Figure_4

Kaitlyn Lagattuta

2024-11-19

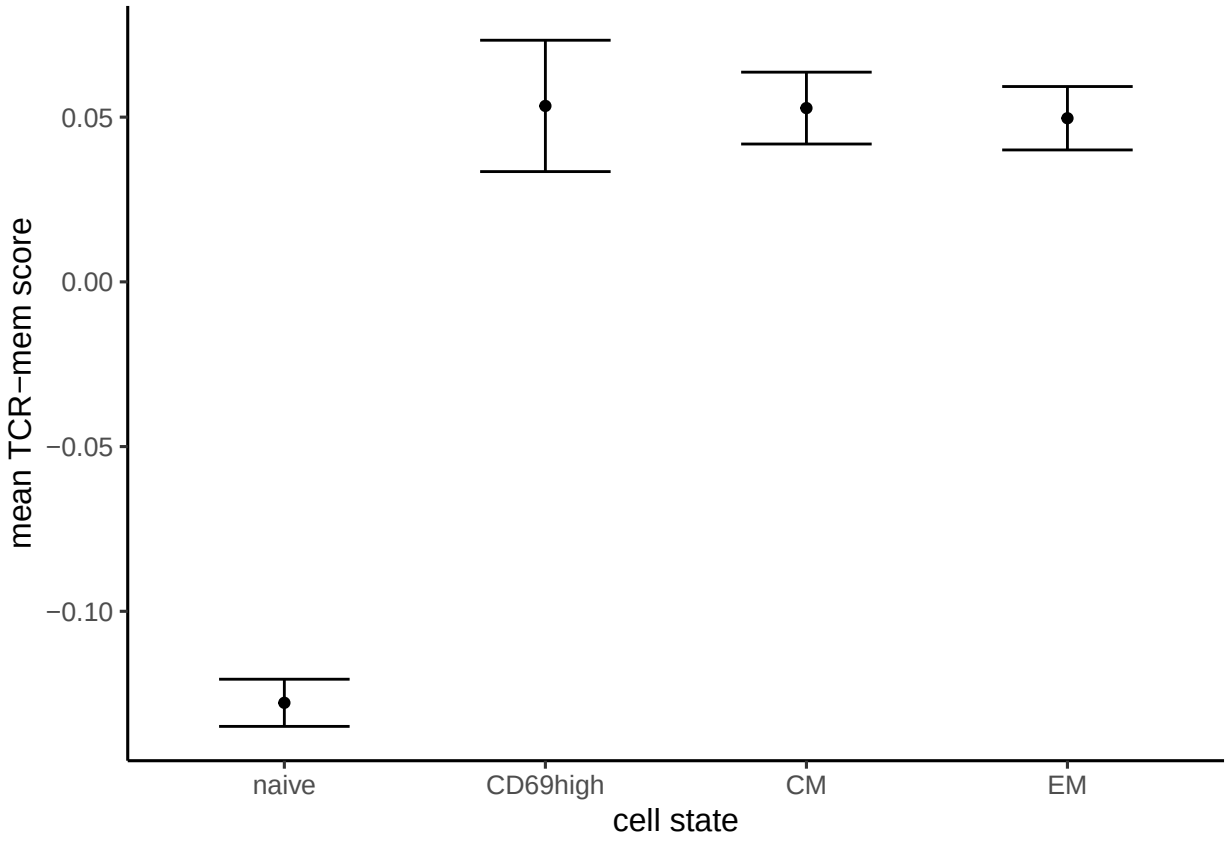
```
suppressPackageStartupMessages(source("utils.R"))
```

