PyNCBIminer Manual

Contents

PyNCBIminer Manual 1
1 Dependencies and installation 1
1.1 Python version and packages for running the source code 1
1.2 Dependencies 1
1.3 Installation 2
2 Quick Start 3
2.1 If you want to perform analysis on your own dataset 3
2.2 Sequence Retrieval 4
2.3 Sequences Filtering 5
2.4 Alignment, trimming and concatenation 7
3 Sequence Retrieval Module 10
3.1 Set target region and target taxa. 10
3.2 Submit BLAST 13
3.3 Load unfinished job 13
3.4 View results 14
4 Supermatrix Construction Module 15
4.1 Sequences Filtering 15
4.2. Sequences Alignment 15
4.3 Alignments Trimming 16
4.4 Alignments concatenation 16

PyNCBIminer Manual

1 Dependencies and installation

1.1 Python version and packages for running the source code

The PyNCBIminer source code consists of multiple Python files. The main script, `pyNCBIminer_00_main.py`, serves as the entry point for executing the program and connects the buttons in the main window to various functions. The file `ui_main.py` defines the graphical user interface (GUI) of the main window. Other Python files contain functions that are utilized by `pyNCBIminer_00_main.py`; these files are not meant to be run directly but are imported and called by other scripts. The `initial_queries` folder stores the pre-defined initial queries implemented by PyNCBIminer, while the `blast_parameters` folder holds the default parameters. Both of these folders should be kept in the same directory as the source code. To run PyNCBIminer, you can use a terminal command such as `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	1.3.0
scipy	1.8.1

1.2 Dependencies

To install and run the core functions of PyNCBIminer on both Windows and macOS, Python 3 (version 3.9.5) must be installed. You can download it from https://peps.python.org/pep-0596/. Additionally, PyNCBIminer requires an active internet connection for several of its modules to function properly. The software has been tested for compatibility with Windows 10 and 11, as well

as macOS versions 13.4 and 14.5.

MacOS users would need to also install Anaconda (https://www.anaconda.com/download/) or miniconda (https://docs.anaconda.com/miniconda/) in order to install the software (see Installation instructions).

PyNCBIminer also provides some extended functionally (e.g. building multiples sequence aligments, aignmet trimming etc.), which are 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. The required executables for MAFFT and trimAl are included with the PyNCBIminer executable, so users do not need to install them separately. However, if users are running the source code, they will need to manually install these tools and add them to the system environment variables.

1.3 Installation

For Windows: Users only need to unzip the package and double click the executable PyNCBIminer.exe file directly.
For MacOS: Users using MacOS would need to follow the 9 Steps describe below to Run PyNCBIminer:

Step1: Download the newest version (1.2.8 on 10-August-2024) of the PyNCBIminer MacOS source code from https://github.com/Xiaoting-Xu/PyNCBIminer and extract to your local directory of choice.

Step2: Open anaconda prompt in Anaconda Navigator or Miniconda (if not installed, we recommend installing Anaconda).

Step3: Change the working directory (cd) to the directory where PyNCBIminer are located. Make sure that anaconda is activated, and you can see the anaconda prompt. All commands in Step 4-9 should be executed in the anaconda prompt. After extracting in a local directory, you should see files such as requirements.txt, pyNCBIminer_00_main.py and so on. It is recommended to add this directory path into environment variable for convenience.

Step4: `conda create -n [name] python==3.9.5` # For example, `conda create -n pyNCBIminer python==3.9.5`. If failed, try `conda create -n [name] python=3.9`.

Step5: `conda activate [name]` # For example, `conda activate pyNCBIminer`

Step6: `pip install requirements.txt` # Install all the dependencies.

Step7: `pip install certifi` # Install certificate package.

Step8: `Install Certificates.command` # Install certificate package to prevent SSL problems.

Step9: `python pyNCBIminer_00_main.py` # run our program

Once you have installed PyNCBIminer and you have created an anaconda environment for it, you only need to follow step3, step5 and step9 in subsequent uses.

