SUMAC: Supermatrix Constructor version 2.0 Manual

William A. Freyman[1]

Department of Integrative Biology, University of California, Berkeley

[1]freyman@berkeley.edu

Contents

Chapter 1

Introduction

1.1 About SUMAC

SUMAC (Supermatrix Constructor) is a Python package to data-mine GenBank, construct phylogenetic supermatrices, and assess the phylogenetic decisiveness of a matrix given the pattern of missing sequence data. It is designed to be run as a command-line program, though the modules can also be imported and used in other Python scripts. SUMAC will assemble supermatrices for any taxonomic group recognized in GenBank, and is optimized to run on multicore systems by utilizing multiple parallel processes.

SUMAC version 2.0 is *significantly* faster than earlier versions due to utilizing the UCLUST clustering algorithm. However, the default values of sequence identity (set with the `-id` flag) should be experimented with. Different values will be optimal depending on the sequence divergence of your group of interest.

When run from the command-line, SUMAC will perform a number of steps to construct a supermatrix. First, SUMAC creates a local SQLite3 (Hipp and Kennedy, 2007) database of the specified GenBank division (e.g. PLN or MAM), automatically downloading sequences from NCBI if necessary. Using NCBI taxonomy, SUMAC searches the local database for all sequences in the user-specified ingroup and outgroup. Found sequences are then clustered as putative homologs in one of two ways: (1) using the UCLUST (Edgar, 2010) algorithm or performing exhaustive all-by-all BLASTn (Camacho et al., 2009) comparisons of each ingroup and outgroup sequence and using a single-linkage hierarchical clustering algorithm, or (2) user-provided guide sequences that typify each cluster are BLASTed against all ingroup and outgroup sequences. For

more details about the clustering algorithms used please see Freyman (2015). Once clustering is complete, SUMAC discards clusters that are not phylogenetically informative (< 4 taxa), and aligns each cluster of sequences using MAFFT (Katoh et al., 2002) with the `--adjustdirection` flag to ensure correct sequence polarity. The individual locus alignments are saved to enable gene tree inference, and then the alignments are concatenated by species binomial (based on the NCBI taxonomy) to create the final supermatrix. Finally, a number of metrics are reported, a graph indicating taxon coverage density is generated, and spreadsheets (in CSV format) are produced with information about each DNA region and GenBank accession used in the supermatrix. There are many options described in detail later in this manual.

1.2 Citing the Program

If you use SUMAC for a published paper please cite the following release note:

Freyman, W.A. 2015. SUMAC: constructing phylogenetic supermatrices and assessing partially decisive taxon coverage. *Evolutionary Bioinformatics* 2015:11 263-266.

1.3 Installation

SUMAC runs on Mac OSX and Linux. MS Windows is not supported and won't work without fiddling.

1.3.1 Dependencies

The following dependencies must be installed to run SUMAC:

- `Python 2.7`
- `Biopython`
- `MAFFT v6.9+`
- `USEARCH` and/or `BLAST+`

BLAST+ only needs to be installed if you plan to use functionality that requires it like guide sequence searches. For clustering, USEARCH's UCLUST algorithm is recommended over all-

by-all BLAST comparisons because it is orders of magnitude faster. `matplotlib` should also be installed if you want SUMAC to generate graphs.

1.3.2 Installing Dependencies on Linux

The following commands install the requirements for Debian GNU/Ubuntu Linux systems:

```
sudo pip install numpy
sudo pip install matplotlib
sudo pip install biopython
sudo apt-get install mafft
```

You must also either install USEARCH (recommended) and/or BLAST+:

```
http://drive5.com/usearch/
sudo apt-get install ncbi-blast+
```

1.3.3 Installing Requirements on Mac

The following commands install the requirements of Mac OS X:

```
sudo pip install numpy
sudo pip install matplotlib
sudo pip install biopython
```

You must also either install USEARCH (recommended) and/or BLAST+. Install USEARCH here:

```
http://drive5.com/usearch/
```

or install BLAST by downloading the appropriate executable from:

```
ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/
```

Install MAFFT from here:

```
http://mafft.cbrc.jp/alignment/software/
```

1.3.4 Installing SUMAC

To install SUMAC use this command:

```
sudo pip install sumac
```

If you prefer to use the development version in the Git repository use these commands:

```
git clone https://github.com/wf8/sumac.git
cd sumac/src
python setup.py install
```

1.4 License and Warranty

SUMAC is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

The program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details (http://www.gnu.org/copyleft/gpl.html).

Chapter 2

Quick Start Tutorial

This chapter provides the quick way to get started using SUMAC. There are many details described in Chapter 3 that would likely be helpful.

2.1 Construct a Supermatrix

The most basic usage of SUMAC is to build a supermatrix with the following command:

```
python -m sumac -d pln -i Onagraceae -o Lythraceae
```

This command is an example of the minimum amount of input required to run SUMAC. The first part of the command `python -m sumac` runs the SUMAC module. The `-d pln` tells SUMAC to download the PLN (Plant) GenBank division and is only necessary the first time one runs SUMAC. If you run SUMAC a second time with the `-d` flag you will download GenBank again, overwriting the previously downloaded version of GenBank. The `-i Onagraceae` and `-o Lythraceae` tells SUMAC to search the PLN division for all sequences within the taxonomic groups Onagraceae (the ingroup) and Lythraceae (the outgroup). Specifying an outgroup (the `-o` flag) is not necessary, but maybe useful when the user wants to combine sequences from two different taxonomic groups. SUMAC will then build clusters of putatively homologous sequences, and construct a supermatrix.

Unless you are on a large multi-core system, the clustering will take a long time to be performed (for the example above) since well over 5000 sequences will be found. To speed up the supermatrix construction, you could make a FASTA file of guide sequences to define each clus-

ter. Each guide sequence should be an example of a sequence commonly used for phylogenetic analysis. Then you could then use this command:

```
python -m sumac -d pln -i Onagraceae -o Lythraceae -g guides.fasta
```

Which approach is better for constructing supermatrices? Using guide sequences makes supermatrix construction much faster, however it requires a priori knowledge of which DNA regions will be used in the supermatrix. Clustering all sequences is computationally more expensive, but it effectively data-mines GenBank in an exploratory fashion, so that sequence data not necessarily used in previous systematic studies can also be incorporated into the supermatrix. The decision will depend on the size of the taxonomic group being analyzed and the computational resources available.

To calculate the partial decisiveness (PD) and Missing Sequence Decisiveness Scores (MSDS) for the data matrix, run SUMAC with the `--decisiveness` flag. See Freyman (2015) for an explanation of these metrics. Please be aware that the decisiveness computations can be time consuming.

2.2 Explanation of the Output Files

SUMAC will output the following files:

1. `alignments/supermatrix_concatenated.fasta` The final aligned and concatenated supermatrix in FASTA format.

2. `alignments/supermatrix_N.fasta` The alignment of gene region N where ($N = 1, 2, 3...$) as it was used in the final supermatrix. The FASTA description contains only the species binomial. These files are useful for gene tree inference.

3. `alignments/N.fasta` The alignment of gene region N, but with the full GeBank descriptions included in the FASTA descriptions.

4. `clusters/N.fasta` The unaligned raw sequence cluster of gene region N.

5. `gb_search_results` File used by SUMAC to save the results of the GenBank sequence search in case the search is re-run. This file is not human readable.

6. `genbank_accessions.csv` A spreadsheet with each GenBank accession used, ordered by gene region and taxon (useful for generating the appendices found in most systematics papers).

7. `gene_regions.csv` A table with the number of taxa, the aligned length, the percent missing data, and the taxon coverage density of each gene region used in the supermatrix.

8. `missing_sequence_decisiveness.csv` A spreadsheet that contains the MSDS scores for all missing sequences. This is useful for prioritizing which sequences to add to the matrix to increase the overall decisiveness. Only generated if SUMAC is run with the `--decisiveness` flag.

9. `sequence_decisiveness_scores_plot.pdf` A graph that shows the MSDS scores for the final supermatrix. Only generated if SUMAC is run with the `--decisiveness` flag.

10. `sequence_data_plot.pdf` A figure that shows how much sequence data was available for each taxon for each gene region.

11. `sumac_log` Log of the SUMAC run that contains details about the supermatrix construction.

Chapter 3

SUMAC in Detail

3.1 Downloading GenBank

3.1.1 GenBank Division

The first time you run SUMAC you must specify which GenBank division to download with the `-d div` option, where `div` is the GenBank designated three letter code of the division (PLN, MAM, etc). Once SUMAC has downloaded the GenBank division, future SUMAC runs should omit the `-d div` option to avoid repeatedly downloading GenBank. If the ingroup and outgroup are in different GenBank divisions, use the `-d2` flag to specify a second GenBank division to download.

3.1.2 GenBank File Path

By default, each SUMAC run searches for the downloaded GenBank files in `./genbank/`, a subdirectory of the current run's directory. It may be useful to save the GenBank files outside of the current working directory, in which case you can specifiy the absolute path of the GenBank files with the `-p path` option. For example, if you want to build multiple supermatrices (or different versions of the same one) each in a different working directory it is helpful to use `-p /genbank` so that all SUMAC runs use the same copy of the GenBank files.

3.2 Specifying Ingroup and Outgroup

The `-i` flag must be used to specify which ingroup to search for. Including an outgroup (the `-o` flag) is not necessary, but maybe useful when the user wants to combine sequences from two different taxonomic groups. The taxonomic names must be those used by GenBank, and the searches for the names are case sensitive. If a SUMAC run is repeated with the same ingroup and outgroup, SUMAC will load the previous search results to save time.

3.3 Setting the Number of CPU Cores

By default, SUMAC will use the maximum number of CPU cores available on the system. You may limit the number of cores used with the `-c` flag.

3.4 Using Guide Sequences

Guide sequences should be in a single standard FASTA file specified using the `-g` option. The names of the guide sequences will be ignored, and each of the ingroup and outgroup sequences will be BLASTed against the guide sequences.

3.5 Homologous Sequence Thresholds

3.5.1 UCLUST ID value

By default, SUMAC uses a threshold default UCLUST ID 0.50. This can be changed with the `-id` option. Different values will be optimal depending on the sequence divergence within your group of interest. This setting will be ignored if using the SLINK or HAC clustering algorithms.

3.5.2 BLAST E-value

By default, SUMAC uses a threshold default BLASTn e-value $1.0e-10$. This can be changed with the `-e` option.

3.5.3 Sequence Length Similarity

SUMAC uses a default threshold of sequence length percent similarity of 0.25. This can be changed with the `-l` option.

3.5.4 Minimum and Maximum Sequence Lengths

By default, SUMAC excludes any sequences less than 100 bp and greater than 5000 bp in length. This is to exclude sequences that maybe fragments or that maybe whole chloroplast or mitochondrial genomes. These defaults can be changed with the `-minl` and `-maxl` options.

3.6 Building Supermatrix from User Edited Alignments

Often the user will want to edit the alignments used by SUMAC and then regenerate the supermatrix (reconcatenating the alignments and recalculating the various metrics and spreadsheets). The user may want to add their own sequences to the GenBank sequences, make taxonomic changes to the GenBank sequences, or remove erroneous sequences. To do this the user should edit the `alignments/N.fasta` files and then rerun SUMAC with the `-sa` flag with the path of the edited alignments as an argument. For example, if the edited alignments are in the `edited_alignments` directory, the user should run the command `python -m sumac -sa edited_alignments/*`. The `*` is a wildcard that specifies all files within the directory.

3.7 Partial Decisiveness

To calculate the partial decisiveness (PD) and Missing Sequence Decisiveness Scores (MSDS) for the data matrix, run SUMAC with the `--decisiveness` flag. See Freyman (2015) for an explanation of these metrics. An example of the plot of MSDS scores that SUMAC generates is given below.

3.8 Clustering Algorithms

Two hierarchical clustering algorithms are implemented in SUMAC. By default, SUMAC clusters sequences using UCLUST (Edgar, 2010), which is by far the fasted clustering method. If run with the command line flag `--SLINK`, the SLINK single-linkage hierarchical clustering algorithm (Sibson, 1973) will be used. If run with the command line flag `--hac`, SUMAC will instead cluster sequences using Sneath's naive hierarchical agglomerative clustering (HAC) algorithm (Sneath, 1957). See Freyman (2015) for more details on their implementation.

3.9 Example Supermatrix Figures

SUMAC generates two PDF plots of the supermatrix. The file `sequence_data_plot.pdf` is a plot of the sequence data coverage (Figure 3.1). If run with the `-de` flag, SUMAC will generate `sequence_decisiveness_scores_plot.pdf`, which is a plot of Missing Sequence Decisiveness Scores (MSDS) (Figure 3.2).

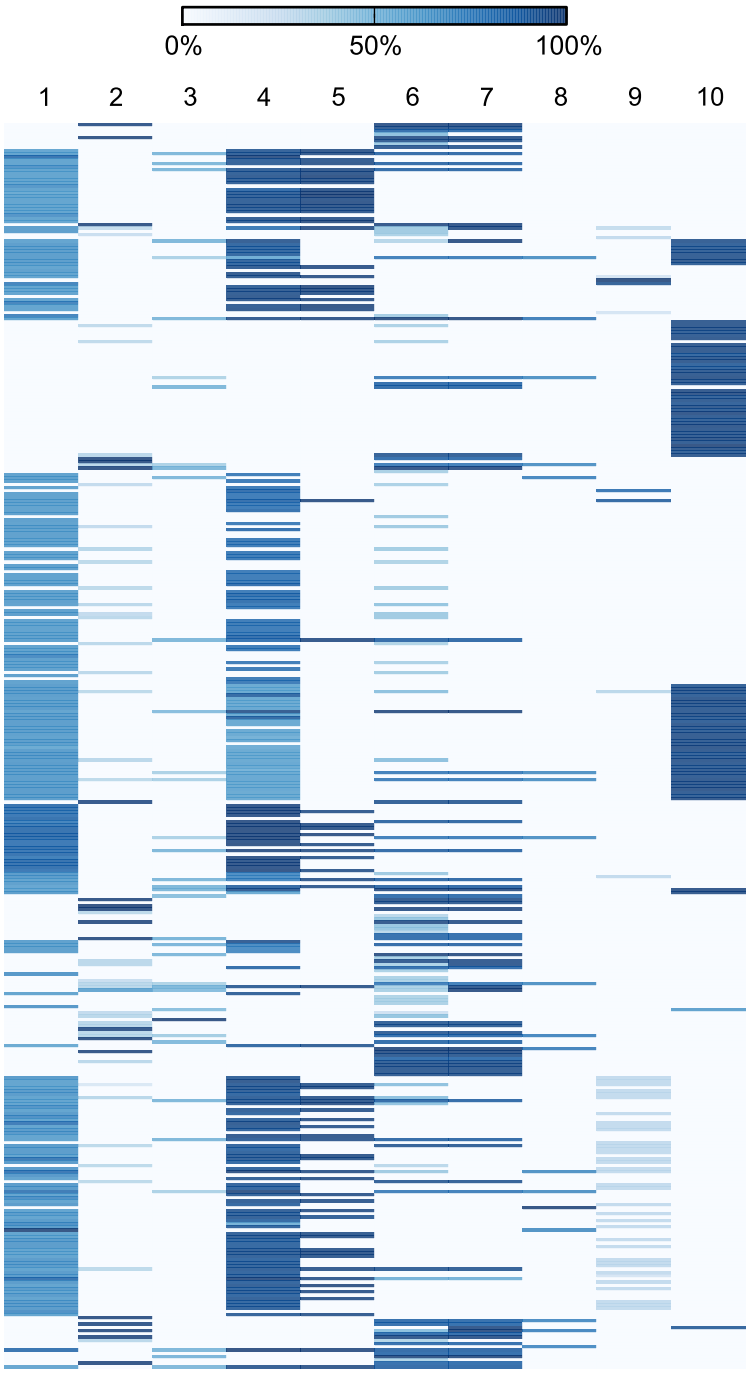

Figure 3.1: Data coverage for a sequence matrix with 10 genes, 384 OTUs, taxon coverage density of 0.26, and partial decisiveness of 0.31. White represents missing data, and shades of blue represent sequences present. Longer sequences are dark blue, short sequences are light blue.

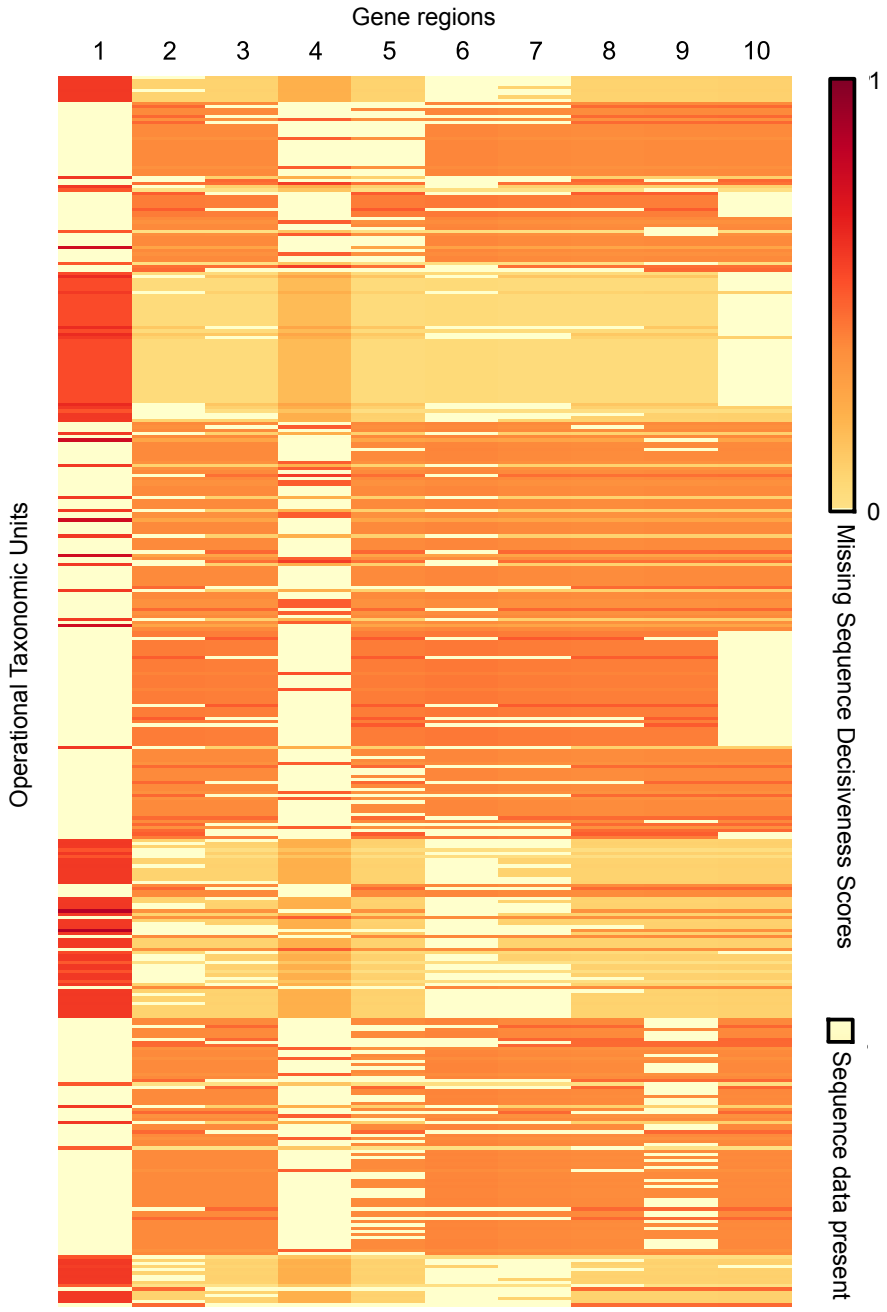

Figure 3.2: Missing Sequence Decisiveness Scores (MSDS) for a sequence matrix with 10 genes, 384 OTUs, taxon coverage density of 0.26, and partial decisiveness of 0.31. Pale yellow represents sequence data present, shades of orange represent missing sequences with low to intermediate MSDS (0-0.75), and red to maroon represents missing sequences with high MSDS (0.75-1.0). MSDS measures how much the individual missing sequence contributes to the decisiveness of the matrix given the overall pattern of missing data. MSDS prioritizes which sequences to add to the matrix; where MSDS is high the addition of new data will increase the decisiveness of the matrix more than where MSDS is low.

Bibliography

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T. L. (2009). BLAST+: architecture and applications. *BMC Bioinformatics*, 10(1):421.

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461.

Freyman, W. (2015). SUMAC: software for constructing phylogenetic supermatrices and assessing partially decisive taxon coverage. *Evolutionary Bioinformatics*, 11:263–266.

Hipp, D. R. and Kennedy, D. (2007). Sqlite. http://www.sqlite.org.

Katoh, K., Misawa, K., Kuma, K.-i., and Miyata, T. (2002). Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic acids research*, 30(14):3059–3066.

Sibson, R. (1973). SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34.

Sneath, P. H. A. (1957). The Application of Computers to Taxonomy. *Journal of General Microbiology*, 17(1):201–226.