genefu: a package for breast cancer gene expression analysis

Deena M.A. Gendoo[1,2], Natchar Ratanasirigulchai[1], Markus Schröder[3], Gianluca Bontempi[4], Christos Sotiriou[5], John Quackenbush[6,7], and Benjamin Haibe-Kains[*1,2]

[1]Bioinformatics and Computational Genomics Laboratory, Princess Margaret Cancer Center, University Health Network, Toronto, Ontario, Canada
[2]Department of Medical Biophysics, University of Toronto, Toronto, Canada
[3]UCD School of Biomolecular and Biomedical Science, Conway Institute, University College Dublin, Belfield, Dublin, Ireland
[4]Machine Learning Group, Université Libre de Bruxelles
[5]Breast Cancer Translational Research Laboratory, Institut Jules Bordet, Université Libre de Bruxelles
[6]Computational Biology and Functional Genomics Laboratory, Dana-Farber Cancer Institute, Harvard School of Public Health
[7]Center for Cancer Computational Biology, Dana-Farber Cancer Institute

June 26, 2015

Contents

1 Introduction

The genefu package is providing relevant functions for gene expression analysis, especially in breast cancer. This package includes a number of algorithms for molecular subtype classification. The package also includes implementations of prognostic prediction algorithms, along with lists of prognostic gene signatures on which these algorithms were based.

Please refer to the manuscript URL and Lab website: http://www.pmgenomics.ca/bhklab/software/genefu

Please also refer to the References section below, for additional information on publications that have cited Version 1 of genefu.

*benjamin.haibe.kains@utoronto.ca

2 Loading package for case studies

First we load the genefu into the workspace. The pacakge is publicly available and can be installed from Bioconductor version 2.8 or higher in R version 2.13.0 or higher.

To install the genefu package:

```
knitr::opts_chunk$set(eval=TRUE,cache=TRUE)
source("http://bioconductor.org/biocLite.R")
biocLite("genefu")
```

For computing the risk scores, estimates of the performance of the risk scores, combining the estimates and comparing the estimates we have to load the genefu and survcomp packages into the workspace. We also load the xtable package to display results inside this document.

```
library(genefu)
library(xtable)
library(rmeta)
```

3 Case Study : Comparing risk prediction models

The following case study compares risk prediction models. This includes computing risk scores, computing estimates of the performance of the risk scores, as well as combining the estimates and comparing them.

The five data sets that we use in the case study are publicly available as experimental data packages on Bioconductor.org. In particular we used:

breastCancerMAINZ: http://www.bioconductor.org/packages/release/data/experiment/html/breastCancerMAINZ.html

breastCancerUPP: http://www.bioconductor.org/packages/release/data/experiment/html/breastCancerUPP.html

breastCancerUNT: http://www.bioconductor.org/packages/release/data/experiment/html/breastCancerUNT.html

breastCancerNKI: http://www.bioconductor.org/packages/release/data/experiment/html/breastCancerNKI.html

breastCancerTRANSBIG:http://www.bioconductor.org/packages/release/data/experiment/html/breastCancerTRANSBIG.ht

Please Note: We don't use the breastCancerVDX experimental package in this case study since it has been used as training data set for GENIUS. Please refer to Haibe-Kains et al, 2010. The breastCancerVDX is found at the following link:

breastCancerVDX: http://www.bioconductor.org/packages/release/data/experiment/html/breastCancerVDX.html

These experimental data packages can be installed from Bioconductor version 2.8 or higher in R version 2.13.0 or higher. For the experimental data packages the commands for installing the data sets are:

```
source("http://www.bioconductor.org/biocLite.R")
biocLite("breastCancerMAINZ")
biocLite("breastCancerTRANSBIG")
biocLite("breastCancerUPP")
biocLite("breastCancerUNT")
biocLite("breastCancerNKI")
```

And to load the packages into R, please use the following commands:

Table 1: Detailed overview for the data sets used in the case study

Dataset	Patients [#]	ER+ [#]	HER2+ [#]	Age [years]	Grade [1/2/3]	Platform
MAINZ	200	155	23	25-90	29/136/35	HGU133A
TRANSBIG	198	123	35	24-60	30/83/83	HGU133A
UPP	251	175	46	28-93	67/128/54	HGU133AB
UNT	137	94	21	24-73	32/51/29	HGU133AB
NKI	337	212	53	26-62	79/109/149	Agilent
Overall	1123	759	178	25-73	237/507/350	Affy/Agilent

```
library(breastCancerMAINZ)
library(breastCancerTRANSBIG)
library(breastCancerUPP)
library(breastCancerUNT)
library(breastCancerNKI)
```

Table1 shows an overview of the data sets and the patients. From those 1123 breast cancer patients we selected only the patients that are node negative and didn't receive any treatment (except local radiotherapy), which results in 713 patients.

