MOtif STAtistic Software Suite v1.1

Utz J. Pape
Max-Planck-Institute for Molecular Genetics
Computational Molecular Biology
Ihnestr. 63-73
14195 Berlin

October 8, 2008

Contents

1 Preliminary Note

All statistics which are computed using this software package assume an i.i.d. sequence. Therefore, as parameter for the sequence model only the GC content is required such that both strands of DNA are symmetric. Thus, before applying these statistics either check that your sequences are indeed i.i.d. sequences (Rice, 1995; Reinert *et al.*, 2005) or be sure that you understand how the background model can influence the results.

2 Compilation

Run for compilation

```
>make all
```

and to clean

```
>make clean
```

3 cstat

Returns the count statistics

Call

```
$ ./cstat <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter> <[regularize]>
. <[output]> <[sequence length]>
```

where the parameters have to be as followed:

- <gc>: gc content, e.g. '.4', for the background model
- <[file:]transfac-file>: file describing the position count matrices (PCM) in transfac format, e.g. 'data/A1.mat'. See data/A1.mat as an example. The program assumes the line tag ID to occur first. Next, it searches for P0, 01, 02 and so on until the next line does not contain the next number. Different PCMs in the file have to be separated by a line only containing '//'. If filename is preceded by list: one can pass a file containing a list of transfac filenames as parameter. Each file of the list is supposed to contain exactly one PCM.
- <threshold-method>: the treshold method:
  - typeI: set threshold such that typeI error is equal to threshold-parameter.
  - typeII: set threshold such that typeII error is equal to threshold-parameter.
  - balanced: set threshold such that typeI error equals typeII, threshold-parameter can be any number (is not used but has to be passed as parameter)
  - typeIext: set threshold to balanced threshold if it's possible such that the probability of a false positive on a sequence length of 500 is less or equal to the threshold-parameter. Otherwise, set typeI error equal to threshold-parameter.
  - threshold: threshold-parameter contains the threshold.
  - nrwords: define the number of words higher than the threshold (is only accurate if gc content is 50%.)
- <[regularize]>: if not set or set to a true value (1), the regularization method from Rahmann et al. 2005 is applied. Otherwise, we just add pseudocounts.
- <output>: if this parameter exist, the running time of the statistical calculation (without reading input/preparing PSSM, and so on) is printed. Depending on the choice of this parameter, following output is printed:
  - parameter: only xi, xi', xi'0, alpha, theta1
  - lambda: in addition: lambda1, lambda2
  - theta: in addition, all thetas until ¡ precision
  - rate: in addition, rate r (give sequence length in next parameter) without theta
  - cpd: in addition, all $P(X \geq x)$ until < precision (give sequence length)
- <[sequence-length]>: length of the sequence

The typeI error is measured as the probability of at least one false positive in a region of length 500 (Pape *et al.*, 2006, α_{500}).

Examples

- Assumes gc-content of 40%, uses matrix given in data/matrixA.mat and sets the threshold to 30.

  ```
  ./cstat .4 data/matrixA.mat threshold 30
  ```

- Iterates over the transfac files given in data/matrix.list and sets for each matrix the threshold such that typeI error is equal to typeII error or (it not possible) the typeI error to .1.

  ```
  ./cstat .3 list:data/matrix.list typeIext .1
  ```

- Returns hit statistic for a sequence of length 10000 with gc-content 40% after setting the threshold for the non-regularized (only pseudo-counts added) such that the type I error is equal to 10%.

  ```
  ./cstat .4 data/matrixA.mat typeI .1 0 cpd 10000
  ```

4 sstat

Returns the similarity between PFMs.

Call

```
$ ./sstat <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. [<partial-execution>] [<return diagonal>] [<bregularize>]
```

Most parameters are the same as for cstat. The new parameters are:

- <partial-execution>: integer i: if not given, whole similarity matrix is computed. if given, only the ith and the n-i th line of the similarity matrix are computed and return in special format (to be read by scluster). if -1 then simstat uses SGE cluster itself.
- <return diagonal>: default: 0 (false). If set to 1, we also return the similarity of each matrix and itself. (Useful for computing the variance for the univariate count distributions.)

Output

Matrix with following columns:

- matrixA: first matrix
- matrixB: second matrix
- Smax: Similarities summarized by using the maximum
- Ssum: Similarity measured by covariance
- imax: Position with the highest similarity (B is shifted against fixed A!)
- bimaxp: maximum similarity is a reverse complementary hit (1) otherwise (0)
- alphaA: probability of a false positive for matrixA
- alphaB: probability of a false positive for matrixB

Examples

Compute similarities between all pairs of matrices from data/matrix.list using a balanced threshold:

```
./sstat .4 list:data/matrix.list balanced .1
```

5 scluster

Returns a clustering of PFMs

Call

```
$ ./scluster <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. [<use-sge>] [<p=.95>] [<LOOCV>] [<regularize>]
```

where most parameters are the same as for cstat. New parameters:

- <use-sge>: 0/1 (standard: 0) uses sge engine to build similarity matrix
- <p>: Two PFMs are considered for merging only if their Smax value is higher than the maximum of the quantile p of all pairwise Smax values and 0.
- <LOOCV>: Performs a Leave-One-Out-Cross-Validation.

