

Figure_5

Kaitlyn Lagattuta

2024-11-20

```
suppressPackageStartupMessages(source("utils.R"))

## Warning: package 'tidyr' was built under R version 4.3.2

## Warning: package 'ggplot2' was built under R version 4.3.2

metadata = read.csv("data/PAN.A01.v01.raw_count.20210429.NKT_metadata.csv", header=TRUE)
metadata = metadata[metadata$organ=="TH",]
## quality control filtering
metadata = metadata[metadata$mito<.1,]
metadata = metadata[metadata$n_genes>=500,]

tcr = readRDS("data/suo_pairedTCRdf.rds")
tcr = tcr[tcr$id %in% metadata$index,]
tcr = label_invariants2(tcr)
tcrscores = tcrpheno::score_tcrs(tcr, chain="ab")

## hash-2.2.6.3 provided by Decision Patterns

## [1] "adding CDR1 and CDR2 based on V gene..."
## [1] "identifying amino acids at each position..."
## [1] "converting amino acids into Atchley factors..."
## [1] 13373
## [1] 13373
## [1] "adding interactions between adjacent residues..."
## [1] "TCRs featurized!"
## [1] "scoring TCRs..."
## [1] "all done!"

tcrscores$index = rownames(tcrscores)
metadata = left_join(metadata, tcr)

## Joining with 'by = join_by(n_counts, n_genes, file, mito, doublet_scores,
## predicted_doublets, old_annotation_uniform, organ, Sort_id, age, method, donor,
## sex, Sample, scvi_clusters, is_maternal_contaminant, anno_lvl_2_final_clean,
## celltype_annotation)'
```

```
tcrscores = left_join(tcrscores, metadata)
```

```
## Joining with 'by = join_by(index)'
```

```
t = table(tcrscores$donor)
tcrscores = tcrscores[tcrscores$donor %in% names(t)[t>=100],]
tcrscores$tphase = "other"
tcrscores$tphase[grepl("DN", tcrscores$celltype_annotation)] = "DN"
tcrscores$tphase[grepl("DP", tcrscores$celltype_annotation)] = "DP"
tcrscores$tphase[tcrscores$celltype_annotation %in% c("CD4+T", "CD8+T", "TREG")] = "SP"

tmp = tcrscores[tcrscores$tphase %in% c("DP", "SP"),]
tmp$Donor = tmp$donor
tmp$lcat = "thymus"
tmp$target4 = tmp$tphase=="SP"
tmp$X4 = tmp$TCR.mem
```

```
plot_TCRscore_validation_bybin(tmp, 4, lcat="thymus", colors = c("turquoise4"), xlab = "TCR-mem score",
```

```
## Joining with 'by = join_by(ctl)'
```

```
## # A tibble: 3 x 7
##   effect   group term          estimate std.error statistic    p.value
##   <chr>    <chr> <chr>          <dbl>    <dbl>    <dbl>    <dbl>
## 1 fixed   <NA> (Intercept)   -0.0169    0.396   -0.0427    0.966
## 2 fixed   <NA> TCRscore      0.141     0.0267    5.28     0.000000128
## 3 ran_pars Donor sd_(Intercept)  0.966     NA        NA        NA
```

