Zemin_CRC_GSE108989-CCR8_Analaysis

Hyun Yong Jin

August 6, 2019

Ver1.1 as of 20190808 Code readability has been improved.

1. Downloaded raw table from GSE108989. This table contained normalized gene expression (12547 genes) of single cells (10807 cells).
2. Downloaded table from Tamatoa. This table includes individual cell ID and cluster information, but less number of cells (7172). I assume they removed some cells with less confident analysis.
3. Merged two tables. Now I have individual cell (7172) with gene expression profile (12546, removed one un-assigned gene).
4. Select cells assigned to CD4_C12-CCR8. Down to 1042 cells.
5. Starting from the CCR8 cluster, I separated the individual cells into two groups. Cells belong to CCR8hi (log2 >8, 330 cells) and CCR8 low (1<log2<4, 47 cells). About half of the cells does not even have significant CCR8 but still clustered as same cluster because other gene expression patterns contributed to the clustering. I focused on cells with significant CCR8 expression.
6. From this point on, I treated the individual cells from CCR8hi group (330 cells) and low group (47 cells) as biological replicates for calculating statistics.
7. I calculated mean, FC, SD, p values and other statistics per individual genes.
8. From this stat(stat_all.csv), I selected FC > 5 and p values <0.01 genes. This table is attached, showing upregulated gene list in CCR8hi cells. CCL22 was the top hit and CCR8 was the third hit.
9. From this stat(stat_all.csv), I selected FC<0.2 genes and p values <0.2 . This table shows downregulated gene list in CCR8hi cells. Stat is very loosened because lowly detected genes have very poor statistics. SIRT1 was downregulated.

```
knitr::opts_chunk$set(fig.width=12, fig.height=8, fig.path='Output/',
                      warning=FALSE)
```

Download Data

```
#GSE108989
library(GEOquery)
```

```
## Loading required package: Biobase
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
```

```
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Setting options('download.file.method.GEOquery'='auto')
```

```
## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

```
getGEOSuppFiles('GSE108989', fetch_files = FALSE) #Check to see what is in the supplement file
s
```

```
##                                              fname
## 1 GSE108989_CRC.TCell.S10805.norm.centered.txt.gz
## 2           GSE108989_CRC.TCell.S11138.TPM.txt.gz
## 3         GSE108989_CRC.TCell.S11138.count.txt.gz
##
                      url
## 1 https://ftp.ncbi.nlm.nih.gov/geo/series/GSE108nnn/GSE108989/suppl//GSE108989_CRC.TCell.S1
0805.norm.centered.txt.gz
## 2           https://ftp.ncbi.nlm.nih.gov/geo/series/GSE108nnn/GSE108989/suppl//GSE108989_CR
C.TCell.S11138.TPM.txt.gz
## 3         https://ftp.ncbi.nlm.nih.gov/geo/series/GSE108nnn/GSE108989/suppl//GSE108989_CRC.
TCell.S11138.count.txt.gz
```

```
#There are three normalization data. Based on their method section, norm.centered.txt.gz is th
e most relevant dataset.
```

```
#Download all, and select "1 GSE108989_CRC.TCell.S10805.norm.centered.txt.gz"

getGEOSuppFiles('GSE108989') #All three files were downloaded in sub-folder /GSE108989
```

```
##
      size
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S10805.norm.centered.txt.gz
 386443604
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.TPM.txt.gz
 368657292
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.count.txt.gz
  73058088
##
 isdir
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S10805.norm.centered.txt.gz
 FALSE
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.TPM.txt.gz
 FALSE
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.count.txt.gz
 FALSE
##
 mode
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S10805.norm.centered.txt.gz
  666
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.TPM.txt.gz
  666
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.count.txt.gz
  666
##
               mtime
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S10805.norm.centered.txt.gz
 2019-08-08 14:01:04
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.TPM.txt.gz
 2019-08-08 14:01:50
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.count.txt.gz
 2019-08-08 14:01:59
##
               ctime
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S10805.norm.centered.txt.gz
 2019-08-06 10:05:50
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.TPM.txt.gz
 2019-08-06 10:07:10
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.count.txt.gz
 2019-08-06 10:07:56
##
               atime
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S10805.norm.centered.txt.gz
 2019-08-06 10:05:50
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.TPM.txt.gz
 2019-08-06 10:07:10
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.count.txt.gz
 2019-08-06 10:07:56
##
```

```
 exe
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S10805.norm.centered.txt.gz
  no
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.TPM.txt.gz
  no
## C:/Users/hjin02/Desktop/Zemin_CRC/GSE108989/GSE108989_CRC.TCell.S11138.count.txt.gz
  no
```

Analysis from the downloaded table

Transpose data

https://stackoverflow.com/questions/6778908/transpose-a-data-frame ##Merge data frame
https://stackoverflow.com/questions/29511215/convert-row-names-into-first-column ##Inner_join
https://rpubs.com/NateByers/Merging

```
## Row data contains gene ID and expression level of each genes. However, cluster information
and selection of cells (excluding low quality cells) information is missing. In contrast, down
loaded file from Tamatoa contains cluster info, and lower number of cells, presumably removed
the low quality cells.

