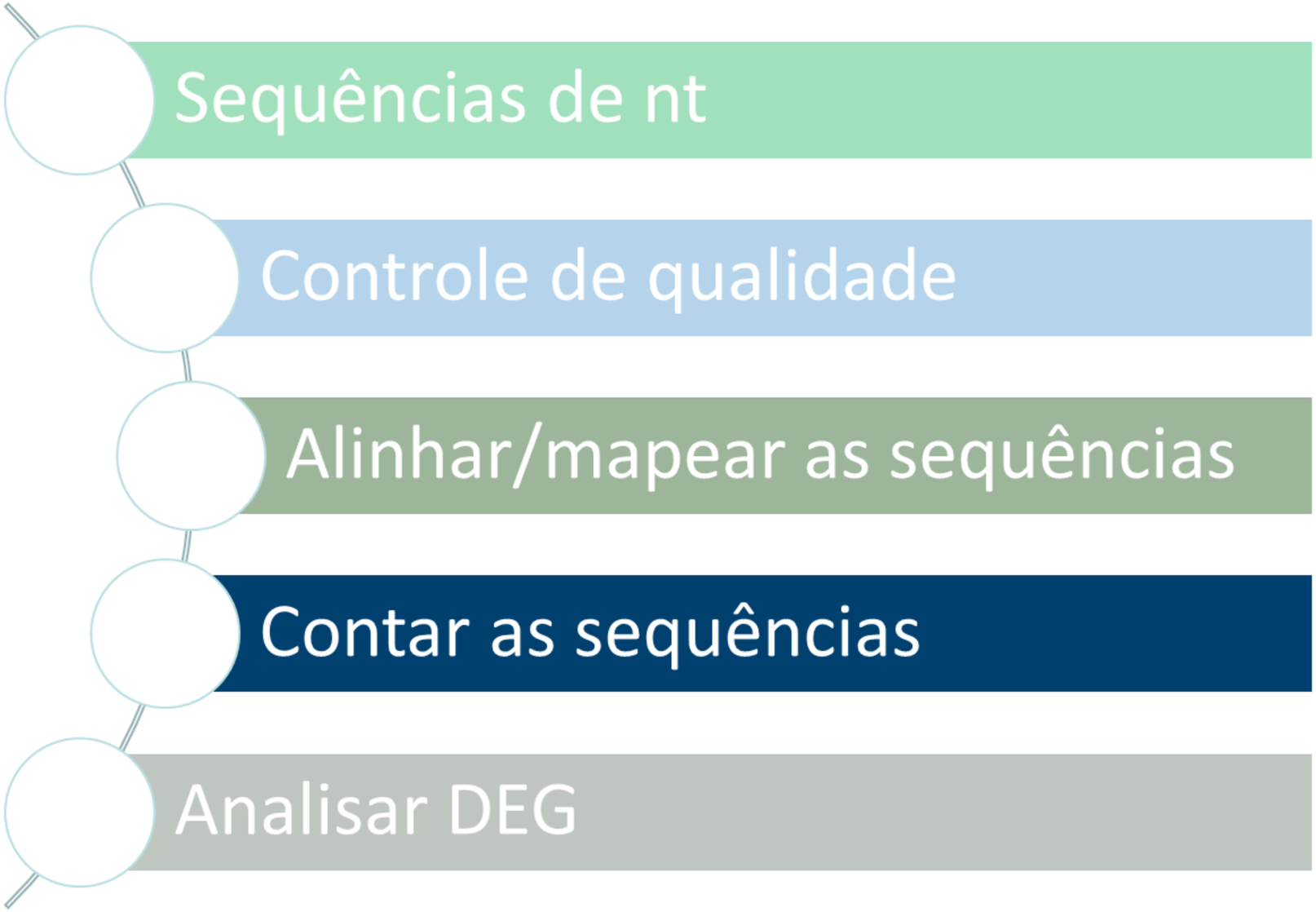
Contagem, análise de expressão diferencial e ontologia gênica

Adriana Mércia Guaratini Ibelli

15 a 19 de maio de 2023

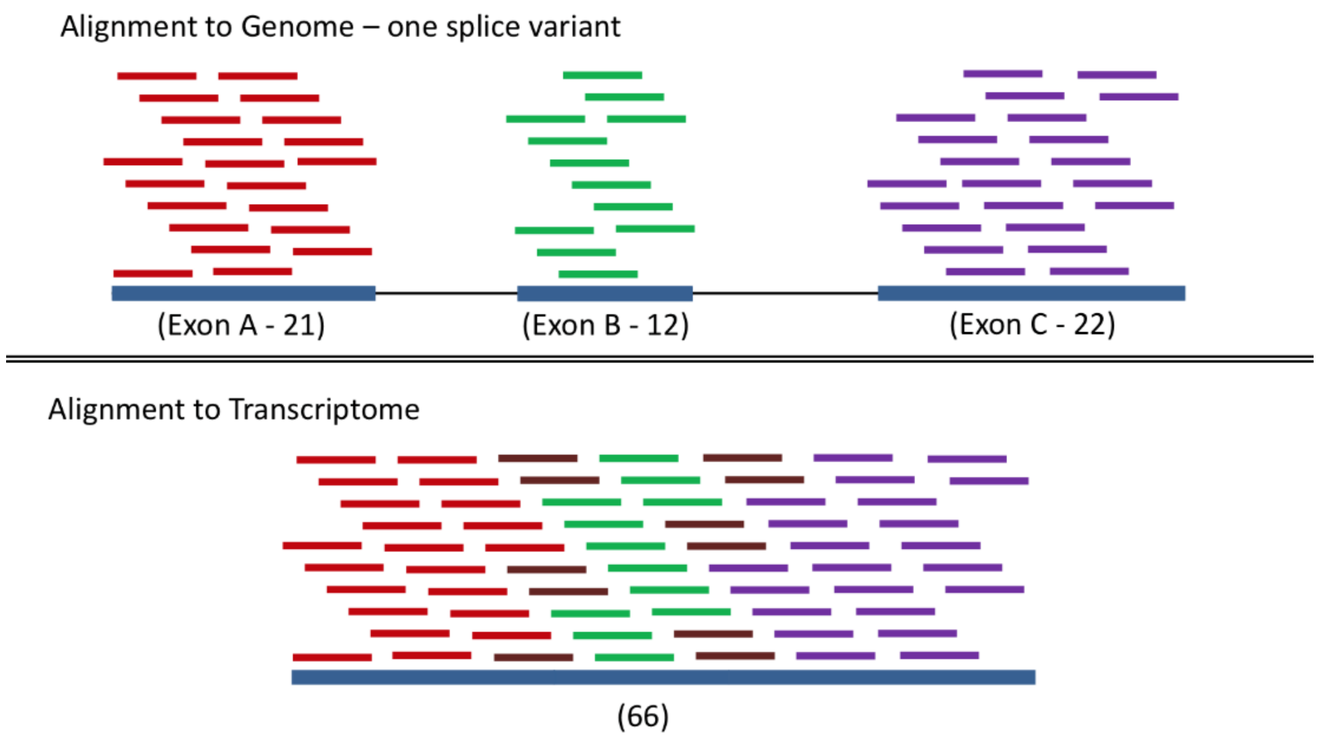
Embrapa 50 ANOS
MINISTÉRIO DA AGRICULTURA E PECUÁRIA
GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Onde estamos ???



Contagem das reads

Quanto > a contagem melhor !!



- Genoma referência
- Quantidade de reads x tamanho do gene
- Qualidade do mapeamento

Quantificação/contagem das reads por gene ou transcritos

Importante saber como o programa trabalha com :

- Overlaping das reads (read completa vs parcial);
- Reads multi-mapeadas;
- Reads que fazem overlapping com muitos genes/features;
- Reads em íntrons;

Ferramentas Populares de contagem:

- Htseq-counts (Anders et al., 2014);
- FeatureCounts (Liao et al., 2014)

Genes

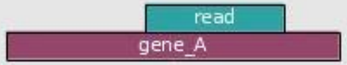
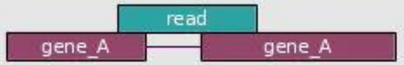
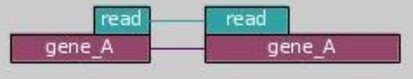
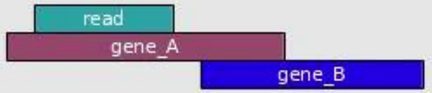
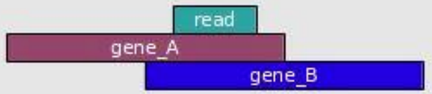
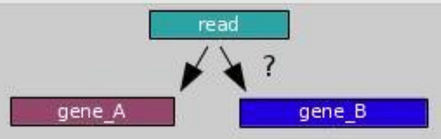
- RSEM (Li;Dewey, 2011)
- Cufflinks (Trapnell et al., 2012)

Transcritos

htseq counts

STAR

htseq e RSEM para contagens

	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous (both genes with --nonunique all)	gene_A	gene_A
	ambiguous (both genes with --nonunique all)		
	alignment_not_unique (both genes with --nonunique all)		

N_unmapped	85176	85176	85176
N_multimapping	64508	64508	64508
N_noFeature	263202	3389193	278428
N_ambiguous	193680	1704	50928
ENSSSCG00000037372	30	3	27
ENSSSCG00000027257	294	1	293
ENSSSCG00000029697	207	2	205
ENSSSCG00000027274	0	0	0
ENSSSCG00000027726	59	0	59
ENSSSCG00000033475	0	0	0
ENSSSCG00000035944	2	2	0
ENSSSCG00000004010	42	0	42
ENSSSCG00000004009	0	0	0
ENSSSCG00000030155	75	117	83
ENSSSCG00000038931	39	11	153
ENSSSCG00000004008	4	0	4
ENSSSCG00000004012	3850	10	3840
ENSSSCG00000035661	3	3	0
ENSSSCG00000032192	0	0	0
ENSSSCG00000004013	56	0	56
ENSSSCG00000037093	0	0	0
ENSSSCG00000031703	0	0	0
ENSSSCG00000004017	0	0	0
ENSSSCG00000004016	0	0	0
ENSSSCG00000004018	242	1	241
ENSSSCG00000038097	0	0	0
ENSSSCG00000027404	9	0	9
ENSSSCG00000022580	0	0	0
ENSSSCG00000032403	6	0	6
ENSSSCG00000037137	0	0	0
ENSSSCG00000024562	2	0	2
ENSSSCG00000034196	0	0	0
ENSSSCG00000004023	39	0	39
ENSSSCG00000004024	462	0	462
ENSSSCG00000004022	109	0	109
ENSSSCG00000032916	90	0	90
ENSSSCG00000040654	0	0	0
ENSSSCG00000004020	154	0	154
ENSSSCG00000004021	0	0	0
ENSSSCG00000036759	0	0	0
ENSSSCG00000034946	0	0	0
ENSSSCG00000036609	0	0	0
ENSSSCG00000036448	0	0	0
ENSSSCG00000004027	43	0	43
ENSSSCG00000034772	0	0	0

HE27_STAR_ReadsPerGene.out.tab

Tabela de contagens bruta

Mas a contagem é feita diretamente usando a quantidade bruta de reads ??

Normalização de Contagens

Normalização → Nos programas de expressão diferencial

Por que precisa ser feita uma normalização ???

Nível de transcrição

Tamanho do gene

Profundidade do sequenciamento

Expressão de outros genes

Metodologias de normalização

Table 13: Normalization methods for the comparison of gene read counts between different conditions. See, for example, Bullard et al. (2010) and Dillies et al. (2013) for comprehensive assessments of the individual methods.

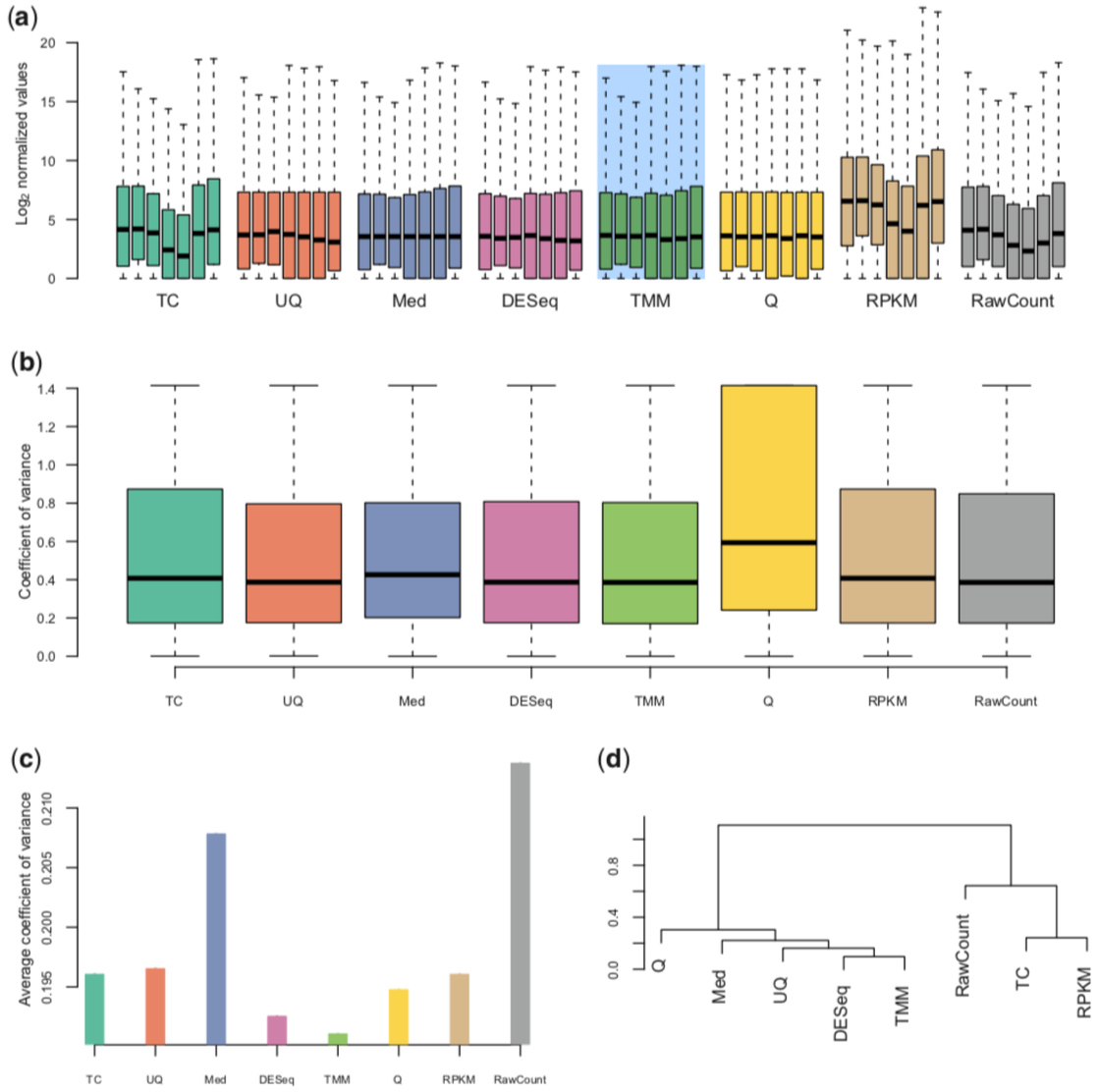
Name	Details	Comment
Total Count	All read counts are divided by the total number of reads (library size) and multiplied by the mean total count across all samples.	• biased by highly expressed genes • cannot account for different RNA repertoire between samples • poor detection sensitivity when benchmarked against qRT-PCR (Bullard et al., 2010)
Counts Per Million	Each gene count is divided by the corresponding library size (in millions).	• see Total Count
DESeq's size factor	1. For each gene, the **geometric mean** of read counts across all samples is calculated. 2. Every gene count is **divided by the geometric mean**. 3. A sample's size factor is the **median of these ratios** (skipping the genes with a geometric mean of zero).	• the size factor is applied to all read counts of a sample • more robust than total count normalization • implemented by the `DESeq` R library (`estimateSizeFactors()` function), also available in `edgeR` (`calcNormFactors()` function with option `method = "RLE"`) • details in Anders and Huber (2010)
Trimmed Mean of M-values (TMM)	TMM is always calculated as the weighted mean of log ratios between two samples: 1. Calculate gene-wise log_2 fold changes (= **M-values**): $M_g = log_2(\frac{Y_{gk}}{N_k})/log_2(\frac{Y_{gk'}}{N_{k'}})$ where Y is the observed number of reads per gene g in library k and N is the total number of reads. 2. **Trimming**: removal of upper and lower 30%. 3. **Precision weighing**: the inverse of the estimated variance is used to account for lower variance of genes with larger counts.	• the size factor is applied to every sample's library size; normalized read counts are obtained by dividing raw read counts by the TMM-adjusted library sizes • more robust than total count normalization • implemented in `edgeR` via `calcNormFactors()` with the default `method = "TMM"` • details in Robinson and Oshlack (2010)
Upper quartile	1. Find the upper quartile value (top 75% read counts after removal of genes with 0 reads). 2. Divide all read counts by this value.	• similar to total count normalization, thus it also suffers from a great influence of highly-expressed DE genes • can be calculated with `edgeR`'s `calcNormFactors()` function (`method = "upperquartile"`)

14: Normalization methods for the comparison of gene read counts within the same sample.

