Finding trans eQTL in the Cheung data

VJ Carey

May 1, 2018

This code is based on a SNPlocs paradigm that has changed. eval=FALSE has been set for all chunks. Revisions will follow when time permits; contact stvjc at channing.harvard.edu if accelerated changes are desired.

```
> options(error=recover)
> #library(chfive40)
> library(cheung2010)
> library(hgfocus.db)
> library(GenomicRanges)
> allst = as.list(hgfocusCHRLOC)
> allen = as.list(hgfocusCHRLOCEND)
> pchrs = sapply(allst, function(x)names(x)[1])
> bad = which(sapply(pchrs,length)==0)
> if (length(bad)>0) {
+ pchrs = pchrs[-bad]
+ pchrs = sapply(pchrs, function(x)x)
+ pchrs = paste("chr", pchrs, sep="")
+ allst = allst[-bad]
+ allen = allen[-bad]
+ }
> st = sapply(allst, function(x) abs(x)[1])
> en = sapply(allen, function(x) abs(x)[1])
> gra = GRanges(seqnames=pchrs, IRanges(st,en))
> names(gra) = names(allst)
> gra = split(gra, seqnames(gra))
> lkg = getSS("cheung2010", "chr5")
> fn5 = featureNames(lkg)
> gra = lapply(gra, function(x)x[ intersect(names(x), fn5) ])
> getSNPlocs = function(chr, as.GRanges=TRUE) {
+    require("SNPlocs.Hsapiens.dbSNP144.GRCh37", character.only=TRUE)
+    sl = SNPlocs.Hsapiens.dbSNP144.GRCh37
+    chr = gsub("ch", "", chr)
```

```
+    ss = snpsBySeqname(sl, chr)
+    if (as.GRanges) return(as(ss, "GRanges"))
+    as.data.frame(ss)
+ }
> #library(snplocsDefault(), character.only=TRUE)
> if (!exists("c1s") && file.exists("c1s.rda")) load("c1s.rda")
> if (!exists("c1s")) c1s = getSNPlocs("ch1", as.GRanges=TRUE)
> if ("ch1" %in% seqlevels(c1s)) seqlevels(c1s) = gsub("ch", "chr", seqlevels(c1s))
> if (!file.exists("c1s.rda")) save(c1s,file="c1s.rda")
> if (!exists("c2s") && file.exists("c2s.rda")) load("c2s.rda")
> if (!exists("c2s")) c2s = getSNPlocs("ch2", as.GRanges=TRUE)
> if ("ch2" %in% seqlevels(c2s)) seqlevels(c2s) = gsub("ch", "chr", seqlevels(c2s))
> if (!file.exists("c2s.rda")) save(c2s,file="c2s.rda")
> if (!exists("c3s") && file.exists("c3s.rda")) load("c3s.rda")
> if (!exists("c3s")) c3s = getSNPlocs("ch3", as.GRanges=TRUE)
> if ("ch3" %in% seqlevels(c3s)) seqlevels(c3s) = gsub("ch", "chr", seqlevels(c3s))
> if (!file.exists("c3s.rda")) save(c3s,file="c3s.rda")
> if (!exists("c4s") && file.exists("c4s.rda")) load("c4s.rda")
> if (!exists("c4s")) c4s = getSNPlocs("ch4", as.GRanges=TRUE)
> if ("ch4" %in% seqlevels(c4s)) seqlevels(c4s) = gsub("ch", "chr", seqlevels(c4s))
> if (!file.exists("c4s.rda")) save(c4s,file="c4s.rda")
> if (!exists("c17s") && !exists("c17s") && file.exists("c17s.rda")) load("c17s.rda")
> if (!exists("c17s")) c17s = getSNPlocs("ch17", as.GRanges=TRUE)
> if ("ch17" %in% seqlevels(c17s)) seqlevels(c17s) = gsub("ch", "chr", seqlevels(c17s
> if (!file.exists("c17s.rda")) save(c17s,file="c17s.rda")
> if (!exists("c19s") && !exists("c19s") && file.exists("c19s.rda")) load("c19s.rda")
> if (!exists("c19s")) c19s = getSNPlocs("ch19", as.GRanges=TRUE)
> if ("ch19" %in% seqlevels(c19s)) seqlevels(c19s) = gsub("ch", "chr", seqlevels(c19s
> if (!file.exists("c19s.rda")) save(c19s,file="c19s.rda")
>
> #if ("multicore" %in% installed.packages()[,1]) library(multicore)
> #system("rm -rf tsco")
> #tr1c = transScores("cheung2010", "chr1", ~sex, snpRanges=c1s, geneRanges=gra[["chr
> #save(tr1c, file="tr1c.rda")
> #tr2c = transScores("cheung2010", "chr2", ~sex, snpRanges=c2s, geneRanges=gra[["chr
> #save(tr2c, file="tr2c.rda")
> #gc()
> #tr3c = transScores("cheung2010", "chr3", ~sex, snpRanges=c3s, geneRanges=gra[["chr
> #save(tr3c, file="tr3c.rda")
> #gc()
> #tr4c = transScores("cheung2010", "chr4", ~sex, snpRanges=c4s, geneRanges=gra[["chr
> #save(tr4c, file="tr4c.rda")
```

We use the probes on chr17 identified as harboring trans regulators by Cheung.

```
> tempfolder = function ()
+ {
+     z = tempfile()
+     system(paste("mkdir", z))
+     z
+ }
> obsfold = tempfolder()
> permfold = tempfolder()
> pr17 = structure(c("209165_at", "203654_s_at", "203367_at", "201508_at",
+ "218676_s_at", "208982_at", "202148_s_at", "214552_s_at", "214299_at",
+ "219282_s_at"), .Names = c("AATF", "COIL", "DUSP14", "IGFBP4",
+ "PCTP", "PECAM1", "PYCR1", "RABEP1", "TOP3A", "TRPV2"))
> options(verbose=TRUE)
> dropNAs = function (x)
+ {
+     if (!(is(x, "smlSet")))
+         stop("works only for smlSet instances")
+     sml <- x@smlEnv$smList
+     maf = snpStats::col.summary(sml[[1]])[, "MAF", drop = FALSE]
+     allrs = rownames(maf)
+     curok = which(!is.na(maf))
+     rm(maf)
+     if (length(curok) == 0)
+         stop("dropNAs eliminates all SNP on a chromosome, cannot proceed")
+     if (length(curok) != length(allrs))
+         x@smlEnv$smList[[1]] = x@smlEnv$smList[[1]][, curok]
+     rm(allrs)
+     x
+ }
> if (!exists("mgrsave")) {
+  mgrsave = list()
+  for (i in 1:22) {
+   cat(i)
+   cc17 = dropNAs(getSS("cheung2010", paste("chr", i, sep="")), probesToKeep=pr17))
+   mgrsave[[i]] = eqtlTests(cc17, ~sex, targdir=obsfold,
+      runname=paste("cc17", i, sep=""))
+   rm(cc17)
+   gc()
+   }
+  save(mgrsave, file="mgrsave.rda")
+ }
```

```
> set.seed(12345)
> if (!exists("mgrsave_perm")) {
+  mgrsave_perm = list()
+  for (i in 1:22) {
+   cat(i)
+   cc17 = dropNAs(getSS("cheung2010", paste("chr", i, sep="")),
+               probesToKeep=pr17, wrapperEndo=permEx))
+   mgrsave_perm[[i]] = eqtlTests(cc17, ~sex, targdir=permfold, runname=paste("pcc17"
+   rm(cc17)
+   gc()
+   }
+  save(mgrsave_perm, file="mgrsave_perm.rda")
+ }
```