deepEA user manual

deepEA installation

- **Step 1**: Docker installation

  i) Docker installation and start (Official installation tutorial)

  For **Windows (Only available for Windows 10 Prefessional and Enterprise version):**

  - Download Docker for windows;
  - Double click the EXE file to open it;
  - Follow the wizard instruction and complete installation;
  - Search docker, select ***Docker for Windows*** in the search results and click it.

  For **Mac OS X (Test on macOS Sierra version 10.12.6 and macOS High Sierra version 10.13.3):**

  - Download Docker for Mac OS;
  - Double click the DMG file to open it;
  - Drag the docker into Applications and complete installation;
  - Start docker from Launchpad by click it.

  For **Ubuntu (Test on Ubuntu 18.04 LTS):**

  - Go to Docker, choose your Ubuntu version, browse to **pool/stable** and choose **amd64, armhf, ppc64el or s390x**. Download the **DEB** file for the Docker version you want to install;
  - Install Docker, supposing that the DEB file is download into following path:***"/home/docker-ce~ubuntu_amd64.deb"***

    ```
    $ sudo dpkg -i /home/docker-ce<version-XXX>~ubuntu_amd64.deb
    $ sudo apt-get install -f
    ```

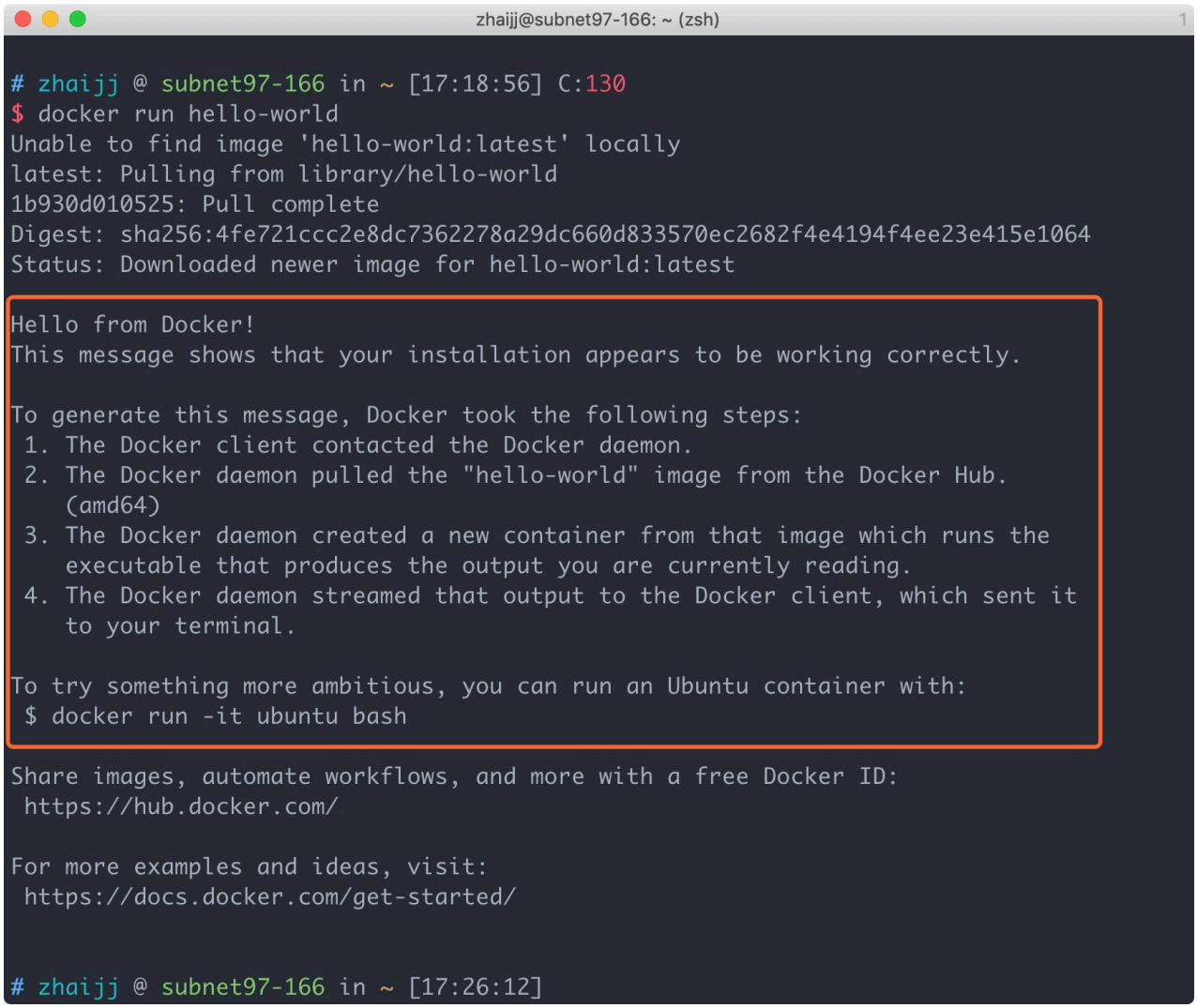
ii) Verify if Docker is installed correctly

Once Docker installation is completed, we can run `hello-world` image to verify if Docker is installed correctly. Open terminal in Mac OS X and Linux operating system and open CMD for Windows operating system, then type the following command:

```
$ docker run hello-world
```

Note: root permission is required for Linux operating system.

- Once Docker is installed successfully, you will see the following message:



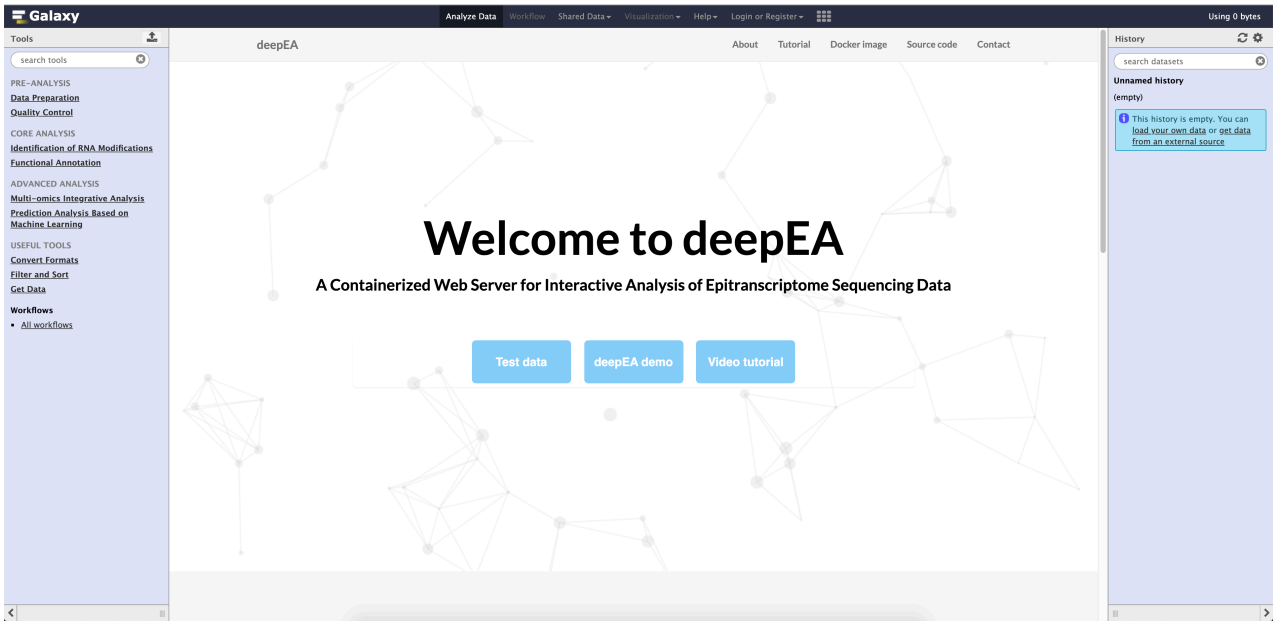
- **Step 2**: deepEA installation from Docker Hub

```
# pull latest deepEA Docker image from docker hub
$ docker pull malab/deepea
```

- **Step 3**: Launch deepEA local server

```
$ docker run -it -p 8080:8080 malab/deepea bash
$ bash /home/galaxy/run.sh
```

Then, deepea local server can be accessed via http://localhost:8080


Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Help
Login or Register
Using 0 bytes
Tools
search tools
PRE-ANALYSIS
Data Preparation
Quality Control
CORE ANALYSIS
Identification of RNA Modifications
Functional Annotation
ADVANCED ANALYSIS
Multi-omics Integrative Analysis
Prediction Analysis Based on Machine Learning
USEFUL TOOLS
Convert Formats
Filter and Sort
Get Data
Workflows
All workflows
deepEA
About
Tutorial
Docker image
Source code
Contact
Welcome to deepEA
A Containerized Web Server for Interactive Analysis of Epitranscriptome Sequencing Data
Test data
deepEA demo
Video tutorial
History
search datasets
Unnamed history
(empty)
This history is empty. You can load your own data or get data from an external source

Data Preparation

0. Introduction for Data Preparation

This module provides three funcitons (see following table for details) to prepare epitranscriptome data.

Tools	Description	Input	Output	Time (test data)	Reference
Obtain Genome Sequences and Annotation	Directly fetch genome, cDNA, CDS sequences and annotation from Ensembl plants database (https://plants.ensembl.org/index.html)	Select a species, a database version, and a data type	Sequences in FASTA format or GTF annotation (depends on user's selection)	Depends on the network speed	In-house scripts
Obtain Epitranscriptome Sequencing Reads	Directly fetch epitranscriptome sequencing reads from NCBI's SRA database	SRR accession or HTTP/FTP link	Sequencing reads in SRA format	Depends on the network speed	SRA Toolkit
Sequence Data Preprocessing	Convert epitranscriptome sequencing reads from SRA to FASTQ format	Epitranscriptome sequencing reads in SRA format	Epitranscriptome sequencing reads in FASTQ format	~2 mins	SRA Toolkit

1. Obtain Genome Sequences and Annotation

This function is designed to download sequences (Genome sequences, cDNA, CDS, proteins) and genome annotation (GTF/GFF3) automatically from Ensembl plants (https://plants.ensembl.org/index.html).

Input

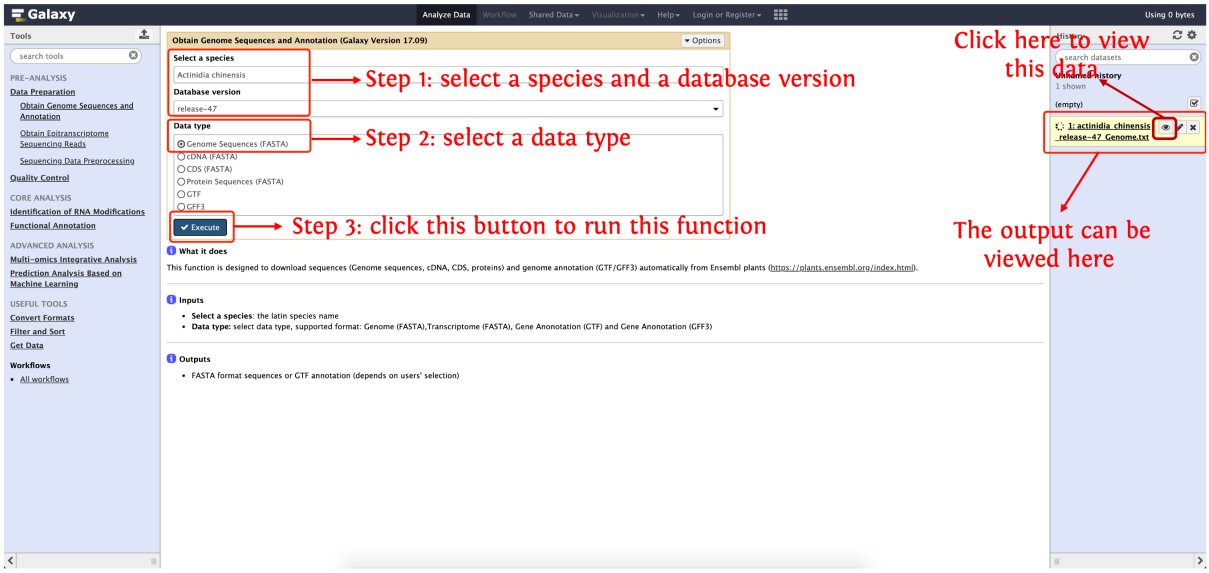
- **Select a species**: the latin name of 61 species are listed
- **Select database version**: Ensembl releases from 25 to 47 are listed
- **Select data type**: Genome sequences, cDNA, CDS, Protein Sequences, GTF and GFF3

Output

- Sequences in FASTA format or annotation in GTF/GFF3 format (depends on users' selection)

How to use this function

- The following screenshot shows us how to download genome sequences using deepEA



2. Obtain Epitranscriptome Sequencing Reads

This function is designed to download epitranscriptome sequencing reads from NCBI SRA (Short Read Archive) database or from an user-specified HTTP/FTP link automatically. For the former, the **prefetch** function implemented in SRA Toolkit is wrapped to enable users to download sequencing data from NCBI SRA database; For the latter, **wget** command line is used to download the file according to an user-specified HTTP/FTP link.