```
## 'summarise()' has grouped output by 'bin'. You can override using the '.groups'
## argument.
```

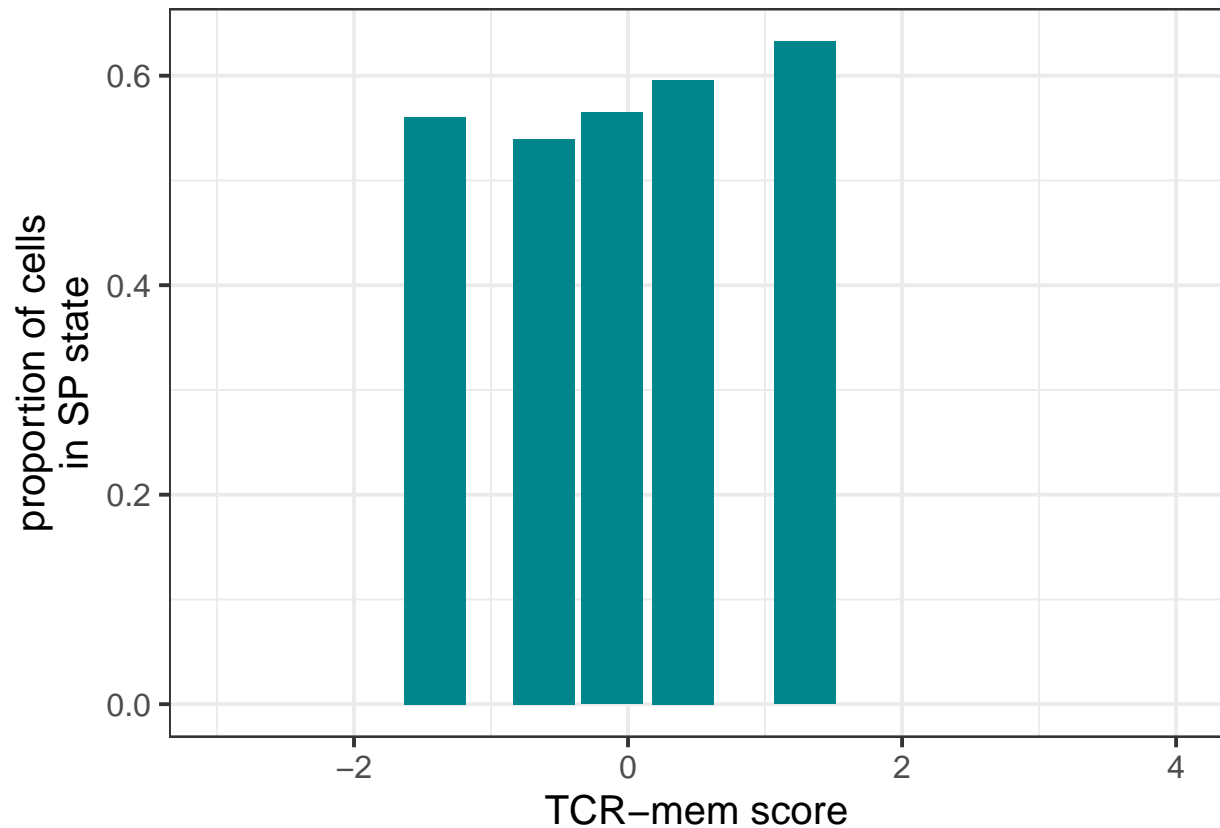
```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
## # A tibble: 3 x 2
##   mmp   prop.pos
##   <chr>    <dbl>
## 1 maxptl  0.773
## 2 minptl  0.533
## 3 other   0.577
```

```
## Joining with 'by = join_by(bin)'
```

```
## # A tibble: 5 x 9
##       bin   OR lowOR highOR term          lcat    qtl minbin maxbin
##   <dbl> <dbl> <dbl> <dbl> <chr>    <chr> <dbl> <dbl> <dbl>
## 1 -1.41  1.08 0.921  1.26 bin-1.40946984417489 thymus    20 -3.82 -0.895
```

```
## 2 -0.119 1.14 0.974 1.33 bin-0.118889573644414 thymus 60 -0.357 0.122
## 3 0.402 1.28 1.09 1.50 bin0.40186517804777 thymus 80 0.122 0.706
## 4 1.29 1.50 1.28 1.76 bin1.2933999374633 thymus 100 0.706 3.46
## 5 -0.617 1 1 1 ref thymus 40 -0.894 -0.357
```



```
thy = tmp
bld = readRDS("data/Dataset3_scored_wannotations.rds")
df = data.frame(data = c(rep("thymus", nrow(thy)), rep("blood", nrow(bld))), donor = c(thy$donor, bld$D
qu = quantile(df$TCR.mem, probs=seq(0.1, 1, by=0.1))
df$qu = sapply(df$TCR.mem, function(x) names(qu)[min(which(qu>=x))])
