DOCUMENTATION

No Promoter Left Behind (NPLB) is used on a fasta file containing a list promoter sequences. It finds the optimal number of promoter architectures (PAs) each with their own set of promoter elements (PEs). It can be downloaded from `https://github.com/computationalBiology/NPLB/`.

NPLB can be used to learn new models using the promoterLearn command and identify new PAs using an existing model with the help of promoterClassify command.

Both promoterLearn and promoterClassify automatically saves an HTML file (*PAlogo.html*) consisting of information about the best model(only a single model is learned in case of promoterClassify) and sequence logos for each PA of the model. It creates the logos using a modified form of Weblogo 3.3.

Apart from the HTML file, promoterLearn and promoterClassify save information about the best model in a text file *modelOut.txt*. It saves the PA labels assigned to each sequence in a file *architectureDetails.txt*. It saves HTML files consisting of similar information for each fold of every model learned.

An image matrix for the clustered (*PAimage.png*) and unclustered (*rawImage.png*) data are also saved. promoterLearn can also save the likelihood plot for each fold of every model learned when run in verbose mode. These plots are helpful in displaying whether the likelihood is maximized. When learning a new model, promoterLearn saves the execution details in a file *settings.txt*.

The best learned model is saved in binary format as *bestmodel.p*. This can then be used later on to determine PAs of another set of promoter sequences. A tab separated text file containing information about the promoter sequences and arranged in the order of the fasta file, and a column number (of the given file) as can also be given as input to plot a pie chart or boxplot based on the learned model depending on the type of data in the column. A column number of the file, consisting of a list of real numbers, can be taken as input and can be used to arrange the PAs based on the median values calculated for each PA.

NPLB can be run with several options in order to optimize the execution time and the results. Note that promoterLearn learns models by varying number of PAs in a method similar to binary search since the cross validation likelihood for model with given number of PAs increases till a particular value and decreases again. Hence, it does not learn models by linearly varying the count of PAs.

Prerequisites

The following packages need to be installed in order to run promoterLearn and promoterClassify:

- Python 2.6+ (Not compatible with Python 3.x)
- python-numpy
- python-ctypes
- python-multiprocessing

- python-re
- python-pickle
- gnuplot 4.6+
- Truetype fonts

Installation

NPLB is freely available at `https://github.com/computationalBiology/NPLB/`. Execute the following commands to download and install NPLB:

```
wget ”https://github.com/computationalBiology/NPLB/archive/v1.0.0.tar.gz”
tar -xvf v1.0.0.tar.gz
cd NPLB-1.0.0/NPLB/
make
```

To execute promoterLearn and promoterClassify from anywhere export the path to NPLB to the PATH variable:

```
export PATH=/path/to/NPLB/:$PATH
```

Note: gnuplot font path should be set to the directory containing truetype fonts. It can be set as:

```
export GDFONTPATH=/path/to/truetype/fonts/directory/:$GDFONTPATH
```

Usage of promoterLearn

To learn new models using promoterLearn:

```
/path/to/NPLB/promoterLearn [options]
```

Options

The various options for running promoterLearn are as follows:

- -f filename

  Compulsory. Data file for which hidden PAs and their corresponding PEs are to be identified. File must be in fasta format and must consist of sequences of equal length.

- -o directory

  Valid directory name. If it exists then a new one is created with given name along with an extension number. Default directory: NPLBoutput<extension> in the current working directory.

- -a positive real number

  Pseudocount. Default value: 1.

- -t times

  Maximum number of iterations over the entire dataset. Default: It iterates at most 1000 times and then loops until the likelihood stops increasing for 10*n consecutive iterations where n is the sum of number of sequences and number of PAs times length of sequences.

- -i 0 or 1

  Flag to save image matrix in the given directory. Default value: 1

- -v 0 or 1

  Flag to save likelihood plots of the models learned in each fold.

- -kfold k

  Value of k in K-fold cross validation. Default value: 5

- -lambda l

  Non negative real number. It is the regularization parameter used for finding the PEs of all PAs. Default value: Learns models by varying lambda value and chooses the best one.

- -lcount lc

  Number of models to be learned while training. Only the best model is considered. Default value: 5

- -minarch mn

  Minimum number of PAs possible for the given dataset. Default value: 1

- -maxarch mx

  Maximum number of PAs possible for the given dataset. Default value: 20

- -proc pr

  Maximum number of processors to be used for computation. Default value is the number of processors the system has.

- -plotExtra <file name>

  Plots data from given tab separated file as pie charts or boxplots (depending on the type of data) for each PA of the best model. Note: -pCol must also be set.

- -pCol pc

  Natural number. Plots the data in the given column of the given file in the form of pie charts or boxplots with respect to the PAs of the best model. Note: -plotExtra must specify a valid filename

- -sortBy sb

  Natural number. Sort PAs in increasing order of median values calculated from values in column -sortBy of file -plotExtra. Note: -plotExtra must specify a valid filename and -pCol must be set

Examples

The following examples illustrate the usage of the options.

- To run with all the default options:

  ```
  promoterLearn -f example.fa
  ```

- To run for number of PAs ranging from 2 to 5 with 100 iterations and save results in directory Output:

  ```
  promoterLearn -f example.fa -minarch 2 -maxarch 5 -t 100 -o Output
  ```

- To run it for number of PAs ranging from 15 to 25 and save likelihood plots:

  ```
  promoterLearn -f example.fa -minarch 15 -maxarch 25 -v 1
  ```

- To plot pie charts in fourth column of file example.info and rearrange PAs according to the sixth column:

  ```
  promoterLearn -f example.fa -plotExtra example.info -pCol 4 -sortBy 6
  ```

- To find the list of options:

  ```
  promoterLearn
  ```

Usage of **promoterClassify**

To use an existing model to learn new PAs and their corresponding PEs using promoterClassify:

```
/path/to/NPLB/promoterClassify [options]
```

Options

The various options for running promoterClassify are as follows:

- -f filename

  Compulsory. Data file for which hidden PAs and their corresponding PEs are to be identified. File must be in fasta format and must consist of sequences of equal length.

- -m model

  Compulsory. Model learned by executing promoterLearn on a given dataset. Model must have sequences of the same length as the input fasta file.

- -o directory

  Valid directory name. If it exists then a new one is created with given name along with an extension number. Default directory: NPLBoutput<extension> in the current working directory.

- -i 0 or 1

  Flag to save image matrix in the given directory. Default value: 1

- -plotExtra <file name>

  Plots data from given tab separated file as pie charts or boxplots (depending on the type of data) for each PA of the best model. Note: -pCol must also be set.

- -pCol pc

  Natural number. Plots the data in the given column of the given file in the form of pie charts or boxplots with respect to the PAs of the best model. Note: -plotExtra must specify a valid filename

- -sortBy sb

  Natural number. Sort PAs in increasing order of median values calculated from values in column -sortBy of file -plotExtra. Note: -plotExtra must specify a valid filename and -pCol must be set

Examples

The following examples illustrate the usage of the options.

- To find labels for dataset example2.fa from model present in directory Output as bestmodel.p:

  ```
  promoterClassify -f example2.fa -m Output/bestmodel.p
  ```

- To find labels for dataset example2.fa from model bestmodel.p and plot pie chart or boxplot for column 4 of example3.info:

  ```
  promoterClassify -f example2.fa -m bestmodel.p -plotExtra example3.info -pCol 4
  ```

- To find the list of options:

  ```
  promoterClassify
  ```