
CONSTAXv2

Release 2.0.17

Julian A. Liber and Gian M. N. Benucci

Mar 15, 2022

CONTENTS:

1	CONSTAX 2.0.17 improves upon 1.0.0 with the following features:	3
2	Developed by	5
3	Funded by	7
4	CONSTAX 1.0.0 was authored by	9
5	Reference	11
5.1	License	11
5.1.1	Contacts	11
5.2	Installation	12
5.2.1	Simple installation with conda for Linux/OSX/WSL	12
5.3	Suggested Reference Databases	13
5.4	CONSTAX Options	13
5.4.1	Options details	15
5.5	Run CONSTAX locally	18
5.6	Run CONSTAX on HPCC	22
5.7	Download and generate SILVA reference database	24
5.8	Downloading the UNITE database	26
5.9	Examine SH (Species Hypothesis) hits from UNITE database	26
5.10	Help	27
6	Indices and tables	29

CONSTAX (*CON*Sensus *TAX*onomy) is a tool, written in Python 3, for improved taxonomic resolution of environmental DNA sequences. Briefly, CONSTAX compares the taxonomic classifications obtained from RDP Classifier, UTX or BLAST, and SINTAX and merges them into an improved consensus taxonomy using a 2 out of 3 rule (e.g. If an OTU is classified as taxon A by RDP and UTX/BLAST and taxon B by SINTAX, taxon A will be used in the consensus taxonomy) and the classification p-value to break the ties (e.g. when 3 different classification are obtained for the same OTU). This tool also produces summary classification outputs that are useful for downstream analyses. In summary, our results demonstrate that independent taxonomy assignment tools classify unique members of the fungal community, and greater classification power (proportion of assigned operational taxonomic units at a given taxonomic rank) is realized by generating consensus taxonomy of available classifiers with CONSTAX.

CONSTAX 2.0.17 IMPROVES UPON 1.0.0 WITH THE FOLLOWING FEATURES:

- Updated software requirements, including Python 3 and Java 8
- Simple installation with conda
- Compatibility with SILVA-formatted databases (for Bacteria, Archaea, protists, etc.)
- Streamlined command-line implementation
- BLAST classification option, due to legacy status of UTAX
- Parallelization of classification tasks
- Isolate matching

CHAPTER
TWO

DEVELOPED BY

- Julian A. Liber
- Gian M. N. Benucci

CHAPTER
THREE

FUNDED BY

- Gregory Bonito

CONSTAX 1.0.0 WAS AUTHORED BY

- Kristi Gdanetz MacCready
- Gian M. N. Benucci
- Natalie Vande Pol
- Gregory Bonito

REFERENCE

Liber JA, Bonito G, Benucci GMN (2021) CONSTAX2: improved taxonomic classification of environmental DNA markers. *Bioinformatics* doi: 10.1093/bioinformatics/btab347

Gdanetz K, Benucci GMN, Vande Pol N, Bonito G (2017) CONSTAX: a tool for improved taxonomic resolution of environmental fungal ITS sequences. *BMC Bioinformatics* 18:538 doi 10.1186/s12859-017-1952-x

See the menu on the left for how to install CONSTAX and how to use it.

5.1 License

MIT License

Copyright (c) 2021 JAL&GMNB&GMB

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

5.1.1 Contacts

Do you have questions about the software license? Please contact [Julian A. Liber](#) or [Gian M. N. Benucci](#)

5.2 Installation

5.2.1 Simple installation with conda for Linux/OSX/WSL

CONSTAX is a command line tool. You will need to open and run commands in a terminal to use it. Windows users can [install WSL](#) to use CONSTAX or [custom install](#) on their machine.

CONSTAX comes in a conda package that contains all the dependencies needed to run the software and can be easily installed as showed below.

```
conda install constax -c bioconda
```

If conda is not installed (you get an error which might include `command not found`), [follow their instructions](#) to install it. Briefly:

1. Download the correct installation for your system, and run it.

- Miniconda installation commands:

Linux / WSL

OSX

```
wget https://repo.anaconda.com/miniconda/Miniconda3-py39_4.10.3-Linux-x86_64.sh
bash Miniconda3-py39_4.10.3-Linux-x86_64.sh
```

```
curl -O https://repo.anaconda.com/miniconda/Miniconda3-py39_4.10.3-MacOSX-x86_64.
↪ sh
bash Miniconda3-py39_4.10.3-MacOSX-x86_64.sh
```

2. Follow the prompts.
3. Close and reopen terminal.
4. Try the command `conda list`.
5. Proceed to installing CONSTAX as above.

Custom installation of USEARCH

If you want to use USEARCH which is a proprietary, instead of VSEARCH, you will have to install it yourself and generate a `pathfile.txt` to specify the binary location. Please see the tutorial sections.

- USEARCH/VSEARCH
 - USEARCH installation from [drive5](#)

Linux/WSL

Windows

OSX

```
wget https://www.drive5.com/downloads/usearch11.0.667_i86linux32.gz
gunzip usearch11.0.667_i86linux32.gz
```

```
curl -O https://www.drive5.com/downloads/usearch11.0.667_win32.gz
gunzip usearch11.0.667_win32.gz
```

```
curl -O https://www.drive5.com/downloads/usearch11.0.667_i86osx32.gz
gunzip usearch11.0.667_i86osx32.gz
```

- **VSEARCH** can be installed by [conda](#), [pip](#), or downloading from [source](#).

5.3 Suggested Reference Databases

Dependent on where your sequences originate (e.g. ITS, 16S, LSU), you will need to have an appropriate database with which to classify them.

For Fungi or all Eukaryotes, the **UNITE** database is preferred. The format of the reference database to use with CONSTAX is one of those under the **General** fasta format. For the latest release (10.05.2021), training with 32GB of RAM for Fungi only or 40GB for all Eukaryotes should be sufficient.

For Bacteria and Archaea, we recommend the **SILVA** reference database. The `SILVA_XXX_SSURef_tax_silva.fasta.gz` file can be gunzip-ped and used.

Note: SILVA taxonomy is not assigned by Linnean ranks (*Kingdom*, *Phylum*, etc.), so instead placeholder ranks 1-n are used. Also, the size of the SILVA database means that a server/cluster is required to train the classifier because 128GB RAM for the RDP training are required. If you have a computer with 32GB of RAM, you may be able to train using the UNITE database. If you cannot train locally for UNITE, the RDP files can be downloaded from [here](#). The `genus_wordConditionalProbList.txt.gz` file should be gunzip-ped after downloading.

