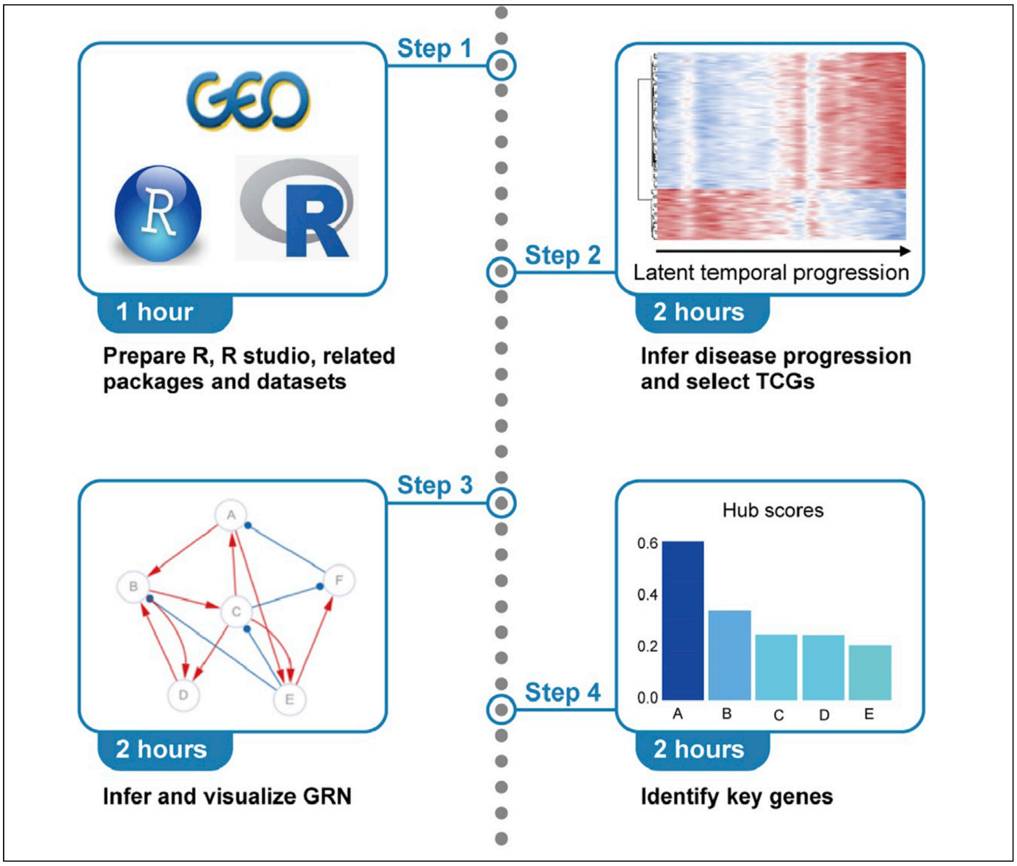
STAR Protocols

CellPress
OPEN ACCESS

Protocol

Inferring disease progression and gene regulatory networks from clinical transcriptomic data using PROB_R



Due to a lack of explicit temporal information, it can be challenging to infer gene regulatory networks from clinical transcriptomic data. Here, we describe the protocol of PROB_R for inferring latent temporal disease progression and reconstructing gene regulatory networks from cross-sectional clinical transcriptomic data. We illustrate the protocol by applying it to a breast cancer dataset to demonstrate its use in recovering pseudo-temporal dynamics of gene expression alongside disease progression, reconstructing gene regulatory networks, and identifying key regulatory genes.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Zhaorui Dong,
Xiaoqiang Sun

sunxq6@mail.sysu.edu.cn,
xiaoqiangsun88@gmail.com

Highlights

Perform downstream analysis of cross-sectional transcriptomic data

Infer temporal progression of disease with a graph-based random walk protocol

Infer causal gene regulatory networks with an ODE-Bayesian-Lasso protocol

Identify key genes from the reconstructed network for further analysis

Dong & Sun, STAR Protocols 3, 101467
September 16, 2022 © 2022 The Authors.
https://doi.org/10.1016/j.xpro.2022.101467

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Inferring disease progression and gene regulatory networks from clinical transcriptomic data using PROB_R

Zhaorui Dong[1] and Xiaoqiang Sun[1,2,3,*]

[1]School of Mathematics, Sun Yat-sen University, Guangzhou 510275, China

[2]Technical contact

[3]Lead contact

*Correspondence: sunxq6@mail.sysu.edu.cn or xiaoqiangsun88@gmail.com

https://doi.org/10.1016/j.xpro.2022.101467

SUMMARY

Due to a lack of explicit temporal information, it can be challenging to infer gene regulatory networks from clinical transcriptomic data. Here, we describe the protocol of PROB_R for inferring latent temporal disease progression and reconstructing gene regulatory networks from cross-sectional clinical transcriptomic data. We illustrate the protocol by applying it to a breast cancer dataset to demonstrate its use in recovering pseudo-temporal dynamics of gene expression alongside disease progression, reconstructing gene regulatory networks, and identifying key regulatory genes.

For complete details on the use and execution of this protocol, please refer to Sun et al. (2021).

BEFORE YOU BEGIN

Download R, RStudio, and related packages

Timing: 1 h

1. This protocol uses R language for all codes. R is a free environment for statistical computing and drawing graphics. R studio (version 1.4.1106) is used as the IDE for R language. The related packages can be downloaded by using `install.packages()` function in R environment. PROB_R is a collection of R functions. The codes containing required functions to run PROB_R can be downloaded from GitHub: https://github.com/SunXQlab/PROB_R or Zenodo: https://doi.org/10.5281/zenodo.6555525. All codes are run under Windows system. We suggest the users to download and install Cytoscape (a software for network analysis and visualization) for better network visualization in step 3.

Select and prepare datasets

Timing: 1 h

2. The PROB_R is designed for analysis of cross-sectional transcriptomic data. In the following steps, we use a microarray dataset of breast cancer samples that was deposited in the Gene Expression Omnibus (GEO) database (https://www.ncbi.nlm.nih.gov/gds) as a case study for illustration purpose. The accession number to this dataset is GEO: GSE7390.

