rmarkdowntest.R

Mcdade

Wed Aug 31 09:09:29 2016

```
setwd("C:/Users/Mcdade/Desktop/EUFLOW")
RNASEQDATA<-read.csv(file="RNASEQDATA.csv",header=TRUE)
RPPADATA<-read.csv(file="RPPADATA.original.csv",header=TRUE)
EvaluationExperimentSet<-RNASEQDATA
#EvaluationExperimentSet<-rbind(RNASEQDATA,RANDOMSET)
ReferenceSet<-RPPADATA



require(IdMappingAnalysis)
```

```
## Loading required package: IdMappingAnalysis

## Warning: package 'IdMappingAnalysis' was built under R version 3.2.2

## Loading required package: R.oo

## Loading required package: R.methodsS3

## R.methodsS3 v1.7.0 (2015-02-19) successfully loaded. See ?R.methodsS3 for help.

## R.oo v1.19.0 (2015-02-27) successfully loaded. See ?R.oo for help.

##
## Attaching package: 'R.oo'

## The following objects are masked from 'package:methods':
##
##     getClasses, getMethods

## The following objects are masked from 'package:base':
##
##     attach, detach, gc, load, save

## Loading required package: rChoiceDialogs

## Loading required package: rJava

## Warning: package 'rJava' was built under R version 3.2.3

##
## Attaching package: 'rJava'
```

```
## The following object is masked from 'package:R.oo':
##
##     clone

## This is rChoiceDialogs 1.0.6 2014-09-05

## This is IdMappingAnalysis 1.14.0 2013-05-24
```

```
# #require(mvbutils)
options(stringsAsFactors = FALSE)


.assign.status<-function(x,status,data.type,version){
    if (status == "workflow_option")
        status_tag<-paste(row.names(x),"_WFO","_",as.character(data.type),"_",as.character(version),sep=
    if (status == "driver")
        status_tag<-paste(row.names(x),"_DRIVER",sep="")
    return(status_tag)
}

WorkflowEvaluationData<-function(EvaluationExperimentSet,ReferenceSet){

    names(EvaluationExperimentSet)[1]<- "Symbol"
    names(ReferenceSet)[1]<- "Symbol"
    row.names(EvaluationExperimentSet)<-EvaluationExperimentSet$Symbol
    row.names(ReferenceSet)<-ReferenceSet$Symbol
    WorkflowList<-strsplit(row.names(EvaluationExperimentSet),"_")
    WorkflowNameVector<-sapply(WorkflowList, "[", 1)
    WorkflowOptionVector<-sapply(WorkflowList, "[", 2)
    #WorkflowNumber<-substr(WorkflowOptionVector,nchar(WorkflowOptionVector),nchar(WorkflowOptionVector
    return(list(ReferenceSet,(split(EvaluationExperimentSet,WorkflowOptionVector))))

}



Workflow.Data<-WorkflowEvaluationData(EvaluationExperimentSet,ReferenceSet)







merge.tag.options<-function(Workflow.Data,ReferenceTag="P",EvaluationTag="RS"){
    EvaluationList<-Workflow.Data[[2]]
    Merged.options<-Workflow.Data[[1]]
    row.names(Merged.options)<-.assign.status(Merged.options,status="driver",ReferenceTag,1)
    for(o in c(1:length(EvaluationList))) {
        Evaluation_dataframe<-as.data.frame(EvaluationList[[o]])
        SymbolList<-strsplit(row.names(Evaluation_dataframe),"_")
        row.names(Evaluation_dataframe)<-sapply(SymbolList, "[", 1)
        row.names(Evaluation_dataframe)<-.assign.status(Evaluation_dataframe,status="workflow_option",E
        Merged.options=rbind(Merged.options,Evaluation_dataframe)
```

```
    }
    return(Merged.options)
}

Merged.options<-merge.tag.options(Workflow.Data)


#count.options<-function(x) {
#    paste(row.names(workflow_options_data[sapply(strsplit(row.names(workflow_options_data),"_"), "[", 
#}


make.workflow.map <- function(Merged.options){
    drivers<-row.names(Merged.options[sapply(strsplit(row.names(Merged.options),"_"), "[", 2) == "DRIVE
    workflow_options_data<-Merged.options[sapply(strsplit(row.names(Merged.options),"_"), "[", 2) == "W
    count.options<-function(x) {
        paste(row.names(workflow_options_data[sapply(strsplit(row.names(workflow_options_data),"_"), "[
    }
    #workflow_options<-row.names(workflow_options_data[sapply(strsplit(row.names(workflow_options_data)
    imax<-max(unique(sapply(strsplit(row.names(workflow_options_data),"_"), "[", 4)),na.rm = TRUE)
    #imin<-min(unique(sapply(strsplit(row.names(workflow_options_data),"_"), "[", 4)),na.rm = TRUE)
    #workflow_options_merged<-paste(count.options(workflow_options_data,1),count.options(workflow_optio
