

# Figure\_3

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
```

## Testing TCR scores within each stratum of individuals (Dataset 1 and Dataset 2)

```
md = read.csv("data/COMBATrenmerged_cellstatetargets_fix0414_fixnames0606.csv")

cmd = read.csv("data/combat_metadata_incNKT.csv")
cmd$DiseaseClassification = as.character(cmd$DiseaseClassification)
cmd$DiseaseClassification[is.na(cmd$DiseaseClassification)] = "Healthy"
md = left_join(md, cmd[,c("X", "DiseaseClassification")], by=c("cell"="X"))

mapped = readRDS("data/ren_mappedtocombat_full_authTplusNKT__500g_20hPCs_tcrfilt0607_nvargenes200_theta")
rmd = mapped$meta_data
md = left_join(md, rmd[,c("SARS.CoV.2..based.on.PCR.", "cellName")], by=c("cell"="cellName"))

md$dataset = ""
md$dataset[md$cell %in% cmd$X] = "COMBAT"
md$dataset[md$cell %in% rmd$cellName] = "Ren et al."

md$condition = "none"
md$condition[grepl("COVID-19", md$DiseaseClassification) | md$SARS.CoV.2..based.on.PCR.=="positive"] = "COVID-19"
md$condition[md$SARS.CoV.2..based.on.PCR.=="negative"] = "COVID-"
md$condition[grepl("Sepsis", md$DiseaseClassification)] = "Sepsis"
md$condition[grepl("Influenza", md$DiseaseClassification)] = "Influenza"

md$strata = paste(md$condition, md$dataset, sep="; ")
md$strata[md$group=="test"] = "test"

xtr = readRDS("data/CRtrtest_061324/CR_xtrain_scored.rds")
xte = readRDS("data/CRtrtest_061324/CR_xtest_scored.rds")
xte = scale_variables(xte, tcrpheno::ABscore_mns, tcrpheno::ABscore_sds)
scored = rbind(xtr, xte)
scored$cell = rownames(scored)

md = left_join(md, scored)
```

```

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

score = c("TCR.innate", "TCR.CD8", "TCR.reg", "TCR.mem")
lineage = c("all", "CD4", "CD8")
strata = c(unique(md$strata), "all; both", "all; Ren et al.", "all; COMBAT")

## Setting up a dataframe to hold mixed effects logistic regression results, testing the effect size of
results = data.frame(expand.grid(score, lineage, strata))
colnames(results) = c("score", "lineage", "strata")

## TCR-innate and TCR-CD8 scores should not be applied within CD4T or within CD8T lineages
results = results[!(results$score=="TCR.innate" & results$lineage!="all"),]
results = results[!(results$score=="TCR.CD8" & results$lineage!="all"),]
results$beta = 0
results$se = 0
results$p = 0
for (i in 1:nrow(results)){
  if (!(grepl("all", results$strata[i]))){
    sub = md[md$strata==results$strata[i],]
  } else if (results$strata[i]=="all; COMBAT") {
    sub = md[md$dataset=="COMBAT" & md$strata!="test",]
  } else if (results$strata[i]=="all; Ren et al.") {
    sub = md[md$dataset!="COMBAT" & md$strata!="test",]
  } else {
    sub = md[md$strata!="test",]
  }
  if (results$lineage[i]=="CD4"){
    sub = sub[sub$target2==0,]
  }
  if (results$lineage[i]=="CD8") {
    sub = sub[sub$target2==1,]
  }
  if (results$score[i]=="TCR.innate"){
    sub$score = sub$TCR.innate
    sub$target = sub$target1
  } else if (results$score[i]=="TCR.CD8") {
    sub$score = sub$TCR.CD8
    sub$target = sub$target2
  } else if (results$score[i]=="TCR.reg"){
    sub$score = sub$TCR.reg
    sub$target = sub$target3
  } else {
    sub$score = sub$TCR.mem
    sub$target = sub$target4
  }

  ##Mixed effects logisitic regression to test the TCRscore in this strata
  stats = tidy(glmmer(target ~ score + (1|donor), data=sub, family="binomial"))
  results$beta[i] = stats$estimate[2]
  results$se[i] = stats$std.error[2]
  results$p[i] = stats$p.value[2]
}

```

```
sts = unique(as.character(results$strata))
results$strata = factor(results$strata, levels=c(sts[sts!="test"], "test"))
results = results[order(results$score, results$strata),]
```

```
results$strata = as.character(results$strata)
results$plot = paste(results$lineage, results$score)
plots = unique(results$score)
```