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Table 1. Example format for the clinical transcriptomic data

Genes/Grades	GSM177885	GSM177886	GSM177887	GSM177888	…
A1CF	7.422311	7.149458	6.974534	8.001110	…
A2M	11.029381	11.564107	13.150140	12.194598	…
…	…	…	…	…	…
ZZZ3	10.420448	8.733127	9.8648944	8.7696577	…
Grade information	3	3	3	3	…

Note: This dataset can be downloaded by using `PROB_GEOinstall()` function and saved as "Gene_GSE7390.csv". When using `PROB_GEOinstall()` function, necessary data cleaning procedures are also performed. The dataset "Gene_GSE7390.csv" contains gene expression profile of 198 breast cancer samples and the associated clinical information such as grade information. Each column is a breast cancer patient sample and each row is the transcriptomic data of a gene. The last row is the clinical grade information for all samples that is also required for the progression inference. If PROB_R is applied to other cross-sectional transcriptomic datasets, make sure that the data matrix is arranged in the format as illustrated in Table1.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Dataset of microarray experiments from primary breast tumors	NCBI Gene Expression Omnibus	GEO: GSE7390
Software and algorithms		
PROB_R	This paper	https://github.com/SunXQlab/PROB_R; or https://doi.org/10.5281/zenodo.6555525
R software (version 4.0.5)	(R Core Team, 2013)	https://www.r-project.org/
RStudio (version 1.4.1106)	(RStudio Team, 2020)	https://rstudio.com/
Cytoscape	Cytoscape Team	https://cytoscape.org/
Biobase package	(Huber et al., 2015)	https://bioconductor.org/packages/release/bioc/html/Biobase.html
ggplot2 package	(Wickham, 2016)	https://ggplot2.tidyverse.org/
GEOquery package	(Davis and Meltzer, 2007)	https://bioconductor.org/packages/release/bioc/html/GEOquery.html
trend package	(Pohlert, 2020)	https://cran.r-project.org/web/packages/trend/index.html
OmnipathR package	(Türei et al., 2016)	https://bioconductor.org/packages/release/bioc/html/OmnipathR.html
monomvn package	(Gramacy, 2019)	https://cran.r-project.org/web/packages/monomvn/index.html
pheatmap package	(Kolde, 2019)	https://www.rdocumentation.org/packages/pheatmap/versions/1.0.12/topics/pheatmap
minerva package	(Albanese et al., 2012)	https://cran.r-project.org/web/packages/minerva/index.html
tidyr package	(Wickham and Girlich, 2022)	https://github.com/tidyverse/tidyr
gprofiler2 package	(Kolberg et al., 2020)	https://cran.r-project.org/web/packages/gprofiler2/
reshape2 package	(Wickham, 2007)	https://cran.r-project.org/web/packages/reshape2/index.html
igraph package	(Csardi and Nepusz, 2006)	https://igraph.org/r/
survival package	(Therneau, 2022)	https://cran.r-project.org/package=survival
Brq package	(Alhamzawi and Ali, 2020)	https://cran.r-project.org/web/packages/Brq/index.html
Other		
Equipment	A laptop with an Intel Core i7 10th generation 2.60 GHz CPU, 16 GB RAM and 64× Windows 10 system	N/A

STEP-BY-STEP METHOD DETAILS

Infer potential temporal disease progression

Timing: 2 h

This section describes the procedures for inferring potential temporal disease progression from the prepared dataset and selecting temporally changing genes (TCGs) of interest.

1. To begin, download and install required R packages in your R environment. Additionally, you need to download the R files we provide and save them in your current working directory. Then you can load the dataset into Rstudio. The file main.R runs following codes.

```
# Package names
packages=c("trend", "pheatmap", "OmnipathR", "tidyr", "gprofiler2", "minerva", "reshape2",
"ggplot2","Biobase","GEOquery","monomvn","igraph","survival","Brq");
# Install packages not yet installed
installed_packages=packages %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
install.packages(packages[!installed_packages])
}
# make sure that these R files are in your current working directory.
source("PROB_GEOinstall.R");
source("Progression_Inference.R");
source("ODE_Bayesian_Lasso.R");
source("BL_to_csv.R");
source("Locate_Key_Genes.R");
source("KM_analysis.R");
source("Time_course.R");
source("trans_cytoscape.R");
# load GSE7390 dataset
Gene_Data=read.csv("Gene_GSE7390.csv");
row.names(Gene_Data)=Gene_Data[,1];
Gene_Data=Gene_Data[,-1];
Gene_Data=data.frame(Gene_Data);
```

2. To infer potential temporal disease progression from your data, you need to use `Progression_Inference()` function. This function is based on a graph-based random walk method and returns the pseudo progression status for each patient and pseudo time-series of expressions of all genes. An example of pseudo temporal expression is shown for gene MCM10 (Figure 1).

```
PI=Progression_Inference(Gene_Data);

save(PI,file="PI.Rdata");

pseudo_series=PI$Ordered_Data;

pseudo_time=PI$Sampled_Time;

plot(pseudo_time,pseudo_series["MCM10",],type="l",xlab="pseudo time",ylab="MCM10
expression");

# You can plot out pseudo time series of any interested genes by replacing "MCM10" here with
other specified gene names.
```

Note: Progression_Inference() is the function to infer the disease progression. The input object is your cross-sectional transcriptomic data, genes in row and samples in column. The last row is grade information of each sample. The output object contains the following elements.

`Accumulated_Transition_Matrix`: the accumulated transition matrix used in the step of feature extraction with diffusion maps.

`Temporal_Progression`: TPD value of each gene.

`Ordered_Data`: reordered samples against temporal progression.

`Sampled_Time`: sampled time points in the smoothed trajectory.

`Order`: The order of each sample in the inferred pseudo-temporal progression.

