

Supplementary Information for teff: estimation of Treatment EFFECTs on transcriptomic data with casual random forest

Alejandro Cáceres and Juan R González

Supplementary Methods

We analyzed publicly available data in the GEO repository, using the R/Bioconductor packages that can be found at <https://www.bioconductor.org/>. Main results are obtained from the application of the package `teff` (<https://github.com/teff-package/teff>). The results discussed in the manuscript can be entirely reproduced with the following code.

Retrieving data from GSE117468

We downloaded transcriptomic and clinical data from 3-phase 3 clinical trials (AMAGINE 1-2-3) as deposited in GEO on the 2nd of April of 2020 with accession number GSE117468

```
library(GEOquery)
gsm <- getGEO("GSE117468", destdir = "./data", AnnotGPL = TRUE)
```

We first obtained clinical data relating to age, BMI, PASI, tissue (lesional or nonlesional), and brodalumab or placebo treatment. We considered all patients under two different brodalumab doses (140mg and 210mg).

```
#obtain phenotype data
phenobb <- pData(phenoData(gsm[[1]]))

#patient and sample IDs
patient <- phenobb$"patientid:ch1"
id <- rownames(phenobb)

#type of visit (baseline W0 or week 12 W12)
visit <- phenobb$"visit:ch1"

#clinical data
age <- as.numeric(phenobb$"age:ch1")
bmi <- as.numeric(phenobb$"bmi:ch1")
eff <- as.numeric(phenobb$"pasi:ch1")
tissue <- phenobb$"tissue:ch1"
t <- factor(factor(phenobb$"treatment:ch1",
  labels = c("brodalumab", "brodalumab",
    "placebo", NA)),
  levels=c("placebo", "brodalumab"))
```

We selected clinical data at baseline (BL) and transcriptomic data for non-lesional skin and stored the information in the `pheno.data.frame`

```

selBLN <- visit=="BL" & tissue=="non-lesional skin"
age <- age[selBLN]
bmi <- bmi[selBLN]
t <- t[selBLN]
id <- id[selBLN]
effbase <- eff[selBLN]

pheno <- data.frame(age, bmi, patient=patient[selBLN], t)
rownames(pheno) <- id

head(pheno)

##          age    bmi    patient      t
## GSM3300910  53 20.750 10216001001 brodalumab
## GSM3300916  51 35.235 10216001004    placebo
## GSM3300920  47 35.471 10216001005    placebo
## GSM3300924  49 27.898 10216001006 brodalumab
## GSM3300928  38 33.272 10216003001 brodalumab
## GSM3300932  47 36.553 10216003002    placebo

```

We selected clinical data at baseline (BL) and transcriptomic data for nonlesional skin and PASI at week 12.

```

#obtain PASI at week 12
effend <- eff[which(visit=="W12" & tissue=="non-lesional skin")]
names(effend) <- patient[visit=="W12" & tissue=="non-lesional skin"]
effend <- effend[as.character(pheno$patient)]

#add effects
pheno <- cbind(pheno,
  eff = as.factor(effbase>effend), #response in PASI
  effdif = (effbase-effend)/effbase, #level of repose in PASI
  effbase = effbase, # PASI at baseline
  effend = effend) # PASI at week 12

#store clinical data, store in phenodat
pheno <- pheno[complete.cases(pheno),]
head(pheno)

##          age    bmi    patient      t    eff    effdif effbase effend
## GSM3300910  53 20.750 10216001001 brodalumab TRUE  1.00000000    12.4    0.0
## GSM3300916  51 35.235 10216001004    placebo TRUE  0.44791667    19.2   10.6
## GSM3300920  47 35.471 10216001005    placebo FALSE -0.16417910    13.4   15.6
## GSM3300928  38 33.272 10216003001 brodalumab TRUE  0.85427136    19.9    2.9
## GSM3300932  47 36.553 10216003002    placebo FALSE -0.67980296    20.3   34.1
## GSM3300936  64 32.189 10216003003    placebo FALSE -0.08116883    30.8   33.3

```

The outcome variables were:

- effbase: PASI at baseline (W0)
- effend: PASI at week 12 (W12)
- eff: categorical improvement given by the improvement in PASI between baseline and week 12 ($W12 < W0$)

- `effdif`: fraction of improvement of PASI from baseline ($\frac{W_0 - W_{12}}{W_0}$)

We then obtained the transcriptomic data for the selected individuals across 53951 transcripts.

```
#obtain annotation data, store in genesIDs
genesIDs <- fData(gsm[[1]])

#obtain transcriptomic data, store in expr
expr <- exprs(gsm[[1]])
expr <- expr[,rownames(pheno)]

genesid <- sapply(strsplit(genesIDs$"Gene symbol", "/"), function(x) x[1])
names(genesid) <- rownames(genesIDs)
genesentrez <- genesIDs$"Gene ID"
names(genesentrez) <- rownames(genesIDs)

dim(expr)

## [1] 53951    96
```

We have the final set of individuals used in the analysis

```
table(pheno$t)

##
##      placebo brodalumab
##          25          71
```

Transcriptome-wide interaction analysis

We used Bioconductor packages `limma` and `sva` to estimate the differential gene expression with the interaction between categorical PASI improvement and treatment type brodalumab or placebo. We extracted the surrogate variables with `sva` and estimated the effects of the interaction with `limma`.

We, therefore, tested the association between gene expression and the interaction between PASI improvement (P) and treatment (t) using the linear model

$$E_{ij} = \alpha_i + \beta_i(P_j \times t_j) + \sum_{r=1 \dots k} \gamma_{ijk} C_{rj} + \epsilon_{ij}$$

where E_{ij} is the post-processed transcript intensity i for individual j with PASI improvement P_j and treatment t_j . C_{rj} are k covariates that include age, BMI and surrogate effects. β_i was the effect of interest that measures the association between the expression level of probe i and the interaction between PASI improvement and treatment. Significant genes were obtained from false discovery rates (FDR) < 0.05 of P-values corrected for multiple comparisons.

```
library(sva)
library(limma)

##interaction between treatment and improvement in PASI: t*eff

#compute SVAs
mod0 <- model.matrix(~ t + eff + age + bmi, data = pheno)
mod <- model.matrix(~ t:eff + t + eff + age + bmi, data = pheno)
ns <- num.sv(expr, mod, method="be")
ss <- sva(expr, mod, mod0, n.sv=ns)$sv
```

```
## Number of significant surrogate variables is: 15
## Iteration (out of 5):1 2 3 4 5

modss <- cbind(mod, ss)

#estimate associations
fit <- lmFit(expr, modss)
fit <- eBayes(fit)
```

The volcano plot showed numerous genes with significant differential expression, downregulated with the interaction. The volcano plot is obtained as follows

```
library(EnhancedVolcano)

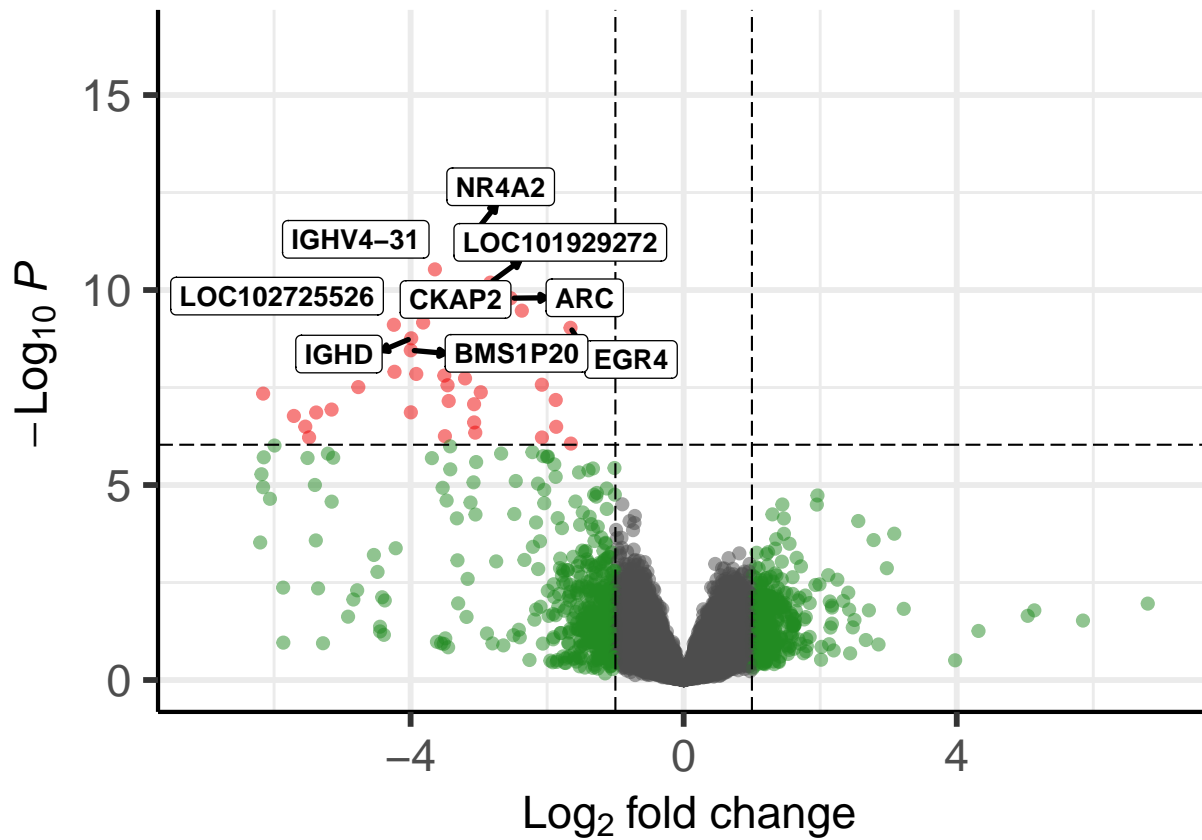
tt <- topTable(fit, coef="tbrodalumab:effTRUE", number=Inf)

gns <- genesid[rownames(tt)]
gns[11:length(gns)] <- ""
tt <- data.frame(genes=gns, tt)

EnhancedVolcano(tt, lab = tt$genes,
  selectLab = na.omit(tt$genes[1:11]),
  x = 'logFC', y = 'P.Value',
  xlim=c(-7, 7),
  pCutoff = 0.05/nrow(tt),
  labSize = 4.0,
  labCol = 'black',
  labFace = 'bold',
  boxedLabels = TRUE,
  legendPosition = 'bottom',
  drawConnectors = TRUE,
  widthConnectors = 1,
  colConnectors = 'black',
  title = "PASI response",
  subtitle = "Differential expression")
```

PASI response

Differential expression



● NS ● Log₂ FC ● p-value ● p-value and log₂ FC

total = 53951 variables

We selected the association that was significant after false-discovery rate correction.

```
tt <- topTable(fit, number=Inf, coef="tbrodalumab:effTRUE")

#Select significant associations
transcriptname <- rownames(tt)
sigGenespso <- transcriptname[tt$adj.P.Val<0.05]

tt <- data.frame(Gene= genesid[sigGenespso], tt[sigGenespso,])

tt[,c(2:4,7)] <- format(tt[,c(2:4,7)], digits=3)
tt[,5:6] <- format(tt[,5:6], digits=3, scientific=TRUE)
```

```
head(tt,20)
```