    workflow_options_matrix<-sapply(1:imax,count.options)
    #workflow_options_merged<-paste(row.names(workflow_options_data[sapply(strsplit(row.names(workflow_
    workflow_options_merged<-sapply(1:dim(workflow_options_matrix)[1],function(i){paste0(as.character(w
    WorkflowMap<-data.frame(drivers,workflow_options_merged)
    ###when your ready class(Merged.options)<- "WorkflowMap"
    return(WorkflowMap)
}

Model.quality.list<-function(Merged.options){
    WorkflowMap.object<-make.workflow.map(Merged.options)
    IdMap.example<-IdMap(DF=WorkflowMap.object,name="Workflowmap.object", primaryKey="drivers",secondary
    secondaryIDs<-unlist(strsplit(WorkflowMap.object$workflow_options_merged,","))
    uniquePairs_workflow <- as.UniquePairs.IdMap(IdMap.example,secondaryIDs)
    reference<-Merged.options[sapply(strsplit(row.names(Merged.options),"_"),"[",2) == "DRIVER",]
    names(reference)[names(reference)=="Symbol"] <- "drivers"
    reference$drivers<-row.names(reference)
    evaluation<-Merged.options[sapply(strsplit(row.names(Merged.options),"_"),"[",2) != "DRIVER",]
    names(evaluation)[names(evaluation)=="Symbol"] <- "workflow_options_merged"
    evaluation$workflow_options_merged<-row.names(evaluation)
    Model.quality.object<-CorrData(uniquePairs_workflow,reference,evaluation)
    return(Model.quality.object)
}

Model.quality.object<-Model.quality.list(Merged.options)


Workflow.Criterion<-function(Model.quality.object){
    Model.quality<- Corr(Model.quality.object,method="spearman",verbose=TRUE)
    return(Model.quality)
}
```

```
Model.Quality<-Workflow.Criterion(Model.quality.object)
```

```
## Performing correlations...
```

```
fit2clusters.workflow<-function(Y, Ysigsq,
                                bootModel,
                                piStart = c(0.5, 0.5),
                                VStart = c(0.1,0.1),
                                psiStart = c(0,0.1),
                                NinnerLoop = 1,
                                nReps=500,
                                psi0Constraint,
                                V0Constraint,
                                sameV=FALSE,
                                estimatesOnly=TRUE,
                                printMe = TRUE,
                                plotMe = TRUE,
                                testMe=FALSE,
                                Ntest = 5000,
                                seed) {
    ### EM algorithm for 2 clusters,
    ### with constraints on the cluster means and variances, and known data variances
    if(testMe) {
        if(missing(seed)) .Random.seed <<- Random.seed.save
        else if(!is.na(seed)) .Random.seed <<- seed
        #  NA ==>  a new dataset.
        simPsi = c(0, 0.4)  ##
        simPi = c(2/3, 1/3)
        simData = data.frame(G = 1+rbinom(Ntest, 1, simPi[2]))
        simV = c(0.05^2, 0.05^2)
        simData$Ysigsq = rgamma(Ntest, 10, 400)
        simData$sd = sqrt(simV[simData$G] +simData$Ysigsq)
        simData = within(simData, Y <- simPsi[G] + rnorm(Ntest)*sqrt(simV[G]) + rnorm(Ntest)*sqrt(Ysigs
        print(summary(simData$Y))
        Y = simData$Y
        Ysigsq = simData$Ysigsq
    }
    ###############  Begining of EM algorithm  ###############
    piStar = piStart
    VStar = VStart
    psiStar = psiStart
    stopMe = FALSE
    iRep = 0
    while(1) {
        iRep = iRep + 1
        #    catn(", ", missing(V0Constraint))
        if(!missing(V0Constraint))
            VStar[1] = V0Constraint
        if(!missing(psi0Constraint))
            psiStar[1] = psi0Constraint
        #    print(psiStar)
        piStarOdds = piStar[2]/piStar[1]
```

```
        piStarOddsGK = piStarOdds *
            dnorm(Y, psiStar[2], sqrt(VStar[2] + Ysigsq)) /
            dnorm(Y, psiStar[1], sqrt(VStar[1] + Ysigsq))
        piStarGK = cbind(1/(1+piStarOddsGK), piStarOddsGK/(1+piStarOddsGK))
        EstarN = apply(piStarGK, 2, sum)
        piStar = apply(piStarGK, 2, mean)
        psiHat = psiStar
        VHat = VStar
        for(iRepInner in 1:NinnerLoop) {
