Vall d'Hebron
Institut de Recerca

INTRODUCCIÓ A L' ANÀLISI DE DADES ÒMIQUES
RNA-seq

Bioinformàtica per a la Recerca Biomèdica

Ricardo Gonzalo Sanz
ricardo.gonzalo@vhir.org

28/11/2018

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

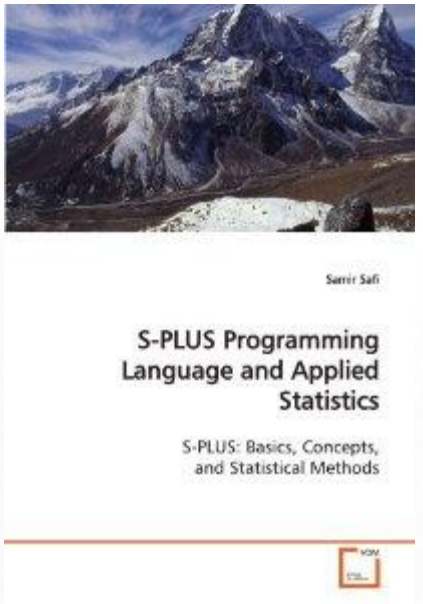
1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)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

Vall d'Hebron
Institut de Recerca

1. **Introduction to R and Bioconductor**
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. 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 is R



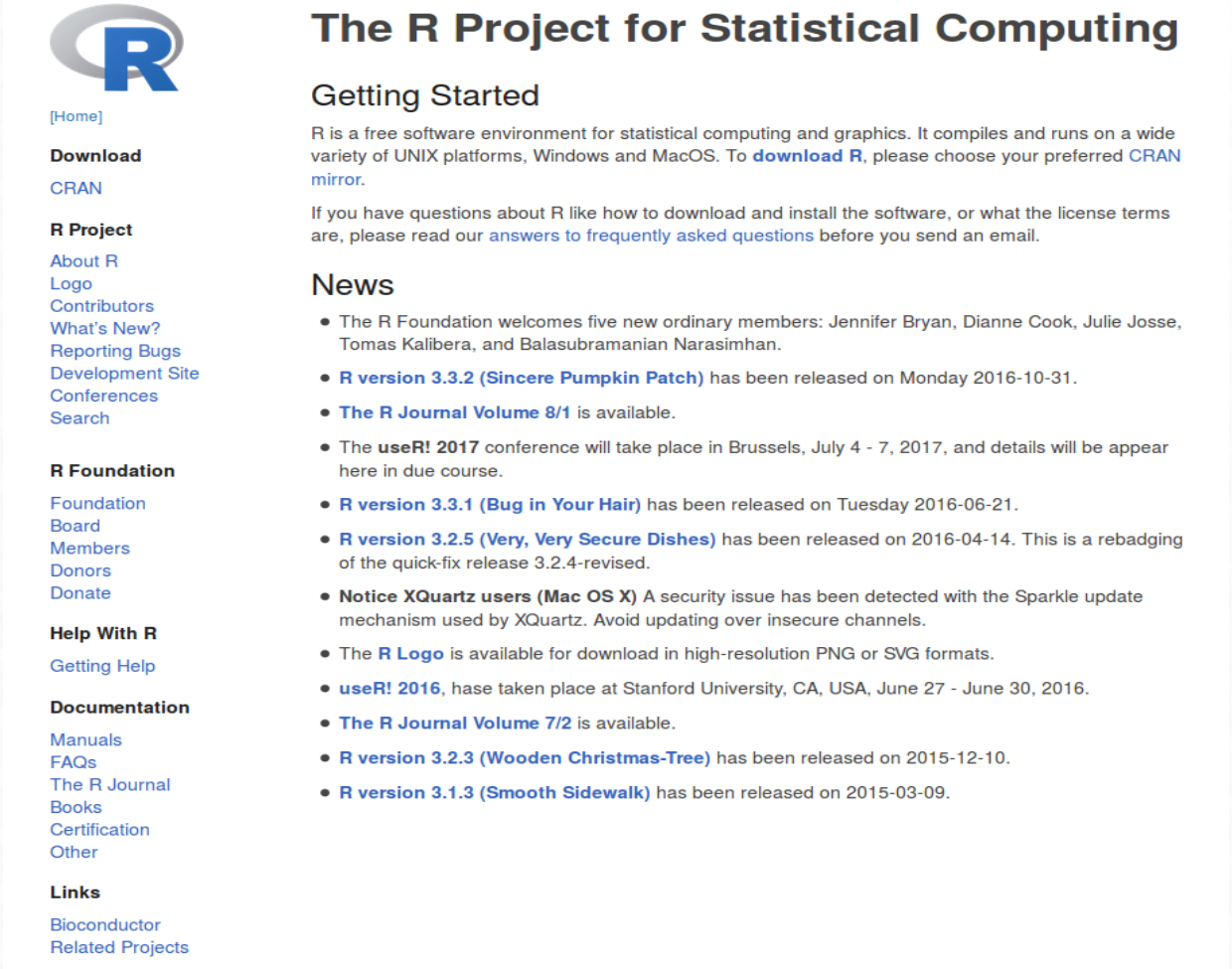
- The S language was developed in 1976 at Bell Laboratories by John Chambers to ...
  - facilitate interactive exploration and visualization of data of varying complexity.
  - allow them to perform on all types of statistical analyzes.

- S language was (and is) commercial.

- R ("GNU" S) is born as a free alternative to S

1. Introduction to R and Bioconductor

Vall d'Hebron
Institut de Recerca



[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting Bugs
Development Site
Conferences
Search

R Foundation

Foundation
Board
Members
Donors
Donate

Help With R

Getting Help

Documentation

Manuals
FAQs
The R Journal
Books
Certification
Other

Links

Bioconductor
Related Projects

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- The R Foundation welcomes five new ordinary members: Jennifer Bryan, Dianne Cook, Julie Josse, Tomas Kalibera, and Balasubramanian Narasimhan.
- **R version 3.3.2 (Sincere Pumpkin Patch)** has been released on Monday 2016-10-31.
- **The R Journal Volume 8/1** is available.
- The **useR! 2017** conference will take place in Brussels, July 4 - 7, 2017, and details will be appear here in due course.
- **R version 3.3.1 (Bug in Your Hair)** has been released on Tuesday 2016-06-21.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016**, hase taken place at Stanford University, CA, USA, June 27 - June 30, 2016.
- **The R Journal Volume 7/2** is available.
- **R version 3.2.3 (Wooden Christmas-Tree)** has been released on 2015-12-10.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.

https://cran.r-project.org/

Vall d'Hebron
Institut de Recerca

Why R?

- Free
- High quality methods implemented
- Platform independent
  - Linux, Mac, windows
- Constantly evolving
  - New version /6 months
- Programming language
  - Powerful & Flexible
  - Open source
  - Great for repetitive tasks

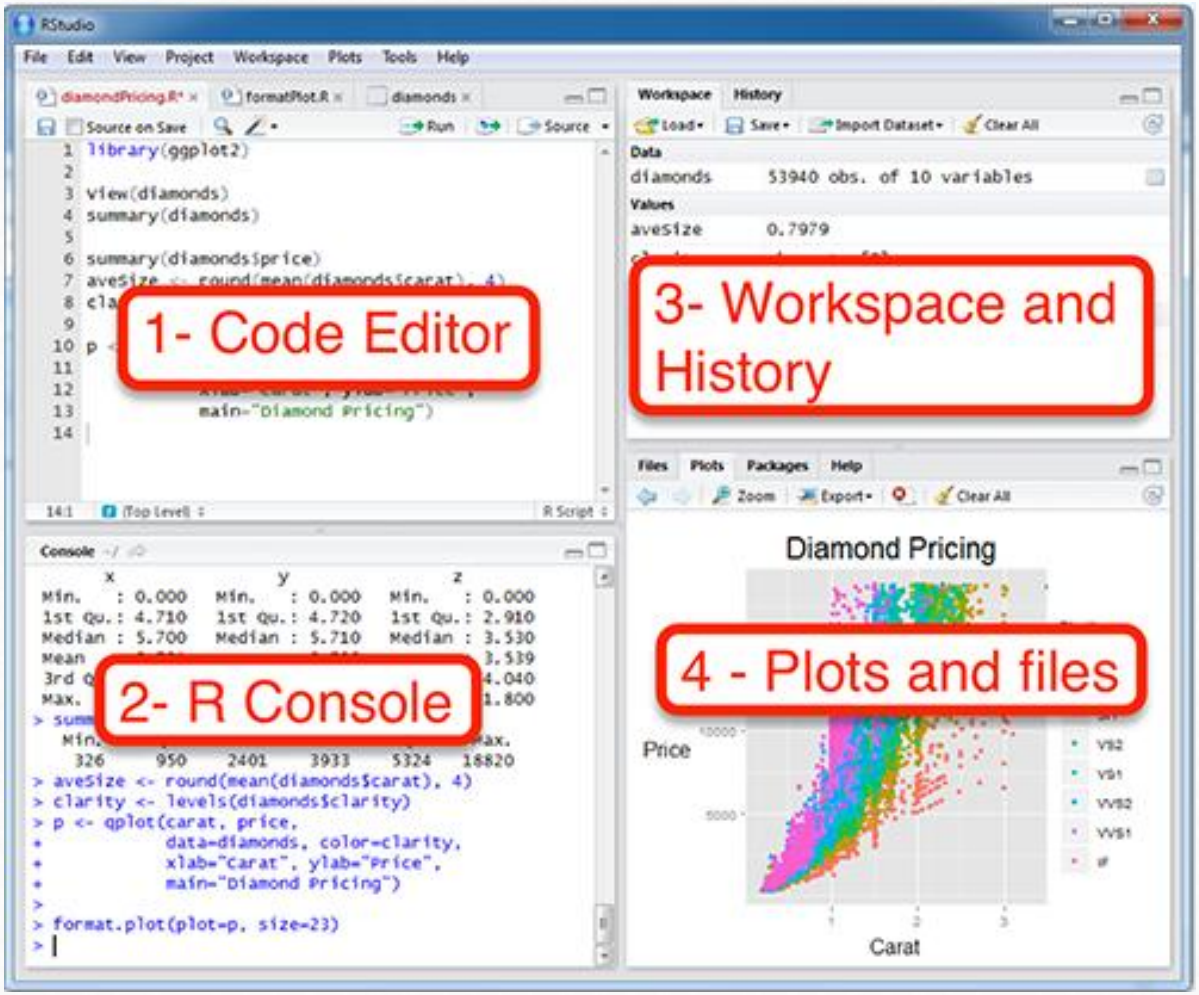
- Statistical tool
  - Modern
  - Most existing methods
    - (new method in R)
  - Great graphics.

Why not R?

- Console-based interface
  - But GUI projects available
    - R-commander, DeduceR
- Community-based quality control
  - No company behind (no money back)
  - But thousands of users for most packages
- Constantly evolving
  - One new version every 6 months

Vall d'Hebron
Institut de Recerca

R interfaces - Rstudio



RStudio®

https://www.rstudio.com/

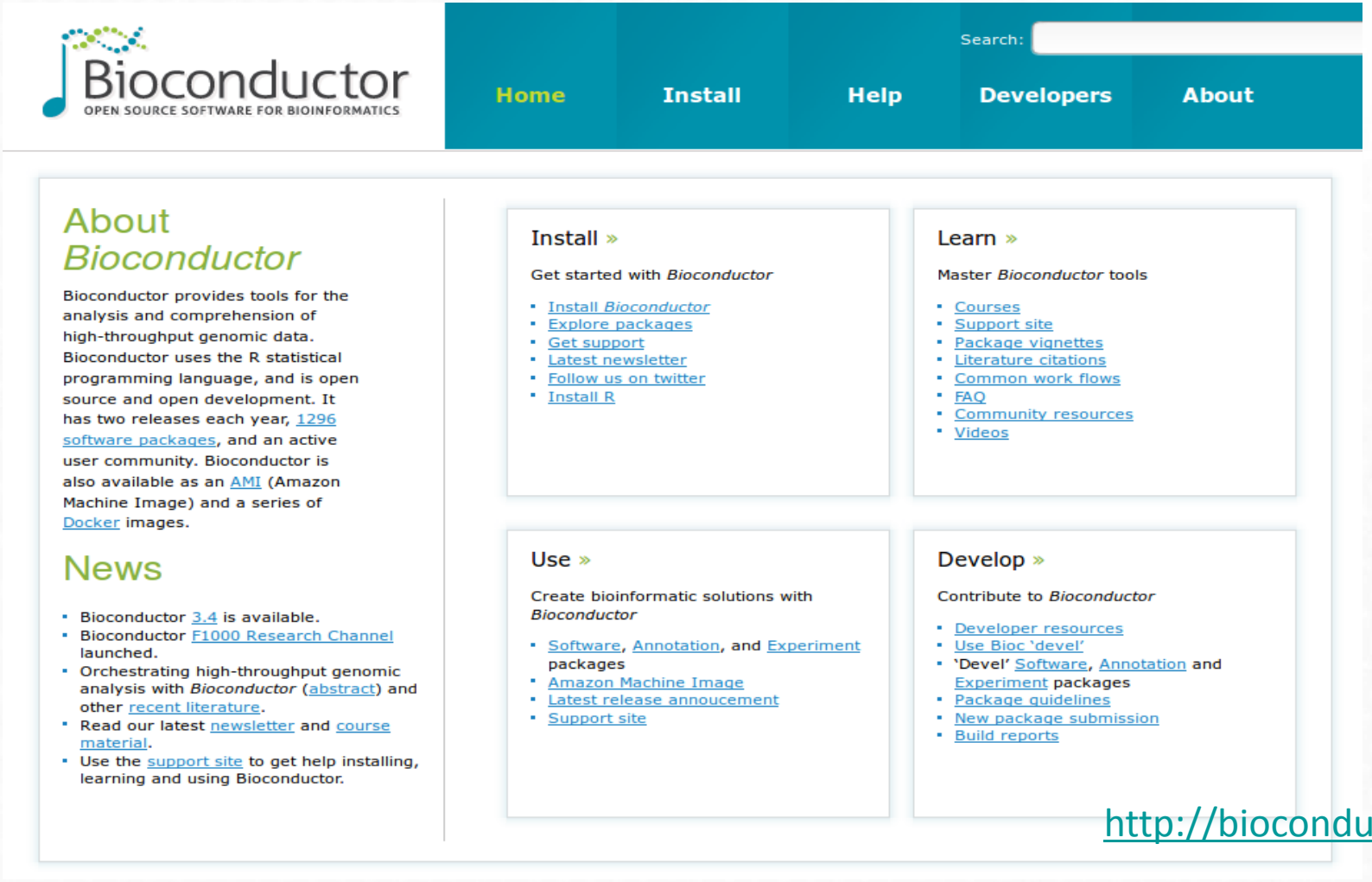
Vall d'Hebron
Institut de Recerca

BIOCONDUCTOR

- An open source and open development software project for the analysis and comprehension of genomic data.
- Started in 2001. The core team is based primarily at the *Fred Hutchinson Cancer Research Center.*
- Primarily based on the R programming language.
- There are two releases of Bioconductor every year.
    - Started with 15packages
    - Now there are more than 1000

