

PyNCBIminer Manual

Contents

PyNCBIminer Manual	1
1. Installation and dependencies	1
1.1 Installation.....	1
1.2 Dependencies	1
2. Quick Start	1
2.1 If you want to perform analysis on your own dataset	1
2.2 Sequence Retrieving.....	2
2.3 Sequences Filtering	3
2.4 Sequences Alignment.....	4
2.5 Alignments Trimming	4
2.6 Alignments concatenation	5
3. Sequence Retrieving Module	5
3.1 Set target region and target taxa.	5
4. Supermatrix Construction Module.....	10
4.1 Sequences Filtering	10
4.2. Sequences Alignment.....	10
4.3 Alignments Trimming	10
4.4 Alignments concatenation	11
5. Python version and packages for running the source code.....	11

PyNCBIminer Manual

1. Installation and dependencies

1.1 Installation

For window: Users only need to unzip the package and double click the executable(.exe) file PyNCBIminer directly.

For MacOS: Users only need to unzip the package and double click the executable (.exe) file of PyNCBIminer directly.

Note that the executable file and the 'initial_queries' folder should always be in the same directory. Right click the executable file to make alias (create a shortcut) for convenience if necessary.

1.2 Dependencies

Network connection needed in several modules of PyNCBIminer. No extra dependencies are required.

PyNCBIminer is based on several tools and software, now including MAFFT (version 7) on <https://mafft.cbrc.jp/alignment/software/macosx.html> and trimAl (v1.2) on <http://trimal.cgenomics.org/downloads>.

2. Quick Start

This part gives a beginner's guide.

Please first download and unzip the package to finish the installation. You can right click the executable file to create a shortcut and place it wherever convenient for you. We highly recommend creating a new director for each step in case of unexpected overwriting.

2.1 If you want to perform analysis on your own dataset

If you need PyNCBIminer to perform some of the analysis, you may need to change the organization of folders and files, and the structure of definition line, to meet the requirements.

File Organization:

In the Supermatrix Construction Module (including Sequences Filtering, Sequences Alignment, Alignments Trimming and Alignments concatenation), the input can be a file or a director. If input

is a file, then PyNCBIminer will conduct an analysis of this input, of course. If input is a director, then PyNCBIminer will conduct an analysis of the readable files directly inside the director as they are one, and the file(s) inside the director(s) which are in the input director (input_director/another_director/file) will NOT be read and analyzed.

Structure of Definition Line:

In the Sequences Filtering sub-module of Supermatrix Construction Module, it is required that the structure of definition line (in FASTA format the line before the nucleotide sequence, called the FASTA definition line, must begin with a carat '>', followed by a unique SeqID) should be '>[accession number][taxon name][further description]', the three parts in the square brackets are separated by a vertical bar '|', for example: ">AB021052.1|Magnolia_schiedeana|Magnolia schiedeana chloroplast atpB, rbcL genes, partial cds, spacer region".

In the Alignments concatenation sub-module of Supermatrix Construction Module, it is recommended that the structure of definition line should be '>[taxon name]' in case of mistakenly added or concatenated sequences, for example ">Magnolia_schiedeana".

2.2 Sequence Retrieving

PyNCBIminer provides Sequence Retrieving module, using iterative BLAST searches to find high quality and complete sequences for a target genetic region and taxonomic group efficiently. Even if you have already collected your data, this module may help to complement the dataset. For a quick start, three parameters are required, and are listed as below:

