomies Reveal Striking Discrepancies among Species Delimitation Methods in *Madascincus* Lizards. PlosONE, 8, e68242.

Miralles A, Ducasse J, Brouillet S, Flouri T, Fujisawa T, Kapli P, Knowles LL, Kumari S, Stamatakis A, Sukumaran J, Lutteropp S, Vences M (2021). SPART, a versatile and standardized data exchange format for species partition information *BioRxiv* (preprint).

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

6.8. spartmapper

Example files provided:

Spartmapper_examplefile1_iTaxoTools_0_1.tab
Spartmapper_examplefile2_iTaxoTools_0_1.spart

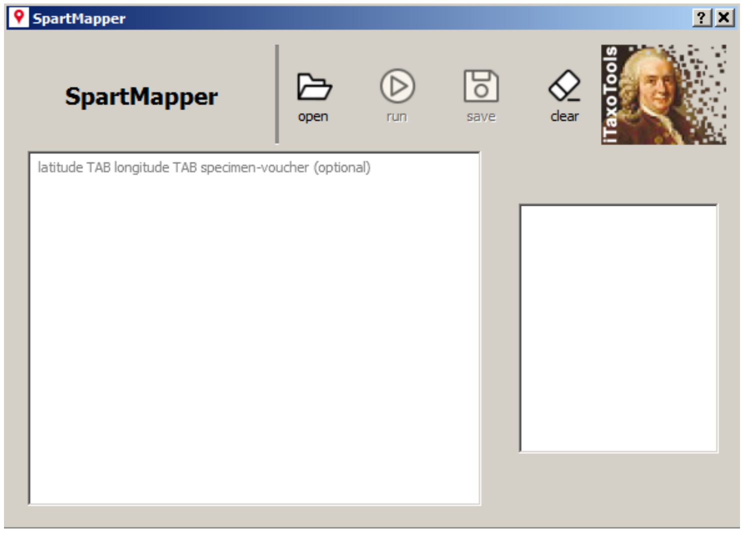
The example files represent (1) a tab-delimited file with geographical coordinates for a series of specimens, and (2) a spart file providing species partition information for these specimens. The tab file is compulsory as input, the spart file provides additional information but is not required for running program.

You can open these exemple files in any text editor. The tab file is best edited in a spreadsheet editor such as Excel, whereas the spart file can be opened and edited in LIMES.

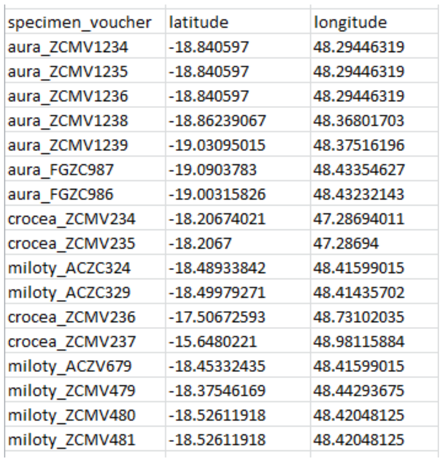
[Note that alternatively, for a quick visualization without species partition, coordinates can be added directly into the text box, in which case they should have the format: latitude TAB longitude TAB specimen voucher.]

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).



