CAncer bioMarker Prediction Pipeline (CAMPP)

Thilde Bagger Terkelsen
Danish Cancer Society Research Center
thilde@cancer.com, thildebate@gmail.com

2019-11-20

Introduction

The CAncer bioMarker Prediction Pipeline (CAMPP) is a simple bioinformatics tool intended to automatize identification of potential diagnostic and prognostic cancer biomarkers. The pipeline is versatile and may be used for analysis of a variety of quantitative biological data from high throughput platforms, including genes, proteins, small RNAs, lipids and glycans. CAMPP currently supports; differential expression/abundance analysis, LASSO/Elastic Net regression, Weighed Gene Co-expression Network Analysis, Correlation analysis and Survival analysis (Cox proportional hazard regression). CAMPP is written in R [1] and runs via a linux command-line with flags specifying arguments.

Contents

Workflow Overview

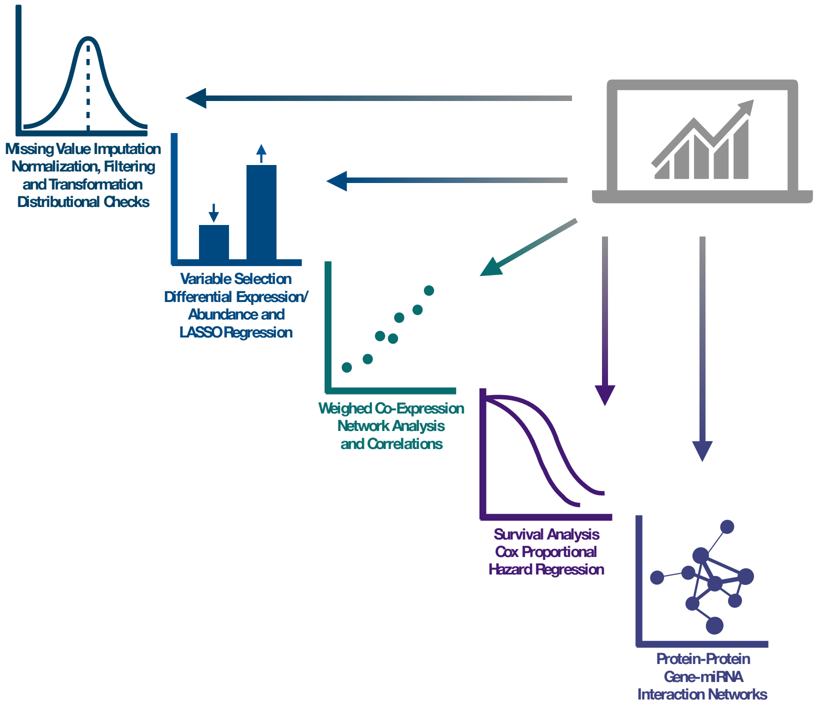


Figure 1: Cancer bioMarker Prediction Pipeline Analysis Flow.

1 About CAMPP

The CAncer bioMarker Prediction Pipeline was developed for internal use at the Danish Cancer Society Research Center (DCRC) providing researchers with an easy way to identify biomarkers for cancer diagnostics and prognostics. In its original form the pipeline was implemented using interstitial fluid samples and tissue samples from patients with breast cancer but it may be applied to any type of data with comparable properties.

The pipeline can perform the following types of analysis (I) Differential expression/abundance analysis (limma [26]), (II) LASSO/Elastic-Net regression (glmnet [12]), (III) Weighed Gene Co-expression Network Analysis (WGCNA [23]), (IV) Correlation analysis (Pearson/Spearman), (V) Survival analysis (Cox proportional hazard regression, survcomp [29]) and (VI) protein-protein/gene-miRNA interaction network analysis (multimiR [27] and the STRING [20]). In addition to different types of analysis the pipeline performs missing value imputation, normalization and transformation, along with data distributional checks. Differential expression/abundance analysis can be performed with as few as 3 biological replicates in each group, whereas LASSO/Elastic-Net regression, WGCNA, correlation and survival analysis is only advisable with a large(r) number of samples. CAMPP may be run with a variety of biological molecules (genes, miRNAs, proteins etc.) from various platforms (high-throughput sequencing, microarray data, liquid chromatography–mass spectrometry, etc.). For differential expression/abundance analysis the limma package [26] for R is utilized. Though limma was originally designed for microarray data, and more recently RNAseq, a number of studies have shown the versatility of this software for the analysis of other -omics data [4, 21, 24]. LIMMA has few underlying statistical assumptions and is known to be powerful for small sample sizes as a result of shrinkage of feature-specific variances [31].

> N.B The user should be careful with LASSO/Elastic-Net regression, as this type of analysis needs a good number of samples, in a balanced group design, to yield reliable results. Recommended is a minium of 30 samples for each group in the design [12]. To perform WGCNA [23] at least 15 samples must be available for analysis (section 5 in FAQ here: `https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/faq.html`), see description of WGCNA on page X.

Survival (over-all, relapse-free) of patients based on the abundance/expression of a given marker is predicted using cox proportional-hazard model. Check for proportional hazard and linearity of continuous covariates are automatically performed. If the user wants to correct for confounders, such as patient age at diagnosis, drug treatment, etc., these may be included in the model.

> N.B For every parameter added to the model more events are needed for appropriate statistical power. A rule of thumb is $\sim$ 10 events for each parameter,

> however, this will somewhat depend on the number of levels within a given parameter. Generally results of the survival analysis should be interpreted with caution as the pipeline is unable to account for all possible options and assumptions related to this type of analysis.
> The check for proportional hazard and linearity of continuous covariates should be OK before interpreting any results!

2 Requirements

To run CAMPP, a working R version 3.6.1 (or newer) is required [1]. In addition, macbook users must have Xcode installed, as well as the Xcode command-line tools. Install Xcode as shown below.

Open a terminal on your computer and type:

Install Xcode

```
xcode-select -install
```

The Pipeline relies on a variety of R-packages, see list below.

> To obtain information on R-packages above, e.g. developers and release information (articles) go to `R` and type `citation("package_name")`.

3 Download

The CAncer bioMarker Prediction Pipeline is easily downloaded by following the steps outlined below.

1. (1) Download the GitHub repository as a .zip locally from the address below. (2) If you wish to work externally on a server you can upload the repository from your computer to the server using scp from the command-line or (3) use git for a direct download:

Download CAMPP

```
(1) https://github.com/ELELAB/CAncer-bioMarker-Prediction
-Pipeline-CAMPP

(2) scp CAncer-bioMarker-Prediction-Pipeline-CAMPP-master.zip
name@login:/path-to-dir.

(3) git clone git://github.com/ELELAB/CAncer-bioMarker-
Prediction-Pipeline-CAMPP.git
```

Table 1: R-packages

Analysis	Packages
Missing value imputation	impute (v 1.56.0) [17]
Distributional checks	fitdistrplus (v 1.0.11) [8]
Excel formatting	openxlsx (v 4.1.0) [34]
Plotting	heatmap.plus (v 1.3), squash (v 1.0.8) viridis (v 0.5.1), ggplot2 (v 3.1.0) [7, 11, 14, 35]
Data management	data.table (v 1.11.8) stackoverflow (v 0.1.2), plyr (v 1.8.4) scales (v 1.0.0) [9, 13, 36, 37]
K-means Clustering	mclust (v 5.4.3) [30]
DE/DA Analysis	sva (v 3.30.0), limma (v 3.38.2) [19, 26]
LASSO/Elastic-Net	glmnet (v 2.0.16) [12]
Co-expression analysis	WGCNA (v 1.66) [23]
Interaction Networks	igraph (v 1.2.4), biomaRt (v 2.38.0) multiMiR (v 1.4.0), devtools (v 2.0.1) [6, 10, 27, 38]
Survival analysis	survminer (v 0.4.3), survcomp (v 1.32.0) [22, 29]

> If you are using the "git clone" option, make sure that the Git software has been installed.

2. Unzip the repository. From the command-line use "unzip" to decompress.

Decompress CAMPP

```
unzip CAncer-bioMarker-Prediction-Pipeline-CAMPP-master.zip
```

> The unzipped repository should contain the following: `Readme.md LICENSE.md, CAMPPFunctions.R CAMPP.R, CAMPPManual, Data and renv.lock`.

4 Installation of R-packages

Download and installation of R-package dependencies is automatically performed the first time the pipeline is run. However, there are two ways of using packages:

1. **Newest R-packages.**
   If the flag *-e* is omitted, CAMPP will check for the required R-packages, and will download and install these if needed. The CRAN mirror default is *Germany*: `https://ftp.fau.de/cran/`, which may work in many cases, however, the user should preferably use the CRAN mirror closest to their geographical location. See mirrors at: `https://cran.r-project.org/mirrors.html` .The user may then specify the mirror by setting the option *-e* to the desired url.

2. **Renv Library.**
   Although we aim at keeping the code updated and compatible with new R-releases and package updates, we also provide the user with a **renv** [33] "freeze" of the R-packages, if "classic" R-package management fails (see option 1. above). We hope that this will stabilize the pipeline, so that if it breaks due to package updates, the user will still be able to run it while potential discrepancies in the code are being resolved.
   When CAMPP is run, the option *-e* should be set to **stable** (not case sensitive). This tells CAMPP.R to use the renv library instead of checking for the newest versions of packages.

   ```
   Warning message:
   In install.packages("name_of_package") :  package ‘name_of_pack-
   age’ is not available...
   ```

   In this case the package may need to be installed directly from the github repository. In order to do this open R. Check that the R-package `devtools` is installed and loaded. Next, use `install.github()` to install.

Install Missing R-packages

```
install.packages("devtools")
library(devtools)
install_github(Bioconductor/name_of_package)
install_github(Bioconductor-mirror/name_of_package)
```

5 Running CAMPP

The CAncer bioMarker Prediction Pipeline consists of two R-scripts; CAMPPFunctions.R and CAMPP.R. These scripts must be placed within the same folder and IF the renv libary is used (see section

4 above), the renv.lock file must also be placed in this directory. The pipeline in run from the command-line using flags. The user may find the example files used for running the example below at `https://github.com/ELELAB/N-glycan-TIF` along with the original publication. Files are located in the *Data/DataExamples* folder.

Mac OS X and Ubuntu users may run CAMPP simply with Rscript, while windows users may need an extension, depending on their system.

First Run Example

OS X and Ubuntu:

```
Rscript CAMPP.R
```

Windows:

```
C:\Program Files\R\R-3.5.1\bin\Rscript" CAMPP.R
```

5.1 Mandatory Data Input

1. **Data (`-d`):** An .xlsx (or .txt) file containing feature expression/abundance. With rows as variables, and columns as samples, e.g. columns are N-glycan, protein, (mi)RNA identifiers and rows are sample IDs. The repository `https://github.com/ELELAB/N-glycan-TIF/tree/master/Data/DataExamples` contains an example with N-glycans named: `glycandata.xlsx`. If two datasets are provided (for correlation and/or network analysis), this option should be specified as a comma separated list (without quotes or parenthesis!) of length two, first entry being data file 1 and second entry data file 2.

2. **Data Variant (`-v`):** The user must specify what type of data is provided in order for the pipeline to pick the appropriate normalization and/or transformation. Options include; array (mircoarray data), seq (high throughput sequencing data), ms (mass spectrometry data) or other (other type). If two datasets are provided (for correlation and/or network analysis), this option should be specified as a comma separated list (without quotes or parenthesis!) of length two, first entry referring to data file 1 and second entry referring to the data file 2.

   - Sequencing data: (**-v** is set to **seq**): Variables with low counts over all groups (tissue, treatment) are filered out, library sizes are scaled (normalization method is weighted trimmed mean of M-values, TMM) and data are voom transformed.

   - Microarray data (**-v** is set to **array**): Data are log transformed and either quantile normalized (normalizeBetweenArrays) or standardized using mean or median (specify option **-z**).

- Mass spectrometry data (**-v** is set to **ms**): **IF** option **-t** is specified, then data will be log transformed (log2, log or logit as specified).

> It should be noted that CAMPP does NOT perform within-array-normalization (normalizeBetweenArrays), which is standard for two color intensity data, e.g. this must be done before hand (see limma manual for more information [26]).

