mlPEA User Manual

(version 1.0)

- mlPEA is a user-friendly, full-functionality pipeline specifically designed to enhance the processing, analysis, and interpretation of m6A-Seq in non-model plants by leveraging machine learning.
- mlPEA comprises four functional modules: **Data Preprocessing, Transcriptome Construction, m^{6}A Calling, and Functional Exploration**.
- mlPEA was powered with an advanced packaging technology, which enables compatibility and portability.
- mlPEA project is hosted on http://github.com/cma2015/mlPEA
- mlPEA docker image is available at http://hub.docker.com/r/malab/mlpea

Transcriptome Construction Module

This module provides step-by-step functions required for transcriptome construction.

Transcriptome Assembly

Several commonly used trascriptome assemblers are wrapped to use. Currently, **Trinity**, **rnaSPAdes**,**TransABySS**,**TransLiG**.

Tools	Description	Input	Output	Time (test data)	Reference
Trinity	Trinity partitions the sequence data into many individual de Bruijn graphs, each representing the transcriptional complexity at a given gene or locus, and then processes each graph independently to extract full-length splicing isoforms and to tease apart transcripts derived from paralogous genes.	Sequencing reads in FASTQ format	Transcripts in FASTA format	~5 mins	(Grabherr *et al.*, 2011)
rnaSPAdes	SPAdes is primarily developed for Illumina sequencing data, but can be used for IonTorrent as well. Most of SPAdes pipelines support hybrid mode, i.e. allow using long reads (PacBio and Oxford Nanopore) as a supplementary data.	Sequencing reads in FASTQ format	Transcripts in FASTA format	~5 mins	(Prjibelski *et al.*, 2020)
TransABySS	a *de novo* short-read transcriptome assembly and analysis pipeline that addresses variation in local read densities by assembling read substrings with varying stringencies and then merging the resulting contigs before analysis.	Sequencing reads in FASTQ format	Transcripts in FASTA format	~5 mins	(Robertson *et al.*, 2010)
TransLiG	TransLiG is shown to be significantly superior to all the salient de novo assemblers in both accuracy and computing resources when tested on artificial and real RNA-seq data.	Sequencing reads in FASTQ format	Transcripts in FASTA format	~5 mins	(Liu *et al.*, 2019)

Transcriptome Screening

Transcriptome screening implements three pipelines for transcripts deduplication, respectively.

Tools	Description	Input	Output	Time (test data)	Reference
Merge and Deduplication	a novel program CD-HIT (Fu *et al*., 2012) for clustering biological sequences to reduce sequence redundancy and improve the performance of other sequence analyses.	multiple Transcripts files in FASTA format	Transcripts in FASTA format	~5 mins	(Fu *et al*., 2012)
Reads Coverage Screening	an ultrafast and memory-efficient tool **Bowtie 2** (Langmead *et al*., 2009; Langmead *et al*., 2012) for aligning sequencing reads, reads coverage was measured by **SAMtools** (Danecek *et al*., 2021) and **BEDTools** (Quinlan *et al*., 2010).	Transcripts in FASTA format	Transcripts in FASTA format	~10 mins	(Langmead *et al*., 2009; Langmead *et al*., 2012) (Danecek *et al*., 2021) (Quinlan *et al*., 2010)
Machine Learning-based Screening	MMSeq2 (Steinegger *et al*., 2017) to cluster the assembled transfrags with both mRNA and ncRNA sequences from the Ensembl Plants across all plant species. Both assembled tranfrags and annotated plant sequences to serve as positive sample set. Then we used the corain package (Wang *et al*., 2023) to encode for RNA from three perspectives, sequence, structure and physical chemical properties. Adopted the improved PSoL(IPSol) algorithm (Wang *et al*., 2006), the initial negative samples were generated from "unlabeled" sample set based on the PU bagging algorithm, the expanded negative samples iteratively using RF classifier until the designated iteration number was reached.	Transcripts in FASTA format	Transcripts in FASTA format	~20mins	(Steinegger *et al*., 2017) (Wang *et al*., 2023) (Wang *et al*., 2006)

Transcriptome Assembly

Currently, mlPEA wrapped four assemblers to *de nove* assemble transcripts, here, we take Trinity as an example to show how to use mlPEA to run transcriptome assembly, the other three assemblers are similar.