Note that the executable file (or python scripts e.g. `pyNCBIminer_00_main.py`) and the `initial_queries` and `blast_parameters` folder should always be in the same directory. Right click the executable file to make alias (create a shortcut) for convenience if necessary.

2 Quick Start

This part provides a quick beginner's guide.

We highly recommend creating a new directory for each step in case of unexpected overwriting.

2.1 If you want to perform analysis on your own dataset

If you wish to use PyNCBIminer to analyze your own datasets, you may need to adjust the folder and file organization, as well as the structure of the definition lines, to meet the software's requirements. The criteria are outlined below, with examples provided in subsequent sections (2.3.1 and 2.4.1).

Within the Supermatrix Construction Module (which includes Sequence Filtering, Sequence Alignment, Alignment Trimming, and Alignment Concatenation), inputs can be either a single file or a directory.

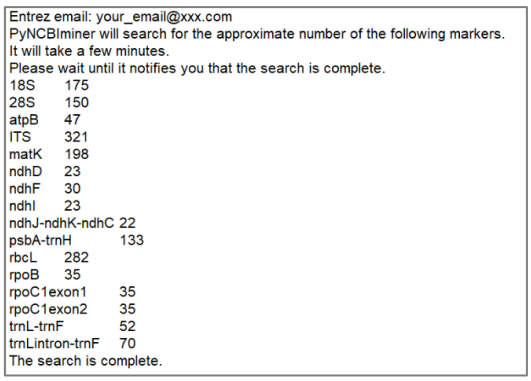
For the Sequence Filtering sub-module of the Supermatrix Construction Module, the FASTA definition line (the line preceding the nucleotide sequence) must follow a specific format. It should begin with a caret '>' and be structured as: `>[accession number]|[taxon name]|[further description]`. Each component in the square brackets should be separated by a vertical bar '|'.

In the Alignment Concatenation sub-module, it is recommended that the definition line be formatted as `>[taxon name]`.

2.2 Summarize widely used marker

To quickly assess data coverage of all available markers for their target group in GenBank, users can input the desired taxonomic group, select "Summarize Widely Used Markers," and then

click the "Entrez Search" button. This will provide an approximate count of sequences available in GenBank for the specified group. Users can perform this quick search for the pre-implemented markers to identify the most commonly used ones before proceeding with BLAST searches.



2.2 Sequence Retrieval

PyNCBIminer provides Sequence Retrieval module, using iterative BLAST searches to find high quality and complete sequences for a targeted 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 as listed below:

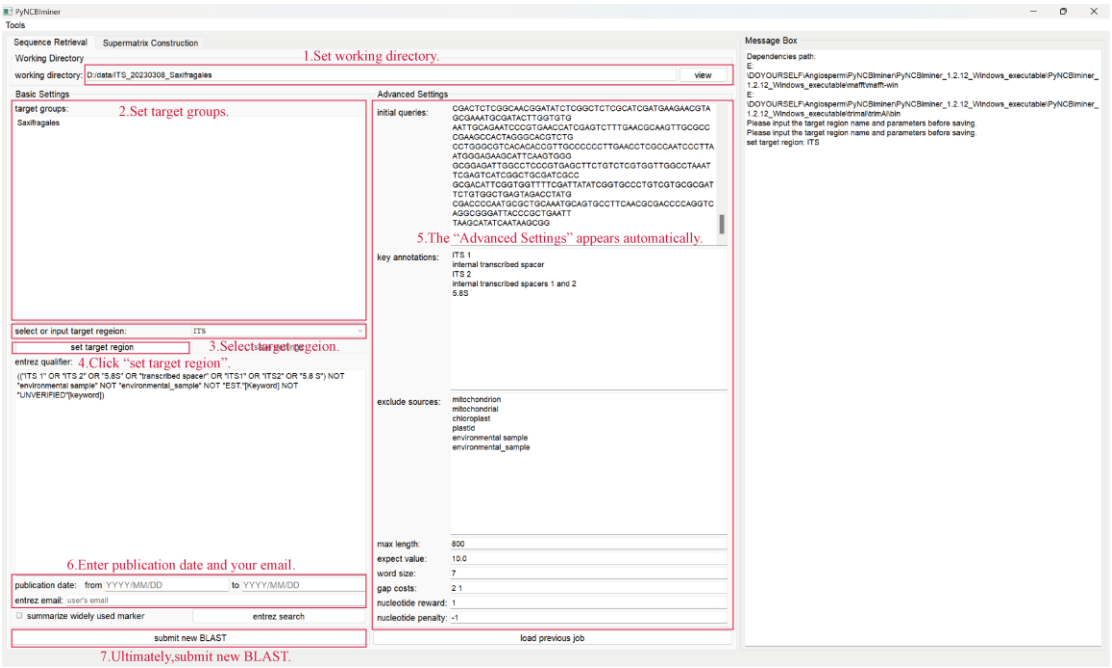
working directory - The destination folder for any output.

