An Introduction to the '`pathclust`' Package, Version 1.0

Zequn Sun,Wei Wei and Dongjun Chung
Department of Public Health Sciences, Medical University of South Carolina (MUSC),
Charleston, SC, 29425.

May 26, 2017

1 Overview

This vignette provides basic information about the `pathclust` package for the pathway-guided identification of cancer subtypes [1]. The proposed approach improves identification of molecularly-defined subgroups of cancer patients by utilizing information from pathway databases in the following four aspects.

(1) Integration of genomic data at the pathway-level improves robustness and stability in identification of cancer subgroups and driver molecular features;

(2) Summarizing multiple genes and genomic platforms at the pathway-level can potentially improve statistical power to identify important driver pathways because moderate signals in multiple genes can be aggregated;

(3) In `pathclust`, we consider the "cooperation" or "interaction" between pathways, instead assuming that each pathway operates independently during the cancer progression, which may be unrealistic;

(4) `pathclust` allows simultaneous inference in multiple biological layers (pathway clusters, pathways, and genes) within a statistically rigorous and unified framework without any additional laborious downstream analysis.

The package can be loaded with the command:

```
R> library("pathclust")
```

2 Input Data

The package requires that the response consist of 4 components: (1) gene expression z-scores in the form of a either data frame or matrix; (2) survival time and censoring indicator in the form of vectors; (3) pathway information in the form of a list, where each element is a vector of the names of gene belonging to the pathway.

In this vignette, a small subset of the Cancer Genome Atlas (TCGA) data is used to illustrate the '`pathclust`' package. Specifically, we consider z-scores for the mRNA expression data of 389 genes for 50 randomly selected high-grade serous ovarian cancer patients, along with their survival times and censoring statuses. This dataset is included as an example data '`TCGA`' in the '`pathclust`' package. This TCGA data was originally downloaded from the cBio Portal (http://www.cbioportal.org/) using the R package '`cgdsr`' and here we consider z-scores for the mRNA expression data. The '`TCGA`' is a list object with four elements, including the '`geneexpr`' data frame of z-scores for the mRNA expression, the '`t`' vector of the survival time, the '`d`' vector of

the censoring status indicator, and the '`pathList`' list of the pathway information. The '`pathList`' has four elements, each of which contains names of genes belonging to each pathway.

This dataset can be loaded as follows:

```
R> data(TCGA)
R> TCGA$geneexpr[1:5,1:5]

        ACLY        ACO1       ACO2         CS       DLAT
1 -2.2410125 -0.48445531 -1.6346455  0.1378804 -3.5310321
2 -2.1301362  0.82116427 -0.9533701  0.6213512  0.6689948
3 -2.9122727 -0.08790649 -1.0975096 -0.2454025 -0.9433900
4 -1.1721514 -0.24249825 -0.7212639  0.1842386 -0.6188785
5  0.5383438  0.98012739 -0.7396043 -0.0699680  1.9573767

R> TCGA$t[1:5]

[1] 43.89 40.97 49.12  2.00 46.59

R> TCGA$d[1:5]

[1] 1 1 0 1 0

R> TCGA$pathList[1]

$KEGG_CITRATE_CYCLE_TCA_CYCLE
 [1] "IDH3B"     "DLST"      "PCK2"      "CS"        "PDHB"      "PCK1"
 [7] "PDHA1"     "LOC642502" "PDHA2"     "LOC283398" "FH"        "SDHD"
[13] "OGDH"      "SDHB"      "IDH3A"     "SDHC"      "IDH2"      "IDH1"
[19] "ACO1"      "ACLY"      "MDH2"      "DLD"       "MDH1"      "DLAT"
[25] "OGDHL"     "PC"        "SDHA"      "SUCLG1"    "SUCLA2"    "SUCLG2"
[31] "IDH3G"     "ACO2"
```

