

Assignment Goals

- i. Compare and contrast algorithms for finding paths in interaction networks.
- ii. Get familiar with the workflow of machine-learning modeling.
- iii. Resolve read mapping uncertainty in RNA-Seq quantification.
- iv. Use Gaussian processes to model biological time series.

Submission Instructions

- To turn in your assignment, please 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/hw3/<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: Source-target paths in networks (25 points)

We will use the `networkx` Python package to compare and contrast two algorithms for finding source-target paths in a network. One algorithm optimizes the min-cost flow similar to ResponseNet. The other finds the k shortest weighted paths. In both cases, you are given an undirected network where each line in the input file lists a pair of nodes followed by its weight, which is interpreted as the cost of transmitting flow by the min-cost flow algorithm. The `networkx` flow algorithms require directed graphs, so we represent an undirected edge as a pair of directed edges with the same weight. In addition, you are provided with a list of source and target nodes. These nodes will be connected to an artificial source and an artificial target, as in ResponseNet.

Our objective is to find connections from the artificial source to the artificial target. Your task is to finish and test a mostly complete program, **`find_paths.py`**, which implements the min-cost flow and shortest path based algorithms.

The `networkx` package is installed on the biostat server. The program is callable from the command line as follows:

```
python find_paths.py --edges=<edges> --flow=<flow> \
    --sources=<sources> --targets=<targets> --out=<out>
```

or

```
python find_paths.py --edges=<edges> --k=<k> \
    --sources=<sources> --targets=<targets> --out=<out>
```

where

- **<edges>** is a text file listing weighted undirected edges one per line.
- **<sources>** is a text file listing source nodes one per line.
- **<targets>** is a text file listing target nodes one per line.
- **<out>** is the name of the text file into which the program will print the identified source-target paths.
- **<flow>** is a positive number specifying the amount of flow to send from the artificial source to the artificial target.
- **<k>** is a positive integer specifying the number of shortest paths to find.

(A) (*Path-finding implementation*) Please complete the five code segments marked as TODO in `find_paths.py`. The [networkx documentation](#) will be useful for learning how it represents the [graph](#) data structure and implements the [min-cost flow](#) and [shortest path](#) based path-finding algorithms. You may use the provided `print_graph` and the `networkx draw` functions to visualize the graph, and the example input files `example_graph.txt`, `example_sources.txt` and `example_targets.txt` to test your code. When `find_paths.py` is run with `--flow=3`, you should obtain `example_paths_flow.txt` or the equally good `example_paths_alt_flow.txt`. When it is run with `--k=7` you should obtain `example_paths_shortest.txt`. (15 points)

(B) (*Min-cost flow vs. k shortest paths*) Based on the `networkx` documentation and your own experiments, discuss the strength and weakness of each method. To begin with, you may examine how the edges 2-5 and 5-11 are used in the flow- and shortest path-based solutions in your test output. (5 points)

(C) (*Special case*) So far we have used infinite capacity on the edges incident to the artificial source and target, and a capacity of one on all real edges in the network. Describe a way to change the capacities such that the min-cost flow algorithm will return essentially the same solution as k shortest paths for some value of k . What value of k is relevant for this special case? (5 points)

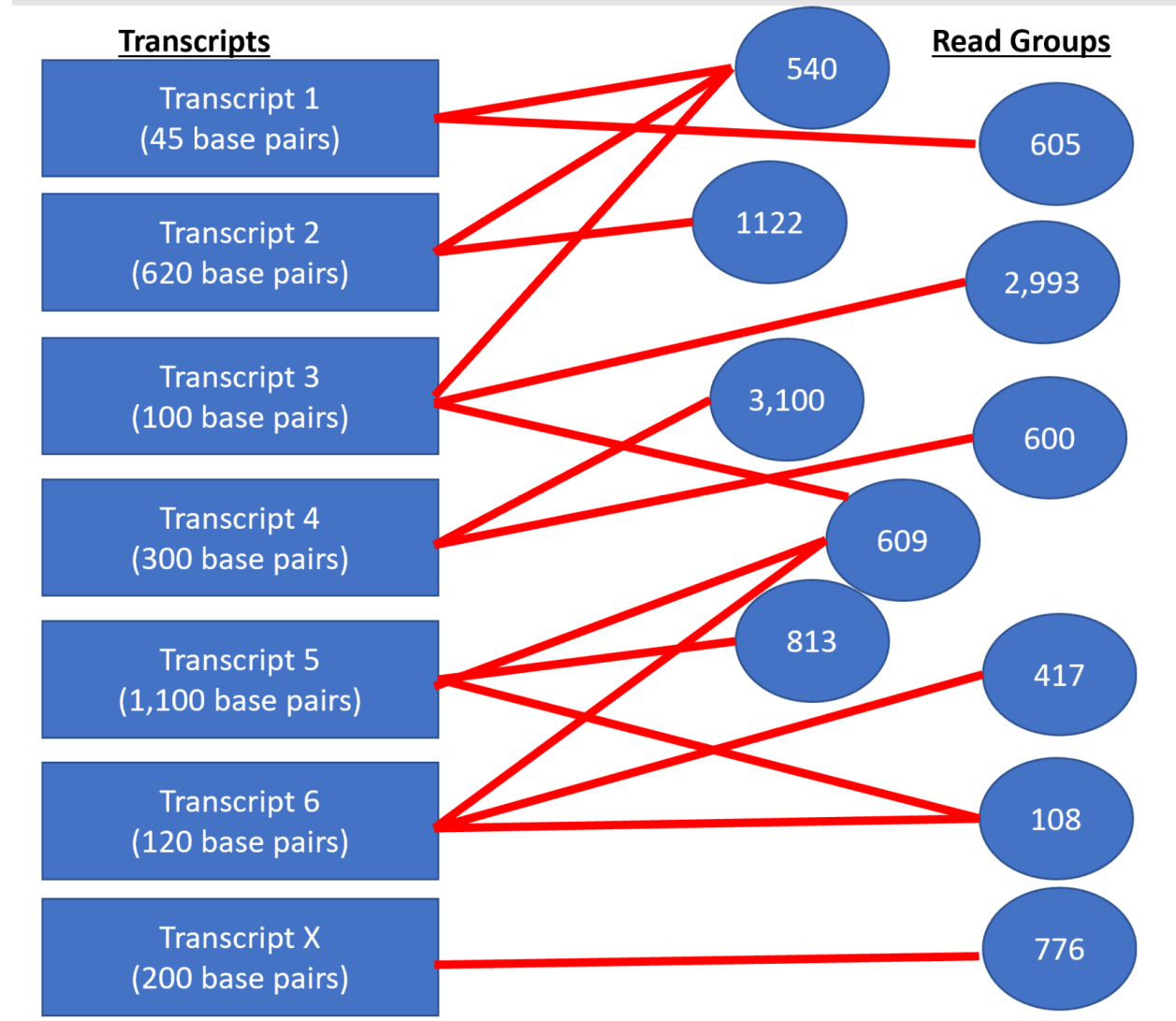
Part 2: Machine-Learning Modeling (30 points)