```
##Meta-analysis across clinical strata for each TCR score
```

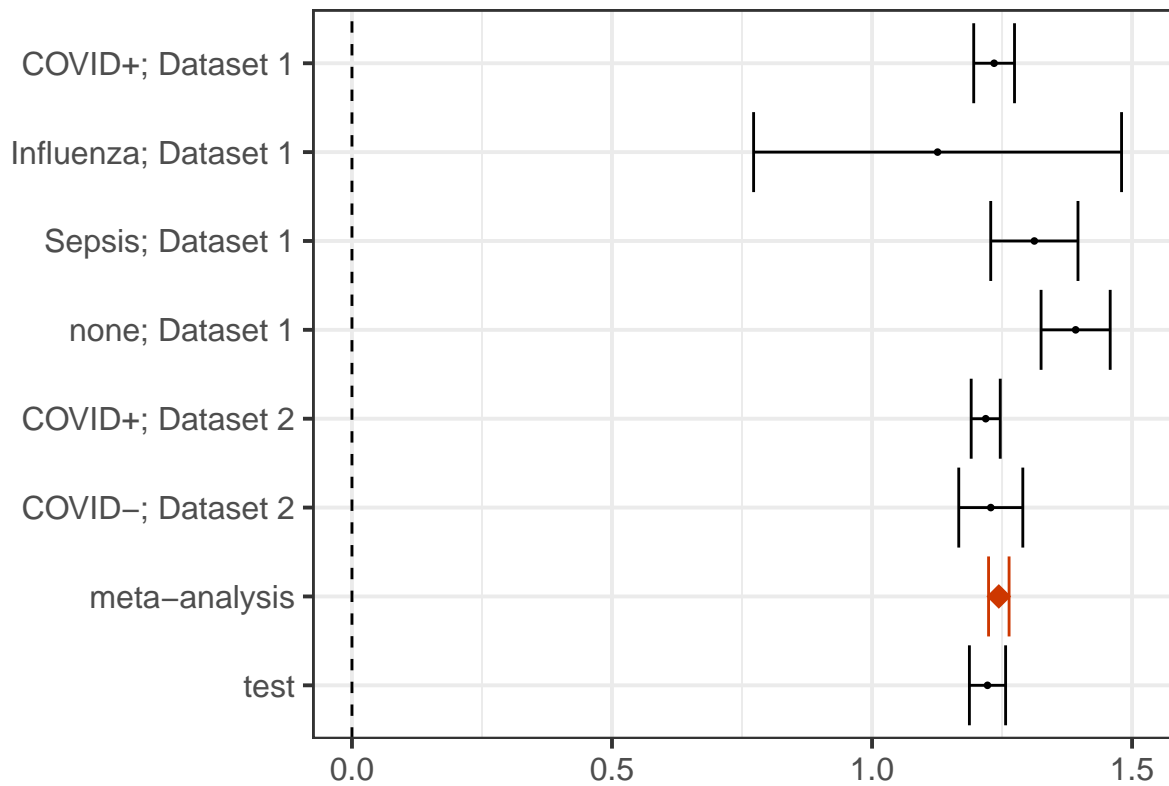
```
for (i in 1:length(plots)){
  sub = results[results$score==plots[i] & results$strata!="test" & results$lineage=="all" & !(grepl("all", results$lineage)),]
  ma = data.frame(sub[1,])
  ma$strata[1] = "meta-analysis"
  ma$beta[1] = get_metabeta(sub$beta, sub$se)
  ma$se[1] = get_metase(sub$se)
  ma$p[1] = get_metap(sub$beta, sub$se)
  ma$p[1] = ifelse(ma$beta>0, ma$p[1]/2, 1-(ma$p[1]/2))
  results = rbind(results, ma)
}
results$type = results$strata=="meta-analysis"
```

```
df = results[!(grepl("all", results$strata)) & results$lineage=="all",]
df$strata = gsub("COMBAT", "Dataset 1", df$strata)
df$strata = gsub("Ren et al.", "Dataset 2", df$strata)
df$strata = factor(df$strata, levels=c("test", "meta-analysis", "COVID-; Dataset 2", "COVID+; Dataset 2"))

