LIFT 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:

- Extraction of features for lncRNA prediction for test set sequences,
- Extraction of features for training set sequences,
- Prediction of lncRNA sequences from test set,
- Identification of optimal features using LiRFFS optimization method,
- Sub-classification of lncRNA sequences using PBC method,
- Function prediction of lncRNA sequences.

Extraction of features for lncRNA prediction for test set 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 LIFT/data/Feature_extraction
```

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

```
bash ../../scripts/LIFT_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 LIFT/data/Feature_extraction
```

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

```
bash ../../scripts/LIFT_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, "LIFT_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 LIFT/data/Feature_extraction
```

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

```
python3 /home/sumukh/Downloads/Framework/LIFT_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 LIFT/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/LIFT_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
```

Sub-classification of lncRNA sequences using PBC method

The sub-classification of lncRNA sequences is performed by "`LIFT_annotateLncRNAs.py`" script. The script classifies the lncRNA sequences based on positional coordinates. The script requires lncRNA and protein-coding sequence files as input files. The example dataset for lncRNA sub-classification is contained in `sub-classification` folder.

For identification of optimal features, open the following directory:

`/home/sumukh/Downloads/LIFT/data/sub-classification`

The example dataset consists of:

- Protein-coding sequences: Test-proteincoding-seqs-ATH-1000.csv
- lncRNA sequences: Test-lncRNA-seqs-ATH.csv

Run the following command for sub-classification of lncRNA sequences:

```
/usr/bin/python2.7 ../../scripts/LIFT_annotateLncRNAs.py -c
Test-proteincoding-seqs-ATH-1000.csv -n Test-lncRNA-seqs-
ATH.csv -o test.out -o1 test1.out
```

Function prediction of lncRNA sequences

For prediction of molecular and regulatory functions of lncRNA sequences, lncRNA and protein-coding genes containing FPKM expression values are required.

The FPKM expression values of lncRNA and protein-coding genes should consist of following format:

- Rows: lncRNA/protein-coding genes
- Columns: FPKM expression values

Below is the description of an example data demonstrating FPKM expression values for lncRNA genes:

genename	Sample.7_FPKM	Sample.8_FPKM		Sample.9_FPKM		Sample.10_FPKM		Sample.11_FPKM		
	Sample.12_FPKM	Sample.13_FPKM		Sample.14_FPKM		Sample.15_FPKM		Sample.16_FPKM		
AT1G10682	979.863	1034.64	1045.23	332.552	522.443	319.876	62.5159	58.979	89.3898	122.328
AT1G15405	603.504	844.275	1610.56	153.374	2679.75	1504.26	396.894	4341.06	250.661	385.227
AT1G49952	0	1.78691	0.834141	1.64234	1.22931	5.55902	10.1756	5.12623	8.38147	14.679
AT1G60545	3.48991	17.0331	21.4972	30.8856	18.0763	34.8773	38.504	24.4306	31.5998	19.5693
AT1G70185	0	2.34902	7.30485	2.09184	8.29688	1.05299	1.15148	0.559789	0.641896	0.678711
AT1G76892	15.2028	5.70073	4.61037	2.76878	2.15998	2.27514	1.32862	0.576894	0.704693	0.698675
AT1G78265	31.9758	8.55873	9.98584	8.59431	6.0147	10.1304	11.5116	9.00249	8.48531	11.3246
AT1G78865	25.9639	24.2719	19.6561	52.4888	13.1994	63.8395	83.6622	29.2235	39.0764	56.2822
AT1G79075	1876.28	944.826	885.469	501.66	1285.5	441.359	406.494	296.719	468.302	411.804

First step in prediction of lncRNA function is data normalization

Step-1: Data normalization

The data is first normalized (scaled) between 0 and 1 to generate relative expression values for calculation of correlation between the expression values of lncRNA and protein-coding genes. The data normalization is performed by "`normalizeLPExpression.R`" file.

For normalization, open the following directory:

`/home/sumukh/Downloads/LIFT/data/ Function-prediction`

The example input files which are required are:

- lncRNA expression values: lncRNA-DGE-FPKM-values.txt
- Protein-coding expression values: proteincoding-DGE-FPKM-values.txt

Run the following command for normalization of lncRNA and protein-coding expression values:

```
Rscript ../../scripts/normalizeLPExpression.R lncRNA-DGE-FPKM-
values.txt proteincoding-DGE-FPKM-values.txt
```

The script produces "set3t-full.csv" and "set4t-full.csv" output files.

Next step is the computation of lncRNA-protein co-expression similarity (LPCS) matrix.

Step-2: LPCS matrix computation

The script "`computeLPCSmatrix.py`" computes the correlation between the expression values of lncRNA and protein-coding sequences with higher positive and negative correlations. The script requires "set3t-full.csv" and "set4t-full.csv" output files as input for computing the correlations.

Run the following command for computing the positive and negative correlation values:

```
/usr/bin/python2.7 ../../scripts/computeLPCSmatrix.py -c
set4t-full.csv -n set3t-full.csv -o lp-correlation.txt
```

Step-3: Prediction of lncRNA functions using lncRNA-protein correlation and protein-protein interaction data

The protein-protein interaction (PPI) data is obtained from the STRING database. The folder consists of "PPI-matrix.txt" file which consists of example protein-protein interaction data from *Arabidopsis thaliana* species.

Concatenate the LPCS matrix and PPI matrix to generate L-P-P-P-matrix file.

```
cat lp-correlation.txt PPI-matrix.txt > L-P-P-P-matrix.txt
```

Function prediction is performed by BMRF method which also requires a GOTerm-associated data with protein-coding genes. The folder contains the example file "P-GOTERM-BMRFFILE.txt" demonstrating the data. The example dataset consists of following files:

- L-P-P-P-matrix.txt
- P-GOTERM-BMRFFILE.txt

The function prediction of lncRNA genes is performed by "`predictFunction.R`" script.

Run the following command for computing the probability of association and prediction of GOTerms with lncRNA genes:

```
Rscript ../../scripts/predictFunction.R L-P-P-P-matrix.txt P-
GOTERM-BMRFFILE.txt lncRNAs-predicted-functions-BMRF.txt
```

The script generates predicted functions for lncRNA sequences with associated probability and GOTerms as an output file "`lncRNAs-predicted-functions-BMRF.txt`".

Step-4: Annotation and filtering of lncRNA sequences based on probability value

The output file "`lncRNAs-predicted-functions-BMRF.txt`" obtained from Step-3 is used for filtering the sequences based on probability cutoff value. The script "`appendFunction.R`" filters the data based on probability cutoff value and annotates the sequences using protein-coding annotation file as input. The script requires following parameters as input:

- Output prediction file: lncRNAs-function-pred-DGE.txt
- Protein-coding GO annotation file: ATH_GO_GOSLIM-for-BMRF.txt
- lncRNA identifiers: lncRNA-IDs.csv
- Cutoff value: 0.8 (default)
- Output file: ATH-pred-function-GOTerm.txt

Run the following command for annotation and filtering of the prediction lncRNA function output file:

```
Rscript ../../scripts/appendFunction.R lncRNAs-function-pred-
DGE.txt ATH_GO_GOSLIM-for-BMRF.txt lncRNA-IDs.csv 0.8 ATH-
pred-function-GOTerm.txt
```