Input

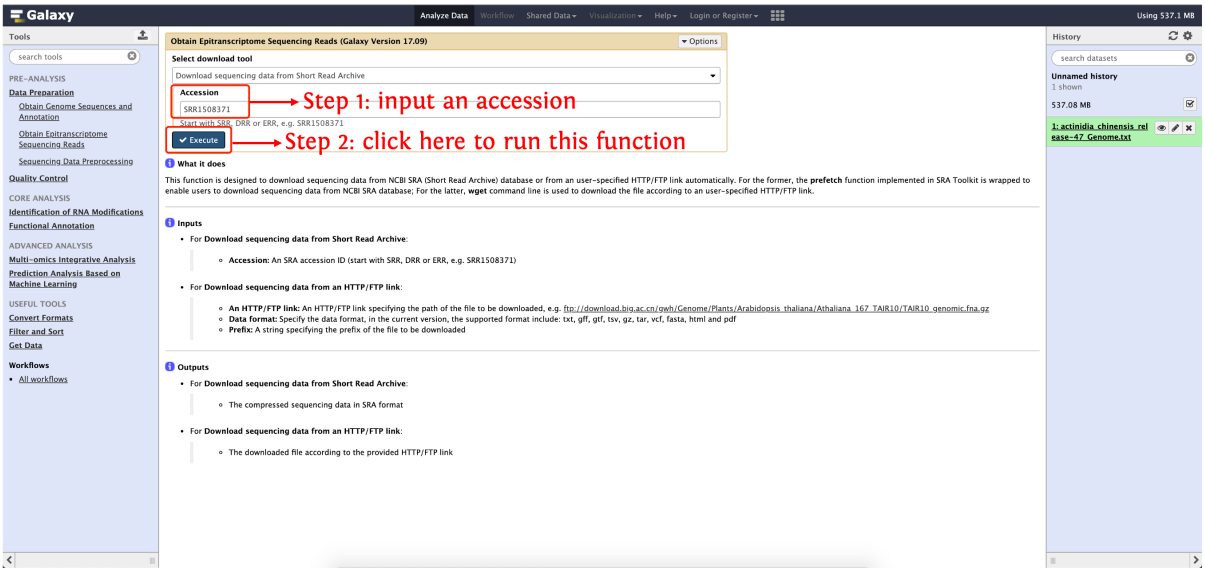
- For **Download sequencing data from Short Read Archive**:
  - **Accession:** An SRA accession ID (start with SRR, DRR or ERR, e.g. SRR1508371)
- For **Download sequencing data from an HTTP/FTP link**:
  - **An HTTP/FTP link**: An HTTP/FTP link specifying the path of the file to be downloaded, e.g. ftp://download.big.ac.cn/gwh/Genome/Plants/Arabidopsis_thaliana/Athaliana_167_TAIR10/TAIR10_genomic.fna.gz
  - **Data format**: Specify the data format, in the current version, the supported format include: txt, gff, gtf, tsv, gz, tar, vcf, fasta, html and pdf
  - **Prefix**: A string specifying the prefix of the file to be downloaded

Output

- For **Download sequencing data from Short Read Archive**:
  - The compressed sequencing data in SRA format
- For **Download sequencing data from an HTTP/FTP link**:
  - The downloaded file according to the provided HTTP/FTP link

How to use this function

- The following screenshot shows us how to download sequencing reads in SRA format



3. Sequence Data Preprocessing

This function wrapped **fastq-dump** function implemented in SRA Toolkit. See http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software for details.

Input

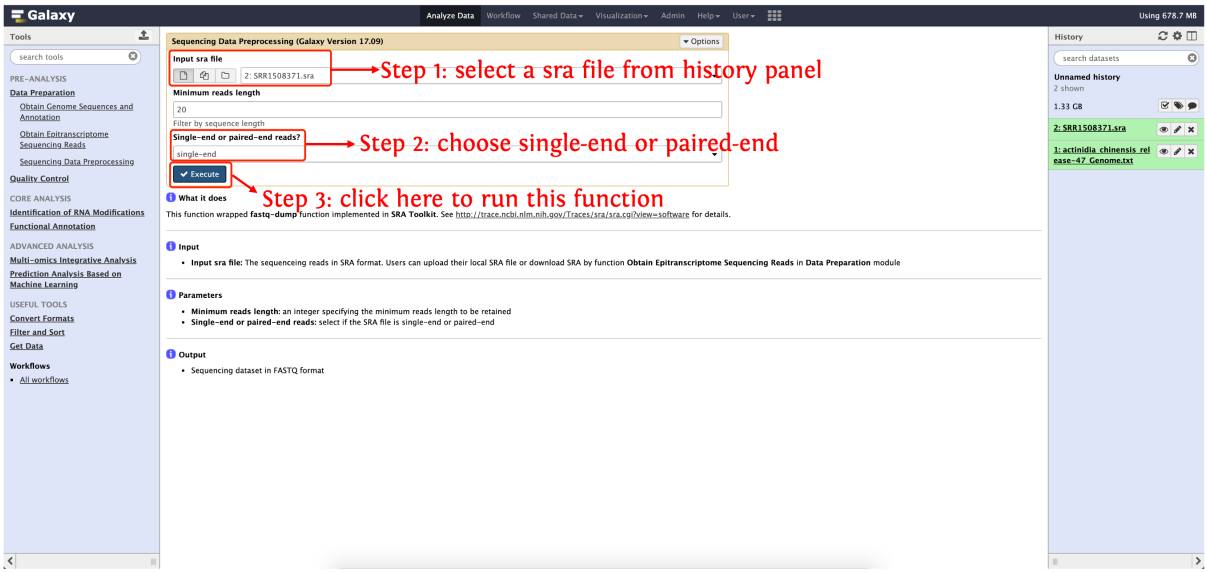
- **Input sra file:** The sequenceing reads in SRA format. Users can upload their local SRA file or download SRA by function **Obtain Epitranscriptome Sequencing Reads** in **Data Preparation** module

Output

- Sequencing dataset in FASTQ format

How to use this function

- The following screenshot shows us how to use this function to convert sequencing reads in SRA format to FASTQ format


Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Admin
Help
User
Using 678.7 MB
Tools
search tools
PRE-ANALYSIS
Data Preparation
Obtain Genome Sequences and Annotation
Obtain Epitranscriptome Sequencing Reads
Sequencing Data Preprocessing
Quality Control
CORE ANALYSIS
Identification of RNA Modifications
Functional Annotation
ADVANCED ANALYSIS
Multi-omics Integrative Analysis
Prediction Analysis Based on Machine Learning
USEFUL TOOLS
Convert Formats
Filter and Sort
Get Data
Workflows
All workflows
Sequencing Data Preprocessing (Galaxy Version 17.09)
Options
Input sra file
2: SRR1508371.sra
Step 1: select a sra file from history panel
Minimum reads length
20
Filter by sequence length
Single-end or paired-end reads?
single-end
Step 2: choose single-end or paired-end
Execute
Step 3: click here to run this function
What it does
This function wrapped fastq-dump function implemented in SRA Toolkit. See http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software for details.
Input
Input sra file: The sequenceing reads in SRA format. Users can upload their local SRA file or download SRA by function Obtain Epitranscriptome Sequencing Reads in Data Preparation module
Parameters
Minimum reads length: an integer specifying the minimum reads length to be retained
Single-end or paired-end reads: select if the SRA file is single-end or paired-end
Output
Sequencing dataset in FASTQ format
History
search datasets
Unnamed history
2 shown
1.33 GB
2: SRR1508371.sra
1: actinidia_chinensis_release-47_Genome.txt

Quality Control

0. Introduction for Quality Control

Quality Control module consists of a suite of tools focused on different levels of quality assessment, including reads quality, alignment quality and RNA modifications quality, Thus, three functions are implemented, including **Assess Reads Quality**, **Assess Alignment Quality** and **Assess RNA modifications Quality**. The details of as follows:

Tool	Description	Input	Output	Time (test data)	Programs	Reference
Assess Reads Quality	This function firstly performs quality control using FastQC and then trims low-quality reads using fastp	Epitranscriptome sequencing reads in FASTQ format	Clean reads in FASTQ format; Reads quality report in HTML format	~40s	FastQC,fastp	Chen *et al*., 2018, Bioinformatics,Babraham Bioinformatics
Assess Alignment Quality	Assess the quality of read-genome alignments (here reads from MeRIP-Seq experiments)	Reads alignments in SAM/BAM format	Alignment quality report in HTML format	~1 min	trumpet	Zhang *et al*., 2018, BMC Bioinformatics
Assess RNA Modifications Quality	Quantify RNA modifications signal strength by counting reads and calculating RPKM	RNA modifications in BED format and read alignments in SAM/BAM format	RNA modifications quantification matrix in HTML format	~1 min	DiffBind	Wu *et al*., 2016, Frontiers in Genetics

1. Assess Reads Quality

In this function, two existing NGS tools **FastQC** (Andrews *et al*., 2010) and **fastp** (Chen *et al*., 2018) are integrated to check sequencing reads quality and obtain high-quality reads, respectively.

Input

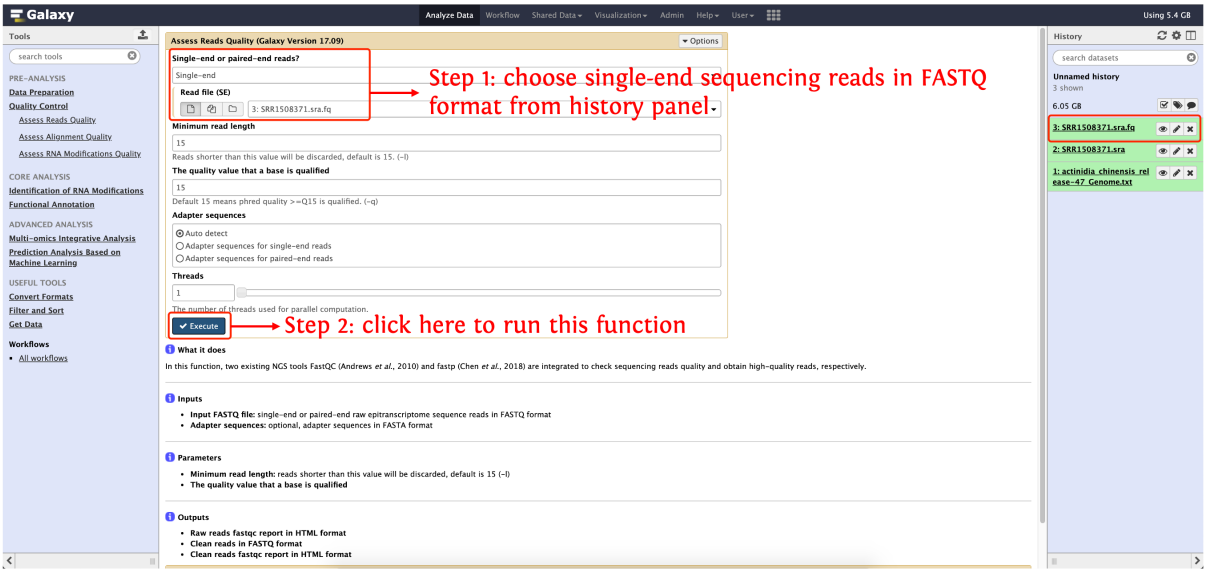
- **Input FASTQ file**: single-end or paired-end raw epitranscriptome sequence reads in FASTQ format
- **Adapter sequences**: optional, adapter sequences in FASTA format

Output

- **Clean reads in FASTQ format**
- **Reads quality report in HTML format**

How to use this function

- The following screenshot shows us how to assess reads quality



2. Assess Alignment Quality

This function is used to generate alignment quality assessment for MeRIP-Seq (methylated RNA immunoprecipitation sequencing) experiments by using an R package "trumpet" (Zhang *et al*., 2018, *BMC Bioinformatics*).

