### Supplemental Information

Yihao Chen, Li-Na He, Yuanzhe Zhang, Yuelong Shu, Di Zhang, Guangchuang Yu, Zhixiang Zuo2024-04-12

#### 1 Installation

To install tigeR package, please enter the following command in R:

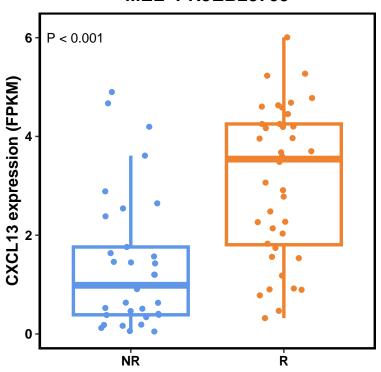
```
if (!requireNamespace("devtools", quietly = TRUE))
  install.packages("BiocManager")
devtools::install_github("YuLab-SMU/tigeR")
devtools::install_github("YuLab-SMU/tigeR.data")
```

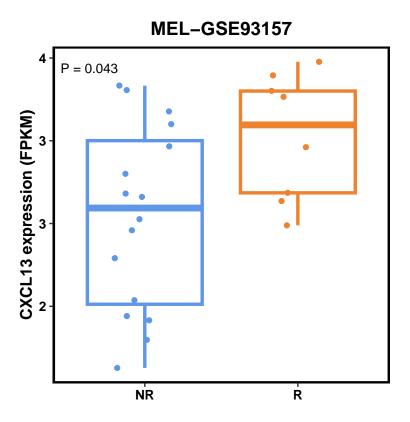
# 2 Evaluating biomarkers associated with immunotherapy response (Figure 3)

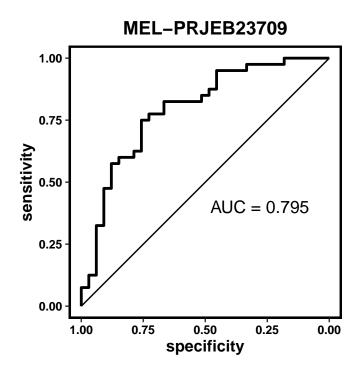
To reproduce the analysis in this document, several extra packages should be library.

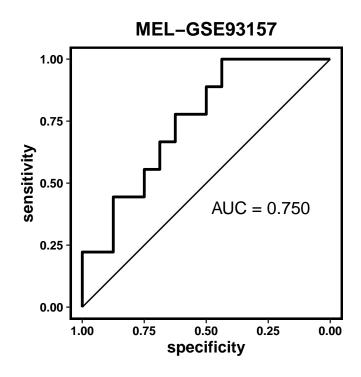
```
library(tigeR)
library(tigeR.data)
library(ggplot2)
library(SummarizedExperiment)
library(patchwork)
```

