Vall d'Hebron
Institut de Recerca

INTRODUCTION TO OMICS DATA ANALYSIS
RNA-seq

Bioinformàtica per a la Recerca Biomèdica

Ricardo Gonzalo Sanz
ricardo.gonzalo@vhir.org

Vall d'Hebron Hospital
Generalitat de Catalunya Departament de Salut
Institut Català de la Salut
UAB Universitat Autònoma de Barcelona
Generalitat de Catalunya Departament d'Economia i Coneixement
Institució CERCA Centres de Recerca de Catalunya
Instituto de Salud Carlos III

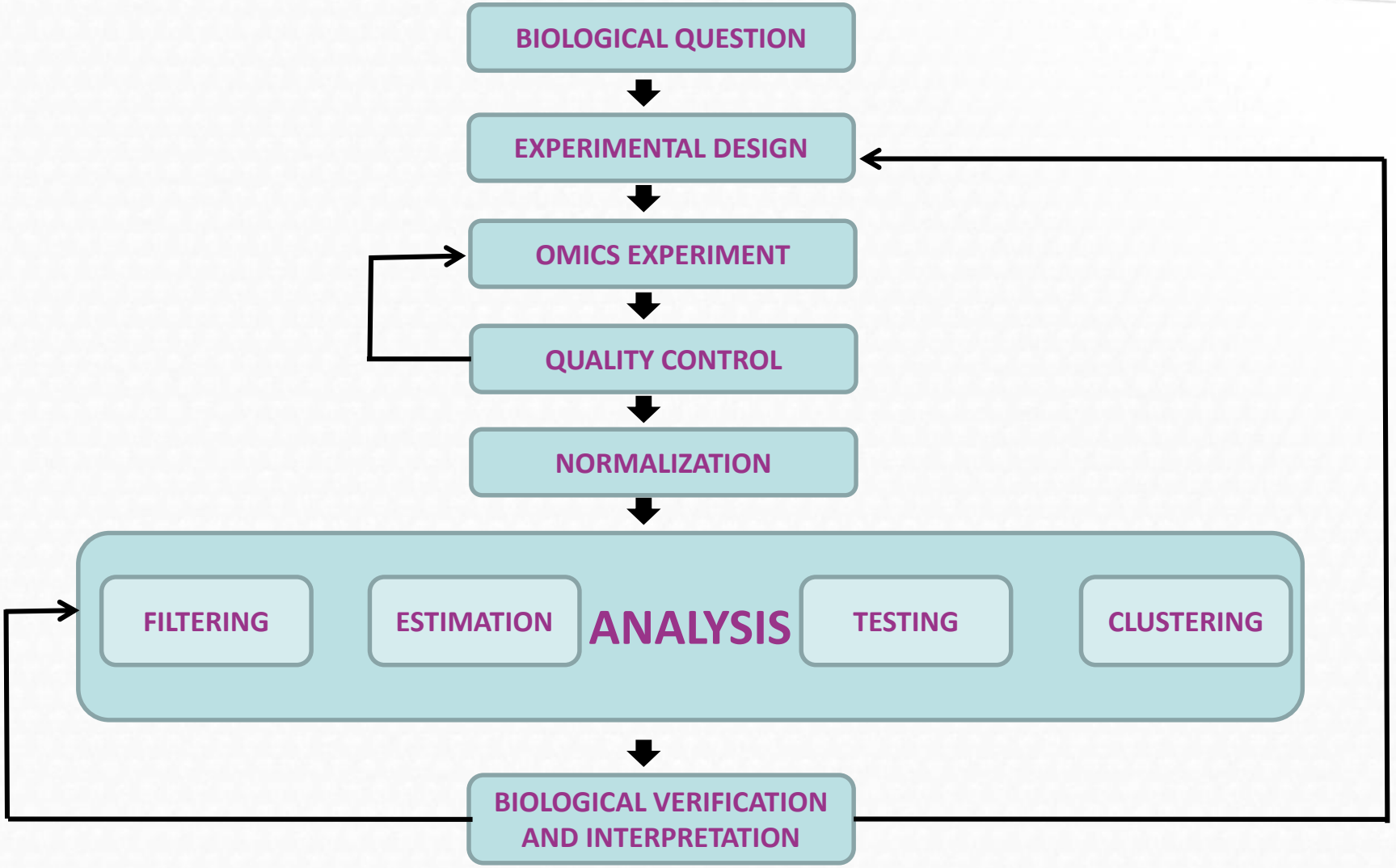
Vall d'Hebron
Institut de Recerca

1. Introduction to omics data analysis
2. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)
    6. Alignment
    7. RNA-seq pipeline with R

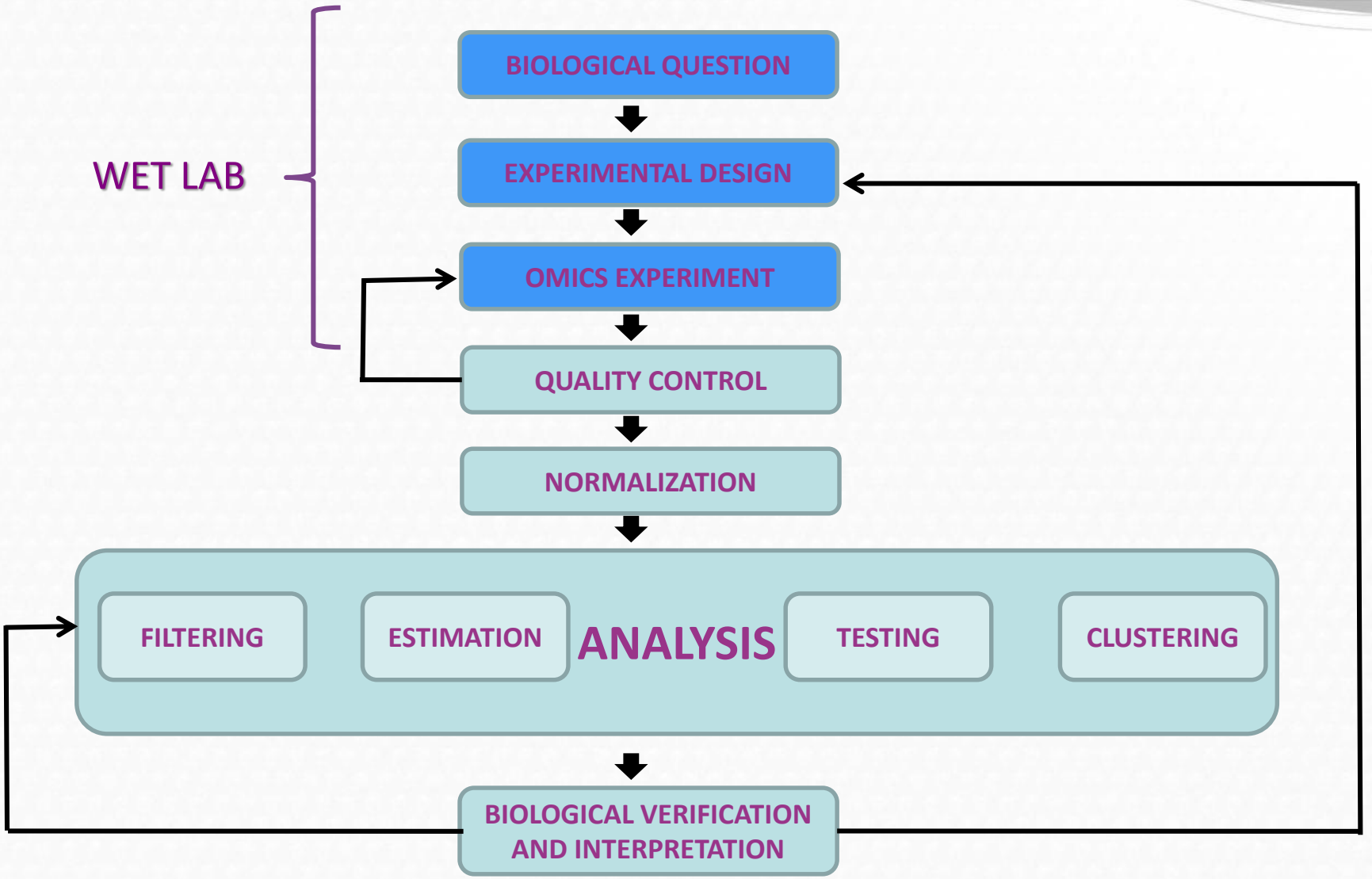
Vall d'Hebron
Institut de Recerca

1. **Introduction to omics data analysis**
2. **An example of omics data analysis. RNA-seq**
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)
    6. Alignment
    7. RNA-seq pipeline with R

1. Introduction to omic data analysis



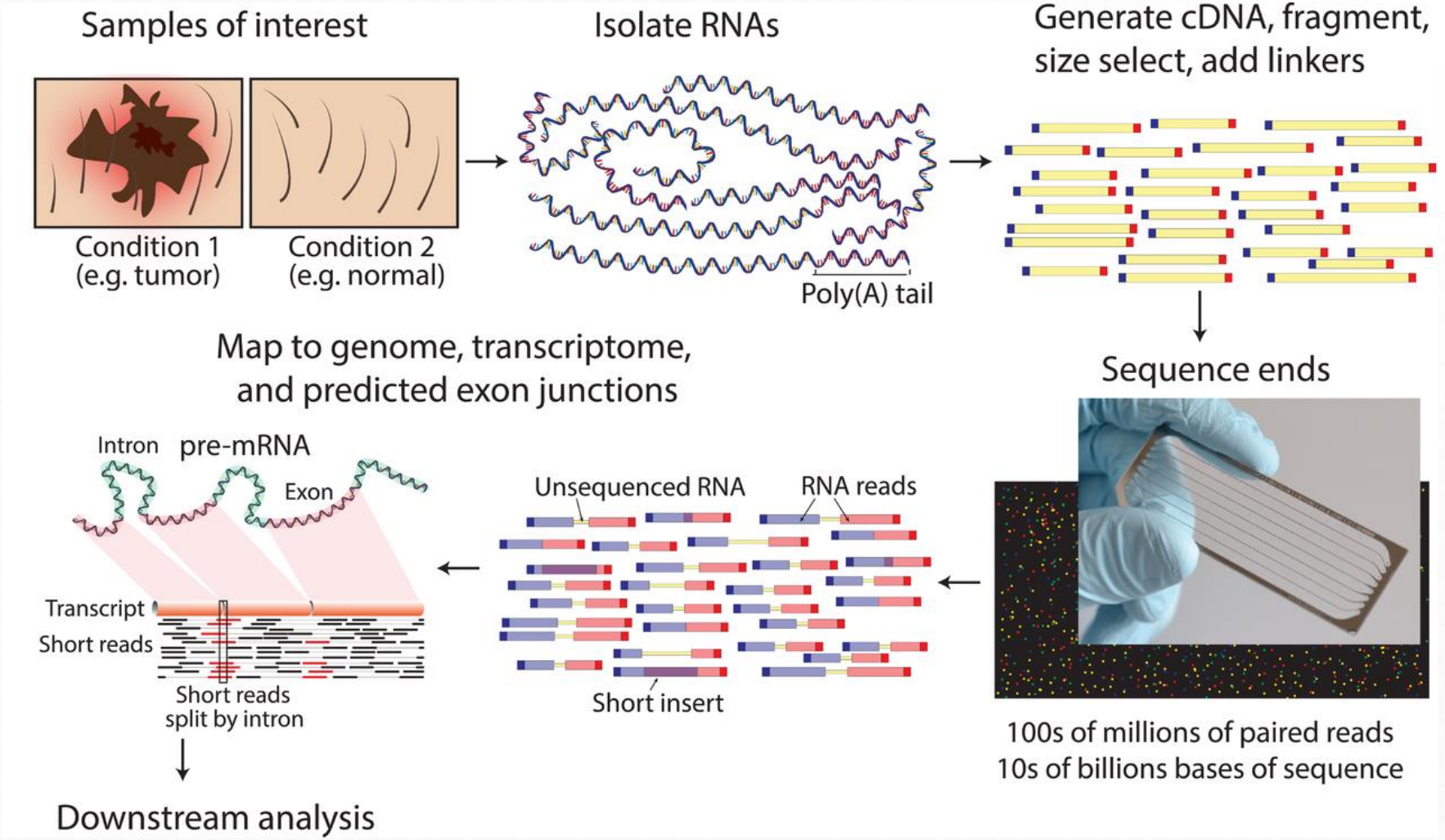
1. Introduction to omic data analysis



Vall d'Hebron
Institut de Recerca

1. **Introduction to omics data analysis**
2. **An example of omics data analysis. RNA-seq**
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)
    6. Alignment
    7. RNA-seq pipeline with R

2.1 What is RNA-seq?



https://en.wikipedia.org/wiki/RNA-Seq

2.1 What is RNA-seq?

- RNA-seq is the high throughput sequencing of cDNA using NGS technologies
- RNA-seq works by sequencing every RNA molecule and profiling the expression of a particular gene by counting the number of time its transcripts have been sequenced.
- The summarized RNA-seq data is widely known as ***count table***

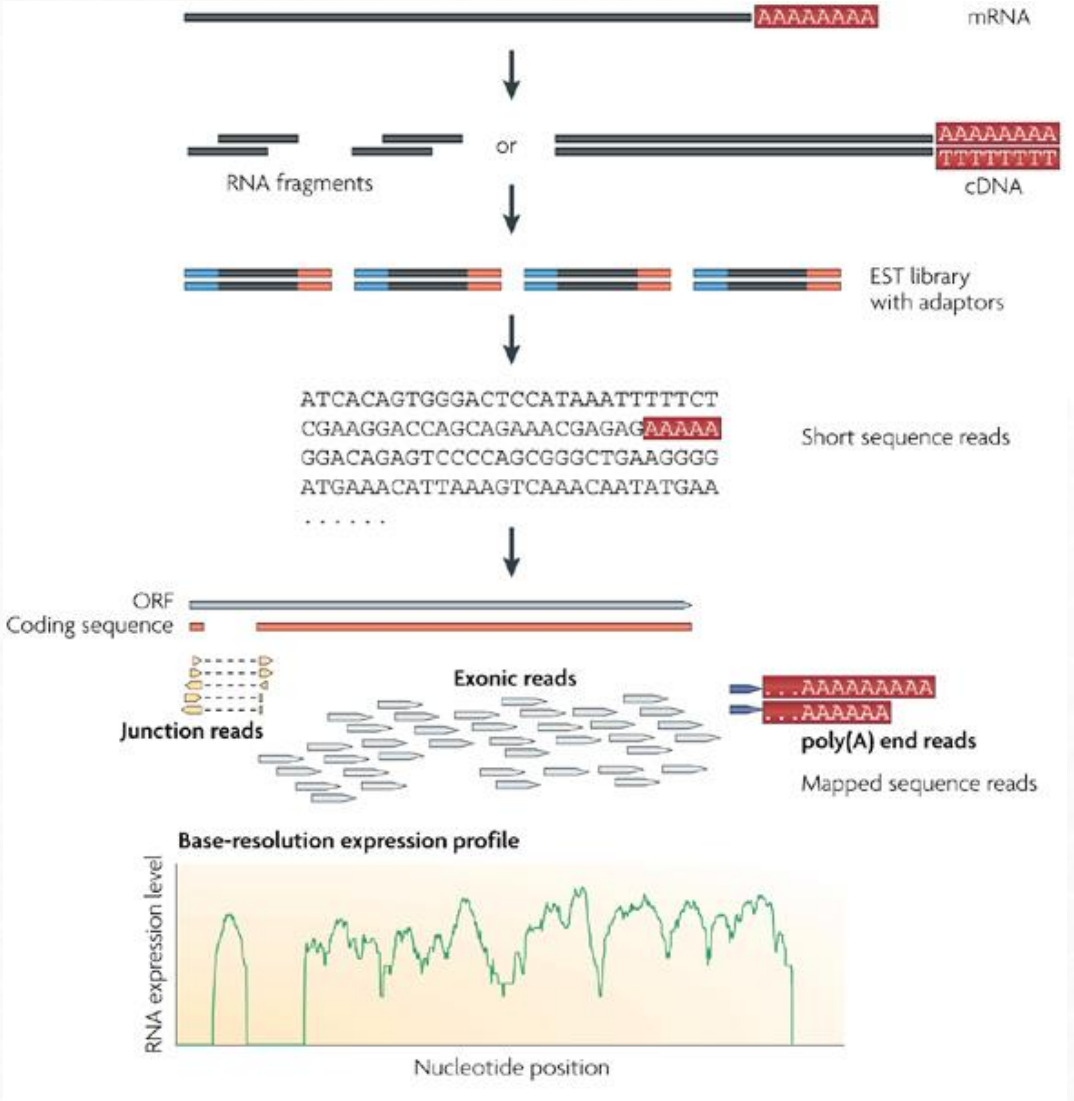
	Condition A			Condition B		
Gene1	4	0	2	12	14	13
Gene2	0	23	50	47	22	0
Gene3	0	2	6	13	11	15
...	...	...	...	...	...	...
GeneG	156	238	37	129	51	118

2.1 What is RNA-seq?

Briefly, long RNAs are first converted into a library of cDNA fragments through either RNA fragmentation or DNA fragmentation (see main text).

Sequencing adaptors (blue) are subsequently added to each cDNA fragment and a short sequence is obtained from each cDNA using high-throughput sequencing technology.

The resulting sequence reads are aligned with the reference genome or transcriptome, and classified as three types: exonic reads, junction reads and poly(A) end-reads. These three types are used to generate a base-resolution expression profile for each gene, as illustrated at the bottom; a yeast ORF with one intron is shown.



Nature Reviews Genetics **10**, 57-63 (January 2009)

2.1 What is RNA-seq?

- Functional studies
  - Genome may be constant but an experimental condition has a pronounced effect on gene expression
    - ✓ e.g. Drug treated vs. untreated cell line
    - ✓ e.g. Wild type versus knock out mice

- Some molecular features can only be observed at the RNA level
  - Alternative isoforms, fusion transcripts, RNA editing

- Predicting transcript sequence from genome sequence is difficult
  - Alternative splicing, RNA editing, etc.

