

# Multivariate Survival Analysis

Performs one feature selection technique (Univariate Cox Regression, PCA, or Diffusion Maps), and multivariate survival analysis (Cox Regression, Random Survival Forests, and Ridge Regularized Cox Regression). If Univariate Cox Regression is used, you can select genes by a significance threshold (p or q-value) or try the m most significance genes (for m = 5, 10, 15, 20, 25, 30). Implements bootstrapping to get average train and out-of-bag concordance indices that are representative of the true values.

Please email [russell.a.yang@gmail.com](mailto:russell.a.yang@gmail.com) with any questions or problems.

## Tutorial – Bootstrapped Survival Analysis (Cox, RSF, Ridge Cox)

1. Create a directory called PRECOG\_DMFS on your computer
2. Inside, create directories called Combined, Original, Pictures, RemovedNA, Results, Scripts, Significant, and Split.
  - a. Original is where you put your datasets to be analyzed
  - b. RemovedNA is an intermediate directory where the script places processed datasets that have the samples with NA removed
  - c. Combined is a directory that contains the clinical annotation file with the selected genes appended at the right
  - d. Pictures is where the script will write the histograms, quantile-quantile plots, scree plots, cumulative scree plots, and diffusion map plots
  - e. Scripts is the directory of scripts used to conduct the analysis
  - f. Significant is a directory that has other subdirectories. Each subdirectory has datasets subsetted by a significance condition
  - g. Split holds gene expression and clinical annotation datasets that have been split into two parts
3. You have 2 datasets – a gene expression dataset and a clinical annotation dataset. The gene expression dataset should look like the picture on the left (genes are the rows and samples are the columns), and the clinical annotation dataset should look like the picture on the right (samples are the rows, and clinical annotations are the columns). In the gene expression dataset, you should have columns for the gene ID and for the Description. In the clinical annotation dataset, two of the clinical annotations must be the time and status variables for the thing you are trying to model (e.g. DMFS\_Time and DMFS\_Status)

	A	B	C	D
1	Name	Description	GSM22365	GSM22366
2	13666	APOBEC3C	0.734146	0.47301
3	8563	KIF17 - kin	-0.27109	-0.24502
4	8434	MMS19 - f	0.264748	0.076877
5	5006	ADAMDEC	0.156043	4.123477
6	3509	DPF2 - D4,	-0.95657	0.442501
7	10309	FGF13 - fib	2.121733	2.039009
8	18581	SHROOM3	3.178149	3.42625
9	10122	NAP1L3 - r	-1.16	-1.52766

	A	B	C	D	E
1	Array	Sample_t	DMFS_Tim	DMFS_Sta	Size
2	GSM22365	microdisse	75	1	3.5
3	GSM22366	microdisse	21	1	3
4	GSM22367	microdisse	38	1	3
5	GSM22368	microdisse	43	1	>2.0
6	GSM22369	microdisse	169	0	1.6
7	GSM22370	microdisse	42	1	0.9
8	GSM22371	microdisse	44	1	2.2
9	GSM22372	microdisse	63	1	3

4. Put the gene expression and clinical annotation datasets that you want to analyze in the Original directory

5. Put all the scripts into the Scripts directory
6. To do bootstrapping analysis, you can use the analysis function. By default, the function will use 30% of the data to select the covariates (e.g. genes), and it will do bootstrapping on the other 70% of the data.
  - a. For the 30% of the data that is used to select the covariates, you have a choice of Univariate Cox Regression, Principal Component Analysis (PCA), and Diffusion Maps. If you choose Univariate Cox Regression, you can either choose genes by a p or q-value significance threshold, or choose the m most significant genes (for m = 5, 10, 15, 20, 25, 30). If you choose PCA or Diffusion Maps, then the m first PCA components or m first diffusion coordinates will be used (for m = 5, 10, 15, 20, 25, 30).
  - b. For the 70% of the data that is used for bootstrapping, the program will randomly sample with replacement n times, where n is the number of samples in the 70% of the dataset. It will use the unused samples (out-of-bag samples) as a testing set. This bootstrapping process is repeated for different random resamplings, and is done 100 times by default. The analysis can be conducted for multivariate Cox Regression, Random Survival Forests, and Ridge Regularized Cox Regression. DeepSurv is run in Python instead of R, and this script does not run it. For each of the multivariate techniques, the outputs are (1) the mean train and out-of-bag concordance indices, (2) histograms of the train concordance indices saved to the Pictures directory, (3) histograms of the out-of-bag concordance indices saved to the Pictures directory, (4) quantile-quantile plots of the train concordance indices (to assess if the histogram is a normal distribution) saved to the Pictures directory, (5) quantile-quantile plots of the test concordance indices (to assess if the histogram is a normal distribution) saved to the Pictures directory, (6) scree & cumulative scree plots if PCA has been conducted, and (7) diffusion map plots if Diffusion Mapping has been conducted.
7. Open R
8. Set your working directory to be the directory that contains PRECOG\_DMFS. For me, this is:
9. Install the following packages if they are not already installed: survival, survminer, ComplexHeatmap, tm, fdrtool, qvalue, preprocessCore, impute, rms, randomForestSRC, survcomp, diffusionMap, ArgumentCheck, glmnet, boot. Some will need to be installed via CRAN, and others via Bioconductor.

```
setwd("/Users/russe/Downloads")
```