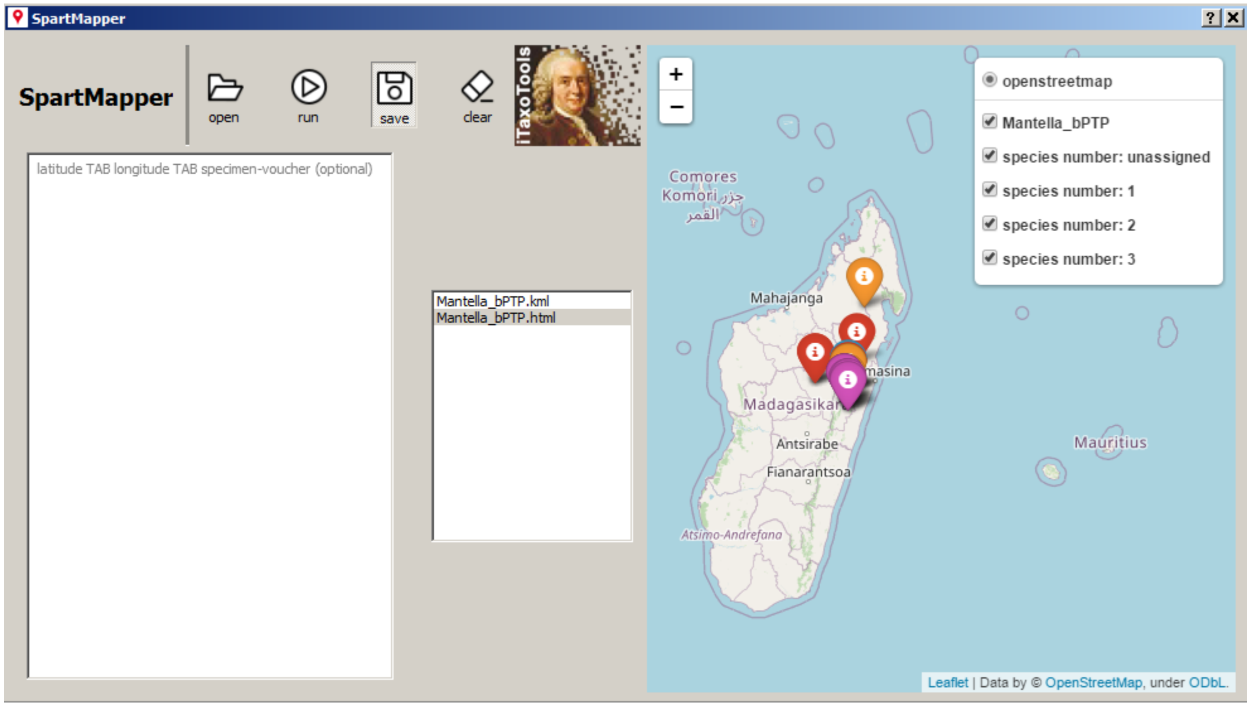
Geographical coordinates must be provided in a tab-delimited format, along with specimen identifier information, in the form of a field "specimen-voucher" (case insensitive; also accepted as "specimen_voucher"), as in the following example file.



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 ile 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".



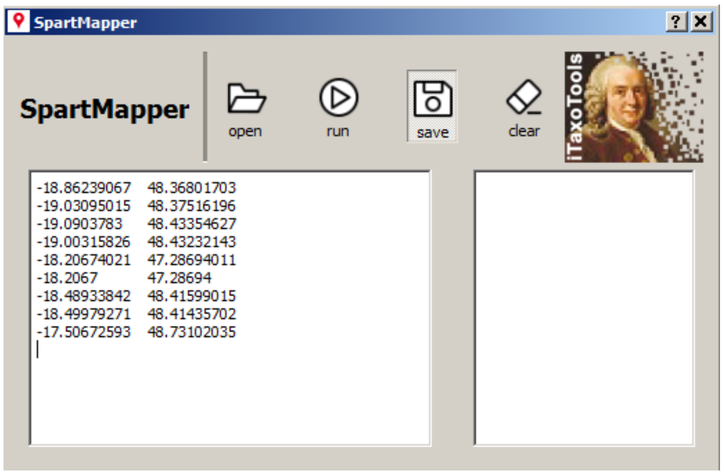
latitude TAB longitude TAB specimen-voucher (optional)

Mantella_bPTP.kml
Mantella_bPTP.html

● openstreetmap
✓ Mantella_bPTP
✓ species number: unassigned
✓ species number: 1
✓ species number: 2
✓ species number: 3