3 Pre-filtering

To refine the candidate set of genes, we first conduct a supervised pre-filtering by fitting a Cox regression model of the mRNA expression measure of each gene on the patient survival. Only the gene expressions associated with patient survival at p-values smaller than a pre-specified cut-off are included in the subsequent analysis. By default, p = 0.5 is used as cut-off point.

```
R> prefilter.results <- prefilter( data=TCGA$geneexpr, time=TCGA$t, status=TCGA$d, plist=TCGA$p
R> prefilter.results

Summary: Pre-filtering results (class: Prefiltered)
--------------------------------------------------
Number of genes before prefiltering: 389
Number of genes after prefiltering: 213
--------------------------------------------------
```

4 Gene Selection

In order to select key genes associated with patient survivals and effectively summarize them by taking into account correlation among them, we fit a sparse partial least squares (SPLS) Cox regression model [3] of patient survivals on gene expression measurements for each pathway.

Using the object '`prefilter.results`', gene-level analysis result can be generated with '`selectGene`' function as follows.

```
R> gene.results <- selectGene(prefilter.results)

R> gene.results

Summary: Gene-level analysis results (class: FitGene)
--------------------------------------------------
Number of prefiltered genes: 213
Number of selected genes: 132
--------------------------------------------------
```

The list of the SPLS regression coefficients of cancer-related genes can be generated using the function `coef()`.

```
R> head(coef(gene.results)[[1]])

  colnames.xx. spls.mod.betahat
1         ACLY        0.0000000
2           CS        0.0000000
3         DLAT        0.0000000
4          DLD        0.0000000
5         MDH1       -0.3560516
6         PCK1       -0.1988449
```

The function '`selectGene`' has two main tuning parameters: '`eta`' represents the sparsity tuning parameter and '`K`' is the number of hidden (latent) components. Parameters can be chosen by (v-fold) cross-validation. Users can search the range for these parameters and the cross-validation procedure searches within these ranges. Note that '`eta`' should have a value between 0 and 1 while '`K`' is integer-valued and can range between 1 and $min\{p, (v-1)n/v\}$, where p is the number of genes and n is the sample size. For example, if 10-fold cross-validation is used (default), '`K`' should be smaller than $min\{p, 0.9n\}$. For the TCGA data, we set the number of folds as 5, '`K`' as 5, and search the optimal '`eta`' in the range between 0.1 and 0.9.

5 Pathway Selection

Next, in order to identify a parsimonious set of pathways associated with patient survivals, we fit a LASSO-penalized Cox regression [4] on latent components derived from all the pathways. Specifically, a pathway is selected if at least one of its latent components has non-zero LASSO coefficient estimate.

This approach has the following two strengths: First, the latent components generated from the SPLS step preserve pathway structure and also reflect correlation among genes and their association

with survival outcomes. Second, this approach can potentially improve the stability of estimation in the subsequent analysis.

Using the '`gene.results`', pathway-level analysis result can be generated with '`selectPath`' function.

```
R> path.results <- selectPath(gene.results)
R> path.results

Summary: Pathway-level analysis results (class: FitPath)
-------------------------------------------------
Number of all pathways: 4
Number of selected pathways: 4

List of selected pathways:
        KEGG_CITRATE_CYCLE_TCA_CYCLE:
        KEGG_MAPK_SIGNALING_PATHWAY:
        KEGG_TGF_BETA_SIGNALING_PATHWAY:
        KEGG_THYROID_CANCER:
-------------------------------------------------
```

LASSO regression coefficients of cancer-related pathways can be generated using the function `coef()`.

```
R> head(coef(path.results))

              rep.pathways..cols.  path.beta
1    KEGG_CITRATE_CYCLE_TCA_CYCLE 0.44388849
2    KEGG_CITRATE_CYCLE_TCA_CYCLE 0.00000000
3    KEGG_CITRATE_CYCLE_TCA_CYCLE 0.00000000
4    KEGG_CITRATE_CYCLE_TCA_CYCLE 0.00000000
5     KEGG_MAPK_SIGNALING_PATHWAY 0.08378862
6 KEGG_TGF_BETA_SIGNALING_PATHWAY 0.37094205
```

Hazard ratio plot associated with each latent component in the selected pathways can be generated using the function `plot()` with the argument `type="HR"`.

```
R> plot(path.results, type="HR")
```

Figure 1 shows the hazard ratio (HR) associated with each latent component in the pathways selected by the `pathclust`. Based on the TCGA data, pathways with the largest effect on survival (HR ≥ 1.15) are `KEGG_CITRATE_CYCLE_TCA_CYCLE` and `KEGG_TGF_BETA_SIGNALING_PATHWAY` pathways.

