Supplemental code file for manuscript titled: Variable gene expression and parasite load predict treatment outcome in cutaneous leishmaniasis

Camila Farias Amorim, Fernanda O. Novais, Ba T. Nguyen, Ana M. Misic, Lucas P. Carvalho, Edgar M. Carvalho, Daniel P. Beiting, Phillip Scott

produced on 2019-08-22

Contents

Background

Patients infected with *Leishmania braziliensis* develop chronic lesions that often fail to respond to treatment with anti-parasite drugs. To determine whether genes whose expression is highly variable in lesions between patients might influence disease outcome, we obtained biopsies of lesions from patients prior to treatment with pentavalent antimony, performed transcriptomic profiling, and identified highly variable genes whose expression correlated with treatment outcome. Amongst the most variable genes in all the patients were components of the cytolytic pathway, the expression of which correlated with parasite load in the skin. We demonstrated that treatment failure was linked to the cytolytic pathway activated during infection. Using a host-pathogen biomarker profile ofof as few as 3x genes, we showed that eventual treatment outcome could be predicted before the start of treatment in two separate patient cohorts (n=46 total). These findings raise the possibility of point-of-care diagnostic screening to identify patients at high risk of treatment failure and provide a rationale for a precision medicine approach to drug selection in cutaneous leishmaniasis, and more broadly demonstrate the value of identifying genes of high variability in other diseases to better understand and predict diverse clinical outcomes.

The code below shows how raw data was processed, mapped, and analyzed to identify potential biomarkers for treatment outcome in Cutaneous Leishmaniasis. This markdown report is configured to be compiled as a PDF, but if the yaml header above is replaced with the header in the 'html_yaml.txt' located in the same directory as this markdown document, then it can be compiled as an HTML file.

Reproducibility and accessibility

Raw fastq files are available from the Gene Expression Omnibus, under accession GSE127831, but are not needed for reproducing the analysis. Findings from our study were validated using a second publicly available dataset obtained from the Sequence Read Archive under bioproject PRJNA307599, which was described previously in Christensen et al, PLOS NTD, 2016. Note that 6 samples from this bioproject (SRR7275002, SRR7275003, SRR7275004, SRR7275005, SRR7275006, SRR7275007) are actually from a separate study of L. amazonensis and disseminated cutaneous leishmaniasis, and therefore were exlcuded from this analysis. In addition, 1 sample (SRR3162874) from Christensen et al., had no information available for treatment outcome, and therefore was also exclued from the study. Prealigned data and all code used in this analysis, including the Rmarkdown document used to compile this supplementary code file, are all available on GitHub **here**. The Github version of this document reflects the most up-to-date and comprehensive analysis, and as such it may differ slightly from the one included as a supplementary file in the manuscript. Once this GitHub repo has been downloaded, navigate to `/Amorim_CutaneousLeish_biomarkers/ANALYSIS/code` to find the Rmarkdown document as well as an RProject file.

R packages used for this analysis

```
library(tidyverse)
library(ggthemes)
library(reshape2)
```

```r
library(edgeR)
library(patchwork)
library(vegan)
library(DT)
library(tximport)
library(gplots)
library(FinCal)
library(ggrepel)
library(gt)
library(ggExtra)
library(EnsDb.Hsapiens.v86)
library(stringr)
library(cowplot)
library(ggpubr)
```

Preprocessing of raw reads

mapping reads to the human transcriptome

Quality control of raw reads was carried out using fastqc. Raw reads were mapped to the *Homo sapiens* reference transcriptome available on Ensembl here using Kallisto, version 0.46. The quality of raw reads, as well as the results of Kallisto mapping were summarized using multiqc. The resulting multiqc report can be found in the github project repo in the `QA` directory. Due to size limitations imposed by GitHub, neither the raw fastq files, nor the reference fasta file used for read mapping could be stored in the GitHub repo.

```
# build index from reference fasta from Ensembl Homo sapiens transcriptome
kallisto index -i Homo_sapiens.GRCh38.cdna.all.Index Homo_sapiens.GRCh38.cdna.all.fa

# Map reads to the indexed reference transcriptome for HOST
# first the healthy subjects (HS)
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_HS01 -t 24 -b 60 --single -l 250 -s 30 HS1
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_HS02 -t 24 -b 60 --single -l 250 -s 30 HS2
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_HS03 -t 24 -b 60 --single -l 250 -s 30 HS3
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_HS04 -t 24 -b 60 --single -l 250 -s 30 HS4
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_HS05 -t 24 -b 60 --single -l 250 -s 30 HS5
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_HS06 -t 24 -b 60 --single -l 250 -s 30 HS6
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_HS07 -t 24 -b 60 --single -l 250 -s 30 HS7
# then the cutaneous leishmaniasis (CL) patients
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL01 -t 24 -b 60 --single -l 250 -s 30 CL1
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL02 -t 24 -b 60 --single -l 250 -s 30 CL2
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL03 -t 24 -b 60 --single -l 250 -s 30 CL3
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL04 -t 24 -b 60 --single -l 250 -s 30 CL4
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL05 -t 24 -b 60 --single -l 250 -s 30 CL5
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL06 -t 24 -b 60 --single -l 250 -s 30 CL6
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL07 -t 24 -b 60 --single -l 250 -s 30 CL7
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL08 -t 24 -b 60 --single -l 250 -s 30 CL8
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL09 -t 24 -b 60 --single -l 250 -s 30 CL9
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL10 -t 24 -b 60 --single -l 250 -s 30 CL10
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL11 -t 24 -b 60 --single -l 250 -s 30 CL11
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL12 -t 24 -b 60 --single -l 250 -s 30 CL12
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL13 -t 24 -b 60 --single -l 250 -s 30 CL13
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL14 -t 24 -b 60 --single -l 250 -s 30 CL14
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL15 -t 24 -b 60 --single -l 250 -s 30 CL15
```

```
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL16 -t 24 -b 60 --single -l 250 -s 30 CL1
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL17 -t 24 -b 60 --single -l 250 -s 30 CL1
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL18 -t 24 -b 60 --single -l 250 -s 30 CL1
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL19 -t 24 -b 60 --single -l 250 -s 30 CL1
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL20 -t 24 -b 60 --single -l 250 -s 30 CL2
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_CL21 -t 24 -b 60 --single -l 250 -s 30 CL2
```

Using R/bioconductor to import RNAseq data

Sample info - Table S2

```
import <- read_tsv("studydesign.txt")
import %>% dplyr::filter(disease == "cutaneous") %>%
  dplyr::select(-2) %>%  gt() %>%
  tab_header(title = md("Clinical metadata from patients with cutaneous leishmaniasis (CL)"),
             subtitle = md("`(n=21)`")) %>%  cols_align(align = "center", columns = TRUE)
```

Clinical metadata from patients with cutaneous leishmaniasis (CL)

(n=21)

sample	treatment_outcome	age_(years)	sex	DTH_(mm2)	lesion_size_(mm2)	Time_to_cure_(days)
host_CL01	failure	35	F	255	79	150
host_CL02	failure	21	M	228	613	130
host_CL03	failure	29	M	180	393	150
host_CL04	cure	37	M	300	19	40
host_CL05	cure	50	M	272	212	80
host_CL06	cure	26	F	285	102	90
host_CL07	cure	28	F	272	163	45
host_CL08	cure	63	M	196	214	62
host_CL09	cure	59	M	255	267	70
host_CL10	cure	22	M	255	657	75
host_CL11	failure	49	M	240	13	140
host_CL12	cure	42	M	225	141	54
host_CL13	cure	69	M	150	177	60
host_CL14	cure	19	M	208	377	45
host_CL15	failure	18	M	130	151	150
host_CL16	failure	46	M	150	431	135
host_CL17	cure	35	M	270	20	90
host_CL18	failure	51	M	225	636	203
host_CL19	cure	25	M	180	118	60
host_CL20	cure	20	F	100	63	80
host_CL21	cure	55	M	1085	1237	70

```
targets.lesion <- import
targets.onlypatients <- targets.lesion[8:28,] # only CL lesions (n=21)

# Making factors that will be used for pairwise comparisons:
# HS vs. CL lesions as a factor:
disease.lesion <- factor(targets.lesion$disease)
# Cure vs. Failure lesions as a factor:
treatment.lesion <- factor(targets.onlypatients$treatment_outcome)
```

Annotation

The tximport package was used to read Kallisto outputs into R environment.

```
# capturing Ensembl transcript IDs (tx) and gene symbols ("gene_name") from EnsDb.Hsapiens.v86 annotati
Tx <- as.data.frame(transcripts(EnsDb.Hsapiens.v86,
                                columns=c(listColumns(EnsDb.Hsapiens.v86, "tx"),
                                          "gene_name")))

Tx <- dplyr::rename(Tx, target_id = tx_id)
row.names(Tx) <- NULL
Tx <- Tx[,c(6,12)]

# getting file paths for Kallisto outputs
paths.all <- file.path("../readMapping/human", targets.lesion$sample, "abundance.h5")
paths.patients <- file.path("../readMapping/human", targets.onlypatients$sample, "abundance.h5")

# importing .h5 Kallisto data and collapsing transcript-level data to genes
Txi.lesion.coding <- tximport(paths.all,
                              type = "kallisto",
                              tx2gene = Tx,
                              txOut = FALSE,
                              ignoreTxVersion = TRUE,
                              countsFromAbundance = "lengthScaledTPM")

# importing againg, but this time just the CL patients
Txi.lesion.coding.onlypatients <- tximport(paths.patients,
                                           type = "kallisto",
                                           tx2gene = Tx,
                                           txOut = FALSE,
                                           ignoreTxVersion = TRUE,
                                           countsFromAbundance = "lengthScaledTPM")
```

Identifying host gene expression associated with treatment failure

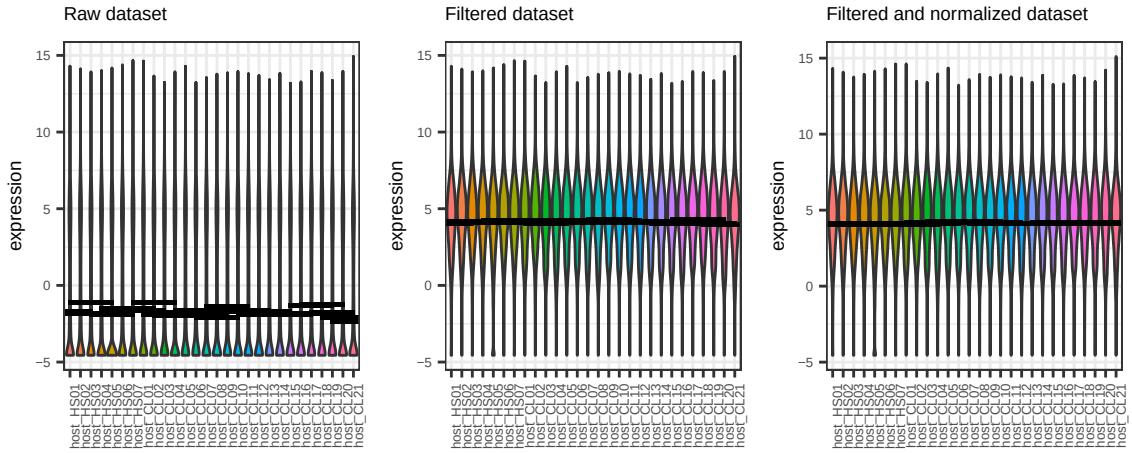
Gene-level counts were converted to counts per million (CPM), filtered to keep only genes with >1 CPM in >= 7 samples, and then normalized using the Trimmed Mean of M-values (TMM method) in the EdgeR package.