5.4 CONSTAX Options

To visualize CONSTAX options:

```
gian@gian-Z390-GY:~/tutorial$ constax --help
```

This is what CONSTAX will display on the terminal

```
# constax --help
usage: constax [-h] [-c CONF] [-n NUM_THREADS] [-m MHITS]
               [-e EVALUATE] [-p P_IDEN] [-d DB] [-f TRAINFILE]
               [-i INPUT] [-o OUTPUT] [-x TAX] [-t] [-b]
               [--select_by_keyword SELECT_BY_KEYWORD] [--msu_hpcc]
               [-s] [--consistent] [--make_plot] [--check]
               [--mem MEM] [--sintax_path SINTAX_PATH]
               [--utax_path UTAX_PATH] [--rdp_path RDP_PATH]
               [--constax_path CONSTAX_PATH] [--pathfile PATHFILE]
               [--isolates ISOLATES]
               [--isolates_query_coverage ISOLATES_QUERY_COVERAGE]
               [--isolates_percent_identity ISOLATES_PERCENT_IDENTITY]
               [--high_level_db HIGH_LEVEL_DB]
               [--high_level_query_coverage HIGH_LEVEL_QUERY_COVERAGE]
               [--high_level_percent_identity HIGH_LEVEL_PERCENT_IDENTITY]
               [--combine_only] [-v]

optional arguments:
  -h, --help                show this help message and exit
  -c CONF, --conf CONF      Classification confidence threshold (default: 0.8)
```

(continues on next page)

(continued from previous page)

```

-n NUM_THREADS, --num_threads NUM_THREADS
    directory to for output files (default: 1)
-m MHITS, --mhits MHITS
    Maximum number of BLAST hits to use, for use with -b
    option (default: 10)
-e EVALUE, --evaluate EVALUE
    Maximum expect value of BLAST hits to use, for use
    with -b option (default: 1.0)
-p P_IDEN, --p_iden P_IDEN
    Minimum proportion identity of BLAST hits to use, for
    use with -b option (default: 0.0)
-d DB, --db DB
    Database to train classifiers, in FASTA format
    (default: )
-f TRAINFILE, --trainfile TRAINFILE
    Path to which training files will be written (default:
    ./training_files)
-i INPUT, --input INPUT
    Input file in FASTA format containing sequence records
    to classify (default: otus.fasta)
-o OUTPUT, --output OUTPUT
    Output directory for classifications (default:
    ./outputs)
-x TAX, --tax TAX
    Directory for taxonomy assignments (default:
    ./taxonomy_assignments)
-t, --train
    Complete training if specified (default: False)
-b, --blast
    Use BLAST instead of UTAX if specified (default:
    False)
--select_by_keyword SELECT_BY_KEYWORD
    Takes a keyword argument and --input FASTA file to
    produce a filtered database with headers containing
    the keyword with name --output (default: False)
--msu_hpcc
    If specified, use executable paths on Michigan State
    University HPCC. Overrides other path arguments
    (default: False)
-s, --conservative
    If specified, use conservative consensus rule (2 False
    = False winner) (default: False)
--consistent
    If specified, show if the consensus taxonomy is
    consistent with the real hierarchical taxonomy
    (default: False)
--make_plot
    If specified, run R script to make plot of classified
    taxa (default: False)
--check
    If specified, runs checks but stops before training or
    classifying (default: False)
--mem MEM
    Memory available to use for RDP, in MB. 32000MB
    recommended for UNITE, 128000MB for SILVA (default:
    32000)
--sintax_path SINTAX_PATH
    Path to USEARCH/VSEARCH executable for SINTAX
    classification (default: False)
--utax_path UTAX_PATH
    Path to USEARCH executable for UTAX classification
    (default: False)
--rdp_path RDP_PATH
    Path to RDP classifier.jar file (default: False)
--constax_path CONSTAX_PATH
    Path to CONSTAX scripts (default: False)
--pathfile PATHFILE
    File with paths to SINTAX, UTAX, RDP, and CONSTAX
    executables (default: pathfile.txt)

```

(continues on next page)

(continued from previous page)

```

--isolates ISOLATES    FASTA formatted file of isolates to use BLAST against
                        (default: False)
--isolates_query_coverage ISOLATES_QUERY_COVERAGE
                        Threshold of sequence query coverage to report isolate
                        matches (default: 75)
--isolates_percent_identity ISOLATES_PERCENT_IDENTITY
                        Threshold of aligned sequence percent identity to
                        report isolate matches (default: 1)
--high_level_db HIGH_LEVEL_DB
                        FASTA database file of representative sequences for
                        assignment of high level taxonomy (default: False)
--high_level_query_coverage HIGH_LEVEL_QUERY_COVERAGE
                        Threshold of sequence query coverage to report high-
                        level taxonomy matches (default: 75)
--high_level_percent_identity HIGH_LEVEL_PERCENT_IDENTITY
                        Threshold of aligned sequence percent identity to
                        report high-level taxonomy matches (default: 1)
--combine_only         Only combine taxonomy without rerunning classifiers
                        (default: False)
-v, --version          Display version and exit (default: False)

```

5.4.1 Options details

```
-c, --conf=0.8
```

Classification confidence threshold, used by each classifier (0,1]. Increase for improved specificity, reduced sensitivity.

```
-n, --num_threads=1
```

Number of threads to use for parallelization. Maximum classification speed at about 32 threads. Training only uses 1 thread.

```
-m, --max_hits=10
```

Maximum number of BLAST hits to use, for use with -b option. When classifying with BLAST, this many hits are kept. Confidence for a given taxa is based on the proportion of these hits agree with that taxa. 5 works well for UNITE, 20 with SILVA (standard, not NR).

```
-e, --evalue=1
```

Maximum expect value of BLAST hits to use, for use with -b option. When classifying with BLAST, only hits under this expect value threshold are used. Decreasing will increase specificity, but decrease sensitivity at high taxonomic ranks.

```
-p, --p_iden=0.8
```

Minimum proportion identity of BLAST hits to use, for use with -b option. Minimum proportion of conserve bases to keep hit.

```
-d, --db
```

Database to train classifiers. UNITE and SILVA formats are supported. See [Datasets](#).

```
-f, --trainfile=./training_files
```