gr = df %>% group_by(qu) %>% dplyr::summarise(meanTCRmem = mean(TCR.mem))
df = left_join(df, gr)
```

```
## Joining with 'by = join_by(qu)'
```

```
cols = c("bin", "OR", "lowOR", "highOR", "term")
df$bin = df$meanTCRmem
df$Donor = df$donor
df$target = df$target4
st.thy = get_bin_stats(df[df$data=="thymus",], cols, quantile_bin=0.1)
st.bld = get_bin_stats(df[df$data!="thymus",], cols, quantile_bin=0.1)

st.thy = st.thy[,2:5]
colnames(st.thy)[1:3] = paste(colnames(st.thy)[1:3], ".thy", sep="")
```

```

st.bld = st.bld[,2:5]
colnames(st.bld)[1:3] = paste(colnames(st.bld)[1:3], ".bld", sep="")

tp = full_join(st.thy, st.bld)

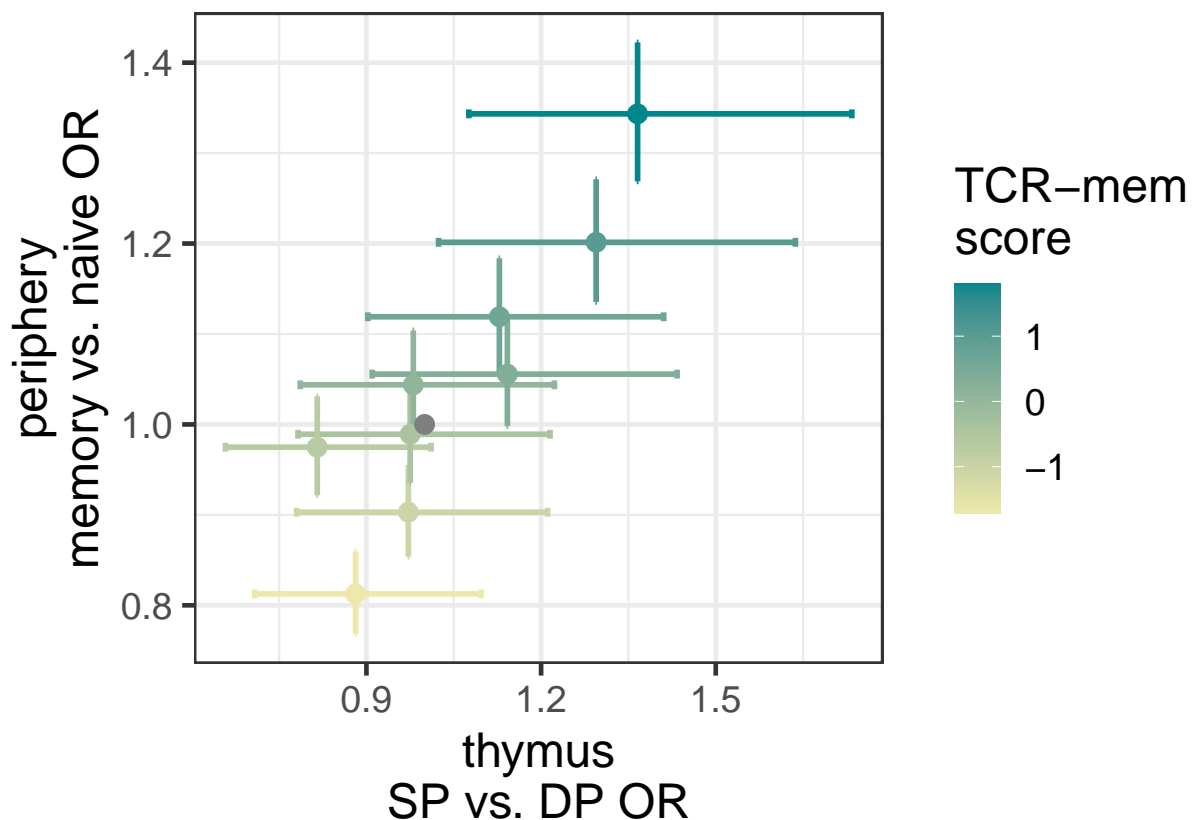
## Joining with 'by = join_by(term)'

tp = tp[!(grepl("Intercept", as.character(tp$term))),]
tp$meanTCRmem = as.numeric(as.character(gsub("bin", "", as.character(tp$term))))

## Warning: NAs introduced by coercion

g = ggplot(tp, aes(OR.thy, OR.bld, color=meanTCRmem))
g = g + geom_errorbar(aes(xmin=lowOR.thy, xmax=highOR.thy), size=1)
g = g + geom_errorbar(aes(ymin=lowOR.bld, ymax=highOR.bld), size=1) + theme_bw(base_size=18)
g = g + geom_point(size=3)
g = g + scale_color_gradient(low="palegoldenrod", high="turquoise4")