Input

- **BAM file in IP sample**: alignments in BAM format of immunoprecipitated (IP) RNA samples
- **BAM file in input sample**: alignments in BAM format of input (control) RNA samples
- **Genome annotation in GTF/GFF3 format**

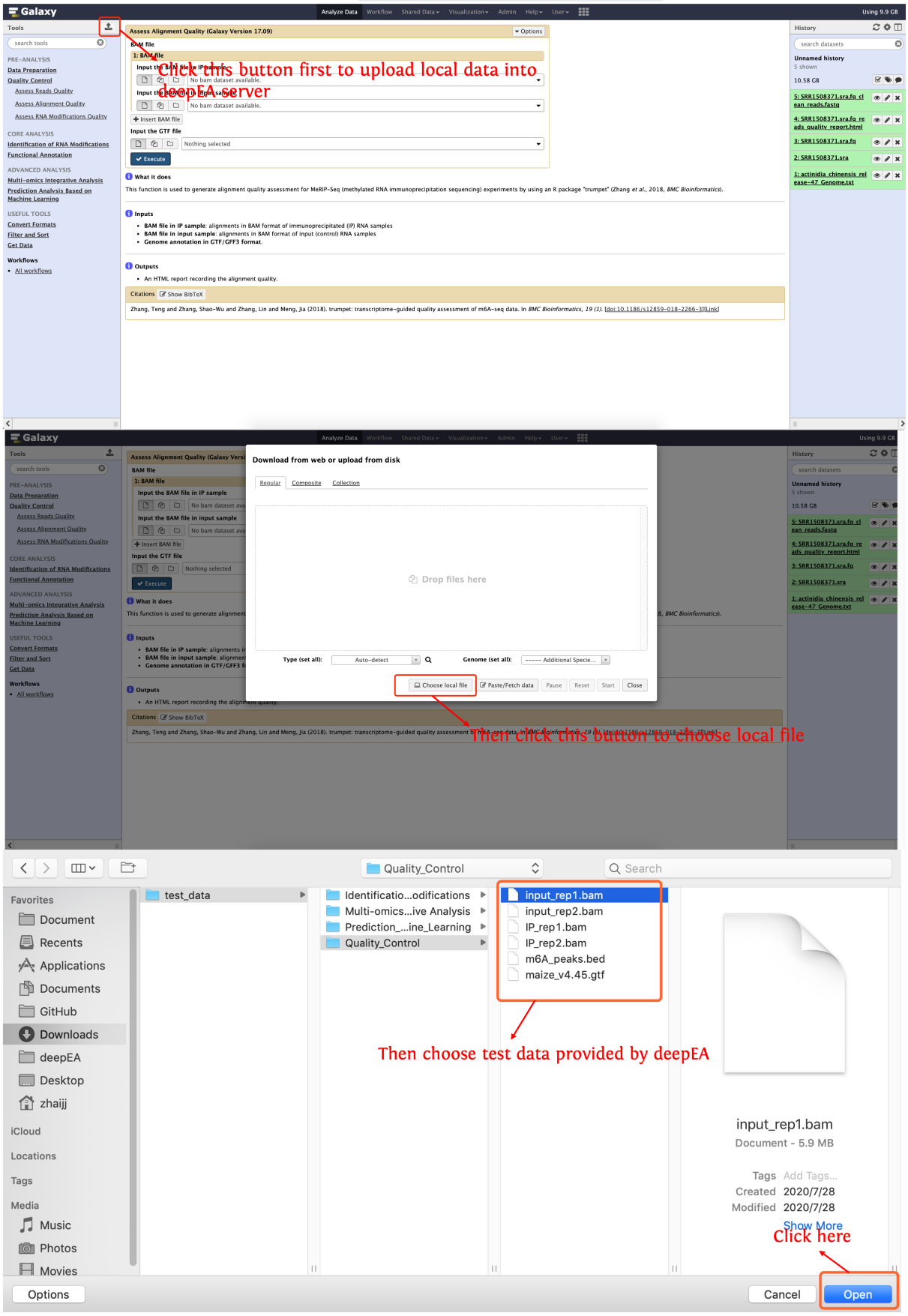
Output

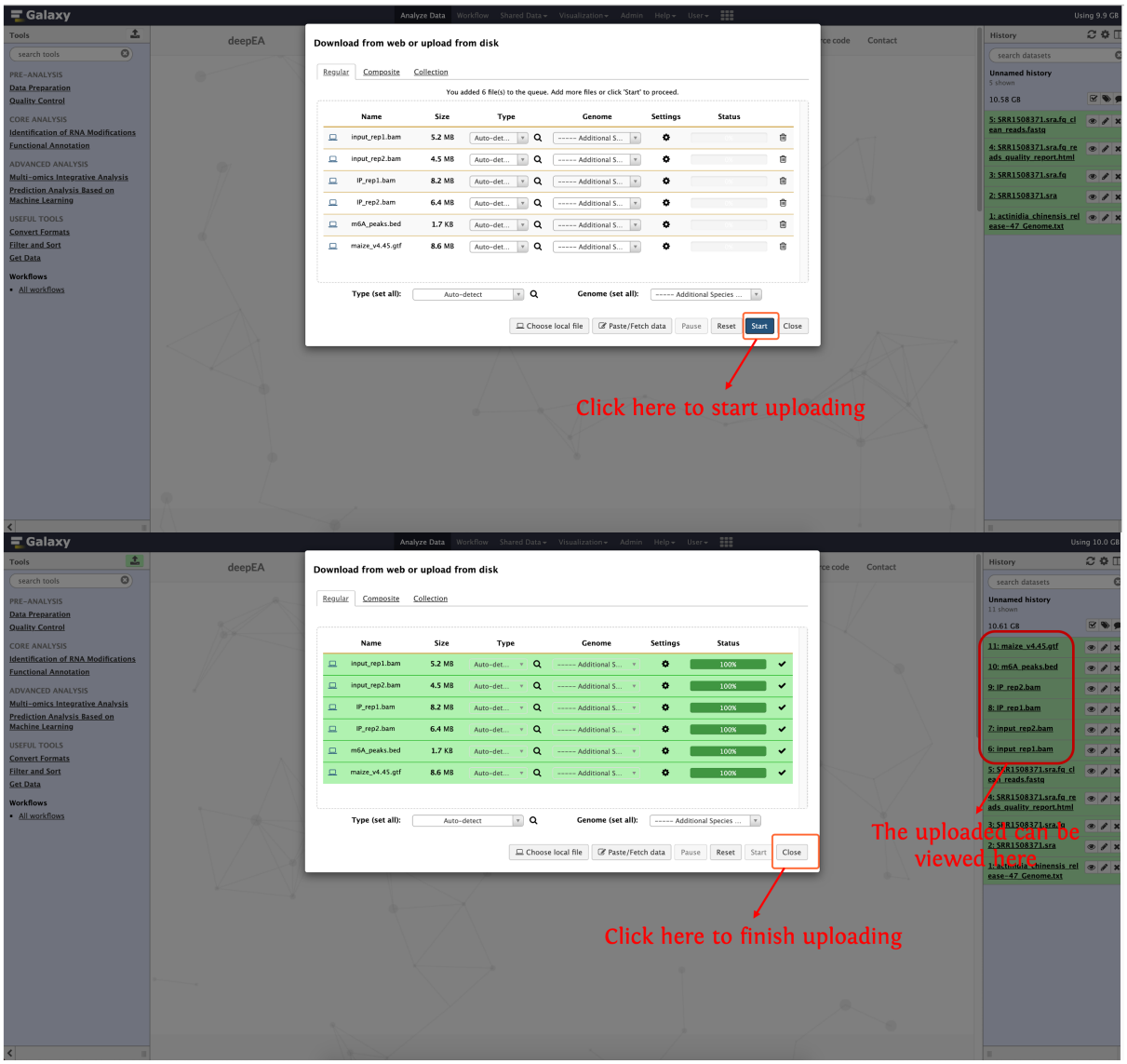
- An HTML report recording the alignment quality.

How to use this function

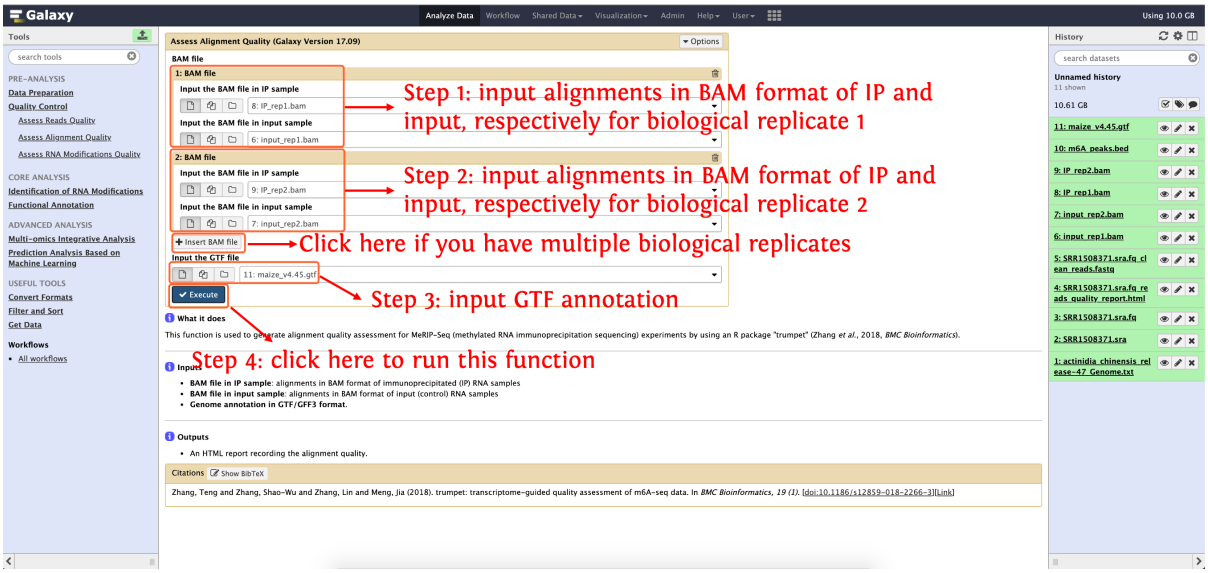
- **Step 1**: download test data provided by deepEA (https://deepea.nwafu.edu.cn/static/test_data.zip)

- **Step 2**: upload test data in directory `test_data/Quality_Control/` to history panel





- **Step 3**: input the corresponding file as the following screenshot shows to run this function:



3. Assess RNA Modifications Quality

This tool aims to quantify RNA modifications (e.g. m6A, m1A) signal strength by counting reads and calculating RPKM in binding site intervals using an R package Diffbind (Wu *et al*., 2016, *Frontiers in Genetics*)

