Bioinformatics: Practical Application of Simulation and Data Mining
FINAL PROJECT

The final project is due on **May 8th 5PM EST**. Choose from either MCDB&MBB or CBB&CPSC project depending on your academic affiliation. The completed assignment should be emailed to cbb752@gersteinlab.org.

If you have another topic on mind, please talk to Mark Gerstein before **April 18th**.

Late policy: Barring a valid medical reason (with supporting documentation), or sufficient advance notice of a schedule conflict (at least two weeks before the due date), late projects will *not* be accepted.

Plagiarism: Following are documents on Yale's policies on academic integrity, and how to avoid plagiarism:
http://www.yale.edu/graduateschool/academics/forms/Avoiding%20plagiarism.pdf
http://www.yale.edu/graduateschool/academics/forms/integrity_resources.pdf

MCDB & MBB

Choose one of the listed two topics and write an 8-10 pages long research project proposal (Times New Roman, 12pt font, double-spaced), with citations at the end (citations do not count towards the 8-10 pages requirement).

Your proposal should include:

1) A review of literature
2) Objectives and detailed methods for implementing the proposed research project

Suggested format of the proposal:

1) Objectives and significance (1-2 pages)
2) Background (2-3 pages)
3) Methods (4-6 pages, including both experimental and computational)
4) References

Citations should be in the format of either EndNote or Zotero.

NOTE: For students taking only one module (i.e. MB&B 753/754), your final project need only be 4-5 pages long.

1. Use ChIP-seq to study gene transcriptional regulation

ChIP-seq is a powerful method to study transcriptional regulation, including but not limited to investigating the impact of transcriptional factor binding, DNA methylation and histone modification on gene expression. Propose a research project that aims at understanding one specific aspect of transcriptional regulation of your interest. ChIP-seq should be the central, but not necessarily the only method used in your project.

Reference

Park P. J. ChIP-seq: advantages and challenges of a maturing technology. *Nat Rev Genet* (2009)
Pepke S. et al. Computation for ChIP-seq and RNA-seq studies. *Nat Methods (*2009)

2. Use biological interaction networks to understand human disease

A network perspective of human disease has enabled us to understand the mechanisms of those diseases far better than would be possible by studying the individual components of those networks. Propose a research project to better understand human disease using biological interaction networks.

CBB & CPSC

Choose one of the listed two programming assignments to implement. The recommended implementation language is Python (Version 2.7); contact Mark Gerstein and the TFs to discuss other language choices. In your email submission, include:

1) Input file(s)
2) Source code
3) Output file(s)
4) A short README file on how to execute your program, and
5) A 1-3 pages write-up on the algorithm implemented along with references

1. Implement a peak-calling algorithm for ChIP-seq data

Identify enriched regions from ChIP-seq data is an important step in the analysis. Various methods have been published in literature (refer to Table 1 in Pepke S. et al. and Wilbanks E.G. and Facciotti M.T. paper for some examples). Please implement an algorithm for calling peaks. It can follow the algorithm from published methods, or it can be your own design as long as you describe in detail how it works.

Suggested dataset: GSM875384-GSM875387 from data series GSE30641 deposited at GEO (http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE30641). They are H3k27ac (an active histone modification) ChIP experiments and input controls for mouse embryonic limb. The aligned and combined reads for ChIP and control experiments are available at http://archive.gersteinlab.org/course/cbb752b12/cbb752_final_project_ChIPseq_data/ (These files contain reads aligned to mouse chromosome 19 only. "e11.5_limb_h3k27ac_chr19.aligned" contains alignments from the H3k27ac ChIP-seq experiment, and "e11.5_limb_input_chr19.aligned" contains alignments from input control experiment). The format description of these alignment files is at http://bowtie-bio.sourceforge.net/manual.shtml#default-bowtie-output. If you choose to use this dataset, you don't need to include input files in your submission. But you need to specify in your README file how these files were used as your input for the program.

In writing the source code, you may use packages for calculating p-values for statistical distributions (e.g. Binomial, Poisson distributions) and multiple hypothesis correction (e.g. Benjamini-Hochberg, Bonferroni correction). You may also be allowed to use packages for other purposes (e.g. signal transformation), for which permission should be granted by TFs.

The output of your program for reporting peaks should be in the ENCODE broadPeak format (description of the format is at http://genome.ucsc.edu/FAQ/FAQformat.html#format13).

In the write-up on algorithm, please include a detailed description on the choice of algorithm and parameters (e.g. how background was modeled; how to calculate p-values for enrichment; etc.). Also include brief explanations for such choices. Note that the choices of algorithms and parameters are also dependent on the dataset (e.g. ChIP-seq experiment on histone modifications versus transcription factor have different characteristics).

Reference

Park P. J. ChIP-seq: advantages and challenges of a maturing technology. *Nat Rev Genet* (2009)

Pepke S. et al. Computation for ChIP-seq and RNA-seq studies. *Nat Methods (*2009)

Wilbanks E.G. and Facciotti M.T. Evaluation of algorithm performance in ChIP-seq peak detection *PLoS ONE* (2010)

2. Implement a program to determine the *betweenness centrality* of each node in a network
Betweenness centrality is a measure of how central a node *v* is in a network by measuring the number of shortest paths between any pair of nodes that passes through *v*. In other words, if each node were trying to communicate with all the other nodes in the most efficient manner possible, the nodes with the highest *betweenness* would have the majority of the communications traffic passing through them. Formally, the *betweenness centrality* of a node *v* in a graph $G := (V, E)$ is computed as follows:

1. For each pair of vertices (s, t), compute the shortest paths between them.
2. For each pair of vertices (s, t), determine the fraction of shortest paths that pass through *v*.
3. Sum this fraction over all pairs of vertices (s, t).

One can also represent the *betweenness centrality* of *v* as follows:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node *s* to node *t*, and $\sigma_{st}(v)$ is the number of those paths that pass through *v*.

Your task, should you choose to accept this project, will be to implement a program that calculates the *betweenness centrality* of each node in a network. Your program will take, as input, a list of the network's edges. Your program must output a two-column list where the first column contains the names of all nodes in the input network, and the second column contains the corresponding *betweenness centrality* score, as explained above. The nodes should be ordered from the highest-scoring *betweenness* nodes to the lowest-scoring *betweenness* nodes.

Three test cases are available at http://archive.gersteinlab.org/course/cbb752b12/cbb752_final_project_network_test_cases/cbb752_network_test_cases.zip to help you test and debug your code.

In writing your program, you may use libraries and pre-existing code for certain "supporting computations", such as matrix multiplication. You may *not* use libraries or pre-existing code for any computation that operates directly on the network, including (but not limited to) storage and retrieval of network nodes and edges, calculating path lengths, and computation of *betweenness centrality*. If you are unsure of whether a particular library is permissible, please contact the TFs first.