3. Based on the expression data reordered by PROB_R, we focus on temporally changing genes (TCGs) that have monotone increasing or decreasing trends. We employ a trend analysis technique based on Mann-Kendall test (Mann, 1945) (Kendall, 1955) to select TCGs. In this case study, we select 100 top TCGs according to the Mann-Kendall test p values.

```
#—Identify TCGs————————————————————————

library(trend)

ngenes=nrow(pseudo_series)-1;

TCG_mark=rep(FALSE,ngenes);

pval_trend=rep(0,ngenes);

for(i in 1:ngenes){

  trial=as.numeric(pseudo_series[i,]);

  res.t=mk.test(trial);

#use Mann-Kendall test to identify temporal trend of each gene

  if(res.t$pvalg<0.05) TCG_mark[i]=TRUE;

  pval_trend[i]=res.t$pvalg;

}

TCGid=order(pval_trend,decreasing=FALSE)[1:100];

save(TCGid,file="TCGid.Rdata");
```

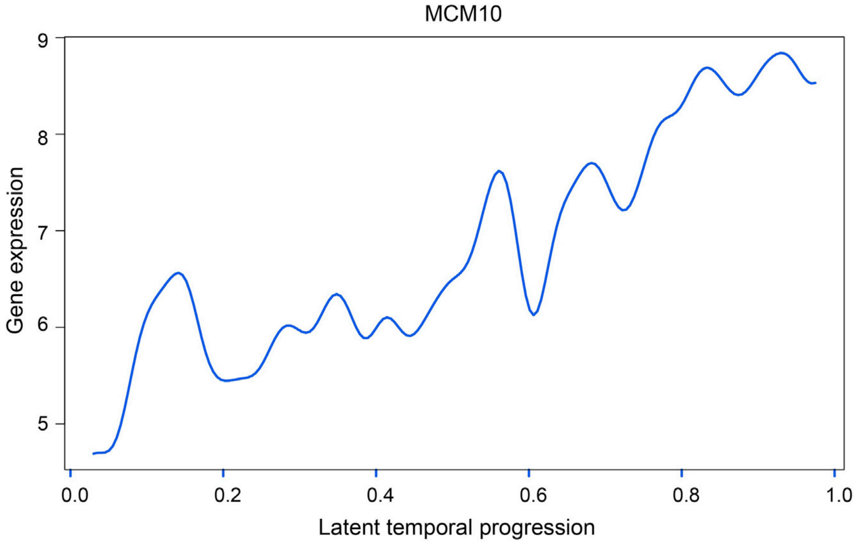


Figure 1. Pseudo temporal dynamics of gene expression along latent disease progression
Shown is an example for the gene MCM10. The values of x-axis and y-axis are standardized.

```
grade2=pseudo_series[ngenes+1,];

TCG_series=rbind(pseudo_series[TCGid,],grade2);

save(TCG_series,file="TCG_series.Rdata");

TCG_names=row.names(pseudo_series)[TCGid];
```

4. You can draw a heatmap to show the expression profile of the selected genes that express with significant trends during breast cancer progression. It shows that the selected TCGs are divided into two groups (temporally upregulated or downregulated alongside the breast cancer progression) (Figure 2).

```
library(pheatmap)

dev.off()

Data=TCG_series[-nrow(TCG_series),];

M=apply(Data,1,max)

A=apply(Data,1,mean)

S=apply(Data,1,sd)

D=(Data-A)/S

dev.new()

pheatmap(D,cluster_row=T, cluster_cols=F, clustering_distance_rows=’euclidean’,clustering_method = "ward.D", color = colorRampPalette(c("CornflowerBlue", "white", "firebrick3"))(200), fontsize=9, fontsize_row=6,labRow=NA, show_colnames = FALSE)
```

Infer GRNs and visualize the network

Timing: 2 h

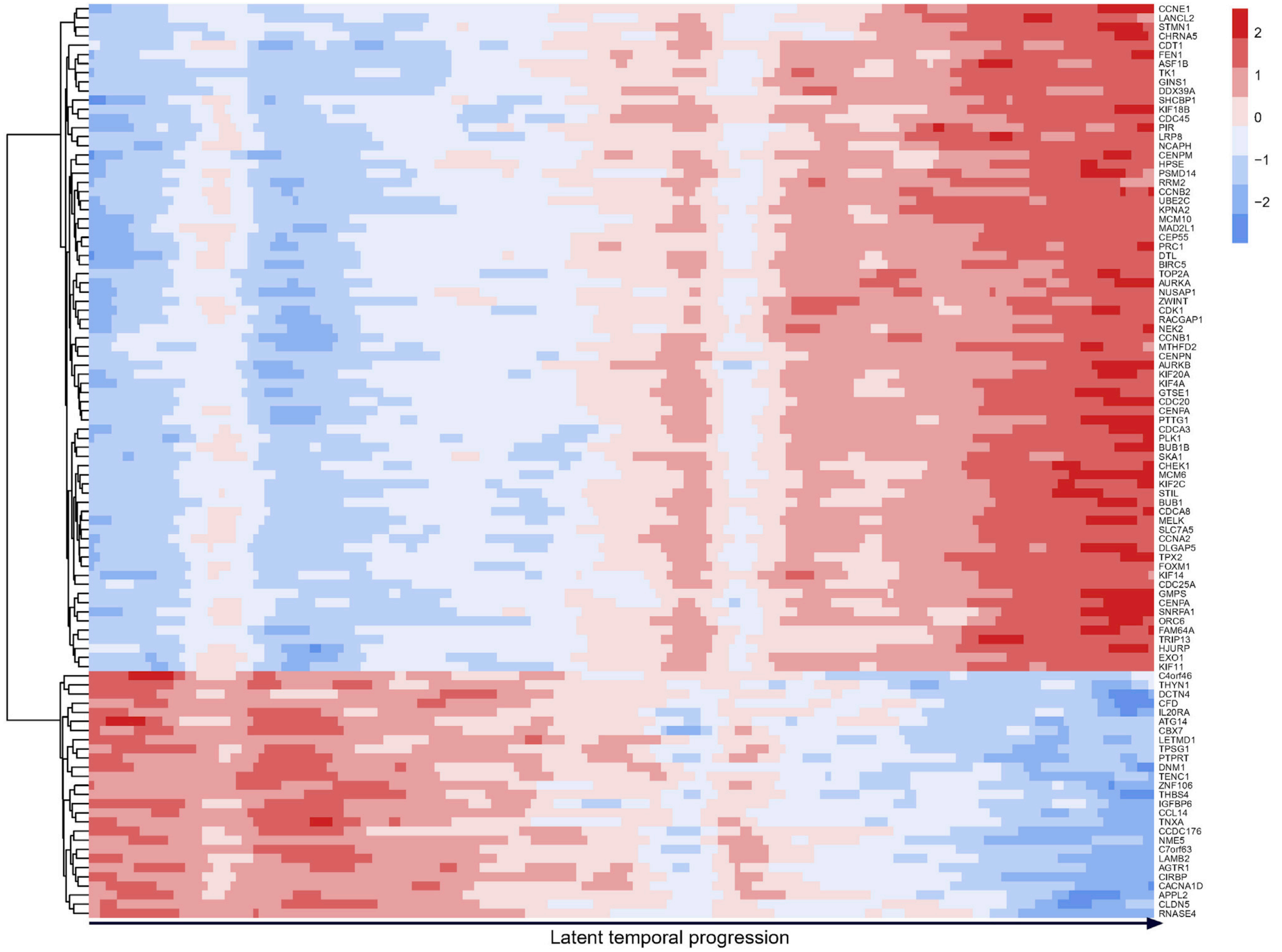


Figure 2. Clustering heatmap of TCGs

The color in each cell represents the standardized expression level of corresponding gene. The left side of the figure shows clustering structure of TCGs.