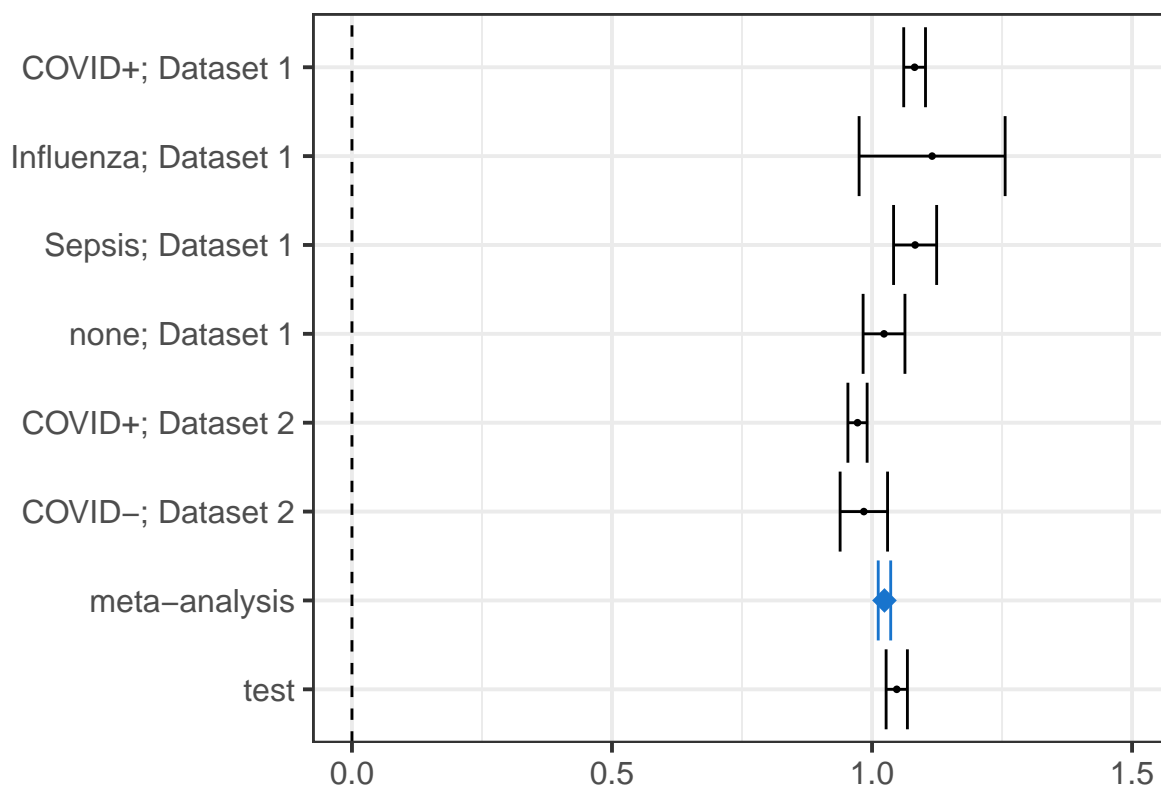
g = ggplot(df[df$score=="TCR.innate",], aes(beta, strata, color=strata=="meta-analysis", shape=strata=="test"))
g = g + scale_shape_manual(values=c(16, 18)) + scale_size_manual(values=c(1,4)) + scale_color_manual(values=c("black", "red", "blue", "green"))
g = g + geom_vline(xintercept=0, linetype="dashed") + geom_point(show.legend=FALSE) + geom_errorbar(aes(xintercept=0, yintercept=0, ylower=-1, yupper=1))
```