Vall d'Hebron
Institut de Recerca

BIOCONDUCTOR



http://bioconductor.org/

Vall d'Hebron
Institut de Recerca

BIOCONDUCTOR

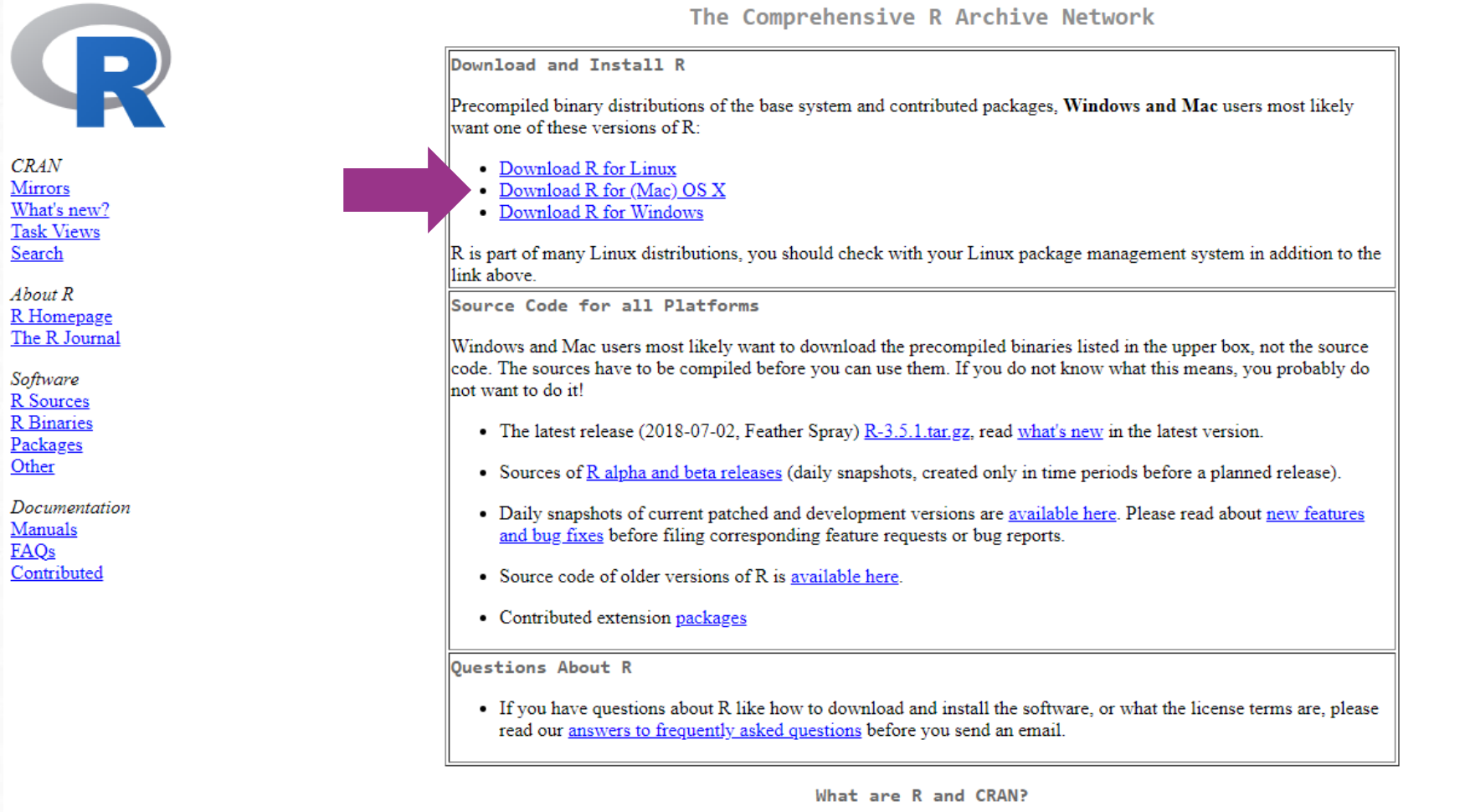
- Essentially Bioconductor is a set of R packages
- A bioconductor package
  - Implements a different, new functionality
    - To manipulate or make tests on omics data
    - To use annotations
    - …
  - It can also be an Annotations database
  - Or even an experimental dataset
- **BioConductor also provides training materials**

Vall d'Hebron
Institut de Recerca

1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. 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. Installation of R and Bioconductor

https://cran.r-project.org/



2. Installation of R and Bioconductor

Vall d'Hebron
Institut de Recerca

R for Windows

Subdirectories:

base	Binaries for base distribution. This is what you want to **install R for the first time**.
contrib	Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

2. Installation of R and Bioconductor

Vall d'Hebron
Institut de Recerca

Subdirectories:

base

contrib

old contrib

Rtools

Please do not submit bin
Windows binaries.

You may also want to rea

Note: CRAN does some

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.6 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.7.1) here. Releases for old Mac OS X systems (through Mac OS X 10.5) and PowerPC Macs can be found in the old directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

As of 2016/03/01 package binaries for R versions older than 2.12.0 are only available from the CRAN archive so users of such versions should adjust the CRAN mirror setting accordingly.

R 3.5.1 "Feather Spray" released on 2018/07/05

Important: since R 3.4.0 release we are now providing binaries for OS X 10.11 (El Capitan) and higher using non-Apple toolkit to provide support for OpenMP and C++17 standard features. To compile packages you may have to download tools from the tools directory and read the corresponding note below.

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type

```
md5 R-3.5.1.pkg
```

in the *Terminal* application to print the MD5 checksum for the R-3.5.1.pkg image. On Mac OS X 10.7 and later you can also validate the signature using

```
pkgutil --check-signature R-3.5.1.pkg
```

Lastest release:

R-3.5.1.pkg MD5-hash: 58eaff65fbd024f267ef1e521e17e7f8 SHA1-hash: 76c01bfa62a6896d5f4a4511e25d17276d149621 (ca. 74MB)	**R 3.5.1** binary for OS X 10.11 (El Capitan) and higher, signed package. Contains R 3.5.1 framework, R.app GUI 1.70 in 64-bit for Intel Macs, Tcl/Tk 8.6.6 X11 libraries and Texinfo 5.2. The latter two components are optional and can be ommitted when choosing "custom install", they are only needed if you want to use the `tcltk` R package or build package documentation from sources. Note: the use of X11 (including `tcltk`) requires XQuartz to be installed since it is no longer part of OS X. Always re-install XQuartz when upgrading your macOS to a new major version. **Important:** this release uses Clang 6.0.0 and GNU Fortran 6.1, neither of which is supplied by Apple. If you wish to compile R packages from sources, you will need to download and install those tools - see the tools directory.
NEWS (for Mac GUI)	News features and changes in the R.app Mac GUI
Mac-GUI-1.70.tar.gz MD5-hash: b1ef5f28552464068 0a22965bb8800f8	Sources for the R.app GUI 1.70 for Mac OS X. This file is only needed if you want to join the development of the GUI, it is not intended for regular users. Read the INSTALL file for further instructions.

2. Installation of R and Bioconductor

Vall d'Hebron
Institut de Recerca

Subdirectories:

base

contrib

old contrib

Rtools

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.6 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.7.1) here. Releases for old Mac OS X systems (through Mac OS X 10.5) and PowerPC Macs can be found in the old directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

As of 2016/03/01 package binaries for R versions older than 2.12.0 are only available from the CRAN archive so users of such versions should adjust the CRAN mirror setting accordingly.

R 3.5.1 "Feather Spray" released on 2018/07/05

Important: since R 3.4.0 release we are now providing binaries for OS X 10.11 (El Capitan) and higher using non-Apple toolkit to provide support for OpenMP and C++17 standard features. To compile packages you may have to download tools from the tools directory and read the corresponding note

Please do not submit bin
Windows binaries.

You may also want to rea

Note: CRAN does some

rupted during the mirroring process. For

u can also validate the signature using

Contains R 3.5.1
braries and Texinfo
hoosing "custom
uild package

ince it is no longer part
ew major version.

f which is supplied by
download and install

f you want to join the
STALL file for further

In Linux is already installed

Index of /bin/linux

Name	Last modified	Size	Description
Parent Directory		-	
debian/	2018-07-03 11:18	-	
redhat/	2014-07-27 21:12	-	
suse/	2012-02-16 15:09	-	
ubuntu/	2018-11-27 04:07	-	

Apache/2.4.10 (Debian) Server at cran.r-project.org Port 443

2. Installation of R and Bioconductor

Vall d'Hebron
Institut de Recerca

Package installation (R):

```
install.packages("name of the package")
```

Package installation (Bioconductor):

```
install.packages("BiocManager")
BiocManager::install("name of the package", version = "X,X")
```

Load the packages

```
library(name of the package)
```

2. Installation of R and Bioconductor

Package installation (R):

install.packages("name of th

Packag

nager")

tall("name of the package", version = "X,X")

the packages

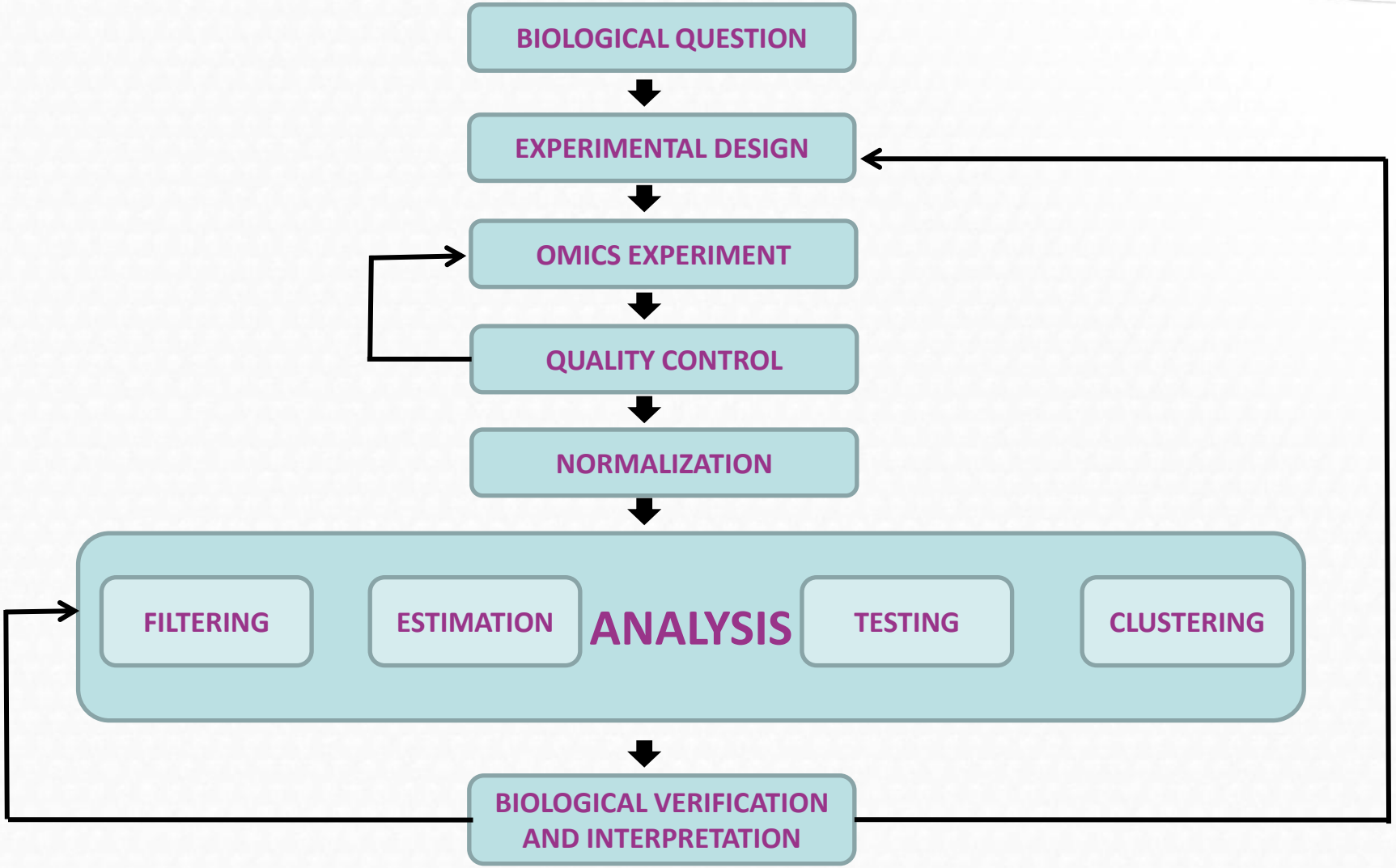
library(name of the package)

All the packages has been already installed previously!!!