This section describes the procedures for inferring and visualizing GRN.

5. Specify prior network structure.

Before applying ODE Bayesian Lasso method (Park and Casella, 2008) to infer GRNs between 100 TCGs, we firstly specify a prior network structure to increase accuracy or interpretation of inferred networks. The prior network structure consists of two parts.

   a. The first part of the prior structure comes from possible TCGs interactions provided by OmniPath. OmniPath is a large integrated resource of prior knowledge of molecular regulatory interactions including protein-protein and gene regulatory interactions, enzyme-PTM relationships, protein complexes, protein annotations and intercellular communication (Türei et al., 2016).

```
library(OmnipathR)

library(tidyr)

library(gprofiler2)

iai_all=import_all_interactions();
```

```
w1_all=which((iai_all$source_genesymbol %in% TCG_names) & (iai_all$target_genesymbol %in% TCG_names));

ntcg=length(TCG_names);

prior_PPI=matrix(rep(0,ntcg*ntcg),nrow=ntcg,ncol=ntcg);

for(i in w1_all){

  sel=iai_all[i,];

prior_PPI[which(TCG_names==sel$target_genesymbol),which(TCG_names==sel$source_genesymbol)]=1;

}
```

b. The second part comes from the highly co-expressed gene pairs based on the assumption that the whole GRN is rather sparse, gene pairs with top 5% mutual information are also included into the prior network structure as additional edges.

```
library(minerva);

Gene_TCGs=t(Gene_Data[TCGid,]);

MI=mine(Gene_TCGs)$MIC;

MI95=quantile(MI,0.95);

prior=(MI>MI95);

for(i in 1:100) prior[i,i]=FALSE;
```

6. Apply `ODE_Bayesian_Lasso()` function on pseudo temporal expression data of selected TCGs to infer the regulatory network by incorporating the prior network information.

```
>set.seed(1);

>Breast_BL=ODE_Bayesian_Lasso(TCG_series,pseudo_time,prior|prior_PPI);
```

Note: `ODE_Bayesian_Lasso()` is a function to infer the GRN with ODE Bayesian Lasso method. The input objects contain your transcriptomic data of selected TCGs, pseudo time samples and your prior network. Elements of the output object are listed below.

`Adjacent_Matrix`: the matrix of the posterior mean of each parameter.

`Adjusted_Adjacent_Matrix`: 0–1 matrix, 1 entry occurs only when edges have 95% or more credible level.

`Presence_Probability`: credible level of each edge.

`Standard_Deviations`: posterior standard deviation of each parameter.

⚠ CRITICAL: Here we use "prior|prior_PPI" to combine two parts of the prior network. If you want to use your own prior network structure, replace "prior|prior_PPI" with a *n* × *n* matrix *A*, here *n* is the number of your TCGs (*n*=100 here). Each element of the matrix should be "TRUE/FALSE" or "1/0". If *A*(*i*,*j*) is TRUE or 1, it means that the edge with the *j*-th TCG as source node and the *i*-th gene as target node is considered as a candidate edge in the GRN.

