

# Drug Screening Signature Generation

Sara Gosline

10/11/2019

## Drug sensitivity data

The first we want to collect the drug sensitivity data from the NF Data Portal. We collect all the data from digoxin screens, including those from the pNF data.

This analysis tries to build a gene signature of digoxin response using GSVA signatures instead of gene data.

```
#tab<-syn$tableQuery("SELECT distinct model_name,response_type,response,response_unit,symptom_name,disease_name")
tab<-syn$tableQuery("SELECT distinct model_name,response_type,response,response_unit,symptom_name,disease_name")
#datatable(tab)

##lets get drugs that are tested in both NF1 and knockout and have more than one sample
##of each
drugs<-tab%>%subset(response_type!='IC50_abs')%>%
  group_by(response_type,DT_explorer_internal_id)%>%
  mutate(numdis=n_distinct(disease_name))%>%
  mutate(numSamps=n_distinct(model_name))%>%
  subset(numdis==2)%>%
  subset(numSamps>4)

#now look for specific drugs that have differential response in NF1 disease!
dtab<-drugs%>%tibble::rowid_to_column()%>%
  mutate(nresponse=as.numeric(response))%>%
  spread(key=disease_name,value=nresponse)%>%
  group_by(response_type,DT_explorer_internal_id)%>%
  mutate(pval=wilcox.test(`no disease`,NF1)$p.value)%>%
  ungroup()%>%group_by(response_type)%>%
  mutate(correctedP=p.adjust(pval))%>%ungroup()

##now we have some that are significant (not passing testing)
sig.drugs<-subset(dtab,pval<0.005)%>%
  subset(response_type%in%c('Min_viability','AUC_Simpson'))%>%
  dplyr::select(DT_explorer_internal_id,response_type,pval)%>%distinct()

dmap<-syn$tableQuery(paste0("SELECT distinct DT_explorer_internal_id,name from syn18506947 where DT_explorer_internal_id in (",
  DT_explorer_internal_id,")"))

pvals.with.names<-sig.drugs%>%left_join(dmap,by='DT_explorer_internal_id')
datatable(pvals.with.names)
```

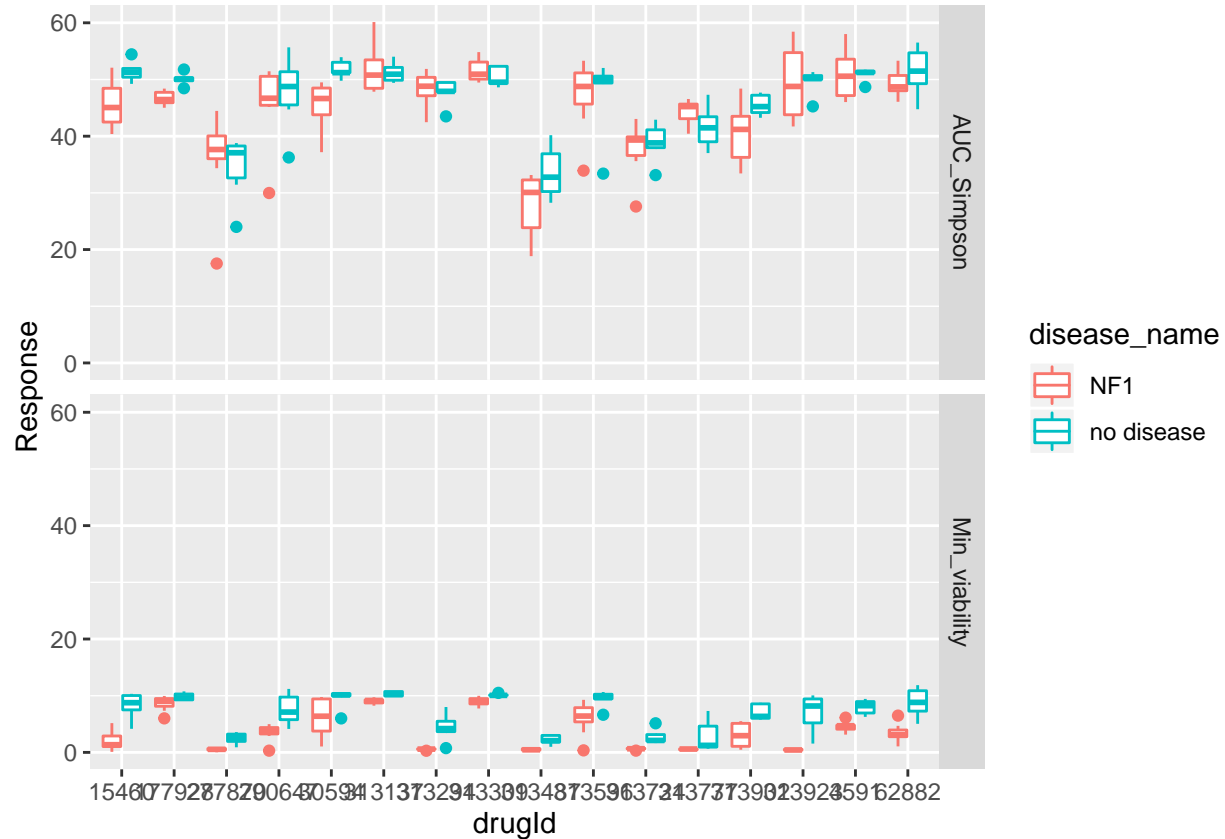
It's important to remember that the 'no symptom' data includes both NF1 knockout cell lines and WT. So we can compare the NF1 disease vs. non-NF1 cells. But this isn't a great comparison since we have the pNFS included. Here are the results without the pNF data

```

auc.data<- subset(tab,DT_explorer_internal_id%in%unique(sig.drugs$DT_explorer_internal_id))%>%mutate(Response_type=
response_type)

p<-ggplot(subset(auc.data,response_type%in%c("AUC_Simpson","Min_viability")))+geom_boxplot(aes(x=drugId,
y=Response,color=disease_name))+facet_wrap(~response_type)
print(p)

```



Now let's dive into individual compounds so we can explore further. maybe with one of their common names

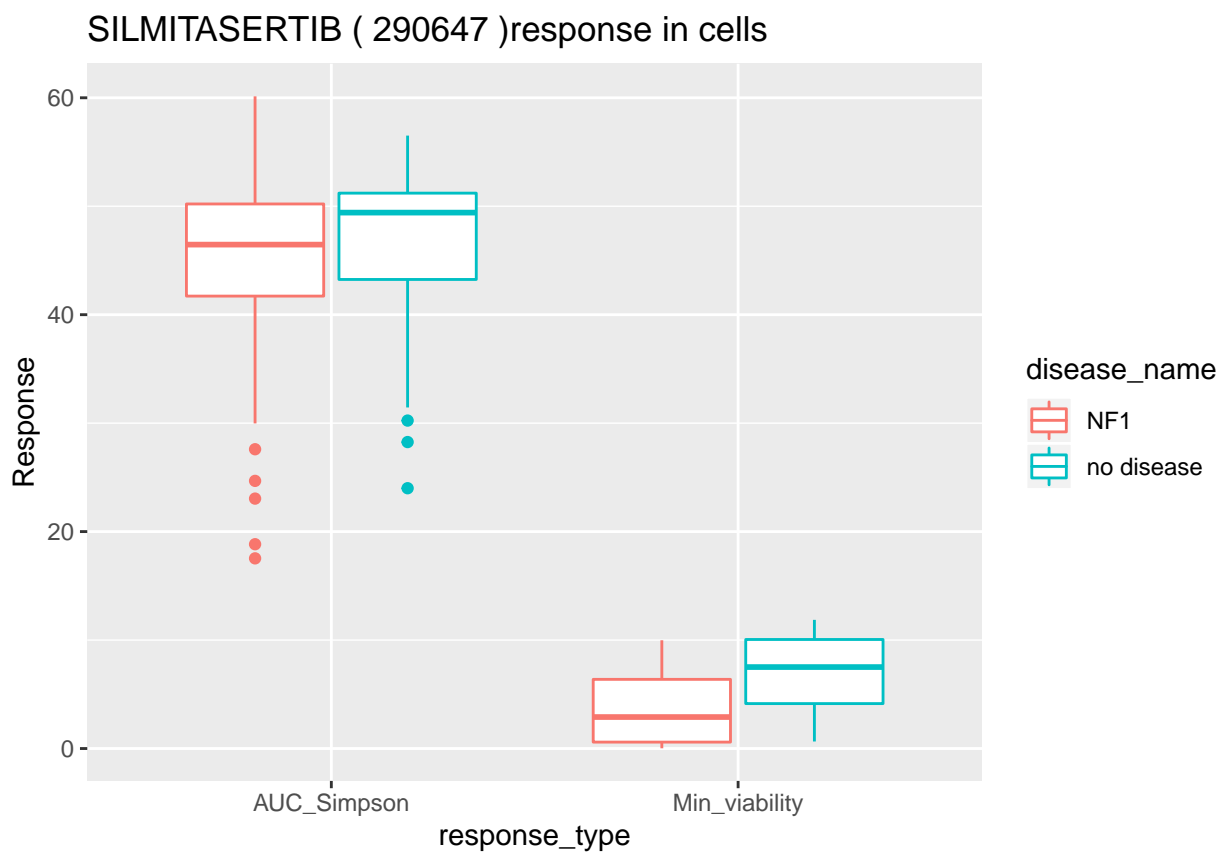
```

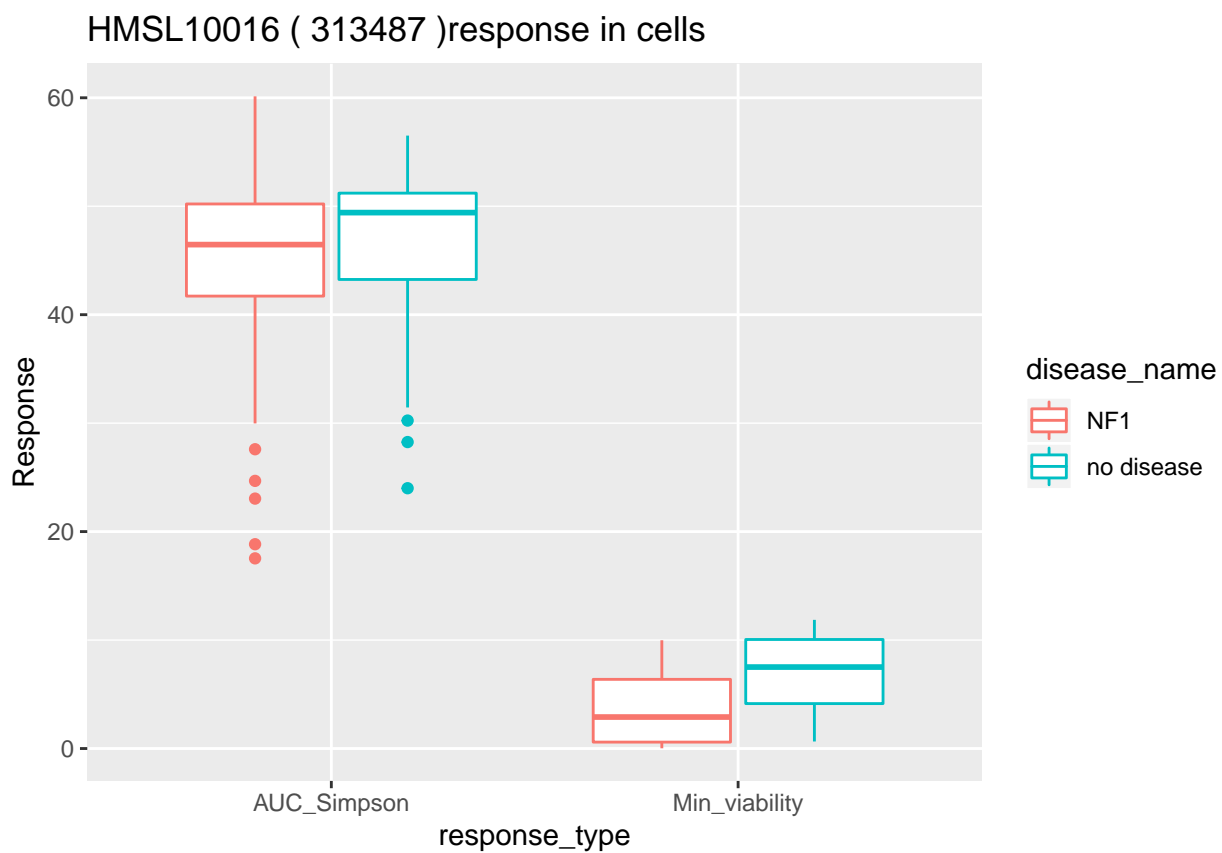
for(comp in unique(auc.data$drugId)){
  ddata<-subset(auc.data,response_type%in%c('AUC_Simpson','Min_viability'))%>%
    subset(drugId==comp)

  dname=dmap$name[which(dmap$DT_explorer_internal_id==comp)[1]]

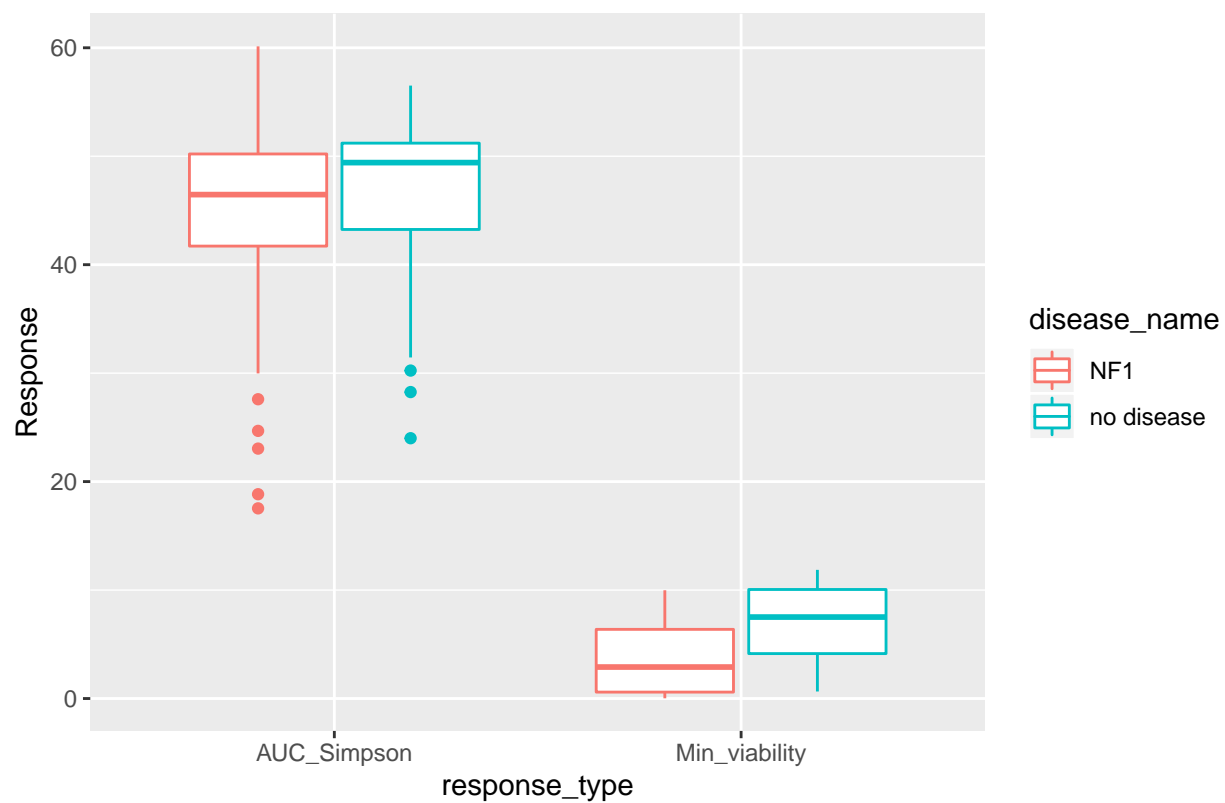
  p<-ggplot(ddata)+geom_boxplot(aes(x=response_type,y=Response,color=disease_name))+ggtitle(paste(dname,
  response_type))
  print(p)
}

