Making Supertrees Superfast

-

Technical Specification Document

UNSW BINF6112 20T3

Caitlin Ramsay, Jade Liang

Weilin Wu, Ying Xu & Kavitha Krishna

Technical documentation

Supertrees superfast (stsf) is a reference-free phylogenomics workflow for short-read data generated from Illumina sequencing. All steps are being connected together using the **Snakemake** workflow management system, which enables reproducible and scalable data analyses.

Product locates on github at *https://github.com/due23/supertrees_BINF6112_2020*
Please contact *z5206348@student.unsw.edu.au* for access. You will need the following libraries/packages to run the program

Resources and Package requires:

1. ABySS genome assembler for short paired-end reads
   github.com/bcgsc/abyss
   ABySS 2.2.0

2. Benchmarking Universal Single-Copy Orthologs (BUSCO) to assess genome assembly completeness
   github.com/openpaul/busco
   busco/4.1.0 and dependencies
   busco/3.0.2b and dependencies for plotting BUSCO result

3. BUSCOMP: BUSCO Compiler and Comparison tool to extract best genes
   github.com/slimsuite/buscomp
   minimap2/2.17

   * Please contact the Edward's lab for BUSCOMP access

4. Muscle for multiple sequence alignment
   github.com/cran/muscle
   muscle/3.8.31

5. IQ-TREE phylogenomic software using maximum likelihood to make gene trees
   github.com/Cibiv/IQ-TREE
   iqtree/2.0.4

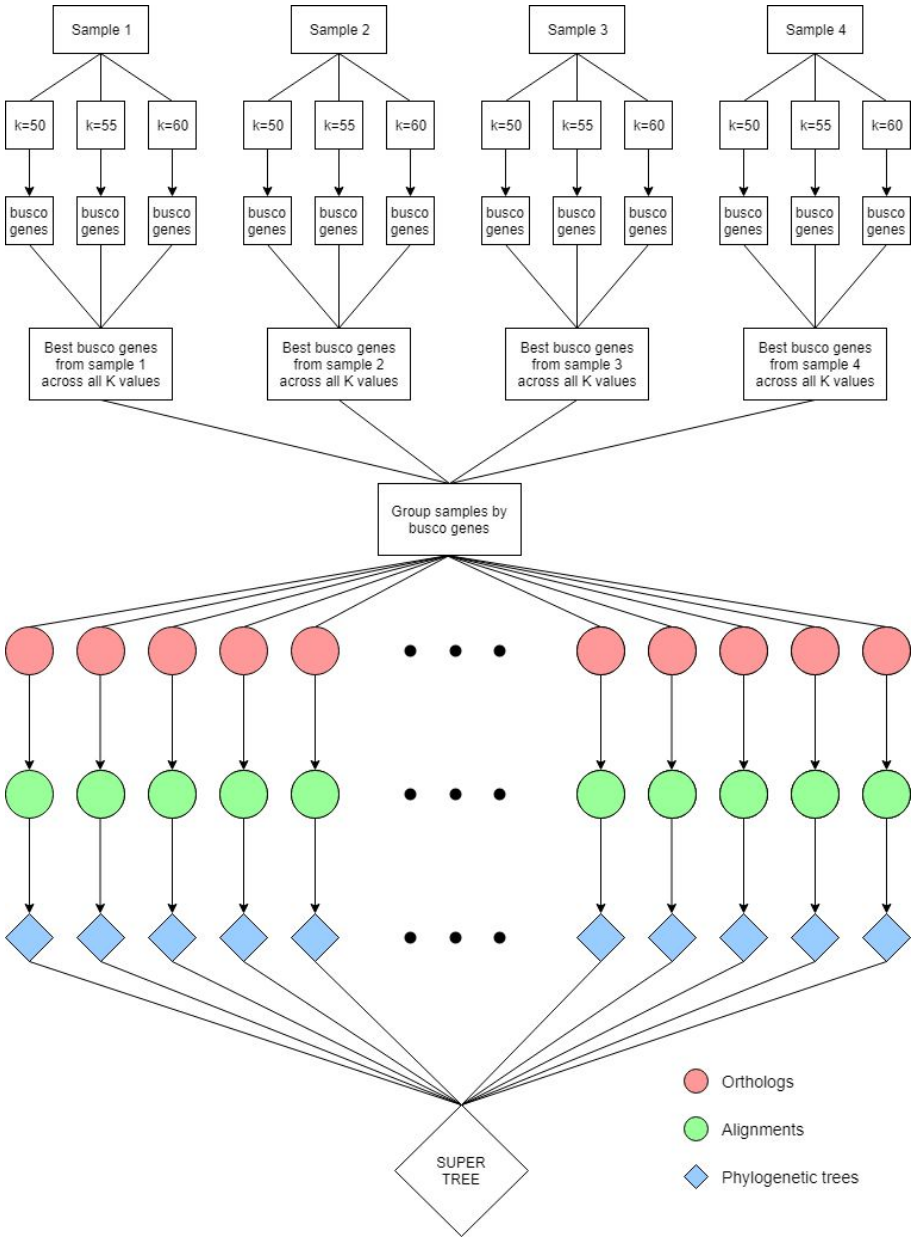
6. ASTRAL for supertrees (species trees)
   github.com/smirarab/ASTRAL
   *Please download the tool from the page provided above.