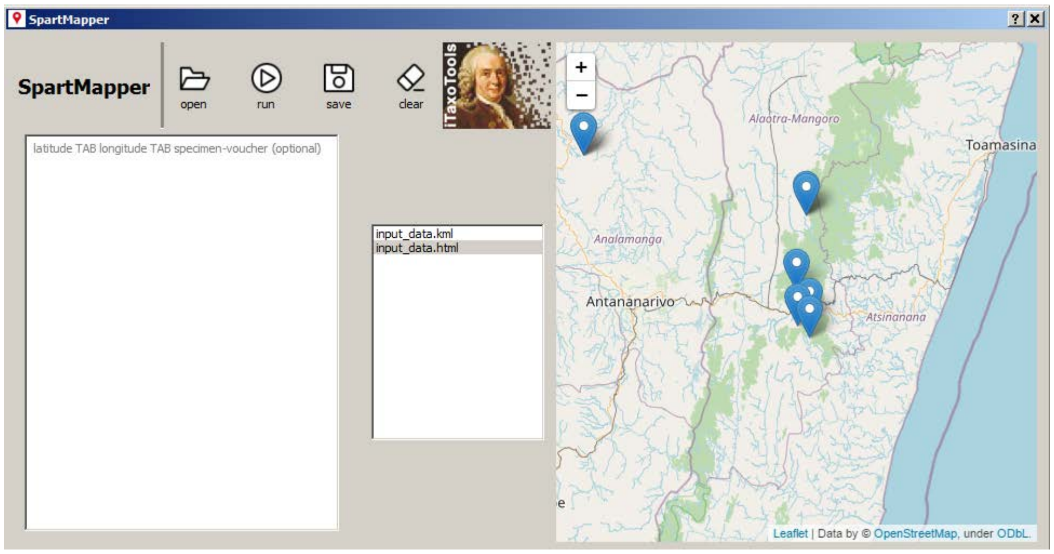
Leaflet | Data by © OpenStreetMap, under ODbL.

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.



```
-18.86239067	48.36801703
-19.03095015	48.37516196
-19.0903783	48.43354627
-19.00315826	48.43232143
-18.20674021	47.28694011
-18.2067	47.28694
-18.48933842	48.41599015
-18.49979271	48.41435702
-17.50672593	48.73102035
```

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

Example files provided:

dnadiagnoser_examplefile1_TyphleotrisCOI_iTaxoTools_0_1.tab
dnadiagnoser_examplefile2_MantellaScaphiophryneCOI_iTaxoTools_0_1.tab
dnadiagnoser_examplefile3_ScaphiophryneCOI_artificialgaps_iTaxoTools_0_1.tab
dnadiagnoser_examplefile4_TyphleotrisCOIaligned_iTaxoTools_0_1.tab

The four example files all are tab-delimited text files with DNA sequences of the mitochondrial COI gene for different genera of amphibians and fishes.

Try the program by using these files as input. In all cases it is recommended to use *Homo sapiens* or *Danio rerio* COI as reference sequence. The output files can be previewed and saved.