Path to which training files will be written.

```
-i, --input=otus.fasta
```

Input file in FASTA format containing sequence records to classify.

```
-o, --output=./outputs
```

Output directory for classifications.

```
-x, --tax=./taxonomy_assignments
```

Directory for taxonomy assignments.

```
-t, --train
```

Complete training if specified. Cannot run classification without training files present, so this option is necessary at least at the first time you run CONSTAX or you changed the taxonomic referenced sequence database.

```
-b, --blast
```

Use BLAST instead of UTAX if specified. If installed with conda, this is the option that will work by default. UTAX is available from [USEARCH](#). BLAST classification generally performs better with faster training, similar classification speed, and greater accuracy.

```
--msu_hpcc
```

If specified, use executable paths on Michigan State University HPCC. Overrides other path arguments.

```
--s, conservative
```

If specified, use conservative consensus rule (2 null = null winner. For example, if BLAST is the only algorithm that classifies OTU_135 to Family Strophariaceae while SINTAX and RDP give no classification, then no classification is reported at the rank of Family for OTU_135 in the CONSTAX taxonomy). According to our tests, works better for SILVA database to use this option.

```
--consistent
```

If specified, show if the consensus taxonomy is consistent with the real hierarchical taxonomy. In this case, a 1 indicates that all subtaxa are contained within each parent taxa. For example, the genus assigned is within the family assigned.

```
--make_plot
```

If specified, run R script to make plot of classified taxa. The plot compares how many OTUs were classified at each rank for RDP, SINTAX, BLAST, and CONSTAX.

```
--check
```

If specified, runs checks but stops before training or classifying.

```
--mem
```

Memory available to use for RDP, in MB. 32000MB recommended for UNITE, 128000MB for SILVA. This is necessary for training the referenced databases.

```
--sintax_path
```

Path to USEARCH/VSEARCH executable for SINTAX classification. Can also be `vsearch` if already on path.

```
--utax_path
```

Path to USEARCH executable for UTAX classification.

```
--rdp_path
```

Path to RDP classifier.jar file, or classifier if on path from RDPTools conda install.

```
--constax_path
```

Path to CONSTAX scripts.

```
--pathfile
```

File with paths to SINTAX, UTAX, RDP, and CONSTAX executables. This useful in your local CONSTAX installation, please the tutorial for how to set a pathfile up in your system.

```
--isolates
```

FASTA formatted file of isolates to use BLAST against.

```
--isolates_query_coverage
```

Threshold of sequence query coverage to report isolate matches, in percent.

```
--isolates_percent_identity
```

Threshold of aligned sequence percent identity to report isolate matches.

```
--high_level_db
```

FASTA database file of representative sequences for assignment of high level taxonomy. For this option you can use the [SILVA](#) NR99 database for SSU/16S/18S sequences or the the [UNITE](#) database for Eukaryotic ITS/28S sequences. This option is useful to match your OTUs representative sequences to a reference using a lower cutoff so you can identify for example, which sequences are Fungi and which ones are not.

```
--high_level_query_coverage
```

Threshold of sequence query coverage to report high-level taxonomy matches, in percent.

```
--high_level_percent_identity
```

Threshold of aligned sequence percent identity to report high-level taxonomy matches.

```
--combine_only
```

If specified, only reruns combine taxonomy without rerunning classifiers. Allows for changing parameters including: `-c`, `--conf`, `-e`, `--evaluate`, `-p`, `--p_iden`, `-s`, `--conservative`, `--isolates_query_coverage`, `--isolates_percent_identity`, `--high_level_query_coverage`, and `high_level_percent_identity`.

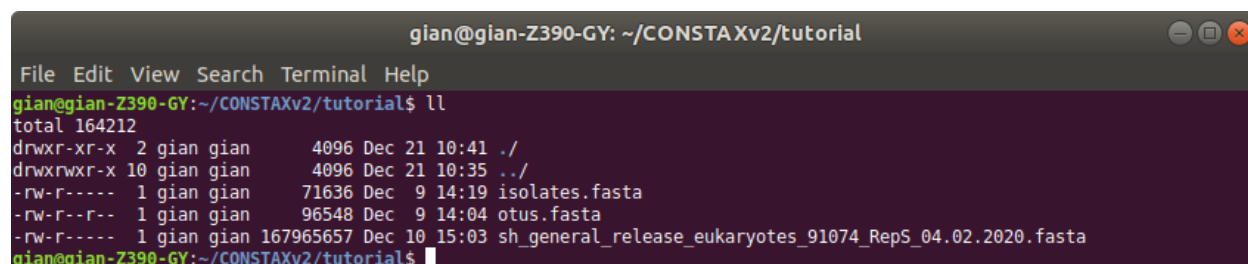
5.5 Run CONSTAX locally

This is a simple tutorial about CONSTAX. We will explain how to run CONSTAX on a local computer like a laptop or a desktop computer.

Before we start, we need to create a folder called `tutorial`. This CONSTAX test will happen inside this folder so you first need to copy all the files you will use before running the software. We need the OTU representative sequence fasta file (e.g. `otus.fasta`), the representative sequence fasta file of your culture isolates if you have any and you want to try to match with the OTUs (e.g. `isolates.fasta`), and the sequence reference database you want

to use, for Fungi (e.g. `sh_general_release_eukaryotes_91074_RepS_04.02.2020.fasta`, see the [Suggested Reference Databases](#) page for details). These files must end in the extensions `.fasta`, `.fa`, or `.fna`.