filtering and normalization

```
# First make a DGEList from the counts:
Txi.lesion.coding.DGEList <- DGEList(Txi.lesion.coding$counts)
colnames(Txi.lesion.coding.DGEList$counts) <- targets.lesion$sample
colnames(Txi.lesion.coding$counts) <- targets.lesion$sample
#write.table(Txi.lesion.coding$counts, "Amorim_GEO_raw.txt", sep = "\t", quote = FALSE)

Txi.lesion.coding.DGEList.OP <- DGEList(Txi.lesion.coding.onlypatients$counts)
colnames(Txi.lesion.coding.DGEList.OP) <- targets.onlypatients$sample

# Convert to counts per million:
Txi.lesion.coding.DGEList.cpm <- edgeR::cpm(Txi.lesion.coding.DGEList, log = TRUE)
Txi.lesion.coding.DGEList.OP.cpm <- edgeR::cpm(Txi.lesion.coding.DGEList.OP, log = TRUE)
```

```
keepers.coding <- rowSums(Txi.lesion.coding.DGEList.cpm>1)>=7
keepers.coding.OP <- rowSums(Txi.lesion.coding.DGEList.OP.cpm>1)>=7

Txi.lesion.coding.DGEList.filtered <- Txi.lesion.coding.DGEList[keepers.coding,]
Txi.lesion.coding.DGEList.OP.filtered <- Txi.lesion.coding.DGEList.OP[keepers.coding.OP,]

# convert back to cpm:
Txi.lesion.coding.DGEList.LogCPM.filtered <- edgeR::cpm(Txi.lesion.coding.DGEList.filtered,
                                                        log=TRUE)
Txi.lesion.coding.DGEList.LogCPM.OP.filtered <- edgeR::cpm(Txi.lesion.coding.DGEList.OP.filtered,
                                                           log=TRUE)

# Normalizing data:
calcNorm1 <- calcNormFactors(Txi.lesion.coding.DGEList.filtered, method = "TMM")
calcNorm2 <- calcNormFactors(Txi.lesion.coding.DGEList.OP.filtered, method = "TMM")

Txi.lesion.coding.DGEList.LogCPM.filtered.norm <- edgeR::cpm(calcNorm1, log=TRUE)
colnames(Txi.lesion.coding.DGEList.LogCPM.filtered.norm) <- targets.lesion$sample
Txi.lesion.coding.DGEList.OP.LogCPM.filtered.norm <- edgeR::cpm(calcNorm2, log=TRUE)
colnames(Txi.lesion.coding.DGEList.OP.LogCPM.filtered.norm) <- targets.onlypatients$sample
# Raw dataset:
V1 <- as.data.frame(Txi.lesion.coding.DGEList.cpm)
colnames(V1) <- targets.lesion$sample
V1 <- melt(V1)
colnames(V1) <- c("sample","expression")

# Filtered dataset:
V1.1 <- as.data.frame(Txi.lesion.coding.DGEList.LogCPM.filtered)
colnames(V1.1) <- targets.lesion$sample
V1.1 <- melt(V1.1)
colnames(V1.1) <- c("sample","expression")

# Filtered-normalized dataset:
V1.1.1 <- as.data.frame(Txi.lesion.coding.DGEList.LogCPM.filtered.norm)
colnames(V1.1.1) <- targets.lesion$sample
V1.1.1 <- melt(V1.1.1)
colnames(V1.1.1) <- c("sample","expression")

# plotting:
ggplot(V1, aes(x=sample, y=expression, fill=sample)) +
  geom_violin(trim = TRUE, show.legend = TRUE) +
  stat_summary(fun.y = "median", geom = "point", shape = 95, size = 10, color = "black") +
  theme_bw() +
  theme(legend.position = "none", axis.title=element_text(size=7),
        axis.title.x=element_blank(), axis.text=element_text(size=5),
        axis.text.x = element_text(angle = 90, hjust = 1),
        plot.title = element_text(size = 7)) +
  ggtitle("Raw dataset") +
ggplot(V1.1, aes(x=sample, y=expression, fill=sample)) +
  geom_violin(trim = TRUE, show.legend = TRUE) +
  stat_summary(fun.y = "median", geom = "point", shape = 95, size = 10, color = "black") +
  theme_bw() +
  theme(legend.position = "none", axis.title=element_text(size=7),
```

```
        axis.title.x=element_blank(), axis.text=element_text(size=5),
        axis.text.x = element_text(angle = 90, hjust = 1),
        plot.title = element_text(size = 7)) +
  ggtitle("Filtered dataset") +
ggplot(V1.1.1, aes(x=sample, y=expression, fill=sample)) +
  geom_violin(trim = TRUE, show.legend = TRUE) +
  stat_summary(fun.y = "median", geom = "point", shape = 95, size = 10, color = "black") +
  theme_bw() +
  theme(legend.position = "none", axis.title=element_text(size=7),
        axis.title.x=element_blank(), axis.text=element_text(size=5),
        axis.text.x = element_text(angle = 90, hjust = 1),
        plot.title = element_text(size = 7)) +
  ggtitle("Filtered and normalized dataset")
```



unfiltered data

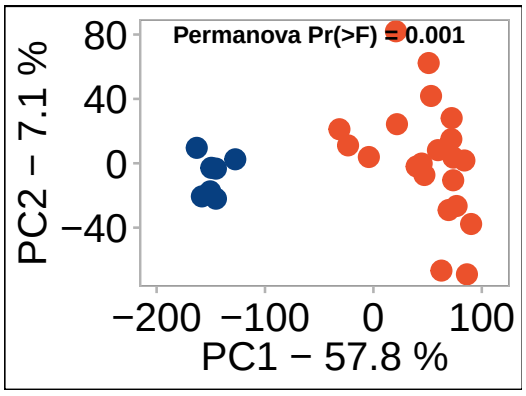
In this session we are creating a normalized dataset with gene counts (in counts per million, CPM) including all the genes aligned to the human reference (not filtering the dataset).
The object **CPM_normData_notfiltered_OP** will be used for:
1) perform differential gene expression analysis between Failure vs. Cure patients to obtain genes with significant P. values and relative fold changes ; 2) the *Variability* and *Treatment outcome* analysis;
3) as dataframe to make plots of gene expression in GraphPad Prism.

```
DataNotFiltered_Norm_OP <- calcNormFactors(Txi.lesion.coding.DGEList[,8:28],
                                           method = "TMM")
DataNotFiltered_Norm_log2CPM_OP <- edgeR::cpm(DataNotFiltered_Norm_OP, log=TRUE)
colnames(DataNotFiltered_Norm_log2CPM_OP) <- targets.onlypatients$sample
CPM_normData_notfiltered_OP <- 2^(DataNotFiltered_Norm_log2CPM_OP)
#write.table(Txi.lesion.coding$counts, "Amorim_GEO_raw.txt", sep = "\t", quote = FALSE)

# Including all the individuals (HS and CL patients) for public domain submission:
DataNotFiltered_Norm <- calcNormFactors(Txi.lesion.coding.DGEList, method = "TMM")
DataNotFiltered_Norm_log2CPM <- edgeR::cpm(DataNotFiltered_Norm, log=TRUE)
colnames(DataNotFiltered_Norm_log2CPM) <- targets.lesion$sample
CPM_normData_notfiltered <- 2^(DataNotFiltered_Norm_log2CPM)
#write.table(DataNotFiltered_Norm_log2CPM, "Amorim_GEO_normalized.txt", sep = "\t", quote = FALSE)
```

Exploratory analysis

PCA comparing infected to naive - Figure 1A

```
pca.res <- prcomp(t(Txi.lesion.coding.DGEList.LogCPM.filtered.norm), scale.=F, retx=T)
pc.var <- pca.res$sdev^2
pc.per <- round(pc.var/sum(pc.var)*100, 1)
data.frame <- as.data.frame(pca.res$x)

# Calculate distance between samples by permanova:
allsamples.dist <- vegdist(t(2^Txi.lesion.coding.DGEList.LogCPM.filtered.norm),
                           method = "bray")

vegan <- adonis2(allsamples.dist~targets.lesion$disease,
        data=targets.lesion,
        permutations = 999, method="bray")

ggplot(data.frame, aes(x=PC1, y=PC2, color=factor(targets.lesion$disease))) +
  geom_point(size=5, shape=20) +
  theme_calc() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        axis.text.x = element_text(size = 15, vjust = 0.5),
        axis.text.y = element_text(size = 15), axis.title = element_text(size = 15),
        legend.position="none") +
  scale_color_manual(values = c("#073F80","#EB512C")) +
  annotate("text", x=-50, y=80, label=paste("Permanova Pr(>F) =",
                                            vegan[1,5]), size=3, fontface="bold") +
  xlab(paste("PC1 -",pc.per[1],"%")) +
  ylab(paste("PC2 -",pc.per[2],"%")) +
  xlim(-200,110)
```



Differential gene expression analysis (DGE analysis)

```
# Model matrices:
# CL lesions vs. HS:
design.lesion <- model.matrix(~0 + disease.lesion)
colnames(design.lesion) <- levels(disease.lesion)

# Failure vs. Cure:
design.lesion.treatment <- model.matrix(~0 + treatment.lesion)
```

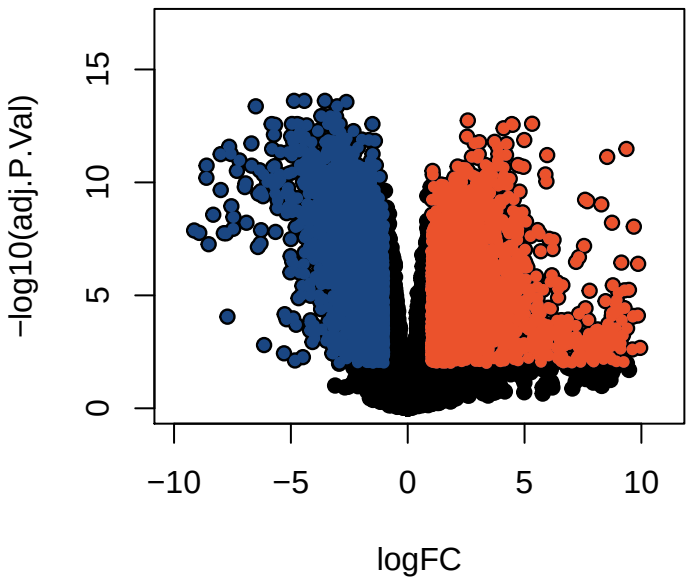
```
colnames(design.lesion.treatment) <- levels(treatment.lesion)

myDGEList.lesion.coding <- DGEList(calcNorm1$counts)
myDGEList.OP.NotFil <- DGEList(CPM_normData_notfiltered_OP)

# Model mean-variance trend and fit linear model to data.
# Use VOOM function from Limma package to model the mean-variance relationship
normData.lesion.coding <- voom(myDGEList.lesion.coding, design.lesion)
normData.OP.NotFil <- voom(myDGEList.OP.NotFil, design.lesion.treatment)

colnames(normData.lesion.coding) <- targets.lesion$sample
colnames(normData.OP.NotFil) <- targets.onlypatients$sample

# fit a linear model to your data
fit.lesion.coding <- lmFit(normData.lesion.coding, design.lesion)
fit.lesion.coding.treatment <- lmFit(normData.OP.NotFil, design.lesion.treatment)

# contrast matrix
contrast.matrix.lesion <- makeContrasts(CL.vs.CON = cutaneous - control,
                                        levels=design.lesion)
contrast.matrix.lesion.treat <- makeContrasts(failure.vs.cure = failure - cure,
                                              levels=design.lesion.treatment)

# extract the linear model fit
fits.lesion.coding <- contrasts.fit(fit.lesion.coding,
                                    contrast.matrix.lesion)
fits.lesion.coding.treat <- contrasts.fit(fit.lesion.coding.treatment,
                                          contrast.matrix.lesion.treat)

# get bayesian stats for your linear model fit
ebFit.lesion.coding <- eBayes(fits.lesion.coding)
ebFit.lesion.coding.treat <- eBayes(fits.lesion.coding.treat)

# TopTable ----
allHits.lesion.coding <- topTable(ebFit.lesion.coding,
                                  adjust ="BH", coef=1,
                                  number=34935, sort.by="logFC")
allHits.lesion.coding.treat <- topTable(ebFit.lesion.coding.treat,
                                        adjust ="BH", coef=1,
                                        number=34776, sort.by="logFC")
myTopHits <- rownames_to_column(allHits.lesion.coding, "geneID")
myTopHits.treat <- rownames_to_column(allHits.lesion.coding.treat, "geneID")

# mutate the format of numeric values:
myTopHits <- mutate(myTopHits, log10Pval = round(-log10(adj.P.Val),2),
                    adj.P.Val = round(adj.P.Val, 2),
                    B = round(B, 2),
                    AveExpr = round(AveExpr, 2),
                    t = round(t, 2),
                    logFC = round(logFC, 2),
                    geneID = geneID)

myTopHits.treat <- mutate(myTopHits.treat, log10Pval = round(-log10(adj.P.Val),2),
```

```
                    adj.P.Val = round(adj.P.Val, 2),
                    B = round(B, 2),
                    AveExpr = round(AveExpr, 2),
                    t = round(t, 2),
                    logFC = round(logFC, 2),
                    geneID = geneID)
#save(myTopHits, file = "myTopHits")
#save(myTopHits.treat, file = "myTopHits.treat")
```