#### MEL-PRJEB23709

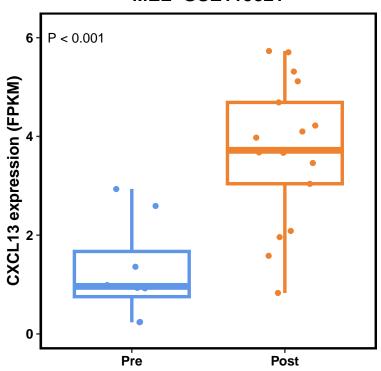




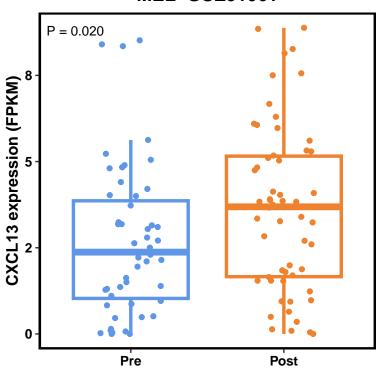


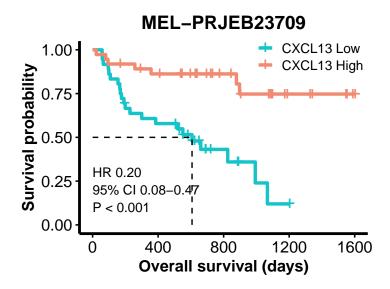


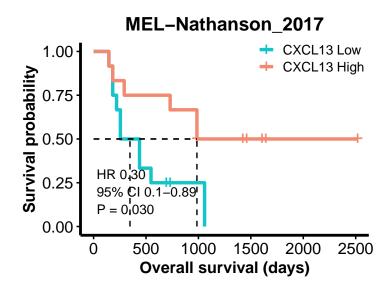
### MEL-GSE115821

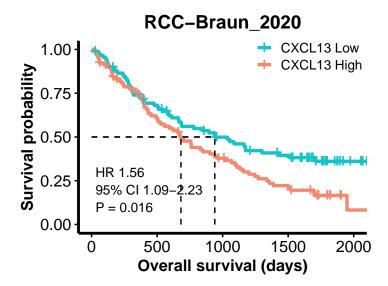


#### MEL-GSE91061

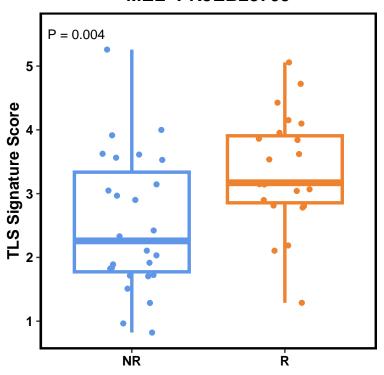


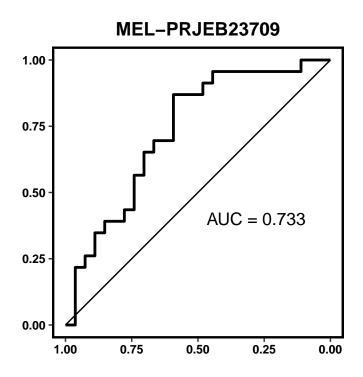


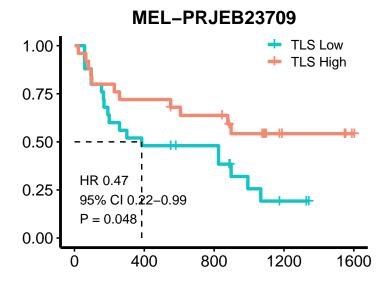


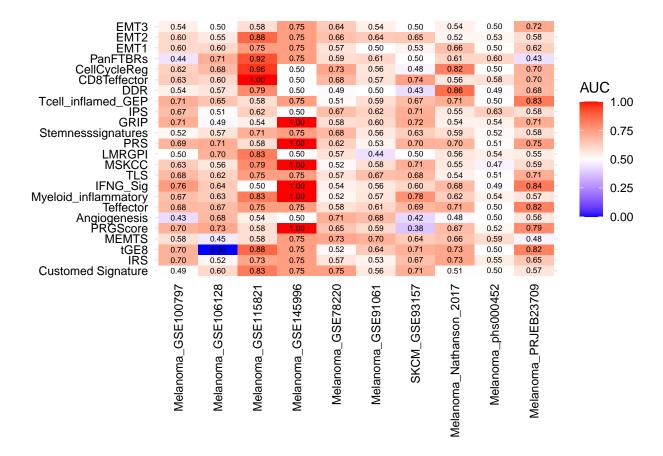


#### MEL-PRJEB23709









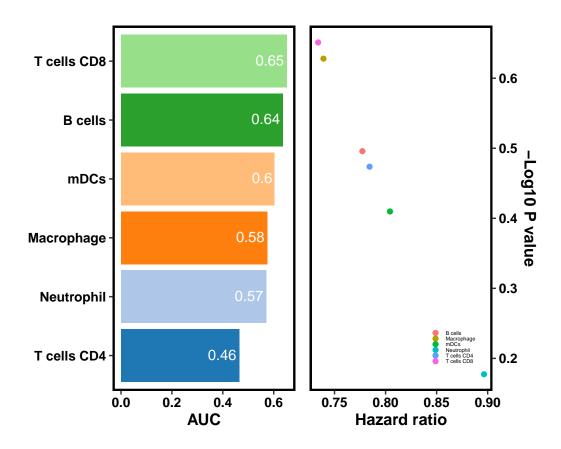
## 3 Identifying tumor microenvironment compositions associated with immunotherapy response (Figure 4)