## Idea: Both table contains cell-ID as common identifier. Merge the two table together for do
wnstream analysis.

library(tidyverse)
```

```
## -- Attaching packages ----------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.1     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts -------------------------------------- tidyverse_conflicts() --
## x dplyr::combine()    masks Biobase::combine(), BiocGenerics::combine()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x ggplot2::Position() masks BiocGenerics::Position(), base::Position()
```

```
tab <- read.delim("GSE108989/GSE108989_CRC.TCell.S10805.norm.centered.txt.gz") #Load file as t
ab delimited txt.

dim(tab)
```

```
## [1] 12547 10807
```

```
tab[c(1:10), c(1:20)]
```

```
##       geneID  geneSymbol NP710.20180123 NP711.20180123 NP71.20180123
## 1          1        A1BG    -0.51541173    -0.51541173    5.51817791
## 2        100         ADA    -1.86224381    -1.86224381   -1.86224381
## 3      10000        AKT3    -0.45803531    -0.45803531   -0.45803531
## 4  100009676  ZBTB11-AS1    -0.66320498    -0.66320498   -0.66320498
## 5      10001        MED6    -1.04021399    -1.04021399    7.16371049
## 6      10003     NAALAD2    -0.10281767    -0.10281767   -0.10281767
## 7  100033438 SNORD116-26    -0.06330997    -0.06330997   -0.06330997
## 8  100037417        DDTL     3.79442161    -1.18408594   -1.18408594
## 9      10004    NAALADL1    -0.31698561    -0.31698561   -0.31698561
## 10 100048912  CDKN2B-AS1    -0.06244128    -0.06244128   -0.06244128
##    NP712.20180123 NP713.20180123 NP714.20180123 NP718.20180123
## 1     -0.51541173     4.82521559    -0.51541173    -0.51541173
## 2     -1.86224381    -1.86224381     4.37329826    -1.86224381
## 3     -0.45803531    -0.45803531    -0.45803531    -0.45803531
## 4      2.58352364    -0.66320498     2.87551653    -0.66320498
## 5     -1.04021399    -1.04021399    -1.04021399     0.49447183
## 6     -0.10281767    -0.10281767    -0.10281767    -0.10281767
## 7     -0.06330997    -0.06330997    -0.06330997    -0.06330997
## 8     -1.18408594    -1.18408594     2.73829929    -1.18408594
## 9     -0.31698561    -0.31698561    -0.31698561     1.94439675
## 10    -0.06244128    -0.06244128    -0.06244128    -0.06244128
##    NP720.20180123 NP721.20180123 NP72.20180123 NP724.20180123
## 1     -0.51541173     2.87784487   -0.51541173    -0.51541173
## 2     -1.86224381    -1.86224381    4.99137433     8.34309578
## 3     -0.45803531    -0.45803531   -0.45803531    -0.45803531
## 4     -0.66320498    -0.66320498    5.46988964    -0.66320498
## 5     -1.04021399    -1.04021399   -1.04021399    -1.04021399
## 6     -0.10281767    -0.10281767   -0.10281767    -0.10281767
## 7     -0.06330997    -0.06330997   -0.06330997    -0.06330997
## 8     -1.18408594    -1.18408594   -1.18408594     5.30138033
## 9     -0.31698561    -0.31698561   -0.31698561    -0.31698561
## 10    -0.06244128    -0.06244128   -0.06244128    -0.06244128
##    NP727.20180123 NP728.20180123 NP731.20180123 NP73.20180123
## 1     -0.51541173    -0.51541173    -0.51541173   -0.51541173
## 2     -1.86224381     4.70996069    -1.86224381   -0.22191188
## 3     -0.45803531    -0.45803531    -0.45803531   -0.45803531
## 4     -0.66320498    -0.66320498    -0.66320498   -0.66320498
## 5     -1.04021399    -1.04021399     8.44352200    0.60011794
## 6     -0.10281767    -0.10281767    -0.10281767   -0.10281767
## 7     -0.06330997    -0.06330997    -0.06330997   -0.06330997
## 8      3.45387168    -1.18408594    -1.18408594   -1.18408594
## 9     -0.31698561    -0.31698561     9.27221833   -0.31698561
## 10    -0.06244128    -0.06244128    -0.06244128   -0.06244128
##    NP732.20180123 NP734.20180123 NP735.20180123
## 1     -0.51541173    -0.51541173    -0.51541173
## 2      5.44462583    -1.86224381    -1.86224381
## 3      8.64238711    -0.45803531    -0.45803531
## 4     -0.66320498    -0.66320498    -0.66320498
## 5     -1.04021399    -1.04021399    -1.04021399
## 6     -0.10281767    -0.10281767    -0.10281767
## 7     -0.06330997    -0.06330997    -0.06330997
## 8      4.43436171    -1.18408594    -1.18408594
## 9     -0.31698561    -0.31698561     1.69219676
## 10    -0.06244128    -0.06244128    -0.06244128
```

```
#Start testing with small scale example.
tab_test<- tab[c(1:10), c(1:5)]
n1 <- tab_test$geneSymbol
t.tab_test <- as.data.frame(t(tab_test[,-c(1:2)])) #remove 1st and 2nd column and transpose
colnames(t.tab_test) <- n1
str(t.tab_test)
```

```
## 'data.frame':    3 obs. of  10 variables:
##  $ A1BG       : num  -0.515 -0.515 5.518
##  $ ADA        : num  -1.86 -1.86 -1.86
##  $ AKT3       : num  -0.458 -0.458 -0.458
##  $ ZBTB11-AS1 : num  -0.663 -0.663 -0.663
##  $ MED6       : num  -1.04 -1.04 7.16
##  $ NAALAD2    : num  -0.103 -0.103 -0.103
##  $ SNORD116-26: num  -0.0633 -0.0633 -0.0633
##  $ DDTL       : num  3.79 -1.18 -1.18
##  $ NAALADL1   : num  -0.317 -0.317 -0.317
##  $ CDKN2B-AS1 : num  -0.0624 -0.0624 -0.0624
```

