Bcell_receptor_analysis

Sedreh

5/10/2019

```
library(dplyr)
library(tidyr)
library(data.table)
library(Biostrings)
library(ggplot2)
library(seqinr)
```

```
########################################
#processing 10x genomics filtered dataset
########################################
```

```
#load the dataset
filtered_data = read.csv ('/home/sedreh/ITMO/semester2/Bcellsproject/final_Rcode/PBMC
s_of_a_healthy_donor/vdj_v1_hs_pbmc_b_filtered_contig_annotations.csv')
head(filtered_data)
```

```
##              barcode is_cell                    contig_id high_confidence
## 1 AAACCTGCACACTGCG-1    True AAACCTGCACACTGCG-1_contig_1            True
## 2 AAACCTGCACACTGCG-1    True AAACCTGCACACTGCG-1_contig_2            True
## 3 AAACCTGCACACTGCG-1    True AAACCTGCACACTGCG-1_contig_5            True
## 4 AAACCTGCAGGTGGAT-1    True AAACCTGCAGGTGGAT-1_contig_1            True
## 5 AAACCTGCAGGTGGAT-1    True AAACCTGCAGGTGGAT-1_contig_3            True
## 6 AAACCTGCAGGTGGAT-1    True AAACCTGCAGGTGGAT-1_contig_5            True
##   length chain   v_gene   d_gene j_gene c_gene full_length productive
## 1    409 Multi     None     None TRAJ10   IGKC       False       None
## 2    652   IGL IGLV3-10     None  IGLJ3  IGLC2        True       True
## 3    652   IGH  IGHV4-4 IGHD2-15  IGHJ4  IGHG1        True       True
## 4    557  None     None     None   None   None       False       None
## 5    560   IGK IGKV2-24     None  IGKJ4   IGKC        True       True
## 6    676   IGL IGLV3-25     None  IGLJ3  IGLC2        True       True
##                cdr3                                                cdr3_nt
## 1              None                                                   None
## 2     CYSTDSSYNHRVF             TGTTACTCAACAGACAGCAGTTATAATCATAGGGTGTTC
## 3 CARVFCGSSSCTAFDSW TGTGCGAGAGTATTTTGTGGTAGTAGTAGCTGTACCGCCTTTGACTCCTGG
## 4              None                                                   None
## 5         CMQATFGRF                         TGCATGCAAGCTACTTTCGGCCGTTTC
## 6     CQSADSSGTYRVF             TGTCAATCAGCAGACAGCAGTGGTACTTATAGGGTGTTC
##   reads umis raw_clonotype_id        raw_consensus_id
## 1 11588   45      clonotype27                    None
## 2  4640   33      clonotype27 clonotype27_consensus_1
## 3  1184    7      clonotype27 clonotype27_consensus_2
## 4   135    2      clonotype28                    None
## 5  1654    8      clonotype28 clonotype28_consensus_2
## 6  2861   20      clonotype28 clonotype28_consensus_1
```

```
summary(filtered_data)
```

```
##                barcode     is_cell                           contig_id
##  GTCTTCGAGGATCGCA-1:   8   True:4216   AAACCTGCACACTGCG-1_contig_1:   1
##  ACCTTTAGTGCAGACA-1:   6               AAACCTGCACACTGCG-1_contig_2:   1
##  CACAGTACACTCGACG-1:   6               AAACCTGCACACTGCG-1_contig_5:   1
##  CATTCGCAGGATTCGG-1:   6               AAACCTGCAGGTGGAT-1_contig_1:   1
##  CCTTTCTAGATCTGAA-1:   6               AAACCTGCAGGTGGAT-1_contig_3:   1
##  CGCTATCCACGACGAA-1:   6               AAACCTGCAGGTGGAT-1_contig_5:   1
##  (Other)           :4178               (Other)                   :4210
##  high_confidence     length           chain            v_gene
##  True:4216       Min.   : 251.0   IGH    :1509   None     :1204
##                  1st Qu.: 572.0   IGK    : 936   IGHV3-23 : 167
##                  Median : 646.0   IGL    : 801   IGKV1D-39: 139
##                  Mean   : 633.1   Multi  : 701   IGKV3-20 : 129
##                  3rd Qu.: 684.0   None   : 123   IGLV2-14 :  93
##                  Max.   :1191.0   TRA    :  76   IGHV4-59 :  90
##                                   (Other):  70   (Other)  :2394
##       d_gene         j_gene         c_gene     full_length  productive
##  None    :2873   IGHJ4  : 721   IGHM   :1168   False:1239   False:  69
##  IGHD3-10: 185   None   : 567   IGKC   :1034   True :2977   None :1522
##  IGHD6-13: 125   IGKJ1  : 401   IGLC2  : 572                True :2625
##  IGHD2-15:  98   IGLJ2  : 380   None   : 375
##  IGHD6-19:  98   IGHJ6  : 357   IGHD   : 248
##  IGHD4-17:  95   IGLJ3  : 285   IGLC1  : 172
##  (Other) : 742   (Other):1505   (Other): 647
##             cdr3                                        cdr3_nt
##  None         :1512   None                                  :1512
##  CQQSYSTPRTF  :  14   TGTCAGGCGTGGGACAGCAGCACTGTGGTATTC      :   9
##  CQQYDNLPLTF  :  13   TGTCAACAGAGTTACAGTACCCCTCGGACGTTC      :   7
##  CQQYGSSPRTF  :  12   TGTCAACAGTATGATAATCTCCCGCTCACTTTC      :   7
##  CGTWDSSLSAVVF:  11   TGCAGCTCATATACAAGCAGCAGCACTCTAGGAGTCTTC:   6
##  CQAWDSSTVVF  :  11   TGCGGAACATGGGATAGCAGCCTGAGTGCTGGGGTATTC:   6
##  (Other)      :2643   (Other)                               :2669
##      reads             umis             raw_clonotype_id
##  Min.   :    27   Min.   :    1.00   None       :  74
##  1st Qu.:   551   1st Qu.:    4.00   clonotype1 :  25
##  Median :  2305   Median :   14.00   clonotype2 :  15
##  Mean   :  4466   Mean   :   44.93   clonotype22:  12
##  3rd Qu.:  5511   3rd Qu.:   33.00   clonotype10:  10
##  Max.   :166221   Max.   :12215.00   clonotype4 :  10
##                                      (Other)    :4070
##                raw_consensus_id
##  None                   :1592
##  clonotype1_consensus_1:   6
##  clonotype1_consensus_2:   6
##  clonotype2_consensus_1:   5
##  clonotype2_consensus_2:   5
##  clonotype3_consensus_1:   3
##  (Other)                :2599
```

```
#As we have several copies of each barcode in the file, in ordre to count the number
 of cells we need to count just one copy of unique barcoe

barcode_summary <- function(filtered_data)
{
  number_of_cells <- filtered_data %>%
    distinct(barcode) %>%
    dplyr::count()

number_of_cells$n
}
barcode_summary(filtered_data)
```

```
## [1] 1363
```

```
#Exploring data
count_table <- function(filtered_data)
{
  filtered_data %>% mutate(
    gene_group = ifelse(
     v_gene != 'None' & j_gene != 'None',
     "v_gene_and_j_gene",
     ifelse(
         v_gene != 'None' | j_gene != 'None',
        "v_gene_or_j_gene",
        "None"
     )
    )
  ) %>%
    group_by(gene_group) %>%
    summarise(count = n())
}
count_table(filtered_data)
```

```
## # A tibble: 3 x 2
##   gene_group        count
##   <chr>             <int>
## 1 None                542
## 2 v_gene_and_j_gene  2987
## 3 v_gene_or_j_gene    687
```

```
##How many IGK,IGH, IGL and Multi we have in the data? (cell distribution based on ch
ain type)
chain_summary <- function(filtered_data)
{
  filtered_data %>%
   group_by(chain) %>%
    summarise(count = n())
}
chain_summary(filtered_data)
```

```
## # A tibble: 8 x 2
##   chain count
##   <fct> <int>
## 1 IGH    1509
## 2 IGK     936
## 3 IGL     801
## 4 Multi   701
## 5 None    123
## 6 TRA      76
## 7 TRB      68
## 8 TRG       2
```

```
#Number of copies of each cell (we want to know how many copy of each cell we have)
barcode_summary <- function(filtered_data)
  {
  filtered_data %>%
    group_by(barcode) %>%
    summarise(count = n())
}
barcode_summary(filtered_data)
```

```
## # A tibble: 1,363 x 2
##    barcode            count
##    <fct>              <int>
##  1 AAACCTGCACACTGCG-1     3
##  2 AAACCTGCAGGTGGAT-1     4
##  3 AAACCTGGTGTTCTTT-1     2
##  4 AAACCTGTCCCGACTT-1     2
##  5 AAACGGGCATGTCCTC-1     3
##  6 AAACGGGTCCGTTGCT-1     4
##  7 AAACGGGTCCTTTCTC-1     3
##  8 AAAGCAACACAGCGTC-1     2
##  9 AAAGCAATCAAAGTAG-1     2
## 10 AAAGCAATCAACGGGA-1     4
## # … with 1,353 more rows
```

```
#B cells distribution by number of chains in a cell
barcode_summary <- function(filtered_data)
  {
  filtered_data%>%
    group_by(barcode) %>%
    summarise(count = n())%>%
    group_by(count) %>%
    summarise(count_total=n()) %>%
  mutate(
    count_total_pct = round(count_total/ sum(count_total)*100, 2)
  )
}
copy_barcode_filtered <- barcode_summary(filtered_data)
copy_barcode_filtered
```

```
## # A tibble: 7 x 3
##   count count_total count_total_pct
##   <int>       <int>           <dbl>
## 1     1          26            1.91
## 2     2         337           24.7
## 3     3         600           44.0
## 4     4         306           22.4
## 5     5          74            5.43
## 6     6          19            1.39
## 7     8           1            0.07
```

```
## occurance of each condition (we can add which condition that we want)
#the purpose of this calculation is that we want to know how many cell with one heavy chain and one light chain there are(if there are too many, so it is good news) and al so how many cells with 2 light chain we can find in the data! with this calculation w e can study dual light chain effect

#we want to know each cell contain how many chain

occurance_of_each_chain <- function(filtered_data)
  {
  many_conditions = c(
  '1IGL', '1IGH', '1IGK', '1IGH_1IGK_1IGL' , '1IGH_1IGL' , '1IGH_1IGK', '1IGH_2IGK', '1IGH_2IGL')

   filtered_data %>%
    group_by(barcode, chain) %>%
    filter(chain %in% c('IGK','IGH','IGL')) %>%
     summarize(count=n()) %>%  #based on barcode we are countingthe occurance of IGH, IGK  and IGL
     unite('result_chain', count, chain, remove=F, sep='')%>% #we want to combine cou nt column with chain column and put it in new column called "result_chain"
     summarize(
       type=paste(result_chain, collapse='_'), #
       count=sum(count)
      )%>%
     mutate(with_condition = ifelse(
       type %in% many_conditions,
       T, F  #for getting other conditions that is not in our condition list, I add n ew column to check if it is in our condition or not! if it is, it will show "True" ot herwise shows "F"
     ))%>%
  group_by(type, with_condition) %>% #at first we should know that is object in condi tion or not! then count the number of objects in out condition
  summarise(total=n())
}
occurance <- occurance_of_each_chain(filtered_data)
head(occurance)
```

```
## # A tibble: 6 x 3
## # Groups:   type [6]
##   type           with_condition total
##   <chr>          <lgl>          <int>
## 1 ""             FALSE             15
## 2 1IGH           TRUE               6
## 3 1IGH_1IGK      TRUE             484
## 4 1IGH_1IGK_1IGL TRUE             118
## 5 1IGH_1IGK_2IGL FALSE              9
## 6 1IGH_1IGL      TRUE             315
```

```
#practice
filtered_data %>%
    group_by(barcode, chain) %>%
    filter(chain %in% c('IGK','IGH','IGL'))%>%
   summarize(count=n()) %>%
  unite('result_chain', count, chain, remove=F, sep='') #we have several different ch
ain for each barcode like 1IGH AND 1IGK! then we want to show the condition for that
 special barcode like "1IGH-1IGK" so we alredy grouped the data with barcode! we got
 the 2 different chain from result_chain column for special barcode and combine them
 by (collapse='')
```

```
## # A tibble: 2,819 x 4
## # Groups:   barcode [1,363]
##    barcode            result_chain chain count
##    <fct>              <chr>        <fct> <int>
##  1 AAACCTGCACACTGCG-1 1IGH         IGH       1
##  2 AAACCTGCACACTGCG-1 1IGL         IGL       1
##  3 AAACCTGCAGGTGGAT-1 1IGH         IGH       1
##  4 AAACCTGCAGGTGGAT-1 1IGK         IGK       1
##  5 AAACCTGCAGGTGGAT-1 1IGL         IGL       1
##  6 AAACCTGGTGTTCTTT-1 1IGH         IGH       1
##  7 AAACCTGGTGTTCTTT-1 1IGK         IGK       1
##  8 AAACCTGTCCCGACTT-1 1IGK         IGK       1
##  9 AAACGGGCATGTCCTC-1 1IGH         IGH       1
## 10 AAACGGGCATGTCCTC-1 1IGL         IGL       1
## # … with 2,809 more rows
```

```
  #      count=sum(count)
  # )
```

```
#here we want our condition! so just filter T, then make data frame from total and ty
pe columns
with_condition <- filtered_data%>%
  occurance_of_each_chain()%>%
  filter(with_condition == T)%>%
  select(type,total)%>%
  data.frame()

#here we will collect just data that are not in our condition
without_condition_total <- filtered_data%>%
  occurance_of_each_chain()%>%
  filter(with_condition == F)%>%
  pull(total)%>%
  sum()

#now we have 2 separate data frame with condition and without! at first we add the ro
w of other to the list

without_condition <- data.frame('Other', without_condition_total)
names(without_condition) <- names(with_condition) #we should make the name of the col
umn of 2 dataset similar!