```

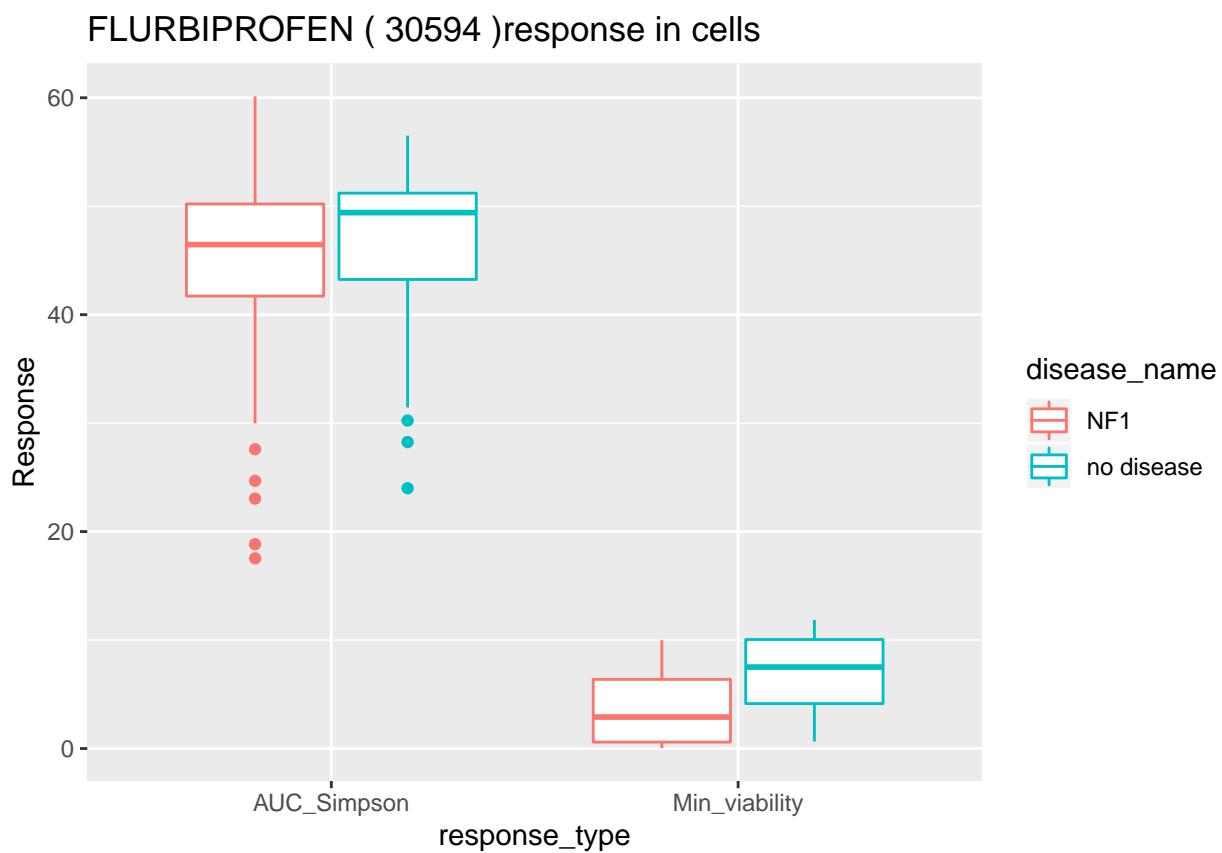


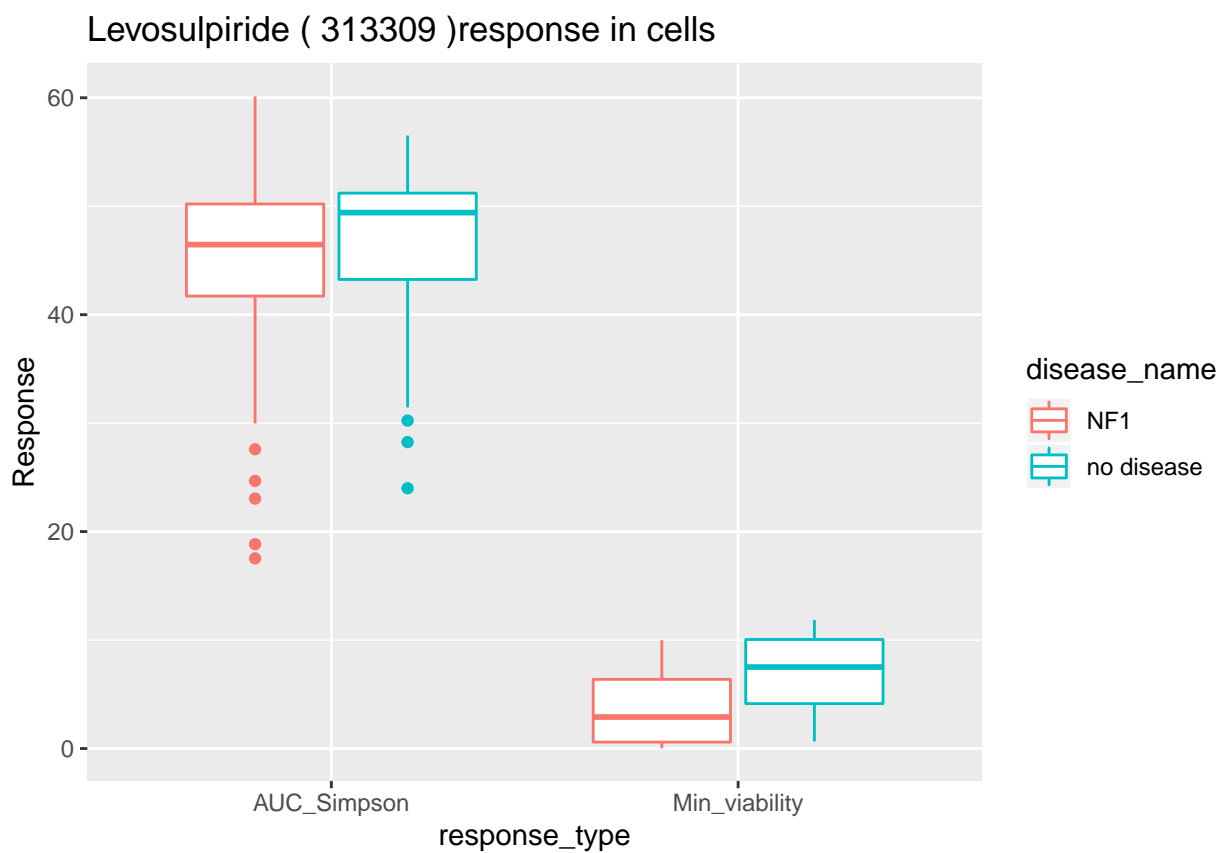


# PF-03716556 ( 313596 )response in cells

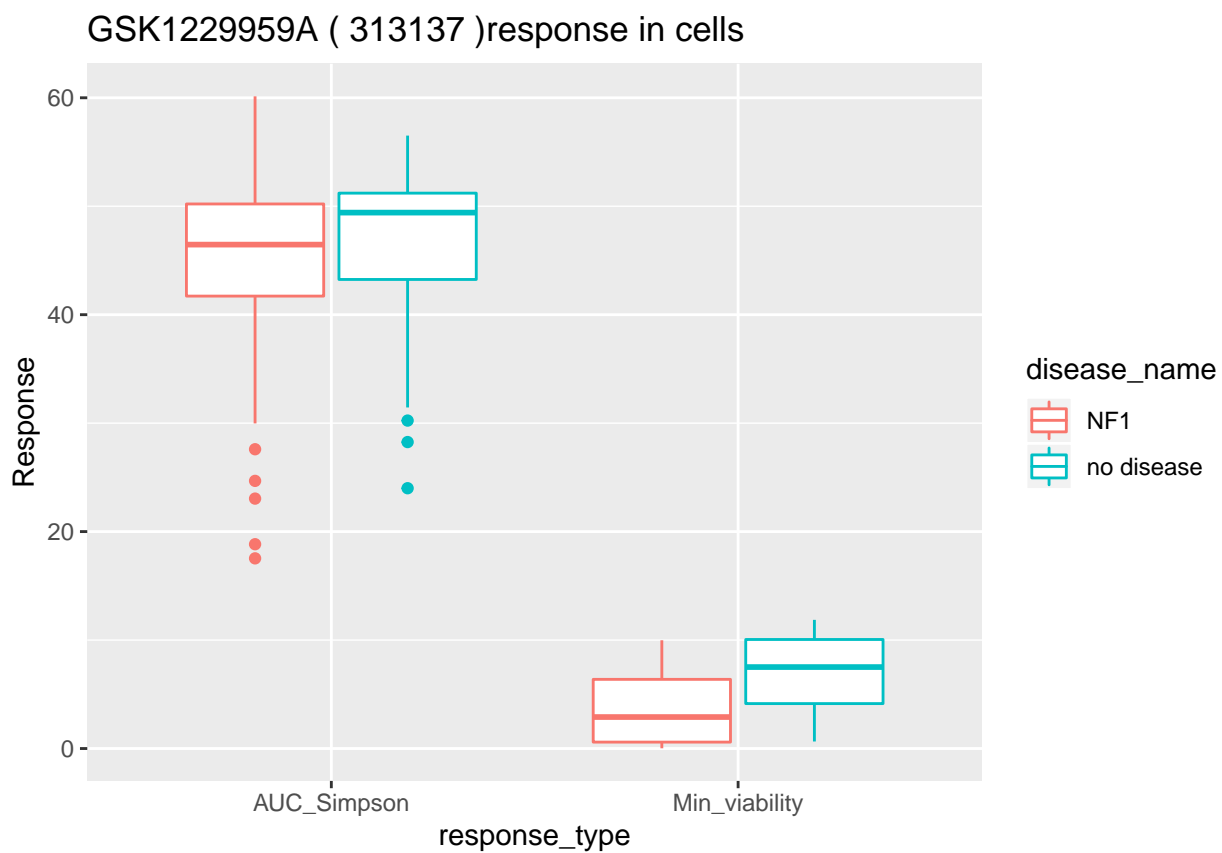


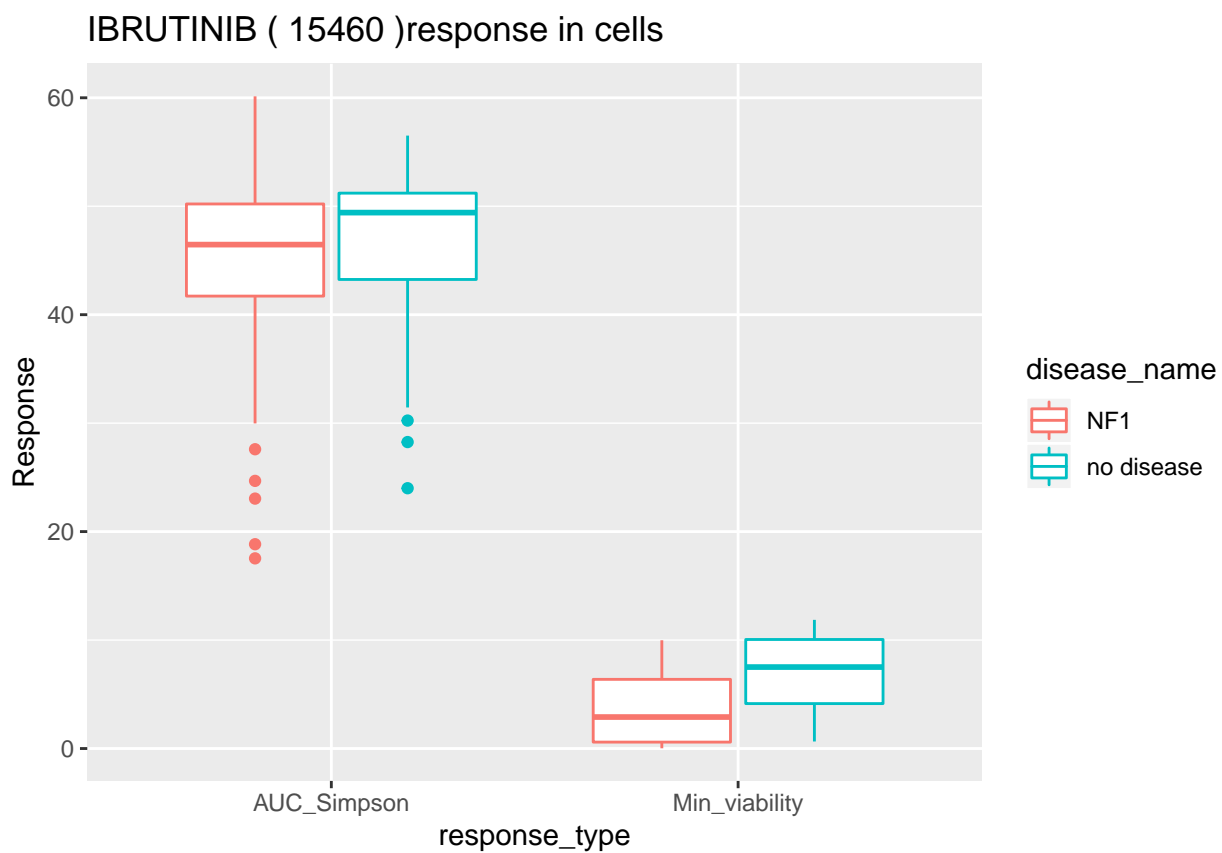


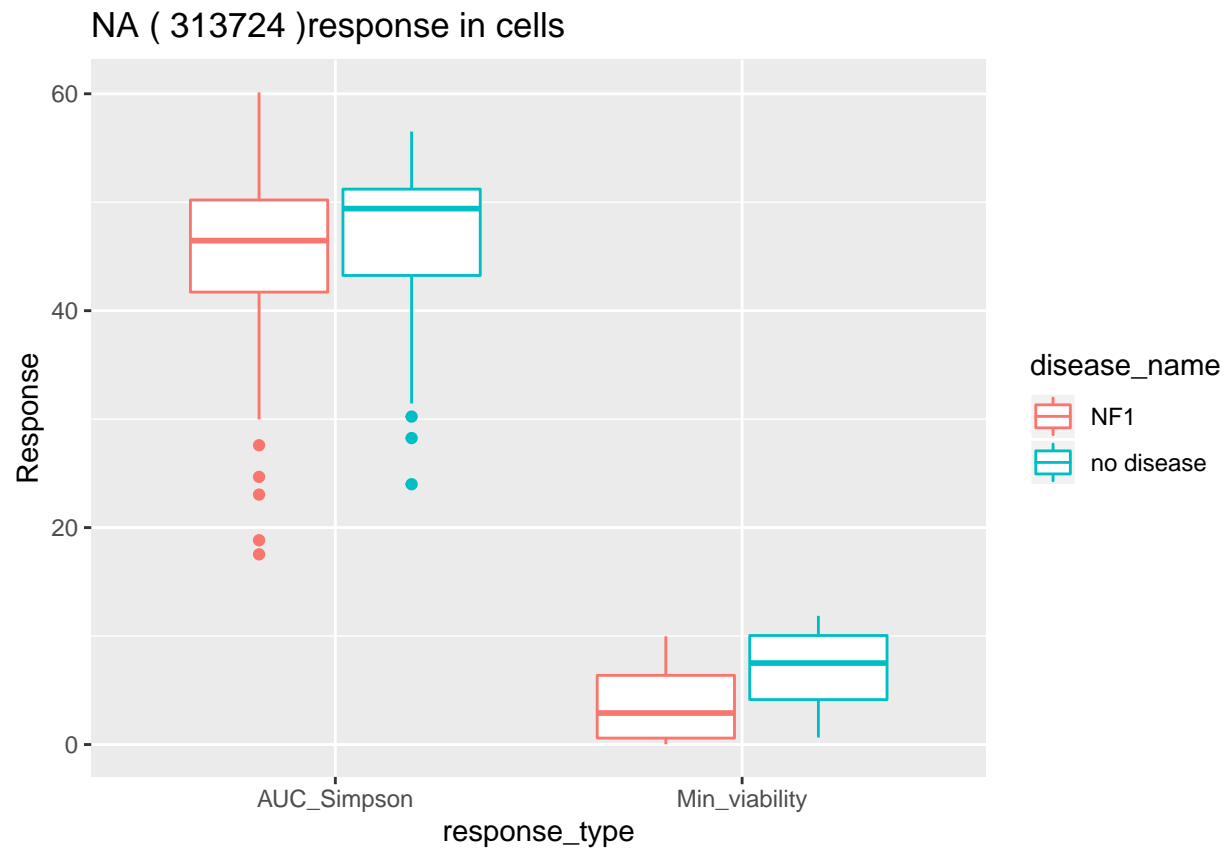


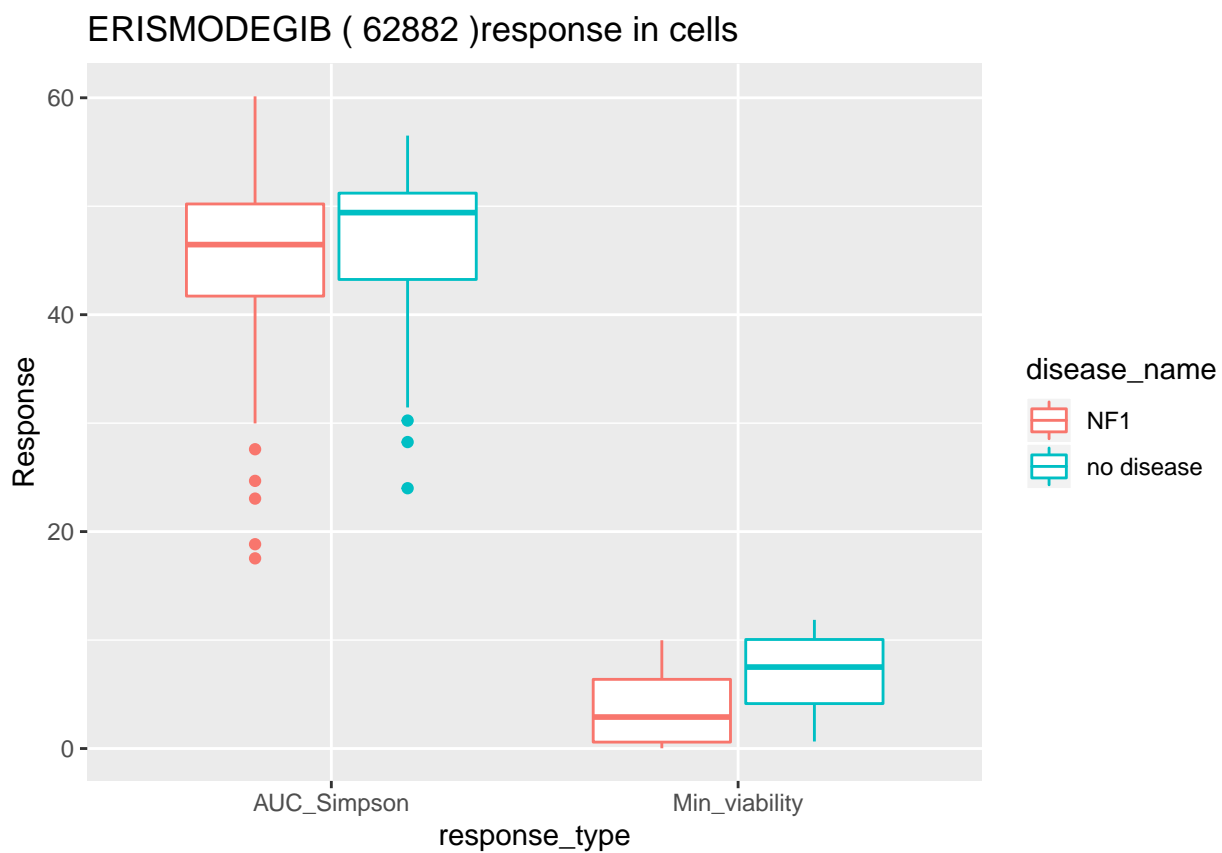


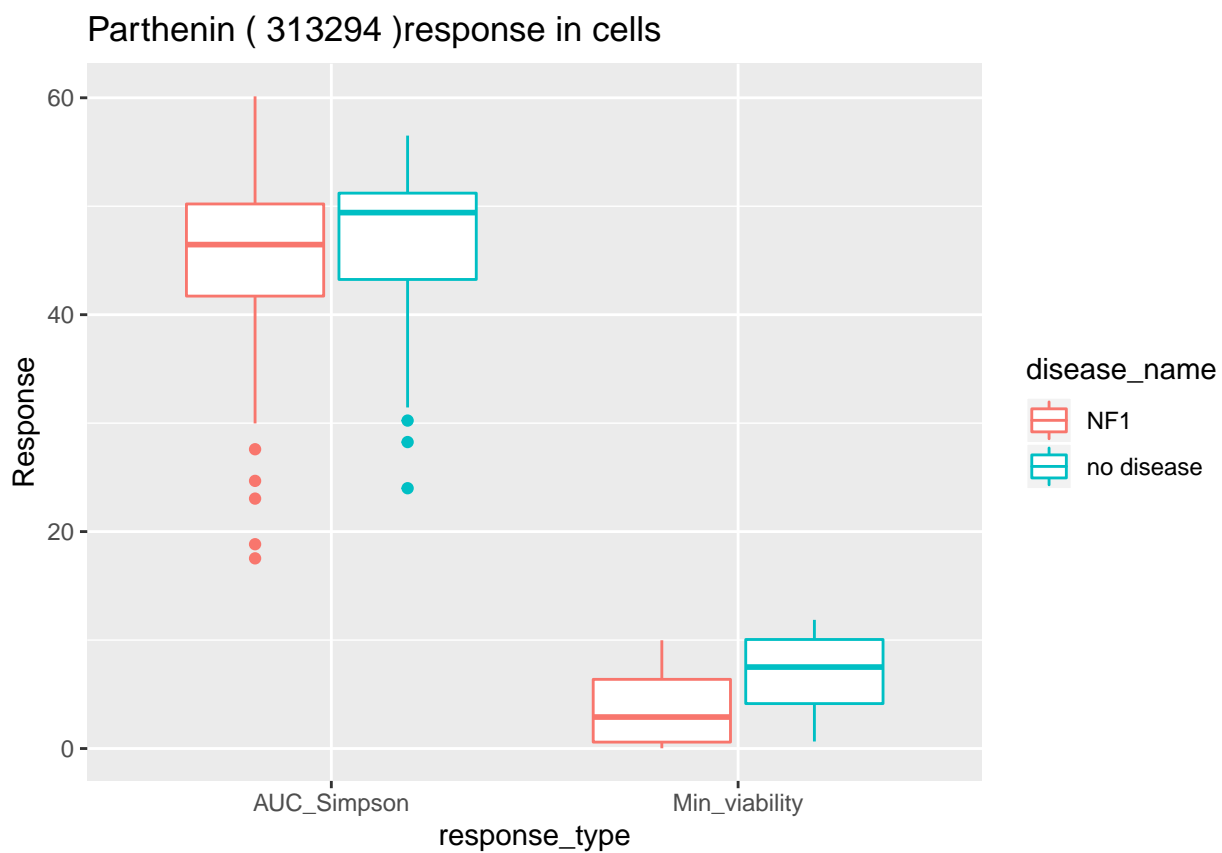


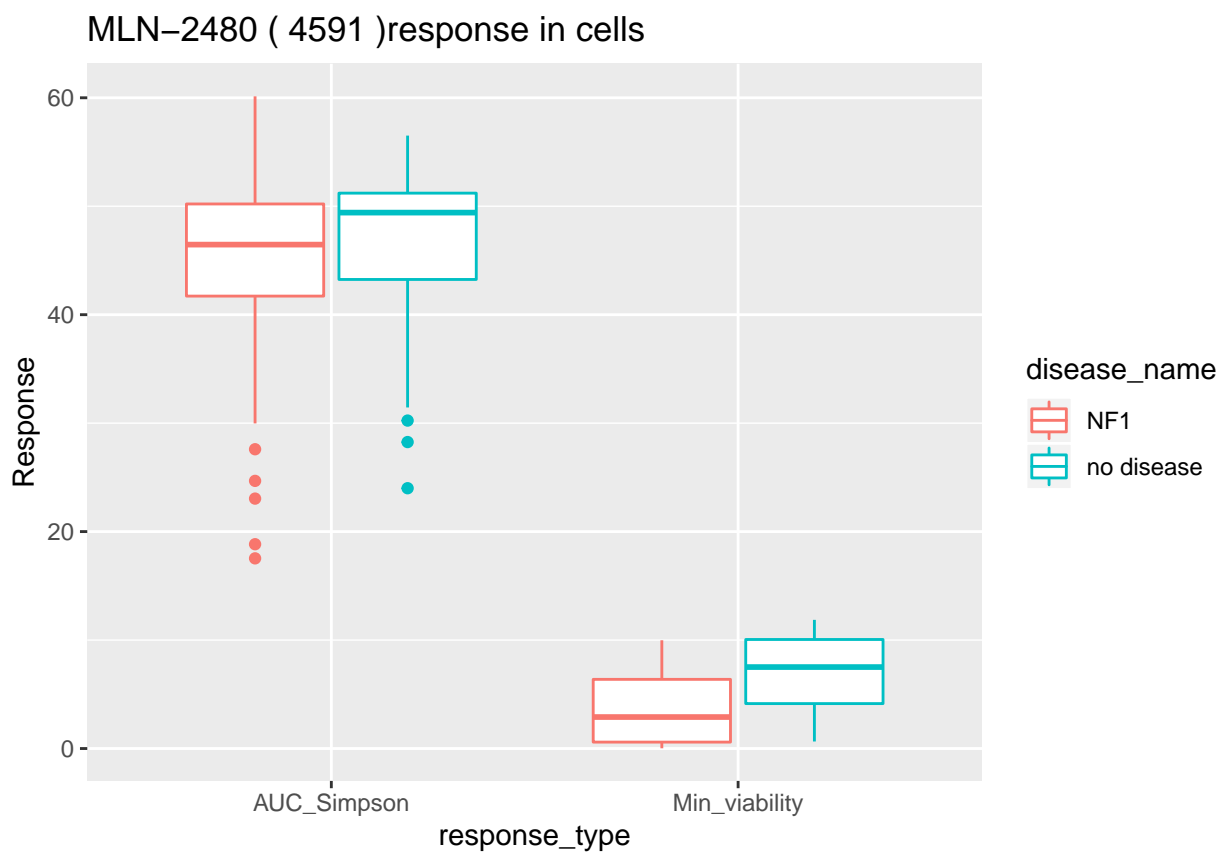




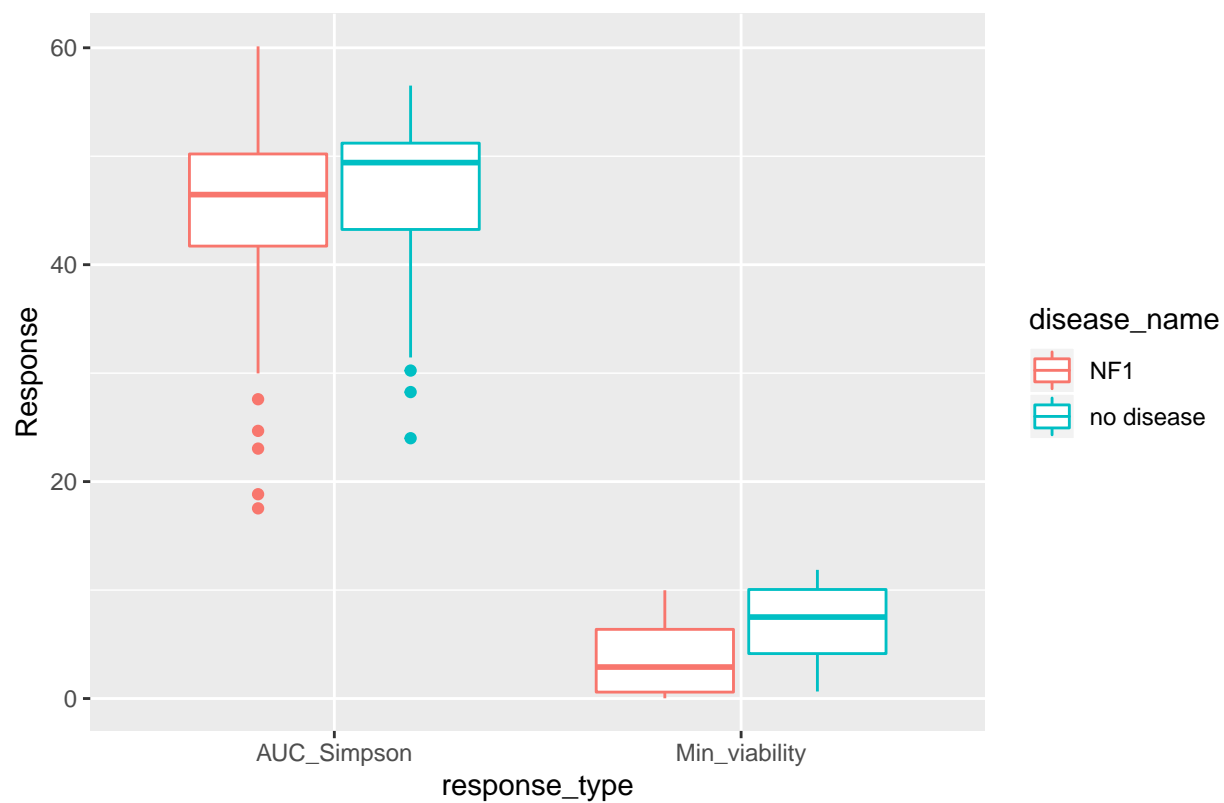


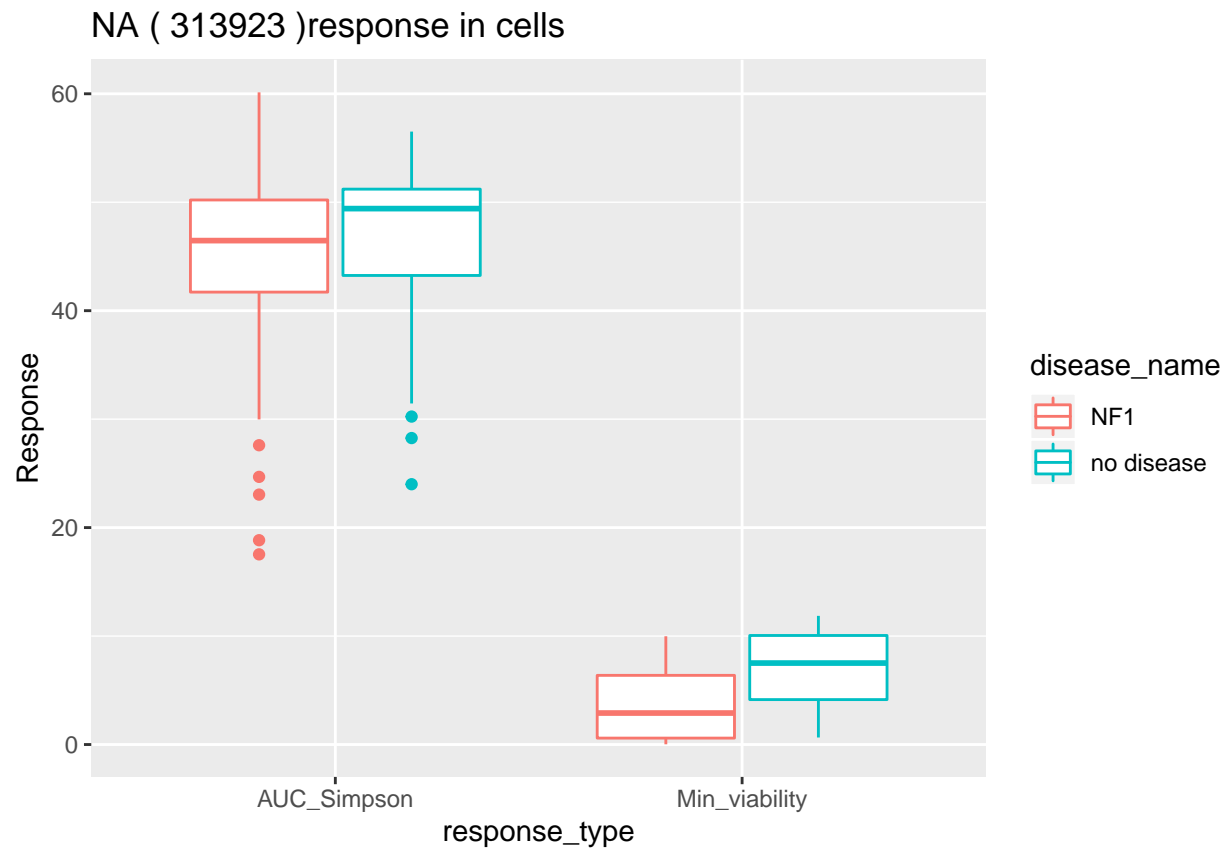




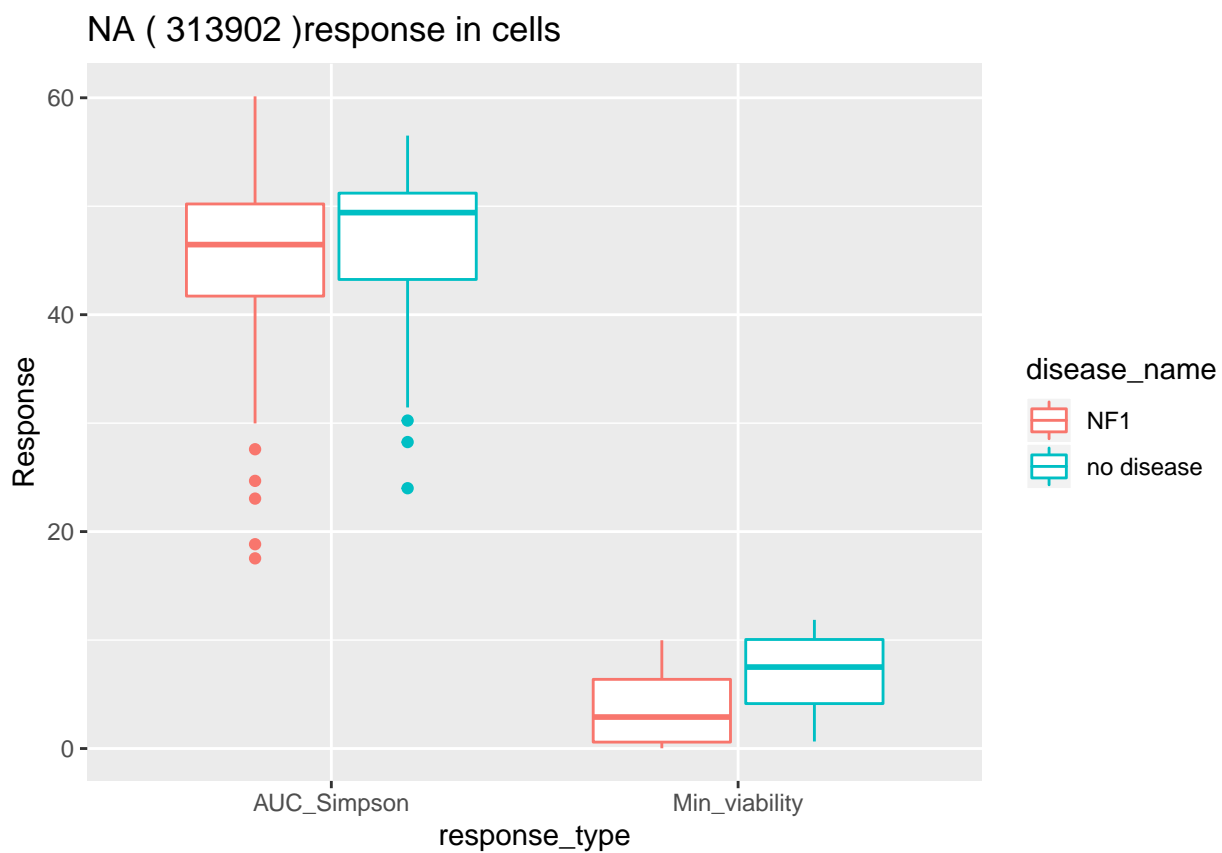


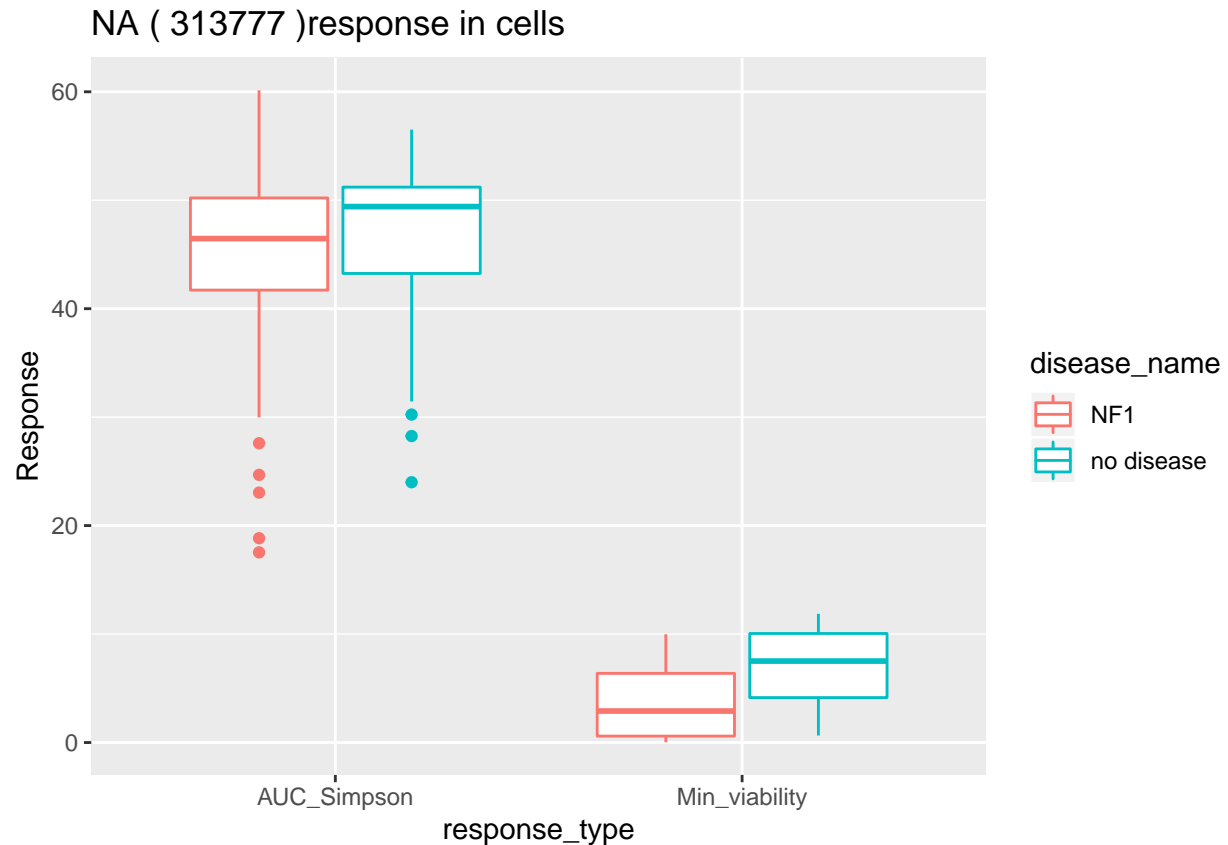
AT-9283 ( 277870 )response in cells











## Gene Expression Data

Now we can get the gene expression data, map to pathways and find pathways that are unique to specific drug responses.

Ok, we have some samples, and now need to compute GSVA on those samples

```
## Loading required package: GSVA
## Loading required package: GSVAdata
## Loading required package: Biobase
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:parallel':
##
##   clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##   clusterExport, clusterMap, parApply, parCapply, parLapply,
##   parLapplyLB, parRapply, parSapply, parSapplyLB
```

```

## The following objects are masked from 'package:dplyr':
##
##   combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##   IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##   anyDuplicated, append, as.data.frame, basename, cbind,