            varHatTotal = colSums(outer(Y, psiHat, "-")^2 * piStarGK)
            sigsqTotal  = Ysigsq %*% piStarGK
            VHat = pmax(0, varHatTotal - sigsqTotal) / EstarN
            if(!missing(V0Constraint))
                VHat[1] = V0Constraint
            psiHat = colSums( Y %*% (piStarGK
                                     / outer(Ysigsq, VHat, "+"))) /
                colSums( piStarGK
                         / outer(Ysigsq, VHat, "+"))
            if(!missing(psi0Constraint))
                psiHat[1] = psi0Constraint
            if(sameV)
                VHat[1] = VHat[2] = mean(VHat[1], VHat[2])
            if(max(abs(psiHat-psiStar), abs(VHat-VStar)) < 1e-7)
                stopMe = TRUE;
            psiHat -> psiStar
            VHat -> VStar
        }
        if(iRep >= nReps) stopMe = TRUE
        if(stopMe) break
    }
    cat(iRep, ifelse(iRep==nReps, ".  Loop exhausted.", ".  Converged."), "\n")

    if(plotMe) {
        options(echo=F)
        plot(col="blue", type = "l",main ="Mixture density", Ytemp<-seq(-1,1,length=100),
             xlab= as.character(names(bootModel[3])), cex.lab = 1.5, ylab="Density", sub = as.character
             piStar[1]*dnorm(Ytemp, psiStar[1], sqrt(VStar[1] + mean(Ysigsq)))
             +
                 piStar[2]*dnorm(Ytemp, psiStar[2], sqrt(VStar[2]  + mean(Ysigsq)))
        )
        for(g in 1:2) lines(col="blue", lty=2, Ytemp<-seq(-1,1,length=100),
                            piStar[g]*dnorm(Ytemp, psiStar[g], sqrt(VStar[g]  + mean(Ysigsq))))
        lines(density(Y), lwd=2, col="black")
        abline(v = 0)
        abline(v = 0.38352)
        ###  Should we make a better choice than the means of the Ysigsq?
        if(testMe) lines(col="red", Ytemp<-seq(-1,1,length=100),
                         piStar[1]*dnorm(Ytemp, simPsi[1], sqrt(simV[1] + mean(simData$Ysigsq)))
                         +
                             piStar[2]*dnorm(Ytemp, simPsi[2], sqrt(simV[2]  + mean(simData$Ysigsq)))
        )
        legend(x=par("usr")[1], y=par("usr")[4],
               legend=c(ifelse(testMe, "truth", ""),
```

```
                        "data smooth", "estimate", " x or 0 component", " + component"),
               col=c("red", "black", "blue", "blue", "blue"),
               lty=c(ifelse(testMe, 1,0),1,1,2,2),
               lwd=c(1,2,1,1,1)
        )
        options(echo=T)
    }
    estimates = c(pi1=piStar[2], psi0=psiHat[1],
                  psi1=psiHat[2], Var0=VStar[1], Var1=VStar[2])

    posteriorOdds =
        piStar[2]*dnorm(Y, psiHat[2], sqrt(VStar[2] + Ysigsq)) /
        piStar[1]/dnorm(Y, psiHat[1], sqrt(VStar[1] + Ysigsq))
    postProb = posteriorOdds/(1+posteriorOdds)
    postProbVar = Ysigsq * (postProb*(1-postProb))^2 *
        ((Y-psiHat[1])/(VStar[1]+Ysigsq) - (Y-psiHat[2])/(VStar[2]+Ysigsq))^2

    if(testMe) {
        simTruth = c(pi1=simPi[2], psi0=simPsi[1],
                     psi1=simPsi[2], Var0=simV[1], Var1=simV[2])
        estimates = data.frame(row.names=c("true","estimated"),
                               rbind(simTruth, estimates))
    }
    if(estimatesOnly) return(estimates)
    else {
        attr(x=posteriorOdds, which="estimates") = estimates
        return(data.frame(posteriorOdds,postProbVar))
    }


}


Workflow.posteriorestimate<-function(Model.quality.object,Model.Quality,postProb=NULL,postProbVar=NULL)
    bootstrap<-Bootstrap(Model.quality.object,Fisher=TRUE,verbose=TRUE)
    bootModel<-as.data.frame(bootstrap)
    bootModel<-bootModel[complete.cases(bootModel),]
    pairs<-bootModel[,1:2]
    #bootModel<-bootModel[c(1:96,99:134),]
    EMtest<-fit2clusters.workflow(bootModel$corr, bootModel$sd^2,bootModel,psi0Constraint=0, sameV=T,es
    #again part of the EMtest
    postProbs<-as.vector(EMtest[[1]]/(1+EMtest[[1]]))  #not needed at the output is a datafrmae from fi
    postProbVar <-as.vector(EMtest[[2]])
    bootMergedWithPairs = merge(data.frame(postProbs=postProbs, postProbVar=postProbVar,bootModel), pai
    return(bootMergedWithPairs)
}


Posterior.dataframe<-Workflow.posteriorestimate(Model.quality.object,Model.Quality)
```