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

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

1. Set working directory by clicking ‘view’ button and navigate to the desired location or paste path from clipboard.
2. Type the target group in the ‘target group’ text field, 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 with default values, and you don’t have to modify them if your search is within the scope of the ‘quick start’ options.
6. Optional: Enter publication date, this will 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. Finally, 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 setup, or refer to 3. Sequence Retrieving Module for a better setting of parameters.

1. If rate of the desired marker mutation is really fast, you can select ‘ITS’ in the pull-down list. If the rate of mutation is 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 3. Sequence Retrieving Module for detailed information.

Results are written in [your working directory]/results/blast_results_checked.fasta.

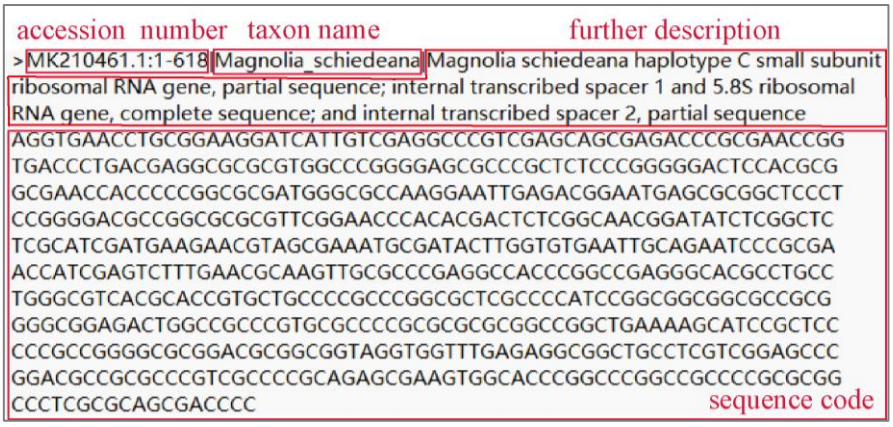
2.3 Sequences Filtering

2.3.1 Analyze your own data

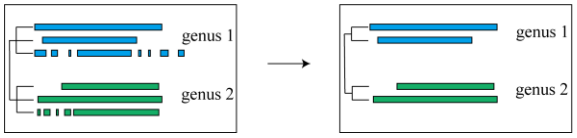
If you want to perform sequence filtering with your own data, the optimal time to add it is after conducting the Extended Segments Refinement, which is tailored for sequences retrieved by PyNCBIminer, but before species-level selection. This ensures that PyNCBIminer retains your data as the representative sequence for each taxon while removing redundant sequences for the same species. To add your data, open the `blast_results_exception_removed.fasta` file in the `results` folder using a text editor, and append your sequences to this file. Make sure to format your sequences

correctly so PyNCBIminer can extract the necessary information. The definition line (in FASTA format, this is the line preceding the nucleotide sequence) must begin with a caret '>', followed by a unique SeqID, and should be structured as `>[accession number]|[taxon name]|[further description]`.

