plotforuk4

Here we plot the correlation matrix and the first 3 eigenvectors of uk4.

```
library('knitr')

knitr::opts_chunk$set(cache=TRUE)
opts_chunk$set(fig.path = "/Users/sarahurbut/Dropbox/PaperEdits/Paper/Figures/")
covmat=readRDS("../Data/covmatAug13withED.rds")

z.stat=read.table("../Data/maxz.txt")
names=colnames(z.stat)
pis=readRDS("../../Dropbox/withzero/piswithzero.rds")$pihat[-1189]
pi.mat=matrix(pis,ncol=54,nrow=22,byrow = T)
```

```
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(ggplot2)
library('colorRamps')
#install.packages("fields")
library(fields)
```

```
## Warning: package 'fields' was built under R version 3.2.5

## Loading required package: spam

## Loading required package: grid

## Spam version 1.3-0 (2015-10-24) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: maps
```

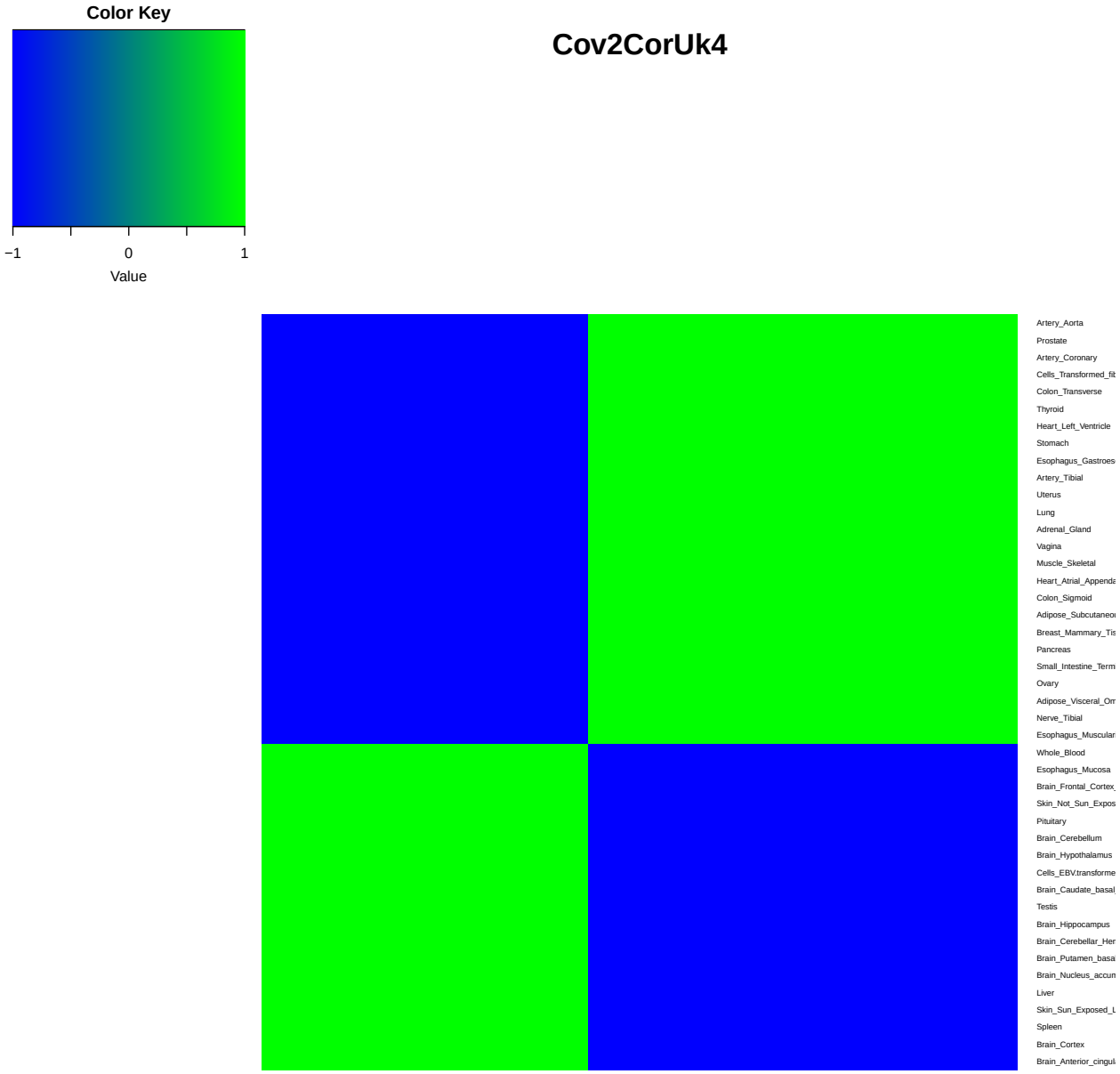
```
##
##  # maps v3.1: updated 'world': all lakes moved to separate new #
##  # 'lakes' database. Type '?world' or 'news(package="maps")'.  #
```

```
k=4
colSums(pi.mat)[k]
```

```
## [1] 0.02440052
```

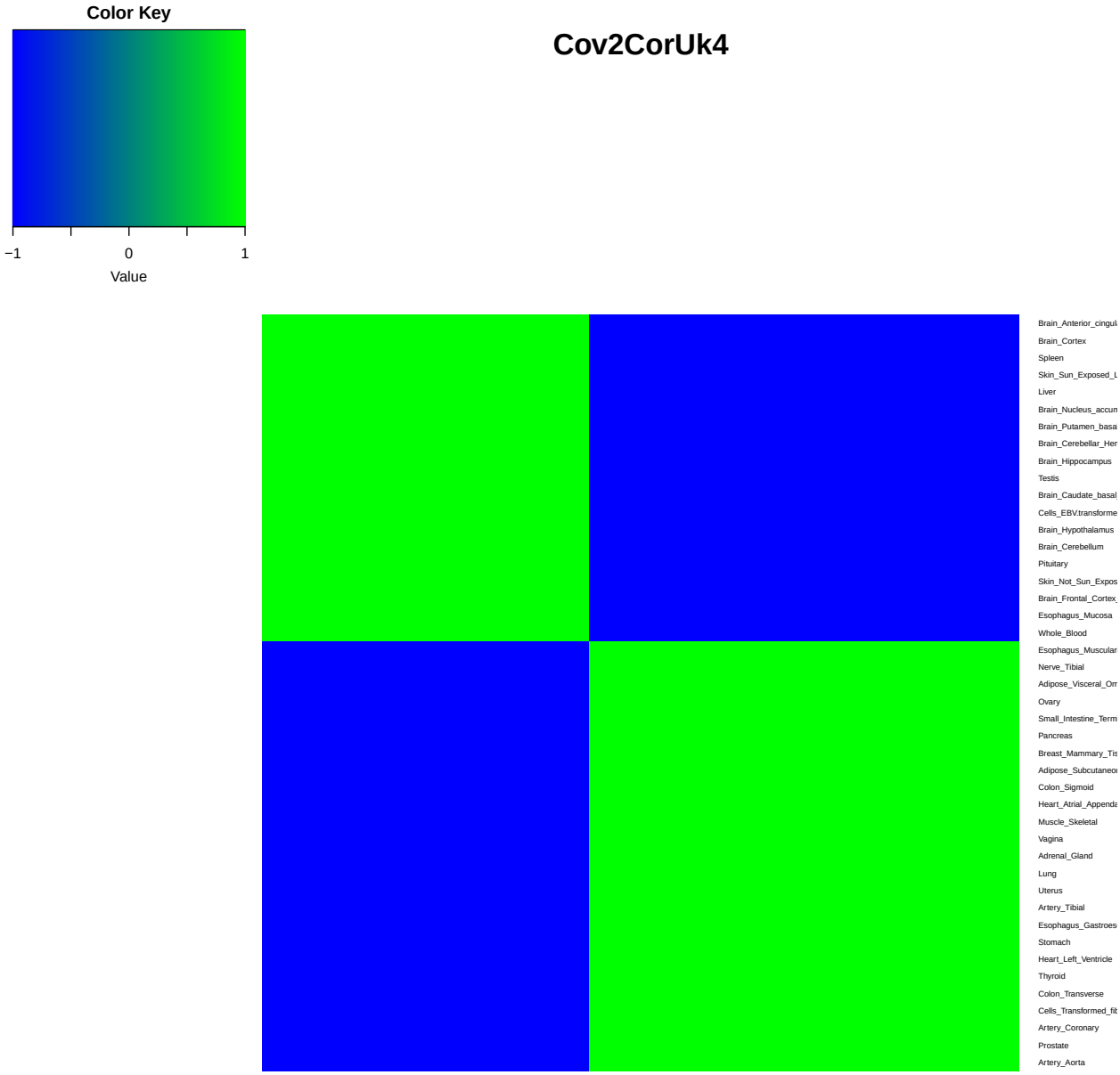
```
hclust.2=function (d, method = "average", members = NULL) {hclust(d, method, members)}
  x=cov2cor(covmat[[k]])
#x[x<0]=0
  colnames(x)=names
  rownames(x)=names


##generate indices
h=heatmap.2(x,#symm=TRUE,
           #Rowv=FALSE,Colv=FALSE,
           dendrogram="none",density="none",trace="none",#col=redblue,
           col=blue2green(256),
           main=paste0("Cov2CorUk",k),
           cexRow=0.5,cexCol=0.5,cex.main=0.5,labCol="")
```




```
###transpose map