3. **Metadata (`-m`):** An .xlsx file (or .txt) containing metadata. This file must contain at least two columns, one with identifiers matching the column names in the data file and one with groups to contrast in analysis e.g. diagnosis (tumor or normal), tumor stage (1,2 or 3), drug treatment (A, B C) ect. If two datasets are provided (for correlation and/or network analysis), this option should be specified as a comma separated list (without quotes or parenthesis!) of length two, first entry being metadata file 1 and second entry metadata file 2. The repository `https://github.com/ELELAB/N-glycan-TIF/tree/master/Data/DataExamples` contains an example with N-glycans named: `glycanmetadata.xlsx`.

   - **Ids and Groups (`-g`):** The user must specify which columns in the `metadata file` corresponds to the sample ids and groups for comparison, receptively. This is done by providing a list of two strings separated by a comma (without quotes or parenthesis!) , indicating the names of the columns which should be used. If two datasets are provided, and both datasets are to be corrected for experimental batch, this option should be specified as a comma separated list (without quotes or parenthesis!) of length four. The first two entries in the list specifying names of columns (ids and groups) to use from metadataset 1 and third and fourth entry specifying names of columns (ids and groups) to use from metadataset 2.

5.2 Arguments

1. **A Simple Run:** To run the pipeline, arguments (flags) must be set to specify the input. In the simplest case we are interested in performing differential abundance/expression analysis and for this we only need to specify the mandatory inputs.

Mandatory Arguments

`-d`:	Quantitative data
`-m`:	Metadata
`-v`:	Data variant
`-g`:	Ids and Groups

We want a custom name for our results, so we set `-n` and specify a sting, in this case we will call it "SimpleRun".

The box below shows an example of the simplest possible CAMPP run.

First Run Example

```
Rscript CAMPP.R -d data.xlsx -m metadata.xlsx -v seq
-g ids,group -n SimpleRun
```

2. The output of the command above will be an .txt (tabular) file with the identifiers, test-statistics, fold changes (logFC) and p-values of any differentially expressed/abundant variables.

Table 2: Example of .txt Output

logFC	t	P.Val	adj.P.Val	B	name	dir.	comp.
2.1	11.6	7.7e-22	4.9e-20	38.9	pA	up	T-N
1.3	8.3	8.4e-14	1.1e-12	20.6	pK	up	T-N
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
-1.2	-10.2	4.1-18	1.3e-16	30.5	pH	down	T-N

> N.B. The pipeline logs while running and produces a text file, `CAMPPlog.txt`, with any errors or comments accumulated during analyses - this file should be viewed carefully.

3. **Help:** In addition to the two mandatory arguments above there are a range of optional arguments which may be utilized. The -h (help) option allows the user to obtain information about all available flags.

CAMPP Help

```
Rscript CAMPP.R -h
```

4. **Other Arguments:** The table below show other available arguments.

Optional Arguments

`-e:`	R-package Management
`-s:`	Multidimensional Scaling Plot
`-t:`	Data Transformation
`-b:`	Batch Correction
`-j:`	Distibutional Checks
`-c:`	Color Scheme
`-r:`	Covariates
`-f:`	Cut-offs for logFC and corrected p-value
`-k:`	Kmeans Clustering
`-l:`	LASSO/Elastic-Net Regression
`-w:`	Weighed Gene Co-expression Analysis
`-o:`	Correlation Analysis
`-u:`	Survival Analysis (Cox Regression)
`-p:`	Protein-Protein Interaction Networks
`-i:`	miRNA-Gene Interaction Networks

- **Renv Library(`-e`):** The flag `-e` may be used to specify that the "freeze" of R-packages should be used instead of the newest versions. If this argument is set to **stable** the renv library will be used. This flag may be set to specify a desired CRAN mirror for "classic" check and installation, default location is *Germany* (see section 4.).
  N.B The user must make sure that the renv.lock file is placed in the same folder as the CAMPP scripts. If `-e` is either, omitted or a CRAN mirror URL is specified, the newest versions of R-packages will be used.

- **Multidimensional Scaling plot (`-s`):** The flag `-s` may be set to generate a preliminary multidimensional scaling (MDS) plot. Multidimensional scaling (MDS) reduces high-dimensional data to two (or more) "observable" dimensions (M1 and M2) in such a way that the inter-sample distance relationship in the simplified dimensions mimic the distance relationship between the original values of samples in n-dimensional space (n = number of variables). MDS was performed with the euclidean distances as the distance metric. A MDS plot may help to determine whether there appears to be any "grouping" of data in a desired way (tumor samples together and normal samples together) or in an undesired way (experimental batches) before performing the DE/DA analysis.

- **Data Transformation(`-t`):** The flag `-t` may be set if the user desires the expression/abundance data to be transformed before analysis. A logarithmic transformation is recommended as the variance of measurements, from most platforms, depend on the expression/abundance level itself. The log-transformation reduces this dependency and additionally pushes the negative binomial distribution, displayed by count data (miRNA, mRNA), towards a normal distribution. The user may choose between log2, log10, logit or voom tranformation. For RNA-seq the voom transformation is recommended [26], while a log transformation may be more appropriate for proteomics and N-glycan abundances. If another sample paired dataset of expression values are provided, this option should be specified as a comma separated list (without quotes or parenthesis!) of length two, first entry referring to data file 1 and second entry referring to the data file 2.

- **Batch (`-b`):** If the data comes from experimental batches and the user wants to correct for this, a column specifying which batch each sample belongs, should also be included in the metadata file. The argument `-b` takes a string (no quotes!) referring to the name of the column in the metadata file denoting batches (e.g. A, B, C, or batch1, batch2, batch3, etc.). Batch type must be noted as a character, meaning numbers alone are not allowed. If two datasets are provided, this option should be specified as a comma separated list (without quotes or parenthesis!) of length two, first entry matching the name of a column in metadata file 1 and the second entry matching the of a column in metadataset 2.

- **Distibutional Checks (`-j`):** The flag `-j` may be set to FALSE to remove the default check of variable distributions. If the flag is not specified (or set to TRUE), the pipeline will produce plots including histograms, quantile-quantile plots and probability plots. By default 10 random variables are picked from the dataframe for plotting (here we are assuming that most variables in an gene expression matrix or protein abundance matrix will belong to the same family of distributions.) **We heavily recommend always running the data checks and to NOT ignore the output, as this vital to whether or not results are reliable!**.

- **Colors (`-c`):** Flag to change group color scheme. Accepted R-colors must be specified in a comma separated list (without parenthesis!) of a length matching the number of groups.

- **Covariates(`-r`):** The user may specify flag `-r` if covariates should be included in the differential expression/abundance analysis and/or the survival analysis. This argument takes a comma separated list (without quotes or parenthesis!). The first element in this list must be either TRUE or FALSE. If TRUE is specified then covariates will be included in both DE/DA analysis and Survival Analsysis. If FALSE is specified covariates will ONLY be used for Survival Analsysis. All other elements of the list after element one (TRUE/FALSE) must be strings matching one or more column names in the metadata file. **Age** is automatically added as a covariate for survival anlysis and should therefore not be specified with `-r`!

- **LogFC and FDR (`-f`):** Cut-offs for log2 fold change and corrected p-value (fdr). Defaults are logFC > 1 or ogFC < -1 and fdr

< 0.05. This argument takes a comma separated list of length two (without quotes or parenthesis!). The first element specifying cut-off for logFC and the second element specifying cut-off for FDR. If two datasets are provided, the list must have length four, e.g. cut-offs for both sets.

- **Kmeans Clustering (`-k`):** The flag `-k` is set to specify K-means clustering. This argument takes a string specifying which column in the metadata file should be used to label the samples in the returned plot(s). If `-k` is set but left empty, no labels are added to the plot(s). The number of clusters tested will be based on number of samples, fewer samples will result in fewer kmeans tested. A folder with MDS plots will be returned for the best n number of kmeans, based on the bayesian information criterion (BIC) [30]. If the dataset has many variables i.e. RNAseq with many genes, multiple samples of 3000 variables will be generated and tested to overcome issues with computational time and the consensus of best n kmeans will be returned.
  Clustering may only be performed one dataset at a time!

- **LASSO/Elastic-Net Regression (`-l`):** The flag `-l` may be set to specify least absolute shrinkage and selection operator (`-l 1.0`) or Elastic-Net (`0.0 < -l < 1.0`) regression. LASSO/EN is performed using the R-package **glmnet** [12]. K-fold (default is 10) cross validation (cv.glmnet) is used to estimate the optimal value for the hyperparameter, lambda. LASSO/EN may be performed in two ways, (I) the dataset is split into training and testing subsets, k-fold cross validation is performed on the training dataset, followed by estimation of specificity and sensitivity (area under the curve = AUC) using the test dataset, or (Il) k-fold cross validation is performed using the full dataset, no AUC is reported. CAMPP will automatically estimate whether the input dataset is large enough to split into training and test subsets and whether EN/LASSO is advisable to perform altogether. Specifying the flag `-l` will produce a list of variables selected by LASSO and a file with the overlap between differential expression/abundance analysis and LASSO regression. LASSO is run with n (default is 10) different random seeds and the consensus set of variables is returned.
  N.B LASSO it not appropriate for heavily unbalanced group designs! When running LASSO a bar-plot with cross-validation errors are returned, as well as an area under the curve (AUC), IF the dataset is large and balanced enough to split into test and training set. LASSO may only be performed on one dataset at a time!

> **N.B** It is important to note setting the flag (`-l`) is a trade-off between retaining information in the regression model without merely including all (or almost all) variables (this happens if `-l` is very low). As the point of elastic-net regression is to aid the user in picking a smaller set of markers for validation from a potentially large set of differentially expressed/abundant variables, we suggest to set `-l` to between 0.5-0.9,

or simply perform LASSO regression, i.e. `-l` is set to 1.0.

- **Weighed Gene Co-expression Network Analysis (`-w`):** The flag `-w` must be set in order to perform Weighed Gene Co-expression Network Analysis with the R-package WGCNA [23]. WGCNA, despite its name, is in this case not exclusive to gene expression data but may be applied to any quantitative expression data. Minimum module size is 10 variables and modules with less than 25% dissimilarity will be merged (default values). The cutoff for top most interconnected variables (genes, proteins ect.) in an identified module is set to the 75th quantile by default. The user may specify different cutoffs for minimum module size, module dissimilarity merging and and % of interconnected variables to report with the flag `-x`.
  N.B the softpower plot generated by a WGCNA run should be inspected before interpretation of results. If the data is very heterogeneous, or if there are too few variables or samples (minimum 15 samples for WGCNA) the scale-free topology fit index (printed to the screen while running) might fail to reach values above 0.8 for reasonable powers (see point 5 in WGCNA tutorial: `https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/faq.html`). If this is the case, the dataset may not be appropriate for WGCNA.
  WGCNA may only be performed with one dataset at a time!

- **Correlation Analysis (`-o`):** The flag `-o` must be set to perform correlation analysis between two matched datasets. For option `-o` the user must provide a string specifying which subset of variables should be included in the correlation analysis; "ALL" = all variables (not advisable, unless dataset is small), "DA" / "DE" = Differentially Expressed/Abundant, "LASSO" / "EN" = LASSO / Elastic Net results or "Consensus" = Overlap between DE and LASSO/EN. Naturally two datasets for must be input in order to perform correlation analysis. Both datasets are given to the argument (-d), with the names of the files separated by a comma (no quotes or parenthesis!). The two dataset do not need to have the same dimensions, but there must be at least a partial overlap in both variables and samples (column names). Column names of these datasets should match the IDs in the `ids` column in the two metadata files.

- **Survival Analysis (`-u`):** The flag `-u` must be set in order to perform survival analysis using cox proportional hazard model. Options for survival analysis are; "ALL" (not advisable!), "DA" / "DE" = Differentially Expressed/Abundant, "LASSO" / "EN" = LASSO / Elastic Net results or "Consensus" = Overlap between DE and LASSO/EN, referring to the set of variables used for cox-regression. For survival analysis the metadata file must contain at least three columns in addition to the sample IDs named;

'outcome.time' (time until end of follow-up, censuring or death in weeks, months or years), 'outcome' (numeric 0 = censuring, 1=dead) and 'survival' (numeric 0 = no survival info, 1=survival info available). If the user wishes to correct for potential confounders (e.g. patient age, tumor grade, hormone levels, drug-treatment ect.) these should also be included in the `metadata.xlsx`. CAMPP checks two underlying assumptions of the cox model before performing survival analysis (I) a linear relationship of continuous covariates with log hazards and (II) proportional hazards of categorical and continuous covariates, e.i. constant relative hazard. If the requirement of linearity is not fulfilled, cubic splines will be added to the covariate(s) in question.