Input

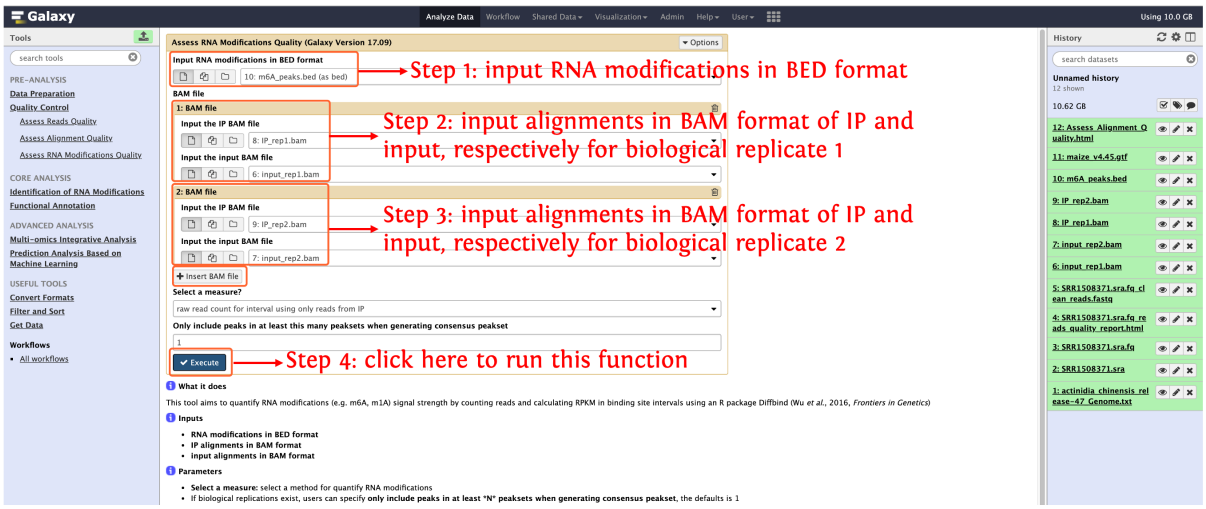
- **RNA modifications in BED format**
- **IP alignments in BAM format**
- **input alignments in BAM format**

Output

- An interactive HTML document recording each RNA modification region's RPKM

How to use this function

- Please see the following screenshots to run this function, test data used in the screenshot is available in `test_data/Quality_Control/`



Identification of RNA Modifications

0. Introduction for Identification of RNA Modifications

This module provides step-by-step functions required for epitranscriptome reads mapping and identification of RNA modifications.

Align Reads to Genome

Several commonly used aligners are wrapped to align epitranscriptome reads to genome. Currently, Tophat2, Bowtie2, STAR, HISAT2, bwa-mem.

Tools	Description	Input	Output	Time (test data)	Reference
Tophat2	Tophat2 is a spliced aligner, which aligns short reads by calling Bowtie2 but alows for variable-length indels with respect to the reference genome.			~50s	Kim *et al.*, 2013, Genome Biology
	Bowtie2 is a short read aligner which achieves a combination of high speed, sensitivity				

Bowtie2	and accuracy by combining the strengths of the full-text minute index with the flexibility and speed of hardware-accelerated dynamic programming algorithms, therefore bowtie2 is suitable for large genomes	Epitranscriptome sequencing reads in FASTQ format and reference genome sequences in FASTA format	Read alignments in SAM/BAM format	~10 s	Langmead *et al.*, 2012, Nature Methods
STAR	STAR is an ultrafast universal RNA-Seq aligner and can discover non-canonical splices and chimeric (fusion) transcripts			~16s	Dobin *et al.*, 2013, Bioinformatics
HISAT2	HISAT2 is an ultrafast spliced aligner with low memory requirements. It supports genomes of any size, including those larger than 4 billion bases			~8s	Kim *et al.*, 2015, Nature Methods
	bwa-mem is a relatively				

bwa-mem	early aligner based on backward search with Burrows–Wheeler Transform			~10s	Li *et al.*, 2009, Bioinformatics

Identify RNA Modifications

Identify RNA Modifications implements three pipelines for MeRIP-Seq, CeU-Seq and RNA-BSSeq, respectively.

Tools	Description	Input	Output	Time (test data)	Reference
Peak Calling from the MeRIP-Seq data	Identify enriched genomic regions from MeRIP-Seq experiment	Read alignments of IP and input in SAM/BAM format and reference genome sequences in FASTA format	RNA modifications in BED format	~36s	Zhai *et al.*, 2018, Bioinformatics
Calling m^5C from the RNA-BSseq data	Perform bisulfite sequencing (BS-Seq) read mapping, comprehensive methylation calling using meRanTK	Sequencing reads in FASTQ format and reference genome sequences in FASTA format	m^5C sites in BED format	~10 mins using 2 threads	Rieder *et al.*, 2016, Bioinformatics
Calling Ψ from CeU-Seq data	Identify pseudouridylation from CeU-Seq	Read alignments in SAM/BAM format and cDNA sequences in FASTA format	Pseudoridylation sites in BED format	~1 mins	Li *et al.*, 2015, Nature Chemical Biology

1. Align reads to genome

Currently, deepEA wrapped five aligners to map epitranscriptome reads to genome, here, we take Tophat2 as an example to show how to use deepEA to run reads mapping, the other four aligners are similar.

Input

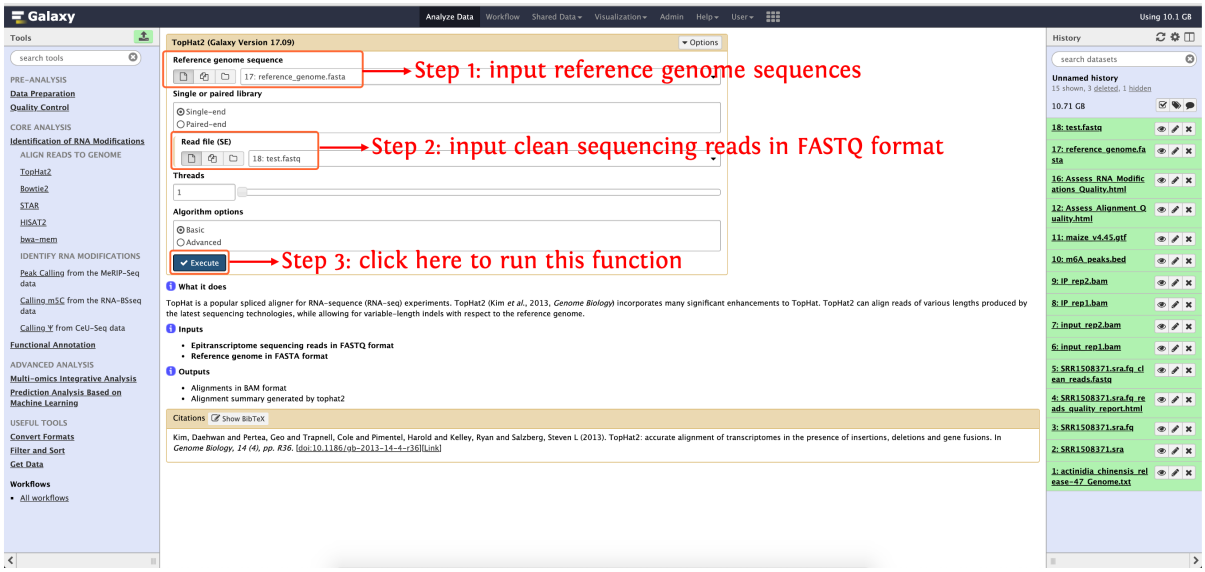
- **Epitranscriptome sequencing reads in FASTQ format**
- **Reference genome in FASTA format**

Output

- Alignments in BAM format
- Alignment summary generated by tophat2

How to use this function

- **Step 1**: upload the data in directory `test_data/Identification_of_RNA_Modifications/Align_Reads_to_Genome/` to history panel, if you are not clear about how to upload local data to deepEA server, please see here for details
- **Step 2**: see the following screenshot to run this function



2. Peak calling from the MeRIP-Seq data

Peak calling is used to identify enriched genomic regions in MeRIP-seq or ChIP-seq experiments. The function is implemented using the **peakCalling** function in PEA package (zhai *et al*., 2018)

Input

- **IP sample:** The IP experiment in BAM format
- **Input sample:** The input control experiment in BAM format
- **Reference genome:** The Reference genome sequences with FASTA format
- **Reference annotation file:** The Reference genome annotation file with GTF/GFF3 format (required for methods: **exomePeak**, **MeTPeak** and **BayesPeak**)

Output

- **The enriched peak region matrix in BED format**
  - For **SlidingWindow** method:

Chromosome	Start(1-based)	End	Bin number	Mean FDR	Max FDR	Minimum FDR	Mean Ratio	Max Ratio	Minimum Ratio
1	67476	67575	4	0.0136	0.0328	0.0001	-1.0012	-0.6334	-1.581
1	330776	330875	4	0.0215	0.0381	0.0007	-1.576	-1.4077	-1.788
1	389201	389300	4	0.0024	0.0070	0.0002	-1.115	-1.0598	-1.190

  - For **exomePeak** metod:

Chromosome	Start (0-based)	End	Gene ID	P.value	Strand
1	30663	30723	AT1G01040	0.0026	+
1	73831	74096	AT1G01160	2.5e-30	+
1	117530	117710	AT1G01300	2.4e-07	+

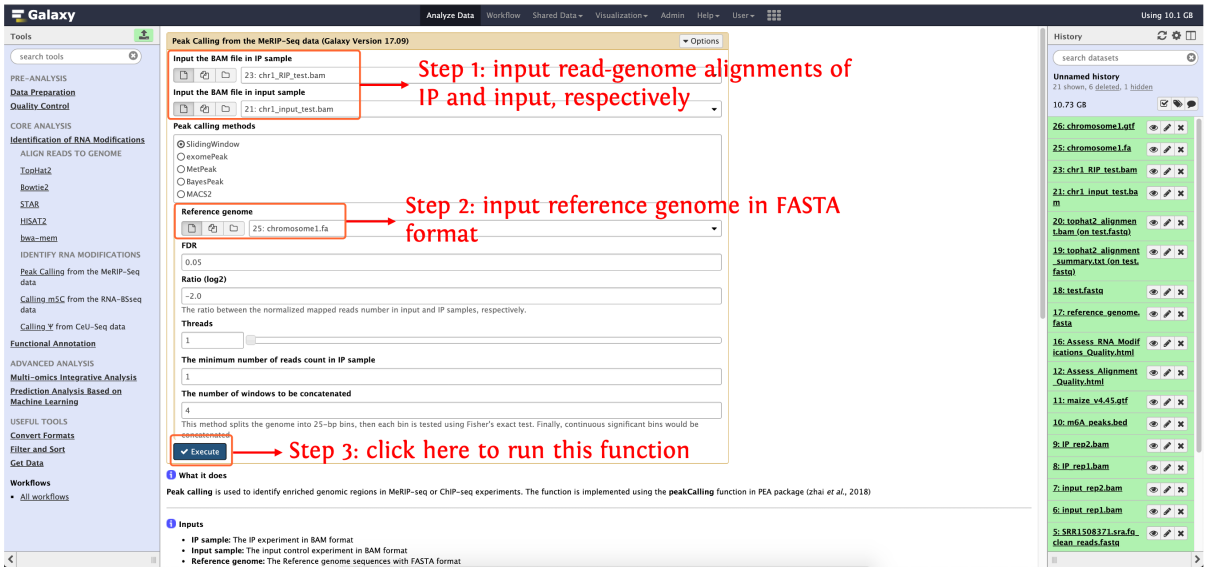
- For **MetPeak** method: it's the same as **exomePeak**
- For **BayesPeak** method:

chr	start	end	PP	job
1	3748	3848	0.0231	2
1	6848	6948	0.0178	2
1	6898	6998	0.9960	1

- For **macs2** method: please see macs2

How to use this function

- **Step 1**: upload the data in directory `test_data/Identification_of_RNA_Modifications/Peak Calling from the MeRIP-Seq data/` to history panel, if you are not clear about how to upload local data to deepEA server, please see here for details
- **Step 2**: see the following screenshot to run this function



3. Calling m^{5}C from the RNA-BSseq data

This function integrated meRanTK (Rieder *et al*., 2016, *Bioinformatics*) to perform RNA bisulfite sequencing (BS-Seq) read mapping, comprehensive methylation calling.