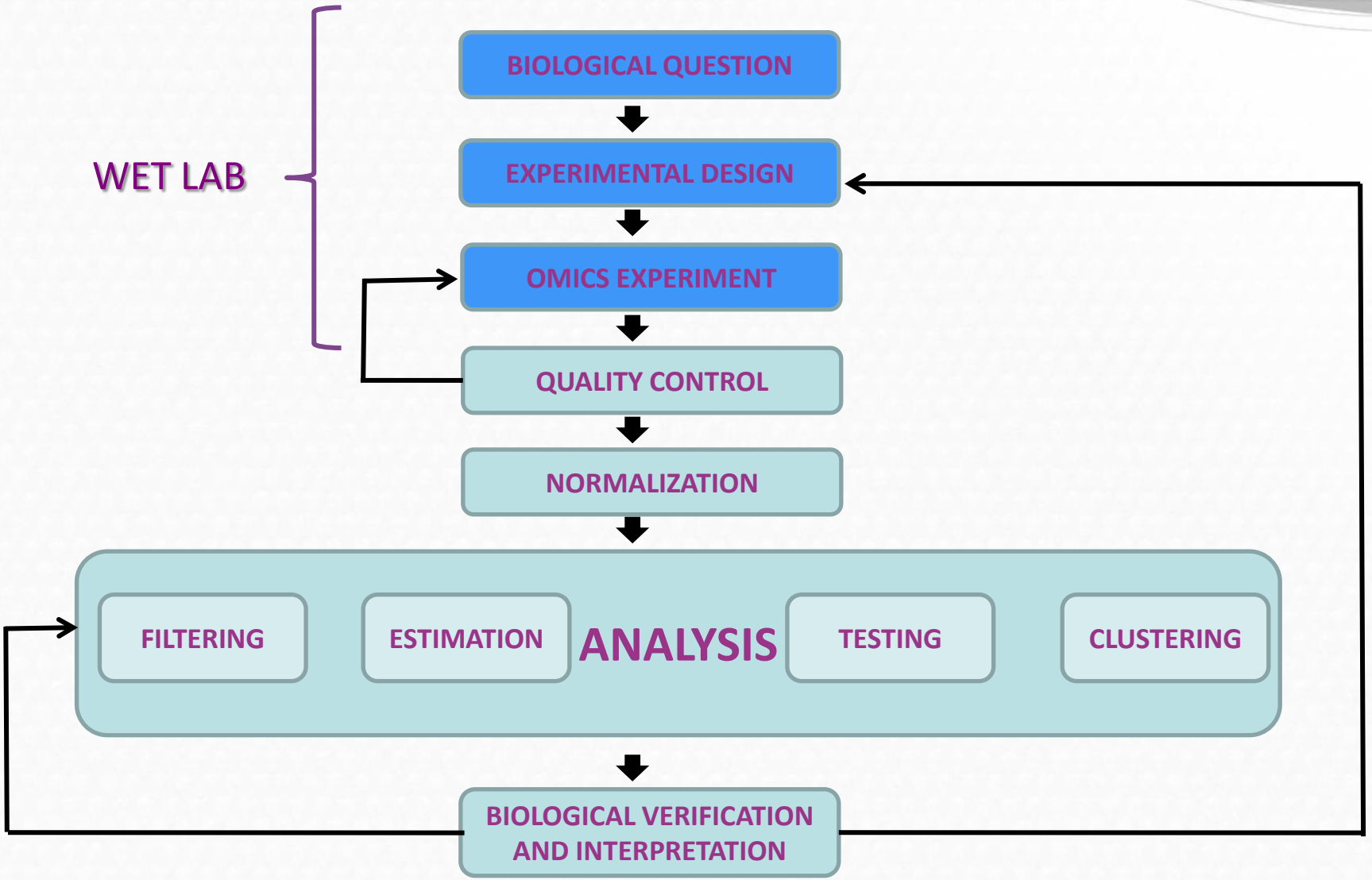
Vall d'Hebron
Institut de Recerca

1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. 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. Introduction to omic data analysis



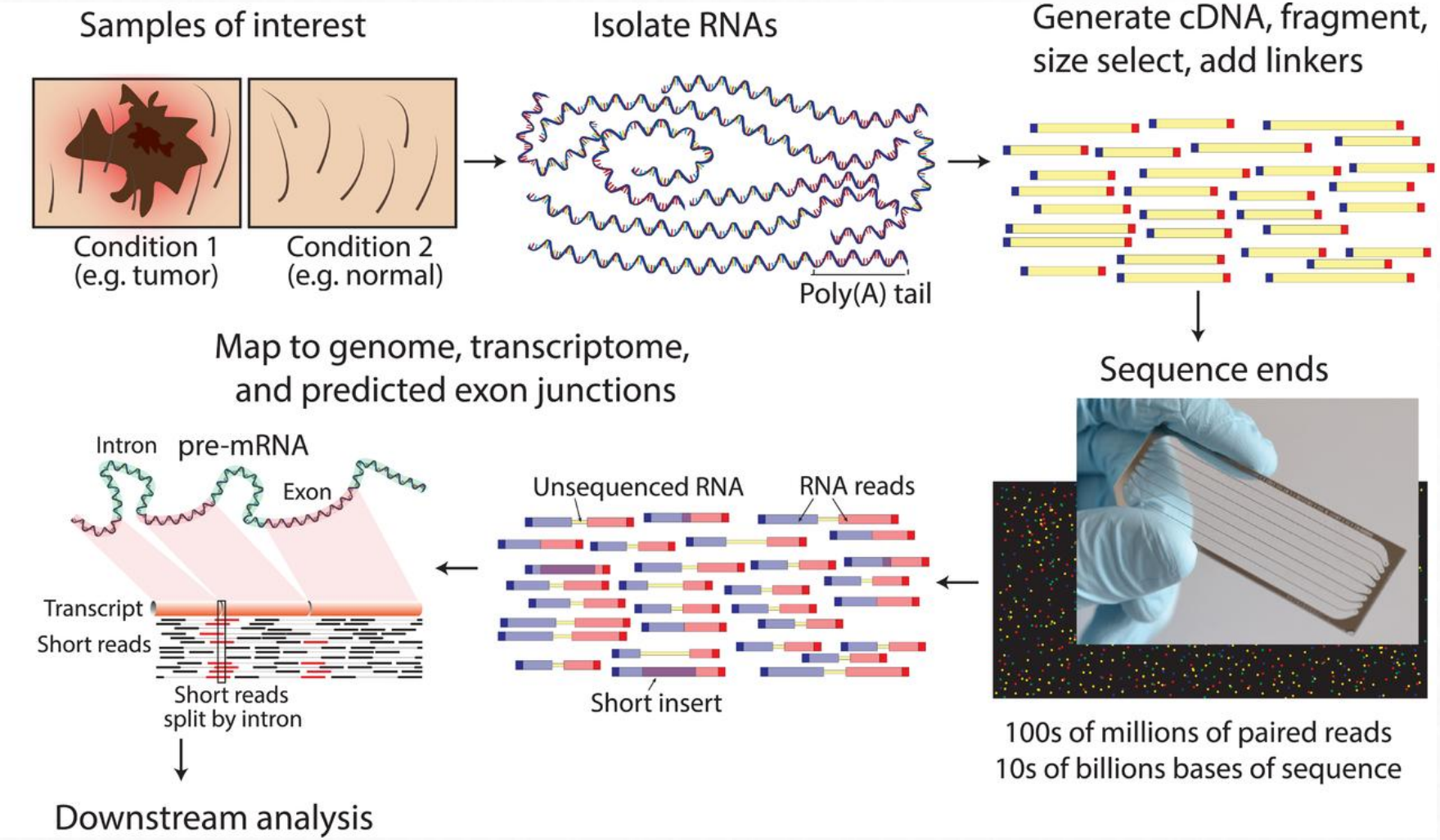
3. Introduction to omic data analysis



Vall d'Hebron
Institut de Recerca

1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. 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

4. An example of data analysis. RNA-seq



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

4.1 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*

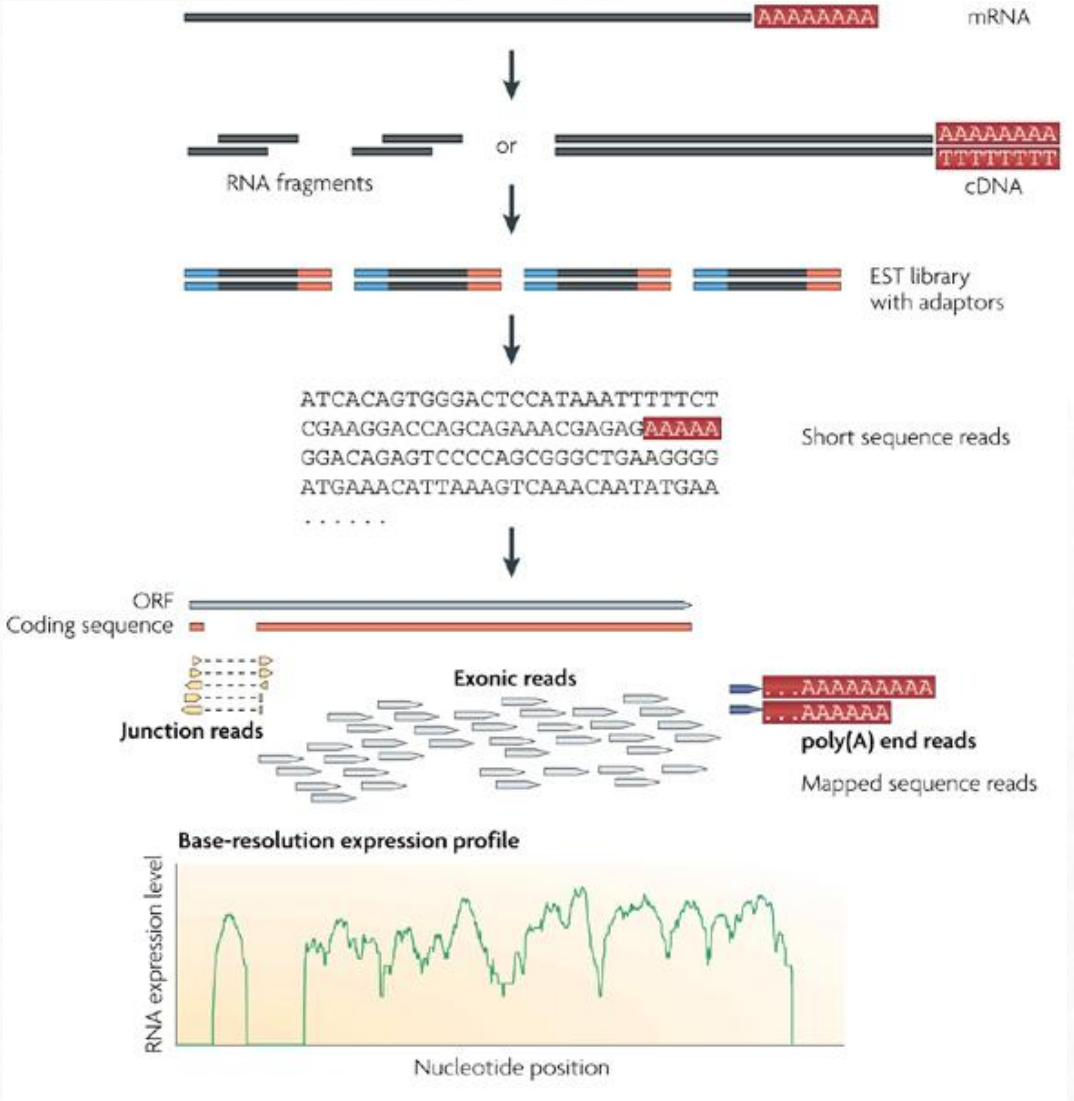
	Condition A			Condition B		
Gene1	4	0	2	12	14	13
Gene2	0	23	50	47	22	0
Gene3	0	2	6	13	11	15
...	...	...	...	...	...	...
GeneG	156	238	37	129	51	118

4.1 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)

4.1 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 R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. 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

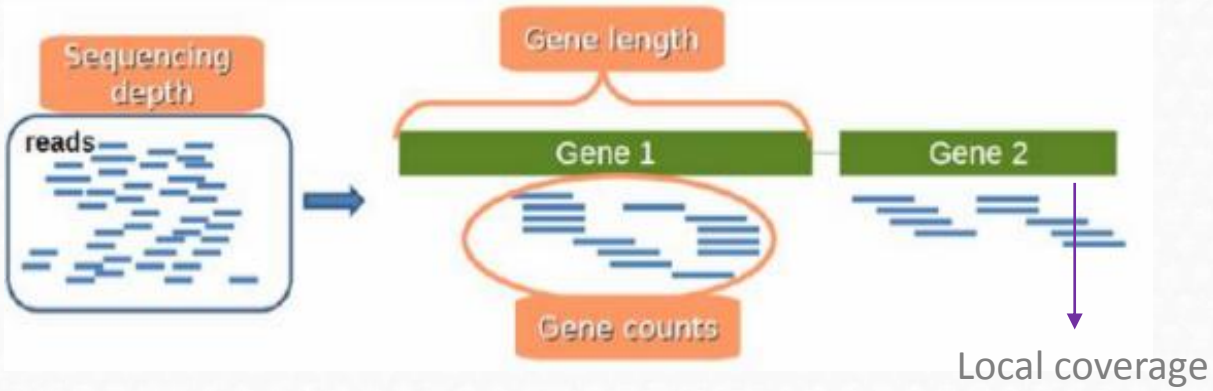
4.2 Basic key concepts?