6 Risk Group Prediction

Risk group predictions can be made using the function `predict()`

```
R> predicted <- predict(path.results)
```

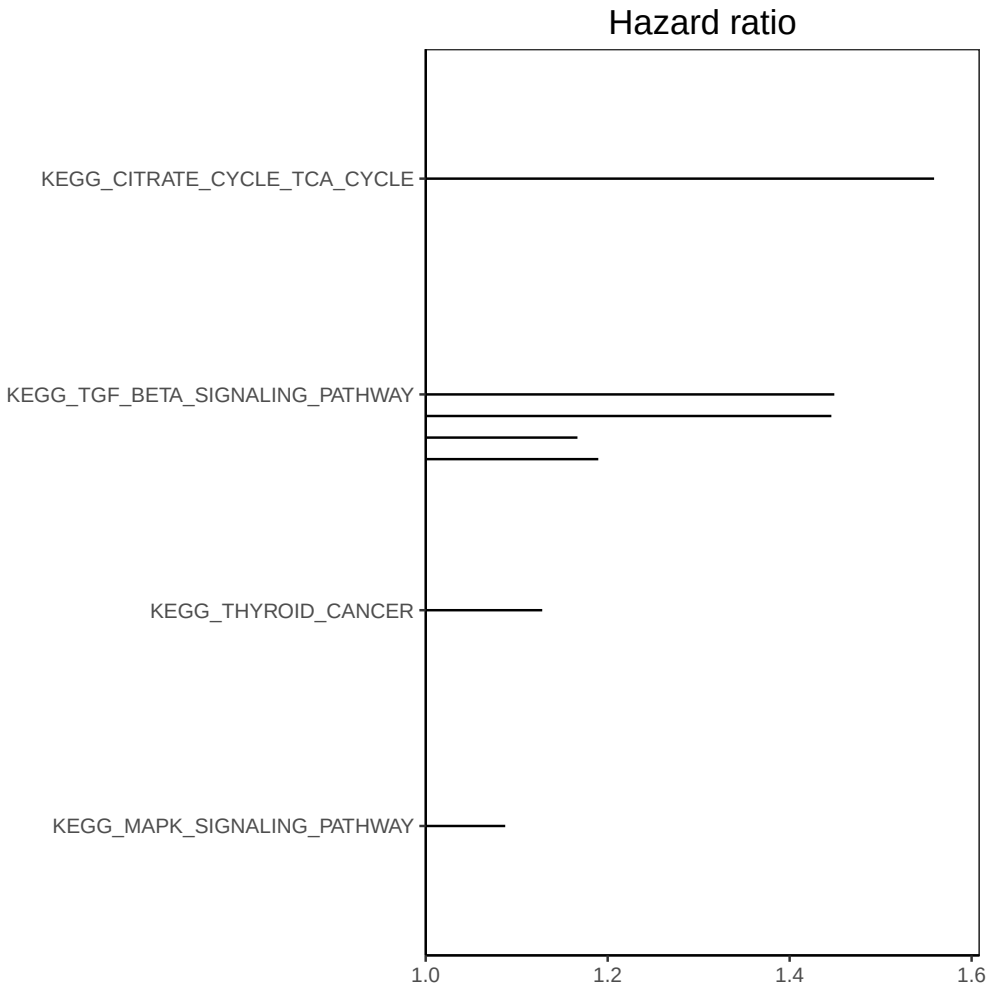

Figure 1: Hazard ratio (HR) associated with each latent component in selected pathways.