Volcano Plot for CL vs. HS - Figure 1B

In this session, we visualize in a volcano plot the upregulated (orange) and downregulated (blue) genes in cutaneous leishmaniasis (CL) lesions relative to skin from healthy subjects (HS).

```
with(allHits.lesion.coding,
     plot(logFC, -log10(adj.P.Val), pch=19, cex=1,
          xlim=c(-10, 11), ylim=c(0, 17), main=""))
with(subset(allHits.lesion.coding, adj.P.Val<0.01 & abs(logFC)>1),
     points(logFC, -log10(adj.P.Val), pch=20, col="#1A4682"))
with(subset(allHits.lesion.coding, adj.P.Val<0.01 & (logFC)>1),
     points(logFC, -log10(adj.P.Val), pch=20, col="#EB522C"))
```

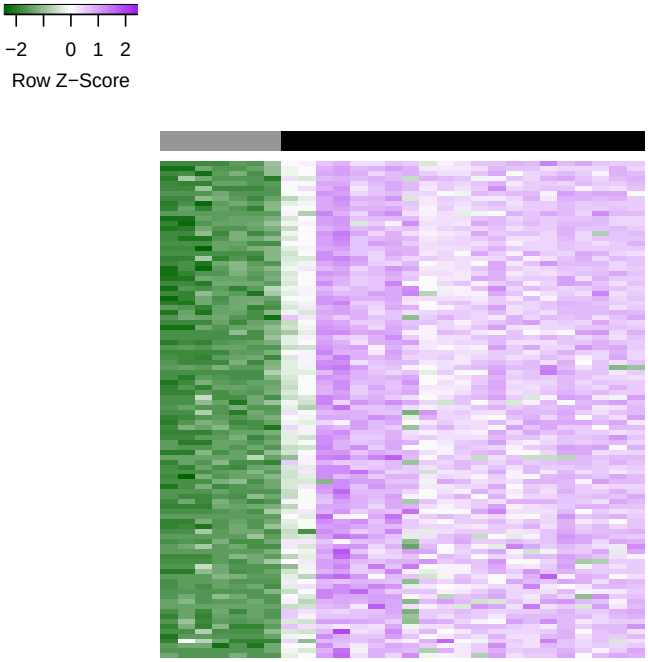


Top 100 upregulated genes in CL vs. HS - Figure S1

In this session, the top 100 upregulate genes in CL lesions relative to HS are represented in a heatmap.

```
color.map1 <- function(disease.lesion) { if (disease.lesion=="control") "#969696"
  else "#000000"} # coloring CL vs. HS
color.map1 <- unlist(lapply(disease.lesion, color.map1))
```

```
sortedupFC <- myTopHits[order(-myTopHits$logFC),]
rownames(sortedupFC) <- sortedupFC$geneID
sorteduptop <- sortedupFC[-grep("IG", sortedupFC$geneID),]

#Top 100 gene supregulated:
top100up <- sortedupFC[1:100,]$geneID
TopUPtable100 <- Txi.lesion.coding.DGEList.LogCPM.filtered.norm[c(top100up),]
TopUPmatrixcoding100 <- as.matrix(TopUPtable100)

colormapX <- colorRampPalette(colors=c("dark green","white","purple"))(50)

HeatmapUP100 <- heatmap.2(TopUPmatrixcoding100,
                          ColSideColors = color.map1,
                          scale = "row", key=TRUE,
                          keysize = 1, key.title = NA,
                          col=colormapX, dendrogram = "none", Rowv = F,
                          margins=c(5,25),
                          labCol = NA, labRow = NA,
                          main = "",
                          density.info="none", trace="none",
                          cexRow=0.8, cexCol=1)
```



Identification of ViTALs

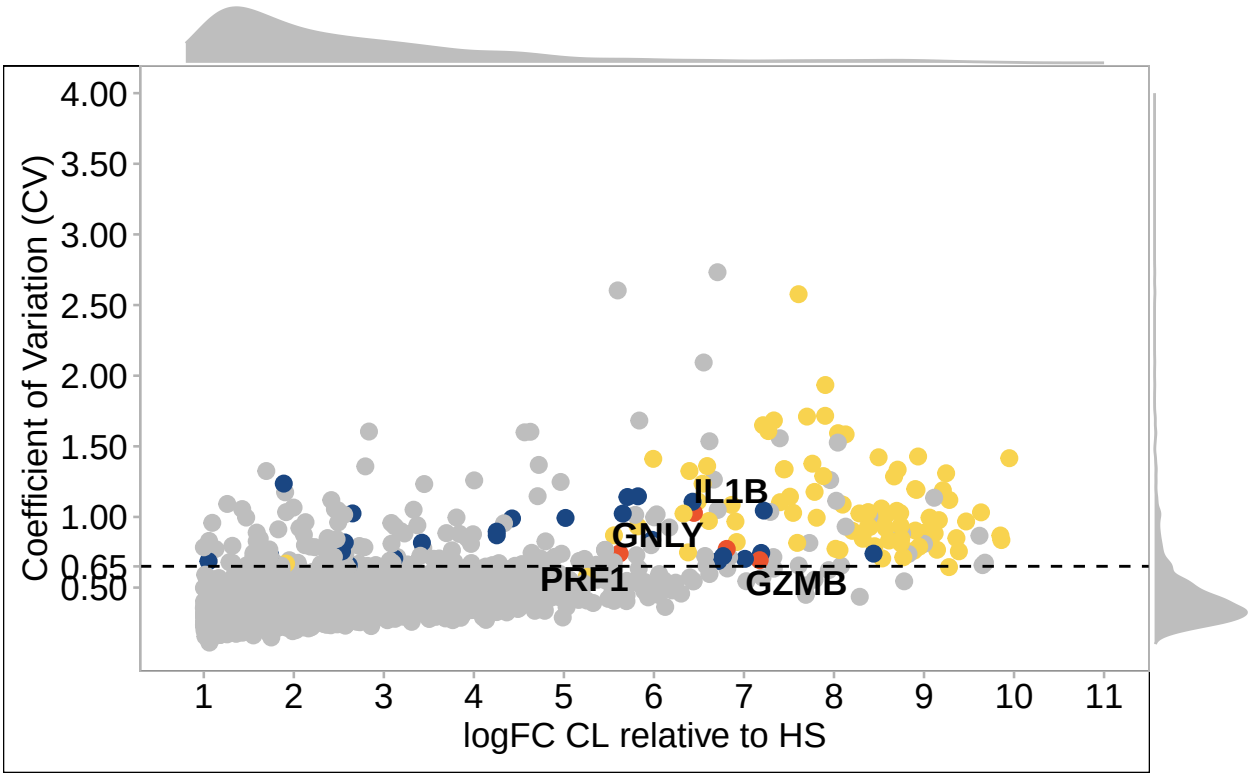
Using Coefficient of Variation (cV) of gene expression across CL patients to identify Transcripts Associated with Leishmaniasis (ViTALs)