heatmap.2(x[h$rowInd,h$rowInd],#symm=TRUE,
          Rowv=FALSE,Colv=FALSE,
          dendrogram="none",density="none",trace="none",#col=redblue,
          col=blue2green(256),
          main=paste0("Cov2CorUk",k),
          cexRow=0.5,cexCol=0.5,cex.main=0.5,labCol="")
```

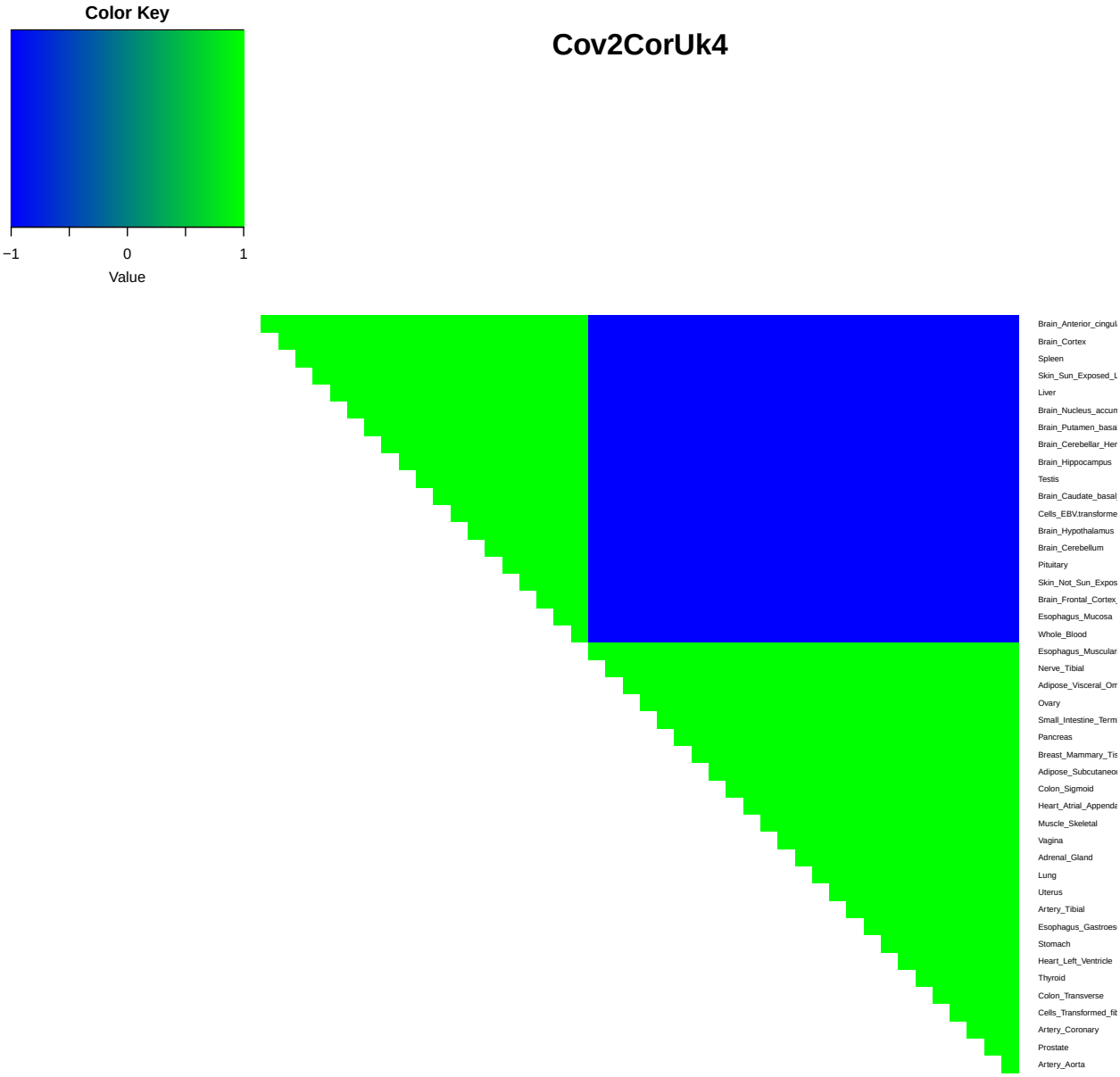



```
write.table(h$rowInd,file = paste0("uk",k,"rowIndices.txt"))
h=read.table(paste0("uk",k,"rowIndices.txt"))[,1]
```

heatmap:

```
smat=(x[h,h])
smat[lower.tri(smat)] <- NA


heatmap.2(smat,#symm=TRUE,
          Rowv=FALSE,Colv=FALSE,
          dendrogram="none",density="none",trace="none",#col=redblue,
          col=blue2green(256),
          main=paste0("Cov2CorUk",k),
          cexRow=0.5,cexCol=0.5,cex.main=0.5,labCol="")
```



As square:

```