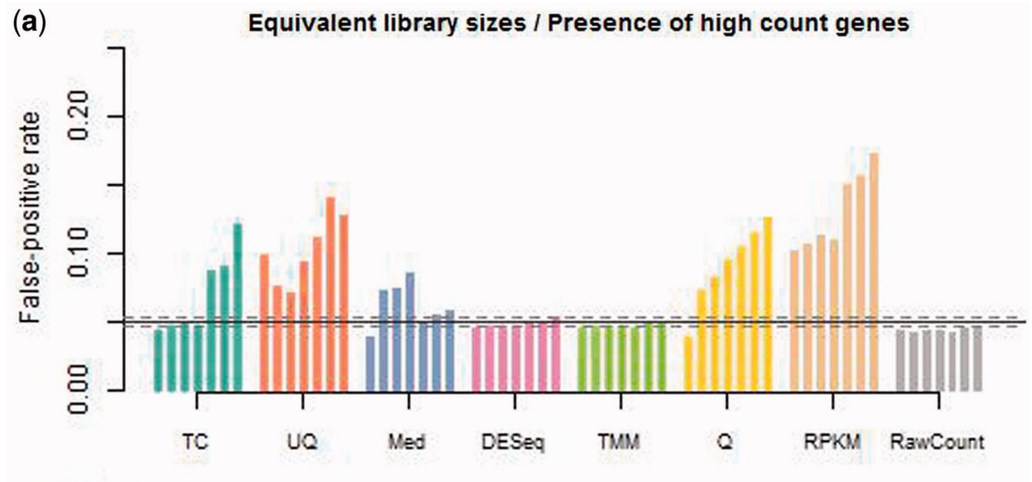
ıe	Details	Comment
ːM (reads ːilobase of ːs per on mapped s)	1. For each gene, count the number of reads mapping to it (X_i). 2. Divide that count by: the length of the gene, l_i, in base pairs divided by 1,000 multiplied by the total number of mapped reads, N, divided by 10^6. $$RPKM_i = \frac{X_i}{(\frac{l_i}{10^3})(\frac{N}{10^6})}$$	• introduces a bia the per-gene variances, in particular for lo expressed genes (Oshlack and Wakefield, 2009) • implemented in `edgeR`'s `rpkm()` function
ːM ːments per ɔase...)	1. Same as RPKM, but for paired-end reads: 2. The number of fragments (defined by two reads each) is used.	• implemented in `DESeq2`'s `fpkm()` function
ı	Instead of normalizing to the total library size, TPM represents the abundance of an individual gene i in relation to the abundances of the other transcripts (e.g., j) in the sample. 1. For each gene, count the number of reads mapping to it and divide by its length in base pairs (= counts per base). 2. Multiply that value by 1 divided by the sum of all counts per base of every gene. 3. Multiply that number by 10^6. $$TPM_i = \frac{X_i}{l_i} * \frac{1}{\sum_j \frac{X_j}{l_k}} * 10^6$$	• details in Wagn et al. (2012)

Dundar et al., 2015

Impacto da metologia de contagem



Dillies et al., 2013



Contagem total e FPKM → Não devem ser usadas para DE

DeSeq e TMM (EdgeR) → mais robustas

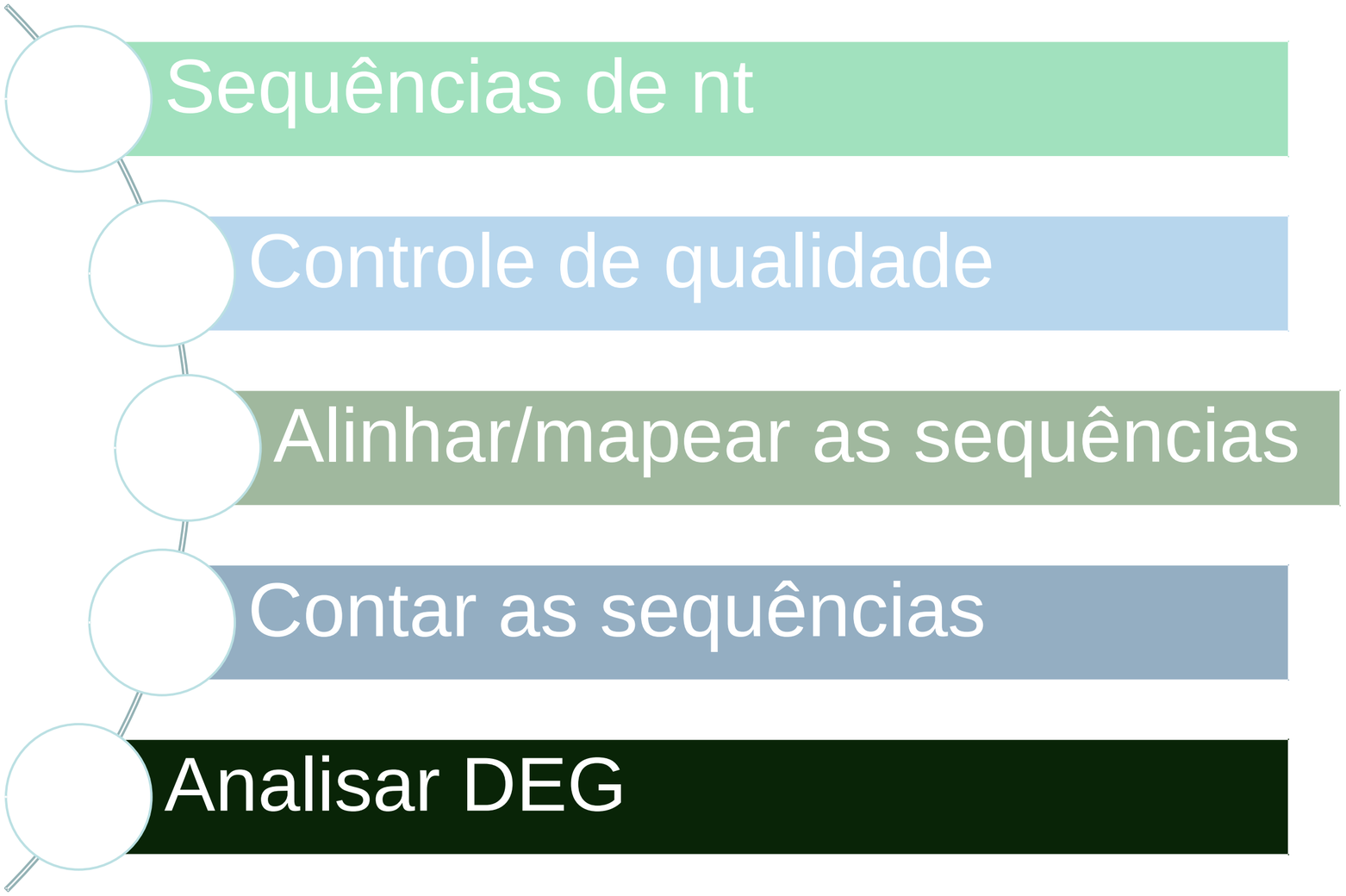
Summary of comparison results for the seven normalization methods under consideration

Method	Distribution	Intra-Variance	Housekeeping	Clustering	False-positive rate
TC	–	+	+	–	–
UQ	++	++	+	++	–
Med	++	++	–	++	–
DESeq	**++**	**++**	**++**	**++**	**++**
TMM	**++**	**++**	**++**	**++**	**++**
Q	++	–	+	++	–
RPKM	–	+	+	–	–

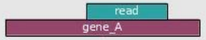
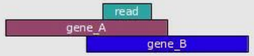
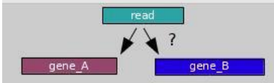
A '–' indicates that the method provided unsatisfactory results for the given criterion, while a '+' and '++' indicate satisfactory and very satisfactory results for the given criterion.

DeSeq e TMM (EdgeR) → recomendados para DEG

Normalization method	Description	Accounted factors	Recommendations for use
CPM (counts per million)	counts scaled by total number of reads	sequencing depth	gene count comparisons between replicates of the same samplegroup; **NOT for within sample comparisons or DE analysis**
TPM (transcripts per kilobase million)	counts per length of transcript (kb) per million reads mapped	sequencing depth and gene length	gene count comparisons within a sample or between samples of the same sample group; **NOT for DE analysis**
RPKM/FPKM (reads/fragments per kilobase of exon per million reads/fragments mapped)	similar to TPM	sequencing depth and gene length	gene count comparisons between genes within a sample; **NOT for between sample comparisons or DE analysis**
DESeq2's **median of ratios** [1]	counts divided by sample-specific size factors determined by median ratio of gene counts relative to geometric mean per gene	sequencing depth and RNA composition	gene count comparisons between samples and for **DE analysis**; **NOT for within sample comparisons**
EdgeR's **trimmed mean of M values (TMM)** [2]	uses a weighted trimmed mean of the log expression ratios between samples	sequencing depth, RNA composition, and gene length	gene count comparisons between and within samples and for **DE analysis**


Sequências de nt
Controle de qualidade
Alinhar/mapear as sequências
Contar as sequências
Analisar DEG

Análise de expressão gênica diferencial (DEG)

	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous (both genes with --nonunique all)	gene_A	gene_A
	ambiguous (both genes with --nonunique all)		
	alignment_not_unique (both genes with --nonunique all)		

DEG

Objetivos:

1- Estimar a magnitude da expressão entre 2 ou mais condições;

2- Estimar a significância das diferenças;

DEG ➔ distribuição Binomial Negativa, Poisson, etc.

Semelhante a análises de microarranjos

DEG

Efeitos específicos da amostras são normalizados:

Específico da amostra:
Profundidade do sequenciamento
Composição do RNA

Conteúdo de GC
Tamanho do gene

Ferramentas de análise

ShrinkSeq

NoiSeq

baySeq

Vsf

Voom

SAMseq

TSPM

DESeq

EBSeq

NBPSeq

edgeR

limma

+ other (not-R)
including CuffDiff

Soneson;Delorenzi,

DEG

Feature	DESeq2	edgeR	limmaVoom	Cuffdiff
Seq. depth normalization	Sample-wise size factor	Gene-wise trimmed median of means (TMM)	Gene-wise trimmed median of means (TMM)	FPKM-like or DESeq-like
Dispersion estimate	Cox-Reid approximate conditional inference with focus on maximum *individual* dispersion estimate	Cox-Reid approximate conditional inference moderated towards the *mean*	squeezes gene-wise residual variances towards the global variance	
Assumed distribution	Neg. binomial	Neg. binomial	*log*-normal	Neg. binomial
Test for DE	Wald test (2 factors); LRT for multiple factors	exact test for 2 factors; LRT for multiple factors	*t*-test	*t*-test
False positives	Low	Low	Low	High
Detection of differential isoforms	No	No	No	Yes
Support for multi-factored experiments	Yes	Yes	Yes	No
Runtime (3-5 replicates)	Seconds to minutes	Seconds to minutes	Seconds to minutes	Hours

Table 2 Summary of the main observations

DESeq	- Conservative with default settings. Becomes more conservative when outliers are introduced.
	- Generally low TPR.
	- Poor FDR control with 2 samples/condition, good FDR control for larger sample sizes, also with outliers.
	- Medium computational time requirement, increases slightly with sample size.
edgeR	- Slightly liberal for small sample sizes with default settings. Becomes more liberal when outliers are introduced.
	- Generally high TPR.
	- Poor FDR control in many cases, worse with outliers.
	- Medium computational time requirement, largely independent of sample size.
NBPSeq	- Liberal for all sample sizes. Becomes more liberal when outliers are introduced.
	- Medium TPR.
	- Poor FDR control, worse with outliers. Often truly non-DE genes are among those with smallest p-values.
	- Medium computational time requirement, increases slightly with sample size.
TSPM	- Overall highly sample-size dependent performance.
	- Liberal for small sample sizes, largely unaffected by outliers.
	- Very poor FDR control for small sample sizes, improves rapidly with increasing sample size. Largely unaffected by outliers.
	- When all genes are overdispersed, many truly non-DE genes are among the ones with smallest p-values. Remedied when the counts for some genes are Poisson distributed.
	- Medium computational time requirement, largely independent of sample size.
voom / vst	- Good type I error control, becomes more conservative when outliers are introduced.
	- Low power for small sample sizes. Medium TPR for larger sample sizes.
	- Good FDR control except for simulation study B_0^{4000}. Largely unaffected by introduction of outliers.
	- Computationally fast.
baySeq	- Highly variable results when all DE genes are regulated in the same direction. Less variability when the DE genes are regulated in different directions.
	- Low TPR. Largely unaffected by outliers.
	- Poor FDR control with 2 samples/condition, good for larger sample sizes in the absence of outliers. Poor FDR control in the presence of outliers.
	- Computationally slow, but allows parallelization.
EBSeq	- TPR relatively independent of sample size and presence of outliers.
	- Poor FDR control in most situations, relatively unaffected by outliers.
	- Medium computational time requirement, increases slightly with sample size.

Soneson;Delorenzi, 2013

edgeR (empirical analysis of DGE in R)

Considera distribuição binomial negativo

limma (Linear Models for Microarray Data)

Log-normal

Pressupostos:

- Cada amostra é sequenciada e as reads são mapeadas apropriadamente no genoma referência.
- O tamanho esperado da contagem: tamanho da biblioteca e a abundância relativa do gene

F1000Research F1000Research 2018, 5:1408 Last updated: 17 APR 2023

Check for updates

SOFTWARE TOOL ARTICLE

REVISED RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR [version 3; peer review: 3 approved]

Charity W. Law[1,2], Monther Alhamdoosh[3], Shian Su[1], Xueyi Dong[1], Luyi Tian[1,2], Gordon K. Smyth[1,4], Matthew E. Ritchie[1,2,4]

[1]The Walter and Eliza Hall Institute of Medical Research, Parkville, 3052, Australia
[2]Department of Medical Biology, The University of Melbourne, Parkville, 3010, Australia
[3]CSL Limited, Parkville, Victoria, 3010, Australia
[4]School of Mathematics and Statistics, The University of Melbourne, Parkville, 3010, Australia

v3 **First published:** 17 Jun 2016, **5**:1408
https://doi.org/10.12688/f1000research.9005.1
Second version: 30 Nov 2016, **5**:1408
https://doi.org/10.12688/f1000research.9005.2
Latest published: 28 Dec 2018, **5**:1408
https://doi.org/10.12688/f1000research.9005.3

Abstract
The ability to easily and efficiently analyse RNA-sequencing data is a key strength of the Bioconductor project. Starting with counts summarised at the gene-level, a typical analysis involves pre-processing, exploratory data analysis, differential expression testing and pathway analysis with the results obtained informing future experiments and validation studies. In this workflow article, we analyse RNA-sequencing data from the mouse mammary gland, demonstrating use of the popular **edgeR** package to import, organise, filter and normalise the data, followed by the **limma** package with its *voom* method, linear modelling and empirical Bayes moderation to assess differential expression and perform gene set testing. This pipeline is further enhanced by the **Glimma** package which enables interactive exploration of the results so that individual samples and genes can be examined by the user. The complete analysis offered by these three packages highlights the ease with which researchers can turn the raw counts from an RNA-sequencing experiment into biological insights using Bioconductor.

Keywords
RNA sequencing, data analysis, gene expression

Open Peer Review

Approval Status ✓ ✓ ✓

	1	2	3
version 3 (revision) 28 Dec 2018			
version 2 (revision) 30 Nov 2016			
version 1 17 Jun 2016	✓ view	✓ view	✓ view

1. **James W. MacDonald**, University of Washington, Seattle, USA
2. **Maria A. Doyle**, Peter MacCallum Cancer Centre, Melbourne, Australia
3. **Jovana Maksimovic**, Murdoch Children's Research Institute, Parkville, Australia

Any reports and responses or comments on the article can be found at the end of the article.

https://f1000research.com/articles/5-1408/v3

F1000Research F1000Research 2020, 9:1444 Last updated: 18 JUL 2022

Check for updates

METHOD ARTICLE

A guide to creating design matrices for gene expression experiments [version 1; peer review: 2 approved]

Charity W. Law[1,2], Kathleen Zeglinski[1,3], Xueyi Dong[1,2], Monther Alhamdoosh[3], Gordon K. Smyth[1,4], Matthew E. Ritchie[1,2,4]

[1]The Walter and Eliza Hall Institute of Medical Research, Parkville, 3052, Australia
[2]Department of Medical Biology, The University of Melbourne, Parkville, 3010, Australia
[3]Research and Development, CSL Limited, Bio21 Institute, Parkville, 3010, Australia
[4]School of Mathematics and Statistics, The University of Melbourne, Parkville, 3010, Australia

v1 **First published:** 10 Dec 2020, **9**:1444
https://doi.org/10.12688/f1000research.27893.1
Latest published: 10 Dec 2020, **9**:1444
https://doi.org/10.12688/f1000research.27893.1

Abstract
Differential expression analysis of genomic data types, such as RNA-sequencing experiments, use linear models to determine the size and direction of the changes in gene expression. For RNA-sequencing, there are several established software packages for this purpose accompanied with analysis pipelines that are well described. However, there are two crucial steps in the analysis process that can be a stumbling block for many -- the set up an appropriate model via design matrices and the set up of comparisons of interest via contrast matrices. These steps are particularly troublesome because an extensive catalogue for design and contrast matrices does not currently exist. One would usually search for example case studies across different platforms and mix and match the advice from those sources to suit the dataset they have at hand. This article guides the reader through the basics of how to set up design and contrast matrices. We take a practical approach by providing code and graphical representation of each case study, starting with simpler examples (e.g. models with a single explanatory variable) and move onto more complex ones (e.g. interaction models, mixed effects models, higher order time series and cyclical models). Although our work has been written specifically with a **limma**-style pipeline in mind, most of it is also applicable to other software packages for differential