Input

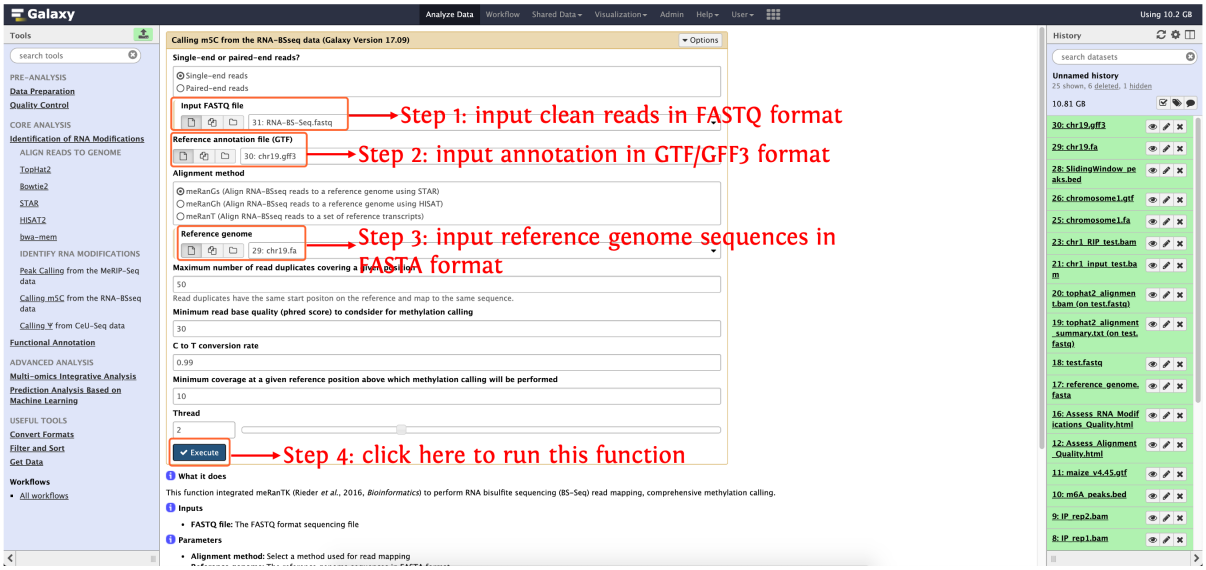
- **FASTQ file:** The FASTQ format sequencing file

Output

- **m5C_out_peaks:** The detected m^5C sites

How to use this function

- **Step 1**: upload the data in directory `test_data/Identification_of_RNA_Modifications/Calling m5C from the RNA-BSseq data/` to history panel, if you are not clear about how to upload local data to deepEA server, please see here for details
- **Step 2**: see the following screenshot to run this function



4. Calling Ψ from CeU-Seq data

This function is used to identify pseudouridylation from CeU-Seq (Li *et al*., 2015). To be specific, for any given position on a reference transcript, the stop rate of position i was calculated using the equation N_i_stop/(N_i_stop + N_i_readthrough), where N_i_stop (stop reads) is the number of reads with the mapping position starting at base i+1 (one nucleotide 3' to position i), and N_i_readthrough (readthrough reads) is the number of reads reading through position i; Then a position i is identified to be Ψ only when all of the following criteria were met:

- the stop reads of position i (N_i_stop) must be no less than 5 in the N3-CMC(+) sample;
- the stop rate in N3-CMC(−) samples must be less than 0.10;
- the difference of stop rate for position i between the N3-CMC(+) samples and the matched N3-CMC(−) samples must be at least 0.30.

Input

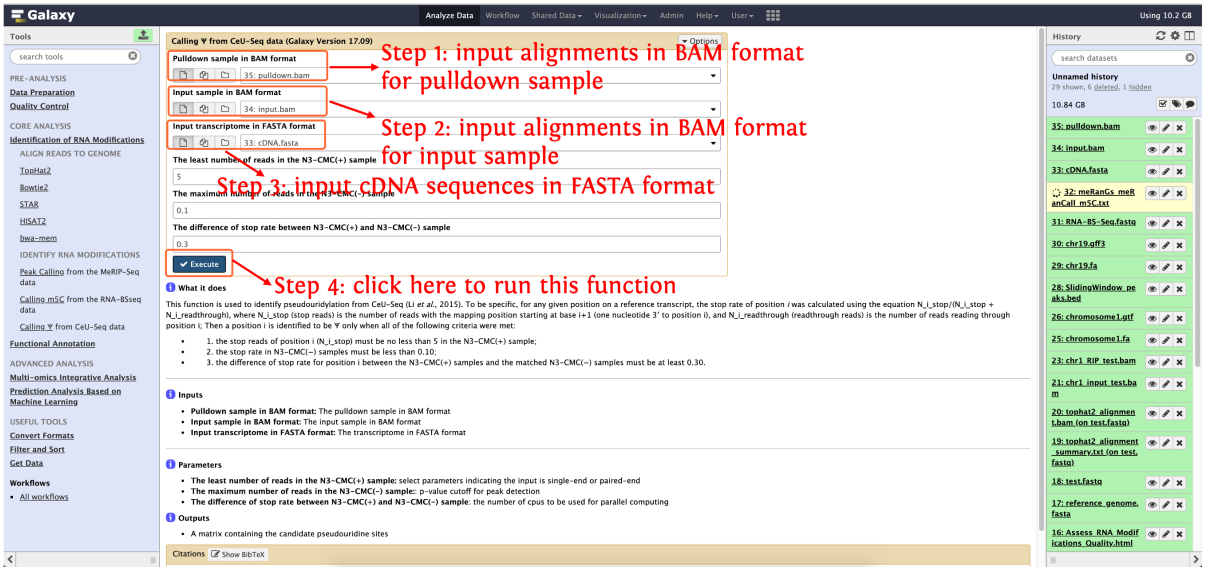
- **Pulldown sample in BAM format:** The pulldown sample in BAM format
- **Input sample in BAM format:** The input sample in BAM format
- **Input transcriptome in FASTA format:** The transcriptome in FASTA format

Output

- A matrix containing the candidate pseudouridine sites

How to use this function

- **Step 1**: upload the data in directory `test_data/Identification_of_RNA_Modifications/Calling pseudouridylation from CeU-Seq/` to history panel, if you are not clear about how to upload local data to deepEA server, please see here for details
- **Step 2**: see the following screenshot to run this function



Functional Annotation

0. Introduction for Functional Annotation

This module provided four functions to perform functional annotation of RNA modifications

Functions	Description	Input	Output	Time (test data)	Reference
RNA Modification Distribution	Visualize the distribution of RNA modifications in the genome and transcriptome, including the number of peaks in genomic features, the regions of enrichment of RNA modifications within transcripts, the enrichment of RNA modifications around transcriptional start/stop site and the enrichment of RNA modifications around splicing sites	RNA modifications in BED format and genome annotation in GTF/GFF3 format	Comprehensive overview of RNA modifications distribution in HTML or PDF format	~1 min	In-house scripts
Motif Analysis	Integrate MEME-ChIP and DREME to perform de-novo motif discovery	RNA modifications in BED format and reference genome sequences in FASTA format	Discovered motifs in HTML format	~4s	Timothy *et al.*, 2011, Bioinformatics,Philip *et al.*, 2011, Bioinformatics,Heinz *et al.*, 2010, Molecular Cell
Link RNA Modifications to Genes	Link RNA modifications to nearest genes based on genomic coordinate	RNA modifications in BED format and genome annotation in GTF/GFF3 format	Detailed RNA modifications-related genes	~5s	In-house scripts
Functional Enrichment Analysis	Perform GO or KEGG enrichment analysis for any species	Gene list	The enriched GO/KEGG terms	~6 mins	Yu *et al.*, 2012, OMICS

1. RNA Modification Distribution

This function is designed to provide insights into spatial and functional associations of RNA modifications. This function takes the RNA modifications in BED format and genome annotation in GTF (Gene Transfer Format) format as input, then the manner of distribution of RNA modifications in the genome and transcriptome is statistically analyzed and visualized, including the number of peaks in genomic feature (e.g. promoter, exon, intron, etc), the regions of enrichment of RNA modifications within transcripts, the enrichment of RNA modifications in transcriptional start/stop site and the enrichment of RNA modifications in splicing sites.

Input

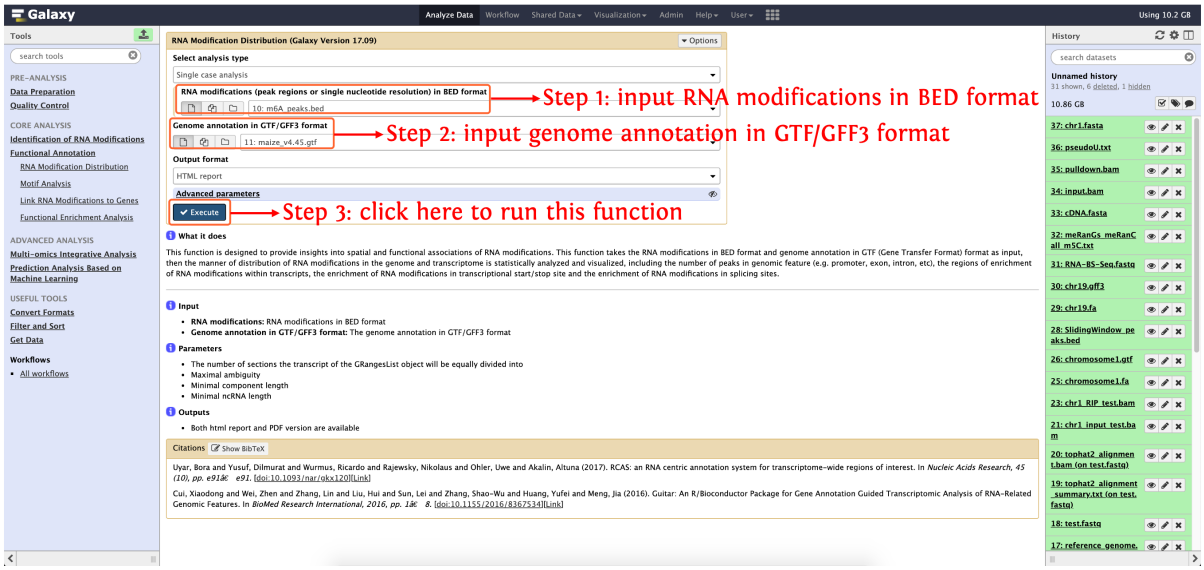
- **RNA modifications:** RNA modifications in BED format
- **Genome annotation in GTF/GFF3 format:** The genome annotation in GTF/GFF3 format

Output

- Both html report and PDF version are available

How to use this function

- **Step 1**: if you don't run the **Quality Control** module, please upload RNA modifications in BED format is in `test_data/Quality_Control/m6A_peaks.bed` and genome annotation in GTF/GFF3 format is in `test_data/Quality_Control/maize_v4.45.gtf` into history panel, please see here for details about how to upload local data to deepEA server
- **Step 2**: see the following screenshot to run this function



2. Motif Analysis

This function integrates MEME-ChIP and DREME to perform *de-novo* motif discovery.

