Ruprecht-Karls-Universität Heidelberg
Fakultät für Biowissenschaften
Bachelorstudiengang Molekulare Biotechnologie

ksdflsdjf
sfdsfd
sfsf

Data Science Project SoSe 2022

Autoreb Max Mustermann, jklsdfksldkfslkdjf
Geburtsort sdfjksafdls
Abgabetermin 20.07.2022

Contents

Thank You

Thank You

1 Introduction

1.1 Cancer

You can cite one or multiple authors. One author (Kumar *et al.*, 2017) and multiple authors (Kumar *et al.*, 2017; Zavidij *et al.*, 2020). Write in **bold** or in *italic* or in both ***bolditalic***. You can also write inline code, e.g. `Seurat::RunUMAP`.

1.2 LUAD

Some information Kumar *et al.* (2017)

1.3 Computational Tools

1.3.1 Gene Set Variation Analysis

1.3.2 Gene Set Enrichment Analysis

1.3.3 UMAP

1.3.4 PCA

1.4 Our Analysis

1.4.1 Pan Cancer

1.4.2 Focused Analysis

1.4.3 Related Work

2 Material and Methods

2.1 TCGA data

What kind of data do we have? ## Used Packages

show a table!

3 Results

3.1 33 tumor types are showing disting clusters in UMAP

hello world!

3.2 blb alsdjflaskdf umap of some tumortypes

Figure generation. You can do it with knitr or with latex formatting. This is knitr:

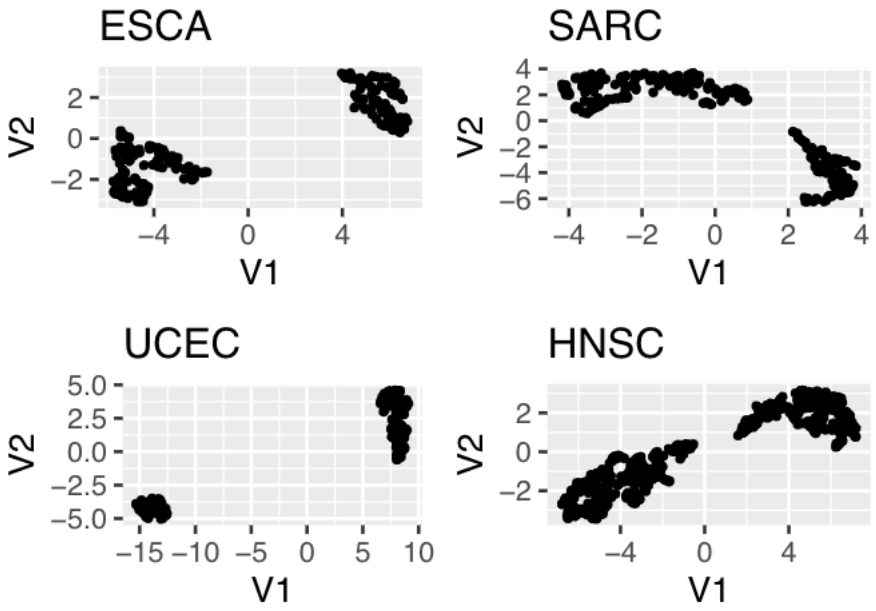

Figure 3.1: Title. Description

easier alternative: this is latex formatting. In Figure 3.2 you can see an UMAP. (+ label your figures, equations etc and then reference with /**??**)