Open Peer Review

Approval Status ✓ ✓

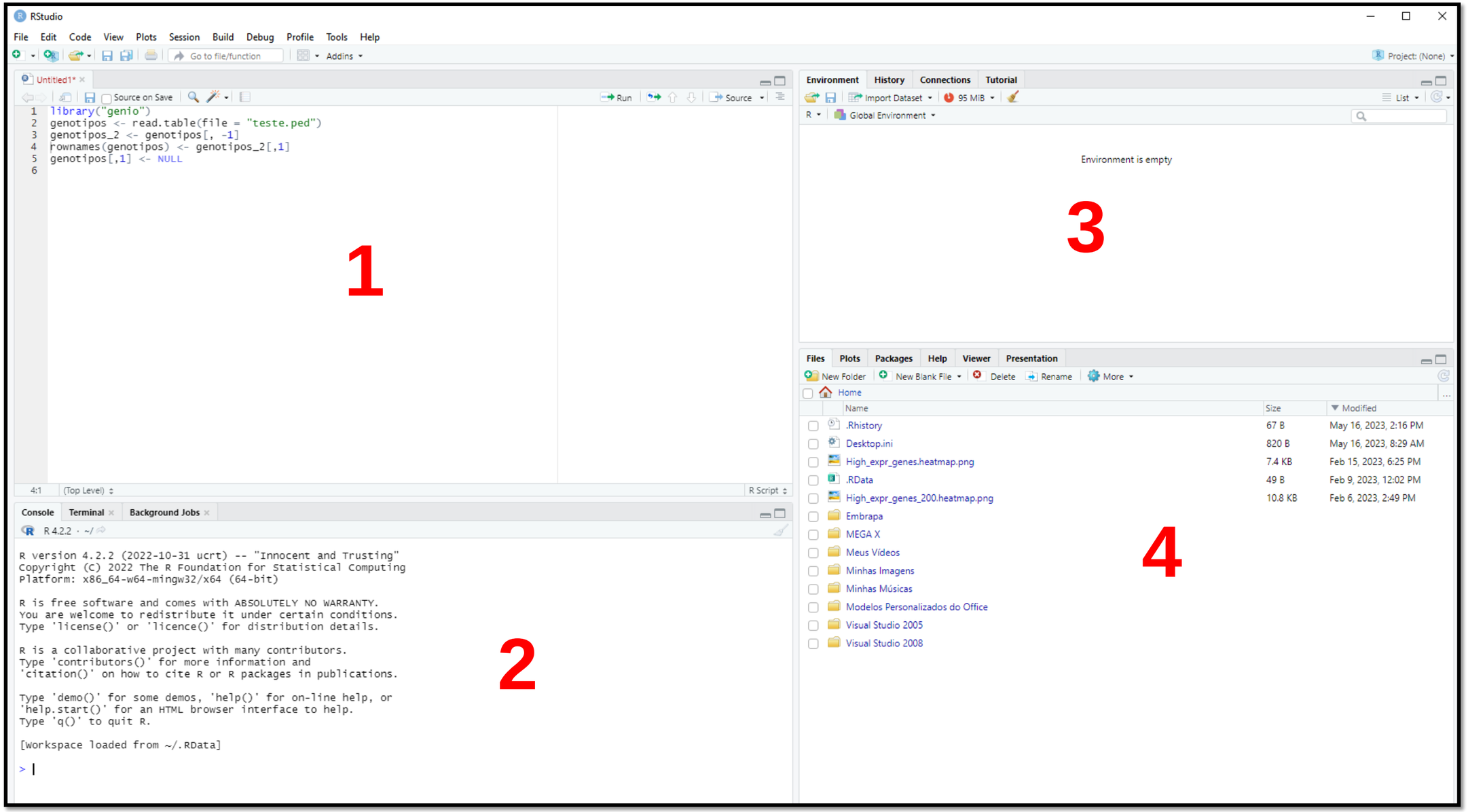
	1	2
version 1 10 Dec 2020	✓ view	✓ view

1. **Federico Marini**, University Medical Center of the Johannes Gutenberg University Mainz, Mainz, Germany
2. **Jenny Drnevich**, University of Illinois, Urbana-Champaign, USA

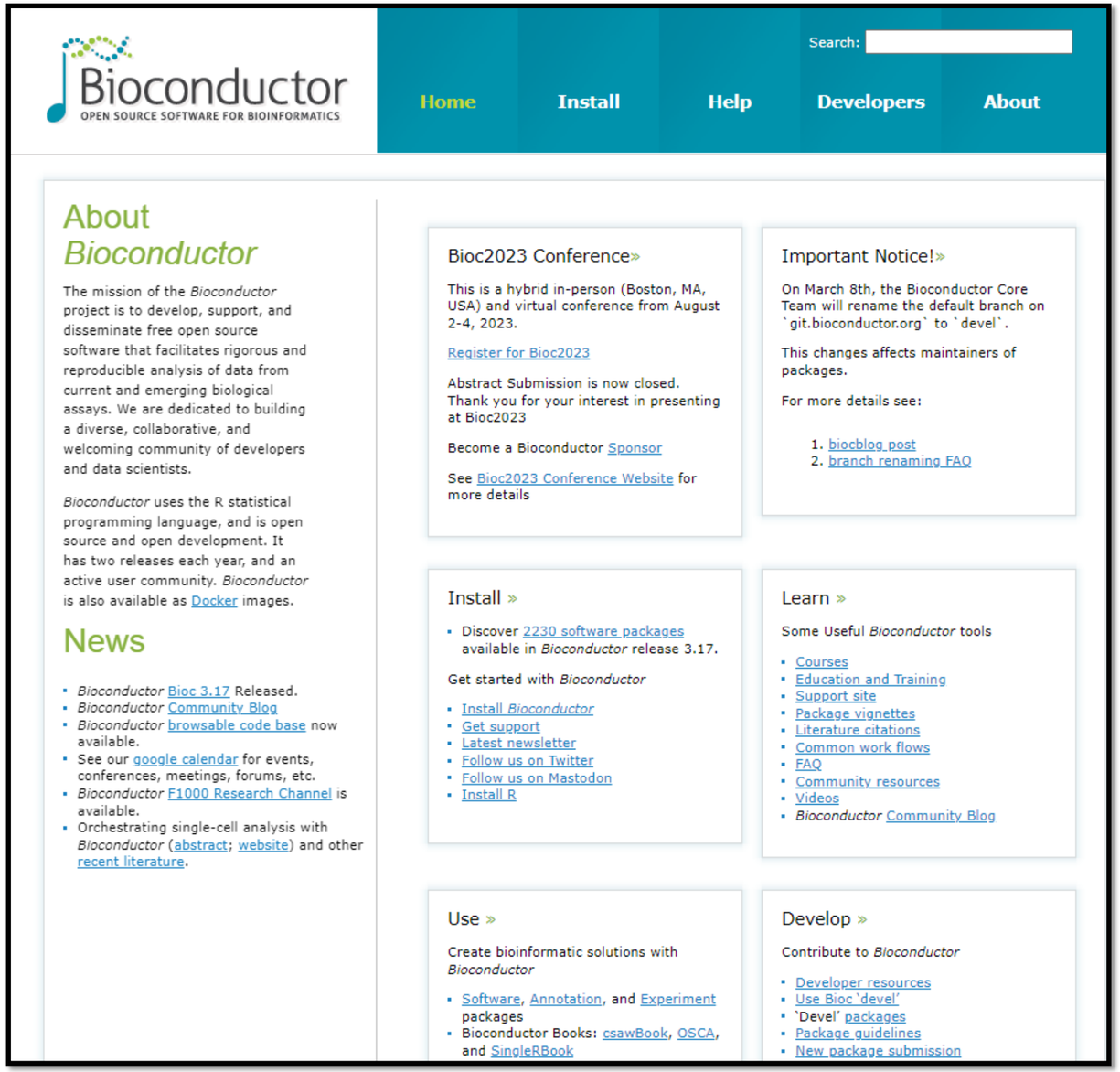
Any reports and responses or comments on the article can be found at the end of the article.

https://f1000research.com/articles/9-1444/v1

No R e Rstudio



No R e Rstudio



Software de bioinformática e dados biológicos

https://bioconductor.org/

No R e Rstudio

Instalar pacotes

Arquivo: instalar_pacotes.R

Instalar pacote individual do Bioconductor:

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

BiocManager::install("Glimma")
```

Instalar pacote individual do CRAN:

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

BiocManager::install("Glimma")
```

Iniciando as análises

#Fazer um arquivo.txt com o design do informação das amostras:

Sample_ID	Animal_ID	Tissue	Treatment
T17N	T17N	osso	controle
T42N	T42N	osso	controle
T44N	T44N	osso	controle
T16A	T16A	osso	afetado
T47A	T47A	osso	afetado
T49A	T49A	osso	afetado

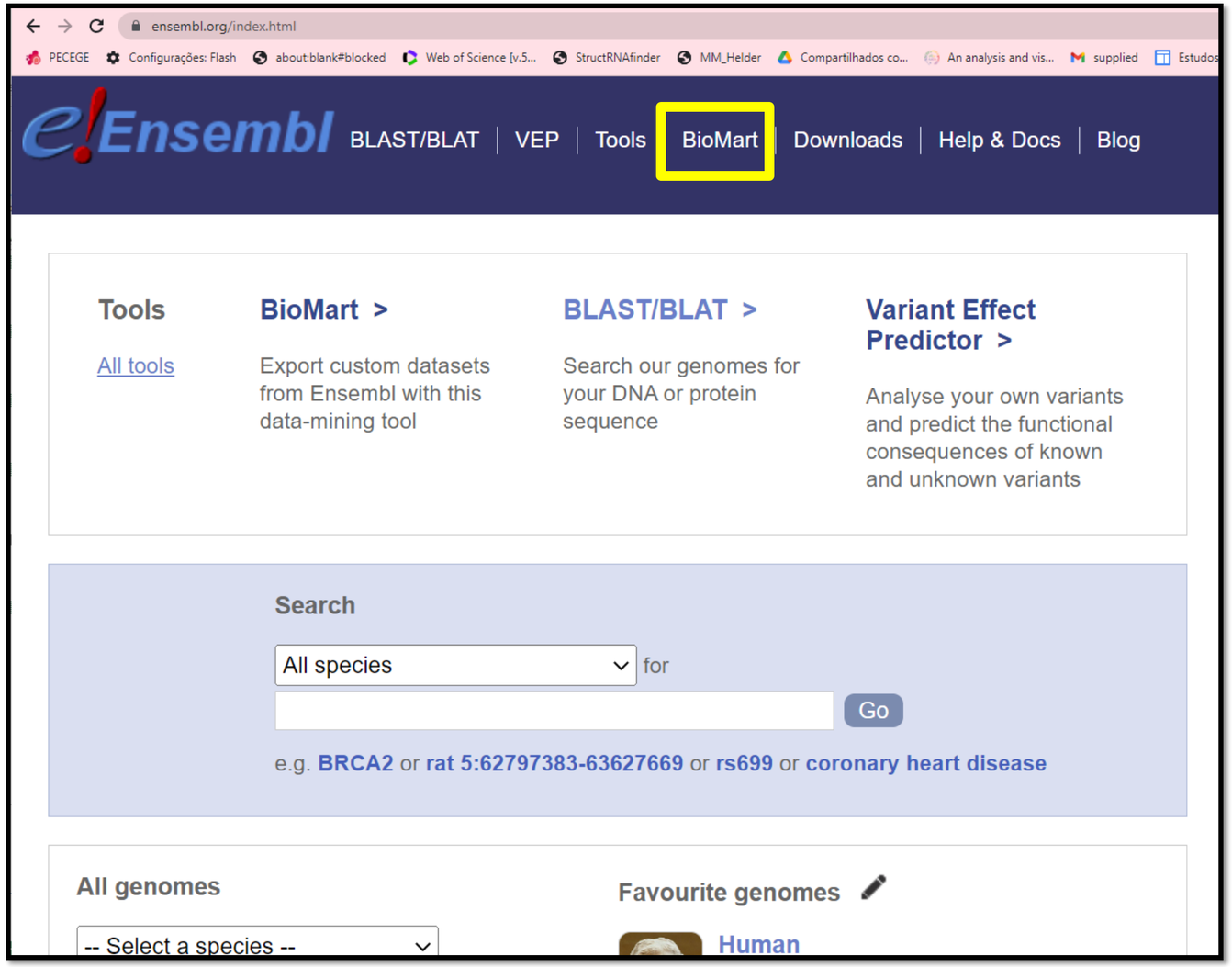
Ler o arquivo dentro do Rstudio e incluir informação dos arquivos de contagem

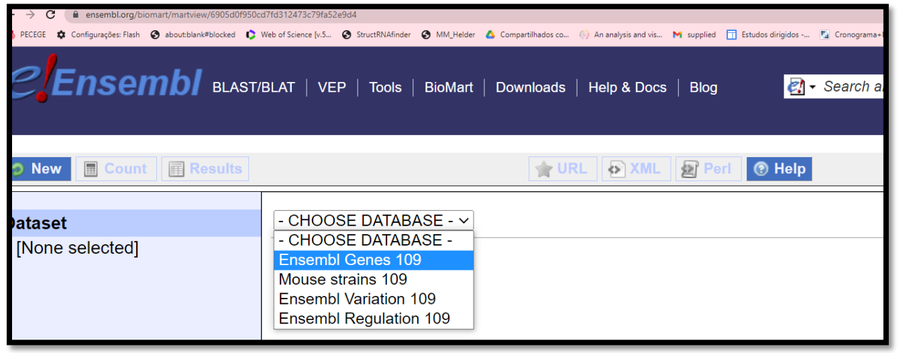
#pode ser feito diretamente no R, mas sempre fazemos um arquivo .txt

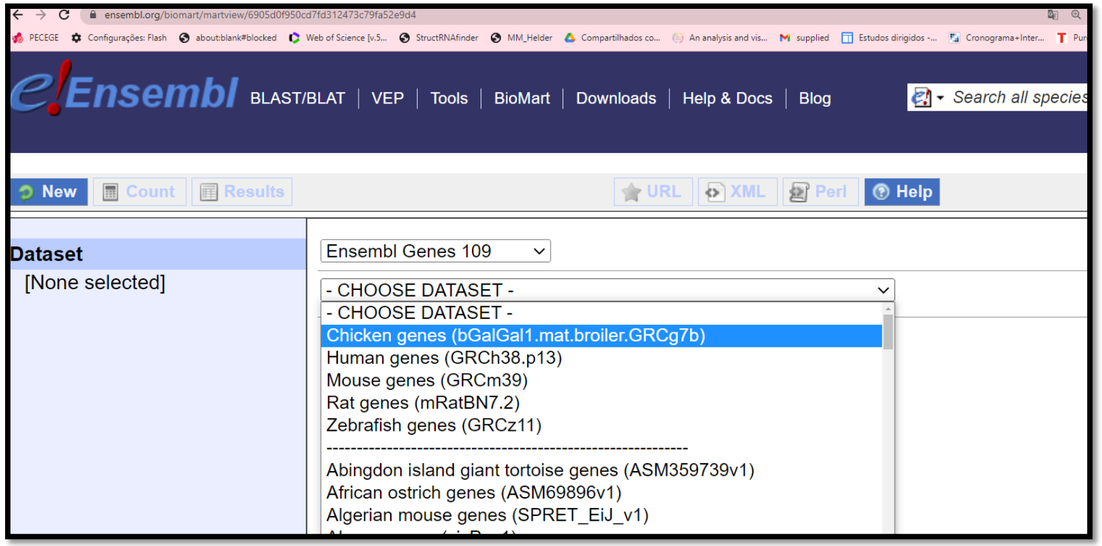
```
$ samples <- read.table("amostras.txt",header=T, as.is=T)
$ files <- c("04-GeneCountsSTAR/T16A_ReadsPerGene.counts",  "04-GeneCountsSTAR/T17N_ReadsPerGene.counts", "04-GeneCountsSTAR/T42N_ReadsPerGene.counts",
            "04-GeneCountsSTAR/T44N_ReadsPerGene.counts",   "04-GeneCountsSTAR/47A_ReadsPerGene.counts","04-GeneCountsSTAR/T49A_ReadsPerGene.counts")

$ amostras <- mutate(amostras, files)
```

	Sample_ID	Animal_ID	Tissue	Treatment	files
1	T16A	T16A	osso	afetado	04-GeneCountsSTAR/T16A_ReadsPerGene.counts
2	T17N	T17N	osso	controle	04-GeneCountsSTAR/T17N_ReadsPerGene.counts
3	T42N	T42N	osso	controle	04-GeneCountsSTAR/T42N_ReadsPerGene.counts
4	T44N	T44N	osso	controle	04-GeneCountsSTAR/T44N_ReadsPerGene.counts
5	T47A	T47A	osso	afetado	04-GeneCountsSTAR/47A_ReadsPerGene.counts
6	T49A	T49A	osso	afetado	04-GeneCountsSTAR/T49A_ReadsPerGene.counts

#Preparar os arquivos de anotação do biomart no site do Ensembl e no R







#Chamar as contagens e incluir informações na DGE-list

```r
#incluindo nome dos files no dataframe:

d <- readDGE(files, header = F)
d$counts <- d$counts[order(rownames(d$counts)),]
class(d)
dim(d$counts)
d.full <- d # objeto backup, caso queria voltar ao original

#Para ver se da match os dois dataframes na mesma ordem, pois counts e anotacao tem que estar na mesma ordem
match(genes$ensembl_gene_id,row.names(d$counts))
summary(match(genes$ensembl_gene_id,row.names(d$counts)))
```

edgeR e limma

```r
colnames(d) <- samplenames
head(d)

d$samples$group <- group

d$genes <- genes
d$genes
```

Avaliando as amostras

Contagem por milhão

Baseada no tamanho da biblioteca sequenciada

```
> y$samples

        group lib.size norm.factors
Sample1     1 10880519            1
Sample2     1  9314747            1
Sample3     1 11959792            1
Sample4     2  7460595            1
Sample5     2  6714958            1
```

