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 n most significance genes (for n = 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.

Tutorial – Boostrapped Survival Analysis

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	Descriptio	GSM22365	GSM22366
2	13666	APOBEC3C	0.734146	0.47301
3	8563	KIF17 - kin	-0.27109	-0.24502
4	8434	MMS19 - I	0.264748	0.076877
5	5006	ADAMDEC	0.156043	4.123477
6	3509	DPF2 - D4,	-0.95657	0.442501
7	10309	FGF13 - fil	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_ti	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 boostrapping 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 is 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
7. Open R
8. Set your working directory to be the directory that contains PRECOG_DMFS. For me, this is:

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

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.

```
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/preprocess.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:
Enter a project name: Breast cancer.GSE3494.HGU133A_EntrezCDF
[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: d
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
[1] "Performing Diffusion Mapping"
Performing eigendecomposition
Computing Diffusion Coordinates
Used default value: 50 dimensions
Elapsed time: 0.01 seconds
[1] "Saved diffusion map images to Pictures directory"
[1] "I will try 5, 10, 15, 20, 25, and 30 diffusion coordinates"
[1] "5 covariates"
[1] "X1" "X2" "X3" "X4" "X5"
[1] "Cox train & test"
[1] 0.66
[1] 0.575
[1] "Coxnet train1se, test1se, trainmin, testmin"
[1] 0.656
[1] 0.59
[1] 0.658
[1] 0.583
[1] "10 covariates"
[1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10"
[1] "Cox train & test"
[1] 0.706
[1] 0.559
[1] "Coxnet train1se, test1se, trainmin, testmin"
[1] 0.699
[1] 0.593
[1] 0.701
[1] 0.58
[1] "15 covariates"
[1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10"
"X11" "X12" "X13" "X14" "X15"
[1] "Cox train & test"
[1] 0.765
[1] 0.593
[1] "Coxnet train1se, test1se, trainmin, testmin"
[1] 0.756
[1] 0.638
[1] 0.77
[1] 0.626
[1] "20 covariates"
[1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10"
"X11" "X12" "X13" "X14" "X15" "X16" "X17" "X18" "X19" "X20"
[1] "Cox train & test"
[1] 0.79
[1] 0.591
[1] "Coxnet train1se, test1se, trainmin, testmin"
[1] 0.768
```

```
[1] 0.61
[1] 0.789
[1] 0.599
[1] "25 covariates"
[1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10"
"X11" "X12" "X13" "X14" "X15" "X16" "X17" "X18" "X19" "X20" "X21"
[22] "X22" "X23" "X24" "X25"
[1] "Cox train & test"
[1] 0.821
[1] 0.57
[1] "Coxnet train1se, test1se, trainmin, testmin"
[1] 0.779
[1] 0.639
[1] 0.801
[1] 0.607
[1] "30 covariates"
[1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10"
"X11" "X12" "X13" "X14" "X15" "X16" "X17" "X18" "X19" "X20" "X21"
[22] "X22" "X23" "X24" "X25" "X26" "X27" "X28" "X29" "X30"
[1] "Cox train & test"
[1] 0.843
[1] 0.539
[1] "Coxnet train1se, test1se, trainmin, testmin"
[1] 0.792
[1] 0.629
[1] 0.823
[1] 0.605
```

12. Mean train and out-of-bag concordance indices for 5, 10, 15, 20, 25, and 30 covariates will be printed to the terminal. 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).
13. 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.

Scripts – Overview

Each script has been documented individually and has example calls at the top

analysis: performs all the preprocessing steps and does bootstrapped analysis for 5, 10, 15, 20, 25, and 30 covariates. Uses other functions as helper functions

bootanalysis: a helper function for analysis that performs the bootstrapped analysis

preprocess: performs preprocessing steps for every dataset in the specified directory. This includes removing samples with NA, finding 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”

changefiletype: 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

unisurvcnsHR: 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