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

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

Assess TCR-mem scores in different memory T subsets

```
## Read in results from Symphony reference mapping and knn label transfer
cmem = readRDS("data/combatCMEMlabels_0405.rds")
md = readRDS("data/COMBAT_metadata.rds")
md$mRNA_cluster = paste("A", as.numeric(as.character(md$cl0.5))+1, sep="")

cmem$cell_state_general = gsub("CD4 |CD8 ", "", cmem$state)
cmem$cell_state_general[cmem$X %in% md$X[md$mRNA_cluster %in% c("A7", "A8")]] = "CD69high"

## Join memory T cell state annotations with TCR scores in these data
xtr = data.frame(readRDS("data/CRtrtest_061324/CR_xtrain_scored.rds"))
xtr$cell = rownames(xtr)
cmem = inner_join(cmem, xtr, by=c("X"="cell"))
summ = cmem[cmem$cell_state_general %in% c("naive", "EM", "CM", "CD69high"),] %>% group_by(cell_state_g
summ$SEM = summ$sd_TCRmem/sqrt(summ$n)
summ$cell_state_general = factor(summ$cell_state_general, levels=c("naive", "CD69high", "CM", "EM"))
g = ggplot(summ, aes(cell_state_general, mean_TCRmem))
g = g + geom_point() + geom_errorbar(aes(ymin=mean_TCRmem + qnorm(0.025)*SEM, ymax=mean_TCRmem + qnorm(
g + xlab("cell state") + ylab("mean TCR-mem score")
```



First TCR transduction experiment: Four TCRs recognizing ELA- antigen peptide from MART-1

```
mart = read.csv("data/TCRs_for_ELAG.csv")
mart$patient_id <- NULL
mart$rank <- NULL
mart = mart[,1:7]
colnames(mart) = c("id", "TCRB_vgene", "TCRB_jgene", "TCRB_cdr3aa", "TCRA_vgene", "TCRA_jgene", "TCRA_c
mart.sc = tcrpheno::score_tcrs(mart, 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] 4
## [1] 4
## [1] "adding interactions between adjacent residues..."
## [1] "TCRs featurized!"
## [1] "scoring TCRs..."
## [1] "all done!"
```

```
mart.sc$TCR = c("MART-a", "MART-b", "MART-c", "MART-d")

data = read.csv("data/Bk10Exp42_ELAG_reactivity.csv")
colnames(data)[1:2] = c("Sample", "Percent_Activated")

data$TCR = "MART-a"
data$TCR[grepl("ELAG2", data$Sample)] = "MART-b"
data$TCR[grepl("ELAG3", data$Sample)] = "MART-c"
data$TCR[grepl("ELAG4", data$Sample)] = "MART-d"
data = left_join(data, mart.sc)
```

