Disease Specific Genomic Analysis: Identifying the Signature of Pathologic Biology

M Nicolau , R Tibshirani, A-L Børresen-Dale, and S S Jeffrey

Disease-specific genomic analysis: identifying the signature of pathologic biology
Bioinformatics 2007 vol. 23 (8) pp.957-65. Epub 2007 Feb 3.

```
# BACKGROUND FUNCTIONS

# COPY THE FUNCTION DEFINITIONS BELOW AND PASTE THEM IN YOUR R SESSION.
```

```
read.pcl <- function(filename,na.type = "",Nrows= -1,Comment.char="",...) {
     x.df <- read.table(paste(filename,"pcl",sep="."),header=TRUE,sep="\t",

quote="\"",as.is=rep(T,3),na.strings=na.type,skip=0,nrows=Nrows,comment.char=
Comment.char,...);
          rownames(x.df)<-x.df[[1]];
return(x.df)};

write.pcl <- function(df,dataname,fileaddress="") {
      dir.address <- paste(fileaddress,dataname,".pcl",sep="");
      X <- write.table(df,file=dir.address,

append=FALSE,quote=FALSE,sep="\t",eol="\n",na="",row.names=FALSE,col.names=TR
UE);
return(X)};
```

```
select.v <- function(x,indx) {y <- x[(indx)]; return(y)};

meshRows.dsga <- function(df1,df2) {
      rn <- rownames(df1)[{rownames(df1) %in% rownames(df2)}];
      ndf1 <- df1[rn,]; ndf2 <- df2[rn,];return(list(ndf1,ndf2))};

normvec <- function(vec) {norm <- sqrt(sum(vec * vec));return(norm)};

mat2pcl <- function(mat,tag) {mat.df <- as.data.frame(mat);
      mat.pcl <- cbind(tag,mat.df);return(mat.pcl)};

#### PCA.collapse.mat function produces list of the principal component
collapse of matrix, up to dimensions: (1:n); up to max n
true.diag <- function(vec) {ifelse(length(vec)>1.5,y <- diag(vec),y <-
as.matrix(vec));return(y)};
plugnew.vec <- function(vec,dimchange,newval=0)
      {vecnew <- vec;vecnew[-(1:dimchange)] <- rep(newval,length(vec) -
dimchange);return(vecnew)};

normvec <- function(vec, na.rm = FALSE) {norm <- sqrt(sum(vec * vec, na.rm =
na.rm));return(norm)}; # euclidean norm, L2 norm
l1vec <- function(vec, na.rm = TRUE) {junk1 <- sum(abs(vec), na.rm =
na.rm);return(junk1)};      # L1 norm

PCA.collapse.mat <- function(mat) {mat.svd <- svd(mat);
U.mat <- mat.svd$u;V.mat <- mat.svd$v;
D.mat <- list();
for(j in 1:length(mat.svd$d)) {D.mat[[j]] <-
true.diag(plugnew.vec(vec=mat.svd$d,dimchange = j))};
N <- list();
for(j in 1:ncol(U.mat)) {N[[j]] <- U.mat %*% D.mat[[j]] %*% t(V.mat)};
NN <- lapply(N,change.attributes,new.atr = attributes(mat));
return(NN)};

PCA.1collapse.mat <- function(mat.svd,indx) {U.mat <- mat.svd$u;V.mat <-
mat.svd$v;
D.mat <- true.diag(plugnew.vec(vec=mat.svd$d,dimchange = indx));
N <- U.mat %*% D.mat %*% t(V.mat)
return(N)};

PC1.col.vec <- function(mat) {svd1 <- svd(mat,nu=1,nv=1);
u.vec <- as.vector(svd1$u,mode="numeric");
lambda <- svd1$d[[1]];
v1 <- svd1$v[1,];
N <- lambda * v1 * u.vec; names(N) <- rownames(mat);
return(N)};


meshRows.norm.dsga <- function(df1,df2,meanFUN = mean, ...) {junk <-
meshRows.dsga(df1,df2);