#here just combine 2 data frame together
Final_results <- rbind(with_condition, without_condition)%>%
  mutate(
    total_pct = round(total/ sum(total)*100, 2)
  )
Final_results
```

```
##             type total total_pct
## 1           1IGH     6      0.44
## 2      1IGH_1IGK   484     35.51
## 3 1IGH_1IGK_1IGL   118      8.66
## 4      1IGH_1IGL   315     23.11
## 5      1IGH_2IGK    55      4.04
## 6      1IGH_2IGL   106      7.78
## 7           1IGK    30      2.20
## 8           1IGL     8      0.59
## 9          Other   241     17.68
```

```
###################################
#processing migmap output
###################################
```

```
#load the dataset
data_migmap <-read.csv ('/home/sedreh/ITMO/semester2/Bcellsproject/final_Rcode/PBMCs_
of_a_healthy_donor/migmap_result_healthy_donor.csv', sep="\t", header=TRUE)
head(data_migmap)
```

```
##                   read.header
## 1 >ACACCCTCAGGCTGAA-1_contig_1
## 2 >ACACCCTCAGGCTGAA-1_contig_3
## 3 >ACCTTTAGTACACCGC-1_contig_2
## 4 >ACTGCTCCAGGATTGG-1_contig_2
## 5 >GTACTCCAGCGCTTAT-1_contig_1
## 6 >TATCAGGCATGGAATA-1_contig_1
##                                                cdr3nt            cdr3aa
## 1 TGTGCGAGAGAGCCCAAAGGCGCAGCAGTGGCTGGACAACTTGACTACTGG CAREPKGAAVAGQLDYW
## 2               TGTCAGCAGTATGGTAGCTCACCTCTATTCACTTTC      CQQYGSSPLFTF
## 3                  TGTCAACAGATTAGTAGTTACCCTCTCACTTTC       CQQISSYPLTF
## 4                  TGTGCGATGTGGGCATGGGAACTAGACTACTGG       CAMWAWELDYW
## 5                  TGTCAACAGAGTTACAGTACCCTTTGGACGTTC       CQQSYSTLWTF
## 6                  TGTCTACAGACTTACAGTGTCCCTCGGACTTTT       CLQTYSVPRTF
##     cdr.insert.qual        mutations.qual   v.segment   d.segment
## 1 IIIIIIIIIIIIIIII                     I IGHV3-21*01 IGHD6-19*01
## 2                 I                      IGKV3-20*01           .
## 3                   IIIIIIIIIIIIIIIIIIII  IGKV1-9*01           .
## 4        IIIIIIIIII      IIIIIIIIIIIIIII IGHV3-23*01  IGHD1-7*01
## 5                II                      IGKV1-39*01           .
## 6                II            IIIIIIIII IGKV1-39*01           .
##   j.segment cdr1.start.in.read cdr1.end.in.read cdr2.start.in.read
## 1  IGHJ4*01                216              240                291
## 2  IGKJ3*01                197              218                269
## 3  IGKJ4*01                172              190                241
## 4  IGHJ4*01                212              236                287
## 5  IGKJ1*01                175              193                244
## 6  IGKJ2*01                193              211                262
##   cdr2.end.in.read cdr3.start.in.read cdr3.end.in.read v.end.in.cdr3
## 1              315                426              477            11
## 2              278                383              419            25
## 3              250                355              388            25
## 4              311                422              455             7
## 5              253                358              391            22
## 6              271                376              409            25
##   d.start.in.cdr3 d.end.in.cdr3 j.start.in.cdr3 v.del d.del.5 d.del.3
## 1              23            35              40     0       6       3
## 2              -1            -1              26     1      -1      -1
## 3              -1            -1              24     1      -1      -1
## 4              17            24              24     4       9       1
## 5              -1            -1              24     4      -1      -1
## 6              -1            -1              27     1      -1      -1
##   j.del                            mutations.nt.FR1
## 1     6
## 2     0
## 3     1
## 4     8 S5:G>A,S7:A>T,S8:G>C,S11:G>C,S34:T>C,S38:G>C
## 5     1
## 6     5
##                         mutations.nt.CDR1
## 1
## 2
## 3 S82:G>A,S83:C>T,S88:G>A,S91:G>A,S92:T>A
## 4
## 5
## 6                                S91:G>T
##                      mutations.nt.FR2                   mutations.nt.CDR2
```

```
## 1
## 2
## 3 S111:C>G,S112:A>T,S124:A>C,S134:G>A                    S148:C>T,S152:A>G
## 4          S102:A>G,S103:G>T,S147:G>A S153:A>G,S165:G>T,S166:G>A,S169:G>A
## 5
## 6          S125:A>T,S134:G>A,S140:G>T
##                                      mutations.nt.FR3
## 1
## 2
## 3 S193:G>C,S195:G>A,S199:C>T,S204:A>G,S226:G>A,S257:T>C
## 4                                    S237:T>G,S242:G>A
## 5
## 6                                              S226:G>A
##                    mutations.nt.CDR3  mutations.nt.FR4    rc complete
## 1                                              S341:A>G false     true
## 2                                                       false     true
## 3                  S270:C>A,S274:A>G S299:A>G,S317:C>G false     true
## 4                                              S323:A>G false     true
## 5                                                       false     true
## 6 S265:A>T,S271:G>C,S279:A>G,S280:C>T          S308:G>C false     true
##   has.cdr3 in.frame no.stop                              mutations.aa.FR1
## 1     true     true    true
## 2     true     true    true
## 3     true     true    true
## 4     true     true    true S1:V>V,S2:Q>L,S2:Q>L,S3:L>L,S11:V>A,S12:Q>H
## 5     true     true    true
## 6     true     true    true
##                         mutations.aa.CDR1                mutations.aa.FR2
## 1
## 2
## 3 S27:G>D,S27:G>D,S29:S>N,S30:S>K,S30:S>K S37:Q>V,S37:Q>V,S41:K>T,S44:K>K
## 4                                                 S34:S>V,S34:S>V,S49:A>T
## 5
## 6                                  S30:S>I         S41:K>N,S44:K>K,S46:L>L
##                mutations.aa.CDR2
## 1
## 2
## 3                 S49:A>V,S50:A>A
## 4 S51:S>G,S55:G>Y,S55:G>Y,S56:S>N
## 5
## 6
##                                    mutations.aa.FR3
## 1
## 2
## 3 S64:S>T,S65:G>R,S66:S>F,S68:T>A,S75:S>N,S85:Y>Y
## 4                                  S79:Y>D,S80:L>L
## 5
## 6                                           S75:S>N
##                mutations.aa.CDR3 mutations.aa.FR4 pol.v pol.d.5 pol.d.3
## 1                                          S113:Q>Q    -1      -1      -1
## 2                                                      -1      -1      -1
## 3                 S90:L>I,S91:N>S S99:G>G,S105:I>M    -1      -1      -1
## 4                                          S107:Q>Q    -1      -1      -1
## 5                                                      -1      -1      -1
## 6 S88:Q>L,S90:S>T,S93:T>V,S93:T>V         S102:K>N    -1      -1      -1
##   pol.j canonical
## 1    -1      true
## 2    -1      true
```

```
## 3    -1      true
## 4    -1      true
## 5    -1      true
## 6    -1      true
##
contignt
## 1
GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCCTGGTCAAGCCTGGGGGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATTCACCT
TCAGTAGCTATAGCATGAACTGGGTCCGCCAGGCTCCAGGGAAGGGGCTGGAGTGGGTCTCATCCATTAGTAGTAGTAGTTA
CATATACTACGCAGACTCAGTGAAGGGCCGATTCACCATCTCCAGAGACAACGCCAAGAACTCACTGTATCTGCAAATGAACAGC
CTGAGAGCCGAGGACACGGCTGTGTATTACTGTGCGAGAGAGCCCAAAGGCGCAGCAGTGGCTGGACAACTTGACTACTGGGGCC
AGGGAACCCTGGTCACCGTCTCCTCAGGGAGTGCATCCGCCCCAACCCTTTTCCCCCTCGTCTCCTGTGAGAATTCCCCGTCGGA
TACGAGCAGCGTG
## 2 GAAATTGTGTTGACGCAGTCTCCAGGCACCCTGTCTTTGTCTCCAGGGGAAAGAGCCACCCTCTCCTGCAGGGCCAGTCA
GAGTGTTAGCAGCAGCTACTTAGCCTGGTACCAGCAGAAACCTGGCCAGGCTCCCAGGCTCCTCATCTATGGTGCATCCAGCAGG
GCCACTGGCATCCCAGACAGGTTCAGTGGCAGTGGGTCTGGGACAGACTTCACTCTCACCATCAGCAGACTGGAGCCTGAAGATT
TTGCAGTGTATTACTGTCAGCAGTATGGTAGCTCACCTCTATTCACTTTCGGCCCTGGGACCAAAGTGGATATCAAACGAACTGT
GGCTGCACCATCTGTCTTCATCTTCCCGCCATCTGATGAGCAGTTGAAATCTGGAACTGCCTCTGTTGTGTGCCTGCTGAATAAC
TTCTATCCCAGAGAGGCCAAAGTACAGTGGAAGGTGGATAACGCCCTCCAATCGGGTAACTCCCAGGAGAGTGTCACAGAGCAGG
ACAGCAAGGACAGCACCTACAGCCTCAGCAGCACCCTGACGCTGAGCAAAGCAGACTACGAGAA
## 3       GACATCCAGTTGACCCAGTCTCCATCCTTCCTGTCTGCATCTGTAGGAGACAGAGTCACCATCACTTGCCGGGC
CAGTCAGGATATTAACAAATATTTAGCCTGGTATCAGGTAAAACCAGGGACAGCCCCTAAACTCCTGATCTATGTTGCGTCCACT
TTGCAAAGTGGGGTCCCATCAAGGTTCAGCGGCACTAGATTTGGGGCAGAATTCACTCTCACAATCAACAGCCTGCAGCCTGAAG
ATTTTGCAACTTACTACTGTCAACAGATTAGTAGTTACCCTCTCACTTTCGGCGGGGGGACCAAGGTGGAGATGAAACGAACTGT
GGCTGCACCATCTGTCTTCATCTTCCCGCCATCTGATGAGCAGTTGAAATCTGGAACTGCCTCTGTTGTGTGCCTGCTGAATAAC
TTCTATCCCAGAGAGGCCAAAGTACAGTGGAAGGTGGATAACGCCCTCCAATCGGGTAACTCCCAGGAGAGTGTCACAGAGCAGG
ACAGCAAGGACAGCACCTACAGCCTCAGCAGCACCCTGACGCTGAGCAAAGCAGACTACGAGAA
## 4                                                                   GAGGTACTCCTCTTGGAGTCTG
GGGGAGGCTTGGCACACCCTGGGGGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATTCACCTTTAGCAGCTATGCCATGGTCTG
GGTCCGCCAGGCTCCAGGGAAGGGGCTGGAGTGGGTCTCAACTATTGGTGGTAGTGGTTATAACACATACTACGCAGACTCCGTG
AAGGGCCGGTTCACCATCTCCAGAGACAATTCCAAGAACACGCTGGATCTACAAATGAACAGCCTGAGAGCCGAGGACACGGCCG
TATATTACTGTGCGATGTGGGCATGGGAACTAGACTACTGGGGCCAGGGAACCCTGGTCACCGTCTCCTCAGCATCCCCGACCAG
CCCCAAGGTCTTCCCGCTGAGCCTCTGCAGCACCCAGCCAGATGGGAACGTGGTCATCGCCTGCCTGGTCCAGGGCTTCTTCCCC
CAGGAGCCACTCAGTGTGACCTGGAGCGAAAGCGGACAGGGCGTGACCGCCAGAAACTTCCCCC
## 5       GACATCCAGATGACCCAGTCTCCATCCTCCCTGTCTGCATCTGTAGGAGACAGAGTCACCATCACTTGCCGGGC
AAGTCAGAGCATTAGCAGCTATTTAAATTGGTATCAGCAGAAACCAGGGAAAGCCCCTAAGCTCCTGATCTATGCTGCATCCAGT
TTGCAAAGTGGGGTCCCATCAAGGTTCAGTGGCAGTGGATCTGGGACAGATTTCACTCTCACCATCAGCAGTCTGCAACCTGAAG
ATTTTGCAACTTACTACTGTCAACAGAGTTACAGTACCCTTTGGACGTTCGGCCAAGGGACCAAGGTGGAAATCAAACGAACTGT
GGCTGCACCATCTGTCTTCATCTTCCCGCCATCTGATGAGCAGTTGAAATCTGGAACTGCCTCTGTTGTGTGCCTGCTGAATAAC
TTCTATCCCAGAGAGGCCAAAGTACAGTGGAAGGTGGATAACGCCCTCCAATCGGGTAACTCCCAGGAGAGTGTCACAGAGCAGG
ACAGCAAGGACAGCACCTACAGCCTCAGCAGCACCCTGACGCTGAGCAAAGCAGACTACGAGAA
## 6       GACATCCAGATGACCCAGTCTCCATCCTCCCTGTCTGCATCTGTAGGAGACAGAGTCACCATCACTTGCCGGGC
AAGTCAGAGCATTAGCATCTATTTAAATTGGTATCAGCAGAAACCAGGGAATGCCCCTAAACTCCTTATCTATGCTGCATCCAGT
TTGCAAAGTGGGGTCCCATCAAGGTTCAGTGGCAGTGGATCTGGGACAGATTTCACTCTCACCATCAACAGTCTGCAACCTGAAG
ATTTTGCAACTTACTACTGTCTACAGACTTACAGTGTCCCTCGGACTTTTGGCCAGGGGACCAACCTGGAGATCAAACGAACTGT
GGCTGCACCATCTGTCTTCATCTTCCCGCCATCTGATGAGCAGTTGAAATCTGGAACTGCCTCTGTTGTGTGCCTGCTGAATAAC
TTCTATCCCAGAGAGGCCAAAGTACAGTGGAAGGTGGATAACGCCCTCCAATCGGGTAACTCCCAGGAGAGTGTCACAGAGCAGG
ACAGCAAGGACAGCACCTACAGCCTCAGCAGCACCCTGACGCTGAGCAAAGCAGACTACGAGAA
```

```
summary(data_migmap)
```

```
##                         read.header
##  >AAACCTGCACACTGCG-1_contig_2:   1
##  >AAACCTGCACACTGCG-1_contig_5:   1
##  >AAACCTGCAGGTGGAT-1_contig_3:   1
##  >AAACCTGCAGGTGGAT-1_contig_5:   1
##  >AAACCTGCAGGTGGAT-1_contig_6:   1
##  >AAACCTGGTGTTCTTT-1_contig_1:   1
##  (Other)                     :2638
##                                    cdr3nt               cdr3aa
##  TGTCAGGCGTGGGACAGCAGCACTGTGGTATTC      :   9   CQQSYSTPRTF  :  14
##  TGTCAACAGAGTTACAGTACCCCTCGGACGTTC      :   7   CQQYDNLPLTF  :  13
##  TGTCAACAGTATGATAATCTCCCGCTCACTTTC      :   7   CQQYGSSPRTF  :  12
##  TGCAGCTCATATACAAGCAGCAGCACTCTAGGAGTCTTC:   6   CGTWDSSLSAVVF:  11
##  TGCGGAACATGGGATAGCAGCCTGAGTGCTGGGGTATTC:   6   CQAWDSSTVVF  :  11
##  TGTCAATCAGCAGACAGCAGTGGTACTTATGTGGTATTC:   6   CQQSYSTPLTF  :  10
##  (Other)                                :2603   (Other)      :2573
##  cdr.insert.qual mutations.qual       v.segment          d.segment
##         : 421           :735    IGHV3-23*01: 164   .          :1383
##  II     : 198    I      :543    IGKV1-39*01: 125   IGHD3-10*01: 176
##  III    : 182    II     :320    IGKV3-20*01: 114   IGHD6-13*01: 119
##  I      : 165    III    :119    IGHV4-59*01:  89   IGHD4-17*01:  93
##  IIIIII : 132    IIIIII :103    IGHV4-39*01:  86   IGHD2-15*01:  87
##  IIII   : 123    IIIII  : 96    IGHV4-34*01:  83   IGHD3-22*01:  83
##  (Other):1423    (Other):728    (Other)    :1983   (Other)    : 703
##     j.segment   cdr1.start.in.read cdr1.end.in.read cdr2.start.in.read
##  IGHJ4*01:568   Min.   : -1.0      Min.   : -1.0    Min.   : -1.0
##  IGLJ2*01:520   1st Qu.:177.0      1st Qu.:204.0    1st Qu.:255.0
##  IGHJ6*01:299   Median :194.0      Median :218.0    Median :269.0
##  IGKJ1*01:263   Mean   :204.1      Mean   :227.7    Mean   :278.6
##  IGKJ4*01:192   3rd Qu.:215.0      3rd Qu.:239.0    3rd Qu.:290.0
##  IGHJ5*01:155   Max.   :681.0      Max.   :711.0    Max.   :762.0
##  (Other) :647
##  cdr2.end.in.read cdr3.start.in.read cdr3.end.in.read v.end.in.cdr3
##  Min.   : -1.0    Min.   :  3.0      Min.   : 69.0    Min.   : 2.00
##  1st Qu.:266.0    1st Qu.:371.0      1st Qu.:408.0    1st Qu.:10.00
##  Median :288.0    Median :396.5      Median :446.0    Median :19.50
##  Mean   :294.7    Mean   :402.7      Mean   :447.3    Mean   :17.87
##  3rd Qu.:314.0    3rd Qu.:425.0      3rd Qu.:477.0    3rd Qu.:25.00
##  Max.   :783.0    Max.   :894.0      Max.   :966.0    Max.   :38.00
##
##  d.start.in.cdr3  d.end.in.cdr3   j.start.in.cdr3     v.del
##  Min.   :-1.000   Min.   :-1.00   Min.   :11.00   Min.   : 0.000
##  1st Qu.:-1.000   1st Qu.:-1.00   1st Qu.:26.00   1st Qu.: 0.000
##  Median :-1.000   Median :-1.00   Median :32.00   Median : 1.000
##  Mean   : 7.839   Mean   :14.51   Mean   :33.21   Mean   : 1.917
##  3rd Qu.:16.250   3rd Qu.:30.00   3rd Qu.:38.00   3rd Qu.: 3.000
##  Max.   :49.000   Max.   :69.00   Max.   :75.00   Max.   :21.000
##
##     d.del.5          d.del.3           j.del
##  Min.   :-1.000   Min.   :-1.000   Min.   : 0.000
##  1st Qu.:-1.000   1st Qu.:-1.000   1st Qu.: 1.000
##  Median :-1.000   Median :-1.000   Median : 3.000
##  Mean   : 2.261   Mean   : 2.136   Mean   : 4.157
##  3rd Qu.: 5.000   3rd Qu.: 4.000   3rd Qu.: 7.000
##  Max.   :29.000   Max.   :26.000   Max.   :28.000
##
##         mutations.nt.FR1       mutations.nt.CDR1 mutations.nt.FR2
```

```