```
## Performing bootstrap R= 200 on correlations...
##
processed: 1 %
```

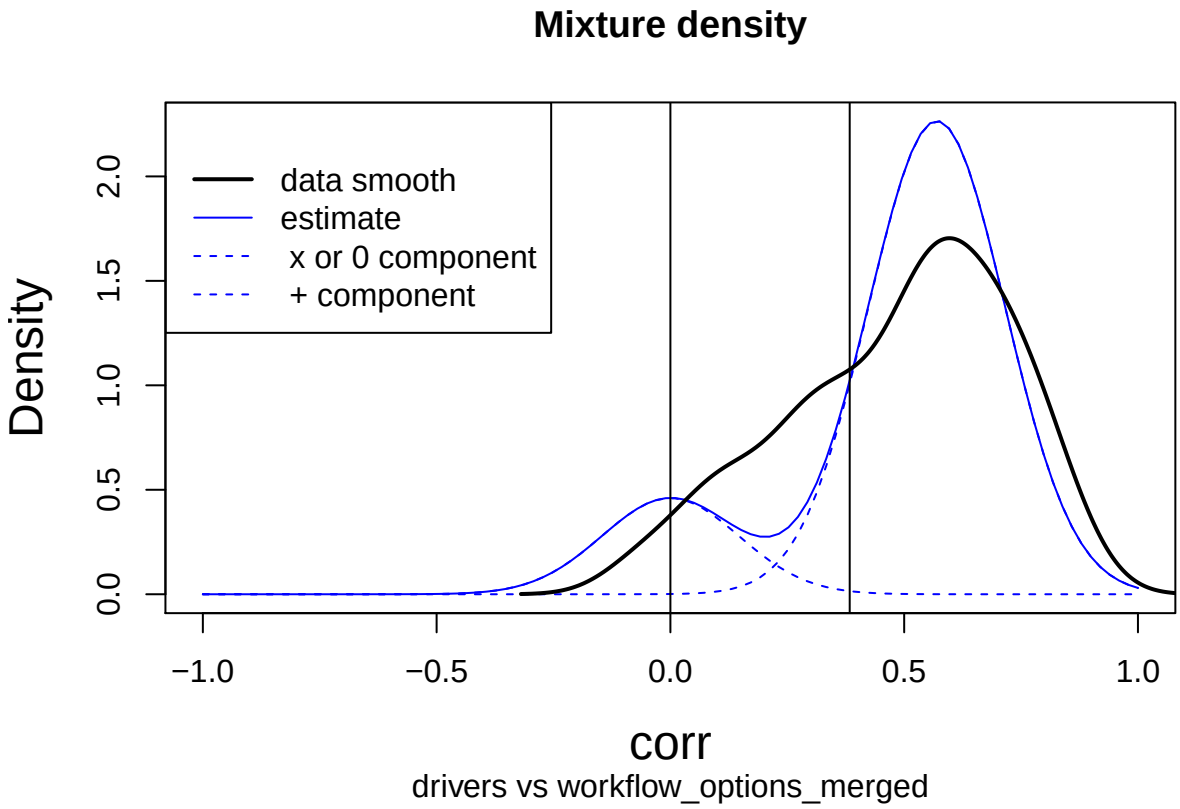
```
processed: 1 %

processed: 2 %

processed: 3 %

processed: 4 %

processed: 4 %

processed: 5 %

processed: 6 %

processed: 7 %

processed: 7 %

processed: 8 %

processed: 9 %

processed: 10 %

processed: 10 %

processed: 11 %

processed: 12 %

processed: 13 %

processed: 13 %

processed: 14 %

processed: 15 %

processed: 16 %

processed: 16 %

processed: 17 %

processed: 18 %

processed: 19 %

processed: 19 %

processed: 20 %

processed: 21 %
```

```
processed: 22 %
processed: 22 %
processed: 23 %
processed: 24 %
processed: 25 %
processed: 25 %
processed: 26 %
processed: 27 %
processed: 28 %
processed: 28 %
processed: 29 %
processed: 30 %
processed: 31 %
processed: 31 %
processed: 32 %
processed: 33 %
processed: 34 %
processed: 34 %
processed: 35 %
processed: 36 %
processed: 37 %
processed: 37 %
processed: 38 %
processed: 39 %
processed: 40 %
processed: 40 %
processed: 41 %
```

```
processed: 42 %
processed: 43 %
processed: 43 %
processed: 44 %
processed: 45 %
processed: 46 %
processed: 46 %
processed: 47 %
processed: 48 %
processed: 49 %
processed: 49 %
processed: 50 %
processed: 51 %
processed: 51 %
processed: 52 %
processed: 53 %
processed: 54 %
processed: 54 %
processed: 55 %
processed: 56 %
processed: 57 %
processed: 57 %
processed: 58 %
processed: 59 %
processed: 60 %
processed: 60 %
processed: 61 %
```

```
processed: 62 %
processed: 63 %
processed: 63 %
processed: 64 %
processed: 65 %
processed: 66 %
processed: 66 %
processed: 67 %
processed: 68 %
processed: 69 %
processed: 69 %
processed: 70 %
processed: 71 %
processed: 72 %
processed: 72 %
processed: 73 %
processed: 74 %
processed: 75 %
processed: 75 %
processed: 76 %
processed: 77 %
processed: 78 %
processed: 78 %
processed: 79 %
processed: 80 %
processed: 81 %
processed: 81 %
```

```
processed: 82 %

processed: 83 %

processed: 84 %

processed: 84 %

processed: 85 %

processed: 86 %

processed: 87 %

processed: 87 %

processed: 88 %

processed: 89 %

processed: 90 %

processed: 90 %

processed: 91 %

processed: 92 %

processed: 93 %