##		Gene	logFC	AveExpr	t	P.Value	adj.P.Val	B
##	204622_x_at	NR4A2	-3.111	6.52	-8.16	4.31e-12	2.33e-07	14.26
##	211868_x_at	IGHV4-31	-3.644	4.87	-7.73	2.95e-11	7.96e-07	12.79
##	215565_at	LOC101929272	-2.831	2.75	-7.55	6.45e-11	1.16e-06	12.19
##	210090_at	ARC	-2.535	3.63	-7.34	1.62e-10	2.18e-06	11.48
##	215036_at	<NA>	-2.371	4.32	-7.18	3.38e-10	3.65e-06	10.91
##	234884_x_at	CKAP2	-3.816	4.43	-7.02	6.82e-10	6.00e-06	10.36
##	217281_x_at	LOC102725526	-4.245	4.51	-6.99	7.79e-10	6.00e-06	10.26
##	207768_at	EGR4	-1.658	3.50	-6.95	9.31e-10	6.28e-06	10.12
##	214973_x_at	IGHD	-3.991	4.02	-6.81	1.73e-09	1.04e-05	9.63
##	217179_x_at	BMS1P20	-3.997	4.13	-6.65	3.51e-09	1.89e-05	9.08
##	217258_x_at	IGLV1-44	-4.236	4.49	-6.35	1.25e-08	6.12e-05	8.08
##	234877_x_at	<NA>	-3.917	4.20	-6.32	1.42e-08	6.38e-05	7.98
##	216248_s_at	NR4A2	-3.506	6.02	-6.30	1.56e-08	6.49e-05	7.91
##	211634_x_at	IGHM	-3.202	3.45	-6.26	1.85e-08	7.11e-05	7.78
##	230494_at	SLC20A1	-2.077	7.41	-6.17	2.68e-08	9.33e-05	7.48
##	204621_s_at	NR4A2	-3.459	4.43	-6.17	2.77e-08	9.33e-05	7.46
##	216984_x_at	IGLJ3	-4.766	4.87	-6.14	3.08e-08	9.78e-05	7.37
##	211881_x_at	IGLJ3	-2.972	6.10	-6.07	4.18e-08	1.25e-04	7.13
##	216401_x_at	MLIP	-6.160	5.19	-6.05	4.53e-08	1.29e-04	7.07
##	234364_at	IGLL5	-1.874	3.20	-5.96	6.59e-08	1.78e-04	6.77

We illustrate top association by violin plots of the residuals of the log-fold change against for the categories: i) Placebo or no improvement of categorical PASI and ii) Brodalumab and improvement of PASI. The significant interaction is illustrated in the violin plots by the difference in gene transcription between those two categories.

```
library(vioplot)

par(mfrow=c(1,3))

top <- rownames(tt)[1]
tr <- log(expr[top,])
res <- summary(lm(tr~modss[, -c(1,2,3,6)]))$residuals
et <- modss[,6]
fc <- factor(modss[,6], labels=c("\n Placebo \n OR \n No response",
                                "\n Brodalumab \n AND \n Response"))

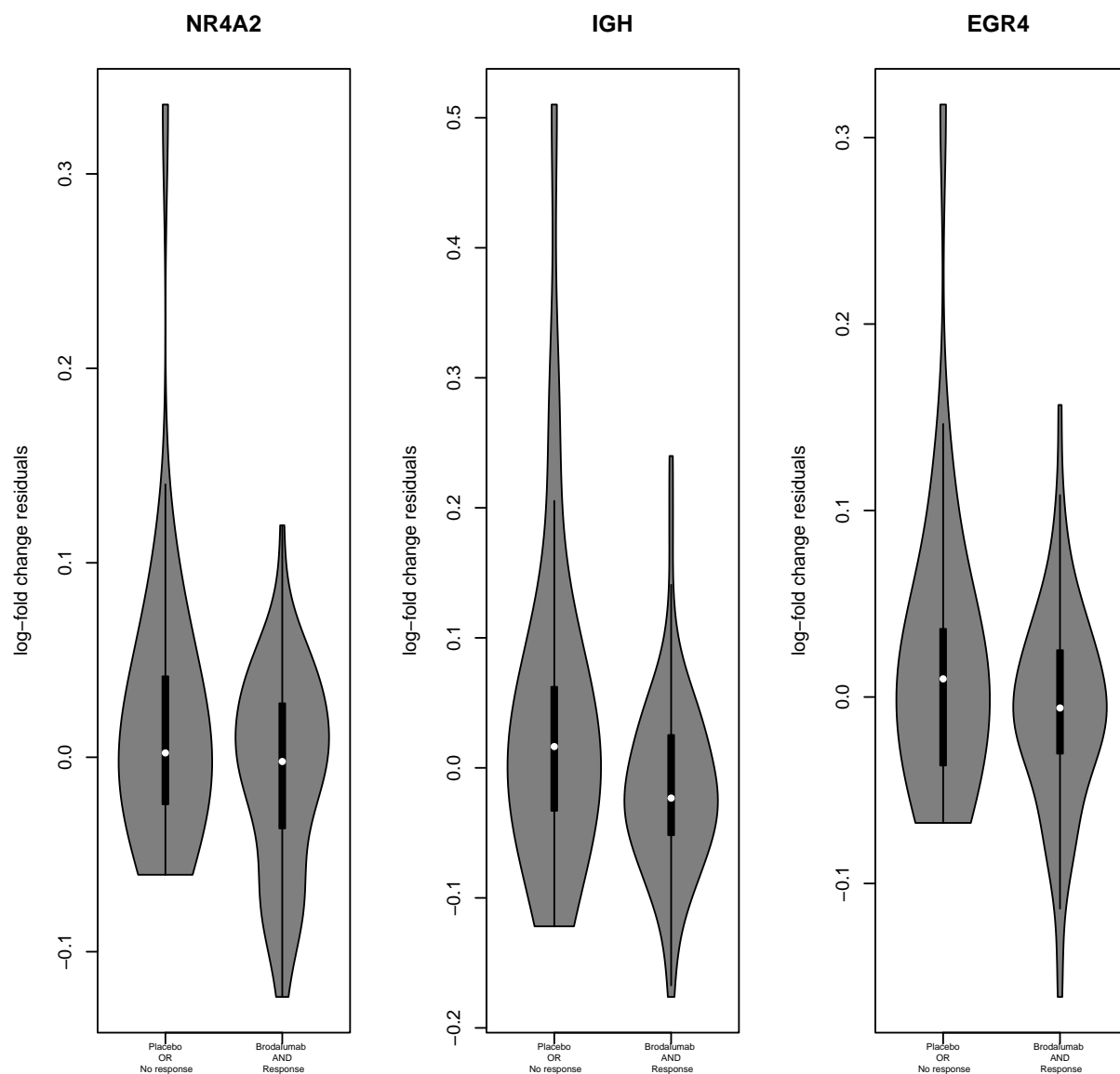
vioplot(res~fc, xlab="", main="NR4A2",
        ylab="log-fold change residuals", cex.names=0.5)

top <- rownames(tt)[2]
tr <- log(expr[top,])
res <- summary(lm(tr~modss[, -c(1,2,3,6)]))$residuals
et <- modss[,6]
fc <- factor(modss[,6],
             labels=c("\n Placebo \n OR \n No response",
                     "\n Brodalumab \n AND \n Response"))

vioplot(res~fc, xlab="", main="IGH",
        ylab="log-fold change residuals", cex.names=0.5)

top <- rownames(tt)[8]
```

```
tr <- log(expr[top,])
res <- summary(lm(tr~modss[, -c(1,2,3,6)]))$residuals
et <- modss[,6]
fc <- factor(modss[,6], labels=c("\n Placebo \n OR \n No response",
                                "\n Brodalumab \n AND \n Response"))
vioplot(res~fc, xlab="", main="EGR4 ",
        ylab="log-fold change residuals", cex.names=0.5)
```



Enrichment analyses were performed for the molecular functions of the gene ontology terms (<http://geneontology.org/>).

```
library(clusterProfiler)

mappedgenesIds <- genesentrez[rownames(tt)]
```

```
mappedgenesIds <- unique(unlist(strsplit(mappedgenesIds, " /// ")))

#run enrichment in GO
GO <- enrichGO(gene = mappedgenesIds, 'org.Hs.eg.db',
               ont="MF", pvalueCutoff=0.05, pAdjustMethod="BH")

GO <- data.frame(ID=GO$ID, Description=GO$Description,
                 Padj=format(GO$p.adjust, digits=3, scientific=TRUE), GeneRatio=GO$GeneRatio)

head(GO)
```

##	ID	Description	Padj	GeneRatio
## 1	GO:0034987	immunoglobulin receptor binding	5.34e-06	5/27
## 2	GO:0003823	antigen binding	5.34e-06	6/27
## 3	GO:0035259	glucocorticoid receptor binding	1.80e-05	3/27
## 4	GO:0035258	steroid hormone receptor binding	1.02e-04	4/27
## 5	GO:0016922	nuclear receptor binding	2.05e-04	4/27
## 6	GO:0140297	DNA-binding transcription factor binding	2.13e-04	6/27

causal random forest

We implemented causal random forest package grf (<https://grf-labs.github.io/grf/>) for transcriptomic data in the software package teff (<https://github.com/teff-package/teff>). For installing teff

```
library(devtools)
install_github("teff-package/teff")
```

We prepared feature data corresponding to the transcriptomic data of the significant transcripts identified in the previous analysis and treatment-effect data corresponding to the treatment received, categorical PSI improvement, and clinical and surrogate covariates.

```
library(teff)

#Prepare data, features: trascription data, teff: treatment, effect and covariates
teffdata <- modss[,-c(1,6)]
colnames(teffdata)[1:2] <- c("t", "eff")
colnames(teffdata)[5:ncol(teffdata)] <- paste0("cov",5:ncol(teffdata))

psoriasis <- list(features=t(expr), teffdata=teffdata)
```