For example: `>AB021052.1|Magnolia_schiedeana|Magnolia schiedeana chloroplast atpB, rbcL genes, partial cds, spacer region`.



2.3.2 Extended segments refinement



each color for sequence of one genus
extended segments with poor alignment results will be removed.

PyNCBlminer can perform 'Extend segments refinement' to check for extended segments. After aligning sequences for each genus, the extended part(s) of sequences will be removed if poorly aligned. For this option, only input and output path are required, and are listed as below:

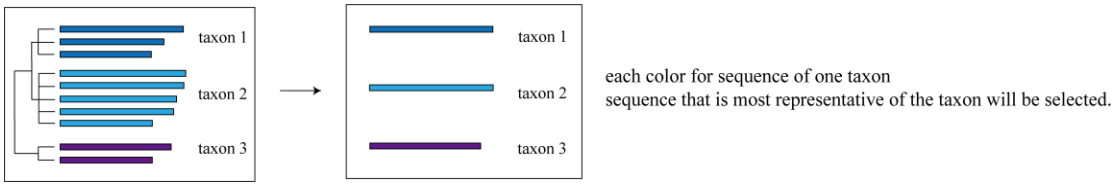
input path - The input file(s) of sequences to be filtered in fasta format.

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.

The refined 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.

2.3.3 Species-level sequence selection



PyNCBIminer can perform 'Species-level sequence selection' to select a single sequence to represent each species. This module consists of two methods. The first method calculates an abnormal index for each sequence by comparing it to a consensus sequence, helping to exclude incorrect data. The second method balances sequence length and alignment quality, assessing alignment via BLAST bit-scores. Preference is given to sequences with a voucher and the most recently published ones when selecting multiple sequences for a species. This module requires only input and output path to be set. Users also can incorporate their own data into the FASTA format sequences generated from the sequence selection process:

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.

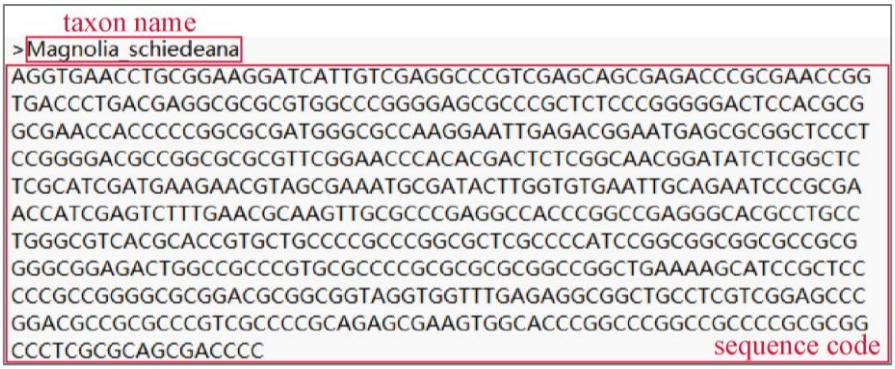
Assuming that you haven't changed the original file name (blast_results_checked.fasta). Results are written in [your output path]/blast_results_checked.fasta. More information of kept results are in [your output path]/blast_results_checked_kept_records.csv

2.4 Alignment, trimming and concatenation

2.4.1 Analyze your own dataset

If you want to use PyNCBIminer to perform alignment, trimming, and concatenation on your own dataset, you will need to adjust the organization of folders and files accordingly. The input can either be a file or a directory. If the input is a file, PyNCBIminer will analyze this specific file. If the input is a directory, the software will analyze all readable files directly within that directory as a single dataset. However, any files located within subdirectories will NOT be processed. For example, in a directory structure like `/input_directory/another_directory/file1`, only files directly under `/input_directory/` will be analyzed, while file1 and any other files within `/input_directory/another_directory/` will be ignored.