Input

- **RNA modifications (peak regions or single nucleotide resolution) in BED format**
- **Reference genome sequences in FASTA format**

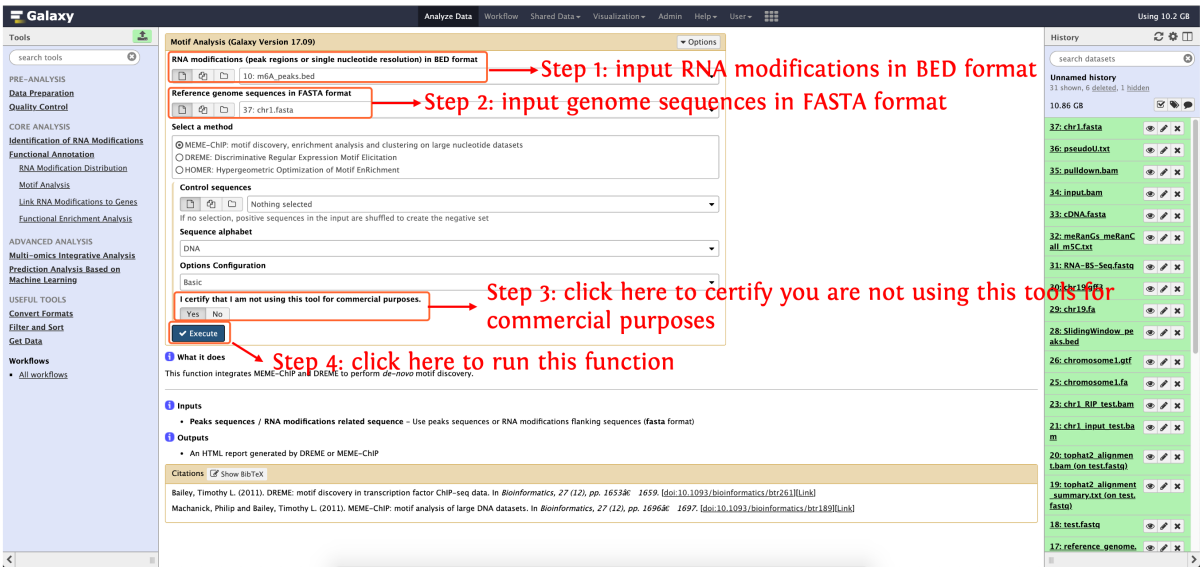
Output

- An HTML report generated by DREME or MEME-ChIP

How to use this function

- **Step 1**: upload RNA modifications in BED format is in `test_data/Quality_Control/m6A_peaks.bed` and genome sequences in FASTA format is in `test_data/Machine Learning-based Modelling Analysis/chr1.fasta` into history panel, please see here for details about how to upload local data to deepEA server

- **Step 2**: see the following screenshot to run this function



3. RNA Modifications Annotation with Gene

This function is designed to annotate RNA modifications with genes, users can specify the minimum overlapped length with genes.

Input

- **RNA modifications:** RNA modifications in BED format which can be obtained by any function in **Identify RNA Modifications**
- **Genome annotation in GTF/GFF3 format:** The genome annotation in GTF/GFF3 format

Output

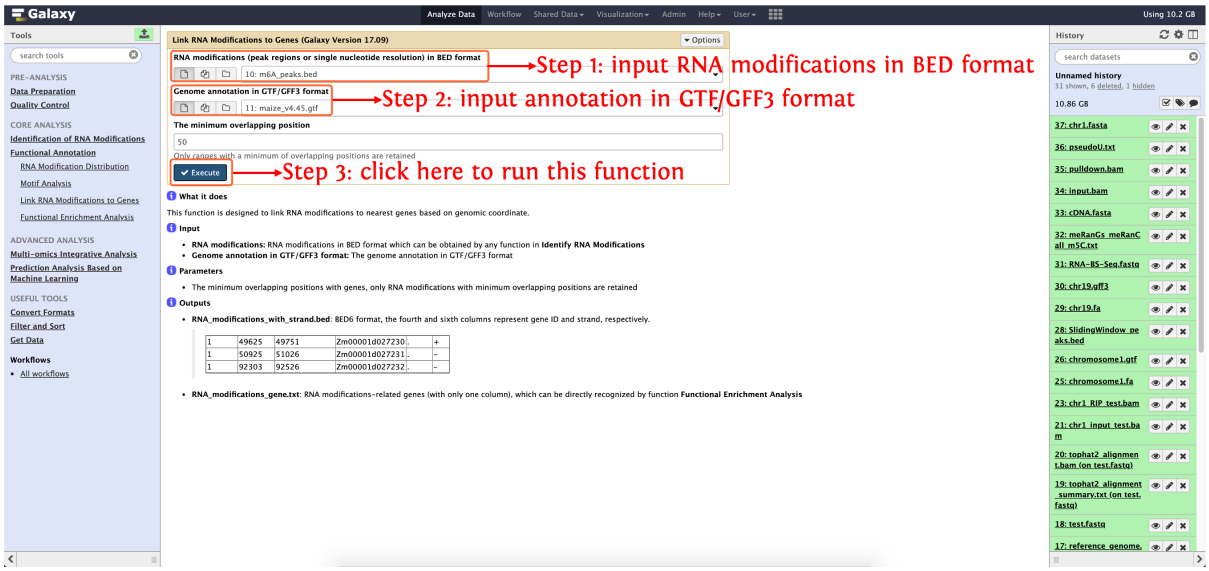
- **RNA_modifications_with_strand.bed**: BED6 format, the fourth and sixth columns represent gene ID and strand, respectively.

Chr	Start	End	GeneID	NA	Strand
1	49625	49751	Zm00001d027230	.	+
1	50925	51026	Zm00001d027231	.	-
1	92303	92526	Zm00001d027232	.	-

- **RNA_modifications_gene.txt**: RNA modifications-related genes (with only one column), which can be directly recognized by function **Functional Enrichment Analysis**

How to use this function

- **Step 1**: if you don't run the **Quality Control** module, please upload RNA modifications in BED format is in `test_data/Quality_Control/m6A_peaks.bed` and genome annotation in GTF/GFF3 format is in `test_data/Quality_Control/maize_v4.45.gtf` into history panel, please see here for details about how to upload local data to deepEA server
- **Step 2**: see the following screenshot to run this function



4. Functional Enrichment Analysis

This function is designed to perform GO or KEGG enrichment analysis for any species through R package "clusterProfiler".

Input

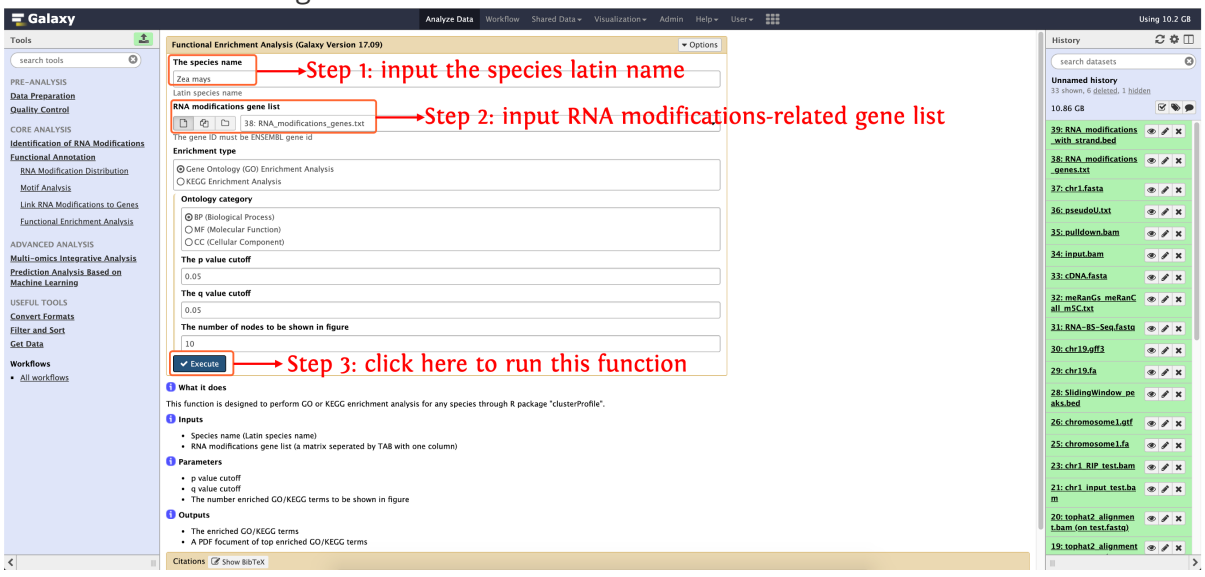
- **Species name (Latin species name)**
- **RNA modifications gene list (a matrix seperated by TAB with one column)**

Output

- The enriched GO/KEGG terms
- A PDF focument of top enriched GO/KEGG terms

How to use this function

- RNA modifications-related gene list can be obtained by function **RNA Modifications Annotation with Gene**
- Please see the following screenshot to run this function



Multi-omics Integrative Analysis

0. Introduction for Multi-omics Integrative Analysis

This module consists of two functions: **Integrative Analysis of Two Omics Data Sets** and **Integrative Analysis of Three Omics Data Sets**.

Functions	Description	Input	Output	Time (test data)	Reference
Integrative Analysis of Two Omics Data Sets	Visualize genes' abundance in two omics data sets via a scatter plot and perform kmeans cluster based on genes' abundace in each omics data.	A quantification matrix of RNA modifications-related genes on two omics data sets	An interactively HTML document recoding the scatter plot and kmeans cluster results	~5s	In-house scripts
Integrative Analysis of Three Omics Data Sets	Group genes into seven categories based on their abundance in each omics data sets and visualize each gene's abundance via a ternary plot	A quantification matrix of RNA modifications-related genes on three omics data sets	An interactively HTML document recoding the ternary plot, categories details defined by multi-omics data	~6s	In-house scripts

1. Integrative Analysis of Two Omics Data Sets

This function is designed to perform integrative analysis of two omics data sets (e.g., m6A and RNA-Seq). Taking a quantification matrix as input, deepEA firstly normalize raw quantification based on user-specific normalization method (currently, three normalization methods including **cumulative distribution, Z-score normalization and min-max normalization** are available), then using kmeans clustering method to cluster genes into four groups and visualize genes by an interactive scatter plot.

Input

- **Two omics quantication data**: a matrix with three columns, seperated by TAB, see following table for details

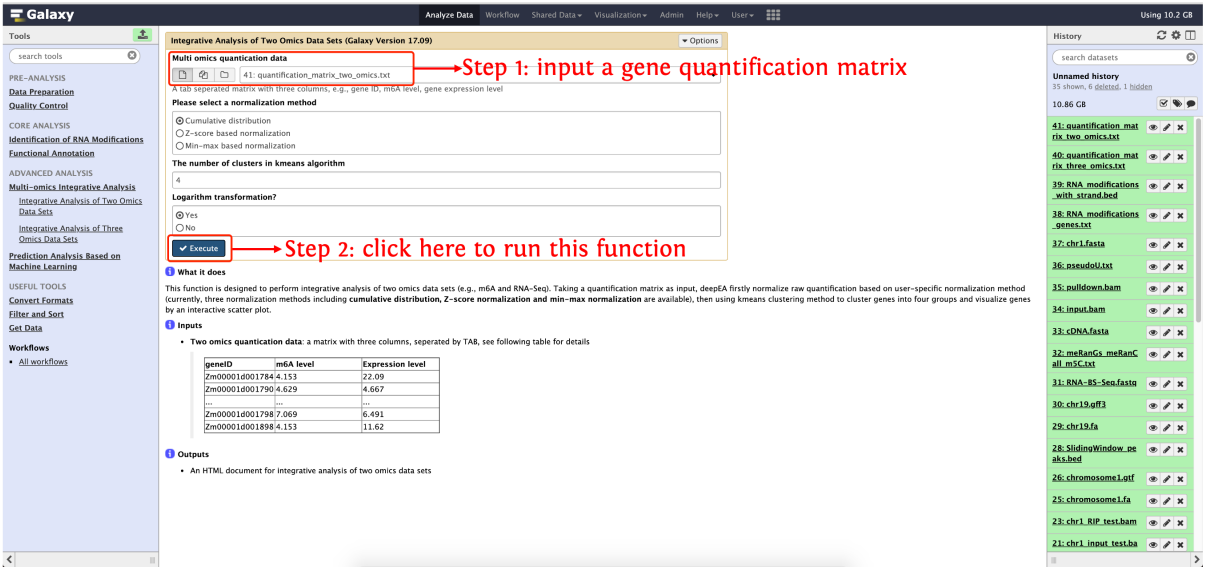
geneID	m^{6}A level	Expression level
Zm00001d001784	4.153	22.09
Zm00001d001790	4.629	4.667
...	...	...
Zm00001d001798	7.069	6.491
Zm00001d001898	4.153	11.62

