ANANAS *DE-NOVO* META-TRANSCRIPTOME ASSEMBLY

The following guidelines bear upon the analyses performed in the Ananas paper. By following the outlined instructions, you will have a methodological framework for implementing your own analysis available. In addition to that, you will be able to replicate the analyses performed in the Ananas paper.

REAL DATASET

For this section, you can download **data for testing** here.

Generating Ananas *de-novo* assembly

1) Run:

```
Ananas -i <file1_r.fastq,file1_l.fastq> -o <output_directory> \
-dir <read_orientation> -n <cores>
```

Ananas can take multiple input files (fastq or fasta, which is automatically detected) when separated by a comma and no blanks between the file names. The fasta or fastq files can be gzipped.

Output: all assembled transcripts will be in the output directory in final.fa. Info about read ids and coordinates in all the assembled contigs will be in the output directory in final.layout.

2) Run:

```
GetTopFromFasta -if final.fa -il final.layout
```

Output: the top assembled transcripts (without isoforms) will be in the output directory in final.fa.top. Info about read ids and coordinates in all the top assembled contigs will be in the output directory in final.layout.top.

Taxonomic annotation

The software Diamond could be used to perform a standard taxonomic annotation of the assembled transcripts.

For more information and detailed method description, see diamond.

SIMULATED DATASET

This section outlines how the analysis of the simulated data described in the Ananas paper was performed.

For this section, you can download **data for testing** here.

Source of RNA-seq data

Download paired-end RNA-seq read sets (fasta format) from the NCBI SRA repository using sratoolkit (Table x). Each file includes RNA-seq data from a single species; the file name should be changed into the species name, specifying the specific set of reads (either left or right).
Example: If you download the files named SRR1234_1.fasta and SRR1234_2.fasta, including data from the "Abcde" species, you should name the obtained files Abcde_1.fasta and Abcde_2.fasta.

Processing the data into a working dataset

At this point you could choose to continue with either the complete datasets or subsets of them. In the first case you will be using the datasets as they are after their download and with their name modified as described above, whereas in the latter case you will create subsets from a selected number of datasets, which include a fixed number of random reads extracted from them.

A) Complete datasets: run the script **1A_FormatFiles_CompleteDatasets.sh**

Run: /my_dir/1A_FormatFiles_CompleteDatasets.sh \
/my_dir/my_input_fasta_directory/my_filename \
/my_dir/my_input_fasta_directory/ \
/my_dir/my_output_fasta_directory/

Example: /Users/Matteo/my_Pipeline/1A_FormatFiles_CompleteDatasets.sh \
/Users/Matteo/my_InputFasta/my_fasta_files \
/Users/Matteo/my_InputFasta/ \
/Users/Matteo/my_OutputFasta/

- Create a file (my_filename) in which you specify the names of the fasta files you want to process. The file names should be WITHOUT set of reads (_1/_2) and WITHOUT extension (.fasta) (for example: Abcde, Abcde2, etc.), and each in a different line. Don't forget to leave an empty line at the end of the list file. **This file should be in the same directory as the fasta files**.
- The name of the processed species (names of the fasta files and names in my_filename) will be used to correctly define the header of each sequence in the output fasta files (i.e. Abcde_1_4Ananas.fasta and Abcde_2_4Ananas.fasta)

Output: **Abcde_1_4Ananas.fasta** & **Abcde_2_4Ananas.fasta** located in the /my_dir/my_output_fasta_directory/

B) Subsets: run the script **1B_FormatFiles_Subsets.sh**

Run: /my_dir/1B_FormatFiles_Subsets.sh \
/my_dir/my_input_fasta_directory/my_filename \
/my_dir/my_input_fasta_directory/ \
/my_dir/my_output_fasta_directory/ \
/my_dir/my_scripts_directory/ \
nr_of_random_reads§

§*nr. of random reads I want to extract for each member of the pair. If I specify here 15,000 reads, this corresponds to 30,000 (15,000x2) reads in total.*

Example: /Users/Matteo/my_Pipeline/1B_FormatFiles_Subsets.sh \
/Users/Matteo/my_InputFasta/my_fasta_files \
/Users/Matteo/my_InputFasta/ \
/Users/Matteo/my_OutputFasta/ \
/Users/Matteo/my_Scripts/ \
15000

- Create a file (my_filename) in which you specify the names of the fasta files you want to process. The file names should be WITHOUT set of reads (_1/_2) and WITHOUT extension (.fasta) (for example: Abcde, Fghil, etc.), and each in a different line. Don't forget to leave an empty line at the end of the list file. **This file should be in the same directory as the fasta files**.
- The name of the processed species (names of the fasta files and names in my_filename) will be used to correctly define the header of each sequence in the output fasta files (i.e. Abcde_1_4Ananas.fasta and Abcde_2_4Ananas.fasta)
- After these pre-processing steps, the **Extract_ReadSubset_fromFasta.pl** perl script will automatically extract a number of random reads you decide (last command line variable, above-shown as §). The **Extract_ReadSubset_fromFasta.pl** perl script should be located in the script directory (/my_dir/my_script_directory/).

Output: **RandomSubset_1_4Ananas.fasta** & **RandomSubset_1_4Ananas.fasta** located in /my_dir/my_output_fasta_directory/

Generating Ananas *de-novo* assembly

Run the script **2_AnanasPipeline.sh**, regardless what dataset you are working with at this stage (either complete datasets or subsets).