```
## 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.
```

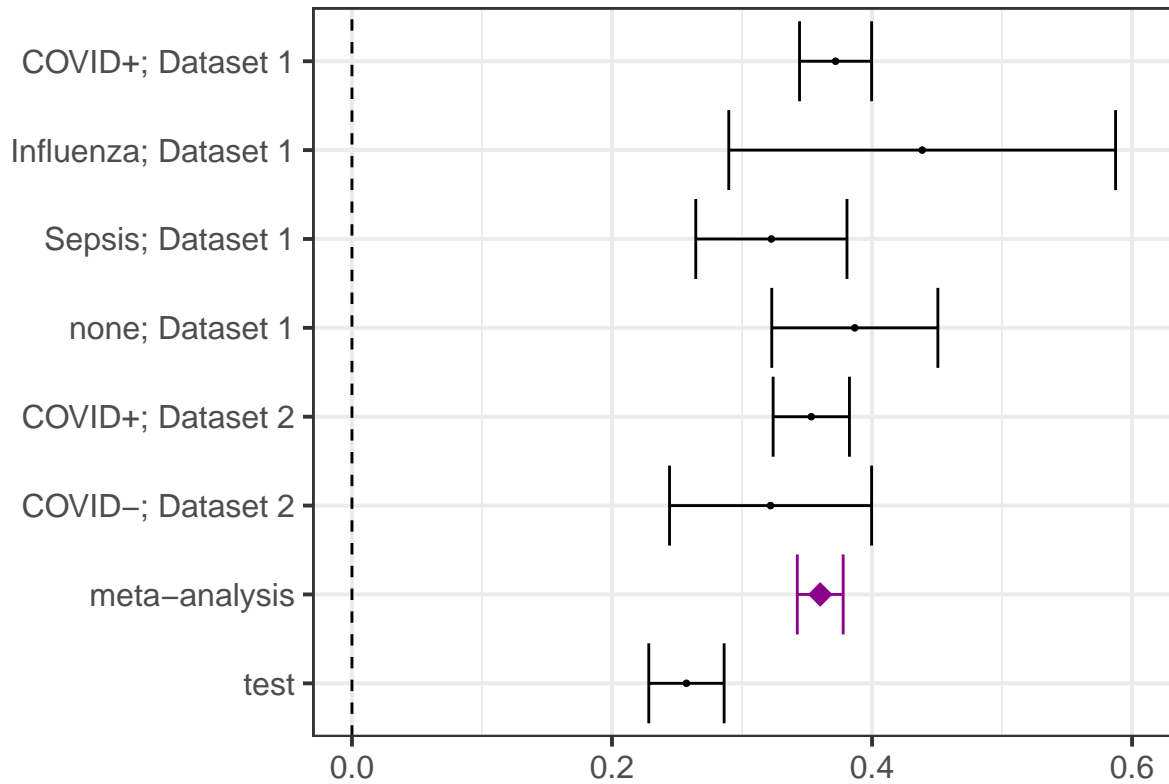
```
g + theme_bw(base_size=15) + xlab("") + ylab("") + xlim(c(0,1.5))
```



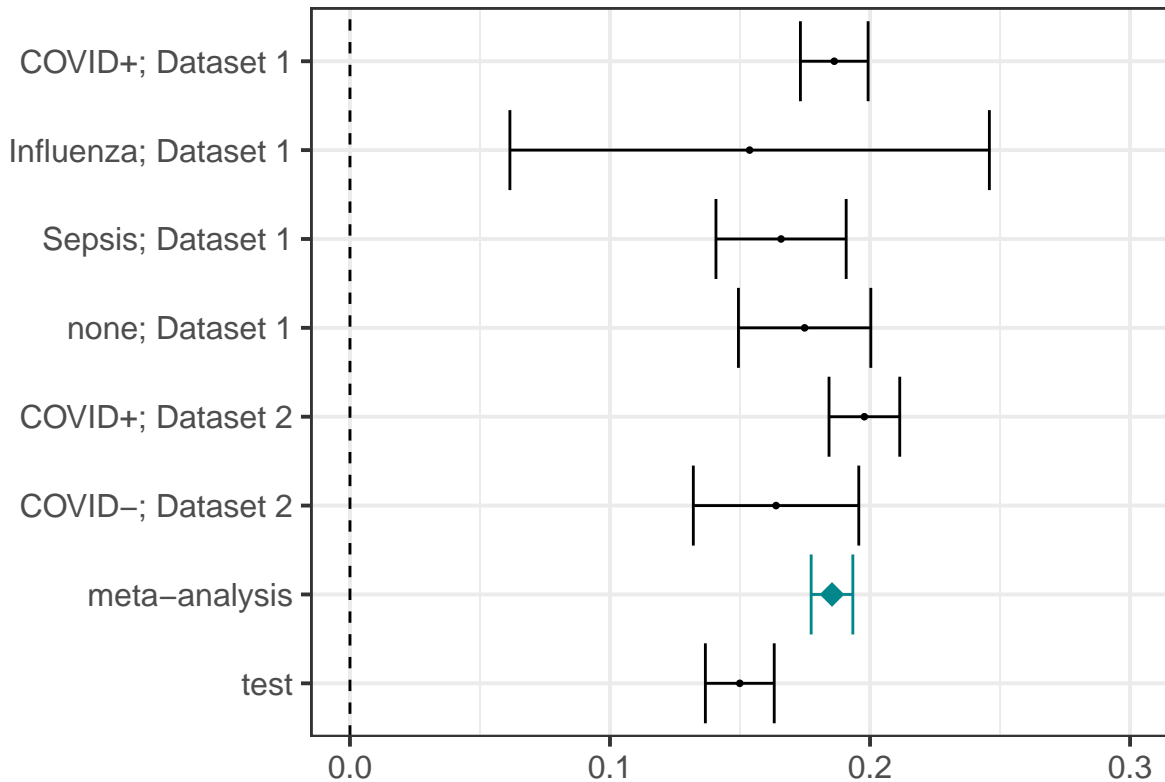
```
g = ggplot(df[df$score=="TCR.CD8",], aes(beta, strata, color=strata=="meta-analysis", shape=strata=="meta-analysis"))
g = g + scale_shape_manual(values=c(16, 18)) + scale_size_manual(values=c(1,4)) + scale_color_manual(values=c("black", "red"))
g = g + geom_vline(xintercept=0, linetype="dashed") + geom_point(show.legend=FALSE) + geom_errorbar(aes(ymin=beta-1.96*se, ymax=beta+1.96*se))
g + theme_bw(base_size=15) + xlab("") + ylab("") + xlim(c(0,1.5))
```



```
g = ggplot(df[df$score=="TCR.reg",], aes(beta, strata, color=strata=="meta-analysis", shape=strata=="meta-analysis"))
g = g + scale_shape_manual(values=c(16, 18)) + scale_size_manual(values=c(1,4)) + scale_color_manual(values=c("black", "blue"))
g = g + geom_vline(xintercept=0, linetype="dashed") + geom_point(show.legend=FALSE) + geom_errorbar(aes(width=2))
g + theme_bw(base_size=15) + xlab("") + ylab("") + xlim(c(0,0.6))
```



```
g = ggplot(df[df$score=="TCR.mem",], aes(beta, strata, color=strata=="meta-analysis", shape=strata=="meta-analysis"))
g = g + scale_shape_manual(values=c(16, 18)) + scale_size_manual(values=c(1,4)) + scale_color_manual(values=c("black", "purple"))
g = g + geom_vline(xintercept=0, linetype="dashed") + geom_point(show.legend=FALSE) + geom_errorbar(aes(width=0.05))
g + theme_bw(base_size=15) + xlab("") + ylab("") + xlim(c(0,0.3))
```



### Testing TCR scores in an external dataset (Dataset 3)