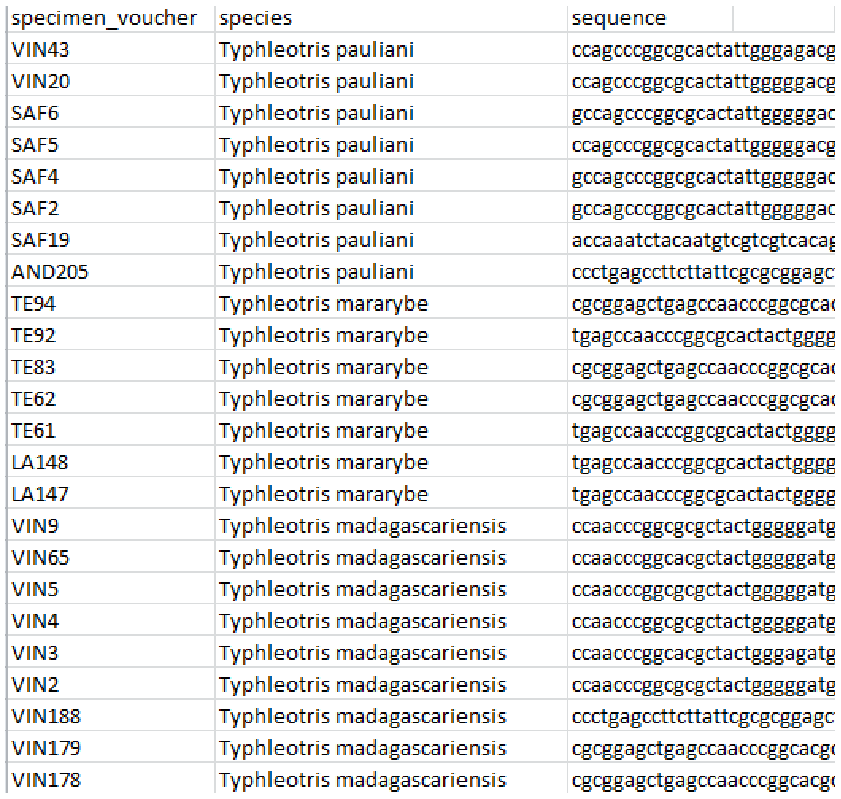
The first example file is very simple and will give you diagnostic differences between three species of cave fish (*Typhleotris*).

The second file is from two genera of frogs (*Mantella* and *Scaphiophryne*) also includes additional fields (columns) for genus and family (other_category) and allows you to specify to output diagnostic differences among these units.

The third file includes sequences of different length and some with a stretch of (articial) gaps, allowing you to test the specification of gaps as diagnostic sites.

The fourth file include already aligned sequences (to be run in the "already aligned" mode of the program), with the *Danio rerio* COI sequence being the first in the list (it will be used as reference for assigning diagnostic sites).