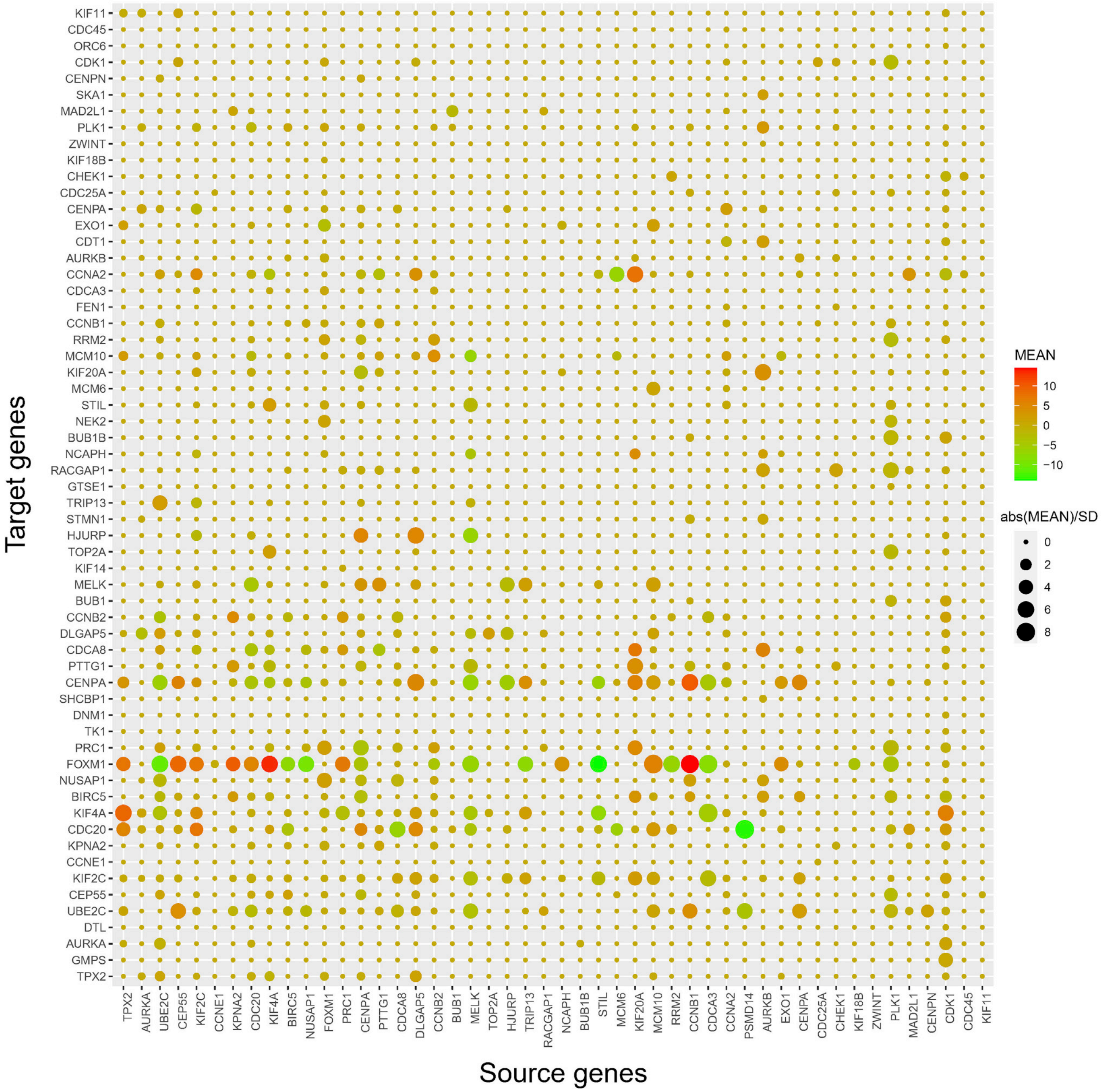


Figure 3. Bubble plot of adjacent matrix of the inferred GRNs

The node color represents the posterior mean of regulatory coefficient for each edge, with red for positive and green for negative. The node size represents the standardized absolute value of the edge coefficient, which is calculated as the absolute posterior mean divided by the standard deviation.

Otherwise, this edge is not considered in the model. (Do not reverse the source and target node!)

7. Use bubble plot to visualize the adjacent matrix of the inferred GRNs (Figure 3).

```
library(reshape2)

library(ggplot2)

BL3=Breast_BL;

am=BL3$Ajacent_Matrix;

rownames(am)=colnames(am)=TCG_names;

sd=BL3$Standard_Deviations;

row_id=(apply(abs(am),1,sum)>1e-5);

col_id=(apply(abs(am),2,sum)>1e-5);

am=am[row_id,col_id];

sd=sd[row_id,col_id];

data_melt=melt(am);

names(data_melt)=c(’Gene1’,’Gene2’,’Value’);

p=ggplot(data_melt,aes(x=Gene2,y=Gene1,size=abs(am)/sd,color=am))+geom_point()+-
theme(axis.text.x = element_text(angle=90,hjust=1))+

  ylab("Target

  genes")+xlab("Genes")+scale_colour_gradient(low="green",high="red");

p
```

8. Choose threshold of credible level of edges to visualize the network. In the returned list of ODE_Bayesian_Lasso() function, the matrix Presence_Probability tells us how credible an edge exists in the GRN. We select a threshold to filter out those edges with low credible level for visualization purpose or for further analysis.
   a. Here we provide a data-driven method to select the threshold of edge credible level. It is based on the assumption that the structures of GRNs tend to follow a scale-free principle. It means that the network degree sequence often follows a power-law distribution (Albert and Barabasi, 2001). For each possible edge threshold (a real number ranging from 0.01 to 0.99), we have a network degree sequence. Then we apply linear regression with a logarithm link function on the empirical density of the degree sequence and use R-square statistic for evaluation. If an edge threshold corresponds to a higher R-square value, it means that this network is more inline with scale-free feature. The code for calculating the R-square is as follows.

```
library(igraph)

BL3=Breast_BL;

cah=(1:99)/100;

power=rep(0,99);

r2=rep(0,99);

for(i in 1:99){

  alpha=cah[i];

  BL3graph=graph_from_adjacency_matrix(BL3$Presence_Probability>alpha);

  degs=degree.distribution(BL3graph);

  degs=degs[-1];

  x=(1:length(degs));
  no0=(degs>0);
  x=x[no0];y=-log(degs[no0]);
  fit=lm(y~x);
  power[i]=fit$coefficients[2];
  r2[i]=cor(x,y)^2;
}
```

b. The edge threshold with highest R-square value is determined as follows:

```
> which(r2==max(r2))/100
> 0.93
```

c. Similarly, if you want to visualize the network based on the scale-free edge threshold, you can use `BL_to_csv()` function to convert the network data to a .csv format file for Cytoscape input.

```
> BL_to_csv(Breast_BL,0.93,’’GSE7390_GRNs.csv’’,TCG_names);
```

Note: In the original paper of PROB (Sun et al., 2021), the threshold is chosen as 0.95, which means that only edges with more than 95% credible level are left in the final network. If you want to follow the original paper, you can use the following code:

```
>BL_to_csv(Breast_BL,0.95,"GSE7390_GRNs.csv",TCG_names);
```

d. Then you can upload this “GSE7390_GRNs.csv” file into Cytoscape to visualize the network. You can choose the layout and edit style of your network in Cytoscape. In this example we use a circular layout. For each edge, red color represents positive regulation and blue color represents negative regulation (Figure 4).

Identify key genes in disease progression

⏱ Timing: 2 h

This section describes the procedures for identifying key genes within the inferred GRN.

9. Identify key genes with eigenvector centrality measure.
   a. use `Locate_Key_Genes()` function to calculate hub scores of each gene in the GRN. We draw a barplot to assist the identification of the key genes with highest hub scores (Figure 5).

```
Eig_scores=Locate_Key_Genes(Breast_BL,TCG_names);
cut=5;
library(ggplot2)
trt=names(Eig_scores)[1:cut];
outcome=Eig_scores[1:cut];
```