##   colMeans, colnames, colSums, dirname, do.call, duplicated,
##   eval, evalq, Filter, Find, get, grep, grepl, intersect,
##   is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##   paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##   Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##   table, tapply, union, unique, unsplit, which, which.max,
##   which.min

## Welcome to Bioconductor
##
##   Vignettes contain introductory material; view with
##   'browseVignettes()'. To cite Bioconductor, see
##   'citation("Biobase)"', and for packages 'citation("pkgname)".

## Loading required package: GSEABase

## Loading required package: annotate

## Loading required package: AnnotationDbi

## Loading required package: stats4

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##   first, rename

## The following object is masked from 'package:tidyr':
##
##   expand

## The following object is masked from 'package:base':
##
##   expand.grid

```

```
##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
## collapse, desc, slice

## The following object is masked from 'package:purrr':
##
## reduce

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:dplyr':
##
## select

## Loading required package: XML

## Loading required package: graph

##
## Attaching package: 'graph'

## The following object is masked from 'package:XML':
##
## addNode

## The following object is masked from 'package:stringr':
##
## boundary

## Loading required package: hgu95a.db

## Loading required package: org.Hs.eg.db

##

##

## 'select()' returned 1:1 mapping between keys and columns

## Estimating ssGSEA scores for 3000 gene sets.
##
|
|
##
|
|=====| 11%
| 0%Using parallel with 4 cores
```



So we have a lot of data, and i'm not sure what to do with it. For now, we can plot a single drug and the pathways it correlates with. First we plot the AUC.

```
#drugid='277870'
for(drugid in unique(filtered.corvals$DT_explorer_internal_id)){
  filtered.corvals$drugname=drugnames=dmap$name[match(filtered.corvals$DT_explorer_internal_id,dmap$DT_explorer_internal_id)]

  drugname=dmap$name[match(drugid,dmap$DT_explorer_internal_id)]

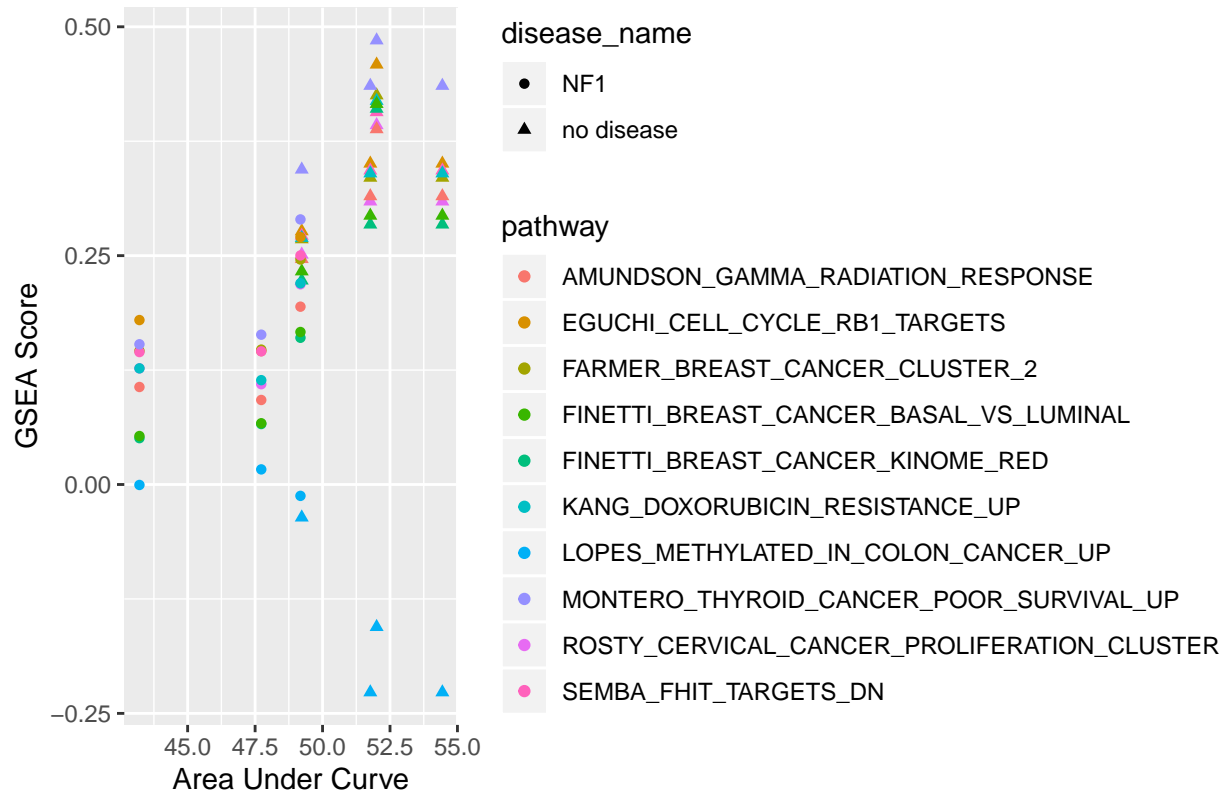
  pathways<- dplyr::select(subset(filtered.corvals,DT_explorer_internal_id==drugid),pathway,corVal)%>%
    arrange(desc(abs(corVal)))%>%
    dplyr::select(pathway)%>%
    unlist()

  if(length(pathways)>10)
    pathways=pathways[1:10]

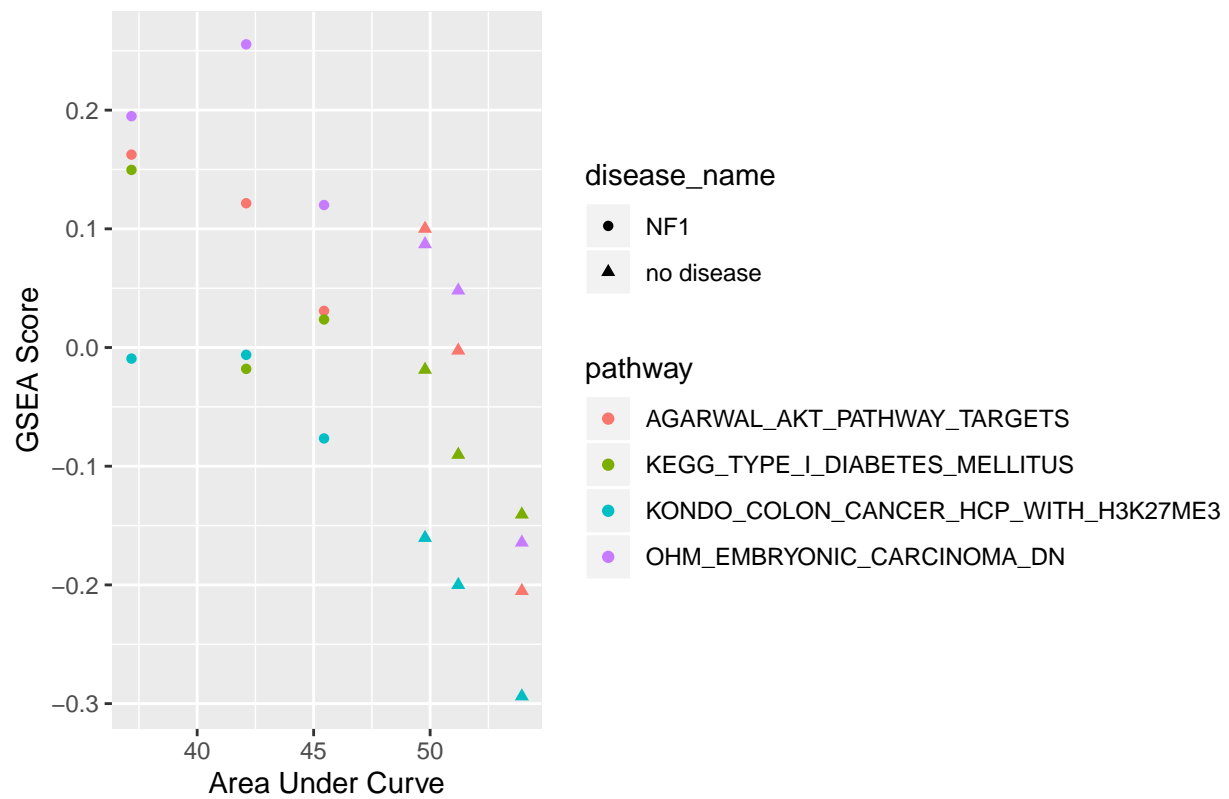
  plot.tab<-dres%>%subset(DT_explorer_internal_id==drugid)%>%
    subset(pathway%in%pathways)%>%
    subset(response_type=='AUC_Simpson')%>%
    mutate('Area Under Curve'=as.numeric(response))%>%
    rename(GSVAscore='GSEA Score')

p<-ggplot(plot.tab)+geom_point(aes(x='Area Under Curve`,y='GSEA Score`,col=pathway,shape=disease_name))+
print(p)
}
```