Your tutorial folder should look like this:



```

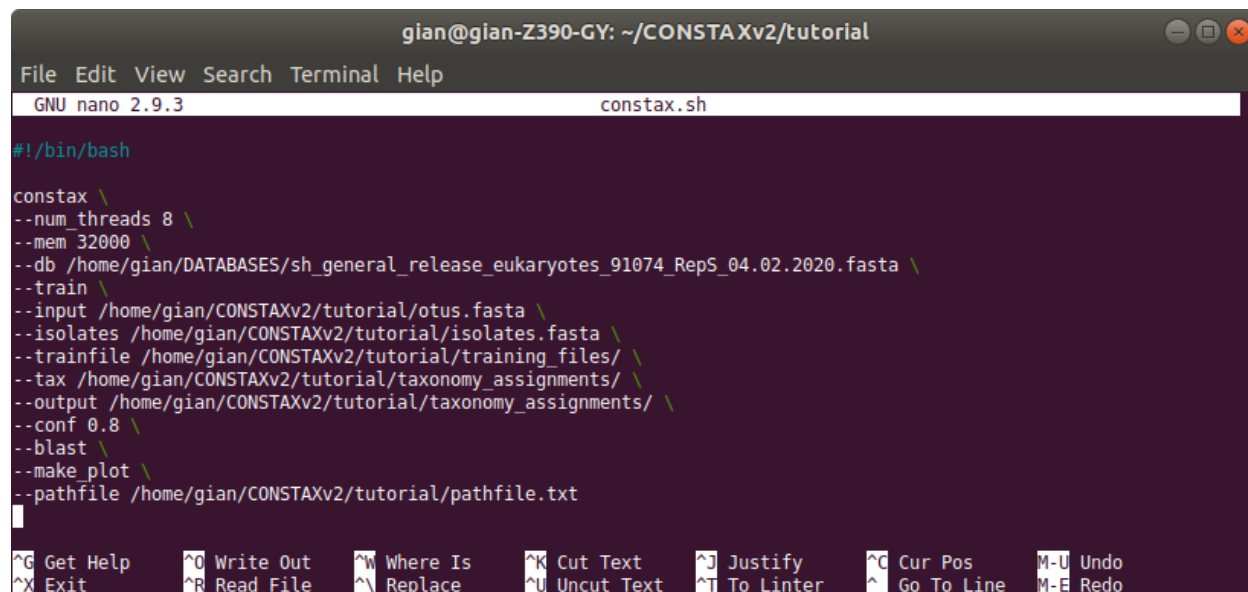
gian@gian-Z390-GY: ~/CONSTAXv2/tutorial
File Edit View Search Terminal Help
gian@gian-Z390-GY:~/CONSTAXv2/tutorial$ ll
total 164212
drwxr-xr-x  2 gian gian   4096 Dec 21 10:41 ./
drwxrwxr-x 10 gian gian   4096 Dec 21 10:35 ../
-rw-r----- 1 gian gian  71636 Dec  9 14:19 isolates.fasta
-rw-r--r--  1 gian gian   96548 Dec  9 14:04 otus.fasta
-rw-r----- 1 gian gian 167965657 Dec 10 15:03 sh_general_release_eukaryotes_91074_RepS_04.02.2020.fasta
gian@gian-Z390-GY:~/CONSTAXv2/tutorial$

```

It is smart to use the `sh` command line interpreter, so we will create a `.sh` file and write the CONSTAX commands in it.

```
gian@gian-Z390-GY:~/tutorial$ nano constax.sh
```

This is how the content of the `.sh` file should look like



```

gian@gian-Z390-GY: ~/CONSTAXv2/tutorial
File Edit View Search Terminal Help
GNU nano 2.9.3 constax.sh

#!/bin/bash

constax \
--num_threads 8 \
--mem 32000 \
--db /home/gian/DATABASES/sh_general_release_eukaryotes_91074_RepS_04.02.2020.fasta \
--train \
--input /home/gian/CONSTAXv2/tutorial/otus.fasta \
--isolates /home/gian/CONSTAXv2/tutorial/isolates.fasta \
--trainfile /home/gian/CONSTAXv2/tutorial/training_files/ \
--tax /home/gian/CONSTAXv2/tutorial/taxonomy_assignments/ \
--output /home/gian/CONSTAXv2/tutorial/taxonomy_assignments/ \
--conf 0.8 \
--blast \
--make_plot \
--pathfile /home/gian/CONSTAXv2/tutorial/pathfile.txt

```

```

constax \
--num_threads 10 \
--mem 32000 \

```

(continues on next page)

(continued from previous page)

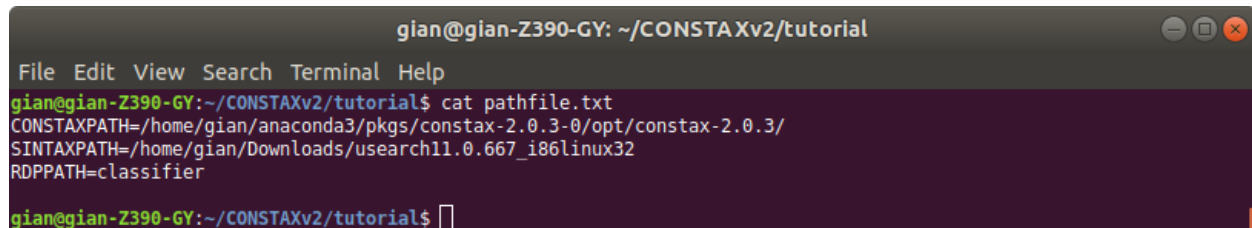
```
--db /home/gian/DATABASES/sh_general_release_eukaryotes_91074_RepS_04.02.2020.fasta \
--train \
--input /home/gian/CONSTAXv2/tutorial/otus.fasta \
--isolates /home/gian/CONSTAXv2/tutorial/isolates.fasta \
--trainfile /home/gian/CONSTAXv2/tutorial/training_files/ \
--tax /home/gian/CONSTAXv2/tutorial/taxonomy_assignments/ \
--output /home/gian/CONSTAXv2/tutorial/taxonomy_assignments/ \
--conf 0.8 \
--blast \
--make_plot \
--pathfile /home/gian/CONSTAXv2/tutorial/pathfile.txt
```

Note: Remember. If using a reference database for the first time, you will need to use the `-t` or `-l-train` flag to train the classifiers on the dataset. The training step is necessary only at first use, you can just point to the `-l-trainfile <PATH>` for the subsequent classifications with the same reference database. For SILVA please see the [Download and generate SILVA reference database](#) page for details on how to create a valid SILVA database before running CONSTAX.

The `--pathfile` option is necessary **ONLY** if you are planning to use USEARCH instead of VSEARCH for your classification. In this case we suggested to create a `pathfile.txt`

```
gian@gian-Z390-GY:~/tutorial$ nano pathfile.txt
```

where you will add the absolute PATHs for the required software. VSEARCH, BLAST, and RDP are already available through the conda environment, what you will need is just USEARCH for the SINTAX classification. The `pathfile.txt` should look like this below:



```
gian@gian-Z390-GY:~/CONSTAXv2/tutorial$ cat pathfile.txt
CONSTAXPATH=/home/gian/anaconda3/pkgs/constax-2.0.3-0/opt/constax-2.0.3/
SINTAXPATH=/home/gian/Downloads/usearch11.0.667_i86linux32
RDPATH=classifier
gian@gian-Z390-GY:~/CONSTAXv2/tutorial$
```

Warning: Remember to navigate through your anaconda installation and find the `constax-2.0.17/` folder. This is the only way to make CONSTAX locate the needed python scripts.