```
#Transpose the original tab table.
n2 <- tab$geneSymbol
t.tab <- as.data.frame(t(tab[,-c(1:2)])) #remove 1st and 2nd column and transpose
colnames(t.tab) <- n2
t.tab[c(1:5), c(1:6)] #sanity test
```

```
##                       A1BG       ADA       AKT3 ZBTB11-AS1      MED6
## NP710.20180123 -0.5154117 -1.862244 -0.4580353  -0.663205 -1.040214
## NP711.20180123 -0.5154117 -1.862244 -0.4580353  -0.663205 -1.040214
## NP71.20180123   5.5181779 -1.862244 -0.4580353  -0.663205  7.163710
## NP712.20180123 -0.5154117 -1.862244 -0.4580353   2.583524 -1.040214
## NP713.20180123  4.8252156 -1.862244 -0.4580353  -0.663205 -1.040214
##                   NAALAD2
## NP710.20180123 -0.1028177
## NP711.20180123 -0.1028177
## NP71.20180123  -0.1028177
## NP712.20180123 -0.1028177
## NP713.20180123 -0.1028177
```

```
#Next is to combine t.tab with data from tamatoa

id<-read.csv("Tamatoa/identifier-cluster_matching.csv", header=T)

#Now issue is the identifier in t.tab is rowname (without header) and id is in column. Both table is data.frame format.
#to merge tables, the cell id in t.tab has to be assigned.

rownames_test <- tibble::rownames_to_column(t.tab_test,"VALUE") #test with small table
```

```
t.tab <-tibble::rownames_to_column(t.tab, "cell_names") #t.tab was overwritten but column was
assigne.

test <- !is.na(names(t.tab)) #64th column nas NA (not assigned) header. Only one missing heade
r. All other headers were fine.

#Let's remove column with header NA from t.tab
t.tab <- t.tab[test] #re-assign only TRUE values

dim(t.tab) #10805 x 12547 (12548 before removing NA)
```

```
## [1] 10805 12547
```

```
dim(id) #7172 x 9
```

```
## [1] 7172    9
```

```
#Inner_join will merge table based on common identifier in the same column name. This function
 is part of dplyr
t.tab$cell_names[1:10]
```

```
##  [1] "NP710.20180123" "NP711.20180123" "NP71.20180123"  "NP712.20180123"
##  [5] "NP713.20180123" "NP714.20180123" "NP718.20180123" "NP720.20180123"
##  [9] "NP721.20180123" "NP72.20180123"
```

```
class(t.tab$cell_names) #character
```

```
## [1] "character"
```

```
id$cell_names[1:10]
```

```
##  [1] NTC10-20170215 NTC11-20170215 NTC1-20170215  NTC13-20170215
##  [5] NTC14-20170215 NTC15-20170215 NTC16-20170215 NTC17-20170215
##  [9] NTC18-20170215 NTC19-20170215
## 7172 Levels: NTC1-0909-ZL NTC1-20161212 NTC1-20161228 ... TTY99-20161012
```

```
class(id$cell_names) #factor
```

```
## [1] "factor"
```

```
#id$cell_names should be converted to character. For example, second column tSNE1 has numeric
value.
#https://stackoverflow.com/questions/2851015/convert-data-frame-columns-from-factors-to-charac
ters

i <- sapply(id, is.factor)
```

```
id[i] <- lapply(id[i], as.character)
class(id$cell_names) #Now the cell_names column is converted to character
```

```
## [1] "character"
```

```
#Merging step.
#There are a few issues so I resolved them.
t.tab$cell_names[10] #Id was connected by dot
```

```
## [1] "NP72.20180123"
```

```
id$cell_names[10] #Id was connected by hyphen
```

```
## [1] "NTC19-20170215"
```

```
#convert hyphen to dot in cell_names columne in id
?gsub #pattern matching and replacement
```

```
## starting httpd help server ...
```

```
##  done
```

```
id$cell_names <- gsub("-", ".", id$cell_names)

#Merge two table and excluded cells from no common id. Used inner_join
merged <- inner_join(id, t.tab, by = "cell_names")
dim(merged) #7172 12555
```

```
## [1]  7172 12555
```

```
#Note that id file (from tamatoa) has 7172, and row file has 10805 cells. Among them, 7172 was overlapped. I assume the 3000 cells were removed due to the low expression.


merged[c(1:8), c(1:12)] #Sanity test.
```

```
##       cell_names     tSNE1     tSNE2       Cluster Patient SampleType
## 1 NTC10.20170215 -16.55942 -26.83424   CD8_C05-CD6   P0215        NTC
## 2 NTC11.20170215 -16.45410 -23.27989   CD8_C05-CD6   P0215        NTC
## 3  NTC1.20170215 -16.46778 -15.64138  CD8_C04-GZMK   P0215        NTC
## 4 NTC13.20170215 -18.40049 -26.08195   CD8_C05-CD6   P0215        NTC
## 5 NTC14.20170215 -13.93536 -27.55328   CD8_C05-CD6   P0215        NTC
## 6 NTC15.20170215 -19.47082 -25.97597   CD8_C05-CD6   P0215        NTC
## 7 NTC16.20170215 -29.78458 -25.04256 CD8_C06-CD160   P0215        NTC
## 8 NTC17.20170215 -30.64388 -29.42102 CD8_C06-CD160   P0215        NTC
##   stype invariantTCR         Units      A1BG       ADA         AKT3
## 1   CD8      diverse log2(TPM + 1)  3.904416 -1.429880 -0.003865219
## 2   CD8      diverse log2(TPM + 1) -1.077304  6.728823 -0.793387736
## 3   CD8      diverse log2(TPM + 1) -1.077304 -2.219403  4.671928970
## 4   CD8      diverse log2(TPM + 1) -1.077304 -2.219403 -0.793387736
## 5   CD8      diverse log2(TPM + 1) -1.077304 -2.219403  0.636907032
## 6   CD8      diverse log2(TPM + 1) -1.077304 -2.219403 -0.793387736
## 7   CD8      diverse log2(TPM + 1) -1.077304 -2.219403  7.966519913
## 8   CD8      diverse log2(TPM + 1) -1.077304  2.386216 -0.793387736
```