```
## Read in Stephenson et al. TCRs (expanded clones de-duplicated, and scored by tcrpheno::score_tcrs())
tcr_scores = readRDS("data/stephenson_LR062424scores_nocex.rds")

## Annotating metadata with target T cell states, based on Symphony reference mapping
md = read.csv("data/stephenson_metadata.csv")

c11 = readRDS("data/stephenson_QCnoLPS_Tselect0326_mappedto_COMBATfull_wcl0.5_5kpreds.rds")
md = md[md$covid_index %in% c11$covid_index,] ##pass RNAqc
md$target1 = md$covid_index %in% c11$covid_index[c11$pred=="8"]
md$target3 = md$covid_index %in% c11$covid_index[c11$pred=="5"]
c12 = readRDS("data/stephenson_QCnoLPS_Tselect0326_mappedto_COMBATfull_w5kpreds_cl2.0.rds")
md$target3[md$covid_index %in% c12$covid_index[c12$c12.0=="15"]] = TRUE
s.cmem = readRDS("data/stephenson_QCnoLPS_Tselect0326_mappedto_COMBATcmem0405_w5kpreds.rds")
md$target2 = md$covid_index %in% s.cmem$covid_index[grepl("CD8", s.cmem$pred)]
md$target2[md$target1==TRUE] = NA
md$target2[md$covid_index %in% s.cmem$covid_index[s.cmem$pred=="DN"]] = NA
md$target4 = md$covid_index %in% s.cmem$covid_index[!(grepl("naive", s.cmem$pred))]
md$target3[md$target1==TRUE] = NA
md$target4[md$target1==TRUE] = NA

md$Donor = as.character(md$patient_id)
md$cell = as.character(md$covid_index)
```

```

tcr_scores$cell = rownames(tcr_scores)
steph = left_join(tcr_scores, md[,c("Donor", "target1", "target2", "target3", "target4", "cell")])

```

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

```
steph$lcat = "blood"
```

```

steph$X1 = staph$TCR.innate
steph$X2 = staph$TCR.CD8
steph$X3 = staph$TCR.reg
steph$X4 = staph$TCR.mem

```

```
plot_TCRscore_validation_bybin(staph, 1, lcat="blood", colors = c("orangered3"), xlab = "TCR-innate 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)	-5.07	0.0961	-52.8	0
## 2	fixed	<NA>	TCRscore	1.18	0.0152	77.9	0
## 3	ran_pars	Donor	sd_ (Intercept)	0.849	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	0.911
## 2	minptl	0.00659
## 3	other	0.0152

```
## # 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)	-5.55	0.168	-33.1	1.27e-239
## 2	fixed	<NA>	mmmaxbin	3.36	0.121	27.7	4.11e-169
## 3	ran_pars	Donor	sd_ (Intercept)	1.13	NA	NA	NA

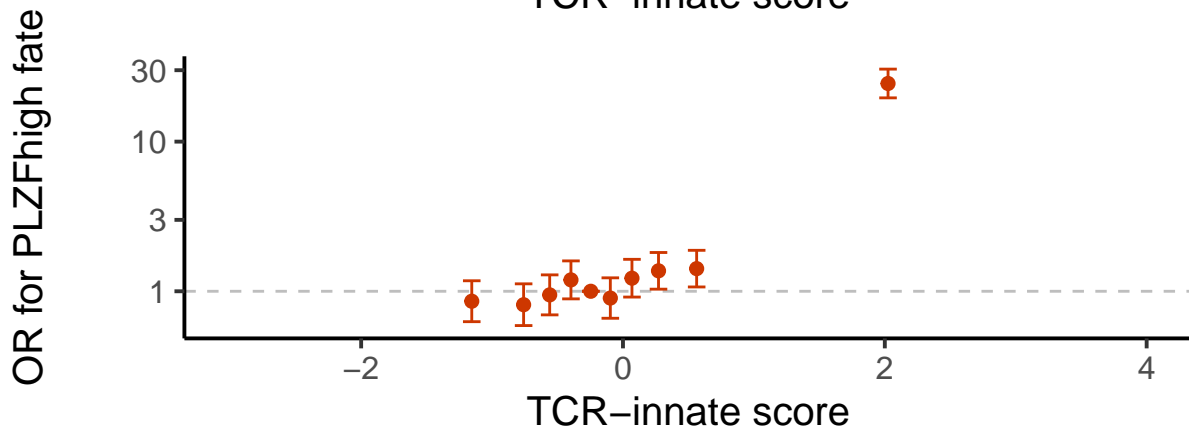
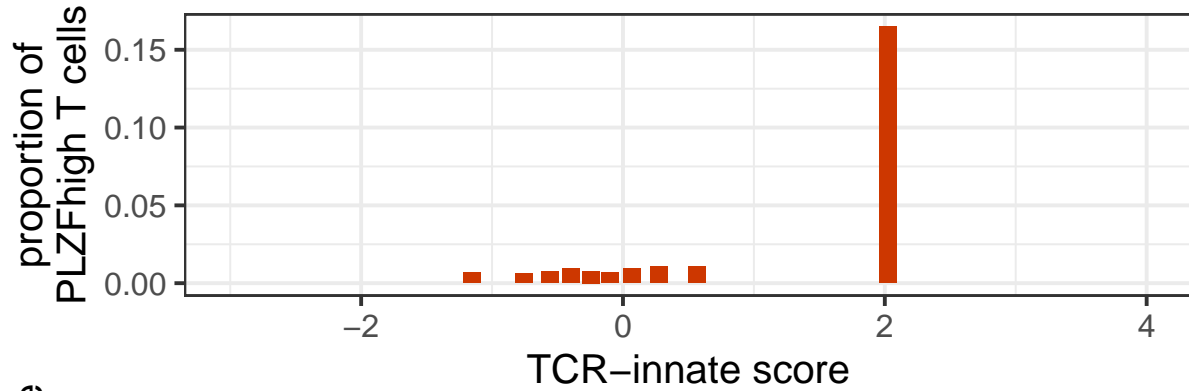
```
## Joining with 'by = join_by(bin)'
```