Before you can run CONSTAX you need to activate your anaconda environment (alternatively, you can include this in the `constax.sh` file).

```
gian@gian-Z390-GY:~/tutorial$ conda activate
```

To see how to set up a conda environment with CONSTAX please refer to [this link](#).

At this point you are ready to give CONSTAX a try.

```
gian@gian-Z390-GY:~/tutorial$ constax
```

And CONSTAX will start running...

```

gian@gian-Z390-GY: ~/CONSTAXv2/tutorial
File Edit View Search Terminal Help
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$ sh constax.sh
Welcome to CONSTAX version 2.0.3 - The CONsensus TAXonomy classifier
This software is distributed under MIT License
© Copyright 2020, Julian A. Liber, Gian M. N. Benucci & Gregory M. Bonito
https://github.com/liberjul/CONSTAXv2

(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$

```

When CONSTAX will be done you will see the outputs in the working directory.

```

gian@gian-Z390-GY: ~/CONSTAXv2/tutorial
File Edit View Search Terminal Help
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$ ll
total 164228
drwxr-xr-x  4 gian gian    4096 Dec 21 10:55 ./
drwxrwxr-x 10 gian gian    4096 Dec 21 10:35 ../
-rw-r----- 1 gian gian   1205 Dec 21 10:51 constax.sh
-rw-r----- 1 gian gian   71636 Dec  9 14:19 isolates.fasta
-rw-r--r--  1 gian gian   96548 Dec  9 14:04 otus.fasta
-rw-r--r--  1 gian gian    152 Dec 10 20:59 pathfile.txt
-rw-r----- 1 gian gian 167965657 Dec 10 15:03 sh_general_release_eukaryotes_91074_RepS_04.02.2020.fasta
drwxr-xr-x  3 gian gian    4096 Dec 10 21:07 taxonomy_assignments/
drwxr-xr-x  2 gian gian    4096 Dec 10 21:06 training_files/
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$

```

Training file and classification results will be stored in the specified folders. In this example the training files will be in `training_files`

```

gian@gian-Z390-GY: ~/CONSTAXv2/tutorial
File Edit View Search Terminal Help
drwxr-xr-x  3 gian gian    4096 Dec 10 21:07 taxonomy_assignments/
drwxr-xr-x  2 gian gian    4096 Dec 10 21:06 training_files/
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$ ll training_files/
total 1789212
drwxr-xr-x  2 gian gian    4096 Dec 10 21:06 ./
drwxr-xr-x  4 gian gian    4096 Dec 21 10:55 ../
-rw-r--r--  1 gian gian 10941604 Dec 10 21:06 bergeyTrainingTree.xml
-rw-r--r--  1 gian gian 876705280 Dec 10 21:06 genus_wordConditionalProbList.txt
-rw-r--r--  1 gian gian 1069317 Dec 10 21:06 logWordPrior.txt
-rw-r--r--  1 gian gian   281 Dec 10 21:06 rRNAClassifier.properties
-rw-r--r--  1 gian gian   20480 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_BLAST.ndb
-rw-r--r--  1 gian gian 27068166 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_BLAST.nhr
-rw-r--r--  1 gian gian 1902792 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_BLAST.nin
-rw-r--r--  1 gian gian 1902548 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_BLAST.not
-rw-r--r--  1 gian gian 26162625 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_BLAST.nsq
-rw-r--r--  1 gian gian 16384 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_BLAST.ntf
-rw-r--r--  1 gian gian 634184 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_BLAST.nton
-rw-r--r--  1 gian gian 105426485 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_RDP.fasta
-rw-r--r--  1 gian gian 16779369 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_RDP_taxonomy_headers.txt
-rw-r--r--  1 gian gian 4957706 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_RDP_taxonomy_trained.txt
-rw-r--r--  1 gian gian 15959202 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_RDP_taxonomy.txt
-rw-r--r--  1 gian gian 120607410 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_RDP_trained.fasta
-rw-r--r--  1 gian gian 124150375 Dec 10 20:59 sh_general_release_eukaryotes_91074_RepS_04.02.2020_UTAX.fasta
-rw-r--r--  1 gian gian 496829831 Dec 10 20:59 syntax.db
-rw-r--r--  1 gian gian    47 Dec 10 21:06 training_check.txt
-rw-r--r--  1 gian gian 957309 Dec 10 21:06 wordConditionalProbIndexArr.txt
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$

```

and the classification in `taxonomy_assignments`

```

gian@gian-Z390-GY: ~/CONSTAXv2/tutorial
File Edit View Search Terminal Help
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$ ll taxonomy_assignments/
total 1220
drwxr-xr-x 3 gian gian 4096 Dec 10 21:07 ./
drwxr-xr-x 4 gian gian 4096 Dec 21 10:55 ../
-rw-r--r-- 1 gian gian 267129 Dec 10 21:06 blast.out
-rw-r--r-- 1 gian gian 11380 Dec 10 21:07 Classification_Summary.txt
-rw-r--r-- 1 gian gian 86289 Dec 10 21:07 combined_taxonomy.txt
-rw-r--r-- 1 gian gian 30492 Dec 10 21:07 consensus_taxonomy.txt
drwxr-xr-x 3 gian gian 4096 Dec 10 21:07 home/
-rw-r--r-- 1 gian gian 95511 Dec 10 21:07 isolates_blast.out
-rw-r--r-- 1 gian gian 439358 Dec 10 21:06 otu_taxonomy.blast
-rw-r--r-- 1 gian gian 31704 Dec 10 21:07 otu_taxonomy_blast_final.txt
-rw-r--r-- 1 gian gian 114 Dec 10 21:07 otu_taxonomy_CountClassified.txt
-rw-r--r-- 1 gian gian 89725 Dec 10 21:07 otu_taxonomy_rdp
-rw-r--r-- 1 gian gian 32746 Dec 10 21:07 otu_taxonomy_rdp_final.txt
-rw-r--r-- 1 gian gian 93803 Dec 10 21:06 otu_taxonomy_sintax
-rw-r--r-- 1 gian gian 34279 Dec 10 21:07 otu_taxonomy_sintax_final.txt
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$