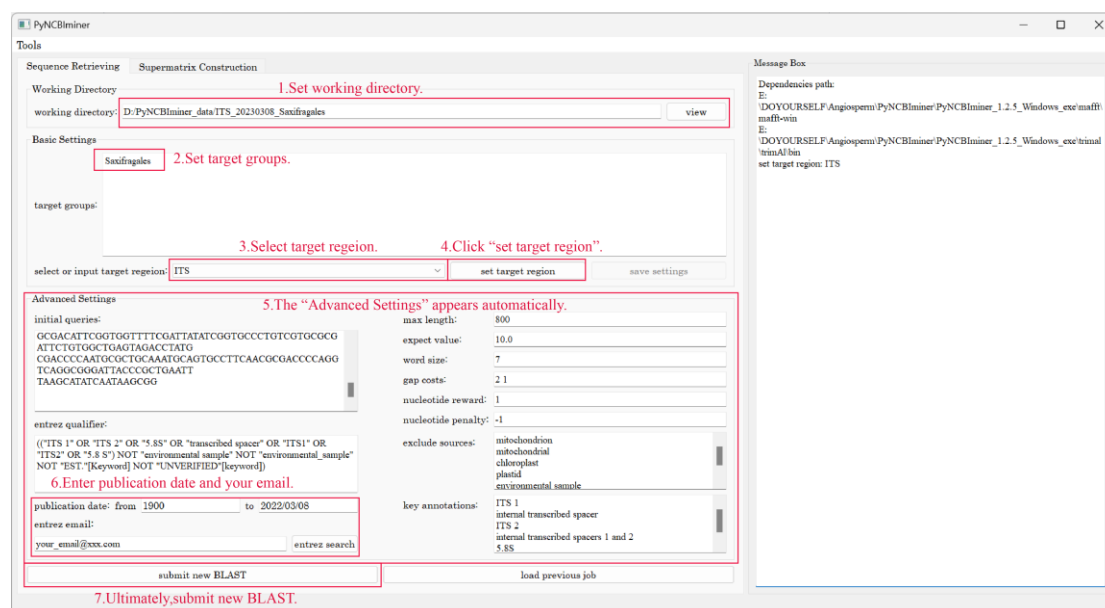
working directory - The destination folder of output.

target region - desired genetic marker (a gene or a spacer), such as *ITS*, *rbcL* or *matK*.

target groups - target taxonomic groups, such as Asterales or Magnolia, one group per line.

1. Set working directory by clicking 'view' button and select or pasting from clipboard.
2. Type the target group in the 'target group' text filed, one group per line.
3. Select target region from the pull-down list. If your desired region is not in the pull-down list, please read "exception" after step 8.
4. Click the 'set target region' button.
5. Blanks in the Advanced Settings will be filled, while you don't have to care about them in the 'quick start'.
6. Enter publication date, It can restrict the search to sequences published within a specific timeframe (default: 1900-now).
7. Enter your email. We recommend users to provide their email address, failure to do so may result in access being blocked by NCBI.
8. Then click the 'submit new BLAST' button, default queries and BLAST parameters will be sent to NCBI web server in batch submissions and wait till the completion. Once the sequence retrieving

is finished , you will see message in the message box on the right of the panel.

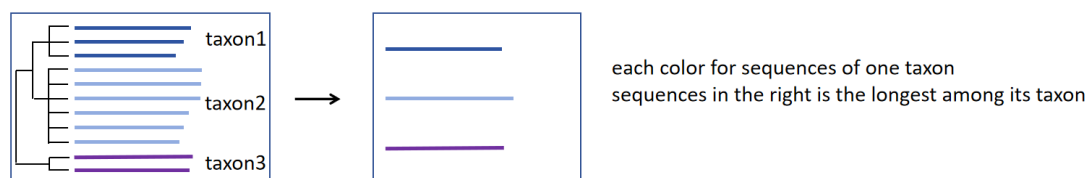


Exception: If the desired region is not in the pull-down list: follow this step for a quick start, or refer to III.Sequence Retrieving Module for a better settings of parameters.

1. If rate of the desired marker mutation is really fast, you can select 'ITS' in the pull-down list. If rate of mutation is relatively moderate or slow, you can select 'rbcL' in the pull-down list.
2. Click 'set target region' button.
3. Modify initial_queries, entrez_qualifier, max_length, exclude_sources, key_annotations in the Advanced Settings according to your desired sequence. Refer to III.Sequence Retrieving Module for detailed information.

Results are written in [your working directory]/results/blast_results_checked.fasta. Besides, pyNCBIminer performs a simple optimization on the ends of sequences in blast results, and the optimized results are written in [your working directory]/results/blast_results_controlled.fasta. You can select a better fasta file based on your judgement or the latter (controlled.fasta) on our advice.