```
## # A tibble: 10 x 9
```

##	bin	OR	lowOR	highOR	term	lcat	ctl	minbin	maxbin
##	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
## 1	-1.16	0.858	0.626	1.18	bin-1.15513292286071	blood	10	-2.92	-0.888
## 2	-0.759	0.813	0.590	1.12	bin-0.7590204905984~	blood	20	-0.888	-0.650
## 3	-0.560	0.946	0.695	1.29	bin-0.5598735424094~	blood	30	-0.650	-0.475
## 4	-0.398	1.19	0.889	1.60	bin-0.3975065579063~	blood	40	-0.475	-0.321
## 5	-0.0967	0.901	0.660	1.23	bin-0.0967000216896~	blood	60	-0.172	-0.0178
## 6	0.0686	1.22	0.915	1.63	bin0.06856301347310~	blood	70	-0.0178	0.160



```
## 7 0.271 1.37 1.03 1.82 bin0.271368282064746 blood 80 0.160 0.392
## 8 0.563 1.42 1.07 1.88 bin0.562755996271955 blood 90 0.392 0.784
## 9 2.02 24.5 19.6 30.5 bin2.0245649554345 blood 100 0.784 7.90
## 10 -0.246 1 1 1 ref blood 50 -0.321 -0.172
```



```
plot_TCRscore_validation_bybin(step, 2, lcat="blood", colors = c("dodgerblue3"), xlab = "TCR-CD8 score")
```

```
## 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)    -1.56    0.0610     -25.5 1.37e-143
## 2 fixed   <NA> TCRscore       1.04    0.00978     107. 0
## 3 ran_pars Donor sd_ (Intercept)  0.608    NA         NA    NA
```

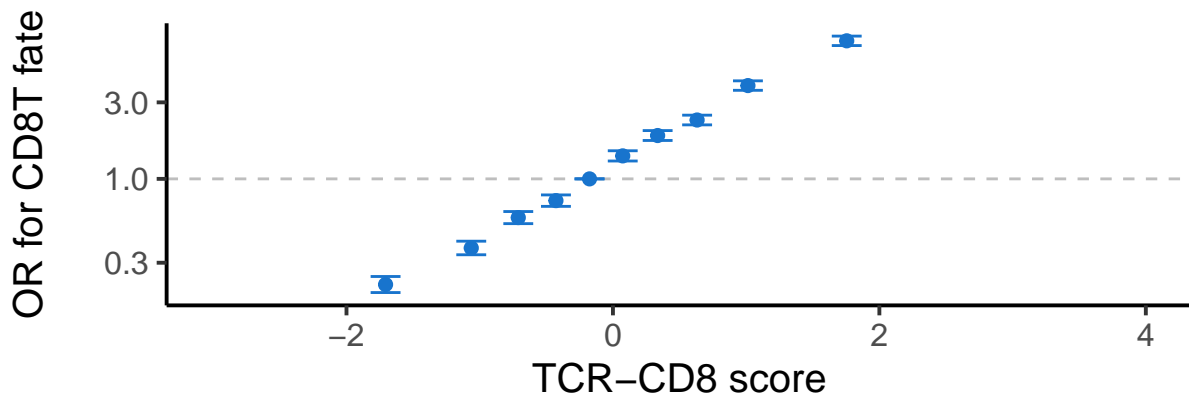
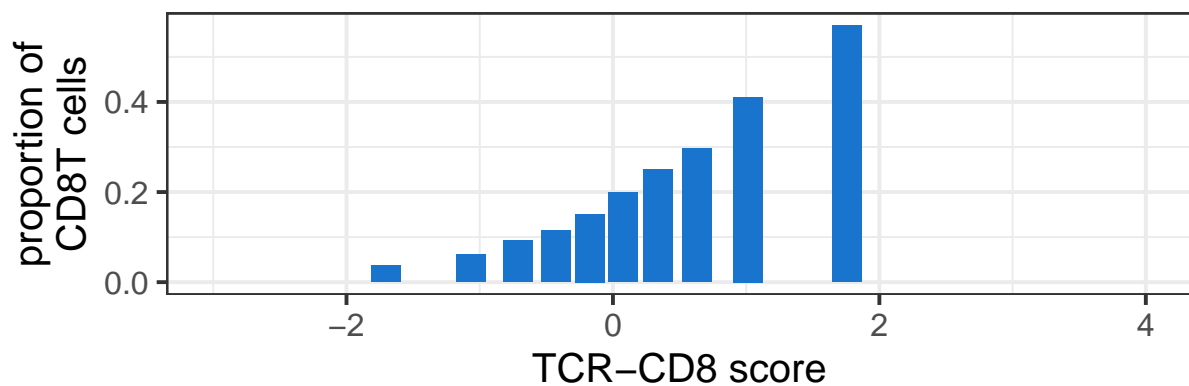
```
## 'summarise()' has grouped output by 'bin'. You can override using the '.groups'
## argument.
```