Since there are duplicated patients in the five data sets, we have to identify the duplicated patients and we subsequently store them in a vector.

```
library(Biobase)
data(breastCancerData)
cinfo <- colnames(pData(mainz7g))
data.all <- c("transbig7g"=transbig7g, "unt7g"=unt7g, "upp7g"=upp7g, "mainz7g"=mainz7g, "nki7g"=nki7g)

idtoremove.all <- NULL
duplres <- NULL

## No overlaps in the MainZ and NKI datasets.

## Focus on UNT vs UPP vs TRANSBIG
demo.all <- rbind(pData(transbig7g), pData(unt7g), pData(upp7g))
dn2 <- c("TRANSBIG", "UNT", "UPP")

## Karolinska
## Search for the VDXKIU, KIU, UPPU series
ds2 <- c("VDXKIU", "KIU", "UPPU")
demot <- demo.all[complete.cases(demo.all[ , c("series")]) & is.element(demo.all[ , "series"], ds2), ]

# Find the duplicated patients in that series
duplid <- sort(unique(demot[duplicated(demot[ , "id"]), "id"]))
duplrest <- NULL
for(i in 1:length(duplid)) {
  tt <- NULL
  for(k in 1:length(dn2)) {
    myx <- sort(row.names(demot)[complete.cases(demot[ , c("id", "dataset")]) &
                                 demot[ , "id"] == duplid[i] & demot[ , "dataset"] == dn2[k]])
    if(length(myx) > 0) { tt <- c(tt, myx) }
  }
  duplrest <- c(duplrest, list(tt))
}
```

```
names(duplrest) <- duplid
duplres <- c(duplres, duplrest)

## Oxford
## Search for the VVDXOXFU, OXFU series
ds2 <- c("VDXOXFU", "OXFU")
demot <- demo.all[complete.cases(demo.all[ , c("series")]) & is.element(demo.all[ , "series"], ds2), ]

# Find the duplicated patients in that series
duplid <- sort(unique(demot[duplicated(demot[ , "id"]), "id"]))
duplrest <- NULL
for(i in 1:length(duplid)) {
  tt <- NULL
  for(k in 1:length(dn2)) {
    myx <- sort(row.names(demot)[complete.cases(demot[ , c("id", "dataset")]) &
                                 demot[ , "id"] == duplid[i] & demot[ , "dataset"] == dn2[k]])
    if(length(myx) > 0) { tt <- c(tt, myx) }
  }
  duplrest <- c(duplrest, list(tt))
}
names(duplrest) <- duplid
duplres <- c(duplres, duplrest)

## Full set duplicated patients
duPL <- sort(unlist(lapply(duplres, function(x) { return(x[-1]) } )))
```

** Computing Risk Scores of Prognostic Signatures for Each Dataset:**
We compute the risk scores using the following list of algorithms (and corresponding genefu functions):

Subtype Clustering Model using just the AURKA gene: scmgene.robust()

Subtype Clustering Model using just the ESR1 gene: scmgene.robust()

Subtype Clustering Model using just the ERBB2 gene: scmgene.robust()

NPI: npi()

GGI: ggi()

GENIUS: genius()

EndoPredict: endoPredict()

OncotypeDx: oncotypedx()

TamR: tamr()

GENE70: gene70()

PIK3CA: pik3cags()

rorS: rorS()

```
dn <- c("transbig", "unt", "upp", "mainz", "nki")
dn.platform <- c("affy", "affy", "affy", "affy", "agilent")

res <- ddemo.all <- ddemo.coln <- NULL
```

```r
for(i in 1:length(dn)) {

  ## load dataset
  dd <- get(data(list=dn[i]))

  #Extract expression set, pData, fData for each dataset
  ddata <- t(exprs(dd))
  ddemo <- phenoData(dd)@data
  dannot <- featureData(dd)@data
  ddemo.all <- c(ddemo.all, list(ddemo))
  if(is.null(ddemo.coln))
  { ddemo.coln <- colnames(ddemo) } else
  { ddemo.coln <- intersect(ddemo.coln, colnames(ddemo)) }
  rest <- NULL

  ## AURKA
  ## if affy platform consider the probe published in Desmedt et al., CCR, 2008
  if(dn.platform[i] == "affy") { domap <- FALSE } else { domap <- TRUE }
  modt <- scmgene.robust$mod$AURKA
  ## if agilent platform consider the probe published in Desmedt et al., CCR, 2008
  if(dn.platform[i] == "agilent") {
    domap <- FALSE
    modt[ , "probe"] <- "NM_003600"
  }
  rest <- cbind(rest, "AURKA"=sig.score(x=modt, data=ddata, annot=dannot, do.mapping=domap)$score)

  ## ESR1
  ## if affy platform consider the probe published in Desmedt et al., CCR, 2008
  if(dn.platform[i] == "affy") { domap <- FALSE } else { domap <- TRUE }
  modt <- scmgene.robust$mod$ESR1
  ## if agilent platform consider the probe published in Desmedt et al., CCR, 2008
  if(dn.platform[i] == "agilent") {
    domap <- FALSE
    modt[ , "probe"] <- "NM_000125"
  }
  rest <- cbind(rest, "ESR1"=sig.score(x=modt, data=ddata, annot=dannot, do.mapping=domap)$score)

  ## ERBB2
  ## if affy platform consider the probe published in Desmedt et al., CCR, 2008
  if(dn.platform[i] == "affy") { domap <- FALSE } else { domap <- TRUE }
  modt <- scmgene.robust$mod$ERBB2
  ## if agilent platform consider the probe published in Desmedt et al., CCR, 2008
  if(dn.platform[i] == "agilent") {
    domap <- FALSE
    modt[ , "probe"] <- "NM_004448"
  }
  rest <- cbind(rest, "ERBB2"=sig.score(x=modt, data=ddata, annot=dannot, do.mapping=domap)$score)

  ## NPI
  ss <- ddemo[ , "size"]
  gg <- ddemo[ , "grade"]
  nn <- rep(NA, nrow(ddemo))
  nn[complete.cases(ddemo[ , "node"]) & ddemo[ , "node"] == 0] <- 1
```

```
  nn[complete.cases(ddemo[ , "node"]) & ddemo[ , "node"] == 1] <- 3
  names(ss) <- names(gg) <- names(nn) <- rownames(ddemo)
  rest <- cbind(rest, "NPI"=npi(size=ss, grade=gg, node=nn, na.rm=TRUE)$score)

  ## GGI
  if(dn.platform[i] == "affy") { domap <- FALSE } else { domap <- TRUE }
  rest <- cbind(rest, "GGI"=ggi(data=ddata, annot=dannot, do.mapping=domap)$score)

  ## GENIUS
  if(dn.platform[i] == "affy") { domap <- FALSE } else { domap <- TRUE }
  rest <- cbind(rest, "GENIUS"=genius(data=ddata, annot=dannot, do.mapping=domap)$score)

  ## ENDOPREDICT
  if(dn.platform[i] == "affy") { domap <- FALSE } else { domap <- TRUE }
  rest <- cbind(rest, "EndoPredict"=endoPredict(data=ddata, annot=dannot, do.mapping=domap)$score)

  # OncotypeDx
  if(dn.platform[i] == "affy") { domap <- FALSE } else { domap <- TRUE }
  rest <- cbind(rest, "OncotypeDx"=oncotypedx(data=ddata, annot=dannot, do.mapping=domap)$score)

  ## TamR
  # Note: risk is not implemented, the function will return NA values
  if(dn.platform[i] == "affy") { domap <- FALSE } else { domap <- TRUE }
  rest <- cbind(rest, "TAMR13"=tamr13(data=ddata, annot=dannot, do.mapping=domap)$score)

  ## GENE70
  # Need to do mapping for Affy platforms because this is based on Agilent. Hence the mapping rule is re
  if(dn.platform[i] == "affy") { domap <- TRUE } else { domap <- FALSE }
  rest <- cbind(rest, "GENE70"=gene70(data=ddata, annot=dannot, std="none",do.mapping=domap)$score)

  ## Pik3cags
  if(dn.platform[i] == "affy") { domap <- FALSE } else { domap <- TRUE }
  rest <- cbind(rest, "PIK3CA"=pik3cags(data=ddata, annot=dannot, do.mapping=domap))

  ## rorS
  # Uses the pam50 algorithm. Need to do mapping for both Affy and Agilent
  rest <- cbind(rest, "rorS"=rorS(data=ddata, annot=dannot, do.mapping=TRUE)$score)

  ## GENE76
  # Mainly designed for Affy platforms. Has been excluded here

  # BIND ALL TOGETHER
  res <- rbind(res, rest)
}
names(ddemo.all) <- dn
```

