PRECISE:
Personalized Integrated Network Modeling

Min Jin Ha

July 18, 2018

1 Overview

```
> library(PRECISE)
```

This vignette describes how to use `R/PRECISE` to estimate cancer-specific integrated networks, infer patient-specific networks and elicit interpretable pathway-level signatures.

2 An example: TCGA KIRC data

We illustrate the usage of `PRECISE` package using TCGA KIRC data. We require two types of input data: (1) data- proteomic data and other (upstream) molecular profiles (e.g., gene expression, microRNA expression, DNA methylation) and (2) prior information- known pathway annotations and PPIs.

2.1 Input Data

We consider 12 key signaling pathways for TCGA RPPA data (Akbani, R. et al., *Nature communications*, 2014).

```
> pw.array = c("Apoptosis","Breast reactive","Cell cycle","Core reactive"
+              ,"DNA damage response","EMT","PI3K/AKT","RAS/MAPK","RTK"
+              ,"TSC/mTOR","Hormone receptor","Hormone signaling (Breast)")
> length(pw.array)
```

```
[1] 12
```

We take "Apoptosis" pathway as an example and the RPPA data, "rppadat" is a list file that includes the protein expression (RPPA) data for the TCGA KIRC patients in the order of the "pw.array".

```
> data(rppadat)
> length(rppadat)
```

```
[1] 12
```

```
> RPPAdat = rppadat[[1]] # RPPA data for Apoptosis pathway
> dim(RPPAdat)
```

```
[1] 469   9
```

The RPPA data for apoptosis pathway includes 9 genes for 469 TCGA KIRC patients.

2.1.1 Prior calibration and causal structure learning

To decide prior probabilities for protein regulators, we estimate the cancer-specific protein (weighted) causal network that determines the protein regulators by the directions of the edges and the contribution to the target proteins by the the weights of the edges. Using RPPA data in each cancer type and each pathway, we construct the weighted causal structure of proteins using the PC algorithm (Kalisch, M. & Buhlmann, P. *J Mach Learn Res*, 2007) which relies on conditional independence tests with a certain significance level α. The causal structure is represented by a graph with directed ($\rightarrow$ or $\leftarrow$) or bi-directed ($\leftrightarrow$) edges. A bi-directed edge means that the direction is not identifiable from the data. In this analysis, we chose a relatively large vaule of the tuning parameter, so that the resulting graph would include a large number of edges and the confidence of those edges are measured by the weights obtained from investigating the stability under a subsampleing procedure (Meinshausen, N. & Buhlmann, P. *J R Stat Soc B* (2010)).

```
> p = ncol(RPPAdat)
> B=100
> alpha = 0.1
> alpha.array = seq(0.0001,0.1,length=100)
> #pcfit = fitPC(dat = RPPAdat,alpha=alpha,stable=TRUE
> #,alpha.array=alpha.array,B=B,labels=as.character(1:p),verbose=T)
> data(pcfit)
> pcfit =  pcfit.list[[1]] # the fitPC result for Apoptosis pathway
> names(pcfit)
```

```
[1] "fit"      "pcA"      "pcSel"    "pcmaxSel" "rownames"
```

The "fit" object includes the PC-algorithm result from the original protein data using $\alpha = 0.1$ and "pcmaxSel" includes maximum selection probabilities for all three possible directions, $A \rightarrow B$, $A \leftarrow B$, and $A \leftrightarrow B$ and the "rownames" object includes all edge names with A in the first column and B in the second column. Then we make the results from the fitPC function into weighted adjacency matrix as follows.

```
> adj.pc = as(pcfit$fit@graph,"matrix")
> addr = matrix(as.numeric(pcfit$rownames),ncol=2)
```

```
> addr.rev = cbind(addr[,2],addr[,1])
> adj.pc[addr] = adj.pc[addr] * pcfit$pcmaxSel[,1]
> adj.pc[addr.rev] = adj.pc[addr.rev] * pcfit$pcmaxSel[,2]
> adj.pc
```

```
  1    2    3 4    5 6    7    8 9
1 0 0.00 0.76 0 0.00 0 0.00 0.89 0
2 0 0.00 0.00 0 0.00 0 0.00 1.00 0
3 0 0.00 0.00 0 0.42 0 0.64 0.48 0
4 0 0.00 0.00 0 0.33 0 0.73 0.91 0
5 0 0.00 0.00 0 0.00 0 0.00 0.00 0
6 0 0.44 0.40 0 0.00 0 0.00 0.00 0
7 0 0.00 0.00 0 0.00 0 0.00 0.00 0
8 0 0.00 0.00 0 0.00 0 0.00 0.00 0
9 0 0.00 0.00 0 0.00 0 0.00 0.96 0
```