##                :1976                   :1906           :1805
##  S13:T>C,S14:G>A:  33    S97:C>G,S98:T>C:  55    S119:G>A:  61
##  S41:C>G        :  33    S91:G>A        :  23    S134:A>G:  60
##  S20:T>C        :  17    S91:G>C        :  23    S146:T>C:  26
##  S5:G>C         :  16    S95:T>C        :  21    S147:G>A:  23
##  S42:C>T        :  15    S92:C>T        :   8    S154:T>G:  17
##  (Other)        : 554    (Other)        : 608    (Other) : 652
##                            mutations.nt.CDR2          mutations.nt.FR3
##                                     :1907                     :1628
##  S155:G>T                           :  63    S179:C>T         :  55
##  S147:G>A,S149:T>G,S152:C>G,S155:C>T:  54    S161:G>A         :  47
##  S152:C>A                           :  37    S242:T>C         :  27
##  S162:T>C,S169:C>T                  :  21    S260:T>C,S263:G>A:  22
##  S165:A>G                           :  19    S219:G>A         :  21
##  (Other)                            : 543    (Other)          : 844
##  mutations.nt.CDR3 mutations.nt.FR4     rc       complete    has.cdr3
##          :1715             :1391    false:2644   true:2644   true:2644
##  S292:G>A:  40     S341:A>G:  93
##  S272:C>T:  27     S338:A>G:  84
##  S330:T>C:  16     S335:A>G:  70
##  S327:T>C:  15     S344:A>G:  57
##  S333:G>A:  14     S347:A>G:  57
##  (Other) : 817     (Other) : 892
##  in.frame    no.stop          mutations.aa.FR1       mutations.aa.CDR1
##  true:2644   true:2644                :1976                     :1906
##                           S13:P>P      :  35    S32:A>G,S32:A>G:  55
##                           S4:M>T,S4:M>T:  33    S31:Y>Y        :  21
##                           S6:S>S       :  18    S30:S>N        :  20
##                           S1:V>V       :  17    S30:S>T        :  20
##                           S14:P>S      :  15    S30:S>S        :   9
##                           (Other)      : 550    (Other)        : 613
##  mutations.aa.FR2                          mutations.aa.CDR2
##         :1805                                      :1907
##  S44:L>L:  62     S51:E>D                          :  63
##  S39:Q>Q:  61     S49:D>K,S49:D>K,S50:A>A,S51:S>S:  56
##  S48:I>I:  26     S50:D>E                          :  36
##  S49:G>R:  22     S54:F>L,S56:T>I                  :  21
##  S51:L>R:  17     S55:N>D                          :  18
##  (Other): 651     (Other)                          : 543
##         mutations.aa.FR3 mutations.aa.CDR3 mutations.aa.FR4
##                 :1628            :1715             :1391
##  S59:Y>Y        :  55    S97:R>K :  40     S113:Q>Q:  93
##  S53:L>L        :  47    S90:D>D :  26     S112:Q>Q:  84
##  S80:D>D        :  27    S110:S>P:  16     S111:Q>Q:  70
##  S86:A>A,S87:A>A:  22    S109:S>P:  14     S114:Q>Q:  57
##  S73:E>K        :  20    S111:V>I:  14     S115:Q>Q:  56
##  (Other)        : 845    (Other) : 819     (Other) : 893
##      pol.v            pol.d.5           pol.d.3            pol.j
##  Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.0000
##  1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000
##  Median :-1.0000   Median :-1.0000   Median :-1.0000   Median :-1.0000
##  Mean   :-0.5034   Mean   :-0.9459   Mean   :-0.7708   Mean   :-0.6032
##  3rd Qu.:-1.0000   3rd Qu.:-1.0000   3rd Qu.:-1.0000   3rd Qu.:-1.0000
##  Max.   :35.0000   Max.   :21.0000   Max.   :51.0000   Max.   :52.0000
##
##  canonical
##  true:2644
##
```

```
##
##
##
##
##
##
contignt
##  CAGTCTGCCCTGACTCAGCCTGCCTCCGTGTCTGGGTCTCCTGGACAGTCGATCACCATCTCCTGCACTGGAACCAGCAGT
GACGTTGGTGGTTATAACTATGTCTCCTGGTACCAACAACACCCAGGCAAAGCCCCCAAACTCATGATTTATGATGTCAGTAATC
GGCCCTCAGGGGTTTCTAATCGCTTCTCTGGCTCCAAGTCTGGCAACACGGCCTCCCTGACCATCTCTGGGCTCCAGGCTGAGGA
CGAGGCTGATTATTACTGCAGCTCATATACAAGCAGCAGCACTCTAGGAGTCTTCGGAACTGGGACCAAGGTCACCGTCCTAGGT
CAGCCCAAGGCCAACCCCACTGTCACTCTGTTCCCGCCCTCCTCTGAGGAGCTCCAAGCCAACAAGGCCACACTAGTGTGTCTGA
TCAGTGACTTCTACCCGGGAGCTGTGACAGTGGCCTGGAAGGCAGATGGCAGCCCCGTCAAGGCGGGAGTGGAGACCACCAAACC
CTCCAAACAGAGCAACAACAAGTACGCGGCCAGCAGCTA:   5
##  CAGTCTGTGCTGACTCAGCCACCCTCAGCGTCTGGGACCCCCGGGCAGAGGGTCACCATCTCTTGTTCTGGAAGCAGCTCC
AACATCGGAAGTAATACTGTAAACTGGTACCAGCAGCTCCCAGGAACGGCCCCCAAACTCCTCATCTATAGTAATAATCAGCGGC
CCTCAGGGGTCCCTGACCGATTCTCTGGCTCCAAGTCTGGCACCTCAGCCTCCCTGGCCATCAGTGGGCTCCAGTCTGAGGATGA
GGCTGATTATTACTGTGCAGCATGGGATGACAGCCTGAATGGTTGGGTGTTCGGCGGAGGGACCAAGCTGACCGTCCTAGGTCAG
CCCAAGGCTGCCCCCTCGGTCACTCTGTTCCCGCCCTCCTCTGAGGAGCTTCAAGCCAACAAGGCCACACTGGTGTGTCTCATAA
GTGACTTCTACCCGGGAGCCGTGACAGTGGCCTGGAAGGCAGATAGCAGCCCCGTCAAGGCGGGAGTGGAGACCACCACACCCTC
CAAACAAAGCAACAACAAGTACGCGGCCAGCAGCTA   :   5
##  TCCTATGAGCTGACACAGCCACCCTCGGTGTCAGTGTCCCCAGGACAGACGGCCAGGATCACCTGCTCTGGAGATGCATTG
CCAAAGCAATATGCTTATTGGTACCAGCAGAAGCCAGGCCAGGCCCCTGTGCTGGTGATATATAAAGACAGTGAGAGGCCCTCAG
GGATCCCTGAGCGATTCTCTGGCTCCAGCTCAGGGACAACAGTCACGTTGACCATCAGTGGAGTCCAGGCAGAAGACGAGGCTGA
CTATTACTGTCAATCAGCAGACAGCAGTGGTACTTATGTGGTATTCGGCGGAGGGACCAAGCTGACCGTCCTAGGTCAGCCCAAG
GCTGCCCCCTCGGTCACTCTGTTCCCGCCCTCCTCTGAGGAGCTTCAAGCCAACAAGGCCACACTGGTGTGTCTCATAAGTGACT
TCTACCCGGGAGCCGTGACAGTGGCCTGGAAGGCAGATAGCAGCCCCGTCAAGGCGGGAGTGGAGACCACCACACCCTCCAAACA
AAGCAACAACAAGTACGCGGCCAGCAGCTA         :   5
##  TCCTATGAGCTGACTCAGCCACCCTCAGTGTCCGTGTCCCCAGGACAGACAGCCAGCATCACCTGCTCTGGAGATAAATTG
GGGGATAAATATGCTTGCTGGTATCAGCAGAAGCCAGGCCAGTCCCCTGTGCTGGTCATCTATCAAGATAGCAAGCGGCCCTCAG
GGATCCCTGAGCGATTCTCTGGCTCCAACTCTGGGAACACAGCCACTCTGACCATCAGCGGGACCCAGGCTATGGATGAGGCTGA
CTATTACTGTCAGGCGTGGGACAGCAGCACTGTGGTATTCGGCGGAGGGACCAAGCTGACCGTCCTAGGTCAGCCCAAGGCTGCC
CCCTCGGTCACTCTGTTCCCGCCCTCCTCTGAGGAGCTTCAAGCCAACAAGGCCACACTGGTGTGTCTCATAAGTGACTTCTACC
CGGGAGCCGTGACAGTGGCCTGGAAGGCAGATAGCAGCCCCGTCAAGGCGGGAGTGGAGACCACCACACCCTCCAAACAAAGCAA
CAACAAGTACGCGGCCAGCAGCTA               :   5
##  CAGTCTGTGCTGACGCAGCCGCCCTCAGTGTCTGGGGCCCCAGGGCAGAGGGTCACCATCTCCTGCACTGGGAGCAGCTCC
AACATCGGGGCAGGTTATGATGTACACTGGTACCAGCAGCTTCCAGGAACAGCCCCCAAACTCCTCATCTATGGTAACAGCAATC
GGCCCTCAGGGGTCCCTGACCGATTCTCTGGCTCCAAGTCTGGCACCTCAGCCTCCCTGGCCATCACTGGGCTCCAGGCTGAGGA
TGAGGCTGATTATTACTGCCAGTCCTATGACAGCAGCCTGAGTGGTTGGGTGTTCGGCGGAGGGACCAAGCTGACCGTCCTAGGT
CAGCCCAAGGCTGCCCCCTCGGTCACTCTGTTCCCGCCCTCCTCTGAGGAGCTTCAAGCCAACAAGGCCACACTGGTGTGTCTCA
TAAGTGACTTCTACCCGGGAGCCGTGACAGTGGCCTGGAAGGCAGATAGCAGCCCCGTCAAGGCGGGAGTGGAGACCACCACACC
CTCCAAACAAAGCAACAACAAGTACGCGGCCAGCAGCTA:   4
##  GAGGTGCAGCTGGTGCAGTCTGGAGCAGAGGTGAAAAAGCCGGGGGAGTCTCTGAAGATCTCCTGTAAGGGTTCTGGATAC
AGCTTTACCAGCTACTGGATCGGCTGGGTGCGCCAGATGCCCGGGAAAGGCCTGGAGTGGATGGGGATCATCTATCCTGGTGACT
CTGATACCAGATACAGCCCGTCCTTCCAAGGCCAGGTCACCATCTCAGCCGACAAGTCCATCAGCACCGCCTACCTGCAGTGGAG
CAGCCTGAAGGCCTCGGACACCGCCATGTATTACTGTGCGAGCCCACTATTGGGATATTGTAGTGGTGGTAGCTGCTACGACTAC
TGGGGCCAGGGAACCCTGGTCACCGTCTCCTCAGGGAGTGCATCCGCCCCAACCCTTTTCCCCCTCGTCTCCTGTGAGAATTCCC
CGTCGGATACGAGCAGCGTG
:   4
##  (Other)
:2616
```

```
# we need to have same columns names the same as filtered data for analysis(for examp
le:In #filtered data we have barcode as a first column name but in migmap the name is
read.heaer).So we need to split read and header to create barcode column.