Expression fold change vs. Coefficient of Variation - Figure 2B

Here, we explored the variability in gene expression between CL lesions. The strategy was to calculate the *Coefficient of Variation (cV)* for all the genes upregulated in the lesions. In this way, we can observe which gene had high expression variability between CL lesions (high variable genes, HVGenes).

```
# Filtering all the genes that were upregulated CL lesions vs. HS with
# more than FC = 2 and a FDR<0.01:
UpregulatedInCL <- subset(allHits.lesion.coding, logFC > 1 & adj.P.Val < 0.01)

# getting the expression levels of these genes for each CL sample:
UpregulatedInCL_names <- rownames(UpregulatedInCL) # names of upregulated genes in CL
all_genes_test2 <- CPM_normData_notfiltered_OP[UpregulatedInCL_names,]

# calculate standard deviation (sd) and mean for upregulated genes in CL lesions
mean_allgenes2 <- rowMeans(all_genes_test2)
sd_test_allgenes2 <- transform(all_genes_test2, SD=apply(all_genes_test2,1, sd,
                                                         na.rm = TRUE))
SDs_allgenes2 <- sd_test_allgenes2$SD
genes_all2 <- rownames(sd_test_allgenes2)

# calculating the cV:
MeanSD_allgenes2 <- cbind(genes_all2, mean_allgenes2, SDs_allgenes2)
CV_allgenes2 <- coefficient.variation(SDs_allgenes2, mean_allgenes2)

# making a dataframe with gene symbols, sd, mean and cV:
allgenes_stats <- cbind(genes_all2, mean_allgenes2, SDs_allgenes2, CV_allgenes2)

# select and filter genes with increased expression variability between CL lesions:
CV_upreg_varies <- subset(allgenes_stats, CV_allgenes2 > 0.6506) # high cV (250 HVGenes)
#CV_upreg_notvaries <- subset(allgenes_stats, CV_allgenes2 < 0.6506)# low cV

# Plot upregulated genes in CL, according to cV and FC (log2)
SD_CV.pre <- allgenes_stats[,c(-1)] # removing a column with gene names

# getting FC from limma DGE analyis Cl vs. HS:
tophits_genes <- allHits.lesion.coding[UpregulatedInCL_names,]

SD_CV <- cbind(tophits_genes, SD_CV.pre)
SD_CV$CV_allgenes2 <- as.numeric(as.character(SD_CV$CV_allgenes2))
SD_CV$SDs_allgenes2 <- as.numeric(as.character(SD_CV$SDs_allgenes2))
SD_CV$genes <- rownames(SD_CV)
SD_CV$genes2 <- SD_CV$genes

# Important: the gene symbols from high varible genes (250 genes) were used to evaluate
# by gene ontology (using DAVID website) the pathways enriched and associated with high
# variability.
# The pathways enriched (FDR<0.01) were:
# Cluster 1: B cell responses
# Cluster 2: chemotaxis
# Cluster 3: cytotoxic granules + IL1B
# Genes contained in these pathways were:
cluster1 <- c("IGHA1","IGHA2","IGHG1","IGHG2","IGHG3","IGHG4","IGHM","IGHV1-18","IGHV1-2",
              "IGHV1-24","IGHV1-46","IGHV1-58","IGHV1OR16-1","IGHV2-70","IGHV3-11",
              "IGHV3-15","IGHV3-21","IGHV3-23","IGHV3-33","IGHV3-43","IGHV3-48",
```

```
              "IGHV3-49","IGHV3-53","IGHV3-62","IGHV3-64","IGHV3-7","IGHV3-72","IGHV3-73",
              "IGHV3-74","IGHV3OR16-9","IGHV4-28","IGHV4-31","IGHV4-39","IGHV4-59",
              "IGHV5-51","IGHV6-1","IGKC","IGKV1-12","IGKV1-16","IGKV1-17","IGKV1-27",
              "IGKV1-39","IGKV1-5","IGKV1-6","IGKV1-9","IGKV1D-33","IGKV1D-39","IGKV1D-8",
              "IGKV2-24","IGKV2-30","IGKV2D-28","IGKV2D-29","IGKV3-11","IGKV3-15",
              "IGKV3-20","IGKV3D-11","IGKV3D-15","IGKV3D-20","IGKV3D-7","IGKV4-1",
              "IGKV6-21","IGLC1","IGLC2","IGLC3","IGLL5","IGLV1-36","IGLV1-40",
              "IGLV1-44","IGLV3-10","IGLV3-19","IGLV3-25","IGLV3-27","IGLV3-9","IGLV4-69",
              "IGLV5-45","IGLV6-57","IGLV7-43","IGLV7-46","CD79A","CR1","KIR3DL2",
              "KIR2DL4","JCHAIN")
cluster2 <- c("CCL18","CCL20","CCL3","CCL4","CCL8","CXCL1","CXCL11","CXCL13","CXCL2",
              "CXCL3","CXCL6","CXCL8","S100A12","S100A8","S100A9","CSF3R","TREM1",
              "ANXA1","NDP","WISP1","ADCYAP1","BHLHA15","CYR61","GJB2","INHBA","RNASE2")
cluster3 <- c("GZMB","PRF1","GNLY","IL1B")

# make objects and conditions that will be used to annotate the gene symbols in the plots:
mygenes <- SD_CV$genes2 %in% cluster3
SD_CV$genes2[!mygenes] <- NA
SD_CV$genes3 <- SD_CV$genes
cytotoxicity <- SD_CV$genes3 %in% cluster3
Igs <- SD_CV$genes3 %in% cluster1
chemokines <- SD_CV$genes3 %in% cluster2
inflammation <- SD_CV$genes3 %in% c("IL1B")

SD_CV$genes3[cytotoxicity] <- "cytotoxic granules"
SD_CV$genes3[Igs] <- "B cell response (immunoglubulins)"
SD_CV$genes3[chemokines] <- "chemotaxis (chemokines)"
SD_CV$genes3[inflammation] <- "IL1B"
SD_CV$genes3[!cytotoxicity & !Igs & !chemokines] <- "others"

# Variability plot:
ggExtra::ggMarginal(
  ggplot(SD_CV, aes(y=CV_allgenes2, x=logFC, color=SD_CV$genes3)) +
    geom_point(size=2.5) +
    theme_calc() + scale_color_manual(values = c("#F8D34F", # yellow
                                                 "#1A4682", # blue
                                                 "#EB522C", # orange
                                                 "#EB522C", # orange
                                                 "grey")) +
    theme(legend.position="none", axis.title = element_text(size = 13),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(), legend.text = element_text(size = 15),
          axis.text.x=element_text(size=13, colour = "black"),
          axis.text.y = element_text(size=13, colour = "black")) +
    labs(colour="") +
    geom_text_repel(aes(label = genes2),
                    size = 4.5, fontface="bold",colour="black") +
    geom_hline(yintercept = 0.6506, linetype=2) +
    scale_x_continuous(limits=c(0.8,11), breaks = c(1,2,3,4,5,6,7,8,9,10,11)) +
    scale_y_continuous(limits=c(0.1,4), breaks = c(0.5,0.65,1,1.5,2,2.5,3,3.5,4)) +
    xlab("logFC CL relative to HS") +
    ylab("Coefficient of Variation (CV)"),
  type = 'density', margins = 'both', size = 10, col = 'grey', fill = 'grey')
```



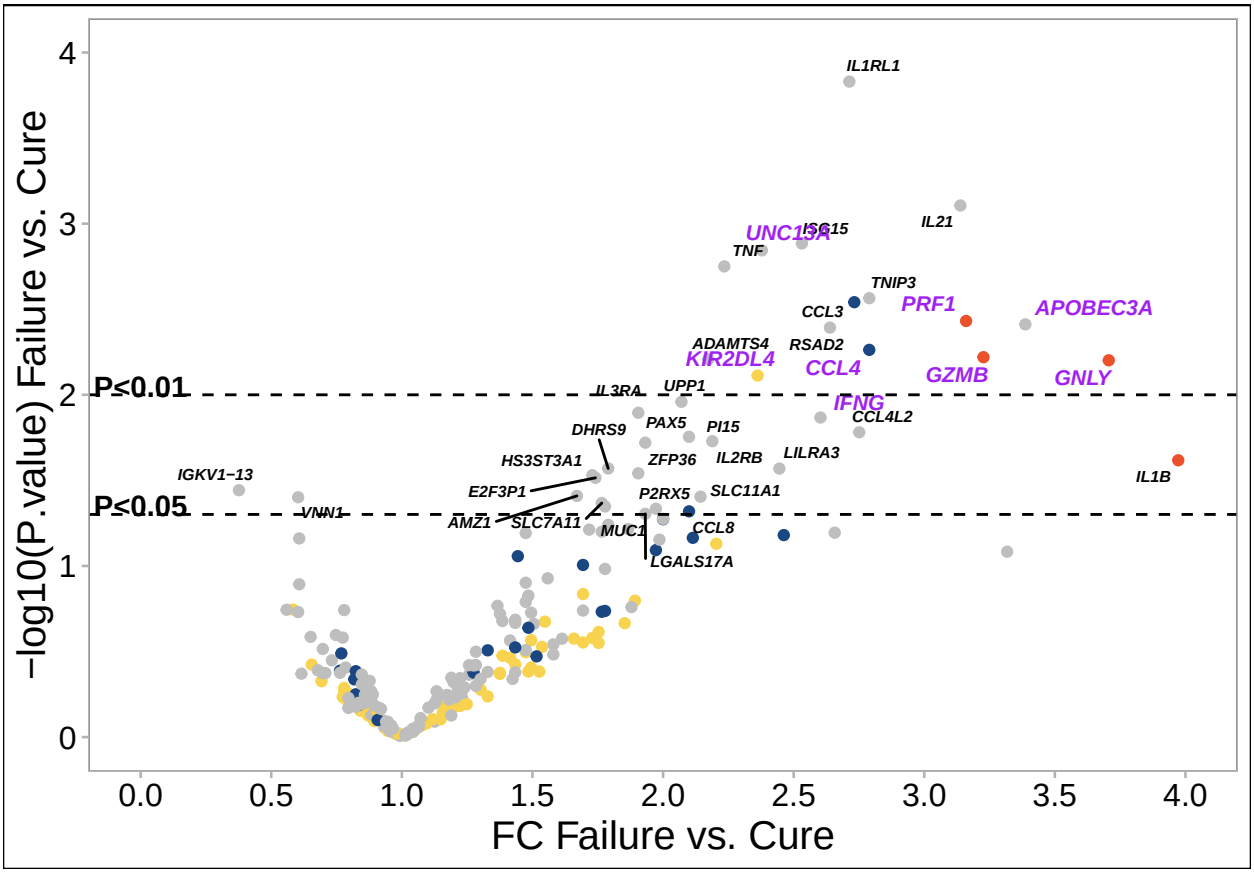
Association of ViTALs with treatment failure

volcano plot of ViTALs - Figure 3B

In this session we identify between the 250 HVGenes which genes were statistically different between Failure vs. Cure patients.

```
merged1 <- myTopHits.treat # accessing the limma DGE analysis between Failure vs. Cure
rownames(merged1) <- merged1$geneID
merged1 <- merged1[c(rownames(CV_upreg_varies)),]
merged1 <- merged1[,c("geneID","logFC","P.Value", "t")]

# selecting the genes with P.value<0.05 to label them in the plot
sigfailure.pre <- subset(merged1, P.Value < 0.05)
sigfailure <- rownames(sigfailure.pre)

# label genes with P.value<0.5 in the plot.
# also, highlight in the plot the genes also found to be differentially expressed between
# Failure vs. Cure (P.value<0.05) in an independente dataset (2016 dataset).
# Reference: Chirstensen SM et al., 2016
dataset2016 <- c("PRF1","GZMB","GNLY","APOBEC3A","CCL4","KIR2DL4","UNC13A","IFNG")

sigfailure <- sigfailure[!sigfailure %in% dataset2016]
merged1$siginfo <- merged1$geneID
sigfailureX <- merged1$siginfo %in% sigfailure
merged1$siginfo[!sigfailureX] <- NA
merged1$bothdata <- merged1$geneID
bothdataX <- merged1$bothdata %in% dataset2016
merged1$bothdata[!bothdataX] <- NA
```

```
# color clusters in the plot:
merged1$geneID2 <- merged1$geneID
cytotoxicity <- merged1$geneID2 %in% cluster3
Igs <- merged1$geneID2 %in% cluster1
chemokines <- merged1$geneID2 %in% cluster2
merged1$geneID2[cytotoxicity] <- "cytotoxic granules"
merged1$geneID2[Igs] <- "B cell response (immunoglubulins)"
merged1$geneID2[chemokines] <- "chemotaxis (chemokines)"
merged1$geneID2[!cytotoxicity & !Igs & !chemokines] <- "others"

# Treatment outcome plot:
ggplot(merged1, aes(x=2^logFC, y=-log10(P.Value),
                    color=merged1$geneID2)) +
  geom_point() +
  theme_calc() + scale_color_manual(values = c("#F8D34F", # yellow
                                               "#1A4682", # blue
                                               "#EB522C", # orange
                                               "grey")) +
  theme(legend.position="none", axis.title = element_text(size = 15),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), legend.text = element_text(size = 17),
        axis.text.x=element_text(size=12, colour = "black"),
        axis.text.y = element_text(size=12, colour = "black")) +
  labs(size="CV") +
  geom_text_repel(aes(label = siginfo),
                  size = 2,
                  fontface=4,
                  color="black") +
  geom_text_repel(aes(label = bothdata),
                  size = 2.8, fontface=4, color="purple") +
  scale_y_continuous(limits=c(0,4)) +
  scale_x_continuous(limits=c(0,4), breaks = c(0,0.5,1,1.5,2,2.5,3,3.5,4)) +
  geom_hline(yintercept = -log10(0.01), linetype=2) +
  geom_hline(yintercept = -log10(0.05), linetype=2) +
  annotate("text", x=0, y=-log10(0.01)+0.05,
           label=paste("P<0.01"), size=4, fontface="bold") +
  annotate("text", x=0, y=-log10(0.05)+0.05,
           label=paste("P<0.05"), size=4, fontface="bold") +
  xlab("FC Failure vs. Cure") +
  ylab("-log10(P.value) Failure vs. Cure")
```



Identifying parasite transcripts in patients

host read filtering

Reads mapping to host were removed using Kneaddata, which uses Bowtie2 to map reads to the human reference genome. Reads that *did not* map to human were be used for mapping to the parasite reference transcriptome in the next step.

```
# first the healthy subjects (HS).
kneaddata -i HS1.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o HS
kneaddata -i HS2.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o HS
kneaddata -i HS3.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o HS
kneaddata -i HS4.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o HS
kneaddata -i HS5.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o HS
kneaddata -i HS6.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o HS
kneaddata -i HS7.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o HS

# then the cutaneous leishmaniasis (CL) patients
kneaddata -i CL1.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o CL
kneaddata -i CL2.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o CL
kneaddata -i CL3.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o CL
kneaddata -i CL4.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o CL
kneaddata -i CL5.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o CL
kneaddata -i CL6.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o CL
kneaddata -i CL7.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o CL
kneaddata -i CL8.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o CL
```

```
kneaddata -i CL9.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o C
kneaddata -i CL10.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL11.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL12.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL13.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL14.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL15.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL16.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL17.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL18.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL19.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL20.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
kneaddata -i CL21.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Index -o
```

