CSC 315, Fall 2019
Lab #10: Bladder Cancer Classification Challenge

Due dates:

Tuesday, 11/24/20, 5:00 PM	**1st submission (text file of predictions)**
Tuesday, 12/01/20, 11:00 AM	**2nd submission (text file of predictions)**
Wednesday, 12/03/20 11:00 AM	**Final notebook**

An important clinical characteristic of a bladder tumor is its stage, which is either non-muscle invasive (NMI) or muscle invasive (MI). NMI tumors are often manageable, but recur (come back) at a very high rate following surgery. MI tumors, on the other hand, are lethal in about 50% of cases with current treatments. Accurate staging of a patient's tumor is important for guiding treatment decisions, and a better understanding of the genomic differences between NMI and MI tumors may lead to advances in targeted therapies in bladder cancer.

In this lab, you will use your knowledge of gene expression data, differential expression, and classification to classify bladder tumors as being NMI or MI. You will also compete against your classmates to see which team can develop the most accurate classifier!

Directions: Modify the *Challenge.R* script as described below. You may work in groups of up to 3 on this assignment.

First Submission:

1. Identify a set of differentially expressed probes using a sensible FDR cutoff, such as 0.05. If desired, you may also use a logFC cutoff (such as logFC > 0.5 or logFC < -0.5).

2. Using these differentially expressed probes, find the balanced accuracy of a *knn* classifier for stage, using leave-one-out cross-validation in the training data. In your classification, you will need to select values for relevant parameters (such as the value of *k* in *knn*). For this step, these values will be fixed.

3. Next, classify the test samples, and e-mail your predictions to dancikg@easternct.edu with the subject: **Bioinformatics Challenge**. In the e-mail, include your team name (be creative!), and team member names, followed by the predictions, with 1 prediction per line. A leaderboard will be posted to Piazza and updated as predictions come in.

Second submission:

4. Next, optimize at least one of the parameters using a classification method of your choice. In addition to the *knn* classifier you may want to consider additional methods such as

5.
    a. *CCM* (https://cran.r-project.org/package=CCM),
    b. support vector machines (available in the *R* library e1071),
    c. *pamr* (https://cran.r-project.org/package=pamr). N

    Note that more advanced classifiers such as these will likely be necessary to obtain a balanced accuracy above 70%. For this problem, a good performance is in the 70% range.

    In order to optimize your classifier, you will need to look at a range of parameter values, and choose the parameter value or values that give the most accurate results. Parameter values to consider include FDR, logFC, and additional classification parameters such as *k*. If you want to optimize over multiple parameters, the *expand.grid* function can be used to generate all combinations of multiple parameters. For example, the code below creates a matrix for all combinations for *k = 1,3,5,7,9,11,13,15* and *FDR* of 0.001, 0.01, and 0.05.

    expand.grid(k = seq(1,15,by=2), FDR = c(.001, .01, .05))

6. Once you have optimized your classifier, classify the test samples and e-mail me your predictions following the directions in (3).

Final Notebook:

7. At the completion of the challenge, you will submit an *R* notebook that shows the work for your two submissions. This *R* notebook should include the leave-one-out cross-validation that goes with your first submission, and the optimization that corresponds to your second submission*, as described above. At the end of the *R* script, you should include a brief description of the methods used for your second submission, the results (including how many probes are in the classifier, and its accuracy in the test dataset), and whether or not the classifier is more accurate for certain cases than others. The notebook must justify this description, for example by explicitly showing the number of probes used, and how you found the optimal classifier. An example with made-up results is below.

We identified 1500 differentially expressed probes using a false discovery rate (FDR) of 10%. We then developed a k-nearest neighbor (knn) classifier. We considered values of k = 1,3, 5, 7, and 9 and optimized the value of k using leave-one-out cross-validation. Our optimal classifier had k = 5 and had a balanced accuracy of 70.3% in the test dataset. Interestingly, the classifier performed better on NMI tumors (90.4% sensitivity) than on MI tumors (50.2% sensitivity), suggesting that many MI tumors may resemble NMI tumors based on gene expression data.

*You may submit more than twice, in which case you would include the R code and results for your most accurate classifier