#for this purpose, we need to do some steps:
preprocess_data <- function(data_migmap){
separate(data_migmap,
         col = "read.header",
         into = c("read", "header"),
         sep = "_")
}
data <- preprocess_data(data_migmap)
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 2644 rows [1, 2,
## 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
head(data)
```

```
##                  read header
## 1 >ACACCCTCAGGCTGAA-1 contig
## 2 >ACACCCTCAGGCTGAA-1 contig
## 3 >ACCTTTAGTACACCGC-1 contig
## 4 >ACTGCTCCAGGATTGG-1 contig
## 5 >GTACTCCAGCGCTTAT-1 contig
## 6 >TATCAGGCATGGAATA-1 contig
##                                                cdr3nt            cdr3aa
## 1 TGTGCGAGAGAGCCCAAAGGCGCAGCAGTGGCTGGACAACTTGACTACTGG CAREPKGAAVAGQLDYW
## 2               TGTCAGCAGTATGGTAGCTCACCTCTATTCACTTTC      CQQYGSSPLFTF
## 3                  TGTCAACAGATTAGTAGTTACCCTCTCACTTTC       CQQISSYPLTF
## 4                  TGTGCGATGTGGGCATGGGAACTAGACTACTGG       CAMWAWELDYW
## 5                  TGTCAACAGAGTTACAGTACCCTTTGGACGTTC       CQQSYSTLWTF
## 6                  TGTCTACAGACTTACAGTGTCCCTCGGACTTTT       CLQTYSVPRTF
##     cdr.insert.qual        mutations.qual   v.segment   d.segment
## 1 IIIIIIIIIIIIIIII                     I IGHV3-21*01 IGHD6-19*01
## 2                 I                      IGKV3-20*01           .
## 3                   IIIIIIIIIIIIIIIIIIII  IGKV1-9*01           .
## 4        IIIIIIIIII      IIIIIIIIIIIIIII IGHV3-23*01  IGHD1-7*01
## 5                II                      IGKV1-39*01           .
## 6                II            IIIIIIIII IGKV1-39*01           .
##   j.segment cdr1.start.in.read cdr1.end.in.read cdr2.start.in.read
## 1  IGHJ4*01                216              240                291
## 2  IGKJ3*01                197              218                269
## 3  IGKJ4*01                172              190                241
## 4  IGHJ4*01                212              236                287
## 5  IGKJ1*01                175              193                244
## 6  IGKJ2*01                193              211                262
##   cdr2.end.in.read cdr3.start.in.read cdr3.end.in.read v.end.in.cdr3
## 1              315                426              477            11
## 2              278                383              419            25
## 3              250                355              388            25
## 4              311                422              455             7
## 5              253                358              391            22
## 6              271                376              409            25
##   d.start.in.cdr3 d.end.in.cdr3 j.start.in.cdr3 v.del d.del.5 d.del.3
## 1              23            35              40     0       6       3
## 2              -1            -1              26     1      -1      -1
## 3              -1            -1              24     1      -1      -1
## 4              17            24              24     4       9       1
## 5              -1            -1              24     4      -1      -1
## 6              -1            -1              27     1      -1      -1
##   j.del                            mutations.nt.FR1
## 1     6
## 2     0
## 3     1
## 4     8 S5:G>A,S7:A>T,S8:G>C,S11:G>C,S34:T>C,S38:G>C
## 5     1
## 6     5
##                         mutations.nt.CDR1
## 1
## 2
## 3 S82:G>A,S83:C>T,S88:G>A,S91:G>A,S92:T>A
## 4
## 5
## 6                                S91:G>T
##                      mutations.nt.FR2                   mutations.nt.CDR2
```

```
## 1
## 2
## 3 S111:C>G,S112:A>T,S124:A>C,S134:G>A                    S148:C>T,S152:A>G
## 4          S102:A>G,S103:G>T,S147:G>A S153:A>G,S165:G>T,S166:G>A,S169:G>A
## 5
## 6          S125:A>T,S134:G>A,S140:G>T
##                                      mutations.nt.FR3
## 1
## 2
## 3 S193:G>C,S195:G>A,S199:C>T,S204:A>G,S226:G>A,S257:T>C
## 4                                    S237:T>G,S242:G>A
## 5
## 6                                              S226:G>A
##                    mutations.nt.CDR3  mutations.nt.FR4    rc complete
## 1                                              S341:A>G false     true
## 2                                                       false     true
## 3                  S270:C>A,S274:A>G S299:A>G,S317:C>G false     true
## 4                                              S323:A>G false     true
## 5                                                       false     true
## 6 S265:A>T,S271:G>C,S279:A>G,S280:C>T          S308:G>C false     true
##   has.cdr3 in.frame no.stop                              mutations.aa.FR1
## 1     true     true    true
## 2     true     true    true
## 3     true     true    true
## 4     true     true    true S1:V>V,S2:Q>L,S2:Q>L,S3:L>L,S11:V>A,S12:Q>H
## 5     true     true    true
## 6     true     true    true
##                         mutations.aa.CDR1                mutations.aa.FR2
## 1
## 2
## 3 S27:G>D,S27:G>D,S29:S>N,S30:S>K,S30:S>K S37:Q>V,S37:Q>V,S41:K>T,S44:K>K
## 4                                                 S34:S>V,S34:S>V,S49:A>T
## 5
## 6                                  S30:S>I         S41:K>N,S44:K>K,S46:L>L
##                mutations.aa.CDR2
## 1
## 2
## 3                 S49:A>V,S50:A>A
## 4 S51:S>G,S55:G>Y,S55:G>Y,S56:S>N
## 5
## 6
##                                    mutations.aa.FR3
## 1
## 2
## 3 S64:S>T,S65:G>R,S66:S>F,S68:T>A,S75:S>N,S85:Y>Y
## 4                                  S79:Y>D,S80:L>L
## 5
## 6                                           S75:S>N
##                mutations.aa.CDR3 mutations.aa.FR4 pol.v pol.d.5 pol.d.3
## 1                                          S113:Q>Q    -1      -1      -1
## 2                                                      -1      -1      -1
## 3                 S90:L>I,S91:N>S S99:G>G,S105:I>M    -1      -1      -1
## 4                                          S107:Q>Q    -1      -1      -1
## 5                                                      -1      -1      -1
## 6 S88:Q>L,S90:S>T,S93:T>V,S93:T>V         S102:K>N    -1      -1      -1
##   pol.j canonical
## 1    -1      true
## 2    -1      true
```

```
## 3    -1      true
## 4    -1      true
## 5    -1      true
## 6    -1      true
##
contignt
## 1
GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCCTGGTCAAGCCTGGGGGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATTCACCT
TCAGTAGCTATAGCATGAACTGGGTCCGCCAGGCTCCAGGGAAGGGGCTGGAGTGGGTCTCATCCATTAGTAGTAGTAGTTA
CATATACTACGCAGACTCAGTGAAGGGCCGATTCACCATCTCCAGAGACAACGCCAAGAACTCACTGTATCTGCAAATGAACAGC
CTGAGAGCCGAGGACACGGCTGTGTATTACTGTGCGAGAGAGCCCAAAGGCGCAGCAGTGGCTGGACAACTTGACTACTGGGGCC
AGGGAACCCTGGTCACCGTCTCCTCAGGGAGTGCATCCGCCCCAACCCTTTTCCCCCTCGTCTCCTGTGAGAATTCCCCGTCGGA
TACGAGCAGCGTG
## 2 GAAATTGTGTTGACGCAGTCTCCAGGCACCCTGTCTTTGTCTCCAGGGGAAAGAGCCACCCTCTCCTGCAGGGCCAGTCA
GAGTGTTAGCAGCAGCTACTTAGCCTGGTACCAGCAGAAACCTGGCCAGGCTCCCAGGCTCCTCATCTATGGTGCATCCAGCAGG
GCCACTGGCATCCCAGACAGGTTCAGTGGCAGTGGGTCTGGGACAGACTTCACTCTCACCATCAGCAGACTGGAGCCTGAAGATT
TTGCAGTGTATTACTGTCAGCAGTATGGTAGCTCACCTCTATTCACTTTCGGCCCTGGGACCAAAGTGGATATCAAACGAACTGT
GGCTGCACCATCTGTCTTCATCTTCCCGCCATCTGATGAGCAGTTGAAATCTGGAACTGCCTCTGTTGTGTGCCTGCTGAATAAC
TTCTATCCCAGAGAGGCCAAAGTACAGTGGAAGGTGGATAACGCCCTCCAATCGGGTAACTCCCAGGAGAGTGTCACAGAGCAGG
ACAGCAAGGACAGCACCTACAGCCTCAGCAGCACCCTGACGCTGAGCAAAGCAGACTACGAGAA
## 3       GACATCCAGTTGACCCAGTCTCCATCCTTCCTGTCTGCATCTGTAGGAGACAGAGTCACCATCACTTGCCGGGC
CAGTCAGGATATTAACAAATATTTAGCCTGGTATCAGGTAAAACCAGGGACAGCCCCTAAACTCCTGATCTATGTTGCGTCCACT
TTGCAAAGTGGGGTCCCATCAAGGTTCAGCGGCACTAGATTTGGGGCAGAATTCACTCTCACAATCAACAGCCTGCAGCCTGAAG
ATTTTGCAACTTACTACTGTCAACAGATTAGTAGTTACCCTCTCACTTTCGGCGGGGGGACCAAGGTGGAGATGAAACGAACTGT
GGCTGCACCATCTGTCTTCATCTTCCCGCCATCTGATGAGCAGTTGAAATCTGGAACTGCCTCTGTTGTGTGCCTGCTGAATAAC
TTCTATCCCAGAGAGGCCAAAGTACAGTGGAAGGTGGATAACGCCCTCCAATCGGGTAACTCCCAGGAGAGTGTCACAGAGCAGG
ACAGCAAGGACAGCACCTACAGCCTCAGCAGCACCCTGACGCTGAGCAAAGCAGACTACGAGAA
## 4                                                                   GAGGTACTCCTCTTGGAGTCTG
GGGGAGGCTTGGCACACCCTGGGGGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATTCACCTTTAGCAGCTATGCCATGGTCTG
GGTCCGCCAGGCTCCAGGGAAGGGGCTGGAGTGGGTCTCAACTATTGGTGGTAGTGGTTATAACACATACTACGCAGACTCCGTG
AAGGGCCGGTTCACCATCTCCAGAGACAATTCCAAGAACACGCTGGATCTACAAATGAACAGCCTGAGAGCCGAGGACACGGCCG
TATATTACTGTGCGATGTGGGCATGGGAACTAGACTACTGGGGCCAGGGAACCCTGGTCACCGTCTCCTCAGCATCCCCGACCAG
CCCCAAGGTCTTCCCGCTGAGCCTCTGCAGCACCCAGCCAGATGGGAACGTGGTCATCGCCTGCCTGGTCCAGGGCTTCTTCCCC
CAGGAGCCACTCAGTGTGACCTGGAGCGAAAGCGGACAGGGCGTGACCGCCAGAAACTTCCCCC
## 5       GACATCCAGATGACCCAGTCTCCATCCTCCCTGTCTGCATCTGTAGGAGACAGAGTCACCATCACTTGCCGGGC
AAGTCAGAGCATTAGCAGCTATTTAAATTGGTATCAGCAGAAACCAGGGAAAGCCCCTAAGCTCCTGATCTATGCTGCATCCAGT
TTGCAAAGTGGGGTCCCATCAAGGTTCAGTGGCAGTGGATCTGGGACAGATTTCACTCTCACCATCAGCAGTCTGCAACCTGAAG
ATTTTGCAACTTACTACTGTCAACAGAGTTACAGTACCCTTTGGACGTTCGGCCAAGGGACCAAGGTGGAAATCAAACGAACTGT
GGCTGCACCATCTGTCTTCATCTTCCCGCCATCTGATGAGCAGTTGAAATCTGGAACTGCCTCTGTTGTGTGCCTGCTGAATAAC