For further analysis and handling of the data we store all information in one object. We also remove the duplicated patients from the analysis and take only those patients into account, that have complete information for nodal, survival and treatment status.

```
ddemot <- NULL
for(i in 1:length(ddemo.all)) {
  ddemot <- rbind(ddemot, ddemo.all[[i]][ , ddemo.coln, drop=FALSE])
}
```

```
res[complete.cases(ddemot[ ,"dataset"]) & ddemot[ ,"dataset"] == "VDX", "GENIUS"] <- NA

## select only untreated node-negative patients with all risk predictions
## ie(incomplete cases (where a risk prediction may be missing for a sample) are subsequently removed))
# Note that increasing the number of risk prediction analyses may increase the number of incomplete case
# In the previous vignette for genefu version1, we were only testing 4 risk predictors, so we had a tota
# Here, we are now testing 12 risk predictors, so we only have 713 complete cases remaining.
# The difference of 9 cases between the two versions are all from the NKI dataset.
myx <- complete.cases(res, ddemot[ , c("node", "treatment")]) &
  ddemot[ , "treatment"] == 0 & ddemot[ , "node"] == 0 & !is.element(rownames(ddemot), duPL)

res <- res[myx, , drop=FALSE]
ddemot <- ddemot[myx, , drop=FALSE]
```

To compare the risk score performances, we compute the concordance index[1]:

Definition of the concordance index for a risk prediction: the probability that, for a pair of randomly chosen comparable samples, the sample with the higher risk prediction will experience an event before the other sample or belongs to a higher binary class.

```
cc.res <- complete.cases(res)
datasetList <- c("MAINZ","TRANSBIG","UPP","UNT","NKI")
riskPList <- c("AURKA","ESR1","ERBB2","NPI", "GGI", "GENIUS",
               "EndoPredict","OncotypeDx","TAMR13","GENE70","PIK3CA","rorS")
setT <- setE <- NULL
resMatrix <- as.list(NULL)

for(i in datasetList)
{
  dataset.only <- ddemot[,"dataset"] == i
  patientsAll <- cc.res & dataset.only

  ## set type of available survival data
  if(i == "UPP") {
    setT <- "t.rfs"
    setE <- "e.rfs"
  } else {
    setT <- "t.dmfs"
    setE <- "e.dmfs"
  }

  # Calculate cindex computation for each predictor
  for (Dat in riskPList)
  {
    cindex <- t(apply(X=t(res[patientsAll,Dat]), MARGIN=1, function(x, y, z) {
    tt <- concordance.index(x=x, surv.time=y, surv.event=z, method="noether", na.rm=TRUE);
    return(c("cindex"=tt$c.index, "cindex.se"=tt$se, "lower"=tt$lower, "upper"=tt$upper)); },
    y=ddemot[patientsAll,setT], z=ddemot[patientsAll, setE]))

    resMatrix[[Dat]] <- rbind(resMatrix[[Dat]], cindex)
  }
}
```

[1]The same analysis could be performed with D index and hazard ratio by using the functions `D.index` and `hazard.ratio` from the *survcomp* package respectively

Using a random-effects model we combine the dataset-specific performance estimated into overall estimates for each risk prediction model:

```
for(i in names(resMatrix)){
  #Get a meta-estimate
  ceData <- combine.est(x=resMatrix[[i]][,"cindex"], x.se=resMatrix[[i]][,"cindex.se"], hetero=TRUE)
  cLower <- ceData$estimate + qnorm(0.025, lower.tail=TRUE) * ceData$se
  cUpper <- ceData$estimate + qnorm(0.025, lower.tail=FALSE) * ceData$se

  cindexO <- cbind("cindex"=ceData$estimate, "cindex.se"=ceData$se, "lower"=cLower, "upper"=cUpper)
  resMatrix[[i]] <- rbind(resMatrix[[i]], cindexO)
  rownames(resMatrix[[i]]) <- c(datasetList, "Overall")
}
```

