

# 10-810, Computational Molecular Biology: a machine learning approach: Problem Set 3

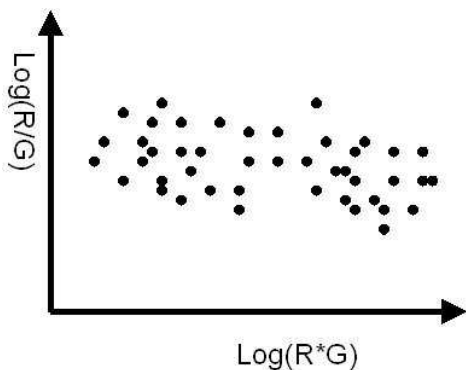
This problem set is due on Monday, 03/28 in class.

## Normalization

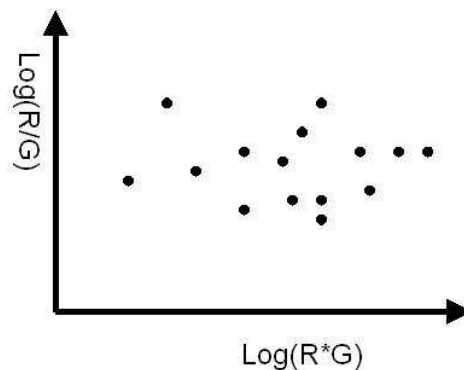
In class, we discussed locally weighted linear regression and mentioned that we can use a Gaussian centered at  $x$  to determine the weight that should be assigned to points (genes) around  $x$ . Here we will explore issues related to this weight.

1. a. What is the effect of having a large variance for such a Gaussian? A small variance?

1. b. Which variance (large or small) would be appropriate to each of the two figures below? Explain.



(a)



(b)

1. c. In gene expression experiments we measure thousands of genes. However, most of these genes are expressed at relatively low levels and only a few are expressed at high levels (high  $R * G$  values). How can we accommodate such a dataset with the Gaussian weighting method? Explain.

## Optimal leaf ordering

In class, we have discussed an  $O(n^3)$  algorithm for optimally ordering the leaves of a hierarchical

clustering tree. Here we will try to improve the running time of this algorithm.

**2. a.** Assume a balanced binary tree (each internal node has exactly two descendants and all leaves are at the same level). Since the algorithm discussed in class is recursive, the last step is to order the lowest (or leaf) level of the resulting tree.

**2.a.1** What is the computational complexity of ordering this level (that is, assume we have already computed  $L_T$  for all levels above this level, what is the complexity of ordering just this level)? Explain.

**2.a.2** How can the algorithm presented in class be modified to reduce the running time **for this level** by a factor of  $n$  ?

**2. b.** Here we will develop an approximation algorithm which runs much faster than the  $O(n^3)$  algorithm discussed in class (though it solves a different problem). Assume that in addition to the similarity values between leaves we have a similarity value between any two nodes (leaves and internal) in the tree. In other words, we are given the complete  $2n - 1$  by  $2n - 1$  similarity matrix for the entire set of nodes in the tree. Again, assume we are working with a balanced binary tree. The **level ordering** problem seeks to optimally order the nodes at level  $l$  holding all levels above  $l$  fixed (the root is level 0 and the leaves are on the lowest level). Note that when level ordering level  $l$  we are only interested in the nodes of level  $l$  and are only allowed to flip between two nodes at this level which have the same parent (their parent resides in level  $l - 1$ ). By optimally ordering we mean maximizing the sum of similarities of neighboring nodes in the ordering (the same as the goal of the algorithm discussed in class, only this time its for all nodes in level  $l$ ).

**2. b. 1.** For the balanced binary tree, discuss an algorithm for level ordering of the nodes in level  $l$  in that tree. What is the computational complexity of your suggested algorithm (the faster the better) ?

**2. b. 2.** What is the computational complexity of level ordering an entire balanced tree with  $n$  leaves (that is, start by level ordering level 1 and proceed to level order all levels from 2 to  $k = \log n$  where  $k$  is the level on which the leaves reside) ?

### Bi-Clustering

**3.** In this problem you will develop and implement a bi-clustering algorithm. A Bi-cluster is a cluster containing a subset of the experiments and a subset of the genes. In this problem we will try to avoid overlap between to Bi-clusters, though other methods allow such overlap.