- **P-P Interactions (`-p`):** Flag `-p` may be specified to perform protein-protein interaction network analysis. Input for this argument is a comma separated list of length two (without quotes or parenthesis!), where the first element specifies the type of gene IDs in the data file, accepted IDs are: uniprotswissprot, ensembl_peptide_id, hgnc_symbol, ensembl_gene_id or ensembl_transcript_id. The second element specifies the protein-protein interaction database to use, currently the only supported database is STRING [20], accepted inputs are: stringdatabase.

- **Gene-miRNA Interactions (`-i`):** Flag `-i` may be specified to perform gene-miRNA interaction network analysis. Input for this argument is a comma separated list of length two (without quotes or parenthesis!), where the first element specifies the type of miRNA IDs in the data file, accepted IDs are: mature_mirna_ids or mature_mirna_accession. The second element specifies the Gene-miRNA interaction database to use (validated, predicted or both), accepted inputs are: targetscan, mirtarbase or tarscanbase.

  N.B: If both `-p` and `-i` are set, CAMPP will integrate P-P and Gene-miR networks. Naturally two data files must be provided to use both `-p` and `-i`, it is assumed that data file 1 contains genes and data file 2 contains miRNAs!

The next page contains an elaborate table (Figure 2) which goes through most of the plots generated by the CAMPP pipeline. In this table the user may find information about in which type of analysis the plot is produced, what it shows and how to interpret it.

6 N-glycan Serum Markers for BC Diagnostics

This section contains an example of how CAMPP is run using different flags in the command-line and what output the user can expect. The files used for running this example may be found in the repository `https://github.com/ELELAB/N-glycan-TIF/tree/master/Data/DataExamples`.

Analysis	Name of Plot	About	X-axis	Y-axis	Color Scheme	Interpretation
Preliminary Data Overview and Normalization	Multidimentional Scaling (MDS) plot	Data overview. The plot shows the relationship between samples (squared euclidian distance) in the dataset. Each dot represents one sample.	Multidimentional Scaling component 1 (M1). The component which best captures the distance relationship (squared euclidian) between samples in n-dimensional space (n = number of variables) decomposed into one dimension.	Multidimentional Scaling component 2 (M2). The component which captured most of the distance relationship (squared euclidian) between samples in n-dimensional space (n = number of variables) decomposed into one dimension, second to M1.	Color scheme denotes the groups provided to CAMPP for DE/DA analysis.	The plot helps the user to evaluate if differentially expression/abundance analysis of sample subsets, is likely to be fruitful. The plot may also reveal if technical variability (i.e. batch effects) is present in the data, in which case the user should run the pipeline with batch correction flag -b. The plot may highlight if Kmeans clustering could be a useful analysis to perform, that is, if the user observes a clustering of samples which cannot be explained by any known sample co-variates, clinical information or batch effects.
	Frey and Cullen graph	The Cullen and Frey graph represents the skewness and kurtosis of a distribution. The skewness refers to the symmetry of the distribution (skewness of a normal distribution = 0) and the kurtosis indicates whether a distribution is "heavy-tailed" or "light tailed" (kurtosis of a normal distribution = 3). Each symbol/line represents a different standard distribution. The star symbol = normal distribution.	Skewness.	Kurtosis.	Blue dot = skewness and kurtosis of dataset, green dots = bootstrapping samples of skewness and kurtosis.	We expect the blue dot to fall within the green cloud of bootstrapping values, and importantly close to the star symbol (normal distribution). \\ For more information: \url{https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm}.\\ If this it not the case, check whether the argument for data variant is set correctly, and subsequently if a log2 transformation of the data will solve the issue.
	Quantile-Quantile (QQ) plot	Check for normality.	Empirical quantiles.	Theoretical quantiles.	NA	Data are normally distributed if the empirical quantiles calculated from the observed data, display a linear relationship with the theoretical quantiles obtained from a standard normal distribution with mean 0 and deviation 1.
	Cumulative Distribution plot	Check for normality.	Empirical cumulative distribution function.	Theoretical cumulative distribution function.	NA	Data are normally distributed if the empirical cumulative distribution function based on the observed data, converges on the cumulative distribution function from a standard normal distribution with mean 0 and deviation 1.
Differnetial Expression/ Abundance Analysis	Heatmap	Heatmap of differentially expressed/ abundant (DE/DA) variables and how they separate samples into subsets.	Each column on the x-axis represents one sample. Names of samples are not included in the plot. The dendrogram at the top of the plot shows the similarity (clustering) of samples based on the variables (genes, proteins or other) in the heatmap.	Each row on the y-axis represents one variable (gene, protein or other). Names of variables are included.	The color bar at the top of the heatmap denotes which groups specified for DE/DA analysis, a given sample belongs to. Colors of the heatmap itself indicates the expression/abundances level and directionality of variables included in the plot between groups. Levels of variables are normally annotated as log2 fold change (logFC). The color key at the bottom of the plot specifies how the color gradient goes from high to low expression/ abundance.	This plot is a graphical represetation of how "good" a given variable (gene, protein or other) or set of variables are at partitioning samples belonging to two or more groups of interest (i.e. cancer vs healthy, condition A vs B vs C, etc.)
Elastic Net / LASSO Regression	Cross Validation Error plot	Bar plot, showing cross-validation errors from 10 elastic net /LASSO runs. Each bar denotes the CV-error from one run. This error is calculated using out-of-bag samples.	Runs 1-10.	Cross-validation error.	NA	If the cross-validation errors are large (> 5-10%) and/or very varying, this indicates that the dataset is either small with a large variance or that it encompasses some structure which makes it very sensitive to removal of certain samples. In either case, the EN/LASSO regression is not trust-worthy and should be discarded.
	Venn Diagram	The venn diagram shows the overlap of variables (genes, proteins or other) found to be both DE/DA and retained in the LASSO/elastic net regression model. DE/DA analysis variables are split into up-regulated and down-regulated set. Numbers in the circles denote the number of variables in the sets	NA	NA	One color refers to one set, either up-regulted in DE/DA analysis, down-regulated in DE/DA analysis or identified by elastic net regression.	If there is no overlap of results from DE/DA analysis and elastic net, this might indicate (I) that the DE/DA analysis is "bad" (check the data distribution plots and potential batch effects), (II) that the elastic-net regression is "bad" (check the plot of cross-validation errors and the AUC if it is returned) or (III) that the dataset has an extremely high variance, compared to number of samples
Weighed Gene Co-expression Analysis (WGCNA)	Dendrogram	Dendogram showing the clustering (similarity) of modules from WGCNA. Each color block represents one module. A module represents a set of co-expressed/ abundant variables, and the dendrogram shows the relationship between these sets.	NA	NA	The dendrogram is used to see how many sets of co-expressed/abundant variables exist in the, different sizes of these sets and their relationship. The information on which variables were both in a module and DE/DA may be seen in the .txt files returned from the WGCNA.	
	Interconnectivity plot(s)	Bar plots, one for each module returned by WGCNA. One bar plot contins the top 25% (default cut-off, see manual or help -h) most interconnected genes from a given module. The hight of the bar indicates how connected a given variable (gene, protein or other) is, i.e. the taller the bar, the more interconnected a gene is. The variable with the tallest bar may be considered a hub gene/protein.	Name of variables (genes, proteins, other).	Interconnectivity scores returned by WGCNA. Read more about interconnectivity in original article: Langfelder, Peter, and Steve Horvath. "WGCNA: an R package for weighted correlation network analysis." BMC bioinformatics 9.1 (2008): 559.	Bars are coloured in the same color as the corresponding module from dendogram.	The plots provide insights into which variables in each module are not only co-expressed but also highly interconnected, potentially revealing variables (genes/ proteins) in specific pathways which are co-expressed/abundant, co-regulated and central to disease pathology. Comparing these plots to .txt files returned by WGCN-analysis with information on which variables from a module were also DE/DA, may point to subsets of genes specific to patient subgroups like cancer subtypes, grade, stage, treatment, etc.
	Heatmap(s)	Heatmaps showing the level of co-expression between variables in a module from WGCNA. These plots are only generated for small modules (< 200 variables), as larger heatmaps are impossible to read.	Variables (genes, proteins, other).	Variables (genes, proteins, other).	Colors indicate strength of co-expression (weighed correlation) ranging from yellow (weak) —> red (strong).	May be used to identify subsets of the most co-expressed variables within a module.
Correlation Analysis	Dot plot	This plot shows the correlation coefficients (spearman) between the same variables from two dataset (e.g. tissue and plama, condition A and condition B, etc.). One dot responds to one variable. Horizontal lines denote a correlation coefficient of 0.5 and -0.5. Correlations are tested for significance and p-values are adjusted for multiple testing (FDR). If there are many variables, this plot will be partitioned into subplots.	Variables.	Correlation coefficients (spearman).	Colors of dots denote the invers FDR, i.e. the smaller the FDR the larger and darker the color value.	This plot helps the user to identify variables which have correlated expression/abundance across different starting materials, conditions or treatments. Correlation analysis may be performed using all variables or only variables from DE/DA analysis, elastic net regression or consensus of these (see manual for specifics). Correlated variables will be dark in color and located above the top horizontal line (positively correlated) or below the bottom horizontal line in the plot (negatively correlated).
	Scatter plot(s)	Individual scatter plots are generated for any variable found to be significantly correlated between to datasets (see Dot plot above). Each dot represents one sample. The blue/red lines are the best fitted linear regression to the data points. The gray shade is the confidence interval which will be narrow in areas of the plot with many points and wide were there are few points.	Expression/abundance values of the variable in dataset 1.	Expression/abundance values of the variable in dataset 2.	Blue line = positive correlation, red line = negative correlation. Gray shade = confidence interval.	The idea behind these individual scatter plots is to visually inspect variables identified as correlated between two datasets. Some variables may have good correlation scores and significant FDR values due to single data points, while there is an overall high uncertainty of correlation for a range of values. This uncertainty will be highlighted in these scatters, helping the user to evaluate the trustworthiness of each correlated variable.
Survival Analysis	Stem plot	Dot plot with confidence intervals (stem plot). This plot is returned from the survival analysis (cox proportional hazard regression). Each dot is the hazard ratio (HR) of a variable on the log2 scale, with accompanying 95% confidence intervals. The size of the dot denotes the size of the HR.	Hazard Ratios on a log2 scale. log2(1) = 0.	Variables (genes, proteins, other).	The color indicates whether a variable was significantly associated with an increased or decreased risk of death by disease (black), or if the variable was not significant (white).	An HR = 1.0 indicates that a variable is not association with patient survival. A HR > 1.0 means an increased risk of outcome (death) is associated with a high/low level of a variable. A HR < 1.0 means a decreased risk of outcome (death) is associated with a high/ low level of a variable. As the HR are on a log2 scale, log2(1.0) = 0.0, log2(2.0) = 1. and log2(0.5) = -1.
Interaction Network Analysis	Arcdiagram plot	Plot depicting protein-protein (gene-gene) and mRNA-miRNA interactions of DE/DA variables. Each dot denotes a variable, while each line (arch) represents a interaction (predicted or supported). This plot can only be generated for miRNA and/or proteins/genes. The size of dot represents the number of interactions a variable has, i.e. larger dot = more interactions.	NA	NA	Colors denote directionality of expression/ abundance of a variable, obtained from the DE/ DA analysis.	The plot shows which DE/DA protein-protein (gene-gene) and miRNA-mRNA pairs are either known to interact (experimentially validated) or predicted to interact (miRNA-mRNAs based on seed region). This plot helps to elucidate which of the candidates from DE/DA analysis and elastic-net regression interact with each other, pointing to subsets of variables involved in the same biological processes and pathways.

The case below uses N-glycans abundances measured using high resolution quantitative Ultra-Performance Liquid Chromatography (UPLC) [28] from interstitial samples [15] and matched serum. Tumor interstitial fluid (TIF), normal interstitial fluid (NIF) and serum samples were collected from $\sim$ 90 women diagnosed with breast cancer (BC). A total of 165 N-glycan groups were identified [32].

Briefly, the involvement of N-glycosylation in development and progression of BC has been documented by both in vitro and in vivo studies [3, 5, 16]. Several circulating N-glycan patterns with altered glycan structures, possibly originating from a primary tumor or from other organs, in response to a neoplastic process, have recently been described in a number of studies by using high-throughput N-glycan profiling [2, 18, 25, 28].

