Gene set-focused analysis of RNA-seq data with MIEP

Corradin, A., Ciminale, V., *MIEP: Make-it-easy-pipeline*, useR! 2024, July 8-12 2024, Salzburg, Austria

Corradin, A., et al., *Gene set-focused analysis of RNA-seq data with MIEP*, Computational Intelligence methods for Bioinformatics and Biostatistics 2024, September 4-6 2024, Benevento, Italy (submitted)

What is MIEP?

Make-it-easy-pipeline (MIEP)

- is an integrated, interactive, and user-friendly R package;
- aggregates a high number of tools for the analysis of RNA-seq data;
- develops/edits new gene sets based on gene ontology enrichment tests and variable importance derived from random forests;
- MIEP R package is composed of more than R 200 scripts.

Where is it?

master | 1 Branch | 0 Tags | Go to file | Add file | Code

AlbertoCorradinPhD Update README.md 1a03086 · now 3 Commits

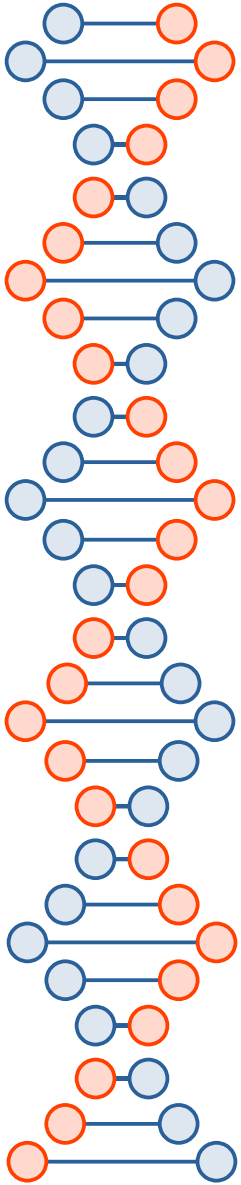
R	first commit	2 days ago
build	first commit	2 days ago
inst	first commit	2 days ago
man	first commit	2 days ago
vignettes	first commit	2 days ago
DESCRIPTION	first commit	2 days ago
NAMESPACE	first commit	2 days ago
README.md	Update README.md	now

README

MIEP

to install MIEP and all its dependencies clone the repository and use the following command: devtools::install(".../Downloads/MIEP", dependencies=TRUE, repos=c(BiocManager::repositories(),'https://github.com/AckerDWM/gg3D'), build=TRUE, build_vignettes=TRUE)

https://github.com/AlbertoCorradinPhD/MIEP



The idea

Pathway disruption plays a critical role in cancer development.

This affects the signaling mechanisms that rule cell function.

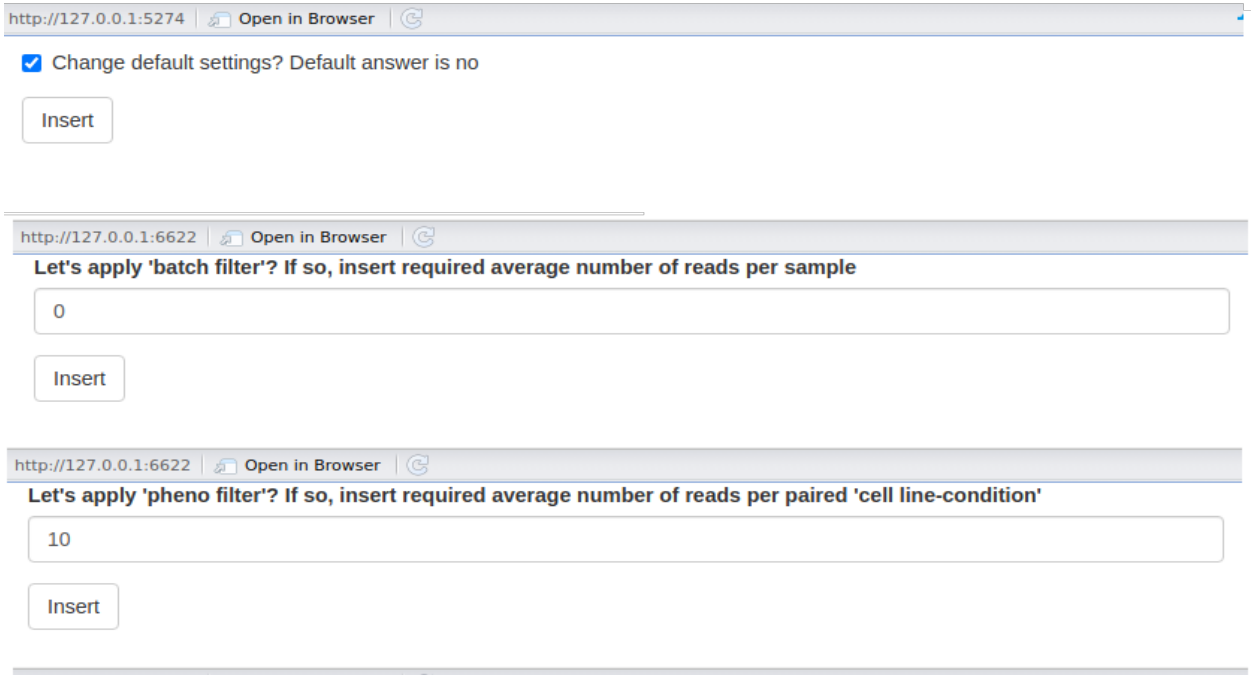
By focusing on gene sets mimicking pathways, we can accelerate the development and testing of treatments.

(supposing these latter act on key elements of signal transduction)

The pipeline in pictures

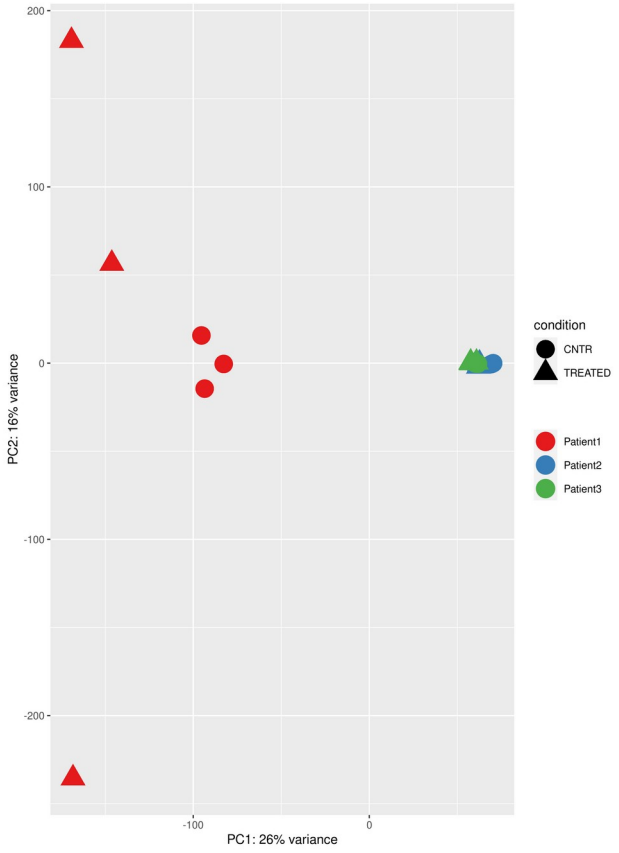
- Pre-filtering low count genes
- Check if data are biased
- Data shrinkage
- Differentially expressed features
- Annotations
- Features selection
- Dimensionality reduction
- Enrichment of Gene Ontology terms
- Conditional random forests
- Focusing on single gene sets