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 R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. 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

4.3 Main challenges in RNA-seq

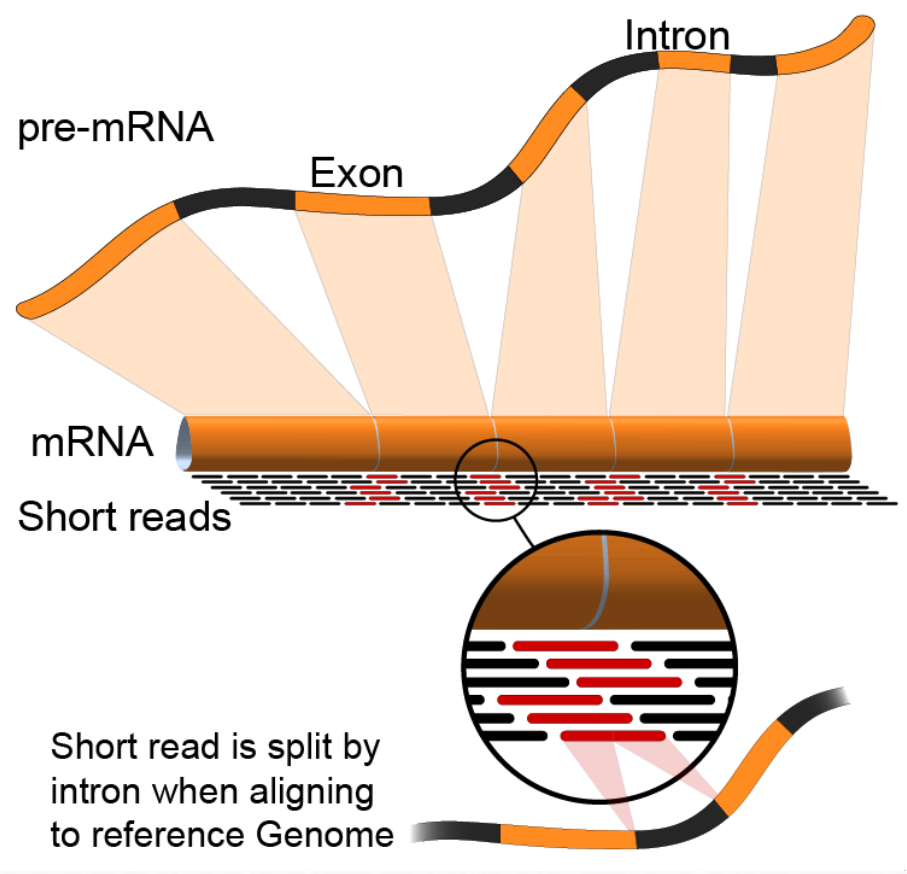
- **Sample**
  - Purity? Quantity? Quality?

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

4.3 Main challenges in RNA-seq

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

arge introns



4.3 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

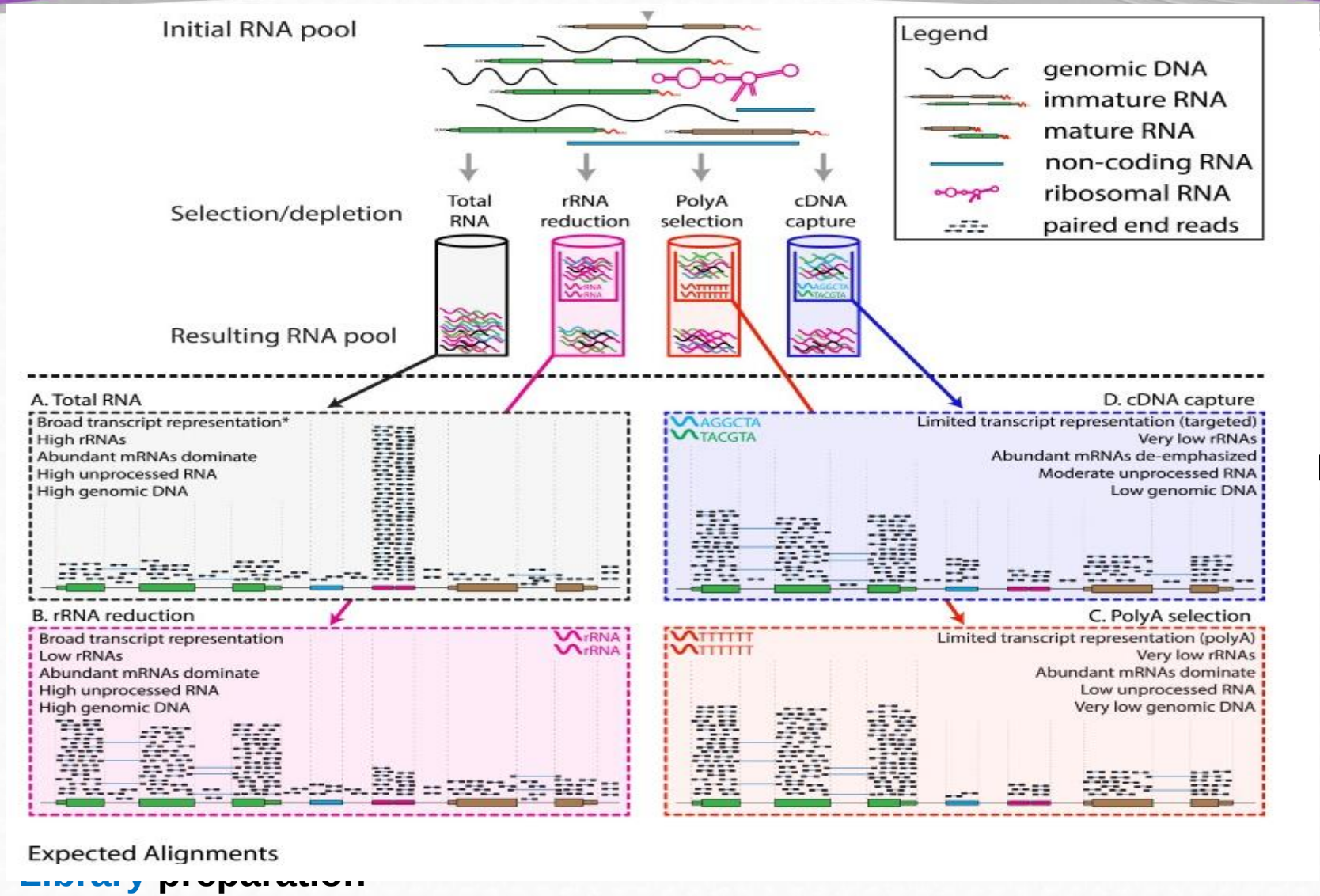
- **The relative abundance of RNAs vary wildly**
  - 10^5 – 10^7 orders of magnitude
  - Since RNA sequencing works by random sampling, a small fraction of highly expressed genes may consume the majority of reads (rRNA)

- **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**

4.3 Main challenges in RNA-seq



hly

Library preparation

4.3 Main challenges in RNA-seq (and other NGS cases)

Vall d'Hebron
Institut de Recerca

- 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 R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. 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

4.4 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

4.4 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.

4.4 RNA-seq vs Microarrays

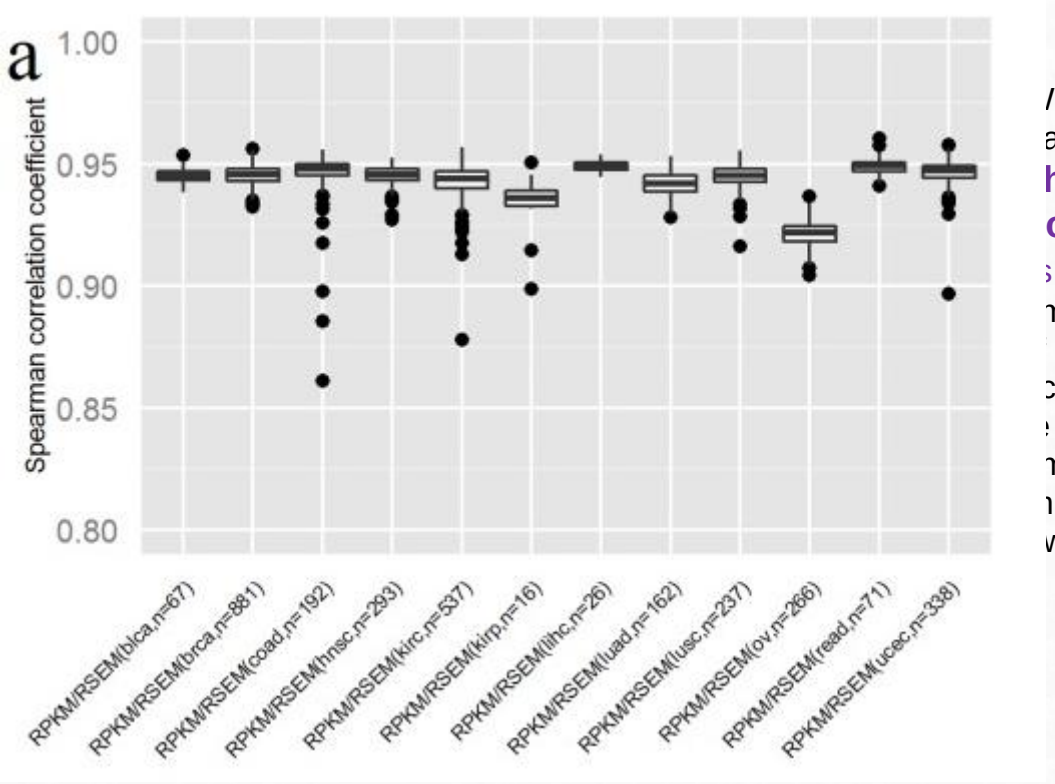
PLoS One. 2013 Aug 20;8(8):e71462. doi: 10.1371/journal.pone.0071462. eCollection 2013.

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

4.4 RNA-seq vs Microarrays

Vall d'Hebron
Institut de Recerca

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

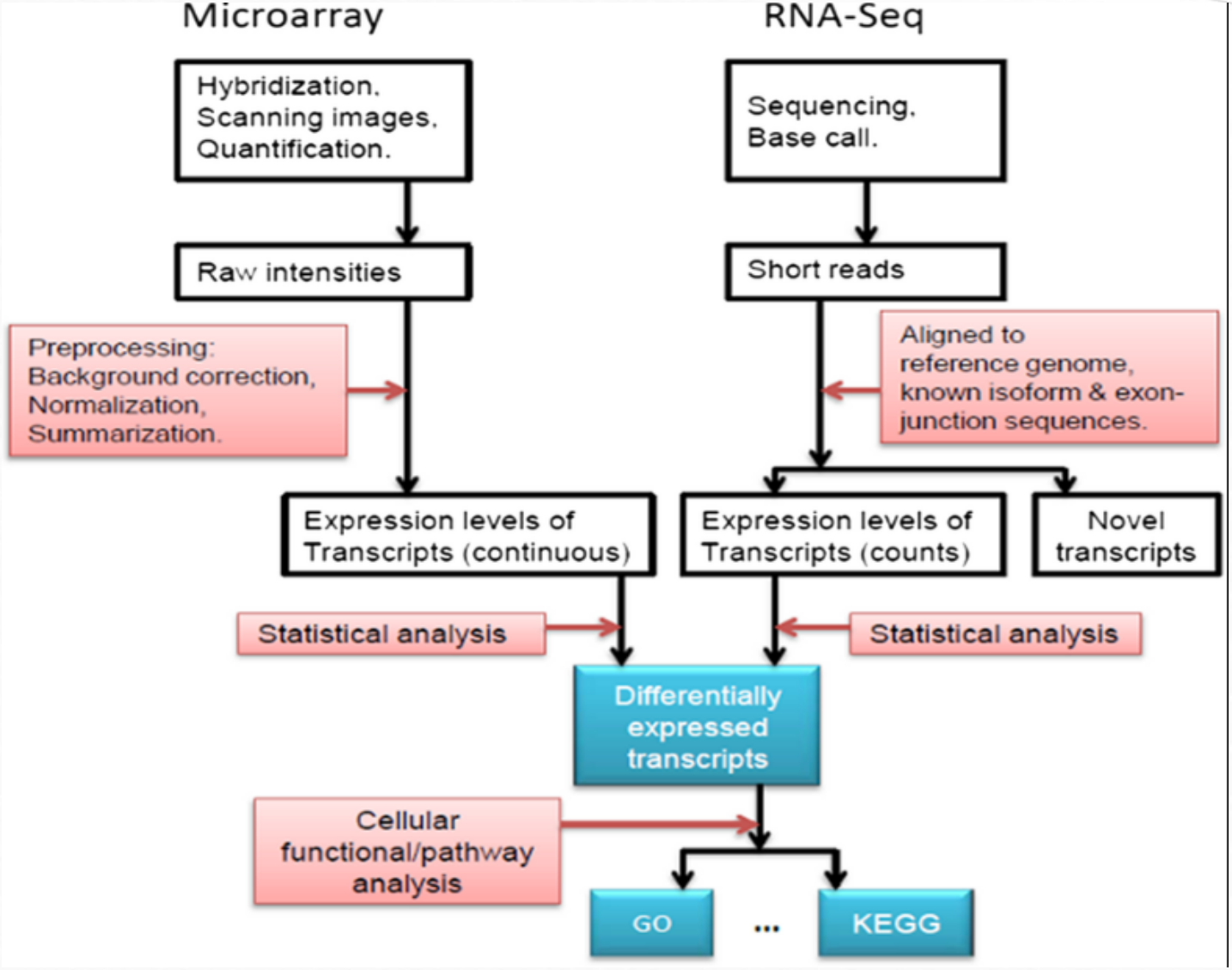
Shanrong Zhao, Wai-Ping Fung-Leung, Anton Bittner, Karen Ngo, Xuejun Liu

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

4.4 RNA-seq vs Microarrays



4.4 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."