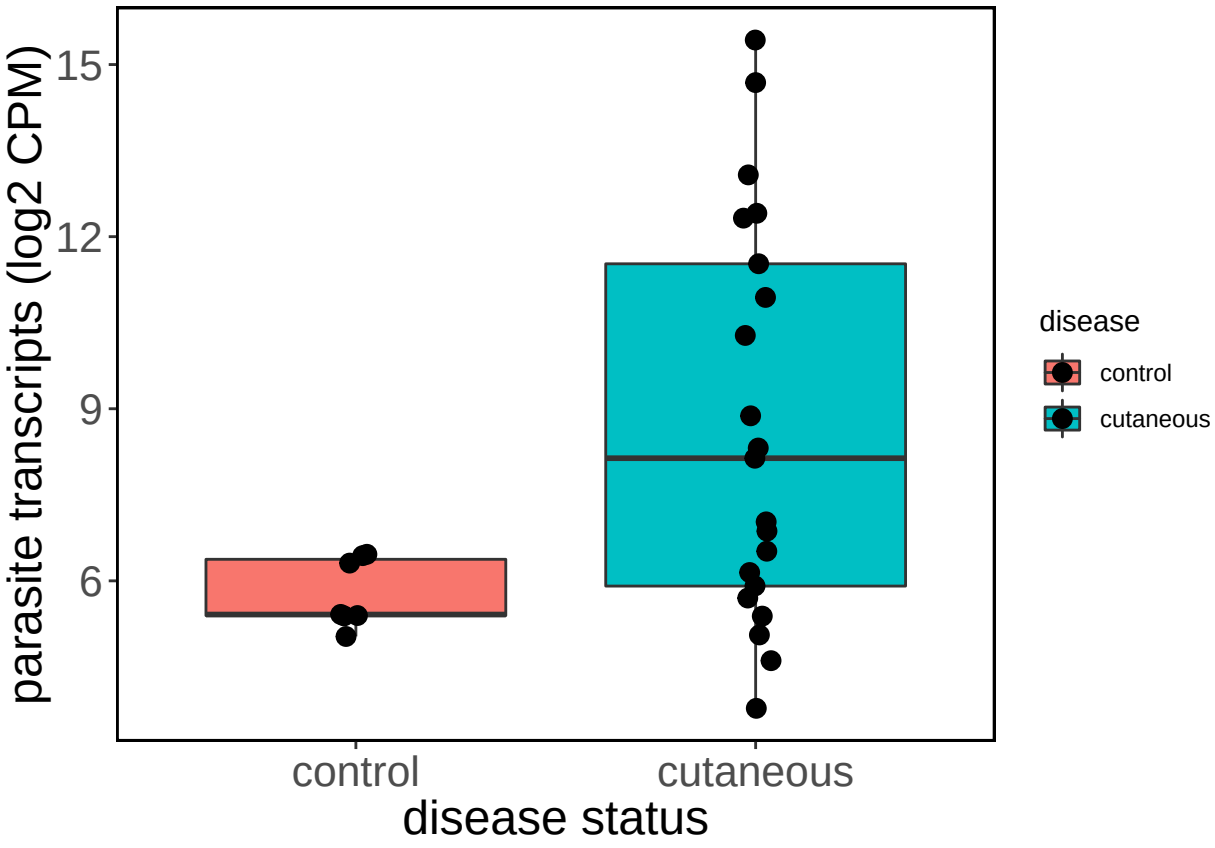
mapping reads to the parasite transcriptome

```
# Map reads to the indexed reference transcriptome for PARASITE
# the input for these alignments are the filtered fastq files produced above by Kneaddata.

# build index from reference fasta from Ensembl L. braziliensis transcriptome
kallisto index -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index Leishmania_braziliens

# first the healthy subjects (HS).  These will serve as a negative control for parasite read mapping
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_HS01 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_HS02 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_HS03 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_HS04 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_HS05 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_HS06 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_HS07 -t


# then the cutaneous leishmaniasis (CL) patients
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL01 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL02 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL03 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL04 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL05 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL06 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL07 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL08 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL09 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL10 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL11 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL12 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL13 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL14 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL15 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL16 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL17 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL18 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL19 -t
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL20 -t
```

```
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_CL21 -t 1
```

parasite transcripts in CL vs HS

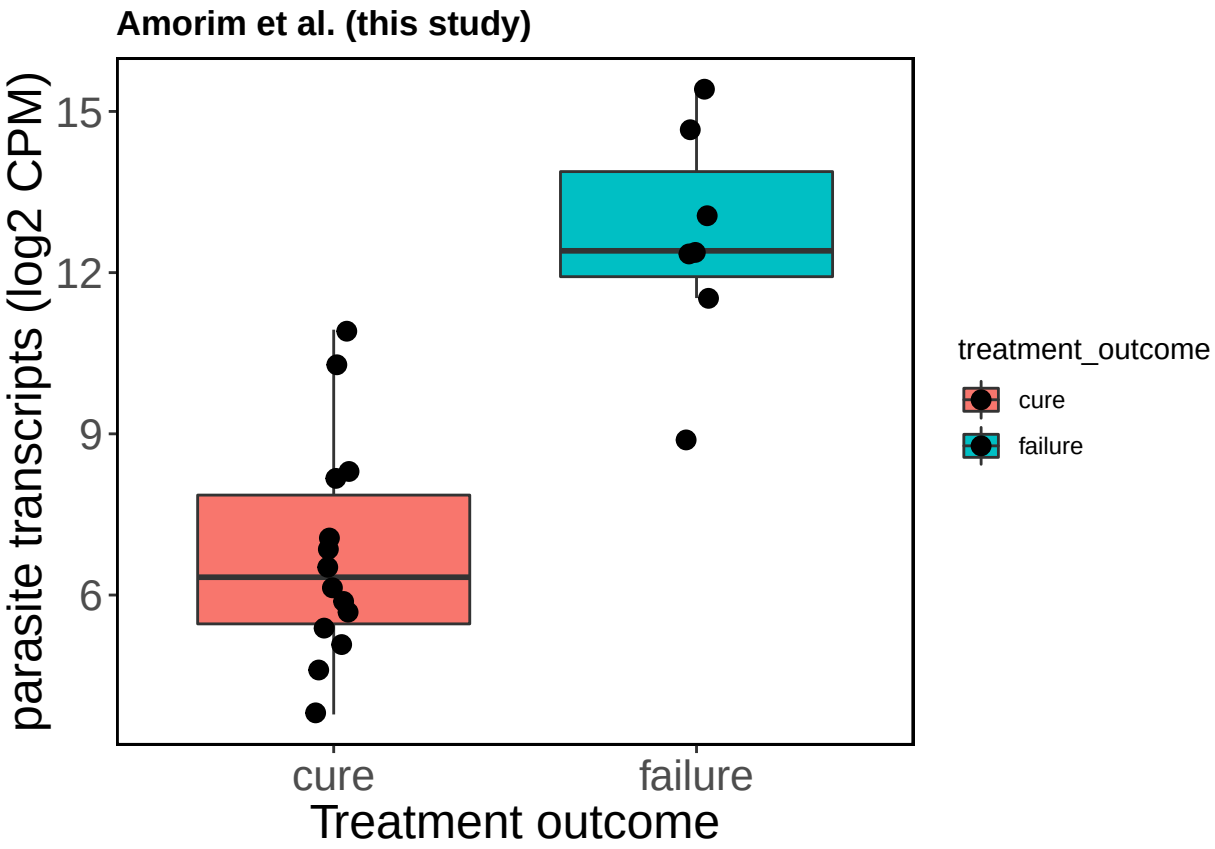
```
list.patients <- 1:21
list.healthy <- 1:7

list.patients <- str_pad(list.patients, 2, pad = "0")
list.healthy <- str_pad(list.healthy, 2, pad = "0")

paths.patients.parasite <- file.path("../readMapping/Lbraz", c(paste0("parasite_CL", list.patients)), "a
paths.healthy.parasite <- file.path("../readMapping/Lbraz", c(paste0("parasite_HS", list.healthy)), "abu

paths.all <- c(paths.healthy.parasite, paths.patients.parasite)

Txi.lesion.all <- tximport(paths.all,
                           type = "kallisto",
                           txOut = TRUE,
                           ignoreTxVersion = TRUE,
                           countsFromAbundance = "lengthScaledTPM")

#number of reads remaining after removing host reads above
librarySize.all <- c(9603863, 6013019, 7269338, 2347419, 641140, 5384651, 6807472, #healthy subjects
                9243184, 4332726, 11725185, 3494176, 16896230, 16427463, 5035568, 10252676,
                8794839, 8145232, 6398817, 6467353, 12087118, 10636911, 12100797, 8457067,
                6116730, 8201034, 12736501, 4582083, 12757299)

librarySize.all <- librarySize.all/1000000

parasiteTx_CPM <- colSums(Txi.lesion.all$counts)/librarySize.all

ggplot(targets.lesion, aes(x=disease, y=log2(parasiteTx_CPM), fill=disease)) +
  geom_boxplot(outlier.shape = NA) +
  labs(y="parasite transcripts (log2 CPM)", x = "disease status") +
  geom_jitter(width = .05, size=3) +
  theme_bw() +
  theme(axis.text=element_text(size=16),
        axis.title=element_text(size=18),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```



comparing parasite transcripts with treatment outcome

```
list <- 1:21
list <- str_pad(list, 2, pad = "0")
paths.patients.parasite <- file.path("../readMapping/Lbraz", c(paste0("parasite_CL", list)), "abundance

Txi.lesion.parasite <- tximport(paths.patients.parasite,
                                type = "kallisto",
                                txOut = TRUE,
                                ignoreTxVersion = TRUE,
                                countsFromAbundance = "lengthScaledTPM")

#modifying the patient-level data to include parasite counts, as well as expresion of selected ViTALs
ViTALs_selected <- Txi.lesion.coding.onlypatients$abundance %>%
  as_tibble(rownames = "geneSymbol") %>%
  dplyr::filter(geneSymbol == "PRF1" | geneSymbol =="GNLY" | geneSymbol == "GZMB" | geneSymbol =="IFNG"
                | geneSymbol == "UNC13A" | geneSymbol =="KIR2DL4" | geneSymbol == "CCL4" | geneSymbol ==

geneSymbols_ViTALS_selected <- ViTALs_selected$geneSymbol
ViTALs_selected <- as.data.frame(t(ViTALs_selected[,-1]))
colnames(ViTALs_selected) <- geneSymbols_ViTALS_selected

#number of reads remaining after removing host reads above
librarySize.CL <- c(9243184, 4332726, 11725185, 3494176, 16896230, 16427463, 5035568, 10252676,
                    8794839, 8145232, 6398817, 6467353, 12087118, 10636911, 12100797, 8457067,
```

```
                6116730, 8201034, 12736501, 4582083, 12757299)

librarySize.CL <- librarySize.CL/1000000
parasiteTx_CPM_Amorim <- colSums(Txi.lesion.parasite$counts)/librarySize.CL

targets.onlypatients <- targets.onlypatients %>%
  dplyr::mutate(parasiteTx_CPM_Amorim = parasiteTx_CPM_Amorim) %>%
  dplyr::mutate(PRF1 = ViTALs_selected$PRF1) %>%
  dplyr::mutate(GZMB = ViTALs_selected$GZMB) %>%
  dplyr::mutate(CCL4 = ViTALs_selected$CCL4) %>%
  dplyr::mutate(GNLY = ViTALs_selected$GNLY) %>%
  dplyr::mutate(UNC13A = ViTALs_selected$UNC13A) %>%
  dplyr::mutate(APOBEC3A = ViTALs_selected$APOBEC3A) %>%
  dplyr::mutate(KIR2DL4 = ViTALs_selected$KIR2DL4) %>%
  dplyr::mutate(IFNG = ViTALs_selected$IFNG)