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

install.packages(c("survival", "survminer", "tm", "fdrtool",
  "rms", "randomForestSRC", "diffusionMap", "ArgumentCheck",
  "glmnet", "boot"))
BiocManager::install("ComplexHeatmap")
BiocManager::install("qvalue")
BiocManager::install("preprocessCore")
BiocManager::install("impute")
BiocManager::install("survcomp")
```

10. Set your working directory to be the directory that contains PRECOG\_DMFS. Source all the scripts, and load all the packages. If you are not using certain functionalities, then you don't need to load all the scripts or load all the packages.

```
setwd("/Users/russe/Downloads") # Change to your directory
source("PRECOG_DMFS/Scripts/Run_Cox_function.R")
source("PRECOG_DMFS/Scripts/unisurvcnsHR.R")
source("PRECOG_DMFS/Scripts/removena.R")
source("PRECOG_DMFS/Scripts/findsignificant.R")
source("PRECOG_DMFS/Scripts/upsetplot.R")
source("PRECOG_DMFS/Scripts/coxsignificant.R")
source("PRECOG_DMFS/Scripts/changefiletype.R")
source("PRECOG_DMFS/Scripts/split.R")
source("PRECOG_DMFS/Scripts/combine.R")
source("PRECOG_DMFS/Scripts/bootanalysis.R")
source("PRECOG_DMFS/Scripts/analysis.R")
source("PRECOG_DMFS/Scripts/normalize.R")
library("survival")
library("survminer")
library("ComplexHeatmap")
library("tm")
library("fdrtool")
library("qvalue")
library("preprocessCore")
library("impute")
library("rms")
library("randomForestSRC")
library("survcomp")
library("diffusionMap")
library("ArgumentCheck")
library("glmnet")
library("boot")
```

11. You can now run the analysis function. The function is interactive, and it will ask a series of questions. First, it will ask for your project. It will then do some preprocessing steps. Then, it will ask if you want to do univariate feature selection, PCA, or diffusion maps. Then, it will ask if you want to do Multivariate Cox Regression, Random Survival Forests, or Ridge Regularized Cox Regression. You can say yes to multiple of these. It will then run your feature selection/dimensionality reduction step of choice: univariate feature selection, PCA, or diffusion mapping. It will save relevant images (such as scree plots) to the Pictures directory. If you choose univariate feature selection, it will ask if you want to choose genes based on some significance threshold or based on a certain number of most significant genes. In either case, it will also ask whether you want to use the p-value or q-value to quantify significance.

```
analysis()
```

```
# Example output:
[1] "You will be asked a series of questions. NOTE: there is no
robust input validation"
Enter a project name: Breast cancer.GSE3494.HGU133A_EntrezCDF
[1] "PREPROCESSING (1/3)"
```

```

[1] "Removing samples with NA"
[1] "Saving new datasets"
[1] "Normalizing, transforming, and imputing missing values"
[1] "Shuffled and split datan"
[1] "Extracted gene expression split"
[1] "Converted to numeric"
[1] "Shuffled and split info"
Do you want to do univariate feature selection (u), PCA (p), or
diffusion maps (d)? Enter u, p, or d: u
Do you want to do Multivariate Cox Regression? Enter y or n: y
Do you want to do Random Survival Forests? Enter y or n: n
Do you want to do Ridge Regularized Cox Regression? Enter y or n:
y
Do you want to select genes by a significance threshold (s) or
try the 5, 10, 15, 20, 25, & 30 most significant genes (n)? Enter
s or n: n
Do you want to select a certain number of most significant genes
by p-value (p) or q-value (q)? Enter p or q: p
[1] "FEATURE SELECTION (2/3)"
[1] 1000
[1] 2000
[1] 3000
[1] 4000
[1] 5000
[1] 6000
[1] 7000
[1] 8000
[1] 9000
[1] 10000
[1] 11000
Step 1... determine cutoff point
Step 2... estimate parameters of null distribution and eta0
Step 3... compute p-values and estimate empirical PDF/CDF
Step 4... compute q-values and local fdr