Vall d'Hebron
Institut de Recerca

1. Introduction to omics data analysis
2. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)
    6. Alignment
    7. RNA-seq pipeline with R

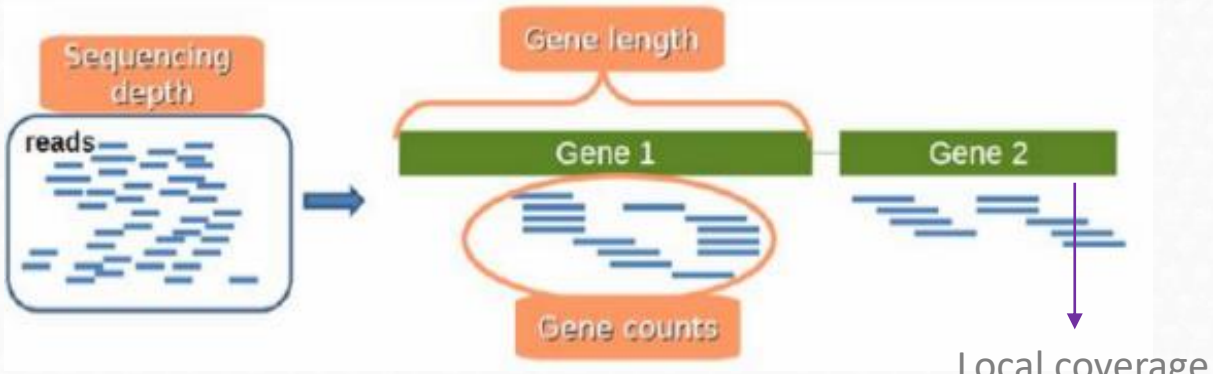
2.2 Basic key concepts?

Sequencing depth: Total number of reads mapped to the genome. (Library size) Could also be applied to samples.

Coverage: Number of reads mapped to a specific region (average of them if we are talking about the whole genome...)

Gene length: Number of bases that a gene has.

Gene counts: Number of reads mapping to that gene (expression measurement)



Vall d'Hebron
Institut de Recerca

1. Introduction to omics data analysis
2. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)
    6. Alignment
    7. RNA-seq pipeline with R

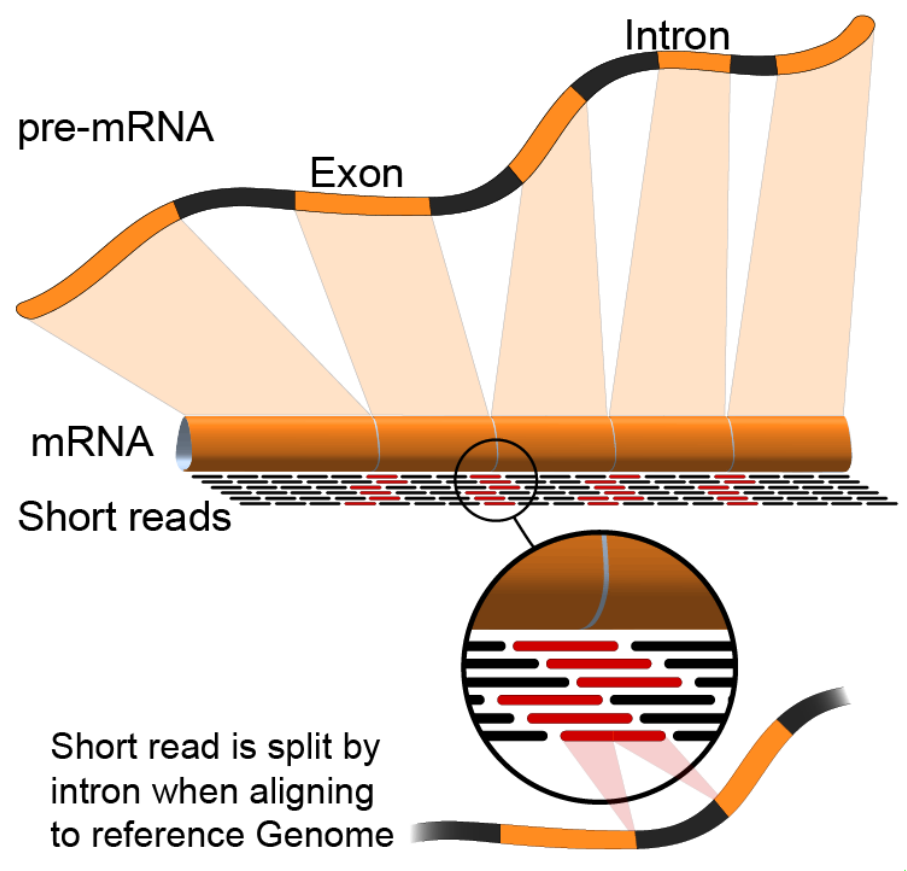
3.3 Main challenges in RNA-seq

- **Sample**
  - Purity? Quantity? Quality?

- **RNAs consist of small exons that may be separated by large introns**
  - Mapping reads to genome is challenging
  - Non-uniformity coverage of the genome

3.3 Main challenges in RNA-seq

- **Sample**
  - Purity? Qua
- **RNAs consist of** arge introns
  - Mapping rea
  - Non-uniform



3.3 Main challenges in RNA-seq

- **Sample**
  - Purity? Quantity? Quality?

- **RNAs consist of small exons that may be separated by large introns**
  - Mapping reads to genome is challenging
  - Non-uniformity coverage of the genome

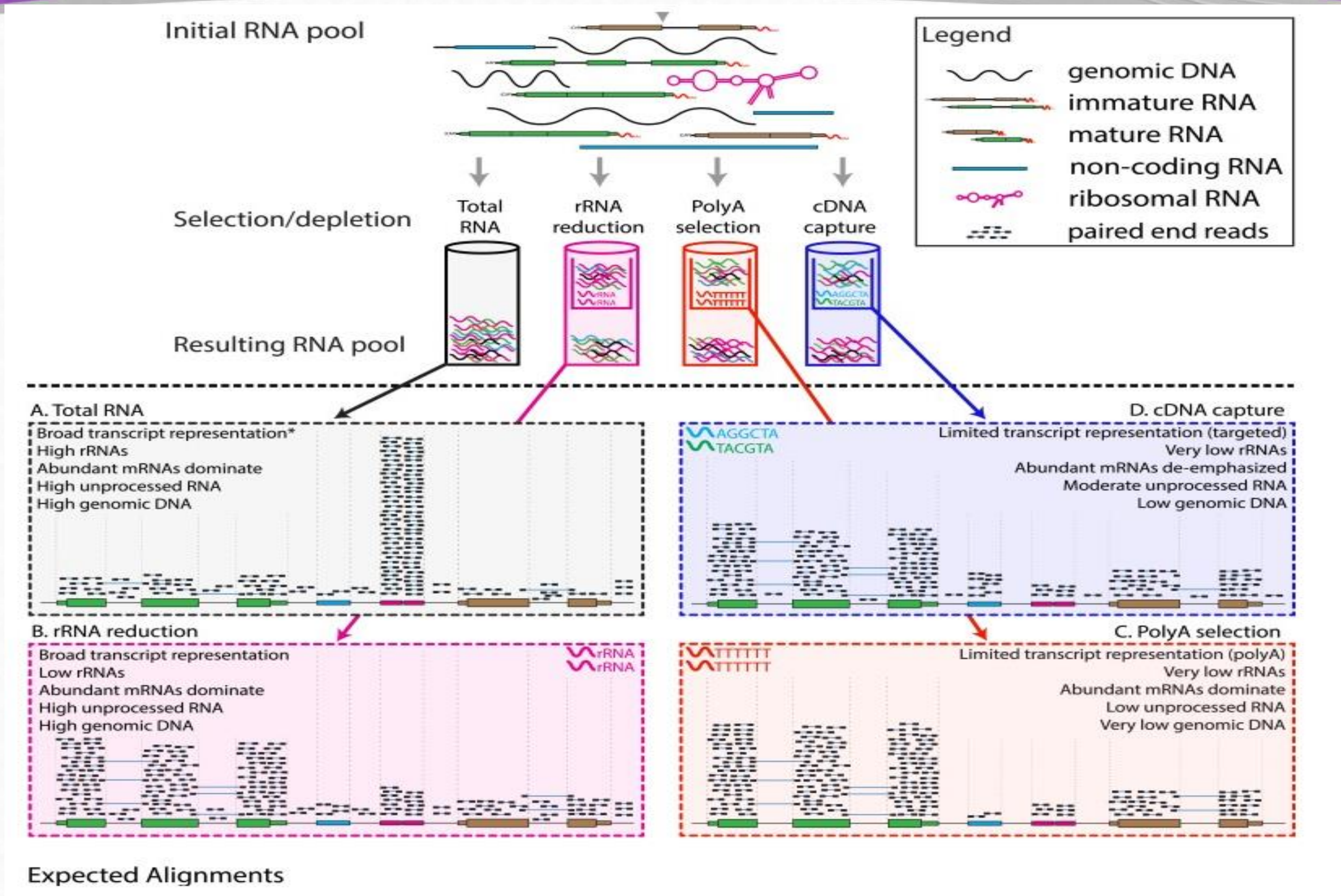
- **The relative abundance of RNAs vary wildly**
  - 10^5 – 10^7 orders of magnitude
  - Since RNA sequencing works by random sampling, a small fraction of highly expressed genes may consume the majority of reads (rRNA)

- **RNAs come in a wide range of sizes**
  - Small RNAs must be captured separately
  - PolyA selection of large RNAs may result in 3' end bias

- **RNA is fragile compared to DNA** (easily degraded)

- **Library preparation**

3.3 Main challenges in RNA-seq



3.3 Main challenges in RNA-seq

- Independently of the software used, one needs to think about

DATA STORAGE & MANAGEMENT!!

1 Illumina Flow Cell equals up to
1.5 Bn individual Clusters
= 3 Bn Reads
= 300 Gbases raw sequence
= 2.5 TByte of disk space (raw data)
> 100 GByte of disk space (fastq data)

T. Stempfl, 2012

Vall d'Hebron
Institut de Recerca

1. Introduction to omics data analysis
2. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)
    6. Alignment
    7. RNA-seq pipeline with R

2.4 RNA-seq vs Microarrays

Published online 15 October 2008 | *Nature* **455**, 847 (2008) |
doi:10.1038/455847a

News

The death of microarrays?

High-throughput gene sequencing seems to be stealing a march on microarrays. Heidi Ledford looks at a genome technology facing intense competition.

Announcing the death of the Micro-array

30 August 2010 by Anthony Fejes, posted in Uncategorized

| **Anthony Fejes**
| About the blog
| Blog homepage

- reproducibility
- only show you what you're looking for
- what about 'indels', inversions, translocations...
- accuracy
- sensitivity

2.4 RNA-seq vs Microarrays

PLoS One. 2013 Aug 20;8(8):e71462. doi: 10.1371/journal.pone.0071462. eCollection 2013.

Large scale comparison of gene expression levels by microarrays and RNAseq using TCGA data.

Guo Y[1], Sheng Q, Li J, Ye F, Samuels DC, Shyr Y.

Abstract

RNAseq and microarray methods are frequently used to measure gene expression level. While similar in purpose, there are fundamental differences between the two technologies. Here, we present the largest comparative study between microarray and RNAseq methods to date using The Cancer Genome Atlas (TCGA) data. **We found high correlations between expression data obtained from the Affymetrix one-channel microarray and RNAseq (Spearman correlations coefficients of ~0.8).** We also observed that the low abundance genes had poorer correlations between microarray and RNAseq data than high abundance genes. As expected, due to measurement and normalization differences, Agilent two-channel microarray and RNAseq data were poorly correlated (Spearman correlations coefficients of only ~0.2). By examining the differentially expressed genes between tumor and normal samples we observed reasonable concordance in directionality between Agilent two-channel microarray and RNAseq data, although a small group of genes were found to have expression changesreported in opposite directions using these two technologies. Overall, RNAseq produces comparable results to microarray technologies in term of expression profiling. The RNAseq normalization methods RPKM and RSEM produce similar results on the gene level and reasonably concordant results on the exon level. Longer exons tended to have better concordance between the two normalization methods than shorter exons.

2.4 RNA-seq vs Microarrays

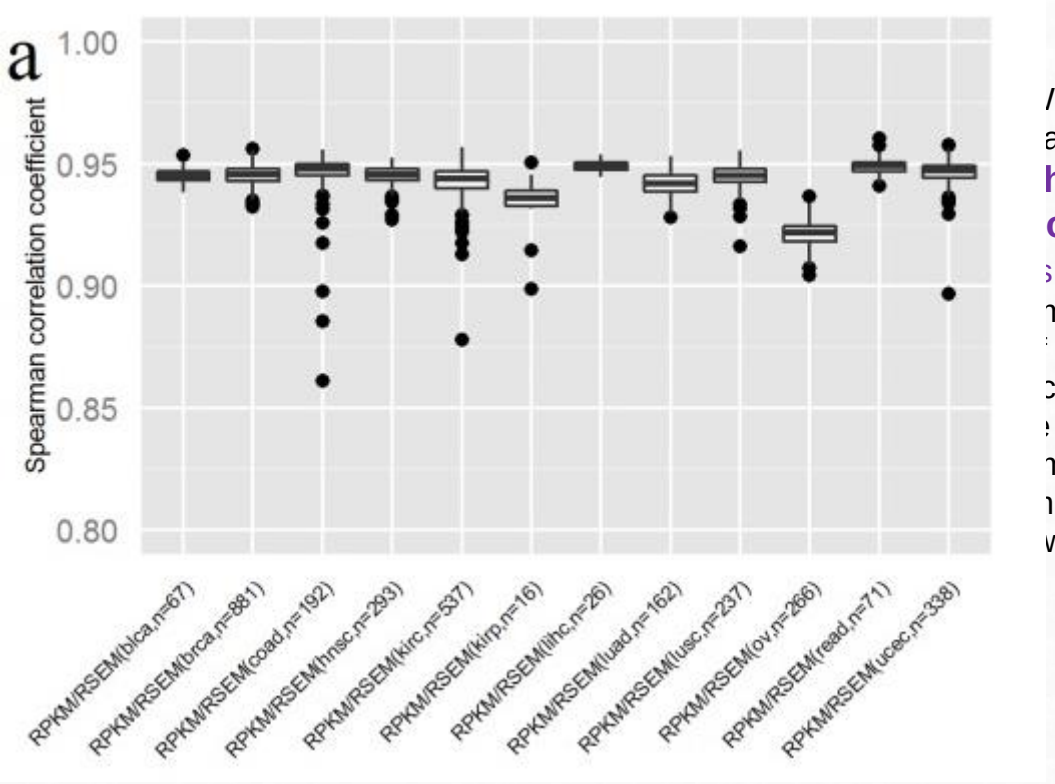
PLoS One. 2013 Aug 20;8(8):e71462. doi: 10.1371/journal.pone.0071462. eCollection 2013.

Large scale comparison of gene expression levels by microarrays and RNAseq using TCGA data.

Guo Y[1], Sheng Q, Li J, Ye F, Samuels DC, Shyr Y.

Abstract

RNAseq and mic
fundamental diffe
and RNAseq me
between expr
correlations c
and RNAseq dat
microarray and F
expressed genes
channel microarr
directions using
of expression pr
concordant resul
shorter exons.



/hile similar in purpose, there are
arative study between microarray
high correlations
oarray and RNAseq (Spearman
s had poorer correlations between microarray
malization differences, Agilent two-channel
only ~0.2). By examining the differentially
ce in directionality between Agilent two-
expression changesreported in opposite
microarray technologies in term
ilar results on the gene level and reasonably
ween the two normalization methods than

2.4 RNA-seq vs Microarrays

Vall d'Hebron
Institut de Recerca

Comparison of RNA-Seq and Microarray in Transcriptome Profiling of Activated T Cells

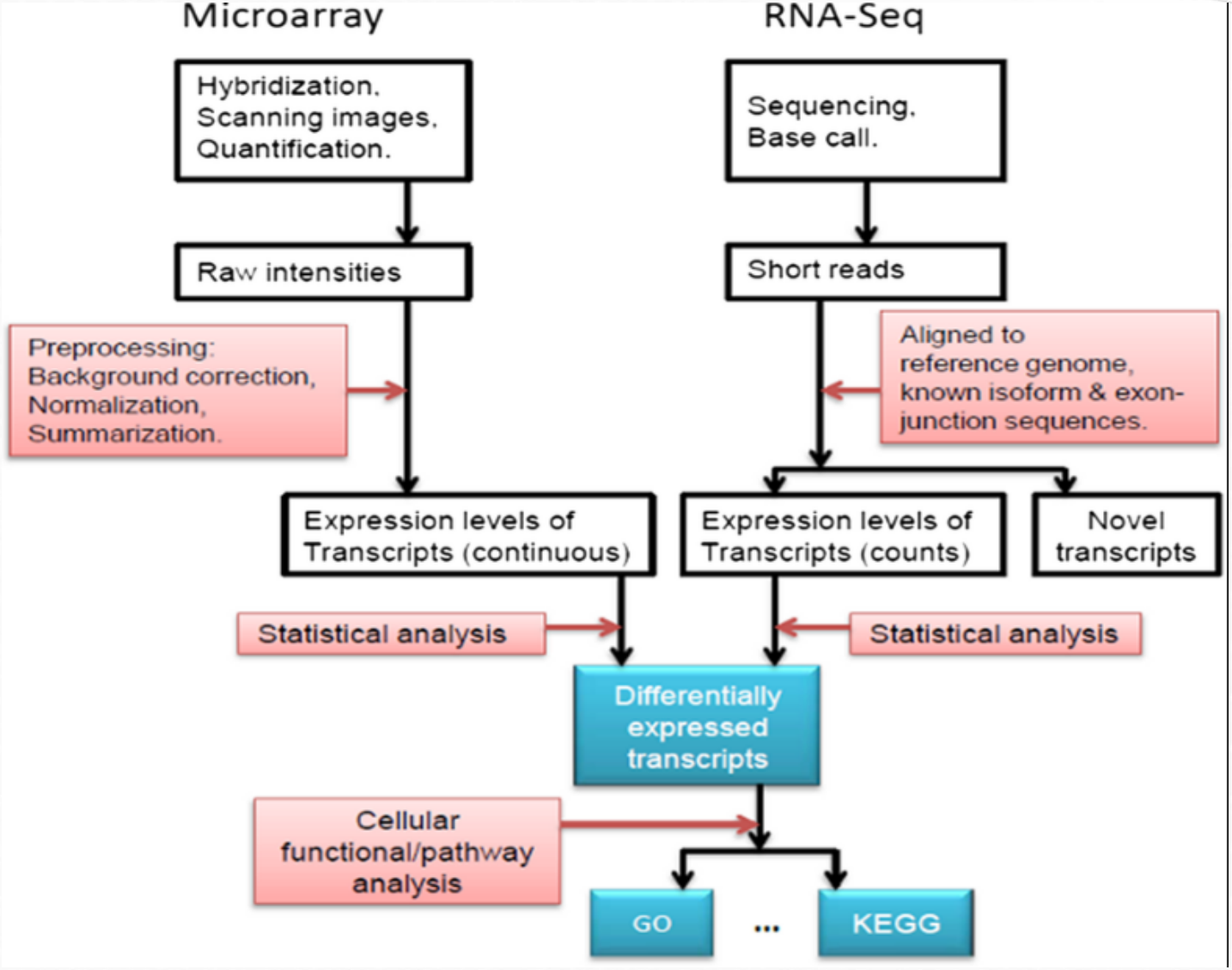
Shanrong Zhao, Wai-Ping Fung-Leung, Anton Bittner, Karen Ngo, Xuejun Liu

Published: January 16, 2014 • DOI: 10.1371/journal.pone.0078644

- RNA-Seq was superior in detecting low abundance transcripts
- also better detecting differentiating biologically isoforms
- RNA-Seq demonstrated a broader dynamic range than microarray.
- RNA-Seq avoid problems inherent to microarray probe performance

!!!The study try to demonstrate the benefits of RNA-Seq over microarray in transcriptome profiling

2.4 RNA-seq vs Microarrays



2.4 RNA-seq vs Microarrays

Vall d'Hebron
Institut de Recerca

Pros and cons of both technologies

Microarrays	**RNA-seq**
Costs,	High reproducibility,
well established methods, small data	not limited to expression
Hybridization bias,	Costs,
sequence must be known	complexity of analysis

"**High correlation** between gene expression profiles generated by the two platforms."