2.3 Sequences Filtering



PyNCBIminer can perform sequence filtering simply according to length of the sequences. Only the longest sequence of each taxon will be kept after filtering. For a quick start (the same is for a normal run, though), only input and output path are required, and are listed as below:

input path - The input file(s) of sequences to be filtered in fasta format.

output path - The destination folder of output, where log files and result will be written.

1. Set the input path and output path by clicking 'view' button or pasting from clipboard.
2. Then click the 'run' button.

Assume that you haven't changed the original file name (blast_results_checked.fasta). Results are written in [your output path]/blast_results_checked.fasta. More information of kept results are in [your output path]/blast_results_checked_kept_records.csv

2.4 Sequences Alignment



PyNCBIminer can perform sequence alignment using MAFFT. For a quick start, only input and output path are required, and are listed as below:

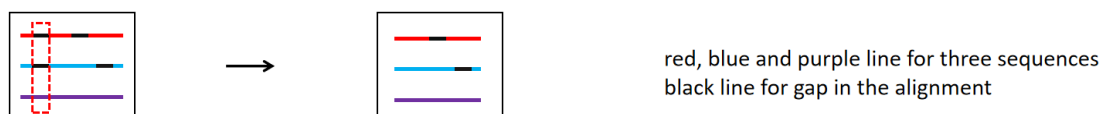
input path - The input file(s) of sequences to be aligned in fasta format.

output path - The destination folder of output, where log files and result will be written.

1. Set the input path and output path by clicking 'view' button or pasting from clipboard.
2. Then click the 'run' button (simply leaving other parameters default or blank is okay).

Assume that you haven't changed the original file name (blast_results_checked.fasta). Results are written in [your output path]/blast_results_checked.fasta

2.5 Alignments Trimming



PyNCBIminer can perform alignment trimming using trimAl. To trim an alignment is to remove of spurious sequences or poorly aligned regions from an alignment, so this step is NOT necessary and depends on the quality of your data (and quality of the alignment). It's a step of optimization. For a quick start, only input and output path are required, and are listed as below:

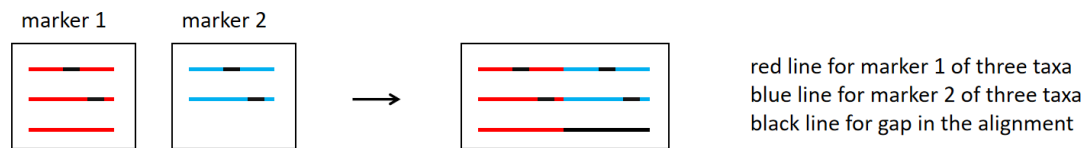
input path - The input file of alignment to be trimmed in fasta format.

output path - The destination folder of output, where log files and result will be written.

1. Set the input path and output path by clicking 'view' button or pasting from clipboard.
2. Then click the 'run' button (simply leaving other parameters default or blank is still okay).

Assume that you haven't changed the original file name (blast_results_checked.fasta). Results are written in [your output path]/blast_results_checked.fasta

2.6 Alignments concatenation



PyNCBIminer can perform alignment concatenation to concatenate alignments of multiple markers to build supermatrix. Markers of each taxon from different input files will be concatenated end to end, and missing markers will be filled with gap '-'. For a quick start (the same is for a normal run), only input and output path are required, and are listed as below:

input path - The input files of alignments to be concatenated in fasta format.

output path - The destination folder of output, where log files and result will be written.

1. Set the input path and output path by clicking 'view' button or pasting from clipboard.
2. Then click the 'run' button.

Results are written in [your output path]/concat.fasta, and we provide phyip format written in [your output path]/concat.phy and a configuration file for PartitionFinder2 written in [your output path]/partition_finder.cfg.

3. Sequence Retrieving Module

To get started, it is recommended that you read the Beginner's Part first if not done so yet.

PyNCBIminer provides Sequence Retrieving module, using iterative BLAST searches to find high quality and complete sequences for a target genetic region and taxonomic group efficiently. Even if you have already collected your data, this module may help to complement the dataset. A brief description of parameters is listed as below.

