Problem Set 2

REVISED FEB 3 2016

Introduction

This problem set will be an exercise in survival analysis, which involves integration of clinical data as well as gene expression data. Survival analysis is an important and often-used technique for determining whether a gene or gene panel is a useful prognostication tool. For our problem set, we'll work with data from The Cancer Genome Atlas, or TCGA (http://cancergenome.nih.gov/). We will first work through a univariate Cox model involving one gene. We will then explore multiple hypothesis testing correction and its importance when considering many possible univariate models. Finally, we'll work through a multivariate model.

To get started

You may do work on your own local machine or on corn.stanford.edu. If you work on your local machine, please make sure your code also works on corn.stanford.edu!

To work on corn do the following:

- ssh to yoursunet@corn.stanford.edu
- Use your stanford password to login.

Part 1: Univariate Cox Model

1. You will start by choosing a cancer of interest from the cancers with available data in TCGA. To do so, go to the Broad Institute Genome Data Analysis Center (GDAC) at http://gdac.broadinstitute.org/. There will be a variety of cancers listed with an acronym, number of cases, previously performed analyses, and the link to the actual raw data. When choosing a cancer, choose one that has at least 500 cases including mRNA-Seq data (this will help improve the quality of the analysis). Be sure that there is RNA-seq data, not just microarray data. If you do not have any preferences, we recommend the data from Head and Neck Squamous Cell Carcinoma (HNSC).

When you go to the data link, there will be a few categories of primary data available. We are most interested in two pieces of data, the clinical data (to be able to gauge survival over time) as well as the gene expression data. The clinical data can be found under the "Clinical" heading in **Merge_Clinical**. When you unzip the file, you'll want the "{CANCER}.clin.merged.txt" file. The gene expression data can be found under the "mRNASeq" heading in "illuminahiseq_rnaseqv2-RSEM_genes_normalized" or a similar name that will say

RSEM_genes_normalized. It will also be a zipped folder that you open to get the text file with the gene counts for each patient. If there are multiple RSEM_genes_normalized (for example, from illuminahiseq and illuminga) use both files to get all patients' RNA-seq datasets. Do not get the isoform data.

For this question, note which cancer you have chosen and the number of cases. (5 pts)

Bonus: You've downloaded the normalized gene expression values. When getting publicly available data it is often important to know exactly what pre-processing was performed on the data. How was the data normalized? Hint: Look around on the GDAC website to find documentation.

2. We now need to clean the data to make it useable in survival analysis. To do this, first make a clinical data table that contains the following information from the original matrix: "patient.bcr_patient_barcode", "patient.days_to_death", "patient.days_to_last_followup", and "patient.vital_status". Remember that fields with days are counting days since the start of observation. This information is intentionally redundant, as there can and will be mistakes in clinical records that we want to catch and remove in our analysis. For the clinical table, use the redundancy across the metrics to remove patients that have inconsistent data - for example, if the patient's last follow up happened after the date of death, remove the patient from the panel. If there are any negative numbers, remove those patients. Add a column which tracks the time to last contact or event (whether it was the last follow-up or the days to death). Also, add one more column that says "alive" if the patient was still alive at the end of observation and "dead" if not. This column will be used for censoring.

How many patients were still alive at the end of observation? What is censoring in the context of survival analysis, and why is it important? (8 pts)

An additional variable that we consider is time. Many things may cause confounding as we follow patients for longer periods of time, so we often set a time cutoff so that we use a reasonable time window for survival analysis. If any patients have a time to last contact or event beyond 4 years (remember that the data is provided to you in days), convert that time into 4*365 days and mark them as "alive".

What might be some possible factors that could lead to confounding over a long time window? How many patients were still alive at the end of the cutoff observation? (7 pts)

3. Additionally, make a table for gene expression. To reduce noise and improve our modeling, we want to remove genes that have low read counts, since they are likely not expressed in the system. We will empirically choose the read count cutoff. To do so, transform the normalized count values with a log2 transform and then plot all read counts (except those that are 0) across all samples in a histogram. You should see a peak with a fat left tail (for those with a statistical background, we're seeing a mixture of a Gaussian and low-level uniform noise); choose a cutoff

that basically removes the fat left tail, and apply to your gene expression matrix, setting all read counts below that value to 0. Remove genes whose max value across all patients is below your chosen cutoff. From here on, use the log2-transformed counts for the following analyses. HINT: if you run into **`-Inf`** problems further on in the analysis, use **`asinh`** as a substitute for **`log2`** which gives a 0 value at 0 instead of **`-Inf`**.