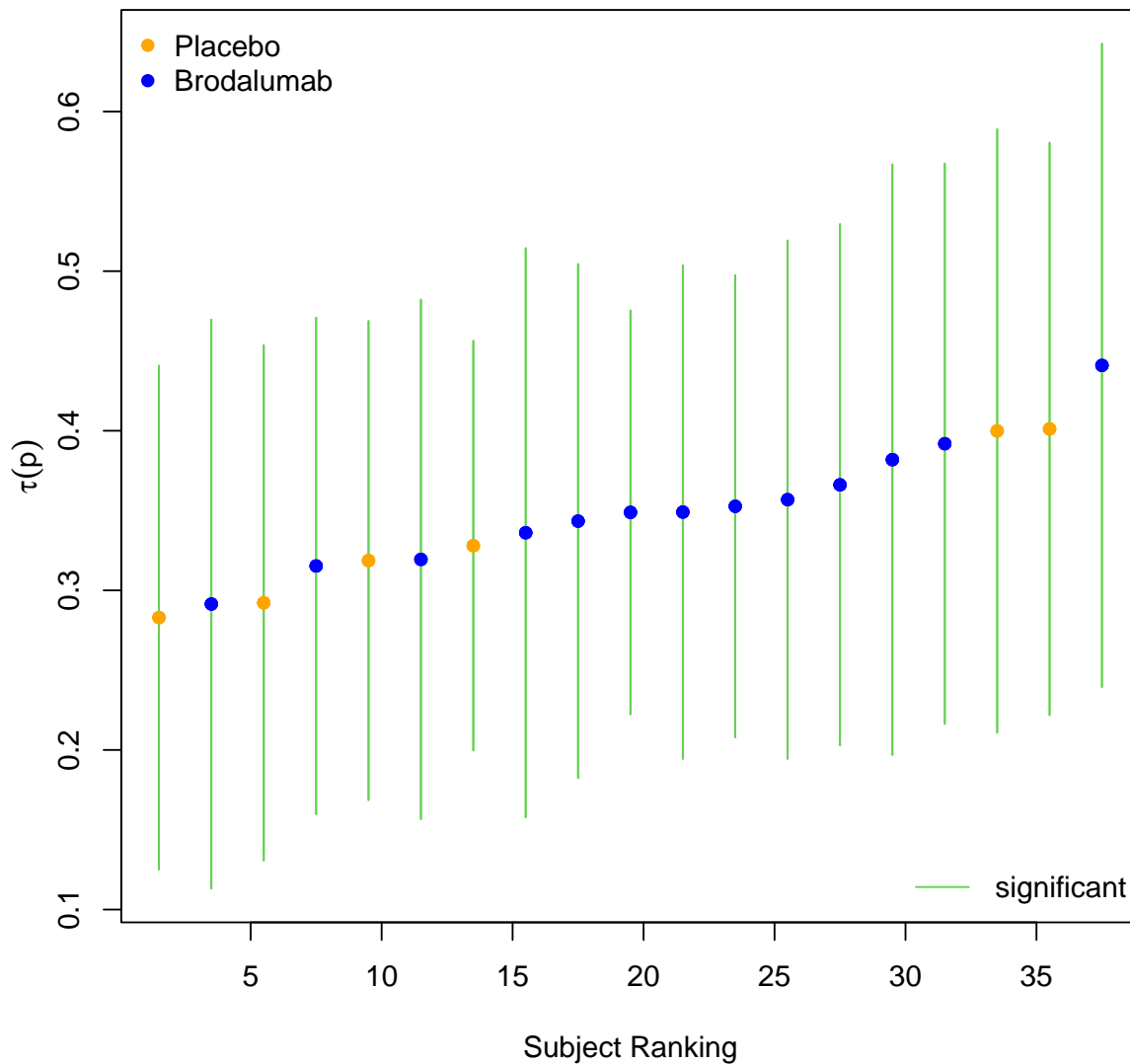
We aimed to estimate for each patient the benefit of a potential brodalumab treatment vs placebo according to their transcription data on nonlesional skin at baseline. We defined the potential effect of brodalumab treatment $\tau(p)$

A main advantage of CRF is that it can estimate the confidence interval (CI) for $\tau(p)$. We applied CRF to the transcription levels of selected genes. First, a random train-set of 80% patients is drawn to grow the forest. The remaining 20% of patients were set aside and not used to grow the forest. These test individuals were used to estimate their $\tau(p)$ and 95% CIs according to the CRF predictor. The application of these procedures was implemented in the function profile of teff


```
pso <-predicteff(psoriasis, featuresinf=sigGenespso, profile=TRUE, dup=TRUE, quant = 0.3)
```

We plot $\tau(p)$ with its 95% CIs, using the function plotPredict

```
plotPredict(pso, lb=expression(tau(p)),
  ctrl.plot = list(lb=c("Placebo", "Brodalumab"),
    wht="topleft", whs = "bottomright"))
```



Logistic relation between $\tau(p)$ and observed PASI improvement

$\tau(p)$ is a measure at baseline for the estimated benefit of a potential treatment with brodalumab vs placebo. We did not observe any correlation of the prediction at baseline with future treatment or with PASI at baseline, for either treatment.

```
treatment <- pso$treatment+1
names(treatment) <- pso$subsids
```

```
tau <- pso$predictions
names(tau) <- pso$subsids
```

```
selsubs <- names(tau)
```

```
response <- pheno[selsubs,"effdif"]
base <- pheno[selsubs,"effbase"]
bmi <- pheno[selsubs,"bmi"]
age <- pheno[selsubs,"age"]
```

```
#association with treatment
```

```
summary(lm(log(tau/(1-tau))~treatment))
```

```
##
```

```
## Call:
```

```
## lm(formula = log(tau/(1 - tau)) ~ treatment)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.28073 -0.13086 -0.01634  0.10937  0.37039
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.75838    0.11013  -6.886 4.63e-08 ***
## treatment    0.07533    0.06303   1.195    0.24
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 0.1806 on 36 degrees of freedom
```

```
## Multiple R-squared:  0.03816, Adjusted R-squared:  0.01144
```

```
## F-statistic: 1.428 on 1 and 36 DF,  p-value: 0.2399
```

```
#association with PASI at baseline in placebo
```

```
summary(lm(log(tau/(1-tau))~base, subset=which(treatment==1)))
```

```
##
```

```
## Call:
```

```
## lm(formula = log(tau/(1 - tau)) ~ base, subset = which(treatment ==
##      1))
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.27111 -0.13512 -0.07517  0.27760  0.27898
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.611399    0.164099  -3.726  0.00394 **
## base        -0.003292    0.006907  -0.477  0.64390
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2278 on 10 degrees of freedom
## Multiple R-squared:  0.02221, Adjusted R-squared:  -0.07557
## F-statistic: 0.2271 on 1 and 10 DF,  p-value: 0.6439
```

```
#association with PASI at baseline in brodalumab
summary(lm(log(tau/(1-tau))~base, subset=which(treatment==2)))

##
## Call:
## lm(formula = log(tau/(1 - tau)) ~ base, subset = which(treatment ==
##      2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24692 -0.11505  0.00883  0.03289  0.33662
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.714707   0.107666  -6.638 7.25e-07 ***
## base         0.005586   0.005376   1.039   0.309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1602 on 24 degrees of freedom
## Multiple R-squared:  0.04304, Adjusted R-squared:  0.003162
## F-statistic: 1.079 on 1 and 24 DF,  p-value: 0.3092
```

To assess the power of the prediction, we tested whether the prediction correlated with the observed levels do response at week 12 after treatment with brodalumab or placebo. We fitted a logistic relationship between the prediction at baseline (dose) with the observed levels of the improvement of PASI (response), given by the percentage of PASI improvement between baseline and week 12. For each treatment, We thus fitted the three-parameter logistic model:

$$PASI(\tau) = \frac{de^{b(\log(\tau)+e)}}{1 + e^{b(\log(\tau)+e)}}$$

where the lower limit is equal to 0. d is the maximum PASI improvement, e the median of τ and b the rate of the effect. We used the function `drm` from the package `drc`, where the rate of change b is parametrized as $-b$.

```
library(drc)

#dose-response under placebo
dresponse <- response[treatment==1]
dtau <- tau[treatment==1]
metP <- drm(dresponse*100~dtau, fct=LL.3())
metP

##
## A 'drc' model.
##
## Call:
```

```
## drm(formula = dresponse * 100 ~ dtau, fct = LL.3())
##
## Coefficients:
## b:(Intercept)  d:(Intercept)  e:(Intercept)
##           1.8484           45.8758           0.2136
```

```
#dose-response under brodalumab
dresponse <- response[treatment==2]
dtau <- tau[treatment==2]
metB <- drm(dresponse*100~dtau, fct=LL.3())
metB

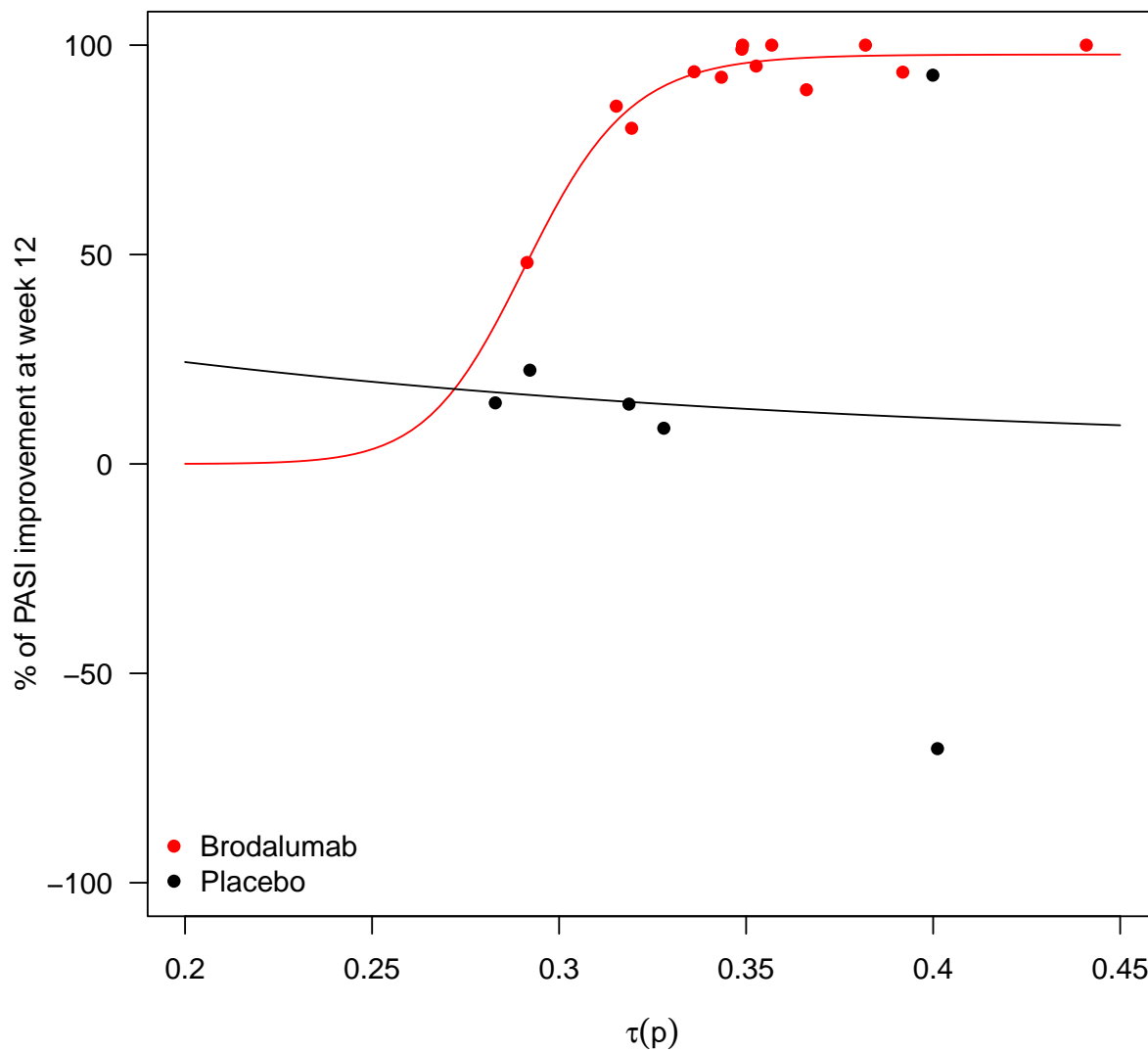
##
## A 'drc' model.
##
## Call:
## drm(formula = dresponse * 100 ~ dtau, fct = LL.3())
##
## Coefficients:
## b:(Intercept)  d:(Intercept)  e:(Intercept)
##          -21.2700           97.7581           0.2919
```

We plot the fitted curves with observed values.

```
plot(metB, log = "", pch=16, col="red", ylim=c(-100,100), xlim=c(0.2,0.45),
     ylab="% of PASI improvement at week 12",
     xlab=expression(tau(p)))

plot(metP, log = "", pch=16, col="black", ylim=c(-100,100), xlim=c(0.2,0.45),
     add=TRUE)

legend("bottomleft", legend=c("Brodalumab", "Placebo"),
     pch=16, col=c("red", "black"), bty="n")
```