In order to compare the different risk prediction models we compute one-sided p-values of the meta-estimates:

```
pv <- sapply(resMatrix, function(x) { return(x["Overall", c("cindex","cindex.se")]) })
pv <- apply(pv, 2, function(x) { return(pnorm((x[1] - 0.5) / x[2], lower.tail=x[1] < 0.5)) })
printPV <- matrix(pv,ncol=length(names(resMatrix)))
rownames(printPV) <- "P-value"
colnames(printPV) <- names(pv)
printPV<-t(printPV)
```

And print the table of P-values:

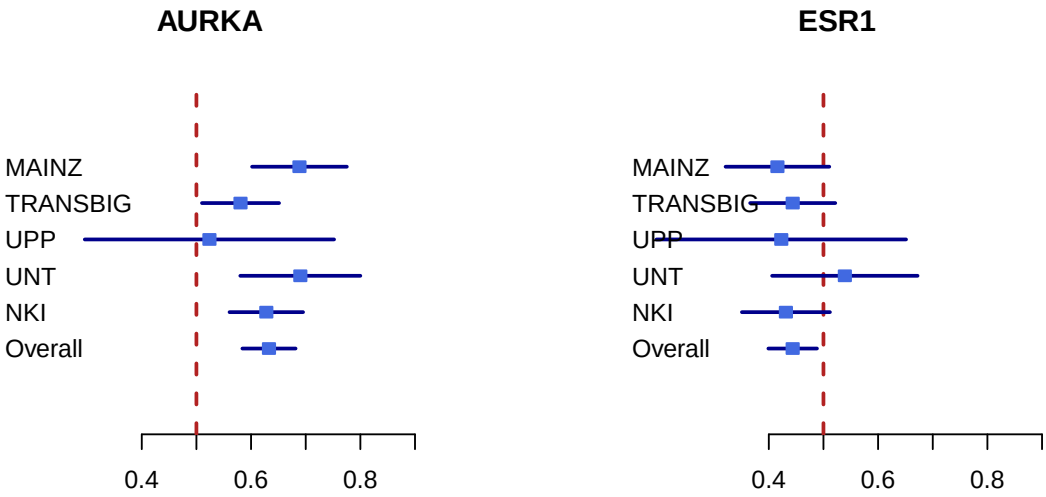
```
xtable(printPV, digits=c(0, -1))
```

	P-value
AURKA	4.5E-08
ESR1	6.5E-03
ERBB2	4.6E-01
NPI	1.8E-15
GGI	2.8E-14
GENIUS	6.1E-23
EndoPredict	7.7E-13
OncotypeDx	9.3E-14
TAMR13	2.5E-07
GENE70	1.8E-10
PIK3CA	2.3E-03
rorS	5.9E-12

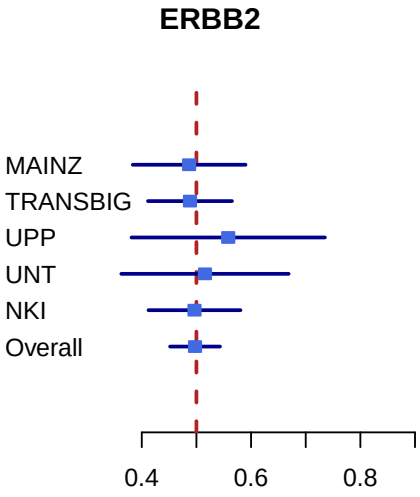
The following figures represent the risk score performances measured by the concordance index each of the prognostic predictors.

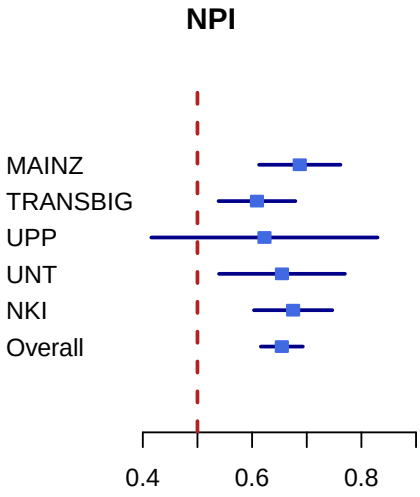
```
RiskPList <- c("AURKA","ESR1","ERBB2","NPI", "GGI", "GENIUS",
               "EndoPredict","OncotypeDx","TAMR13","GENE70","PIK3CA","rorS")
datasetListF <- c("MAINZ","TRANSBIG","UPP","UNT","NKI", "Overall")
myspace <- "   "
par(mfrow=c(2,2))
  for (RP in RiskPList)
  {
```

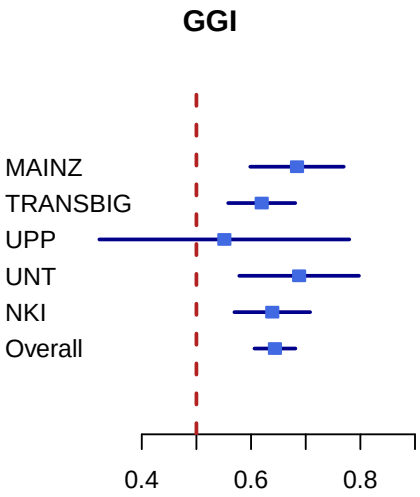
```
#<<forestplotDat,fig=TRUE>>=
## Forestplot
tt <- rbind(resMatrix[[RP]][1:5,],
          "Overall"=resMatrix[[RP]][6,])

tt <- as.data.frame(tt)
labeltext <- (datasetListF)

r.mean <- c(tt$cindex)
r.lower <- c(tt$lower)
r.upper <- c(tt$upper)

metaplot.surv(mn=r.mean, lower=r.lower, upper=r.upper, labels=labeltext, xlim=c(0.4,0.9),
              boxsize=0.5, zero=0.5,
              col=meta.colors(box="royalblue",line="darkblue",zero="firebrick"),
              main=paste(RP))

}
```