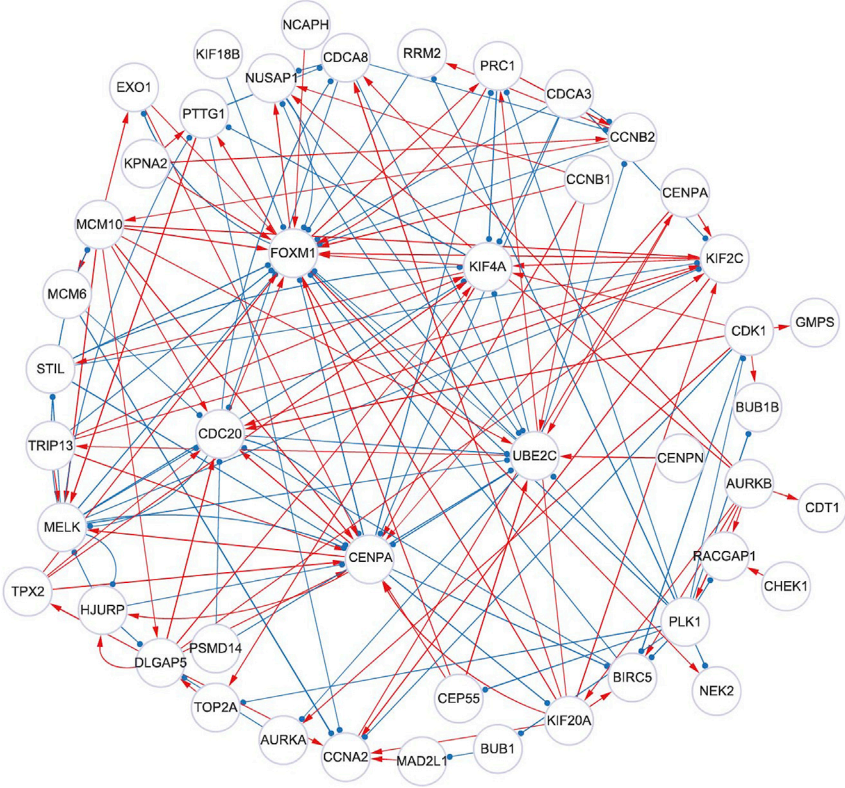


Figure 4. Gene regulatory networks with threshold 0.95 as edge credible level

The color of the edge represents the sign of the edge, i.e., red for positive regulation and green for negative regulation, respectively.

```
df=data.frame(trt,outcome)

p=ggplot(df, aes(reorder(trt,-outcome), outcome)) +

  geom_bar(aes(fill=outcome),stat="identity")+xlab("")+ylab("Hub scores")+

  scale_fill_gradient(low = "Yellow", high = "Red", na.value = NA)+

  theme_minimal()#+theme(axis.text.x = element_text(angle=90, hjust=1));

  p
```

b. You can also draw time-course graphics of the key genes using `Time_course()` function, as shown in Figure 6.

```
> Time_course(Eig_scores,cut=5,TCG_series,pseudo_time)
```

10. Further analysis or validation of the key genes.
    a. The barplot above (Figure 5) suggests that FOXM1 is the most important gene in the inferred GRN during breast cancer progression. To verify the statistical significance of the prognostic power of FOXM1, we analyze the association of FOXM1 expression with survival time data of breast cancer patients in the GSE7390 dataset. The survival time data can be extracted using the following codes:

```
Sys.setenv(VROOM_CONNECTION_SIZE=1e8);
library(GEOquery)
library(Biobase)
my_id="GSE7390";
gset=getGEO(my_id,GSEMatrix =TRUE, getGPL=FALSE);
FUN7=function(strg){return(as.numeric(substring(strg,first=7)))};
FUN8=function(strg){return(as.numeric(substring(strg,first=8)))};
FUN9=function(strg){return(as.numeric(substring(strg,first=9)))};
os=as.array(gset[["GSE7390_series_matrix.txt.gz"]]@phenoData@data
[["characteristics_ch1.15"]]);
os=apply(os,1,FUN=FUN7)/365;
eos=as.array(gset[["GSE7390_series_matrix.txt.gz"]]@phenoData@data
[["characteristics_ch1.16"]]);
eos=apply(eos,1,FUN7);
rfs=as.array(gset[["GSE7390_series_matrix.txt.gz"]]@phenoData@data
[["characteristics_ch1.13"]]);
rfs=apply(rfs,1,FUN=FUN8)/365;
erfs=as.array(gset[["GSE7390_series_matrix.txt.gz"]]@phenoData@data
[["characteristics_ch1.14"]]);
erfs=apply(erfs,1,FUN8);
dmfs=as.array(gset[["GSE7390_series_matrix.txt.gz"]]@phenoData@data
[["characteristics_ch1.17"]]);
dmfs=apply(dmfs,1,FUN=FUN9)/365;
edmfs=as.array(gset[["GSE7390_series_matrix.txt.gz"]]@phenoData@data
[["characteristics_ch1.18"]]);
edmfs=apply(edmfs,1,FUN=FUN9);
#os,rfs,dmfs: survival time data with respect to overall survival, relapse-free survival and
distant metastasis-free survival.
#eos,erfs,edmfs: 0/1 vector for samples labeling whether corresponding events are observed.
```

b. Then you can use `KM_analysis()` function to draw survival probability curves of patients with high or low expression of FOXM1. The function also calculates the log-rank p-values with respect to overall survival (OS), relapse-free survival (RFS) and distant metastasis-free survival (DMFS) (Figure 7).

```
#—KM analysis
cut=1;
top_id=names(Eig_scores);
par(mfcol=(c(cut,3)))
for(i in 1:cut){
  ng=top_id[i]; #ng is the gene that you want to apply KM analysis.
```

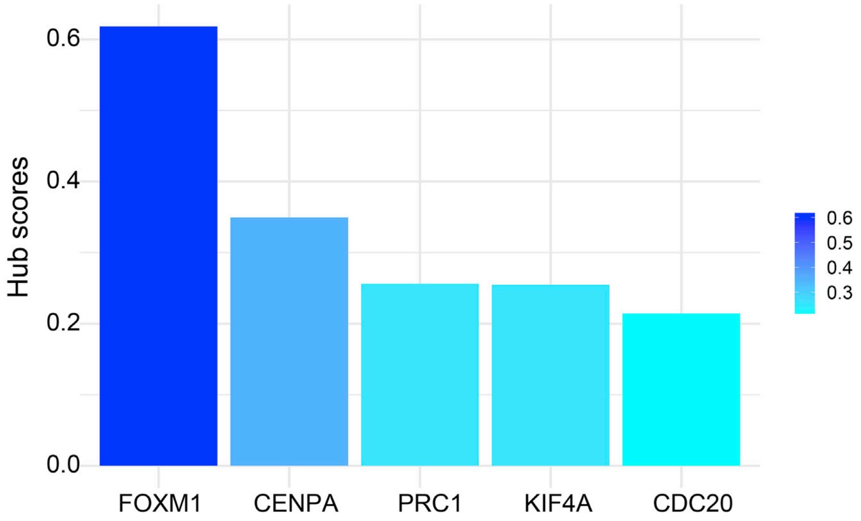


Figure 5. Top 5 genes ranked according to the hub scores

```
KM_analysis(os,eos,ng);text(4,0.4,"OS",cex=1.5);
KM_analysis(rfs,erfs,ng);text(4,0.4,"RFS",cex=1.5);
KM_analysis(dmfs,edmfs,ng);text(4,0.4,"DMFS",cex=1.5);}
```