We tested whether there was a significant logistic relationship between τ and the levels of improvement in PASI for each treatment, using a log-likelihood test between the model and a model where the response is on average constant. We observed a strong relationship for brodalumab but not for placebo.

```
noEffect(metB)

## Chi-square test          Df      p-value
## 6.815728e+01    2.000000e+00  1.554312e-15

noEffect(metP)

## Chi-square test          Df      p-value
## 0.03134458    2.00000000  0.98444988
```

We assessed the logistic relationship between PASI at baseline on PASI improvement after treatment.

```

#dose-response under placebo
dresponse <- response[treatment==1]
dbase <- base[treatment==1]
metP<-drm(dresponse*100~dbase, fct=LL.3())

dresponse <- response[treatment==2]
dbase <- base[treatment==2]
metB<-drm(dresponse*100~dbase, fct=LL.3())

noEffect(metB)

## Chi-square test          Df          p-value
##      6.31003378          2.00000000          0.04263768

noEffect(metP)

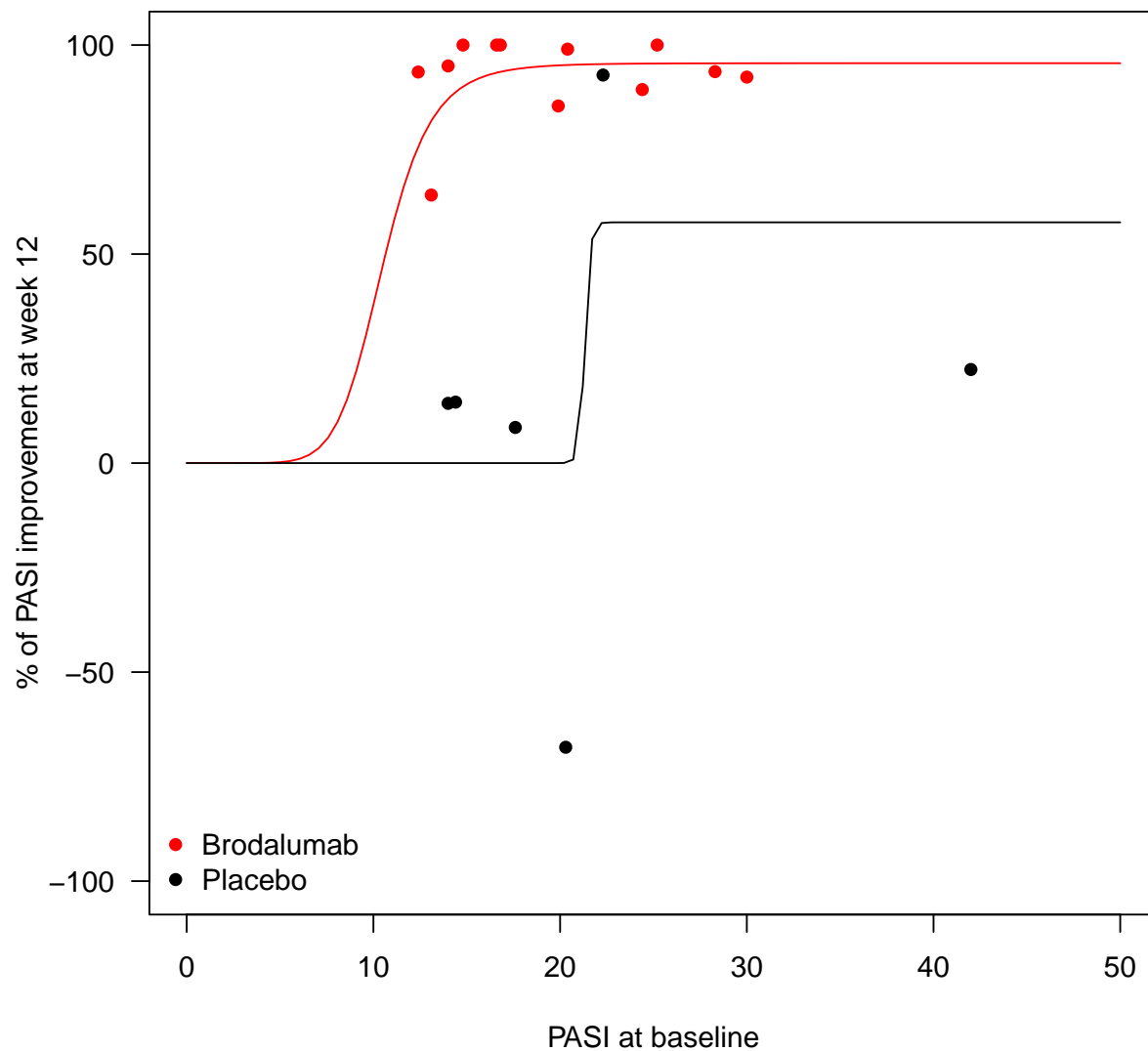
## Chi-square test          Df          p-value
##      6.464632            2.000000          0.039466

plot(metB, log = "", pch=16, col="red", ylim=c(-100,100), xlim=c(0,50),
     ylab="% of PASI improvement at week 12",
     xlab="PASI at baseline")

plot(metP, log = "", pch=16, col="black", ylim=c(-100,100), xlim=c(0,50),
     add=TRUE)

legend("bottomleft", legend=c("Brodalumab", "Placebo"),
     pch=16, col=c("red", "black"), bty="n")

```



We assessed the logistic relationship between BMI on PASI improvement after treatment.

```
#dose-response under placebo
dresponse <- response[treatment==1]
dbmi <- bmi[treatment==1]
metP<-drm(dresponse*100~dbmi, fct=LL.3())

dresponse <- response[treatment==2]
dbmi <- bmi[treatment==2]
metB<-drm(dresponse*100~dbmi, fct=LL.3())

noEffect(metB)
```

##	Chi-square test	Df	p-value
##	1.3206427	2.0000000	0.5166853

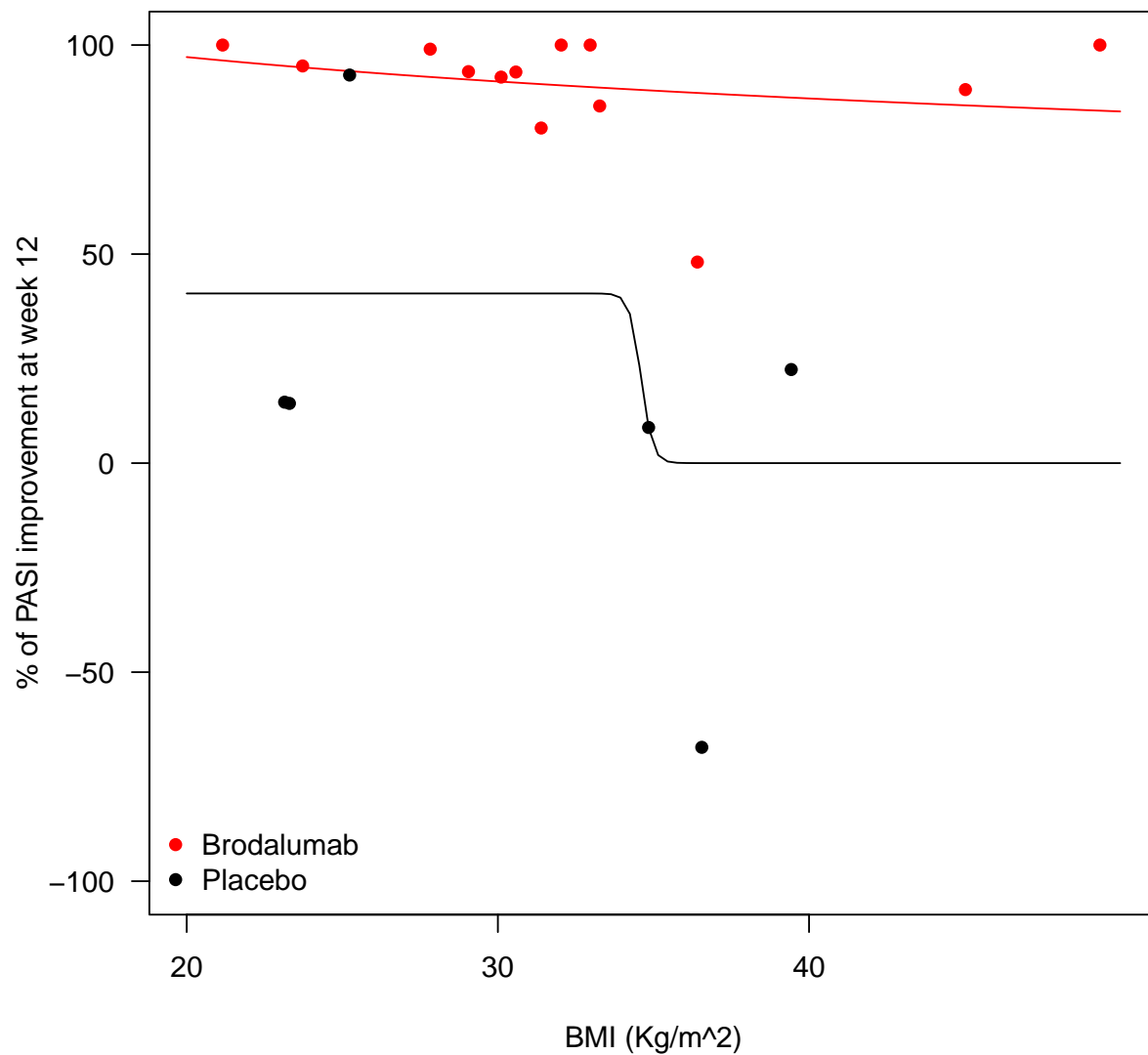
```
noEffect(metP)
```

## Chi-square test	Df	p-value
## 4.1560384	2.0000000	0.1251779

```
plot(metB, log = "", pch=16, col="red", ylim=c(-100,100), xlim=c(20,50),  
      ylab="% of PASI improvement at week 12",  
      xlab="BMI (Kg/m^2)")
```

```
plot(metP, log = "", pch=16, col="black", ylim=c(-100,100), xlim=c(20,50),  
      add=TRUE)
```

```
legend("bottomleft", legend=c("Brodalumab", "Placebo"),  
      pch=16, col=c("red", "black"), bty="n")
```

We assessed the logistic relationship between age on PASI improvement after treatment.

```
#dose-response under placebo
dresponse <- response[treatment==1]
dage <- age[treatment==1]
metP<-drm(dresponse*100~dage, fct=LL.3())

dresponse <- response[treatment==2]
dage <- age[treatment==2]
metB<-drm(dresponse*100~dage, fct=LL.3())

noEffect(metB)

## Chi-square test          Df      p-value
## -0.0002472357      2.0000000000  1.0000000000
```