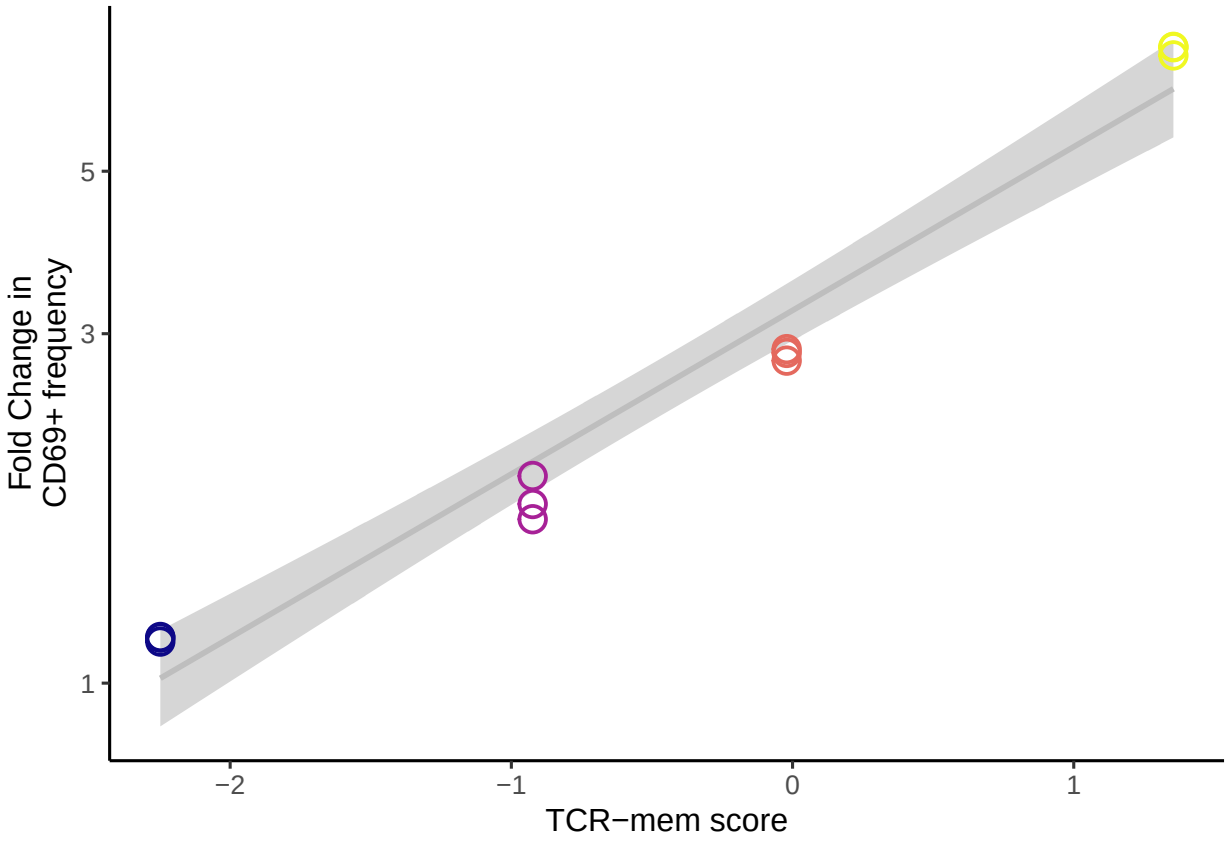
```
## Joining with `by = join_by(TCR)`
```

```
data$WT = TRUE
data$WT[grepl("Het", data$Sample)] = FALSE

data$exposure = ifelse(grepl("MLANA", data$Sample), "ELAG", "background")

data$label = paste(data$TCR, data$exposure, sep=", ")

## Calculate background activation
bg = data[data$WT==TRUE & data$exposure=="background",] %>% group_by(TCR) %>% dplyr::summarise(mean_bg_p

data = left_join(data, bg)
```

```
## Joining with `by = join_by(TCR)`
```

```
data$fold_change = data$Percent_Activated/data$mean_bg_percent_activation

g = ggplot(data[data$exposure=="ELAG" & data$WT==TRUE,], aes(TCR.mem, fold_change, color=TCR.mem))
g = g + geom_smooth(color="gray", method="lm") + scale_y_log10()
g = g + geom_point(size=4, show.legend = FALSE, pch=1, stroke=1) + theme_classic(base_size=12) + scale_
g + xlab("TCR-mem score") + ylab("Fold Change in\nCD69+ frequency")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Second TCR transduction experiment: Four TCRs recognizing NLV- antigenic peptide from pp65 of CMV

```
nlv = data.frame(id =c("WT", "SW1", "SW3", "SP4"),
                      TCRA_cdr3aa = c("CAGPMKTSYDKVIF", "CAGPMITSQDKVIF", "CAGPMLTSQDKVIF", "CAGPNPTTYDKV
                      TCRA_vgene = rep("TRAV35", 4), TCRA_jgene = rep("TRAJ50", 4),
                      TCRB_vgene = rep("TRBV12-4", 4), TCRB_jgene = rep("TRBJ1-2", 4),
                      TCRB_cdr3aa = rep("CASSSANYGYTF", 4))

nlv.sc = tcrpheno::score_tcrs(nlv, chain="ab") ##to minimize confusion
```

```
## [1] "adding CDR1 and CDR2 based on V gene..."
## [1] "identifying amino acids at each position..."
## [1] "converting amino acids into Atchley factors..."
## [1] 4
## [1] 4
## [1] "adding interactions between adjacent residues..."
## [1] "TCRs featurized!"
## [1] "scoring TCRs..."
## [1] "all done!"
```

```r
nlv.sc$TCR = rownames(nlv.sc)

data = read.csv("data/NLV_CD69_052524_results.csv")
data$pct_CD69 = as.numeric(as.character(data$live.cells.Single.Cells.Jurkats.CD69pos...Freq..of.Parent))
data$column = as.numeric(as.character(sapply(data$Sample., function(x) substr(x, 10, nchar(x)-4))))