```
write.csv(merged, file="merged_all.csv")

#Selection of CCR8 cluster only
merged_ccr8only<- merged %>% filter(Cluster =="CD4_C12-CCR8")
dim(merged_ccr8only) # 1042 12555 #~1/7 cells were ccr8+ cluster
```

```
## [1]  1042 12555
```

```
write.csv(merged_ccr8only, file="merged_ccr8.csv")
#Export the merged dataset.
```

```
#How about rowVars? But rowVars detects the most variable genes between individual replicates.

merged_ccr8only[c(1:50), c(1:12)]
```

```
##         cell_names       tSNE1    tSNE2      Cluster Patient SampleType
## 1   NTH14.20170215   4.1966610 30.39126 CD4_C12-CCR8   P0215        NTH
## 2   NTH50.20170215   3.0459613 25.21051 CD4_C12-CCR8   P0215        NTH
## 3   NTR10.20170215   2.7369125 36.85508 CD4_C12-CCR8   P0215        NTR
## 4   NTR11.20170215  -5.9585143 42.93301 CD4_C12-CCR8   P0215        NTR
## 5    NTR1.20170215   0.9936695 32.19331 CD4_C12-CCR8   P0215        NTR
## 6   NTR12.20170215  -4.1315597 33.77234 CD4_C12-CCR8   P0215        NTR
## 7   NTR15.20170215   0.9260567 33.48127 CD4_C12-CCR8   P0215        NTR
## 8   NTR17.20170215  -1.4855544 35.62769 CD4_C12-CCR8   P0215        NTR
## 9   NTR20.20170215  -4.8935730 43.74594 CD4_C12-CCR8   P0215        NTR
## 10  NTR21.20170215  -4.8880420 45.13257 CD4_C12-CCR8   P0215        NTR
## 11   NTR2.20170215  -3.1531354 45.50898 CD4_C12-CCR8   P0215        NTR
## 12   NTR4.20170215  -6.4693281 31.50204 CD4_C12-CCR8   P0215        NTR
## 13   NTR6.20170215   5.4560333 38.04615 CD4_C12-CCR8   P0215        NTR
## 14   NTR7.20170215   5.0017723 24.31386 CD4_C12-CCR8   P0215        NTR
## 15   NTR9.20170215  -3.7971227 44.78517 CD4_C12-CCR8   P0215        NTR
## 16  TTH10.20170215  -0.3311651 30.98576 CD4_C12-CCR8   P0215        TTH
## 17 TTH102.20170215  -3.7418018 22.57369 CD4_C12-CCR8   P0215        TTH
## 18 TTH122.20170215  -3.8231895 39.85634 CD4_C12-CCR8   P0215        TTH
## 19  TTH16.20170215   1.8756551 41.93914 CD4_C12-CCR8   P0215        TTH
## 20  TTH17.20170215   3.8084360 31.25363 CD4_C12-CCR8   P0215        TTH
## 21  TTH19.20170215  -4.2862863 17.33803 CD4_C12-CCR8   P0215        TTH
## 22  TTH28.20170215   5.5966068 28.78028 CD4_C12-CCR8   P0215        TTH
## 23  TTH50.20170215  -6.8525564 26.89016 CD4_C12-CCR8   P0215        TTH
## 24   TTH6.20170215  -1.6928506 20.54383 CD4_C12-CCR8   P0215        TTH
## 25  TTH76.20170215   1.3148417 43.32272 CD4_C12-CCR8   P0215        TTH
## 26  TTH85.20170215  -0.5425033 31.04282 CD4_C12-CCR8   P0215        TTH
## 27  TTH88.20170215  -2.0611939 33.18780 CD4_C12-CCR8   P0215        TTH
## 28  TTH89.20170215  -5.6135468 34.94681 CD4_C12-CCR8   P0215        TTH
## 29  TTH96.20170215   3.9579806 33.36961 CD4_C12-CCR8   P0215        TTH
## 30  TTH98.20170215   4.6451936 29.08508 CD4_C12-CCR8   P0215        TTH
## 31 TTR102.20170215  -6.5052937 43.97598 CD4_C12-CCR8   P0215        TTR
## 32 TTR104.20170215   2.7659632 36.90233 CD4_C12-CCR8   P0215        TTR
## 33 TTR108.20170215  -7.6930241 25.44515 CD4_C12-CCR8   P0215        TTR
## 34 TTR110.20170215   6.1271683 37.48952 CD4_C12-CCR8   P0215        TTR
## 35 TTR111.20170215   3.9719513 32.28359 CD4_C12-CCR8   P0215        TTR
## 36  TTR11.20170215  -3.0266363 45.29051 CD4_C12-CCR8   P0215        TTR
## 37 TTR114.20170215  -7.9398176 34.96567 CD4_C12-CCR8   P0215        TTR
## 38 TTR116.20170215   2.0217907 42.90070 CD4_C12-CCR8   P0215        TTR
## 39 TTR117.20170215  -3.7643328 44.67531 CD4_C12-CCR8   P0215        TTR
## 40 TTR119.20170215 -10.1664135 39.55565 CD4_C12-CCR8   P0215        TTR
## 41   TTR1.20170215  -3.7757006 45.13887 CD4_C12-CCR8   P0215        TTR
## 42 TTR120.20170215  -5.0997930 22.50982 CD4_C12-CCR8   P0215        TTR
## 43 TTR123.20170215  -0.8318460 43.43088 CD4_C12-CCR8   P0215        TTR
## 44 TTR124.20170215   0.9843536 33.35588 CD4_C12-CCR8   P0215        TTR
## 45 TTR125.20170215  -8.4349954 42.70343 CD4_C12-CCR8   