```

The taxonomic classification of your OTUs representative sequences will be in `constax_taxonomy.txt`.

```

gian@gian-Z390-GY: ~/CONSTAXv2/tutorial
File Edit View Search Terminal Help
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$ head taxonomy_assignments/consensus_taxonomy.txt
OTU_ID Kingdom Phylum Class Order Family Genus Species Isolate Isolate percent id
A31_2375258 Fungi Basidiomycota Atractiellomycetes Atractiellales Incertae sedis Helicogloea 0
A73_1655555 Fungi Basidiomycota Agaricomycetes Agaricales Agaricaceae Tulostoma 0
A86_195 Fungi 0 0 0
A104_1257856 0 0
A31_758777 0
A102_766143 Fungi Basidiomycota Agaricomycetes 0
A108_786407 Fungi Zoopagomycota 0
A61_60220 0
A124_452 Fungi Basidiomycota Tremellomycetes Cystofilobasidiales Mrakiaceae Tausonia Tausonia pullulans 0
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$

```

While classifications performed by each classifier will be store in `combined_taxonomy.txt`

```

gian@gian-Z390-GY: ~/CONSTAXv2/tutorial
File Edit View Search Terminal Help
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$ head taxonomy_assignments/combined_taxonomy.txt
OTU_ID Kingdom RDP Kingdom_BLAST Kingdom_SINTAX Kingdom_Consensus Phylum_RDP Phylum_BLAST Phylum_SINTAX Phylum_Consensus Family_RDP F
Class_RDP Class_BLAST Class_SINTAX Class_Consensus Order_RDP Order_BLAST Order_SINTAX Order_Consensus Species_RDP Species_
amily_BLAST Family_SINTAX Family_Consensus Genus_RDP Genus_BLAST Genus_SINTAX Genus_Consensus Species_RDP Species_
BLAST Species_SINTAX Species_Consensus
A31_2375258 Fungi Fungi Fungi Fungi Basidiomycota Basidiomycota Basidiomycota Basidiomycota Atractiellomycetes Atractie
llomycetes Atractiellomycetes Atractiellomycetes Atractiellales Atractiellales Atractiellales Atractiellales Incertae_sedis I
ncertae_sedis Incertae_sedis Helicogloea Helicogloea
A73_1655555 Fungi Fungi Fungi Fungi Basidiomycota Basidiomycota Basidiomycota Basidiomycota Agaricomycetes Agaricomycetes A
garicomycetes Agaricomycetes Agaricales Agaricales Agaricales Agaricales
garicaceae Tulostoma Tulostoma Tulostoma Tulostoma
A86_195 Fungi Fungi Fungi Fungi
A104_1257856
A31_758777
A102_766143 Fungi Fungi Fungi Fungi Basidiomycota Basidiomycota Basidiomycota Basidiomycota Agaricomycetes Agaricomycetes
A108_786407 Fungi Fungi Fungi Fungi Zoopagomycota Zoopagomycota Agaricomycetes Agaricomycetes
A61_60220
A124_452 Fungi Fungi Fungi Fungi Basidiomycota Basidiomycota Basidiomycota Basidiomycota Tremellomycetes TremellomycetesT
remellomycetes Tremellomycetes Cystofilobasidiales Cystofilobasidiales Cystofilobasidiales Cystofilobasidiales Mrakiaceae M
rakiaceae Mrakiaceae Mrakiaceae Tausonia Tausonia Tausonia Tausonia pullulans
ausonia pullulans
(base) gian@gian-Z390-GY:~/CONSTAXv2/tutorial$

```

Please explore other CONSTAX outputs, such as `Classification_Summary.txt`.

If you want to use some test `otus.fasta` to practice the use of CONSTAX you can find some in [THIS](#) github repo of CONSTAX.

Now. We can try to run CONSTAX again changing some parameters to see some other options. For example, modify the `constax.sh` script as showed below.

5.6 Run CONSTAX on HPCC

To run CONSTAX on the high performance cluster computer or [HPCC](#) available at Michigan State University, you can set the paths just using `--msu_hpcc` flag to your `constax.sh` file

The code will look like as below

```
benucci@dev-intel16:~/CONSTAX_v2/tutorial
File Edit View Search Terminal Help
GNU nano 2.3.1 File: constax test.sb

#!/bin/bash --login

#SBATCH --time=10:00:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=20
#SBATCH --mem=128G
#SBATCH --job-name constax_fungi
#SBATCH -A shade-cole-bonito

cd ${SLURM_SUBMIT_DIR}

conda activate py3

constax \
--num_threads $SLURM_CPUS_PER_TASK \
--mem $SLURM_MEM_PER_NODE \
--db /mnt/home/benucci/DATABASES/sh_general_release_fungi_35077_RepS_04.02.2020.fasta \
--train \
--trainfile /mnt/home/benucci/CONSTAX_v2/tutorial/training_files_fungi/ \
--input /mnt/home/benucci/CONSTAX_v2/tutorial/ITS1_soil_500_otu.fasta \
--isolates /mnt/home/benucci/CONSTAX_v2/tutorial/isolates.fasta \
--isolates_query_coverage=97 \
--isolates_percent_identity=97 \
--high_level_db /mnt/home/benucci/DATABASES/sh_general_release_fungi_35077_RepS_04.02.2020.fasta \
--high_level_query_coverage=85 \
--high_level_percent_identity=60 \
--tax /mnt/home/benucci/CONSTAX_v2/tutorial/taxonomy_assignments_fungi07/ \
--output /mnt/home/benucci/CONSTAX_v2/tutorial/taxonomy_assignments_fungi07/ \
--conf 0.7 \
--blast \
--msu_hpcc \
--make_plot

conda deactivate

scontrol show job $SLURM_JOB_ID

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit          ^J Justify      ^W Where Is    ^V Next Page    ^U UnCut Text   ^T To Spell
```

```
#!/bin/bash --login

#SBATCH --time=10:00:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=20
#SBATCH --mem=32G
#SBATCH --job-name constax_fungi
#SBATCH -A shade-cole-bonito

cd ${SLURM_SUBMIT_DIR}

conda activate py3

constax \
--num_threads $SLURM_CPUS_PER_TASK \
--mem $SLURM_MEM_PER_NODE \
```

(continues on next page)

(continued from previous page)

```

--db /mnt/home/benucci/DATABASES/sh_general_release_fungi_35077_RepS_04.02.2020.fasta_
↪\
--train \
--trainfile /mnt/home/benucci/CONSTAX_v2/tutorial/training_files_fungi/ \
--input /mnt/home/benucci/CONSTAX_v2/tutorial/ITS1_soil_500_otu.fasta \
--isolates /mnt/home/benucci/CONSTAX_v2/tutorial/isolates.fasta \
--isolates_query_coverage=97 \
--isolates_percent_identity=97 \
--high_level_db /mnt/home/benucci/DATABASES/sh_general_release_fungi_35077_RepS_04.02.
↪2020.fasta \
--high_level_query_coverage=85 \
--high_level_percent_identity=60 \
--tax /mnt/home/benucci/CONSTAX_v2/tutorial/taxonomy_assignments_fungi07/ \
--output /mnt/home/benucci/CONSTAX_v2/tutorial/taxonomy_assignments_fungi07/ \
--conf 0.7 \
--blast \
--msu_hpcc \
--make_plot

conda deactivate

scontrol show job $SLURM_JOB_ID

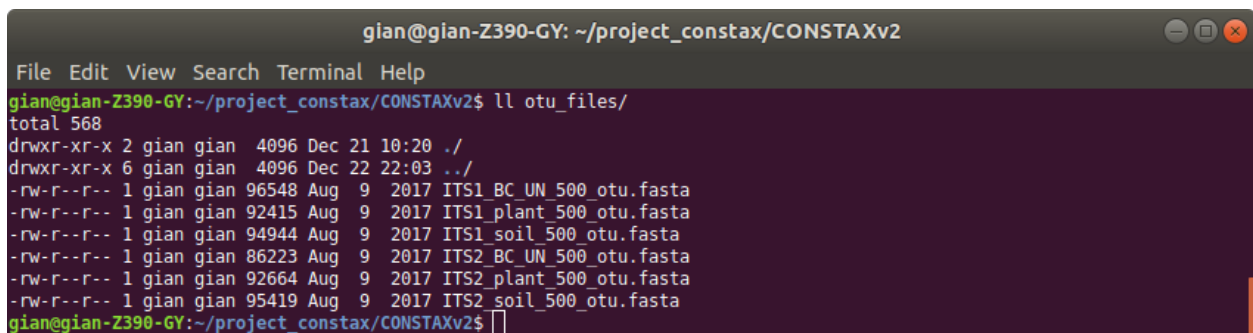
```

Note: As you can see this time `constax.sh` does not contain the `--train` option,

since the reference database has been already trained it is not required any additional training. This will improve the speed and therefore the running time will be less. The resources you need to compute just the classification are much less than those needed for training. You can then set the `num_threads` option to a lower number as well as the amount of RAM `--mem`.

Additionally no `--isolates` is provided in this run of CONSTAX and the `--hpcc_msu` is specified at the end of the script.

To access some other representative OTU sequences files please follow [THIS](#) link. These are the available files.



```

gian@gian-Z390-GY: ~/project_constax/CONSTAXv2
File Edit View Search Terminal Help
gian@gian-Z390-GY:~/project_constax/CONSTAXv2$ ll otu_files/
total 568
drwxr-xr-x 2 gian gian 4096 Dec 21 10:20 ./
drwxr-xr-x 6 gian gian 4096 Dec 22 22:03 ../
-rw-r--r-- 1 gian gian 96548 Aug 9 2017 ITS1_BC_UN_500_otu.fasta
-rw-r--r-- 1 gian gian 92415 Aug 9 2017 ITS1_plant_500_otu.fasta
-rw-r--r-- 1 gian gian 94944 Aug 9 2017 ITS1_soil_500_otu.fasta
-rw-r--r-- 1 gian gian 86223 Aug 9 2017 ITS2_BC_UN_500_otu.fasta
-rw-r--r-- 1 gian gian 92664 Aug 9 2017 ITS2_plant_500_otu.fasta
-rw-r--r-- 1 gian gian 95419 Aug 9 2017 ITS2_soil_500_otu.fasta
gian@gian-Z390-GY:~/project_constax/CONSTAXv2$

```

5.7 Download and generate SILVA reference database

This is a tutorial about how to generate a reference database, that can be used with CONSTAX. from the SILVA database that contains Bacteria and Archaea sequences.

First thing to do is to download the [SILVA reference database here](#). You should use the latest release such as 138. Go to `release_<XXX> > Exports` where `<XXX>` is the release number, and download a gzipped fasta such as `SILVA_138_SSURef_tax_silva.fasta.gz` with the name ending in `_SSURef_tax_silva.fasta.gz`.

Linux/WSL

OSX

```
wget https://www.arb-silva.de/fileadmin/silva_databases/release_138/Exports/SILVA_138_
↳SSURef_tax_silva.fasta.gz
gunzip SILVA_138_SSURef_tax_silva.fasta.gz
```

```
curl -O https://www.arb-silva.de/fileadmin/silva_databases/release_138/Exports/SILVA_
↳138_SSURef_tax_silva.fasta.gz
gunzip SILVA_138_SSURef_tax_silva.fasta.gz
```

Then, the best way is to create a script (it can be a `.sh` file or a `.sb` file depending if you are running CONSTAX locally or on the MSU HPCC) that generates the Bacteria and the Archaea `fasta` files and directly concatenate them together.

This is how the content of the `.sh` file should look like

```

benucci@dev-intel18:~/CONSTAX_v2/tutorial
File Edit View Search Terminal Help
GNU nano 2.3.1 File: constax format silva.sb

#!/bin/bash --login

#SBATCH --time=01:30:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=20
#SBATCH --mem=32G
#SBATCH --job-name format_silva
#SBACTH -A shade-cole-bonito

cd ${SLURM_SUBMIT_DIR}

conda activate py3

/mnt/research/bonito_lab/CONSTAX_May2020/constax.sh \
--num_threads $SLURM_CPUS_PER_TASK \
--mem $SLURM_MEM_PER_NODE \
-i /mnt/home/benucci/DATABASES/formatting_SILVA/SILVA_138.1_SSURef_tax_silva.fasta \
--select_by_keyword " Bacteria;" \
--msu_hpcc \
--output silvaDb_bacteria.fasta

/mnt/research/bonito_lab/CONSTAX_May2020/constax.sh \
--num_threads $SLURM_CPUS_PER_TASK \
--mem $SLURM_MEM_PER_NODE \
-i /mnt/home/benucci/DATABASES/formatting_SILVA/SILVA_138.1_SSURef_tax_silva.fasta \
--select_by_keyword " Archaea;" \
--msu_hpcc \
--output silvaDb_archaea.fasta

cat silvaDb_bacteria.fasta silvaDb_archaea.fasta > SILVA_138.1_SSURef_bact_arch.fasta

rm silvaDb_bacteria.fasta silvaDb_archaea.fasta

scontrol show job $SLURM_JOB_ID    ### write job information to output file

conda deactivate

```

You can copy and paste this code below as a guideline.

```

#!/bin/bash

constax \
-i SILVA_138_SSURef_tax_silva.fasta \
--select_by_keyword " Bacteria;" \
--output silva_Db_bacteria.fasta

constax \
-i SILVA_138_SSURef_tax_silva.fasta \
--select_by_keyword " Archaea;" \
--output silva_Db_archaea.fasta

cat silva_Db_bacteria.fasta silva_Db_archaea.fasta > SILVA_138_SSURef_bact_arch.fasta
rm silva_Db_bacteria.fasta silva_Db_archaea.fasta

```

Warning: Remember to specify the keywords correctly, as they appear in the SILVA reference. For example, to target the domain Bacteria the right keyword is " Bacteria;" with a space before the name and ";" after it.

When the scripts are finished running you can inspect the results.

```

benucci@dev-intel18:~/CONSTAX_v2/tutorial
File Edit View Search Terminal Help
[benucci@dev-intel18 tutorial]$ grep "^>" -m 10 SILVA_138.1_SSURef_bact_arch.fasta
>AY855839.1.1390 Bacteria;Proteobacteria;Alphaproteobacteria;Rickettsiales;Mitochondria;Maytenus hookeri
>FW343016.1.1511 Bacteria;Firmicutes;Bacilli;Lactobacillales;Carnobacteriaceae;Atopostipes;unidentified
>AY835431.189876.191345 Bacteria;Cyanobacteria;Cyanobacteriia;Chloroplast;Tupiella akineta
>FW369114.1.1462 Bacteria;Proteobacteria;Alphaproteobacteria;Rhizobiales;Xanthobacteraceae;Bradyrhizobium;unidentified
>FW369795.1.1413 Bacteria;Proteobacteria;Alphaproteobacteria;Acetobacterales;Acetobacteraceae;Gluconacetobacter;unidentified
>AB001440.1.1538 Bacteria;Proteobacteria;Gammaproteobacteria;Pseudomonadales;Pseudomonadaceae;Pseudomonas;Pseudomonas coronafaciens pv. atropurpurea
>HG529995.1.1433 Bacteria;Proteobacteria;Alphaproteobacteria;Rhodobacterales;Rhodobacteraceae;Rhodobacter;Rhodobacter sp. AK39
>HG529997.1.1373 Bacteria;Proteobacteria;Gammaproteobacteria;Enterobacteriales;Pseudoalteromonadaceae;Pseudoalteromonas;Pseudoalteromonas sp. AK46
>FW562653.1.1495 Bacteria;Firmicutes;Clostridia;Oscillospirales;Hungateiclostridiaceae;Thermoclostridium;unidentified
>FW555182.1.1505 Bacteria;Proteobacteria;Gammaproteobacteria;Enterobacteriales;Shewanellaceae;Shewanella;unidentified
[benucci@dev-intel18 tutorial]$

```

```
grep "^>" -m 10 SILVA_138.1_SSURef_bact_arch.fasta
```

The headers are formatted correctly and you can now use the newly created reference to classify your sequences.

5.8 Downloading the UNITE database

This tutorial is about how to obtain a reference database for classification of fungi or eukaryotes in general. These will be downloaded from [UNITE](#).

For classification of fungi, we have had tested with the [RepS 44343 General Release FASTA](#).

The eukaryote database with [96423 RepS sequences](#) provides better information about the kingdom classification of the sequence, but requires slightly more RAM (~40GB). Using the `--high_level_taxonomy` option can provide a similar result but with reduced RAM requirements.

```

curl https://files.plutof.ut.ee/public/orig/E7/28/
↪E728E2CAB797C90A01CD271118F574B8B7D0DAEAB7E81193EB89A2AC769A0896.gz > sh_general_
↪release_04.02.2020.tar.gz
tar -xzf sh_general_release_04.02.2020.tar.gz

```

Use the FASTA called `sh_general_release_fungi_35077_RepS_04.02.2020.fasta` within the expanded directory for your fungal reference database, specified with `-d` or `--db` in your `constax` command.

For the `--high_level_db` option, the eukaryotes database found here <https://plutof.ut.ee/#/doi/10.1515/BIO/1280127> can be used. This will help to remove non-fungal OTUs from your dataset, or can be used as the main database (`-d`, `--db`) for projects amplifying other eukaryotes.

5.9 Examine SH (Species Hypothesis) hits from UNITE database

This tutorial is about how to examine poorly classified fungal OTUS by comparing to SHs from the UNITE database, which often don't have species names associated with them but are consistent taxa which could be of interest to the user.

This will require a [downloaded UNITE database](#).

You can do this two separate ways:

1. Use the same database for both `-d/--db` and for `--isolates`.

```

constax \
-i otus.fasta \
-b \
-t \
-d sh_general_release_fungi_35077_RepS_04.02.2020.fasta \
--isolates sh_general_release_fungi_35077_RepS_04.02.2020.fasta

```

The accessions found in the `constax_taxonomy.txt` file in the output directory is searchable at the [UNITE search page](#).

2. Examine the `blast.out` file in the directory specified by `-x/--tax` or the default `./taxonomy_assignments` directory.

```
# BLASTN 2.10.0+
# Query: OTU_1
# Database: /mnt/ufs18/rs-022/bonito_lab/CONSTAX_May2020/UNITE_Fungi_tf/sh_
↳general_release_fungi_35077_RepS_04.02.2020__BLAST
# Fields: query acc., subject acc., evalue, bit score, % identity, % query_
↳coverage per subject
# 5 hits found
OTU_1    KC306753      1.04e-96      351      99.482  100
OTU_1    AF377107     2.25e-93      340      98.446  100
OTU_1    AF377107     2.25e-93      340      98.446  100
OTU_1    KC306757     8.16e-88      322      96.891  100
OTU_1    KC306757     8.16e-88      322      96.891  100
```

The second column is an accession number that can be searched at the [UNITE search page](#).

5.10 Help

If you need help please open a ticket in the [CONSTAX repo](#) on Github.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`