Random Forest Model

Code

Using *varSelRF* package for variable selection using random forest

https://cran.r-project.org/web/packages/varSelRF/varSelRF.pdf

I normalized the pre-filtered gene counts of the 92 AIH and healthy patients (no liver transplants, no outliers and no complex cases) using cpm() function of EdgeR, log = TRUE.

I defined the two groups as AIH and healthy.

After this, I ran the random forest model with differing standard deviations, starting at 2 and subtracting 0.25 in every loop. As the standard deviation became less stringent (lower), the number of genes increased. I recorded the genes in a dataframe.

Hide

```
# load in genecounts and metadata
genecounts_rf <- genecountspc_filt
metadata_rf <- metadata_filt

# normalize gene counts
genecounts_filt_cpm <- cpm(genecountspc_filt, log = TRUE)

# set up the column of interest, remove empty levels
metadata_rf$factors <- droplevels.factor(metadata_filt$case_hl_du)

#declare the factors for random forest model (AIH vs healthy)
RF_factors <- metadata_rf$factors

#initialize data frame, standard deviation, number of genes
RF_results <-data.frame(stddev = integer(), num.genes = integer(), genes = character())
stddev = 2
num.genes = 0

#run random forest until I have more than 50 genes in a group
while (num.genes < 50) {

#random forest command
varselRF.AIH <- varSelRF(t(genecounts_filt_cpm), as.factor(RF_factors), c.sd = stddev, mtryFactor = 1, ntree
= 5000, ntreeIterat = 2000, vars.drop.num = NULL, vars.drop.frac = 0.2, whole.range = FALSE, recompute.var.i
mp = FALSE, verbose = TRUE, returnFirstForest = TRUE, fitted.rf = NULL, keep.forest = TRUE)

#store data into data frame
genes = genemap[match(varselRF.AIH$selected.vars, genemap$ensembl_gene_id),]$hgnc
num.genes = varselRF.AIH$best.model.nvars
genes_t <- data.frame(t(c(stddev, num.genes, paste(unlist(genes), collapse=', '))))
colnames(genes_t) <- c("stddev","num.genes","genes")
RF_results<-rbind(RF_results,genes_t)

#reduce standard deviation by 0.25
stddev = stddev - 0.25

}
```

Hide

```
knitr::kable(RF_results)
```

stddev	num.genes	genes
2	2	PTOV1, GIGYF1
1.75	2	PTOV1, GIGYF1
1.5	2	FLYWCH1, GIGYF1
1.25	5	FLYWCH1, ARHGAP4, PTOV1, SLC4A10, GIGYF1
1	3	COL5A3, PTOV1, GIGYF1

stddev	num.genes	genes
0.75	5	FLYWCH1, COL5A3, PTOV1, SLC4A10, GIGYF1
0.5	99	TSPOAP1, MATK, NISCH, FLYWCH1, ZNF275, SUGP2, PDZD4, RASGRP2, MGAT4A, COL5A3, ARHGAP4, KLHL22, LMF2, ARFGAP1, PIGU, MIB1, CENPT, KLHDC4, CCDC130, PTOV1, TNPO2, ABCA2, EIF3A, EZH1, UST, ZAP70, STAT1, GBP1, CEP89, MTERF4, AGO3, IRF3, MPRIP, RRAS2, PRPF38B, STK26, TUT4, GOLGA1, SYTL2, USP8, MAPK8IP3, RHOT2, SGSM2, TMC6, IFITM3, PGGHG, ADAMTS10, SYTL1, SLC4A10, DDX46, FAM193B, GIGYF1, ADAM12, DGKZ, PLCH2, KLRF1, FCGR1A, DLG5, ZNF256, SCOC, TTC39B, TYSND1, DIP2A, CCDC78, MEGF6, TYW3, SNED1, INTS1, NEMF, TMEM170A, ENGASE, LENG8, TMC8, ABHD15, JMJD7-PLA2G4B, CEL, ZDHHC14, TBC1D10C, ANAPC2, DTX3, MTA1, ANO9, ZBTB37, MYBL1, NPIPB4, CCDC84, ZNF600, ANAPC7, SZT2, TPM2, DIO1, SCART1, CCNL2, AP5Z1, HAUS5, , C19orf84, MYO15B, DGKK