MSCBIO 2070/02-710: Computational Genomics, Spring 2015

A4: spline, HMM, clustering, time-series data analysis, RNA-folding

Due: April 13, 2015 by email to Silvia Liu (silvia.shuchang.liu@gmail.com)
TA in charge: Silvia Liu (Q1, Q3) & Marta Wells (Q2, Q4)

Your goals in this assignment are to

1. Understand cubic spline interpolation;
2. Master the HMM algorithm;
3. Implement bi-clustering algorithm into time-series dataset;
4. Implement Nussinov algorithm in RNA-folding.

What to hand in. Write a short report addressing each of the questions below (either a hand-written report or a report submitted as a pdf file is acceptable). The report should be self-contained, we should not have to run your code to be convinced that your code is correct. Be sure to comment your code and use any programming language you like (if not specified). Also, include instructions on how to run your code. In your report you can assume that we know the context of the questions, so do not spend time repeating material in the hand-out or in class notes. Email a zip file containing the complete code (if any) and pdf file (if any) to: Silvia Liu, *silvia.shuchang.liu@gmail.com* with the Subject: *2070S15 A4 YourName*.

1 [15 points] Cubic Spline

Cubic Spline Method

Given a set of n control points, cubic spline constructs $n-1$ piecewise third-order polynomials between the points. The splines need to satisfy the following properties:

- Each spline need to pass through its left-handed and right-handed control points.
- The splines located on the left and right hand of a control point should be continuous and have the equal first derivative value at that point.
- The splines located on the left and right hand of a control point should have the equal second derivative value at that point.
- The second derivatives at the endpoints (first and last control points) should be zero.

Your tasks:

1. (5 points) Assume we have a value of v for point i. Let $S_1 = ax^3 + bx^2 + cx + d$ be the spline to the left of this point, and $S_2 = ex^3 + fx^2 + gx + h$ be the spline to its right. You may parameterize both splines as $x \in [0, 1]$. How many equations are defined by point i? Write all these equations and simplify each as much as you can. *[Hint: You may need to implement the properties listed above.]*

   There are four equations defined by each internal point (and one by each of the two endpoints). The four equations are:

   1. Continuous value at end point

   $$a1^3 + b1^2 + c1 + d = v \quad \Rightarrow \quad a + b + c + d = v$$

   2. Continuous value at start point

   $$e0^3 + f0^2 + g0 + h = v \quad \Rightarrow \quad h = v$$

   3. Equality of first derivative

   $$3a1^2 + 2b1 + c = 3e0^2 + 2f0 + g \quad \Rightarrow \quad 3a + 2b + c = g$$

   4. Equality of second derivative

   $$6a1 + 2b = 6e0 + 2f \quad \Rightarrow \quad 3a + b = f$$

2. (5 points) Assume we have a value of u for the first control point. Let $S = \alpha x^3 + \beta x^2 + \gamma x + \theta$ be the spline located on its right. Also parameterize the spline as $x \in [0, 1]$. How many equations can you write for this point? Try to simplify each equation. In order to get all the splines, how many equations for n control points do we need?

   There are two equations defined by the first point.

   1. Continuous value

   $$\alpha 0^3 + \beta 0^2 + \gamma 0 + \theta = u \quad \Rightarrow \quad \theta = u$$

   2. Second derivative equals to zero

   $$6\alpha 0 + 2\beta = 0 \quad \Rightarrow \quad \beta = 0$$

   For n points, we need $n-1$ splines. Each spline has 4 parameters to estimate. So totally we need $4(n-1)$ equations. Each internal point can define 4 equations and each end point can define 2 equations. Totally there are $4(n-2) + 2 \times 2 = 4(n-1)$ equations. These are exactly what we need to estimate all the parameters.

3. (5 points) In class we actually discussed approximating splines, that is splines that contain less control points than the number of actual measured points. One important issue is how to choose the number of control points to assign. If we assign too many, we will overfit the data. But if we assign too few, we might not be able to accurately reconstruct the underlying expression curve. Assume control points are uniformly spaced. Suggest a method for determining the number of control points we should use.

