General Guidelines:
This assignment is part of your test in the course. It should be done INDEPENDENTLY, without any help from others. If you use articles or books, please include the reference in your relevant answer.

**Credit:** Solve 10 items out of 11 for full credit. Each item in each question has equal weight. Extra items solved will accumulate towards the next assignments.

1. Describe an efficient implementation of the lowess normalization procedure described in class. You start with two vectors storing the Cy3 and Cy5 values (the intensities of the two colors in a 2-dye experiment). You should use a 100 gene window to compute the local regression and normalize the data. Make your algorithm as efficient as possible and describe its exact complexity (including constants).

2. The appropriate threshold for declaring a p-value as significant becomes complex in multiple testing, due to increased chance of making false rejections of null hypotheses. The classical Familywise Error Rate (FWER) methods (e.g., Bonferroni correction) control the chance of making even a single false rejection. Alternatively, the False Discovery Rate (FDR) method controls the expected proportion of false rejections.

   (a) Prove that any procedure that controls FWER controls also FDR, but controlling FDR will not always control FWER.

   (b) Define the power of a method as $100\% - \beta$ where $\beta$ is the proportion of failures to reject the non-true null hypotheses, i.e., failures to recognize the true effects as significant. Prove that the power of FDR is higher or equal to the power of FWER.

   (c) Show that with growing number of tests, the power of FWER becomes lower. What happens to the power of FDR in such case?

3. Wilcoxon Rank-Sum Test In class you saw several ways for testing differential expression of a gene (e.g., using a t-test). Suppose we observe $n$ expression values of the gene in a sample $x_1, \ldots, x_n$. Suppose that the label of the $i$th sample can be either
+1 or −1 (e.g., +1 represents healthy and −1 sick). The Wilcoxon Rank-Sum Test is a test for difference between the positive and negative samples. Let $r_i$ be the rank of the $i$th sample if we sort them by the expression values. Define $w_{+1} = \sum_i I(l_i = +1)r_i$, where $I$ is the indicator function. That is, $w_{+1}$ is the sum of ranks of the samples in the positive group. Similarly we can define $w_{-1}$ to be the rank of the negative group. The null hypothesis is that the expression values are randomly permuted (thus, keeping the same numbers of positive and negative samples). Let $W_{+1}$ be the random variable denoting the sum rank in a random sample of labels. The p-value of of $w_{+1}$ is the probability $P(W_{+1} \geq w_{+1})$. If this value is small we can reject the null hypothesis.

Suppose we want to compute the p-value exactly. Naively, this requires summing over all the permutations of the ranks. Develop a dynamic programming algorithm for this task. What is the complexity of your algorithm?

4. The version of the $k$-means algorithm described in class aims to solve a minimization problem, looking for a partitioning of the universe into $k$ subsets such that the total distance of elements from their sets’ center is minimized. In class we saw how to use a simple hill-climbing strategy that converged to a local optimum when no moving of a single element between clusters improves the score. Describe how a simulated annealing heuristic can improve the hill climbing strategy. Formally write down the equations of how to use simulated annealing in $k$-means. (For an introduction to simulated annealing, see, e.g., *Statistical Methods in Bioinformatics: An Introduction* by Warren J. Ewens and Gregory Grant (2001), sections 10.5.1 and 10.5.3.)

5. Let $G = (V, E, w)$ be a weighted graph where all edge weights are non-negative integers. One formulation of the clustering problem is as follows: Given integers $k$ and $B$, we wish to decide if there exists a partition into $k$ sets such that the maximum edge weight within each set is at most $B$ (i.e. $\max_i \max_{u,v \in S_i} w(u,v) \leq B$). In class we proved that the problem is NP-hard even for $k = 3$. Prove that the problem is polynomial for $k = 2$, but that the version where we replace the maximum by the sum (i.e. the sum of edge weights within each sets is at most $B$, namely, $\max_i \sum_{u,v \in S_i} w(u,v) \leq B$) is NP-hard for $k = 2$.

6. In many clustering applications we only know if two elements are related or not. That is, for every pair of elements, we are told if they are expected to be in the same cluster or not. If they are expected to be in the same cluster then the pair is labeled "r" (for a real pair), otherwise the pair is labeled "n" (for a pair that is not real). Given a clustering solution $C$, for every gene pair $(u,v)$ we define two options: (1) Hit: if $u$ and $v$ are in the same cluster and $(u,v)$ is labeled "r", or if $u$ and $v$ are in different clusters
Shamir: Analysis of DNA Chips and Gene Networks © Tel Aviv Univ., Spring ’12

and (u,v) is labeled ”n”. (2) Error: if u and v are in the same cluster and (u,v) is labeled ”n”, or if u and v are in different clusters and (u,v) is labeled ”r”.

(a) Prove that finding a clustering solution with a minimum number of errors is NP-hard. What can you say about maximizing the number of hits?

Hint: deciding if a graph can be partitioned into triangles is NP-hard.

(b) Give a constant factor polynomial time approximation algorithm for maximizing the number of hits.

Hint: Keep it simple

7. Practical part

In this question you are asked to analyze real data using the EXPANDER software (http://acgt.cs.tau.ac.il/expander/). Gene expression data are represented as a matrix in which rows correspond to genes and columns correspond to experiments. In addition, every row and every column has a name. In EXPANDER, you can load such a matrix by selecting File → New session→Expression Data→ Tabular Data File. In the pop-up window, select the file that you want to analyze. For the data you are given here, select Organism→Human, select ”Use probe IDs as gene IDs”, and modify ”Data type” to ”Relative Intensities”.

Download the gene expression data from the course website (filename: ex1_data) and upload it to EXPANDER. This data set contains information on 1500 genes and 3 conditions. This data set comprises three known gene clusters.

Once the data are uploaded to EXPANDER, you can cluster the genes by going to Unsupervised Grouping→Clustering→AlgorithmName.

(a) Compare the clustering solution of K-means (k=3) to a clustering solution of SOM that also provides three clusters. Which solution is better in your opinion and why?

To view a 2-dimensional projection of a clustering solution you can use Visualizations→PCA in EXPANDER.

(b) Cluster the data, using K-means, for k=1...8 clusters. Plot the homogeneity and separation of the different solutions as a function of k. Is the prior knowledge that the genes are in fact from three known clusters reflected in your plot?