data$TCR = ""
data$TCR[grepl("1.fcs", data$Sample.) | grepl("6.fcs", data$Sample.)] = "no TCR"
data$TCR[grepl("2.fcs", data$Sample.) | grepl("7.fcs", data$Sample.)] = "WT"
data$TCR[grepl("3.fcs", data$Sample.) | grepl("8.fcs", data$Sample.)] = "SW1"
data$TCR[grepl("4.fcs", data$Sample.) | grepl("9.fcs", data$Sample.)] = "SW3"
data$TCR[grepl("5.fcs", data$Sample.) | grepl("10.fcs", data$Sample.)] = "SP4"

data$APCs = TRUE
data$APCs[grepl("02-Well", data$Sample.) & data$column<6] = FALSE

data$conc = 0
data$conc[data$column>=6 & (grepl("11-D", data$Sample.) | grepl("11-E", data$Sample.) | grepl("11-F", da
data$conc[data$column<6 & (grepl("11-D", data$Sample.) | grepl("11-E", data$Sample.) | grepl("11-F", dat
data$conc[data$column>=6 & (grepl("11-A", data$Sample.) | grepl("11-B", data$Sample.) | grepl("11-C", da
data$conc[data$column<6 & (grepl("11-A", data$Sample.) | grepl("11-B", data$Sample.) | grepl("11-C", dat

data$conc[grepl("02-Well", data$Sample.)] = 0

data = left_join(data, nlv.sc[,c("TCR", "TCR.mem")])
```

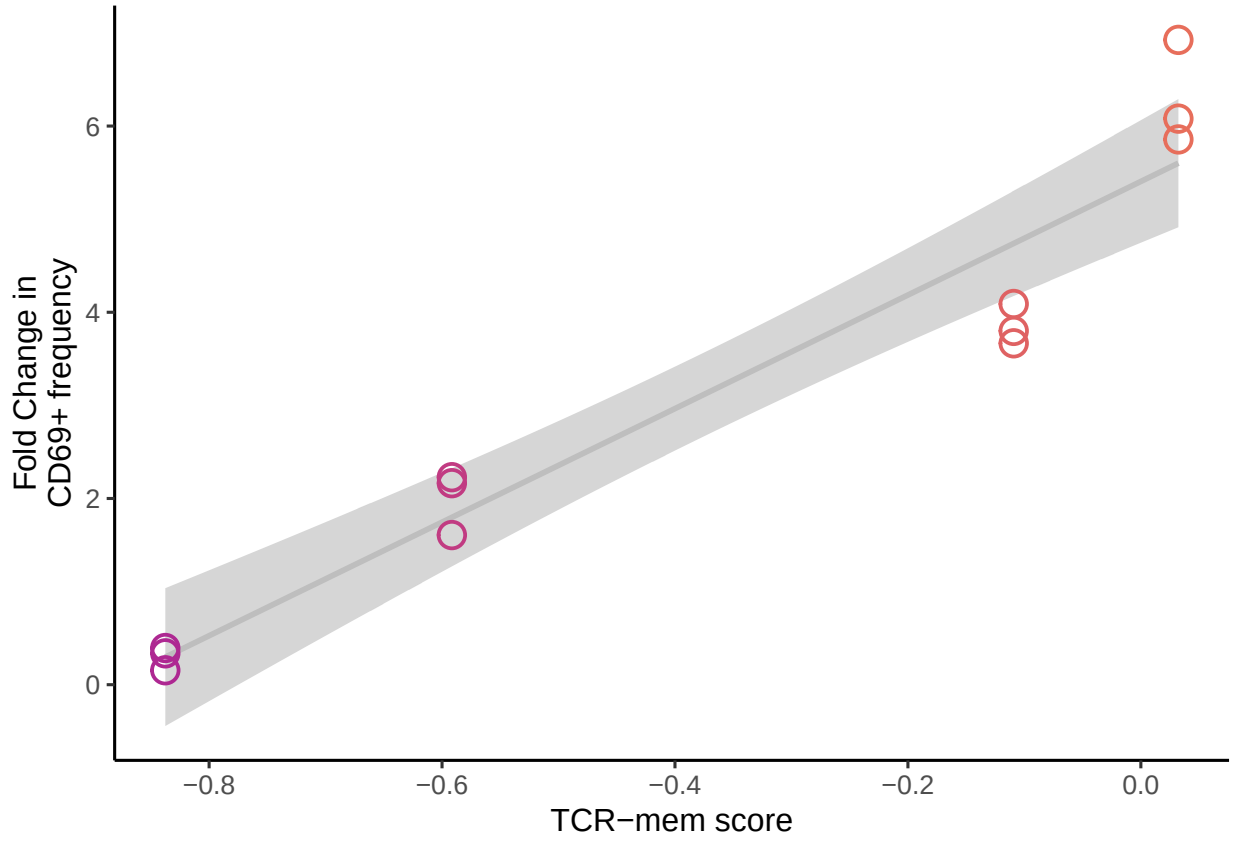
```
## Joining with `by = join_by(TCR)`
```