Alignment and trimming steps do not impose any specific requirements on the format of the definition lines in FASTA files. However, during the concatenation step, it is necessary to reformat your files. In the Alignment Concatenation sub-module of the Supermatrix Construction Module, it is recommended that the definition line follows the format `>[taxon name]`, as taxon names are used to match sequences from the same organisms when concatenating multiple matrices.



2.4.2 Sequences Alignment

PyNCBIminer can perform ‘Sequence Alignments’ using MAFFT. For a fast run, 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, the default path is the same as the input path 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

2.4.3 Alignments Trimming

PyNCBIminer can perform 'Alignment Trimming' using trimAl to remove 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). For a fast run, only input and output path are required, and are listed as below:

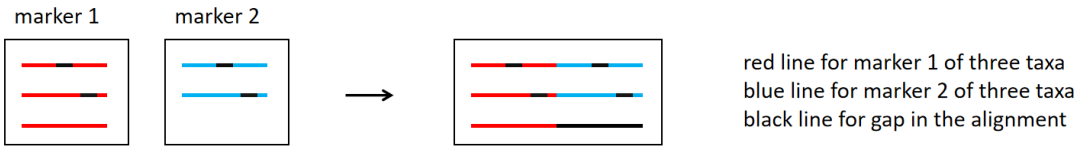
input path - The input file of alignment to be trimmed in fasta format.

output path - The destination folder of output, the default path is the same as the input path, 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).

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

2.4.4 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 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, the default path is the same as the input path, 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 phylip format written in [your output path]/concat.phy and a configuration file for PartitionFinder2 written in [your output path]/partition_finder.cfg.

3 Sequence Retrieval Module

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

PyNCBIminer provides Sequence Retrieval module, that uses 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.