CPM de 1 corresponde a 6-7 na menor amostra sequenciada

RPKM/FPKM e TPM

RPKM (Reads Per Kilobase Million)

$$RPKM\ of\ a\ gene = \frac{Number\ of\ reads\ mapped\ to\ a\ gene \times 10^3 \times 10^6}{Total\ number\ of\ mapped\ reads\ from\ given\ library \times gene\ length\ in\ bp}$$

Single-end

FPKM (Fragments Per Kilobase Million)

Paired-end

CPM (Copies Per Million)

$$\text{RPM or CPM} = \frac{\text{Number of reads mapped to gene} \times 10^6}{\text{Total number of mapped reads}}$$

Não leva em consideração tamanho do gene

TPM (Transcripts Per Million)

$$\text{TPM} = 10^6 * \frac{reads\ mapped\ to\ transcript\ /\ transcript\ length}{Sum\ (reads\ mapped\ to\ transcript\ /\ transcript\ length)}$$

Melhor para verificar a proporção das reads em cada gene

https://www.rna-seqblog.com/rpkm-fpkm-and-tpm-clearly-explained/, https://www.youtube.com/watch?v=S1PcT5rp8c4, https://www.biostars.org/p/273537/

TPM

Gene lengths	Genes	Technical replicate 1	Technical replicate 2	Technical replicate 3
1.5kb	Gene A	4.71	3.99	5.57
2kb	Gene B	5.29	6	4.426
		9.99	9.99	9.99

FPKM

Gene lengths	Genes	Technical replicate 1	Technical replicate 2	Technical replicate 3
1.5kb	Gene A	6.66	5.55	14.16
2kb	Gene B	7.5	8.33	11.25
		14.16	13.88	25.41

https://www.youtube.com/watch?v=S1PcT5rp8c4

Genes pouco expressos

Filtrar os genes com poucas reads

Mas.. O que considerar pouco expresso ?

```
$ keep = rowSums(cpms >1) >=5 & !noint
$ counts = counts[keep,]
$ colnames(counts) = samples$SAMPLE_ID
$ dim(counts)
```

Não há um consenso

Depende de cada experimento

```
> keep <- rowSums(cpm(y) > 0.5) >= 2
> table(keep)
```

Sempre fazer em cpm

Genes pouco expressos

```
keep.exprs <- filterByExpr(x, group=group)
x <- x[keep.exprs,, keep.lib.sizes=FALSE]
dim(x)
```

edgeR

`y`	matrix of counts, or a `DGEList` object, or a `SummarizedExperiment` object.
`design`	design matrix. Ignored if `group` is not `NULL`.
`group`	vector or factor giving group membership for a oneway layout, if appropriate.
`lib.size`	library size, defaults to `colSums(y)`.
`min.count`	numeric. Minimum count required for at least some samples.
`min.total.count`	numeric. Minimum total count required.
`large.n`	integer. Number of samples per group that is considered to be "large".
`min.prop`	numeric. Minimum proportion of samples in the smallest group that express the gene.
`...`	any other arguments. For the `DGEList` and `SummarizedExperiment` methods, other arguments will be passed to the default method. For the default method, other arguments are not currently used.

Normalizando para o tamanho das bibliotecas

```
x <- calcNormFactors(x, method = "TMM")
x$samples$norm.factors

## [1] 0.894 1.025 1.046 1.046 1.016 0.922 0.996 1.086 0.984
```

Scaling factor

Não altera o tamanho das bibliotecas

edgeR e limma ➔ TMM

```
calcNormFactors(object,
                method = c("TMM","TMMwsp","RLE","upperquartile","none"),
           refColumn = NULL, logratioTrim = .3, sumTrim = 0.05, doWeighting = TRUE,
                Acutoff = -1e10, p = 0.75, ...)
```

Como o edgeR faz a normalização das amostras ??

Genes	Amostras	
	T16A	T17N
ENSGALG00010000002	0	0
ENSGALG00010000003	2832	3255
ENSGALG00010000004	0	0
ENSGALG00010000005	2084	2129
ENSGALG00010000006	0	0
ENSGALG00010000007	10000	17142
ENSGALG00010000008	0	0
ENSGALG00010000009	0	0
ENSGALG00010000010	0	0
ENSGALG00010000012	0	0
ENSGALG00010000013	0	0
ENSGALG00010000014	0	0
ENSGALG00010000015	0	0
ENSGALG00010000016	0	0
ENSGALG00010000017	106566	135908
ENSGALG00010000018	0	0
Total de reads	**121482**	**158434**

1. Remove os genes pouco expressos

Genes	Amostras	
	T16A	T17N
ENSGALG00010000003	2832	3255
ENSGALG00010000005	2084	2129
ENSGALG00010000007	10000	17142
ENSGALG00010000017	106566	135908
Total de reads	**121482**	**158434**

2. Escolhe uma amostra referência

Scaling fator ➔ amostra mais mediana

Amostras	
T16A	T17N
2832/121482	3255/158434
2084/121482	2129/158434
10000/121482	17142/158434
10324/121482	12844/158434

Amostras	
T16A	T17N
0,023	0,021
0,017	0,013
0,082	0,108
0,877	0,858

Média Q75% = 0,095

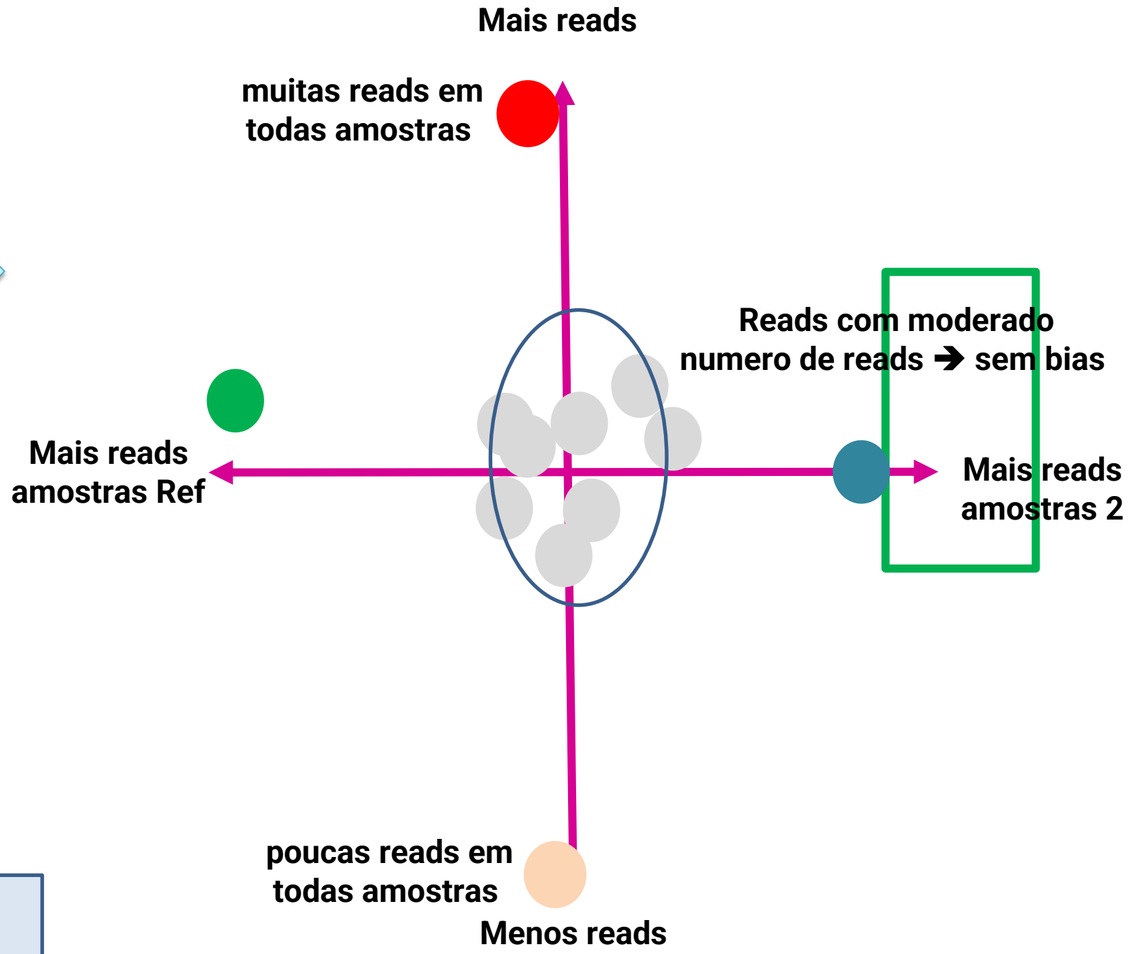
AMOSTRA REFERÊNCIA

2.1 Calcula o quartil de 75%
2.2. Faz a média dos quartis de 75%
2.3. Escolhe a amostra que tem o valor mais próximo da média dos quartis75%

Como o edgeR faz a normalização das amostras ??

3. Escolhe genes para fazer o *scaling factor*

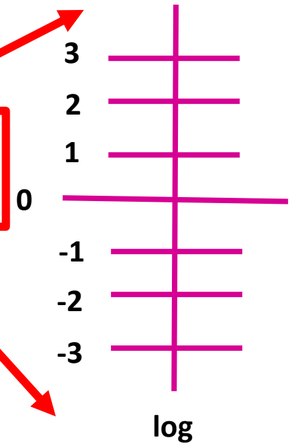
Amostras		Amostras	
T16A	T17N	T16A	T17N
2832/121482	3255/158434	0,023	0,021
2084/121482	2129/158434	0,017	0,013
10000/121482	17142/158434	0,082	0,108
10324/121482	12844/158434	0,877	0,858



edgeR filtra os genes “tendenciosos”

➔ log fold das diferenças

Remoção dos genes com valores extremos



$$log2\left(\frac{ref, gene\ 1}{amostra\ 2, gene\ 1}\right)$$

Como o edgeR faz a normalização das amostras ??

3. Escolhe genes para fazer o *scaling factor*

$$log2\left(\frac{ref, gene\ 1}{amostra\ 2, gene\ 1}\right)$$

Remove genes com log2 Inf

Tabela 1 ➔ genes tendenciosos

$$\left(\frac{log2(ref, gene\ 1) + log2(\#2, gene\ 1)}{2}\right)$$

Calcula média geométrica dos logs para todos os genes e amostras

Tabela 2 ➔ Verifica genes altamente expressos e pouco expressos

Ordena em ordem crescente

Filtra 30% dos genes de cada extremo

Filtra 5% dos genes de cada extremo

Genes comuns ➔ scaling factors

Como o edgeR faz a normalização das amostras ??

Genes comuns ➔ scaling factors

4. Calcula a média ponderada das razões de log2 remanescentes

“WEIGHTED TRIMMED MEAN OF THE LOG2 RATIOS”

“Trima” ou filtra os genes extremos

DIMINUI O EFEITO DE OUTLIERS

Como o edgeR faz a normalização das amostras ??

Genes que passaram no filtro

4. Calcula a média ponderada das razões de log2 remanescentes

Genes com mais reads ➔ mais peso

Genes pouco expressos ➔ > variância

	Sample #1	Sample #2	log_2(ratio)
Gene #1	202	101	1
Gene #2	204	101	1.01
Gene #3	206	101	1.02
Gene #4	2	1	1
Gene #5	4	1	2
Gene #6	6	1	2.6

5. Calcula a média ponderada das razões de log2 "normal numbers"

$2^{média\ ponderada\ das\ log_2\ \#2}$ = *scaling factor*

faz para todas amostras

Valores são centrados em torno de 1

Centered scaling factor

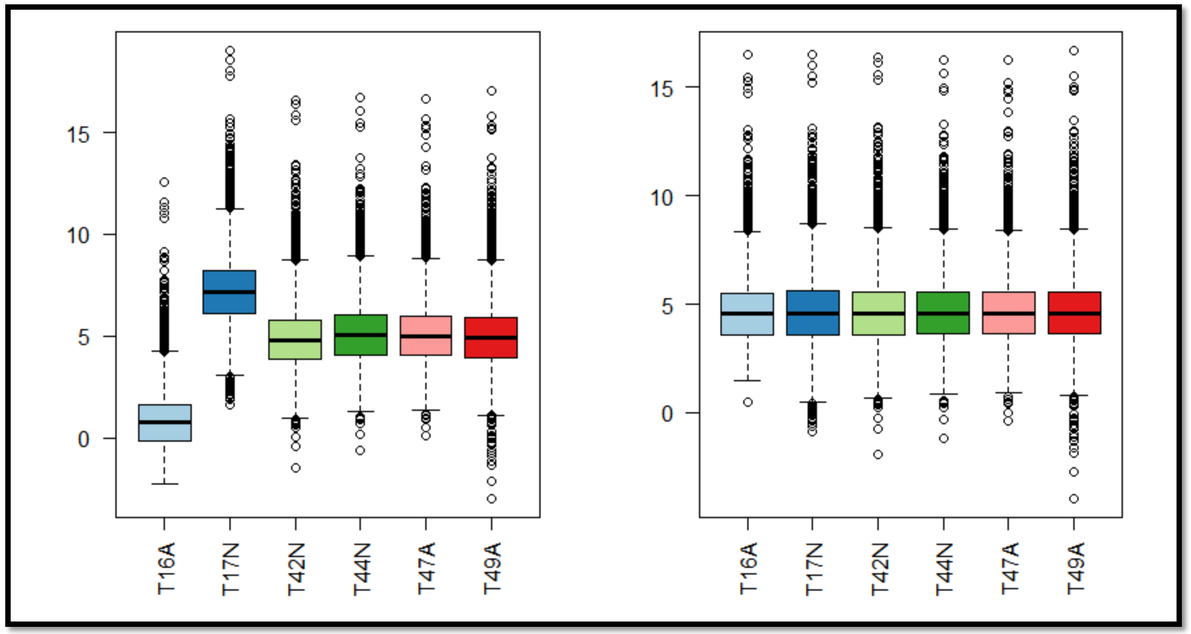
$$\left(\frac{raw\ scaling\ factor}{média\ geométrica\ da\ amostras}\right)$$

ANTES

```
> d$samples
                                                files     group lib.size norm.factors
T16A 04-GeneCountsSTAR/T16A_ReadsPerGene.counts  afetado 11447263            1
T17N 04-GeneCountsSTAR/T17N_ReadsPerGene.counts controle 17097830            1
T42N 04-GeneCountsSTAR/T42N_ReadsPerGene.counts controle 13150041            1
T44N 04-GeneCountsSTAR/T44N_ReadsPerGene.counts controle 21096609            1
T47A 04-GeneCountsSTAR/T47A_ReadsPerGene.counts  afetado 15119253            1
T49A 04-GeneCountsSTAR/T49A_ReadsPerGene.counts  afetado 18723518            1
```

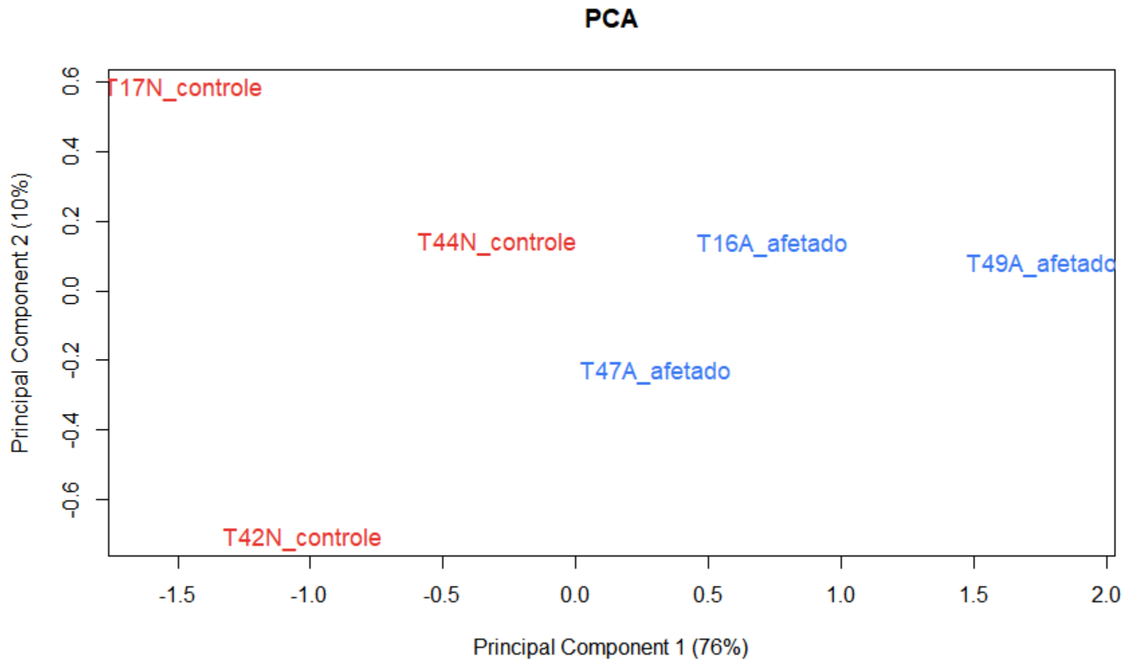
DEPOIS

```
> d$samples
                                                files     group lib.size norm.factors
T16A 04-GeneCountsSTAR/T16A_ReadsPerGene.counts  afetado 11447263    1.0067830
T17N 04-GeneCountsSTAR/T17N_ReadsPerGene.counts controle 17097830    0.9470692
T42N 04-GeneCountsSTAR/T42N_ReadsPerGene.counts controle 13150041    0.9366670
T44N 04-GeneCountsSTAR/T44N_ReadsPerGene.counts controle 21096609    1.0818763
T47A 04-GeneCountsSTAR/T47A_ReadsPerGene.counts  afetado 15119253    1.0604755
T49A 04-GeneCountsSTAR/T49A_ReadsPerGene.counts  afetado 18723518    0.9759306
```