```r
data$label = data$TCR
data$label[data$conc==0] = paste(data$label[data$conc==0], ", 0% PP65", sep="")
data$label[data$conc==10] = paste(data$label[data$conc==10], ", 10% PP65", sep="")
data$label[data$conc==25] = paste(data$label[data$conc==25], ", 25% PP65", sep="")
data$label[data$conc==50] = paste(data$label[data$conc==50], ", 50% PP65", sep="")
data$label[data$conc==100] = paste(data$label[data$conc==100], ", full PP65", sep="")
data$label[data$APCs==FALSE] = paste(data$TCR[data$APCs==FALSE], ", 0 APCs", sep="")
data$label[grepl("SP4", data$label)] = paste("t", data$label[grepl("SP4", data$label)])

## Calculate background activation
bg = data[data$conc==0 & data$APCs==TRUE,]
bg = bg %>% group_by(TCR) %>% dplyr::summarise(mean_bg = mean(pct_CD69))

data = data[data$APCs==TRUE & data$conc!=0,]

data = left_join(data, bg)
```

```
## Joining with `by = join_by(TCR)`
```

```r
data$FC_CD69_bg = data$pct_CD69/data$mean_bg

g = ggplot(data[data$TCR!="no TCR" & data$conc==50,], aes(TCR.mem, FC_CD69_bg, color=TCR.mem))
g = g + geom_smooth(color="gray", method="lm")
g = g + geom_point(size=4, show.legend = FALSE, pch=1, stroke=1) + theme_classic(base_size=12) + scale_
g + xlab("TCR-mem score") + ylab("Fold Change in\nCD69+ frequency")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

