REEFFIT Documentation

Release 0.5

Pablo Cordero G.

February 13, 2014

REEFFIT (RNA Ensemble Extraction From Footprinting Instights Tool) is a method to fit RNA secondary structure ensembles to multi-dimensional chemical mapping data. Currently, the method can take data from one and multiple mutate-and-map or mutate-bind-and-map (mutate-and-map plus ligand titrations, e.g. for riboswitches) experiments. Output is a set of weights and expected reactivities for each structure that linearly combine to form the data. If the structural ensemble is not provided, REEFFIT uses RNAstructure to generate a suboptimal set of structures of all mutants.

CHAPTER 1

Technical summary

REEFFITs core framework is essentially non-negative factorization with a Gaussian error model (i.e. a form of factor analysis). In this framework, the data is modeled as a linear combination of positive hidden variables:

$$D_{obs} = W D + \epsilon$$

Where D_{obs} is the data, W are the weights and D are the hidden variables; ϵ is a noise term that has mean 0 and position-wise variance values $\Psi_i, \forall i = 1, ...n$. In standard factor analysis, the hidden variables are normally distributed, and the likelihood function to obtain the weights and the noise covariance matrix is maximized using the EM algorithm. Fortunately, when the covariance matrix is assumed diagonal (that is, the measured variables are not well correlated), then the E and M steps of the EM algorithm can be written in closed form. However, we have a different, more complicated prior on the hidden variables D, since in our case these variables are the chemical reactivities for a given structure that are expected to be drawn from a chemical reactivity distribution shaped by the structure. To simplify things, we use chemical reactivity distributions obtained from the RMDB database (http://rmdb.stanford.edu) splitted into two classes: distributions for unpaired and paired nucleotides. Because these priors on D are far from normal, we cannot use the standard factor analysis EM-algorithm solutions; in fact, the likelihood function derived from the E-step cannot be calculated analytically. Instead, we either use Bayesian inference (Markov Chain Monte Carlo simulations) or maximum a posteriori estimation to solve the optimization problem. The M-step is much simpler, as it can be solved as a quadratic optimization problem with convex constraints (the weights need to be positive and sum to one). For error estimation, we use bootstrapping, although REEFFIT is able to calculate stnadard errors withouth bootstrapping as well.

CHAPTER 2

Contents of this documentation:

2.1 Getting started with REEFFIT

2.1.1 Requirements

REEFFIT is written in python (routinely tested in 2.7.3) and requires the following python packages:

- Scipy (0.9 or greater)
- Numpy (1.6.1 or greater)
- Matplotlib (1.1.1 or greater)
- PyMC (2.2 or greater)
- joblib (0.5.4 or higher)
- CVXOPT (dev. releast)
- RDATkit (dev. version, see https://github.com/hitrace/rdatkit)

It is recommended that these dependencies (except joblib and RDATkit) be obtained via the enthought package (via canopy).

2.1.2 Installation

To install REEFFIT, follow these steps:

- Go to REEFFIT's home directory and type: `python setup.py install`
- In your profile (e.g. .bashrc), include an environment variable REEFFIT_HOME that points to the REEFFIT home directory. For example: `export REEFFIT_HOME=/path/to/reeffit`
- Add the `REEFFIT_HOME/bin` directory to your path: `export PATH=$PATH:REEFFIT_HOME/bin`
- Check that REEFFIT is correctly installed. Execute the reeffit command by running `reeffit` in your shell.

2.1.3 Running REEFFIT

Almost all of REEFFIT's functionality is accessed through the `reeffit` command, located in the `REEFFIT_HOME/bin` directory that was added to the PATH variable during installation. There are four steps that are usually performed in a REEFFIT analysis, all of which are accessed through the `reeffit` command: obtaining

a secondary structure ensemble, preparing cross-validation or bootstrapping shell files, executing the shell files, and compiling the results. The following subsections explain how to run each.

General form of the `reeffit` command

In each of the following analyses, the `reeffit` command is used in various ways. REEFFIT's main input is an RDAT file containing the data and details of the chemical mapping experiment. Besides this, the only other necessary option is the output directory. The general form of the `reeffit` command is:

```
reeffit RDAT_FILE OUTPUT_DIR [OPTIONS]
```

The options given to REEFFIT will dictate its behavior.

Obtaining the structural ensemble

If a hypothesized secondary structure ensemble is not available, REEFFIT can estimate it using one of several methods:

- Sampling of the suboptimal ensemble for all sequences (option `--modelselect=sample`): Up to 200 suboptimal structures that lie in at most 5% distance to the minimum free energy secondary structure are sampled for each sequence in the ensemble. The full resulting set of structures (usually on the order of 200 to 1000 for RNAs of 100 to 200 nucleotides in length), located in `OUTPUT_DIR/all_structures.txt` are used for subsequent analyses.
- Selection of a small number of structures in the suboptimal ensemble that best agree with the data a priori (option `modelselect=heuristic`): The same procedure as in the point above is used to obtain a large number of suboptimal structures. These are then clustered into a small number of clusters (calculated by maximizing the Calinksi-Barabasz index) and representatives for each cluster are chosen by scoring them against each chemical mapping profile in the data using 1-dimensional mapping secondary struture directed modeling. Additional high-scoring structures are added if they decrease the AIC information score after a few EM iterations.