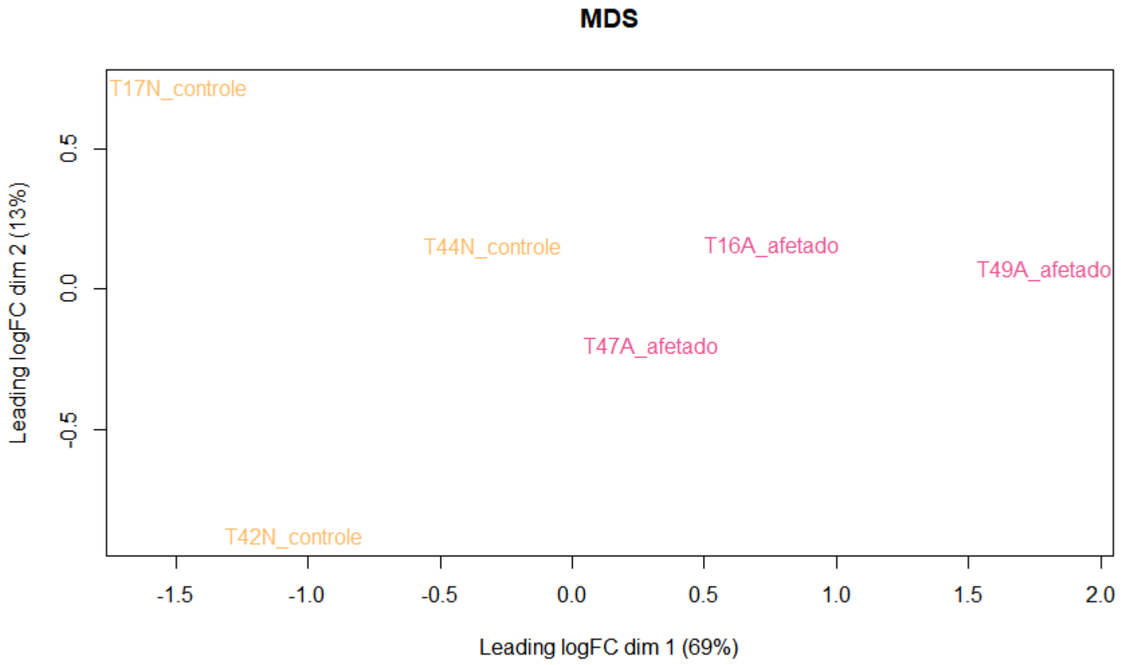


Verificando a distribuição das amostras

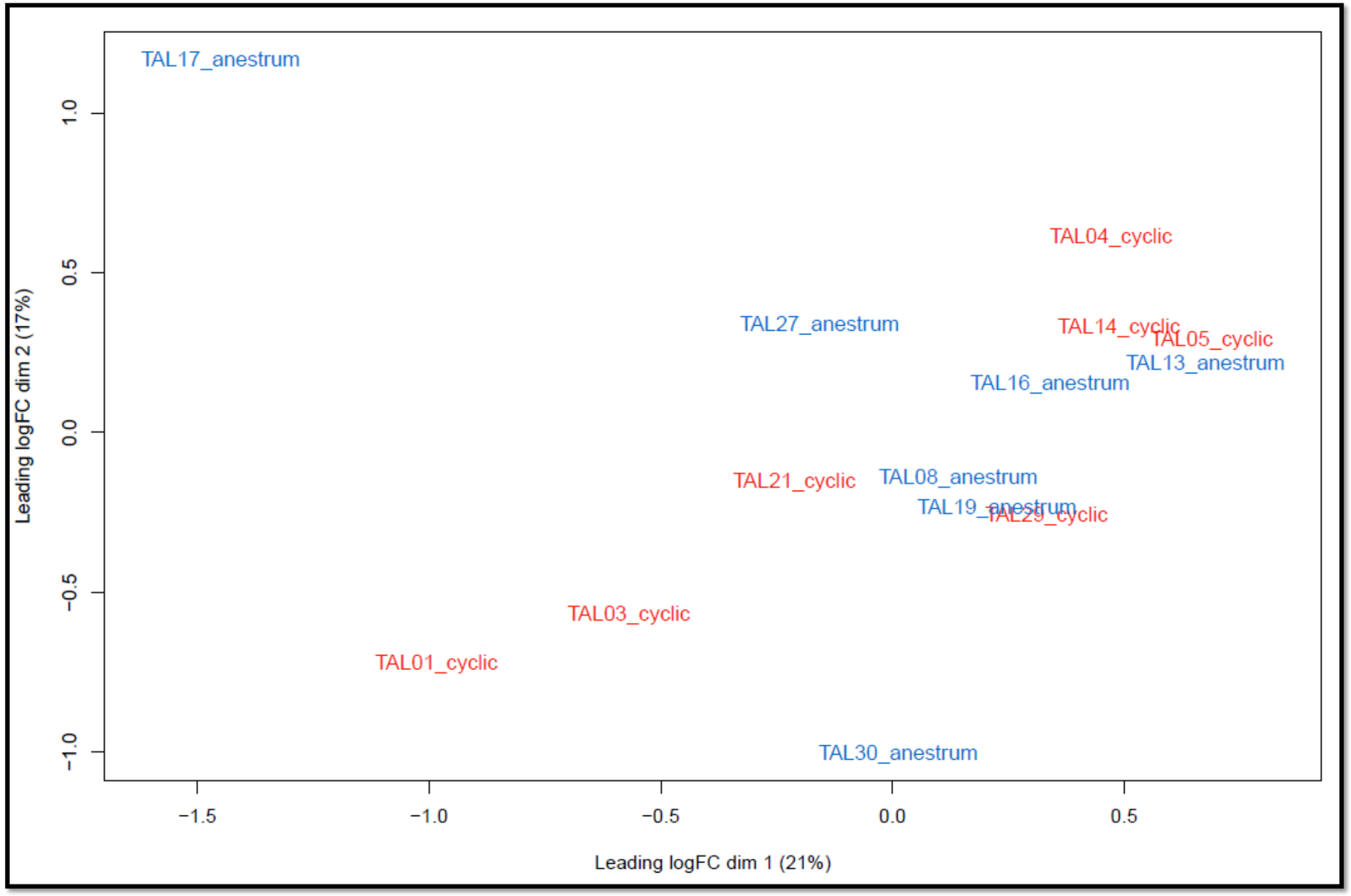
PCA



MDS



E neste caso ????


TAL17_anestrum
TAL04_cyclic
TAL27_anestrum
TAL14_cyclic
TAL05_cyclic
TAL13_anestrum
TAL16_anestrum
TAL21_cyclic
TAL08_anestrum
TAL19_anestrum
TAL29_cyclic
TAL03_cyclic
TAL01_cyclic
TAL30_anestrum
Leading logFC dim 2 (17%)
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
Leading logFC dim 1 (21%)

Criando a matriz de design e contrastes

Várias formas de fazer ➔ modelo de análise

```r
```{r}
design <- model.matrix(~0+group)
colnames(design) <- gsub("group", "", colnames(design))
rownames(design) <- samplenames

contr.matrix <- makeContrasts("Afetado-Normal"=afetado-normal, levels=design)
```
```

F1000Research F1000Research 2020, 9:1444 Last updated: 18 JUL 2022

Check for updates

METHOD ARTICLE

A guide to creating design matrices for gene expression experiments [version 1; peer review: 2 approved]

Charity W. Law[1,2], Kathleen Zeglinski[1,3], Xueyi Dong[1,2], Monther Alhamdoosh[3], Gordon K. Smyth[1,4], Matthew E. Ritchie[1,2,4]

[1]The Walter and Eliza Hall Institute of Medical Research, Parkville, 3052, Australia
[2]Department of Medical Biology, The University of Melbourne, Parkville, 3010, Australia
[3]Research and Development, CSL Limited, Bio21 Institute, Parkville, 3010, Australia
[4]School of Mathematics and Statistics, The University of Melbourne, Parkville, 3010, Australia

v1 **First published:** 10 Dec 2020, **9**:1444
https://doi.org/10.12688/f1000research.27893.1
Latest published: 10 Dec 2020, **9**:1444
https://doi.org/10.12688/f1000research.27893.1

Abstract

Differential expression analysis of genomic data types, such as RNA-sequencing experiments, use linear models to determine the size and direction of the changes in gene expression. For RNA-sequencing, there are several established software packages for this purpose accompanied with analysis pipelines that are well described. However, there are two crucial steps in the analysis process that can be a stumbling block for many -- the set up an appropriate model via design matrices and the set up of comparisons of interest via contrast matrices. These steps are particularly troublesome because an extensive catalogue for design and contrast matrices does not currently exist. One would usually search for example case studies across different platforms and mix and match the advice from those sources to suit the dataset they have at hand. This article guides the reader through the basics of how to set up design and contrast matrices. We take a practical approach by providing code and graphical representation of each case study, starting with simpler examples (e.g. models with a single explanatory variable) and move onto more complex ones (e.g. interaction models, mixed effects models, higher order time series and cyclical models). Although our work has been written specifically with a **limma**-style pipeline in mind, most of it is also applicable to other software packages for differential

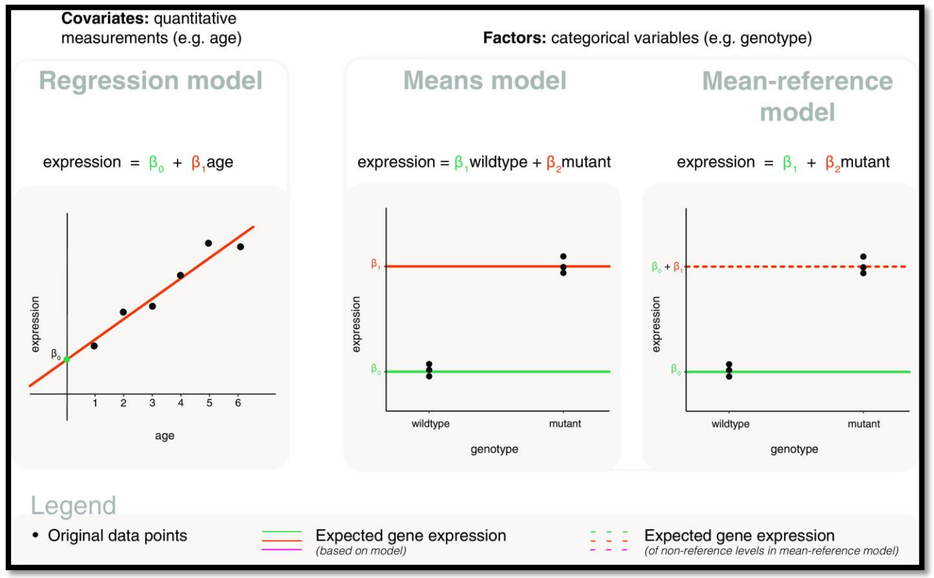
Open Peer Review

Approval Status ✓ ✓

	1	2
version 1 10 Dec 2020	✓ view	✓ view

1. **Federico Marini**, University Medical Center of the Johannes Gutenberg University Mainz, Mainz, Germany
2. **Jenny Drnevich**, University of Illinois, Urbana-Champaign, USA

Any reports and responses or comments on the article can be found at the end of the article.



Explanatory variables	Design matrix	Section
age	model.matrix(~age)	Covariates: With intercept
	model.matrix(~0+age)	Covariates:Without intercept
group *HEALTHY, SICK*	model.matrix(~group)	Factors: With intercept
	model.matrix(~0+group)	Factors: Without intercept

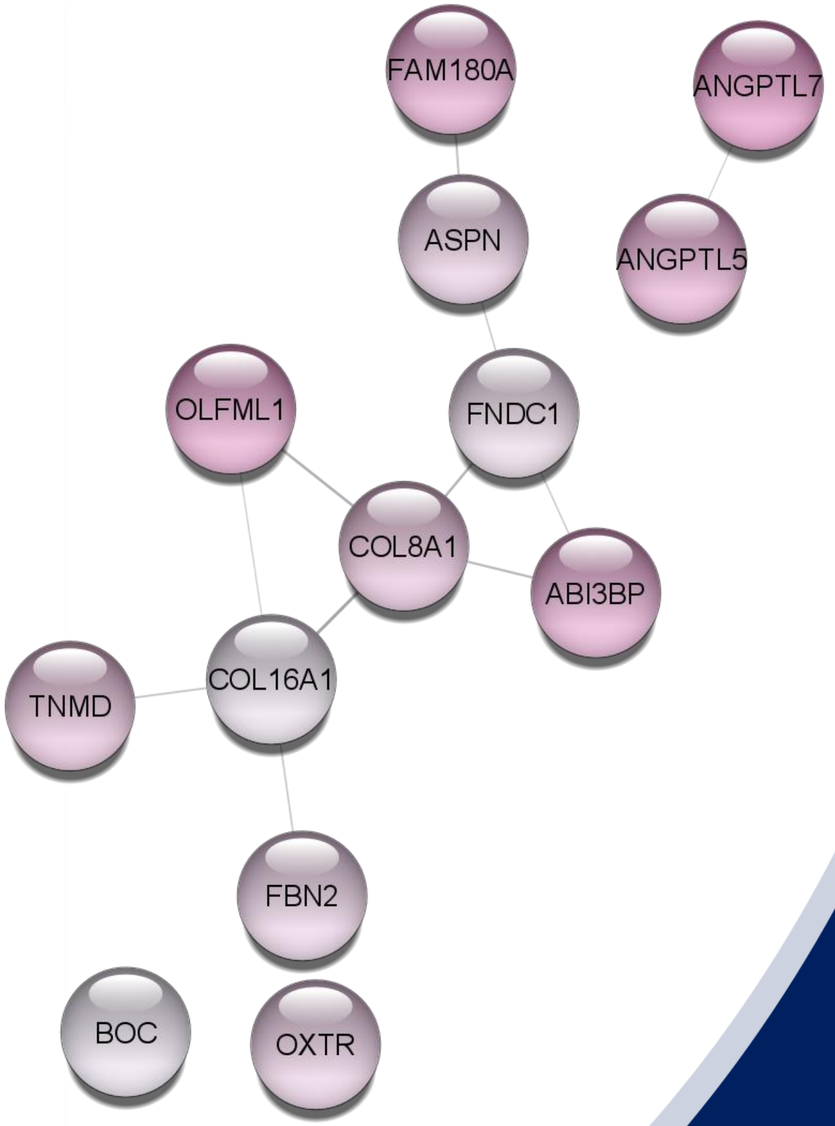
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7873980/

Correção de múltiplos testes

Testes dos genes não é independente ➔ tem que considerar na análise

Benjamini and Hochberg False Discovery Rate

Common Methods Ranked by Stringency	Mais falso negativos
Bonferroni	
Bonferroni Step-Down	
Westfall and Young Permutation	
Benjamini and Hochberg False Discovery Rate	Mais falsos positivos
None	



Análises Funcionais

Achei meus genes DE, e agora???

UM MUNDO DE OPÇÕES SE ABRE!!!!!!

Quais genes eu vou estudar ?
Quantos genes tenho que avaliar?
O que esses genes fazem?
De onde vieram ????
E onde vivem??

Análises funcionais

Análise de enriquecimento fucional

Análise de vias metabólicas

Análise de Redes Gênicas

Integração com outros dados, …….