6.1 Data Normalization, Transformation and Distributional Checks

If an input dataset contains missing values CAMPP will automatically impute these, unless missing per column > 70%. Depending on which data input is given CAMPP will perform data normalization and transformation **(argument -v, data variant, see section on mandatory input)**. The quantitative N-glycan data (used in this example) are relative (fractional) meaning that the sum of all values within one sample yields 100. Abundances of N-glycans were quantified using liquid chromatography tandem mass spectrometry (LC-MS/MS). The dataset had already been standardized by the MS-software, therefore the option **-v** was set to **ms** and the option -t was set to log2. As the N-glycan data were quantified over three LC-MS/MS runs, the argument **-b** was set, e.g. data were corrected for experimental batch.

Before performing any analysis it is advisable to evaluate the distribution of the normalized data. CAMPP automatically generates distributional plots for n (default is 10) randomly selecting input variables for the user to evaluate - to skip this step the argument **-j** may be set to FALSE).

Output:

1. A 5-panel plot for each of the n (default is 10) randomly selected variables. The plot contains, quantile -and probability plots, a histogram of expression/abundance values with fitted distributions and a skewness-kurtosis plot (Cullen and Frey graph, see fitdistrplus manual [8]) - See **Figure 2**.

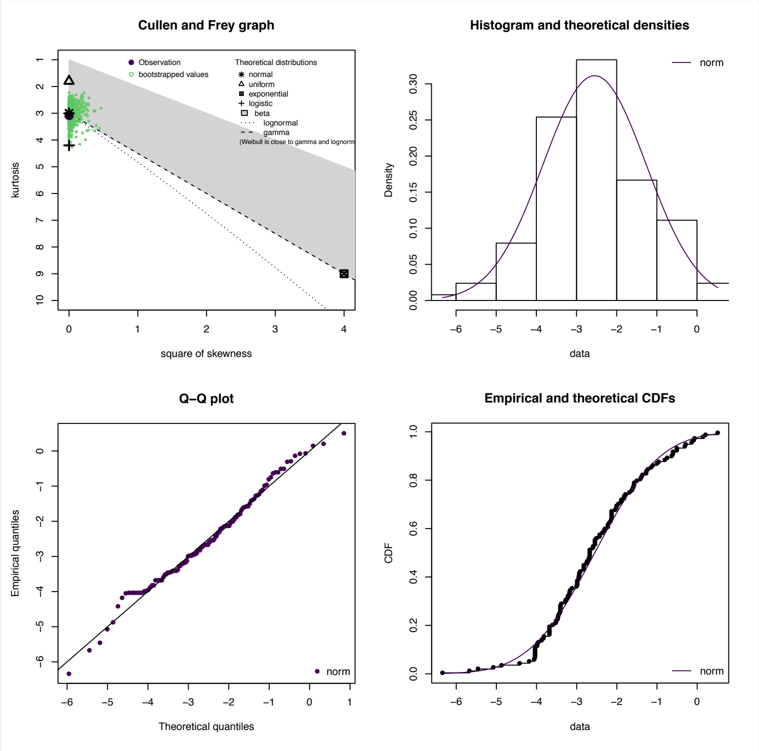


Figure 2: Data distributional check for N-glycan GP13 (randomly selected from dataset).

Based on all four plots above, log2 transformed GP13 abundances appear to follow a normal distribution. In brief, we expect to see a bell curve shape for the histogram, and a diagonal scatter for the qqplot. The Cullen and Frey graph says something about the skewness and kurtosis of a distribution. The skewness here refers to the symmetry of the distribution (skewness of a normal distribution = 0) and the kurtosis indicates whether a distribution is "heavy-tailed" or "light tailed" (kurtosis of a normal distribution = 3). As seen from the plot above both the normal distribution as well as other types of distributions are included in the graph. We expect the variable of interest (blue dot) to fall within the green cloud of bootstrapping values, and close to the star symbol (normal).

For more information: `https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm`.

Although the statistical framework behind limma is robust to some divergence of data from the expected normal distribution, the user should be careful with interpretation of results if data checks fail, especially for all/most of the variables checked. In this case the first thing is to check whether the argument for data variant `-v` is set correctly (seq, array, ms) and subsequently if a log transformation of the data will solve the issue, see section on argument `-t`.

6.2 Differential Abundance Analysis and LASSO/Elastic-Net Regression

Differential Abundance Analysis and LASSO regression was run with correction for experimental batch (p1, p2, p3), in this particular case the column name specifying batch (in metadata file) was "pool". The data were log2 transformed (flag `-t`) and a preliminary MDS plot was generated (flag `-s`.) The cut-off for a significant hit was an FDR <= 0.05 (default) - no logFC cut-off was set to define N-glycan differential abundance (flag `-f`). The `-l` was set to 1.0, specifying LASSO (for Elastic-Net `0.0 < -l < 1.0`). Files used for this example may be found at `https://github.com/ELELAB/N-glycan-TIF/tree/master/Data/DataExamples`.

The box below shows an example of DAA and LASSO regression with CAMPP.

Differential Abundance Analysis.

```
Rscript CAMPP.R -d glycandata.xlsx -m glycanmetadata.xlsx
-v ms -g sids,cn -t log2 -b pool -l 0.5 -s TRUE
-f 0,0.05 -n FirstRunOutput
```

Output:
The run above generates three outputs:

1. A multidimensional scaling plot (`FirstRunOutput_MDSplot.pdf`) using the abundance/expression of variables (in this case N-glycans). The components M1 and M2 in the plot below are those which best retained the distance relationship between samples in two dimensions. - See **Figure 3**.
2. A tabular .txt file (`FirstRunOutput_DE.txt`) with the stats for significant differentially abundant/expressed variables (in this case N-glycans).
3. A heatmap (`FirstRunOutput_heatmap.pdf`) visualizing the power of the identified DA/DE variables (in this case N-glycans) for partitioning of the samples in groups/condition/treatment (in this case NIF and TIF samples)- See **Figure 4.**
4. A tabular .txt file (`FirstRunOutput_LASSO.txt`) with the LASSO selected variables for group classification.
5. A tabular .txt file (`FirstRunOutput_DEA_LASSO_Consensus.txt`) containing the overlap of DAA results and LASSO results.
6. Bar-plot (`FirstRunOutput_CrossValidationPlot.pdf`), with cross-validation errors from each LASSO run. - **Figures not shown.**
7. A .txt file (`FirstRunOutput_AUC.txt`, with the area under the curve (AUC) IF the dataset was large enough and groups balanced enough to split the dataset into testing and training set. CAMPP will automatically decided if the dataset can be split.

If we wanted to use the "freeze" of R-packages, instead of the newest versions (see section on CAMPP installation) we would add the option (–e) and set this to **stable**, see below.

Differential Abundance Analysis.

```
Rscript CAMPP.R -d glycandata.xlsx -m glycanmetadata.xlsx
-v ms -g sids,cn -t log2 -b pool -l 0.5 -s TRUE
-f 0,0.05 -e stable -n FirstRunOutput
```

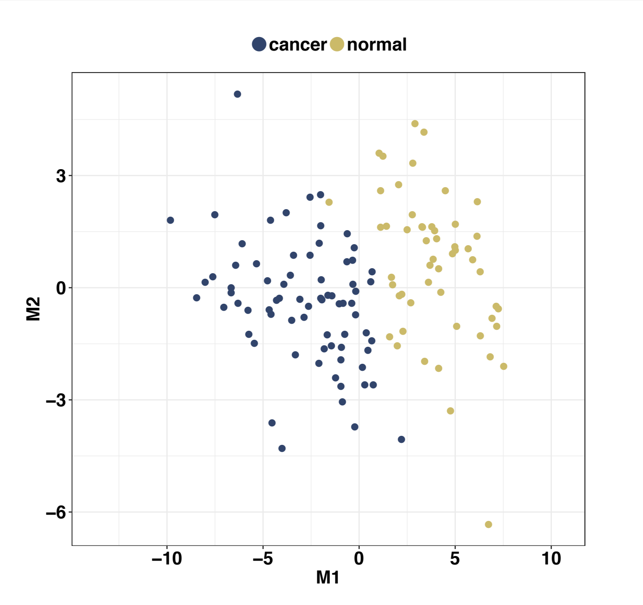


Figure 3: Multidimensional Scaling Plot showing the separation of tumor interstitial and normal interstitial fluid samples based on N-glycan abundances. The data was batch corrected before plotting.

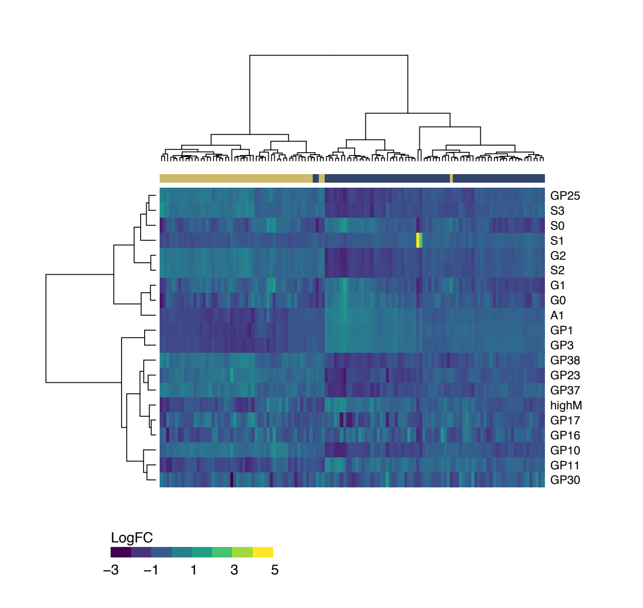


Figure 4: Heatmap showing the partitioning of TIF and NIF samples based on 20 N-glycan peaks/groups identified as differentially abundant.

The multidimensional scaling plot in **Figure 3** indicates that N-glycan abundance patterns may confer a separation of tumor interstitial fluid and normal interstitial fluid samples from patients with breast cancer - at least when all N-glycans variables are retained. The MDS plot may be used as an indicator of whether the differential expression analysis is likely to yield any results, e.g. no clustering of samples from different groups/conditions/treatments implies that the biological data does NOT reflect the sample classification.

6.3 Correlation of abundances in TIF and serum

In order to determine whether N-glycans with differential abundances in tumour -and normal interstitial fluids displayed corresponding patterns in serum, correlation was performed using TIF samples and matched serum.

For correlation analysis with CAMPP, the user must provide:

1. Two .xlsx (or .txt) files with expression/abundance data (rows as variables and columns as samples). See section 5.2.4 for specifications. Examples may be found in the repository `https://github.com/ELELAB/N-glycan-TIF/tree/master/Data/DataExamples`.
2. Two .xlsx (or .txt) files with metadata for each dataset. Examples may be found in the repository `https://github.com/ELELAB/N-glycan-TIF/tree/master/Data/DataExamples`.
3. A string specifying which subset of variables to use for correlation analysis, options are; "ALL", "DE" / "DA", "LASSO"/ "EN" or "Consensus" - see section X.

Files used in the example below, can be found in the repository `https://github.com/ELELAB/N-glycan-TIF/tree/master/Data/DataExamples`. Command-line box 6.2.3 shows an example of a CAMPP correlation analysis run with TIF and serum N-glycan abundances. Both data and serumdata were produced by tandem mass spectrometry and -v was therefore set to ms,ms. Also, both sets were log2 transformed (argument -t) before analysis. Only differentially abundant N-glycans where included in the correlation analysis As serum data were not produced in batches argument `-e` was unneeded.

Correlation Analysis.

```
Rscript CAMPP.R -d glycandata.xlsx, glycanSdata.xlsx
-m glycanmetadata.xlsx, -m glycanSmetadata.xls
-b pool -v ms,ms -g sids,cn,sids,cn -t log2,log2 -o DA
-n SecondRunOutput
```

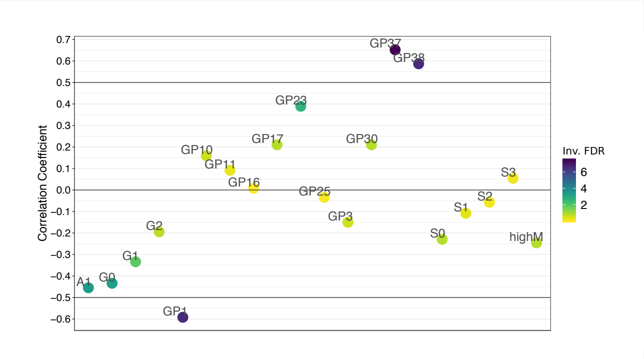


Figure 5: Correlation plot depicting correlation coefficients associated with abundances of N-glycans in TIF and matched serum. Correlation was performed with the 20 N-glycans identified as DA in normal and tumor fluids. Dots are colored in accordance with inverse (scaled) FDRs, e.g. darker shade indicates smaller adjust p-value and vice versus.