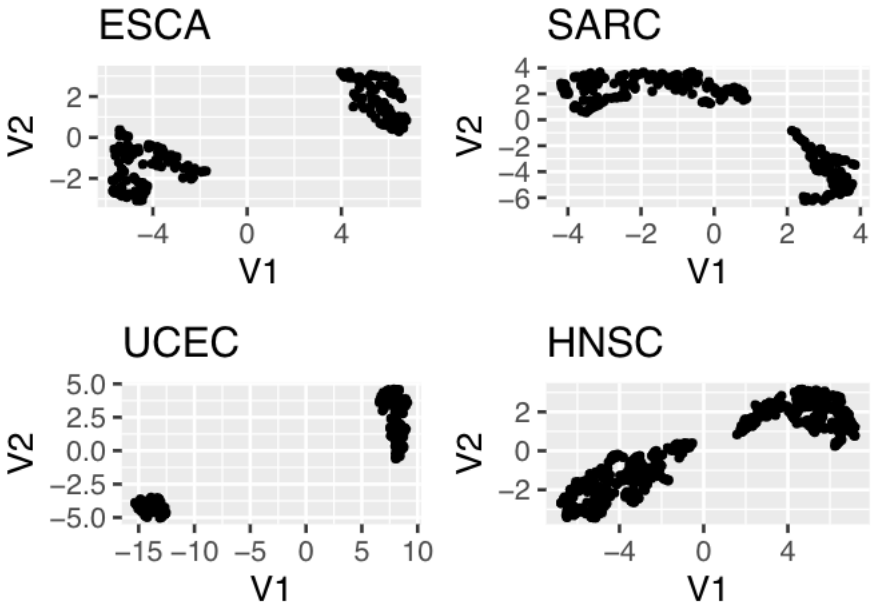

Figure 3.2: Title. Description.

4 Discussion

4.1 Immune pathways are significantly upregulated in X

5 References

Kumar, SK, Rajkumar, V, Kyle, RA, Duin, M van, Sonneveld, P, Mateos, M-V, Gay, F, and Anderson, KC (2017). Multiple myeloma. Nat Rev Dis Primers 3, 17046.
Zavidij, O et al. (2020). Single-cell RNA sequencing reveals compromised immune microenvironment in precursor stages of multiple myeloma. Nat Cancer 1, 493–506.

6 Appendix

6.1 Plots

hello ## Code world

```
#createn einer liste mit allen patienten in dfs sortiert nach krebs
cancers = list();cancers = vector('list',length(table(tcga_anno$cancer_type_abbreviatio
names(cancers) = names(table(tcga_anno$cancer_type_abbreviation))
i=1
for (i in 1:length(cancers)){
  cancers[[i]] = tcga_exp_cleaned[,tcga_anno$cancer_type_abbreviation == names(cancers)
}
#function die einen krebstypen df und genesets als input nimmt und ein df mit pvalues
enrichment = function(expressiondata, genesets = genesets_ids){
  ESmatrix = sapply(genesets, FUN = function(x){
    ins = na.omit(match(x,rownames(expressiondata)))#indices der gene im aktuellen set
    outs = -ins#indices der gene nicht im aktuellen set
    #gibt einen vektor der für jeden patienten den pval für das aktuelle gene enthält
    res = NULL
    for (i in 1:ncol(expressiondata)){#testet für jeden patienten
      res[i] = wilcox.test(expressiondata[ins,i],expressiondata[outs,i],'two.sided')$p.
    }
    return(res)
  })
  row.names(ESmatrix) = colnames(expressiondata); return(ESmatrix)
}
pvalueslist = lapply(cancers, enrichment)#für die tests für jeden krebstypen durch
```

```
get_top10pathways_from_pvalues = function(df_p_values, length_genesets) {
```

```
require(ggplot2)

results <- list()

df_p_values_log10 <- -log10(as.data.frame(df_p_values))

mean_pathway <- as.data.frame(apply(df_p_values_log10, 1, mean))
rownames(mean_pathway) <- rownames(df_p_values_log10)

ordered_score <- mean_pathway[order(-mean_pathway[ ,1]), 1]
top_10 <- data.frame(ordered_score[1:10])
colnames(top_10) <- "mean_pathway"

ordered_names <- order(-mean_pathway[ ,1])
top_10_names <- ordered_names[1:10]
top_10$pathway_names <- row.names(mean_pathway)[top_10_names]

results[[1]] <- top_10

results[[2]] <- ggplot(data = top_10, aes(x = mean_pathway, y = reorder(pathway_names
  geom_bar(stat = "identity")+
  coord_cartesian(xlim =c(3, 3.75))+
  labs(title = names(df_p_values),
       x = "mean p-value pathway",
       y = "pathway name")

pathway_size <- order(-mean_pathway[ ,1])
top_10_size <- pathway_size[1:10]
top_10$pathway_size <- length_genesets[top_10_size]

results[[3]] <- ggplot(data = top_10, aes(x = mean_pathway, y = reorder(pathway_names
                                                                        mean_pathway)
  geom_point(aes(size = pathway_size))+
  labs(title = names(df_p_values),
       x = "mean p-value pathway",
       y = "pathway name")
```

```
  return(results)
}
```