3.1 Set target region and target taxa.

3.1.1 Working Directory

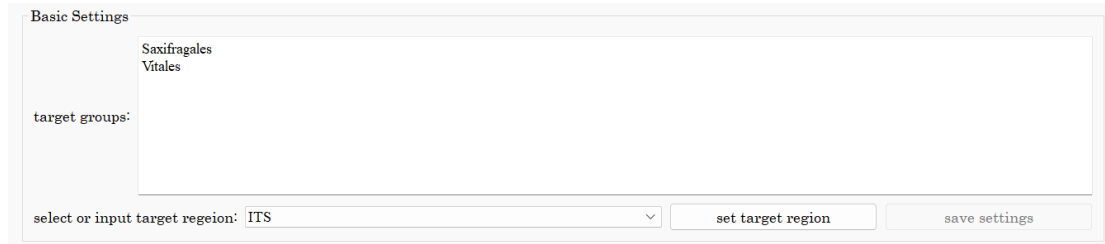
working directory - The destination folder of output.

Working Directory	
working directory: D:\PyNCBIminer_data\ITS_20230308_Saxifragales	view

3.1.2 Basic Settings

target region - desired genetic marker (a gene or a spacer), such as *ITS*, *rbcL* or *matK*.

target groups - target taxonomic groups, such as Saxifragales or Vitales, one group per line.



Basic Settings

target groups:

Saxifragales
Vitales

select or input target region: ITS

set target region

save settings

3.1.3 Advanced Settings

initial queries - Search query.

entrez qualifier - Entrez query results.

publication date - Range of sequence publication dates, restrict the search to sequences published within a specific timeframe.

entrez email - The user's email.

max length - Maximum allowed length for downloaded sequences, sequences exceeding this length will not be downloaded.

expect value - Expect value, the lower value, the higher the similarity expected between downloaded sequences and reference sequences.

word size - Size of word for initial matches.

gap costs - Gap existence and extension costs.

nucleotide reward - Reward for matching base.

nucleotide penalty - Cost for mismatched bases.

exclude sources - Sequences with this annotation information need to be excluded.

key annotations - Sequences with this annotation information need to be retained.

Advanced Settings

initial queries:

CCTGGGCGTCACACACCGTTGCCCCCTTGAACCTCGCCA
ATCCCTTAATGGGAGAAGCATTCAAGTGGG
GCGGAGATTGGCCTCCCGTGAGCTTCTGTCTCGTGGTTGG
CCTAAATTCGAGTCATCGGCTGCGATCGCC
GCGACATTCGGTGGTTTTCGATTATATCGGTGCCCTGTCGT
GCGCGATTCTGTGGCTGAGTAGACCTATG
CGACCCCAATGCGCTGCAAATGCAGTGCCTTCAACGCGAC
CCCAGGTCAGGCGGGATTACCCGCTGAATT
TAAGCATATCAATAAGCGG

entrez qualifier:

(("ITS 1" OR "ITS 2" OR "5.8S" OR "transcribed spacer" OR "ITS1"
OR "ITS2" OR "5.8 S") NOT "environmental sample" NOT
"environmental_sample" NOT "EST." [Keyword] NOT
"UNVERIFIED"[keyword])

publication date:

from

YYYY/MM/DD

to

YYYY/MM/DD

entrez email:

user's email

entrez search

max length:

800

expect value:

10.0

word size:

7

gap costs:

2 1

nucleotide reward:

1

nucleotide penalty:

-1

exclude sources:

mitochondrion
mitochondrial
chloroplast
plastid
environmental sample
environmental_sample

key annotations:

ITS 1
internal transcribed spacer
ITS 2
internal transcribed spacers 1 and 2
5.8S

submit new BLAST

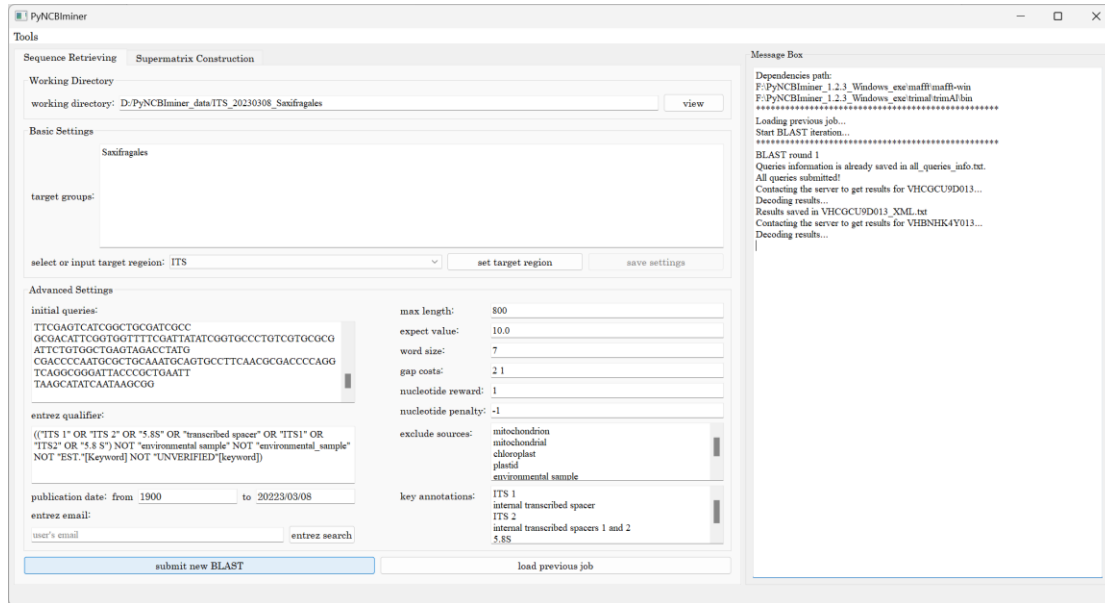
load previous job

You can customize your own queries and edit BLAST parameters in the ‘Advanced Settings’ panel. It is recommended including both distantly and closely related species of the target taxonomic group to cover a larger variation space and reduce BLAST iteration running times. Moreover, it is better to include at least one complete reference sequence in the initial queries dataset to guarantee the completeness of BLAST results. A general suggestion for BLAST is to use higher ‘expect value’ and shorter ‘word size’ for highly variable sequences such as intergenic spacer (*ITS*) and use lower ‘expect value’ and longer ‘word size’ for highly conserved sequences such as chloroplast gene *rbcL*.

The specific descriptions of each BLAST parameter and the allowable values can be accessed on the BLAST website (<https://ncbi.github.io/blast-cloud/dev/api.html>).

