1. ID检查
    1. 发现SP数据库中的CD97在10X数据里没有，经NCBI检查发现其标准名称实际为ADGRE5，此名称在10X数据里有，故需要做系统的ID检查
    2. 用reutils将SP数据库提供的全部蛋白的ENTREZ gene id用efetch获取其数据库记录，获得其对应的标准基因名称，再和原来一开始SP数据库里所有ENTREZ gene symbol合起来去10X里取对应基因的行，获得了1000+个基因的表达谱，其中有70+个基因通过上述方法救回来了

2. 枚举所有gene pair的表达谱，并对每一个gene pair，检查三大类细胞中各有百分之多少的细胞同时表达了这个gene pair的2个基因（表达的标准：标准化后表达量>0或1；另：标准化后表达量范围为0~10，均值和中位数均在1.5附近），最后用百分比筛选，要求t和b中都只有不到10%的细胞同时表达、但tlb mem中至少有60%的细胞同时表达
    1. 原始一共有56万多个pair（是组合数，不是排列数，下同）
    2. 至少在一个细胞里2者同时测到（即表达量>0）的pair有39万个，其中至少在一个细胞里2者同时>1的pair有36万个

大于0的：

```
gene.1 gene.2     b.cell      t.cell tlb.mem.cell
1:    CD79A   CD3D 0.03300624 0.031045446    0.8695652
2:    CD79A   CD3G 0.01427297 0.024125678    0.7246377
3:    CD79A   IL7R 0.05441570 0.027679072    0.7826087
4:    CD79B   CD3D 0.01962533 0.082850196    0.8115942
5:    CD79B   CD3G 0.01070473 0.071067889    0.6521739
6:    CD79B   IL7R 0.04371097 0.064709183    0.7101449
7: HLA-DQA1   CD3D 0.01605709 0.050121563    0.6666667
8: HLA-DQA1   IL7R 0.03211418 0.028053114    0.6231884
9: HLA-DQA2   IL7R 0.04995540 0.071628951    0.7101449
10: HLA-DQB1   CD3D 0.01962533 0.082476155    0.6521739
11:     IGHM   CD3D 0.01516503 0.010286142    0.7391304
12:     IGHM   IL7R 0.03657449 0.009164017    0.6376812
13:    MS4A1   CD3D 0.02943800 0.009725079    0.8985507
14:    MS4A1   CD3G 0.01427297 0.005610623    0.7246377
15:    MS4A1   IL7R 0.05173952 0.006732747    0.7681159
```

大于1的：

```
     gene.1 gene.2     b.cell      t.cell tlb.mem.cell
1:    CD79A   CD3D 0.03211418 0.026930989    0.8115942
2:    CD79A   IL7R 0.05084746 0.024312699    0.7101449
3:    CD79B   CD3D 0.01873327 0.072564055    0.7391304
4:    CD79B   IL7R 0.04103479 0.057228352    0.6086957
5: HLA-DQA2   IL7R 0.04727921 0.064522162    0.6521739
6:     IGHM   CD3D 0.01516503 0.008602955    0.6376812
7:    MS4A1   CD3D 0.02854594 0.008976996    0.8115942
8:    MS4A1   IL7R 0.04995540 0.005984664    0.6521739
```