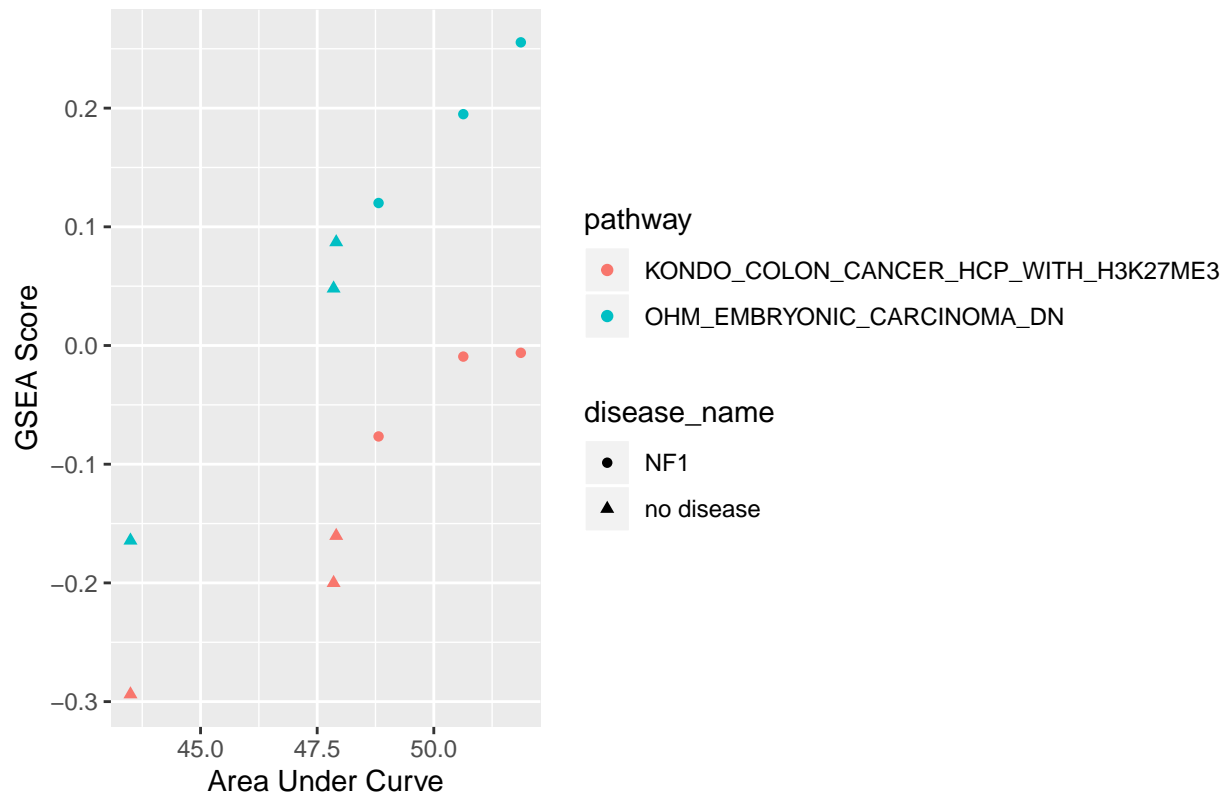
## Pathways correlated with IBRUTINIB treatment



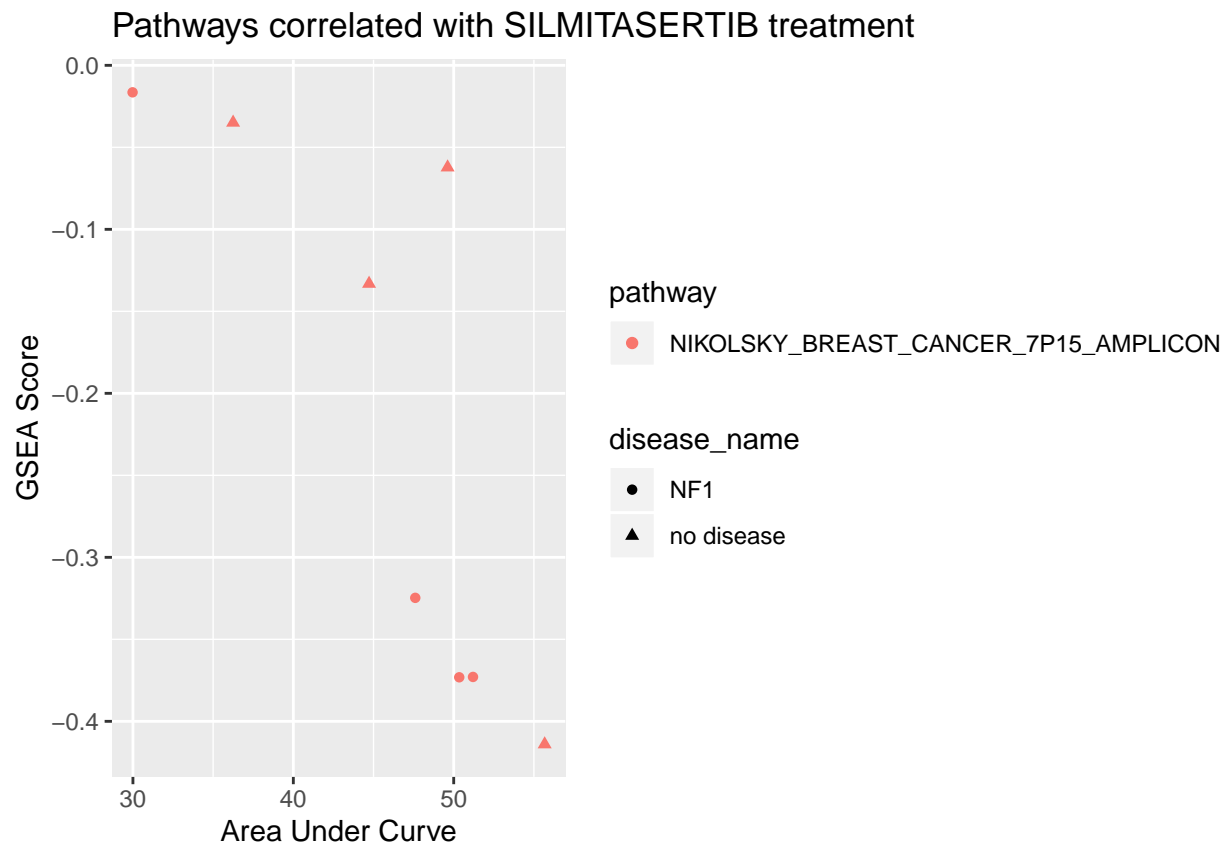
Pathways correlated with FLURBIPROFEN treatment