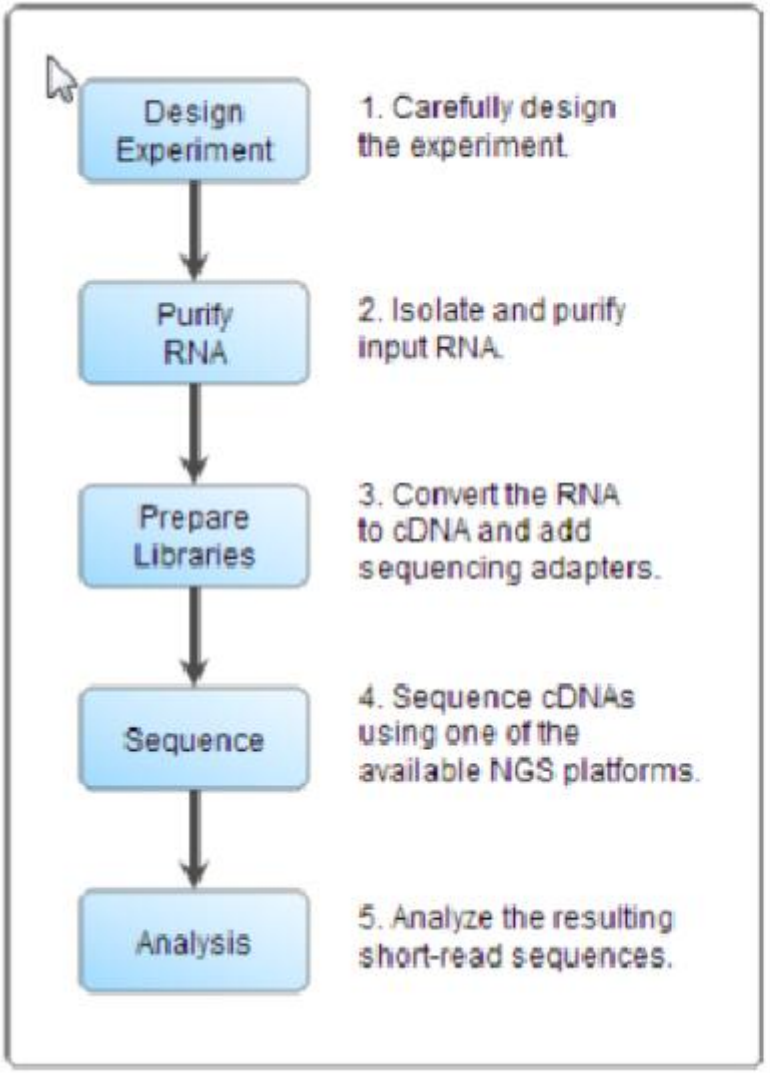
"RNA-Seq sequencing technology is new to most researchers, more expensive than microarray, **data storage is more challenging** and **analysis is more complex**."

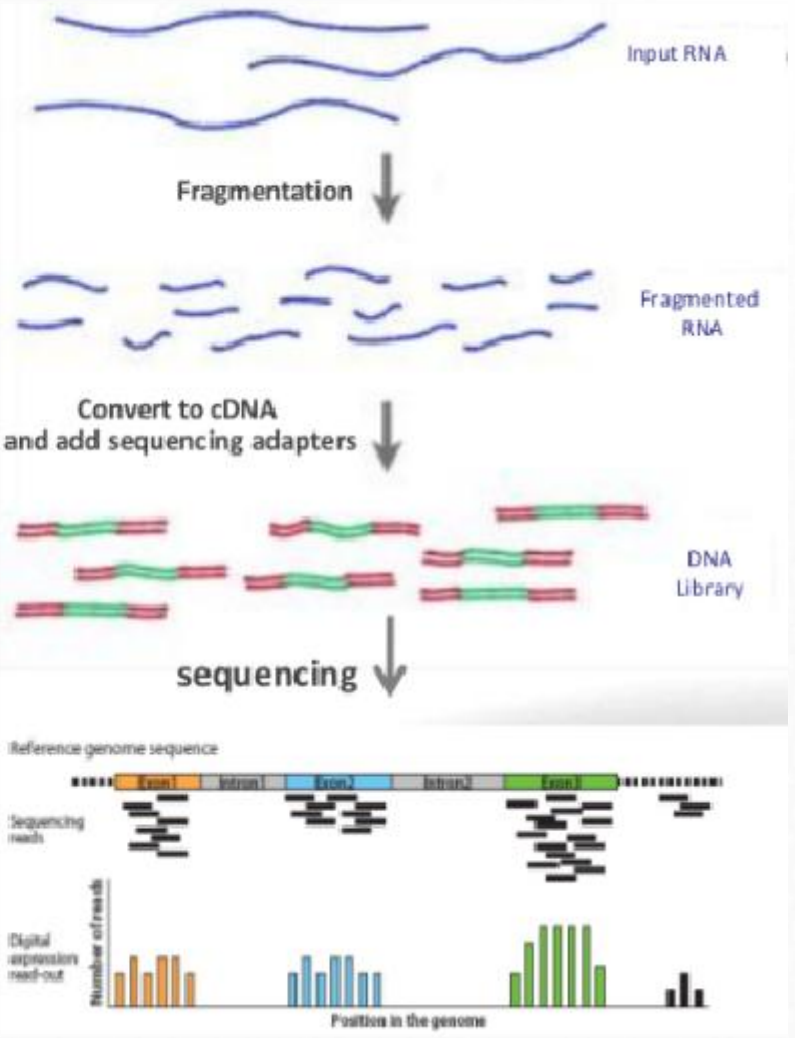
Zhao et al, 2014

Vall d'Hebron
Institut de Recerca

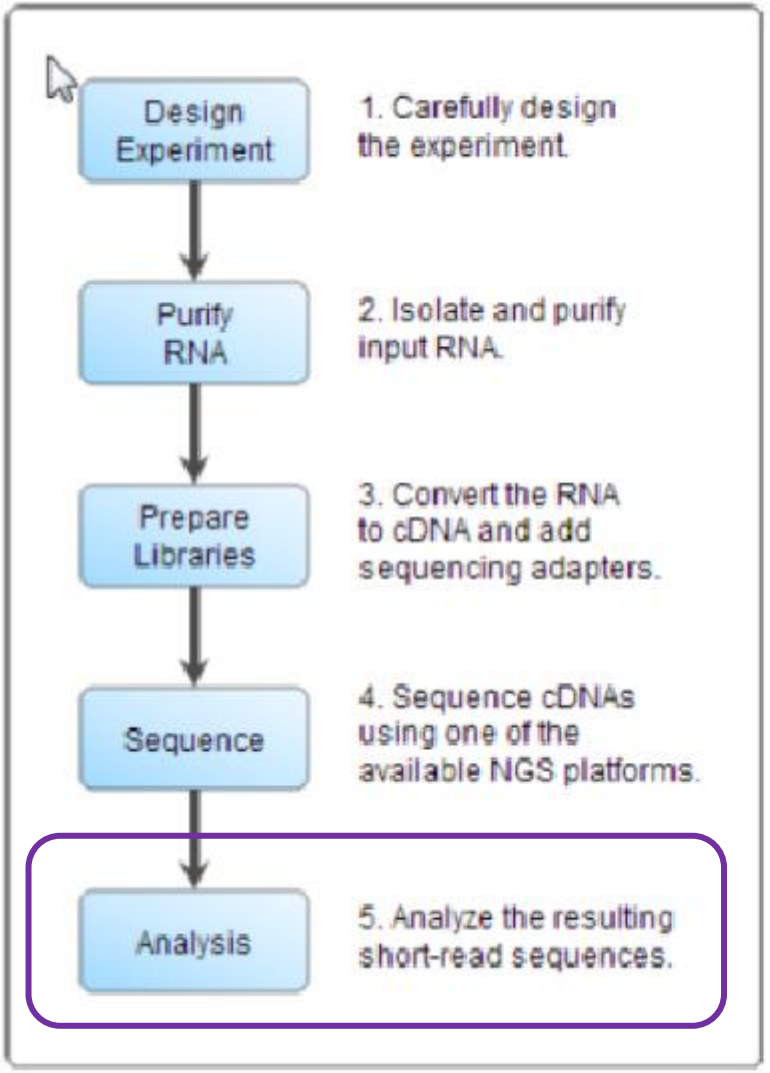
1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)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

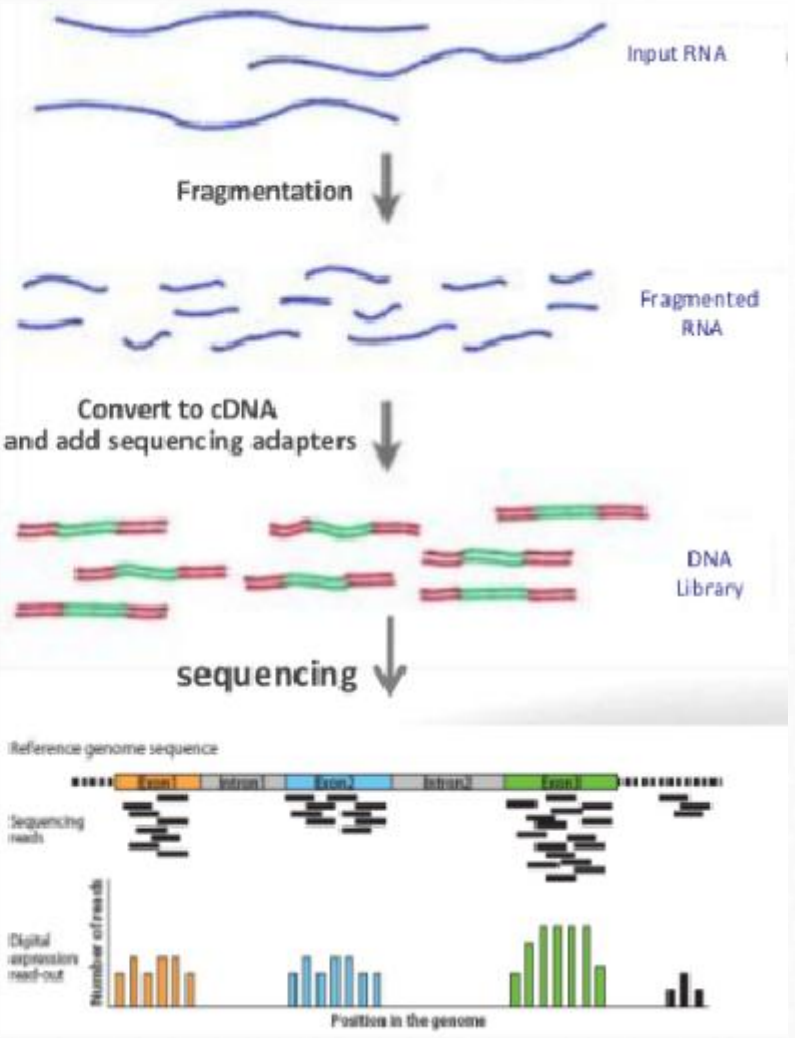
4.5 RNA-seq analysis pipeline(s)