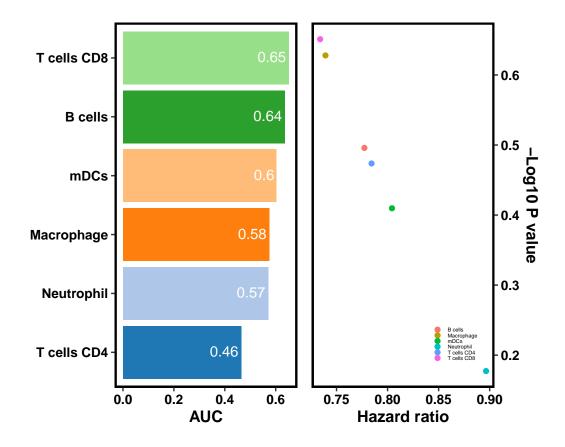
Tumor microenvironment analysis is crucial for understanding tumor immune evasion mechanisms and predicting the efficacy of imunotherapy. It provides insights into the interactions between tumor cells, immune cells, and stromal components that influence the anti-tumor immune response and treatment outcomes. Here we analyze the relation between tumor microenvironment and immunotherapy outcome.

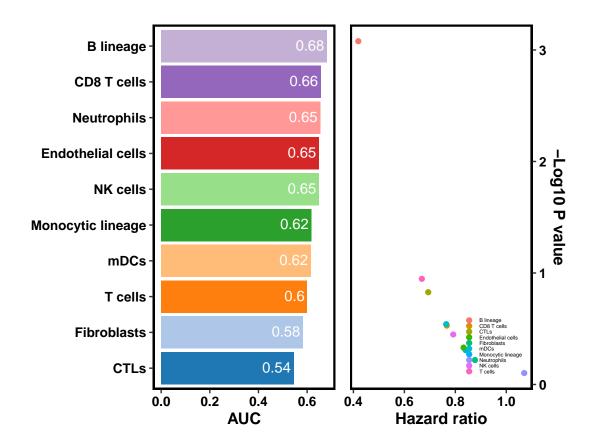
```
## MCPCounter
frac3 <- deconv_TME(MEL_GSE78220,method="MCPCounter")</pre>
cell3 <- c("CTLs", "Fibroblasts", "T cells", "mDCs", "Monocytic lineage",
           "NK cells", "Endothelial cells", "Neutrophils", "CD8 T cells", "B lineage")
pie3 <- fraction_pie(cell_name_filter(frac3),</pre>
                      feature=factor(cell3, levels = cell3))
## xCell
frac4 <- deconv_TME(MEL_GSE78220,method="xCell")</pre>
cell4 <- c("Tgd cells", "CD8+ Tem", "Osteoblast", "Megakaryocytes", "CD8+ Tcm",</pre>
            "ly Endothelial cells", "Eosinophils", "mv Endothelial cells",
           "Endothelial cells", "Smooth muscle")
pie4 <- fraction_pie(cell_name_filter(frac4),</pre>
                      feature=factor(cell4, levels = cell4))
## IPS
frac5 <- deconv_TME(MEL_GSE78220,method="IPS")</pre>
cell5 <- c("IPS", "MHC", "CP", "AZ", "SC", "EC")
bar5 <-
draw_bar(cell_name_filter(frac5),
         feature=factor(cell5, levels = cell5))
frac6 <- deconv_TME(MEL_GSE78220,method="epic")</pre>
cell6 <- c("NKcells", "CD4_Tcells", "CD8_Tcells", "Bcells", "CAFs",</pre>
           "Macrophages", "Endothelial", "otherCells")
pie6 <- fraction_pie(cell_name_filter(frac6),</pre>
                      feature=factor(cell6, levels = cell6))
## ESTIMATE
frac7 <- deconv TME(MEL GSE78220,method="ESTIMATE")</pre>
cell7 <- c("StromalScore", "ESTIMATEScore", "TumorPurity", "ImmuneScore")</pre>
bar7 <- draw_bar(cell_name_filter(frac7),</pre>
                  feature=factor(cell7, levels = cell7))
## ABIS
frac8 <- deconv_TME(MEL_GSE78220,method="ABIS")</pre>
cell8 <- c("Macrophages M1", "Plasma cells", "CD4+ T memory activated",
            "T cells follicular helper", "DCs activated", "T cells CD8",
            "NK cells activated", "Neutrophils", "Tregs", "B cells naive")
pie8 <- fraction_pie(cell_name_filter(frac8),</pre>
                      feature=factor(cell8, levels = cell8))
## ConsensusTME
frac9 <- deconv_TME(MEL_GSE78220,method="ConsensusTME")</pre>
cell9 <- c("T cells CD4", "NK cells", "Yd T cells", "Mast cells",</pre>
           "T cells CD8", "Immune Score", "Tregs", "Plasma cells",
           "B cells", "Endothelial")
pie9 <- fraction_pie(cell_name_filter(frac9),</pre>
                      feature=factor(cell9, levels = cell9))
## quanTIseq
frac10 <- deconv TME(MEL GSE78220,method="quanTIseq")</pre>
cell10 <- c("Neutrophils", "T cells CD4", "DCs", "Tregs", "Other", "B cells",
            "Macrophages.M2", "Macrophages.M1", "T cells CD8", "Monocytes")
pie10 <- fraction_pie(cell_name_filter(frac10),</pre>
                       feature=factor(cell10,levels = cell10))
bar_IPS <-
  gridExtra::grid.arrange(bar5[[1]],bar5[[2]],bar5[[3]],
                           bar5[[4]],bar5[[5]],bar5[[6]],nrow=1)
```