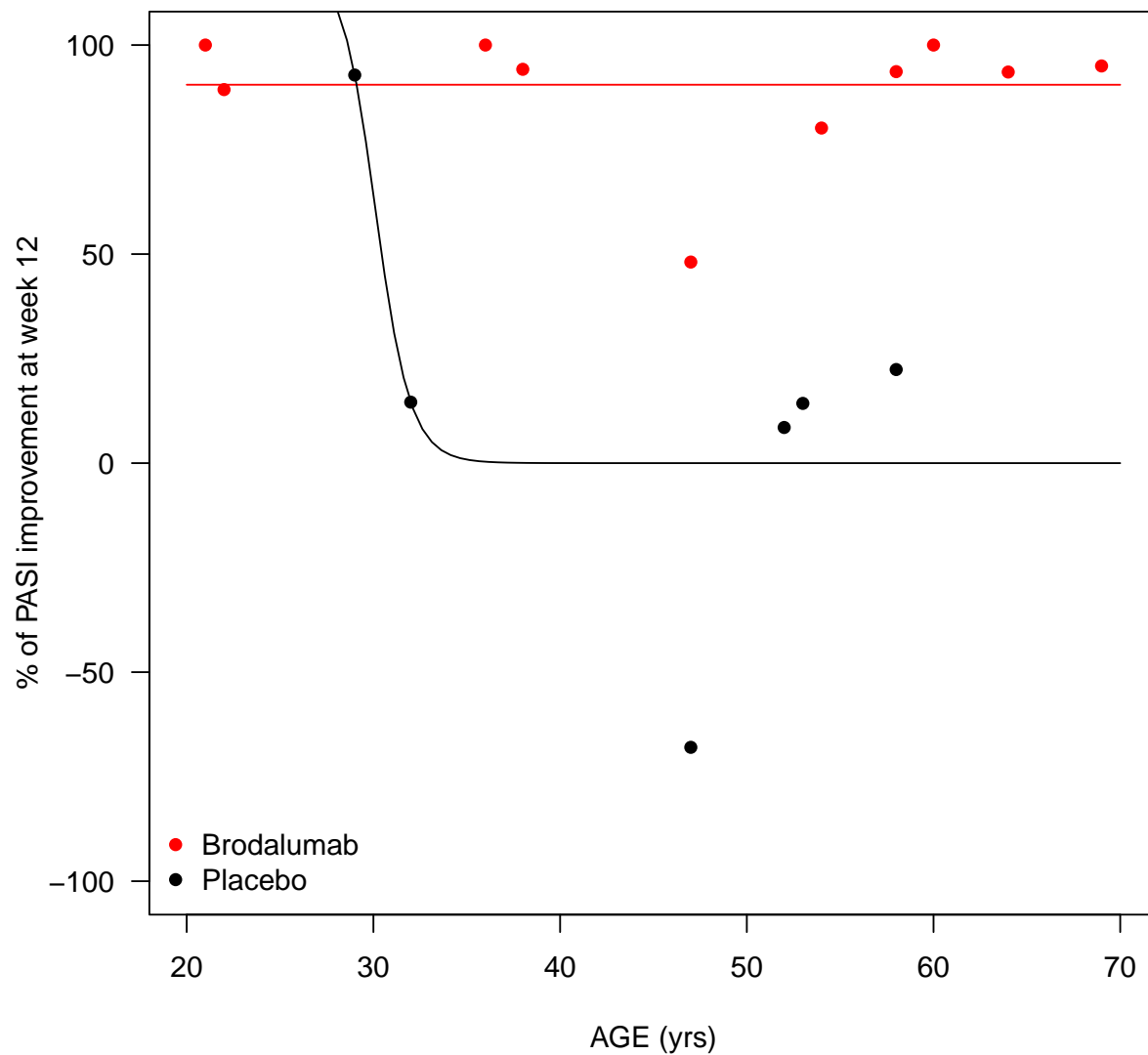
```
noEffect(metP)
```

##	Chi-square test	Df	p-value
##	10.576978641	2.000000000	0.005049382

```
plot(metB, log = "", pch=16, col="red", ylim=c(-100,100), xlim=c(20,70),  
     ylab="% of PASI improvement at week 12",  
     xlab="AGE (yrs)")
```

```
plot(metP, log = "", pch=16, col="black", ylim=c(-100,100), xlim=c(20,70),  
     add=TRUE)
```

```
legend("bottomleft", legend=c("Brodalumab", "Placebo"),  
      pch=16, col=c("red", "black"), bty="n")
```



Targeting

We selected individuals with statistically significant $\tau(p)$ greater than 0.2. This was consistent with an significant increase PASI improvement of at least 25% as given by the logistic relationship between τ and PASI improvement, as described in the previous section.

```
dresponse <- response[treatment==1]
dttau <- tau[treatment==1]
metP <- drm(dresponse*100~dttau, fct=LL.3())
predict(metP, data.frame(dttau=0.2))

## Prediction
## 24.32748
```

The function `predicteff` extracts the individuals with $\tau > 0.2$ and builds the binary transcriptomic profile for individuals with high expected brodalumab benefit. See, Table 3 in the manuscript where FALSE is Down and TRUE is High.

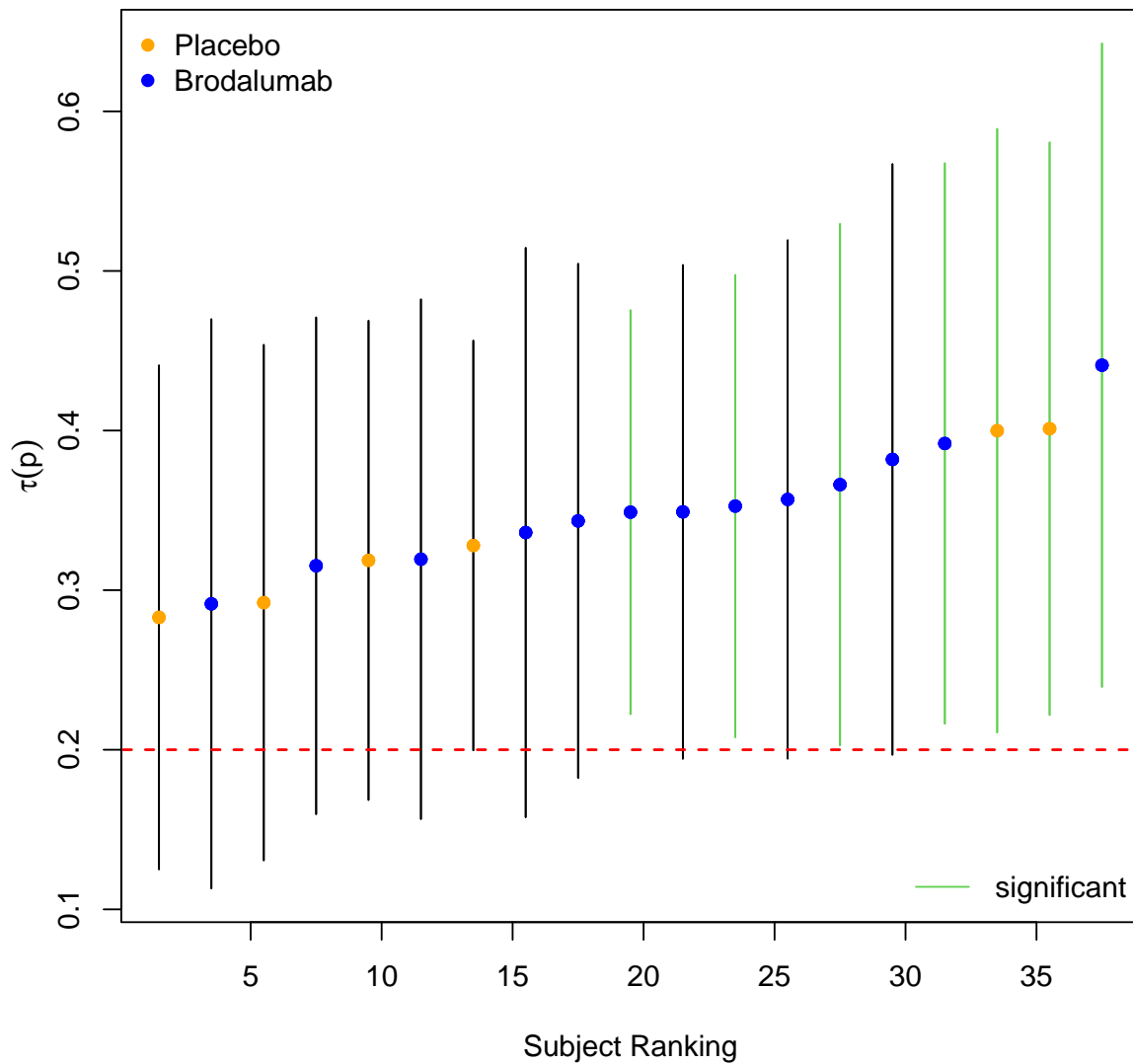
```
pso <-predicteff(psoriasis, featuresinf=sigGenespso,
                 profile=TRUE, dup=TRUE, quant=0.5, resplevel = 0.2)

pso$profile$profpositive
```

	234366_x_at	201236_s_at	214973_x_at	217378_x_at	215214_at	216979_at	210809_s_at
## [1,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
	235094_at	214777_at	207768_at	230494_at	234884_x_at	1558623_at	1558078_at
## [1,]	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
	211639_x_at	211881_x_at	216852_x_at	238472_at	217157_x_at		
## [1,]	FALSE	FALSE	FALSE	FALSE	FALSE		

The binary profile can be used to target individuals in other studies. To study the consistency of the targeting we first target all the individuals in the brodalumab study

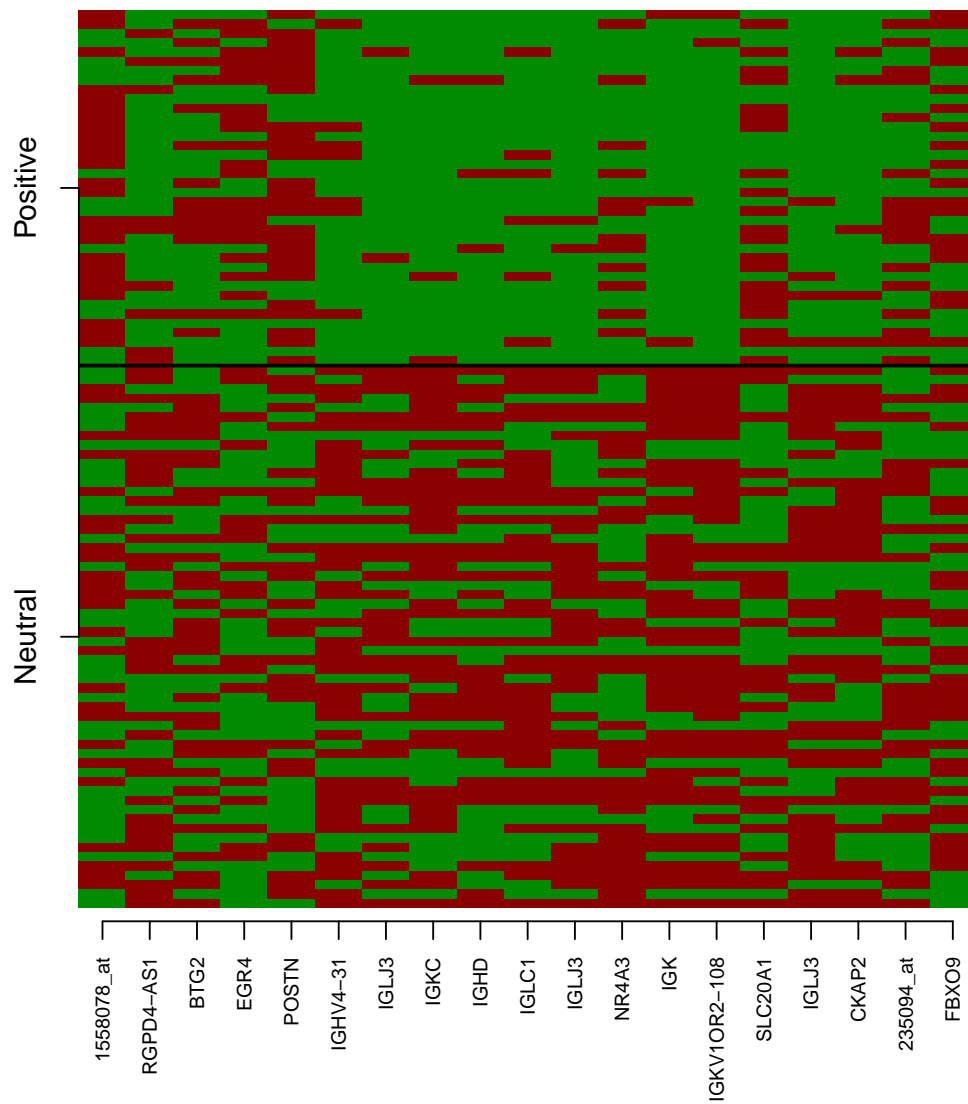
```
plotPredict(pso, lb=expression(tau(p)),
            ctrl.plot = list(lb=c("Placebo", "Brodalumab"),
                             wht="topleft", whs = "bottomright"))
```



