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

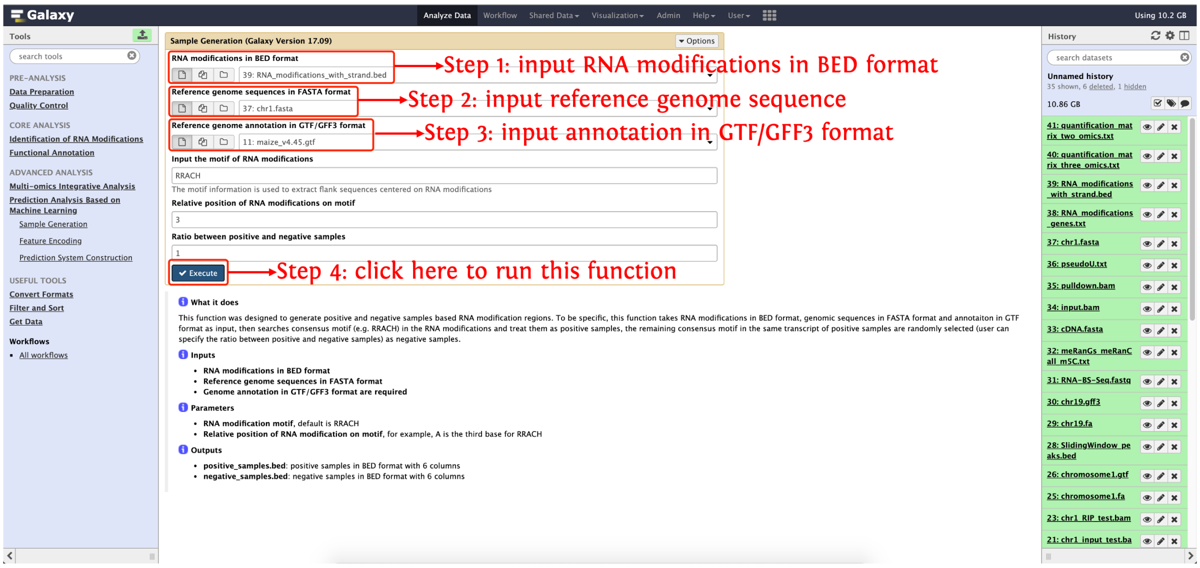
- **RNA modifications in BED format**
- **Reference genome sequences in FASTA format**
- **Genome annotation in GTF/GFF3 format are required**

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:



Sample Generation (Galaxy Version 17.09)

RNA modifications in BED format: 39: RNA_modifications_with_strand.bed

Step 1: input RNA modifications in BED format

Reference genome sequences in FASTA format: 37: chr1.fasta

Step 2: input reference genome sequence

Reference genome annotation in GTF/GFF3 format: 11: maize_v4.45.gtf

Step 3: input annotation in GTF/GFF3 format

Input the motif of RNA modifications: RRACH

The motif information is used to extract flank sequences centered on RNA modifications

Relative position of RNA modifications on motif: 3

Ratio between positive and negative samples: 1

✓ Execute

Step 4: click here to run this function

What it does

This function was designed to generate positive and negative samples based RNA modification regions. To be specific, this function takes RNA modifications 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 modifications 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.

Inputs

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

Parameters

- **RNA modification motif**, default is RRACH
- **Relative position of RNA modification on motif**, for example, A is the third base for RRACH

Outputs

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

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

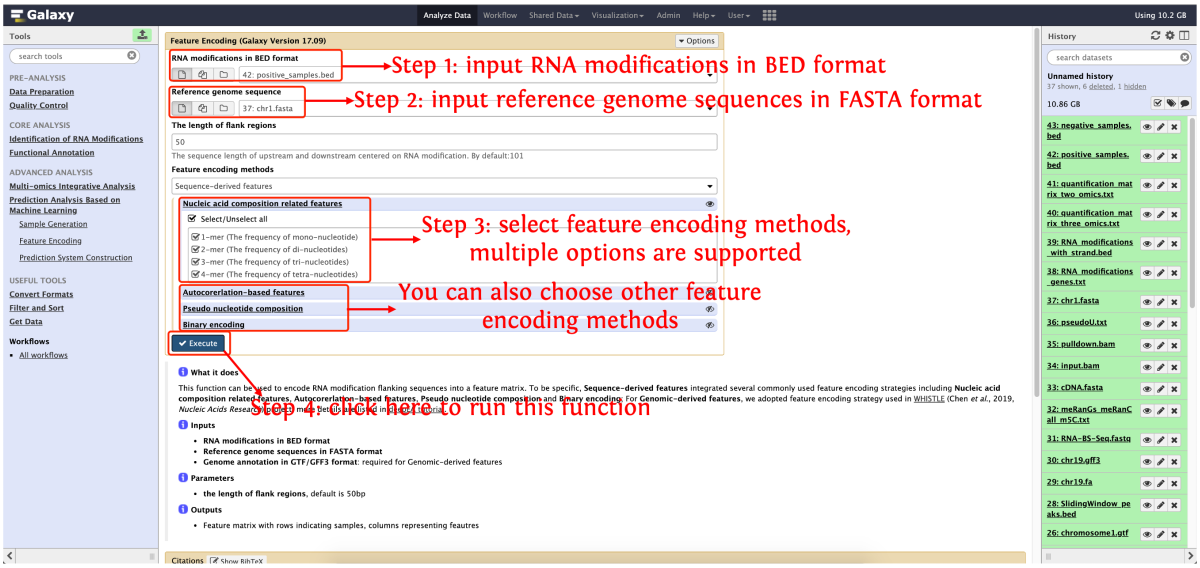
Output

- 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.