MÉTODOS DE ENRIQUECIMENTOS MAIS COMUNS:

Over-representation analysis (ORA)

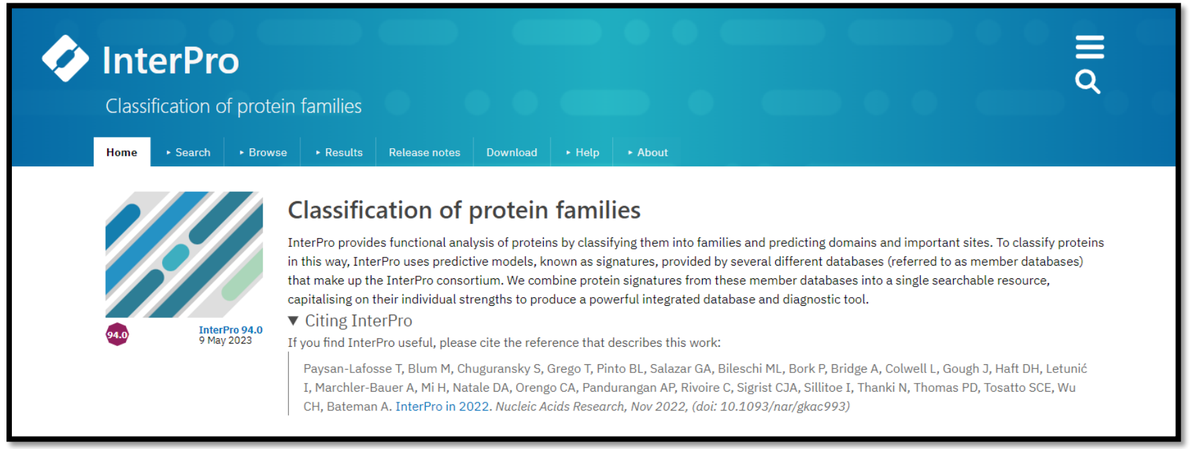
- **Genes diferencialmente expressos**
- **Genes expressos (*background*)**

Functional class scoring (FCS)

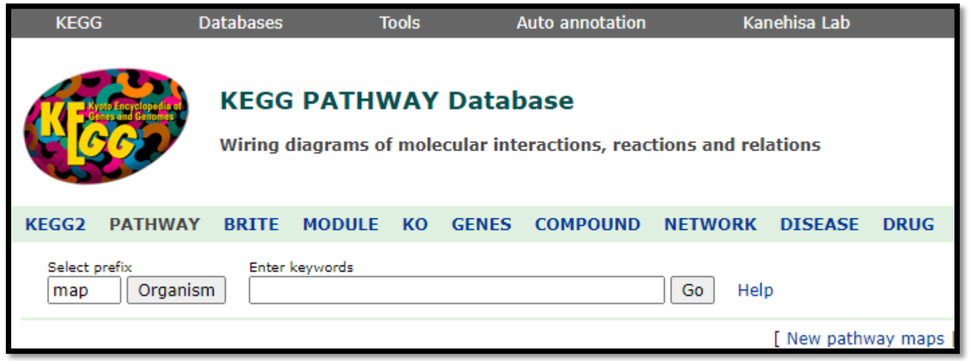
- **Genes diferencialmente expressos + níveis de expressão (logFC)**
- **Genes expressos (*background*)**
- **Gene Set Enrichment Analysis (GSEA)**

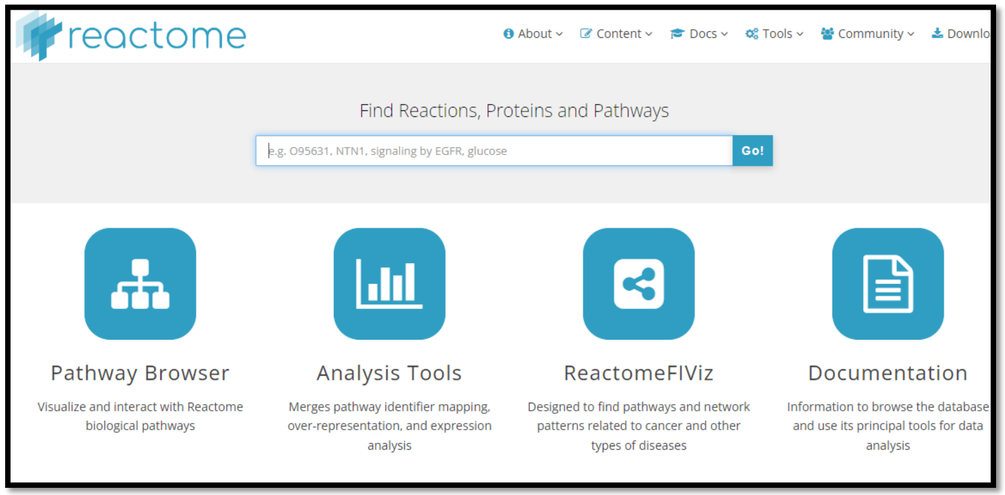
Análises funcionais

Ontologias gênicas



Vias metabólicas





Ontologias gênicas

Gene Ontology overview

An ontology is a formal representation of a body of knowledge within a given domain. Ontologies usually consist of a set of classes (or terms or concepts) with relations that operate between them. The Gene Ontology (GO) describes our knowledge of the biological domain with respect to three aspects:

Molecular Function — Molecular-level activities performed by gene products. Molecular function terms describe activities that occur at the molecular level, such as "catalysis" or "transport". GO molecular function terms represen…an the entities (molecules or complexes) that perform the actions, and do not specify where, when, or in what context the action takes place. Molecular functions generally …can be performed by individual gene products (*i.e.* a protein or RNA), but some activities are performed by molecular complexes composed of multiple gene products. Example…l terms are *catalytic activity* and *transporter activity*; examples of narrower functional terms are *adenylate cyclase activity* or *Toll-like receptor binding*. To avoid confusion between gene product names and their molecular functions, GO molecular functions are often appended with the word "activity" (a *protein kinase* would have the GO molecular function *protein kinase activity*).

Cellular Component — …lar structures in which a gene product performs a function, either cellular compartments (*e.g.*, *mitochondrion*), or stable macromolecular complexes of which they are parts (*e.g.*, the…e other aspects of GO, cellular component classes refer not to processes but rather a cellular anatomy.

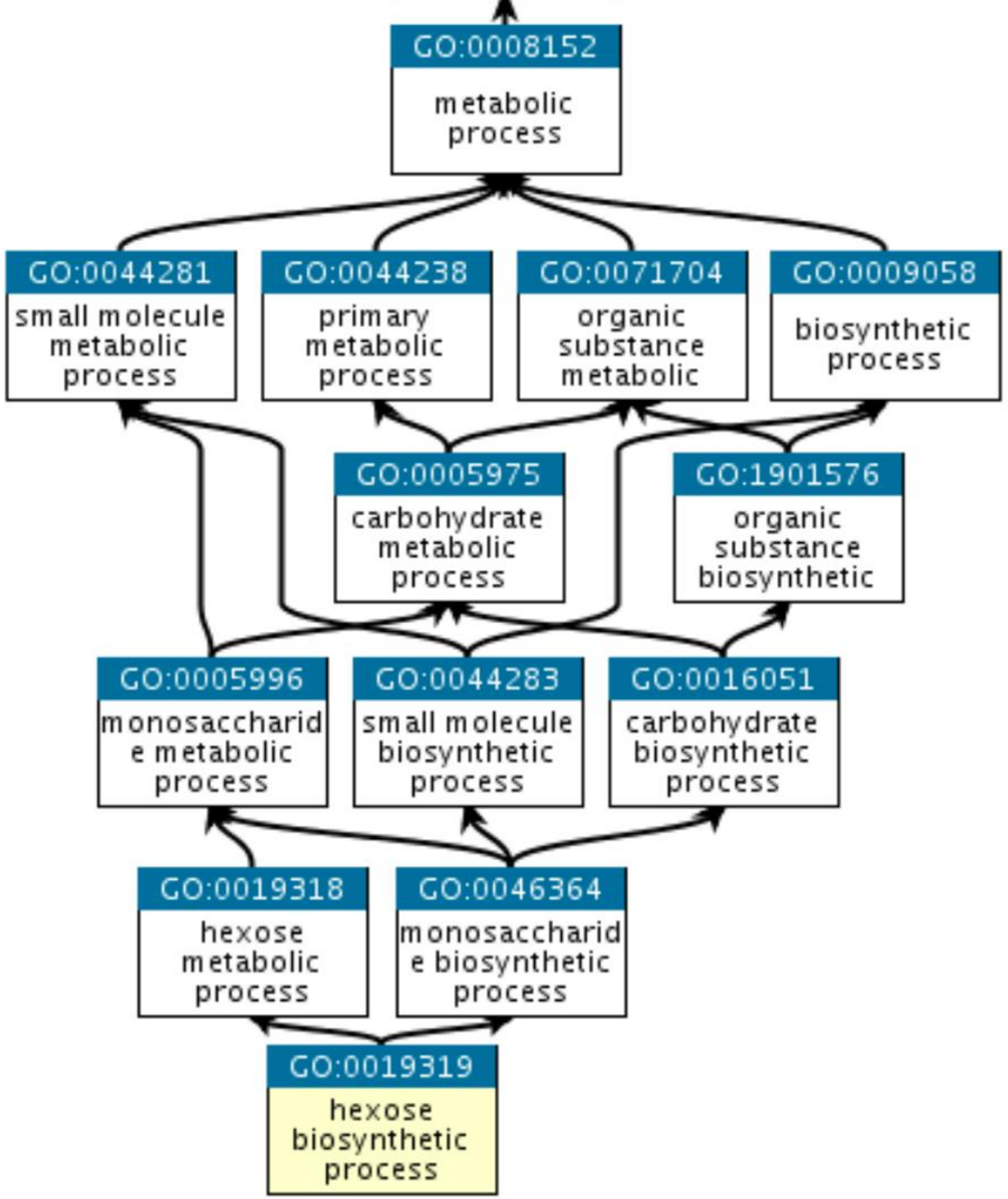
Biological Process — The larger processes, or 'biological p…ed by multiple molecular activities. Examples of broad biological process terms are *DNA repair* or *signal transduction*. Examples of more specific terms are *pyrimidine nucle…glucose transmembrane transport*. Note that a biological process is not equivalent to a pathway. At present, the GO does not try to represent the dynamics or dependencies that woul… describe a pathway.

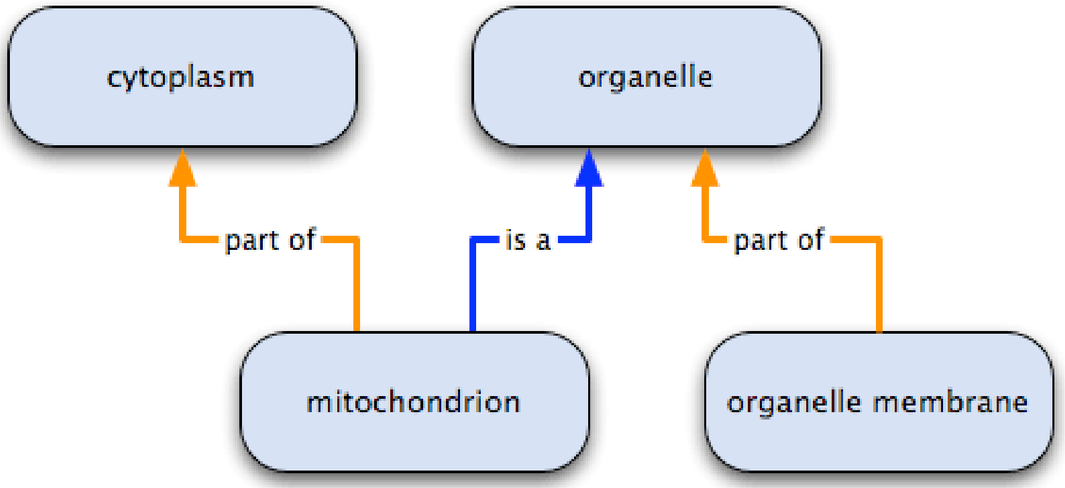
MF

CC

BP

http://geneontology.org/docs/ontology-documentation/



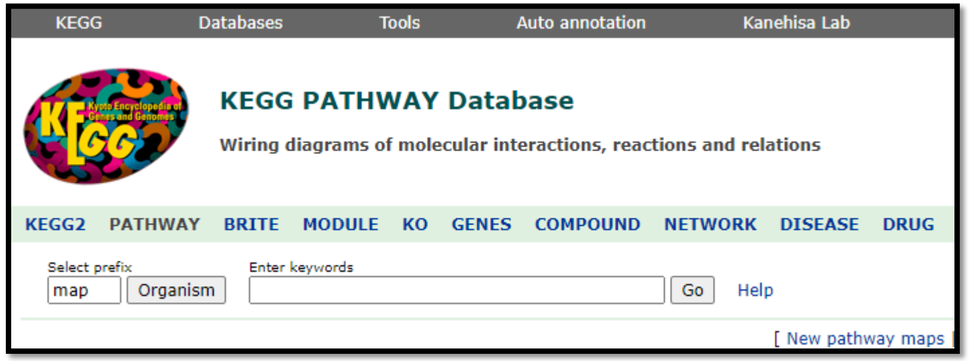


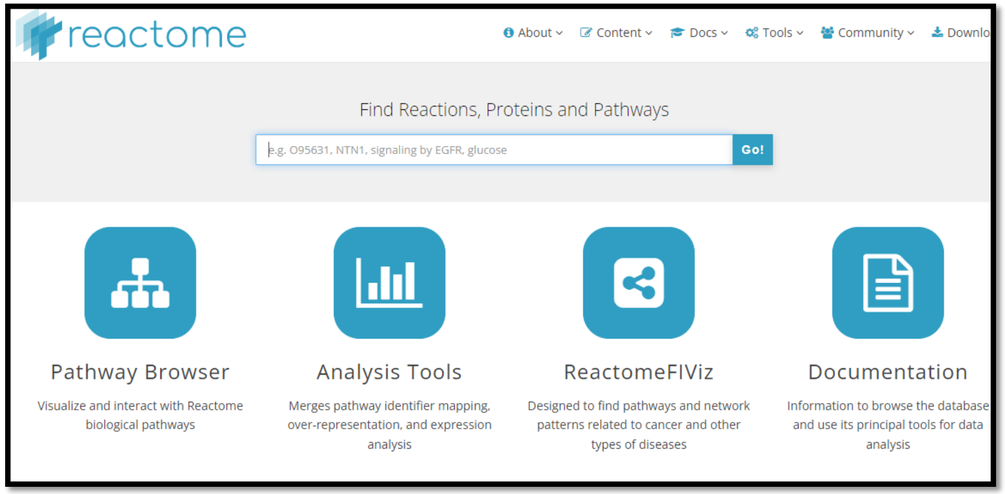
http://geneontology.org/docs/ontology-documentation/