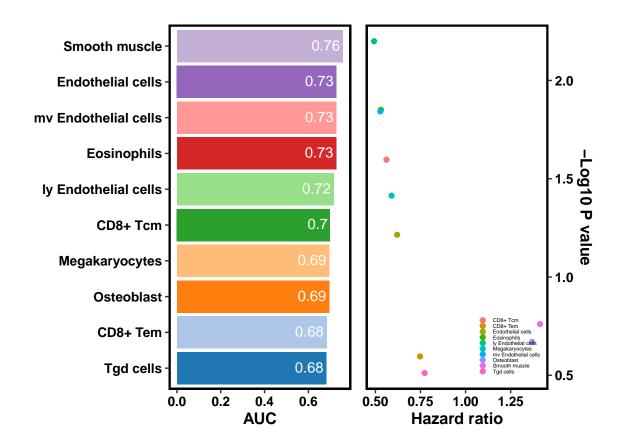
```
bar_ESTIMATE <-</pre>
  gridExtra::grid.arrange(bar7[[1]],bar7[[2]],
                            bar7[[3]],bar7[[4]],nrow=1)
grob_list <-</pre>
  list(pie1, pie2, pie3, pie4, pie6, pie8, pie9, pie10,
       bar_IPS, bar_ESTIMATE)
gridExtra::grid.arrange(grobs = grob_list, ncol = 1,
                          heights = c(1, 1, 1, 1, 1, 1, 1, 1, 2, 2))
                                 ■ CTLs ■ Fibroblasts ■ T cells ■ mDCs ■ Monocytic lineage ■ NK cells ■ Endothelial cells ■ Neutrophils ■ CD8 T cells ■ B lineage
## TIMER
TM <- deconv_TME(MEL_GSE91061,method = "TIMER")</pre>
```

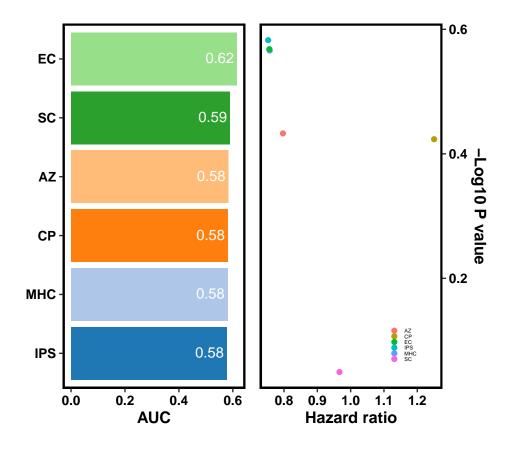
## Found 125 genes with uniform expression within a single batch (all zeros); these will ## not be adjusted