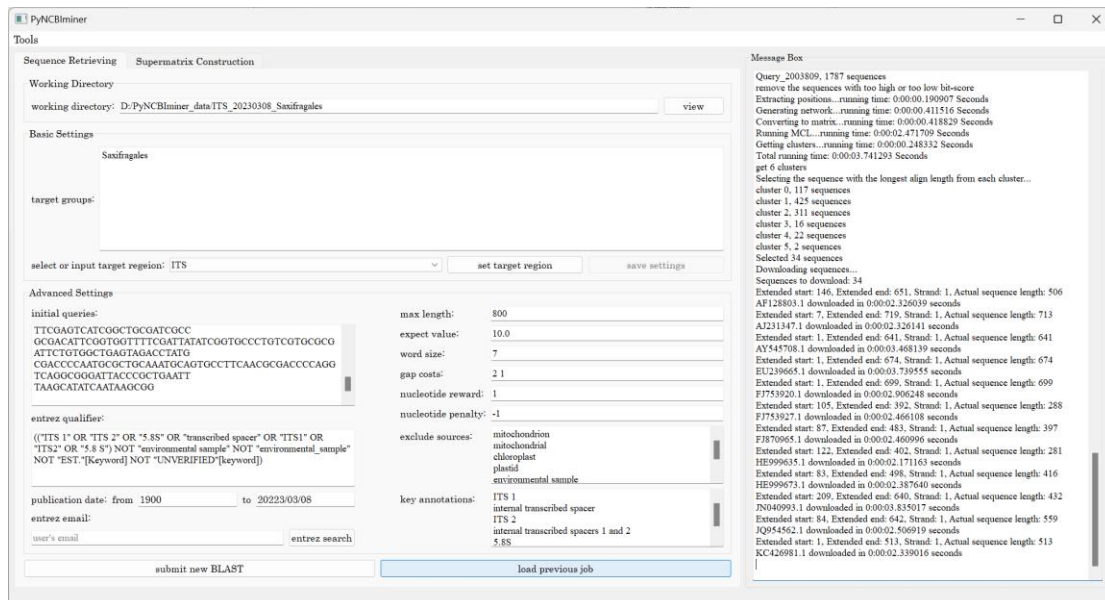
3.1.4 Submit BLAST

Click the 'Submit New BLAST' button in the 'Working directory' section to initiate the BLAST process. The subsequent steps require no user intervention, as PyNCBIminer software will automatically select representative reference sequences from each round of BLAST for iteration. The process continues until no new sequences can be found, at which point the BLAST stops, and sequence downloading begins.



3.1.5 Load unfinished job

If the user does not want to submit new BLAST but to continue running a previous job, they can use the ‘load previous job’ button to resume the terminated job. Please be cautious not to alter the directory structure under 'working directory' or change the names of intermediate files, as doing so may prevent the program from correctly reloading.



3.1.6 View results

Three folders will be created in the working directory.

- ① The parameters folder saves BLAST parameters and query sequences.

blast_parameters.txt <file>: The BLAST parameters

initial_queries.fasta <file>: The fasta file of initial reference sequences.

all_new_queries_info.txt <file>: Information about the newly selected reference sequences in each round.

ref_seq <folder>: The newly selected reference sequences in fasta format.

ref_msa <folder>: The newly selected reference sequence alignment results.

- ② The results folders contain BLAST result and downloaded sequences.

blast_results.txt <file>: The final BLAST results.

blast_results_checked.fasta <file>: The fasta file for correctly annotated sequences.(We typically use this file for subsequent phylogenetic analysis.)

blast_results_checked_seq_info.txt <file>: The sequence information for correctly annotated sequences.

erroneous_blast_results_checked.fasta <file>: The fasta file for erroneous annotation sequences.

erroneous_blast_results_checked_seq_info.txt <file>: The sequence information for erroneous annotation sequences.

- ③ The tmp_files folder saves original HitTables of each round of BLAST.

The 'tmp_files' folder stores intermediate results for each round of BLAST, creating multiple subfolders under this directory with a prefix 'BLAST_' followed by numerical identifiers, each independently saving the results of each round of BLAST. Within each round's result folder, files with the suffix '_XML.txt' represent the original XML files of the BLAST results (deleted after parsing into a HitTable to save space). Files with the suffix '_HitTable.txt' contain sequence information extracted from the XML file in tabular form. Files with the suffix '_joined.txt' present the results after merging hits with the same accession. 'hits_selected.txt' represents the final results obtained after merging, filtering, and extending in that round of BLAST.

The remaining files are intermediate files generated during the process of selecting new reference sequences. 'hits_clustered.fasta' contains candidate reference sequences clustered based on query start and query end. 'new_queries.fasta' includes the newly selected reference sequences further clustered based on sequence similarity.

4. Supermatrix Construction Module

In short, the ‘Supermatrix Construction’ module provides tools and integrated phylogenetic analysis programs for sequences filtering, sequence alignment, alignments trimming and alignments concatenation. To get started, it is recommended that you read the Beginner's Part first if not done so yet.

NOTE: In this module, if input is a director, PyNCBIminer will conduct analysis of the readable files DIRECTLY inside the director AS THEY ARE ONE and ignore others that are not directly inside this director.

4.1 Sequences Filtering

PyNCBIminer can perform sequence filtering simply according to length of the sequences. Only the longest sequence of each taxon will be kept after filtering, and a brief description of parameters is listed as below.

input path - The input file(s) of sequences to be filtered in fasta format.

output path - The destination folder of output, where log files and result will be written.