and confirmed that the targeting significantly interacts with treatment on treatment response (eff). Patients in the positive group are those with high predicted benefit to brodalumab while patients in the neutral groups are with low benefit.

```
nmf <- colnames(psoriasis$features)
nmf <- nmf[nmf%in%colnames(pso$profile$profpositive)]
l1 <- genesid[nmf]
l1[is.na(l1)] <- names(l1)[is.na(l1)]

res <- target(psoriasis, pso, plot=TRUE, nmcov = c("bmi", "age"),
              effect="positive", match=0.6, model=NULL,
              lb=l1)
```



```
library(arm)

y <- psoriasis$teffdata[,"eff"]
x <- factor(res$classification, labels=c("Low benefit", "High benefit"))
w <- psoriasis$teffdata[,"t"]

summary(bayesglm(y ~ x*w, family="binomial"))

##
## Call:
## bayesglm(formula = y ~ x * w, family = "binomial")
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.7360  0.1536  0.2190  0.2190  1.6256
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.2881    0.7986   2.865  0.00417 **
## xHigh benefit  -3.2991    1.0082  -3.272  0.00107 **
## w              1.4309    1.0884   1.315  0.18864
## xHigh benefit:w  4.0146    1.8678   2.149  0.03161 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 64.155  on 95  degrees of freedom
## Residual deviance: 27.885  on 92  degrees of freedom
## AIC: 35.885
##
## Number of Fisher Scoring iterations: 24
```

We investigated biological correlates of the targeting with biological conditions relevant for psoriasis etiology. We inferred the abundance of T-cell in non-lesional skin at baseline with immunedeconv and correlated it with the classification of individuals into predicted high and low brodalumab benefit. We used to infer T-cell count from transcriptomic data.

```
library(immunedeconv)

gns <- genesid[rownames(expr)]
rownames(expr) <- gns

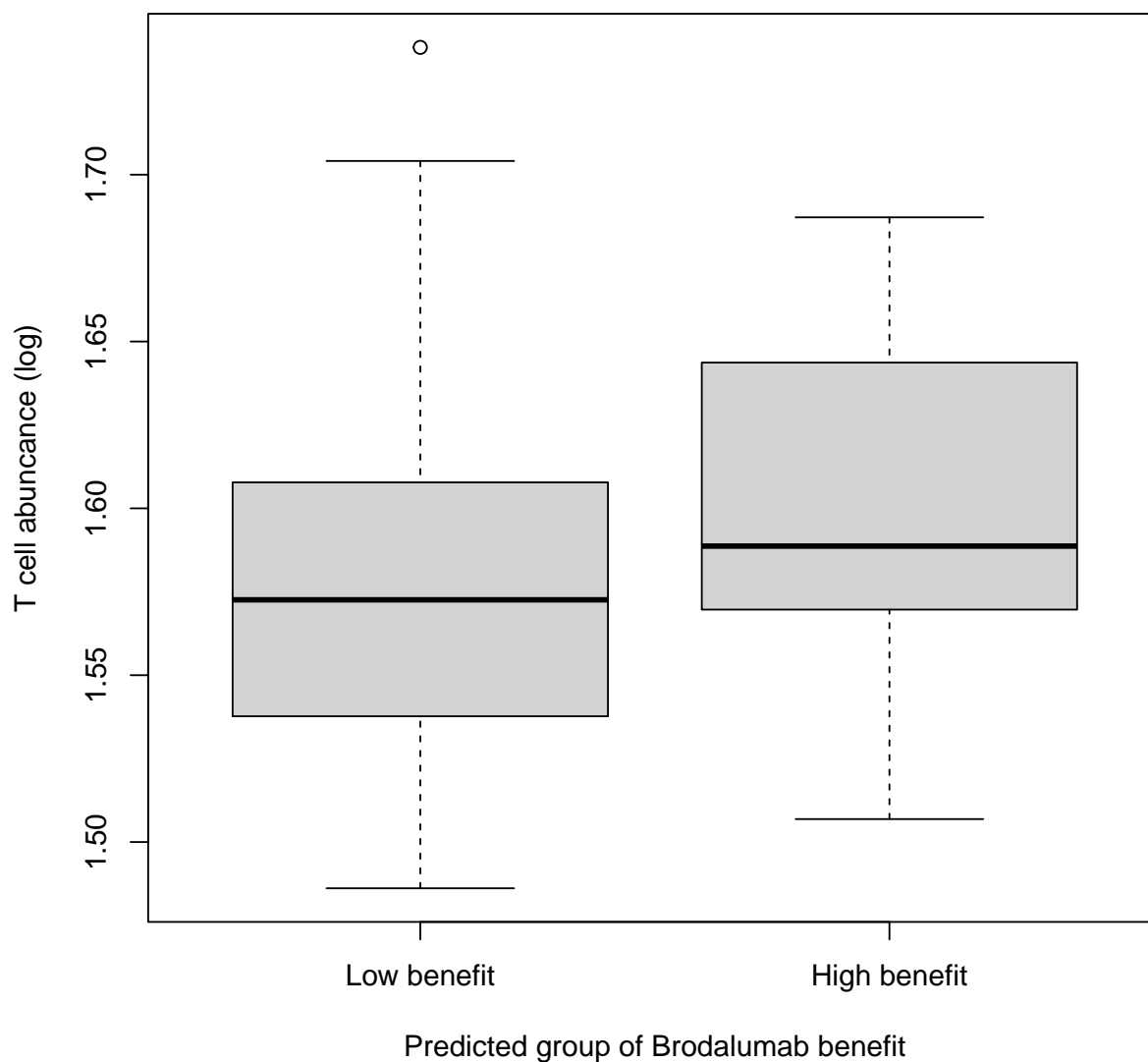
cellcomp2 <- deconvolute(expr, "mcp_counter", arrays=TRUE, column="Symbol")
cellnames <- cellcomp2$cell_type
cm <- matrix(as.numeric(t(cellcomp2)[-1,]), ncol=length(cellnames))
colnames(cm) <- cellnames
rownames(cm) <- colnames(cellcomp2)[-1]
tcell <- cm[, "T cell"]

boxplot(log(tcell) ~ x,
        xlab="Predicted group of Brodalumab benefit",
        ylab="T cell abundance (log)")

summary(lm(log(tcell) ~ x[names(tcell)]))

##
## Call:
## lm(formula = log(tcell) ~ x[names(tcell)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09149 -0.03431 -0.00662  0.03424  0.16257
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.575589    0.006669  236.267  <2e-16 ***
```

```
## x[names(tcell)]High benefit 0.022780 0.010599 2.149 0.0342 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05079 on 94 degrees of freedom
## Multiple R-squared:  0.04684, Adjusted R-squared:  0.0367
## F-statistic: 4.619 on 1 and 94 DF,  p-value: 0.03419
```



Etanercept study

We downloaded data from an etanercept study from GEO with accession number GSE11903. We retrieved transcriptomic and treatment response data for non-lesional skin at baseline.


```

gsms1 <- getGEO("GSE11903", destdir = "./data", AnnotGPL = TRUE)
phenobb <- pData(phenoData(gsms1[[1]]))

#patient and sample IDs
patient <- sapply(strsplit(phenobb$title, "_"), function(x) x[[1]])
id <- rownames(phenobb)

#time of visit
visit <- phenobb$Time:ch1

#clinical data
eff <- as.numeric(factor(phenobb$Group:ch1))-1
selbase <- visit=="0" & phenobb$Condition:ch1=="non-lesional"
phenost1 <- data.frame(patient=patient, id=id, eff=eff)[selbase,]

rownames(phenost1) <- phenost1$id
phenost1 <- phenost1[complete.cases(phenost1),]

```

We observe 11 patients that responded after 12 weeks to the weekly administration of 50mg of etanercept

```

head(phenost1)

##          patient      id eff
## GSM300749      A GSM300749  1
## GSM300755      B GSM300755  0
## GSM300761      C GSM300761  1
## GSM300767      D GSM300767  1
## GSM300773      E GSM300773  1
## GSM300779      F GSM300779  0

table(phenost1$eff)

##
##  0  1
##  4 11

```

Transcriptomic data of non-lesional skin at baseline was collected with Affymetrix Human Genome U133A 2.0 Array.

```

genesIDs <- fData(gsms1[[1]])

#obtain transcriptomic data, store in expr
expr <- exprs(gsms1[[1]])
expr <- expr[,rownames(phenost1)]

genesidS1 <- sapply(strsplit(genesIDs$Gene symbol, "/"), function(x) x[1])
names(genesidS1) <- rownames(genesIDs)

rownames(expr) <- genesidS1

dim(expr)

## [1] 22277      15

```

We used transcriptomic data to infer T-cell abundance in non-lesional skin at baseline using `mcp_counter` and fitted a regression model of response to treatment of the log-T cell levels.