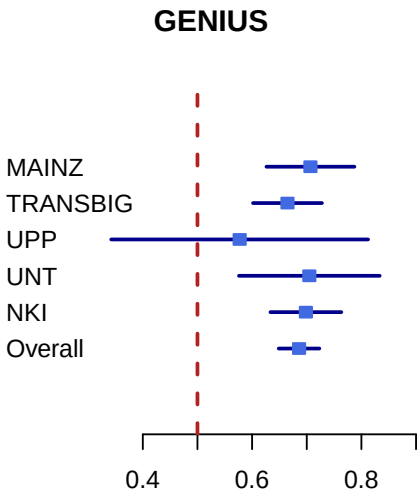

AURKA
MAINZ
TRANSBIG
UPP
UNT
NKI
Overall
0.4
0.6
0.8

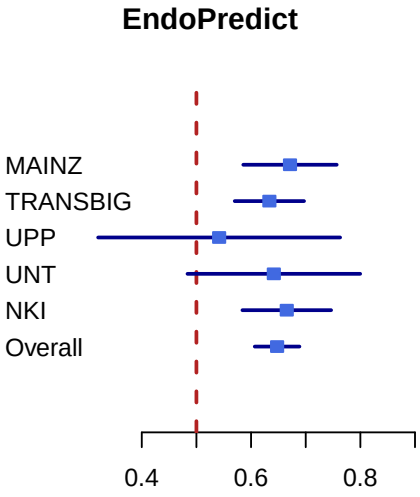
ESR1
MAINZ
TRANSBIG
UPP
UNT
NKI
Overall
0.4
0.6
0.8