The function '`predict`' returns a list with the following three elements: (1) `risk.index`: number of pathways with elevated activity for each patient; (2) `riskcat`: risk group prediction for each patient; (3) `cuts`: cut off to determine low, intermediate and high risk groups.

```
R> predicted

$risk.index
 [1] 3 4 0 4 2 4 4 0 0 1 4 0 2 3 3 3 0 1 0 2 1 0 1 1 2 2 2 0 0 0 2 0 1 0 0 0 1 0
[39] 4 4 2 4 4 4 4 4 3 4 3 3

$riskcat
 [1] "med"  "high" "low"  "high" "med"  "high" "high" "low"  "low"  "med"
[11] "high" "low"  "med"  "med"  "med"  "med"  "low"  "med"  "low"  "med"
[21] "med"  "low"  "med"  "med"  "med"  "med"  "med"  "low"  "low"  "low"
[31] "med"  "low"  "med"  "low"  "low"  "low"  "med"  "low"  "high" "high"
[41] "med"  "high" "high" "high" "high" "high" "med"  "high" "med"  "med"
```

```
$cuts
[1] 0.00 3.75

$time
 [1]  43.89  40.97  49.12   2.00  46.59  18.50  11.86  65.44  63.01  48.72
[11]  21.55  63.93  49.25  32.49  44.29  33.64  76.48  71.42  56.51  19.97
[21]  38.34 118.99  84.14  57.69  35.51   9.95  38.41  89.26  41.56  57.53
[31]  30.46   6.11  35.12 112.29  71.06  63.05  39.72  12.45  25.89  81.80
[41]  18.46  34.17  32.85  14.65   6.67  35.38  19.97  20.89   6.21  24.57

$status
 [1] 1 1 0 1 0 1 1 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0
[39] 1 1 1 1 1 1 1 1 1 1 1 1
```

7 Survival Curve

The predictive performance of `pathclust` method can be presented by Kaplan-Meier curves. Kaplan-Meier curves of predicted patient subgroups can be generated with `plot()` function with argument `type="KM"`.

```
R> plot(path.results, type="KM")
```

Figure 2 shows the Kaplan-Meier curves of predicted patient subgroups and indicates that the `pathclust` approach successfully separates out high, intermediate and low risk groups.

8 Survival ROC

The predictive performance of `pathclust` method can be further evaluated based on area under the time dependent receiver operating curve (ROC). ROC plot can be generated using `plot()` function with argument `type="ROC"`.

```
R> plot(path.results, type="ROC")
```

Figure 3 shows the ROC curves for survival, and for the TCGA data, the area under curve (AUC) associated with the `pathclust` approach was 0.886.

References

[1] Wei W., Zequn S., Willian S., Zhenning Y., Andrew L., Gary H., Linda K., Dongjun C. (2017), "pathclust: Pathway-guided identification of cancer subtypes". (submitted).

[2] Cancer Genome Atlas Research Network (2011), "Integrated genomic analyses of ovarian carcinoma". Nature, 474(7353), 609-615.

[3] Bastien, P., Bertrand, F., Meyer, N., Maumy-Bertrand, M. (2014), " Deviance residuals-based sparse PLS and sparse kernel PLS regression for censored data". Bioinformatics, 31(3), 397-404.

[4] Tibshirani, R. (1997), "The lasso method for variable selection in the cox model". Statistics in Medicine, 16(4), 385-395.