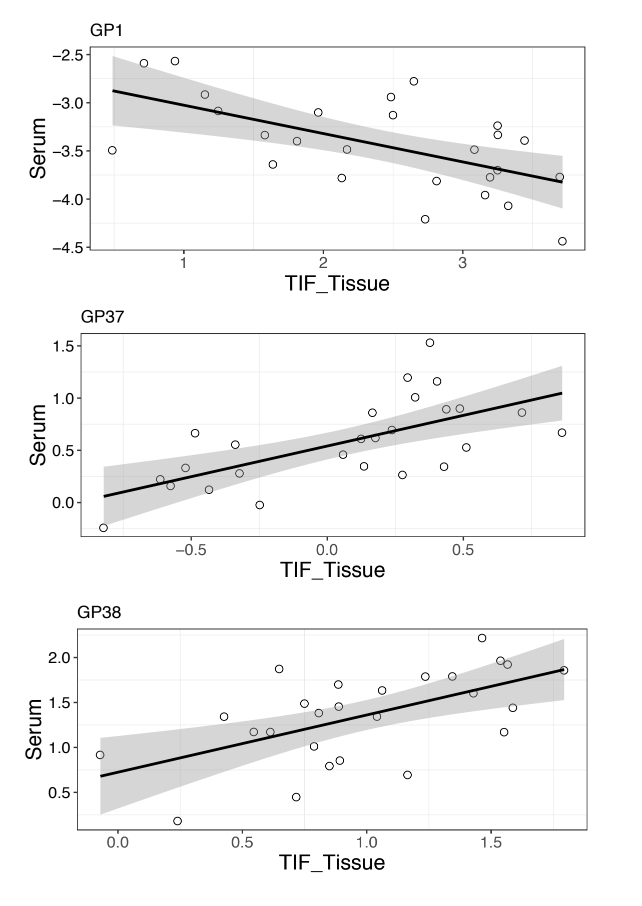


Figure 6: Scatter plots of the 3 N-glycans (GP1, GP37 and GP38) with significant correlations between abundances in TIF and serum. Shaded area indicates confidence of regression line.

Output:
The run above generates three outputs:

1. A tabular .txt file (`SecondRunOut_corr_serum_.txt`) with the stats for variables (in this case N-glycans) displaying significant correlation of abundance/expression in tumor sample (interstitial fluid) and matched serum.

2. A correlation plot (`SecondRunOutput_corrplot.pdf`) visualizing the correlation coefficients and adjusted p-values of all differentially expressed/abundant variables - in this case N-glycans found to partition TIF and NIF. See **Figure 5**.

3. Individual scatter plots (`SecondRunOutput_individual_corrplots.pdf`) for the variables displaying a significant correlation between expression/abundance in TIF/tissue and serum (FDR <= 0.05 is default). See **Figure 6**.

Based on the .txt file with coefficients and adjusted p-values (not shown), three N-glycans, GP1, GP37 and GP38 where found to display significant correlations of abundances in TIF and serum (see Figure 5). It is worth noting that levels of GP1 in TIF and serum are negatively correlated indicating an accumulation of this N-glycan within the tumor interstitium, perhaps as a result of primary tumor response. GP37 and GP38 have positive correlation coefficients and may be considered as potential diagnostic serum markers.
Figure 6 shows the scatter plots (TIF abundance plotted against serum abundance) of GP1, GP37 and GP28, with confidence shading of regression lines.

6.4 Weighed Gene Co-expression Network Analysis

In order explore the abundance relationship between N-glycans in an unsupervised way, Weighed Gene Co-expression Network Analysis was performed with WGCNA [23]. The cutoff for top most interconnected variables (genes, proteins ect.) in an identified module is set to 25% by default. The user may specify a different cutoff with the flag `-x`. Cutoff for module similarity merging is 0.25 and cutoff for minimum module size is 10 (defaults from the WGCNA tutorial here: `https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/faq.html`). These default parameters cannot be changed with a flag but may easily be modified in the source code of the CAMPP.R script if desired.

Weighed Gene Co-expression Network Analysis.

```
Rscript CAMPP.R -d glycandata.xlsx -m glycanmetadata.xlsx
-v ms -g sids,cn -t log2 -b pool -w TRUE -n ThirdRunOutput
```

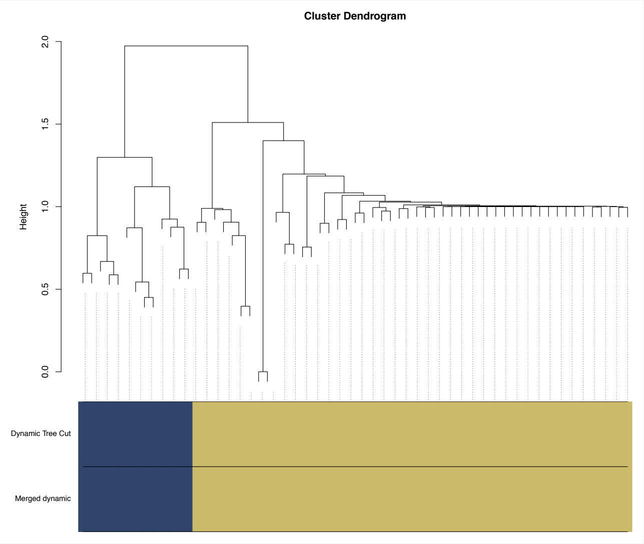


Figure 7: WGCNA module dendogram showing hclust of N-glycans and which modules each N-glycan was assigned to. The plot shows the modules, in this case two, before and after merging on module similarity. As similarity was not enough to merge, the two original modules were retained in this example.

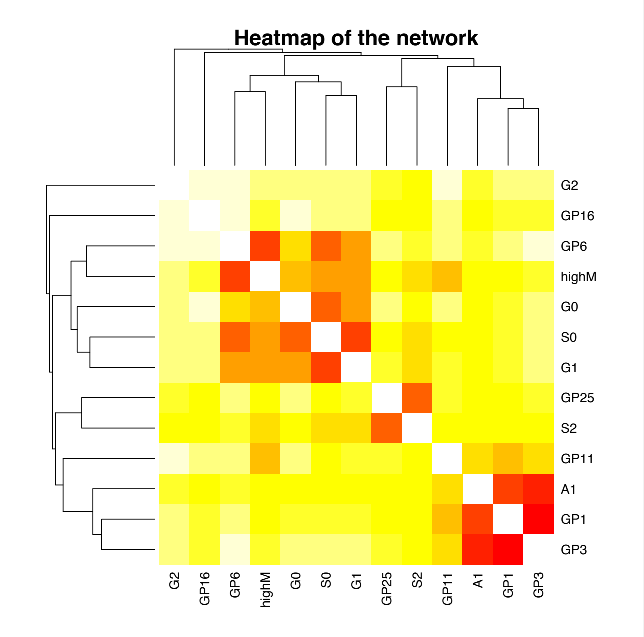


Figure 8: Heatmap of the blue module (heatmap of turquoise module not shown).

Output:
The run above generates three outputs:

1. Tabular .txt files, one for each module (`ThirdRunOut_X_moduleRes.txt`), with the stats for variables (in this case N-glycans) which were the among the top n % (default = 25%) most interconnected variables within a module, e.g. those with the highest kWithin values.
2. WGCNA dendogram, e.g. clustering of variables (N-glycans), with modules colored before and after mergining (`ThirdRunOut_WGCNA_moduleTree Tree.pdf`). See **Figure 7**.
3. Barplots, one for each module, with the top most interconnected variables in a module (default = 25%). The hight of a bar denotes how interconnected a variable (gene, protein, etc.) is with other variables in the module (`ThirdRunOut_moduleIC.pdf`).
4. Heatmaps for small modules (< 100 variables), colored according to variable (N-glycan) co-expression (`ThirdRunOut_moduleHM.pdf`). CAMPP does not return heatmaps from large modules as these become unreadable. See **Figure 8**.

The heatmap in **Figure 8** below shows that within the blue module N-glycan features A1, GP1 and GP3 were the most highly co-expressed, followed by N-glycan features S0, G1, GP6 and highM. These results are consistent with both the results of the TIF and serum correlation analysis (**Figure 5**), was well as the variables selected from LASSO regression and differential expression analysis.

6.5 Survival Analysis

Cox proportional hazard model was used to assess whether any differentially abundant N-glycans had potential as prognostic biomarkers for the survival of breast cancer patients.
CAMPP only provides cox proportional hazard models for overall survival and simple relapse-free survival, e.g. one time entry per patient/sample.
CAMPP checks two underlying assumptions of the cox model before performing survival analysis (I) a linear relationship of continuous covariates with log hazards and (II) proportional hazards of categorical and continuous covariates, e.i. constant relative hazard. If the requirement of linearity is not fulfilled, cubic splines will be added to the covariate(s) in question, and analysis is continued. If the proportional hazard assumption is violated for a categorical variable the user may apply stratification and re-run the pipeline. However, the pipeline does not currently handle continuous variables with non-proportional hazards, e.i. that the user should ignore any cox models where this is the case.

Mandatory Columns for Survival Analysis (metadata.xlsx):
In order to perform cox proportional hazard regression with CAMPP,

the user must ensure that the metadata file contains at least three columns in addition to the sample IDs named; 'outcome.time' (time until end of follow-up, censuring or death in weeks, months or years), 'outcome' (numeric 0 = censuring, 1=dead) and 'survival' (numeric 0 = no survival info, 1=survival info available). If the user wishes to correct for potential confounders (e.g. patient age, tumor grade, hormone levels, drug-treatment ect.) these should also be included in the `metadata.xlsx`. The flag `-u` is set to "DA" specifying that the variables which were differentially expressed should be used for survival analysis, alternatives are; "LASSO", "EN" or "Consensus".

Files used in the example below, can be found in repository `https://github.com/ELELAB/N-glycan-TIF/tree/master/Data/DataExamples`. Command-line box 6.3.2 shows an example of a CAMPP survival analysis run. TIF N-glycan abundances were corrected for experimental batch and log2 transformed before regression analysis.

Survival Analysis.

```
Rscript CAMPP.R -d glycandata.xlsx -m glycanmetadata.xlsx
-v ms -g sids,cn -t log2 -b pool -u DA -n FourthRunOutput
```

The example above shows the simplest version of a cox model which does not include any covariates/confounders, however these may be included with the flag **`-r`**. If multiple confounders are added these should be separated by commas and their names should match those of the desired columns within `metadata.xlsx`. **N.B** the first argument specified in the `-r` list should always be either TRUE or FALSE. TRUE means that covariates should be included both in the design matrix for differential expression analysis and survival analysis, whereas FALSE specifies that covariates should only be used for cox regression. In the run below, survival analysis was performed with tumor infiltrating lymphocyte status (name: TILS, type: categorical) and tumor size (name: TSize, type: continuous) as covariates - See the `metadata.xlsx` sheet in the `DataExampels` folder.

> **N.B** It is important to note that each time a covariate is added to the model we increase the degrees of freedom. It is therefore essential to have a dataset of an appropriate size for the number of covariates added. The good rule of thumb is 10 events per parameter/variable.