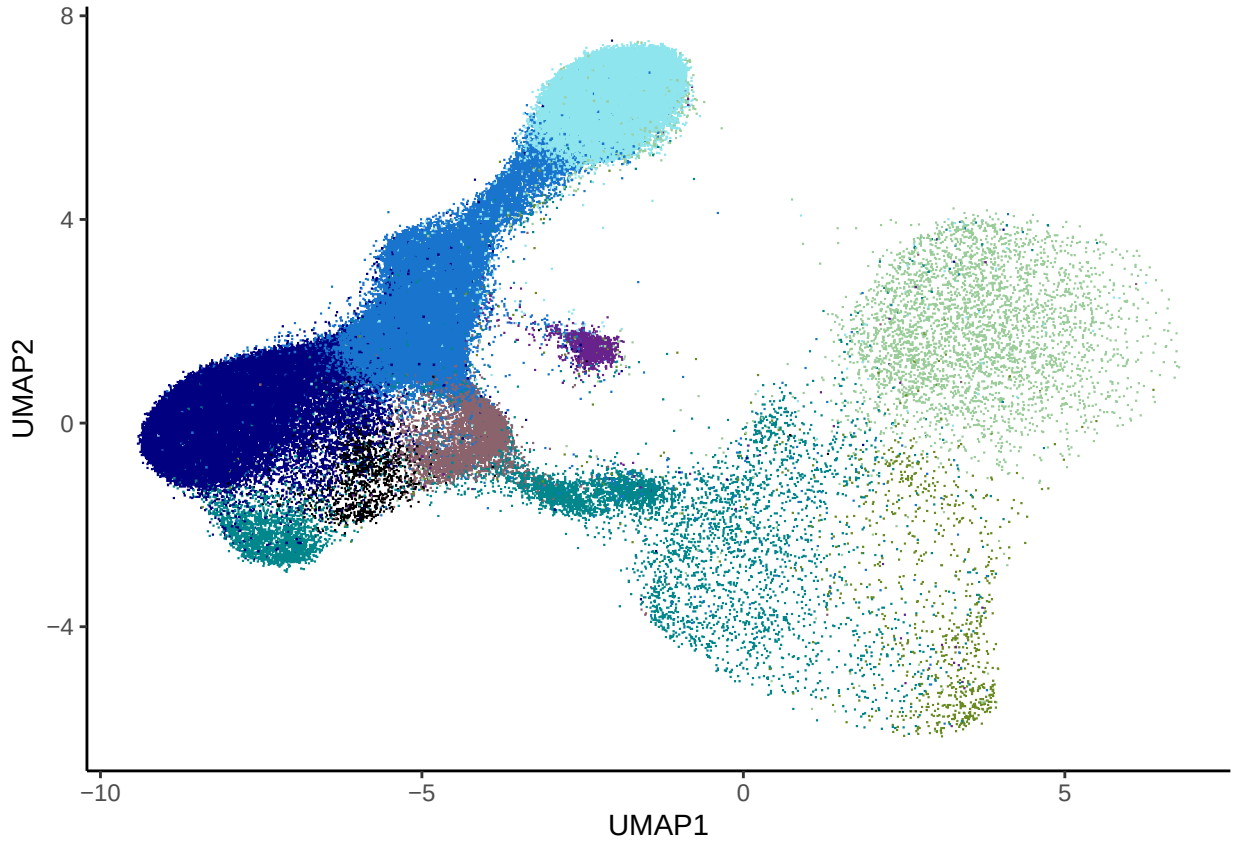



```
data = readRDS("data/Dextramer_dataset.rds")
mapped = readRDS("data/Dataset5_mapped_to_multimodal_reference.rds")
umap = data.frame(mapped$umap)
colnames(umap) = c("UMAP1", "UMAP2")
umap$cell = mapped$meta_data$cell
data = left_join(data, umap)
```

```
## Joining with `by = join_by(cell)`
```

```
g = ggplot(data, aes(UMAP1, UMAP2, color=cmem))
g = g + geom_point_rast(shape=".", size=0.001, show.legend=FALSE) + theme_classic() + scale_color_manual
g + xlab("UMAP1") + ylab("UMAP2") + labs(color="cell state")
```

```
## Warning: Removed 4236 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
## Deduplicate expanded clones
set.seed(27)
datadd = data[!(duplicated(data$clone)),]

## Filter out cells that are missing a UMAP projection results because they did not pass QC
datadd = datadd[!(is.na(datadd$UMAP1)),]

## Remove the innate-like (PLZF-high cluster); cells that recognize MHC class I Dextramer should be CD8
datadd = datadd[datadd$knn_mRNA_cluster!="A9",]

## Read results from tcrpheno::score_tcrs() on these data
df = readRDS("data/10xG_LR062424scores.rds")
df$cell = rownames(df)
datadd = left_join(datadd, df)
```

```
## Joining with `by = join_by(cell)`
```

```
## Call antigen specificities based on results from negative binomial regression
print(load("data/normalized_Dextramer_values.RData"))
```

```
## [1] "R"          "gmm_plots" "gmm_means" "gmm_vars"
```

```
thresh_file = read.csv("data/Dextramer_thresholds.csv")
thresholds = thresh_file$threshold
```