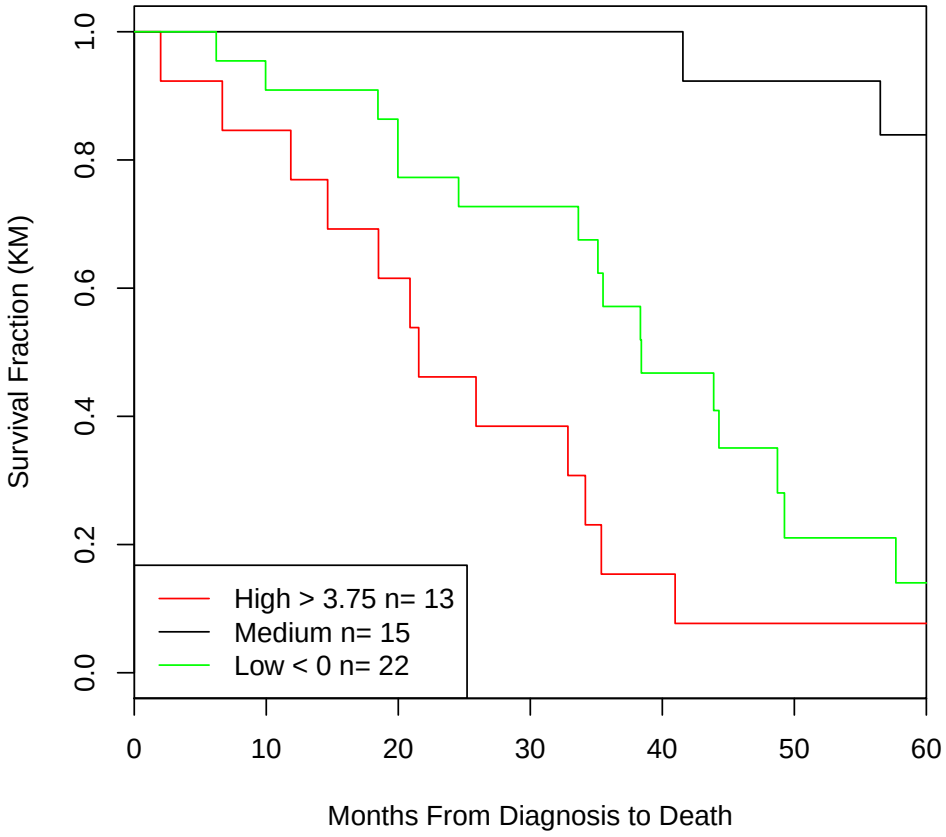

Figure 2: The observed survival curves for patient subgroups.

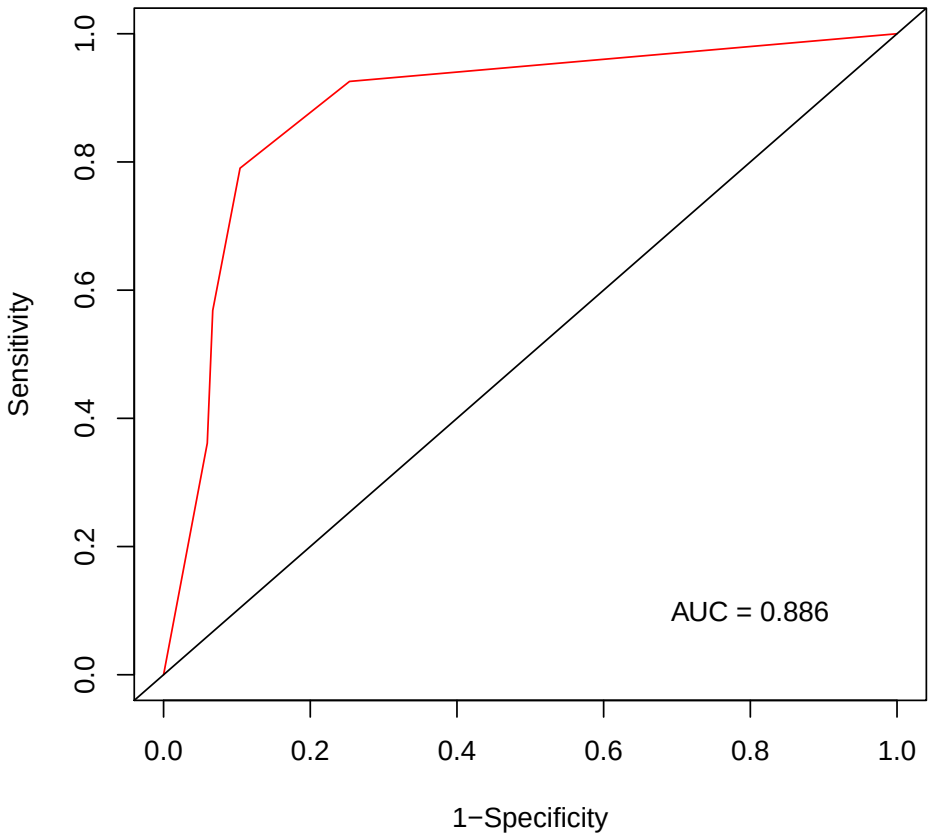

Figure 3: Time dependent receiver operating curve