Survival Analysis (confounders).

```
Rscript CAMPP.R -d glycandata.xlsx -m glycanmetadata.xlsx
-v ms -g sids,cn -t log2 -b pool -u DA -r FALSE,age,TILs
-n FourthRunOutput
```

If a CAMPP run produces the warning message below, it means that some of the specified covariates violate the proportional hazard assumption:

```
WARNING: The following variables and/or covariates failed the test of
proportional hazard:  TILS.
IF the covariates that failed are categorical you may use strata by
re-running the pipline adding flag -y followed by the names of the
categorical covariates to stratify (if multiple, separate by comma).
N.B, this pipeline does not handle continuous variables that violate
the proportional hazard assumption, if any of these failed PH test,
the hazard ratios of these should NOT be evaluated.
```

If the s are categorical we can re-run the pipeline with stratification using the flag **-y**:

Survival Analysis (confounders and stratification).

```
Rscript CAMPP.R -d glycandata.xlsx -m glycanmetadata.xlsx -v ms
-g sids,cn -t log2 -b pool -u DA -r FALSE,age,TILs -y TILs
-n FourthRunOutput
```

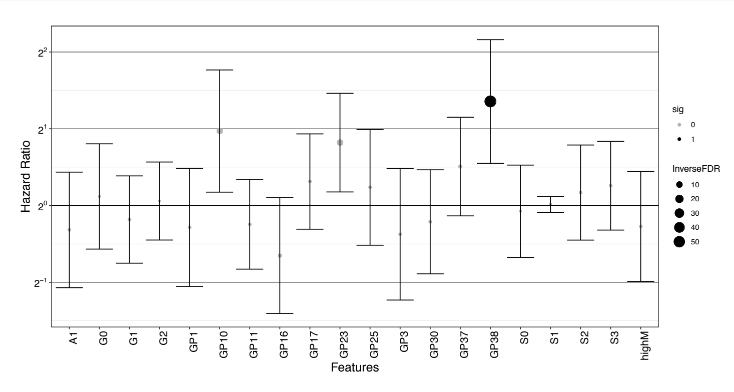


Figure 9: Summary plot of hazard ratios (and confidence intervals) for cox regression with each of the 20 N-glycans found to be differentially abundant between TIF and NIF. A hazard ratio (HR) < 1 indicates that a high level of a given N-glycan is associated with a positive outcome (longer overall survival after diagnosis), while a HR > 1 suggests that a high level of a given N-glycan predicts poorer overall survival. The dot size denotes inverse FDR, e.g. a larger dot is associated with a smaller FDR. Significant N-glycans (FDR <= 0.05) have blue dots, while non-significant N-glycans have orange dots.

Output:
The runs above generate two outputs:

1. A tabular .txt file (`FourthRunOut_survival.txt`) with the stats for variables (in this case N-glycans) displaying significant associations between abundance/expression in tumor sample (interstitial fluid) and patient outcome.
2. A plot of the hazard ratios for variables (in this case N-glycans) with confidence intervals and significance (`FourthRunOut_survivalplot.pdf`). See **Figure 8**.

Figure 9 shows that one N-glycan, GP38, was found to be significantly associated with survival, e.i. a high level of this N-glycan was predictive for poor overall survival. GP38 was one of the three N-glycans displaying a correlation between abundances in TIF and serum, suggestion that this glycan structure may have potential as a prognostic serum biomarker.

6.6 Protein-Protein / Gene-miR Interaction Networks

CAMPP may be used to perform protein-protein and/or gene-miRNA interaction network analysis. Interactions are returned for differentially expressed genes/miRNAs. In this case LASSO/EN may not be specified instead, as this type of analysis does not yield the statistics necessary for the analysis.

P-P interactions are extracted from the STRING database [20], with a lower score cut-off > 25th quantile. MiRNAs are retreived using the multimiR R-package [27], the user may specify whether to use predicted miRNA targets (TargetScan, lower score cut-off > 25th quantile), validated gene-miRNA pairs (miRTarBase) or both of these. In the following section we given an example of how to run this analysis with miRNA and gene expression data. Here we cannot use the N-glycan dataset as N-glycans are neither genes nor miRNAs. Instead we will show a fictive example. Note that gene -and miRNA identifiers must be in the approved list of inputs - See specifics under the Arguments section above.

The box below shows an example of protein-protein interaction network analysis with CAMPP. Here, the data file contains gene symbols and so the first element in the list given to argument *-p* is hgnc_symbol. The second element is the p-p interaction database, in this case the STRING database.

Protein-Protein Network Interaction Analysis.

```
Rscript CAMPP.R -d Genedata.xlsx -m Genemetadata.xlsx
-v seq -g gID,gCN -b gBatch
-p hgnc_symbol,stringdatabase -n FifthRunOut
```

The box below shows an example of gene-miRNA interaction network analysis with CAMPP. Here, the data file contains miRNA IDs and so the first element in the list given to argument *-i* is mature_mirna_ids. We decide that we want predicted gene-miRNA interaction pairs, so we set the second element in list *-i* to targetscan.

Protein-Protein Network Interaction Analysis.

```
Rscript CAMPP.R -d miRNAdata.xlsx -m miRNAmetadata.xlsx
-v seq -g miRID,miRCN -b miRBatch
-i mature_mirna_ids,targetscan -n FifthRunOut
```

The box below shows an example of CAMPP run where both gene expression data and paired miRNA expression data are available. Here, we can integrate protein-protein and gene-miRNA interactions into one network.

Protein-Protein Network Interaction Analysis.

```
Rscript CAMPP.R -d Genedata.xlsx,miRNAdata.xlsx
-m Genemetadata.xlsx,miRNAmetadata.xlsx
-v seq,seq -g gID,gCN,miRID,miRCN -b gBatch,miRBatch
-p hgnc_symbol,stringdatabase -i mature_mirna_ids,targetscan
-n FifthRunOut
```

Output:
The runs above will return the following outputs:

1. A tabular .txt file (`FifthRunOut_AllInteractions.txt`) which contains all interaction pairs with accompanying logFCs, FDRs and interaction scores. This file may be used to visualize networks in Cytoscape or with another similar tool. If more than two groups where contrasted in the DE/DA analysis, then a file for each pairwise contrast will be returned.

2. A plot of the top 100 most i significant interactions, based on absolute logFCs and inter-connectivity. (`FiftRunOut_TopInteracions.pdf`). If more than two groups where contrasted in the DE/DA analysis, then a plot for each pairwise contrast will be returned. See **Figure 10**.

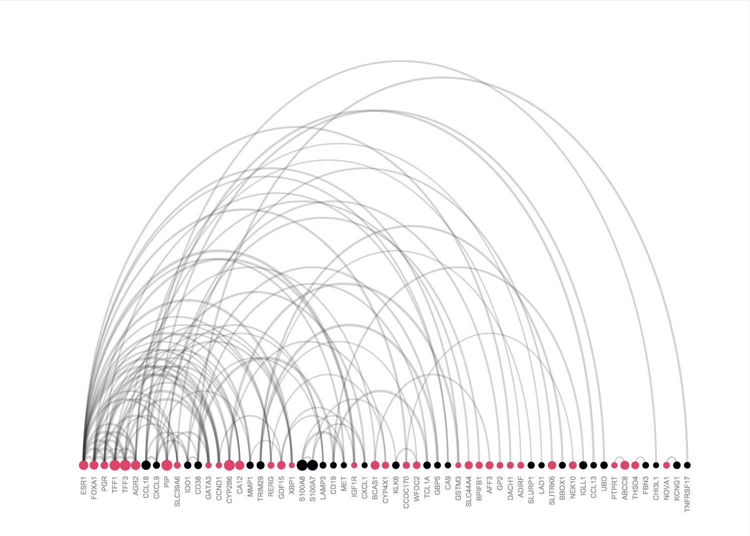


Figure 10: Top 100 most significant interactions from Protein-Protein interaction Network Analysis. Color denotes directionality of genes in the comparison. Red = up and Dark Purple = down. Nodes are ordered according to number of interactions. Size of node is the absolute logFC and thickness of arch is the interaction score.

7 BC-related Gene Markers - RNA Sequencing Data

In the example below we use CAMPP to analyse RNA sequencing data from The Cancer Genome Atlas (TCGA, GDC data portal). We downloaded and compiled breast cancer data into the data format accepted by CAMPP, i.e. a table (.xlsx or .txt) of count data and a clinical file with sample information. The dataset contained 416 samples and 55150 mRNA transcripts (some isoforms of the same gene). After running the pipeline one time with only mandatory parameters, we observed what could be a batch effect in the data. As such, we performed two subsequent runs with two different potential batch covariates, sample plate and tissue source site (tss), available from the clinical metadata. The code box below shows how this was accomplished for tss, using the covariate as group in the -g option and *Figure 11* depicts the MDS plots coloured according to either (A) condition (cancer and normal) or (B) tissue source site. We specify custom colors with the argument *-c*. The figure shows that there is bias in tss as almost all normals are collected from one tss (BH), as such we decide to corrected for this as a batch effect.

7.1 Batch Effects and Data Overview

Multidimensional Scaling - Breast Cancer Data from TCGA

```
(A) Color by cancer and normal (contrast of interest):

Rscript CAMPP.R -d BRCA_data.txt -m BRCA_clinical.txt
-v seq -g ids,cn -c "#FE5F55","#7FC6A4" -n SimpleRun1

(B) Color by tissue source site (potential batch effect):

Rscript CAMPP.R -d BRCA_data.txt -m BRCA_clinical.txt
-v seq -g ids,tss -c "#90323D","#475B5A","#6D213C","#0B3142",
"#F78764","#86DEB7","#AAA4C4","#083D77" -n SimpleRun2
```

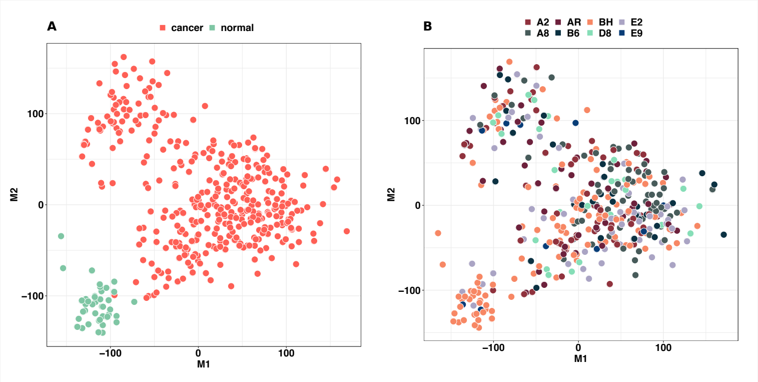


Figure 11: Figure showing MDS plots from two CAMPP runs above. A = colored by condition(cancer or normal) and B = colored by tissue source site (potential batch effect).

7.2 Differential Expression Analysis and Elastic-Net Regression

In the CAMPP run below we apply all types of analysis to the 416 x 55150 gene count matrix. For a dataset this size WGCNA will be slow and memory consuming (approx. 20 GB) to run on a small laptop, due to the topology matrix calculation - we therefore restrain this analysis to include differentially expressed genes only. Although WCGNA can in principle be applied to a dataset with 50.000 variables or more, the user should consider whether this analysis makes biological sense and whether the output can be interpreted in a meaningful way. If we are interested in identifying cancer biomarkers, the co-expressed variables of greatest interest are likely those which are also differentially expressed between sub-groups of patients. There can of course be scenarios in which is make sense to perform WGCNA on all data, an if the dataset is "small", the process it will be quick and the results fewer and more easily digestible, so in this case the full dataset can be input. In the CAMPP run shown below, we set the cut-off for differential expression (DE) to log fold change of 2 and an FDR < 0.01 (*-f* 2,0.01), and we performed EN regression (*-l* 0.5).

