PLIT Tutorial

User Manual

To get familiar with the scripts and data, an example dataset has been provided in the "data" directory. This tutorial provides an overview for:

- Extract FASTA sequences from Cufflinks GTF file
- Extract FASTA sequences from CANTATAdb GTF file
- Feature extraction from FASTA sequences
- Extraction of features for training set sequences
- Prediction of lncRNA sequences from test set
- Identification of optimal features using LiRFFS optimization method

Extract FASTA sequences from Cufflinks GTF file

The example dataset does not contain BAM file. However, the file can be obtained from read alignment step in RNA-seq data analysis. The data consists of an R script which extracts the FASTA sequences based on the coordinates (comma delimited file) obtained from GTF file.

- First step is to create a VCF file, namely, 'VCF-calls.vcf' using samtools and bcftools.
- Second step is to create an index using Tabix.
- Input files required are: Genome FASTA file and BAM file. The command below takes the Tophat2 generated BAM file as input. Please note that the genome must be indexed before running these commands. To index the genome, just type: *samtools faidx ref.fasta*
- Follow the steps as mentioned below:

```
samtools mpileup -uf genome.fa accepted_hits.bam | bcftools
call -mv -Oz -o VCF-calls.vcf.gz

tabix calls.vcf.gz
```

Filter the Gene Transfer Format (GTF) file for extracting consensus FASTA sequences from RNA-seq data.

- Run the following commands for filtering the GTF file:

  bash filterGTF.sh merged.gtf

  Rscript filterGTF.R merged-cols.txt

- Extract the FASTA sequences using reference genome and VCF-calls.vcf.gz files:

Rscript extractFASTA.R merged-cols1.csv genome.fa VCF-calls.vcf.gz output-file.txt

- Remove FASTA sequences containing 'NA' or 'NNN…NNN' bases:

bash filterOutputFASTA.sh output-file.txt

Extract FASTA sequences from CANTATAdb GTF file

For extraction of FASTA sequences from any plant species from CANTATAdb database, steps outlined below should be followed. For example, for extracting the FASTA sequences from *Glycine Max* plant GTF file, follow the steps below:

- http://cantata.amu.edu.pl/DOWNLOADS/Glycine_max_lncrnas.gtf
- Using Glycine Max plant data – "Glycine_max_lncrnas.gtf"
- Run the following commands for filtering the GTF file:

  bash filterGTF_CANTATAdb.sh Glycine_max_lncrnas.gtf

- Extract the FASTA sequences using reference genome and VCF-calls.vcf.gz files:

Rscript extractFASTA.R merged-cols1.csv genome.fa VCF-calls.vcf.gz output-file.txt

- Remove FASTA sequences containing 'NA' or 'NNN…NNN' bases:

bash filterOutputFASTA.sh output-file.txt

Feature extraction from FASTA sequences

The example dataset consists of *Solanum tuberosum* FASTA sequences. The dataset consists of coding sequences, noncoding sequences and test set sequences:

- Coding sequences: ST_mRNA_500-1.fasta
- Noncoding sequences: ST_lncRNA_500-1.fasta
- Test set sequences: Test-set-sequences-ST.fasta-7.fa

Open the Feature_extraction directory:

```
cd PLIT/data/Feature_extraction
```

For extraction of features from the test set sequences, execute the following command:

```
bash ../../scripts/PLIT_extractFeatures_testSet.sh -testc
../ST_mRNA_500-1.fasta -testl ../ST_lncRNA_500-1.fasta -fasta
../Test-set-sequences-ST.fasta-7.fa -output ST-features.csv -
scripts ../../lib -b ../../lib -cpat ../../lib
```

The extracted features will be stored as "ST-features.csv" file.

Extraction of features for training set sequences

For extraction of features from training set, coding and noncoding sequences are required.

- Coding sequences: ST_mRNA_500-1.fasta
- Noncoding sequences: ST_lncRNA_500-1.fasta

Open the Feature_extraction directory:

```
cd PLIT/data/Feature_extraction
```

For extraction of features from the training set sequences, execute the following command:

```
bash ../../scripts/PLIT_extractFeatures_trainingSet.sh -coding
../ST_mRNA_500-1.fasta -noncoding ../ST_lncRNA_500-1.fasta -
output ST-features-training.csv -scripts ../../lib -b
../../lib -cpat ../../lib
```

The extracted features will be stored as "ST-features-training.csv" file.

Prediction of lncRNA sequences from test set

For prediction of lncRNA sequences, "PLIT_lncRNAPredict_new.py" script is required. The script requires training and test set feature matrices. For annotation of test set prediction results, test set FASTA file with comments is required:

- Training set matrix: ST-features-training.csv
- Test set matrix: ST-features-noInf.csv
- Test set FASTA sequences: Test-set-sequences-ST.fasta

Open the Feature_extraction directory:

```
cd PLIT/data/Feature_extraction
```

For prediction of lncRNA sequences in test set, execute the following command:

```
python3 /home/sumukh/Downloads/Framework/PLIT_lncRNAPredict.py
-tr ST-features-training.csv -te ST-features-noInf.csv -tef
Test-set-sequences-ST.fasta -o ST-testset-predict.csv
```

Identification of optimal features using LiRFFS optimization method

The algorithm provides minimal and maximal number of optimal features. The algorithm also provides output training and validation set files with optimal features.

For identification of optimal features, open the following directory:

```
cd PLIT/data/LiRFFS-optimization
```

Two input files are required, training and validation set files:

- Training set: 6-plant-train.csv
- Validation set: 6-plant-validation.csv

Additional parameters used for extraction of optimal features are as follows:

- lambdaL: 0.00001
- lambdaU: 0.1
- lambdaS: 0.00001
- tol: 0.7
- otrMin: optimal-train-features-min.csv (output training set with minimal optimal features)
- oteMin: optimal-test-features-min.csv (output validation set with minimal optimal features)
- otrMax: optimal-train-features-max.csv (output training set with maximal optimal features)
- oteMax: optimal-test-features-max.csv (output validation set with maximal optimal features)

Run the following command for identification of optimal features:

```
python3 ../../scripts/PLIT_LiRFFS.py -tr 6-plant-train.csv -te
6-plant-validation.csv -lambdaL 0.00001 -lambdaU 0.1 -lambdaS
0.00001 -tol 0.7 -otrMin optimal-train-features-min.csv -
oteMin optimal-test-features-min.csv -otrMax optimal-train-
features-max.csv -oteMax optimal-test-features-max.csv
```