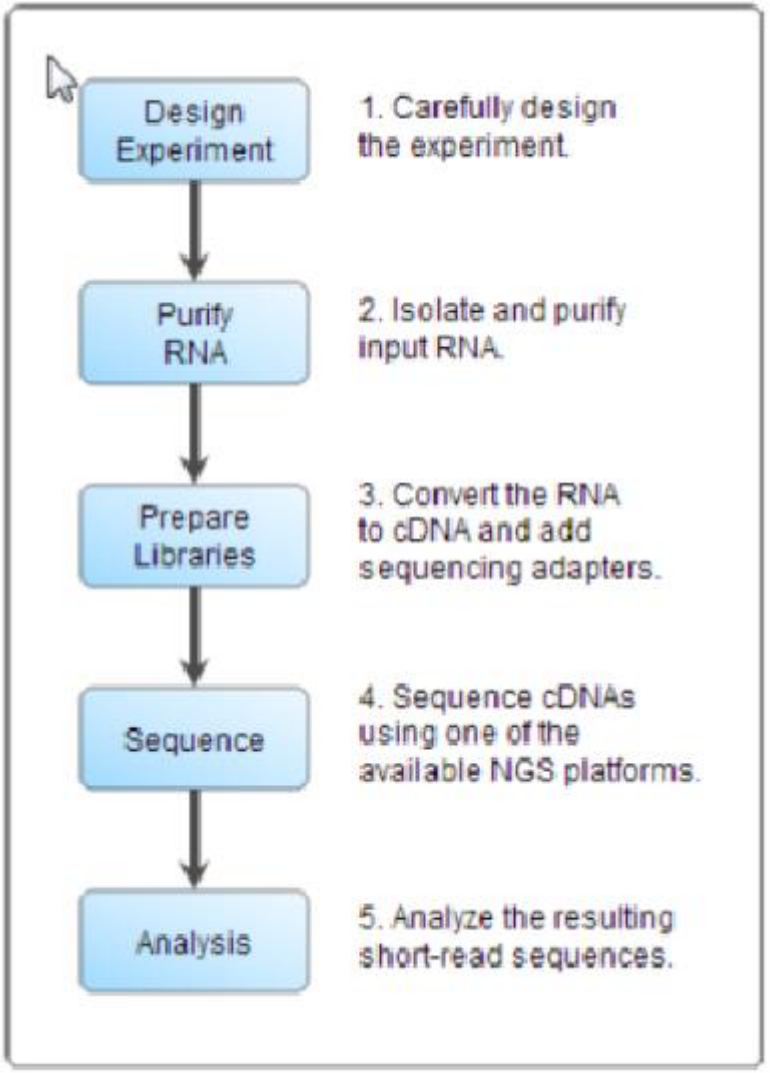
"RNA-Seq sequencing technology is new to most researchers, more expensive than microarray, **data storage is more challenging** and **analysis is more complex**."

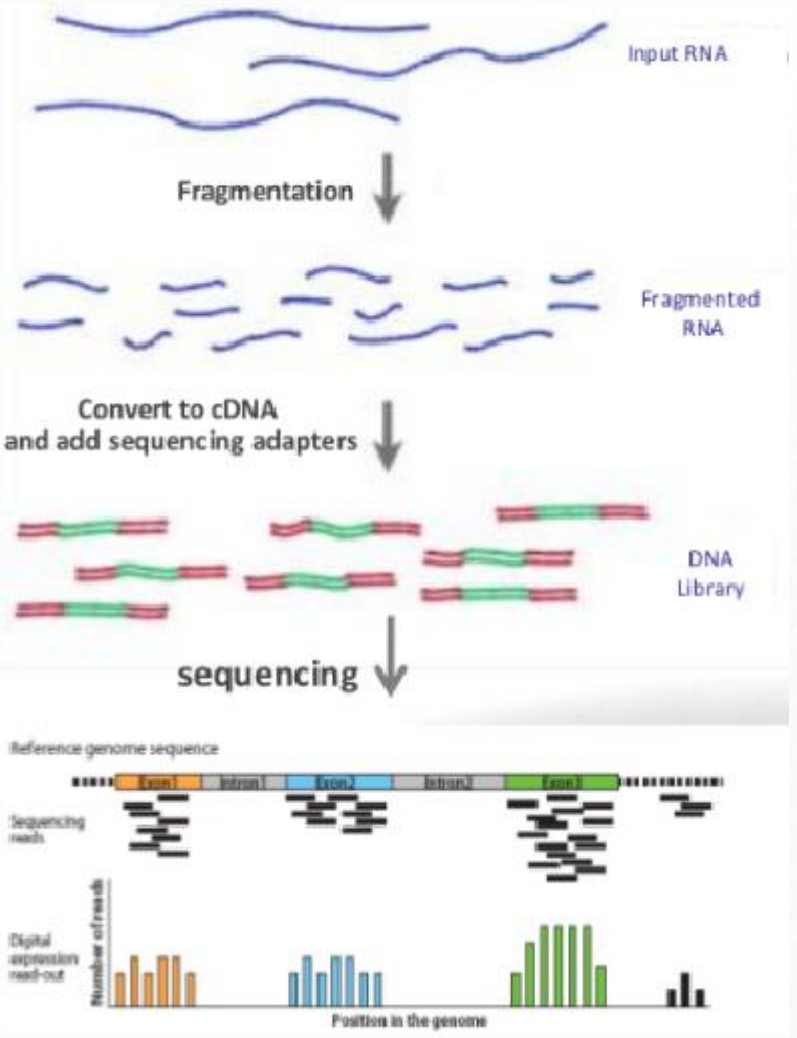
Zhao et al, 2014

Vall d'Hebron
Institut de Recerca

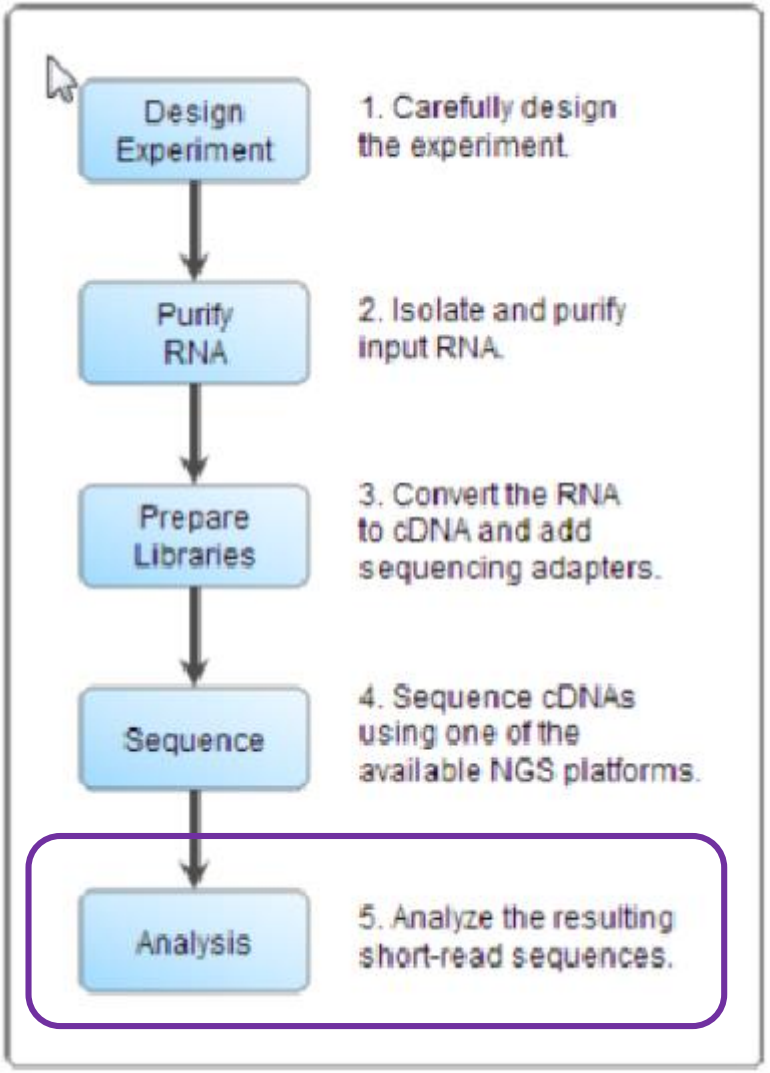
1. Introduction to omics data analysis
2. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)
    6. Alignment
    7. RNA-seq pipeline with R

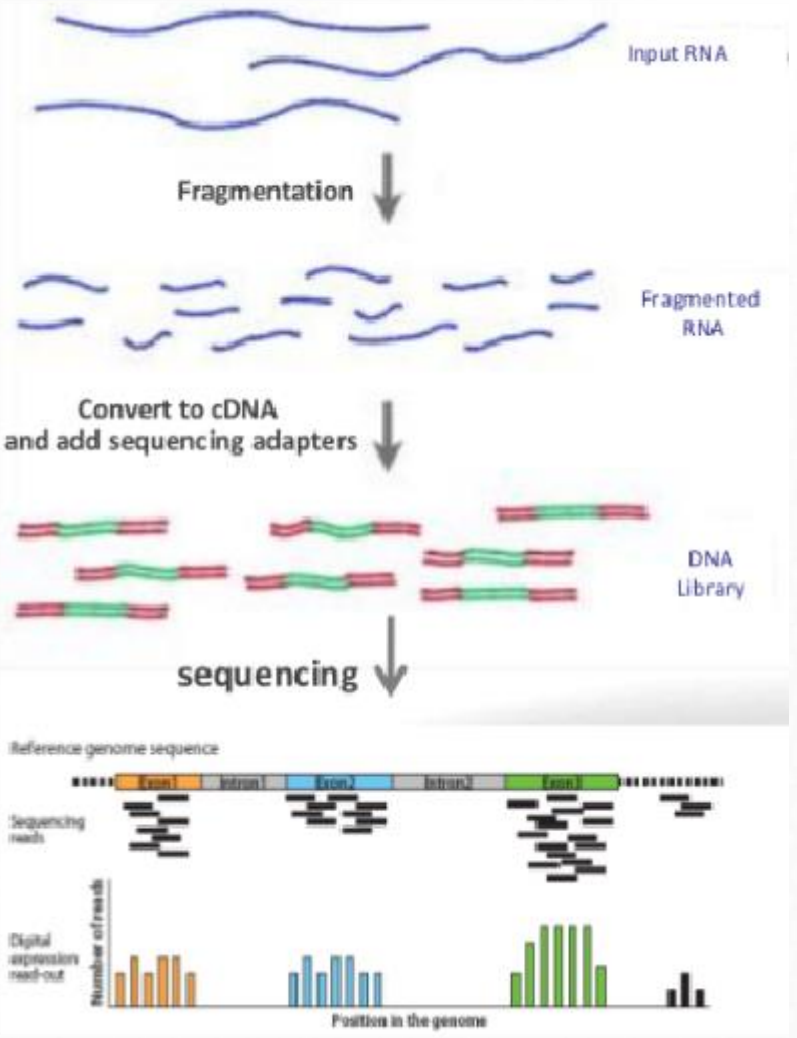
2.5 RNA-seq analysis pipeline(s)