g = g + xlab("thymus\nSP vs. DP OR") + ylab("periphery\nmemory vs. naive OR")
g + labs(color="TCR-mem\nscore")

```



```

yi <- c(0.136, 0.141) # observed effect size estimates
vi <- c(0.00655^2, 0.0267^2) # corresponding sampling variances
rma(yi, vi, method="FE")

```

```
##
## Fixed-Effects Model (k = 2)
##
## I^2 (total heterogeneity / total variability): 0.00%
## H^2 (total variability / sampling variability): 0.03
##
## Test for Heterogeneity:
## Q(df = 1) = 0.0331, p-val = 0.8557
##
## Model Results:
##
## estimate      se      zval      pval      ci.lb      ci.ub
## 0.1363 0.0064 21.4236 <.0001 0.1238 0.1488 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
igor.thy = process_igor_output("data/suo_igorresults_117.rds")
tcrscores = left_join(tcrscores, igor.thy[,c("id", "ab_ins")], by=c("index"="id"))
```

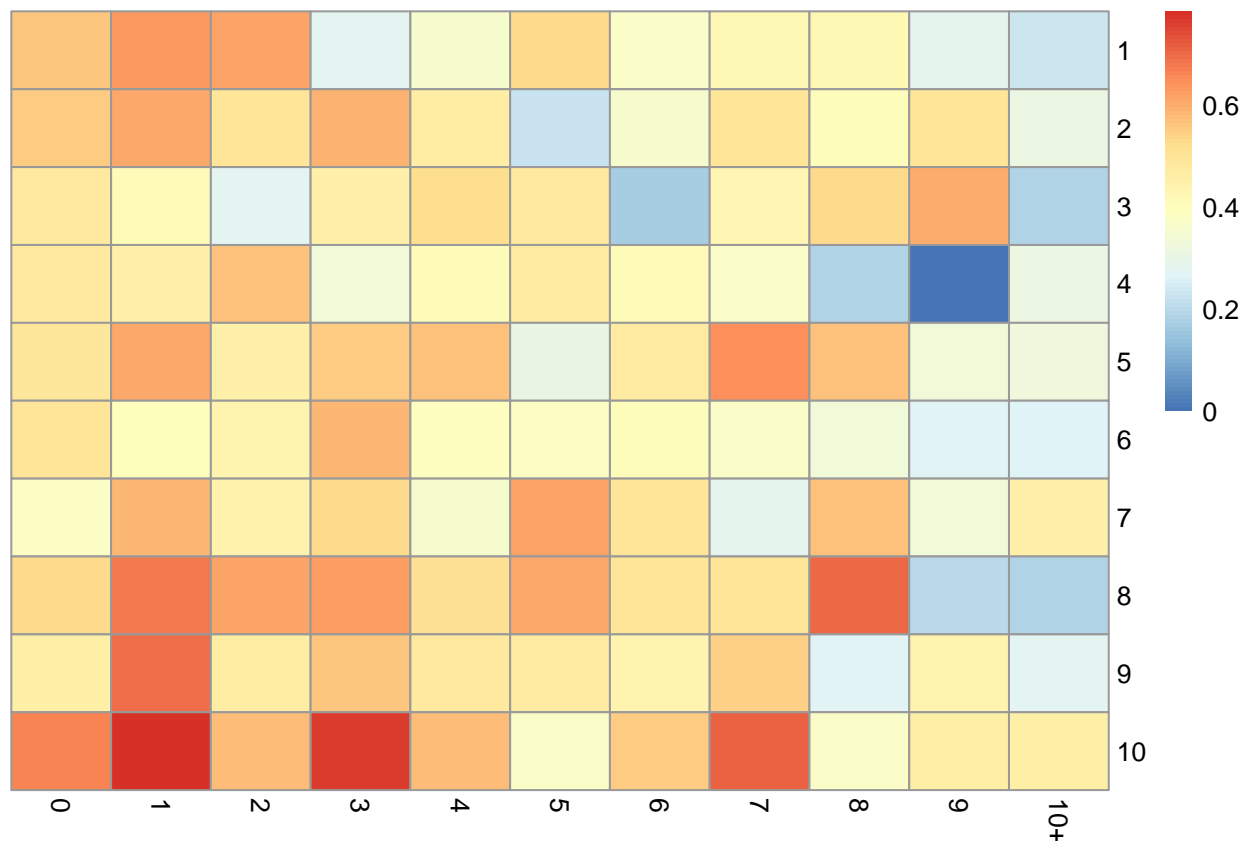
```
tidy(glmmer(tphase=="SP" ~ TCR.mem + ab_ins + (1|donor), data=tcrscores[tcrscores$tphase %in% c("SP", "DP"),]))
```

```
## # A tibble: 4 x 7
##   effect   group term          estimate std.error statistic    p.value
##   <chr>    <chr> <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 fixed   <NA> (Intercept)  -0.0374    0.424    -0.0882  9.30e- 1
## 2 fixed   <NA> TCR.mem        0.216     0.0424     5.08    3.78e- 7
## 3 fixed   <NA> ab_ins       -0.0910    0.0115    -7.94    2.03e-15
## 4 ran_pars donor sd_ (Intercept)   1.02      NA         NA       NA
```

```
tcrscores$TCR.mem_decile = convert_to_quantile(tcrscores$TCR.mem, step=0.1)
tcrscores$ab_ins = as.numeric(as.character(tcrscores$ab_ins))
tcrscores$ab_ins2 = as.character(tcrscores$ab_ins)
tcrscores$ab_ins2[tcrscores$ab_ins>=10] = "10+"
ins.gr = tcrscores[!(is.na(tcrscores$TCR.mem_decile)) & tcrscores$tphase %in% c("SP", "DP"),] %>% group_by(tcrscores$ab_ins2)
```

```
## 'summarise()' has grouped output by 'ab_ins2'. You can override using the
## '.groups' argument.
```

```
tpl = ins.gr[,c("ab_ins2", "TCR.mem_decile", "p")]
tp = data.frame(pivot_wider(tpl, names_from=ab_ins2, values_from=p))
rownames(tp) = tp$TCR.mem_decile
tp = tp[,c(paste("X", seq(0, 9), sep=""), "X10.")]
colnames(tp) = c(as.character(seq(0, 9)), "10+")
rownames(tp) = seq(1, 10)
pheatmap(tp, cluster_rows = FALSE, cluster_cols = FALSE)
```



```
combat_reference = readRDS("data/sref_combat_full_authTplusNKT_20hPCs_tcrfilt0607_nvargenes200_sampTH1..")