BASES DE DADOS MAIS COMUNS

Ontologias gênicas

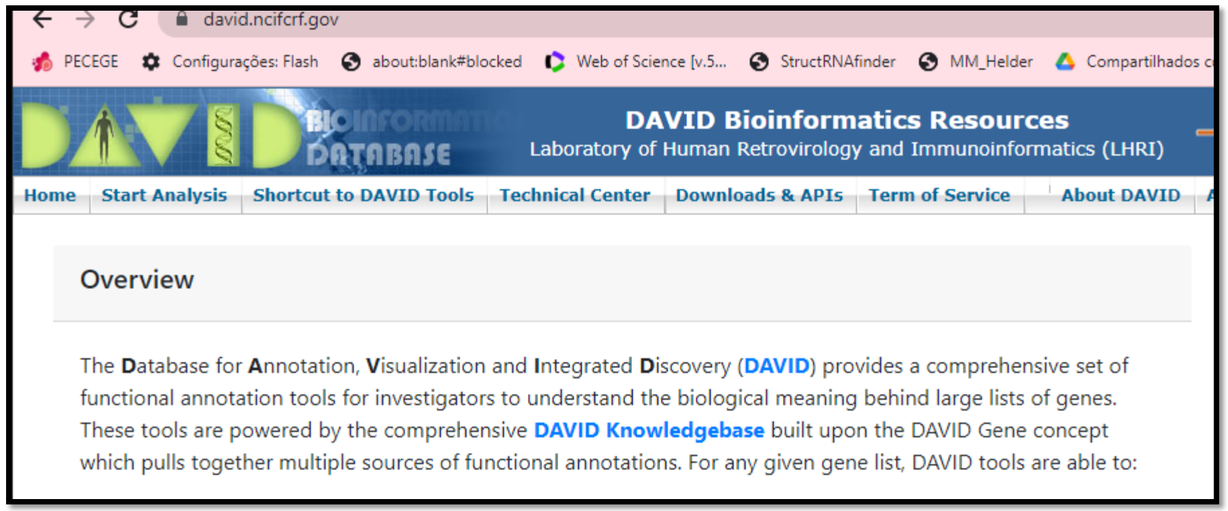
Vias metabólicas

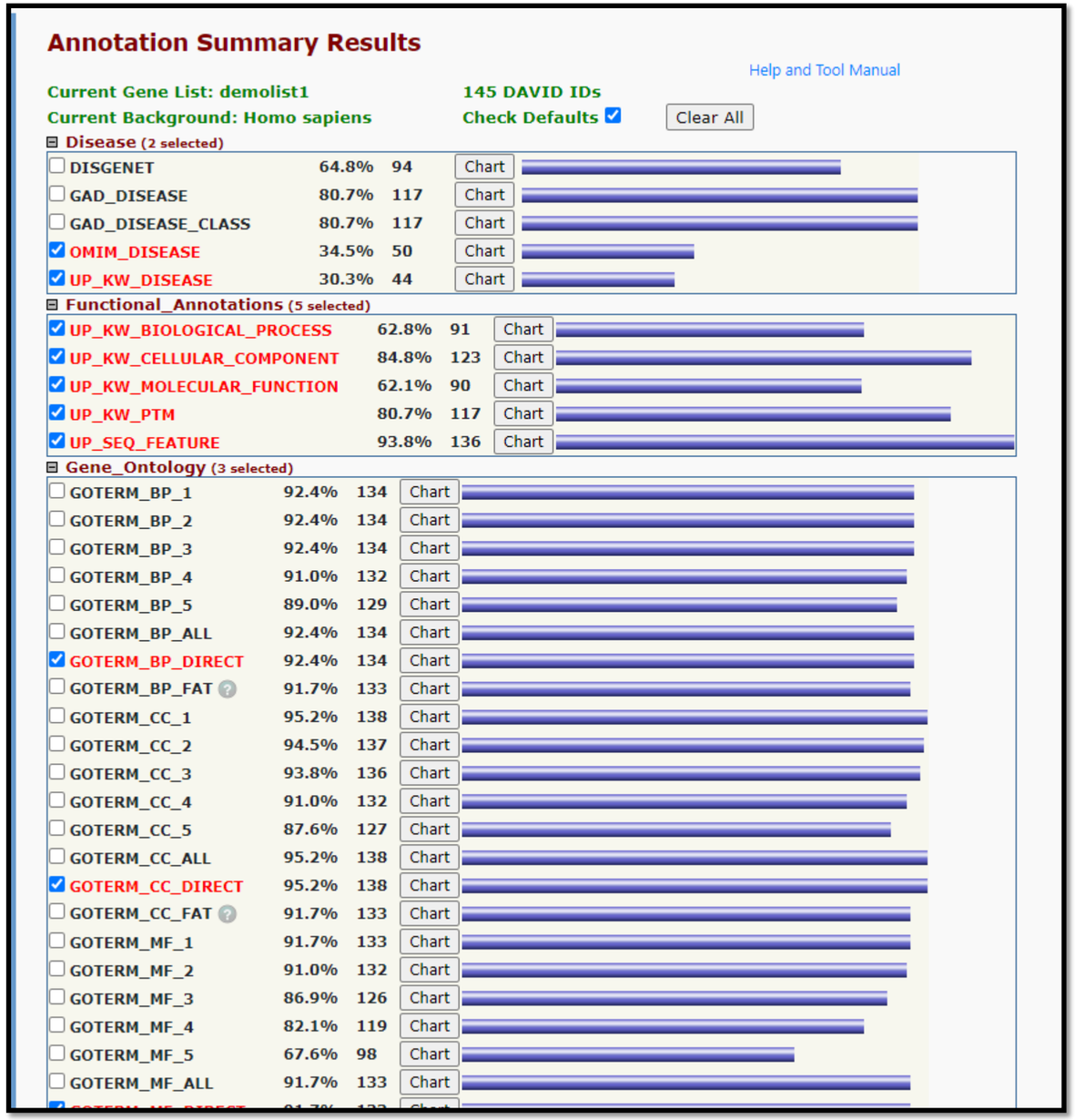




BASES DE DADOS MAIS COMUNS

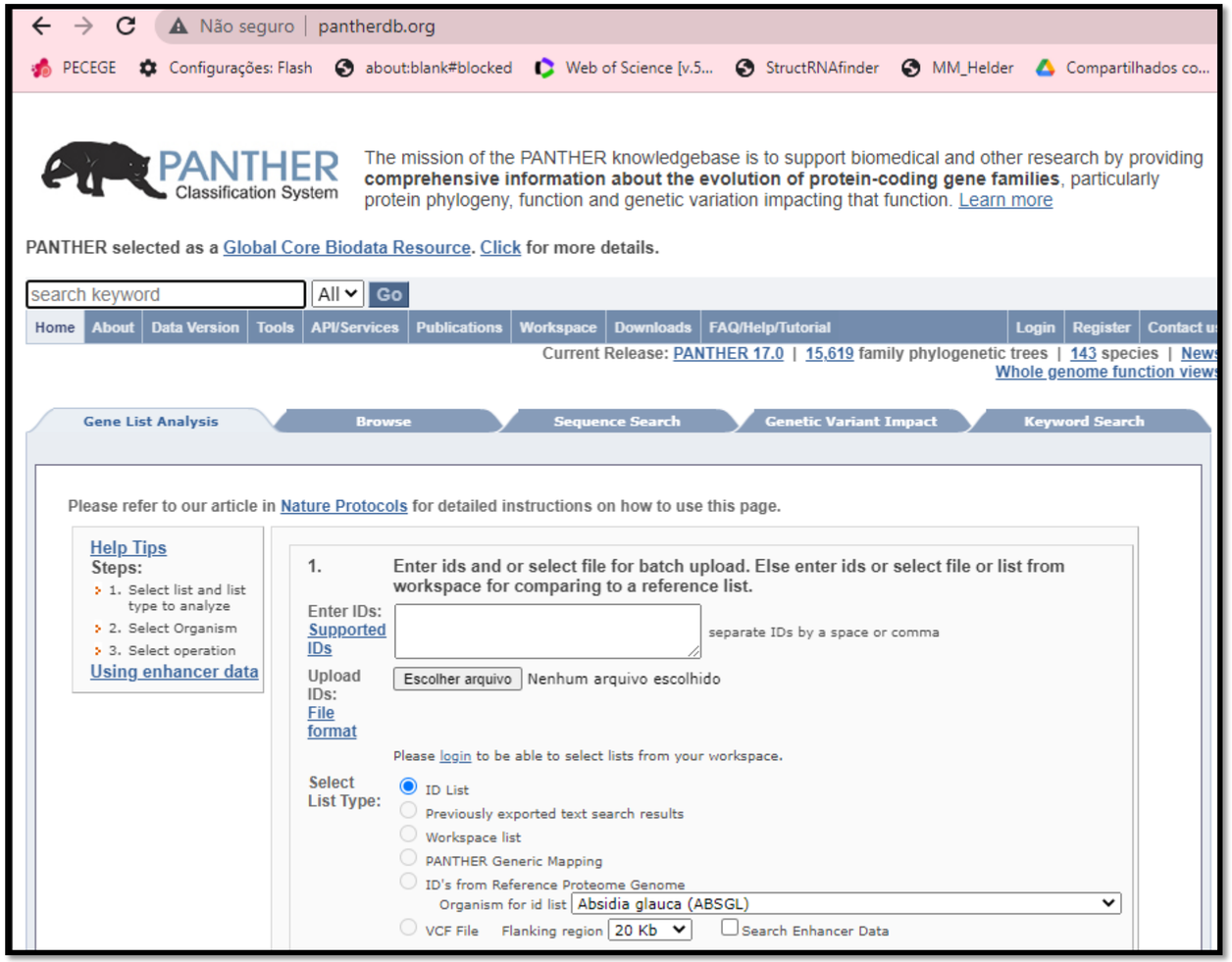
DAVID

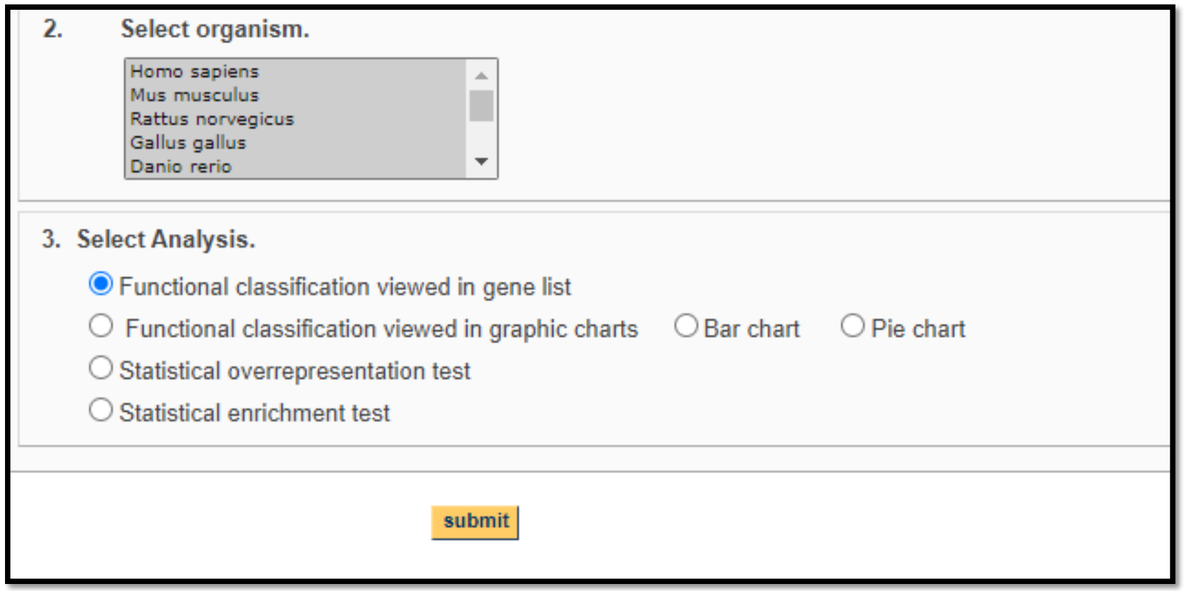




BASES DE DADOS MAIS COMUNS

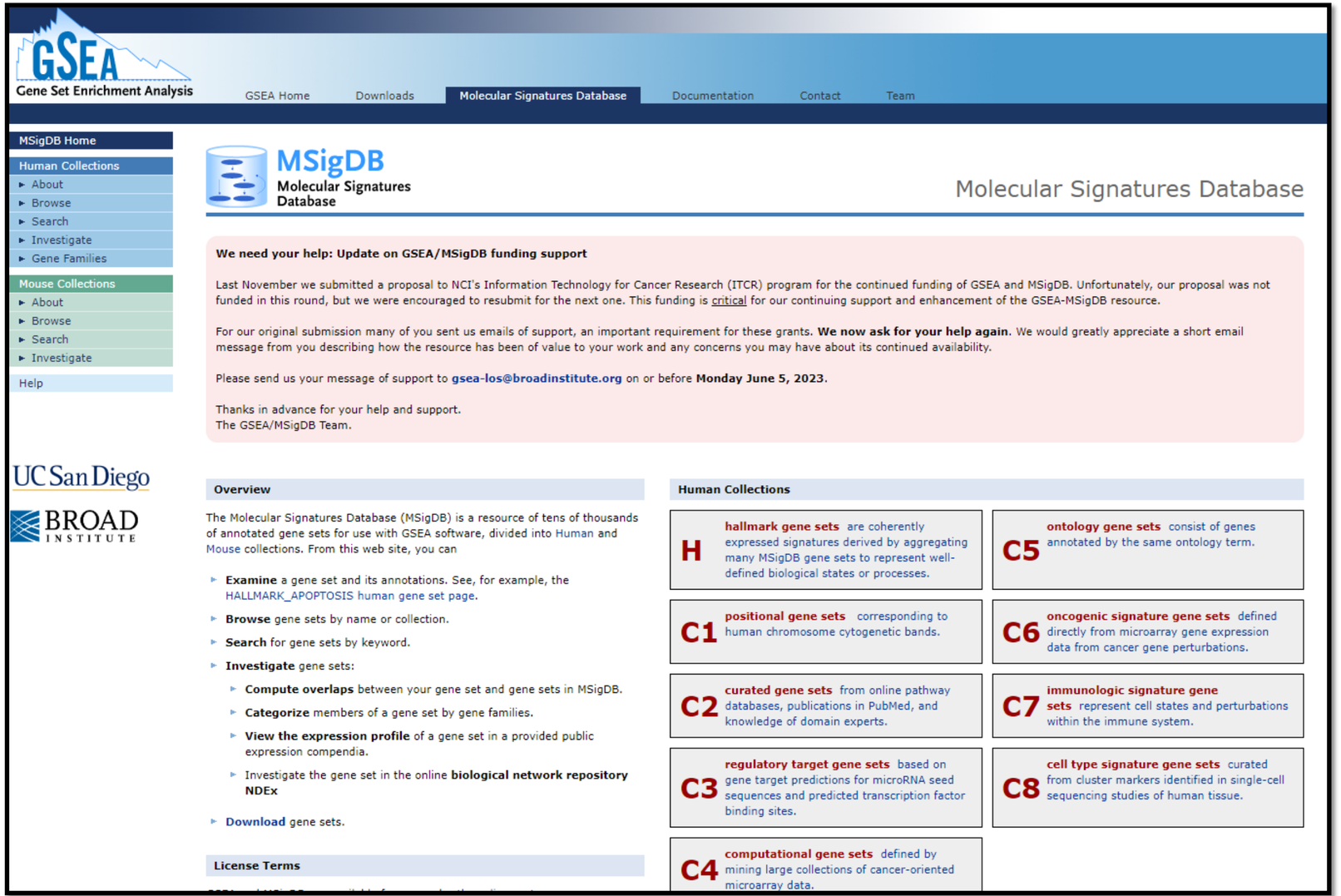
PANTHER



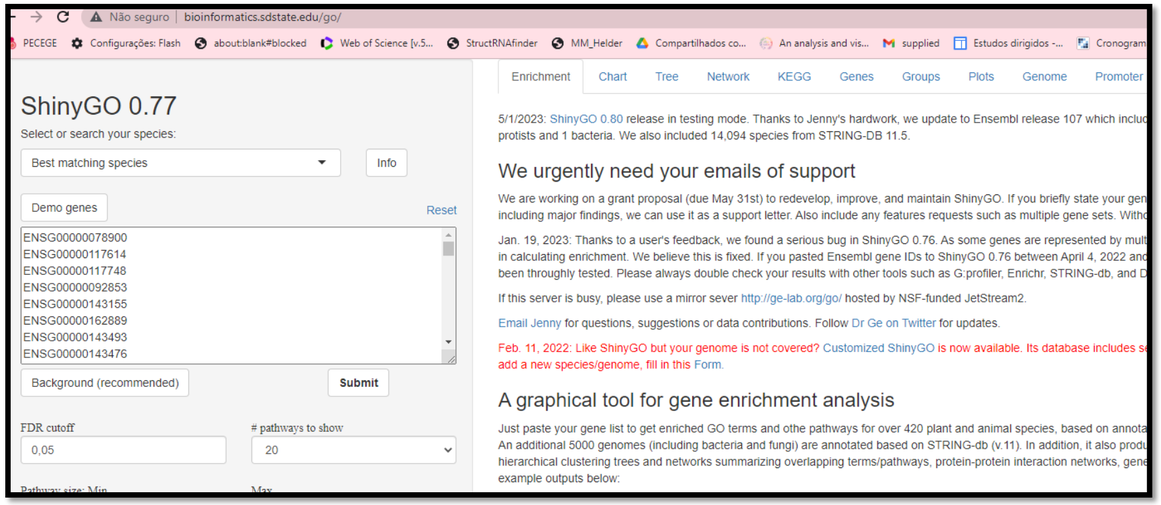


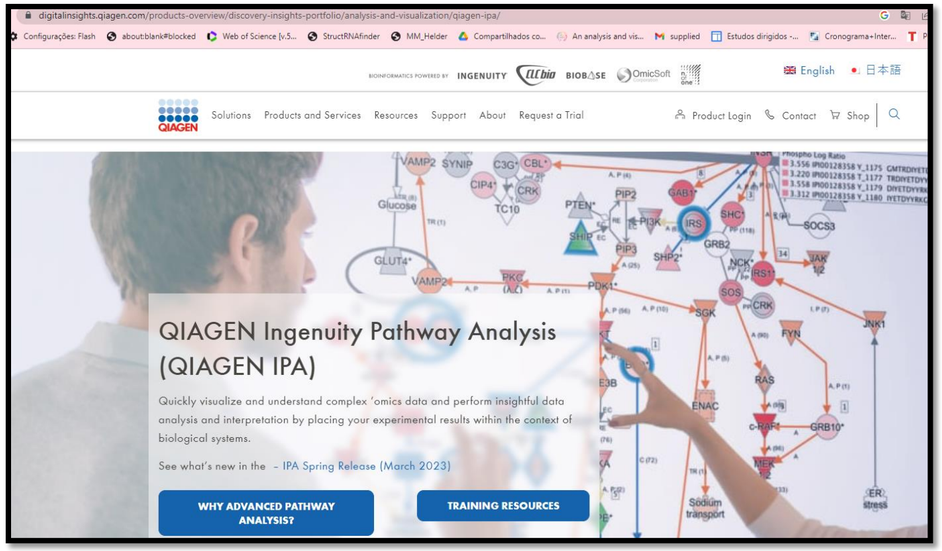
BASES DE DADOS MAIS COMUNS

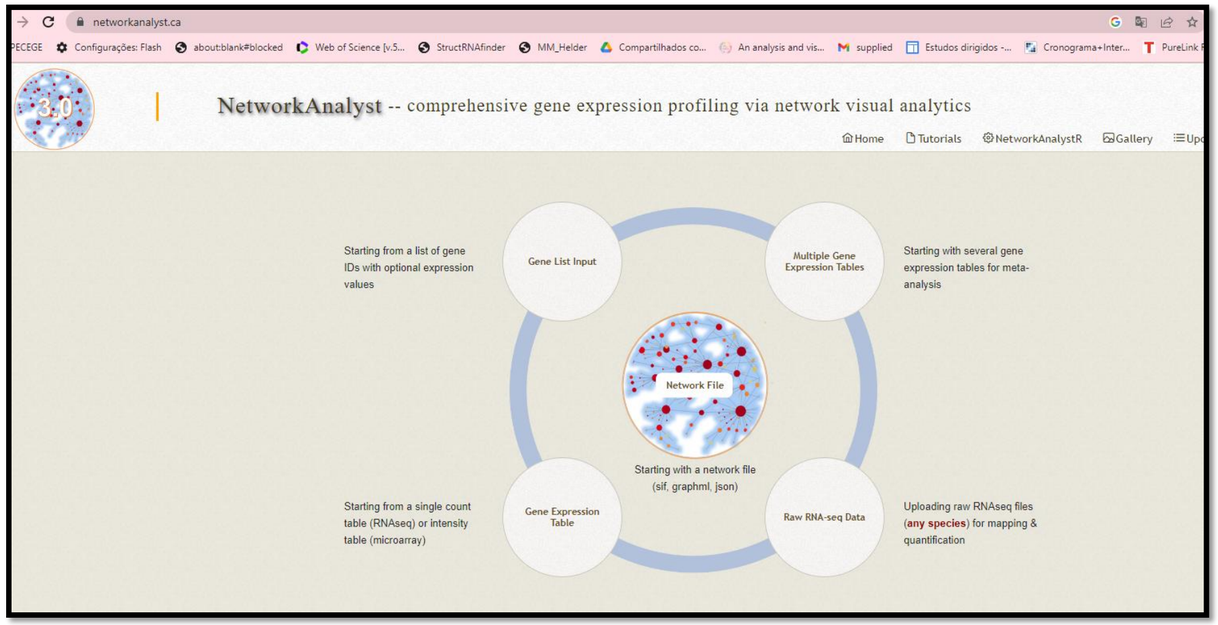
MSigdb

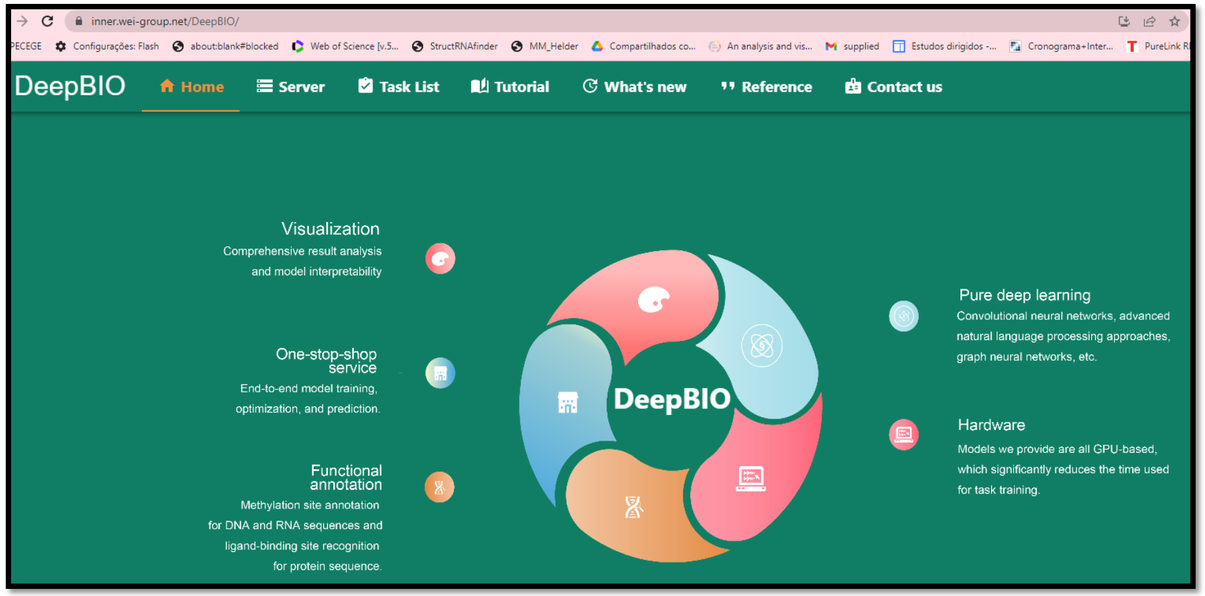


Ferramentas/Softwares









Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home Install Help

Home » Bioconductor 3.17 » Software Packages » clusterProfiler

clusterProfiler

platforms all | rank 41 / 2229 | support 1 2 / 2 1 | in Bioc 12 years
build ok | updated < 1 month | dependencies 128