Pre-filtering low count genes



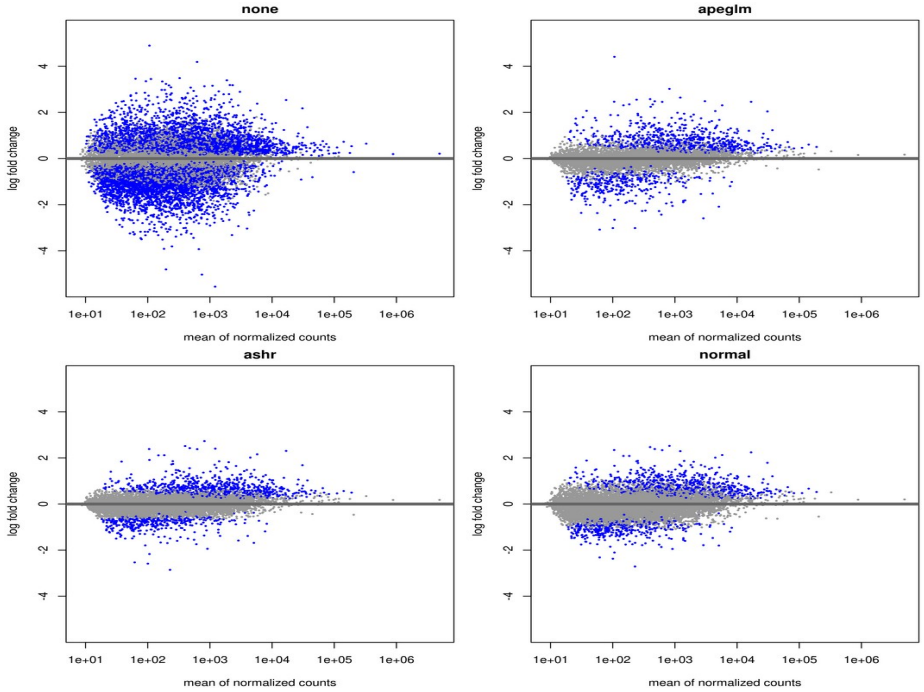
N.2 distinct filters: Filtering is customizable based on experimental conditions

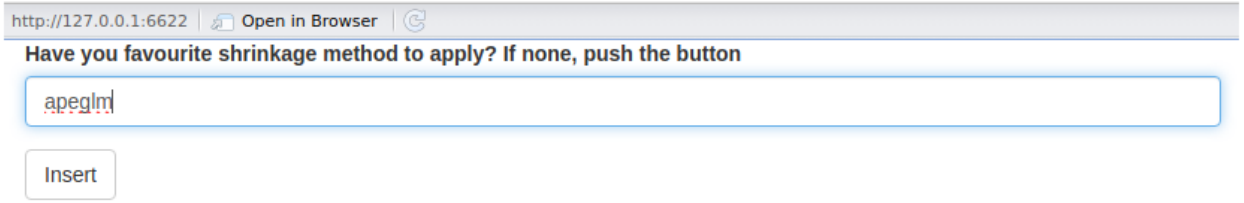
Check if data are biased



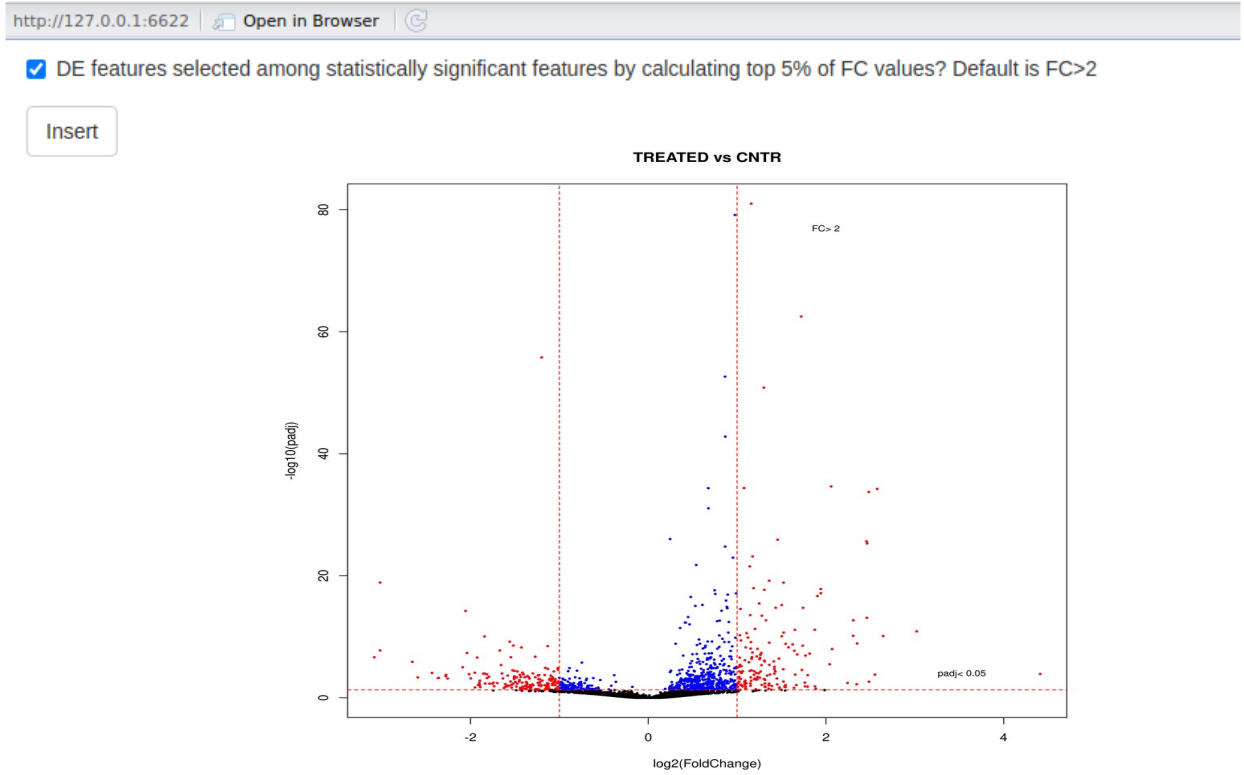
Preliminary Principal Component Analysis show potential bias due to patient specificity.