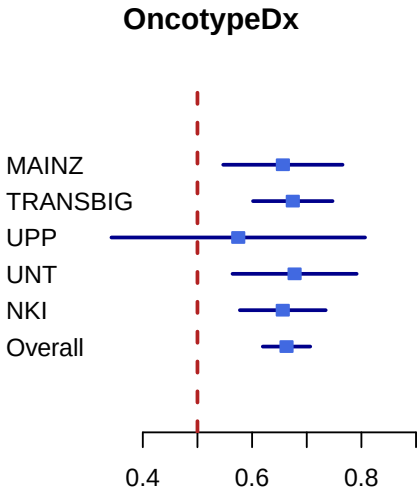

ERBB2
MAINZ
TRANSBIG
UPP
UNT
NKI
Overall
0.4
0.6
0.8

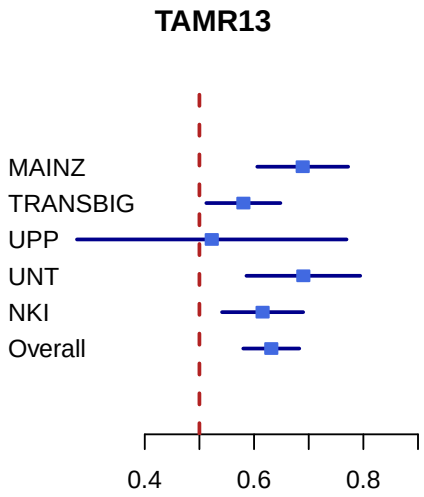

NPI
MAINZ
TRANSBIG
UPP
UNT
NKI
Overall
0.4
0.6
0.8

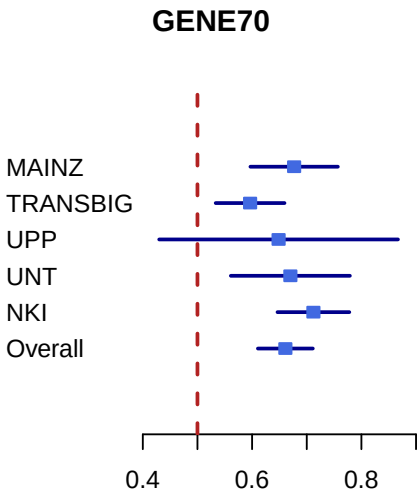

GGI
MAINZ
TRANSBIG
UPP
UNT
NKI
Overall
0.4
0.6
0.8

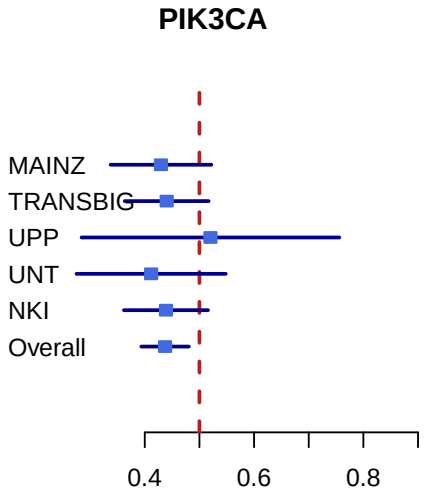

GENIUS
MAINZ
TRANSBIG
UPP
UNT
NKI
Overall
0.4
0.6
0.8

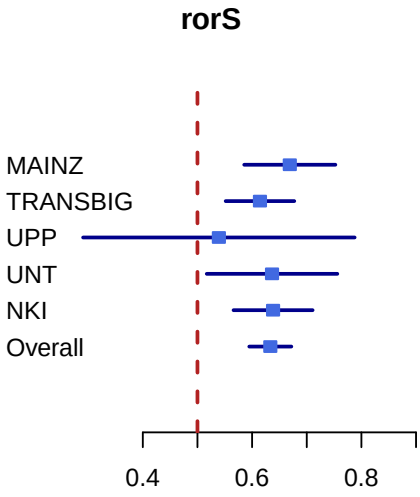

EndoPredict
MAINZ
TRANSBIG
UPP
UNT
NKI
Overall
0.4
0.6
0.8


OncotypeDx
MAINZ
TRANSBIG
UPP
UNT
NKI
Overall
0.4
0.6
0.8







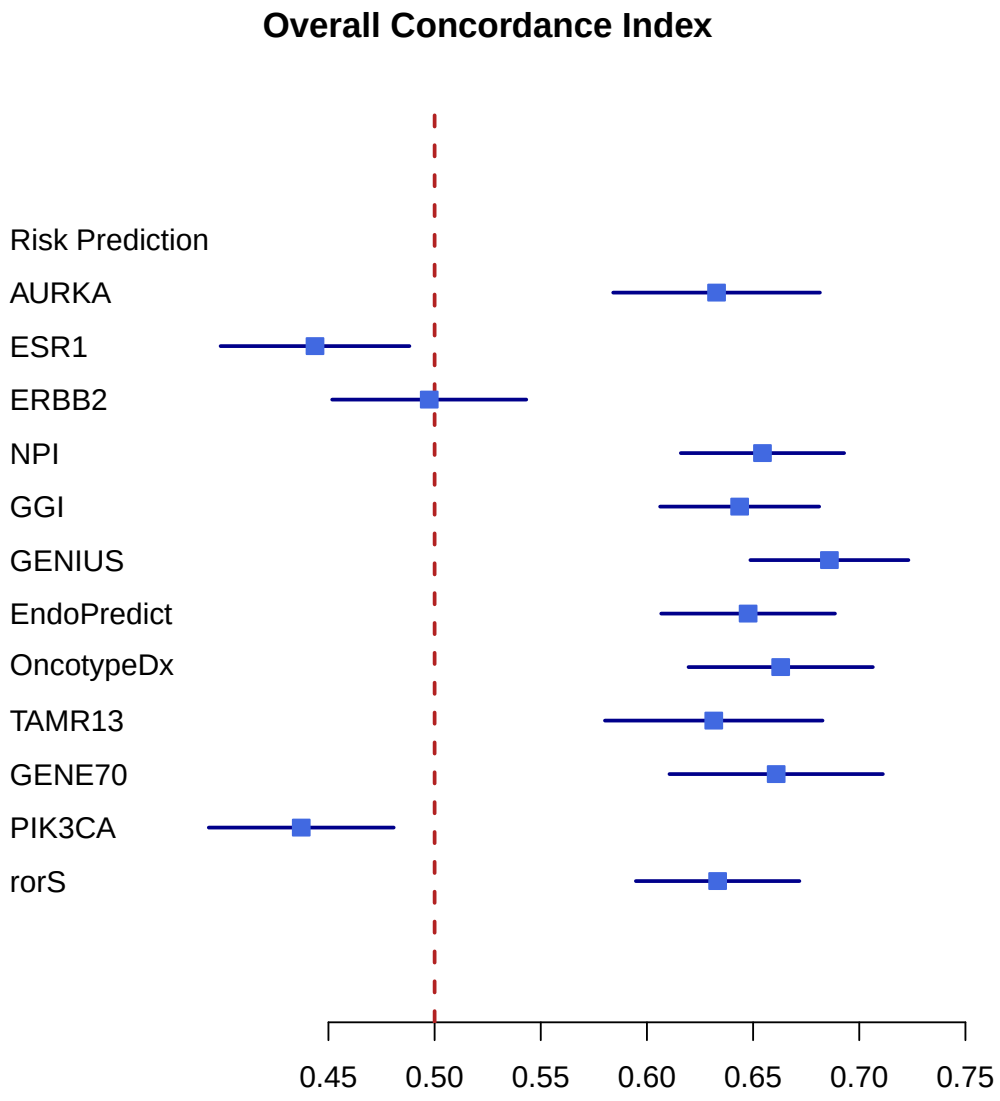



```
#@
#
```

We can also represent the overall estimates across all prognostic predictors, across all the datasets.

```
## Overall Forestplot
mybigspace <- "       "
tt <- rbind("OverallA"=resMatrix[["AURKA"]][6,],
            "OverallE1"=resMatrix[["ESR1"]][6,],
            "OverallE2"=resMatrix[["ERBB2"]][6,],
            "OverallN"=resMatrix[["NPI"]][6,],
          "OverallM"=resMatrix[["GGI"]][6,],
          "OverallG"=resMatrix[["GENIUS"]][6,],
          "OverallE3"=resMatrix[["EndoPredict"]][6,],
```

```
          "OverallOD"=resMatrix[["OncotypeDx"]][6,],
          "OverallT"=resMatrix[["TAMR13"]][6,],
          "OverallG70"=resMatrix[["GENE70"]][6,],
          "OverallP"=resMatrix[["PIK3CA"]][6,],
          "OverallR"=resMatrix[["rorS"]][6,]
          )

tt <- as.data.frame(tt)
labeltext <- cbind(c("Risk Prediction","AURKA","ESR1","ERBB2","NPI",
                     "GGI","GENIUS","EndoPredict","OncotypeDx","TAMR13","GENE70","PIK3CA","rorS"))

r.mean <- c(NA,tt$cindex)
r.lower <- c(NA,tt$lower)
r.upper <- c(NA,tt$upper)

metaplot.surv(mn=r.mean, lower=r.lower, upper=r.upper, labels=labeltext, xlim=c(0.45,0.75),
              boxsize=0.5, zero=0.5,
              col=meta.colors(box="royalblue",line="darkblue",zero="firebrick"),
              main="Overall Concordance Index")
```