EXPECTED OUTCOMES

Running each section of codes in "step-by-step method details" generates corresponding figures. In the step of progression inference, we obtain smoothed trajectories of expression for each gene as well as the pseudo time for each sample (e.g., Figures 1 and 2). In the step of GRNs inference, the outcomes contain posterior mean and standard deviation for each regulatory parameter in the GRN model (e.g., Figure 3). In the step of network visualization, we obtain a .csv file that can be imported into Cytoscape (e.g., Figure 4). In the step of

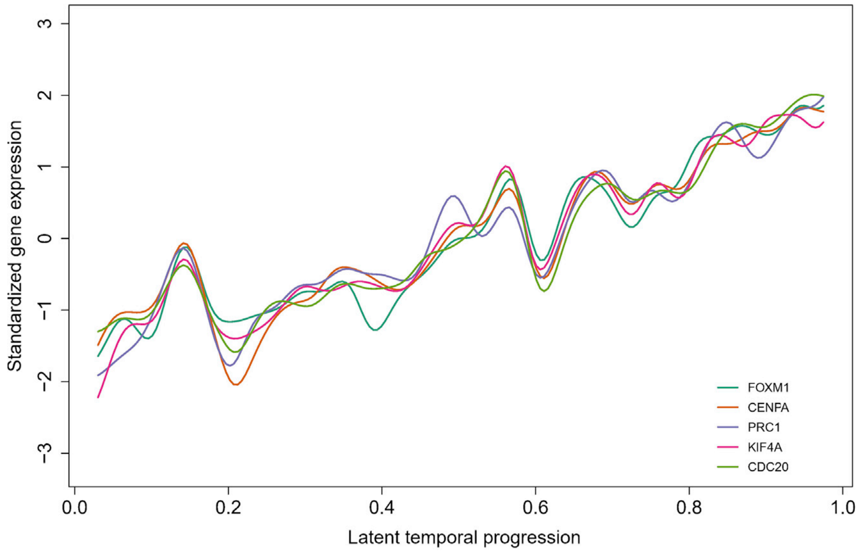


Figure 6. Time course curves of the top 5 genes

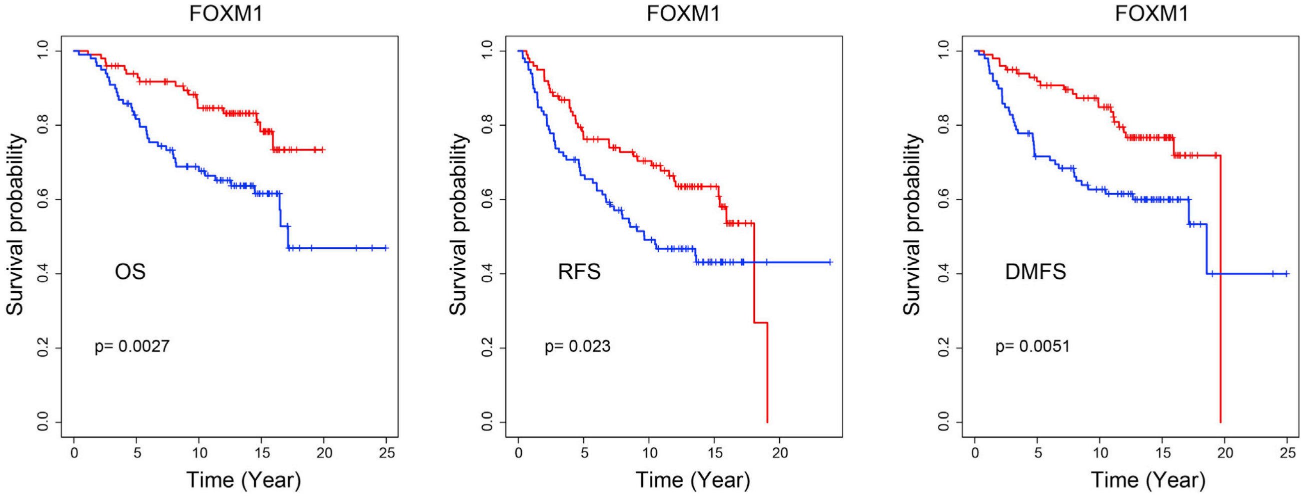


Figure 7. Clinical relevance of FOXM1 for breast cancer patients

The red curve corresponds to low level of gene expression and the blue curve corresponds to high expression level. The log-rank test p values are used to assess the statistical significance of difference between the two K-M survival curves.

identification of key genes, we get the hub score for each gene in the GRN (e.g., Figure 5) as well as the time course expression profile of the top ranked genes (e.g., Figure 6). In the step of further analysis of the key genes, we may conduct experimental validation for the top ranked genes or perform some downstream analyses such as K-M survival analysis (e.g., Figure 7). The file is saved in the current working directory.

LIMITATIONS

PROB_R only leverages gene expression and staging information for latent-temporal progression inference. Other covariates (e.g., genetic mutation, molecular subtypes) might also be useful for progression inference, which could be integrated into PROB_R for better inference of disease progression. Although our protocol is illustrated for clinical transcriptomic data, PROB_R can be applied to other types of cross-sectional data, for instance, time-stamped single cell RNA-seq data, by substituting grade information with time course points in PROB_R. However, the dropout issue in the scRNA-seq data should be considered for GRN inference when using `ODE_Bayesian_Lasso()` function. On the other hand, the current algorithm only considers single trajectory of progression but fall short on inferring multiple branching sub-trajectories of disease progression or cell differentiation. Additionally, the K-M estimator assumes that the event time is independent of the censored time, hence the method of validation for the top gene might be inaccurate for data that violates this assumption. Furthermore, this protocol cannot deal with missing data in clinical transcriptomic data. In the future study, we will continue to improve PROB_R for wider applications.

TROUBLESHOOTING

Problem 1

In step 1, the computer fails to download the dataset (such as GEO: GSE7390 dataset) from the GEO or other databases.

Potential solution

The problem occurs frequently with a warning "The size of the connection buffer is not large enough". If so, you can try increase the buffer size by using the following codes:

```
>Sys.setenv(VROOM_CONNECTION_SIZE=1e8)
```

Or you can adjust the buffer size that is suitable for your device until no such warnings occur. If that does not solve the problem, you may need to check whether required dependencies are installed.