Submit your density plot of log2-transformed read counts. What cutoff did you choose? How many genes are remaining after your threshold? (5 pts)

4. When we do survival analysis, we can only use cases that have both gene expression data as well as survival data. Determine the subset of patients that have both clinical data and gene expression data, and reduce both tables to be from that subset. HINT: some patients may have more than 1 RNA-seq experiment associated with them. Take the one with sample barcode 01 to use in your analysis (the cancer sample) and if there are multiple of those, take the first one, so that you have only one RNA-seq experiment per patient.

How many patients are in the subset that have both gene expression data and clinical data? (5 pts)

5. Before we actually do the analysis, it can be useful to do a quick visualization of the data. This allows us to quickly look for structure in the data, which can give us early confidence that our analyses will find something of interest. To do this, we take a random subsample of patients (up to 200 patients), and we take the top 2000 most variable genes (biggest standard deviations per each gene, when calculated across patients). Then, convert the read counts to z-scores (by row, ie for each gene across patients). Then, perform a k-means clustering (10 clusters) across the genes (as hierarchical clustering will be too computationally intensive for thousands of genes) and hierarchical clustering across patients, and plot the result. HINT: hierarchical clustering of patients can be done in the plotting step with **`heatmap.2`**. You can take a random subsample using **`sample`**.

Share your heatmap. Comment on any characteristics or structure you see in the data visualization (or any lack of characteristics you see). (10 pts)

6. Now we will build a univariate model, using one gene: CDKN1B (if using HNSC data). Or if you have another favorite gene, feel free to use that as well - this problem will not be graded on significance of your result, but rather correctness of the analysis. Work through the following steps to build your model.

a. For the survival data: first build a survival object (from the **`survival`** library in R, use **`Surv`**) that takes in the days to last contact (cutoff adjusted) as well as the censoring data.
b. Using **`coxph`**, build a univariate Cox model with your gene of interest. Remember to use the actual gene's values for the formula, as it is a regression model.

c. We now want to know whether this gene has an effect on patient outcomes, based on our regression model. Use **`summary`** to extract the following metrics: Wald Test, hazard ratio, confidence bounds (high and low) on the hazard ratio, and the z-score. Remember that the hazard ratio value is the exp(coefficient).
d. Then, calculate the median for this gene's values across the samples and then mark each sample with whether the gene is high or low, relative to the median. These will be your two groups of samples for prediction, one group with high expression of your gene of interest and one group with low expression. **Bonus:** use **`maxstat`** to find an optimal threshold. Comment on the difference between the median threshold and the maxstat threshold.
e. Now, if we actually used the two levels of gene expression (high vs low, thresholded by the median) as a prognostic indicator, how well does that metric actually separate the patients into a better prognostic group and a poor prognostic group? Use **`survdiff`** to get the log-rank test metric to check significance of the split of the groups, ie the significance level of the two groups being different. HINT: **`survdiff`** gives you a chi-squared statistic for a test of equality, using **`pchisq`** can give you the probability on a chi-squared statistic (make sure to set the degrees of freedom correctly).

Report the summary metrics for the regression model and the log-rank test metric for the median-thresholded group prediction. What is the meaning of each of these metrics and why are they important to know? (15 pts)

7. Plot the result as a Kaplan-Meier curve. Use **`survfit`** to set up your data in plot format. Submit your Kaplan-Meier curve plot. In a sentence or two, interpret the plot. (5 pts)

Part 2: Multiple hypothesis testing

1. In Part 1, we looked at one gene of interest. However, we have data on all genes from the RNA-seq experiment - we naturally will want to consider all genes to see which individual gene might be most predictive of outcome. Build a univariate model for every gene as in Part 1 (building a Cox regression model and then predicting outcomes using two groups based on the median value), and save the metrics to a table (there are 6 metrics, same as above). Do NOT plot a Kaplan-Meier curve for each, just collect the metrics for each gene. If a gene does not separate your patients into two classes, ignore the gene. Then, sort by the log-rank test p-value and submit the top 10 in a table. How many genes are significant to a log-rank test p-value of less than 0.05? What does it mean that the log-rank test p-value is less than 0.05? NOTE: this question may take many minutes to run! Pre-allocate your results table beforehand to save computation time. (10 pts)