In order to assess the difference between the risk scores, we compute the concordance indices with their p-values and compare the estimates with the `cindex.comp.meta` with a paired student t test.

```
cc.res <- complete.cases(res)
datasetList <- c("MAINZ","TRANSBIG","UPP","UNT","NKI")
riskPList <- c("AURKA","ESR1","ERBB2","NPI","GGI","GENIUS",
               "EndoPredict","OncotypeDx","TAMR13","GENE70","PIK3CA","rorS")
setT <- setE <- NULL
resMatrixFull <- as.list(NULL)

for(i in datasetList)
{
  dataset.only <- ddemot[,"dataset"] == i
  patientsAll <- cc.res & dataset.only
```

```
  ## set type of available survival data
  if(i == "UPP") {
    setT <- "t.rfs"
    setE <- "e.rfs"
  } else {
    setT <- "t.dmfs"
    setE <- "e.dmfs"
  }

  ## cindex and p-value computation per algorithm
  for (Dat in riskPList)
  {
    cindex <- t(apply(X=t(res[patientsAll,Dat]), MARGIN=1, function(x, y, z) {
    tt <- concordance.index(x=x, surv.time=y, surv.event=z, method="noether", na.rm=TRUE);
    return(tt); },
    y=ddemot[patientsAll,setT], z=ddemot[patientsAll, setE]))

    resMatrixFull[[Dat]] <- rbind(resMatrixFull[[Dat]], cindex)
  }
}

for(i in names(resMatrixFull)){
  rownames(resMatrixFull[[i]]) <- datasetList
}

ccmData <- tt <- rr <- NULL
for(i in 1:length(resMatrixFull)){
  tt <- NULL
  for(j in 1:length(resMatrixFull)){
    if(i != j) { rr <- cindex.comp.meta(list.cindex1=resMatrixFull[[i]],
                                        list.cindex2=resMatrixFull[[j]], hetero=TRUE)$p.value }
    else { rr <- 1 }
    tt <- cbind(tt, rr)
  }
  ccmData <- rbind(ccmData, tt)
}
ccmData <- as.data.frame(ccmData)
colnames(ccmData) <- riskPList
rownames(ccmData) <- riskPList
```

Table 2 displays the for multiple testing uncorrected p-values for the comparison of the different methods:

```
xtable(ccmData, digits=c(0, rep(-1,ncol(ccmData))))
```

We correct the p-value with Holms method:

```
ccmDataPval <- matrix(p.adjust(data.matrix(ccmData), method="holm"),ncol=length(riskPList),
                      dimnames=list(rownames(ccmData),colnames(ccmData)))
```

Table 3 displays the corrected p-values:

```
xtable(ccmDataPval, digits=c(0, rep(-1,ncol(ccmDataPval))))
```

	AURKA	ESR1	ERBB2	NPI	GGI	GENIUS	EndoPredict	OncotypeDx	TAMR13
AURKA	1.0E+00	1.0E-07	5.0E-05	7.8E-01	6.9E-01	9.8E-01	7.3E-01	8.8E-01	4.8E-01
ESR1	1.0E+00	1.0E+00	9.5E-01	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
ERBB2	1.0E+00	4.6E-02	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
NPI	2.2E-01	3.5E-11	2.8E-07	1.0E+00	3.3E-01	9.0E-01	3.9E-01	6.3E-01	2.1E-01
GGI	3.1E-01	5.0E-10	1.3E-06	6.7E-01	1.0E+00	9.7E-01	5.8E-01	8.2E-01	3.1E-01
GENIUS	2.3E-02	2.2E-15	4.8E-10	1.0E-01	2.9E-02	1.0E+00	5.6E-02	1.8E-01	2.1E-02
EndoPredict	2.7E-01	2.3E-09	1.1E-06	6.1E-01	4.2E-01	9.4E-01	1.0E+00	7.6E-01	2.7E-01
OncotypeDx	1.2E-01	9.7E-10	2.4E-07	3.7E-01	1.8E-01	8.2E-01	2.4E-01	1.0E+00	1.4E-01
TAMR13	5.2E-01	1.6E-07	7.7E-05	7.9E-01	6.9E-01	9.8E-01	7.3E-01	8.6E-01	1.0E+00
GENE70	1.5E-01	9.1E-09	3.1E-06	4.1E-01	2.3E-01	8.2E-01	3.0E-01	5.3E-01	1.6E-01
PIK3CA	1.0E+00	5.9E-01	9.8E-01	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
rorS	4.9E-01	9.8E-09	6.1E-06	8.1E-01	7.1E-01	9.9E-01	7.6E-01	9.2E-01	4.7E-01

	AURKA	ESR1	ERBB2	NPI	GGI	GENIUS	EndoPredict	OncotypeDx	TAMR13
AURKA	1.0E+00	1.3E-05	6.0E-03	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
ESR1	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
ERBB2	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
NPI	1.0E+00	4.9E-09	3.5E-05	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
GGI	1.0E+00	6.9E-08	1.5E-04	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
GENIUS	1.0E+00	3.1E-13	6.5E-08	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
EndoPredict	1.0E+00	3.1E-07	1.4E-04	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
OncotypeDx	1.0E+00	1.3E-07	3.0E-05	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
TAMR13	1.0E+00	2.0E-05	9.1E-03	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
GENE70	1.0E+00	1.2E-06	3.7E-04	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
PIK3CA	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00
rorS	1.0E+00	1.3E-06	7.3E-04	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00	1.0E+00

4 References

The following is a list of publications that have cited genefu (Version 1) in the past.