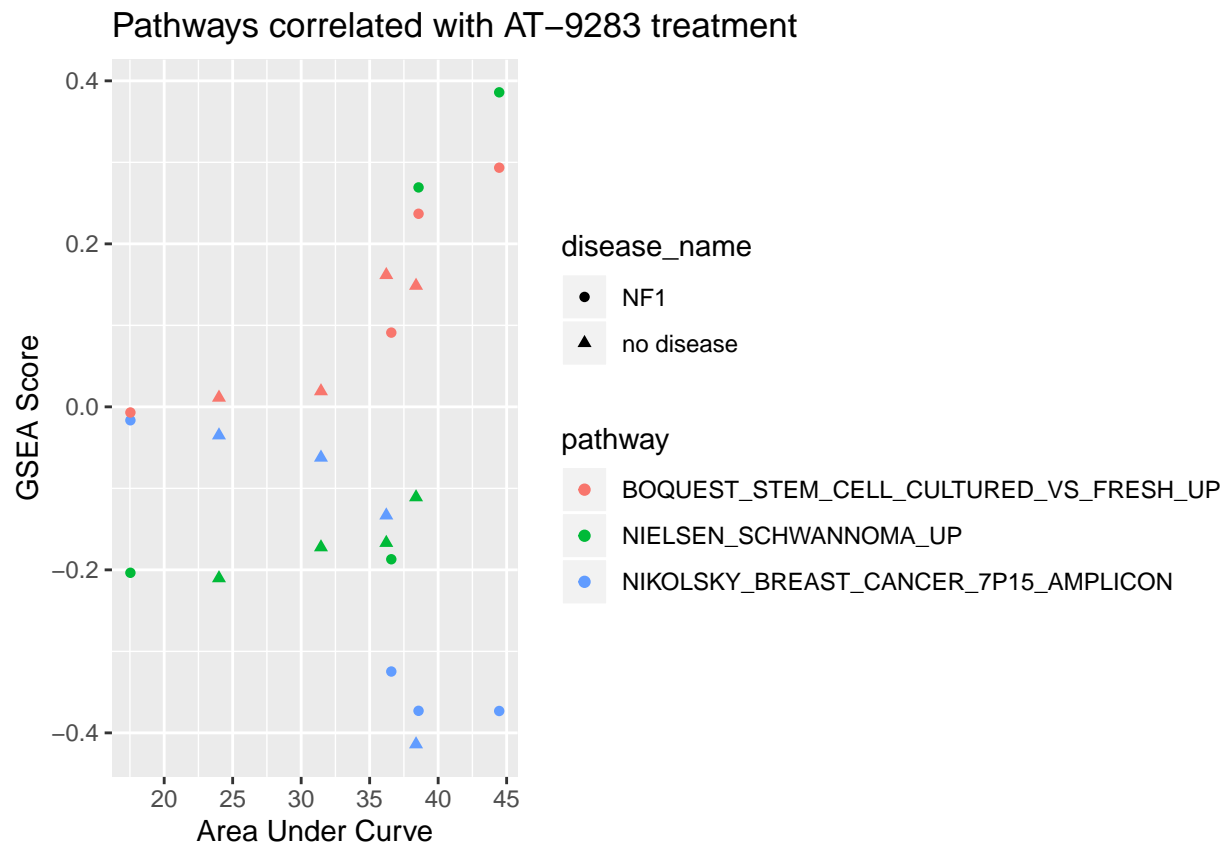


Pathways correlated with Parthenin treatment

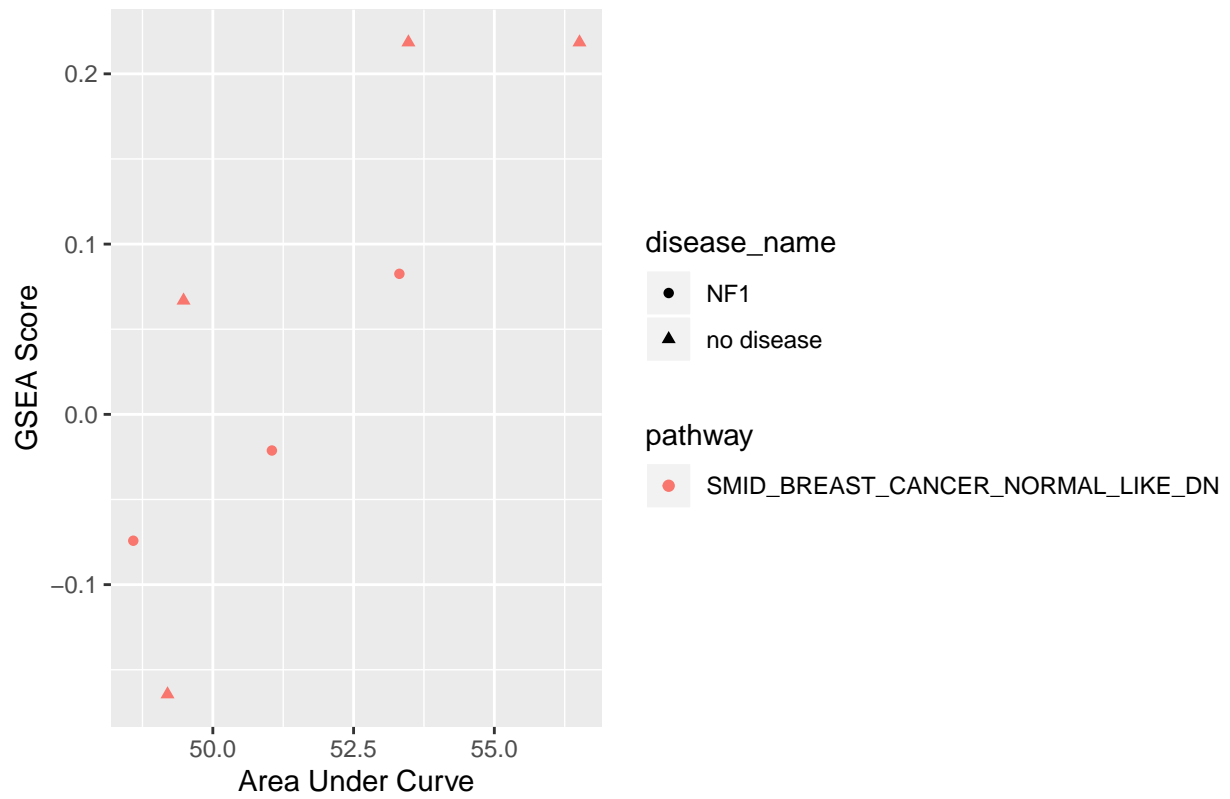




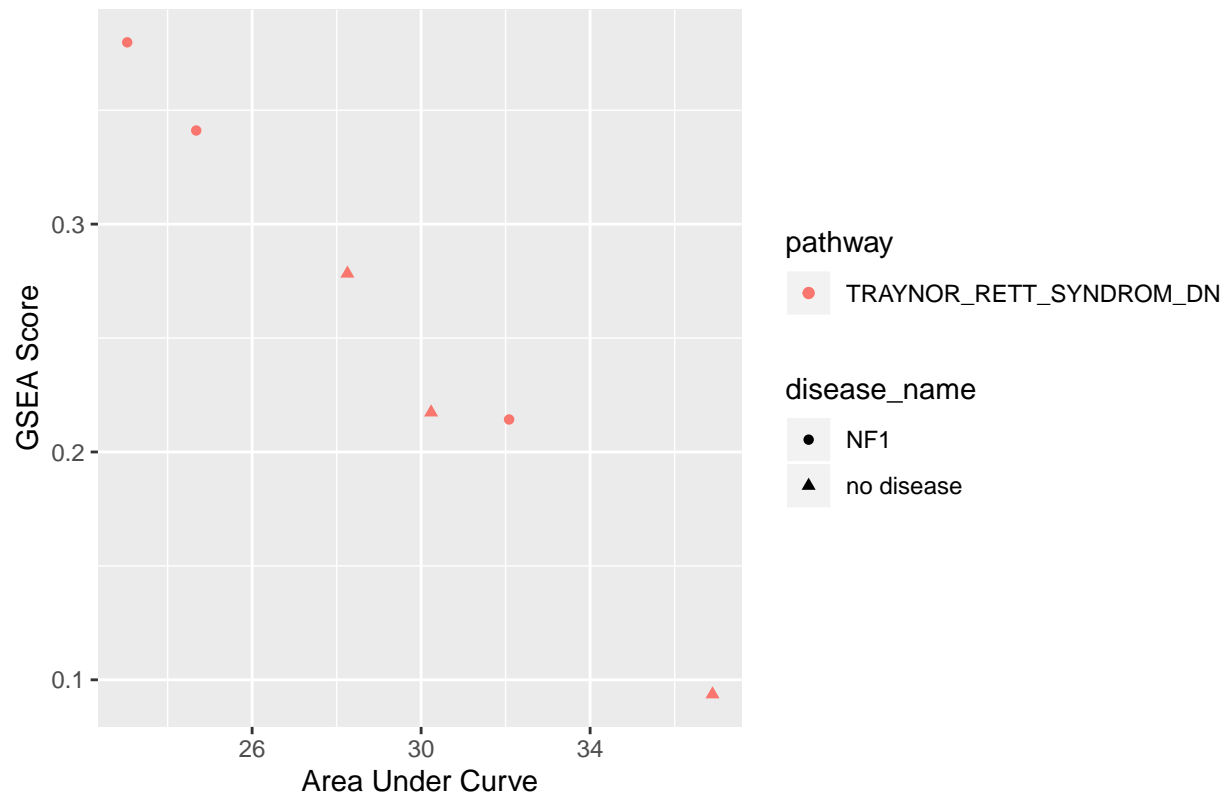




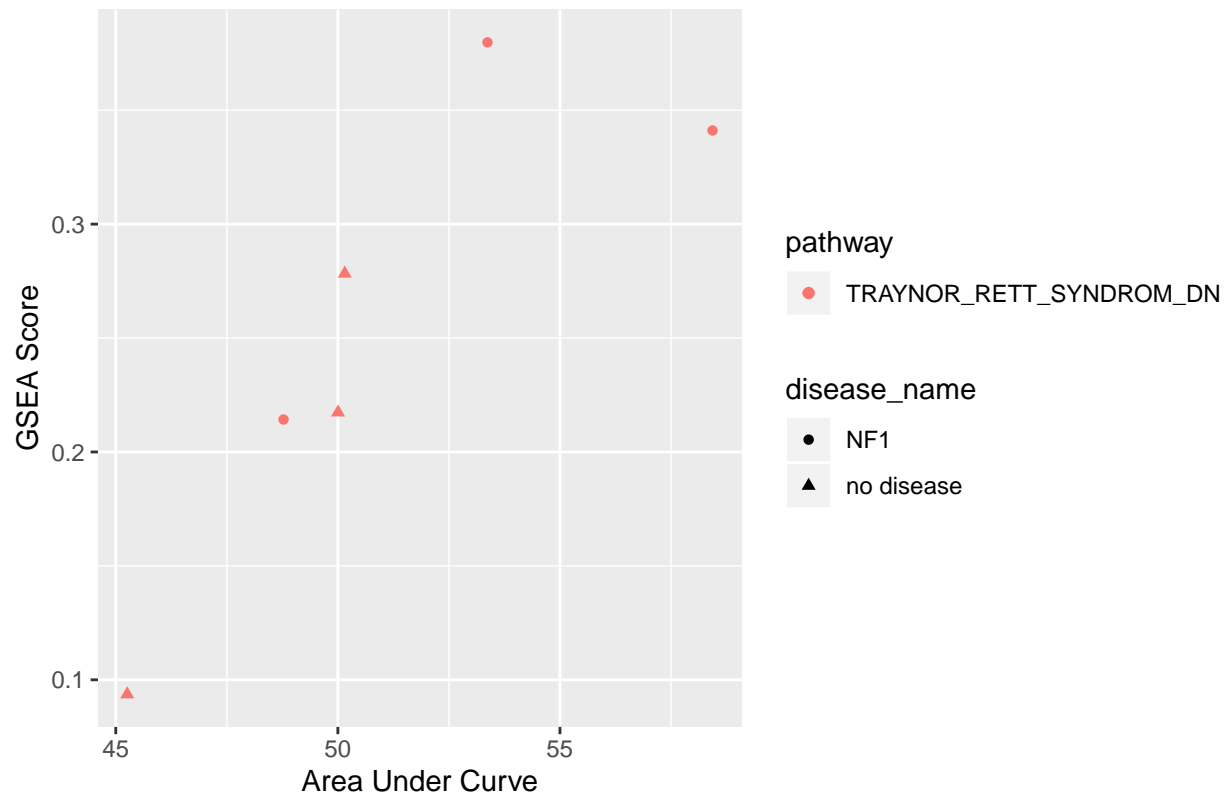
Pathways correlated with ERISMODEGIB treatment



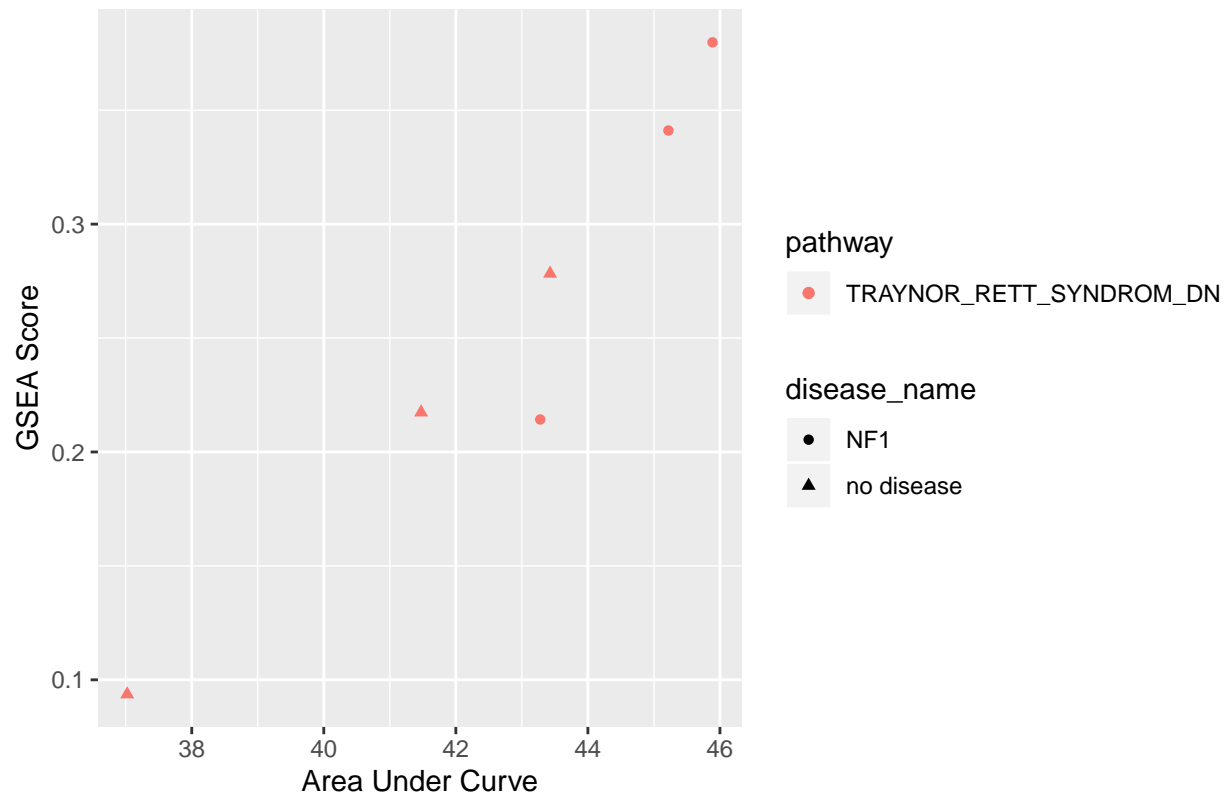
Pathways correlated with HMSL10016 treatment



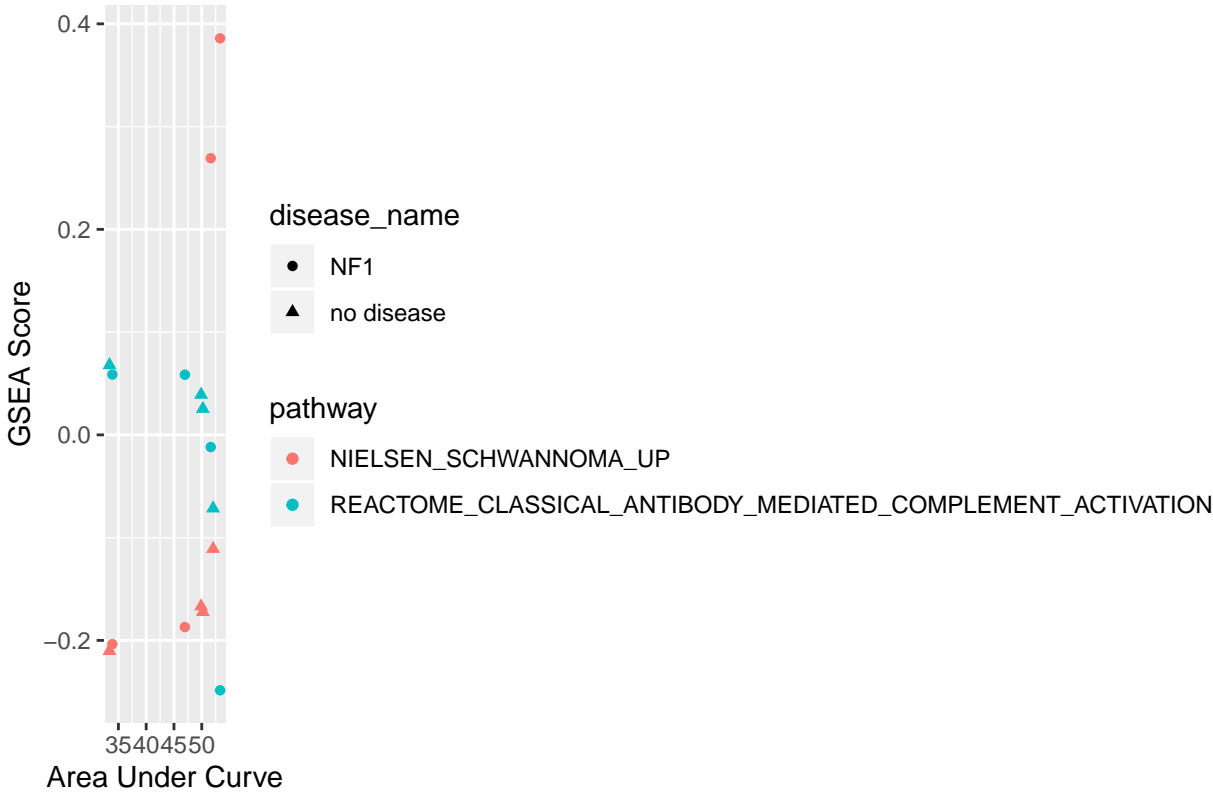
Pathways correlated with NA treatment

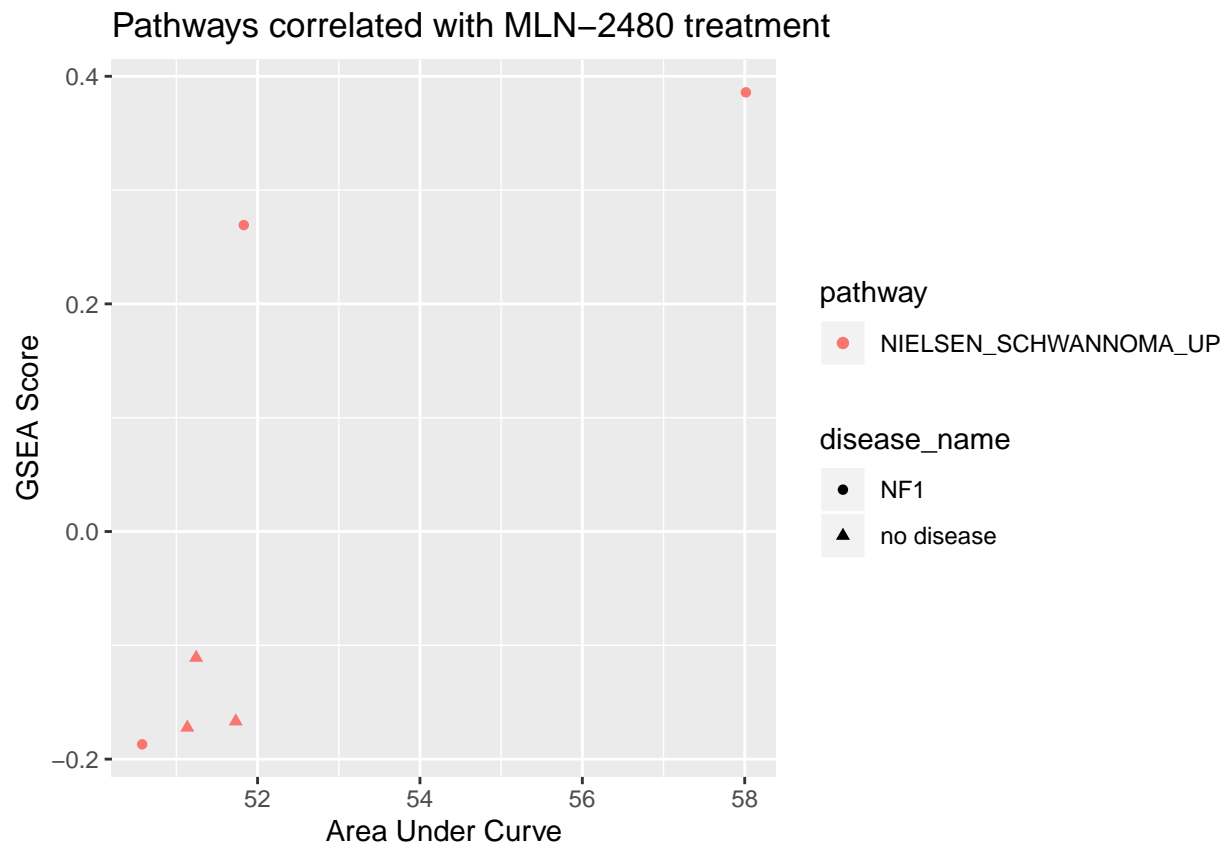


Pathways correlated with NA treatment



## Pathways correlated with PF-03716556 treatment







## Pathways correlated with Levosulpiride treatment



Because the minimum viability boxplots up top are a little more believable, i will plot those as well.

```
#drugid='277870'
for(drugid in unique(filtered.corvals$DT_explorer_internal_id)){

  filtered.corvals$drugname=drugnames=dmap$name[match(filtered.corvals$DT_explorer_internal_id,dmap$DT_explorer_internal_id)]

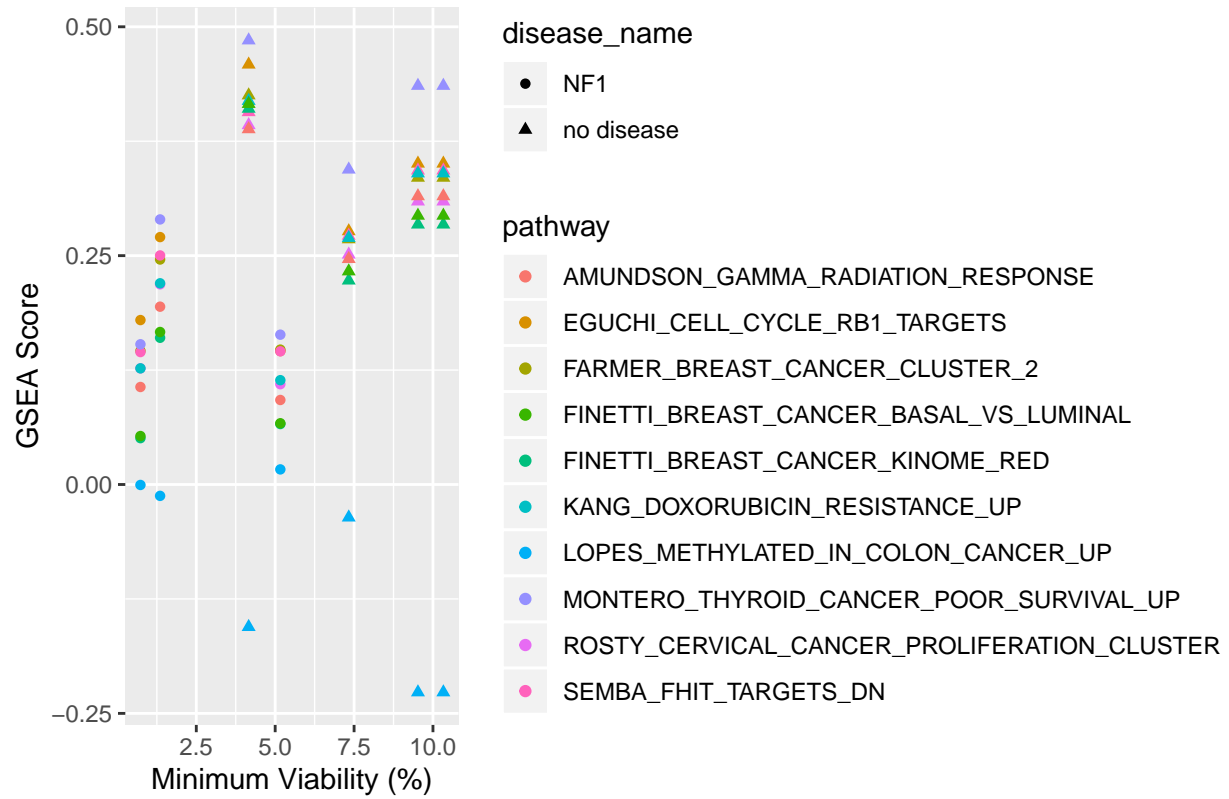
  drugname=dmap$name[match(drugid,dmap$DT_explorer_internal_id)]

  pathways<- dplyr::select(subset(filtered.corvals,DT_explorer_internal_id==drugid),pathway,corVal)%>%
    arrange(desc(abs(corVal)))%>%
    dplyr::select(pathway)%>%
    unlist()
  if(length(pathways)>10)
    pathways=pathways[1:10]

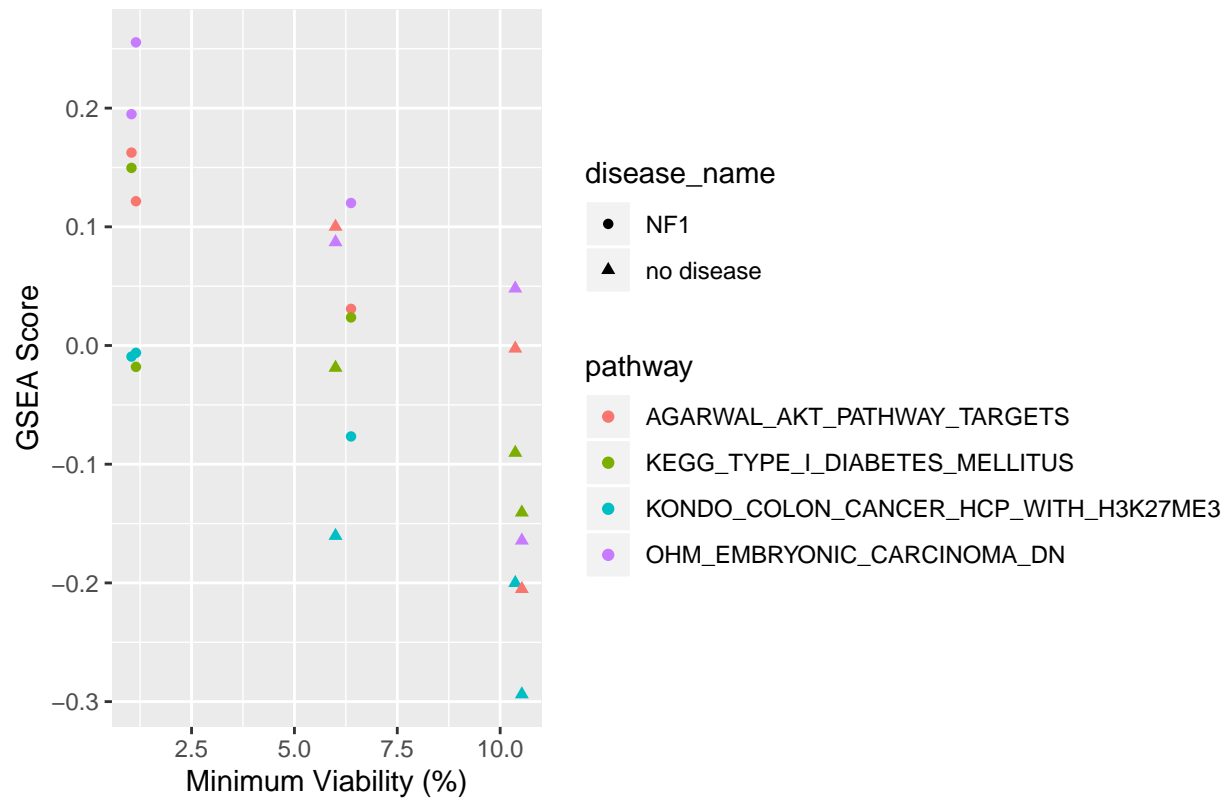
  plot.tab<-dres%>%subset(DT_explorer_internal_id==drugid)%>%
    subset(pathway%in%pathways)%>%
    subset(response_type=='Min_viability')%>%
    mutate(`Minimum Viability (%)`=as.numeric(response))%>%
    rename(GSVAscore='GSEA Score')

  p<-ggplot(plot.tab)+geom_point(aes(x=`Minimum Viability (%)`,y=`GSEA Score`,col=pathway,shape=disease_name))
  print(p)
}
```