N.B. In this step you need R locally installed.

Run: /my_dir/2_AnanasPipeline.sh \
/my_dir/my_scripts_directory/ \
/my_dir/my_output_fasta_directory/ \
/my_dir/Ananas_assembler_directory/ \
/my_dir/Ananas_output_directory/ \
Bp_overlap_value* Reads_direction* n_parameter* n2_parameter* no_parameter*

*OPTIONAL
*If the optional parameters are not specified, default values will be run (35 as Bp_overlap_value, fr as Reads_direction, 1 as n_parameter, 1 as n2_parameter, 2 as no_parameter).

Example: /Users/Matteo/my_Pipeline/2_AnanasPipeline.sh \
/Users/Matteo/my_Scripts/ \
/Users/Matteo/my_OutputFasta/ \
/Users/Matteo/Ananas_Assembler/ \
/Users/Matteo/Ananas_Results/

- In my_output_fasta_directory there should be the fasta files coming from the previous steps (either complete datasets or subsets).
- After the assembly, the **Ananas_dowstream.pl** perl script will automatically run downstream commands, in order to get a suitable file in which some statistics will be calculated and plots generated. The **Ananas_dowstream.pl** perl script should be located in the script directory (/my_dir/my_script_directory/).

Output: Ananas final.fa.top and final.layout.top will be located in /my_dir/Ananas_output_directory/ together with a file showing the accuracy of the analysis (accuracy.txt), one showing how many of the produced contigs have the maximum ratio value equal to 1 (max_ratio_eq1.txt), one showing how many contigs are assembled in total (max_ratio_all.txt). In the same directory you can also find R plots showing various statistics and relevant correlations:

- histogram_contig_length_log.pdf
- corr_contig_length_max_ratio_diff_spec.pdf
- corr_coverage_max_ratio_diff_spec.pdf
- corr_contig_length_coverage_diff_spec.pdf

N.B. To dowload and run Trinity, see here.
To run the benchmarking Trinity pipeline, see here.

Generating Ananas statistics

Run the script **3_Statistics_Ananas.sh**, regardless what dataset you are working with at this stage (either complete datasets or subsets).

N.B. In this step you need the software BedTools2.

Run:
```
/my_dir/3_Statistics_Ananas.sh \
/my_dir/my_filename HowManySpecies \
/my_dir/Reference_genomes_files_directory/ \
/my_dir/Ananas_output_directory/ \
/my_dir/Ananas_statistics_directory/ \
/my_dir/my_scripts_directory/ \
/my_dir/BedTools2_directory/
```

Example:
```
/Users/Matteo/my_Pipeline/3_Statistics_Ananas.sh \
/Users/Matteo/Reference_Genomes/my_fasta_files \
TwoSpecies \
/Users/Matteo/Reference_Genomes/ \
/Users/Matteo/Ananas_Results/ \
/Users/Matteo/Ananas_Statistics/ \
/Users/Matteo/my_Scripts/ \
/Users/Matteo/BedTools2_Folder/
```

- create a file (my_filename) in which you specify the names of the species included in the Ananas assembly. Each name should be in a different line. Don't forget to leave an empty line at the end of the list file (same concept as the previous pipeline steps). **This file should be in the Reference files directory.**
- HowManySpecies should specify how many species are included in the analysis (for example TwoSpecies, ThreeSpecies, FourSpecies, etc).
- The Reference_genomes_files_directory should contain:

   1. Reference genome file for each species included in the analysis. Every reference genome should be named Abcde_ReferenceGenome (same species name as in the my_filename). N.B: Replace the name of the assembly (generally a number in the header) with the name of the species.
   2. Gff file for the corresponding reference genome. For each reference genome of the included species, you should download the corresponding gff file. Every gff file should be named Abcde_gff.txt (same species name as in the my_filename).

- The reference genomes and the corresponding gff files can be downloaded from NCBI RefSeq.
- The Ananas_output_directory is the directory where the Ananas assembly results obtained in the previous step are located.
- In order to generate the assembly statistics, the following scripts should be located in the script directory (/my_dir/my_script_directory/): **Ananas_GetContigsLength.pl**,

Get_ContigPercentBpOverlappingWithGff.pl,
Statistics_BlastDowstream.pl,
Statistics_BlastGetBpGaps.pl.

Output: All output files will be located in /my_dir/Ananas_Statistics_directory/. The output files are the following:
Ananas_number_of_contigs: number of contigs in the assembly.
Ananas_TotalBpAssembled: total number of bp assembled by Ananas.
Abcde_Ananas_BpGaps: how many bp gaps are present in the assembly of that specific species compared to the reference genome.
Abcde_Ananas_BpGenesOverlap: how many bp in total in the assembly of that specific species overlap with the gff gene annotation coordinates.
Abcde_Ananas_TotalBpMatch: how many bp in the assembly of that specific species match with the concatenated reference genome including all analyzed species after blasting Ananas contigs back to it.
Abcde_Ananas_BpIntersectionWithGenesGff: for each annotated gene, how many bp in the assembly of that specific species overlap with every gff gene coordinates.

N.B. The commands included in the 3_Statistics_Ananas.sh scripts are compatible with Trinity, and the user might run them as Trinity downstream processing (see here)