[1] "MULTIVARIATE ANALYSIS (3/3)"
[1] "5 covariates"
[1] "Now processing Breast cancer.GSE3494.HGU133A_EntrezCDF"
[1] "Saving new .tsv files ..."
[1] "X26872_at" "X3101_at" "X55335_at" "X57088_at" "X79840_at"
[1] "Cox train & test"
[1] 0.65
[1] 0.544
[1] "Coxnet trainlse, testlse, trainmin, testmin"
[1] 0.64
[1] 0.568
[1] 0.644
[1] 0.564
[1] "10 covariates"
[1] "Now processing Breast cancer.GSE3494.HGU133A_EntrezCDF"
[1] "Saving new .tsv files ..."

```

```

[1] "X10471_at" "X26872_at" "X3101_at" "X3182_at" "X55335_at"
"X57088_at" "X79840_at"
[8] "X81847_at" "X81873_at" "X971_at"
[1] "Cox train & test"
[1] 0.717
[1] 0.551
[1] "Coxnet trainlse, testlse, trainmin, testmin"
[1] 0.7
[1] 0.584
[1] 0.712
[1] 0.567
[1] "15 covariates"
[1] "Now processing Breast cancer.GSE3494.HGU133A_EntrezCDF"
[1] "Saving new .tsv files ..."
[1] "X10471_at" "X23768_at" "X26872_at" "X27129_at" "X3101_at"
"X3182_at" "X51070_at" "X55314_at" "X55335_at" "X57088_at"
"X79840_at" "X81847_at" "X81873_at" "X9710_at" "X971_at"
[1] "Cox train & test"
[1] 0.775
[1] 0.6
[1] "Coxnet trainlse, testlse, trainmin, testmin"
[1] 0.746
[1] 0.591
[1] 0.774
[1] 0.598
[1] "20 covariates"
[1] "Now processing Breast cancer.GSE3494.HGU133A_EntrezCDF"
[1] "Saving new .tsv files ..."
[1] "X10471_at" "X23768_at" "X26872_at" "X27129_at" "X3101_at"
"X3182_at" "X51070_at" "X51412_at" "X55314_at" "X55335_at"
"X57088_at" "X7402_at" "X79840_at" "X81847_at" "X81873_at"
[16] "X8404_at" "X8721_at" "X9651_at" "X9710_at" "X971_at"
[1] "Cox train & test"
[1] 0.821
[1] 0.643
[1] "Coxnet trainlse, testlse, trainmin, testmin"
[1] 0.772
[1] 0.575
[1] 0.815
[1] 0.609
[1] "25 covariates"
[1] "Now processing Breast cancer.GSE3494.HGU133A_EntrezCDF"
[1] "Saving new .tsv files ..."
[1] "X10471_at" "X23284_at" "X23768_at" "X26872_at" "X27129_at"
"X2791_at" "X307_at" "X3101_at" "X3182_at" "X51070_at"
"X5119_at" "X51412_at" "X55314_at" "X55335_at" "X57088_at"
[16] "X7402_at" "X79673_at" "X79840_at" "X81847_at" "X81873_at"
"X8404_at" "X8721_at" "X9651_at" "X9710_at" "X971_at"
[1] "Cox train & test"
[1] 0.846
[1] 0.633

















```

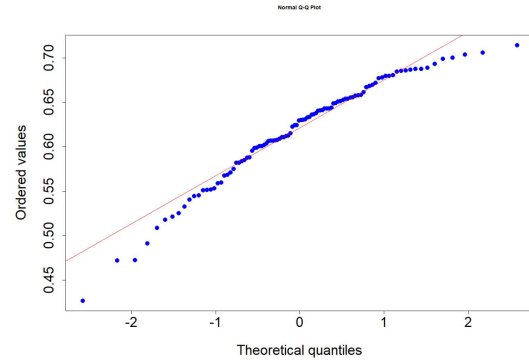
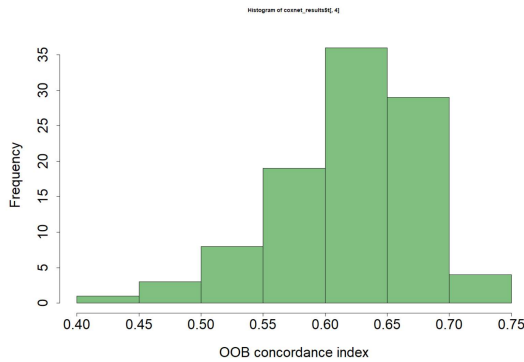
```