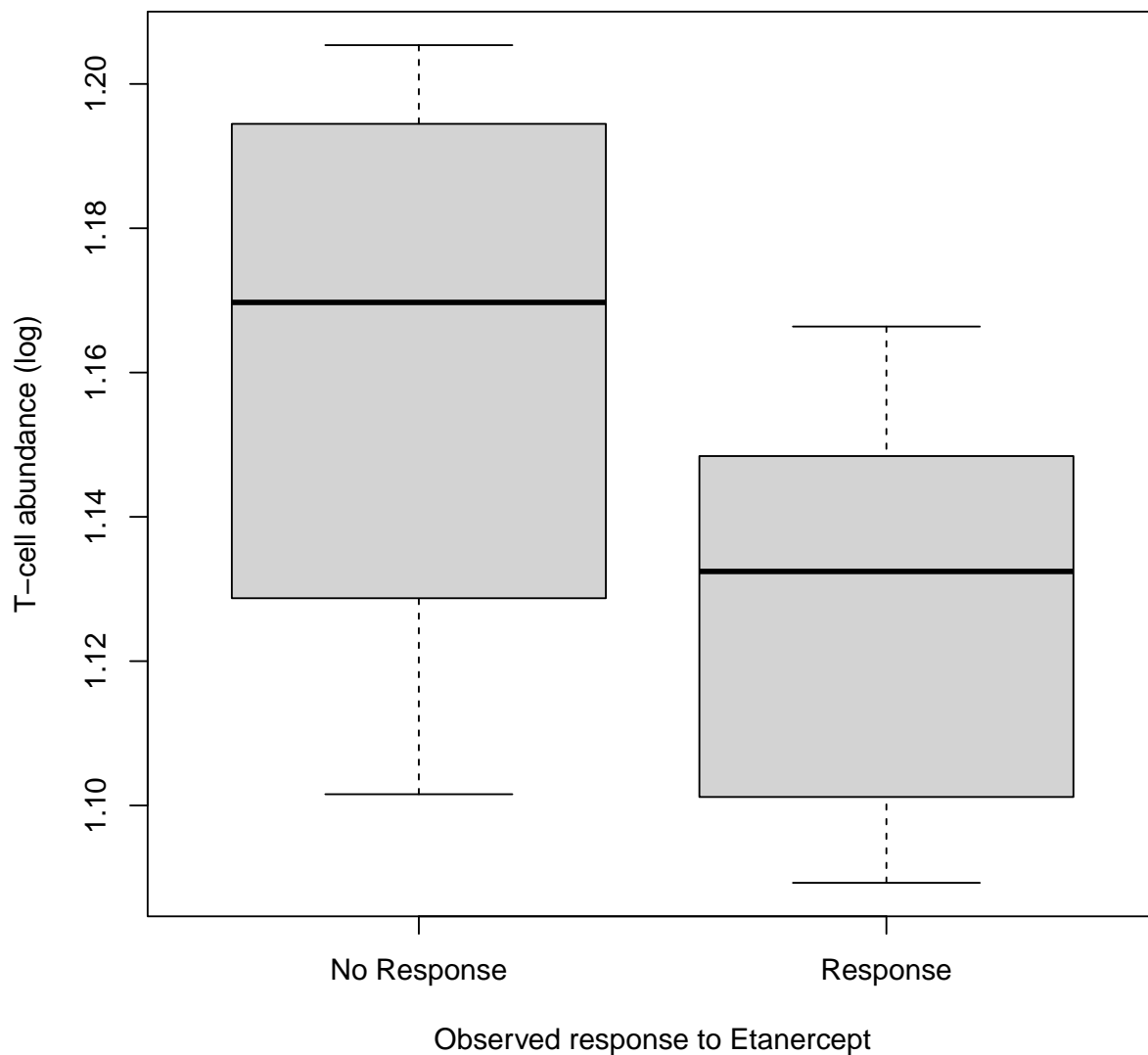
```
cellcomp2 <- deconvolute(expr, "mcp_counter", arrays=TRUE, column = "Symbol")
cellnames <- cellcomp2$cell_type
cm <- matrix(as.numeric(t(cellcomp2)[-1,]), ncol=length(cellnames))
colnames(cm) <- cellnames
rownames(cm) <- colnames(cellcomp2)[-1]
tcell <- cm[, "T cell"]

phenost1$tcell <- tcell
y <- factor(phenost1$eff, labels=c("No Response", "Response"))

boxplot(log(tcell) ~ y, ylab="T-cell abundance (log)", xlab="Observed response to Etanercept")

summary(glm(log(tcell) ~ y))

##
## Call:
## glm(formula = log(tcell) ~ y)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06004  -0.02606   0.00520   0.02222   0.04378
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.16159    0.01573  73.851  <2e-16 ***
## yResponse    -0.03436    0.01837  -1.871   0.0841 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.0009895992)
##
##      Null deviance: 0.016328  on 14  degrees of freedom
## Residual deviance: 0.012865  on 13  degrees of freedom
## AIC: -57.352
##
## Number of Fisher Scoring iterations: 2
```



We formatted data for targeting individuals with high brodalumab benefit, using the profile from the GSE117468 study

```
#compute SVAs
mod0 <- model.matrix( ~ 1, data = phenost1)
mod <- model.matrix( ~ tcell, data = phenost1)
ns <- num.sv(expr, mod, method="be")
ss <- sva(expr, mod, mod0, n.sv=ns)$sv

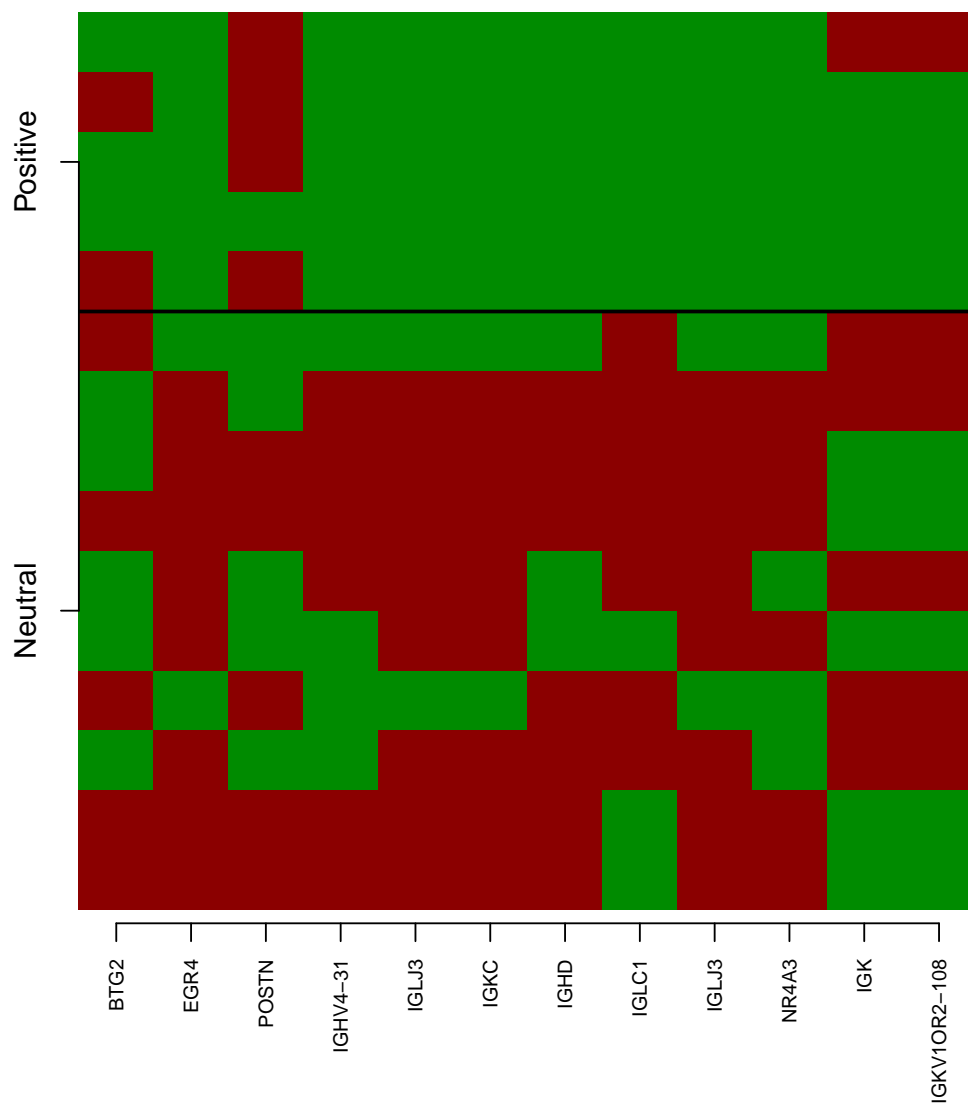
## Number of significant surrogate variables is: 4
## Iteration (out of 5):1 2 3 4 5

modss <- cbind(mod, ss)
teffdata <- modss
```

```
colnames(teffdata) <- c("t","eff", paste("cov",1:(ncol(teffdata)-2), sep=""))
rownames(expr) <- names(genesidS1)
study1 <- list(teffdata=teffdata, features=t(expr))
```

We selected common transcript IDS in the brodalumab profile (Manuscript Table 3) and the etanercept study. We targeted individuals with available transcripts and classified them into high and low brodalumab benefit at baseline if they matched the profile in more than 60% of the transcripts.

```
nmf <- colnames(study1$features)
nmf <- nmf[nmf%in%colnames(pso$profile$profpositive)]
l1 <- genesid[nmf]
l1[is.na(l1)] <- names(l1)[is.na(l1)]
res <- target(study1, pso, plot=TRUE, effect="positive", match=0.6, model=NULL, lb=l1)
```



We tested the association between the targeting and response to etanercept treatment

```
library("epiR")

y <- phenost1$eff
x <- res$classification

tb <- table(x,y)

fisher.test(tb)

##
## Fisher's Exact Test for Count Data
##
## data:  tb
```

```
## p-value = 0.07692
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.001283895 1.806255144
## sample estimates:
## odds ratio
## 0.09381563

epi.tests(table(x,as.numeric(y==0)), conf.level = 0.95)

##           Outcome +      Outcome -      Total
## Test +           9          1          10
## Test -           2          3           5
## Total           11          4          15
##
## Point estimates and 95% CIs:
## -----
## Apparent prevalence           0.67 (0.38, 0.88)
## True prevalence               0.73 (0.45, 0.92)
## Sensitivity                   0.82 (0.48, 0.98)
## Specificity                   0.75 (0.19, 0.99)
## Positive predictive value     0.90 (0.55, 1.00)
## Negative predictive value     0.60 (0.15, 0.95)
## Positive likelihood ratio     3.27 (0.59, 18.28)
## Negative likelihood ratio     0.24 (0.06, 0.96)
## -----
```