Problem 2

In step 1, when running the `PROB_GEOinstall()` function, it appears an error “One or more parsing issues, see ‘problems()’ for details”.

Potential solution

This error is possibly due to different R versions. We have tested and verified that the codes can be successfully run in the environment of R 4.0.5. Confirm that your R environment has the same version, or try to run the codes within this function line-by-line.

Problem 3

In step 1, the RStudio can’t find functions required to execute the codes.

Potential solution

If the problem occurs with a warning “Can’t find packages called…”, install that package mentioned. Otherwise, make sure that you download executable functions we provide and execute them in your working environment.

Problem 4

In step 3, when applied to the user’s dataset for analysis, PROB_R fails in the step of progression inference.

Potential solution

Make sure that your data matrix follows the format in Table 1. Check whether rows and columns are reversed or grade information is correctly included.

Problem 5

In step 8, the Cytoscape cannot read the output file for network visualization.

Potential solution

Check your version of Cytoscape and update it to the most recent one. Detailed guides of operations in Cytoscape can be found from its official website. Alternatively, you can try opening the .csv file with Microsoft Excel and save it in another format, such as a tab delimited text file, and then try to create the network in Cytoscape with this file.

Problem 6

In step 7, I want to infer a GRN without a prior network structure. How should I adjust the input parameters in the `ODE_Bayesian_Lasso()` function?

Potential solution

If you want to infer a GRN without a prior network structure, you can skip step 5 and apply ODE_Bayesian_Lasso() function in step 6 as follows.

```
no_prior= matrix(rep(1,100*100),100,100);

Breast_BL=ODE_Bayesian_Lasso(TCG_series,pseudo_time,no_prior);
```

Problem 7

In step 10, can I use `KM_analysis()` function to test prognostic significance of the importance of any other genes in addition to the identified key genes?

Potential solution

Yes. For example, if you want to use overall survival time data to examine prognostic significance of MCM10 gene in the breast cancer progression, you can use the following codes:

```
> KM_analysis(os,eos,’’MCM10’’);
```

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Xiaoqiang Sun, sunxq6@mail.sysu.edu.cn or xiaoqiangsun88@gmail.com.

Materials availability

This study did not generate new unique reagents.

Data and code availability

The code generated during this study is available at GitHub: https://github.com/SunXQlab/PROB_R and at Zenodo: https://doi.org/10.5281/zenodo.6555525.

ACKNOWLEDGMENTS

This work was supported by grants from the National Key R&D Program of China (2021YFF1200903), the National Natural Science Foundation of China (11871070), the Guangdong Basic and Applied Basic Research Foundation (2020B151502120).

AUTHOR CONTRIBUTIONS

X.S. conceived this study; Z.D. and X.S. designed and tested the protocol; Z.D. and X.S. wrote the manuscript; X.S. supervised this study; and all authors have read and agreed to the published version of the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Albert, R., and Barabasi, A.-L. (2001). Statistical mechanics of complex networks. Rev. Mod. Phys. *74*. https://doi.org/10.1103/RevModPhys.74.47.

Albanese, D., Filosi, M., Visintainer, R., Riccadonna, S., Jurman, G., and Furlanello, C. (2012). Minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers. Bioinformatics *29*, 407–408. https://doi.org/10.1093/bioinformatics/bts707.

Alhamzawi, R., and Ali, H.T.M. (2020). Brq: an R package for Bayesian quantile regression. METRON *78*, 313–328. https://doi.org/10.1007/s40300-020-00190-6.

Csardi, G., and Nepusz, T. (2006). The igraph software package for complex network research. InterJ. Complex Syst. *1695*, 1–9.

Davis, S., and Meltzer, P.S. (2007). GEOquery: a bridge between the gene expression Omnibus (GEO) and BioConductor. Bioinformatics *23*, 1846–1847. https://doi.org/10.1093/bioinformatics/btm254.

Gramacy, R.B. (2019). Monomvn: Estimation for MVN and Student-t Data with Monotone Missingness (The Comprehensive R Archive Network (CRAN).). R package version 1.9-13.

Huber, W., Carey, V.J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B.S., Bravo, H.C., Davis, S., Gatto, L., Girke, T., et al. (2015). Orchestrating high-throughput genomic analysis with bioconductor. Nat. Methods *12*, 115–121.

Kendall, M. (1955). Further contributions to the theory of paired comparisons. Biometrics *11*, 43–62. https://doi.org/10.2307/3001479.

Kolde, R. (2019). Pheatmap: Pretty Heatmaps (The Comprehensive R Archive Network (CRAN).). R package version 1.0.12.

Kolberg, L., Raudvere, U., Kuzmin, I., Vilo, J., and Peterson, H. (2020). gprofiler2 – an R package for gene list functional enrichment analysis and namespace conversion toolset g:Profiler. F1000Research *9*, ELIXIR-709.

Mann, H.B. (1945). Nonparametric tests against trend. Econometrica *13*, 245. https://doi.org/10.2307/1907187.

Park, T., and Casella, G. (2008). The Bayesian Lasso. J. Am. Stat. Assoc. *103*, 681–686.

Pohlert, T. (2020). Trend: Non-Parametric Trend Tests and Change-Point Detection (The Comprehensive R Archive Network (CRAN).). R package version 1.1.4.

R Core Team (2013). R: A Language and Environment for Statistical Computing (R Foundation for Statistical Computing).

RStudio Team (2020). RStudio: Integrated Development for R (RStudio (PBC)).

Sun, X., Zhang, J., and Nie, Q. (2021). Inferring latent temporal progression and regulatory networks from cross-sectional transcriptomic data of cancer samples. PLoS Comput. Biol. *17*, e1008379. https://doi.org/10.1371/journal.pcbi.1008379.

Therneau, T. (2022). A Package for Survival Analysis in R (The Comprehensive R Archive Network (CRAN).). R package version 3.3-1.

Türei, D., Korcsmáros, T., and Saez-Rodriguez, J. (2016). OmniPath: guidelines and gateway for literature-curated signaling pathway resources. Nat. Methods *13*, 966–967.

Wickham, H. (2007). Reshaping data with the reshape package. J. Stat. Softw. *21*, 1–20.

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis (Springer-Verlag).

Wickham, H., and Girlich, M. tidyr: Tidy Messy Data. https://github.com/tidyverse/tidyr.