Data shrinkage





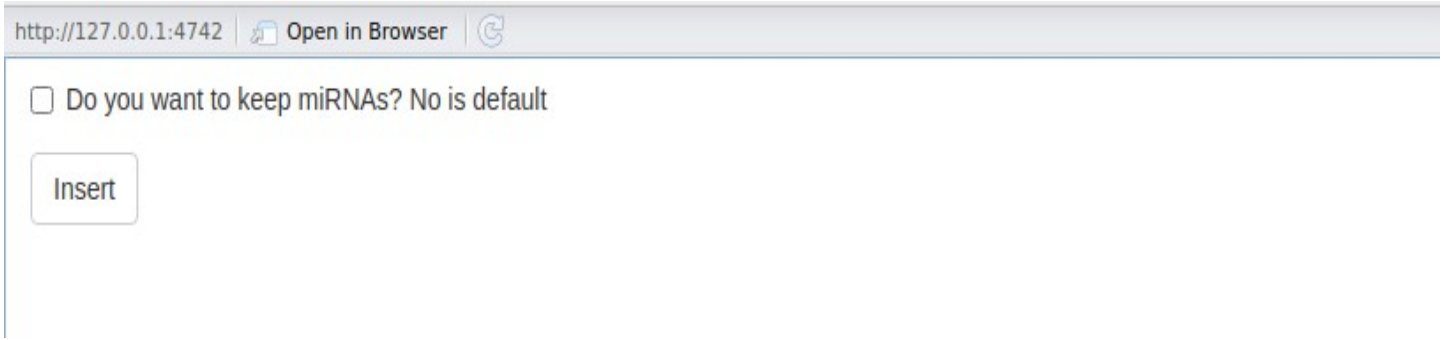
Differentially expressed features

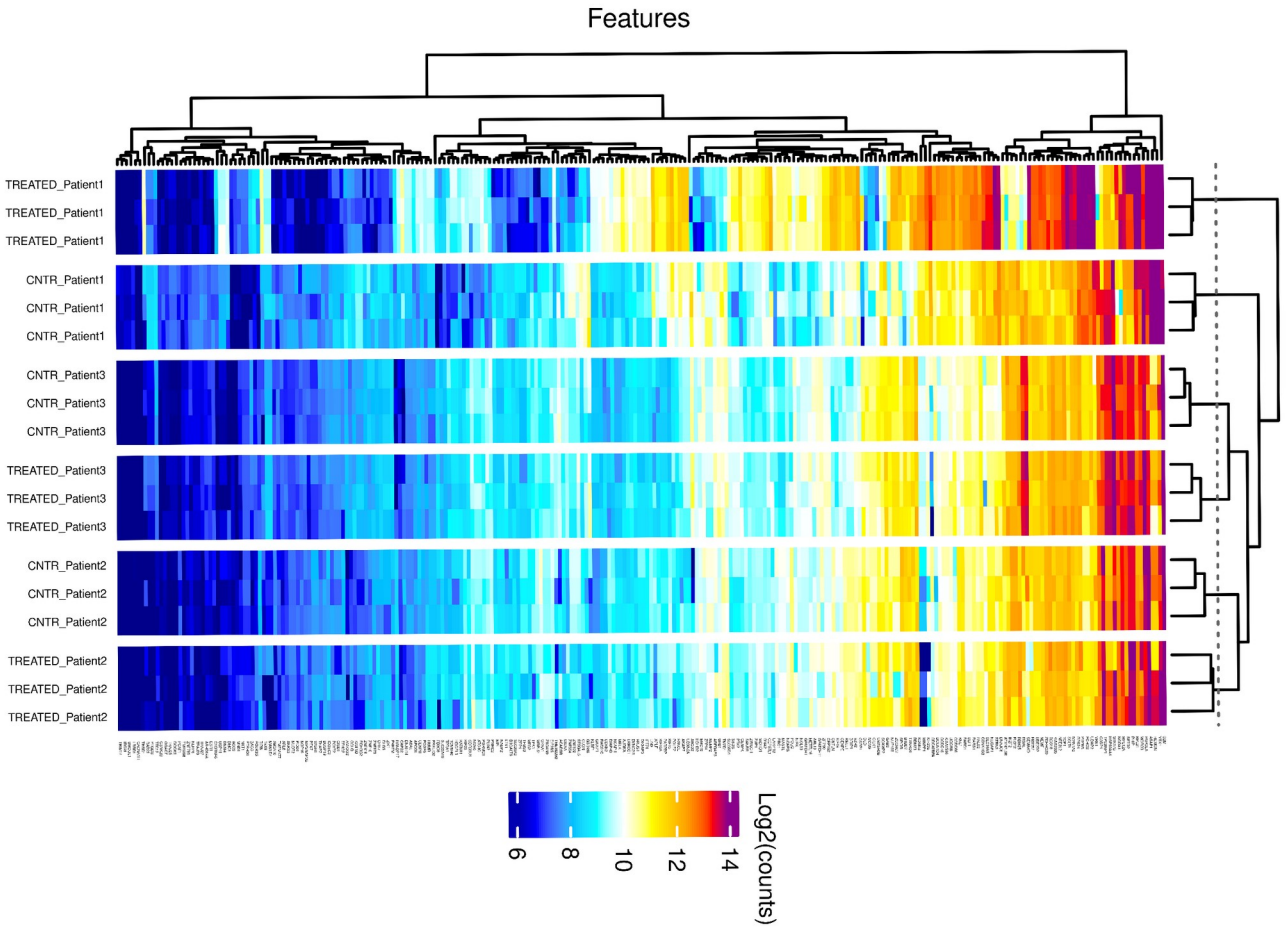


DE features = statistically significant (Wald test +BH correction for multiple testing) and FC> FC cut-off