7. Snakemake workflow management system
   https://snakemake.readthedocs.io/en/stable/
   python/3.8.3

User documentation

The Snakemake Workflow

The pipeline takes in Illumina short-read data across different samples and assembles multiple genomes according to a range of k-values. Each of these genomes is then assessed for quality and completeness before the best BUSCO genes are extracted. These BUSCO genes across all the k assembled genomes are grouped by a BUSCO gene ID into read files and then multiple sequence alignment is performed on them. The multiple sequence alignments for each BUSCO gene ortholog set is used to generate multiple gene trees which are then merged together to create a final supertree. Many parts of the workflow can be run in parallel to speed up the runtime of the pipeline. A summary of the workflow and details on how tasks are run is shown in the diagram below.



Configure the Snakemake Workflow

The user is required to construct a file called "**snakemake.config**". This should contain the AbySS, BUSCO and workflow configuration. It enables the users to specify the:

- Sample names (these are the names that will be used throughout and will be displayed in the gene trees and final supertree)
- Path of all input files, both single and paired end data ------> [line 11 "read: "]. It is recommended to specify the absolute file paths
- Range of k values and increment -------> [line 9, "k: "]
- BUSCO configuration
- Lineage selection

A sample "**snakemake.config**" is provided in github.

Execution of the Snakemake workflow

Run the following command to set up the working environment:
`$ source setup.sh`

The setup script will load the correct version of python required and the abyss module.

***Once the environment is set up, user can run the snakemake workflow as the following:**

Sample parameter:

`--cores N` : snakemake can automatically determine which parts of the workflow can be run in parallel. By specifying the number of available cores. The number of cores that is specified is equivalent to the maximum number of tasks that can be run in parallel.

`--until / -U` : Runs the pipeline until it reaches the specified rules or files.

`-n:` Only dry run the whole workflow to display all the tasks. Nothing will be run with this.

`-p:` To print all the commands from all programs to the terminal as the workflow is running.

Command Line:

To only dry-run the workflow:
`$ snakemake -n`

To run the whole workflow to generate the super tree:
N: snakemake can automatically determine which parts of the workflow can be run in parallel. By specifying the number of available cores. The number of cores that is specified is equivalent to the maximum number of tasks that can be run in parallel.

`$snakemake --cores N`

To run sections of the Snakemake workflow:

If user is running the whole workflow from assembly to generating final supertree
`$ snakemake --cores N`

If user only wants to run assembly and BUSCO
`$ snakemake -U assess_all --cores N`

If user wants to run the workflow up to BUSCOMP
`$ snakemake -U extract_all --cores N`

If user wants to run the workflow up to MUSCLE Alignment
`$ snakemake -U align_all --cores N`

If user wants to run the workflow up to individual tree (IQ Tree)
`$ snakemake -U tree_all --cores N`

To plot the BUSCO results and assessment of genome assembly

`$ snakemake busco_figure.png --cores 1`

In the Snakefile, there is a rule BUSCO_plot which will plot the BUSCO assessment results. This rule will make use of busco_plot.config.ini file.

Outputs and results

The output after each program will be stored in separate directories. The directories are:

- BUSCO: stores the BUSCO outputs and is organised by samples and k values
- BUSCOMP: stores the BUSCOMP outputs and is organised by samples. BUSCOMP results contain a fasta file with all the BUSCO genomes summarised from all the k values.
- genomes: stores all the assemblies for all k values
- msa: contains all the alignments between all genes
- trees: contains all the individual gene trees
- final.tree: this is the file that contains the final supertree. It will contain all the samples (with the sample names given in snakemake.config).

Clean Up

If the user wants to remove all intermediate result files, run the following command:

```
$ ./cleanup.sh
```