#also storing this plot for use later
p1 <- ggplot(targets.onlypatients, aes(x=treatment_outcome, y=log2(parasiteTx_CPM_Amorim), fill=treatmen
  geom_boxplot(outlier.shape = NA) +
  labs(y="parasite transcripts (log2 CPM)", x = "Treatment outcome",
       title = "Amorim et al. (this study)") +
  geom_jitter(width = .05, size=3) +
  theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=16),
        axis.title=element_text(size=18),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

ggplot(targets.onlypatients, aes(x=treatment_outcome, y=log2(parasiteTx_CPM_Amorim), fill=treatment_out
  geom_boxplot(outlier.shape = NA) +
  labs(y="parasite transcripts (log2 CPM)", x = "Treatment outcome",
       title = "Amorim et al. (this study)") +
  geom_jitter(width = .05, size=3) +
  theme_bw() +
  theme(axis.text=element_text(size=16),
        axis.title=element_text(size=18),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```



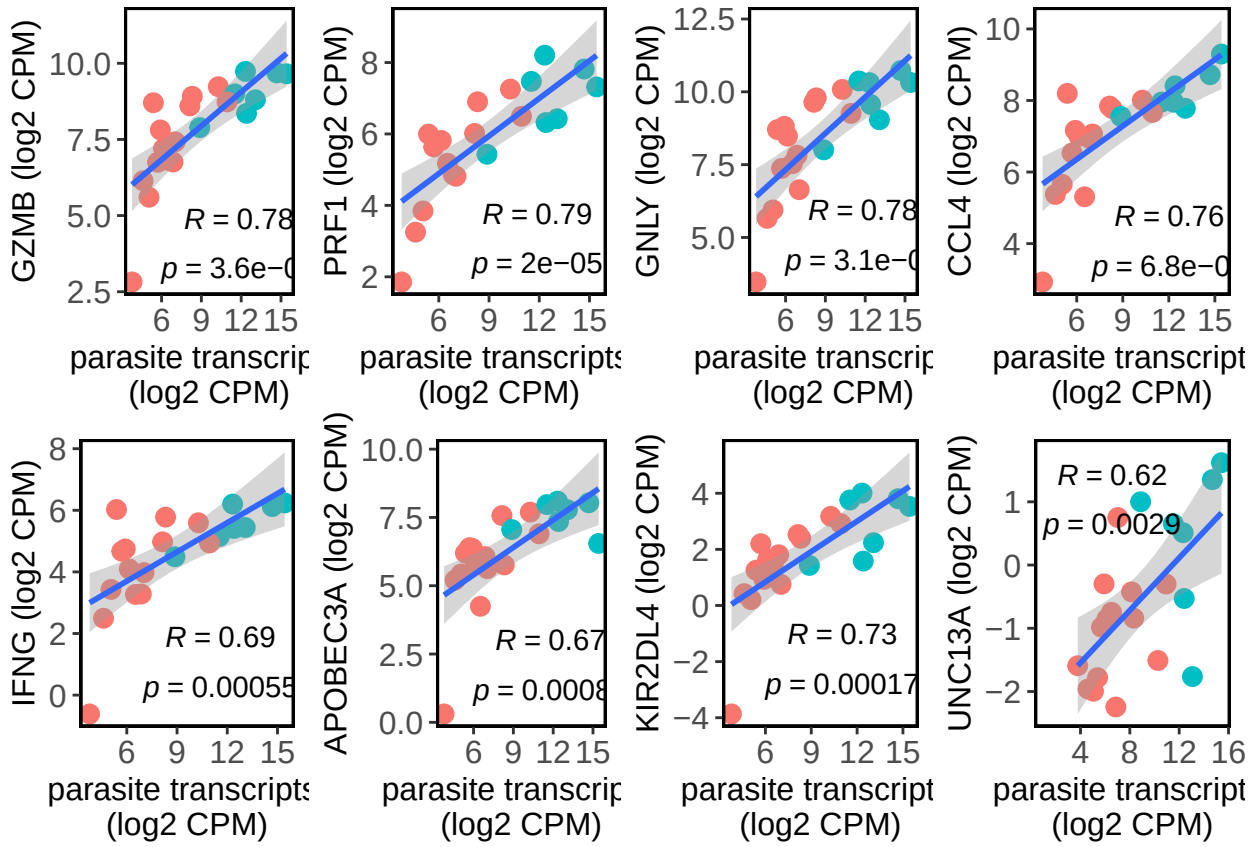
correlation of parasite transcripts with ViTALs - Figure 5D

```
p_PRF1 <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Amorim), y = log2(PRF1))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("PRF1 (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 8, label.y = 3, label.sep = "\n") +
  theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_GZMB <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Amorim), y = log2(GZMB))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("GZMB (log2 CPM)") +
```

```
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 6, label.y = 4, label.sep = "\n") +
  theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_GNLY <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Amorim), y = log2(GNLY))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("GNLY (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 6, label.y = 5, label.sep = "\n") +
  theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_IFNG <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Amorim), y = log2(IFNG))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("IFNG (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 7, label.y = 1, label.sep = "\n") +
  theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_UNC13A <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Amorim), y = log2(UNC13A))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("UNC13A (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 1, label.y = 1, label.sep = "\n") +
```

```
  theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_APOBEC3A <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Amorim), y = log2(APOBEC3A))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("APOBEC3A (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 6, label.y = 2, label.sep = "\n") +
  theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_KIR2DL4 <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Amorim), y = log2(KIR2DL4))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("KIR2DL4 (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 6, label.y = -2, label.sep = "\n") +
  theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_CCL4 <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Amorim), y = log2(CCL4))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("CCL4 (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 7, label.y = 4, label.sep = "\n") +
  theme_bw() +
  theme(legend.position = "none",
```

```
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

plot_grid(p_GZMB, p_PRF1, p_GNLY, p_CCL4, p_IFNG, p_APOBEC3A, p_KIR2DL4, p_UNC13A, nrow=2)
```



Validation on patient cohort from Christensen et al., PLOS NTD, 2016

Sample info

```
import <- read_tsv("StudyDesign_Christensen_plosNTD_2016.txt")
import %>% dplyr::filter(disease == "CL" & treatment_outcome != "unclear") %>%
  gt() %>%
  tab_header(title = md("Clinical metadata from patients with cutaneous leishmaniasis (CL) in Christens
             subtitle = md("`(n=24)`")) %>%  cols_align(align = "center", columns = TRUE)
```

Clinical metadata from patients with cutaneous leishmaniasis (CL) in Christensen et al.
(n=24)

sample	disease	disease_stage	treatment_outcome	lesion_size	age_(years)	sex	Time_to_cure_(days)

SRR3162852	CL	early	cure	36	44	F	60
SRR3162853	CL	early	failure	40	24	M	more
SRR3162854	CL	early	failure	56	31	F	more
SRR3162855	CL	early	failure	12	25	F	90
SRR3162856	CL	early	failure	80	33	M	90
SRR3162857	CL	early	failure	120	25	F	90
SRR3162858	CL	early	failure	25	30	F	more
SRR3162859	CL	early	failure	4	40	M	more
SRR3162860	CL	late	failure	100	25	M	more
SRR3162861	CL	late	cure	440	28	M	90
SRR3162862	CL	late	cure	100	19	M	60
SRR3162863	CL	late	failure	560	33	F	more
SRR3162867	CL	late	cure	180	18	M	60
SRR3162864	CL	late	cure	252	27	M	90
SRR3162865	CL	late	cure	100	45	F	60
SRR3162866	CL	late	failure	48	21	M	90
SRR3162868	CL	late	failure	550	37	M	90
SRR3162869	CL	late	cure	380	25	M	60
SRR3162870	CL	late	failure	250	30	M	more
SRR3162871	CL	late	cure	100	33	F	60
SRR3162872	CL	late	cure	440	25	M	90
SRR3162873	CL	late	failure	192	19	F	90
SRR3162875	CL	late	failure	48	26	F	90
SRR3162876	CL	late	failure	960	19	M	90

```
targets.lesion <- dplyr::filter(import, treatment_outcome != "unclear")
# only CL lesions where treatment outcome was clear (n=24)
targets.onlypatients <- dplyr::filter(import, disease == "CL" & treatment_outcome != "unclear")

# Making factors that will be used for pairwise comparisons:
# HS vs. CL lesions as a factor:
disease.lesion <- factor(targets.lesion$disease)
# Cure vs. Failure lesions as a factor:
treatment.lesion <- factor(targets.onlypatients$treatment_outcome)
```

mapping raw reads to human reference

```
# First the healthy controls
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162842 -b 60 -t 24 SRR3162842_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162843 -b 60 -t 24 SRR3162843_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162844 -b 60 -t 24 SRR3162844_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162845 -b 60 -t 24 SRR3162845_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162846 -b 60 -t 24 SRR3162846_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162847 -b 60 -t 24 SRR3162847_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162848 -b 60 -t 24 SRR3162848_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162849 -b 60 -t 24 SRR3162849_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162850 -b 60 -t 24 SRR3162850_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162851 -b 60 -t 24 SRR3162851_1.fastq.g

# Then the cutaneous leishmaniasis patients
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162852 -b 60 -t 24 SRR3162852_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162853 -b 60 -t 24 SRR3162853_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162854 -b 60 -t 24 SRR3162854_1.fastq.g
```

```
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162855 -b 60 -t 24 SRR3162855_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162856 -b 60 -t 24 SRR3162856_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162857 -b 60 -t 24 SRR3162857_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162858 -b 60 -t 24 SRR3162858_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162859 -b 60 -t 24 SRR3162859_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162860 -b 60 -t 24 SRR3162860_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162861 -b 60 -t 24 SRR3162861_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162862 -b 60 -t 24 SRR3162862_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162863 -b 60 -t 24 SRR3162863_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162864 -b 60 -t 24 SRR3162864_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162865 -b 60 -t 24 SRR3162865_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162866 -b 60 -t 24 SRR3162866_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162867 -b 60 -t 24 SRR3162867_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162868 -b 60 -t 24 SRR3162868_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162869 -b 60 -t 24 SRR3162869_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162870 -b 60 -t 24 SRR3162870_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162871 -b 60 -t 24 SRR3162871_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162872 -b 60 -t 24 SRR3162872_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162873 -b 60 -t 24 SRR3162873_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162874 -b 60 -t 24 SRR3162874_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162875 -b 60 -t 24 SRR3162875_1.fastq.g
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.Index -o host_SRR3162876 -b 60 -t 24 SRR3162876_1.fastq.g
```

QC of raw data and read mapping

As was done before, quality control of raw reads was carried out using fastqc. The quality of raw reads, as well as the results of Kallisto mapping for the Christensen et al. dataset were summarized using multiqc. The resulting multiqc report can be found in the github project repo in the `QA` directory. *Note*: due to size limitations imposed by Github, neither the raw fastq files, nor the reference fasta file used for read mapping could be stored in the GitHub repo.