   We can use cross validation for this task. Start with interpolating splines (the largest number of control points) and hide a complete column (one experiments), say out of 10 time points we are hiding time point 5. Following spline assignment we can compute the predicted values for TP 5 (since we have a continuous curve). Next, we test how far are the values predicted from the values we have hidden. This is repeated for other internal TPs and an average loss is computed. Next, we reduce the number of control points by one and repeat this process until we end with 4 control points (the lowest number possible). Then we pick the number of control points that yielded the lowest average error between the predicted and hidden values and use these.

2 [15 points] Hidden Markov Model

In the class you learned how to identify CpG islands in a sequence. Here you will extend the HMM model to capture the basic properties of human genes.

We will take a simple view where a gene consisting of a continuous sequence of DNA is transcribed first into premessenger RNA (pre-mRNA). This pre-mRNA consists of an alternate sequence of exons and introns. After transcription, introns are edited/spliced out of the pre-mRNA to form mRNA which is then translated into a protein.

For the HMM that you will build here, assume that you have access to two different learning models:

L1. You can invoke a semihidden Markov Model (sHMM) for a state, where all transition probabilities from a state to itself are zero, and when the Markov process visits a state, it produces not just a single symbol from the alphabet but rather an ENTIRE sequence.

To generate the sequence, we first pick a length value L from a given distribution. Then we generate a random sequence of length L from another known distribution. For now, we will not worry about the nature of these distributions.

L2. You can use position weight matrices (PWM) of a "suitable size" for training a state.

EACH QUESTION BELOW PROVIDES PARTIAL KNOWLEDGE ABOUT THE STRUCTURE OF A GENE. YOU WILL INCORPORATE THIS KNOWLEDGE INTO YOUR HMM. YOU WILL SPECIFY:

(a) the states in a HMM that you will introduce to capture the partially stated structure of a gene;

(b) the models (PWM, sHMM) you will invoke to learn the parameters of the introduced states;

(c) do not forget to mention any other properties that states might possess;

AND DRAW

where applicable, the transition arrows from the newly introduced state(s) to the state(s) already defined.

Your task:

1. (5 points) **Encoding Intergenic Region:** DNA sequence is very long with genes that encode for proteins and regions that do not encode. Introduce a state for intergenic regions, to identify regions between genes.

   Introduce a node say N whose parameters will be learned with sHMM.

2. (5 points) **Encoding Promoter Region:** Promoter region of a gene is the region where specialized proteins bind and initiate transcription. Thus, promoter regions appear before the start of the transcribed region. Say you decided to recognize the signal called the TATA box which is located 28-34 bases upstream of transcription. Show how you will incorporate this knowledge in your HMM by answering the questions listed in (a, b, c) above.

   Introduce a node TATA box and another node N1 that spits out a sequence of length 28-34 bases; learn the parameters of TATA with PWM and N1 with sHMM; N1 can transit to TATA and N1 but no arrows in the reverse direction; TATA connects to N1 but not the other way round.

3. (5 points) **Encoding 5′ UTR Region:** 5′ untranslated region which does not get translated into a protein follows the promoter. There is a cap end of 5′UTR Layer that is 8 bases long. Near the other end of 5′UTR is a translation initiation signal/end (TIE) following which

is the start codon. TIE will have a span of 18 bases. Show how you will incorporate this knowledge in your HMM by answering the questions listed in (a, b, c) above.

Introduce a cap end state and learn it with 8-base weight matrix; followed by an intergenic model N2; followed by TIE node learned with a 18-base weight matrix; N1 transits into cap end; cap-end transits to N2 and N2 transits to TIE.

In the interest of time, we will skip layers encoding exon and intron regions, and also the post-translational region.

3 [40 points] Biclustering: Application in Time-series Data

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 not allow overlap between the bi-clusters, though other methods allow such overlap. By answering the questions below you will develop (and implement) a method that uses bipartite graphs for bi-clustering.