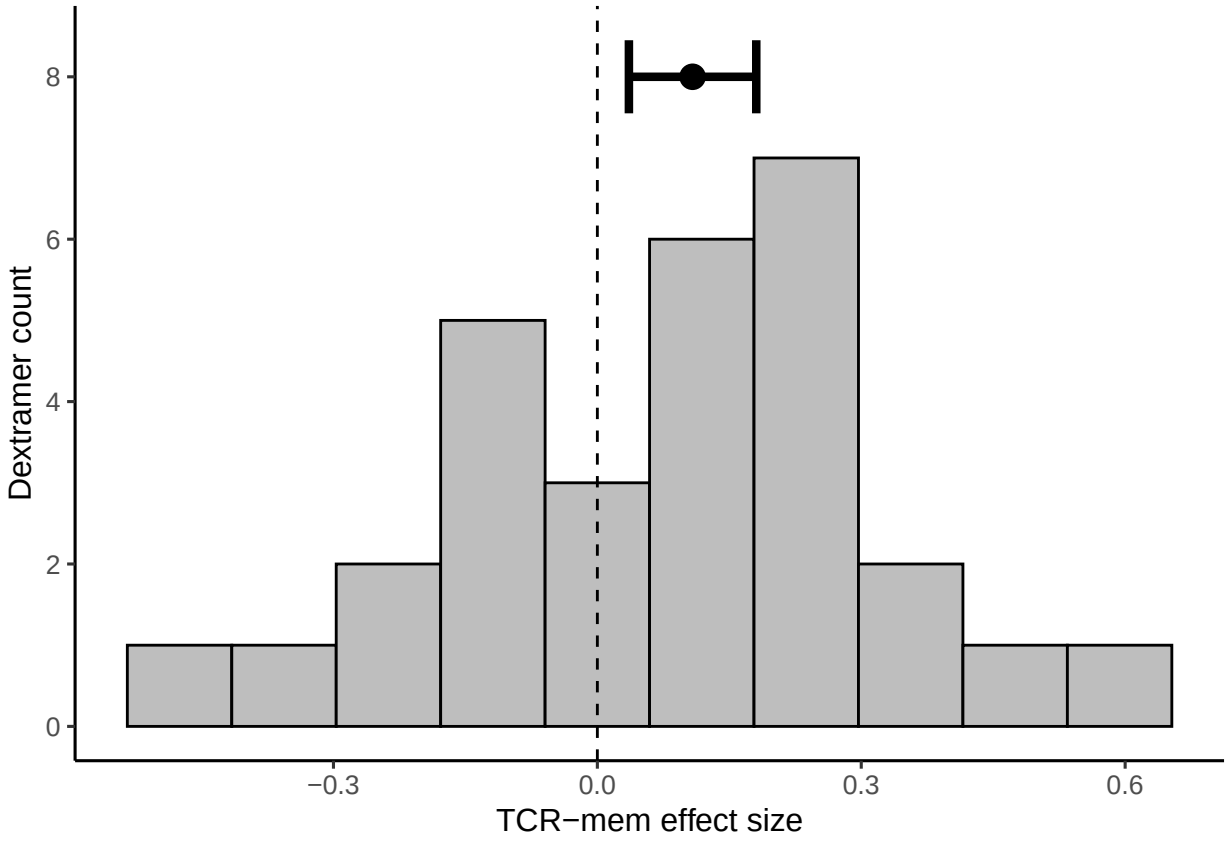
```r
names(thresholds) = paste(thresh_file$antigen, thresh_file$donor)

call = call_antigens(R=R, thresholds=thresholds, data=data)
```

```
## [1] "number of calls per cell:"
##
##      0      1      2      3      4      5      6      7      8      9     10
## 136651  34775  11773   4088   1405    493    197     78     36     14      1
##     11
##      1
```

```r
betas = testTCRscore_perantigen(call, datadd, stain_cov=TRUE, remove_CD4s = TRUE) ##cells that recogniz
```

```
## [1] "analyzing 10860 cells"
## [1] "analyzing 32 antigens prior to N/M thresh"
## [1] "29 antigens retained"
## [1] "positive beta?"
##
## FALSE  TRUE
##    10    19
## [1] "(nominally significant):"
##
## TRUE
##    5
```

```r
meta_analysis = rma(betas$b, sei=betas$se, measure="OR", method="ML")

g = ggplot()
g = g + geom_histogram(aes(x=betas$b), color="black", fill="gray", bins=10) + geom_vline(xintercept=0,
g + geom_point(aes(x=meta_analysis$beta[1], y=8), size=4) + geom_errorbar(aes(x=meta_analysis$beta[1],
```




```
print(meta_analysis)
```

```
##
## Random-Effects Model (k = 29; tau^2 estimator: ML)
##
## tau^2 (estimated amount of total heterogeneity): 0.0107 (SE = 0.0094)
## tau (square root of estimated tau^2 value):      0.1036
## I^2 (total heterogeneity / total variability):   29.15%
## H^2 (total variability / sampling variability):  1.41
##
## Test for Heterogeneity:
## Q(df = 28) = 39.6131, p-val = 0.0716
##
## Model Results:
##
## estimate      se    zval    pval   ci.lb   ci.ub
##   0.1083  0.0369  2.9316  0.0034  0.0359  0.1807  **
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```