md = combat_reference$meta_data
md$UMAP1 = combat_reference$umap$embedding[,1]
md$UMAP2 = combat_reference$umap$embedding[,2]

print(load("data/TH_PAN.A01.v01.20210429.NKT_TCRorauthT_mapped.RData"))
```

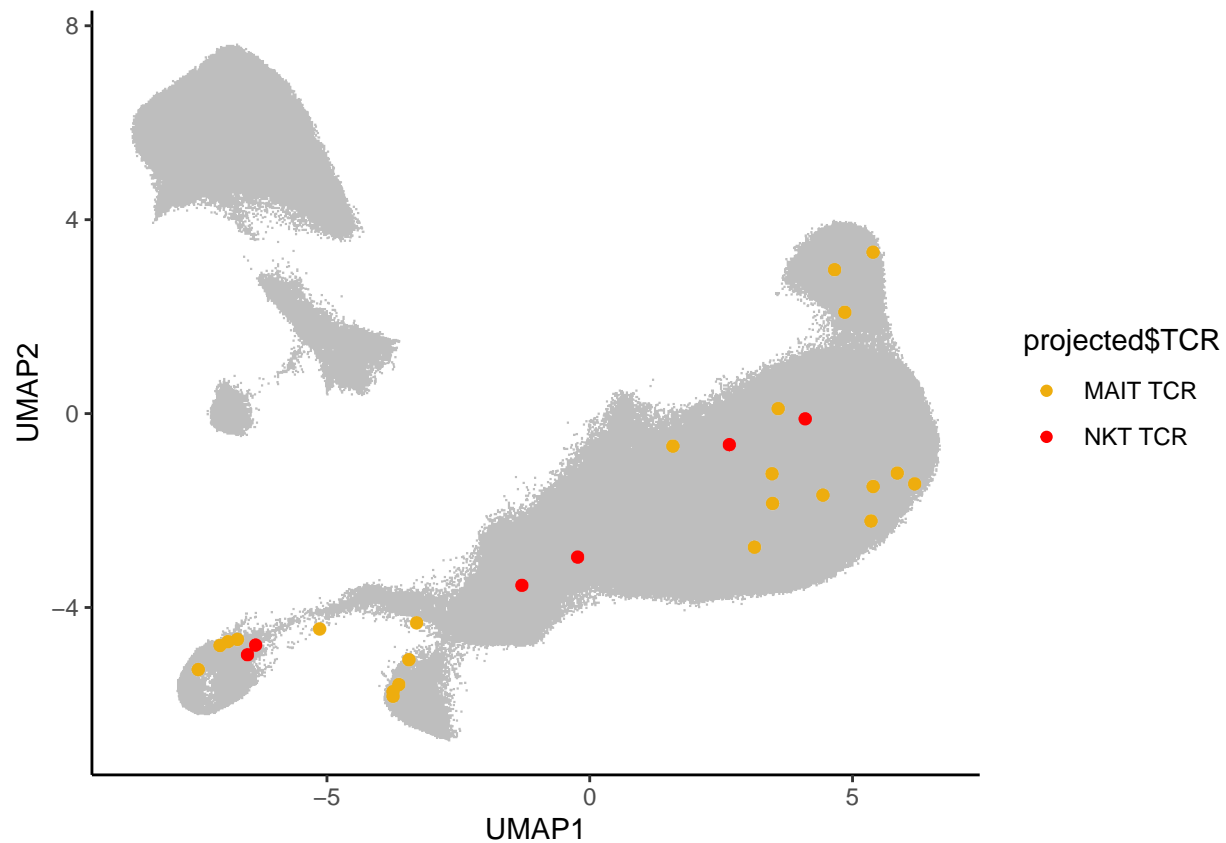
```
## [1] "res" "preds"
```

```
projected = data.frame(res$umap)
projected$index = res$meta_data$index
projected = projected[projected$index %in% metadata$index[metadata$my_abMAIT==TRUE | metadata$my_abNKT==TRUE],]
projected = left_join(projected, metadata[,c("my_abMAIT", "my_abNKT", "index")])
```

```
## Joining with 'by = join_by(index)'
```

```
projected$TCR = ifelse(projected$my_abMAIT==TRUE, "MAIT TCR", "NKT TCR")

g = ggplot()
g = g + geom_point_rast(aes(x=md$UMAP1, md$UMAP2), color="gray", shape=".")