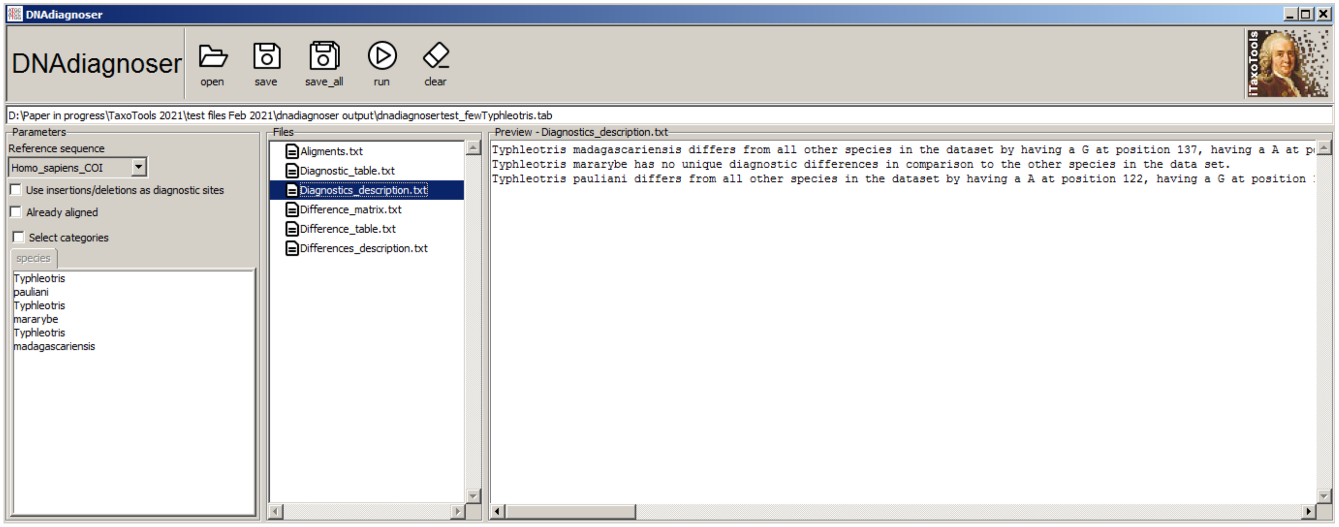
In order to facilitate the use of such DNA characters in differential diagnoses of new species, we implemented a crucial new tool for DNA taxonomy named dnadiagnoser. This tool takes as input tab-delimited text files in which one column specifies the unit for analysis (typically the species). The files 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	ccagcccggcgcactattgggagacg	
VIN20	Typhleotris pauliani	ccagcccggcgcactattgggggacg	
SAF6	Typhleotris pauliani	gccagcccggcgcactattgggggac	
SAF5	Typhleotris pauliani	ccagcccggcgcactattgggggacg	
SAF4	Typhleotris pauliani	gccagcccggcgcactattgggggac	
SAF2	Typhleotris pauliani	gccagcccggcgcactattgggggac	
SAF19	Typhleotris pauliani	accaaatctacaatgtcgtcgtcacag	
AND205	Typhleotris pauliani	ccctgagccttcttattcgcgcggagc	
TE94	Typhleotris mararybe	cgcggagctgagccaacccggcgcac	
TE92	Typhleotris mararybe	tgagccaacccggcgcactactgggg	
TE83	Typhleotris mararybe	cgcggagctgagccaacccggcgcac	
TE62	Typhleotris mararybe	cgcggagctgagccaacccggcgcac	
TE61	Typhleotris mararybe	tgagccaacccggcgcactactgggg	
LA148	Typhleotris mararybe	tgagccaacccggcgcactactgggg	
LA147	Typhleotris mararybe	tgagccaacccggcgcactactgggg	
VIN9	Typhleotris madagascariensis	ccaacccggcgcgctactgggggatg	
VIN65	Typhleotris madagascariensis	ccaacccggcacgctactgggggatg	
VIN5	Typhleotris madagascariensis	ccaacccggcgcgctactgggggatg	
VIN4	Typhleotris madagascariensis	ccaacccggcgcgctactgggggatg	
VIN3	Typhleotris madagascariensis	ccaacccggcacgctactgggagatg	
VIN2	Typhleotris madagascariensis	ccaacccggcgcgctactgggggatg	
VIN188	Typhleotris madagascariensis	ccctgagccttcttattcgcgcggagc	
VIN179	Typhleotris madagascariensis	cgcggagctgagccaacccggcacgc	
VIN178	Typhleotris madagascariensis	cgcggagctgagccaacccggcacgc	

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.



D:\Paper in progress\TaxoTools 2021\test files Feb 2021\dnadiagnoser output\dnadiagnosertest_fewTyphleotris.tab

Typhleotris
pauliani
Typhleotris
mararybe
Typhleotris
madagascariensis

```
Typhleotris madagascariensis differs from all other species in the dataset by having a G at position 137, having a A at p
Typhleotris mararybe has no unique diagnostic differences in comparison to the other species in the data set.
Typhleotris pauliani differs from all other species in the dataset by having a A at position 122, having a G at position
```

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

Example files provided:

MolD_examplefile1_PontohedyleCOI_iTaxoTools_0_1.fas
MolD_examplefile2_LophiotomaNICOI_iTaxoTools_0_1.fas
MolD_examplefile3_LophiotomaCOI_iTaxoTools_0_1.fas
MolD_examplefile4_ConusCOI_iTaxoTools_0_1.fas

These four example files all are fasta files with DNA sequences of the mitochondrial COI gene for different genera of snails.