2. When you test many genes for their predictive ability, this is a situation known as multiple hypothesis testing. Each gene is a hypothesis (is it predictive of outcome?), and you are testing

thousands of these hypotheses. This introduces a problem with using the p-value to test for significance. What is that problem, and why is the p-value inappropriate in multiple hypothesis testing? To correct this problem, we will convert these p-values to q-values within the False Discovery Rate (FDR) framework. Install the “qvalue” package (easiest through Bioconductor) and generate q-values for each gene. What is the q-value? Using q-values, how many genes are now significant? Submit the top 10 as a table. (10 pts)

Part 3: Multivariate Model

1. We’ve now seen that we can build predictive models using single genes. However, we can improve these models by using groups of genes (a multivariate model). Follow the steps below to build a regularized multivariate Cox regression model. We will be using **`glmnet`**. (20 pts)

a. Split your patients into a training dataset and a testing dataset. Perform an 75-25 split, where 75% of the data goes to training and 25% goes to test. Do this for both the gene expression data and the clinical data, making sure each patient’s gene expression and clinical data goes into the correct splits. Remember that, as with the univariate models, you will need survival objects using **`Surv`** - so be sure to make separate train and test survival objects. We will use the training set to build the model (with cross validation), and the test set will be used to determine the effectiveness of the model. Why is it important to have a training and test set? HINT: use **`set.seed`** to make reproducible train and test sets.
b. At this point, we could simply throw our training set into glmnet and get a model. However, glmnet allows for an optimization of lambda (your regularization parameter). Using the training data and the function **`cv.glmnet`**, where your nfolds is 3, determine the optimal (minimum) lambda value. What is regularization? (Just give an intuitive explanation) What is your optimal lambda value? What is cross validation and what are we using it for here? HINT: you are making a Cox regression, keep the alpha parameter as 1.
c. Now using **`glmnet`** on the training data with your optimal lambda value, build your model.
d. Using **`predict`**, get predictions of patient outcomes based on your model on your test set. Using these test predictions, build a Cox model and extract the same summary statistics you pulled from the univariate models. Use the median of the predictions to separate the predictions into two groups (above and below the median threshold) and use **`survdiff`** to get the log-rank test metric as well. How do your metrics compare to your best univariate model?

2. Plot the result on the test data as a Kaplan-Meier curve. Use **`survfit`** to set up your data in plot format. Submit your Kaplan-Meier curve plot. In a sentence or two, interpret the plot. (5 pts)

3. Your model should have most coefficients for variables set to 0, with a few variables having real coefficients. Extract those genes and provide those genes and their coefficients as a table. Provide some biological interpretation - does it make sense that these genes should help in predicting patient outcomes? Do the signs of the coefficients make sense, given the gene functions? Provide 3-5 sentences at most. (10 pts)

Bonus: there are a variety of other datasets available on TCGA. Build a more extensive model that utilizes these other datasets (methylation, imaging, miRNA, etc) and report back the relevant metrics, a Kaplan-Meier curve, and a few sentences of biological interpretation.

Useful notes

Be very aware of character vs numeric. R can still sort character strings and will sort those differently from if the values were numeric. Remember to set stringsAsFactors to FALSE. **`apply`** will be very useful for things like **`max`**, **`scale`**, etc.

Submission

Please follow the instructions on the course website that explain how to move your files onto corn, test that your code runs there, and actually submit your work.

You will need to submit the following files:
1. All code/scripts. We need be able to run your code on corn. Make sure we will be able to do so (test it yourself, especially if you developed it on your own computer!). Your code must be well commented. Try to adhere to Google's R style guide as best as possible.
2. A file called “readme.txt” explaining your technical code details. Write down exactly how to run your code. If you used libraries that we should install, note them here.
3. A PDF file called “ps2.pdf”, which contains answers to our questions. This is where you explain your work, which is important for assigning partial credit. Please try to be as concise as possible.

Grade breakdown

You will need to explain your work.

- 30 pts for well-commented working code. You can get partial credit for partially working code or non-working code that is well-commented.
- 5 pts for readme.txt that clearly, concisely describes how to run your code.
- 115 pts for ps2.pdf, for answers to the questions
- Total: 150 points

Collaboration policy

You are encouraged to discuss your problems in the piazza group page. You must submit your own individual work and are discouraged from working on the problem set or comparing answers with others.