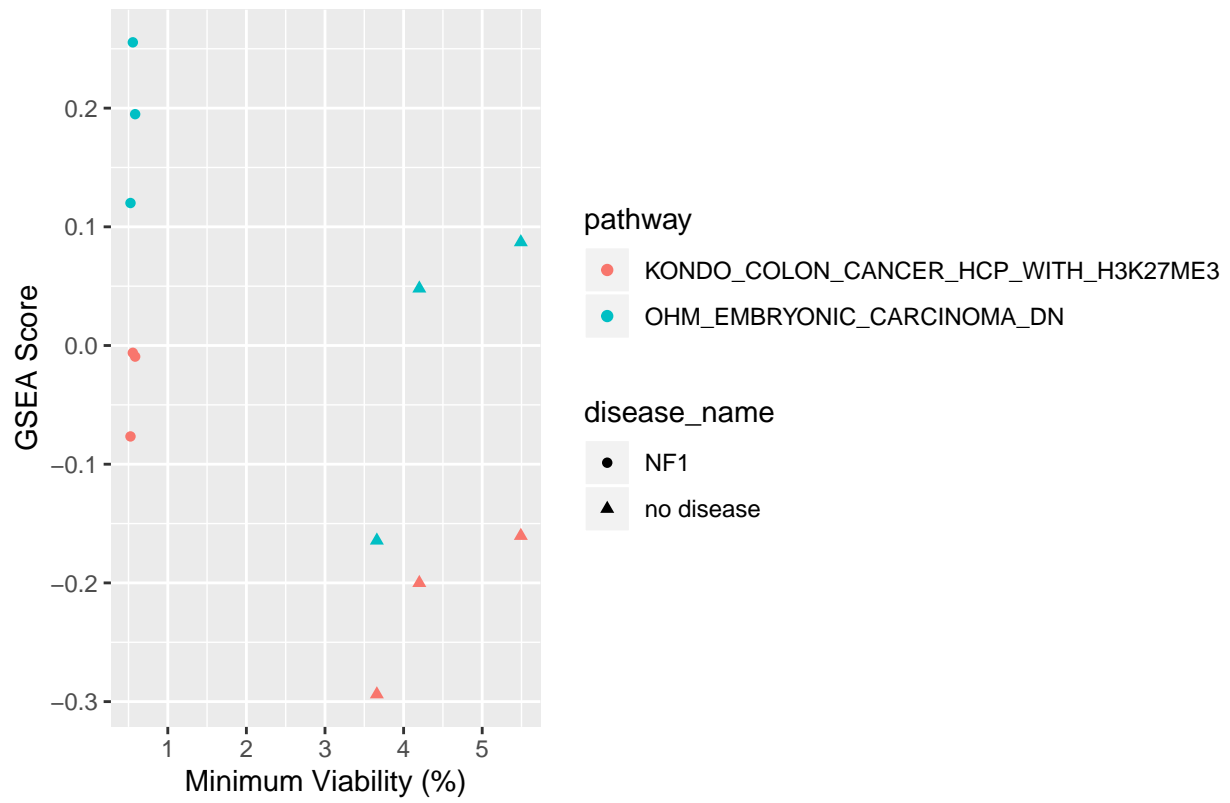
## Pathways correlated with IBRUTINIB treatment