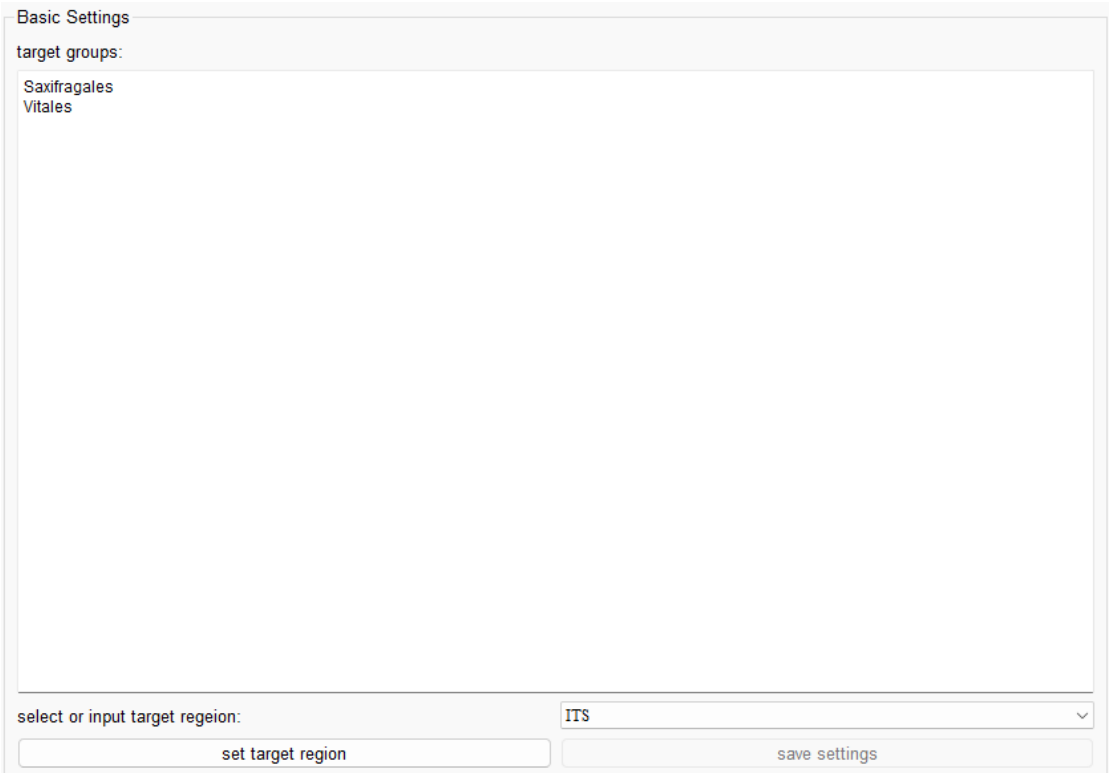
3.1.1 Working Directory

working directory - The destination folder of output.

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.



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 - the length of the short sequence of nucleotides (or amino acids) that BLAST uses to find initial matches between the query sequence and sequences in the database.

gap costs - the penalties applied when introducing gaps (insertions or deletions) in alignments to account for differences between sequences. The gap cost is typically divided into two components: Gap existence cost and Gap extension cost.

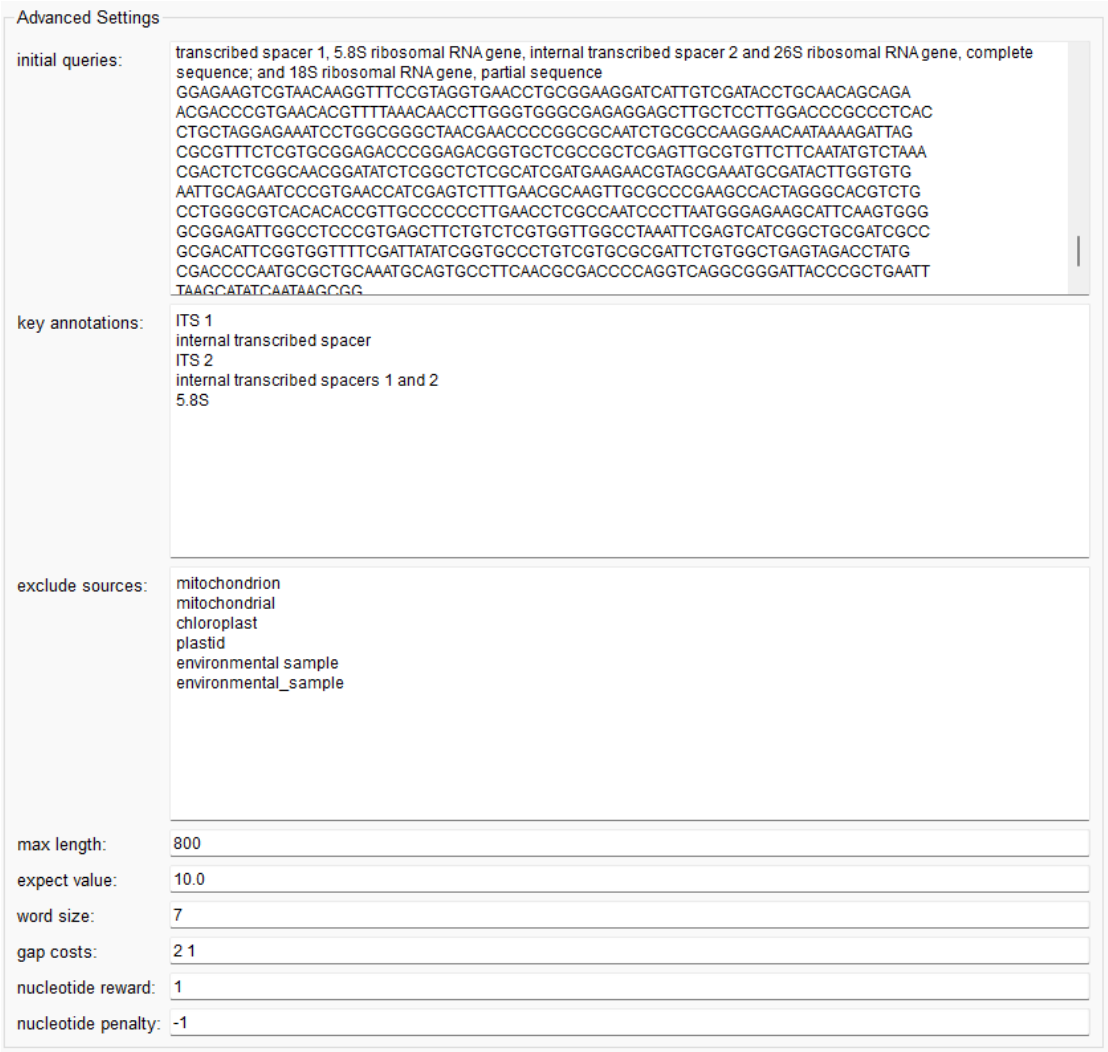
nucleotide reward - the score added for each correctly matched base (A, T, C, or G) during the alignment. A higher reward value will prioritize matching exact bases over introducing gaps or mismatches, making the alignment stricter in preserving exact matches.