CAMPP Run with Breast Cancer Data from TCGA - Tumour vs Normal is contrasted.

```
Rscript CAMPP.R -d BRCA_data.txt -m BRCA_clinical.txt
-v seq -g ids,cn -b tss -f 2,0.01 -l 0.5 -w DE
-a Consensus -s TRUE -k "" -n FullRunSeq
```

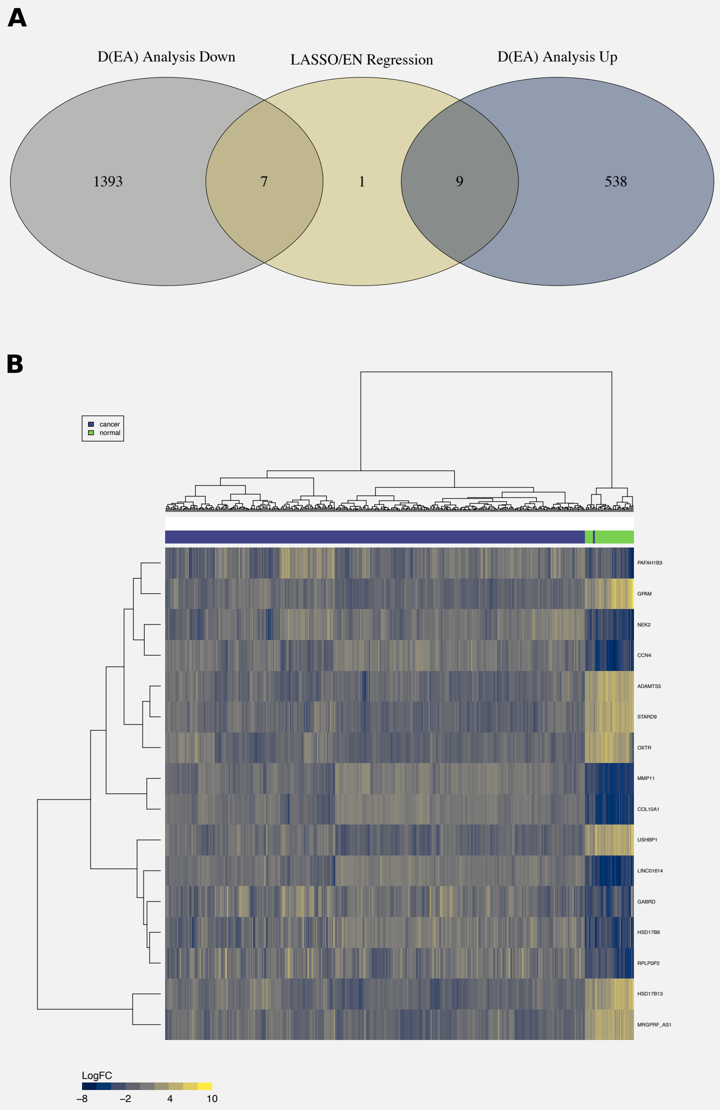


Figure 12: A = Overlap of genes from differential expression analysis and elastic net regression from TCGA breast cancer dataset (416 samples, 55150 genes). A total of 16 genes were shared between DEA and EN. B = Heatmap showing expression of these 16 genes and how they discriminate between breast tumor and normal tissues.

With these criteria we obtained 1944 DE genes and 17 classifier genes from EN, out of which 16 were identified by both approaches, see *Figure 12, Panel A*. The 16 genes retained from the consensus of DEA and EN contained genes previously related to breast cancer development and progression *Figure 12, Panel B*. Among those with well-known oncogenic properties were NEK2 [39], COL10A1 [40], MMP11 [41], CCN4/WISP1 [42] and KIF4A [44], which in accordance with their respective functions were up-regulated in breast cancer tissue vs normal paired tissues. A literature search revealed three genes to be directly associated with anti-tumorigenic properties; tumor suppressors PAMR1 [45] and ADAMTS5 [46] and anti-metastasis modulator PPP1R12B [47], all of which had significantly lower expression levels in BC vs normal samples.

Kmeans clustering returned six clusters *Figure 13, Panel A*. After inspection of the labels, these clusters seem to somewhat fit with annotated breast cancer subtype (Luminal A, Luminal B, Basal-like and Her2-enriched), with Basal-like samples displaying the most distinct expression profile. We therefore re-ran the pipeline using subtype as the group contrast of interest. We pick LASSO regression (not elastic net) to restrict the number of results returned from regression, as LASSO with multiple groups often results in more variables (-*l* 1.0). We also leave out the k-means clustering, as we already know what clusters we are interested in.

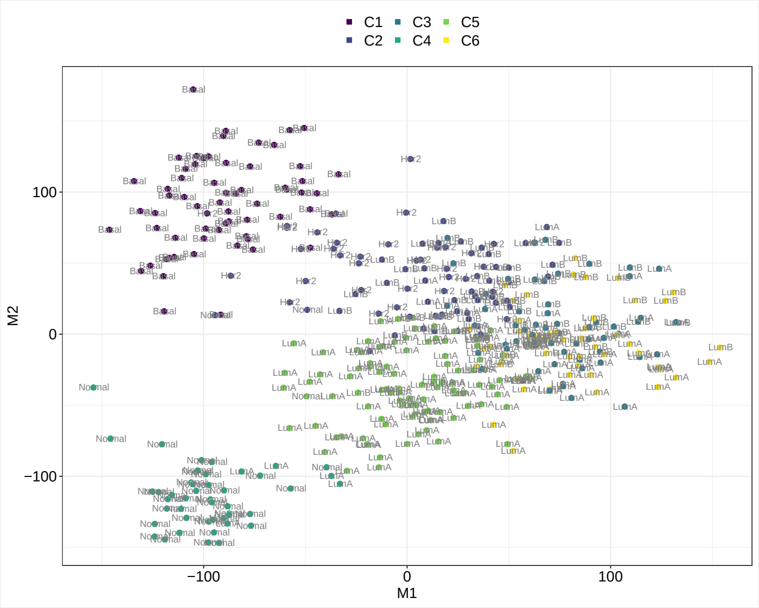


Figure 13: MDS plot, colored by kmeans clusters (six in total) and labelled by BC subtype.

CAMPP Run with Breast Cancer Data from TCGA - Subtype are contrasted.

```
Rscript CAMPP.R -d BRCA_data.txt -m BRCA_clinical.txt -v seq
-g ids,subtype -b tss -f 2,0.01 -l 1.0 -w DE -n FullRunSeq
```

Let's first have a look at the LASSO results. The cross-validation errors from the 10 LASSO runs with different seeds ranged from 1.0-5.0 % -

see *Figure 14*. In total 13 genes were returned by LASSO regression, out of which 12 were also differentially expressed. The combined area under the curve (AUC) for these was **0.98**. I quick look into these 12 gene signature revealed that it encompassed four pam50 genes: ESR1, FOXC1, GRB7 and NAT [48].

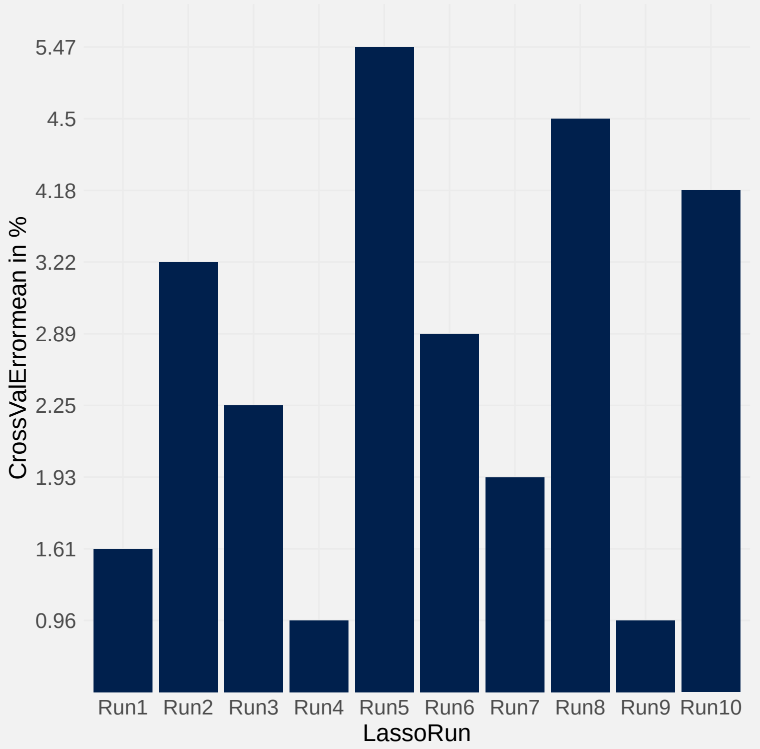


Figure 14: Cross-validation error plot from 10 LASSO run with breast cancer subtype (Luminal A, Luminal B, Her2-enriched and Basal-like) as response variables. Y-axis shows cross validation error in %.

> **N.B** It is important that the LASSO/EN cross-validation errors are low (< 10 %) for most of the 10 runs. If the cross validation plot reveals very different error rates, this might indicate that the dataset is unstable and/or uninformative, which could either be inherent for the data or it could be related to missing or inappropriate data normalization. In this case we advise the user against trusting the results of LASSO/EN regression.

7.3 WGCNA with Differentially Expressed Genes

Next, we look at the results of the weighed gene co-expression network analysis. If WGCNA is carried out using DE/DA variables only, then co-expression modules will be output with information about in which contrast(s) variables in a module were identified as differentially expressed/abundant. For DE/DA analysis with more than one pairwise

group comparison, each module is output in an individual file.
In the plot below is the dendrogram (panel A) from WGCNA of the TCGA BC dataset, and the interconnectivity plot of the pink module (panel B).

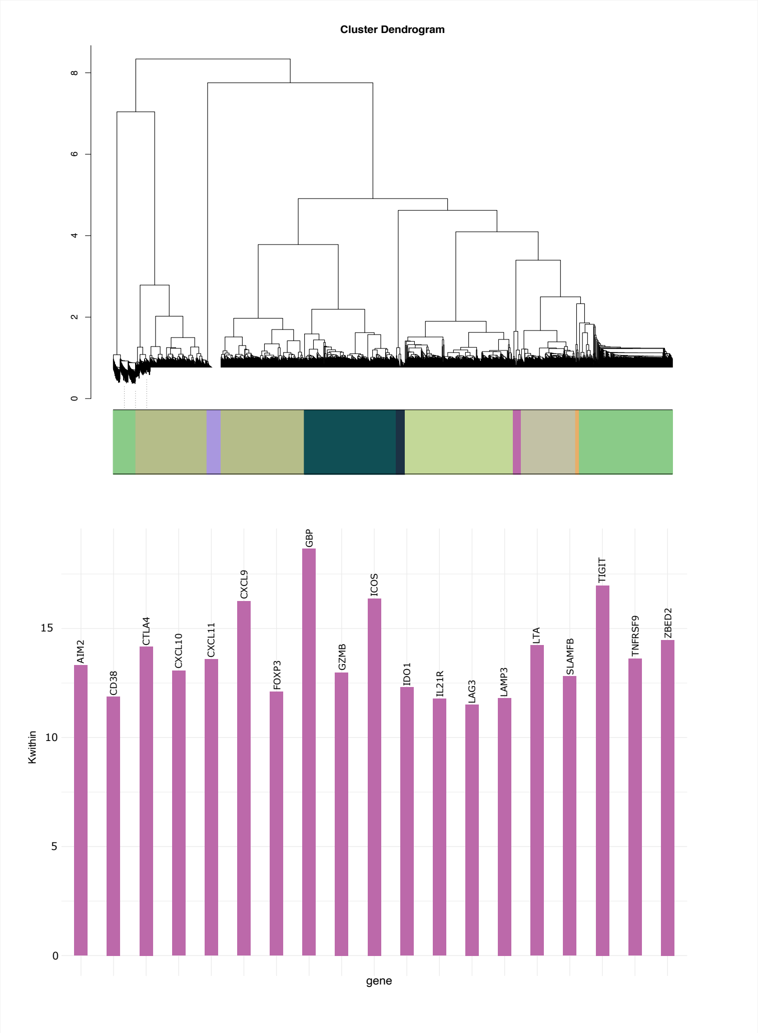


Figure 15: Panel A = dendrogram of similarity/clustering of nine co-expression modules from WGCNA. Panel B = Pink module of co-expressed gene. X-axis has the name of the gene and y-axis/height of bar is interconnectivity score of a gene.

By inspection of the accompanying .txt file from the analysis, we see that most of the genes in the pink module were up-regulated in Basal and Her2 breast cancer vs Luminal types and normal samples. The most interconnected gene in this module was GBP5 (Guanylate Binding Protein 5), followed by; TIGIT (T Cell Immunoreceptor With Ig And ITIM Domains), CXCL9 (C-X-C Motif Chemokine Ligand 9) and ICOS (Inducible T Cell Costimulator).

We copy the gene names from this file (first column) and put them into STRING [49] (https://string-db.org/). STRING [49] analysis revealed that all the genes in the pink module are known to interact with each other, and are involved in positive regulation of immune system processes.