P0215        TTR
## 46  TTR13.20170215  -1.8683164 45.17080 CD4_C12-CCR8   P0215        TTR
## 47  TTR14.20170215  -3.1471864 39.16172 CD4_C12-CCR8   P0215        TTR
## 48  TTR16.20170215  -0.7480891 35.19703 CD4_C12-CCR8   P0215        TTR
## 49  TTR17.20170215  -1.3429923 43.77283 CD4_C12-CCR8   P0215        TTR
## 50  TTR21.20170215  -4.2967627 38.12282 CD4_C12-CCR8   P0215        TTR
##    stype invariantTCR         Units        A1BG        ADA       AKT3
## 1    CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026  0.3834663
## 2    CD4      diverse log2(TPM + 1)  2.79622105 -1.4048718 -0.7933877
## 3    CD4      diverse log2(TPM + 1)  5.65500181 -2.2194026 -0.7933877
## 4    CD4      diverse log2(TPM + 1) -0.08840157  2.4030699 -0.7933877
## 5    CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 6    CD4      diverse log2(TPM + 1) -1.07730380  1.8135679 -0.7933877
## 7    CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 8    CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 9    CD4      diverse log2(TPM + 1) -0.48383380  4.2895055 -0.7933877
## 10   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 11   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 12   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 13   CD4      diverse log2(TPM + 1) -0.23467839 -2.2194026 -0.7933877
## 14   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 15   CD4      diverse log2(TPM + 1) -1.07730380 -1.3938988 -0.7933877
## 16   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 17   CD4      diverse log2(TPM + 1) -1.07730380 -0.4300742 -0.7933877
## 18   CD4      diverse log2(TPM + 1) -1.07730380 -1.2681352 -0.7933877
## 19   CD4      diverse log2(TPM + 1) -1.07730380 -1.2574555 -0.7933877
## 20   CD4      diverse log2(TPM + 1)  5.35855681 -2.2194026  5.3566647
## 21   CD4      diverse log2(TPM + 1) -1.07730380 -1.3221381 -0.7933877
## 22   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 23   CD4      diverse log2(TPM + 1) -1.07730380  2.2501199  6.2619334
## 24   CD4      diverse log2(TPM + 1) -1.07730380  3.0687655 -0.7933877
## 25   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026  0.2216188
## 26   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 27   CD4      diverse log2(TPM + 1)  4.18456877  5.2221475 -0.7933877
## 28   CD4      diverse log2(TPM + 1) -1.07730380  4.1400097 -0.7933877
## 29   CD4      diverse log2(TPM + 1) -1.07730380 -0.9607812 -0.7933877
## 30   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 31   CD4      diverse log2(TPM + 1) -1.07730380 -1.4874674 -0.7933877
## 32   CD4      diverse log2(TPM + 1) -1.07730380 -1.2621495 -0.7933877
## 33   CD4      diverse log2(TPM + 1) -1.07730380  3.6738557 -0.7933877
## 34   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 35   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 36   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 37   CD4      diverse log2(TPM + 1) -1.07730380  5.8021996 -0.7933877
## 38   CD4      diverse log2(TPM + 1) -1.07730380  5.6625432 -0.7933877
## 39   CD4      diverse log2(TPM + 1) -1.07730380 -1.3955893 -0.7933877
## 40   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 41   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 42   CD4      diverse log2(TPM + 1) -1.07730380  3.6412543 -0.7933877
## 43   CD4      diverse log2(TPM + 1) -1.07730380 -2.2194026 -0.7933877
## 44   CD4      diverse log2(TPM + 1)  2.95743725 -2.2194026 -0.7933877
## 45   CD4      diverse log2(TPM + 1) -1.07730380 -1.0164626 -0.7933877
## 46   CD4      diverse log2(TPM + 1) -1.07730380 -1.5533131 -0.7933877
## 47   CD4      diverse log2(TPM + 1) -0.45126668 -2.2194026 -0.7933877
## 48   CD4      diverse log2(TPM + 1)  3.17091242 -1.0265861 -0.7933877
## 49   CD4      diverse log2(TPM + 1)  5.29148492  6.1644368 -0.7933877
## 50   CD4      diverse log2(TPM + 1) -0.22696796 -2.2194026 -0.7933877
```