The recomended method to obtain the structural ensemble is the first one: using all the structures sampled. We have observed that this captures most of the data better and can be properly regularized with several priors to account for the large number of variables to fit (see below). Therefore, to obtain the structural ensemble, the reeffit command would be:

```
reeffit RDAT_FILE OUTPUT_DIR --modelselect=sample
```

Important options for fitting the ensemble to the data

There are several options that affect the way REEFFIT performs the fit to the data. The most important ones to consider are the ones below:

- Number of parallel jobs (option `--njobs`): Some of REEFFIT's calculations can be performed quickly in parallelized form. This option can specify the number of jobs to split the calculations into.
- Number of EM iterations (option `--refineiter`): Number of maximum EM iterations to perform. The default is 10.
- Mode of inference for the E-step (option `--softem` to use soft EM (MCMC inference) instead of the default hard EM (MAP estimation) mode): The most rigorous way to run REEFFIT is using the `--softem` option to perform MCMC inference at each E-step in the calculation (set the number of simulation steps to take with the `--nsim` option, which defaults to 1000). This however, is terribly costly computationally and can come at an expense of other important downstream analyses, like bootstrapping. In our experience, performing hard EM does not alter the results terribly and can therefore be safely used.

- Motif decomposition (option `--decompose`, default is no decomposition): When the ensemble to fit is large, it may be useful to decompose the structures into overlapping secondary structure motifs, forcing all motifs that are structurally the same to have the same reactivity profile. This greatly reduces the number of variables in the model and speeds up the computation. Note that this is not activated by default.
- Data normalization (option `--boxnormalize`): When handling capillary electrophoresis chemical mapping data that has not been rigorously normalize (through, for example, a dilution series), it is recommended to normalize the data to conform to the prior reactivities coded into REEFFIT. This option should box the data into values from 0 to 2 and get rid of outliers. Note that if normalization has been previously peformed on the dataset, this additional normalization will produce large artifacts and will essentially "flatten" the data.

Preparing bootstrap files

In order to robustly estimate population fraction errors, we perform boostrapping. Because bootstrapping is computationally expensive, we recommend using the `reeffit` command to prepare shell "worker" scripts that will perform the bootstrapping in parallel. To achieve this, we use the `--preparebootstrap` option. To set up the scripts, we have to divide the number of bootstraps into the number of worker scripts that are going to be running simultaneously using the `--nworkers` and `--ntasks` options. For example, to set up 100 bootstraps for 5 workers, we would use the options `--nworkers=5 --ntasks=20`. It is important to note that every option passed to the command will be passed to the worker scripts. Assuming that we have the structural ensemble in `OUTPUT_DIR/all_structures.txt` and we want to activate motif decomposition, the REEFFIT command for this task would be:

```
reeffit RDAT_FILE OUTPUT_DIR --structfile=OUTPUT_DIR/all_structures.txt --decompose --preparebootstra
```

This will write several `bootstrap_workerN.sh` scripts to the output directory, as well as a `master_bootstrap_script.sh`

Executing the boostrap workers and compiling results

Once the bootstrapping files are set up, we can execute them and compile the results using the master script. To execute the bootstrap workers prepared in the section above, execute the generated master script:

```
sh OUTPUT_DIR/master_bootstrap_script.sh execute
```

All workers will then execute in parallel and store their results in `OUTPUT_DIR/bootN` directories. After the workers are done, you can compile their results using the master script as well:

```
sh OUTPUT_DIR/master_bootstrap_script.sh compile
```

This will take some time, since it will do a REEFFIT fit will the full data in addition to compile the bootstrapping results.

Generating a PDF report

Optionally, REEFFIT can produce a PDF report of the bootstrap results. This is achieved with the `reeffit_report` command, which is added to your PATH variable during installation:

```
reeffit_report OUTPUT_DIR NAME PREFIX
```

Here, the NAME option is just to give a name to output structures in the report. The PREFIX option specifies which result files REEFFIT will use to generate the report. For example, all result files in the bootstrap analysis by default start with the bootstrap prefix. Therefore, to generate a report using the bootstrap results, PREFIX would be set to bootstrap.

CHAPTER 3

Contact

If you have any questions please contact us:

- Pablo Cordero: tsuname at stanford dot edu
- Rhiju Das: rhiju at stanford dot edu

CHAPTER 4

License

This project is licensed under the GNU Public Licence.