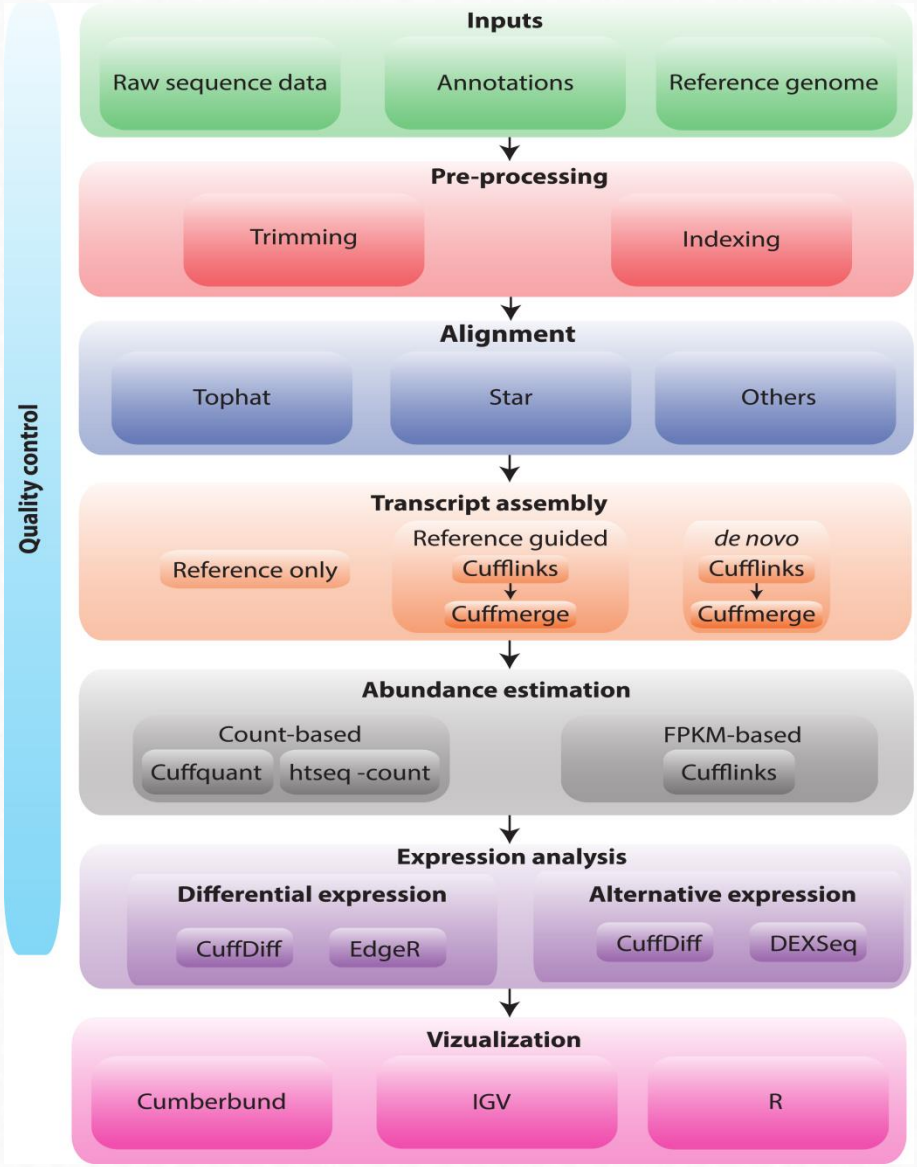


2.5 RNA-seq analysis pipeline(s)



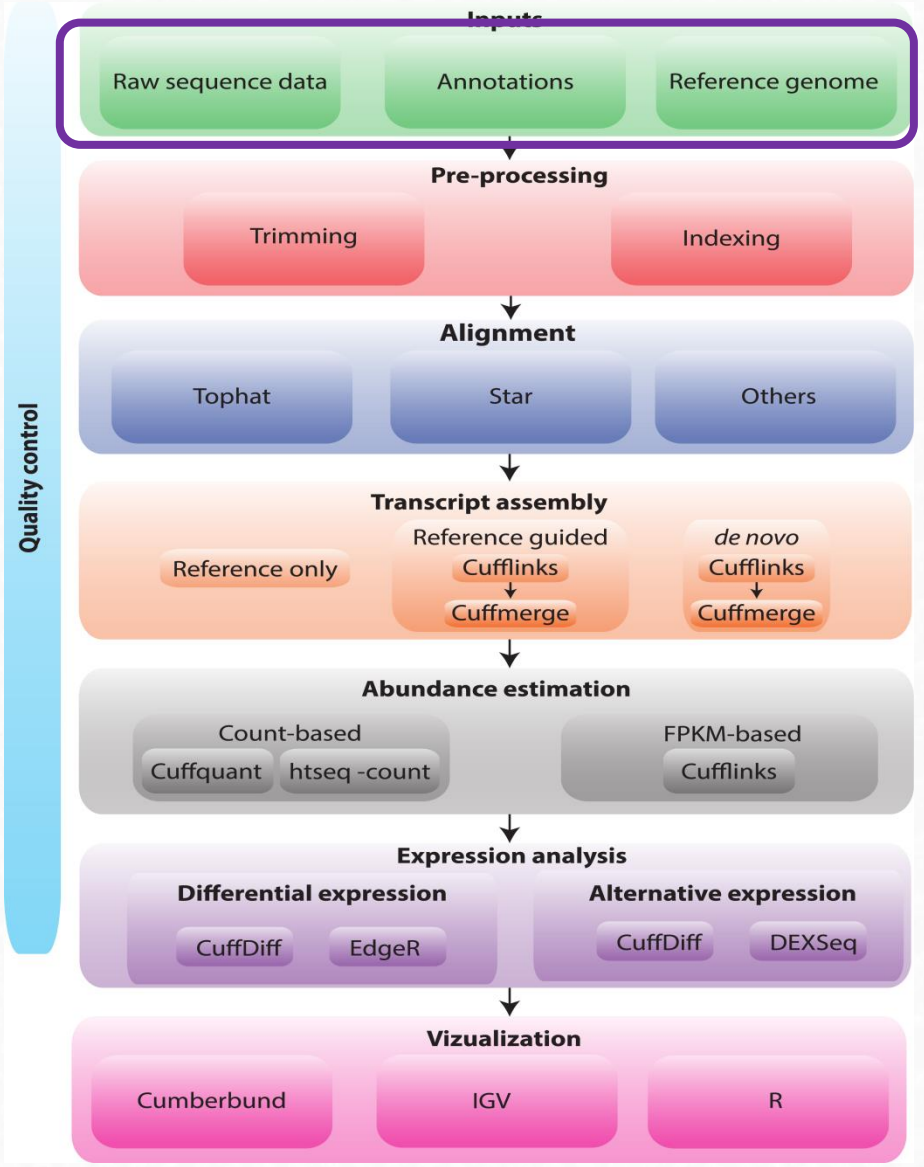


2.5 RNA-seq analysis pipeline(s)



Griffith, 2015 DOI: 10.1371/journal.pcbi.1004393

2.5 RNA-seq analysis pipeline(s)

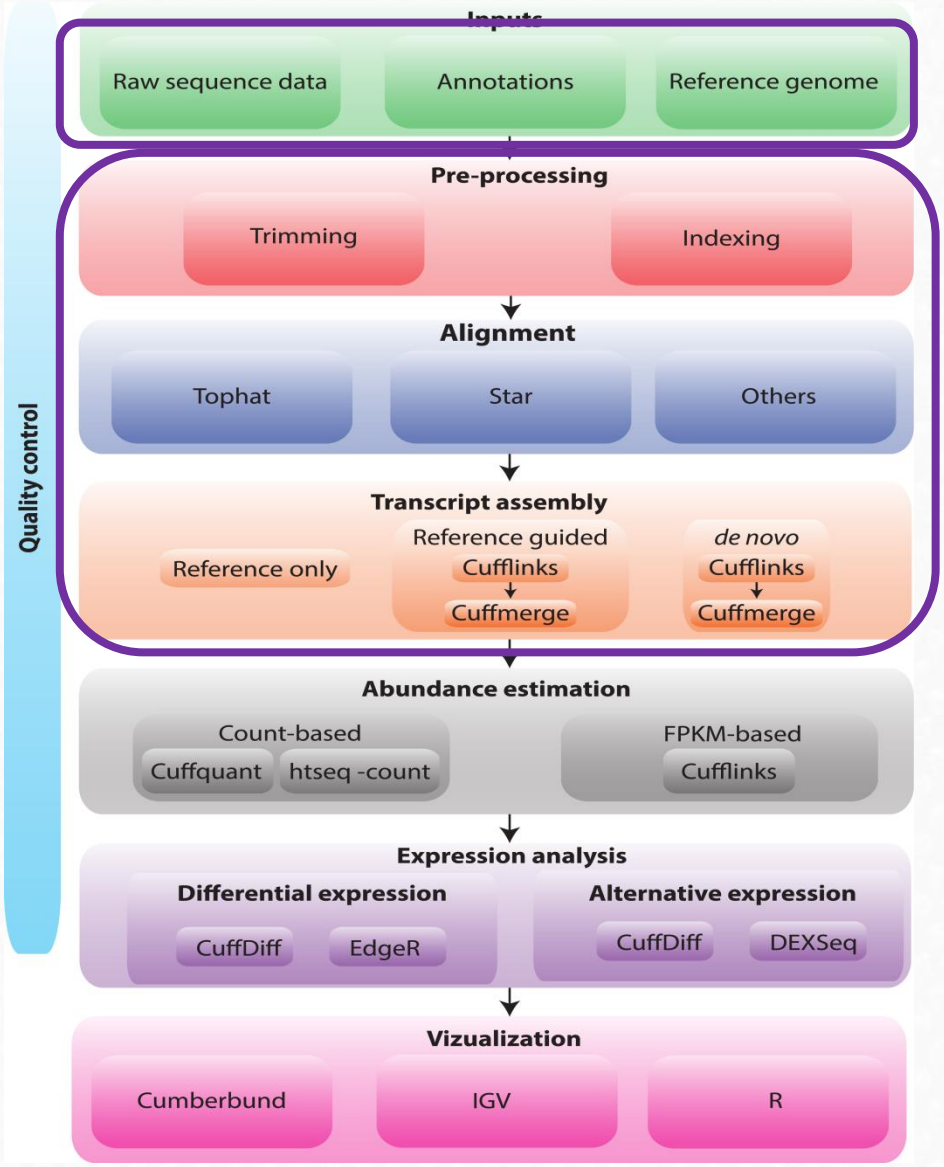


Fastq files

FASTQC (quality control)

Griffith, 2015 DOI: 10.1371/journal.pcbi.1004393

2.5 RNA-seq analysis pipeline(s)

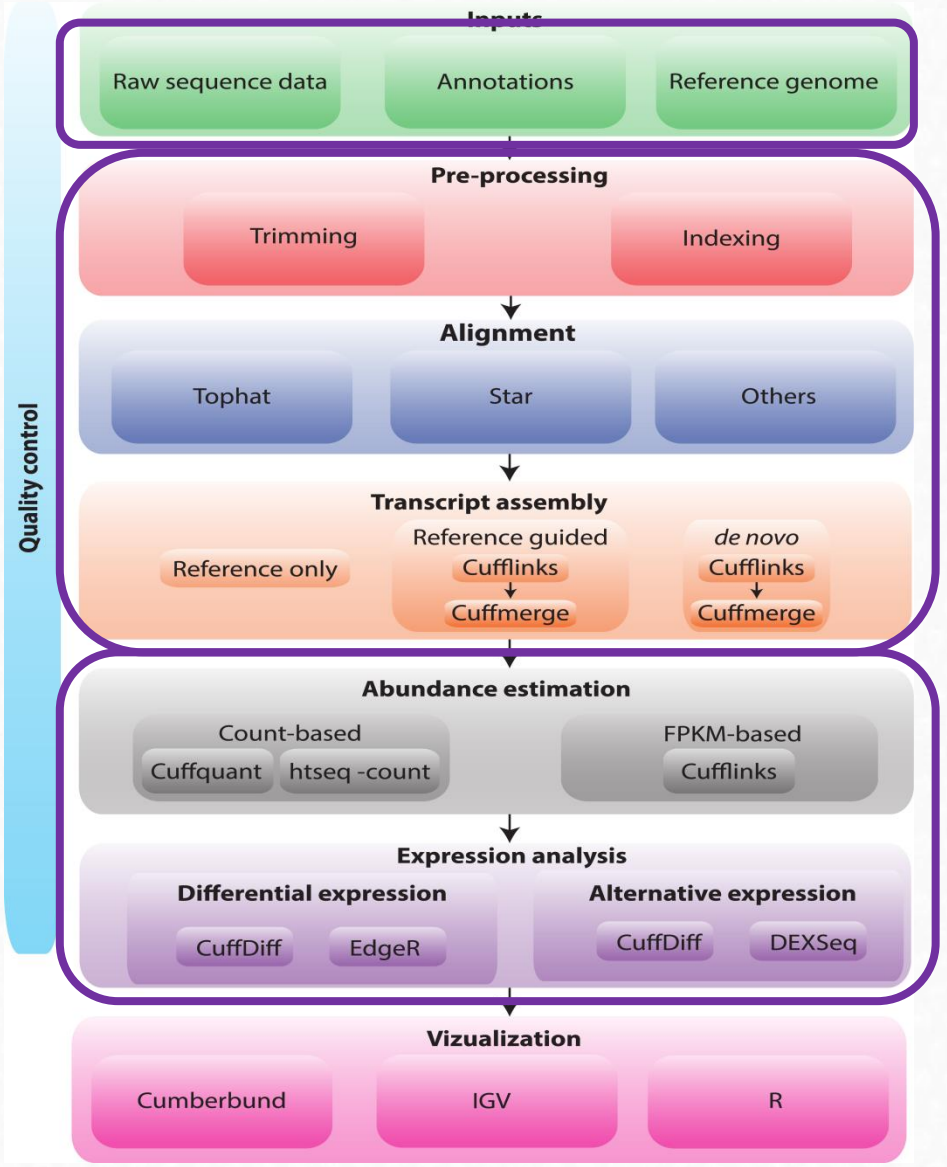


- Fastq files
- FASTQC (quality control)

- Alignment

Griffith, 2015 DOI: 10.1371/journal.pcbi.1004393

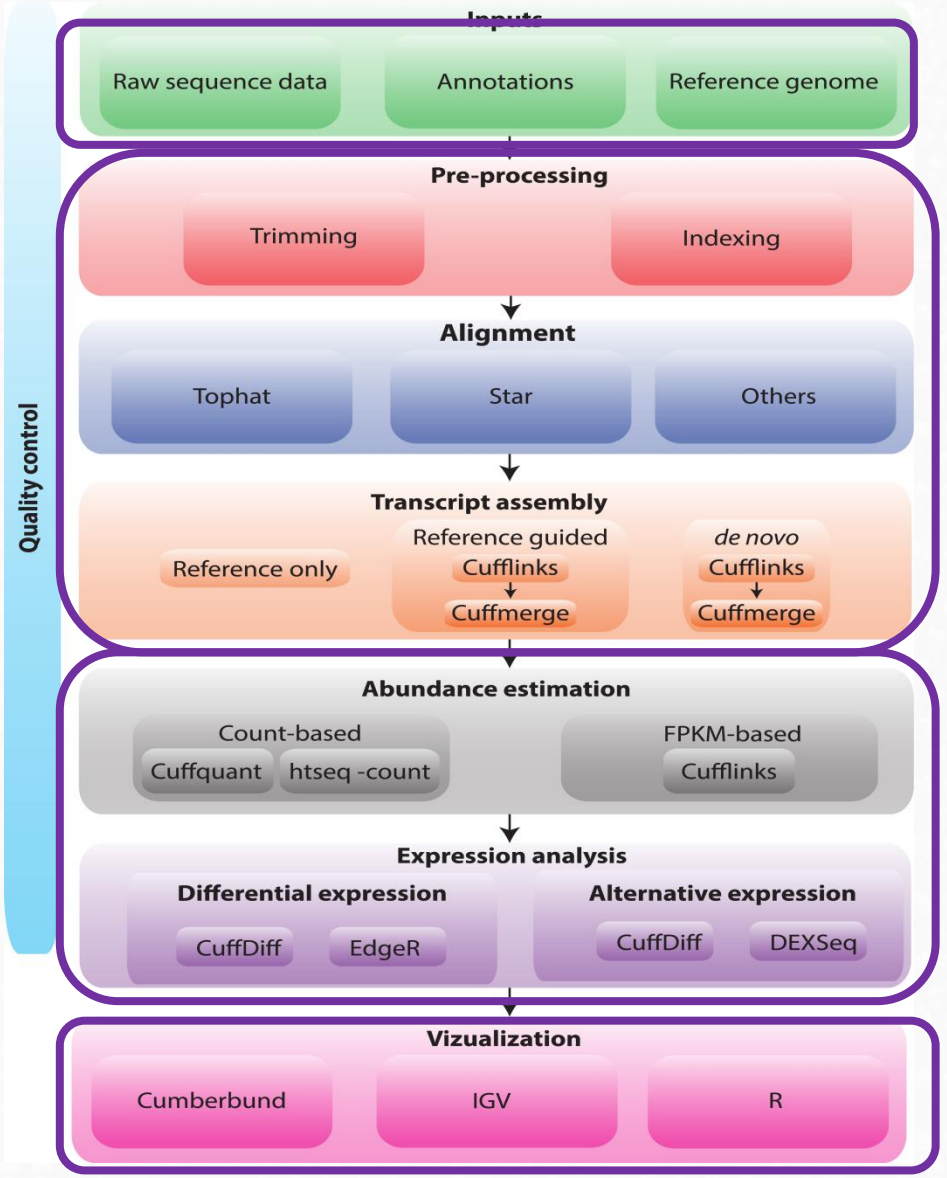
2.5 RNA-seq analysis pipeline(s)



- Fastq files
- FASTQC (quality control)
- Alignment
- DEG analysis

Griffith, 2015 DOI: 10.1371/journal.pcbi.1004393

2.5 RNA-seq analysis pipeline(s)



- Fastq files
- FASTQC (quality control)
- Alignment
- DEG analysis

+ Analysis of biological significance

Griffith, 2015 DOI: 10.1371/journal.pcbi.1004393

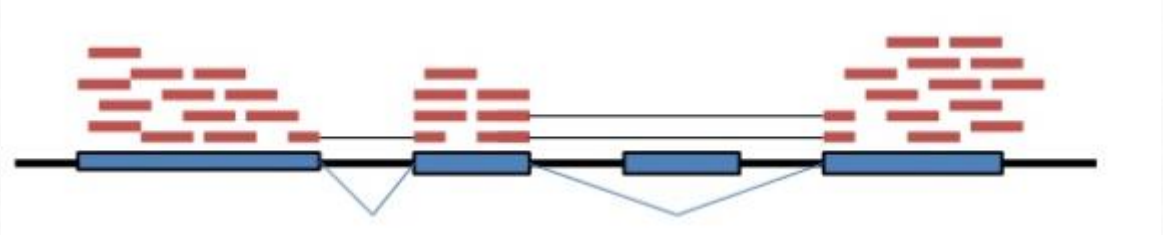
Vall d'Hebron
Institut de Recerca

1. Introduction to omics data analysis
2. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)
    6. Alignment
    7. RNA-seq pipeline with R

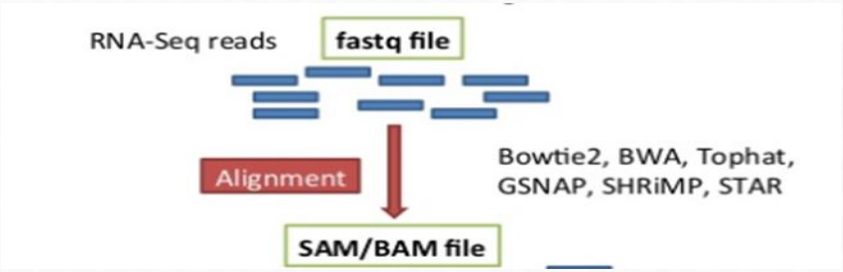
Vall d'Hebron
Institut de Recerca

What to map to?

Map to the genome, with knowledge of transcript annotations



- Well annotated genome reference is required.
- To effectively map to exon junctions, you need a mapping algorithm that can divide the sequencing reads and map portions independently.
- Identifying alternative transcript isoforms involves complex algorithms



Which sequence mappers to use?

- RNASeq Alignment algorithm must be
  - Fast
  - Able to handle SNPs, indels, and sequencing errors
  - Maintain accurate quantification
  - Allow for introns for reference genome alignment (spliced alignment detection)

Vall d'Hebron
Institut de Recerca

Which sequence mappers to use?

- RNASeq Alignment algorithm must be
  - Fast
  - Able to handle SNPs, indels, and sequencing errors
  - Maintain accurate quantification
  - Allow for introns for reference genome alignment (spliced alignment detection)

- Burrows-Wheeler Transform (BWT) mappers
  - Fast
  - Limited mismatches allowed (<3)
  - Limited indel detection ability
  - Examples: Bowtie2, BWA, Tophat, HISAT2
  - Use cases: large and conserved genome and transcriptomes

- Hash Table mappers
  - Require large amount of RAM for indexing
  - More mismatches allowed
  - Indel detecIon
  - Examples: GSNAP, SHRiMP, STAR
  - Use case: highly variable or smaller genomes, transcriptomes

Vall d'Hebron
Institut de Recerca

Which sequence mappers to use?

- RNASeq Alignment algorithm must be
  - Fast
  - Able to handle SNPs, indels, and sequencing errors

Front Genet. 2018; 9: 35.
PMCID: PMC5834436
Published online 2018 Feb 26. doi: [10.3389/fgene.2018.00035]
PMID: 29535759

Comparison of Burrows-Wheeler Transform-Based Mapping Algorithms Used in High-Throughput Whole-Genome Sequencing: Application to Illumina Data for Livestock Genomes[1]

Brittney N. Keel* and Warren M. Snelling

- Hash Table mappers
  - Require large amount of RAM for indexing
  - More mismatches allowed
  - Indel detection
  - Examples: GSNAP, SHRiMP, STAR
  - Use case: highly variable or smaller genomes, transcriptomes

2.6 Alignment

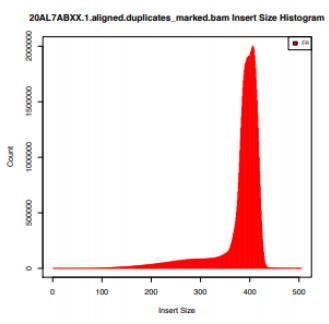
Steps with TopHat

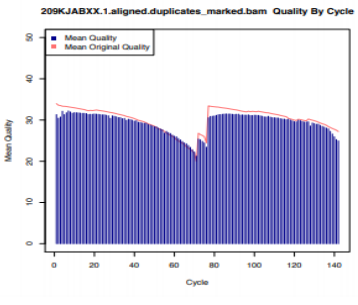
1. Unspliced reads are mapped to locate exons (with Bowtie)

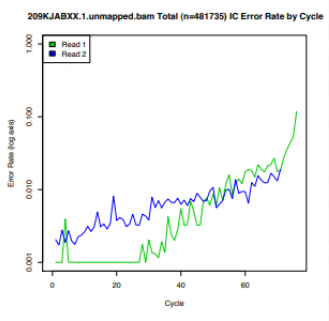
2. Unmapped reads are then split and aligned independently to identify exon

Important to check the quality of mapping process (percentage of mapped reads)

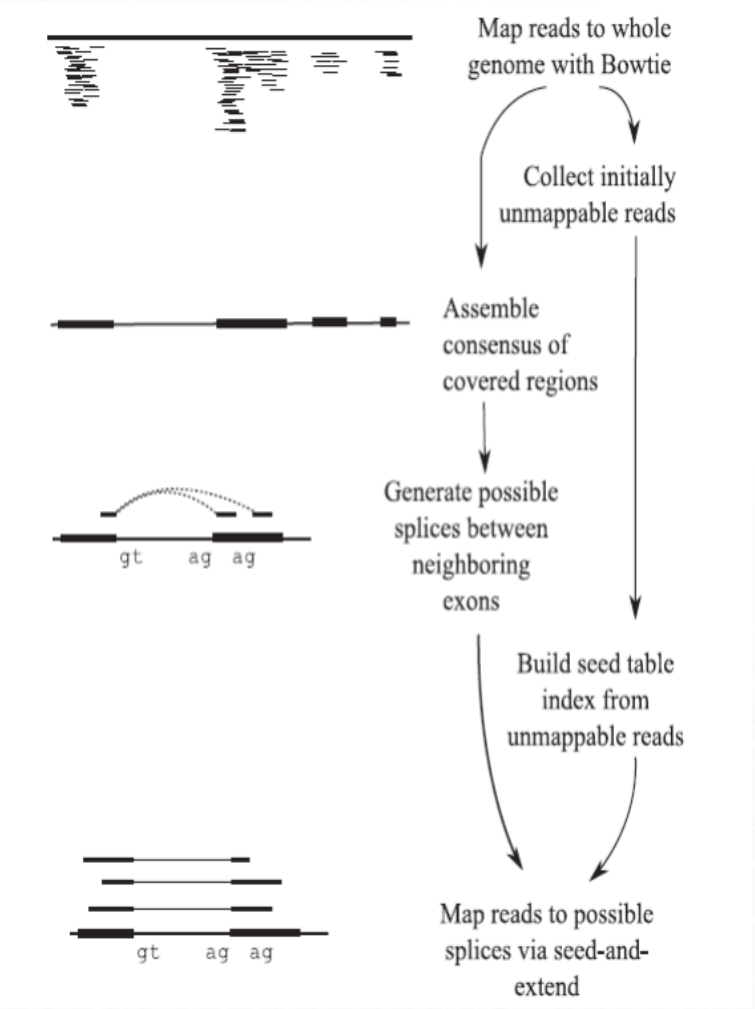
Picard can be used for quality control of mapping







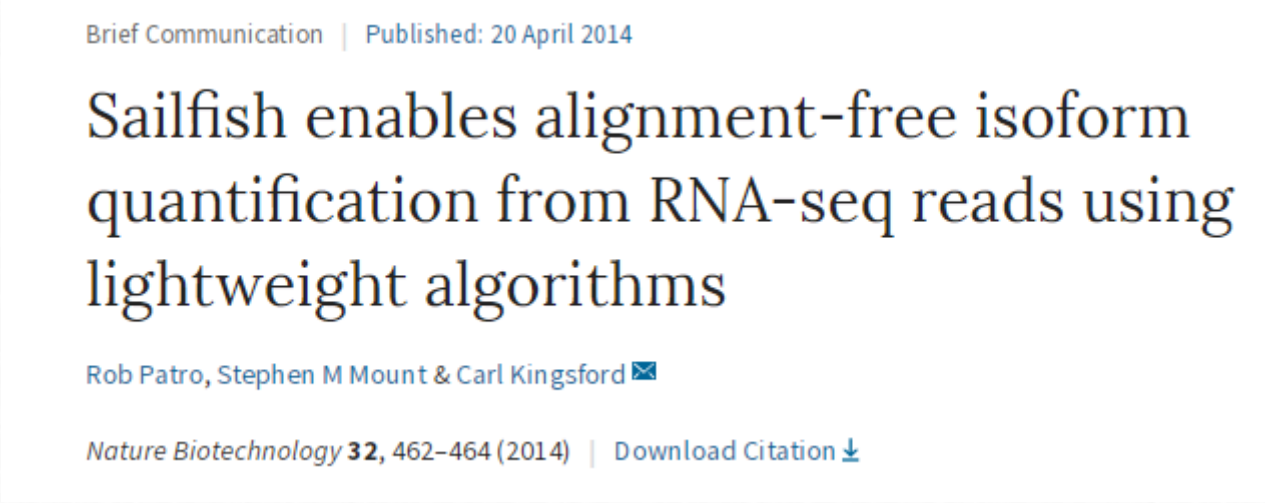
TopHat Pipeline



Vall d'Hebron
Institut de Recerca

Alignment-independent quantification for RNA-Seq

- Alignment steps are computationally heavy and can be very time-consuming even with multi-threading.
- In 2014, Sailfish method, demonstrated that it was not necessary to actually align each read to the genome in order to obtain accurate transcript each read.