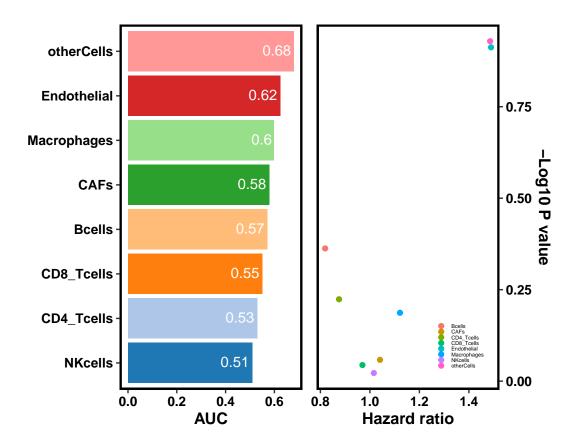


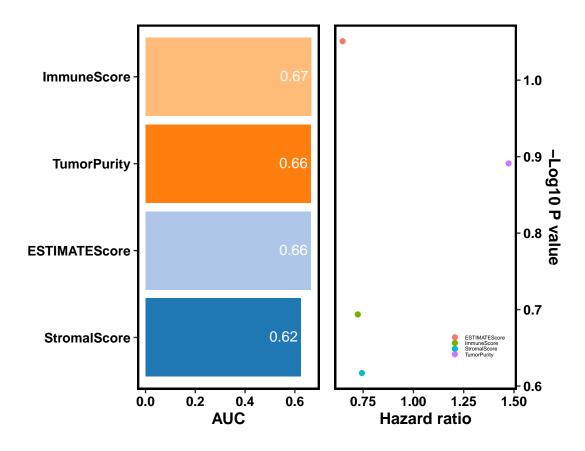


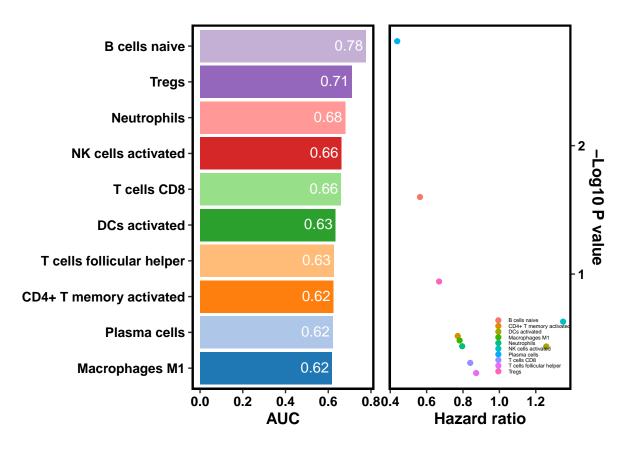




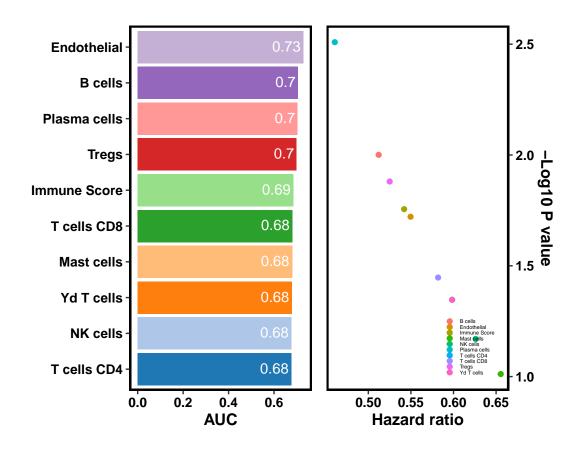


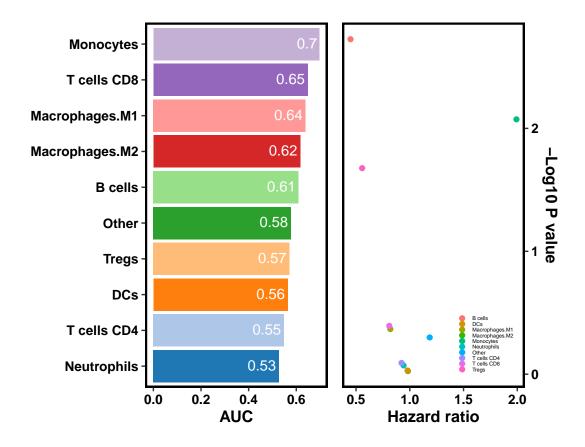


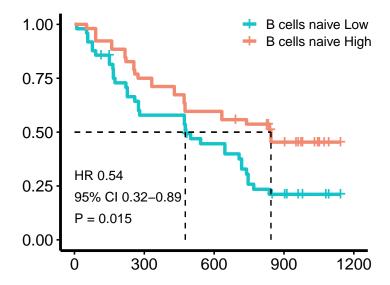


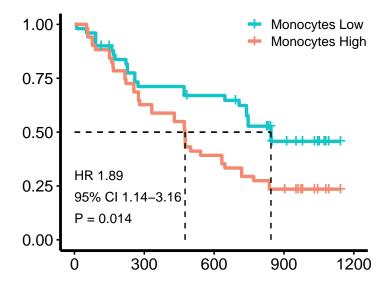


```
## ConsensusTME
CTM <- deconv_TME(MEL_GSE91061,
                  method="ConsensusTME")
## Producing ConsensusTME Estimates Using The Following Parameters:
## Statistical Framework: "ssgsea"
## Gene Sets For Cancer Type: "SKCM"
## Sample Size: 109
## Estimating ssGSEA scores for 19 gene sets.
## [1] "Calculating ranks..."
## [1] "Calculating absolute values from ranks..."
##
##
## [1] "Normalizing..."
CTM1 <- cell_name_filter(CTM)</pre>
CTM_SE <- SummarizedExperiment(assays=SimpleList(CTM1),</pre>
                                colData=MEL_GSE91061@colData)
fig.4J <- browse_biomk(SE=CTM_SE)</pre>
fig.4J
```









### 4 Constructing immunotherapy response prediction model (Figure 5)

We then construct a Prediction Model module for robust immunotherapy response prediction.

library(tigeR)

TIMER\_SE(type="SKCM") %>%

to1()

```
library(SummarizedExperiment)
library(magrittr)
# Data preparation
SE_origin <- cbind(MEL_GSE91061,</pre>
                   MEL_phs000452,
                   MEL_Nathanson_2017)
exp <-
  dataProcess_SE(data_standardization(SE_origin, type = c(1,2,3,4)),
                 Signature = NULL,rmBE = TRUE,response_NR=TRUE,
                 turn2HL=FALSE)
## Found 24344 genes with uniform expression within a single batch (all zeros); these will
## not be adjusted
cf <-
  dataProcess_SE(data_standardization(SE_origin,type=c(1,2)),
                        Signature = NULL,rmBE = FALSE,response_NR=TRUE,
                        turn2HL=FALSE) %>%
```

## Found 7 genes with uniform expression within a single batch (all zeros); these will not ## be adjusted for

```
sg <-
  dataProcess_SE(data_standardization(SE_origin,type=c()),
                  Signature = NULL,rmBE = FALSE,response_NR=TRUE,
                  turn2HL=FALSE) %>%
  score_biomk_SE(sg_SE,PT_drop = FALSE) %>%
  to1()
final_SE <- rbind(exp,cf,sg)</pre>
set.seed(8)
idx <- sample(1:275,size=187)</pre>
SE_obj_train <- final_SE[,idx]</pre>
SE_obj_test <- final_SE[,-idx]</pre>
m1 <- t(
  apply(assay(exp)[,idx], 1, function(x){
    if(length(which(x==0))*5>length(x))