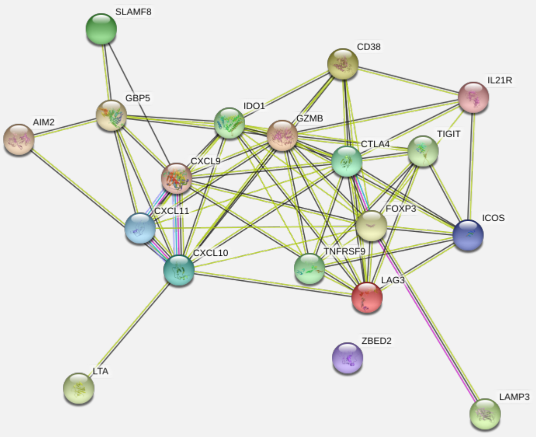
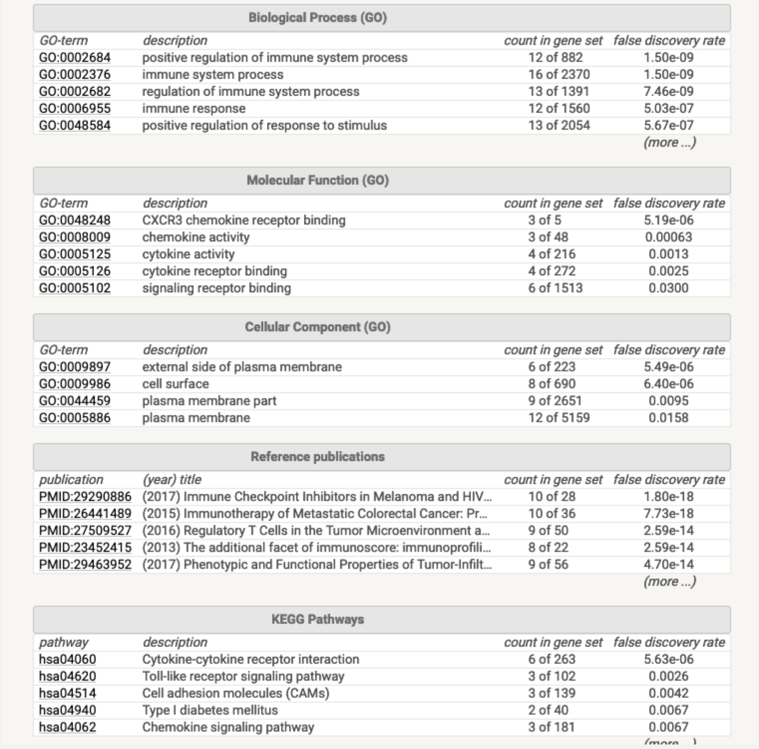


Figure 16: String network of differentially co-expressed genes from the pink module.



Biological Process (GO)

GO-term	description	count in gene set	false discovery rate
GO:0002684	positive regulation of immune system process	12 of 882	1.50e-09
GO:0002376	immune system process	16 of 2370	1.50e-09
GO:0002682	regulation of immune system process	13 of 1391	7.46e-09
GO:0006955	immune response	12 of 1560	5.03e-07
GO:0048584	positive regulation of response to stimulus	13 of 2054	5.67e-07

(more ...)

Molecular Function (GO)

GO-term	description	count in gene set	false discovery rate
GO:0048248	CXCR3 chemokine receptor binding	3 of 5	5.19e-06
GO:0008009	chemokine activity	3 of 48	0.00063
GO:0005125	cytokine activity	4 of 216	0.0013
GO:0005126	cytokine receptor binding	4 of 272	0.0025
GO:0005102	signaling receptor binding	6 of 1513	0.0300

Cellular Component (GO)

GO-term	description	count in gene set	false discovery rate
GO:0009897	external side of plasma membrane	6 of 223	5.49e-06
GO:0009986	cell surface	8 of 690	6.40e-06
GO:0044459	plasma membrane part	9 of 2651	0.0095
GO:0005886	plasma membrane	12 of 5159	0.0158

Reference publications

publication	(year) title	count in gene set	false discovery rate
PMID:29290886	(2017) Immune Checkpoint Inhibitors in Melanoma and HIV...	10 of 28	1.80e-18
PMID:26441489	(2015) Immunotherapy of Metastatic Colorectal Cancer: Pr...	10 of 36	7.73e-18
PMID:27509527	(2016) Regulatory T Cells in the Tumor Microenvironment a...	9 of 50	2.59e-14
PMID:23452415	(2013) The additional facet of immunoscore: immunoprofili...	8 of 22	2.59e-14
PMID:29463952	(2017) Phenotypic and Functional Properties of Tumor-Infilt...	9 of 56	4.70e-14

(more ...)

KEGG Pathways

pathway	description	count in gene set	false discovery rate
hsa04060	Cytokine-cytokine receptor interaction	6 of 263	5.63e-06
hsa04620	Toll-like receptor signaling pathway	3 of 102	0.0026
hsa04514	Cell adhesion molecules (CAMs)	3 of 139	0.0042
hsa04940	Type I diabetes mellitus	2 of 40	0.0067
hsa04062	Chemokine signaling pathway	3 of 181	0.0067

(more ...)

Figure 17: Gene ontology (GO) terms and pathway enrichment results from STRING [49] analysis.

References

[1] R Core Team. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/. (2014).

[2] Abd Hamid, U.M et al. *A strategy to reveal potential glycan markers from serum glycoproteins associated with breast cancer progression.* Glycobiology 18, (2008): 1105–1118.

[3] Abbott, K.L. et al. *Targeted glycoproteomic identification of biomarkers for human breast carcinoma.* Journal of Proteome Research 7, (2008):1470–1480.

[4] Castello, A., et al. *Insights into RNA biology from an atlas of mammalian mRNA-binding proteins.* Cell 149.6 (2012): 1393-1406.

[5] Christiansen M.N et al. *Cell surface protein glycosylation in cancer.* Proteomics 14, (2014):525–546.

[6] Csardi G. anf Nepusz T. *The igraph software package for complex network research* InterJournal, Complex Systems 1695. 2006. http://igraph.org.

[7] Day, A. *heatmap.plus: Heatmap with more sensible behavior.* R package (2012) version 1.3. https://CRAN.R-project.org/package=heatmap.plus

[8] Delignette-Muller, M. L., and Dutang, C. *fitdistrplus: An R package for fitting distributions.* Journal of Statistical Software 64.4 (2015): 1-34.

[9] Dowle, M. and Srinivasa, A. *data.table: Extension of 'data.frame'.* R package (2019) version 1.12.0. https://CRAN.R-project.org/package=data.table

[10] Durinck, S. et al. *Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt.* Nature Protocols 4, 1184-1191 (2009).

[11] Eklund, A. C. *squash: Color-Based Plots for Multivariate Visualization.* R package (2017) version 1.0.8. https://CRAN.R-project.org/package=squash

[12] Friedman, J., Hastie, T., and Tibshirani, R. *Regularization paths for generalized linear models via coordinate descent.* Journal of statistical software 33.1 (2010): 1.

[13] Fultz, N. and the StackOverflow.com community. *stackoverflow: Stack Overflow's Greatest Hits.* R package (2018) version 0.3.0. https://CRAN.R-project.org/package=stackoverflow

[14] Garnier, S. *viridis: Default Color Maps from 'matplotlib'.* R package (2018) version 0.5.1. https://CRAN.R-project.org/package=viridis

[15] Gromov, P. et al. *Tumor interstitial fluid—a treasure trove of cancer biomarkers*. Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics 1834.11 (2013): 2259-2270.

[16] Guo, H.B. et al. *Specific posttranslational modification regulates early events in mammary carcinoma formation*. Proceedings of the National Academy of Sciences 107, (2010): 21116–21121.

[17] Hastie, T. et al. *impute: impute: Imputation for microarray data.* R package (2018) version 1.56.0.

[18] Haakensen, V.D. et al. *Serum N-glycan analysis in breast cancer patients – Relation to tumour biology and clinical outcome*. Journal of Molecular Oncology, (2015):1–14.

[19] Jeffrey T. et al. *sva: Surrogate Variable Analysis.* R package (2019) version 3.30.1.

[20] Jensen, L. J., et al. *STRING 8 - a global view on proteins and their functional interactions in 630 organisms.* Nucleic acids research 37 (suppl-1), D412-D416, (2008).

[21] Kammers, K., et al. *Detecting significant changes in protein abundance.* EuPA open proteomics 7 (2015): 11-19.

[22] Kassambara, A. and Kosinski, M. *survminer: Drawing Survival Curves using 'ggplot2'.* R package (2018) version 0.4.3. https://CRAN.R-project.org/package=survminer

[23] Langfelder, P. and Horvath, S. *WGCNA: an R package for weighted correlation network analysis.* BMC Bioinformatics (2008), 9:559.

[24] Pineda, A.L., et al. *On Predicting lung cancer subtypes using 'omic'data from tumor and tumor-adjacent histologically-normal tissue.* BMC cancer 16.1 (2016): 184.

[25] Potapenko, I.O. et al. *Glycan-related gene expression signatures in breast cancer subtypes; relation to survival.* Journal of Molecular Oncology 9, (2013): 861–876.

[26] Ritchie, M.E., et al. *limma powers differential expression analyses for RNA-sequencing and microarray studies.* Nucleic Acids Research 43(7), e47.

[27] Ru, Y. et al. *The multiMiR R package and database: integration of microRNA–target interactions along with their disease and drug associations.* Nucleic Acids Res 42 (17) (2014). e133. doi: 10.1093/nar/gku631

[28] Saldova, R., et al. *Association of N-glycosylation with breast carcinoma and systemic features using high-resolution quantitative UPLC..* Journal of proteome research 13.5 (2014): 2314-2327.

[29] Schroeder MS., et al. *survcomp: an R/Bioconductor package for performance assessment and comparison of survival models.* Bioinformatics 27(22): 3206-3208. (2011)

[30] Scrucca et al. *mclust 5: clustering, classification and density estimation using Gaussian finite mixture models.* The R journal 8.1 (2016): 289.

[31] Soneson, C., and Delorenzi, M. et al. *A comparison of methods for differential expression analysis of RNA-seq data.* BMC bioinformatics 14.1 (2013): 91.

[32] Terkelsen, T. and Haakansen, V.D., et al. *N-glycan signatures identified in tumor interstitial fluid and serum of breast cancer patients: association with tumor biology and clinical outcome.* Molecular oncology 12.6 (2018): 972–990.

[33] Ushey, K., et al. *renv: Project Environments.* 2019, R package version 0.8.3. https://CRAN.R-project.org/package=renv

[34] Walker, A. *openxlsx: Read, Write and Edit XLSX Files.* R package version 4.1.0. https://CRAN.R-project.org/package=openxlsx

[35] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, (2016).

[36] Wickham, H. *The Split-Apply-Combine Strategy for Data Analysis.* Journal of Statistical Software, 40(1), 1-29, (2011). URL: http://www.jstatsoft.org/v40/i01/.

[37] Wickham, H. *scales: Scale Functions for Visualization.* R package (2018) version 1.0.0. https://CRAN.R-project.org/package=scales.

[38] Wickham, H., et al. *devtools: Tools to Make Developing R Packages Easier.* R package version 2.0.1. (2018). https://CRAN.R-project.org/package=devtools

[39] Fang, Y., and Zhang, X. *Targeting NEK2 as a promising therapeutic approach for cancer treatment.* Cell Cycle 15.7 (2016): 895–907.

[40] Chapman, K. B., et al. *COL10A1 expression is elevated in diverse solid tumor types and is associated with tumor vasculature.* Future oncology 8.8 (2012): 1031-1040.

[41] Zhang, X., et al. *Insights into the distinct roles of MMP-11 in tumor biology and future therapeutics.* International journal of oncology 48.5 (2016): 1783–1793.

[42] Chiang, K.C., et al. *WNT-1 inducible signaling pathway protein-1 enhances growth and tumorigenesis in human breast cancer.* Scientific reports 5 (2015): 8686.

[43] Wu, Y., et al. *ASF1a inhibition induces p53-dependent growth arrest and senescence of cancer cells.* Cell death and disease 10.2 (2019): 76.

[44] Xue, Dan, et al. *An integrated bioinformatical analysis to evaluate the role of KIF4A as a prognostic biomarker for breast cancer.* OncoTargets and therapy 11 (2018): 4755.

[45] Lo, P. H Y., et al. *Identification of novel epigenetically inactivated gene PAMR1 in breast carcinoma.* Oncology reports 33.1 (2015): 267-273.

[46] Huang, J., et al. *ADAMTS5 acts as a tumor suppressor by inhibiting migration, invasion and angiogenesis in human gastric cancer.* Gastric Cancer 22.2 (2019): 287-301.

[47] Ding, C., et al. *The PEAK1–PPP1R12B axis inhibits tumor growth and metastasis by regulating Grb2/PI3K/Akt signalling in colorectal cancer.* Cancer letters 442 (2019): 383-395.

[48] Parker, J. S., et al. *Supervised risk predictor of breast cancer based on intrinsic subtypes.* Journal of clinical oncology 27.8 (2009): 1160.

[49] Szklarczyk, D. et al. *The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible* Nucleic acids research (2016): gkw937