Brief Communication | Published: 20 April 2014

Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms

Rob Patro, Stephen M Mount & Carl Kingsford

Nature Biotechnology **32**, 462–464 (2014) | Download Citation

https://cgatoxford.wordpress.com/2015/10/12/road-testing-kallisto/

Vall d'Hebron
Institut de Recerca

Alignment-independent quantification for RNA-Seq

- Alignment steps are computationally heavy and can be very time-consuming even with multi-threading.
- In 2014, Sailfish method, demonstrated that it was not necessary to actually align each read to the genome in order to obtain accurate transcript each read.
- All you actually need to do is establish the most likely transcript for each read

1. shredding the transcriptome and reads into kmers (short overlapping sequences
2. matching the transcriptome and read kmers (is a very fast and low memory usage)

https://cgatoxford.wordpress.com/2015/10/12/road-testing-kallisto/

Vall d'Hebron
Institut de Recerca

Alignment-independent quantification for RNA-Seq

- Nowadays there exist various tools:

  - Salmon, Kallisto, Sailfish

PROS:

- Extremely Fast & Lightweight (can quantify 20 million reads in five minutes on a laptop computer)
- Easy to use

Vall d'Hebron
Institut de Recerca

Alignment-independent quantification f

- Nowadays there exist various tools:
  - Salmon, Ka

antify 20 ues on a laptop computer)

Limitations of alignment-free tools in total RNA-seq quantification

Douglas C. Wu[1,2], Jun Yao[1,2], Kevin S. Ho[1,2], Alan M. Lambowitz[1,2] and Claus O. Wilke[1,3*]

Wu et al. BMC Genomics (2018) 19:510
https://doi.org/10.1186/s12864-018-4869-5

Conclusion: We have shown that alignment-free and traditional alignment-based quantification methods perform similarly for common gene targets, such as protein-coding genes. However, we have identified a potential pitfall in analyzing and quantifying lowly-expressed genes and small RNAs with alignment-free pipelines, especially when these small RNAs contain biological variations.

Vall d'Hebron
Institut de Recerca

1. Introduction to omics data analysis
2. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)
    6. Alignment
    7. RNA-seq pipeline with R

2.7 RNA-seq pipeline with R

We will use a workflow from Bioconductor project

https://www.bioconductor.org/packages/devel/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html

RNA-seq workflow: gene-level exploratory analysis and differential expression

***Michael I. Love*[1,2], *Simon Anders*[3], *Vladislav Kim*[4] *and Wolfgang Huber*[4]**

[1]Department of Biostatistics, UNC-Chapel Hill, Chapel Hill, NC, US
[2]Department of Genetics, UNC-Chapel Hill, Chapel Hill, NC, US
[3]Zentrum für Molekulare Biologie der Universität Heidelberg, Heidelberg, Germany
[4]European Molecular Biology Laboratory (EMBL), Heidelberg, Germany

16 October, 2019

This is one….but there are others….

https://f1000research.com/articles/4-1070

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Steps covered by the tutorial:

- Start with the FASTQ files (how they are alignment to the genome)
- Prepare a count matrix
- Exploratory data analysis (EDA) for quality assessment
- Differential gene expression analysis
- Visually explore the results

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Steps covered by the tutorial:

- ~~Start with the FASTQ files (how they are alignment to the genome)~~
- Prepare a count matrix
- Exploratory data analysis (EDA) for quality assessment
- Differential gene expression analysis
- Visually explore the results

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Data for the analysis:

- RNA-seq experiment with airway smooth muscle cells
- Cells were treated with dexamethasone (dexa)
- 4 primary human airway smooth muscle cell lines
- Experimental design
  - 4 samples treated with 1mM dexa 18 hours
  - 4 samples without treatment

2.7 RNA-seq pipeline with R

Vall d'Hebron Institut de Recerca

Data for the analysis:

- RNA-seq experiment w
- Cell were treated with
- 4 cell lines
- Experimental design
  - 4 samples treated
  - 4 samples without

NCBI

Gene Expression Omnibus

HOME | SEARCH | SITE MAP | GEO Publications | FAQ | MIAME | Email GEO

NCBI > GEO > **Accession Display** | Not logged in | Login

Scope: Self | Format: HTML | Amount: Quick | GEO accession: GSE52778 | GO

Series GSE52778 | Query DataSets for GSE52778

Status	Public on Jan 01, 2014
Title	Human Airway Smooth Muscle Transcriptome Changes in Response to Asthma Medications
Organism	Homo sapiens
Experiment type	Expression profiling by high throughput sequencing
Summary	Rationale: Asthma is a chronic inflammatory airway disease. The most common medications used for its treatment are β2-agonists and glucocorticosteroids, and one of the primary tissues that these drugs target in the treatment of asthma is the airway smooth muscle. We used RNA-Seq to characterize the human airway smooth muscle (HASM) transcriptome at baseline and under three asthma treatment conditions. Methods: The Illumina TruSeq assay was used to prepare 75bp paired-end libraries for HASM cells from four white male donors under four treatment conditions: 1) no treatment; 2) treatment with a β2-agonist (i.e. Albuterol, 1μM for 18h); 3) treatment with a glucocorticosteroid (i.e. Dexamethasone (Dex), 1μM for 18h); 4) simultaneous treatment with a β2-agonist and glucocorticoid, and the libraries were sequenced with an Illumina Hi-Seq 2000 instrument. The Tuxedo Suite Tools were used to align reads to the hg19 reference genome, assemble transcripts, and perform differential expression analysis using the protocol described in https://github.com/blancahimes/taffeta
Overall design	mRNA profiles obtained via RNA-Seq for four primary human airway smooth muscle cell lines that were treated with dexamethasone, albuterol, dexamethasone+albuterol or were left untreated.

PLoS One. 2014 Jun 13;9(6):e99625. doi: 10.1371/journal.pone.0099625. eCollection 2014.

RNA-Seq transcriptome profiling identifies CRISPLD2 as a glucocorticoid responsive gene that modulates cytokine function in airway smooth muscle cells.

Himes BE[1], Jiang X[2], Wagner P[2], Hu R[2], Wang Q[2], Klanderman B[3], Whitaker RM[4], Duan Q[4], Lasky-Su J[4], Nikolos C[5], Jester W[5], Johnson M[5], Panettieri RA Jr[5], Tantisira KG[4], Weiss ST[6], Lu Q[2].

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

The data

- airway package contains **eight files** with a small subset of the total number of reads in the experiment (bam files).
- The reads were selected which aligned to a small region of chromosome 1.
- Also contains a **targets file**: file which correlates samples (fastq) with experimental conditions

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

- Load the package where the data is stored

```
library("airway")
#to find out where on your computer the files have been
installed.
indir <- system.file("extdata", package="airway", mustWork=TRUE)
list.files(indir)
```

- See what files we have to perform the analysis

```
##  [1] "GSE52778_series_matrix.txt"
##  [2] "Homo_sapiens.GRCh37.75_subset.gtf"
##  [3] "sample_table.csv"
##  [4] "SraRunInfo_SRP033351.csv"
##  [5] "SRR1039508_subset.bam"
##  [6] "SRR1039509_subset.bam"
##  [7] "SRR1039512_subset.bam"
##  [8] "SRR1039513_subset.bam"
##  [9] "SRR1039516_subset.bam"
## [10] "SRR1039517_subset.bam"
## [11] "SRR1039520_subset.bam"
## [12] "SRR1039521_subset.bam"
```

2.7 RNA-seq pipeline with R

```
#read the targets file of the experiment
csvfile <- file.path(indir, "sample_table.csv")
sampleTable <- read.csv(csvfile, row.names = 1)
sampleTable
```

	SampleName	cell	dex	albut	Run	avgLength	Experiment	Sample	BioSample
SRR1039508	GSM1275862	N61311	untrt	untrt	SRR1039508	126	SRX384345	SRS508568	SAMN02422669
SRR1039509	GSM1275863	N61311	trt	untrt	SRR1039509	126	SRX384346	SRS508567	SAMN02422675
SRR1039512	GSM1275866	N052611	untrt	untrt	SRR1039512	126	SRX384349	SRS508571	SAMN02422678
SRR1039513	GSM1275867	N052611	trt	untrt	SRR1039513	87	SRX384350	SRS508572	SAMN02422670
SRR1039516	GSM1275870	N080611	untrt	untrt	SRR1039516	120	SRX384353	SRS508575	SAMN02422682
SRR1039517	GSM1275871	N080611	trt	untrt	SRR1039517	126	SRX384354	SRS508576	SAMN02422673
SRR1039520	GSM1275874	N061011	untrt	untrt	SRR1039520	101	SRX384357	SRS508579	SAMN02422683
SRR1039521	GSM1275875	N061011	trt	untrt	SRR1039521	98	SRX384358	SRS508580	SAMN02422677

2.7 RNA-seq pipeline with R

```
#read the targets file of the experiment
csvfile <- file.path(indir, "sample_table.csv")
sampleTable <- read.csv(csvfile, row.names = 1)
sampleTable
```

	SampleName	cell	dex	albut	Run	avgLength	Experiment	Sample	BioSample
SRR1039508	GSM1275862	N61311	untrt	untrt	SRR1039508	126	SRX384345	SRS508568	SAMN02422669
SRR1039509	GSM1275863	N61311	trt	untrt	SRR1039509	126	SRX384346	SRS508567	SAMN02422675
SRR1039512	GSM1275866	N052611	untrt	untrt	SRR1039512	126	SRX384349	SRS508571	SAMN02422678
SRR1039513	GSM1275867	N052611	trt	untrt	SRR1039513	87	SRX384350	SRS508572	SAMN02422670
SRR1039516	GSM1275870	N080611	untrt	untrt	SRR1039516	120	SRX384353	SRS508575	SAMN02422682
SRR1039517	GSM1275871	N080611	trt	untrt	SRR1039517	126	SRX384354	SRS508576	SAMN02422673
SRR1039520	GSM1275874	N061011	untrt	untrt	SRR1039520	101	SRX384357	SRS508579	SAMN02422683
SRR1039521	GSM1275875	N061011	trt	untrt	SRR1039521	98	SRX384358	SRS508580	SAMN02422677

You can create it with a spreadsheet software

Vall d'Hebron
Institut de Recerca

Generate count matrices

Once the reads have been aligned, there are a number of tools that can be used to count the number of reads/fragments that can be assigned to genomic features for each sample. These often take as input SAM/BAM alignment files and a file specifying the genomic features, e.g. a GFF3 or GTF file specifying the gene models.

function	package	framework	output	*DESeq2* input function
summarizeOverlaps	*GenomicAlignments*	R/Bioconductor	*SummarizedExperiment*	*DESeqDataSet*
featureCounts	*Rsubread*	R/Bioconductor	matrix	*DESeqDataSetFromMatrix*
tximport	*tximport*	R/Bioconductor	list of matrices	*DESeqDataSetFromTximport*
htseq-count	HTSeq	Python	files	*DESeqDataSetFromHTSeq*

Vall d'Hebron
Institut de Recerca

Generate count matrices

Here we will proceed using *summarizeOverlaps*, using he *Run* column in the sample table, we construct the full paths to the files we want to perform the counting operation on:

```
filenames <- file.path(indir, paste0(sampleTable$Run, "_subset.bam"))
file.exists(filenames)
filenames
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
filenames
```

```
## [1] "/home/rgonzalo/R/x86_64-pc-linux-gnu-library/3.2/airway/extdata/SRR1039508_subset.bam"
## [2] "/home/rgonzalo/R/x86_64-pc-linux-gnu-library/3.2/airway/extdata/SRR1039509_subset.bam"
## [3] "/home/rgonzalo/R/x86_64-pc-linux-gnu-library/3.2/airway/extdata/SRR1039512_subset.bam"
## [4] "/home/rgonzalo/R/x86_64-pc-linux-gnu-library/3.2/airway/extdata/SRR1039513_subset.bam"
## [5] "/home/rgonzalo/R/x86_64-pc-linux-gnu-library/3.2/airway/extdata/SRR1039516_subset.bam"
## [6] "/home/rgonzalo/R/x86_64-pc-linux-gnu-library/3.2/airway/extdata/SRR1039517_subset.bam"
## [7] "/home/rgonzalo/R/x86_64-pc-linux-gnu-library/3.2/airway/extdata/SRR1039520_subset.bam"
## [8] "/home/rgonzalo/R/x86_64-pc-linux-gnu-library/3.2/airway/extdata/SRR1039521_subset.bam"
```

Vall d'Hebron
Institut de Recerca

Generate count matrices

Note: make sure that the chromosome names of the genomic features in the annotation you use are consistent with the chromosome names of the reference used for read alignment. Otherwise, the scripts might fail to count any reads to features due to the mismatching names. For example, a common mistake is when the alignment files contain chromosome names in the style of `1` and the gene annotation in the style of `chr1`, or the other way around. See the *seqlevelsStyle* function in

Generate count matrices

- We indicate in Bioconductor that these files are BAM files using the BamFileList function from the Rsamtools package that provides an R interface to BAM files.
- Here we also specify details about how the BAM files should be treated, e.g., only process 2 million reads at a time

```
library("Rsamtools")
bamfiles <- BamFileList(filenames, yieldSize=2000000)
```

2.7 RNA-seq pipeline with R

2.5 Defining gene models

Next, we need to read in the gene model that will be used for counting reads/fragments. We will read the gene model from an Ensembl GTF file (Flicek et al. 2014), using *makeTxDbFromGFF* from the *GenomicFeatures* package. GTF files can be downloaded from Ensembl's FTP site or other gene model repositories. A *TxDb* object is a database that can be used to generate a variety of range-based objects, such as exons, transcripts, and genes. We want to make a list of exons grouped by gene for counting read/fragments.

```
library("GenomicFeatures")
gtffile <- file.path(indir,"Homo_sapiens.GRCh37.75_subset.gtf")
txdb <- makeTxDbFromGFF(gtffile, format = "gtf", circ_seqs = character())
txdb
```

2.7 RNA-seq pipeline with R

```
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: C:/Users/rgonz/Documents/R/win-
      library/4.0/airway/extdata/Homo_sapiens.GRCh37.75_subset.gtf
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
# Nb of transcripts: 65
# Db created by: GenomicFeatures package from Bioconductor
# Creation time: 2020-12-04 17:40:04 +0100 (Fri, 04 Dec 2020)
# GenomicFeatures version at creation time: 1.40.1
# RSQLite version at creation time: 2.2.1
# DBSCHEMAVERSION: 1.2
```