nucleotide penalty - the score deducted for each mismatched base in the alignment. Higher

penalties discourage mismatches, leading to more conservative alignments where only very close matches are considered.

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

key annotations - Sequences without any of this annotation information need to be excluded.

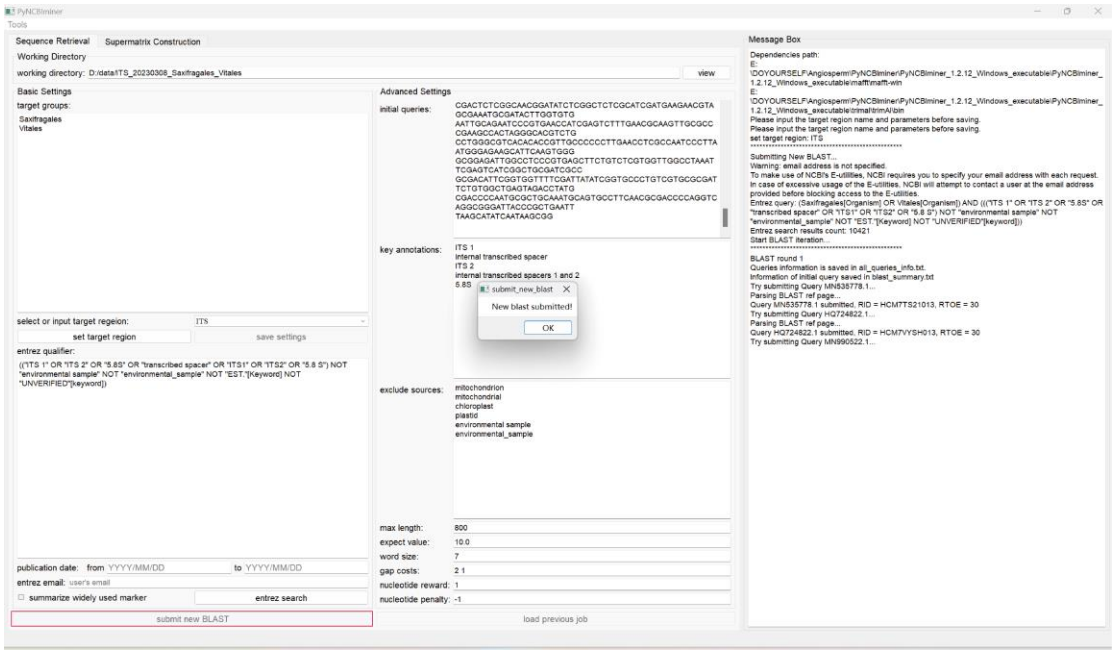


You can customize your own queries and edit BLAST parameters in the 'Advanced Settings' panel. It is recommended including both distantly and closely related species of the target taxonomic group to cover a larger variation space and reduce BLAST iteration running times. Moreover, it is better to include 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).