Output

- An HTML document for integrative analysis of two omics data sets

How to use this function

- **Step 1**: upload gene quantification matrix in `test_data/Quality_Control/quantification_matrix_two_omics.txt` into history panel, please see here for details about how to upload local data to deepEA server
- **Step 2**: see the following screenshot to run this function



2. Integrative Analysis of Three Omics Data Sets

This function is designed to perform integrative analysis of three omics data sets. According RNA modifications-related genes' relative abundace in three omics data sets, deepEA grouped genes into seven categories. For example, if users would like to integrate m6A with gene expression and translation, the following categories will be illustrated in a ternary plot:

- **Balanced**
- **m^{6}A dominant**
- **m^{6}A suppressed**
- **Expression dominant**
- **Expression suppressed**

- **Translation dominant**
- **Translation suppressed**

Input

- **Multi omics quantication data**: see following table for details

geneID	RNA modification level	Expression level	Translation level
Zm00001d001784	4.153	22.09	35.18
Zm00001d001790	4.629	4.667	5.406
...	...	...	...
Zm00001d001798	7.069	6.491	7.891
Zm00001d001898	4.153	11.62	24.42

- Duplicated gene pairs (homoeologs): see following table for details:

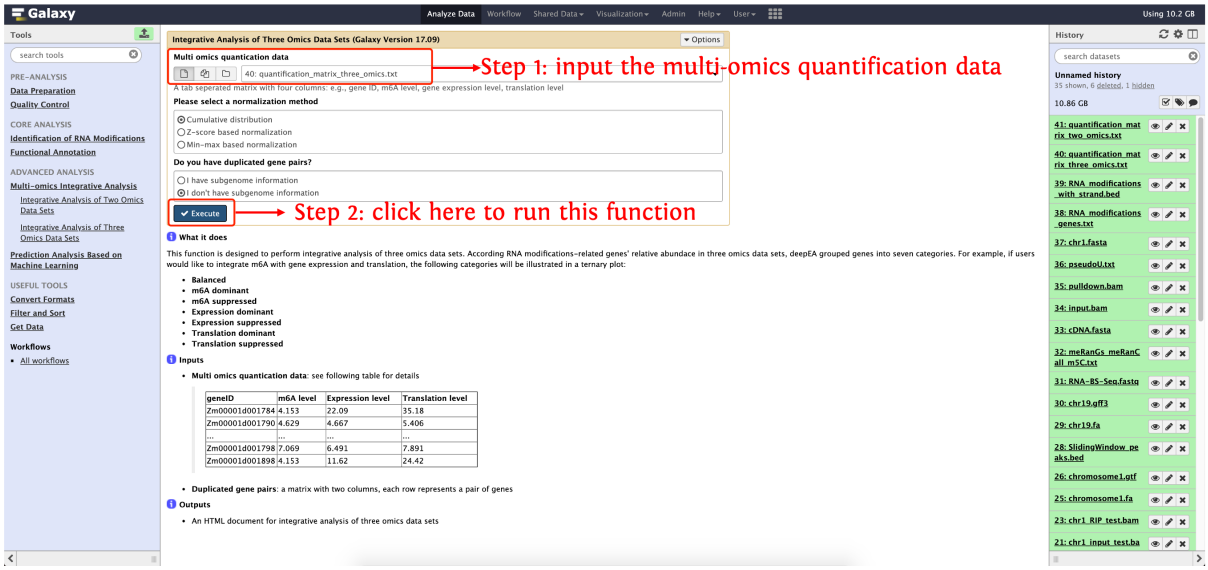
Gene 1	Gene 2
Zm00001d034918	Zm00001d012811
Zm00001d034901	Zm00001d012816
...	...
Zm00001d034896	Zm00001d012817
Zm00001d034876	Zm00001d012830

Output

- An HTML document for integrative analysis of three omics data sets

How to use this function

- **Step 1**: upload gene quantification matrix in `test_data/Quality_Control/quantification_matrix_three_omics.txt` into history panel, please see here for details about how to upload local data to deepEA server
- **Step 2**: see the following screenshot to run this function


Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Admin
Help
User
Using 10.2 GB
Tools
search tools
PRE-ANALYSIS
Data Preparation
Quality Control
CORE ANALYSIS
Identification of RNA Modifications
Functional Annotation
ADVANCED ANALYSIS
Multi-omics Integrative Analysis
Integrative Analysis of Two Omics Data Sets
Integrative Analysis of Three Omics Data Sets
Prediction Analysis Based on Machine Learning
USEFUL TOOLS
Convert Formats
Filter and Sort
Get Data
Workflows
All workflows
Integrative Analysis of Three Omics Data Sets (Galaxy Version 17.09)
Options
Multi omics quantication data
40: quantification_matrix_three_omics.txt
A tab seperated matrix with four columns: e.g., gene ID, m6A level, gene expression level, translation level
Please select a normalization method
Cumulative distribution
Z-score based normalization
Min-max based normalization
Do you have duplicated gene pairs?
I have subgenome information
I don't have subgenome information
Execute
Step 1: input the multi-omics quantification data
Step 2: click here to run this function
What it does
This function is designed to perform integrative analysis of three omics data sets. According RNA modifications-related genes' relative abundace in three omics data sets, deepEA grouped genes into seven categories. For example, if users would like to integrate m6A with gene expression and translation, the following categories will be illustrated in a ternary plot:
Balanced
m6A dominant
m6A suppressed
Expression dominant
Expression suppressed
Translation dominant
Translation suppressed
Inputs
Multi omics quantication data: see following table for details
geneID | m6A level | Expression level | Translation level
Zm00001d001784 | 4.153 | 22.09 | 35.18
Zm00001d001790 | 4.629 | 4.667 | 5.406
... | ... | ... | ...
Zm00001d001798 | 7.069 | 6.491 | 7.891
Zm00001d001898 | 4.153 | 11.62 | 24.42
Duplicated gene pairs: a matrix with two columns, each row represents a pair of genes
Outputs
An HTML document for integrative analysis of three omics data sets
History
search datasets
Unnamed history
35 shown, 6 deleted, 1 hidden
10.86 GB
41: quantification_matrix_two_omics.txt
40: quantification_matrix_three_omics.txt
39: RNA_modifications_with_strand.bed
38: RNA_modifications_genes.txt
37: chr1.fasta
36: pseudoU.txt
35: pulldown.bam
34: input.bam
33: cDNA.fasta
32: meRanGs_meRanCall_m5C.txt
31: RNA-BS-Seq.fastq
30: chr19.gff3
29: chr19.fa
28: SlidingWindow_peaks.bed
26: chromosome1.gtf
25: chromosome1.fa
23: chr1_RIP_test.bam
21: chr1_input_test.ba

Prediction Analysis Based on Machine Learning

0. Introduction for Prediction Analysis Based on Machine Learning

This module provides a pipeline for transcriptome-wide RNA modification prediction using machine learning technology. This pipeline is consisted of **Sample generation, Feature Encoding and Prediction System Construction**.

Functions	Description	Input	Output	Time (test data)	Reference
Sample Generation	Generate positive and negative samples for machine learning	RNA modifications in BED format; Reference genome sequences in FAST format; Genome annotation in GTF/GFF3 format	Positive and negative samples in BED format	~3 mins	In-house scripts
Feature Encoding	Characterize each sample with more than 900 numeric features.	Genome sequences in FASTA format and RNA modifications in BED format	Feature matrix seperated by TAB	~6 mins	In-house scripts
Prediction System Construction	Several commonly-used machine learning classification algorithms are provided to construct a predictor to classify RNA modifications from non RNA modifications.	Positive feature matrix and negative feature matrix	A predictor and model evaluation results	~15s	In-house scripts

1. Sample Generation

This function was designed to generate positive and negative samples based RNA modification regions. To be specific, this function takes RNA modification regions in BED format, genomic sequences in FASTA format and annotaiton in GTF format as input, then searches consensus motif (e.g. RRACH) in the RNA modification regions and treat them as positive samples, the remaining consensus motif in the same transcript of positive samples are randomly selected (user can specify the ratio between positive and negative samples) as negative samples.

Input

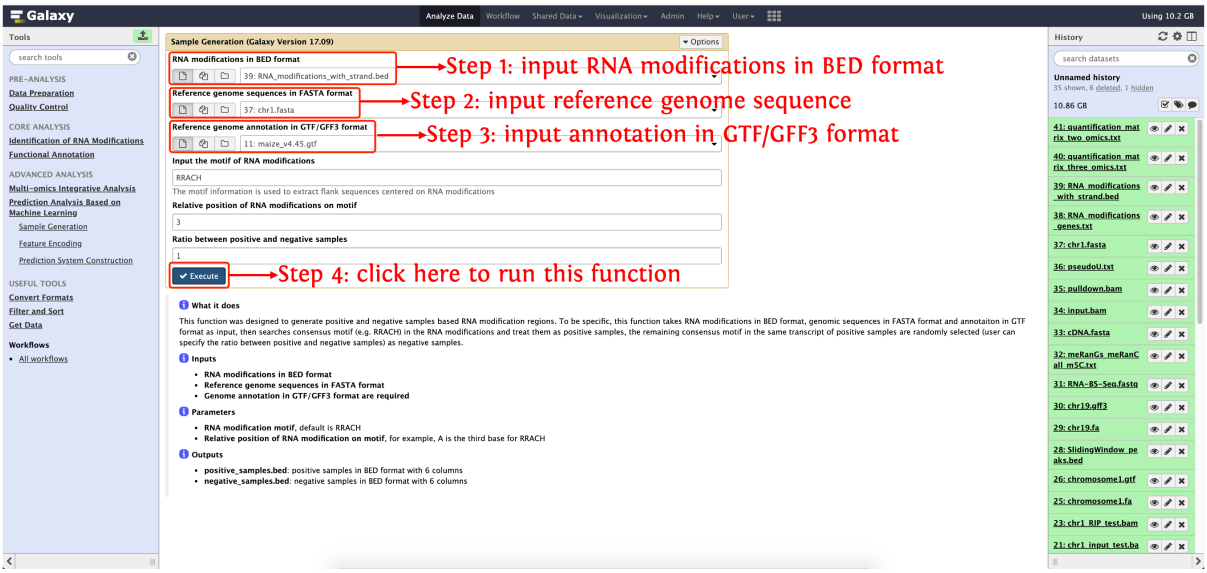
- **RNA modifications in BED format**
- **Reference genome sequences in FASTA format**
- **Genome annotation in GTF/GFF3 format are required**

Output

- **positive_samples.bed**: positive samples in BED format with 6 columns
- **negative_samples.bed**: negative samples in BED format with 6 columns

How to use this function

- **Step 1**: RNA modifications in BED format can be generated by function **Link RNA Modifications to Genes** in **Functional Annotaion** module.
- **Step 2**: upload reference genome sequences in directory `test_data/Prediction_Analysis_Based_on_Machine_Learning/chr1.fasta` and annotation in GTF/GFF3 format in directory `test_data/Quality_Control/maize_v4.45.gtf` to history panel.
- **Step 3**: see the following screenshot to run this function:



2. Feature Encoding

This function can be used to encode RNA modifications flanking sequences into a feature matrix. To be specific, **Sequence-derived features** integrated several commonly used feature encoding strategies including **Nucleic acid composition related features**, **Autocorerlation-based features**, **Pseudo nucleotide composition** and **Binary encoding**; For **Genomic-derived features**, we adopted feature encoding strategy used in WHISTLE (Chen *et al*., 2019, *Nucleic Acids Research*) project.