Annotations

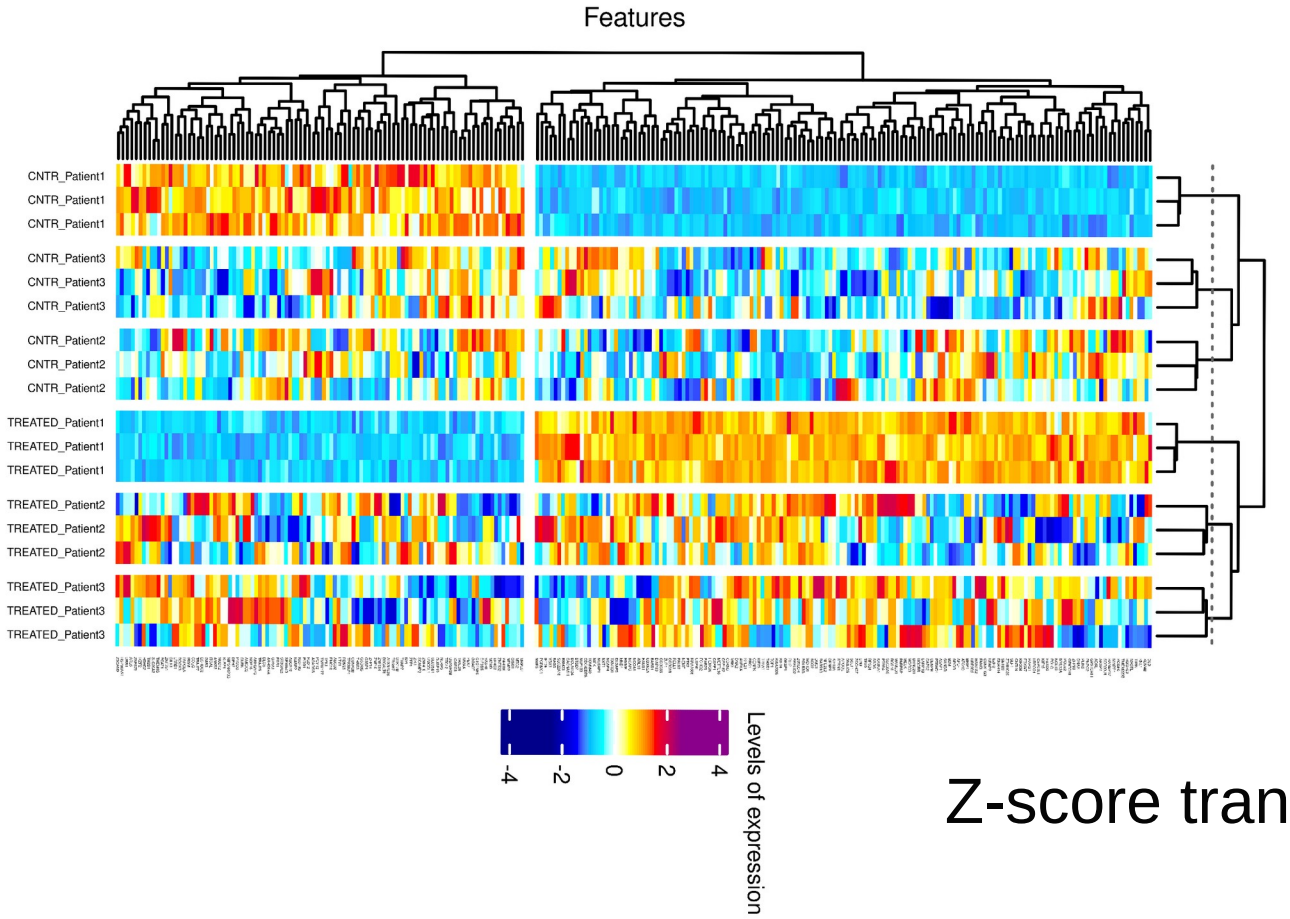
- Annotation of sequences based on Hugo Gene nomenclature (HGCN, https://www.genenames.org/).
- Cleaning: long intergenic non-protein coding RNA (lncRNA), uncharacterized open reading frames families with sequence similarity, small nucleolar RNA, pseudogenes, … are discarded (we focus on coding genes).
- Do you want to keep miRNAs? (customizable)





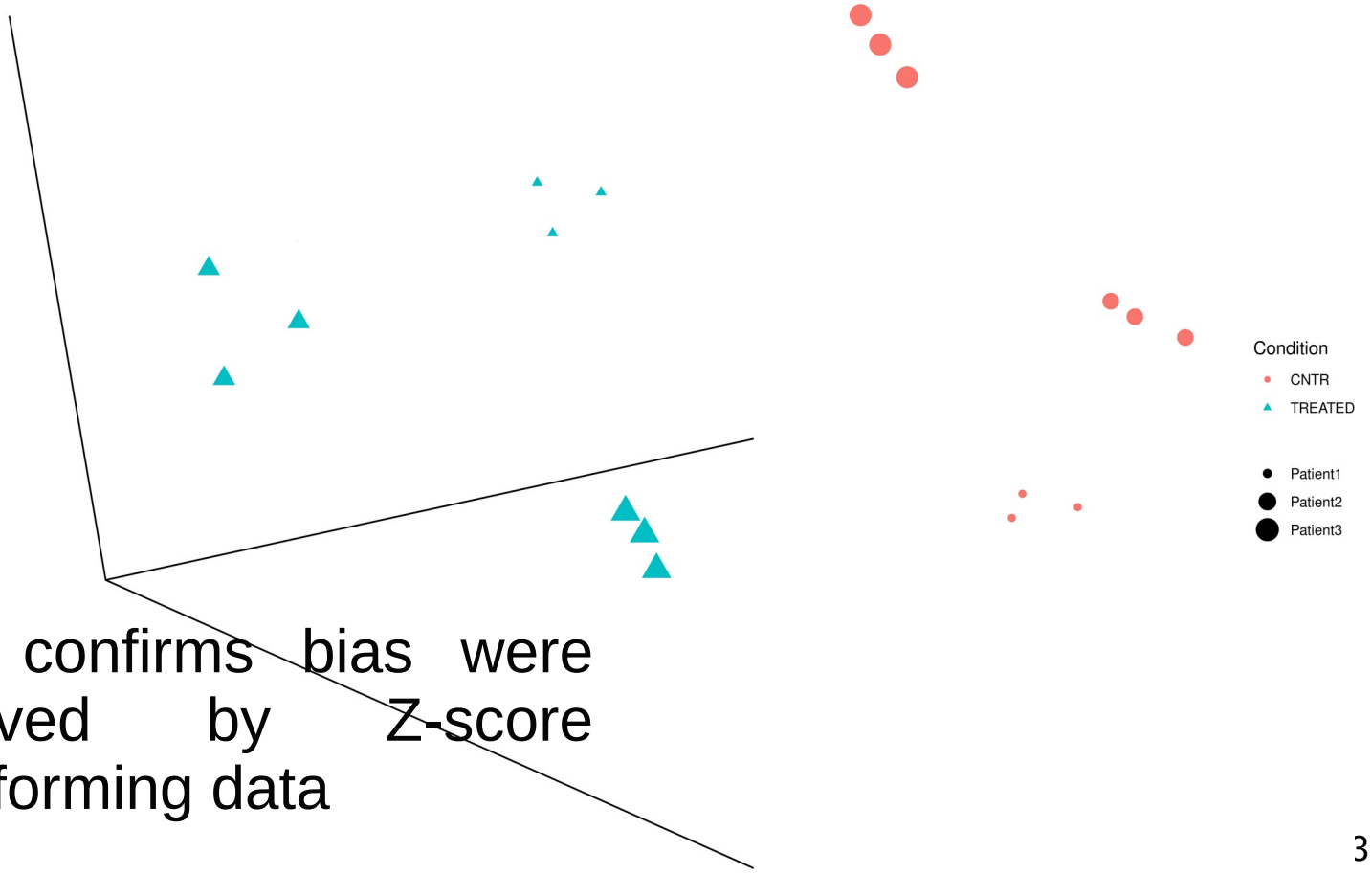
Samples are initially clustered by patient because of patient specificity