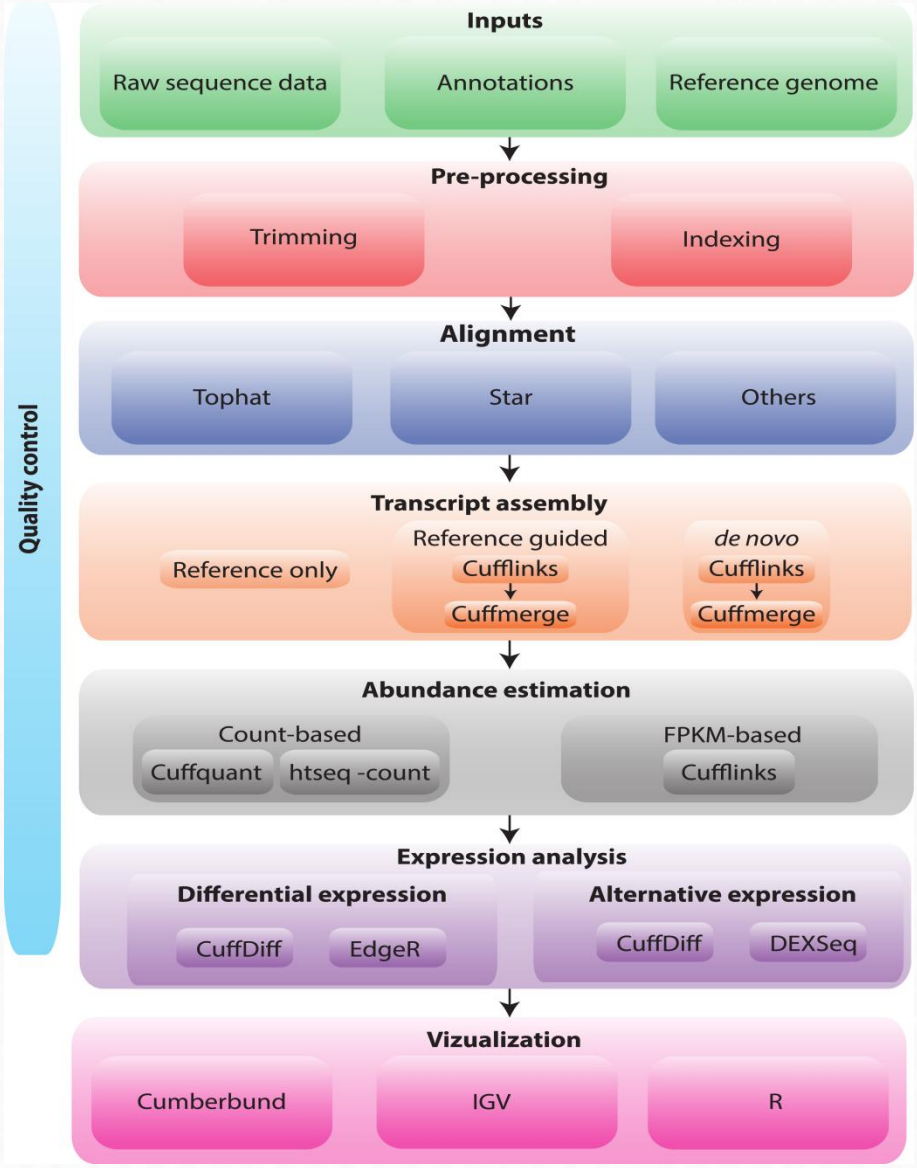


4.5 RNA-seq analysis pipeline(s)



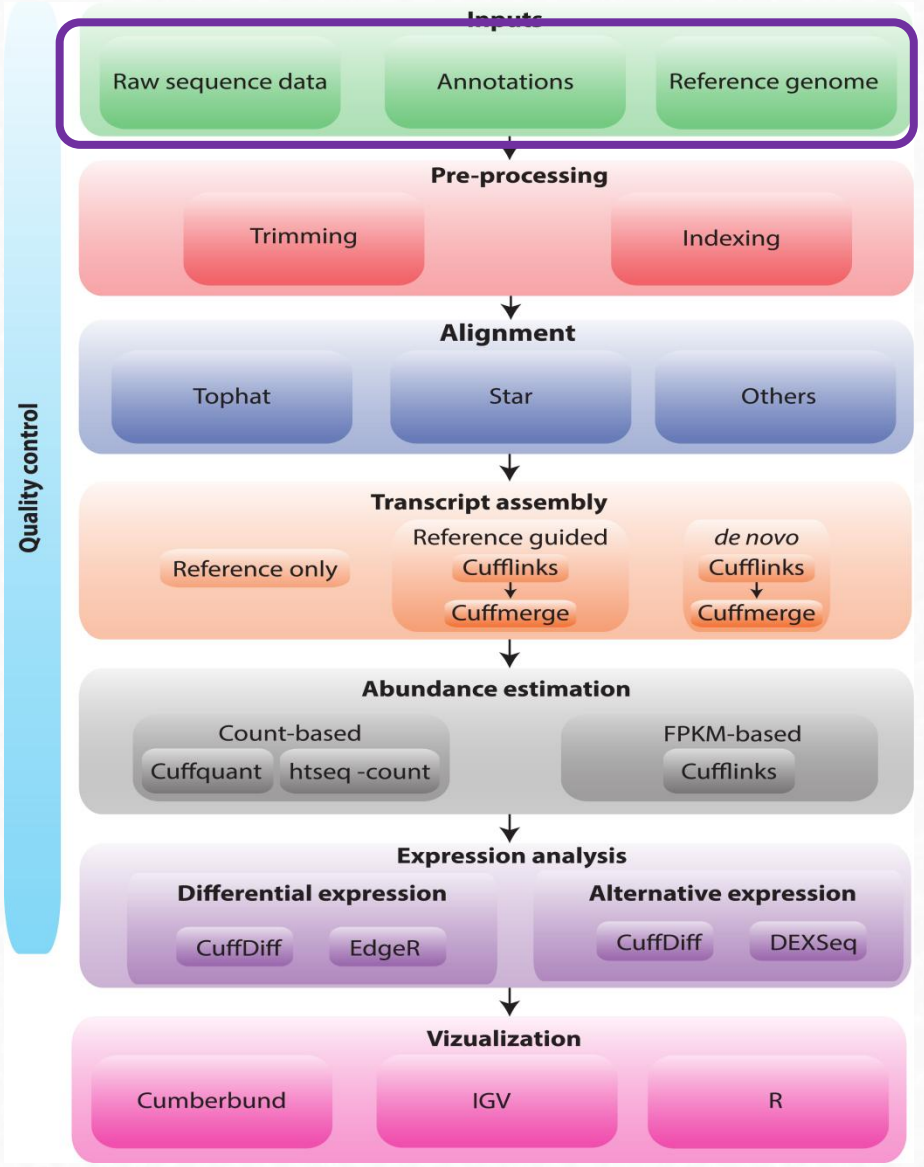


4.5 RNA-seq analysis pipeline(s)



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

4.5 RNA-seq analysis pipeline(s)

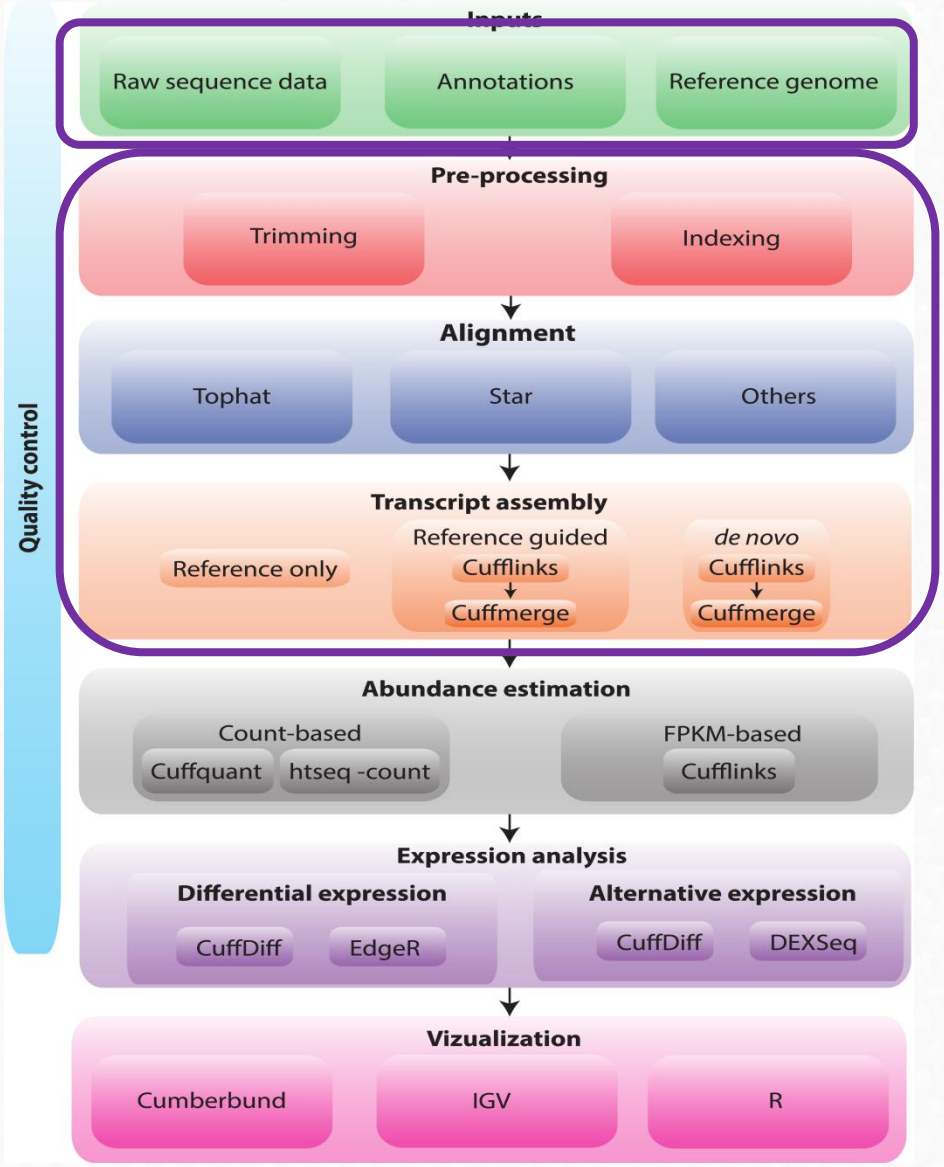


Fastq files

FASTQC (quality control)

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

4.5 RNA-seq analysis pipeline(s)

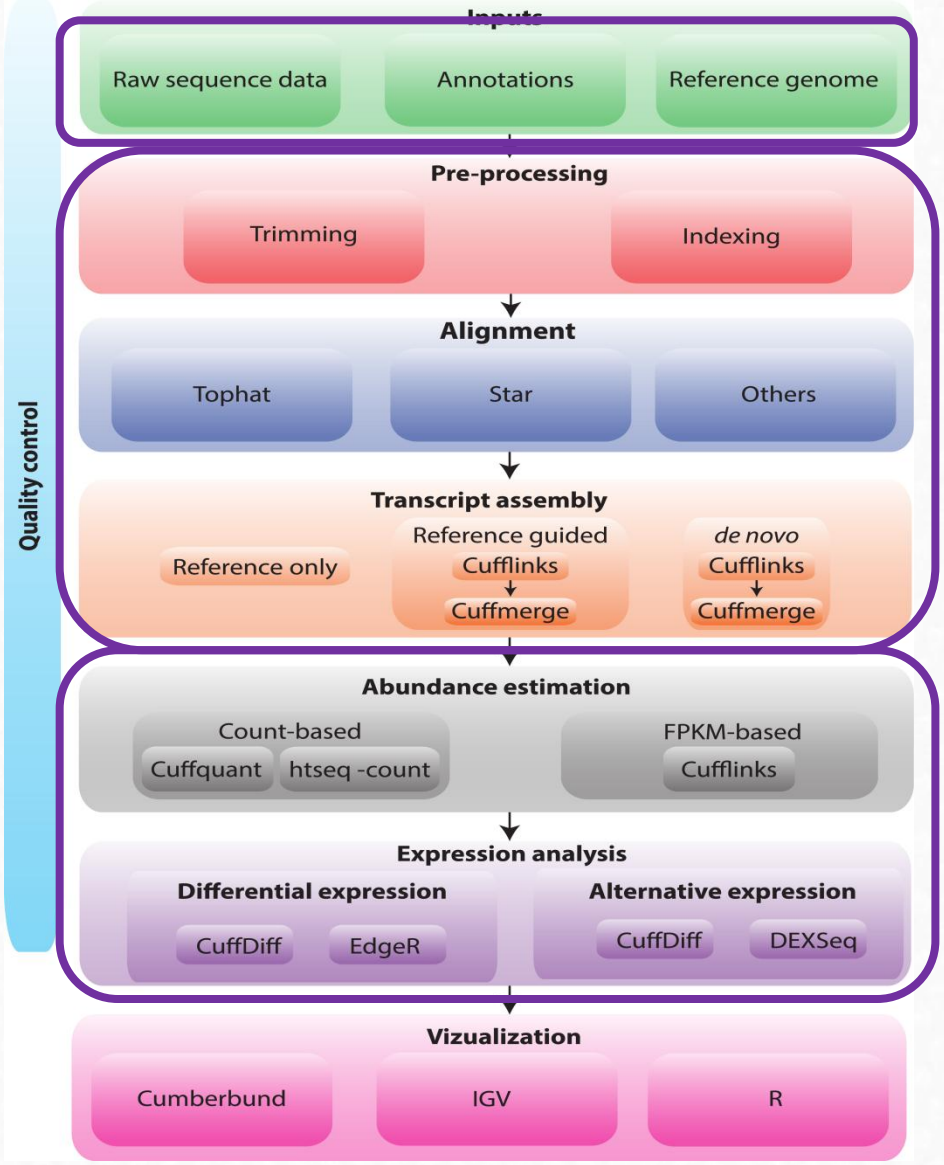


- Fastq files
- FASTQC (quality control)

- Alignment

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

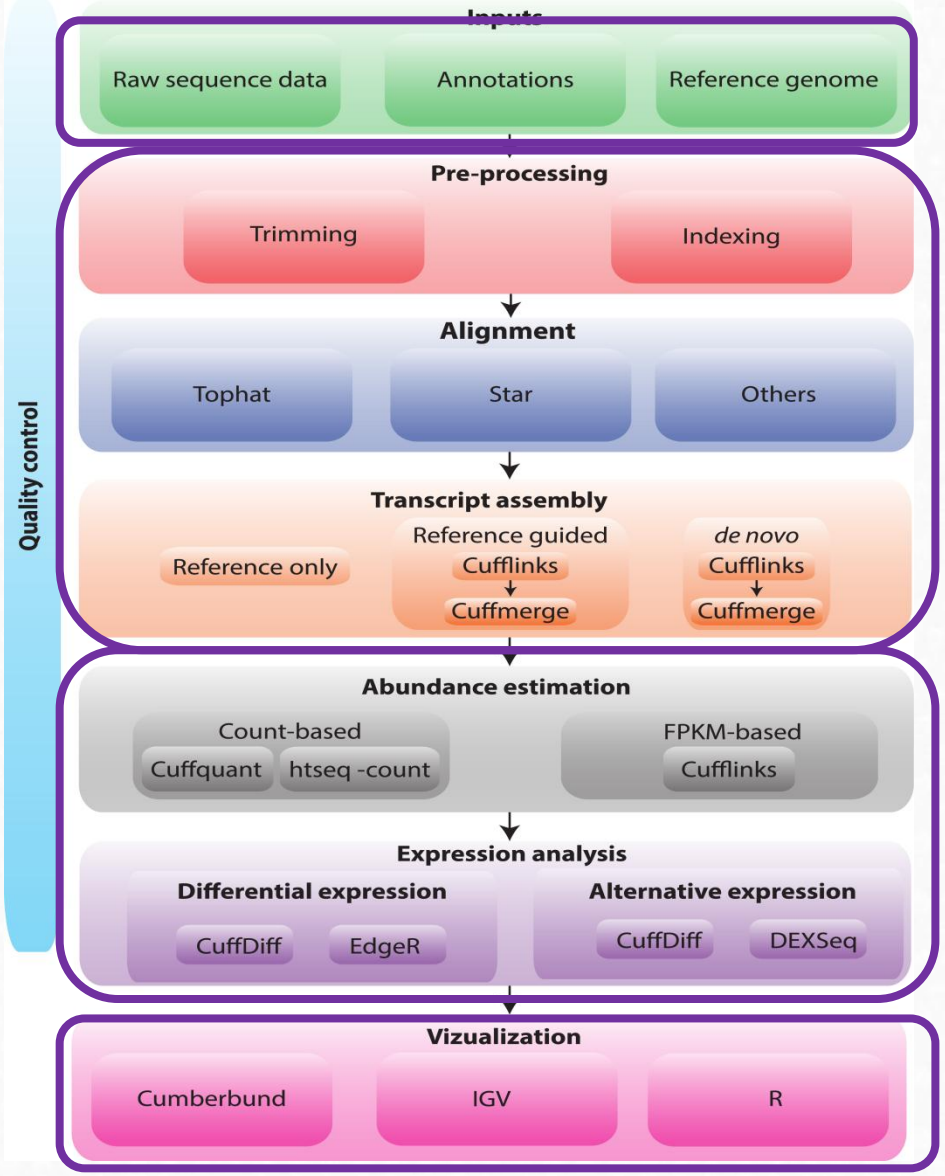
4.5 RNA-seq analysis pipeline(s)