Df1 <- junk[[1]];Df2 <- junk[[2]];
rm(junk);
mat1 <- as.matrix(Df1[-(1:3)]);mat2 <- as.matrix(Df2[-(1:3)]);
junk1 <- mean(apply(mat1,2,normvec));
```

```
junk2 <- mean(apply(mat2,2,normvec));
junk <- meanFUN(c(junk1,junk2),...);
rm(mat1,mat2,junk1,junk2);
Df1 <- fast.Knormalize.pcl(Df1,K = junk);
Df2 <- fast.Knormalize.pcl(Df2,K = junk);
return(list(Df1,Df2))};

FLAT.dsga <- function(mat) {
      f1 <- function(x,m) {nx <- fitted(lm(x ~(m - 1)));return(nx)};
      f2 <- function(j,m) {v <- m[,j];mm <- m[,-(j)];nv <-
f1(v,mm);return(nv)};
      bigindx <- rbind(1: ncol(mat));  colnames(bigindx) <- colnames(mat);
      matt <- apply(bigindx,2,f2,m = mat); return(matt)};

wold.dsga <- function(mat) {
      f1 <- function(svd.list,l) {
      n <- nrow(svd.list$u);k <- ncol(svd.list$u);lam.square <- {svd.list$d}
^2;
      junk <- {{lam.square[[l]]} / {sum(lam.square[-(1:l)])}} * {{n - l - 1}
* {k - l} / {n + k - (2 * l)}};
      return(junk)};
      svd.list <- svd(mat);
      dim.list <- as.list((1:(length(svd.list$d)-1)));
      junk <- lapply(dim.list,f1,svd.list=svd.list);
      junk <- unlist(junk);
      return(junk)};

plot.wold.dsga <- function(x,x.Lbound = 1,x.Ubound = length(x),main.extra="")
{plot.range <- (x.Lbound : x.Ubound);z <- x[plot.range];
      y <- plot(plot.range,z,type="l",lty="solid",xlab="Dimension of PC space
= l",ylab="W(l)",
      main=paste("Wold invariant",main.extra),col="blue",log =
"y");return(y)};

pca.dsga <- function(mat,j) {
      td <- function(vec){if(length(vec)>1.5) y=diag(vec) else
y=as.matrix(vec); return(y)};
      pn <- function(vec) {vn <- vec;vn[-(1:j)] <- rep(0,length(vec) -
j);return(vn)};
      mat.svd <- svd(mat);
      U.mat <- mat.svd$u;V.mat <- mat.svd$v;D.mat <- td(pn(vec=mat.svd$d));
      matt <- U.mat %*% D.mat %*% t(V.mat);
      attributes(matt) <- attributes(mat);
      return(matt)};

leave1out.dsga <- function(mat,j) {
      f1 <- function(x) {y <- pca.dsga(mat = FLAT.dsga(x),j = j);return(y)};
      f2 <- function(i,x) {v <- cbind(x[,i]);z <- x[,-(i)];y <- f1(z);z <-
disease.dsga(Dmat = v,Nmodel = y);return(cbind(z))};
      f3 <- function(lst) {vec <- unlist(lst);y <-
matrix(vec,ncol=length(lst),byrow=FALSE);colnames(y) <-
names(lst);rownames(y) <- names(lst[[1]]);return(y)};
      ls <- as.list(1:ncol(mat));ls.mat <- lapply(ls,f2,x = mat);
      newmat <- f3(ls.mat);attributes(newmat) <-
attributes(mat);colnames(newmat) <- paste("L1O",colnames(newmat),sep=".");
```

```
      return(newmat)};

normal.dsga <- function(Dmat,Nmodel,new.cnames = "Norm") {
      mat <- cbind(lm(Dmat ~ (Nmodel - 1))$fitted.values);
      colnames(mat) <- paste(colnames(Dmat),new.cnames,sep=".");
      return(mat)};

normal.coefficients.dsga <- function(Dmat,Nmodel) {
      mat <- lm(Dmat ~ (Nmodel - 1))$coefficients;
            return(mat)};

normal.coefficients.mag1.dsga <- function(Dmat,Nmodel) {