Try the program by using these files as input. The output files will be saved into the folder that you specify. To get an idea of the performance and output of the program, try with the following settings:

Example file 1 (*Pontohedyle*):
Query taxon: brasilensis / settings: len(draftDNC)=7, len(mDNC)=4
Query taxa: brasilensis, verrucosa / settings: len(draftDNC)=7, len(mDNC)=4
Query taxa: ALL / settings: len(draftDNC)=7, len(mDNC)=4
Query taxon:brasilensis / settings: len(draftDNC)=7, len(mDNC)=4, cutoff >1

Example file 1 (*Conus*):
Query taxa: Conus_ebraeus / settings: default_parameters
Query taxa: Conus_ebraeus, Conus_chaldaeus / settings: len(draftDNC)=7, len(mDNC)=4
Query taxa: Conus_ebraeus, Conus_chaldaeus / settings: len(draftDNC)=7, len(mDNC)=4

Note 1: Before running the program, choose the taxon rank at which you want to obtain results (1 = species, 2 = above species) and whether gaps should be counted as diagnostic sites (yes or no).

Note 2: When inputting more than one query taxon in the respective box in the GUI, make sure they are added each on a new line, for instance, to enter "brasilensis, verrucosa":
"verrucosa
brasilensis"

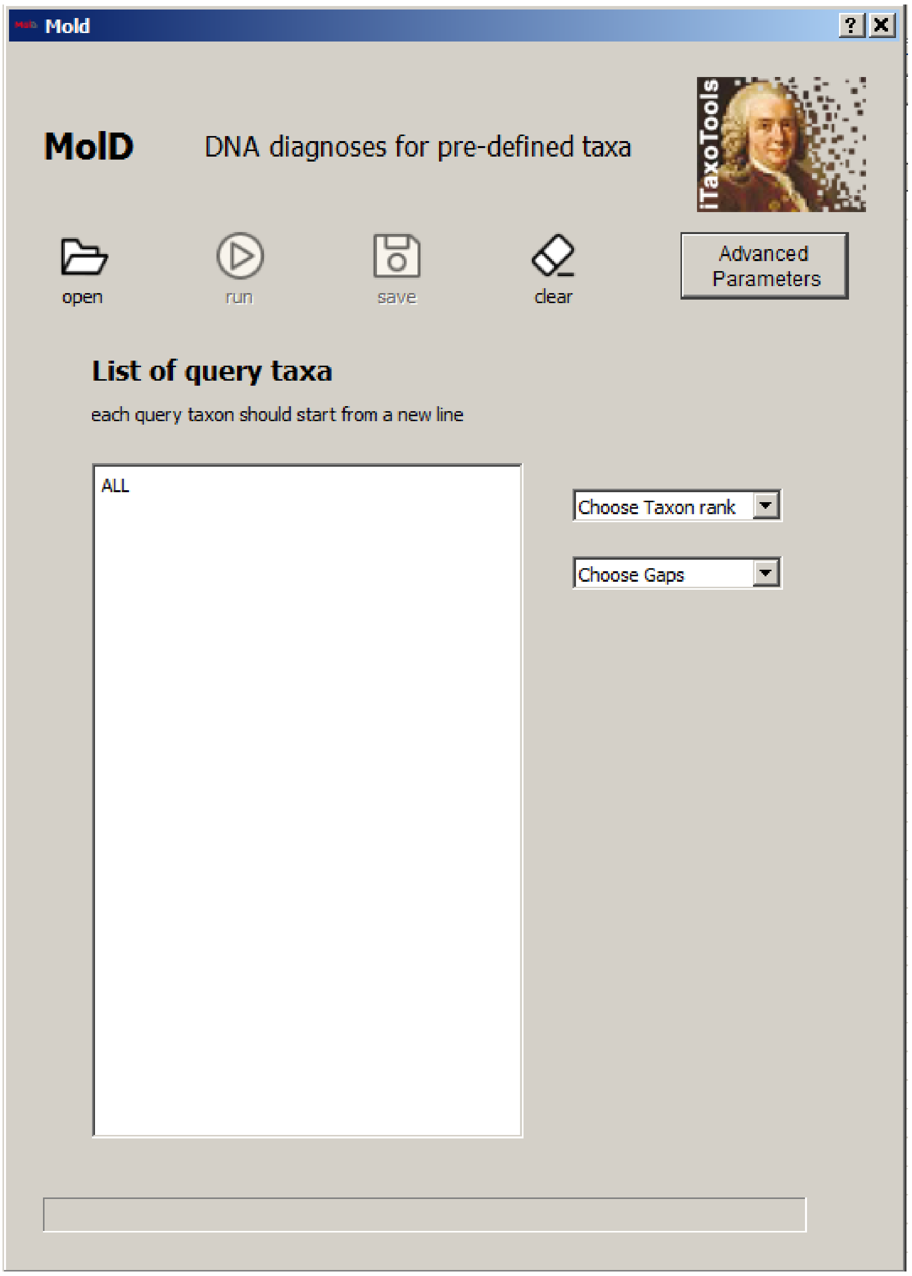
As a second program for species diagnosis, iTaxoTools implements a GUI version of MolD (Fedosov et al. 2020). This program is tailored for recovering DNA-based diagnoses in large DNA dataset, and is capable of identifying diagnostic combinations of nucleotides (DNCs) in addition to single (pure) diagnostic sites. The crucial and unique functionality of MolD 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 MolD 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 MolD 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.