Bias correction



Z-score transformed data

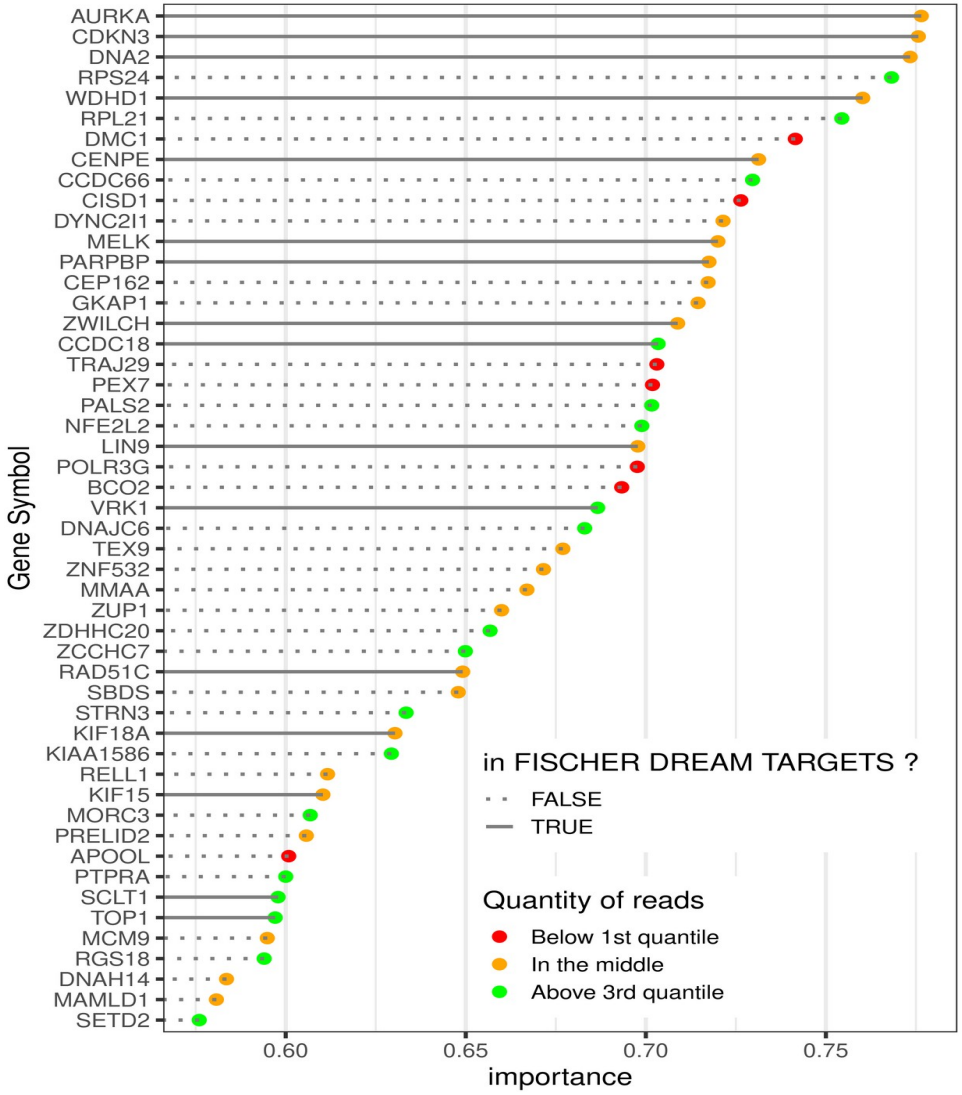
Dimensionality reduction



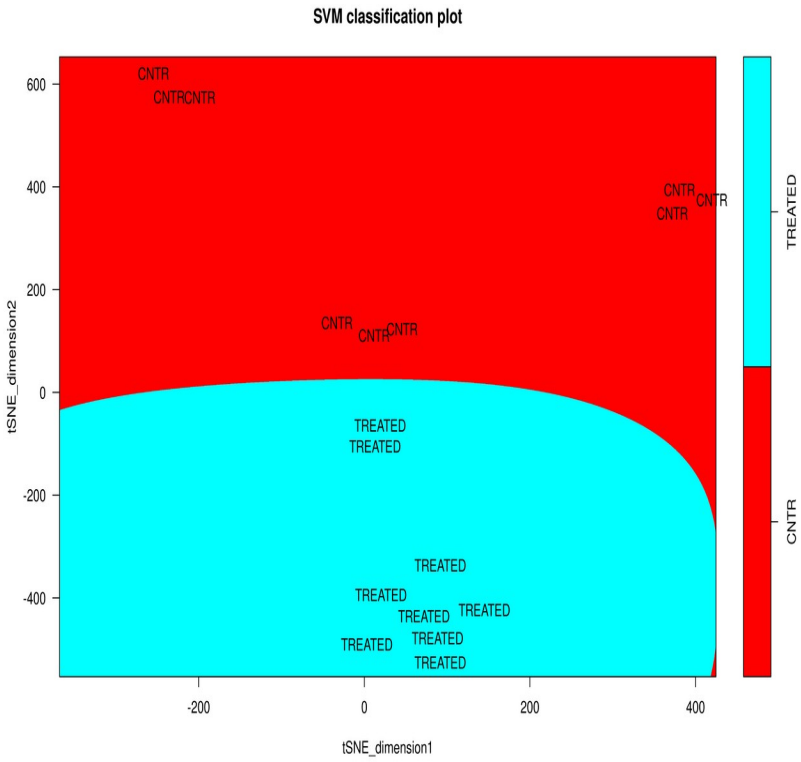
PCA confirms bias were removed by Z-score transforming data

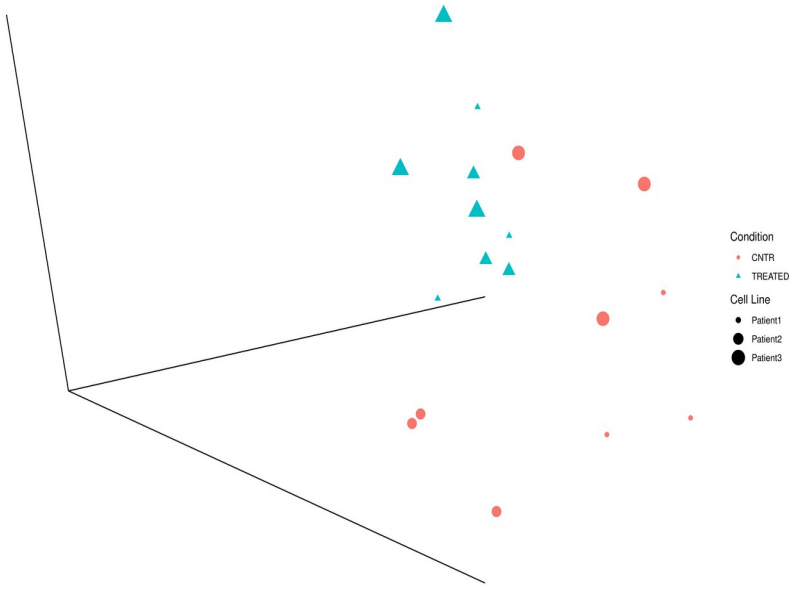
Features' importance: weights in composing first principal component