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 25,000 genes following exposure to 24 toxins and 30 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. **(15 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. **(15 points)**

Part 3: RNA-Seq Rescue Algorithm (25 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.



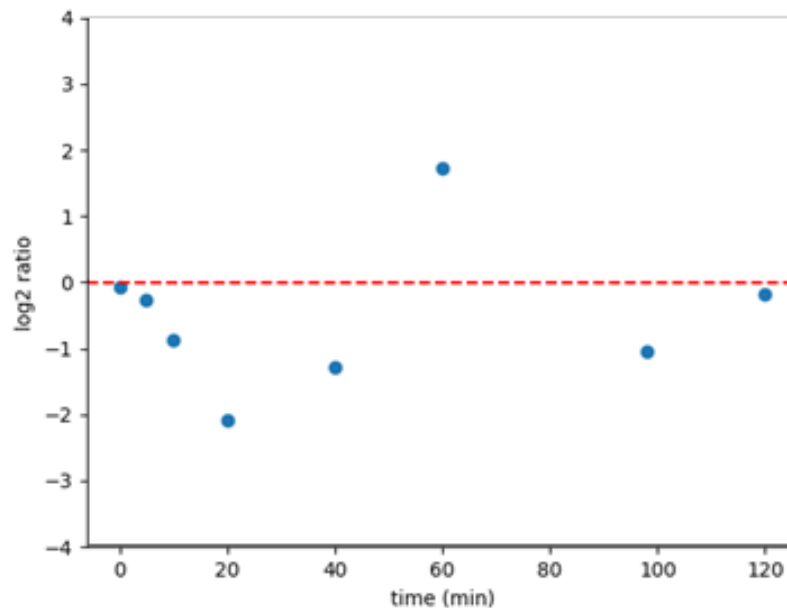
(A) Use the rescue method to calculate the *relative* abundance for the five transcripts. (15 points)

(B) Transcript X is an RNA spike-in. 850 copies of transcript X were mixed into the experimental sample when preparing the sample for RNA-Seq, meaning its absolute abundance is 850. Use the relative abundance from (A) to calculate the *absolute* abundances for the other six transcripts, rounded to the nearest integer. (10 points)

Part 4: Gaussian processes for biological time series (20 points)

Suppose we are studying how neural stem cells respond to an environmental toxin. We perform RNA-Seq on cells in the control and treatment groups to obtain gene expression data at 0, 5, 10, 20, 40, 60, 90 and 120 min. Gaussian processes (GP) with a squared exponential kernel are well suited for modeling biological data collected over a time course. Following GP regression, the posterior mean is smooth over time, and the confidence intervals track uncertainty between the measured time points.

- (A) Shown below is a time series of \log_2 fold change of gene expression (i.e., \log_2 ratio between treatment and control). Assume a GP prior with zero mean and a squared exponential kernel. Sketch the mean and 95% confidence interval of the GP posterior (note the irregular time intervals). You may further assume that the kernel parameters (i.e., length scale l and variance σ^2) have been optimized to maximize the data likelihood. Explain your reasoning. (8 points)



- (B) Differential expression analysis and clustering are the natural first steps toward understanding gene functions. Describe a GP-based statistical test that can be applied separately to each gene to assess whether its temporal expression profile in the normal condition differs from that under drug exposure, or a GP-based

probabilistic model that clusters genes by their temporal expression profiles to reveal shared biological functions. You are encouraged (but not required) to work with the \log_2 ratios as in (A). Please clearly state and justify your test/model formulation as well as your assumptions. **(12 points)**