STAP: Sequence To binding Affinity Prediction
Xin He <xinhe2@illinois.edu>
10/28/2009

Description

The program use a biophysical model to analyzes transcription factor (TF)-DNA binding data, such as ChIP-chip or ChIPSeq data. The program assumes that the measured affinity of a sequence to a TF (TF_exp) in some ChIP-chip or ChIP-seq experiment is determined by: 1) the number and strength of binding sites of TF_exp in this sequence; 2) the presence of other sites that may interact cooperatively with the sites of TF_exp in the neighborhood. Specifically, it takes as input a set of DNA sequences, their binding affinities to some TF as measured by experiments (TF_exp), and the position weight matrices (PWMs) of a set of TFs, including TF_exp. It will learn the relevant parameters of the biophysical model of TF-DNA interaction, including those of TF-DNA interaction and those of TF-TF cooperative interactions. The program can be used for several purposes:
(1) Test if a given TF binding motif can predict the binding affinities of the sequences. It predicts the binding sites based on this motif, and computes the theoretical values of the binding affinities of the sequences. The predicted values will be compared with the observations to judge the sucess of the model.
(2) When multiple motifs are given as inputs, the program assumes the first motif is the one of the experimetnal TF (TF_exp), and the rest are the motifs that may cooperatively interact with TF_exp (meaning that the adjacent sites of other factors can facilicate DNA binding of TF_exp). The program will learn which motifs are likely to interact cooperatively with TF_exp.
(3) Once a biophysical model is learned, it can be applied to predict affinities of sequences not used in training the model. This would be useful, for example, for analyze sequences in a different organism.

Installation

The program needs GNU Scientific Library (GSL). If it is not installed in your system, go to:
http://www.gnu.org/software/gsl/
Note that after installing GSL, you need to change the start-up script of your shell, e.g., .bash_profile at your home directory if you are using bash. Suppose the GSL installati
on directory is /raid/apps/gsl-1.8/lib:

```
LD_LIBRARY_PATH=/raid/apps/gsl-1.8/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

After extracting the program, change the GSL directory in src/Makefile, e.g.:
GSL_DIR = my_gsl_dir

Then simply type:
make

Running the program

Usage:
./seq2binding -s <seqFile> -d <dataFile> -m <motifFile>

The program takes three arguments as input:

seqFile: the FASTA format file of sequences. See examples/Nanog_top_500.fa.

```
>chr1:136351629-136351631 136351630 -250 +250
gtggtgatgcccaaccacagaattattttgttgctactttataactgtaattttgatcct
>chr3:122137593-122137593 122137593 -250 +250
atttctagttccagtgactgggagactgaaacaagagagtcacttgagtacaggagtgca
```

dataFile: the binding data of all sequences in the seqFile. The first column is the sequence id (must be the same as those used in seqFile, and in the same order), and the second column is the measured strength of binding. See examples/Nanog_top_500.txt.

```
chr1:136351629-136351631        312
chr3:122137593-122137593        307
```

motifFile: the motifs of the TFs. It could contain multiple motifs. The header line consists of motif name, length and pseudocount (0.5 should be OK for most motifs). The first motif should be the one of TF_exp, and the rest are putative TFs that interact cooperatively with TF_exp. See examples/Nanog_Oct4_Sox2.wtmx and examples/Nanog.wtmx.

```
>Nanog  9       0.5
20      225     46      209
70      0       19      411
50      66      381     3
434     45      0       21
55      5       66      374
17      32      222     229
74      18      325     83
8       243     146     103
48      145     6       301
<
```

Output:
(1) Estimated parameters: binding parameter (how strongly the TF binds with its binding site); the interaction parameters between any pair of TFs (the order of motifs in the

matrix follows the order defined in motifFile): greater than 1 if favorable interaction, less than 1 unfavorable, 1 if no interaction.
(2) Pearson correlation between predicted binding and observed binding.

Examples:
(1) Run with a single factor: test if the provided Nanog motif explains the binding of top 500 Nanog sequences in experiments: (under examples/ directory)
../src/seq2binding -m Nanog.wtmx -s Nanog_top_500.fa -d Nanog_top_500.txt

(2) Run with multiple factors (TF_exp, and other factors): test if Nanog interacts cooperatively with Oct4 and Sox2 in the top 500 Nanog sequences: (under examples/ directory)
../src/seq2binding -m Nanog_Oct4_Sox2.wtmx -s Nanog_top_500.fa -d Nanog_top_500.txt

Advanced options

-ts <testSeqFile> -td <testDataFile>: test the trained model in additional testing data. The format of testSeqFile and testDatafile is the same as seqFile and dataFile.

-n <nExps>: the number of experiments being analyzed. The default value is 1, i.e. only one experiment (binding data of one TF) is analyzed. When analyzing binding data of multiple TFs, set nExps as the number of TFs. In this case, it is assumed that seqFile and dataFile contain the concantenation of data of multiple factors (assume the number of records of each TF is equal, thus no explicit delimiter is needed between data of different TFs).

-cv <K>: K-fold cross validation, report the average performance (correlation)

-p <trainPredictionFile>: print the predicted binding intensities (of the training sequences in seqFile) in the file trainPredictionFile.

-co coopOption: the option of cooperative binding. 0 - no cooperativity at all; 1 - no self-cooperativity, but hetero-cooperativity; 2 - allow all cooperativities

-io interactionOption: the option for modeling factor-factor interaction. 0 - binary; 1 - linear; 2 - periodic.

-dt <d_max>: the maximum distance of interaction (beyond which there will be no interaction)

There are other parameters that control the factor-factor interaction model. The file utils/run_pair.sh contains examples of using these parameters. In most cases, you probably do not need to set these parameters.

Utilities

In utils/ directory, some useful scripts are included. However, not all of them can be ready to execute in your system. Some of them are included only for the purpose of demonstrating the use of program.

run_pair.sh: demonstrate the use of program (for analyzing cooperative interactions using binding data of two TFs).

create_null_distr.sh: suppose we want to find cooperative factors of one TF (TF_exp). First run the program using TF_exp and the test motif, and obtain the correlation coefficient (CC) of the test motif. Then run this script to get the null distribution of the CC. The script will sample random motifs from a specified collection of motif, and calculate CC of the random motifs.

shuffle_wtmx.pl: random shuffling of a motif, used by create_null_distr.sh.

split_wtmx.pl: split a file of many PWMs into multiple files, each of which contains a single motif.