For example, we have an edge node 6 $\rightarrow$ node 2 with the weight 0.44. We use this weighted causal structure to determine the priors for the regression in combination with the known PPIs (`http://string-db.org`).

```
> data(ppi)
> names(ppi)
```

```
[1] "pathwaydat" "ppi"
```

The pathwaydat object includes gene/protein membership of the 12 pathways and ppi object includes the weights for PPIs from the String DB in each pathway. The existing PPI scores are similarly transformed to weighted adjacency matrix and combined with that from the denovo causal structure.

```
> pw="Apoptosis"
> genelist = strsplit(colnames(RPPAdat),split=", ")
> membership = rep(1:p,lapply(genelist,length))
> indigenes = unlist(genelist)
> stringscore = ppi$ppi[[which(names(ppi$ppi) == pw)]]
> # select edges included in the RPPAdat
> stringscore = stringscore[rowSums(cbind(as.numeric(stringscore[,1] %in% indigenes)
+                     ,as.numeric(stringscore[,2] %in% indigenes))) == 2,]
> address = cbind(match(stringscore[,1],indigenes),match(stringscore[,2],indigenes))
> adj.string = matrix(0,ncol=length(indigenes),nrow=length(indigenes))
> adj.string[address] = as.numeric(stringscore[,3])/1000
> adj.string.cl = matrix(0,nrow=max(membership),ncol=max(membership))
> id = 1:length(membership)
> w.off = rbind(which(upper.tri(adj.string.cl),arr.ind=T)
+               ,which(lower.tri(adj.string.cl),arr.ind=T))
> v.off = c(which(upper.tri(adj.string.cl),arr.ind=F)
+           ,which(lower.tri(adj.string.cl),arr.ind=F))
```

```
> for (i in 1:nrow(w.off)) {
+   addr1 = id[membership == w.off[i,1]]
+   addr2 = id[membership == w.off[i,2]]
+   adj.string.cl[v.off[i]] = mean(adj.string[addr1,addr2])
+ }
> Gmat = adj.string.cl/2 + adj.pc/2 ### Make averages
```

2.1.2 Upstream Data

The upstream data, gene expression, miRNA expression, and DNA methylation data are downloaded.

```
> data(mRNA)
> data(miRNA)
> data(Methylation)
> dim(mRNA$Data)

[1] 174 454

> dim(miRNA$Data)

[1] 296 454

> dim(Methylation$Data)

[1] 324 454

> head(mRNA$Des)

     GeneSymbol EntrezID
[1,] "YWHAE"    "7531"
[2,] "EIF4EBP1" "1978"
[3,] "TP53BP1"  "7158"
[4,] "ACACA"    "31"
[5,] "ACACB"    "32"
[6,] "AKT1"     "207"

> head(miRNA$Des)

[1] "hsa-mir-185"  "hsa-mir-586"  "hsa-mir-519d"
[4] "hsa-mir-520h" "hsa-mir-17"   "hsa-mir-106a"

> head(Methylation$Des)

     REF          GeneSymbol ChromosomeID CoordinateID
[1,] "cg21137823" "PRKCZ"    "1"          "1981270"
[2,] "cg25007680" "PARK7"    "1"          "8021821"
[3,] "cg19560758" "ERRFI1"   "1"          "8086721"
[4,] "cg04508649" "MTOR"     "1"          "11249046"
[5,] "cg21223353" "MTOR"     "1"          "11249539"
[6,] "cg07029998" "MTOR"     "1"          "11322191"
```

Each of the data are obtained from TCGA Assembler (Zhu, Y. et al., *Nature methods*, (2014)) includes Data and Des objects for data matrix (no. of features $\times$ no. of samples) and the feature information, respectively. The Des will be used for making covariate data for each of the 9 proteins by matching at the gene-level.