```
tbl <- merged_ccr8only %>%
  select(-cell_names, -tSNE1, -tSNE2, -Cluster, -Patient, -SampleType, -stype, -invariantTCR,
-Units) #select the necessary columnes only

tbl[c(1:20), c(1:10)]
```

```
##           A1BG        ADA       AKT3 ZBTB11-AS1       MED6    NAALAD2
## 1  -1.07730380 -2.2194026  0.3834663 -0.7937445 -1.8261215 -0.1578804
## 2   2.79622105 -1.4048718 -0.7933877 -0.7937445  5.1358763 -0.1578804
## 3   5.65500181 -2.2194026 -0.7933877 -0.7937445 -1.8261215 -0.1578804
## 4  -0.08840157  2.4030699 -0.7933877  0.1951578  4.2251793  0.2516189
## 5  -1.07730380 -2.2194026 -0.7933877 -0.7937445 -0.6442732 -0.1578804
## 6  -1.07730380  1.8135679 -0.7933877 -0.7937445 -1.8261215 -0.1578804
## 7  -1.07730380 -2.2194026 -0.7933877 -0.7937445 -1.8261215 -0.1578804
## 8  -1.07730380 -2.2194026 -0.7933877 -0.7937445  5.5503074 -0.1578804
## 9  -0.48383380  4.2895055 -0.7933877 -0.7937445  6.3988387 -0.1578804
## 10 -1.07730380 -2.2194026 -0.7933877 -0.7937445  3.9639649 -0.1578804
## 11 -1.07730380 -2.2194026 -0.7933877 -0.7937445 -1.8261215 -0.1578804
## 12 -1.07730380 -2.2194026 -0.7933877 -0.7937445 -1.8261215 -0.1578804
## 13 -0.23467839 -2.2194026 -0.7933877 -0.7937445 -1.8261215 -0.1578804
## 14 -1.07730380 -2.2194026 -0.7933877  3.6888721 -1.8261215 -0.1578804
## 15 -1.07730380 -1.3938988 -0.7933877 -0.7937445 -1.8261215 -0.1578804
## 16 -1.07730380 -2.2194026 -0.7933877 -0.7937445 -1.8261215 -0.1578804
## 17 -1.07730380 -0.4300742 -0.7933877 -0.7937445 -1.8261215 -0.1578804
## 18 -1.07730380 -1.2681352 -0.7933877 -0.7937445 -0.8748541 -0.1578804
## 19 -1.07730380 -1.2574555 -0.7933877 -0.7937445  4.8523629 -0.1578804
## 20  5.35855681 -2.2194026  5.3566647  4.7125904  0.6189161 -0.1578804
##    SNORD116-26      DDTL   NAALADL1 CDKN2B-AS1
## 1    0.9320694 -1.834174  0.5662379 -0.1570629
## 2   -0.2447846  4.566021 -0.6106161 -0.1570629
## 3   -0.2447846 -1.834174 -0.6106161 -0.1570629
## 4   -0.2447846  2.668083 -0.6106161 -0.1570629
## 5   -0.2447846  3.019363 -0.6106161 -0.1570629
## 6   -0.2447846 -1.834174 -0.6106161 -0.1570629
## 7   -0.2447846  4.232533 -0.6106161 -0.1570629
## 8   -0.2447846  2.867610 -0.6106161  0.8436362
## 9    0.3486854 -1.834174 -0.6106161 -0.1570629
## 10  -0.2447846  5.388901 -0.6106161 -0.1570629
## 11  -0.2447846 -1.834174 -0.6106161 -0.1570629
## 12  -0.2447846 -1.834174 -0.6106161 -0.1570629
## 13  -0.2447846 -1.834174  0.2320093 -0.1570629
## 14  -0.2447846  4.303501 -0.6106161 -0.1570629
## 15  -0.2447846 -1.834174 -0.6106161 -0.1570629
## 16  -0.2447846 -1.834174 -0.6106161 -0.1570629
## 17  -0.2447846 -1.834174 -0.6106161 -0.1570629
## 18  -0.2447846 -1.834174 -0.6106161 -0.1570629
## 19  -0.2447846 -1.834174 -0.6106161 -0.1570629
## 20  -0.2447846 -1.834174 -0.6106161 -0.1570629
```

```
#transpose data and maintain the first column as header
t_tbl <- as.data.frame(t(tbl)) #Transposing number only is much faster.
colnames(t_tbl) <- merged_ccr8only$cell_names #Add colname back