Data Description

We will implement the bi-clustering algorithm into a time-series data set. The data file named *alphaCycle.txt* can be downloaded online from our website. In the data matrix, each row represents a gene and each column corresponds to a time point (experiment). So each value in row i and column j corresponds to the expression level (log ratio) of gene i in time point j. File *alphaGenes.txt* contains the corresponding gene names.

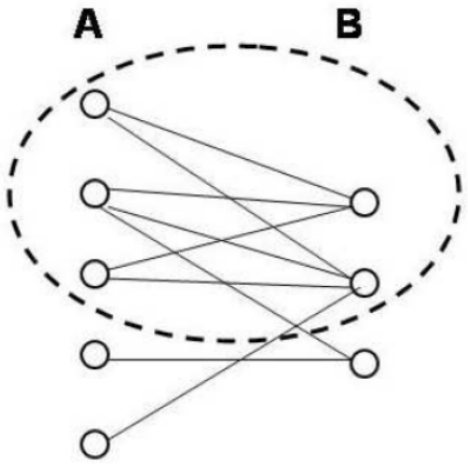

Figure 1: A bipartite graph. The dashed circle contains a complete subgraph in this graph, serving as a good candidate for a bi-cluster.

Your task:

1. (3 points) How could we use an undirected bipartite graph to represent the gene expression data matrix? What do the nodes and edges represent?

   We can have the left nodes represent genes and the right nodes represent time points. An edge in the graph means that the gene is either expressed or repressed in that time point.

2. (4 points) Since gene expression data contains non-discrete values, we will first discretized the data. Every value above 0.9 will be set to 1 and every value below -0.9 will be set to -1. Values between -0.9 and 0.9 will be set to 0. Discretize your data matrix by this method and show how many values are assigned to 1 and -1 respectively.

   Totally 520 values are assigned to 1 and 416 values are assigned to -1.

3. (3 points) Using one unweighted bipartite graph (that is, all edges have the same weight of 1) as you have described above, how can you represent both activation (1) and repression (-1)? (Remember, we would like to cluster activated genes in a different cluster than the repressed ones.)

We will use two nodes for each time point. One will be connected to all over-expressed genes and the other to the under-expressed ones. Thus, we will have twice the number of nodes than the number of time points.

Alternatively, we can use both two copies of gene nodes and two copies of time nodes. The edges between first copy of gene nodes and time nodes will represent the higher expression, and the edges connecting second copy of gene and time nodes will represent the lower expression.

4. (3 points) Assume the graph has a bounded out degree on the left (that is, no node on the left side has more than d outgoing edges). Also, assume that we are looking for complete subgraphs (figure 1). That is a subset of the nodes on the left ($l \subseteq A$) and a subset of the nodes on the right ($r \subseteq B$), where each node in l is connected to all nodes in r and vice versa. What is the largest possible size of r?

   Since each gene node (left node) is connected to at most d experiment nodes (right nodes), we can only have at most d nodes from B in a complete subgraph.

5. (5 points) Let n be the number of genes. You are asked to develop a $O(n2^d)$ algorithm for finding the maximal complete subgraph (where maximal means that it has the most number of edges). Explain your algorithm and its complexity.
   [Hint: Here is a nice paper FYI, `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.8302&rep=rep1&type=pdf`*.]*

   Since each node in A is connected to at most d nodes in B, the total number of subsets of the d nodes is 2^d. For each of these subsets we need to determine a score. This can be done by starting with the singleton sets and increasing them by one node at a time. Each time we add a node we need to perform a merging step to merge the set of nodes in A connected to the subset we have with the nodes connected to the new node we are adding. While this can take a time linear in the number of nodes connected to the new node we had, this cost can be amortized over all nodes by not considering a subset more than once. Thus, the total running time will be $O(n2^d)$.

6. (8 points) Implement the algorithm in the discretized data matrix to find the bi-cluster. After finding one maximal complete subgraph, you can set the edges in that subgraph to 0 and repeat the method to find the next bi-cluster. Draw a table to show out the dimensions of the top 5 bi-clusters you detected.