2.7 RNA-seq pipeline with R

The following line produces a *GRangesList* of all the exons grouped by gene (Lawrence et al. 2013). Each element of the list is a *GRanges* object of the exons for a gene.

```
ebg <- exonsBy(txdb, by="gene")
ebg
```

```
GRangesList object of length 20:
$ENSG00000009724
GRanges object with 18 ranges and 2 metadata columns:
       seqnames            ranges strand |   exon_id       exon_name
          <Rle>         <IRanges>  <Rle> | <integer>     <character>
   [1]        1 11086580-11087705      - |        98 ENSE00000818830
   [2]        1 11090233-11090307      - |        99 ENSE00000472123
   [3]        1 11090805-11090939      - |       100 ENSE00000743084
   [4]        1 11094885-11094963      - |       101 ENSE00000743085
   [5]        1 11097750-11097868      - |       102 ENSE00003482788
   ...      ...                 ...    ... .       ...             ...
  [14]        1 11106948-11107176      - |       111 ENSE00003467404
  [15]        1 11106948-11107176      - |       112 ENSE00003489217
  [16]        1 11107260-11107280      - |       113 ENSE00001833377
  [17]        1 11107260-11107284      - |       114 ENSE00001472289
  [18]        1 11107260-11107290      - |       115 ENSE00001881401
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

2.6 Read counting step

After these preparations, the actual counting is easy. The function *summarizeOverlaps* from the *GenomicAlignments* package will do this. This produces a *SummarizedExperiment* object that contains a variety of information about the experiment, and will be described in more detail below.

Note: If it is desired to perform counting using multiple cores, one can use the *register* and *MulticoreParam* or *SnowParam* functions from the *BiocParallel* package before the counting call below. Expect that the `summarizeOverlaps` call will take at least 30 minutes per file for a human RNA-seq file with 30 million aligned reads. By sending the files to separate cores, one can speed up the entire counting process.

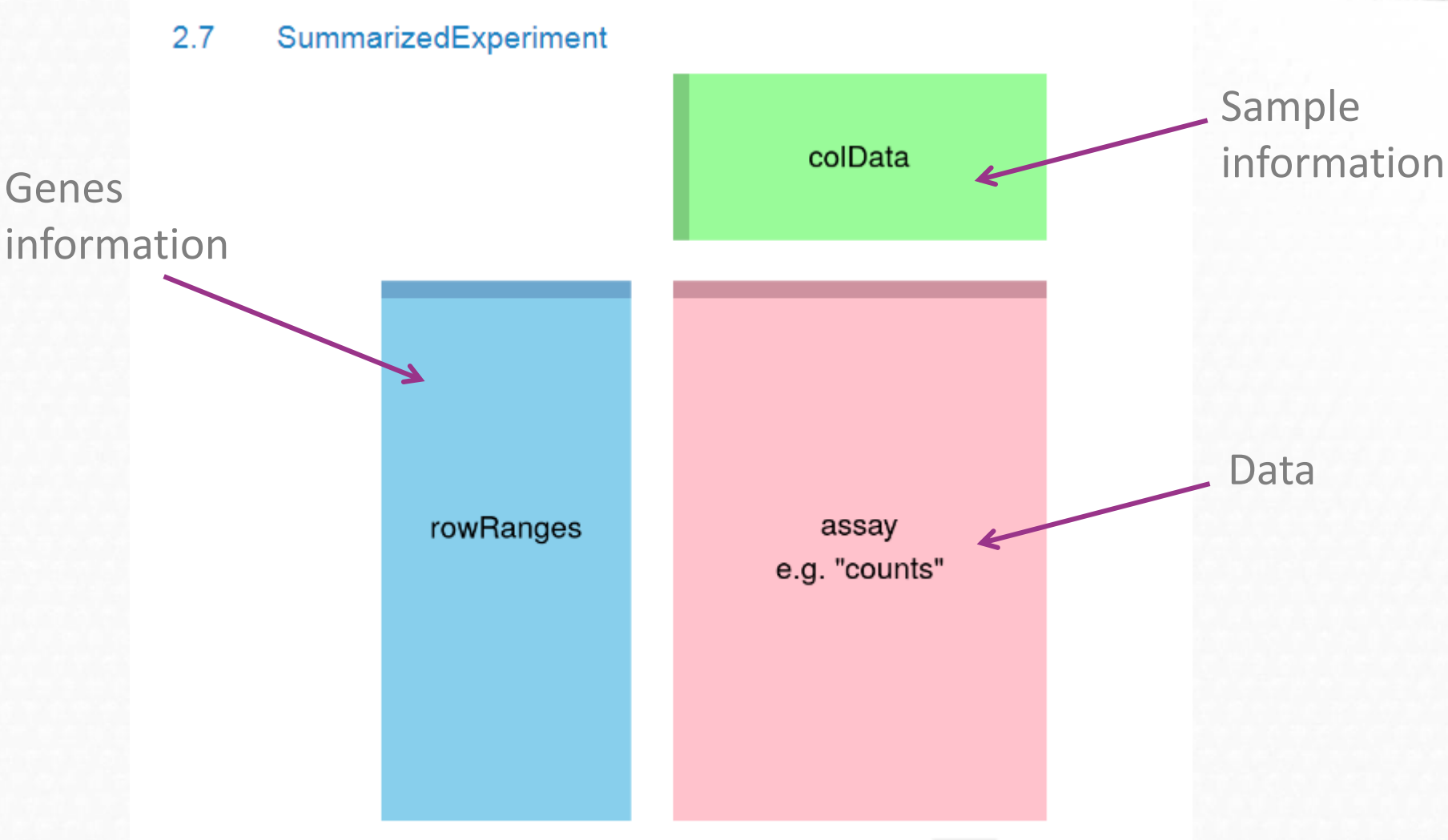
2.7 RNA-seq pipeline with R

```
library("GenomicAlignments")
library("BiocParallel")
library("DESeq2")

register(SerialParam())
```

The following call creates the SummarizedExperiment object with counts:

```
se <- summarizeOverlaps(features=ebg, reads=bamfiles,
                        mode="Union",
                        singleEnd=FALSE,
                        ignore.strand=TRUE,
                        fragments=TRUE )
```

2.7 RNA-seq pipeline with R



2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

```
##see the count matrix and its
dimensions
se
dim(se)
head(assay(se), 3)
```

```
> assay(se)
                 SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517 SRR1039520 SRR1039521
ENSG00000009724         38         28         66         24         42         41         47         36
ENSG00000116649       1004       1255       1122       1313       1100       1879        745       1536
ENSG00000120942        218        256        233        252        269        465        207        400
ENSG00000120948       2751       2080       3353       1614       3519       3716       2220       1990
ENSG00000171819          4         50         19        543          1         10         14       1067
ENSG00000171824        869       1075       1115       1051        944       1405        748       1590
ENSG00000175262          0          0          4          1          0          0          1          0
ENSG00000198793       1546       1719       1745       1970       2099       3280       1237       2521
ENSG00000207213          0          0          0          0          0          0          0          0
ENSG00000207451          0          0          0          1          0          0          0          0
ENSG00000215785          0          0          1          0          0          0          0          0
ENSG00000225602          1          0          0          1          0          0          0          0
ENSG00000226849          2          0          1          1          2          6          1          2
ENSG00000230337          2          0          4          4          0          5          1          3
ENSG00000238173          0          0          0          0          0          0          0          0
ENSG00000238199          0          1          0          0          2          0          0          0
ENSG00000253086          1          0          0          0          0          0          0          0
ENSG00000264181          0          0          0          0          0          0          0          0
ENSG00000271794          0          0          0          0          0          0          0          0
ENSG00000271895         42         37         36         26         31         42         33         23
> #dimensions of the count matrix
> dim(se)
[1] 20  8
```

2.7 RNA-seq pipeline with R

We have created the object ***gse*** with three matrices:

- **“counts”** - the estimated fragment counts for each gene and sample,
- **“abundance”** - the estimated transcript abundances in TPM, and
- **“length”** - the effective gene lengths which include changes in length due to biases as well as due to transcript usage.

Vall d'Hebron
Institut de Recerca

Analysis of differential expressed genes

Now he will began with full count matrix corrsponding to all the samples and all the data

```
data(gse)
gse
```

```
dim(gse)
[1] 58294 8
```

```
assayNames(gse)
## [1] "counts"    "abundance" "length"
head(assay(gse), 3)
##                    SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
## ENSG00000000003.14    708.164    467.962    900.992    424.368   1188.295
## ENSG00000000005.5       0.000      0.000      0.000      0.000      0.000
## ENSG00000000419.12    455.000    510.000    604.000    352.000    583.000
##                    SRR1039517 SRR1039520 SRR1039521
## ENSG00000000003.14   1090.668    805.929    599.337
## ENSG00000000005.5       0.000      0.000      0.000
## ENSG00000000419.12    773.999    409.999    499.000
```

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

- We can quickly check the millions of fragments that uniquely aligned to the genes

```
round(colSums(assay(gse)) / 1e6, 1)
SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517 SRR1039520 SRR1039521
      21.1       19.3       26.1       15.7       25.3       31.9       19.7       21.8
```

- to make sure that the object contains all the necessary information about the samples

```
colData(gse)
DataFrame with 8 rows and 3 columns
                names    donor     condition
              <factor> <factor>      <factor>
SRR1039508 SRR1039508  N61311  Untreated
SRR1039509 SRR1039509  N61311  Dexamethasone
SRR1039512 SRR1039512  N052611 Untreated
SRR1039513 SRR1039513  N052611 Dexamethasone
SRR1039516 SRR1039516  N080611 Untreated
SRR1039517 SRR1039517  N080611 Dexamethasone
SRR1039520 SRR1039520  N061011 Untreated
SRR1039521 SRR1039521  N061011 Dexamethasone
```

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

- We can rename our variables if we want. Let's use cell to denote the donor cell line, and dex to denote the treatment condition

```
gse$cell <- gse$donor
gse$dex <- gse$condition
```

- We can also change the names of the levels. It is critical when one renames levels to not change the order. Here we will rename "Untreated" as "untrt" and "Dexamethasone" as "trt":

```
levels(gse$dex) <- c("untrt", "trt")
```

2.7 RNA-seq pipeline with R

- We can construct a DESeqDataSet object that will be the starting point of the analysis. We add an appropriate design for the analysis:

```
library("DESeq2")
dds <- DESeqDataSet(gse, design = ~ cell + dex)
```

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

4 Exploratory analysis and visualization

4.1 Pre-filtering the dataset

Our count matrix with our *DESeqDataSet* contains many rows with only zeros, and additionally many rows with only a few fragments total. In order to reduce the size of the object, and to increase the speed of our functions, we can remove the rows that have no or nearly no information about the amount of gene expression. Here we apply the most minimal filtering rule: removing rows of the *DESeqDataSet* that have no counts, or only a single count across all samples. Additional weighting/filtering to improve power is applied at a later step in the workflow.

2.7 RNA-seq pipeline with R

- It is recommended to filter for **lowly expressed genes** before differential expression testing.

  - provide little evidence for differential expression and they interfere with some of the statistical approximations used
  - Add to the multiple testing burden when estimating FDR, reducing power to detect differential expressed genes.

```
##Filter out those rows without any count
> nrow(dds)
[1] 58294
> dds <-[rowSums(counts(dds)) > 1, ]
Error: inesperado '[' in "dds <-["
> dds <- dds[rowSums(counts(dds)) > 1, ]
> nrow(dds)
[1] 31604
```

4.2 The variance stabilizing transformation and the rlog

Many common statistical methods for exploratory analysis of multidimensional data, for example clustering and *principal components analysis* (PCA), work best for data that generally has the same range of variance at different ranges of the mean values. For RNA-seq counts, however, the expected variance grows with the mean (genes with *highest* counts show the largest absolute differences between samples)

A simple and often used strategy to avoid this is to take the logarithm of the normalized count values plus a pseudocount of 1

DESeq2 offers transformations for count data that stabilize the variance across the mean: the ***variance stabilizing transformation*** **(VST)**

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

```
vsd <- vst(dds, blind = FALSE)
```

```
> head(assay(vsd), 3)
                  SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517 SRR1039520 SRR1039521
ENSG00000000003.14  10.105781   9.852029  10.169726   9.991545  10.424865  10.194490  10.315814  10.002177
ENSG00000000419.12   9.692244   9.923647   9.801921   9.798653   9.763455   9.874703   9.683211   9.845507
ENSG00000000457.13   9.449592   9.312186   9.362754   9.459168   9.281415   9.395937   9.477971   9.477027
```

2.7 RNA-seq pipeline with R

4.3 Sample distances

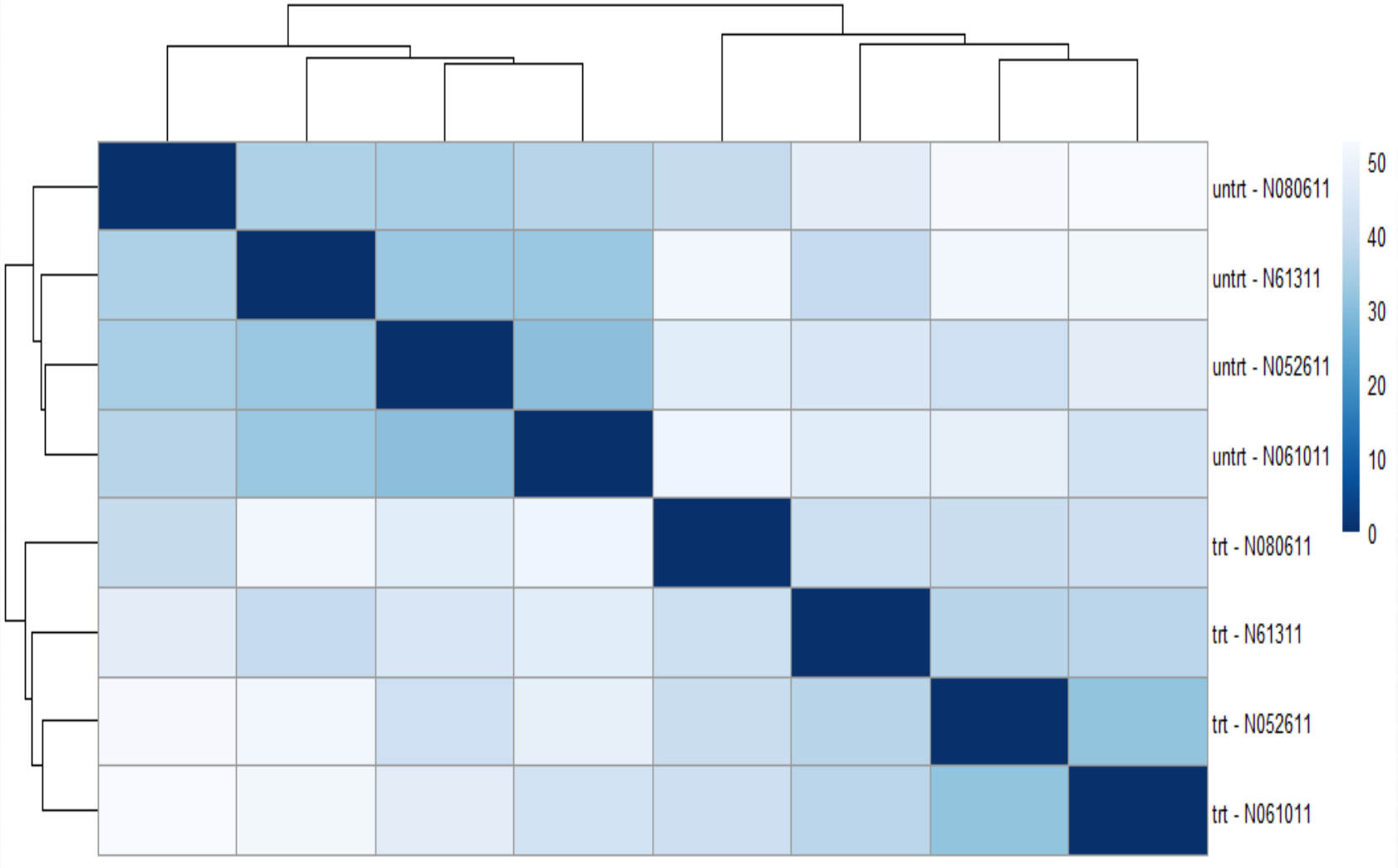
A useful first step in an RNA-seq analysis is often to assess overall similarity between samples: Which samples are similar to each other, which are different? Does this fit to the expectation from the experiment's design?

```
#Sample distances

sampleDists <- dist(t(assay(vsd)))
sampleDists
```

Heatmap + Hierarchical Clustering

```
library("pheatmap")
library("RColorBrewer")

sampleDistMatrix <- as.matrix( sampleDists )
rownames(sampleDistMatrix) <- paste( vsd$dex, vsd$cell, sep = " - " )
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows = sampleDists,
         clustering_distance_cols = sampleDists,
         col = colors)
```

2.7 RNA-seq pipeline with R



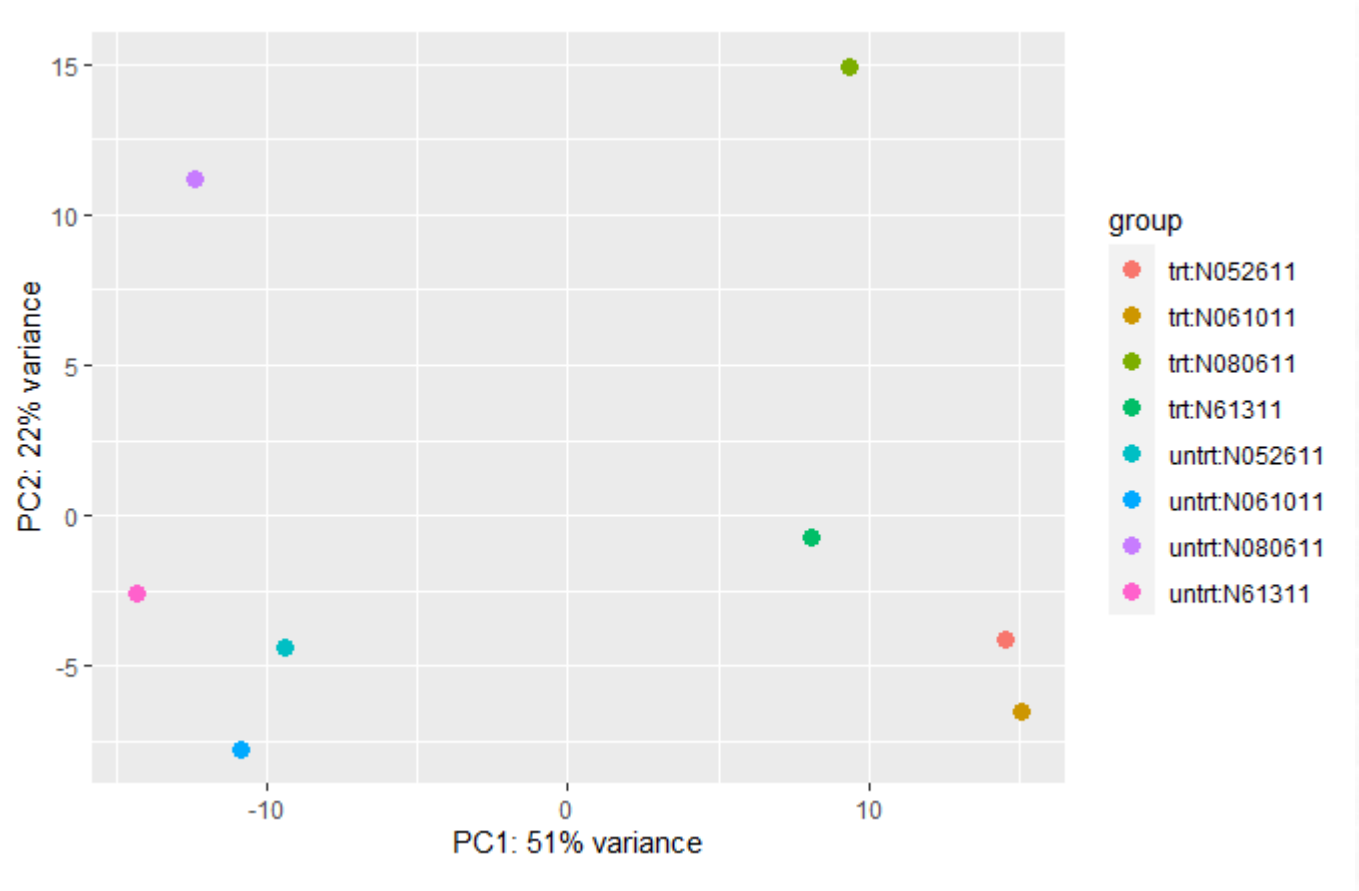
Vall d'Hebron
Institut de Recerca

Principal component Analysis Plot