t_tbl[c(1:20), c(1:10)]
```

```
##             NTH14.20170215 NTH50.20170215 NTR10.20170215 NTR11.20170215
## A1BG            -1.0773038      2.7962211      5.6550018    -0.08840157
## ADA             -2.2194026     -1.4048718     -2.2194026     2.40306989
## AKT3             0.3834663     -0.7933877     -0.7933877    -0.79338774
## ZBTB11-AS1      -0.7937445     -0.7937445     -0.7937445     0.19515776
## MED6            -1.8261215      5.1358763     -1.8261215     4.22517934
## NAALAD2         -0.1578804     -0.1578804     -0.1578804     0.25161891
## SNORD116-26      0.9320694     -0.2447846     -0.2447846    -0.24478457
## DDTL            -1.8341736      4.5660206     -1.8341736     2.66808276
## NAALADL1         0.5662379     -0.6106161     -0.6106161    -0.61061609
## CDKN2B-AS1      -0.1570629     -0.1570629     -0.1570629    -0.15706290
## ACOT8           -1.0084181     -1.0084181     -1.0084181     0.84988045
## ABI1             5.5798733     -4.5190316      2.1230108     2.60859206
## GNPDA1          -1.1689779     -1.1689779     -1.1689779     6.48127597
## ZBTB33          -0.7109166     -0.7109166     -0.7109166    -0.71091660
## SNHG8           -2.9220104      6.2077393      3.3900494     2.88899197
## GTF2IP4         -0.6900022     -0.6900022      2.2069143     0.51970004
## TANK             4.6469376     -5.3770755      3.7601186     2.22524761
## POM121C         -1.6553951     -1.6553951      0.7502938     1.38390911
## ZSCAN30          0.6902883     -0.4865657     -0.4865657    -0.48656569
## MCTS2P          -0.3656114     -0.3656114     -0.3656114    -0.36561144
##             NTR1.20170215 NTR12.20170215 NTR15.20170215 NTR17.20170215
## A1BG           -1.0773038     -1.0773038     -1.0773038     -1.0773038
## ADA            -2.2194026      1.8135679     -2.2194026     -2.2194026
## AKT3           -0.7933877     -0.7933877     -0.7933877     -0.7933877
## ZBTB11-AS1     -0.7937445     -0.7937445     -0.7937445     -0.7937445
## MED6           -0.6442732     -1.8261215     -1.8261215      5.5503074
## NAALAD2        -0.1578804     -0.1578804     -0.1578804     -0.1578804
## SNORD116-26    -0.2447846     -0.2447846     -0.2447846     -0.2447846
## DDTL            3.0193628     -1.8341736      4.2325329      2.8676103
## NAALADL1       -0.6106161     -0.6106161     -0.6106161     -0.6106161
## CDKN2B-AS1     -0.1570629     -0.1570629     -0.1570629      0.8436362
## ACOT8          -1.0084181     -1.0084181     -1.0084181     -1.0084181
## ABI1            3.2233586      3.7079322      4.6880888      4.0215214
## GNPDA1         -1.1689779     -1.1689779     -1.1689779     -0.1682787
## ZBTB33          7.9713098     -0.7109166     -0.7109166     -0.7109166
## SNHG8          -2.9220104     -1.8531005      1.8694100     -2.9220104
## GTF2IP4        -0.6900022     -0.6900022     -0.6900022     -0.6900022
## TANK            3.1795343      2.9370679      2.7111245      3.6714437
## POM121C         3.0656194     -1.6553951     -1.6553951     -1.6553951
## ZSCAN30        -0.4865657     -0.4865657     -0.4865657     -0.4865657
## MCTS2P         -0.3656114     -0.3656114      1.5640776     -0.3656114
##             NTR20.20170215 NTR21.20170215
## A1BG            -0.4838338     -1.0773038
## ADA              4.2895055     -2.2194026
## AKT3            -0.7933877     -0.7933877
## ZBTB11-AS1      -0.7937445     -0.7937445
## MED6             6.3988387      3.9639649
## NAALAD2         -0.1578804     -0.1578804
## SNORD116-26      0.3486854     -0.2447846
## DDTL            -1.8341736      5.3889012
## NAALADL1        -0.6106161     -0.6106161
## CDKN2B-AS1      -0.1570629     -0.1570629
## ACOT8           -1.0084181     -1.0084181
## ABI1             4.6471271      4.4080336
## GNPDA1           5.1134546      6.4784364
## ZBTB33          -0.7109166     -0.7109166
## SNHG8            2.1469948     -2.9220104
## GTF2IP4          0.3227403     -0.6900022
## TANK             3.3364873      2.9580800
## POM121C          3.7269168      2.3078120
## ZSCAN30         -0.4865657      5.7383334
## MCTS2P           3.8678552     -0.3656114
```

```r
#Note that this filter is based on log2 value.
ccrhi <- tbl %>% filter(CCR8>8) %>% arrange(desc(CCR8)) #330 obs
ccrlo <- tbl %>% filter(CCR8>1 & CCR8 <4) %>% arrange(desc(CCR8)) #47 obs

#Now treat individual cells as individual replicate.
#Transpose the table
tccrhi <- as.data.frame(t(ccrhi))
tccrlo <- as.data.frame(t(ccrlo))

#Also note that stat should be done in linear values.
#Writing quick function to make the whole table to linear values
lin <- function (x, na.rm=FALSE) (2^x)
lin_tccrhi <- lin(tccrhi)
lin_tccrlo <- lin(tccrlo)

#Make sure both table contains same numbers of genes. 12546
dim(lin_tccrhi)
```

```
## [1] 12546   330
```

```
dim(lin_tccrlo)
```

```
## [1] 12546    47
```

```
#Calculate fold change CCRhi/CCRlo
datFC <- t_tbl %>%
  rownames_to_column("gene_name") %>%
  mutate(hi_mean = rowMeans(lin_tccrhi)) %>%
  mutate(lo_mean = rowMeans(lin_tccrlo)) %>%
  mutate(FC=hi_mean/lo_mean)

datFC[c(1:5), c(1:10)] #first column becomes gene_name
```

```
##    gene_name NTH14.20170215 NTH50.20170215 NTR10.20170215 NTR11.20170215
## 1       A1BG     -1.0773038      2.7962211      5.6550018    -0.08840157
## 2        ADA     -2.2194026     -1.4048718     -2.2194026     2.40306989
## 3       AKT3      0.3834663     -0.7933877     -0.7933877    -0.79338774
## 4 ZBTB11-AS1     -0.7937445     -0.7937445     -0.7937445     0.19515776
## 5       MED6     -1.8261215      5.1358763     -1.8261215     4.22517934
##   NTR1.20170215 NTR12.20170215 NTR15.20170215 NTR17.20170215
## 1    -1.0773038     -1.0773038     -1.0773038     -1.0773038
## 2    -2.2194026      1.8135679     -2.2194026     -2.2194026
## 3    -0.7933877     -0.7933877     -0.7933877     -0.7933877
## 4    -0.7937445     -0.7937445     -0.7937445     -0.7937445
## 5    -0.6442732     -1.8261215     -1.8261215      5.5503074
##   NTR20.20170215
## 1     -0.4838338
## 2      4.2895055
## 3     -0.7933877
## 4     -0.7937445
## 5      6.3988387
```