	# gens	# time points	# edges	activated / repressed
1	99	1	99	repressed
2	66	1	66	activated
3	58	1	58	activated
4	55	1	55	activated
5	51	1	51	activated

7. (8 points) Did you find any problem with this method? Try to come up with a solution to the problem and implement revised algorithm into the discretized data. Again, draw a table to show out the dimensions of the top 5 bi-clusters.

   When checking the bi-cluster dimensions in the above table, we find that all the top bi-clusters only have one time point. Biologically, we are more interested in the genes that have high/low expression levels over multiple time points. Here I change the code slightly, only recording the bi-clusters that have more than 3 time points. The dimensions of the new top 5 bi-clusters are shown in the following table. *This is an open question. Any reasonable answers will be accepted.*

	# gens	# time points	# edges	activated / repressed
1	18	3	54	repressed
2	10	3	30	repressed
3	7	4	28	activated
4	9	3	27	activated
5	8	3	24	activated

8. (6 points) After detecting the bi-clusters, we need to know whether the genes detected are meaningful (or what we could learn form the bi-clusters). GO enrichment analysis is one of the popular methods. For each of the top bi-cluster detected in step 7,

   (a) Go to the FuncAssociate website, http://llama.mshri.on.ca/funcassociate/.

   (b) In step 1, select *species* as *Saccharomyces cerevisiae*.

   (c) In step 2, select *namespace* as *sgd_systematic*.

   (d) In step 3, paste the names of the genes in the bi-cluster in *Query List*.

   (e) Click *Functionate!* button.

   (f) Sort the results by p-values in ascending order. Draw a table to list the top 3 GO categories, showing the Gene-Ontology-ID, Gene-Ontology-Attribute and p-value in each column.

   Briefly explain the GO analysis and what we could learn form the table?

   First bi-cluster:

	Gene-Ontology-ID	Gene-Ontology-Attribute	p-value
1	0000788	nuclear nucleosome	1.138×10^{-19}
2	0000786	nucleosome	8.853×10^{-19}
3	1990104	DNA bending complex	8.853×10^{-19}

   Second bi-cluster:

	Gene-Ontology-ID	Gene-Ontology-Attribute	p-value
1	0005576	extracellular region	8.448×10^{-14}
2	0009277	fungal-type cell wall	1.998×10^{-11}
3	0005618	cell wall	3.060×10^{-11}

   Third bi-cluster:

	Gene-Ontology-ID	Gene-Ontology-Attribute	p-value
1	0000788	nuclear nucleosome	4.125×10^{-21}
2	0000786	nucleosome	2.145×10^{-20}
3	1990104	DNA bending complex	2.145×10^{-20}

   Fourth bi-cluster:

	Gene-Ontology-ID	Gene-Ontology-Attribute	p-value
1	0005576	extracellular region	8.270×10^{-8}
2	0009277	fungal-type cell wall	9.677×10^{-8}
3	0005618	cell wall	1.306×10^{-7}

   Fifth bi-cluster: no result.

   GO analysis is to check whether the selected genes are enriched in some important pathways. From the table we could see that most of the pathways are related to cell cycle (for instance, related to nucleosome or cell wall pathways).

4 [30 points] RNA-folding

Write code to implement the Nussinov algorithm and fold the RNA sequence ACCAGAACUGGU. Use a score of 1 for base-pairing, 0 otherwise. In your implementation, you will skip base pairing if the distance between bases is less than 4. That is, if i and j are base indices and $j > i$, if j–$i < 4$, then the score is 0. Report the traceback (which residues bind), as well as the matrix you build using the Nussinov algorithm. For binding, report the indices and letter of the residues. For example: (A1-U12).

The binding pairs are:
A1-U12
C2-G11
C3-G10
A4-U9
The score matrix is

i, j	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	1	2	3	4
2	0	0	0	0	0	0	0	0	1	2	3	3
3	0	0	0	0	0	0	0	0	1	2	2	2
4	0	0	0	0	0	0	0	0	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0