We finally fitted a logistic regression model of brodalumab benefit and T-cell abundancy in non-lesional skin at baseline on the observed response to a 12-week treatment with etarnecept. We computed the likelihood ratio test and the variance explained by the model ($R^2 = 0.751$) with the function `lrm` from `rms`.

```
library(rms)
mod <- lrm(y ~ x + tcell, x=TRUE)
mod

## Logistic Regression Model
##
## lrm(formula = y ~ x + tcell, x = TRUE)
##
##           Model Likelihood      Discrimination      Rank Discrim.
##           Ratio Test           Indexes           Indexes
## Obs           15    LR chi2      10.08    R2           0.713    C           0.977
## 0              4    d.f.           2      g           3.946    Dxy          0.955
## 1             11    Pr(> chi2) 0.0065    gr          51.729    gamma         0.955
## max |deriv| 0.0002      gp          0.375    tau-a         0.400
##           Brier          0.083
##
##           Coef      S.E.    Wald Z Pr(>|Z|)
## Intercept  73.1799 40.4613  1.81  0.0705
## x          -5.0954  2.9600 -1.72  0.0852
## tcell      -22.1948 12.4613 -1.78  0.0749
##
prob <- predict(mod, type="fitted.ind")
```

```

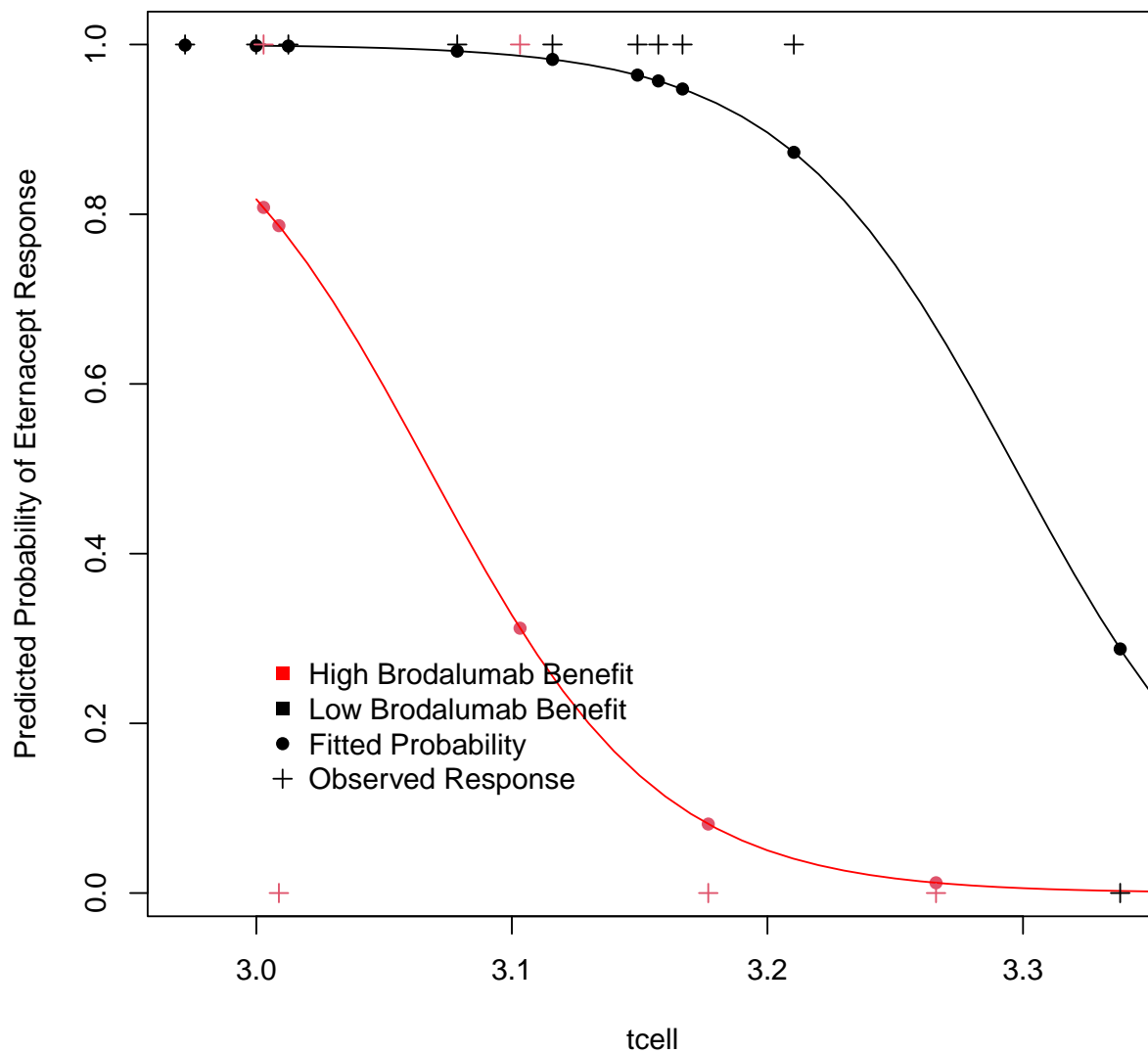
d1 <- data.frame(tcell=seq(3,4,0.01),x=0)
l1 <- predict(mod, d1, type="fitted.ind")

d2 <- data.frame(tcell=seq(3,4,0.01),x=1)
l2 <- predict(mod, d2, type="fitted.ind")

plot(tcell, prob, col=x+1, pch=16, ylab="Predicted Probability of Eternacept Response")
lines(seq(3,4,0.01), l1)
lines(seq(3,4,0.01), l2, col="red")
points(tcell, y, col=x+1, pch=3)

legend(3,0.3,
      legend = c("High Brodalumab Benefit", "Low Brodalumab Benefit",
                  "Fitted Probability", "Observed Response"),
      col=c("red", "black", "black", "black"),
      pch=c(15,15,16,3),
      bty = "n")

```



```
sessionInfo()

## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Spanish_Spain.1252 LC_CTYPE=Spanish_Spain.1252
## [3] LC_MONETARY=Spanish_Spain.1252 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Spain.1252
##
## attached base packages:
```



```

## [1] stats4      parallel  stats      graphics  grDevices utils      datasets  methods
## [9] base
##
## other attached packages:
## [1] rms_6.2-0           SparseM_1.81          Hmisc_4.5-0
## [4] Formula_1.2-4       lattice_0.20-45       epiR_2.0.36
## [7] survival_3.2-13     immunedeconv_2.0.4    EPIC_1.1.5
## [10] arm_1.11-2          lme4_1.1-27.1         Matrix_1.3-4
## [13] drc_3.0-1           MASS_7.3-54           teff_0.1.0
## [16] org.Hs.eg.db_3.13.0 AnnotationDbi_1.54.1   IRanges_2.26.0
## [19] S4Vectors_0.30.1    clusterProfiler_4.0.5 vioplot_0.3.7
## [22] zoo_1.8-9           sm_2.2-5.7            xtable_1.8-4
## [25] EnhancedVolcano_1.10.0 ggrepel_0.9.1         ggplot2_3.3.5
## [28] limma_3.48.3        sva_3.40.0            BiocParallel_1.26.2
## [31] genefilter_1.74.0    mgcv_1.8-37           nlme_3.1-153
## [34] GEOquery_2.60.0      Biobase_2.52.0        BiocGenerics_0.38.0
## [37] knitr_1.36
##
## loaded via a namespace (and not attached):
## [1] Rsamtools_2.8.0      foreach_1.5.1         lmtest_0.9-38
## [4] crayon_1.4.1         rhdf5filters_1.4.0    backports_1.2.1
## [7] GOSemSim_2.18.1      rlang_0.4.11          XVector_0.32.0
## [10] readxl_1.3.1         nloptr_1.2.2.2        extrafontdb_1.0
## [13] minfi_1.38.0         filelock_1.0.2        data.tree_1.0.0
## [16] extrafont_0.17       rjson_0.2.20          bit64_4.0.5
## [19] glue_1.4.2           rngtools_1.5.2        vipor_0.4.5
## [22] DOSE_3.18.2          haven_2.4.3           tidyselect_1.1.1
## [25] SummarizedExperiment_1.22.0 rio_0.5.27            XML_3.99-0.8
## [28] tidyr_1.1.4          proj4_1.0-10.1         ggpubr_0.4.0
## [31] GenomicAlignments_1.28.0 MatrixModels_0.5-0    magrittr_2.0.1
## [34] evaluate_0.14        cli_3.0.1             zlibbioc_1.38.0
## [37] rstudioapi_0.13      doRNG_1.8.2           sp_1.4-5
## [40] rpart_4.1-15         betareg_3.1-4         fastmatch_1.1-3
## [43] treeio_1.16.2        maps_3.4.0            xfun_0.26
## [46] askpass_1.1          multtest_2.48.0       cluster_2.1.2
## [49] tidygraph_1.2.0      KEGGREST_1.32.0       quantreg_5.86
## [52] tibble_3.1.5         lpSolve_5.6.15        base64_2.0
## [55] ape_5.5              scime_1.3.5           Biostrings_2.60.2
## [58] png_0.1-7            reshape_0.8.8         withr_2.4.2
## [61] bitops_1.0-7         ggforce_0.3.3         plyr_1.8.6
## [64] cellranger_1.1.0     coda_0.19-4           pillar_1.6.3
## [67] bumpHunter_1.34.0    cachem_1.0.6          GenomicFeatures_1.44.2
## [70] multcomp_1.4-17      flexmix_2.3-17        raster_3.4-13
## [73] DelayedMatrixStats_1.14.3 vctrs_0.3.8           ellipsis_0.3.2
## [76] generics_0.1.0       tools_4.1.1           foreign_0.8-81
## [79] beeswarm_0.4.0       munsell_0.5.0         tweenr_1.0.2
## [82] fgsea_1.18.0         DelayedArray_0.18.0   fastmap_1.1.0
## [85] compiler_4.1.1       abind_1.4-5           rtracklayer_1.52.1
## [88] beanplot_1.2         GenomeInfoDbData_1.2.6 gridExtra_2.3
## [91] edgeR_3.34.1         utf8_1.2.2            dplyr_1.0.7
## [94] BiocFileCache_2.0.0  jsonlite_1.7.2        scales_1.1.1
## [97] tidytree_0.3.5       carData_3.0-4         sparseMatrixStats_1.4.2
## [100] lazyeval_0.2.2       car_3.0-11            latticeExtra_0.6-29

```

## [103] checkmate_2.0.0	openxlsx_4.2.4	ash_1.0-15
## [106] nor1mix_1.3-0	sandwich_3.0-1	cowplot_1.1.1
## [109] siggenes_1.66.0	forcats_0.5.1	pander_0.6.4
## [112] downloader_0.4	igraph_1.2.6	HDF5Array_1.20.0
## [115] yaml_2.2.1	plotrix_3.8-2	htmltools_0.5.2
## [118] memoise_2.0.0	modeltools_0.2-23	BiocIO_1.2.0
## [121] locfit_1.5-9.4	graphlayouts_0.7.1	quadprog_1.5-8
## [124] viridisLite_0.4.0	digest_0.6.28	assertthat_0.2.1
## [127] rappdirs_0.3.3	Rttf2pt1_1.3.9	BiasedUrn_1.07
## [130] RSQLite_2.2.8	yulab.utils_0.0.2	data.table_1.14.2
## [133] testit_0.13	blob_1.2.2	preprocessCore_1.54.0
## [136] splines_4.1.1	labeling_0.4.2	Rhdf5lib_1.14.2
## [139] illuminaio_0.34.0	RCurl_1.98-1.5	broom_0.7.9
## [142] hms_1.1.1	rhdf5_2.36.0	colorspace_2.0-2
## [145] base64enc_0.1-3	ggbeeswarm_0.6.0	GenomicRanges_1.44.0
## [148] aplot_0.1.1	ggrastr_0.2.3	nnet_7.3-16
## [151] Rcpp_1.0.7	mclust_5.4.7	mvtnorm_1.1-2
## [154] enrichplot_1.12.2	fansi_0.5.0	conquer_1.0.2
## [157] tzdb_0.1.2	R6_2.5.1	grid_4.1.1
## [160] polyspline_1.1.19	lifecycle_1.0.1	zip_2.2.0
## [163] curl_4.3.2	ggsignif_0.6.3	minqa_1.2.4
## [166] limSolve_1.5.6	D0.db_2.9	qvalue_2.24.0
## [169] TH.data_1.1-0	RColorBrewer_1.1-2	iterators_1.0.13
## [172] stringr_1.4.0	htmlwidgets_1.5.4	polyclip_1.10-0
## [175] biomaRt_2.48.3	purrr_0.3.4	MCPcounter_1.2.0
## [178] shadowtext_0.0.9	gridGraphics_0.5-1	openssl_1.4.5
## [181] htmlTable_2.2.1	patchwork_1.1.1	lubridate_1.7.10
## [184] codetools_0.2-18	matrixStats_0.61.0	G0.db_3.13.0
## [187] gtools_3.9.2	prettyunits_1.1.1	dbplyr_2.1.1
## [190] GenomeInfoDb_1.28.4	grf_2.0.2	gtable_0.3.0
## [193] DBI_1.1.1	ggfun_0.0.4	httr_1.4.2
## [196] highr_0.9	KernSmooth_2.23-20	stringi_1.7.5
## [199] vroom_1.5.5	progress_1.2.2	reshape2_1.4.4
## [202] farver_2.1.0	annotate_1.70.0	viridis_0.6.1
## [205] ggtree_3.0.4	xml2_1.3.2	boot_1.3-28
## [208] ggalt_0.4.0	restfulr_0.0.13	readr_2.0.2
## [211] ggplotify_0.1.0	bit_4.0.4	scatterpie_0.1.7
## [214] jpeg_0.1-9	MatrixGenerics_1.4.3	ggraph_2.0.5
## [217] pkgconfig_2.0.3	rstatix_0.7.0	