```
## # A tibble: 3 x 2
##   mmp   prop.pos
##   <chr>   <dbl>
## 1 maxptl 0.716
## 2 minptl 0.0271
```

```
## 3 other      0.215
## # 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)    -3.27    0.0737    -44.4     0
## 2 fixed   <NA> mmmmaxbin      3.51    0.0562     62.4     0
## 3 ran_pars Donor sd_ (Intercept)    0.511    NA         NA        NA
```

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

```
## # A tibble: 10 x 9
##       bin    OR lowOR highOR term          lcat    qtl minbin maxbin
##       <dbl> <dbl> <dbl> <dbl> <chr>      <chr> <dbl> <dbl> <dbl>
## 1 -1.71    0.219 0.196 0.246 bin-1.70661959635454 blood    10 -3.77 -1.29
## 2 -1.06    0.371 0.336 0.409 bin-1.0634290969332 blood    20 -1.29 -0.869
## 3 -0.710   0.574 0.526 0.626 bin-0.710491346132192 blood    30 -0.869 -0.564
## 4 -0.428   0.732 0.674 0.794 bin-0.427761436691705 blood    40 -0.563 -0.297
## 5 0.0735   1.39 1.29 1.50 bin0.0735008905460693 blood    60 -0.0533 0.202
## 6 0.336    1.86 1.74 2.00 bin0.335541333322811 blood    70 0.202 0.475
## 7 0.632    2.33 2.17 2.50 bin0.631904416596891 blood    80 0.475 0.801
## 8 1.01     3.82 3.57 4.09 bin1.01335080314039 blood    90 0.801 1.27
## 9 1.75     7.26 6.78 7.77 bin1.75494605243512 blood   100 1.27 4.00
## 10 -0.176 1      1      1      ref          blood    50 -0.297 -0.0533
```



```
plot_TCRscore_validation_bybin(steph, 3, lcat="blood", colors = c("darkmagenta"), xlab = "TCR-reg score")
```

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

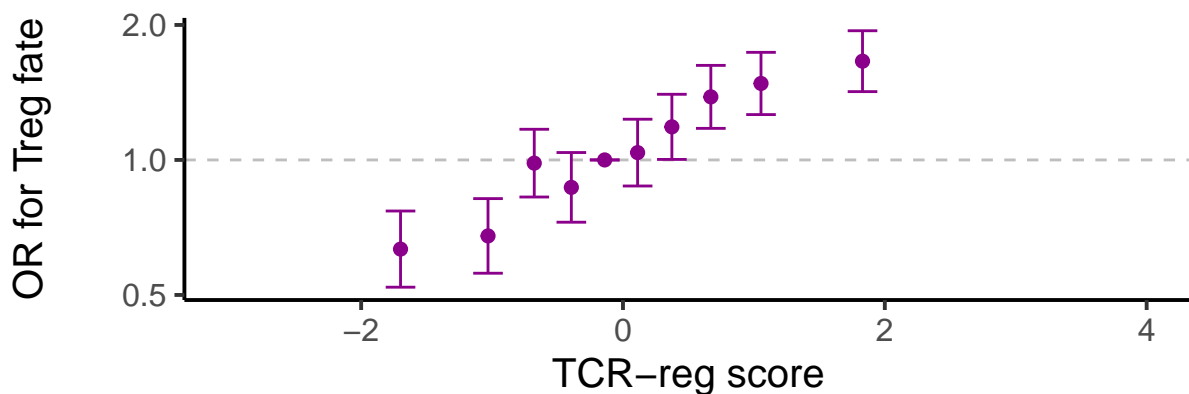
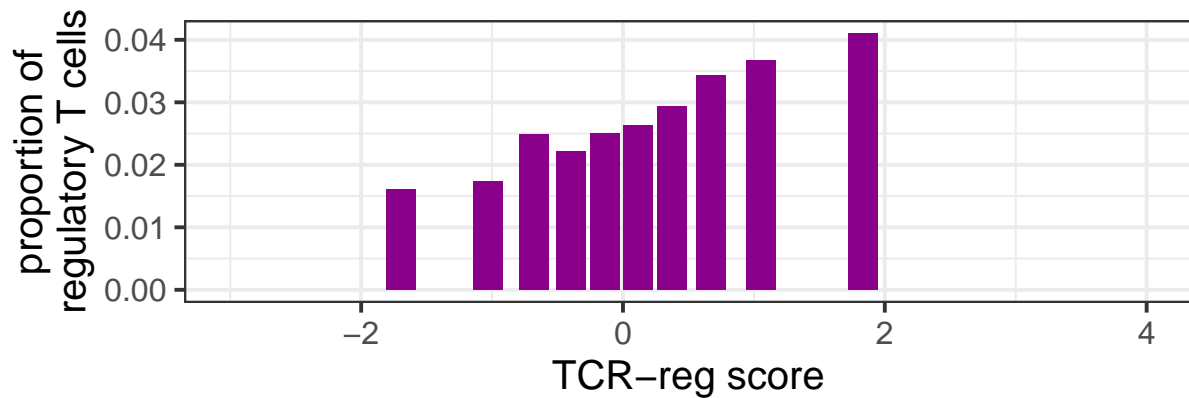
```
## # A tibble: 3 x 7
##   effect   group term          estimate std.error statistic   p.value
##   <chr>   <chr> <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 fixed   <NA> (Intercept)    -3.65     0.0557    -65.5     0
## 2 fixed   <NA> TCRscore      0.291     0.0187     15.6 9.25e-55
## 3 ran_pars Donor sd_ (Intercept)  0.512     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 0.0541
## 2 minptl 0.00869
## 3 other  0.0272
## # 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)    -4.18     0.100    -41.7     0
## 2 fixed   <NA> mmmmaxbin      0.971     0.0918     10.6 3.71e-26
## 3 ran_pars Donor sd_ (Intercept)  0.592     NA         NA      NA
```

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