Input

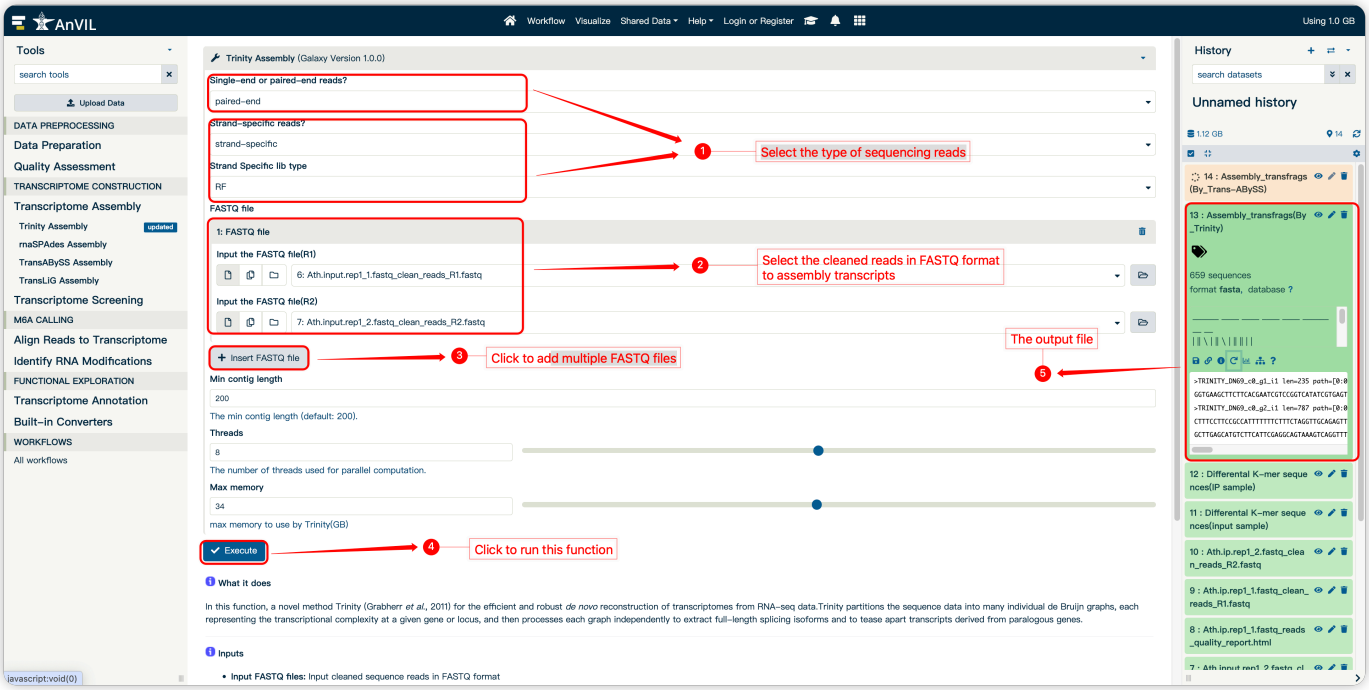
- **Input FASTQ files:** Input cleaned sequence reads in FASTQ format

Output

- **de novo transcriptome assembly by Trinity in FASTA format**

How to use this function

- The following screenshot shows us how to use this function



Merge and Deduplication

In this function, a novel program CD-HIT (Fu *et al*., 2012) for clustering biological sequences to reduce sequence redundancy and improve the performance of other sequence analyses.