DOI: 10.18129/B9.bioc.clusterProfiler

A universal enrichment tool for interpreting omics data

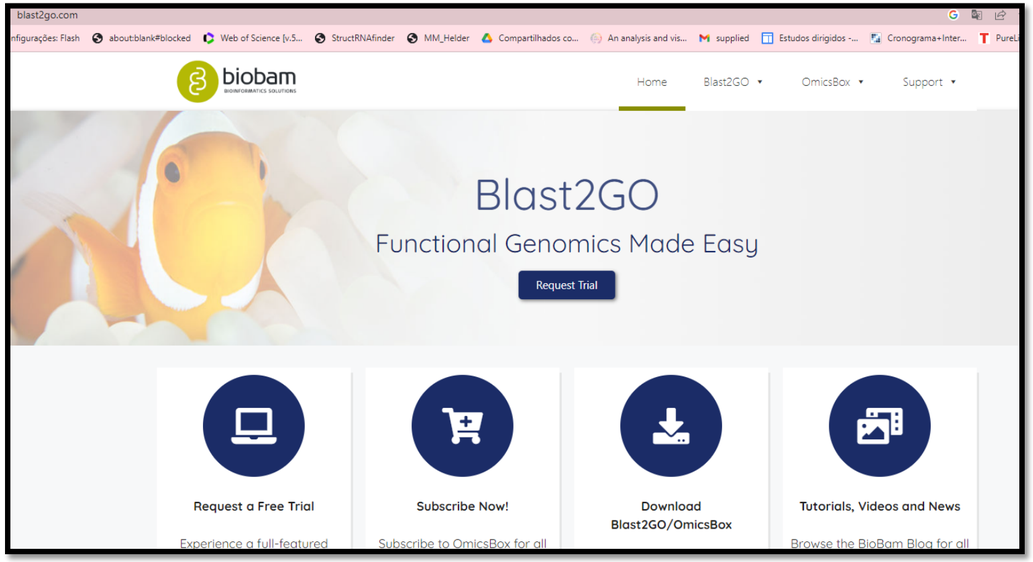
Bioconductor version: Release (3.17)

This package supports functional characteristics of both coding and non-coding genomics data for thousands of species with up-to-date gene annotation. It provides a univeral interface for gene functional annotation from a variety of sources and thus can be applied in diverse scenarios. It provides a tidy interface to access, manipulate, and visualize enrichment results to help users achieve efficient data interpretation. Datasets obtained from multiple treatments and time points can be analyzed and compared in a single run, easily revealing functional consensus and differences among distinct conditions.

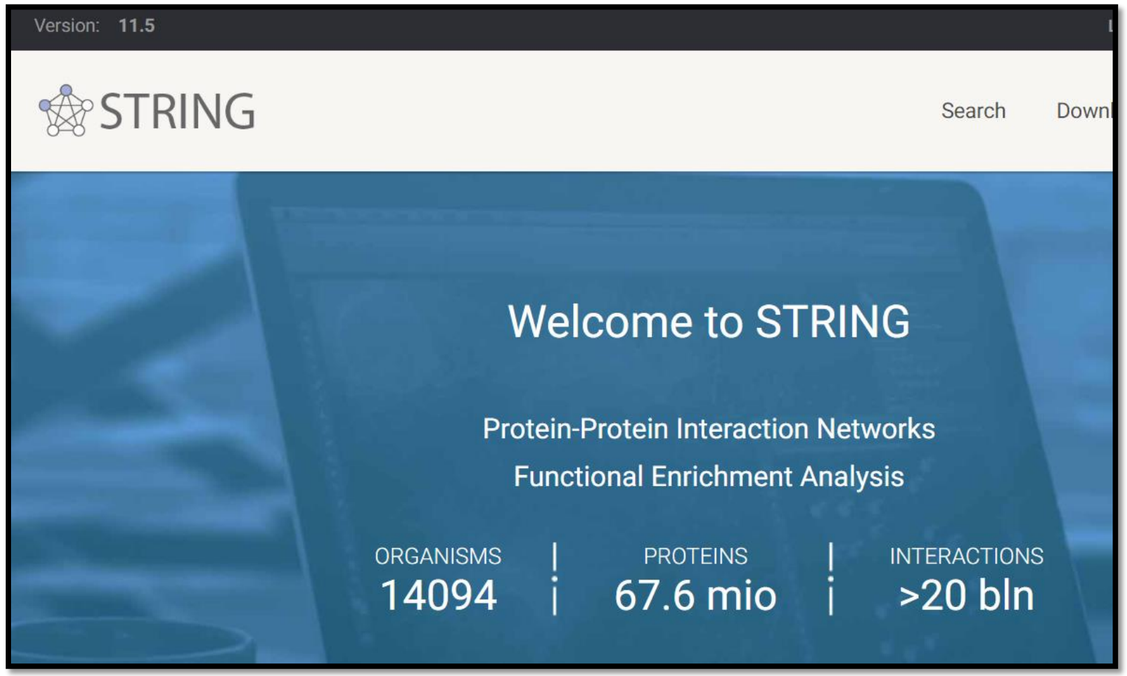
Author: Guangchuang Yu [aut, cre, cph], Li-Gen Wang [ctb], Erqiang Hu [ctb], Xiao Luo [ctb], Meijun Chen [ctb], Giovanni Dall'Olio [ctb], Wanqian Wei [ctb], Chun-Hui Gao [ctb]

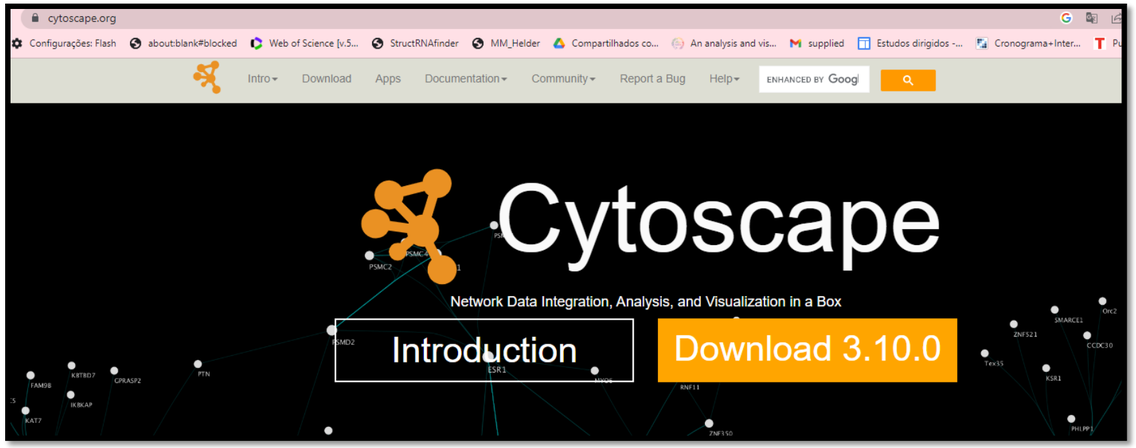
Maintainer: Guangchuang Yu <guangchuangyu at gmail.com>

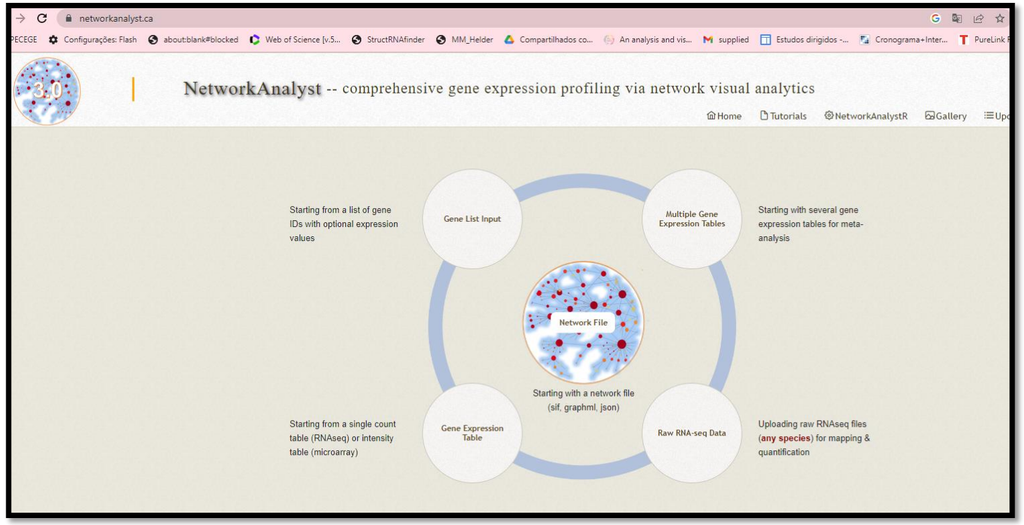
Citation (from within R, enter `citation("clusterProfiler")`):

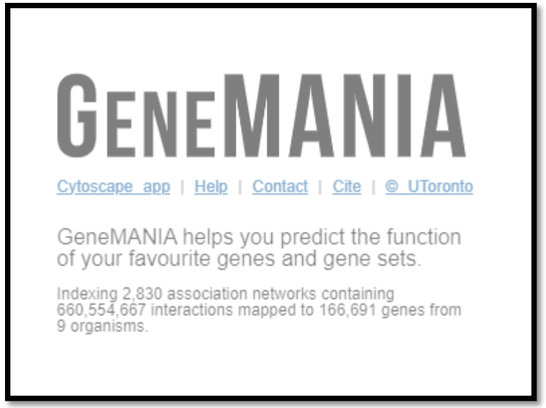


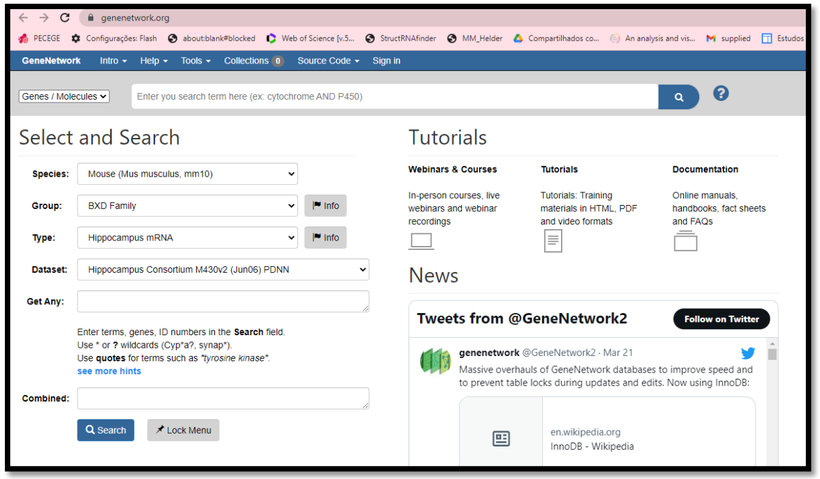
Ferramentas/Softwares para redes gênicas











E quais ferramentas usar ???

E quais ferramentas usar ???

OXFORD

Briefings in Bioinformatics, 2023, **24(1)**, 1–17
https://doi.org/10.1093/bib/bbac529
Review

The hitchhikers' guide to RNA sequencing and functional analysis

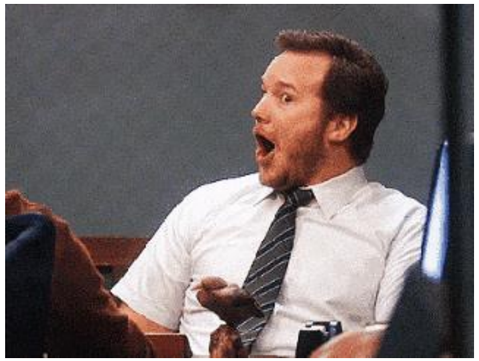
Jiung-Wen Chen, Lisa Shrestha, George Green, André Leier and Tatiana T. Marquez-Lago

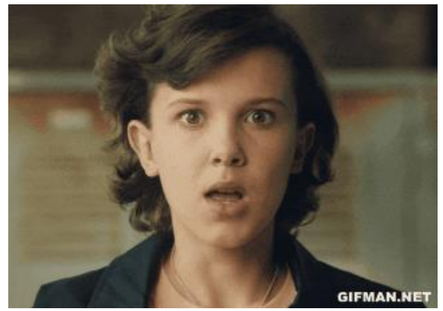
Corresponding author. Tatiana T. Marquez-Lago, Department of Genetics, University of Alabama at Birmingham, School of Medicine, Birmingham, AL, USA. Tel.: +1 (205) 9343194. E-mail: tmarquez@uab.edu

Abstract

DNA and RNA sequencing technologies have revolutionized biology and biomedical sciences, sequencing full genom
tomes at very high speeds and reasonably low costs. RNA sequencing (RNA-Seq) enables transcript identification and q
once sequencing has concluded researchers can be easily overwhelmed with questions such as how to go from raw d
expression (DE), pathway analysis and interpretation. Several pipelines and procedures have been developed to this ef
there is no unique way to perform RNA-Seq analysis, it usually follows these steps: 1) raw reads quality check, 2) al
to a reference genome, 3) aligned reads' summarization according to an annotation file, 4) DE analysis and 5) gene se
functional enrichment analysis. Each step requires researchers to make decisions, and the wide variety of options ar
volumes of data often lead to interpretation challenges. There also seems to be insufficient guidance on how best t
information and derive actionable knowledge from transcription experiments. In this paper, we explain RNA-Seq st
outline differences and similarities of different popular options, as well as advantages and disadvantages. We also di
RNA analysis, multi-omics, meta-transcriptomics and the use of artificial intelligence methods complementing the
available to researchers. Lastly, we perform a complete analysis from raw reads to DE and functional enrichment
illustrating how results are not absolute truths and how algorithmic decisions can greatly impact results and interpr

Keywords: RNA sequencing, differential expression, functional analysis, machine learning, multi-omics





PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Urgent need for consistent standards in functional enrichment analysis

Kaumadi Wijesooriya[1], Sameer A. Jadaan[2], Kaushalya L. Perera[1], Tanuveer Kaur[1], Mark Ziemann[1]*

1 Deakin University, School of Life and Environmental Sciences, Geelong, Australia, **2** College of Health and Medical Technology, Middle Technical University, Baghdad, Iraq

* m.ziemann@deakin.edu.au

Check for updates

OPEN ACCESS

Citation: Wijesooriya K, Jadaan SA, Perera KL, Kaur T, Ziemann M (2022) Urgent need for consistent standards in functional enrichment analysis. PLoS Comput Biol 18(3): e1009935. https://doi.org/10.1371/journal.pcbi.1009935

Editor: Melissa L. Kemp, Georgia Institute of Technology and Emory University, UNITED STATES

Received: December 7, 2021

Accepted: February 18, 2022

Published: March 9, 2022

Abstract

Gene set enrichment tests (a.k.a. functional enrichment analysis) are among the most frequently used methods in computational biology. Despite this popularity, there are concerns that these methods are being applied incorrectly and the results of some peer-reviewed publications are unreliable. These problems include the use of inappropriate background gene lists, lack of false discovery rate correction and lack of methodological detail. To ascertain the frequency of these issues in the literature, we performed a screen of 186 open-access research articles describing functional enrichment results. We find that 95% of analyses using over-representation tests did not implement an appropriate background gene list or did not describe this in the methods. Failure to perform p-value correction for multiple tests was identified in 43% of analyses. Many studies lacked detail in the methods section about the tools and gene sets used. An extension of this survey showed that these problems are not associated with journal or article level bibliometrics. Using seven independent RNA-seq datasets, we show misuse of enrichment tools alters results substantially. In conclusion, most published functional enrichment studies suffered from one or more major flaws, highlighting the need for stronger standards for enrichment analysis.

Obrigada!

adriana.Ibelli@embrapa.br

@dricaibelli

Embrapa 50 ANOS

MINISTÉRIO DA AGRICULTURA E PECUÁRIA

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO