Symtyper Documentation

Release 0.1

August 13, 2014

Contents:

CHAPTER

ONE

WEB-BASED SYMTYPER

1.1 Brief Overview

To run SymTyper from the web, follow these instructions:

First, invoke symtyper main submission page (http://www.symtyper.com). You should be presented with the following page in *SymTyper's Main Screen*.

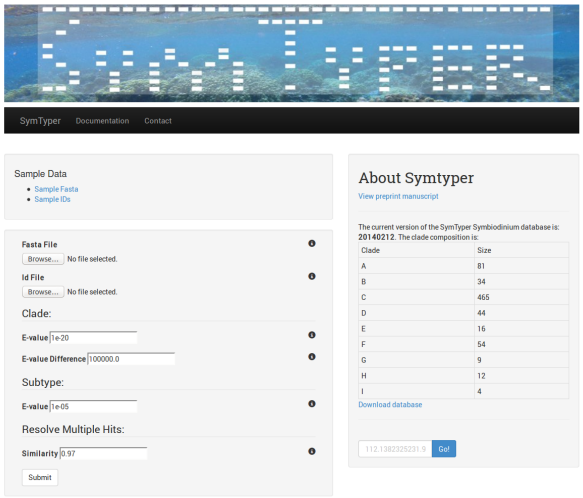

Figure 1.1: SymTyper's Main Screen

To submit a new analysis, browse and select your input fasta file and a valid ids file (*Fasta Input Format*) and then click submit (See *New SymTyper Analysis Screen*).

The next screen will provide you with the URL where the output can be accessed. Depending on the input size, the processing can take between few minutes to hours (See *Processing Screen and Job URL*). Please copy the URL for future access. Job will be hosted on the SymTyper site for 15 days.

If the anlysis completed successfully, you will be presented with the a summary table where the various componenents of the analysis can be accessed (*SymTyper Results Main Screen*). The results are gouped by section: Clades, Subtypes, Multiples, Trees, Breakdown. These sections are explained below.

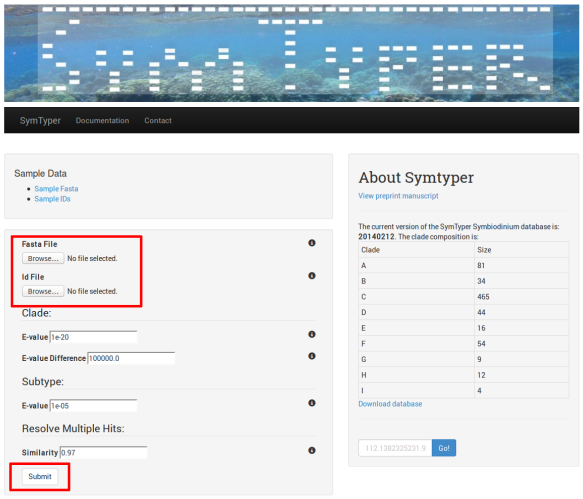

Figure 1.2: New SymTyper Analysis Screen

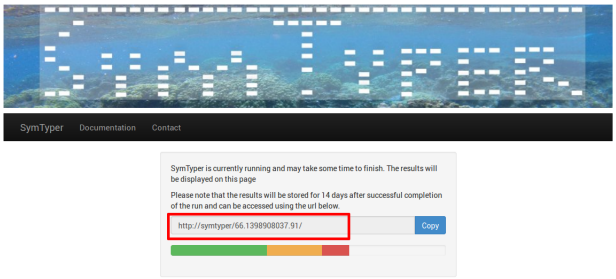

Figure 1.3: Processing Screen and Job URL

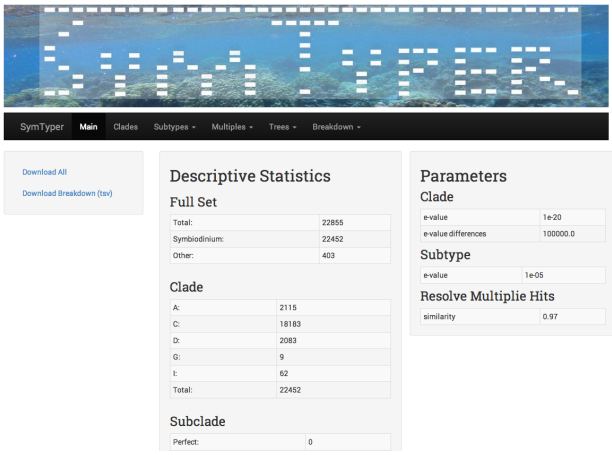

Figure 1.4: SymTyper Results Main Screen

Section	Description
Clades	Shows the breakdown of clades per sample. The results can be viewed or download as a matrix or show as a piechart per sample
Subtypes	Shows the breakdowns of subtypes per sample. The results can be viewed independently for the *Perfect*, *Unique* and *ShortNew* subtypes
Multiples	The graphs shows the distribution of sequences for each clade contaning multiple hits. The definition of a Multiple hits is described in the *Multiples* section
Trees	Describes the breakdown of the number of sequences assigned to the internal nodes, and the clades per sample. The tree representation show the combined count for all the samples
Breakdown	Shows a Sunburst representation of the Clades and subtypes by sample

1.1.1 Clades View

The Clades View shows a table view of the distribution of HITS, NOHITS, LOW and AMBIGUOUS hits per sample. Clicking the View Chart provides access to the clades distribution for each sample. The complete results and disbtribution of clades per sample can be downloaded from the results main page (see *Pie Chart Distribution of Clade per Sample*).

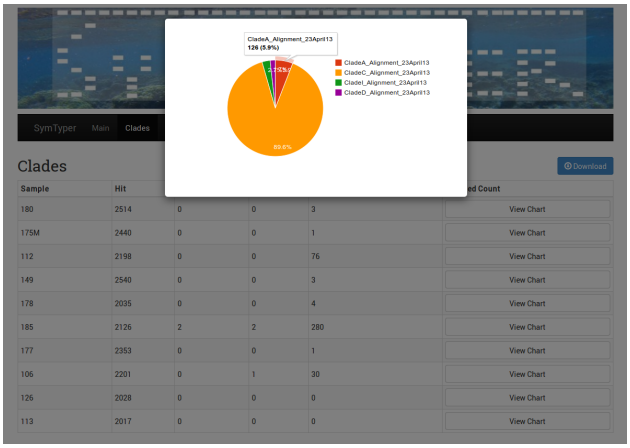

Figure 1.5: Pie Chart Distribution of Clade per Sample

1.1.2 Subtypes View

The Subtypes Views shows the breakdown of subtypes per sample. The results can be viewed independently for the *Perfect*, *Unique* and the *ShortNew* subtypes. The subtypes are assigned based on the blast results of the query sequences to the clade specific references (See *Subtypes Distribution per Clade*).

Section	Description
Perfect	A query sequence that aligns perfectly or with very high similarity to a unique symbiont reference in the database (e.g., 100% similarity to 100% of the length of the target)
Unique	A query sequence that aligns unambiguously to symbiont reference in the database. (e.g., >= user defined % similarity to 100% target length and the bit score for the best hit is at least 3 orders or magnitude larger than than that for the second hit);
ShortNew	A query sequence shorter than the average sequence in the reference database but aligns with high similarity to a unique reference according to the dynamic similarity threshold (See *Dynamic Similarity*)

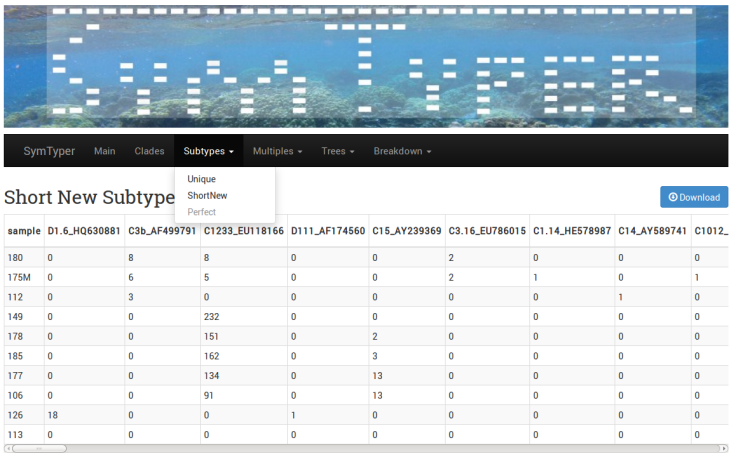

Figure 1.6: Subtypes Distribution per Clade

1.1.3 Multiples View

The Multiples View is a graphical representation of the corrected subtypes count to which ambiguous sequences map. The algortihm used to resolved multiple hits is described in the *Ambiguous Hit Correction* and detailed in the manuscript (See *Subtypes Distribution for the Corrected Ambiguous Hits*).

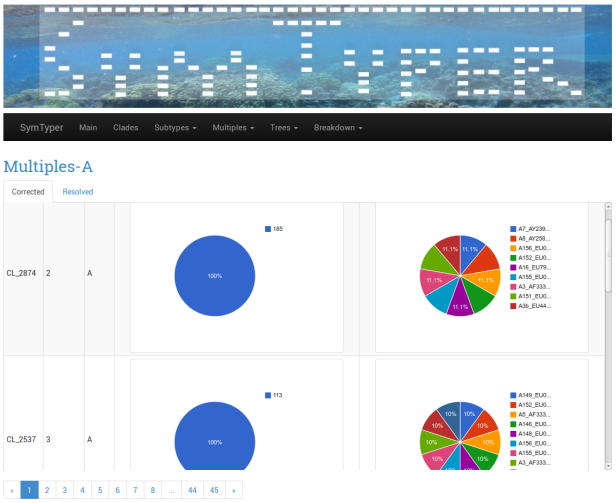

Figure 1.7: Subtypes Distribution for the Corrected Ambiguous Hits

The breakedown of subtypes for *Resolved* under the "Resolved tab"

1.1.4 Trees View

For each clade phylogeny, this view compiles the number of times a *Lowest Common Ancestor* was identified for an ambiguous sequence (after the *Ambiguous Hit Correction* stage). The tree can be downloaded in the Newick format and viewed or parsed in phylogeny applications. A matrix file comparing results across samples can be be found in output archive available for download from the main page.

1.1.5 Breakdown View

Using user-friendly graphical Sunburst representation, this view summarizes the intricate structure of Symbiodinium clades and subtypes in a single or multi-sample view. Highlighting a level of the Sunburst charts display its structure and the percentage of sample reads assigned to it (See *Subtype Breakdown Vizualization*).

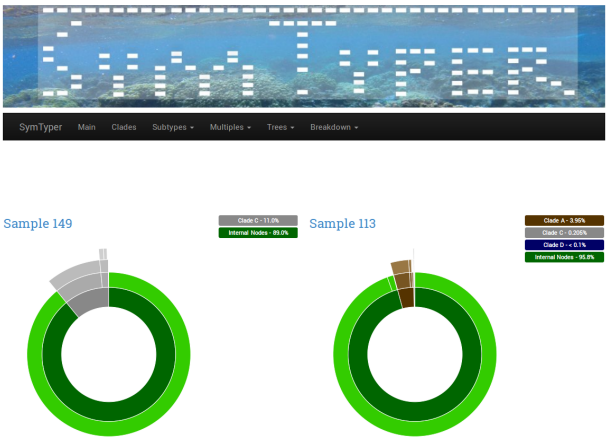

Figure 1.8: Subtype Breakdown Vizualization

CHAPTER

TWO

COMMAND LINE SYMTYPER

2.1 Installing Symtyper's requirements

The most recent version of Symtyper is a self-contained Python script that can be run without being explicitly installed on the system. However, Symtyper depends on other applciation for its exectution. These applications are

Application	Version	Notes
HMMER	>=3.0	http://selab.janelia.org/software/hmmer3/
Blast	>=2.2.25	. Currently, only legacy Blast is supported
cd-hit	>= 4.5.6	
biopython	>= 1.61	
ete2	>= 2.2	
xvfb		Required for executing Symtyper on a remote server via ssh

2.2 Running Symtyper sub-program

Symtyper is comprised of 5 subprograms that each carry out a specific funciton. These programs are: *clade*, subtype, resolveMultipleHits, builPlacementTree and makeTSV. The details for each program is described below.

2.2.1 clade

usage

usage: symTyper.py clade [-h] -s SAMPLESFILE -i INFILE [-e EVALUE]

Input

REQUIRED

Param	Description
-i, –inFile	File containing the sequencing reads in fasta fomat. Note that this files requires the ids to be fomatted using the following *Fasta Input Format*
-s, –samplesFile	The *Samples File*

Ouptut

fasta/: Directory cotaining a collection of fasta sequences representing the intput fasta file split by sample **hmmer_output/**: Directory containing HMMER output files, broken down by sample **hmmer_parsedOutput/**: Directory containing listing of *AMBIGUOUS OUTPUT*, *HITS OUTPUT*, *NOHITS OUTPUT* and *LOWOUT* for each of the input samples **hmmer_hits/**: Directory containing fasta files, split by clade, of sequences having hits against the clade database.

2.2.2 subtype

usage: symTyper.py subtype [-h] -s SAMPLESFILE -H HITSDIR -b BLASTOUTDIR -r BLASTRESULTS -f FASTAFILESDIR Directory contains HMMER output files, broken down by sample

Input

The input to "clade" is expected to be in the same format as that produced by the *clade* subprogram

Param	Description
-f, –fastaFilesDir	Directory cotaining sequences from the input fasta file, split by *clade* (fasta directory)
-s, –samplesFile	The *Samples File*
-H, –hitsDir	HMMER fasta hits output directory produced by *clade* (hmmer_hits directory)
-b, –blastOutDir	Blast output directory
-r, –blastResults	Parsed blast results directory

Output

blast_output/: Directory containing Blast output files, broken down by sample **blastResults/**: Directory containing informaiton on the *Perfect*, *Unique*, *New*, *ShortNew* and *Short*

The output formats for the files in blastResults/ can be found here:

PERFECT OUTPUT UNIQUE OUTPUT NEWOUT SHORTNEW OUTPUT SHORT OUTPUT

2.2.3 resolveMultipleHits

Input

usage: symTyper.py resolveMultipleHits [-h] -s SAMPLESFILE -m MULTIPLEFASTADIR -c CLUSTERSDIR

The input to **resolveMultipleHits** is expected to be in the same format as that produced by the *subtype* subprogram

Param	Description
-s, –samplesFile	The *Samples File*
-m, –multipleFastaDir	Directory containing sequences with multiple hits, split by *clade* (x directory)
-c, –clustersDir	Directory that will contain cluster information

Output

resolveMultiples/Reps: Representatives from each cluster **resolveMultiples/clusters**: Clusters produced for each sample **resolveMultiples/correctedMultiplesHits**: Contains output files from clustering and multiple hit resolution

The resolveMultiples/correctedMultiplesHits directory contains the following files and directory:

- correctedOutputFile_all_clades: *Corrected Output All Clade*
- resolvedOutputFile_all_clades: *Resolved Output All Clades*
- corrected/: Contains *Corrected Output Per Clade*, split by *clade*

2.2.4 builPlacementTree

usage: symTyper.py builPlacementTree [-h] -c CORRECTEDRESULTSDIR -n NEWICKFILESDIR -o OUTPUTDIR