```
## # A tibble: 10 x 9
##   bin    OR lowOR highOR term          lcat   ctl minbin maxbin
##   <dbl> <dbl> <dbl> <dbl> <chr>          <chr> <dbl> <dbl> <dbl>
## 1 -1.70 0.633 0.520 0.769 bin-1.69921469490498 blood   10 -6.95 -1.26
## 2 -1.03 0.677 0.559 0.820 bin-1.03091821155198 blood   20 -1.26 -0.837
## 3 -0.678 0.984 0.827 1.17 bin-0.677743921856654 blood   30 -0.837 -0.530
## 4 -0.394 0.869 0.726 1.04 bin-0.394476804726651 blood   40 -0.530 -0.264
## 5 0.112 1.04 0.875 1.23 bin0.111763686267537 blood   60 -0.0172 0.239
## 6 0.373 1.19 1.00 1.40 bin0.373388763124888 blood   70 0.239 0.513
## 7 0.671 1.38 1.18 1.63 bin0.671063630068232 blood   80 0.513 0.842
## 8 1.05 1.48 1.26 1.74 bin1.05431050945128 blood   90 0.842 1.30
## 9 1.83 1.66 1.42 1.94 bin1.82964083638695 blood  100 1.30 5.81
## 10 -0.140 1 1 1 ref blood   50 -0.264 -0.0172
```



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

```
## 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)    0.101  0.0620      1.63 1.03e- 1
## 2 fixed   <NA> TCRscore      0.136  0.00655    20.8 4.75e-96
## 3 ran_pars Donor sd_ (Intercept)    0.623   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 0.590
## 2 minptl 0.420
## 3 other  0.497
## # 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)   -0.163  0.0632     -2.58 9.83e- 3
## 2 fixed   <NA> mmmmaxbin      0.508  0.0291     17.4 3.49e-68
## 3 ran_pars Donor sd_ (Intercept)    0.597   NA          NA    NA
```

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

```
## # A tibble: 10 x 9
```

##	bin	OR	lowOR	highOR	term	lcat	qtl	minbin	maxbin
##	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
## 1	-1.72	0.812	0.768	0.860	bin-1.7219700916915	blood	10	-3.90	-1.28
## 2	-1.06	0.902	0.852	0.954	bin-1.05549944439346	blood	20	-1.28	-0.865
## 3	-0.709	0.968	0.915	1.02	bin-0.709242295117964	blood	30	-0.865	-0.562
## 4	-0.430	0.996	0.942	1.05	bin-0.429548881439132	blood	40	-0.562	-0.301
## 5	0.0642	1.04	0.981	1.10	bin0.0642404201507223	blood	60	-0.0573	0.189
## 6	0.322	1.06	0.997	1.12	bin0.322297082227014	blood	70	0.189	0.458
## 7	0.616	1.11	1.05	1.18	bin0.616265147356619	blood	80	0.458	0.784
## 8	0.994	1.19	1.12	1.26	bin0.993947601202813	blood	90	0.785	1.25
## 9	1.77	1.34	1.27	1.42	bin1.77025701863124	blood	100	1.25	4.80
## 10	-0.178	1	1	1	ref	blood	50	-0.301	-0.0573