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

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

4.5 RNA-seq analysis pipeline(s)



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

+ Analysis of biological significance

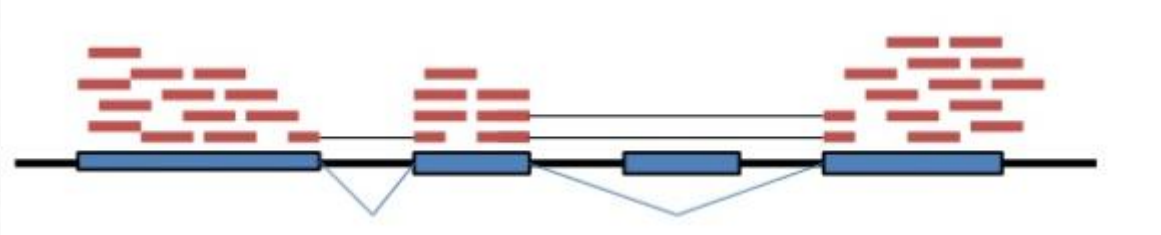
Griffith, 2015 DOI: 10.1371/journal.pcbi.1004393

Vall d'Hebron
Institut de Recerca

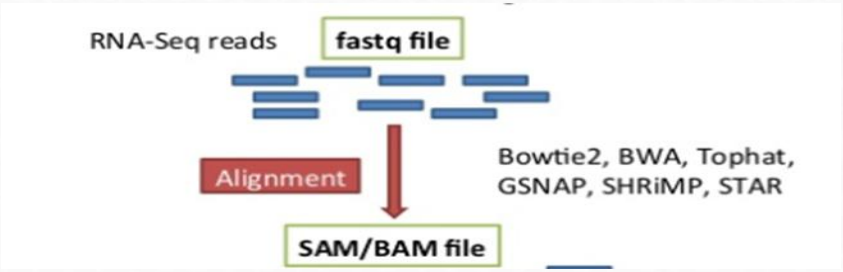
1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. An example of omics data analysis. RNA-seq
    1. What is RNA-seq
    2. Basic key concepts
    3. Main challenges in RNA-seq
    4. RNA-seq vs Microarrays
    5. RNA-seq analysis pipeline(s)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

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



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)

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

4.6 Alignment

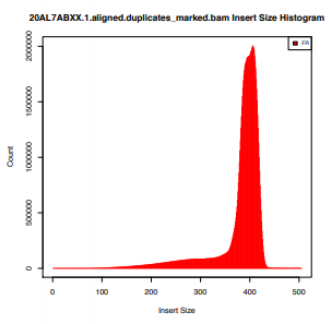
Steps with TopHat

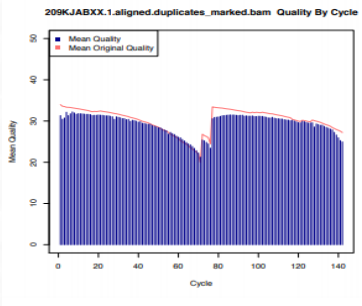
1. Unspliced reads are mapped to locate exons (with Bowtie)

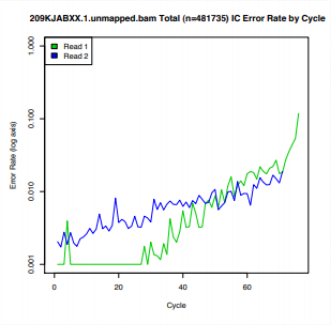
2. Unmapped reads are then split and aligned independently to identify exon

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

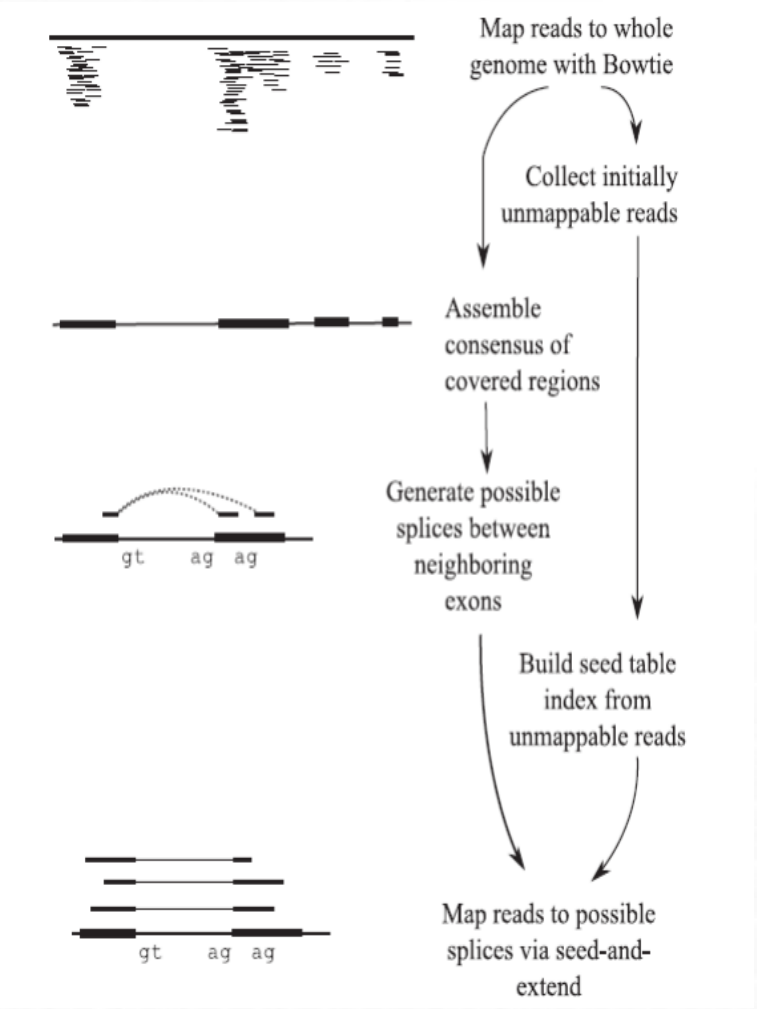
Picard can be used for quality control of mapping







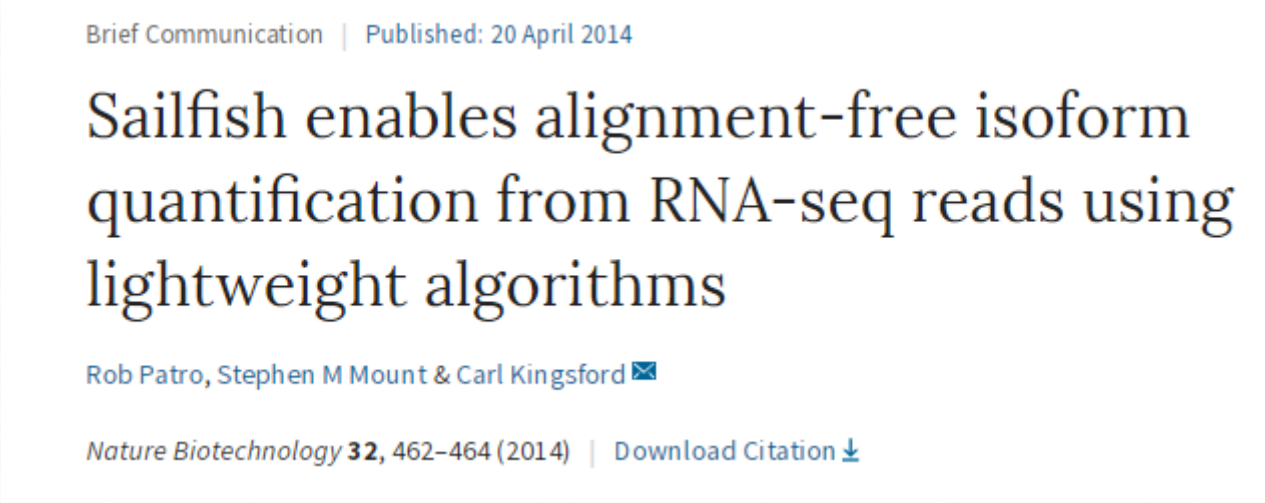
TopHat Pipeline



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

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.

antify 20 ... tes on a laptop computer)

Vall d'Hebron
Institut de Recerca

1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to omics data analysis
4. 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

4.7 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#running-the-differential-expression-pipeline

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

11 April, 2018

This is one….but there are others….

4.7 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

4.7 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

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Data for the analysis:

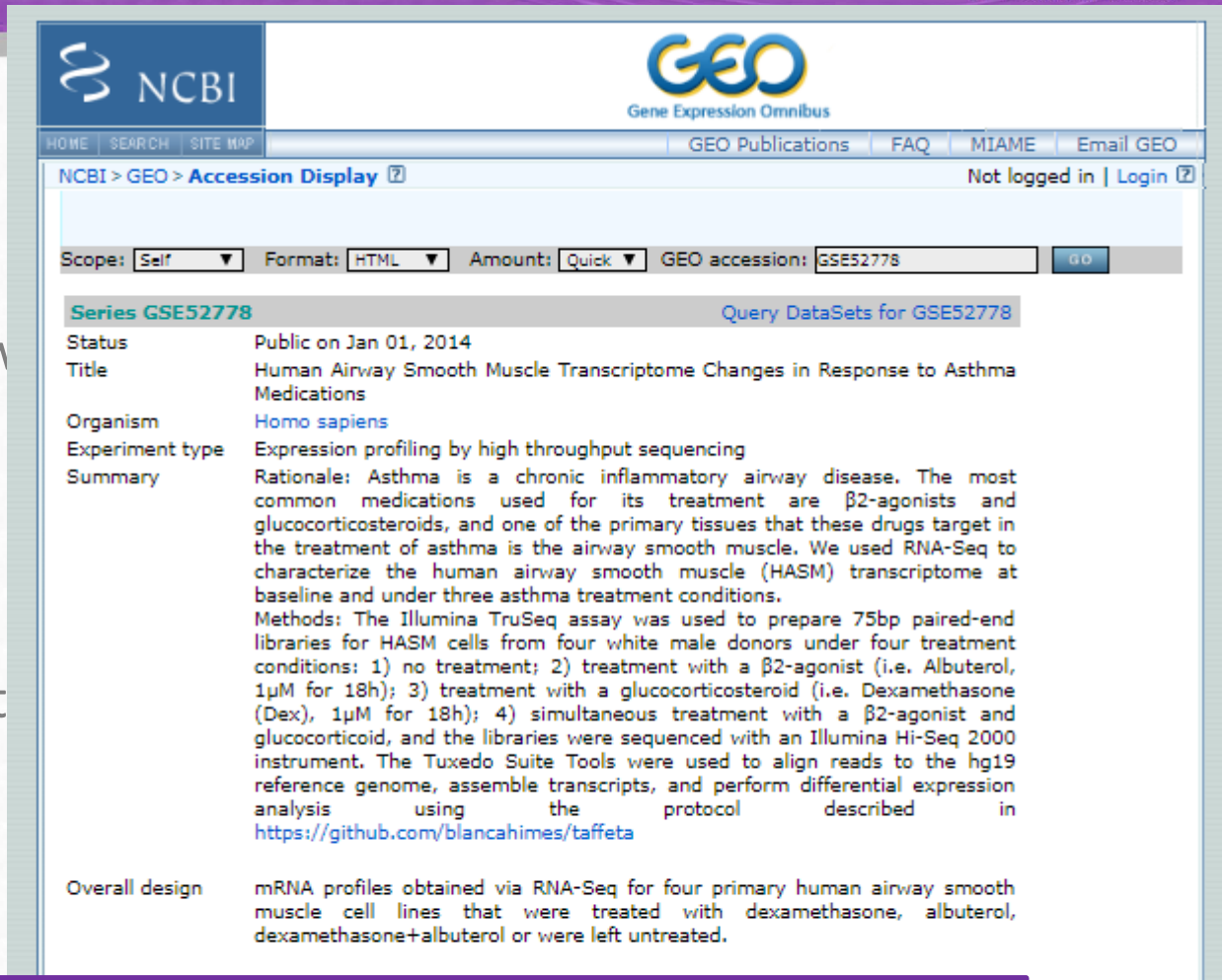
- RNA-seq experiment with airway smooth muscle cells
- Cells were treated with dexamethasone (dexa)
- 4 cell lines
- Experimental design
  - 4 samples treated with 1mM dexa 18 hours
  - 4 samples without treatment

4.7 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].