g + geom_point_rast(aes(x=projected$X1, y=projected$X2, color=projected$TCR)) + scale_color_manual(values=c("MAIT TCR", "NKT TCR"))
```



```
tmp$target2 = grepl("CD8", tmp$celltype_annotation)
tmp$X2 = tmp$TCR.CD8
plot_TCRscore_validation_bybin(tmp[tmp$tpphase=="SP",], 2, lcat="thymus", colors = c("dodgerblue3"), xlab="UMAP1", ylab="UMAP2")
```

```
## Joining with 'by = join_by(qtl)'
```

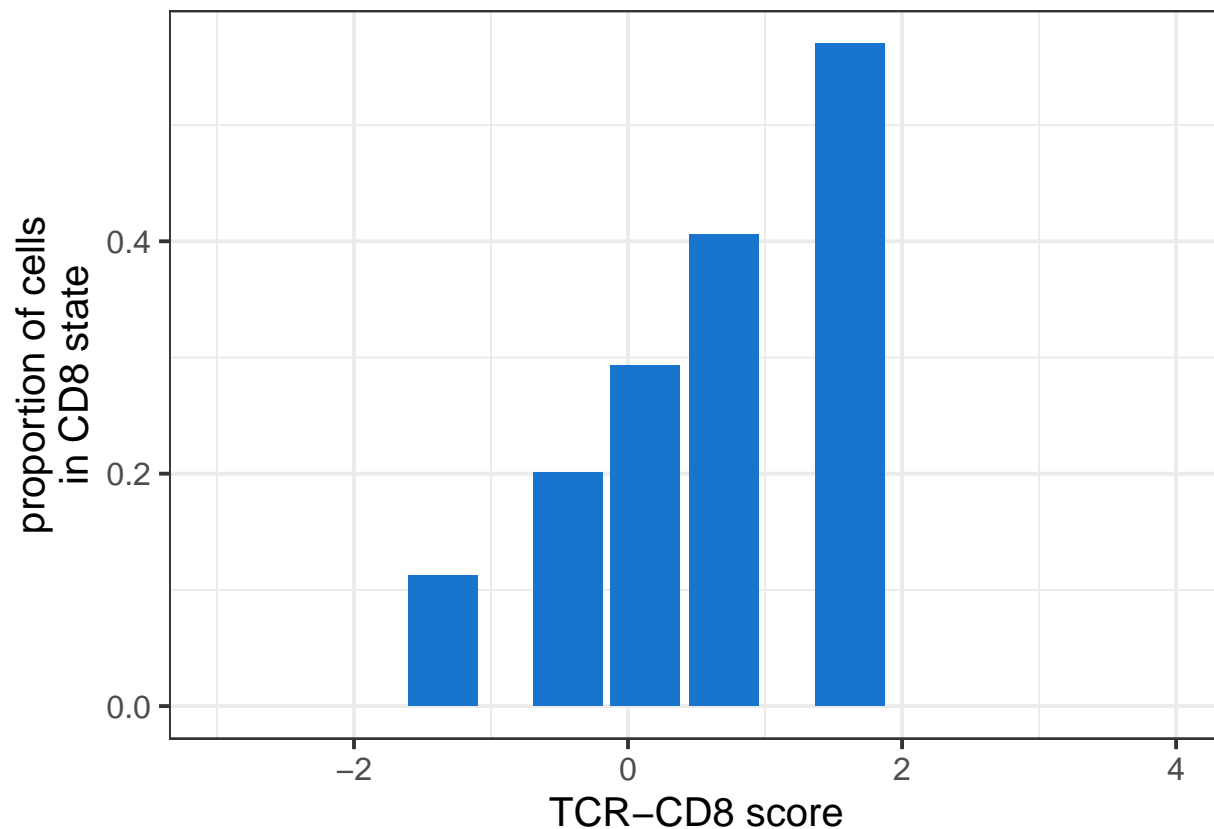
```
## # A tibble: 3 x 7
##   effect   group term          estimate std.error statistic    p.value
##   <chr>   <chr> <chr>          <dbl>    <dbl>    <dbl>    <dbl>
## 1 fixed   <NA> (Intercept)   -1.09    0.145     -7.52 5.37e-14
## 2 fixed   <NA> TCRscore      0.784    0.0375    20.9 4.28e-97
## 3 ran_pars Donor sd_(Intercept) 0.322    NA        NA    NA
```