Feature Encoding (Galaxy Version 17.09)

RNA modifications in BED format: 42: positive_samples.bed — Step 1: input RNA modifications in BED format

Reference genome sequence: 37: chr1.fasta — Step 2: input reference genome sequences in FASTA format

The length of flank regions: 50

The sequence length of upstream and downstream centered on RNA modification. By default:101

Feature encoding methods: Sequence-derived features

Nucleic acid composition related features

☑ Select/Unselect all

☑ 1-mer (The frequency of mono-nucleotide)
☑ 2-mer (The frequency of di-nucleotides)
☑ 3-mer (The frequency of tri-nucleotides)
☑ 4-mer (The frequency of tetra-nucleotides)

Step 3: select feature encoding methods, multiple options are supported

Autocorerlation-based features
Pseudo nucleotide composition
Binary encoding

You can also choose other feature encoding methods

Execute — Step 4: click here to run this function

What it does

This function can be used to encode RNA modification 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. more details are listed in deepEA tutorial.

Inputs

- **RNA modifications in BED format**
- **Reference genome sequences in FASTA format**
- **Genome annotation in GTF/GFF3 format**: required for Genomic-derived features

Parameters

- **the length of flank regions**, default is 50bp

Outputs

- Feature matrix with rows indicating samples, columns representing feautres

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

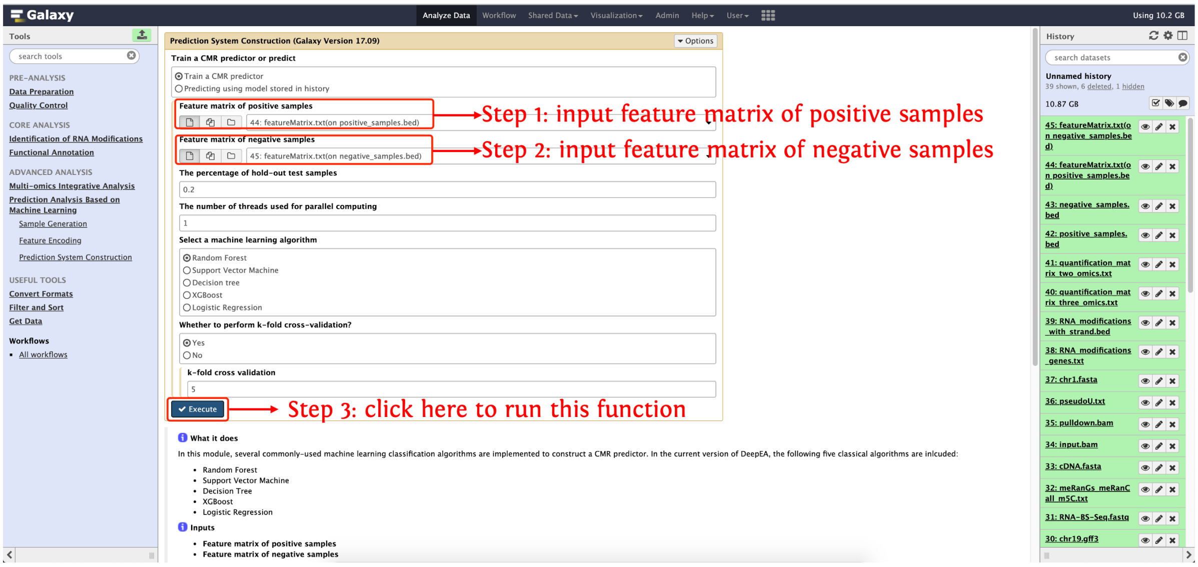
- **Feature matrix of positive samples**
- **Feature matrix of negative samples**

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



Prediction System Construction (Galaxy Version 17.09)

Train a CMR predictor or predict

● Train a CMR predictor
○ Predicting using model stored in history

Feature matrix of positive samples: 44: featureMatrix.txt(on positive_samples.bed)

Step 1: input feature matrix of positive samples

Feature matrix of negative samples: 45: featureMatrix.txt(on negative_samples.bed)

Step 2: input feature matrix of negative samples

The percentage of hold-out test samples: 0.2

The number of threads used for parallel computing: 1

Select a machine learning algorithm

● Random Forest
○ Support Vector Machine
○ Decision tree
○ XGBoost
○ Logistic Regression

Whether to perform k-fold cross-validation?

● Yes
○ No

k-fold cross validation: 5

Step 3: click here to run this function

What it does

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

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

Inputs

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

Unnamed history
39 shown, 6 deleted, 1 hidden
10.87 GB

45: featureMatrix.txt(on negative_samples.bed)
44: featureMatrix.txt(on positive_samples.bed)
43: negative_samples.bed
42: positive_samples.bed
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