RNA Sequencing Differential Analysis Project

Gavin Fortenberry

July 25, 2018

Contents

Introduction

Myleofibrosis is a type of bone marrow cancer in which an rapidly increasing number of blood forming cells form a fibrous like structure that sometimes leads to acute leukemia. Certain genotypes like the JAK2 V617F mutation have been a determining factor of Blast Transformation in Myleofibrosis. - What is differential gene expression? Differential gene expression is a way to analyze factors in different groups that may or may not be associated with different gene counts, from RNA sequence data.

- The groups of phenotype data being used can be sorted into two overarching categories: biological factors and technical factors. Biological factors include "Tissue_type", "genotype_jak2", "genotype_calr", and "genotype_mpl". Technical factors include: "collection_type", "time_to_processing", and "extraction_type".
- The reason why I am splitting the data/analysis into two categories is to attempt to see whether certain factors from each category have an impact on specific gene counts or not.
- To be completed: Why im picking each of the columns within those sections
- I will separate demographic groups by phenotype, and compare gene expression counts for each of the genes between the two groups, groups defined by conditions of that factor. For example, the conditions of the factor of age could be different age groups, sex could be male or female, genotype could be present or not present, etc. ultimate goal is to make observations/conclusions about possible differences in gene counts between different groups that may or may not be significant in diagnosis of Myleofibrosis.

Data Import

Install R Packages

Installs packages of functions that can be used to help manipulate variables and data libraries. These packages can be installed through base R code that downloads them and installs them into R studio. Then certain programming phrases can be used to do new functions in connection with base R programming phrases, with an indicator of the package being used mentioned before using that package's functions. For example, if I wanted to use a function of the "dplyr" package, I would write -> "dplyr::(insert function here)(insert arguments of functions here)". Note that functions or phrases that are in base R programming vocabulary

do not require a name indicator of the package being used, as it is from the base software and the software knows how to interperet those phrases without an idicator of a package. It is still required to write names of functions from Base R that are wanting to be used though, and if shortcut variables for functions are desired then you can assign new variables to those functions.

```
install.packages("dplyr")
install.packages("knitr")
install.packages("rmarkdown")
```

Load Libraries

```
library(plyr); library(dplyr)
```

Read in Project Metadata to R

Reads the two CSV's and combines them into one data frame: The code below creates three data frames: one labeled as "df_pheno" for data of a file called phenotable, which contains demographics and general information from patients and samples taken from them, another labeled as "df_molecular" for data from another file called molecularDataSets, which contains more specific info on samples collected from patients and their diagnoses from these samples, and another labeled as "df_combined" that combines the two data frames according to common columns.

The overall purpose of this code chunk is to create the combined data frame.

The first two data frames created ("df_pheno" and "df_molecular") were created using base R code "read.csv" function, which reads in data into a data frame from a CSV, a comma seperated values document.

In the process of formatting the df_pheno dataframe in order to be compatible when joining with the df_pheno dataframe, the "rename" function from the dplyr package is used on the df_pheno dataframe to change the name of a mismatching column name.

The function used to actually combine the two dataframes into one is the "full_join" function from the dplyr package which joins two indicated dataframes together by a common column.

"str()" is short for structure, and is a base R function that allows the user to get an idea of the format of the data they are looking at

```
df_pheno <- read.csv(file="../RNASeqData/phenotypeTable.csv",
                     header =TRUE)
df_pheno <-df_pheno %>% rename( assay_material_id = ends_with("assay_material_id"))

df_molecular <- read.csv(file="../RNASeqData/molecularDataSets.csv",
                         header =TRUE, sep=",")

df_combined <- dplyr::full_join(df_pheno, df_molecular,
                                by = "assay_material_id")
head(df_combined, n = 3)
```

```
##   assay_material_id     sex    race age_range diagnosis collection_event
## 1             R0297 unknown unknown   unknown        mf        diagnosis
## 2             R0299    male unknown elderly61        mf        diagnosis
## 3             R0298 unknown unknown   unknown        mf        diagnosis
##   acquisition_date     tissue_type tissue_type_origin collection_type
## 1          8/14/13 peripheralblood                 NA            edta
## 2          8/22/13 peripheralblood                 NA            edta
```

```
## 3          8/19/13 peripheralblood                 NA            edta
##    processing_type   time_to_processing       storage shipping
## 1 mononuclearcells repositoryprocessing cryopreserved    fresh
## 2 mononuclearcells repositoryprocessing cryopreserved    fresh
## 3 mononuclearcells repositoryprocessing cryopreserved    fresh
##   genotype_jak2 genotype_calr  genotype_mpl material_type extraction_type
## 1      positive      negative notdetermined           RNA          column
## 2      negative      positive notdetermined           RNA          column
## 3      negative      positive notdetermined           RNA          column
##   na_260280 na_260230   rin_range jak2_vaf molecular_id genomics_types
## 1      2.05      1.86 highquality       70    M00000298         rnaseq
## 2      2.05      1.81 highquality        0    M00000300         rnaseq
## 3      2.07      1.78 highquality        0    M00000299         rnaseq
##   omics_sample_name omics_contact_id omics_date
## 1        JAK2-6-1-D          jradich    2/27/14
## 2       JAK2-10-1-D          jradich    2/27/14
## 3           MF-D-07          jradich    3/24/14
##                seq_flowcell_id seq_readlength seq_paired seq_libtype
## 1 140227_SN367_0370_AH8JPDADXX             99        yes      truseq
## 2 140227_SN367_0370_AH8JPDADXX             99        yes      truseq
## 3 140324_SN367_0381_BH929TADXX             99        yes      truseq
```