** Where genefu was used in subtyping:**

Larsen, M.J. et al., 2014. Microarray-Based RNA Profiling of Breast Cancer: Batch Effect Removal Improves Cross-Platform Consistency. BioMed Research International, 2014, pp.1-11.

Miller, T.W. et al., 2011. A gene expression signature from human breast cancer cells with acquired hormone independence identifies MYC as a mediator of antiestrogen resistance. Clinical cancer research : an official journal of the American Association for Cancer Research, 17(7), pp.2024-2034.

Karn, T. et al., 2011. Homogeneous Datasets of Triple Negative Breast Cancers Enable the Identification of Novel Prognostic and Predictive Signatures S. Ranganathan, ed. PloS one, 6(12), p.e28403.

** Where genefu was used in Comparing Subtyping Schemes:**

Haibe-Kains, B. et al., 2012. A three-gene model to robustly identify breast cancer molecular subtypes. Journal of the National Cancer Institute, 104(4), pp.311-325.

Curtis, C. et al., 2012. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature.

Balko, J.M. et al., 2012. Profiling of residual breast cancers after neoadjuvant chemotherapy identifies DUSP4 deficiency as a mechanism of drug resistance. Nature medicine, 18(7), pp.1052-1059.

Paquet, E.R. and Hallett, M.T., 2015. Absolute Assignment of Breast Cancer Intrinsic Molecular Subtype. Journal of the National Cancer Institute, 107(1), pp.dju357-dju357.

Patil, P. et al., 2015. Test set bias affects reproducibility of gene signatures. Bioinformatics, p.btv157.

** Where genefu was used to Compute Prognostic gene signature scores:**

Haibe-Kains, B. et al., 2008. A comparative study of survival models for breast cancer prognostication based on microarray data: does a single gene beat them all? Bioinformatics, 24(19), pp.2200-2208.

Haibe-Kains, B. et al., 2010. A fuzzy gene expression-based computational approach improves breast cancer prognostication. Genome biology, 11(2), p.R18.

Madden, S.F. et al., 2013. BreastMark: An Integrated Approach to Mining Publicly Available Transcriptomic Datasets Relating to Breast Cancer Outcome. Breast Cancer Research, 15(4), p.R52.

Fumagalli, D. et al., 2014. Transfer of clinically relevant gene expression signatures in breast cancer: from Affymetrix microarray to Illumina RNA-Sequencing technology. BMC genomics, 15(1), p.1008.

Beck A.H. et al., 2013. Significance Analysis of Prognostic Signatures. PLoS Computational Biology, 9(1), e1002875.

** As well as other publications **

APOBEC3B expression in breast cancer reflects cellular proliferation, while a deletion polymorphism is associated with immune activation.Cescon DW, Haibe-Kains B, Mak TW. Proc Natl Acad Sci U S A. 2015 Mar 3;112(9):2841-6. doi: 10.1073/pnas.1424869112. Epub 2015 Feb 17. PMID: 25730878

Radovich M. et al., 2014. Characterizing the heterogeneity of triple-negative breast cancers using microdissected normal ductal epithelium and RNA-sequencing. Breast cancer research and treatment, 143(1), pp.57-68.

Tramm T. et al., 2014. Relationship between the prognostic and predictive value of the intrinsic subtypes and a validated gene profile predictive of loco-regional control and benefit from post-mastectomy radiotherapy in patients with high-risk breast cancer. Acta Oncologica 53(10), pp.1337-1346.

Doan, T.B. et al., 2014. Breast cancer prognosis predicted by nuclear receptor-coregulator networks. Molecular oncology 8(5), pp.998-1013.

5 Session Info

```
## \begin{itemize}\raggedright
##   \item R version 3.2.0 Patched (2015-05-20 r68389), \verb|x86_64-apple-darwin10.8.0|
##   \item Locale: \verb|en_CA.UTF-8/en_CA.UTF-8/en_CA.UTF-8/C/en_CA.UTF-8/en_CA.UTF-8|
##   \item Base packages: base, datasets, graphics, grDevices, grid,
##     methods, parallel, stats, utils
##   \item Other packages: Biobase~2.28.0, BiocGenerics~0.14.0,
##     biomaRt~2.24.0, breastCancerMAINZ~1.6.0,
##     breastCancerNKI~1.6.0, breastCancerTRANSBIG~1.6.0,
##     breastCancerUNT~1.6.0, breastCancerUPP~1.6.0, genefu~1.18.0,
##     knitr~1.10.5, mclust~5.0.1, prodlim~1.5.1, rmeta~2.16,
##     survcomp~1.18.0, survival~2.38-1, xtable~1.7-4
##   \item Loaded via a namespace (and not attached): amap~0.8-14,
##     AnnotationDbi~1.30.1, bitops~1.0-6, bootstrap~2015.2,
##     DBI~0.3.1, evaluate~0.7, formatR~1.2, GenomeInfoDb~1.4.0,
##     highr~0.5, IRanges~2.2.2, KernSmooth~2.23-14, lava~1.4.0,
##     magrittr~1.5, RCurl~1.95-4.6, RSQLite~1.0.0, S4Vectors~0.6.0,
##     splines~3.2.0, stats4~3.2.0, stringi~0.4-1, stringr~1.0.0,
##     SuppDists~1.1-9.1, survivalROC~1.0.3, tools~3.2.0,
##     XML~3.98-1.1
## \end{itemize}
```