Input

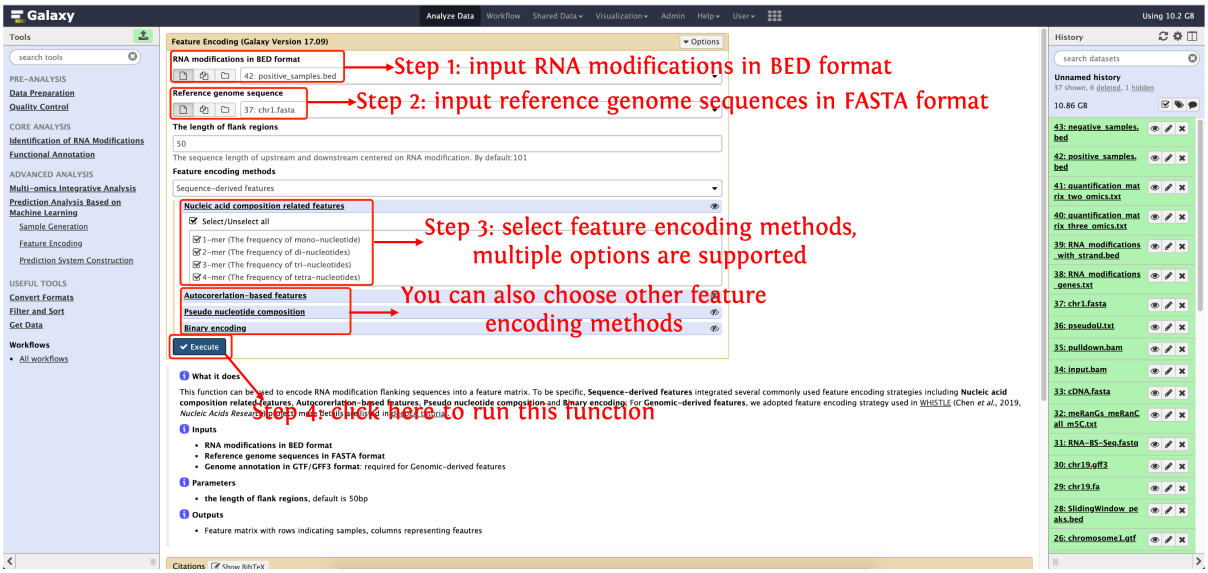
- **RNA modification in BED format**: which can be generated by function **Sample Generation**
- **Reference genome sequences in FASTA format**
- **Genome annotation in GTF/GFF3 format**: required for Genomic-derived features

Output

- Feature matrix with rows indicating samples, columns representing feautres

How to use this function

- RNA modifications in BED format: the output from last function (**Sample Generation**)
- Reference genome sequences is in `test_data/6-Machine Learning-based Modelling Analysis/chr1.fasta`
- **Note: please run this function two times to generate positive feature matrix and negative feature matrix, respectively**. The following screenshot shows how to use this function to generate positive feature matrix, run this function again but replace the first input with `negative_samples.bed` to generate negative feature matrix.



3. Prediction System Construction

In this module, several commonly-used machine learning classification algorithms are implemented to construct an RNA modification predictor. In the current version of DeepEA, the following five classical algorithms are inlcuded:

- Random Forest
- Support Vector Machine
- Decision Tree
- XGBoost
- Logistic Regression

Input

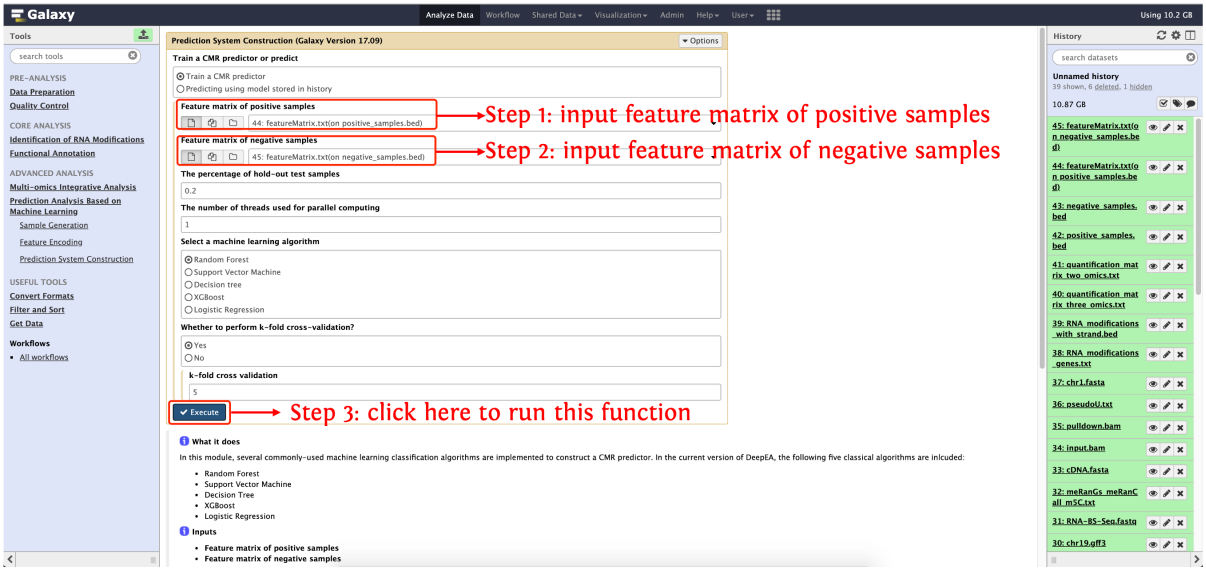
- **Feature matrix of positive samples**
- **Feature matrix of negative samples**

Output

- An RNA modification predictor in binary format
- Cross validation evaluation results in PDF format

How to use this function

- Both positive feature matrix and negative feature matrix can be generated by function **Feature Encoding**
- The following screenshot shows how to use this function to train a m^6 predictor, and evaluate using 5-fold cross validation

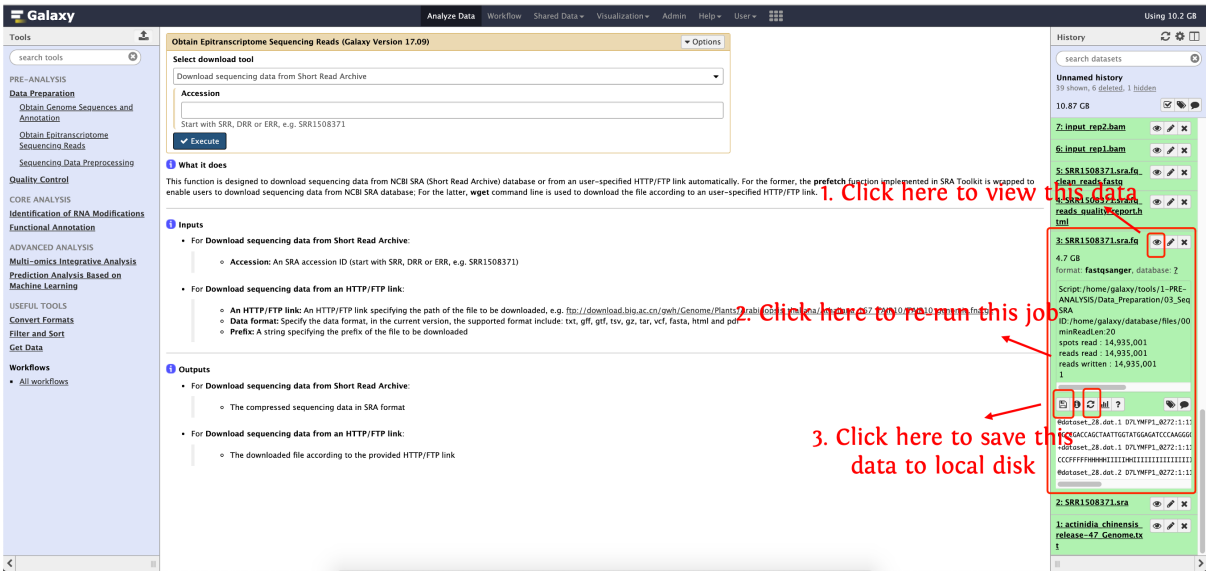


Tips and FAQs

Tips

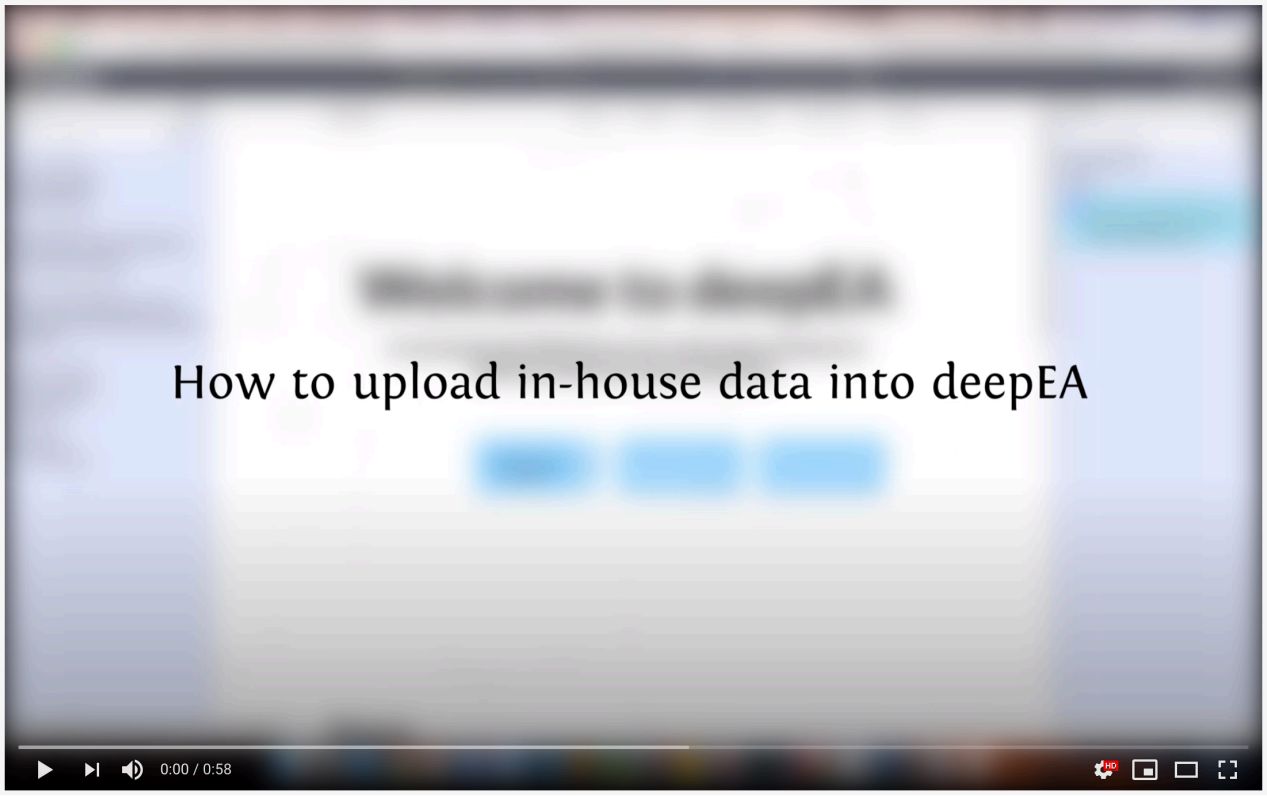
The folloing screenshot shows us:

- **How to view the data in deepEA**
- **How to re-run a job**
- **How to save data to your local disk**



Frequently Asked Questions (FAQs)

How to upload in-house data into deepEA local server



How to become an admin user

- First, register with email `admin@example.org`, set the password arbitrary.
- Then, login with `admin@example.org`

How to stop deepEA local server

- Press `Ctrl + C` (for windows and unix users) or `Cmd + C` (for Mac OS users)

How to re-launch deepEA local server when I exited the docker container

- First, using the following command to check the container ID

```
docker ps -a
```

- Then run the following command

```
docker container start  container ID
docker exec -it container ID bash
bash /home/galaxy/run.sh
```

How to mount local disk into deepEA docker container

```
docker run -it -v /your home directory:/home/galaxy/database/files/000 -p
8080:8080 malab/deepea bash
```