      dmat <- apply(Dmat,2,fast.normalize); nmodel <-
apply(Nmodel,2,fast.normalize);
      mat <- lm(dmat ~ (nmodel - 1))$coefficients;
            return(mat)};

disease.dsga <- function(Dmat,Nmodel,new.cnames = "Dis") {
      mat <- cbind(lm(Dmat ~ (Nmodel - 1))$residuals);
      colnames(mat) <- paste(colnames(Dmat),new.cnames,sep=".");
      return(mat)};


dsga_part1 <- function(normal.pcl ,tumor.pcl, normalname, dataname,
org.directory = "" )
{
   record <- list();
   record$ntumors.original <- ncol(tumor.pcl) - 3;
   record$ngenes.tumor.original <- nrow(tumor.pcl) ;
   record$nnormal.original <- ncol(normal.pcl) - 3;
   record$ngenes.normal.original <- nrow(normal.pcl) ;
   norm.tum.list <- meshRows.norm.dsga(df1 = normal.pcl, df2 = tumor.pcl,
meanFUN = select.v, indx = 1);
   rm(normal.pcl,tumor.pcl);
   Normal.pcl <- norm.tum.list[[1]];
   Disease.pcl <- norm.tum.list[[2]];
      write.pcl(Normal.pcl,paste(org.directory,normalname,".normMesh",sep =
""));
      write.pcl(Disease.pcl,paste(org.directory,dataname,".normMesh",sep =
""));
   record$ngenes.common <- nrow(Normal.pcl);
   rm(norm.tum.list);
   tag.pcl <- Disease.pcl[(1:3)];
   Normal.mat <- as.matrix(Normal.pcl[-(1:3)]);
   Disease.mat <- as.matrix(Disease.pcl[-(1:3)]);
   rm(Normal.pcl,Disease.pcl);
   ####  Start DSGA program
   flat.Nmat <- FLAT.dsga(Normal.mat);
   wold.Nmat <- wold.dsga(flat.Nmat);
junk <- list(tag.pcl = tag.pcl,Normal.mat = Normal.mat,Disease.mat =
Disease.mat,flat.Nmat = flat.Nmat,wold.Nmat = wold.Nmat,record = record);
return(junk)
};

dsga_part2 <- function(x,k,dataname,normalname)
```

```
{
     tag.pcl <- x$tag.pcl;
     Normal.mat <- x$Normal.mat;
     Disease.mat <- x$Disease.mat;
     flat.Nmat <- x$flat.Nmat;
     wold.Nmat <- x$wold.Nmat;
     record <- x$record;
     record$K <- k;
      rm(x);
Normal.model <- pca.dsga(mat = flat.Nmat,j = k);
L1.mat <- leave1out.dsga(mat = Normal.mat,j = k);
Dc.Dmat <- disease.dsga(Dmat = Disease.mat,Nmodel = Normal.model);
Nc.Dmat <- normal.dsga(Dmat = Disease.mat,Nmodel = Normal.model);
Dc.Nmat <- disease.dsga(Dmat = Normal.mat,Nmodel = Normal.model);
Nc.Nmat <- normal.dsga(Dmat = Normal.mat,Nmodel = Normal.model);
Org.Dmat  <- Dc.Dmat + Nc.Dmat
Org.Nmat  <- L1.mat + Nc.Nmat

Dc.Dpcl <- mat2pcl(mat = Dc.Dmat,tag = tag.pcl);
write.pcl(Dc.Dpcl,paste(dataname,"Tdis",sep = "."));
Nc.Dpcl <- mat2pcl(mat = Nc.Dmat,tag = tag.pcl);
write.pcl(Nc.Dpcl,paste(dataname,"Tnorm",sep = "."));
Dc.Npcl <- mat2pcl(mat = Dc.Nmat,tag = tag.pcl);
write.pcl(Dc.Npcl,paste(normalname,"Ndis",sep = "."));
Nc.Npcl <- mat2pcl(mat = Nc.Nmat,tag = tag.pcl);
write.pcl(Nc.Npcl,paste(normalname,"Nnorm",sep = "."));
NormMod.pcl <- mat2pcl(mat = Normal.model, tag = tag.pcl);
write.pcl(NormMod.pcl,paste(normalname,"NormalModel",sep = "."));
L1.pcl <- mat2pcl(mat = L1.mat, tag = tag.pcl);