Output

Matrix with following columns:

- matrixA: first matrix
- matrixB: second matrix
- QA: power of matrix A
- QB: power of matrix B
- icA: information content of matrix A
- icB: information content of matrix B
- Smax: Similarities summarized by using the maximum
- imax: Position with the highest similarity (B is shifted against fixed A!)
- bimaxp: maximum similarity is a reverse complementary hit (1) otherwise (0)
- Q: power of new matrix
- ic: information content of new matrix

Furthermore, following files are written (in the same directory where the input files are):

- <transfac-file>.matrices: contains all familial binding profiles (cluster representatives) for each cluster including intermediate representatives.
- <transfac-file>.cluster: contains the final familial binding profiles for each cluster and all remaining singletons.

Example

Computes clustering of all pairs of matrices from data/matrix.list using a balanced threshold.

```
$ ./scluster .4 list:data/matrix.list balanced .1
```

6 costat for co-occurences

This program returns the probability (or the rates you need to compute this probability) to have at least one hit of TF A and at least one hit of TF B in a window. The calculation is performed for all pairs of the given set of TFs.

Call

```
$ ./costat <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. [<window size>] [<bregularize>] [<file with empirical alphas>]
```

where most parameters are the same as for cstat. New parameters:

- <window size>: if the parameter is not given, the program only outputs the rates. In case of a given window size, it returns the probability to have at least one hit for A and one hit of B.
- <file with empirical alphas>: instead of using the theoretically derived alphas (probability for a false positive - at one position!), you can supply the empirical probability (count the number of hits and divide by twice of the sequence length due to the complementary strand). This corrects against a bias occuring for unexpected frequent motifs. The file should contain one probability per line in the same order as the matrices in the transfac file.

Output

Output if window length was not given Matrix with following columns:

- matrixA: first matrix
- matrixB: second matrix
- rA: rate for the occurence of matrix A
- rB: rate for the occurence of matrix B
- rAB: rate for the occurence of matrix A and B
- alphaA: typeI error for matrixA
- alphaB: typeI error for matrixB

Output for given window length Matrix with following columns:

- matrixA: first matrix
- matrixB: second matrix
- p: probability to observe at least one hit of matrixA and one hit of matrixB in a window of given length.

Example

For rare rate output:

```
$ ./costat .4 list:data/matrix.list typeIext .1
```

and for probability output with a window size of 500

```
$ ./costat .4 list:data/matrix.list typeIext .1 500
```

7 bsanno for clustering

Annotates sequences with binding sites

Call

```
$ ./bsanno <sequence-file> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. [<bregularize>] [<statistics>] [<gc content for global option>]
```

where most parameters are the same as for cstat. New parameters:

- <sequence-file>: a FASTA file containing sequences for annotation, e.g. data/seq.fasta
- <statistics>: default: false; if true then pvalue per sequence per PFM are reported, otherwise position of binding sites are reported.
- <gc-content for global option>: if this parameter is not set, we use for each sequence annotation for the background model (for PSSM generation and threshold determiniation) the gc content given by the selected sequence (default option!). If a gc content (like .3) is given, we define the background model (for PSSM generation and thresholding) by this given gc content (here .3) for all sequences.

Output

Matrix with following columns:

- matrix: matrix for binding site
- gene: name of gene
- strand: 1 for binding site on given sequences, -1 for reverse complementary strand
- pos.start: starting position of the binding site (ignoring its orientation) counted for upstream sequences (this means we assume the TSS at the end of the given sequences, therefore, the last sequence position is 0, the second last 1, and the first position is the sequence length-1), in fact, we enumerate the sequence positions from right to left.
- pos.end: corresponding ending position of binding site, pos.end¡pos.start always holds.
- seq.start: starting position of the binding site while enumerating the positions from right to left. Thus, first position of the sequence corresponds to 0 and the last position is sequence length-1. Again, we ignore the orientation of the binding site.
- seq.end: ending position of the binding site. seq.end > seq.start.

Example

Annotates sequences in seq.fasta by the binding sites contained in matrix.list using a balanced threshold.

```
./bsanno data/seq.fasta list:data/matrix.list balanced .1
```

8 pfmqual to compute quality of PFMs

Computes the correlation between set of binding sites (given in Transfac file in lines starting with BS <sequence>) and the PFM. Make sure that BS is followed either by a tab and the sequence or by two spaces.

Call

```
$ ./pfmqual <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. [<bregularize>] [<bnr>]
```

where all parameters except <threshold-method> are the same as before.