Input

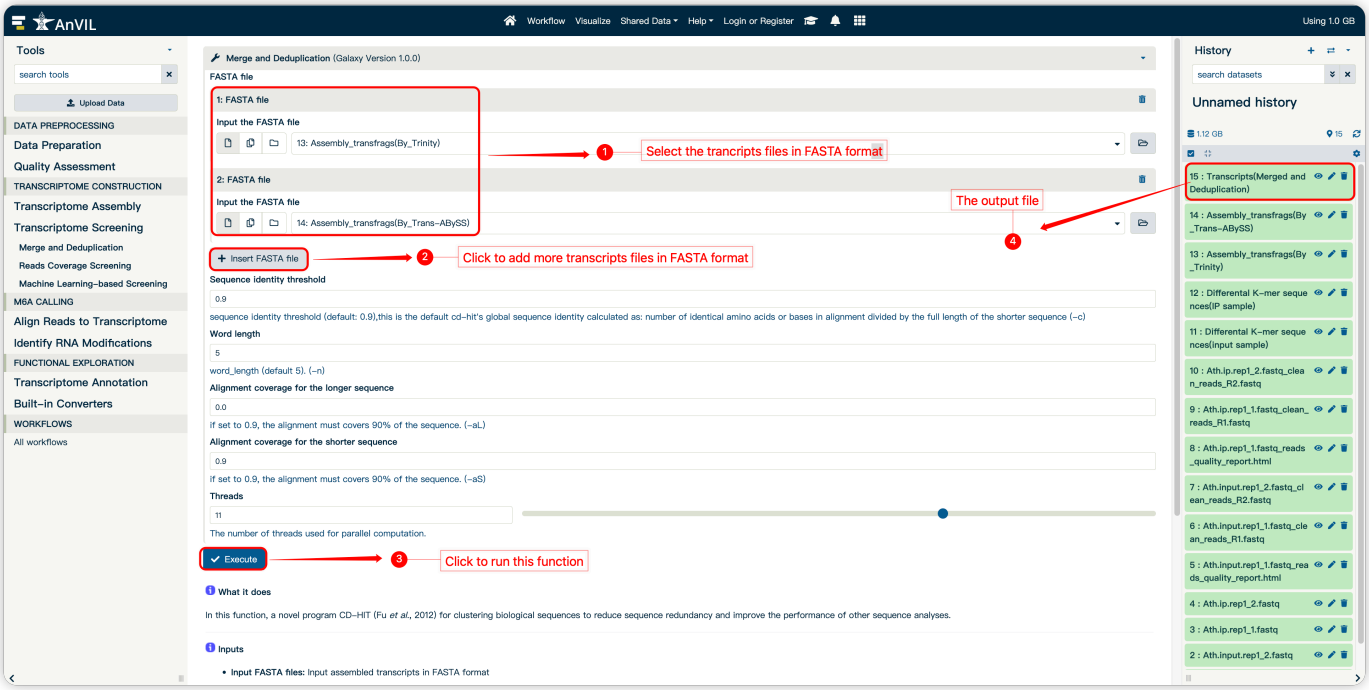
- **Input FASTA files:** Input assembled transcripts in FASTA format

Output

- **Transcripts after deduplication in FASTA format**

How to use this function

- The following screenshot shows us how to use this function



Reads Coverage Screening

In this function, we developed a pipline for screening transcripts based on reads coverage. **Bowtie 2** (Langmead *et al*., 2009; Langmead *et al*., 2012) for aligning sequencing reads, reads coverage was measured by **SAMtools** (Danecek *et al*., 2021) and **BEDTools** (Quinlan *et al*., 2010).

Input

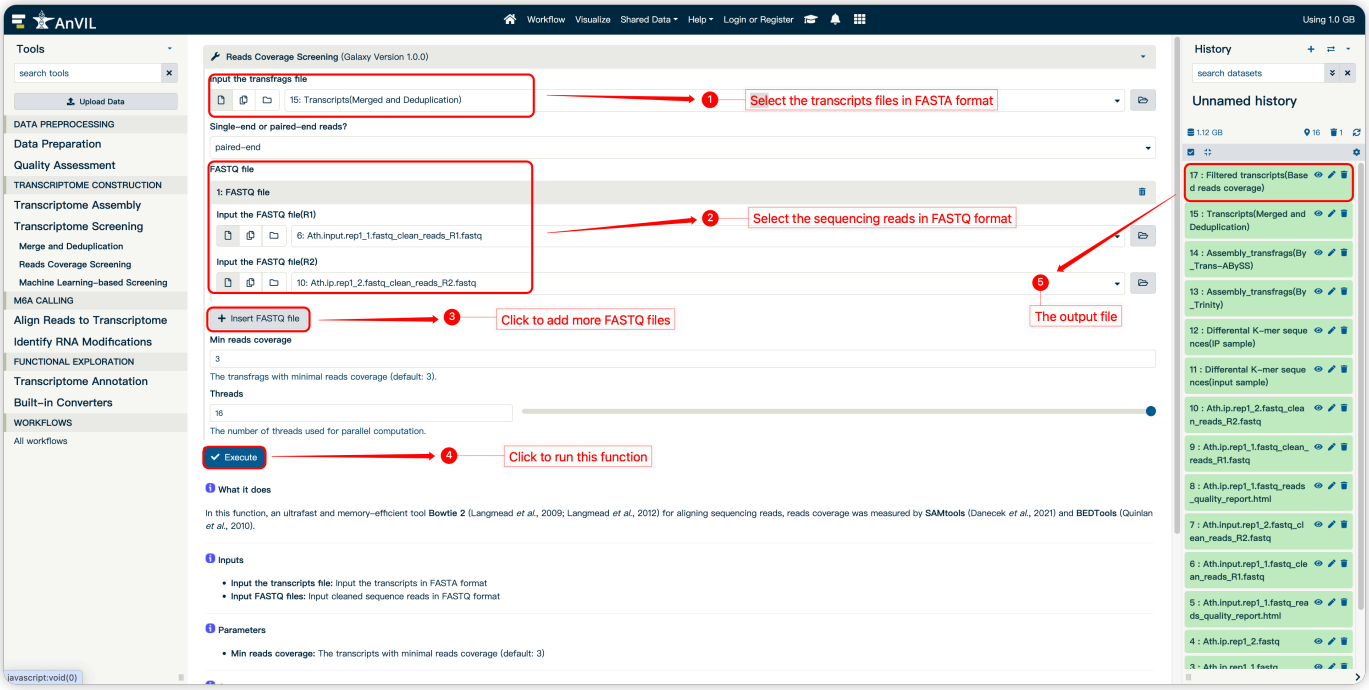
- **Input FASTQ files:** Input cleaned sequence reads in FASTQ format