write.pcl(L1.pcl,paste(normalname,"L1out",sep = "."));
Org.Dpcl <- mat2pcl(mat = Org.Dmat,tag = tag.pcl);
write.pcl(Org.Dpcl,paste(dataname,"normMesh",sep = "."));
Org.Npcl <- mat2pcl(mat = Org.Nmat,tag = tag.pcl);
write.pcl(Org.Npcl,paste(normalname,"normMesh",sep = "."));
#rm(Dc.Dpcl,Nc.Dpcl,Dc.Npcl,Nc.Npcl,NormMod.pcl, Org.Dpcl, Org.Npcl);
junk <- list(Dc.Dmat = Dc.Dmat,L1.pcl = L1.pcl,record = record);
return(junk)};
```

```
 # RUN DSGA BY ENTERING THE NAMES OF DATA
FILES AND THEN RUNNING THE 2 FUNCTIONS:

# dsga_part1()
# dsga_part2()

# this will generate and store all the
```

```
necessary data files in your working
directory
# you must first upload the original data
files in your R session, by following the
instructions below.
# data cannot have any missing values.
# If your .pcl files have missing values,
you must run an(y) algorithm to impute
missing data.
# We recommend knn-impute with k = 10
nearest neighbors.

# knn.impute.information:
# package :DMwR in Bioconductor
# function knnImputation
# k=10 neighbors

# reference:
# Torgo, L. (2010) Data Mining using R: learning with case studies,
CRC Press (ISBN: 9781439810187).
```

```
# DSGA step-by-step:


#  Read in 2 pcl files
#  original names of tumor data and normal data, place
these two .pcl files in a subdirectory called "xtra"
#        org.directory <- "xtra/"

org.tumorName <- "       ";    # USER ENTER THE NAME OF
THE TUMOR DATA, WITHOUT THE pcl EXTENTION.
org.normalName <- "       ";  # USER ENTER THE NAME OF
THE NORMAL DATA, WITHOUT THE pcl EXTENTION.
```

```
org.directory <- "xtra/";

NORMAL.pcl <-
read.pcl(paste(org.directory,org.normalName,sep = ""));
TUMOR.pcl <-
read.pcl(paste(org.directory,org.tumorName,sep = ""));
####  USER -- Choose a name for the data when it is stored -
this could be a shorter name than the original name
DataName <- "       ";            # USER ENTER A SHORT
NAME FOR THE TUMOR DATA, WITHOUT THE pcl EXTENTION. IT
CAN BE THE SAME AS ORIGINAL NAME
NormalName <- "       ";         # USER ENTER A SHORT
NAME FOR THE NORMAL DATA, WITHOUT THE pcl EXTENTION. IT
CAN BE THE SAME AS ORIGINAL NAME
Org.directory <- "xtra/";

####      DATA UPLOAD & HEALTHY STATE MODEL -

DSGA_part1 <- dsga_part1(normal.pcl =
NORMAL.pcl,tumor.pcl = TUMOR.pcl, normalname =
NormalName, dataname = DataName, org.directory =
Org.directory);

plot.wold.dsga(DSGA_part1$wold.Nmat, main.extra =
NormalName);


K <-            ;      # USER choose dimension K where the
wold plot peaks (has a local maximum).

# K <-  24;   for kidney
# K <-  10;   for NKI


abline(v = K, lty = "dotted", col = "red")
legend(x = K, y = 0.2 * DSGA_part1$wold.Nmat[[1]],
legend = paste("dim =", K), bty = "n");
```

```
####      DATA TRANSFORMATION -
####  DATA IS DECOMPOSED INTO DISEASE & NORMAL
COMPONENTS

DSGA_part2 <- dsga_part2(x = DSGA_part1,k =
K,dataname = DataName, normalname = NormalName);
```