#smat=(x[h,h])


# heatmap.2(smat,#symm=TRUE,
#           Rowv=FALSE,Colv=FALSE,
#           dendrogram="none",density="none",trace="none",#col=redblue,
#           col=blue2green(256),
#           main=paste0("Cov2CorUk3"),
#           cexRow=0.5,cexCol=0.5,cex.main=0.5,labCol="")
```

```
missing.tissues=c(7,8,19,20,24,25,31,34,37)
color.gtex=read.table("../Data/GTExColors.txt",sep = '\t', comment.char = '')[-missing.tissues,]
col = as.character(color.gtex[,2])
```

```
library('corrplot')
```

```
## Warning: package 'corrplot' was built under R version 3.2.5
```

```
corrplot((x[h,h]),type="upper",#cl.lim=c(-1,1),
         tl.col=col[h],tl.cex=0.8,method="color",col=rep(blue2green(256),2))
```

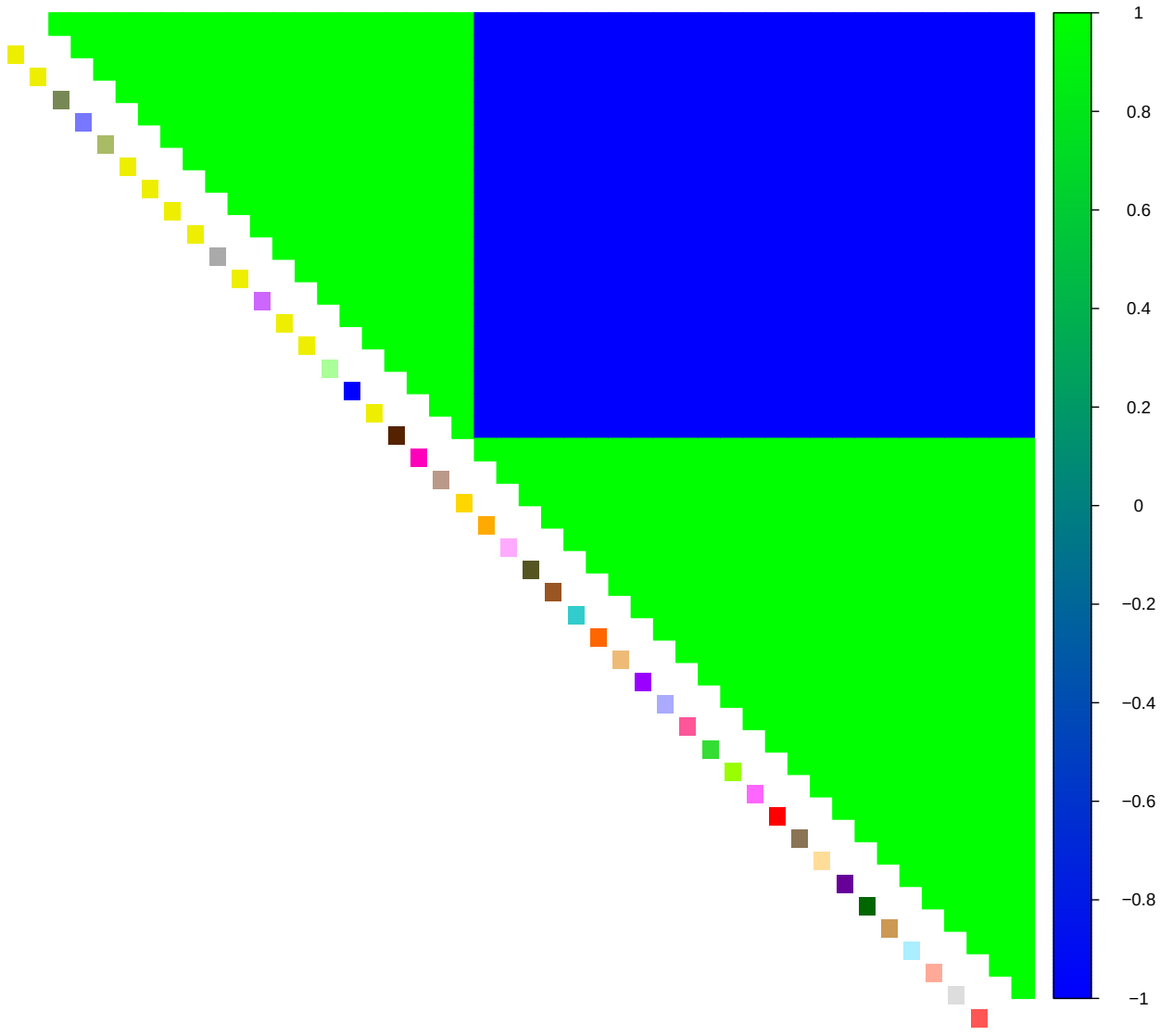
```
## Warning in ind1:ind2: numerical expression has 2 elements: only the first
## used
```

```
## Warning in ind1:ind2: numerical expression has 2 elements: only the first
## used
```


```
colnames(x)=NULL
rownames(x)=rep(".",44)
corrplot((x[h,h]),type="upper",#cl.lim=c(-1,1),
         tl.col=col[h],tl.cex=8,method="color",col=rep(blue2green(256),2))
```

```
## Warning in ind1:ind2: numerical expression has 2 elements: only the first
## used

## Warning in ind1:ind2: numerical expression has 2 elements: only the first
## used
```



And the SVD Plots:

```
k=4
h=read.table(paste0("uk",k,"rowIndices.txt"))[,1]
for(g in 1:3){
v=svd(covmat[[k]])$v[h,]
rownames(v)=colnames(v)=names(h)
```

```
par(mar=c(8,4.1,4.1,2.1))
barplot(v[,g]/v[which.max(abs(v[,g])),g],las=2,main=paste("Eigenvector",g,"of Uk",k),cex.names = 0.5,co
```