Two files will be created in the output path after sequences are filtered. The text file in fasta format contains the filtered sequences. The csv table contains which sequences are kept.

4.2. Sequences Alignment

PyNCBIminer can perform sequence alignment using MAFFT, a multiple sequence alignment program, and a brief description of parameters is listed as below. For a more detailed explanation of the parameters corresponding to those from MAFFT, please refer to the manual <https://mafft.cbrc.jp/alignment/software/manual/manual.html>.

input path - The input file(s) of sequences to be aligned in fasta format.

output path - The destination folder of output, where log files and result will be written.

algorithm - MAFFT parameter Algorithm, default is ‘auto’.

thread - The number of threads, -1 if unsure.

reorder - If true, the sequences will be reordered according to similarity.

In output folder, the text file in fasta format contains the alignment.

4.3 Alignments Trimming

PyNCBIminer can perform alignment trimming using trimAl, a tool for the automated removal

of spurious sequences or poorly aligned regions from a multiple sequence alignment, and a brief description of parameters is listed as below. For a more detailed explanation of the parameters corresponding to those from trimAl, please refer to the manual [use of the command line trimAl v1.2 \[trimAl\] \(cgenomics.org\)](http://cgenomics.org/trimAl/v1.2)

input path - The input file of alignment to be trimmed in fasta format.

output path - The destination folder of output, where log files and result will be written.

htmlout - If true, a summary of trimal's work in an HTML file will be provided.

bp length - If true, the kept length of the alignment will be included in the definition line.

implement methods - See trimal parameter [gappyout, strict, strictplus, automated1].

gt - Trimal parameter gt: gapthreshold, 1 - (fraction of sequences with a gap allowed).

st - Trimal parameter st: simthreshold, minimum average similarity allowed.

ct - Trimal parameter ct: conthreshold, minimum consistency value allowed.

cons - Trimal parameter cons: Minimum percentage of positions in alignment to conserve.

A file (and maybe several folders, depends on parameter settings) will be created in the output path after the alignment is trimmed. The text file in fasta format contains the trimmed alignment.

4.4 Alignments concatenation

PyNCBIminer can perform alignment concatenation to concatenate alignments of multiple markers to build supermatrix. Markers of each taxon from different input files will be concatenated end to end, and missing markers will be filled with gaps '-'. A brief description of parameters is listed as below.

input path - The input files of alignments to be concatenated in fasta format.

output path - The destination folder of output, where log files and result will be written.

Several folders will be created in the output path after the alignments are concatenated. In completion.result (if exists), log file records which missing taxa are added with gaps, and fasta files are edited files (gaps added if necessary). In concat.result and phylip.result, files in fasta format and phylip format are results of concatenation, and the cfg file (partition_finder.cfg) is the configuration file for PartitionFinder2, if needed.

5. Python version and packages for running the source code

The source code of PyNCBIminer includes three python files, pyNCBIminer_00_main.py, pyNCBIminer_01_tools.py and pyNCBIminer_02_ui.py. The python file pyNCBIminer_00_main.py serves as the starting point for program execution and connects buttons in the main widow with various functions. The python file pyNCBIminer_00_tools.py contains the functions used by pyNCBIminer_00_main.py and cannot be executed directly but be imported and

called by other scripts. The python file `pyNCBIminer_02_ui.py` defines the main window of graphical user interface. The “initial_queries” folder is where the implemented initial queries of PyNCBIminer are saved and needs to be kept in the same directory with the source code. PyNCBIminer source code can be run in the terminal by input the command line like “python `pyNCBIminer_00_main.py`” or “python3 `pyNCBIminer_00_main.py`”.

Python version: 3.9.5

Package	Version

biopython	1.79
func-timeout	4.3.5
markov-clustering	0.0.6.dev0
matplotlib	3.5.1
networkx	2.8.4
numpy	1.21.2
pandas	1.3.4
PySide2	5.15.2.1
scikit-learn	1.1.1
scipy	1.8.1