Input

The input to **builPlacementTree** is expected to be in the same format as that produced by the *resolveMultipleHits* subprogram

Param	Description
-c, –correctedResultsDir	Directory containing corrected Clade placements (the correctedMultiplesHits/corrected directory from resolveMultipleHits)
-n, –newickFilesDir	Newick directory for input calde phylogenies in Newick format
-o, –outputDir	Dir that will contain the newick and interenal nodes information

Output

The output directory containing the placement information for, broken down by sample

2.2.5 makeTSV

Input

Files and directories produced by the *builPlacementTree* subprogram

Output

CHAPTER

THREE

SYMTYPER'S CONCEPTS

3.1 Definitions

3.1.1 HIT

This is a clade-relevant definition. To be a HIT against a clade reference sequence, a query needs to unambiguously align with a defined similarity over a defined percentage of its length. Furthermore, the e-value of the first hit needs to be at least K orders of magnitude larger than that of an alternative clade.

3.1.2 NOHIT

This is a clade-relevant definition. A Sequence is considered a NOHIT if it does not have any satisfactory alignments against a clade.

3.1.3 AMBIGUOUS

This is a clade-relevant definition. An ambiguous sequence is one that has more than one satisfctory clade hit.

3.1.4 Perfect

This is a subtype-relevant definition. Perfect refers to a query sequence that aligns unambiguously to one sequence in the reference database (e.g., 100% similarity to 100% of the length of the target) for which the best hit's raw bit score is at least 3 orders of magnitude larger than the raw bit score for the second hit.

3.1.5 Unique

This is a subtype-relevant definition. Unique refers to a query sequence that aligns to a single reference in the database with a user-defined (e.g., >= user defined % similarity to 100% target length) for which the best hit's raw bit score is at least 3 orders of magnitude larger than the raw bit score for the second hit.

3.1.6 New

This is a subtype-relevant definition. A New subtype applies to a sequnence with no significant hit to any of the subtype database sequences.

3.1.7 ShortNew

This is a subtype-relevant definition. ShortNew refers to a query sequence that aligns with high similarity to a unique reference sequence according to the dynamic similarity threshold (Equation 1: *Dynamic Similarity*) below.

3.1.8 Multiples

This is a subtype-relevant definition. A query sequence of type multiple is a sequence that aligns with equal similarity to multiple subtypes sequences.

3.1.9 Short

This is a subtype-relevant definition. A query of type short, is one that does not meet the minimum similarity and length requirements (e.g., $< 90\%$ similarity to $< 90\%$ of the length of the target).

3.1.10 Dynamic Similarity

The dynamic similarity threshold is computed to allow query sequences that are shorter than the database references to be considered as potential hits. However, the shorter the sequnces, the higher the required stringency. The dynamic similarity threshold is computed as:

$$required\_similarity = 100 - \frac{C - min_c}{1 - min_c} * (100 - min_s)$$

where:

C	is the coverage fraction of the query over the hit sequences
min_c	is the minimum accepted coverage fraction of the query and the hit sequences
min_s	is the minimum similarity threshold between the query and the hit sequences

3.1.11 Ambiguous Hit Correction

An ambiguous hit occurs when a sequences aligns with multiple subtypes. To try to infer the correct subtype of the sequence, we employ a strategy similar to the wisdom of the crowd, and allow similar sequences to help contribute information about the closest subtype of the sequence. To do so, ambiguous sequences are clustered using high stringency and a subtype distribution (or spectrum) is computed for each cluster.

Suppose a cluster has a distribution: 88 C1.1, 45 C1.18, 6 C1.21 and 2 C1.28. This means that at least 88 sequences in the cluster were subtyped as C1.1. and only 1 was subtyped as C1.28.