TTCTATCCCAGAGAGGCCAAAGTACAGTGGAAGGTGGATAACGCCCTCCAATCGGGTAACTCCCAGGAGAGTGTCACAGAGCAGG
ACAGCAAGGACAGCACCTACAGCCTCAGCAGCACCCTGACGCTGAGCAAAGCAGACTACGAGAA
## 6       GACATCCAGATGACCCAGTCTCCATCCTCCCTGTCTGCATCTGTAGGAGACAGAGTCACCATCACTTGCCGGGC
AAGTCAGAGCATTAGCATCTATTTAAATTGGTATCAGCAGAAACCAGGGAATGCCCCTAAACTCCTTATCTATGCTGCATCCAGT
TTGCAAAGTGGGGTCCCATCAAGGTTCAGTGGCAGTGGATCTGGGACAGATTTCACTCTCACCATCAACAGTCTGCAACCTGAAG
ATTTTGCAACTTACTACTGTCTACAGACTTACAGTGTCCCTCGGACTTTTGGCCAGGGGACCAACCTGGAGATCAAACGAACTGT
GGCTGCACCATCTGTCTTCATCTTCCCGCCATCTGATGAGCAGTTGAAATCTGGAACTGCCTCTGTTGTGTGCCTGCTGAATAAC
TTCTATCCCAGAGAGGCCAAAGTACAGTGGAAGGTGGATAACGCCCTCCAATCGGGTAACTCCCAGGAGAGTGTCACAGAGCAGG
ACAGCAAGGACAGCACCTACAGCCTCAGCAGCACCCTGACGCTGAGCAAAGCAGACTACGAGAA
```

```
colnames(data)
```

```
##  [1] "read"               "header"             "cdr3nt"
##  [4] "cdr3aa"             "cdr.insert.qual"    "mutations.qual"
##  [7] "v.segment"          "d.segment"          "j.segment"
## [10] "cdr1.start.in.read" "cdr1.end.in.read"   "cdr2.start.in.read"
## [13] "cdr2.end.in.read"   "cdr3.start.in.read" "cdr3.end.in.read"
## [16] "v.end.in.cdr3"      "d.start.in.cdr3"    "d.end.in.cdr3"
## [19] "j.start.in.cdr3"    "v.del"              "d.del.5"
## [22] "d.del.3"            "j.del"              "mutations.nt.FR1"
## [25] "mutations.nt.CDR1"  "mutations.nt.FR2"   "mutations.nt.CDR2"
## [28] "mutations.nt.FR3"   "mutations.nt.CDR3"  "mutations.nt.FR4"
## [31] "rc"                 "complete"           "has.cdr3"
## [34] "in.frame"           "no.stop"            "mutations.aa.FR1"
## [37] "mutations.aa.CDR1"  "mutations.aa.FR2"   "mutations.aa.CDR2"
## [40] "mutations.aa.FR3"   "mutations.aa.CDR3"  "mutations.aa.FR4"
## [43] "pol.v"              "pol.d.5"            "pol.d.3"
## [46] "pol.j"              "canonical"          "contignt"
```

```
#from the migmap output we will choose the column that we will work with them and cha
nge the columns' name (same as filtered data)
preprocess <- function(data) {
data <- data %>% select(barcode = read, v_gene = v.segment, d_gene = d.segment, j_gen
e= j.segment)
data
}
data <- preprocess(data)
head(data)
```

```
##                barcode      v_gene      d_gene   j_gene
## 1 >ACACCCTCAGGCTGAA-1 IGHV3-21*01 IGHD6-19*01 IGHJ4*01
## 2 >ACACCCTCAGGCTGAA-1 IGKV3-20*01           . IGKJ3*01
## 3 >ACCTTTAGTACACCGC-1  IGKV1-9*01           . IGKJ4*01
## 4 >ACTGCTCCAGGATTGG-1 IGHV3-23*01  IGHD1-7*01 IGHJ4*01
## 5 >GTACTCCAGCGCTTAT-1 IGKV1-39*01           . IGKJ1*01
## 6 >TATCAGGCATGGAATA-1 IGKV1-39*01           . IGKJ2*01
```

```
#Number of cells (As we have several copies of each barcode,we need to count just one
copy of unique barcoe and calculate number of cells)

barcode_summary <- function(data)
{
  number_of_cells <- data %>%
    distinct(barcode) %>%
    dplyr::count()

number_of_cells$n
}
barcode_summary(data)
```

```
## [1] 1321
```

```
#look at to the quality of chains

count_table <- function(data)
{
  data %>% mutate(
    gene_group = ifelse(
     v_gene != 'None' & j_gene != 'None',
     "v_gene_and_j_gene",
     ifelse(
         v_gene != 'None' | j_gene != 'None',
        "v_gene_or_j_gene",
        "None"
     )
    )
  ) %>%
    group_by(gene_group) %>%
    summarise(count = n())
}
count_table(data)
```

```
## # A tibble: 1 x 2
##   gene_group        count
##   <chr>             <int>
## 1 v_gene_and_j_gene  2644
```

```
# In midmap data we do not have chain column that shows the type of chain in each cel
l. Therefore we need to creat chain column based on v_gene, d_gene and j_gene (the sa
me as filtered data)

matrix_gene_data <- as.matrix(data[,2:4])
 matrix_gene_data <- substr(matrix_gene_data, 1, 3) # get only first three characters
 data$chain <- apply(matrix_gene_data, 1, function(x) {
   x <- x[!(x %in% ".")] # removeing .
   x <- unique(x) # get unique value from row
   if(length(x) == 0) { # if all are . then return none
     "None"
   } else if (length(x) > 1) { # if more than 1 unique value then it's multi
     "Multi"
   } else { # otherwise just single chain value
     x
   }
 })
 head(data)
```

```
##               barcode      v_gene      d_gene   j_gene chain
## 1 >ACACCCTCAGGCTGAA-1 IGHV3-21*01 IGHD6-19*01 IGHJ4*01   IGH
## 2 >ACACCCTCAGGCTGAA-1 IGKV3-20*01           . IGKJ3*01   IGK
## 3 >ACCTTTAGTACACCGC-1  IGKV1-9*01           . IGKJ4*01   IGK
## 4 >ACTGCTCCAGGATTGG-1 IGHV3-23*01  IGHD1-7*01 IGHJ4*01   IGH
## 5 >GTACTCCAGCGCTTAT-1 IGKV1-39*01           . IGKJ1*01   IGK
## 6 >TATCAGGCATGGAATA-1 IGKV1-39*01           . IGKJ2*01   IGK
```

```
#Number of copies of each cell

barcode_summary <- function(data)
  {
  data %>%
    group_by(barcode)%>%
    summarize(count=n())
}
barcode_summary <- barcode_summary(data)
head(barcode_summary)
```

```
## # A tibble: 6 x 2
##   barcode             count
##   <chr>               <int>
## 1 >AAACCTGCACACTGCG-1     2
## 2 >AAACCTGCAGGTGGAT-1     3
## 3 >AAACCTGGTGTTCTTT-1     2
## 4 >AAACGGGCATGTCCTC-1     2
## 5 >AAACGGGTCCGTTGCT-1     2
## 6 >AAACGGGTCCTTTCTC-1     2
```

```
#B cells distribution by number of chains in a cell
barcode_summary <- function(data)
  {
  data %>%
    group_by(barcode) %>%
    summarize(count=n()) %>%
    group_by(count)%>%
    summarize(count_total=n())%>%
  mutate(
    count_total_pct = round(count_total/ sum(count_total)*100, 2)
  )
}
copy_barcode_migmap <- barcode_summary(data)
copy_barcode_migmap
```

```
## # A tibble: 4 x 3
##   count count_total count_total_pct
##   <int>       <int>           <dbl>
## 1     1          75            5.68
## 2     2        1177           89.1
## 3     3          61            4.62
## 4     4           8            0.61
```

```
#we want to know each cell contain how many chain

occurance_of_each_chain <- function(data)
  {
  many_conditions = c(
  '1IGL', '1IGH', '1IGK', '1IGH_1IGK_1IGL' , '1IGH_1IGL' , '1IGH_1IGK', '1IGH_2IGK',
'1IGH_2IGL')

   data %>%
    group_by(barcode, chain) %>%
    filter(chain %in% c('IGK','IGH','IGL')) %>%
     summarize(count=n()) %>%  #based on barcode we are counting the occurance of IG
H, IGK  and IGL
     unite('result_chain', count, chain, remove=F, sep='')%>% #we want to combine cou
nt column with chain column and put it in new column called "result_chain"
     summarize(
       type=paste(result_chain, collapse='_'), #
       count=sum(count)
      )%>%
     mutate(with_condition = ifelse(
       type %in% many_conditions,
       T, F  #for getting other conditions that is not in our condition list, I add n
ew column to check if it is in our condition or not! if it is, it will show "True" ot
herwise shows "F"
     ))%>%
  group_by(type, with_condition) %>% #at first we should know that is object in condi
tion or not! then count the number of objects in out condition
  summarise(total=n())
}
occurance_migmap <- occurance_of_each_chain(data)
head(occurance_migmap)
```

```
## # A tibble: 6 x 3
## # Groups:   type [6]
##   type           with_condition total
##   <chr>          <lgl>          <int>
## 1 1IGH           TRUE              10
## 2 1IGH_1IGK      TRUE             656
## 3 1IGH_1IGK_1IGL TRUE              23
## 4 1IGH_1IGL      TRUE             521
## 5 1IGH_2IGK      TRUE              10
## 6 1IGH_2IGL      TRUE              21
```

```
#there we want our condition! so just filter T, then make data frame from total and t
ype columns
with_condition <- data%>%
  occurance_of_each_chain()%>%
  filter(with_condition == T)%>%
  select(type,total)%>%
  data.frame()

#here we will collect just data that are not in our condition
without_condition_total <- data%>%
  occurance_of_each_chain()%>%
  filter(with_condition == F)%>%
  pull(total)%>%
  sum()

#now we have 2 separate data frame with condition and without! at first we add the ro
w of other to the list

without_condition <- data.frame('Other', without_condition_total)
names(without_condition) <- names(with_condition) #we should make the name of the col
umn of 2 dataset similar!
#here just combine 2 data frame together
Final_results <- rbind(with_condition, without_condition)%>%
  mutate(
    total_pct = round(total/ sum(total)*100, 2)
  )
Final_results
```

```
##             type total total_pct
## 1           1IGH    10      0.76
## 2      1IGH_1IGK   656     49.66
## 3 1IGH_1IGK_1IGL    23      1.74
## 4      1IGH_1IGL   521     39.44
## 5      1IGH_2IGK    10      0.76
## 6      1IGH_2IGL    21      1.59
## 7           1IGK    44      3.33
## 8           1IGL    21      1.59
## 9          Other    15      1.14
```

```
########################################
      #calculating chain distance
########################################
#for this purpos we need to have read.header in migmapdata instead of barcode because
it contains barcode and "contig"! here we need contig for recognizing the type of cha
in in "fasta file"! so we should not delete it like previous studies!