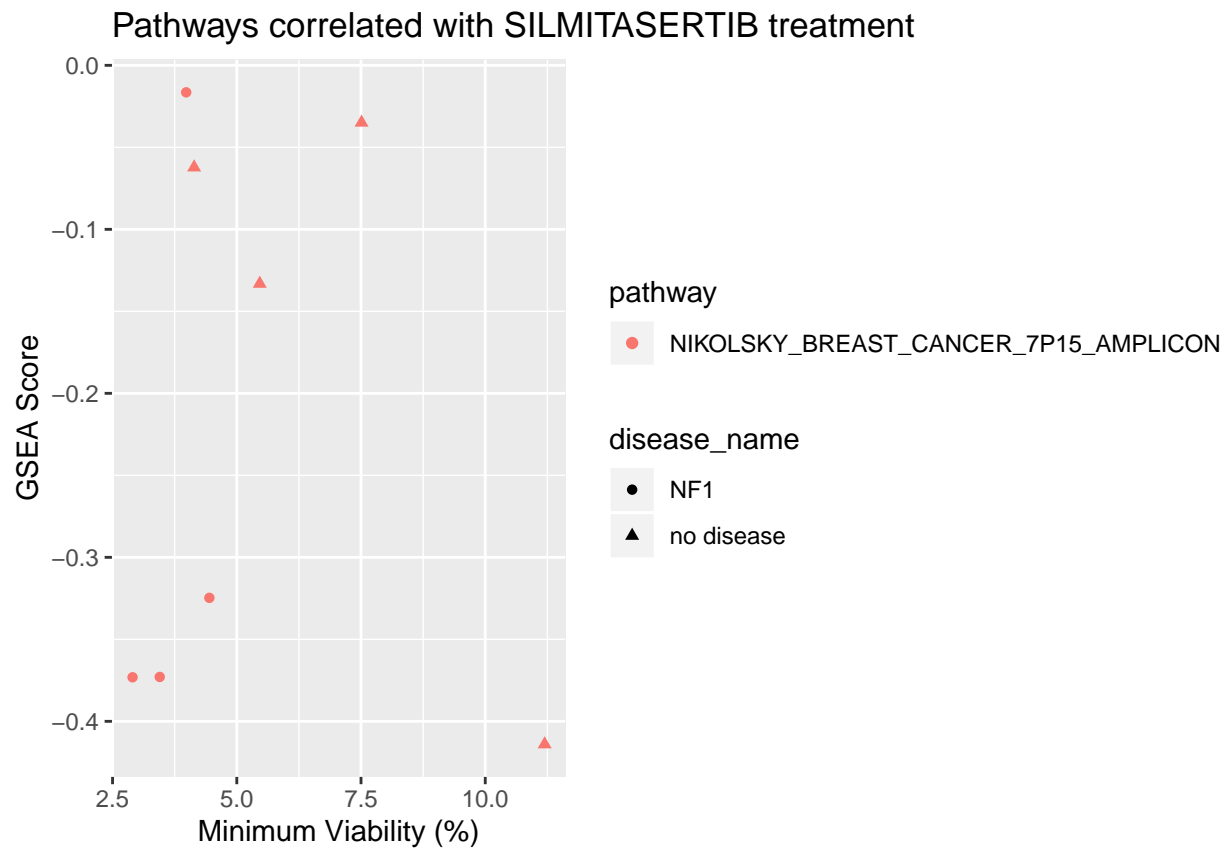


## Pathways correlated with FLURBIPROFEN treatment

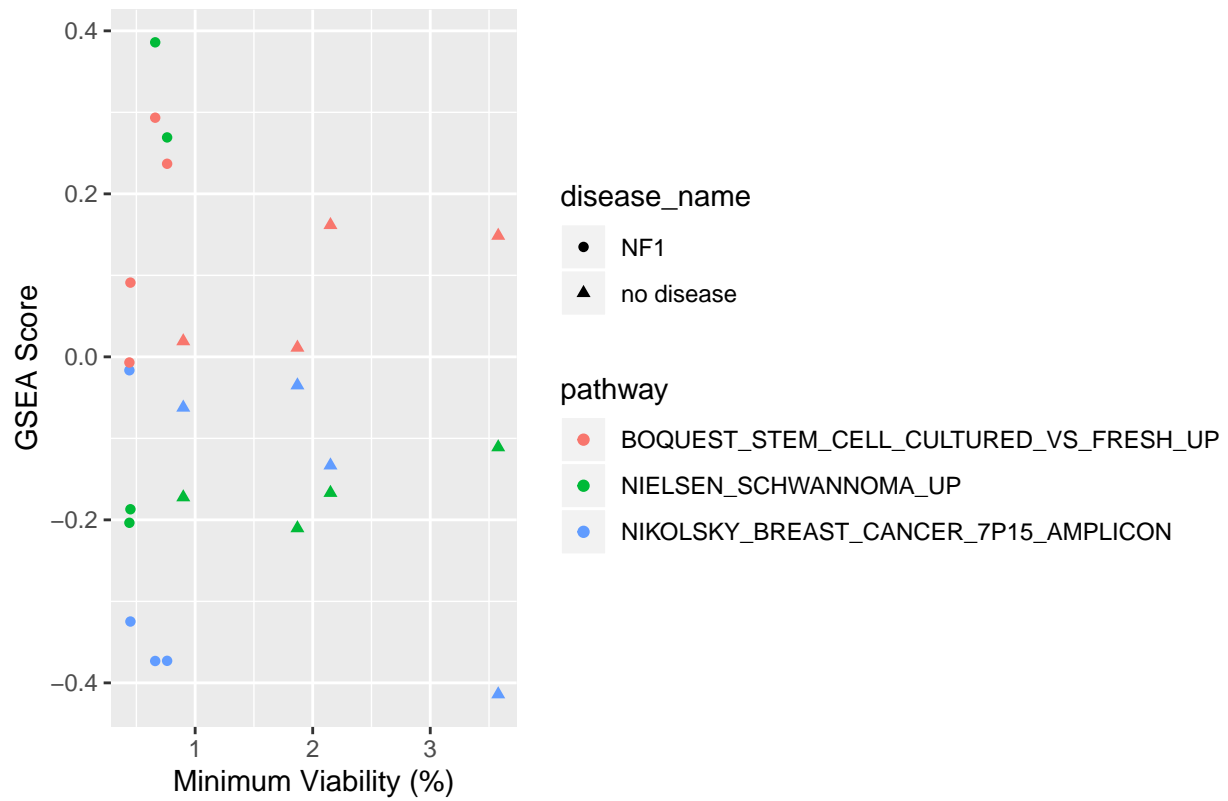


Pathways correlated with Parthenin treatment

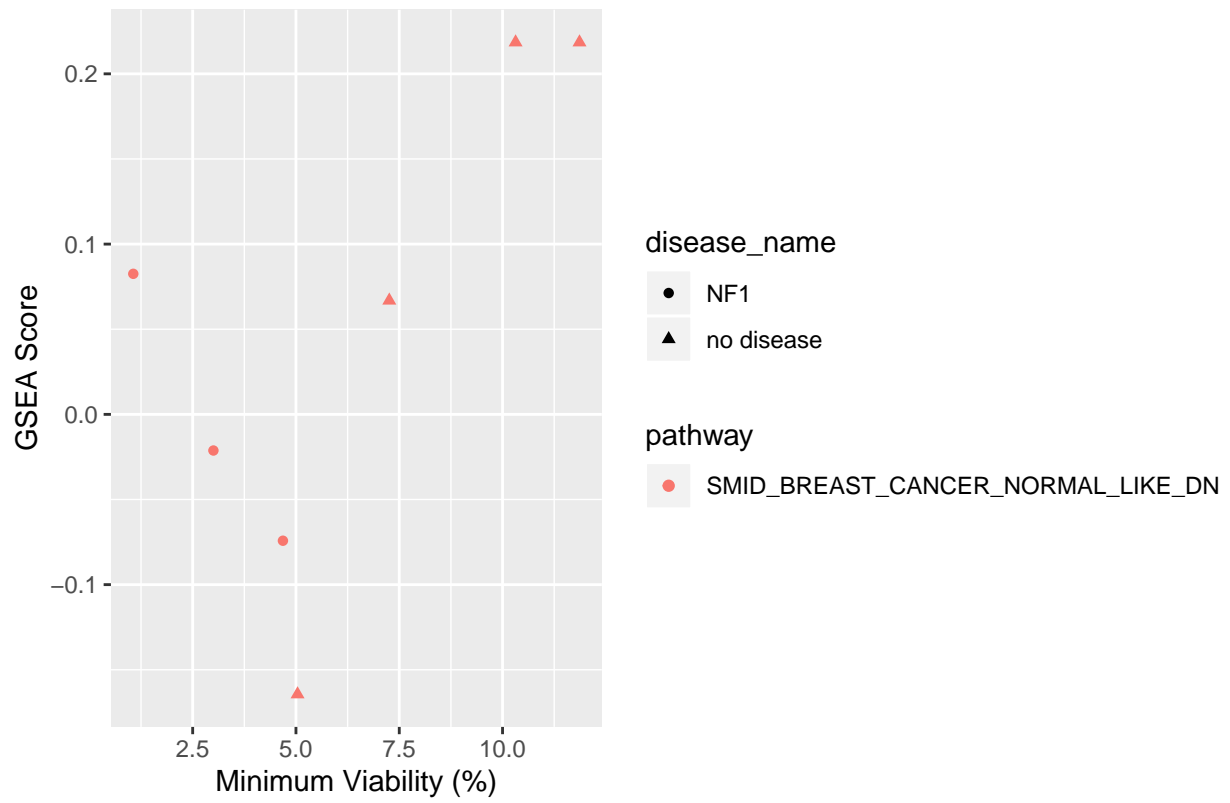




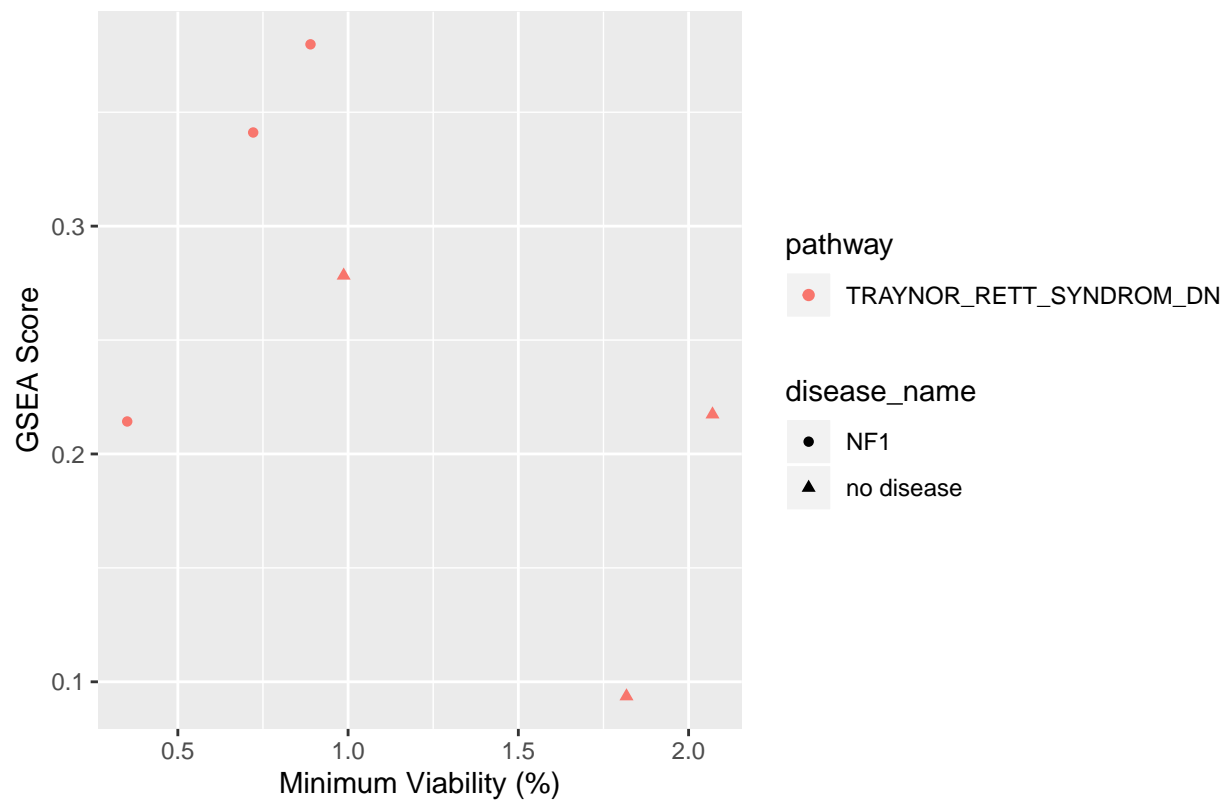
Pathways correlated with AT-9283 treatment



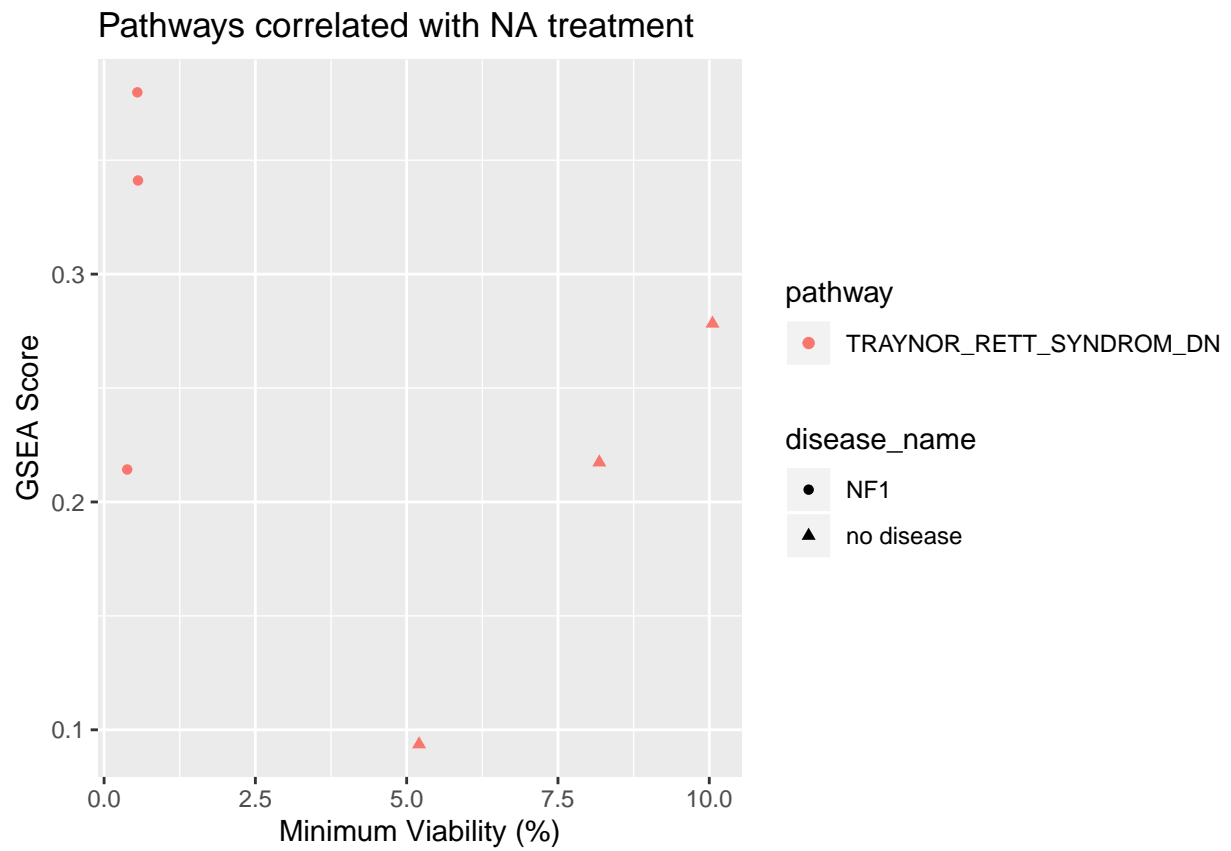
Pathways correlated with ERISMODEGIB treatment



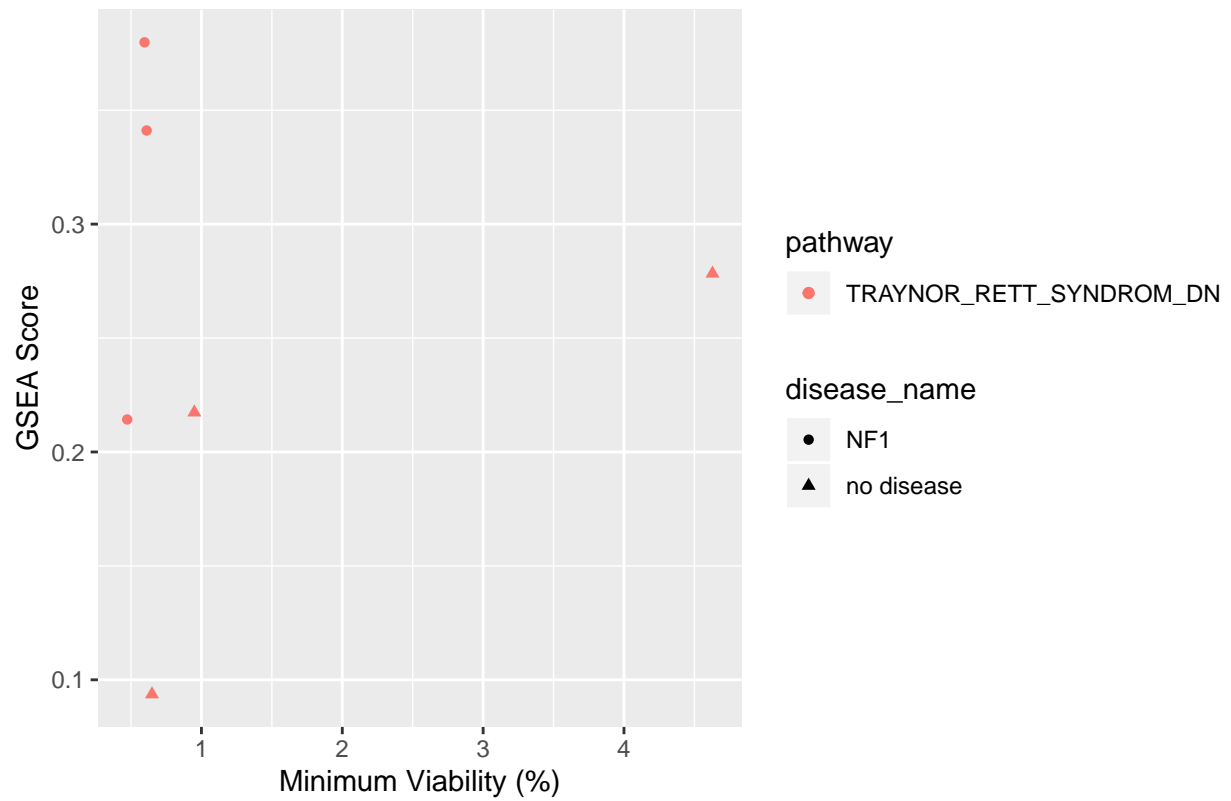
Pathways correlated with HMSL10016 treatment



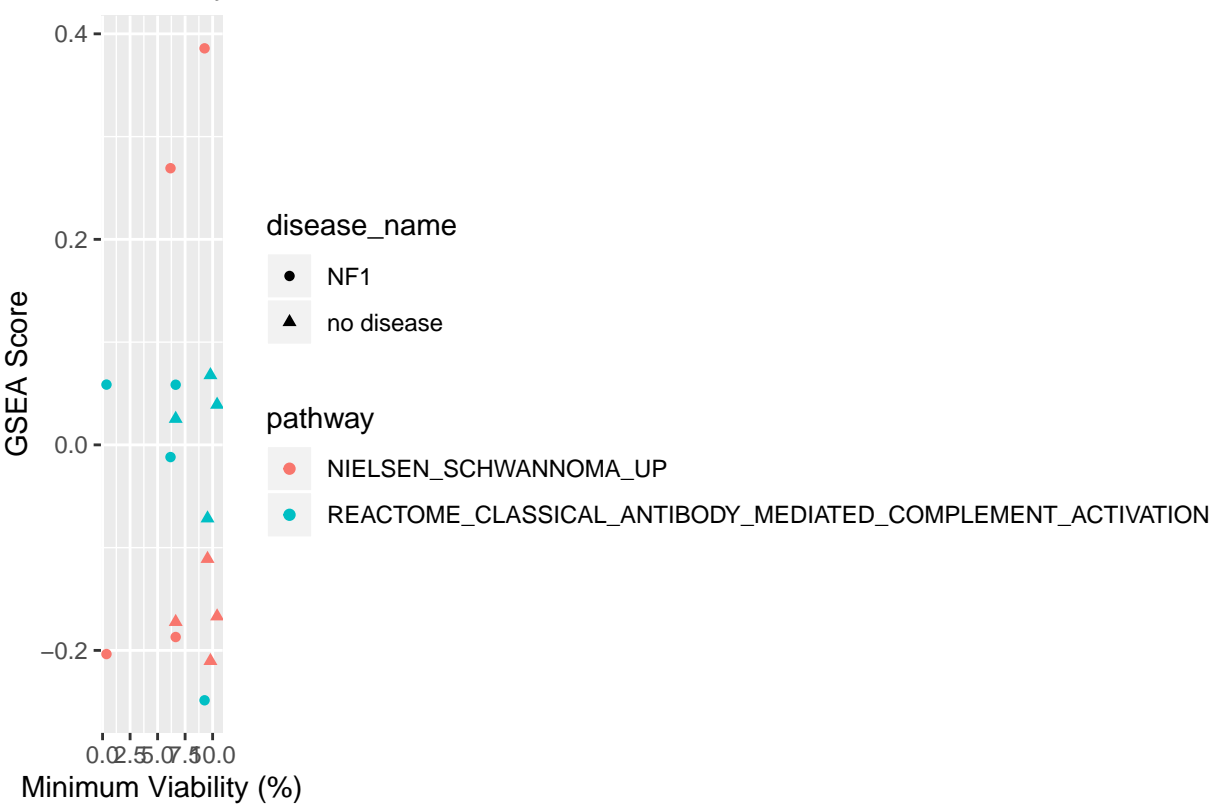


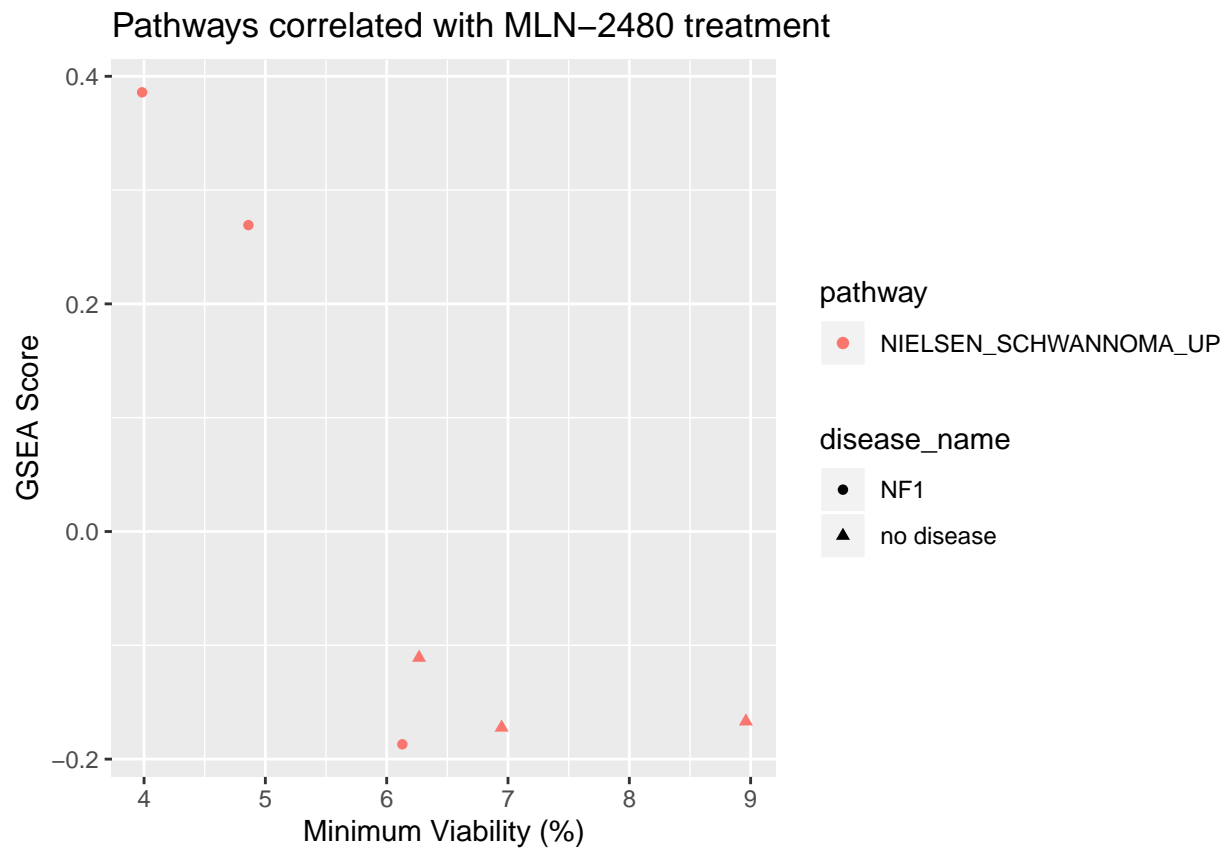


Pathways correlated with NA treatment



Pathways correlated with PF-03716556 treatment





## Pathways correlated with Levosulpiride treatment