Clusters' distributions are usually highly skewed with few high frequency subtypes and a greater number of low frequency types. Since there distributions are subsequently used to infer the *Lowest Common Ancestor* (LCA) sequence as a proxy, it is very improtant to rid the data of unlikely subtype that can bias the computation of the LCA. For the previous distribution, the wisdom of the crowd tells us that this cluster of sequences is closest to C1.1. and unlikely to be C1.28 and therfore drops it for the C1.28. The same can be said about C1.21 since only 6 sequences have been aligned to it. The corrected distribution is thus likely 88 C1.1, 45 C1.18. This distribution will be subsequently used to map the reads to the common ancestor in the phylogeny.

The algoirthm used to correct the subtypes distribution uses a similar approach by formalizing which subtypes to drop for the distribution using a strigency parameter p. To do so, we iteratively drop the the subtypes that have counts within the p^{th} percentile of the distribution and stop when no subtypes can be dropped.

3.1.12 Resolved

An ambiguous read is said to be resolved if its filtered distribution after the *Ambiguous Hit Correction* contains a single subtype.

3.1.13 Lowest Common Ancestor

In a phylogenetic tree, an internal node, N, is the lowest common ancestor (or most recent common ancestor) of a set of leaves L, if N is the first common parent of all the leaves of in L

3.1.14 Placement Tree

A phylogeny of the subtypes in each clade where an internal node can be labeled using the number of seqeuencing reads for which is considered to be the most recent ancestor

3.1.15 TSV Format

A file with tab delimited columns

3.1.16 Samples File

A file cotaining the samples – one per line – in the dataset.

3.2 Input File Formats

3.2.1 Fasta Input Format

Sequence ids in the fasta file are required to have the following format.

Sample_ID::Seq_Number

- **Sample_ID**: refers to the sample to which the sequence belongs. The sampleID should be present in the *Samples File*
- **Seq_Number**: is a unique identifier for a the sequence.

Note that the two colons (**::**) are used to separate the Sample_ID and the Seq_Number.

3.3 Clade Output Format

3.3.1 HITS OUTPUT

- Query sequence id
- Hit start in query
- Hit end in query
- First hit id
- Second hit id

- First hit e-value
- Second hit e-value

3.3.2 NOHITS OUTPUT

- Query sequence id

3.3.3 AMBIGUOUS OUTPUT

- Query sequence id
- First hit id
- Second hit id
- First hit e-value
- Second hit e-value

3.3.4 LOWOUT

- Query sequence id
- First hit id
- Hit e-value

3.3.5 MULTIPLE OUTPUT

- Query sequence id
- List of hits ids

3.4 Subtype Output Formats

3.4.1 NEWOUT

- Query sequence id

3.4.2 PERFECT OUTPUT

- Query sequence id
- Best hit id
- Query length / Hit length
- Percent identity

3.4.3 SHORT OUTPUT

- Query sequence id
- Query length
- Best hit id
- Best hit lenght

3.4.4 SHORTNEW OUTPUT

- Query sequence id
- Best hit id
- Query length / Hit length
- Percent identity

3.4.5 UNIQUE OUTPUT

- Query sequence id
- Best hit id

3.5 ResolveMultipleHits Output Formats

3.5.1 Corrected Output All Clade

Tab separated fields and colon separated values. Ex.

```
Cluster:  CL_415 numSeq:  6 clade:  C breakDown:180:4 175M:2 subtypes:
C3.24_HE579012:  6, C3k_AY589737:  6, C3.23_HE579011:  6
```

The previous line tell us that CL_145 representes 6 Sequences, 2 form sample 175M and 4 from sample 180. These sequences are in Clade C and have the subtype distribution listed in *subtype* list.

3.5.2 Resolved Output All Clades

- Cluster ID
- Number of sequences in the cluster
- Clade
- Subtype of sequences in the cluster

3.5.3 Corrected Output Per Clade

This file format is similar to that in *Corrected Output All Clade* except that the *subtype* list represents the corrected (or effective), rather than initial, subtypes.