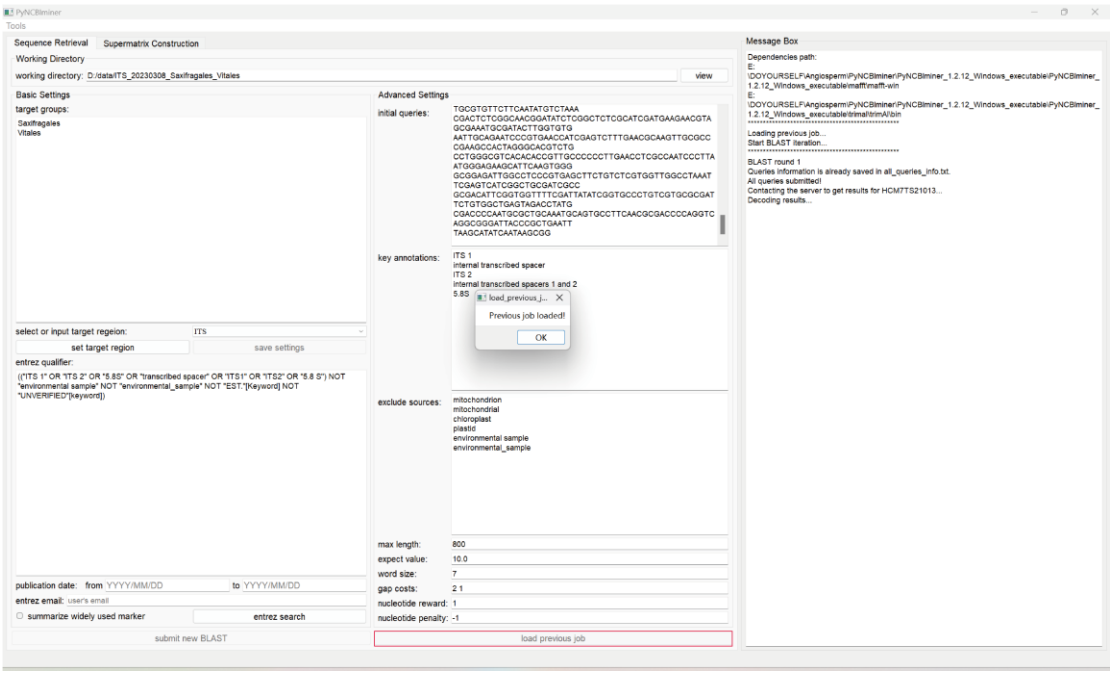
3.2 Submit BLAST

Click the 'Submit New BLAST' button in the 'Working directory' section to initiate the BLAST process, and then a dialog box "New blast submitted" will appear. 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.



3.3 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, and then a dialog box "Previous job loaded" will appear. 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 reloading correctly.



3.4 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 for sequences filtering, sequence alignment, alignments trimming and alignments concatenation. These are run locally on the user’s computer, so if one downloads large datasets he has to consider if it is feasible to do that locally. To get started, it is recommended that you read the Quick Start first if not done so yet.

NOTE: In this module, if input is a directory, PyNCBIminer will conduct analysis of the readable files DIRECTLY inside that director and will ignore files in subfolders.

4.1 Sequences Filtering

PyNCBIminer in ‘Sequences Filtering’ module can perform ‘Extended segments refinement’ and ‘Species-level sequence selection’ to check for extended segments and select a single sequence to represent each species. User also can incorporate their own data into FASTA format sequences generated from the sequence selection process:

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. Choose -1 if unsure, and the number of threads will be determined automatically.

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)

A short overview of the implemented parameters is as follows:

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.