importing human data

```
# Importing .h5 Kallisto outputs and annotate transcripts to gene symbols:
paths.all <- file.path("../Christensen_plosNTD_2016/human", paste0("host_", targets.lesion$sample), "abu
paths.patients <- file.path("../Christensen_plosNTD_2016/human", paste0("host_", targets.onlypatients$sa

Txi.lesion.Christensen <- tximport(paths.all,
                                   type = "kallisto",
                                   tx2gene = Tx,
                                   txOut = FALSE,
                                   ignoreTxVersion = TRUE,
                                   countsFromAbundance = "lengthScaledTPM")

Txi.lesion.Christensen.onlypatients <- tximport(paths.patients,
                                                type = "kallisto",
                                                tx2gene = Tx,
                                                txOut = FALSE,
                                                ignoreTxVersion = TRUE,
                                                countsFromAbundance = "lengthScaledTPM")
```

filtering out host reads

```
# first the healthy subjects (HS).
kneaddata -i SRR3162842.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162843.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162844.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162845.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162846.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162847.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162848.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162849.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162850.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162851.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde

# then the cutaneous leishmaniasis (CL) patients
kneaddata -i SRR3162852.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162853.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162854.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162855.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162856.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162857.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162858.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162859.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162860.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162861.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162862.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162863.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162867.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162864.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162865.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162866.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162868.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162869.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162870.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162871.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162872.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162873.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162874.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162875.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
kneaddata -i SRR3162876.fastq.gz -db /data/reference_db/Homo_sapiens/Ensembl/GRCh37/Sequence/Bowtie2Inde
```

mapping filtered reads to *L. braziliensis*

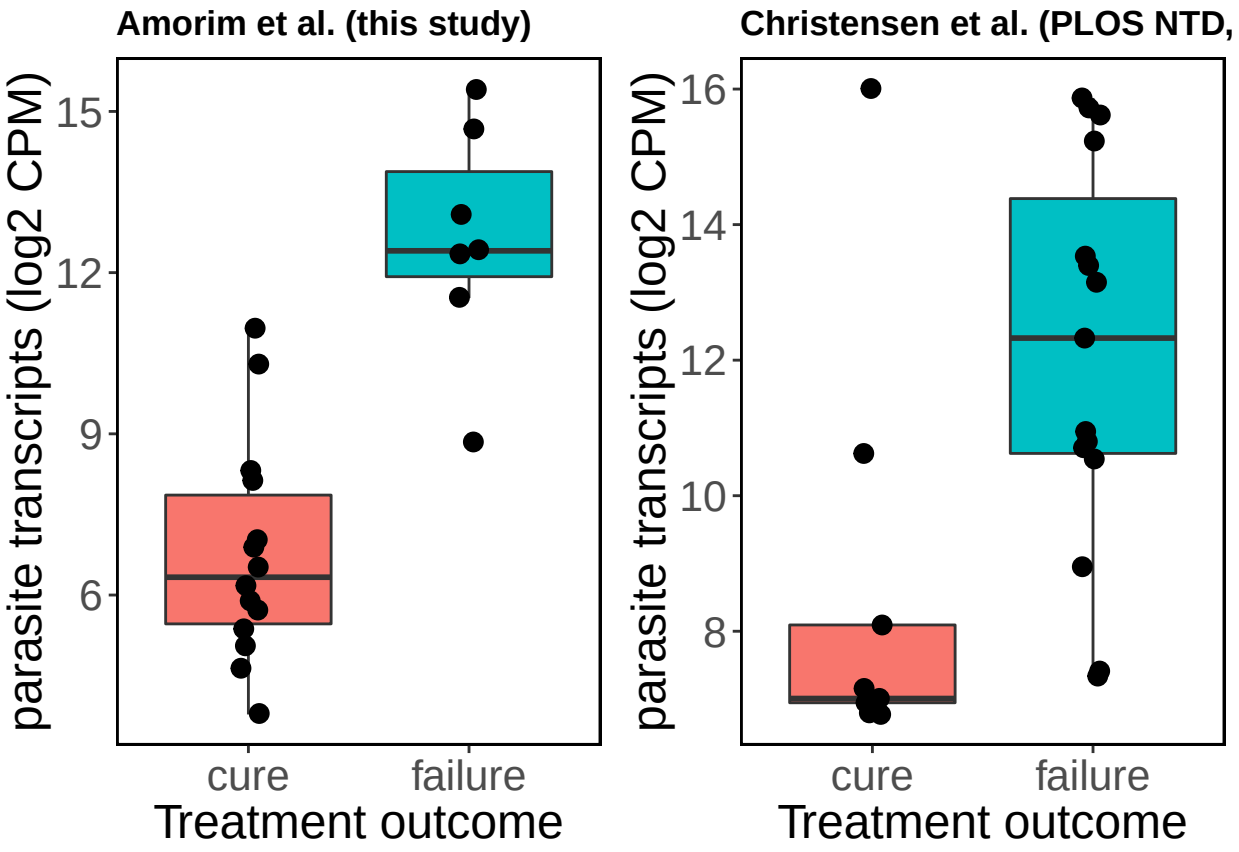
```
# first the healthy subjects (HS).
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR316284
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR316284
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR316284
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR316284
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR316284
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR316284
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR316284
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR316284
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR316285
```

```
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628

# then the cutaneous leishmaniasis (CL) patients
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
kallisto quant -i Leishmania_braziliensis_mhom_br_75_m2904.ASM284v2.cdna.all.index -o parasite_SRR31628
```

Parasite reads by treatment outcome - Figure 6B

```
paths.patients.Christensen.parasite <- file.path("../Christensen_plosNTD_2016/Lbraz", paste0("parasite_

Txi.lesion.Christensen.parasite <- tximport(paths.patients.Christensen.parasite,
                                            type = "kallisto",
                                            txOut = TRUE,
                                            ignoreTxVersion = TRUE,
                                            countsFromAbundance = "lengthScaledTPM")


#modifying the patient-level data to include parasite counts, as well as expresion of selected ViTALs
ViTALs_selected_Christensen <- Txi.lesion.Christensen.onlypatients$abundance %>%
  as_tibble(rownames = "geneSymbol") %>%
  dplyr::filter(geneSymbol == "PRF1" | geneSymbol =="GNLY" | geneSymbol == "GZMB" | geneSymbol =="IFNG"
                | geneSymbol == "UNC13A" | geneSymbol =="KIR2DL4" | geneSymbol == "CCL4" | geneSymbol ==


geneSymbols_ViTALS_selected <- ViTALs_selected_Christensen$geneSymbol
ViTALs_selected_Christensen <- as.data.frame(t(ViTALs_selected_Christensen[,-1]))
colnames(ViTALs_selected_Christensen) <- geneSymbols_ViTALS_selected

#number of reads remaining after removing host reads above
librarySize.CL <- c(5112799, 5138357, 9712437, 10867257, 7564381,
```

```
                    10339099, 3576442, 3774463, 5588509, 5341356,
                    5721003, 4766394, 4538135, 5060671, 5166608,
                    5067688, 4414686, 4334193, 8849480, 3886005,
                    4313191, 4033585, 3485099, 5532895)

librarySize.CL <- librarySize.CL/1000000
parasiteTx_CPM_Christensen <- colSums(Txi.lesion.Christensen.parasite$counts)/librarySize.CL

targets.onlypatients <- targets.onlypatients %>%
  dplyr::mutate(parasiteTx_CPM_Christensen = parasiteTx_CPM_Christensen) %>%
  dplyr::mutate(PRF1 = ViTALs_selected_Christensen$PRF1) %>%
  dplyr::mutate(GZMB = ViTALs_selected_Christensen$GZMB) %>%
  dplyr::mutate(CCL4 = ViTALs_selected_Christensen$CCL4) %>%
  dplyr::mutate(GNLY = ViTALs_selected_Christensen$GNLY) %>%
  dplyr::mutate(UNC13A = ViTALs_selected_Christensen$UNC13A) %>%
  dplyr::mutate(APOBEC3A = ViTALs_selected_Christensen$APOBEC3A) %>%
  dplyr::mutate(KIR2DL4 = ViTALs_selected_Christensen$KIR2DL4) %>%
  dplyr::mutate(IFNG = ViTALs_selected_Christensen$IFNG)

p2 <- ggplot(targets.onlypatients, aes(x=treatment_outcome, y=log2(parasiteTx_CPM_Christensen), fill=tr
  geom_boxplot(outlier.shape = NA) +
  labs(y="parasite transcripts (log2 CPM)", x = "Treatment outcome",
       title = "Christensen et al. (PLOS NTD, 2016)") +
  geom_jitter(width = .05, size=3) +
  theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=16),
        axis.title=element_text(size=18),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

plot_grid(p1, p2)
```



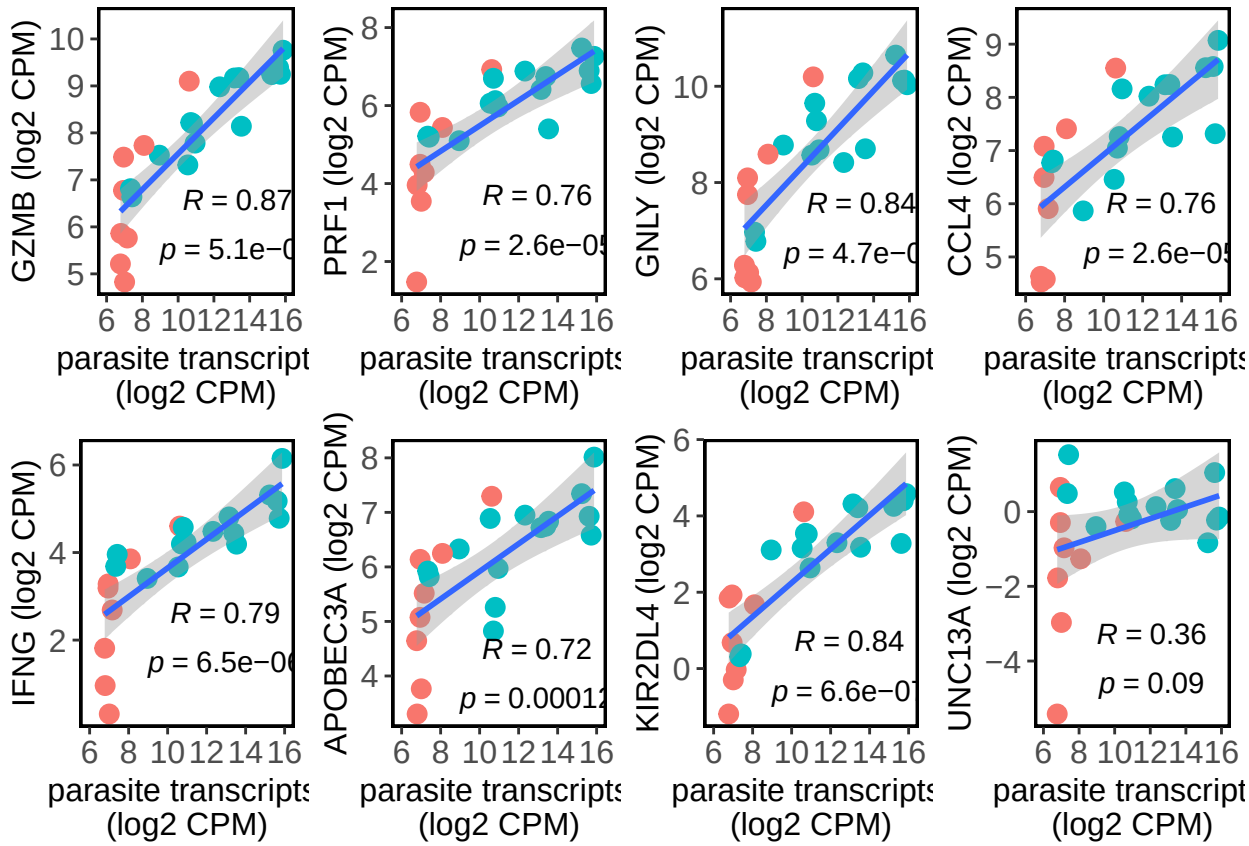
validation of 8 ViTALs conserved between Amorim et al. and Christensen et al.

```
p_PRF1 <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Christensen), y = log2(PRF1))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("PRF1 (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 9, label.y = 3, label.sep = "\n") +
  theme_bw() +
  xlim(6, 16) +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_GZMB <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Christensen), y = log2(GZMB))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
```

```
  ylab("GZMB (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 9, label.y = 6, label.sep = "\n") +
  theme_bw() +
  xlim(6, 16) +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_GNLY <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Christensen), y = log2(GNLY))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("GNLY (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 9, label.y = 7, label.sep = "\n") +
  theme_bw() +
  xlim(6, 16) +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_IFNG <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Christensen), y = log2(IFNG))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("IFNG (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 9, label.y = 2, label.sep = "\n") +
  theme_bw() +
  xlim(6, 16) +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_UNC13A <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Christensen), y = log2(UNC13A))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
```

```
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("UNC13A (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 9, label.y = -4, label.sep = "\n") +
  theme_bw() +
  xlim(6, 16) +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_APOBEC3A <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Christensen), y = log2(APOBEC3A)
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("APOBEC3A (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 9, label.y = 4, label.sep = "\n") +
  theme_bw() +
  xlim(6, 16) +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_KIR2DL4 <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Christensen), y = log2(KIR2DL4)))
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("KIR2DL4 (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 9, label.y = 0, label.sep = "\n") +
  theme_bw() +
  xlim(6, 16) +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

p_CCL4 <- ggplot(targets.onlypatients, aes(x = log2(parasiteTx_CPM_Christensen), y = log2(CCL4))) +
  geom_point(shape = 19, size = 3, aes(colour = as.factor(treatment_outcome))) +
  #geom_smooth(colour = "red", fill = "lightblue", method = 'lm') +
```

```
  #ggtitle("Correlation between parasite load (by RNAseq) and IL1B expression") +
  xlab("parasite transcripts \n(log2 CPM)") +
  ylab("CCL4 (log2 CPM)") +
  geom_smooth(method='lm') +
  stat_cor(method = "pearson", label.x = 9, label.y = 5.5, label.sep = "\n") +
  theme_bw() +
  xlim(6, 16) +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=12),
        plot.title = element_text(face="bold"),
        panel.border = element_rect(colour = "black", fill=NA, size=1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

plot_grid(p_GZMB, p_PRF1, p_GNLY, p_CCL4, p_IFNG, p_APOBEC3A, p_KIR2DL4, p_UNC13A, nrow=2)
```