You can provide additional indications to biologists for next experiments



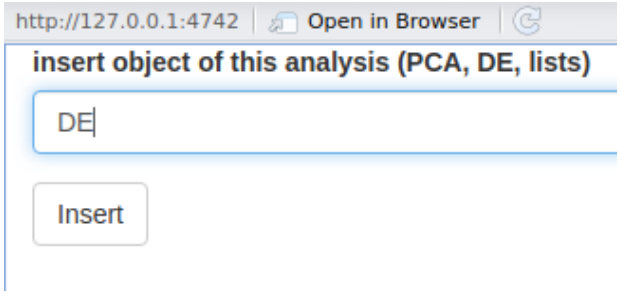
TSNE+SVM and UMAP

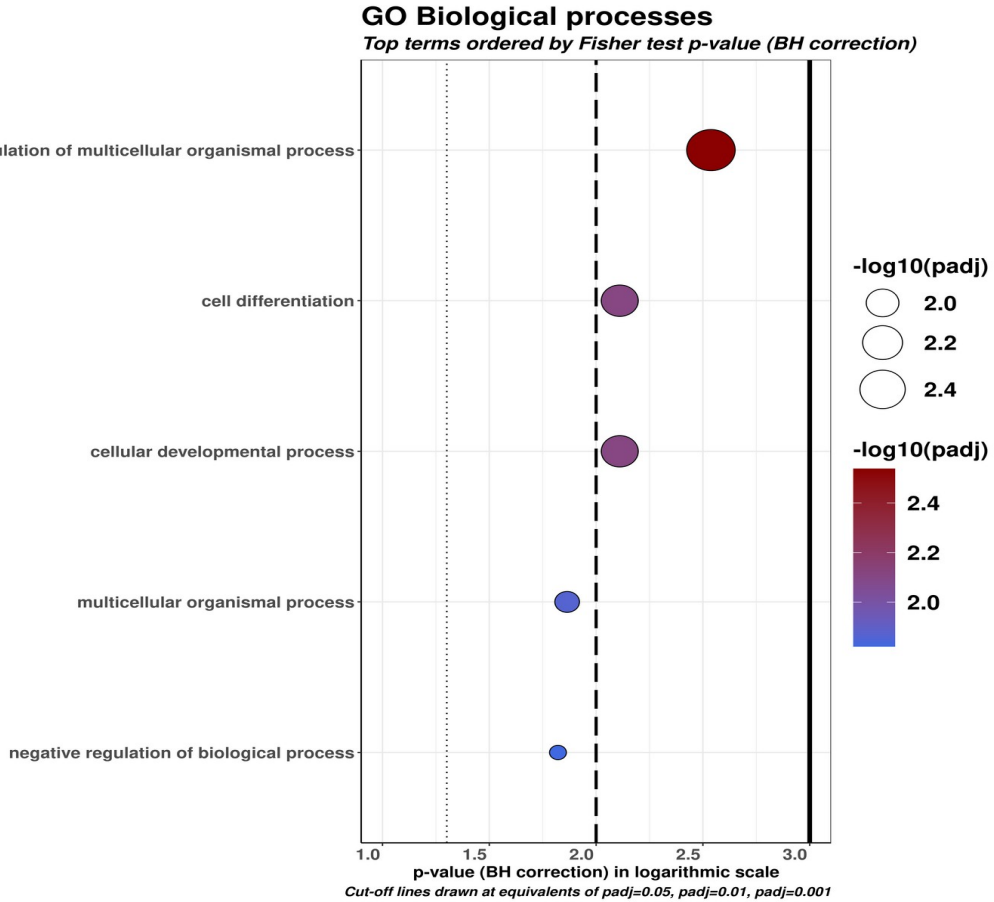




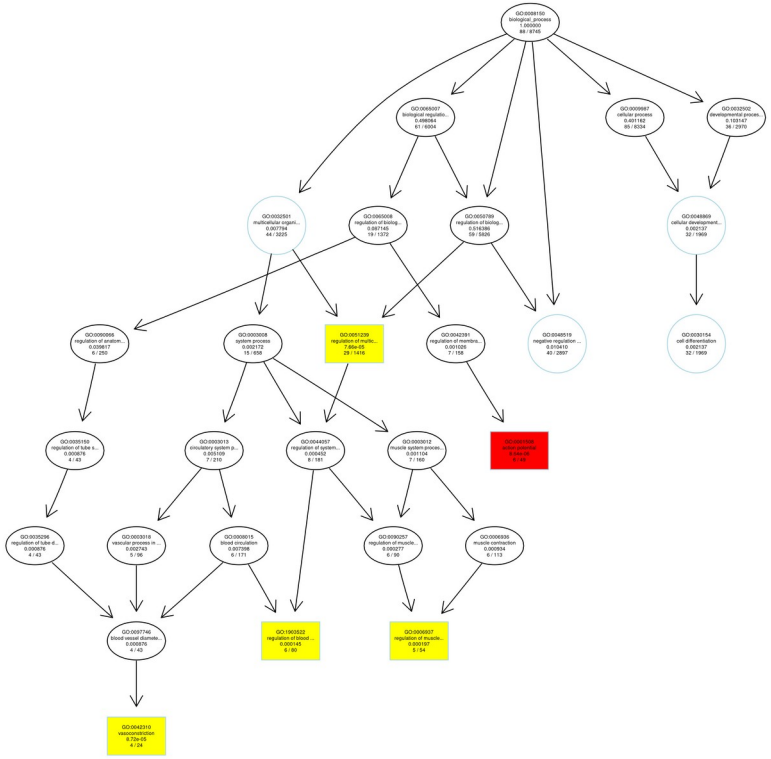
Enrichment of Gene Ontology terms

Customizable subset of interest to test:





Nodes graph

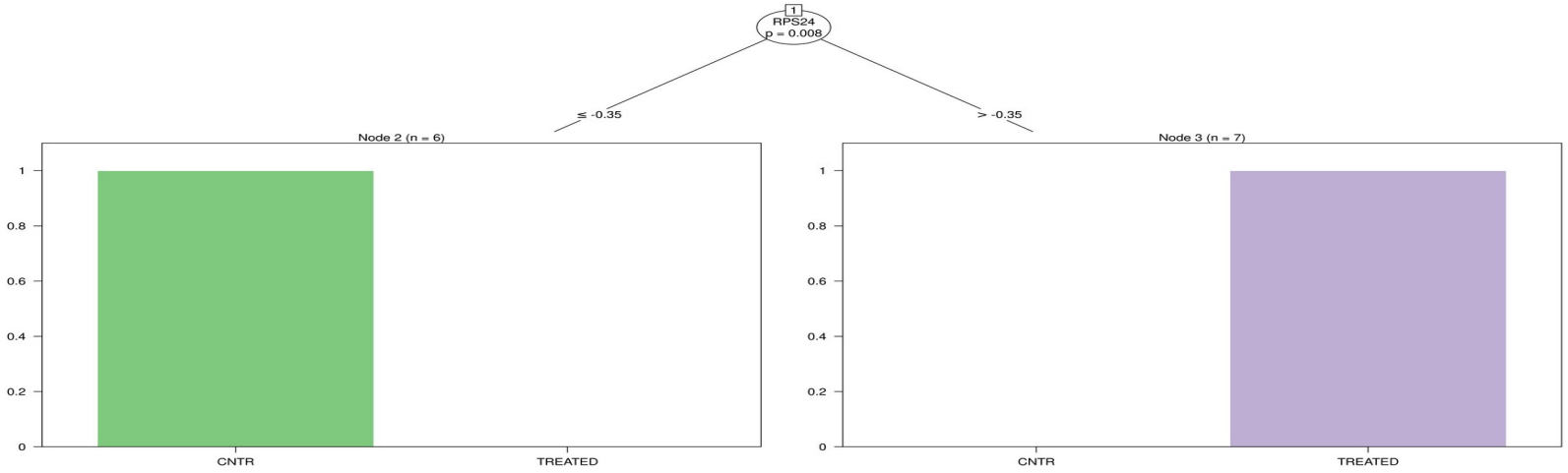


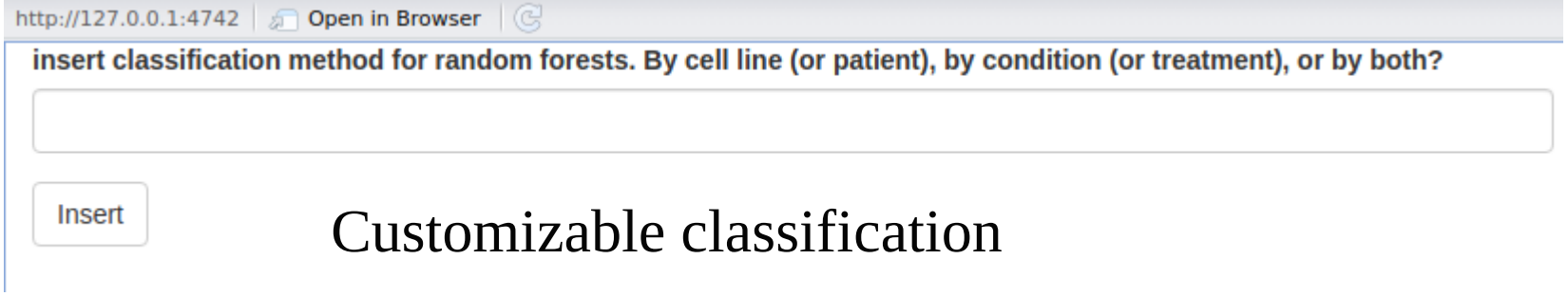
New gene sets are edited based on nodes graph:

multicellular_organismal_process_in_featuresSelection_for_Patient2 > GO:0032501 from GO enrichment test

AKAP6	ALPK3	APLF
ARHGAP42	BCL7A	BTG2
CAMK2D	CAMK4	CDH23
CDKN1A	CLMN	CRYBG3
DNHD1	ERMN	IL15
IL7R	KAT14	KCNMB4
LFNG	LILRB3	LOXL3
MAPK13	MC1R	MSR1
NUP155	OMG	OR52K2
OVGP1	P2RX1	PDK4
PROK2	RAPGEF1	SEMA4D
SIDT2	SLA2	SLC27A1
SLC8A1	TBX19	TNFSF8
TPH1	TRIM32	TYROBP
VCAN	ZMYND15	

