

---

# BINDING SITE GRAPHS README

## OVERVIEW

The included code is a collection of programs used to construct and analyze Binding Site Graphs. Overall, the analysis is a two step process that can be performed on separate computers. First, ensemble Gibbs sampling is performed, preferably on a massively parallel computer, and the results are collected. The resulting output files are then compressed (using gzip) to conserve disk space. Then, in a second step, the compressed output files are analyzed and predictions are made, as described in (Reddy, DeLisi et al. 2007)

The main workflow I use here is as follows. Bold indicates major analysis steps. These steps are described in detail below.

- 1) Collect promoter sets and an appropriate background file (commonly genomic intergenic sequences).
- 2) Dust filter the input promoter set (I use "dust" from the wublast package)
- 3) Run the Ensemble Gibbs sampling.**
- 4) Gzip all the result files, and optionally transfer them onto an analysis computer.
- 5) Perform Binding Site Graph analysis to predict transcription factor binding sites. ("eval\_basic.sh")**
- 6) Align the predicted binding sites (using sampling or EM) to generate a PWM for the TF.

## ENSEMBLE SAMPLING PROCEDURE

The ensemble sampling procedure is the most computationally intensive aspect of the Binding Site Graph analysis, and is best performed on massively parallel supercomputers such as the BlueGene Machines. The sampling itself is performed in a process similar to that of BioProspector (Liu, Brutlag et al. 2001). The code uses MPI to divide processes on separate computers. As a general framework, each Gibbs sampling iteration returns the best scoring result out of 60 random restarts. The process of selecting the best result is repeated many times, and all the results are collected together and written to an output file. To perform the sampling process in parallel, the processors are divided into groups of 4 CPUs, and each group has a group leader. Additionally, a master node is designated. The master node is responsible for handing out requests for Gibbs sampling results to each group leader. Within each group, the 4 nodes compute the 60 Gibbs sampling iterations in parallel, and report their results to the group leader. The group leader, in turn, selects the best result from the group's iterations, and reports back to the master node. The master node compiles results from all the group leaders, and writes the results to disk.

The main executable that performs the ensemble sampling procedure is "gibtigs\_tester.rts". (The name is a vestige of a previous version of the code). Makefiles are provided for compilation on a couple different systems, although they will likely require modification for non-BlueGene/L platforms. The code is divided into two directories. The "gibtigs/libfindmotif/" subdirectory contains all the Gibbs sampling code. The main directory, "gibtigs/", contains other iterative Gibbs sampling code.

The Ensemble sampling takes 3 arguments: (1) The input file; (2) The background file; and (3) the output file. Their file formats are detailed below:

- (1) The input is a set of input sequences (i.e. promoters) in a restricted FASTA format, in which the DNA sequence is always on a single line. That is, a file containing 3 sequences will have 6 lines, as follows:

```
>sequence 1
ACTAGATAGAG...
>sequence 2
<all of sequence 2 on one line>
>sequence 3
<all of sequence 3 on one line>
```

- (2) The background is a FASTA file in the same format, from which a higher order markov background model is derived. The default is a 3bp background model. Generally, the background contains all promoters from the organism being studied.
- (3) The output file is given as a base filename. One output file is created for each motif width by appending ".wXX" to the end of the base filename provided. Here, XX is replaced by the corresponding Gibbs sampling width. Therefore, for example, if the base output filename is "testset.fasta", then there will be output files of the form "testset.fasta.links.w6", "testset.fasta.links.w7", ..., "testset.fasta.links.w18"

Within each output file is a list of tab delimited links (graph edges) of the format:

```
Sequence1 <TAB> Pos1 <TAB> Sequence2 <TAB> Pos2 <TAB> Sampling # <TAB> Motif Width <TAB> Score
```

where "<TAB>" is actually a tab character (\t). For example, a line that reads:

```
seq1<TAB>10<TAB>seq2<TAB>100 <TAB>1<TAB>10<TAB>30.1
```

indicates that the nucleotide at position 10 in sequence1 is connected to the nucleotide at position 100 in sequence 2. The link occurred in the first Gibbs sampling run using motif width 10. The score for the sampling run was 30.1 (the score is currently ignored, but may be useful later for weighting edges).

The primary reason to write all the links to disk was to conserve memory. However, output files are often very large, and there have been problems with output files exceeding the maximum allowed file size on some computers. A potential compromise is to write directly to compressed (gzipped) files. Currently, this is not implemented because the BlueGene/L platform used does not have gzip libraries available. Instead, the output files are gzipped afterwards in order to save space. As an optional step, the gzipped output files can be transferred to another computer for analysis.

## MOTIF PREDICTION PROCEDURE

The analysis/feature selection process is carried out by a separate set of scripts, found in the "gibtigs/analysis/" subdirectory. These scripts take in the gzipped link file, cluster all the links together into a weighted graph, and then perform all the subsequence analysis that ultimately leads to a set of FASTA files that are the predictions for the set. The analysis is described in detail in Plos Computational Biology (Reddy, DeLisi et al. 2007).

The main script here is "eval\_simple.sh" which takes as input two arguments:

- (1) The base name of the output link files
- (2) The name of the original input FASTA file. The input FASTA file is needed to reconstruct binding site predictions from the link files.

So, for example, if ensemble sampling produced output files of the form "DATASET.fasta.links.wXX.gz", where XX is the motif width, then the base name is "DATASET.fasta.links". The script automatically iterates through each of the motif widths and constructs a graph accordingly. Other arguments are available, but may require modifying the "eval\_simple.sh" script

The workhorse of the "eval\_simple.sh" script is the executable

"group\_links\_sparse\_gzip\_weighted\_cc\_noconf"

for which source code and a simple Makefile is included. This binary depends on the gzip library (standard on unix machines) and the boost graph library, which you may need may need to be installed. There are also some perl scripts which will create the output file, and make a FASTA file of the results. To run the perl scripts, additional perl libraries may be required.

## BIBLIOGRAPHY

- Liu, X., D. L. Brutlag, et al. (2001). "BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes." *Pac Symp Biocomput*: 127-38.
- Reddy, T. E., C. DeLisi, et al. (2007). "Binding site graphs: a new graph theoretical framework for prediction of transcription factor binding sites." *PLoS Comput Biol* **3**(5): e90.