In order to find a Bi-cluster we need to specify a target function, and look for large subsets (in terms of the number of genes and the number of experiments) that maximize (or minimize in this case) the target function. For this problem the target function will be:

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

where  $a_{ij}$  is the (log) expression of gene  $i$  at experiment  $j$ ,  $a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}$ ,  $a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$  and

$$a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij}$$

Our goal will be to find Bi-clusters with the maximal number of genes ( $|I|$ ) and experiments ( $|J|$ ) that achieve  $H(I, J) \leq \delta$  where  $\delta$  is a user defined threshold.

**3. a.** Motivate the selection of  $H(I, J)$  as the value to minimize. In particular, what does a value of 0 mean? Are there always Bi-clusters with a value of 0?

**3. b.** Show that the problem of finding the largest square Bi-cluster (below a specified threshold) is NP-hard (Hint: assume equal number of rows and columns and reduce Max-Clique to this problem).

Since we cannot reach an optimal solution, we will use a heuristic method to solve this problem and identify the clusters. This method will remove rows and columns until we arrive at a Bi-cluster that satisfies our requirements. Assume we have a potential Bi-cluster (initially all rows and all columns). For a row  $i \in I$  set  $d(i)$  to

$$d(i) = \frac{1}{|J|} \sum_{j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

and similarly for a column  $j \in J$ :

$$d(j) = \frac{1}{|I|} \sum_{i \in I} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

At each step the algorithm will compute  $d(i)$  for all rows and  $d(j)$  for all columns, and will select the highest resulting row  $i'$  and column  $j'$ . If  $d(i') > d(j')$  row  $i'$  will be removed, otherwise column  $j'$  will be removed. This process will be iterated until we arrive at a Bi-cluster satisfying our threshold.

**3. c.** Suggest a way to handle missing values in this case?

**3. d.** Since the above is a deterministic algorithm, and our goal is to find many Bi-clusters, we will need to somehow mask the Bi-cluster we have found and re-run the algorithm. How should this masking be performed?

**3. e.** Implement the above algorithm, with your answers to **c.** and **d.**. Go to: <http://genome-www.stanford.edu/clustering/Figure2.txt> and download this tab delimited file (this file contains log ratio expression values, and missing values are indicated by two or more consecutive tabs). This file is the file used by Eisen *et al* for their hierarchical clustering paper we read in class. Your algorithm will work on these log ratios so there is no need to change these values except for overcoming missing values. Perform the Bi-clustering algorithm on this dataset. You still need to assign one parameter:  $\delta$  (any reasonable choice of  $\delta$  is O.K.). Hand in the number of clusters your algorithm identified and plots for the expression pattern of each of these clusters. For each cluster, generate a file with the ORF names (first column in the file you downloaded) of all genes in that cluster.

**3. f.** Download the files: cin.txt catNames.txt catSize.txt geneNames.txt and compSigClust.m from <http://www-2.cs.cmu.edu/~zivbj/class05/hw3Files/>. The first four files contain functional assignment to the yeast genes (or ORFs). As discussed in class, we can use these functional annotations to validate the results of the clustering algorithm. For the clusters identified in **e.**, use the matlab file you downloaded (compSigClust.m) to compute the p-value for the overlap between each of

these clusters and each of the functional categories. Hand in the output for each cluster which is the top three categories (ordered by increasing p-values) for that cluster. For each cluster

**3. g.** As discussed in class, since we are performing multiple hypothesis testing, it is not clear which of the p-values is really significant. How would you determine which p-value cutoff to use? Determine the appropriate cutoff. Hand in the selected cutoff and, for each cluster the set of categories that were significant using your threshold (not more than three for each cluster). Can expression data identify functionally related genes?

In addition to your answers to the questions in problem 3, you will need to hand in the following:

1. Create a directory with your program, the input files you used and a README file that explains how to perform **e,f,g** using your program. Email me (zivbj@cs.cmu.edu) a zipped version of this directory.
2. For **e** plots for the clusters you identified. For **e,f** the categories that were determined to be significant for each cluster with the p-value assigned to these intersections.