[1] "Coxnet trainlse, testlse, trainmin, testmin"
[1] 0.784
[1] 0.613
[1] 0.835
[1] 0.634
[1] "30 covariates"
[1] "Now processing Breast cancer.GSE3494.HGU133A_EntrezCDF"
[1] "Saving new .tsv files ..."
  [1] "X10471_at" "X11284_at" "X151011_at" "X23284_at"
"X23768_at" "X26872_at" "X27129_at" "X2791_at" "X28952_at"
"X307_at"
[11] "X3101_at" "X3182_at" "X51070_at" "X5119_at"
"X51412_at" "X55314_at" "X55335_at" "X57088_at" "X7105_at"
"X7402_at"
[21] "X79673_at" "X79840_at" "X81847_at" "X81873_at"
"X8404_at" "X8470_at" "X8721_at" "X9651_at" "X9710_at"
"X971_at"
[1] "Cox train & test"
[1] 0.86
[1] 0.612
[1] "Coxnet trainlse, testlse, trainmin, testmin"
[1] 0.779
[1] 0.602
[1] 0.838
[1] 0.618

```

12. Mean train and out-of-bag concordance indices for 5, 10, 15, 20, 25, and 30 covariates will be printed to the terminal if you use the “n” method. After the script is done running, the Pictures folder will be populated with the histograms, quantile-quantile plots, scree plots & cumulative scree plots (if PCA is done), and diffusion map plots (if Diffusion Mapping is done).

<input type="checkbox"/> Name	<input type="checkbox"/> Name
 Combined	 coxnetoobminq30.p...
 Original	 coxnetoobmin30.png
→  Pictures	 coxnettrainminq30.p...
 RemovedNA	 coxnettrainmin30.png
 Results	 coxnetoob1seq30.p...
 Scripts	 coxnetoob1se30.png
 Significant	 coxnettrain1seq30.p...
 Split	 coxnettrain1se30.png



13. You can adjust the script according to your analysis. For example, you can choose numbers of covariates to try or a different number of bootstrap iterations to run. You can also try a different output variable by changing all instances of DMFS to OS, etc. Each script has its own documentation.

## Tutorial – Bootstrapped Survival Analysis (DeepSurv)

1. After running the analysis above, the necessary intermediate files will be available in the Significant and Split directories. These include the properly subsetting bootstrapping set (70% of the data) and the properly subsetting info datasets (70% of the data).
2. Install the libraries and dependencies required by DeepSurv (<https://github.com/jaredleekatzman/DeepSurv>). The included instructions did not work for me at first, but that could be because of my specific laptop.
3. Download the Jupyter Notebook files and place them inside the Notebooks directory in DeepSurv's file structure.
4. Open Jupyter Notebook
5. Go to the Notebooks directory in Jupyter Notebook and open the notebook you would like to run. For example, use gene5.ipynb to run the analysis using Univariate Cox Regression with 5 genes on DeepSurv. Use pca5.ipynb to run the analysis using Principal Component Analysis with 5 genes on DeepSurv. Use diff5.ipynb to run the analysis using Diffusion Maps with 5 genes on DeepSurv, etc.
6. Scroll down to the 5<sup>th</sup> cell and change the file paths (model\_data\_fp and model\_info\_fp) to point to the dataset on your computer that you want to analyze.
7. You can run all cells at once by clicking Kernel → Restart & Run All
8. The analysis will take a long time. To run it overnight, you cannot run multiple notebooks using the same Jupyter Notebook instance, as there are not enough allocated resources and the runs will fail. Instead, you can open multiple instances of Jupyter Notebook and run one DeepSurv analysis on each instance. Using this method, it will take about 15 hours to run DeepSurv for 5, 10, 15, 20, 25, & 30 covariates using one feature selection/dimensionality reduction technique.
9. When each Jupyter Notebook is done running, it will automatically calculate the average train and out-of-bag concordance indices, and it will also plot the histogram of out-of-bag concordance indices and the quantile-quantile plot.

## Scripts – Overview

Each script has been documented individually

*analysis*: an interactive function that will ask the user what kind of analysis and feature selection/dimensionality reduction they want to do and do it

*bootanalysis*: a helper function for analysis that performs bootstrapped analysis on a dataset according to the parameters passed to it

*coxsignificant*: for one dataset, this function finds significant covariates either by a threshold (such as  $q < 0.05$ ) or by nlowest (such as the 10 most significant genes by lowest q-value). Also combines the gene expression and clinical annotation files into “combined files”

*change filetype*: converts between RData and tsv files

*combine*: a helper function used to combine gene expression and clinical annotation files

*findsignificant*: finds significant genes either by using a threshold (p or q is less than some number n), or by nlowest (a certain # of genes by lowest p or q value)

*removena*: removes samples with NA in the output columns from both gene expression and clinical annotation datasets

*Run\_Cox\_function*: performs univariate cox regression gene-by-gene

*unisurvconsHR*: a helper function for Run\_Cox\_function

*split*: a helper function to shuffle and split a dataset into two parts

*upsetplot*: a function to visualize intersections of statistically significant genes