Output

- **de novo transcriptome assembly by Trinity in FASTA format**

How to use this function

- The following screenshot shows us how to use this function



Machine Learning-based Screening

In this function, we developed a pipline for screening transcripts based on machine learning. **MMSeq2** (Steinegger *et al*., 2017) to cluster the assembled transfrags with both mRNA and ncRNA sequences from the Ensembl Plants across all plant species. Both assembled tranfrags and annotated plant sequences to serve as positive sample set. Then we used the **corain** package (Wang *et al*., 2023) to encode for RNA from three perspectives, sequence, structure and physical chemical properties. Adopted the **PSoL** algorithm (Wang *et al*., 2006), the initial negative samples were generated from "unlabeled" sample set based on the PU bagging algorithm, the expanded negative samples iteratively using RF classifier until the designated iteration number was reached.

Input

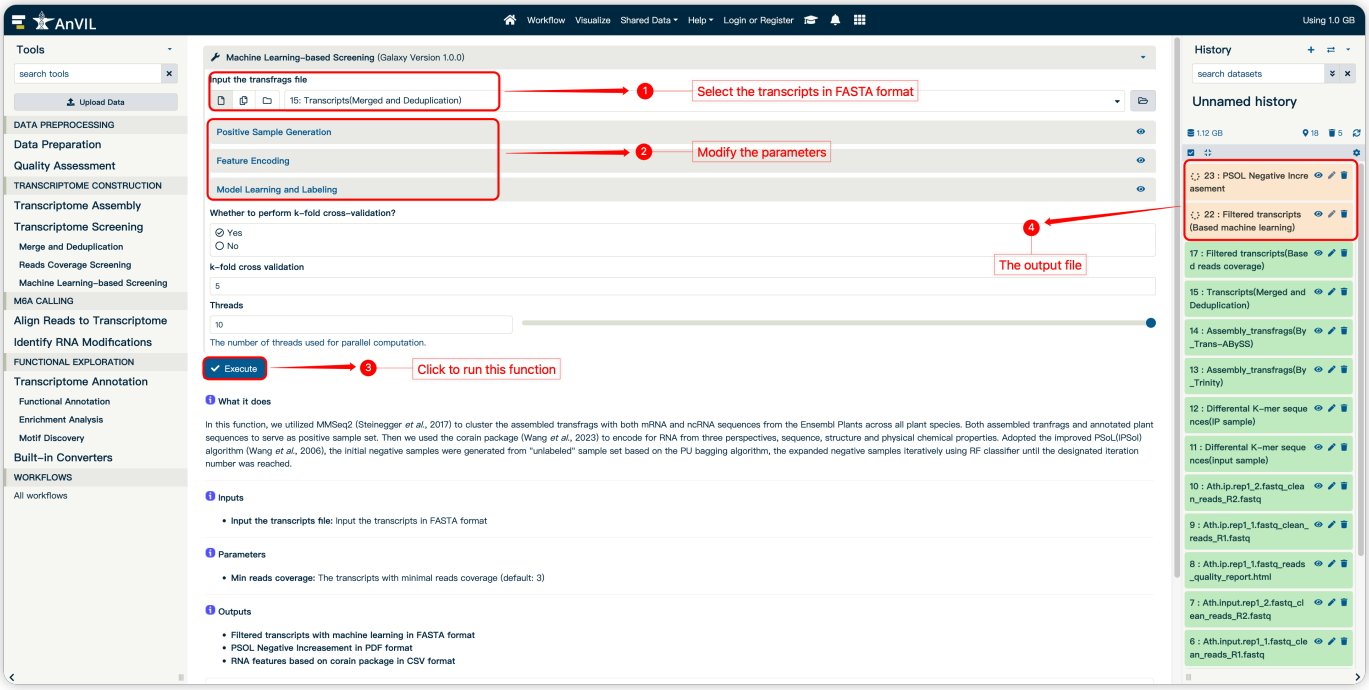
- **Input FASTQ files:** Input cleaned sequence reads in FASTQ format