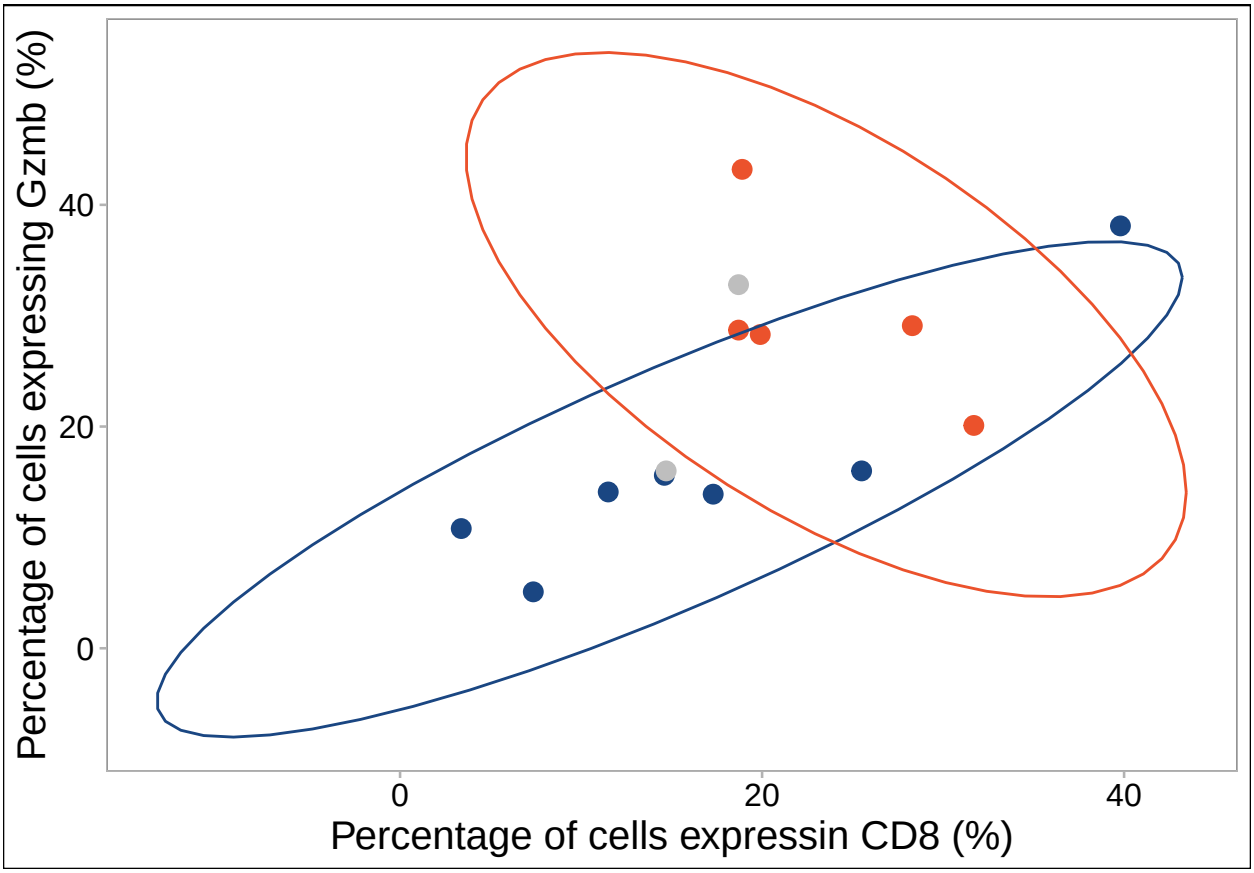
Flow cytometry validation of CD8/granzyme expression vs treatment outcome - Figure 4D

In this session, we imported the results from flow cytometry analysis, where a independent set of biopsies from CL patients were stained for CD8 and Gzmb. Clinical outcome information of the patients was incorporated in the data and it will be used to create a *ggplot* and apply ellipses with 95% confidence for statistical analysis.

```
flowdataframe <- read_delim("flowdataframe.txt", "\t", escape_double = FALSE,
                            col_types = cols(CD8 = col_number(), Gzmb = col_number()),
                            trim_ws = TRUE)
flowdataframe
```

```
## # A tibble: 14 x 3
##    status    CD8  Gzmb
##    <chr>   <dbl> <dbl>
##  1 Failure 28.3  29.1
##  2 Failure 18.9  43.2
##  3 unknown 18.7  32.8
##  4 Failure 19.9  28.3
##  5 Cure    11.5  14.1
##  6 Cure     3.38 10.8
##  7 Cure    17.3  13.9
##  8 Cure    14.6  15.6
##  9 Cure     7.36  5.09
## 10 Failure 31.7  20.1
## 11 unknown 14.7  16
## 12 Cure    25.5  16
## 13 Cure    39.8  38.1
## 14 Failure 18.7  28.7
```

```
ggplot(flowdataframe, aes(x=CD8, y=Gzmb,
                    color=status)) +
  geom_point(size=3) +
  theme_calc() + scale_color_manual(values = c("#1A4682", # blue
                                               "#EB522C", # orange
                                               "grey")) +
  theme(legend.position="none", axis.title = element_text(size = 15),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), legend.text = element_text(size = 17),
        axis.text.x=element_text(size=12, colour = "black"),
        axis.text.y = element_text(size=12, colour = "black")) +
  stat_ellipse(level = 0.95) +
  xlab("Percentage of cells expressin CD8 (%)") +
  ylab("Percentage of cells expressing Gzmb (%)")
```



Session Info

R version 3.6.0 (2019-04-26) Platform: x86_64-apple-darwin15.6.0 (64-bit) Running under: macOS Mojave 10.14.5

Matrix products: default BLAS: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale: [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages: [1] stats4 parallel stats graphics grDevices utils datasets [8] methods base

other attached packages: [1] ggpubr_0.2.1 magrittr_1.5
[3] cowplot_0.9.4 EnsDb.Hsapiens.v86_2.99.0 [5] ensembldb_2.8.0 AnnotationFilter_1.8.0
[7] GenomicFeatures_1.36.1 AnnotationDbi_1.46.0
[9] Biobase_2.44.0 GenomicRanges_1.36.0
[11] GenomeInfoDb_1.20.0 IRanges_2.18.0
[13] S4Vectors_0.22.0 BiocGenerics_0.30.0
[15] ggExtra_0.8 gt_0.1.0
[17] ggrepel_0.8.1 FinCal_0.6.3
[19] gplots_3.0.1.1 tximport_1.12.0
[21] DT_0.6 vegan_2.5-5
[23] lattice_0.20-38 permute_0.9-5
[25] patchwork_0.0.1 edgeR_3.26.0
[27] limma_3.40.0 reshape2_1.4.3
[29] ggthemes_4.2.0 forcats_0.4.0
[31] stringr_1.4.0 dplyr_0.8.1

[33] purrr_0.3.2 readr_1.3.1
[35] tidyr_0.8.3 tibble_2.1.2
[37] ggplot2_3.1.1 tidyverse_1.2.1
[39] knitr_1.23 rmarkdown_1.13

loaded via a namespace (and not attached): [1] colorspace_1.4-1 ggsignif_0.5.0
[3] XVector_0.24.0 rstudioapi_0.10
[5] bit64_0.9-7 fansi_0.4.0
[7] lubridate_1.7.4 xml2_1.2.0
[9] splines_3.6.0 zeallot_0.1.0
[11] jsonlite_1.6 Rsamtools_2.0.0
[13] broom_0.5.2 cluster_2.0.9
[15] shiny_1.3.2 compiler_3.6.0
[17] httr_1.4.0 backports_1.1.4
[19] assertthat_0.2.1 Matrix_1.2-17
[21] lazyeval_0.2.2 cli_1.1.0
[23] later_0.8.0 htmltools_0.3.6
[25] prettyunits_1.0.2 tools_3.6.0
[27] gtable_0.3.0 glue_1.3.1
[29] GenomeInfoDbData_1.2.1 Rcpp_1.0.1
[31] cellranger_1.1.0 vctrs_0.1.0
[33] Biostrings_2.52.0 gdata_2.18.0
[35] nlme_3.1-140 rtracklayer_1.44.0
[37] xfun_0.7 rvest_0.3.4
[39] mime_0.6 miniUI_0.1.1.1
[41] gtools_3.8.1 XML_3.99-0
[43] MASS_7.3-51.4 zlibbioc_1.30.0
[45] scales_1.0.0 ProtGenerics_1.16.0
[47] hms_0.4.2 promises_1.0.1
[49] SummarizedExperiment_1.14.0 rhdf5_2.28.0
[51] curl_3.3 yaml_2.2.0
[53] memoise_1.1.0 sass_0.1.0.9000
[55] biomaRt_2.40.1 stringi_1.4.3
[57] RSQLite_2.1.1 checkmate_1.9.3
[59] caTools_1.17.1.2 BiocParallel_1.18.0
[61] matrixStats_0.54.0 rlang_0.3.4
[63] pkgconfig_2.0.2 commonmark_1.7
[65] bitops_1.0-6 evaluate_0.14
[67] Rhdf5lib_1.6.0 labeling_0.3
[69] GenomicAlignments_1.20.0 htmlwidgets_1.3
[71] bit_1.1-14 tidyselect_0.2.5
[73] plyr_1.8.4 R6_2.4.0
[75] generics_0.0.2 DelayedArray_0.10.0
[77] DBI_1.0.0 pillar_1.4.1
[79] haven_2.1.0 withr_2.1.2
[81] mgcv_1.8-28 RCurl_1.95-4.12
[83] modelr_0.1.4 crayon_1.3.4
[85] utf8_1.1.4 KernSmooth_2.23-15
[87] progress_1.2.2 locfit_1.5-9.1
[89] grid_3.6.0 readxl_1.3.1
[91] blob_1.1.1 digest_0.6.19
[93] xtable_1.8-4 httpuv_1.5.1
[95] munsell_0.5.0