原始代码，需要准备all_cell, t_cell, b_cell, tlb_mem_cell

```
library("readxl")
library("data.table")
library("magrittr")
library("foreach")
library("reutils")

## needs `all_cell`

rownames(all_cell)

db.dt <- read_excel("./S2_File.xlsx", sheet=1) %>% data.table

ENTREZ.ids.to.check.vector <- setdiff(db.dt[`ENTREZ gene symbol` %in%
rownames(all_cell) == FALSE, ENTREZ_gene_ID] %>% unique %>% sort, 0)


foreach(temp.start=seq(1, length(ENTREZ.ids.to.check.vector), 50)) %do% {
  temp.end <- min(temp.start + 50 -1, length(ENTREZ.ids.to.check.vector))
  cat(date(), "fetching ids between ", temp.start, " and ", temp.end, "\n")
  date(); ENTREZ.ids.efetch <- efetch(uid =
ENTREZ.ids.to.check.vector[temp.start:temp.end], db = "gene", retmax = 10000,
                                      outfile=paste(sep="", "test-", temp.start,
"-", temp.end, ".xml")); date()
}

## rescued gene symbols from mislabeling in XX db
ENTREZ.gene.symbols.cleaned.vector <- system("grep Gene-ref_locus test-*-*.xml |
sed -E -e 's/.*>(.*)<.*/\\1/'", intern=TRUE)



################ copied from dingyang_tlb.R

t_expr <- t_cell@assays$RNA@data
b_expr <- b_cell@assays$RNA@data
tlb_mem_expr <- TLB_mem@assays$RNA@data



t.expr.melt.dt <- t_expr %>%
  {
    temp.dt <- summary(.) %>% as.data.table;
    temp.dt[, gene:=rownames(.)[i]]
    temp.dt[, cell.barcode:=colnames(.)[j]]
  }

b.expr.melt.dt <- b_expr %>%
  {
    temp.dt <- summary(.) %>% as.data.table;
    temp.dt[, gene:=rownames(.)[i]]
    temp.dt[, cell.barcode:=colnames(.)[j]]
  }

tlb.mem.expr.melt.dt <- tlb_mem_expr %>%
  {
```

```
    temp.dt <- summary(.) %>% as.data.table;
    temp.dt[, gene:=rownames(.)[i]]
    temp.dt[, cell.barcode:=colnames(.)[j]]
  }

b.expr.melt.dt[, cell.type:="b.cell"][cell.barcode %in% tlb.mem.expr.melt.dt[,
cell.barcode], cell.type:="tlb.mem.cell"]
t.expr.melt.dt[, cell.type:="t.cell"]

combined.dt <- list(b.expr.melt.dt, t.expr.melt.dt) %>% rbindlist
combined.dt[, cell.type.count:=c('b.cell'=ncol(b_expr), 't.cell'=ncol(t_expr),
'tlb.mem.cell'=ncol(tlb_mem_expr))[cell.type]]



valid.surface.protein.gene.names.vector <- c(db.dt[, `ENTREZ gene symbol`],
ENTREZ.gene.symbols.cleaned.vector) %>% intersect(combined.dt[, gene])

combined.SP.dt <- combined.dt[gene %in% valid.surface.protein.gene.names.vector]

valid.SP.pairs.dt <- combn(combined.SP.dt[, gene] %>% unique, 2) %>% t %>%
data.table %>% setnames(c("gene.1", "gene.2"))

date();
valid.SP.pairs.with.expr.dt <- merge(x=valid.SP.pairs.dt, y=combined.SP.dt[,
list(cell.barcode, cell.type, cell.type.count, gene.1=gene, gene.1.expr=x)],
by="gene.1", all=FALSE, allow.cartesian=TRUE) %>%
  {merge(x=., y=combined.SP.dt[, list(cell.barcode, gene.2=gene, gene.2.expr=x)],
by=c("gene.2", "cell.barcode"), all=FALSE)}
date()
## 2min for all SP pairs and all cells

fwrite(valid.SP.pairs.with.expr.dt, "./210419-
valid.SP.pairs.with.expr.dt.txt.gz")

date();
valid.SP.pairs.with.expr.metrics.dt <- valid.SP.pairs.with.expr.dt %>%
  {.[gene.1.expr>1 & gene.2.expr>1, data.table(count.of.expressed=.N,
pct.of.expressed=.N/cell.type.count), list(gene.1, gene.2, cell.type,
cell.type.count)]}
date()
## 2min

valid.SP.pairs.with.expr.metrics.dt %>%
  dcast(gene.1 + gene.2 ~ cell.type, value.var="pct.of.expressed", fill=-1) %>%
  data.table %>%
  {.[b.cell<0.1 & t.cell <0.1 & tlb.mem.cell>0.6]}
```