```
stat <- datFC %>%
  select(gene_name, hi_mean, lo_mean, FC) #extracting FC stats only. with rownames

#Calculate SD
library(genefilter)
```

```
##
## Attaching package: 'genefilter'
```

```
## The following object is masked from 'package:readr':
##
##     spec
```

```
datFCSD <- stat %>%
  mutate(hi_SD = rowSds(lin_tccrhi)) %>%
  mutate(lo_SD = rowSds(lin_tccrlo))

#Writing for-loop to calculate t-test in row-wise.
#See my 20190110 IBD analysis as reference

library(broom) #for tidy function
testresults <- vector("list", nrow(datFCSD))

#Start for-loop. Takes some time.
for (j in seq(nrow(datFCSD))) {
  testresults[[j]] <-tidy(t.test(as.data.frame(lin_tccrhi[j,]), as.data.frame(lin_tccrlo[j,]))
)
}

t_stats = do.call(rbind, testresults)
head(t_stats)
```

```
## # A tibble: 6 x 10
##   estimate estimate1 estimate2 statistic p.value parameter conf.low
##      <dbl>     <dbl>     <dbl>     <dbl>   <dbl>     <dbl>    <dbl>
## 1    1.57       5.62      4.06     0.762  0.448       81.7  -2.52
## 2   -5.76       9.64     15.4     -0.799  0.428       48.9 -20.3
## 3   -1.88       3.09      4.98    -0.423  0.674       48.9 -10.8
## 4   -1.42       3.66      5.08    -0.389  0.699       48.3  -8.77
## 5    0.575     12.6      12.1      0.126  0.900       56.3  -8.56
## 6    0.701      1.65      0.950    2.02   0.0445     330.    0.0173
## # ... with 3 more variables: conf.high <dbl>, method <chr>,
## #   alternative <chr>
```

```
all_stats <-bind_cols(datFCSD, t_stats)

dim(all_stats)
```

```
## [1] 12546    16
```

```
head(all_stats)
```

```
##    gene_name   hi_mean    lo_mean        FC     hi_SD      lo_SD
## 1       A1BG  5.624701  4.0595055 1.3855631 18.886206 12.1525398
## 2        ADA  9.637158 15.4005060 0.6257689 22.776546 48.7223838
## 3       AKT3  3.094601  4.9781848 0.6216324 14.066178 30.0276339
## 4 ZBTB11-AS1  3.659279  5.0819698 0.7200513 10.313471 24.7399129
## 5       MED6 12.648734 12.0736096 1.0476348 25.797292 29.6965277
## 6    NAALAD2  1.651442  0.9501306 1.7381207  6.311207  0.1005881
##     estimate estimate1  estimate2  statistic    p.value parameter
## 1  1.5651953  5.624701  4.0595055  0.7616461 0.44846328  81.73732
## 2 -5.7633477  9.637158 15.4005060 -0.7986190 0.42837304  48.90139
## 3 -1.8835839  3.094601  4.9781848 -0.4234773 0.67380342  48.91354
## 4 -1.4226909  3.659279  5.0819698 -0.3894504 0.69865489  48.30116
## 5  0.5751246 12.648734 12.0736096  0.1261646 0.90005088  56.32840
## 6  0.7013110  1.651442  0.9501306  2.0168254 0.04452288 330.16711
##       conf.low conf.high                  method alternative
## 1  -2.52308378  5.653474 Welch Two Sample t-test   two.sided
## 2 -20.26647228  8.739777 Welch Two Sample t-test   two.sided
## 3 -10.82236942  7.055202 Welch Two Sample t-test   two.sided
## 4  -8.76650557  5.921124 Welch Two Sample t-test   two.sided
## 5  -8.55552398  9.705773 Welch Two Sample t-test   two.sided
## 6   0.01726493  1.385357 Welch Two Sample t-test   two.sided
```

```
#Note that my manucal mean calculation and calculation of tidy (estimate1& estimate2) is identical.


write.csv(all_stats, file="stat_all.csv") #Summarized stat table.

#Select genes with high fold change and significant p values
up_in_ccr8hi <- all_stats %>%
  select(-estimate1, -estimate2) %>%
  filter(p.value <0.01 & FC >5) %>%
  arrange (desc(FC))

down_in_ccr8hi <- all_stats %>%
  select(-estimate1, -estimate2) %>%
  filter(p.value <0.2 & FC <0.2) %>%
  arrange (FC) #default is ascending order

write.csv(up_in_ccr8hi, file="5-fold_up_in_ccr8hi_sig.csv") #Summarized stat table.
write.csv(down_in_ccr8hi, file="5-fold_down_in_ccr8hi.csv") #Summarized stat table.

#Comments: Many cells with low abundent TPM contains exactly same values. Statistics from these values may not represent true statistics. Removing cells with low abundant values are not feasible because essentially all the cells contain low abundant mRNAs. In these case, fold-change could be more reliable values.
```

Visualization

```
#Tried to find our Ruggero lab volcano plot + ggrepel script I wrote.
#I would generate a seperate Markdown file for volcano/ggrepel combination
```