```
plotPCA(vsd, intgroup = c("dex", "cell"))
```

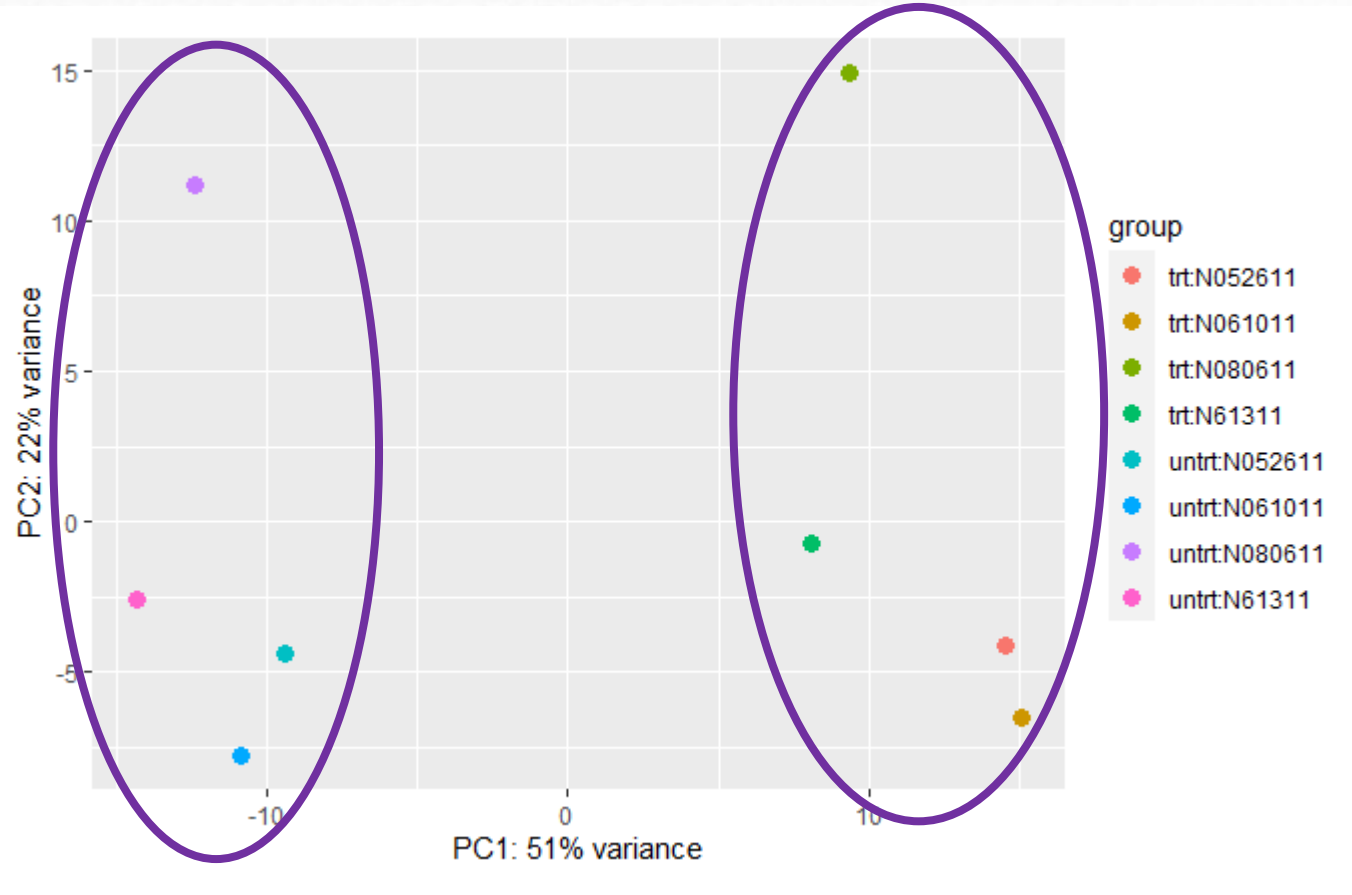
2.7 RNA-seq pipeline with R

Principal component Analysis Plot



Vall d'Hebron
Institut de Recerca

Principal component Analysis Plot



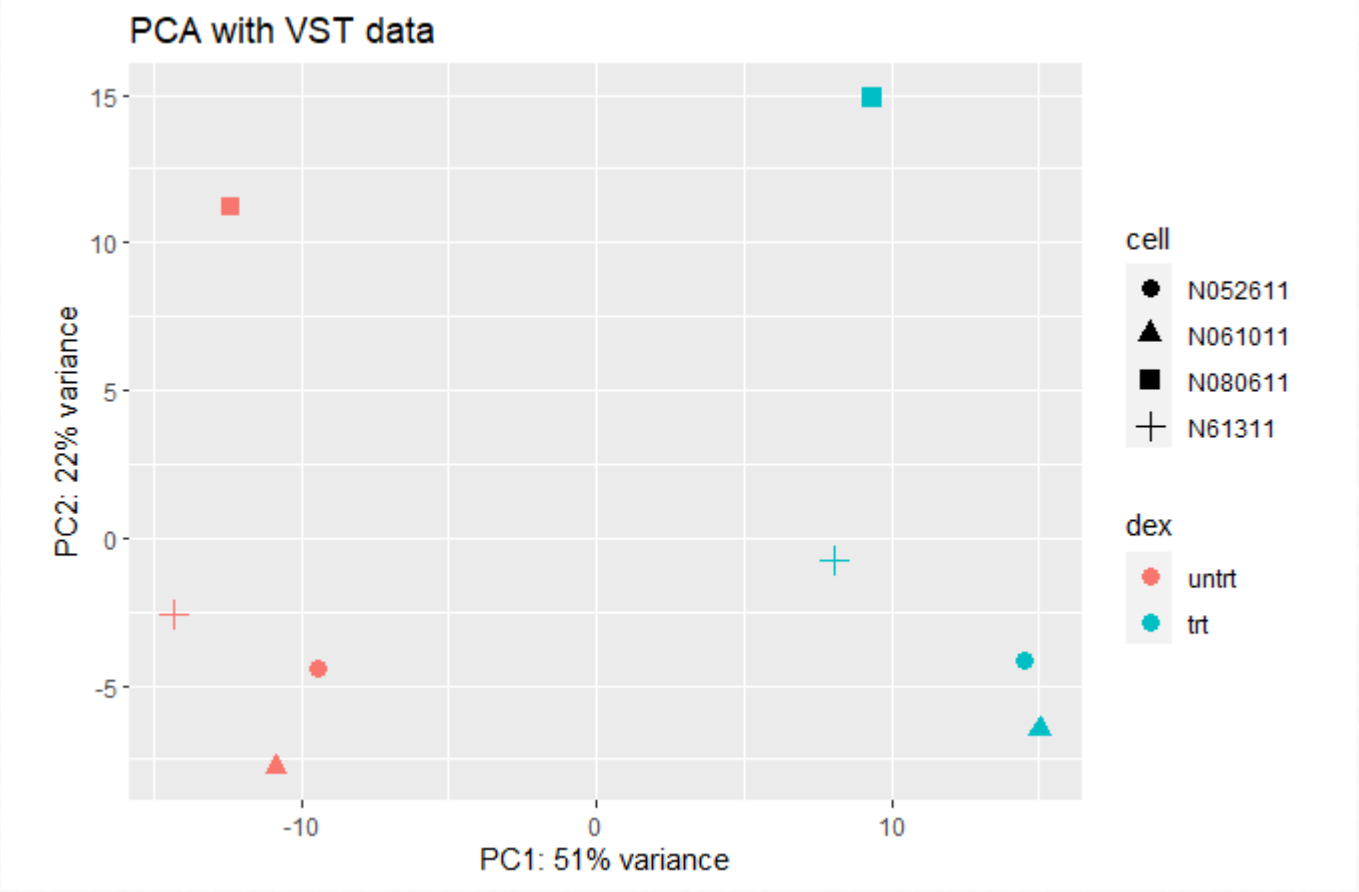
Vall d'Hebron
Institut de Recerca

Principal component Analysis Plot

```
pcaData <- plotPCA(vsd, intgroup = c( "dex", "cell"), returnData = TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(x = PC1, y = PC2, color = dex, shape = cell)) +
  geom_point(size =3) +
  xlab(paste0("PC1: ", percentVar[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar[2], "% variance")) +
  coord_fixed() +
  ggtitle("PCA with VST data")
```

Vall d'Hebron
Institut de Recerca

Principal component Analysis Plot



Vall d'Hebron
Institut de Recerca

5 Differential expression analysis

5.1 Running the differential expression pipeline

```
dds <- DESeq(dds, parallel =TRUE)

estimating size factors
using 'avgTxLength' from assays(dds), correcting for library size
estimating dispersions
gene-wise dispersion estimates: 1 workers
mean-dispersion relationship
final dispersion estimates, fitting model and testing: 1 workers
```

Vall d'Hebron
Institut de Recerca

Data Normalization

The counts of mapped reads for each gene is proportional to the expression of RNA (“interesting”) in addition to many other factors (“uninteresting”).

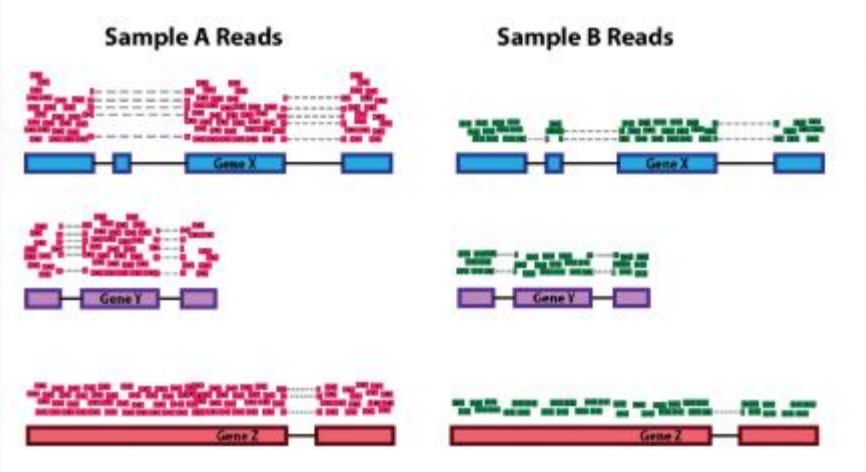
Normalization is the process of scaling raw count values to account for the “uninteresting” factors.

https://hbctraining.github.io/DGE_workshop/lessons/02_DGE_count_normalization.html

2.7 RNA-seq pipeline with R

Main factors often considered during normalization are:

1. **Sequencing depth:** total number or reads mapped to the genome

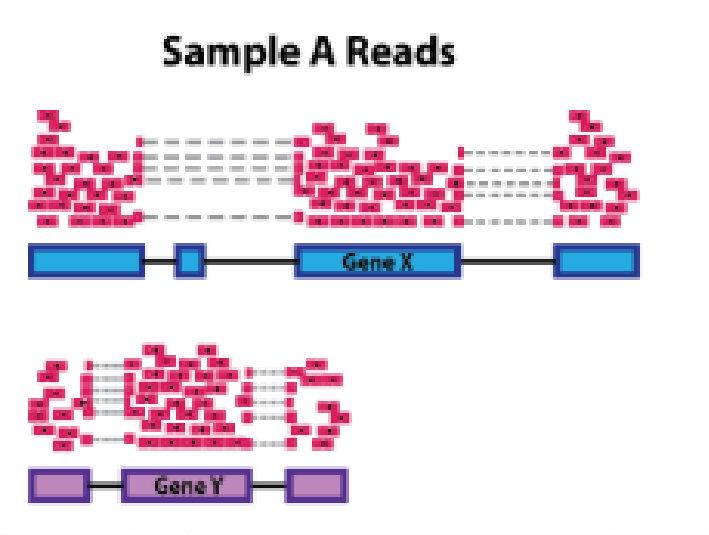


https://hbctraining.github.io/DGE_workshop/lessons/02_DGE_count_normalization.html

2.7 RNA-seq pipeline with R

Main factors often considered during normalization are:

2. **Gene length:** Accounting for gene length is necessary for comparing expression between different genes within the same sample.

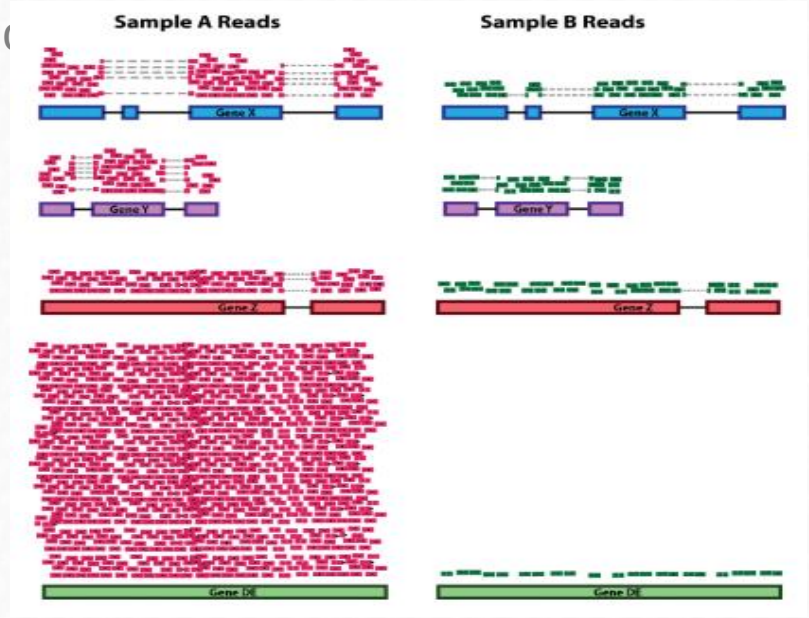


https://hbctraining.github.io/DGE_workshop/lessons/02_DGE_count_normalization.html

2.7 RNA-seq pipeline with R

Main factors often considered during normalization are:

3. **RNA composition:** A few highly differentially expressed genes between samples, differences in the number of genes expressed between samples, or presence of contamination can skew some types of normalizati



https://hbctraining.github.io/DGE_workshop/lessons/02_DGE_count_normalization.html

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

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**

https://hbctraining.github.io/DGE_workshop/lessons/02_DGE_count_normalization.html

Vall d'Hebron
Institut de Recerca

Check the results

```
res <- results(dds, contrast=c("dex","trt","untrt"))
res
```

```
log2 fold change (MLE): dex trt vs untrt
Wald test p-value: dex trt vs untrt
DataFrame with 31604 rows and 6 columns
                   baseMean log2FoldChange     lfcSE      stat      pvalue       padj
                  <numeric>      <numeric> <numeric> <numeric>   <numeric>  <numeric>
ENSG00000000003.14 739.940717     -0.3611537  0.106869 -3.379419 0.000726392 0.00531137
ENSG00000000419.12 511.735722      0.2063147  0.128665  1.603509 0.108822318 0.29318870
ENSG00000000457.13 314.194855      0.0378308  0.158633  0.238479 0.811509461 0.92255697
ENSG00000000460.16  79.793622     -0.1152590  0.314991 -0.365912 0.714430444 0.87298038
ENSG00000000938.12   0.307267     -1.3691185  3.503764 -0.390757 0.695977205         NA
...                       ...            ...       ...       ...         ...        ...
ENSG00000285979.1   38.353886      0.3423657  0.359511  0.952310    0.340940   0.600750
ENSG00000285987.1    1.562508      0.7064145  1.547295  0.456548    0.647996         NA
ENSG00000285990.1    0.642315      0.3647333  3.433276  0.106235    0.915396         NA
ENSG00000285991.1   11.276284     -0.1165515  0.748601 -0.155692    0.876275   0.952921
ENSG00000285994.1    3.651041     -0.0960094  1.068660 -0.089841    0.928414         NA
```

metadata with information on the meaning of the columns:

```
mcols(res, use.names = TRUE)

DataFrame with 6 rows and 2 columns
                       type                                 description
                 <character>                                 <character>
baseMean       intermediate mean of normalized counts for all samples
log2FoldChange      results  log2 fold change (MLE): dex trt vs untrt
lfcSE               results          standard error: dex trt vs untrt
stat                results          Wald statistic: dex trt vs untrt
pvalue              results       Wald test p-value: dex trt vs untrt
padj                results                    BH adjusted p-values
```

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

We can also summarize the results with the following line of code:

```
summary(res)

out of 31604 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 2373, 7.5%
LFC < 0 (down)     : 1949, 6.2%
outliers [1]       : 0, 0%
low counts [2]     : 14706, 47%
(mean count < 9)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

2.7 RNA-seq pipeline with R

Vall d'Hebron Institut de Recerca

Other comparisons: Here we extract results for the log2 of the fold change of one cell line over another

```
results(dds, contrast = c("cell", "N061011", "N61311"))
```

```
log2 fold change (MLE): cell N061011 vs N61311
Wald test p-value: cell N061011 vs N61311
DataFrame with 31604 rows and 6 columns
                    baseMean log2FoldChange     lfcSE       stat    pvalue      padj
                   <numeric>      <numeric> <numeric>  <numeric> <numeric> <numeric>
ENSG00000000003.14 739.940717       0.270945  0.152171   1.780534 0.0749886  0.378828
ENSG00000000419.12 511.735722      -0.071831  0.182817  -0.392912 0.6943842  0.936703
ENSG00000000457.13 314.194855       0.179881  0.225122   0.799036 0.4242696  0.820733
ENSG00000000460.16  79.793622      -0.119482  0.441594  -0.270570 0.7867217  0.960662
ENSG00000000938.12   0.307267       0.000000  4.997580   0.000000 1.0000000        NA
...                       ...            ...       ...        ...       ...       ...
ENSG00000285979.1   38.353886      0.0589757  0.512391  0.1150989  0.908367   0.98371
ENSG00000285987.1    1.562508      1.0216804  2.201861  0.4640078  0.642642        NA
ENSG00000285990.1    0.642315     -3.0956404  4.852715 -0.6379193  0.523526        NA
ENSG00000285991.1   11.276284     -0.8779628  1.046963 -0.8385804  0.401705        NA
ENSG00000285994.1    3.651041     -0.0192351  1.513236 -0.0127112  0.989858        NA
```