```

```
#load migmap output file
migmap <-read.csv ('/home/sedreh/ITMO/semester2/Bcellsproject/final_Rcode/PBMCs_of_a_
healthy_donor/migmap_result_healthy_donor.csv', sep="\t", header=TRUE)
#load fasta file for recognizing light chains in cells
d <- read.fasta('/home/sedreh/ITMO/semester2/Bcellsproject/final_Rcode/PBMCs_of_a_hea
lthy_donor/vdj_v1_hs_pbmc_b_filtered_contig.fasta')
head(d, n=1)
```

```
## $`ACACCCTCAGGCTGAA-1_contig_1`
##   [1] "t" "g" "g" "g" "g" "a" "g" "c" "t" "c" "t" "g" "a" "g" "a" "g" "a"
##  [18] "g" "g" "a" "g" "c" "c" "t" "t" "a" "g" "c" "c" "c" "t" "g" "g" "a"
##  [35] "t" "t" "c" "c" "a" "a" "g" "g" "c" "c" "t" "a" "t" "c" "c" "a" "c"
##  [52] "t" "t" "g" "g" "t" "g" "a" "t" "c" "a" "g" "c" "a" "c" "t" "g" "a"
##  [69] "g" "c" "a" "c" "c" "g" "a" "g" "g" "a" "t" "t" "c" "a" "c" "c" "a"
##  [86] "t" "g" "g" "a" "a" "c" "t" "g" "g" "g" "g" "c" "t" "c" "c" "g" "c"
## [103] "t" "g" "g" "g" "t" "t" "t" "t" "c" "c" "t" "t" "g" "t" "t" "g" "c"
## [120] "t" "a" "t" "t" "t" "t" "a" "g" "a" "a" "g" "g" "t" "g" "t" "c" "c"
## [137] "a" "g" "t" "g" "t" "g" "a" "g" "g" "t" "g" "c" "a" "g" "c" "t" "g"
## [154] "g" "t" "g" "g" "a" "g" "t" "c" "t" "g" "g" "g" "g" "g" "a" "g" "g"
## [171] "c" "c" "t" "g" "g" "t" "c" "a" "a" "g" "c" "c" "t" "g" "g" "g" "g"
## [188] "g" "g" "t" "c" "c" "c" "t" "g" "a" "g" "a" "c" "t" "c" "t" "c" "c"
## [205] "t" "g" "t" "g" "c" "a" "g" "c" "c" "t" "c" "t" "g" "g" "a" "t" "t"
## [222] "c" "a" "c" "c" "t" "t" "c" "a" "g" "t" "a" "g" "c" "t" "a" "t" "a"
## [239] "g" "c" "a" "t" "g" "a" "a" "c" "t" "g" "g" "g" "t" "c" "c" "g" "c"
## [256] "c" "a" "g" "g" "c" "t" "c" "c" "a" "g" "g" "g" "a" "a" "g" "g" "g"
## [273] "g" "c" "t" "g" "g" "a" "g" "t" "g" "g" "g" "t" "c" "t" "c" "a" "t"
## [290] "c" "c" "a" "t" "t" "a" "g" "t" "a" "g" "t" "a" "g" "t" "a" "g" "t"
## [307] "a" "g" "t" "t" "a" "c" "a" "t" "a" "t" "a" "c" "t" "a" "c" "g" "c"
## [324] "a" "g" "a" "c" "t" "c" "a" "g" "t" "g" "a" "a" "g" "g" "g" "c" "c"
## [341] "g" "a" "t" "t" "c" "a" "c" "c" "a" "t" "c" "t" "c" "c" "a" "g" "a"
## [358] "g" "a" "c" "a" "a" "c" "g" "c" "c" "a" "a" "g" "a" "a" "c" "t" "c"
## [375] "a" "c" "t" "g" "t" "a" "t" "c" "t" "g" "c" "a" "a" "a" "t" "g" "a"
## [392] "a" "c" "a" "g" "c" "c" "t" "g" "a" "g" "a" "g" "c" "c" "g" "a" "g"
## [409] "g" "a" "c" "a" "c" "g" "g" "c" "t" "g" "t" "g" "t" "a" "t" "t" "a"
## [426] "c" "t" "g" "t" "g" "c" "g" "a" "g" "a" "g" "a" "g" "c" "c" "c" "a"
## [443] "a" "a" "g" "g" "c" "g" "c" "a" "g" "c" "a" "g" "t" "g" "g" "c" "t"
## [460] "g" "g" "a" "c" "a" "a" "c" "t" "t" "g" "a" "c" "t" "a" "c" "t" "g"
## [477] "g" "g" "g" "c" "c" "a" "g" "g" "g" "a" "a" "c" "c" "c" "t" "g" "g"
## [494] "t" "c" "a" "c" "c" "g" "t" "c" "t" "c" "c" "t" "c" "a" "g" "g" "g"
## [511] "a" "g" "t" "g" "c" "a" "t" "c" "c" "g" "c" "c" "c" "c" "a" "a" "c"
## [528] "c" "c" "t" "t" "t" "t" "c" "c" "c" "c" "c" "t" "c" "g" "t" "c" "t"
## [545] "c" "c" "t" "g" "t" "g" "a" "g" "a" "a" "t" "t" "c" "c" "c" "c" "g"
## [562] "t" "c" "g" "g" "a" "t" "a" "c" "g" "a" "g" "c" "a" "g" "c" "g" "t"
## [579] "g"
## attr(,"name")
## [1] "ACACCCTCAGGCTGAA-1_contig_1"
## attr(,"Annot")
## [1] ">ACACCCTCAGGCTGAA-1_contig_1"
## attr(,"class")
## [1] "SeqFastadna"
```

```
#select needed columns from data and rename them like filtered data
preprocess_data <- function(migmap) {
migmap <- migmap %>% select(barcode = read.header, v_gene = v.segment, d_gene = d.seg
ment, j_gene= j.segment)
migmap
}
migmap <- preprocess_data(migmap)
head(migmap)
```

```
##                         barcode      v_gene      d_gene   j_gene
## 1 >ACACCCTCAGGCTGAA-1_contig_1 IGHV3-21*01 IGHD6-19*01 IGHJ4*01
## 2 >ACACCCTCAGGCTGAA-1_contig_3 IGKV3-20*01           . IGKJ3*01
## 3 >ACCTTTAGTACACCGC-1_contig_2  IGKV1-9*01           . IGKJ4*01
## 4 >ACTGCTCCAGGATTGG-1_contig_2 IGHV3-23*01  IGHD1-7*01 IGHJ4*01
## 5 >GTACTCCAGCGCTTAT-1_contig_1 IGKV1-39*01           . IGKJ1*01
## 6 >TATCAGGCATGGAATA-1_contig_1 IGKV1-39*01           . IGKJ2*01
```

```
#determine the chain type for each cell based on v, d and j gene and make "chain" col
umn

matrix_gene_data <- as.matrix(migmap[,2:4])
 matrix_gene_data <- substr(matrix_gene_data, 1, 3) # get only first three characters
 migmap$chain <- apply(matrix_gene_data, 1, function(x) {
   x <- x[!(x %in% ".")] # removeing .
   x <- unique(x) # get unique value from row
   if(length(x) == 0) { # if all are . then return none
     "None"
   } else if (length(x) > 1) { # if more than 1 unique value then it's multi
     "Multi"
   } else { # otherwise just single chain value
     x
   }
 })
 head(migmap)
```

```
##                         barcode      v_gene      d_gene   j_gene chain
## 1 >ACACCCTCAGGCTGAA-1_contig_1 IGHV3-21*01 IGHD6-19*01 IGHJ4*01   IGH
## 2 >ACACCCTCAGGCTGAA-1_contig_3 IGKV3-20*01           . IGKJ3*01   IGK
## 3 >ACCTTTAGTACACCGC-1_contig_2  IGKV1-9*01           . IGKJ4*01   IGK
## 4 >ACTGCTCCAGGATTGG-1_contig_2 IGHV3-23*01  IGHD1-7*01 IGHJ4*01   IGH
## 5 >GTACTCCAGCGCTTAT-1_contig_1 IGKV1-39*01           . IGKJ1*01   IGK
## 6 >TATCAGGCATGGAATA-1_contig_1 IGKV1-39*01           . IGKJ2*01   IGK
```

```
#look at the data
look <- migmap %>%
  select(
    v_gene, j_gene, d_gene, chain
  )
head(look)
```

```
##        v_gene   j_gene      d_gene chain
## 1 IGHV3-21*01 IGHJ4*01 IGHD6-19*01   IGH
## 2 IGKV3-20*01 IGKJ3*01           .   IGK
## 3  IGKV1-9*01 IGKJ4*01           .   IGK
## 4 IGHV3-23*01 IGHJ4*01  IGHD1-7*01   IGH
## 5 IGKV1-39*01 IGKJ1*01           .   IGK
## 6 IGKV1-39*01 IGKJ2*01           .   IGK
```

```
occurance_of_each_chain <- function(migmap)
  {
  migmap %>%
    group_by(barcode, chain) %>%
    filter(chain %in% c('IGK','IGH','IGL')) %>%
    summarize(count=n()) %>%
    unite('result_chain', count, chain, remove=F, sep='') %>%
    summarize(type=paste(result_chain, collapse='_'), count=sum(count))
}
occurance_of_each_chain <- occurance_of_each_chain(migmap)
head(occurance_of_each_chain)
```

```
## # A tibble: 6 x 3
##   barcode                      type  count
##   <fct>                        <chr> <int>
## 1 >AAACCTGCACACTGCG-1_contig_2 1IGL      1
## 2 >AAACCTGCACACTGCG-1_contig_5 1IGH      1
## 3 >AAACCTGCAGGTGGAT-1_contig_3 1IGK      1
## 4 >AAACCTGCAGGTGGAT-1_contig_5 1IGL      1
## 5 >AAACCTGCAGGTGGAT-1_contig_6 1IGH      1
## 6 >AAACCTGGTGTTCTTT-1_contig_1 1IGH      1
```

```
#In this step we need to recognize the type of each chain with contig to search the s
equence in fasta file
estimate_condition <- function(migmap) {
  migmap %>%
  separate(barcode, into=c('bc','contig'),sep = '1_',remove = F) %>% # spliting barco
de into bc(barcode without contig) and contig
    mutate(bc=substr(bc,2,18)) %>% # slicing only barcode (ex:without > and -1 form >
AAACGGGTCCGTTGCT-1)
  filter(chain %in% c('IGK','IGH','IGL')) %>%
     group_by(bc, chain) %>%
  summarize(contig=paste(contig, collapse = ','), count=n()) %>% # for every barcode
 and contig, counting the occurance
  unite('result_chain', count, chain, remove=F, sep='') %>% # combining the count cal
culated before with chain
  unite('contig', result_chain, contig, sep = '-', remove = F) %>% # combining contig
with result_chain calculated just before
  group_by(bc) %>%
  summarize(
    type=paste(result_chain, collapse='_'),
    count=sum(count),
    contig=paste(contig, collapse = '@')) # for every barcode estimating condition (t
ype) and contig (to be used for get_score function)
}
 estimate_condition(migmap)
```

```
## # A tibble: 1,321 x 4
##    bc               type         count contig
##    <chr>            <chr>        <int> <chr>
##  1 AAACCTGCACACTGC… 1IGH_1IGL        2 1IGH-contig_5@1IGL-contig_2
##  2 AAACCTGCAGGTGGA… 1IGH_1IGK_1…     3 1IGH-contig_6@1IGK-contig_3@1IGL-co…
##  3 AAACCTGGTGTTCTT… 1IGH_1IGK        2 1IGH-contig_1@1IGK-contig_4
##  4 AAACGGGCATGTCCT… 1IGH_1IGL        2 1IGH-contig_1@1IGL-contig_3
##  5 AAACGGGTCCGTTGC… 1IGH_1IGK        2 1IGH-contig_2@1IGK-contig_1
##  6 AAACGGGTCCTTTCT… 1IGH_1IGK        2 1IGH-contig_2@1IGK-contig_1
##  7 AAAGCAACACAGCGT… 1IGK             1 1IGK-contig_2
##  8 AAAGCAATCAAAGTA… 1IGH_1IGK        2 1IGH-contig_2@1IGK-contig_1
##  9 AAAGCAATCAACGGG… 1IGH_1IGL        2 1IGH-contig_1@1IGL-contig_4
## 10 AAAGCAATCACGAAG… 1IGH_1IGL        2 1IGH-contig_2@1IGL-contig_1
## # … with 1,311 more rows
```