processed: 93 %

processed: 94 %

processed: 95 %

processed: 96 %

processed: 96 %

processed: 97 %

processed: 98 %

processed: 99 %

processed: 99 %

processed: 100 %
## 35 .  Converged.
```




```
expectedUtility<-function(dataset, label="", Lfp=1,Utp=1,deltaPlus=1,guarantee=1e-5)
{
    postProbVar = pmax(dataset$postProbVar, guarantee)
    PrPlus = sum(dataset$postProbs/postProbVar)/
        sum(1/postProbVar)
    result = data.frame(label=label,
                        Utp=Utp, Lfp=Lfp, deltaPlus=deltaPlus,
                        nPairs=nrow(dataset),
                        PrPlus= PrPlus,
                        PrTrue= PrTrue<-PrPlus / deltaPlus,
                        PrFalse= PrFalse<-1 - PrTrue,
                        Utrue=  Utrue<-PrTrue * Utp,
                        Lfalse= Lfalse<-PrFalse * Lfp,
                        Eutility1= Utrue-Lfalse,
                        Eutility= nrow(dataset)*(Utrue-Lfalse))
    rownames(result) = label
    return(result)
}

Workflow.Evaluation.table<-function(Posterior.dataframe,Lfp=1,Utp=1,deltaPlus=1,guarantee=1e-5){
    WorkflowStats<-data.frame(sapply(strsplit(Posterior.dataframe$workflow_options_merged,"_"),"[",1),sa
    colnames(WorkflowStats)[1]<-"Marker"
    colnames(WorkflowStats)[2]<-"WorkflowID"
    og<-expectedUtility(label="Use All", dataset=WorkflowStats,Lfp=Lfp,Utp=Utp,deltaPlus=deltaPlus,guara
    for(p in unique(WorkflowStats$WorkflowID)){
        set<-expectedUtility(label=as.character(WorkflowStats$WorkflowID[as.numeric(p)]), dataset=Workfl
```

```
        og=rbind(og,set)
    }
    return(og)
}

Workflow.Evaluation.table(Posterior.dataframe)
```

```
##           label Utp Lfp deltaPlus nPairs    PrPlus    PrTrue    PrFalse
## Use All Use All   1   1         1    133 0.9543752 0.9543752 0.04562483
## 1             1   1   1         1     67 0.9390894 0.9390894 0.06091061
## 2             2   1   1         1     66 0.9698081 0.9698081 0.03019194
##             Utrue     Lfalse Eutility1  Eutility
## Use All 0.9543752 0.04562483 0.9087503 120.86380
## 1       0.9390894 0.06091061 0.8781788  58.83798
## 2       0.9698081 0.03019194 0.9396161  62.01466
```