2.7 RNA-seq pipeline with R

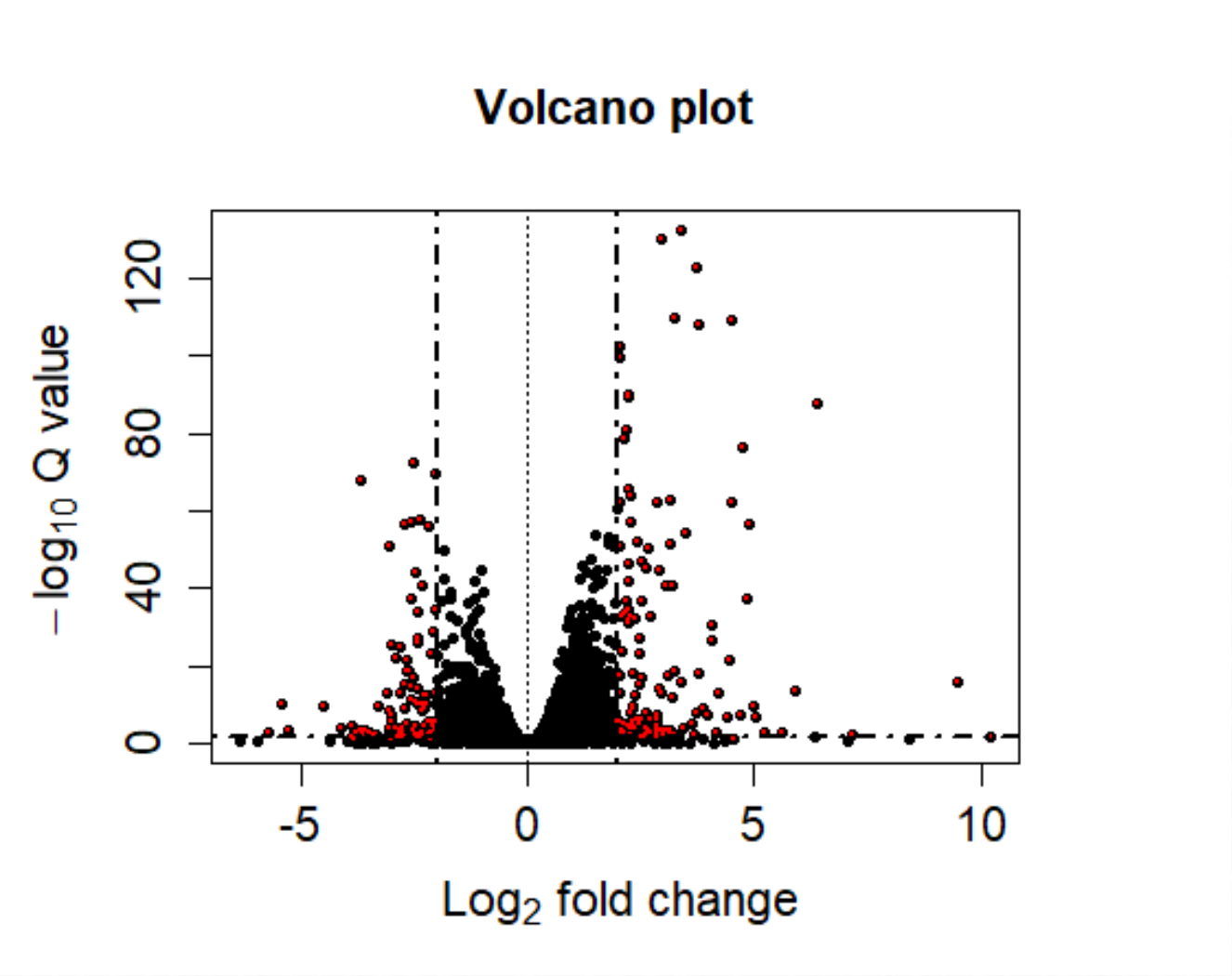
Vall d'Hebron
Institut de Recerca

Volcano plot

```
par(mar=c(5,5,5,5), cex=1.0, cex.main=1.4, cex.axis=1.4, cex.lab=1.4)

topT <- as.data.frame(res)

#Adjusted P values (FDR Q values)
with(topT, plot(log2FoldChange, -log10(padj), pch=20, main="Volcano plot",
cex=1.0, xlab=bquote(~Log[2]~fold~change), ylab=bquote(~-
log[10]~Q~value)))
with(subset(topT, padj<0.05 & abs(log2FoldChange)>2),
points(log2FoldChange, -log10(padj), pch=20, col="red", cex=0.5))

#Add lines for absolute FC>2 and P-value cut-off at FDR Q<0.05
abline(v=0, col="black", lty=3, lwd=1.0)
abline(v=-2, col="black", lty=4, lwd=2.0)
abline(v=2, col="black", lty=4, lwd=2.0)
abline(h=-log10(max(topT$pvalue[topT$padj<0.05], na.rm=TRUE)),
col="black", lty=4, lwd=2.0)
```

2.7 RNA-seq pipeline with R

Volcano plot



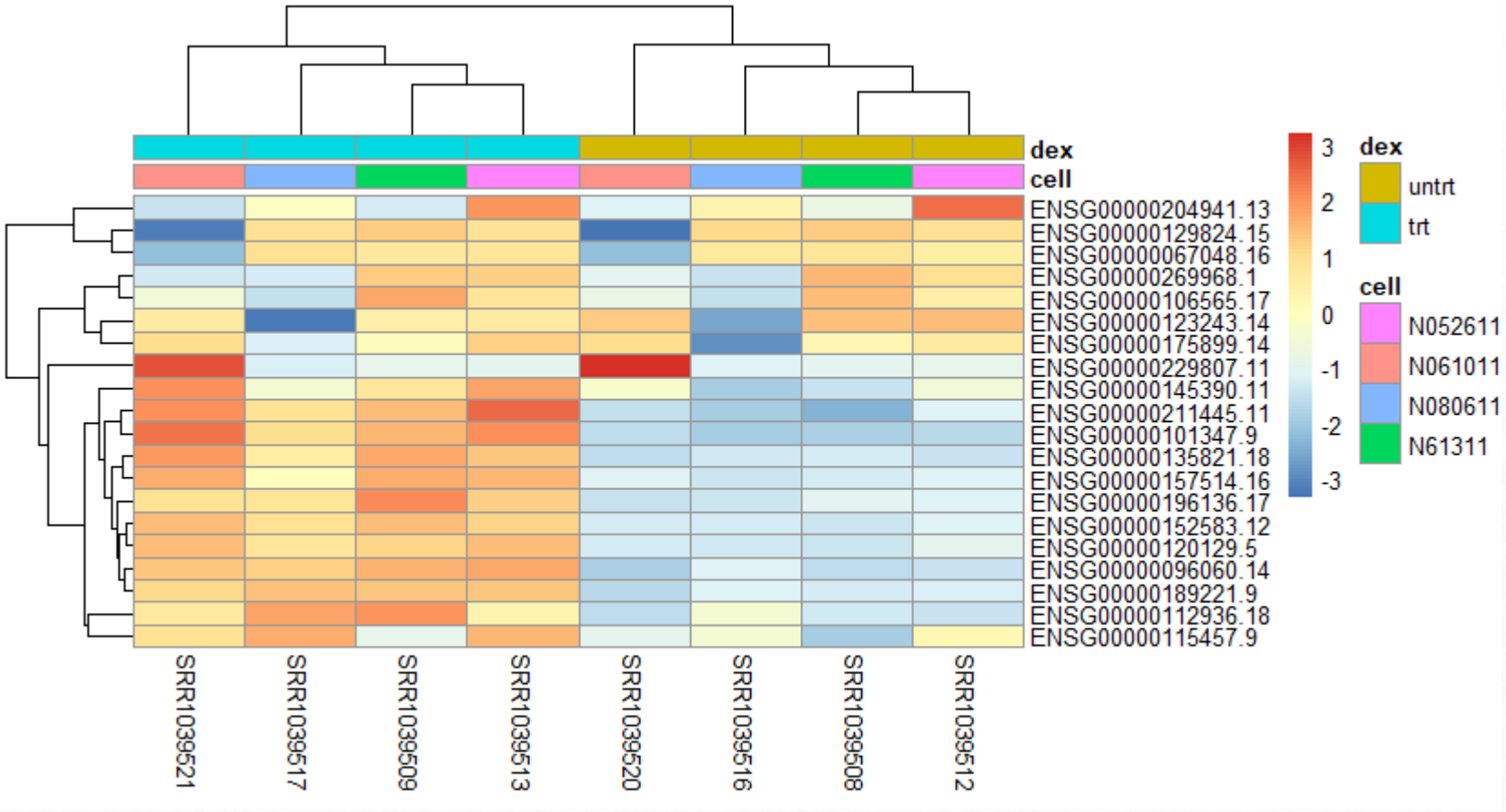
2.7 RNA-seq pipeline with R

6.3 Gene clustering

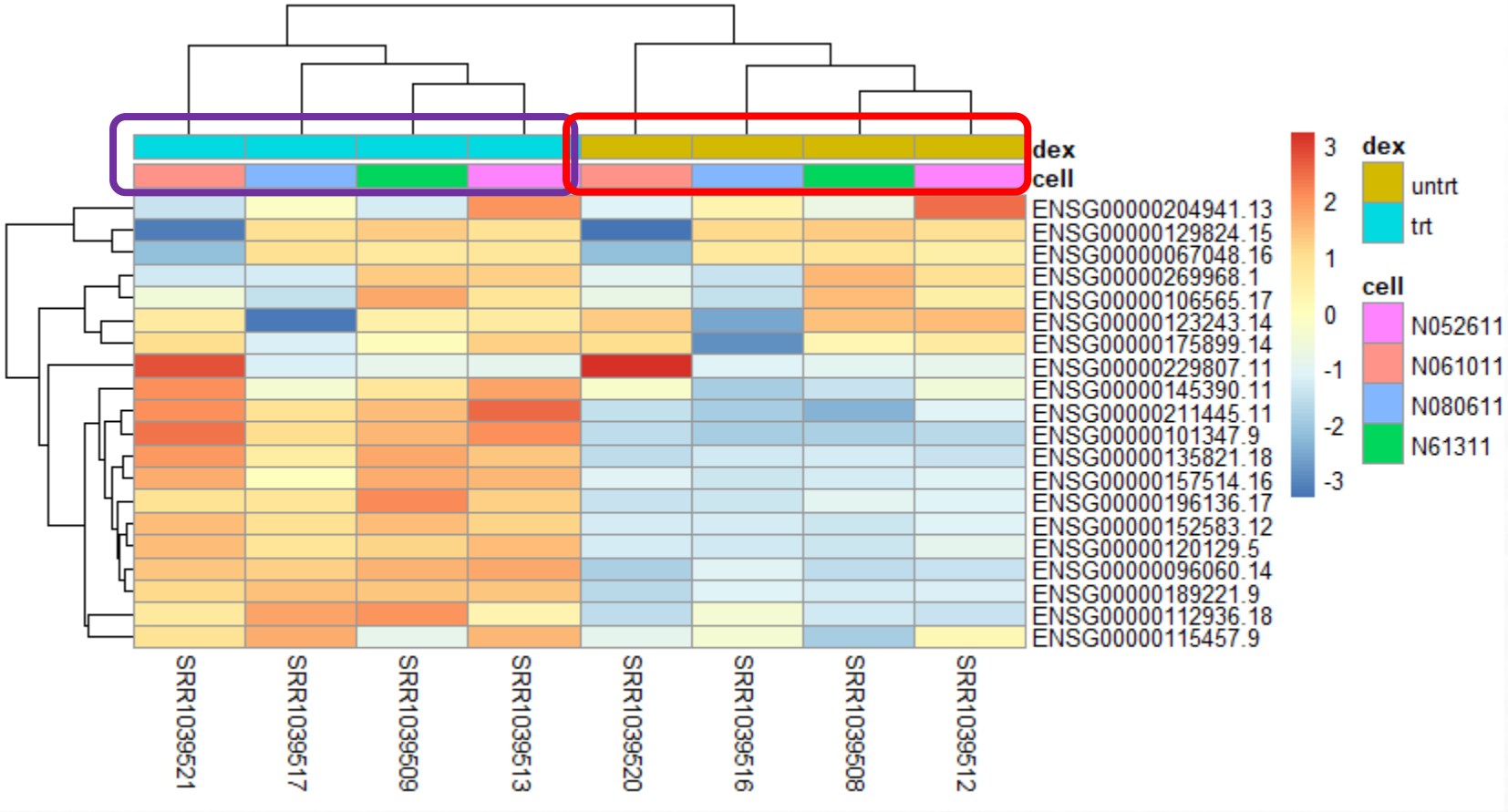
In the sample distance heatmap made previously, the dendrogram at the side shows us a hierarchical clustering of the samples. Such a clustering can also be performed for the genes. Since the clustering is only relevant for genes that actually carry a signal, one usually would only cluster a subset of the most highly variable genes. Here, for demonstration, let us select the 20 genes with the highest variance across samples. We will work with the VST data.

```
library("genefilter")
topVarGenes <- head(order(rowVars(assay(vsd)), decreasing = TRUE), 20)
mat  <- assay(vsd)[ topVarGenes, ]
mat  <- mat - rowMeans(mat)
anno <- as.data.frame(colData(vsd)[, c("cell","dex")])
pheatmap(mat, annotation_col = anno)
```

2.7 RNA-seq pipeline with R



2.7 RNA-seq pipeline with R



Vall d'Hebron
Institut de Recerca

7 Annotating and exporting results

Our result table so far only contains the Ensembl gene IDs, but alternative gene names may be more informative for interpretation. Bioconductor's annotation packages help with mapping various ID schemes to each other. We load the *AnnotationDbi* package and the annotation package *org.Hs.eg.db*:

```
library("AnnotationDbi")
library("org.Hs.eg.db")

ens.str <- substr(rownames(res), 1, 15)
res$symbol <- mapIds(org.Hs.eg.db,
                     keys=ens.str,
                     column="SYMBOL",
                     keytype="ENSEMBL",
                     multiVals="first")
res$entrez <- mapIds(org.Hs.eg.db,
                     keys=ens.str,
                     column="ENTREZID",
                     keytype="ENSEMBL",
                     multiVals="first")
resOrdered <- res[order(res$pvalue),]
head(resOrdered)
```

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

```
log2 fold change (MLE): dex trt vs untrt
Wald test p-value: dex trt vs untrt
DataFrame with 6 rows and 8 columns
                  baseMean log2FoldChange     lfcSE      stat       pvalue         padj      symbol
                 <numeric>      <numeric> <numeric> <numeric>    <numeric>    <numeric> <character>
ENSG00000189221.9   2373.805        3.39828  0.136172   24.9558 1.84494e-137 3.11758e-133        MAOA
ENSG00000120129.5   3420.727        2.97394  0.120300   24.7211 6.35042e-135 5.36547e-131       DUSP1
ENSG00000101347.9  14125.584        3.75333  0.156399   23.9984 2.88983e-127 1.62775e-123      SAMHD1
ENSG00000196136.17  2710.217        3.24734  0.143001   22.7086 3.68488e-114 1.55668e-110    SERPINA3
ENSG00000152583.12   974.737        4.50460  0.199169   22.6170 2.94551e-113 9.95466e-110     SPARCL1
ENSG00000211445.11 12512.792        3.77212  0.167464   22.5250 2.36246e-112 6.65348e-109        GPX3
                      entrez
                 <character>
ENSG00000189221.9         4128
ENSG00000120129.5         1843
ENSG00000101347.9        25939
ENSG00000196136.17          12
ENSG00000152583.12        8404
ENSG00000211445.11        2878
```

2.7 RNA-seq pipeline with R

7.1 Exporting results

You can easily save the results table in a CSV file that you can then share or load with a spreadsheet program such as Excel. The call to *as.data.frame* is necessary to convert the *DataFrame* object (*IRanges* package) to a *data.frame* object that can be processed by *write.csv*. Here, we take just the top 100 genes for demonstration.

```
resOrderedDF <- as.data.frame(resOrdered)[1:100, ]
write.csv(resOrderedDF, file = "results.csv")
```

2.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

A1

	A	B	C	D	E	F	G	H	I
1		baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol	entrez
2	ENSG00000189221.9	2373.80530666857	3.39828300997249	0.136171802447729	24.9558495142704	1.84493930984867e-137	3.11757844578229e-133	MAOA	4128
3	ENSG00000120129.5	3420.72722668767	2.97393572235113	0.120299687463265	24.7210594230284	6.35042375601189e-135	5.36547303145445e-131	DUSP1	1843
4	ENSG00000101347.9	14125.5841210142	3.75332984242088	0.156399170199782	23.9983999763326	2.8898330027183e-127	1.62774660266446e-123	SAMHD1	25939
5	ENSG00000196136.17	2710.21742603852	3.2473447475938	0.143000730118923	22.7085885847801	3.68488166639805e-114	1.55667825996985e-110	SERPINA3	12
6	ENSG00000152583.12	974.737366684742	4.50460352399075	0.199168515073065	22.6170462853439	2.94551443720581e-113	9.95466059198077e-110	SPARCL1	8404
7	ENSG00000211445.11	12512.7915263227	3.77212369837512	0.167463991514841	22.5249838144508	2.36246306452666e-112	6.65348347739525e-109	GPX3	2878
8	ENSG00000162614.18	5511.42886062423	2.011431976101	0.0916515987321489	21.946501795123	9.35274410616913e-107	2.25775242722923e-103	NEXN	91624
9	ENSG00000157214.13	3032.09554258641	2.01585616882548	0.0932194832636305	21.6248374079109	1.04886341857547e-103	2.21546175588604e-100	STEAP2	261729
10	ENSG00000154734.14	29608.2224675216	2.24074279330348	0.109100635375382	20.538311125263	9.79083450972645e-94	1.8382835727262e-90	ADAMTS1	9510
11	ENSG00000125148.6	3691.40735646806	2.23631771684738	0.109076549821485	20.5022777169551	2.05448558846141e-93	3.47166974738209e-90	MT2A	4502
12	ENSG00000109906.13	438.193981971508	6.37749569518999	0.313810468763127	20.3227627182951	8.08933590243127e-92	1.24266907344803e-88	ZBTB16	7704
13	ENSG00000134243.11	5543.59360082063	2.20367729753381	0.113059010050248	19.4913903505295	1.29903234899347e-84	1.8292540527743e-81	SORT1	6272
14	ENSG00000139132.14	1215.52553921277	2.15357770223809	0.111944039496327	19.2379845494923	1.77995994966619e-82	2.3136740945738e-79	FGD4	121512
15	ENSG00000127954.12	493.200428518741	4.74668284740483	0.249950406448032	18.9904986147391	2.04378727860063e-80	2.46685124527096e-77	STEAP4	79689
16	ENSG00000178695.5	2646.11820131036	-2.53174047028604	0.136781570320757	-18.5093683626312	1.73523393416814e-76	1.95479886797154e-73	KCTD12	115207
17	ENSG00000106484.15	942.144729137826	-2.03422440717542	0.112373164195998	-18.1024039122667	3.05058486294693e-73	3.22179893837983e-70	MEST	4232
18	ENSG00000162692.11	504.943122753552	-3.67370446529305	0.204815511891146	-17.9366515327487	6.10232947617832e-72	6.06571549932125e-69	VCAM1	7412
19	ENSG00000166741.7	7493.99425894884	2.24432150615166	0.127527776514408	17.5986876545135	2.52098328723491e-69	2.36664308820531e-66	NNMT	4837
20	ENSG00000145390.11	8699.60480031948	2.29560371933772	0.132352454095727	17.3446252660897	2.16585998229454e-67	1.92624747267437e-64	USP53	54532