```
#just for explanation
look1 <- migmap %>%
  separate(barcode, into=c('bc','contig'),sep = '1_',remove = F)

head(look1)
```

```
##                         barcode                bc   contig      v_gene
## 1 >ACACCCTCAGGCTGAA-1_contig_1 >ACACCCTCAGGCTGAA- contig_1 IGHV3-21*01
## 2 >ACACCCTCAGGCTGAA-1_contig_3 >ACACCCTCAGGCTGAA- contig_3 IGKV3-20*01
## 3 >ACCTTTAGTACACCGC-1_contig_2 >ACCTTTAGTACACCGC- contig_2  IGKV1-9*01
## 4 >ACTGCTCCAGGATTGG-1_contig_2 >ACTGCTCCAGGATTGG- contig_2 IGHV3-23*01
## 5 >GTACTCCAGCGCTTAT-1_contig_1 >GTACTCCAGCGCTTAT- contig_1 IGKV1-39*01
## 6 >TATCAGGCATGGAATA-1_contig_1 >TATCAGGCATGGAATA- contig_1 IGKV1-39*01
##        d_gene   j_gene chain
## 1 IGHD6-19*01 IGHJ4*01   IGH
## 2           . IGKJ3*01   IGK
## 3           . IGKJ4*01   IGK
## 4  IGHD1-7*01 IGHJ4*01   IGH
## 5           . IGKJ1*01   IGK
## 6           . IGKJ2*01   IGK
```

```
#just for explanation
look2 <- migmap %>%
  separate(barcode, into=c('bc','contig'),sep = '1_',remove = F)  %>%
  mutate(bc=substr(bc,2,18))%>%
  filter(chain %in% c('IGK','IGH','IGL')) %>%
  group_by(bc, chain) %>%
  summarize(contig=paste(contig, collapse = ','), count=n())

head(look2)
```

```
## # A tibble: 6 x 4
## # Groups:   bc [3]
##   bc                chain contig   count
##   <chr>             <chr> <chr>    <int>
## 1 AAACCTGCACACTGCG- IGH   contig_5     1
## 2 AAACCTGCACACTGCG- IGL   contig_2     1
## 3 AAACCTGCAGGTGGAT- IGH   contig_6     1
## 4 AAACCTGCAGGTGGAT- IGK   contig_3     1
## 5 AAACCTGCAGGTGGAT- IGL   contig_5     1
## 6 AAACCTGGTGTTCTTT- IGH   contig_1     1
```

```
#just for explanation
look3 <- migmap %>%
  separate(barcode, into=c('bc','contig'),sep = '1_',remove = F)  %>%
  mutate(bc=substr(bc,2,18))%>%
  filter(chain %in% c('IGK','IGH','IGL')) %>%
  group_by(bc, chain) %>%
  summarize(contig=paste(contig, collapse = ','), count=n()) %>%
  unite('result_chain', count, chain, remove=F, sep='') %>% # combining the count cal
culated before with chain
  unite('contig', result_chain, contig, sep = '-', remove = F) %>%
  group_by(bc) %>%
  summarize(
    type=paste(result_chain, collapse='_'),
    count=sum(count),
    contig=paste(contig, collapse = '@'))

head(look3)
```

```
## # A tibble: 6 x 4
##   bc               type          count contig
##   <chr>            <chr>         <int> <chr>
## 1 AAACCTGCACACTGC… 1IGH_1IGL         2 1IGH-contig_5@1IGL-contig_2
## 2 AAACCTGCAGGTGGA… 1IGH_1IGK_1I…     3 1IGH-contig_6@1IGK-contig_3@1IGL-co…
## 3 AAACCTGGTGTTCTT… 1IGH_1IGK         2 1IGH-contig_1@1IGK-contig_4
## 4 AAACGGGCATGTCCT… 1IGH_1IGL         2 1IGH-contig_1@1IGL-contig_3
## 5 AAACGGGTCCGTTGC… 1IGH_1IGK         2 1IGH-contig_2@1IGK-contig_1
## 6 AAACGGGTCCTTTCT… 1IGH_1IGK         2 1IGH-contig_2@1IGK-contig_1
```

```
#calculate score of distance between 2 light chain in cells with dual light chain con
dition
mat <- nucleotideSubstitutionMatrix(match = 0, mismatch = 1, baseOnly = TRUE)

get_alignment <- function(x) {

    # There are two types of contigs
    # 1. 1IGH-contig_5@1IGL-contig_2 (with any two) (same type of light chain)
    # 2. 1IGH-contig_6@1IGK-contig_3@1IGL-contig_5 (with all three) (different tyeps
 of light chain)

    # once we run this code, we'll know if it's type 1 or type 2
    # once we split by '@' type 1 will have lenght 2(becauze one @), but type 2 will
 have length 3(becauze two @)
    contig <- strsplit(x['contig'], '@')[[1]]

    if(length(contig) == 2) {
      # if contig is of type 1 then we run this part becauze we have same type of lig
ht chain
      # so we use contig only once here
      contig <- strsplit(contig[2], '-')[[1]]
      contig <- strsplit(contig[2], ',')[[1]]

      # we get barcode for each light chains
      barcodes <- c(
        paste(x['bc'], contig[1], sep = '1_'),
        paste(x['bc'], contig[2], sep = '1_'))
    } else {
      # if contig is of type 2 then we run this part bcz we have two different types
 of light chain
      # so we use contig1 and contig2 here
      contig1 <- strsplit(contig[2], '-')[[1]]
      contig1 <- strsplit(contig1[2], ',')[[1]]

      contig2 <- strsplit(contig[3], '-')[[1]]
      contig2 <- strsplit(contig2[2], ',')[[1]]

      # we get barcode for each light chains
       barcodes <- c(
        paste(x['bc'], contig1[1], sep = '1_'),
        paste(x['bc'], contig2[1], sep = '1_'))
    }

    # finally we use this part to calcualte the alignment
    s1 <- DNAString(
      toupper(
        paste(d[barcodes[1]][[1]], collapse = '')
      )
    )

    s2 <- DNAString(
      toupper(
        paste(d[barcodes[2]][[1]], collapse = '')
      )
    )

    globalAlign <-
```

```
      pairwiseAlignment(s1, s2, substitutionMatrix = mat,
                        gapOpening = 0, gapExtension = -1, scoreOnly=T)
    globalAlign
  }

get_score <- function(s, condition) {
  estimate_condition(s) %>% # properly estimate condition
    filter(type %in%  c(condition)) %>% # THIS PART
    apply(1, get_alignment) %>%
    unlist

}
scores_1IGH_2IGL <- -get_score(migmap, '1IGH_2IGL')
scores_1IGH_2IGL
```

```
##  [1] 149 158 180 136 156 113 102 199 139 162 175 185 192  22 174 181 172
## [18] 226 110 202 194
```

```
scores_1IGH_1IGK_1IGL <- -get_score(migmap, '1IGH_1IGK_1IGL')
scores_1IGH_1IGK_1IGL
```

```
##  [1] 316 310 305 408 316 338 435 316 366 337 350 352 303 294 319 293 333
## [18] 298 285 335 309 350 282
```

```
scores_1IGH_2IGK <- -get_score(migmap, '1IGH_2IGK')
scores_1IGH_2IGK
```

```
##  [1] 130 130 411 232 113 279 128 213 127 135
```

```
#Making dataframe from all conditions(with two light chain) with distance score

IGH_2IGk.data <- data.frame("1IGH_2IGk",scores_1IGH_2IGK) %>% select(condition = "X.1
IGH_2IGk.", score = "scores_1IGH_2IGK")
#names(IGH_2IGK.data)

IGH_2IGL.data <- data.frame('1IGH_2IGL', scores_1IGH_2IGL)  %>%
  data.frame("1IGH_2IGL",scores_1IGH_2IGL) %>% select(condition = "X.1IGH_2IGL.", sco
re = "scores_1IGH_2IGL")
#names(IGH_2IGL.data)

IGH_IGL_IGK.data <- data.frame("1IGH_1IGK_1IGL",scores_1IGH_1IGK_1IGL) %>%
  select(condition = "X.1IGH_1IGK_1IGL.", score = "scores_1IGH_1IGK_1IGL")
#names(IGH_IGL_IGK.data)

migmap <- rbind(IGH_2IGk.data, IGH_2IGL.data, IGH_IGL_IGK.data)
migmap$condition <- as.factor(migmap$condition)
migmap$condition
```

```
##  [1] 1IGH_2IGk      1IGH_2IGk      1IGH_2IGk      1IGH_2IGk
##  [5] 1IGH_2IGk      1IGH_2IGk      1IGH_2IGk      1IGH_2IGk
##  [9] 1IGH_2IGk      1IGH_2IGk      1IGH_2IGL      1IGH_2IGL
## [13] 1IGH_2IGL      1IGH_2IGL      1IGH_2IGL      1IGH_2IGL
## [17] 1IGH_2IGL      1IGH_2IGL      1IGH_2IGL      1IGH_2IGL
## [21] 1IGH_2IGL      1IGH_2IGL      1IGH_2IGL      1IGH_2IGL
## [25] 1IGH_2IGL      1IGH_2IGL      1IGH_2IGL      1IGH_2IGL
## [29] 1IGH_2IGL      1IGH_2IGL      1IGH_2IGL      1IGH_1IGK_1IGL
## [33] 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL
## [37] 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL
## [41] 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL
## [45] 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL
## [49] 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL
## [53] 1IGH_1IGK_1IGL 1IGH_1IGK_1IGL
## Levels: 1IGH_2IGk 1IGH_2IGL 1IGH_1IGK_1IGL
```

```
#plot distance between 2 light chain in cells with dual light chain condition
p <- ggplot(migmap, aes(x=condition, y=score, fill=condition)) +
  geom_boxplot(outlier.colour="red", outlier.shape=1,
               outlier.size=1)+
  labs(title="Plot of distance between two light chain",x="condition", y = "score")+
scale_color_brewer(palette="Dark2")+
coord_cartesian(ylim = c(2, 600))
#one of the reasons for using is puting limit for x and y

#p + theme_classic()
p
```