```
## 'summarise()' has grouped output by 'bin'. You can override using the '.groups'
## argument.
```

```
## # A tibble: 3 x 2
##   mmp   prop.pos
##   <chr>    <dbl>
## 1 maxptl 0.727
## 2 minptl 0.0909
## 3 other 0.315
```

```
## Joining with 'by = join_by(bin)'
```

```
## # A tibble: 5 x 9
##   bin    OR lowOR highOR term          lcat    qtl minbin maxbin
##   <dbl> <dbl> <dbl> <dbl> <chr>      <chr> <dbl> <dbl> <dbl>
## 1 -1.35  0.506 0.386  0.663 bin-1.35160116964468 thymus    20 -3.45 -0.774
## 2  0.127 1.67  1.33  2.09 bin0.126512275429661 thymus    60 -0.155 0.398
## 3  0.704 2.70  2.17  3.35 bin0.703523049094082 thymus    80  0.398 1.03
## 4  1.62  5.11  4.11  6.34 bin1.61775834493033 thymus   100  1.03  3.77
## 5 -0.441 1      1      1      ref          thymus    40 -0.773 -0.155
```



```
tmp$target3 = grepl("TREG", tmp$celltype_annotation)
tmp$X3 = tmp$TCR.reg
plot_TCRscore_validation_bybin(tmp[tmp$tpase=="SP",], 3, lcat="thymus", colors = c("darkmagenta"), xlab="TCR-CD8 score", ylab="proportion of cells in CD8 state")
```

```
## Joining with 'by = join_by(qtl)'
## boundary (singular) fit: see help('isSingular')
```

```
## # A tibble: 3 x 7
##   effect    group term          estimate std.error statistic    p.value
##   <chr>    <chr> <chr>          <dbl>    <dbl>    <dbl>    <dbl>
## 1 fixed    <NA> (Intercept)   -0.986   0.0361   -27.3  1.10e-164
## 2 fixed    <NA> TCRscore      0.139   0.0331    4.19  2.76e- 5
## 3 ran_pars Donor sd_ (Intercept)    0      NA      NA      NA
```

```
## boundary (singular) fit: see help('isSingular')
## 'summarise()' has grouped output by 'bin'. You can override using the '.groups' argument.boundary (s
```



```
## # A tibble: 3 x 2
##   mmp      prop.pos
##   <chr>      <dbl>
## 1 maxptl    0.477
## 2 minptl    0.205
## 3 other     0.280
```

```
## Joining with 'by = join_by(bin)'
```

```
## # A tibble: 5 x 9
##       bin      OR lowOR highOR term          lcat      qtl minbin maxbin
##   <dbl> <dbl> <dbl> <dbl> <chr>      <chr> <dbl> <dbl> <dbl>
## 1 -1.08  0.940  0.759   1.17 bin-1.08031881407862 thymus    20 -2.98 -0.571
## 2  0.268  0.946  0.763   1.17 bin0.267881944207646 thymus    60  0.0121  0.527
## 3  0.822  1.19   0.968   1.47 bin0.822383871247654 thymus    80  0.528   1.17
## 4  1.79   1.28   1.04   1.58 bin1.78697337205181 thymus   100  1.17   3.80
## 5 -0.275  1       1       1   ref          thymus    40 -0.571  0.0119
```