2.2 Step 1: Bayesian estimation of integrated cancer-specific networks

We aim to estimate integrated cancer-specific networks using Bayesian regression methods on each of the proteins with other upstream molecular profiling data. Before performing regressions for all proteins, we need to construct design matrices for each proteins by matching the samples and the features at the gene-level across all data types. Matching samples are performed as follows.

```
> data(rppasample) ## sample names for the KIRC RPPAdat
> covsample = colnames(mRNA$Data)
> intsample = intersect(covsample,rppasample)
> RPPAdat = RPPAdat[match(intsample,rppasample),]
> mRNA$Data = mRNA$Data[,match(intsample,covsample)]
> miRNA$Data = miRNA$Data[,match(intsample,covsample)]
> Methylation$Data = Methylation$Data[,match(intsample,covsample)]
```

To make the covariate data matrices, we match the features from miRNA and methylation using the following annotation files.

```
> data(anno.miRNA)
> miRNAname =  gsub("mir","miR",miRNA$Des)
> anno.miRNA = anno.miRNA[anno.miRNA[,1]%in%miRNAname,c(7,1)]
> ## reduce the annotation file with microRNAs in the dataset
> anno.miRNA[,2] = gsub("miR","mir",anno.miRNA[,2])
> data(anno.methyl)
```

We obtain RPPA response vectors and those corresponding covariate matrices from mRNA, miRNA, methylation data and protein regulators from the prior calibration in Section 2.1.1. Note that the gene expression data are decomposed into two parts modulated by DNA methylation and independent of DNA methylation. Using the annotation files, upstream data and the prior weighted causal structure for the proteins, we construct scaled response vectors and the corresponding scaled covariate matrices.

```
> rownames(RPPAdat) = intsample
> if (!is.null(miRNA$Des))miRNA$Des = as.matrix(miRNA$Des)
> dat = getregcovDat(Gmat = Gmat,RPPAdat=RPPAdat,mRNA=mRNA,miRNA=miRNA
+           ,Methylation=Methylation,anno.miRNA=anno.miRNA,anno.methyl=anno.methyl)
> names(dat)

[1] "ylist" "Xlist"
```

```
> length(dat$ylist)

[1] 9

> length(dat$Xlist)

[1] 9
```

Using the input response vectors and covariate matrices for the 9 proteins in the Apoptosis pathway and the prior weighted protein causal network, we perform Bayesian model averaging for each of the proteins based on linear regression models with Zellner's g-prior on the regression coefficients and compute predictive densities for each sample.

```
> bmsfit= getBMS(dat,Gmat)

> names(bmsfit)

[1] "outlist" "pdlist"
```

For each regression of a protein, its integrated cancer-specific regulators that have edges directed toward the protein are defined by the proteins or other upstream covariates that have a posterior inclusion probability greater than 0.5 (median probability model). Therefore, among proteins, the network includes both directed regulatory edges - when regulator proteins of a protein are not targets of the protein and correlative edges- where both proteins in a link are regulators and targets. The KIRC-specific integrative Apoptosis network is obtained by:

```
> nodes = names(dat$ylist)
> netfit = getPosteriors(bmsfit$outlist,nodes)
> names(netfit)

[1] "G"        "intGlist"
```

The protein network adjusted by upstream covariates is as follows.

```
> netfit$G>0.5 # median probability model

          BAK1   BAX   BID BCL2L11 CASP7   BAD  BCL2 BCL2L1
BAK1    FALSE FALSE  TRUE   FALSE FALSE FALSE FALSE   TRUE
BAX     FALSE FALSE  TRUE   FALSE FALSE  TRUE  TRUE   TRUE
BID      TRUE  TRUE FALSE   FALSE FALSE  TRUE  TRUE   TRUE
BCL2L11 FALSE FALSE FALSE   FALSE FALSE FALSE  TRUE   TRUE
CASP7   FALSE FALSE  TRUE   FALSE FALSE FALSE FALSE  FALSE
BAD     FALSE  TRUE  TRUE   FALSE FALSE FALSE FALSE  FALSE
BCL2    FALSE  TRUE  TRUE    TRUE FALSE FALSE FALSE  FALSE
BCL2L1   TRUE  TRUE  TRUE    TRUE  TRUE FALSE FALSE  FALSE
BIRC2    TRUE FALSE FALSE   FALSE FALSE FALSE FALSE   TRUE
```

```
        BIRC2
BAK1     TRUE
BAX      TRUE
BID     FALSE
BCL2L11 FALSE
CASP7   FALSE
BAD     FALSE
BCL2    FALSE
BCL2L1   TRUE
BIRC2   FALSE
```