- <threshold-method>: 'optimize' to optimize the threshold such that quality is maximixed. bnr: if 1 then binding sites are transformed to unique sequences, if 0 (default) we use the binding sites as given in the transfac file.

Output

Matrix with following columns:

- matrix: name of matrix
- ic: information content of PFM
- alpha: probability of at least one false positive on a sequence region of length 500.
- beta: typeII error
- len: length of PFM
- t: threshold

Example

```
./pfmic .4 list:data/matrix.list balanced .1
```

9 Parallel Computing

Some of the programs support parallel computing. Since we support OMP (one memory, multiple processors) and the Sun Grid Engine (multiple memory, multiple processors), we divide this section correspondingly.

9.1 OMP

If you have an OMP ready compiler, you just have to uncomment the two lines in the Makefile:

```
compileoption += -fopenmp
linkoption = -lgomp -o
```

And, perhaps, change the parameter for using openmp fitting to the compiler you use. We are using C++ compiler v. 4.2.0 for 64bit machines. If you have compiled the program with OMP enabled, the clustering will perform much faster in recomputing the similarities of each new representatives with all other nodes.

9.2 The Sun Grid Engine

As the parameters suggested (above), the programs support the Sun Grid Engine - although we have to admit that the implementation is rather proprietary. Anyways, some inspection of the sge.h and sge.cpp should clarify the implementation and give the possibility to extend it. In fact, all classes which can use the SGE engine (CSimilarityMatrix and CClusterMatrix since it is inherited from CSimilarityMatrix but does not need any further adjustment.) are derived from the interface ISGEClient. Two adjustment might be needed:

1. In the client class CSimilarityMatrix you might want to modify the path of sstat. We assume that it is contained in the path - then - you don't need any modifications.
2. In `sge.cpp`, three main task are done - and might need some adjustments:
   (a) Initialization: The constructor of CSGEMaster needs a temporary directory (default: sgetemp). Be aware that each construction might delete files within this directory.
   (b) Job Submission: Implemented in the method submit. Here, you have to change the format/directives and the program to submit the job (default:submit2sge) as well as the queue and other parameters such that they fit you environment.
   (c) Waiting: After the jobs are submitted, the class waits until all jobs are done (method finish()). Here, we use the program qstat to see which jobs are done (using the job id caught at submission) and also perform some basic error handling. Depending on your output of qstat, your error logs and so on, you might want to change some code there, as well. (By the way, if we see that a job was finished successfully, we call the callback function sge_merge (for which we in fact use the interface ISGEClient) to read the output).

10 Website, Citation and License

Website

Check out the website http://mosta.molgen.mpg.de for new revisions and an online version of the program. Send comments, suggestions etc. to utz.pape@molgen.mpg.de.

Citation

Depending on the program you are using, please cite the following articles (and, if possible, give a link to http://mosta.molgen.mpg.de):

- Pape *et al.* (2008*b*): `cstat`
- Pape *et al.* (2008*c*): `sstat` and `scluster`
- Pape & Vingron (2008): `costat`
- Pape *et al.* (2008*a*): `pfmqual`
- Pape (2008): programs which are not listed above

License

References

Pape, U. J. (2008). *Statistics for Transcription Factor Binding Sites.* PhD thesis, Free University of Berlin, IMPRS for Computional Biology and Scientific Computing, Max Planck Institute for Molecular Genetics.

Pape, U. J., Grossmann, S., Hammer, S., Sperling, S. & Vingron, M. (2006) A new statistical model to select target sequences bound by transcription factors. *Genome Informatics,* **17** (1), 134–140.

Pape, U. J., Rahmann, S., Richard, H. & Vingron, M. Quality of Binding Site Representation by Position Frequency Matrices and Threshold Optimization. Submitted to Bioinformatics.

Pape, U. J., Rahmann, S., Sun, F. & Vingron, M. (2008*b*) Compound Poisson approximation of number of occurrences of a Position Frequency Matrix (PFM) on both strands. *J. Comput. Biol.,* **15** (6), 547–564.

Pape, U. J., Rahmann, S. & Vingron, M. (2008*c*) Natural Similarity Measures between Position Frequency Matrices with an Application to Clustering. *Bioinformatics,* **24** (3), 350–357.

Pape, U. J. & Vingron, M. (2008) Statistics for Co-Occurrence of DNA Motifs. In *Proceedings of the 4th International Workshop on Applied Probability*, (Chiquet, J., Glaz, J., Limnios, N. & Moyal, P., eds),.

Reinert, G., Schbath, S. & Waterman, M. (2005) Probabilistic and Statistical Properties of Finite Words in Finite Sequences. In *Applied Combinatorics on Words*, (Berstel, J. & Perrin, D., eds),. Cambridge University Press.

Rice, J. (1995) *Mathematical Statistics and Data Analysis.* Duxbury Press.