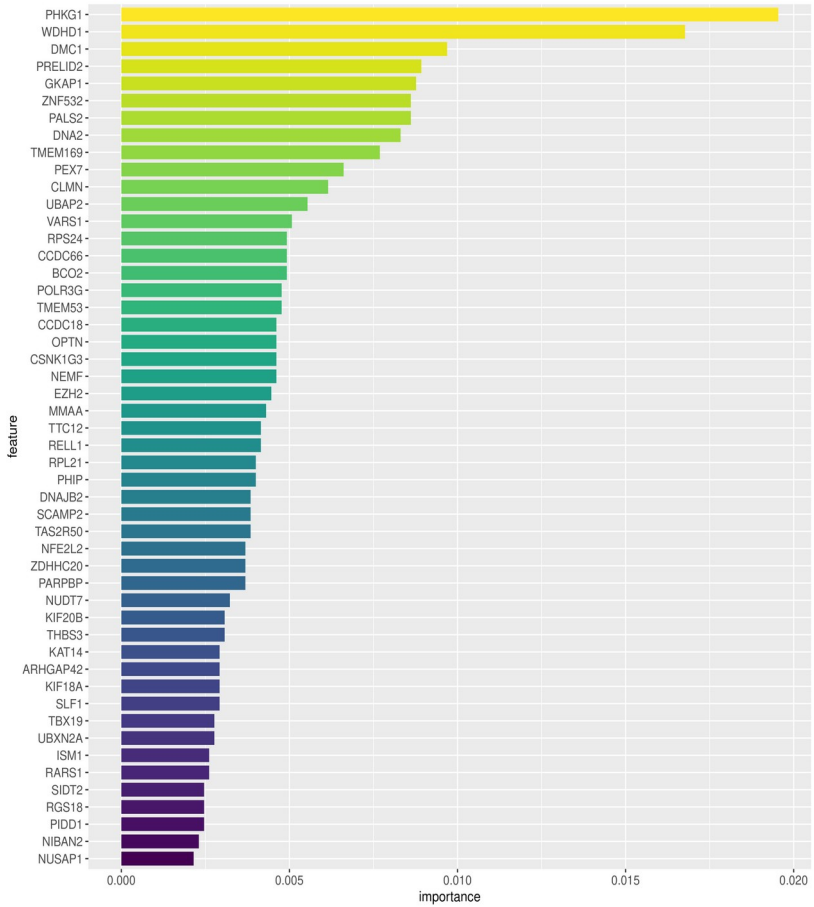
Conditional random forests





Customizable classification

Variables' importance

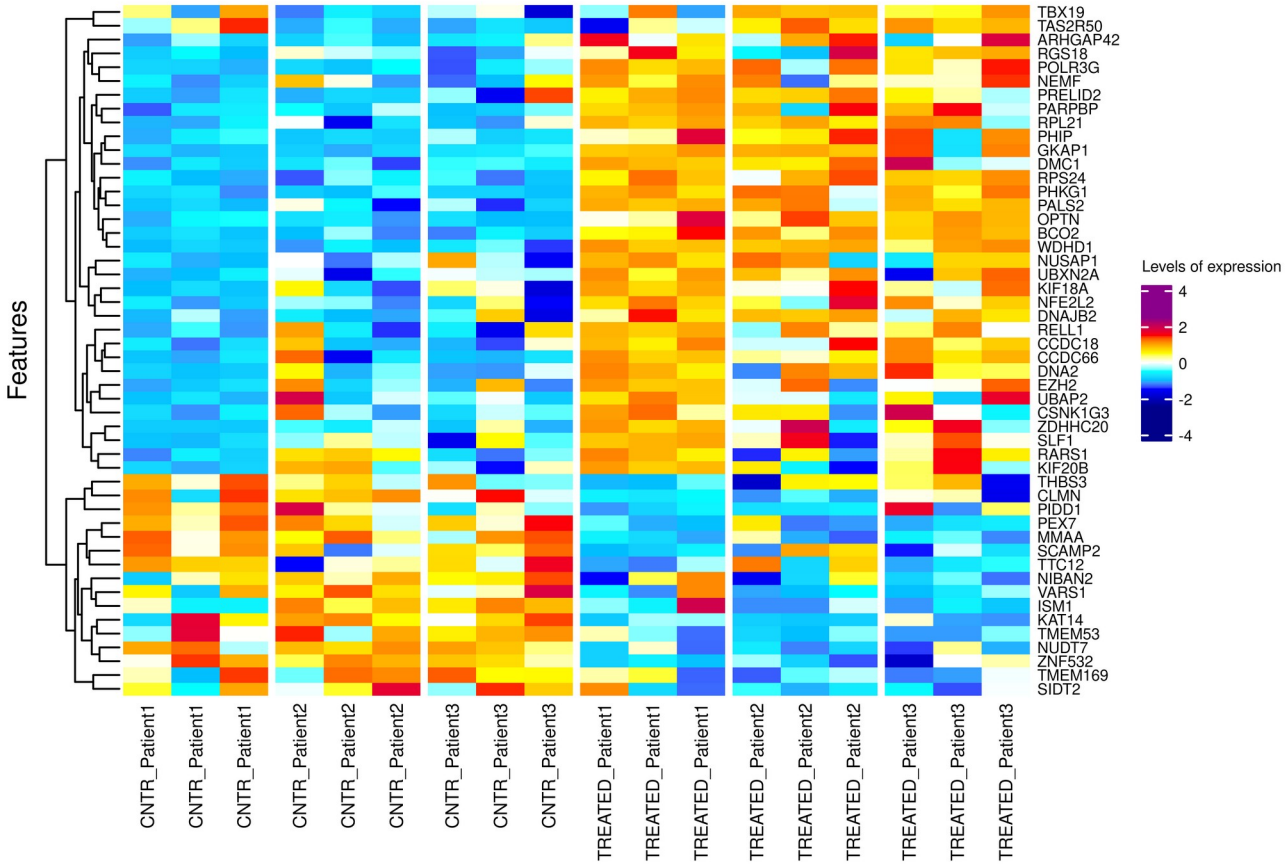


New gene sets are edited based on variables importance:

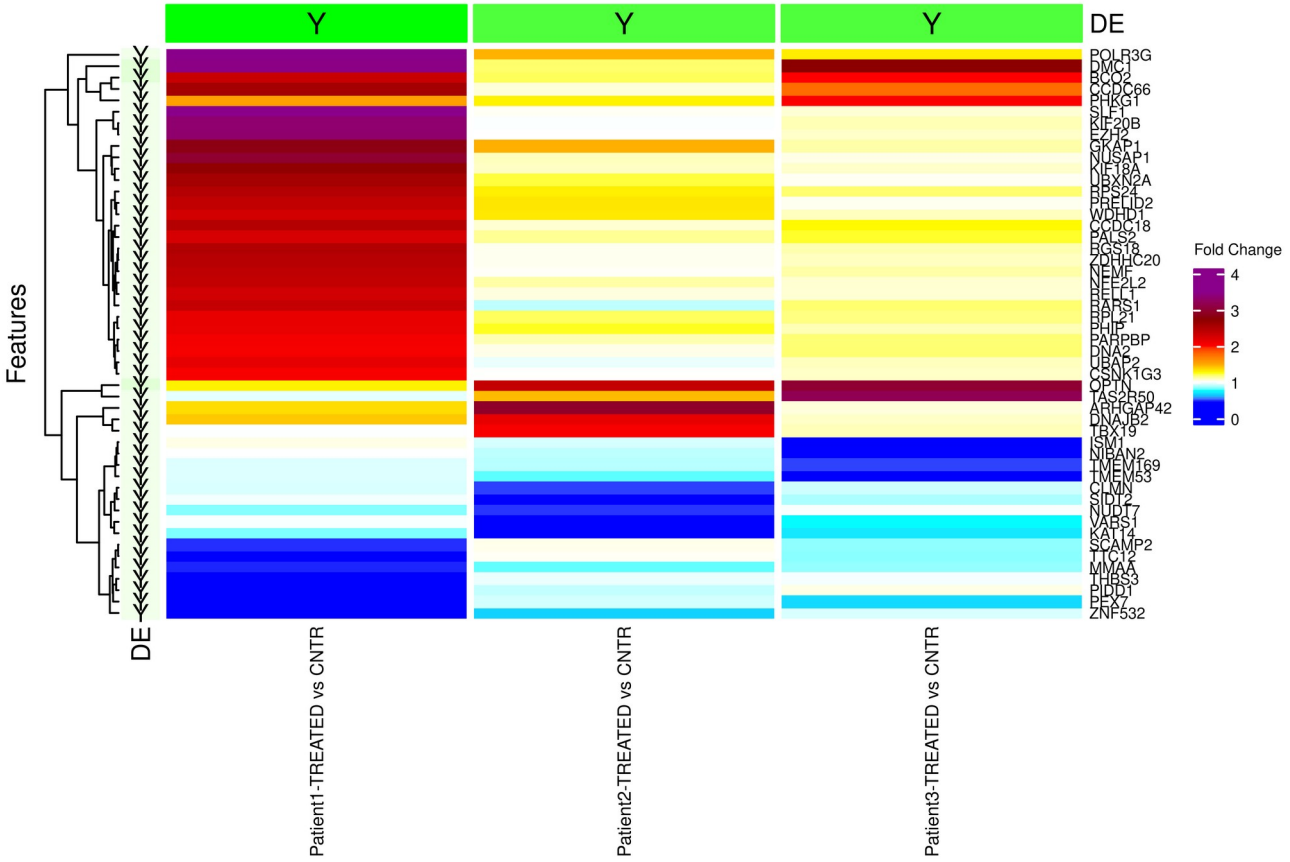
topFeaturesClassicByCondition > from classic variable

importance	PHKG1	WDHD1	DMC1
	PRELID2	GKAP1	DNA2
	PALS2	ZNF532	TMEM169
	CLMN	PEX7	UBAP2
	CCDC18	POLR3G	OPTN
	TMEM53	BCO2	RPS24
	NEMF	PHIP	RELL1
	PARPBP	TTC12	NUDT7
	TAS2R50	CSNK1G3	VARS1
	DNAJB2	EZH2	MMAA
	CCDC66	RPL21	ZDHHC20
	TBX19	UBXN2A	ARHGAP42
	KAT14	SCAMP2	NFE2L2
	RARS1	KIF20B	ISM1
	THBS3	KIF18A	SIDT2
	RGS18	PIDD1	SLF1
	NUSAP1	NIBAN2	

Focusing on single gene sets



Heat map of fold changes



Why MIEP?

MIEP's functions include:

- Dimensionality reduction and visualizations by PCA, UMAP, tSNE, SVM
- Enrichment of Gene Ontology (GO) terms
- Calculation of features' importance subsequent to classification (conditional random forests)
- Gene set editing based on the ranking of GO terms or features' importance
- Analyses focused on gene sets and user-friendly graphical representations.

These characteristics and careful handling of exceptions make MIEP an easy-to-use tool for biologists with basic programming skills.

… and it's for free!

MIEP's technical features (1/2)

- The settings of conditional random forests' parameters are optimized by a grid search approach, whose computational burden is eased by parallel computing;
- Distinct implementations of parallel computing are automatically tested before grid search, to select the fastest;
- Random seeds were considered by registering them from the global environment

MIEP's technical features (2/2)

- An ad hoc environment of the package was declared in the main script to facilitate the transmission of variables along the grid search or to object constructors calling functions;
- Since expanded swap memory is often used as a surrogate when RAM is limited, swappiness in Linux OS is automatically increased for machine learning by invoking system commands. A shiny app allows inserting a superuser password for the execution of system calls.

Taken together, these characteristics allow running the pipeline with limited computational resources.