      return(rep(NA,length(x)))
    else
      return(x)
  }))
m2 <- t(
  apply(m1, 1, function(x){
    # x[x==0] <- NA
    return(x)
  }))
colnames(m2) <- colnames(assay(exp)[,idx])</pre>
vars <- na.omit(</pre>
  apply(m2,1,var,na.rm=TRUE))
selected_genes <- names(vars[vars>0.1])
NB_model <- build_Model(SE_obj_train,</pre>
                          Model = "NB",
                          feature_genes = c(selected_genes,
                                             rownames(cf),
                                             rownames(sg)),
                          rmBE = FALSE,
                          response_NR = TRUE,
                          laplace=0)
RF_model <- build_Model(SE_obj_train,</pre>
                          Model = "RF",
                          feature_genes = c(selected_genes,
                                             rownames(cf),
                                             rownames(sg)),
                          rmBE = FALSE,
                          response NR = TRUE)
SVM_model <- build_Model(SE_obj_train,</pre>
                           Model = "SVM",
```

```
feature_genes = c(selected_genes,
                                             rownames(cf),
                                             rownames(sg)),
                          rmBE = FALSE,
                          response_NR = TRUE)
CC_model <- build_Model(SE_obj_train,</pre>
                         Model = "CC",
                         feature_genes = c(selected_genes,
                                            rownames(cf),
                                            rownames(sg)),
                         rmBE = FALSE,
                         response_NR = TRUE)
ADB_model <- build_Model(SE_obj_train,
                          Model = "ADB",
                          feature_genes = c(selected_genes,
                                             rownames(cf),
                                             rownames(sg)),
                          rmBE = FALSE,
                          response_NR = TRUE)
LGB_model <- build_Model(SE_obj_train,</pre>
                          Model = "LGB",
                          feature_genes = c(selected_genes,
                                             rownames(cf),
                                             rownames(sg)),
                          rmBE = FALSE,
                          response_NR = TRUE)
```

