

Workflow for *The Drosophila Dicer-2 partner Loquacious-PD is specific to processing of hairpin- and 3' cis-NAT-derived siRNAs*

Setup

Input files

Raw read libraries for each dataset should be stored in their directory under **Input/Libraries**.

Due to size constraints, With the exception of the Genome fasta file, General Feature Format (GFF) file and RepeatMasker output, all required input files for Bowtie Index generation are provided in **Input/Downloads**. Refer to Methods for more details on the required input files.

Config files

A single config file should be setup as **Configfiles/main_config.yaml**, which contains essential config parameters for the workflow. These parameters are used in the various Snakefiles and should be modified when appropriate. Further documentation can be found within the file itself.

Helper Scripts and Figures

All helper functions are stored in **Scripts/Python/helpers.py**. Further documentation can be found within the file itself.

All plots are created by the R scripts stored in **Scripts/R**.

Both Figure-1 and Figure-2 are stitched together using **patchwork.R**, after creating the subplots using **Main_figures.R**, **Main_figures2.R**, **hpRNA_figure.R**, **hpRNA_figure2.R**.

Main_figures.R and **Main_figures2.R** are identical and are used to create the bar plots for counts across the various small RNA features. They should be run separately with the required parameters to generate the sub-plots for both *GSE37443Head* and *GSE37443Ovary*.

Similarly **hpRNA_figure.R** and **hpRNA_figure2.R** are identical and are used to create the bar plots for hpRNA features that are in Figure-1 and Figure-2.

TE_figure.R generates bar plots for Transposable Element features and **Read_position.R** generates the read position plots.

Usage

Detailed below is the order in which the Snakefiles are run. '' act as placeholders for required config parameters for each run of the Snakefile. More details on individual config parameters can be found within each Snakefile.

1. Generate Bowtie indexes
`snakemake -s Index_generation.Snakefile`
2. Pre-process libraries
`snakemake -s Okamuralib_preprocessing.Snakefile`

`snakemake -s Externallib_preprocessing.Snakefile --config dataset=''`
3. Run Sequential Mapping and gather counts for each Bowtie mapping
`snakemake -s Okamuralib_Sequential_mapping.Snakefile --config dataset='' runID='' lengthfilter = '' minlength = '' maxlength= ''`

`snakemake -s GSE37443_Sequential_mapping.Snakefile --config dataset='' runID='' lengthfilter = '' minlength = '' maxlength= ''`
4. Run Multi-mapped Sequential Mapping
`snakemake -s Multi_sequential_mapping.Snakefile --config dataset='' runID='' minlength = '' maxlength= ''`
5. Extract counts for each feature present in target bowtie output
`snakemake -s Feature_counts.Snakefile --config dataset='' runID='' lengthfilter='' minlength='' maxlength=''`
6. Extract read position output for each feature present in target bowtie output
`snakemake -s Read_position.Snakefile --config dataset='' runID='' lengthfilter ='' minlength = '' maxlength= ''`

Support