

### Assignment Goals

- i. Implement the Nussinov algorithm for RNA secondary structure prediction.
- ii. Resolve read mapping uncertainty in RNA-Seq quantification.
- iii. Use probabilistic splice graphs as a compact representation for gene isoforms and their transcript frequencies.
- iv. Get familiar with the workflow of machine-learning modeling.

### Submission Instructions

- To turn in your assignment, log in to the server **mi1.biostat.wisc.edu** or **mi2.biostat.wisc.edu** using your BMI (biostat) username and password.
- Copy all relevant files to the directory

**/u/medinfo/handin/bmi776/hw4/<USERNAME>**

where **<USERNAME>** is your BMI (biostat) username. Submit all of your Python source code and test that it runs on the biostat server.

- For the rest of the assignment, compile all of your answers in a single file and submit as **solution.pdf**.
- Write the number of late days you used at the top of **solution.pdf**.
- For the written portions of the assignment, show your work for partial credit.

### Part 1: RNA Secondary Structure Prediction (55 points)

- (A) Write a program, **nussinov.py**, that takes as input an RNA sequence and outputs the base-paired positions of a secondary structure that maximizes the number of base pairings in the absence of pseudo-knots. Base pairing within subsequences of length  $\leq 4$  is not allowed. If multiple optimal secondary structures exist, prefer the middle-road (i.e., diagonal) path to the high- and low-road paths in the score matrix when doing traceback.

Your program should be callable from the command line as follows:

```
python nussinov.py --out=<out> <sequence>
```

where

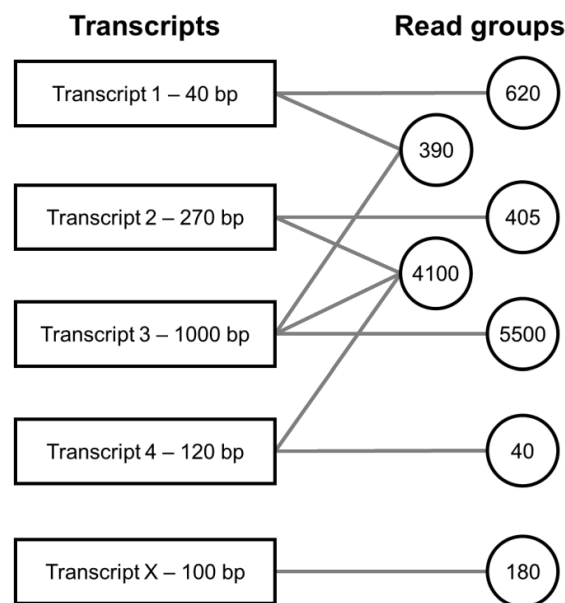
- **<sequence>** is a text file containing a string of A, C, G and U characters.
- **<out>** is the name of the text file into which the program will print all pairs of base-paired positions (using one-based indexing) in lexicographical order.

Example input files **example1.txt** and **example2.txt**, their corresponding output files, and the template **nussinov.py** with argument parsing code can be found in the **hw4\_files** directory. Your program will be evaluated on the example inputs and additional datasets that will be kept private. **(50 points)**

- (B) Draw the secondary structure of the RNA sequence in **example2.txt** based on the base pairings in **example2\_out.txt**. Identify the bulges, stems, internal loops and hairpin loops present in the structure. (5 points)

**Part 2: RNA-Seq Rescue Algorithm (15 points)**







The full RSEM algorithm is too complicated to execute manually, but we can use the RNA-Seq rescue method to approximate one iteration of expectation-maximization. The bipartite graph below contains two types of nodes: transcripts and read groups. The transcript nodes contain a transcript ID and the transcript length in base pairs (bp). The read nodes contain the read counts for a group of reads that all align to the same transcripts. The edges designate the transcripts to which each read group aligns.



- (A) Use the rescue method to calculate the *relative* abundance for the five transcripts. (10 points)
- (B) Transcript X is an RNA spike-in. 1000 copies of transcript X were mixed into the experimental sample when preparing the sample for RNA-Seq, meaning its absolute abundance is 1000. Use the relative abundance from (A) to calculate the *absolute* abundances for the other four transcripts, rounded to the nearest integer. (5 points)

**Part 3: Probabilistic Splice Graphs (10 points)**

In this problem, we will examine how probabilistic splice graphs may produce a more compact representation for the possible isoform structures of a gene and their frequencies. Shown in the figure below are six possible isoform structures for a gene.

	<b>A</b>	<b>B</b>
	0.1	0.1
	0.025	0.1
	0.4	0.4
	0.1	0.025
	0.25	0.25
	0.125	0.125

Give the *most compact* probabilistic splice graph representation for this gene for the isoform frequencies given in scenarios A and B, respectively.

#### **Part 4: Machine-Learning Modeling (20 points)**

Neural stem cell-based in vitro models can be used for pre-clinical screening of neurotoxic compounds. In collaboration with stem cell biologists, you as a bioinformatician want to build a predictive machine-learning model for neurotoxicity based on changes in global gene expression of neural tissue cultures exposed to known neurotoxic and control compounds. Your collaborators performed RNA-Seq and obtained measurements for 20,000 genes following exposure to 20 toxins and 40 nontoxic controls, with five biological replicates for each compound.

- (A) (*Exploratory data analysis*) It is common practice to apply unsupervised learning methods (clustering, dimensionality reduction, etc.) on the measurement data in order to understand the intragroup variability among replicates and the intergroup variability among samples treated with different compounds. Outline *two* unsupervised learning methods that you think would serve this purpose. Describe what you would expect these methods to uncover. **(10 points)**
- (B) (*Learning a classifier*) Now that you have gained some insight from the data through unsupervised learning, you would like to proceed and build a support vector machine (SVM) for neurotoxicity classification. Given a gene expression profile measured following drug exposure, the SVM will classify the drug as either toxic or nontoxic. Describe a workflow for training and evaluating the SVM. Beware of the high feature dimensionality relative to the sample size and the class imbalance problem. **(10 points)**