Save Prepared Metadata

Writes a CSV (a comma seperated values document) from the new combined data frame(df_combined) of dataframes df_pheno and df_molecular into a document called "combineddata.csv" stored in the working directory (the main location of the files created from the Rstudio

```
write.csv(df_combined, file = "../ProjectFiles/combineddata.csv",
          row.names = FALSE)
```

The read.csv function is used to re-read the newly written CSV to make sure the data is the same as it was when written. The "str()" and "summary" functions are used to compare the statistics of the new dataframe to the original created one to confirm similarity.

```
test_set_combined_data <- read.csv("combineddata.csv")
str(test_set_combined_data)
```

Read in RNA Sequencing Data from HiSat2/htseqcount

Makes a list of directories(folders) with the R base function "list.dirs" which takes the indicated path of the main directory containing the directories its making a list of in, and the econd argument written "recursive" is set to false becuse the main directory is not desired to be listed in the list of its components.

```
#":/Users/gfortenb/Documents/GitHub/bioDS-bootcamp/RNASeqData"
RNADirectoryList = list.dirs(path = "../RNASeqData", recursive = FALSE)
```

Makes a List of files within each folder of each directory in the main directory. Uses a function to go through the files within each folder and only list files with a certain phrase in the name of the file, "htseq.txt", using the pattern function (only argument usedin function is name of character phrase its looking for).

Binds path's of files(locations of the files in the computer) on the list made to the molecular ID of the files with "as.data.frame"" function"/"cbind" function, created by finding key character sequences in the title of the folders containing the "htseq.txt"" files, using the gsub function (from base R). The first argument used

in the gsub function is a character phrase that indicates where in the string to look for the character phrase of the molecular ID, and the second argument is the list of data of file locations/names to look for the ID in.

The "colnames" function (base R) sets the column names if the newly created dataframe.

```
FileList1 = sapply(RNADirectoryList,
                   function(x){list.files(path = x,
                                          full.names = TRUE,
                                          pattern = "htseq.txt") })

FileIDList <- as.data.frame(cbind(FileList1,
                                  gsub("^.*-","", RNADirectoryList)),
                            stringsAsFactors = F)
colnames(FileIDList) <- c("Path", "molecular_id")
head(FileIDList, n = 3)
```

```
##
## ../RNASeqData/JAK2-10-1-D-R0299-M00000300 ../RNASeqData/JAK2-10-1-D-R0299-M00000300/JAK2-10-1-D.htsec
## ../RNASeqData/JAK2-30-D-R0301-M00000302       ../RNASeqData/JAK2-30-D-R0301-M00000302/JAK2-30-D.htsec
## ../RNASeqData/JAK2-36-D-R0303-M00000304       ../RNASeqData/JAK2-36-D-R0303-M00000304/JAK2-36-D.htsec
##                                           molecular_id
## ../RNASeqData/JAK2-10-1-D-R0299-M00000300    M00000300
## ../RNASeqData/JAK2-30-D-R0301-M00000302      M00000302
## ../RNASeqData/JAK2-36-D-R0303-M00000304      M00000304
```

Reads all data from list of selected files and compiles into different data frames/list of different data frames using LApply function.

```
listOf_alldf <- lapply(seq(1:nrow(FileIDList)),
                  function(i){
                    X <- read.delim(file = FileIDList$Path[i],
                                    header = FALSE);
                    colnames(X) <- c("Gene", FileIDList$molecular_id[i]);
                    return(X)
                    } )
```

TO DO: Prepare RNA Sequencing Data for Analysis

Make the list of dataframes into one dataframe w/all contents of each dataframe in the dataframe of dataframes as columns using the "join_all" function from the plyr package. This produces a dataframe with the molecular ID and Genes in each sample assigned to a molecular ID columns.

```
all_joined_df <- plyr:: join_all(listOf_alldf, by = NULL,
                                 type = "full", match = "all")
head(all_joined_df, n = 3)
```

```
##       Gene M00000300 M00000302 M00000304 M00000305 M00000298 M00000297
## 1     A1BG       226       138       233       438       130       148
## 2 A1BG-AS1         7         7         8         4        10         6
## 3     A1CF         0         0         0         0         0         0
##   M00000299 M00000301 M00000303 M00000306 M00000307 M00000308 M00000019
## 1        88       163        88       132       232       240       119
## 2         0         7         6         0         6        22        10
## 3         0         0         0         0         0         0         0
##   M00000020 M00000021 M00000022 M00000002 M00000001 M00000003 M00000004
```

```
## 1        164        230        254        129         63        140        128
## 2         26         39         43         21         13         29         32
## 3          0          0          1          0          0          2          1
##    M00000007 M00000005 M00000006 M00000008
## 1        160        125        133        203
## 2         28         52         25         45
## 3          5          0          3          3
```

Create a Summarized Experiment Data set

Normalize RNA Sequencing Counts - in new normalized data frame

(Normalize each sample's counts data based on over all library size for each sample.)

TO DO: Differential Gene Expression Analysis with DESeq2

Install Bioconductor, the DESeq2 Package and ggplot2

Bioconductor packages need to be installed by "biocLite" rather than install.packages which is for CRAN/base R

Load Additional Packages

```
library(DESeq2); library(ggplot2)
```