Output

- **Filtered transcripts with machine learning in FASTA format**
- **PSOL Negative Increasement in PDF format**
- **RNA features based on corain package in CSV format**

How to use this function

- The following screenshot shows us how to use this function


AnVIL
Workflow
Visualize
Shared Data
Help
Login or Register
Using 1.0 GB
Tools
search tools
Upload Data
DATA PREPROCESSING
Data Preparation
Quality Assessment
TRANSCRIPTOME CONSTRUCTION
Transcriptome Assembly
Transcriptome Screening
Merge and Deduplication
Reads Coverage Screening
Machine Learning-based Screening
M6A CALLING
Align Reads to Transcriptome
Identify RNA Modifications
FUNCTIONAL EXPLORATION
Transcriptome Annotation
Functional Annotation
Enrichment Analysis
Motif Discovery
Built-in Converters
WORKFLOWS
All workflows
Machine Learning-based Screening (Galaxy Version 1.0.0)
Input the transfrags file
15: Transcripts(Merged and Deduplication)
1
Select the transcripts in FASTA format
Positive Sample Generation
Feature Encoding
2
Modify the parameters
Model Learning and Labeling
Whether to perform k-fold cross-validation?
Yes
No
k-fold cross validation
5
Threads
10
The number of threads used for parallel computation.
Execute
3
Click to run this function
What it does
In this function, we utilized MMSeq2 (Steinegger et al., 2017) to cluster the assembled transfrags with both mRNA and ncRNA sequences from the Ensembl Plants across all plant species. Both assembled tranfrags and annotated plant sequences to serve as positive sample set. Then we used the corain package (Wang et al., 2023) to encode for RNA from three perspectives, sequence, structure and physical chemical properties. Adopted the improved PSoL(IPSol) algorithm (Wang et al., 2006), the initial negative samples were generated from "unlabeled" sample set based on the PU bagging algorithm, the expanded negative samples iteratively using RF classifier until the designated iteration number was reached.
Inputs
Input the transcripts file: Input the transcripts in FASTA format
Parameters
Min reads coverage: The transcripts with minimal reads coverage (default: 3)
Outputs
Filtered transcripts with machine learning in FASTA format
PSOL Negative Increasement in PDF format
RNA features based on corain package in CSV format
History
search datasets
Unnamed history
1.12 GB
18
5
23 : PSOL Negative Increasement
22 : Filtered transcripts (Based machine learning)
4
The output file
17 : Filtered transcripts(Based reads coverage)
15 : Transcripts(Merged and Deduplication)
14 : Assembly_transfrags(By_Trans-ABySS)
13 : Assembly_transfrags(By_Trinity)
12 : Differental K-mer sequences(IP sample)
11 : Differental K-mer sequences(input sample)
10 : Ath.ip.rep1_2.fastq_clean_reads_R2.fastq
9 : Ath.ip.rep1_1.fastq_clean_reads_R1.fastq
8 : Ath.ip.rep1_1.fastq_reads_quality_report.html
7 : Ath.input.rep1_2.fastq_clean_reads_R2.fastq
6 : Ath.input.rep1_1.fastq_clean_reads_R1.fastq