The posterior inclusion probabilities (PIPs) for upstream covariates are displayed

```
> length(netfit$intGlist)

[1] 9

> nodes[1]

[1] "BAK1"

> netfit$intGlist[[1]] # the integrative network for nodes[1]

                   type    gene
miRNA_hsa-mir-451 "miRNA" "hsa-mir-451"
miRNA_hsa-mir-363 "miRNA" "hsa-mir-363"
ME_BAK1           "ME"    "BAK1"
NME_BAK1          "NME"   "BAK1"
                   pip
miRNA_hsa-mir-451 "0.453115618707226"
miRNA_hsa-mir-363 "0.0388620233601076"
ME_BAK1           "0.051000260789684"
NME_BAK1          "0.0771015898285164"
```

, where ME stands for gene expression modulated by DNA methylation and NME is for gene expression independent of DNA methylation. Because all the PIPs are less than 0.5, the posteior network include no upstream covariates for BAK1 gene in the KIRC-specific integrative Apoptosis network.

2.3 Step 2: Constructing PRECISE (patient-specific) networks

A PRECISE network is the integrated cancer-specific network with patient-specific labels on the nodes (proteins). Specifically, the activation statuses of the nodes are evaluated by estimating the posteior predictive density of each protein for each patient. To determine the activation status of a protein for a patient , we computed the posterior probabilities of the protein to lie in the δ-interval

around zero, to be greater than, or less than. Then, we decided whether a protein is neutral, activated, or suppressed, depending on the maximum of the three posterior probabilities. Thus, patients with the same tumor type have different node labels, suppressed, neutral or activated while the structure of the networks is the same.

```
> delta = 0.5
> psNet = getPRECISE(bmsfit$outlist,bmsfit$pdlist,nodes,delta)

> names(psNet)

[1] "net"        "net.status" "score.mat"  "samplename"
```

The patient-specific node labels for the 9 proteins in the Apoptosis pathway are stored in the "net" object with -1 (suppressed), 1 (activated) or 0 (neutral).

```
> dim(psNet$net)

[1] 454   9

> head(psNet$net)

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,]    0   -1    1    0    0   -1    0    0    0
[2,]    0    0    1   -1    0    0   -1    0    0
[3,]   -1   -1    0    0    0    0    0   -1    0
[4,]   -1    0    0    0    0    0    1   -1    0
[5,]    0    0    1   -1    0    1    0    1    0
[6,]    0    0    0    0    0    0    1    0    0
```

2.4 Step 3: Calibrating PRECISE (patient-specific) scores

To compute pathway activity scores for each patient, we derive summary measures from the PRECISE networks obtained from the Step 2, indicating that the entire pathway is suppressed, neutral or activated. These patient-specifc pathway scores are weighted averages of the posterior probabilities for suppressed, neutral, and activated statuses of proteins by the number of target or correlative proteins. Therefore, hub proteins in the pathway, that excercise more control over the network through higher target or correlative proteins are given higher weights towards determining the cumulative network score.

```
> dim(psNet$score.mat)

[1] 454   3

> head(psNet$score.mat)
```

```
  Netscore.pos Netscore.neg Netscore.neu
1    10.566950    11.403292     16.02976
2    11.926841    12.704271     13.36889
3     6.417883    17.548730     14.03339
4     8.859715    12.678395     16.46189
5    14.990619     6.416420     12.59296
6    10.927817     9.869793     17.20239
```

The score.mat object displays the three types of network scores for activated (first column), suppressed (second column) and neutral (third column). For a given pathway and each patient, the PRECISE pathway status- which indicates that the pathway is suppressed, neutral, or activated for the patient, can be decided by the statuses, activated, suppressed, or netrual that have the maximum of the three types of pathway scores.

```
> length(psNet$net.status)

[1] 454

> head(psNet$net.status)

[1]  0  0 -1  0  1  0

> head(apply(psNet$score.mat,1,which.max))

[1] 3 3 2 3 1 3
```