Note: On Windows, the GUI version may not be able to save the output if executed from a logical drive different from the one where the TEMP folder is located (usually C:/).



The following instructions are verbatim copied from theoriginal 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) – Puilllandre et al. 2014]

```
###################################################
>Conasprella_alisi1|alisi
TATAAGATTTTGGCTTTTACCTCCTGCCCTTCTTTTACTCCTTTCTTCAGCT
>Conasprella_alisi2|alisi
TATAAGATTTTGGCTTTTACCTCCTGCCCTTCTTTTACTCCTTTCTTCAGCT
>Conasprella_alisi3|alisi
TATAAGATTTTGGCTTTTACCTCCTGCTCTTCTTTTACTCCTTTCTTCAGCT
>Conasprella_baileyi1|baileyi
TATAAGATTTTGACTTTTGCCTCCGGCCCTTCTTTTACTTCTTTCTTCAGCC
>Conasprella_baileyi2|baileyi
TATAAGATTTTGACTTTTGCCCCCGGCCCTTCTTTTACTTCTTTCTTCAGCC
>Conasprella_boholensis|boholensis
TATAAGATTTTGACTTTTACCTCCTGCGCTTCTTTTACTTCTTTCTTCAGCT
>Conasprella_boucheti|boucheti
TATAAGATTTTGACTTTTACCTCCCGCACTTCTTTTACTTCTTTCTTCAGCT
>Conasprella_comatosa|comatosa
TATAAGATTTTGACTTTTACCTCCTGCGTTGCTTCTACTCTTATCTTCAGCT
>Conasprella_coriolisi|coriolisi
TATAAGATTTTGACTTTTACCCCCTGCGTTGCTTCTACTCCTATCTTCAGCT
###################################################
```

!

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).

`qTAXA` (**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 themleave respective fields blank, and the defaults will be used.]

`Cutoff`

-integer (default **100**) -integer prepended by ('>')	denotes the **number of informative sites** to be considered for inclusion into a mDNC; 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`	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:

`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, x*L*/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)	**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. **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. If it doesn't help, **at last resort:** -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)	**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. **Too strict parameters for scoring rDNC/** Try relaxing (using lower values) each of the following parameters* in the following order : **NMaxSeq** – setting it at 5 is acceptable, below is not recommen. **Pdiff** – for species-level it should be either 1 or 2. **Scoring** – 'lousy' should only be used at last resort *note that by relaxing each or all parameters you compromise reliability of the resulting diagnosis.

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., Droege, 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., Mabee, P., Ford, L.S., Guralnick, R., Gropp, R.E., Revelez, M., Cobb, N., Seltmann, 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., Cibrian, 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.