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

2 Preparing count matrices

As input, the count-based statistical methods, such as *DESeq2* (Love, Huber, and Anders 2014), *edgeR* (Robinson, McCarthy, and Smyth 2009), *limma* with the voom method (Law et al. 2014), *DSS* (Wu, Wang, and Wu 2013), *EBSeq* (Leng et al. 2013) and *baySeq* (Hardcastle and Kelly 2010), expect input data as obtained, e.g., from RNA-seq or another high-throughput sequencing

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

2 Preparing count matrices

As input, the count-based statistical methods, such as *DESeq2* (Love, Huber, and Anders 2014), *edgeR* (Robinson, McCarthy, and Smyth 2009), *limma* with the voom method (Law et al. 2014), *DSS* (Wu, Wang, and Wu 2013), *EBSeq* (Leng et al. 2013) and *baySeq* (Hardcastle and Kelly 2010), expect input data as obtained, e.g., from RNA-seq or another high-throughput sequencing

2.1 Recommended: transcript abundances and the *tximport* pipeline

Before we demonstrate how to align and then count RNA-seq fragments, we mention that a newer and faster alternative pipeline is to use transcript abundance quantification methods such as Salmon (Patro et al. 2017), Sailfish (Patro, Mount, and Kingsford 2014), kallisto (Bray et al. 2016), or RSEM (Li and Dewey 2011), to estimate abundances without aligning reads, followed by the *tximport* package for assembling estimated count and offset matrices for use with Bioconductor differential gene expression packages.

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

2 Preparing count matrices

As input, the count-based statistical methods, such as *DESeq2* (Love, Huber, and Anders 2014), *edgeR* (Robinson, McCarthy, and Smyth 2009), *limma* with the voom method (Law et al. 2014), *DSS* (Wu, Wang, and Wu 2013), *EBSeq* (Leng et al. 2013) and *baySeq* (Hardcastle and Kelly 2010), expect input data as obtained, e.g., from RNA-seq or another high-throughput sequencing

2.1 Recommended: transcript abundances and the *tximport* pipeline

Before we demonstrate how to align and then count RNA-seq fragments, we mention that a newer and faster alternative pipeline is to use transcript abundance quantification methods such as Salmon (Patro et al. 2017), Sailfish (Patro, Mount, and Kingsford 2014), kallisto (Bray et al. 2016), or RSEM (Li and Dewey 2011), to estimate abundances without aligning reads, followed by the *tximport* package for assembling estimated count and offset matrices for use with Bioconductor differential gene expression packages.

2.2 Aligning reads to a reference genome

The computational analysis of an RNA-seq experiment begins from the FASTQ files that contain the nucleotide sequence of each read and a quality score at each position. These reads must first be aligned to a reference genome or transcriptome, or the abundances and estimated counts per transcript can be estimated without alignment, as described above. In either case, it is important to know if the sequencing experiment was single-end or paired-end, as the alignment software will require the user to specify both FASTQ files for a paired-end experiment. The output of this alignment step is commonly stored in a file format called SAM/BAM.

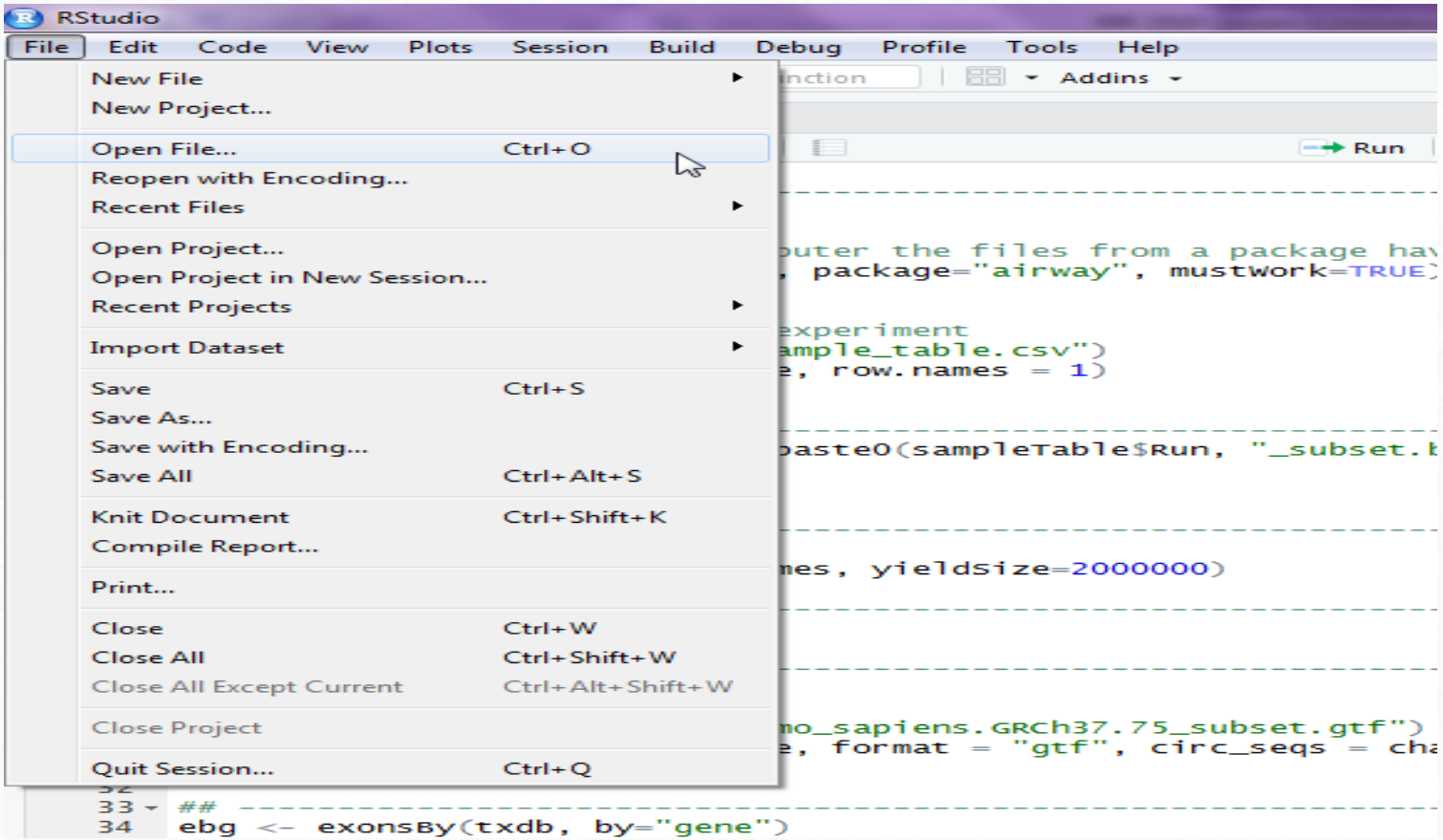
4.7 RNA-seq pipeline with R

2.3 Locating alignment files

Besides the count matrix that we will use later, the *airway* package also contains eight files with a small subset of the total number of reads in the experiment. The reads were selected which aligned to a small region of chromosome 1. Here, for demonstration, we chose a subset of reads because the full alignment files are large (a few gigabytes each), and because it takes between 10-30 minutes to count the fragments for each sample. We will use these files to demonstrate how a count matrix can be constructed from BAM files. Afterwards, we will load the full count matrix corresponding to all samples and all data, which is already provided in the same package, and will continue the analysis with that full matrix.

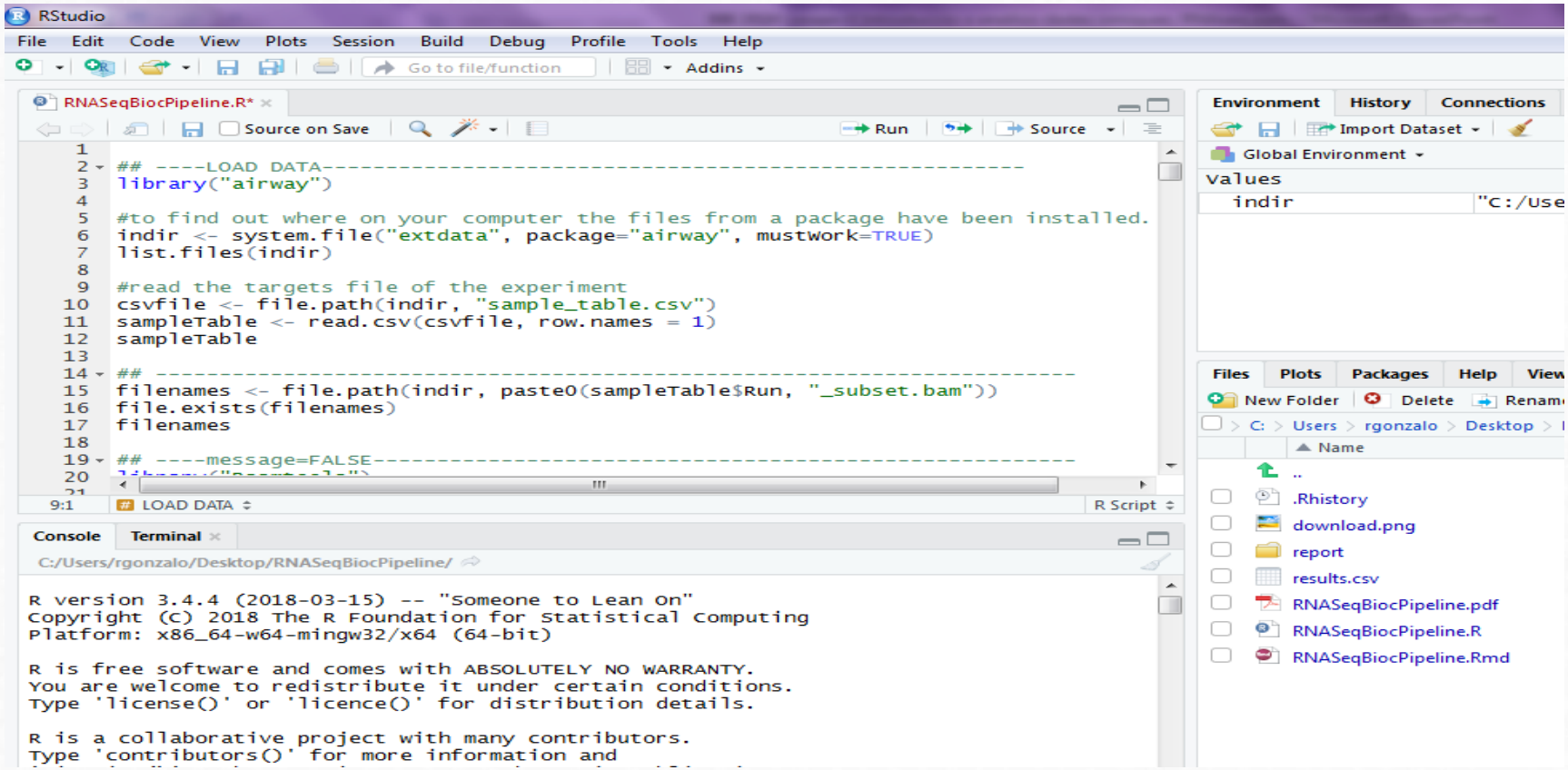
4.7 RNA-seq pipeline with R

1. Open R-Studio
2. Open the file RNAseq_airway.R (course web page)



4.7 RNA-seq pipeline with R

1. Open R-Studio
2. Open the file RNAseq_airway.R (course web page)



4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

- Load the package where the data is stored

```
1
2  ## ----LOAD DATA----------------------------------------------------------
3  library("airway")
4
5  #to find out where on your computer the files from a package have been installed.
6  indir <- system.file("extdata", package="airway", mustWork=TRUE)
7  list.files(indir)
8
```

- See what files we have to perform the analysis

```
> list.files(indir)
 [1] "GSE52778_series_matrix.txt"       "Homo_sapiens.GRCh37.75_subset.gtf"
 [3] "sample_table.csv"                 "SraRunInfo_SRP033351.csv"
 [5] "SRR1039508_subset.bam"            "SRR1039509_subset.bam"
 [7] "SRR1039512_subset.bam"            "SRR1039513_subset.bam"
 [9] "SRR1039516_subset.bam"            "SRR1039517_subset.bam"
[11] "SRR1039520_subset.bam"            "SRR1039521_subset.bam"
>
```

Vall d'Hebron
Institut de Recerca

Targets File

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

```
> 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

```
14  ##store the bam files in object filenames
15  filenames <- file.path(indir, paste0(sampleTable$Run, "_subset.bam"))
16  file.exists(filenames)
17  filenames
18
19  ##Indicate that these are bam files
20  library("Rsamtools")
21  bamfiles <- BamFileList(filenames, yieldSize=2000000)
22
```

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

4.7 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.

```
27  library("GenomicFeatures")
28
29  gtffile <- file.path(indir,"Homo_sapiens.GRCh37.75_subset.gtf")
30  txdb <- makeTxDbFromGFF(gtffile, format = "gtf", circ_seqs = character())
31  txdb
32
```

We indicate that none of our sequences (chromosomes) are circular using a 0-length character vector.

4.7 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.

```
24  ##Defining gene models
25  library("GenomicFeatures")
26
27  gtffile <- file.path(indir,"Homo_sapiens.GRCh37.75_subset.gtf")
28  txdb <- makeTxDbFromGFF(gtffile, format = "gtf", circ_seqs = character())
29  txdb
30
31  ##resume the exons by gene
32  ebg <- exonsBy(txdb, by="gene")
33  ebg
```

Vall d'Hebron
Institut de Recerca

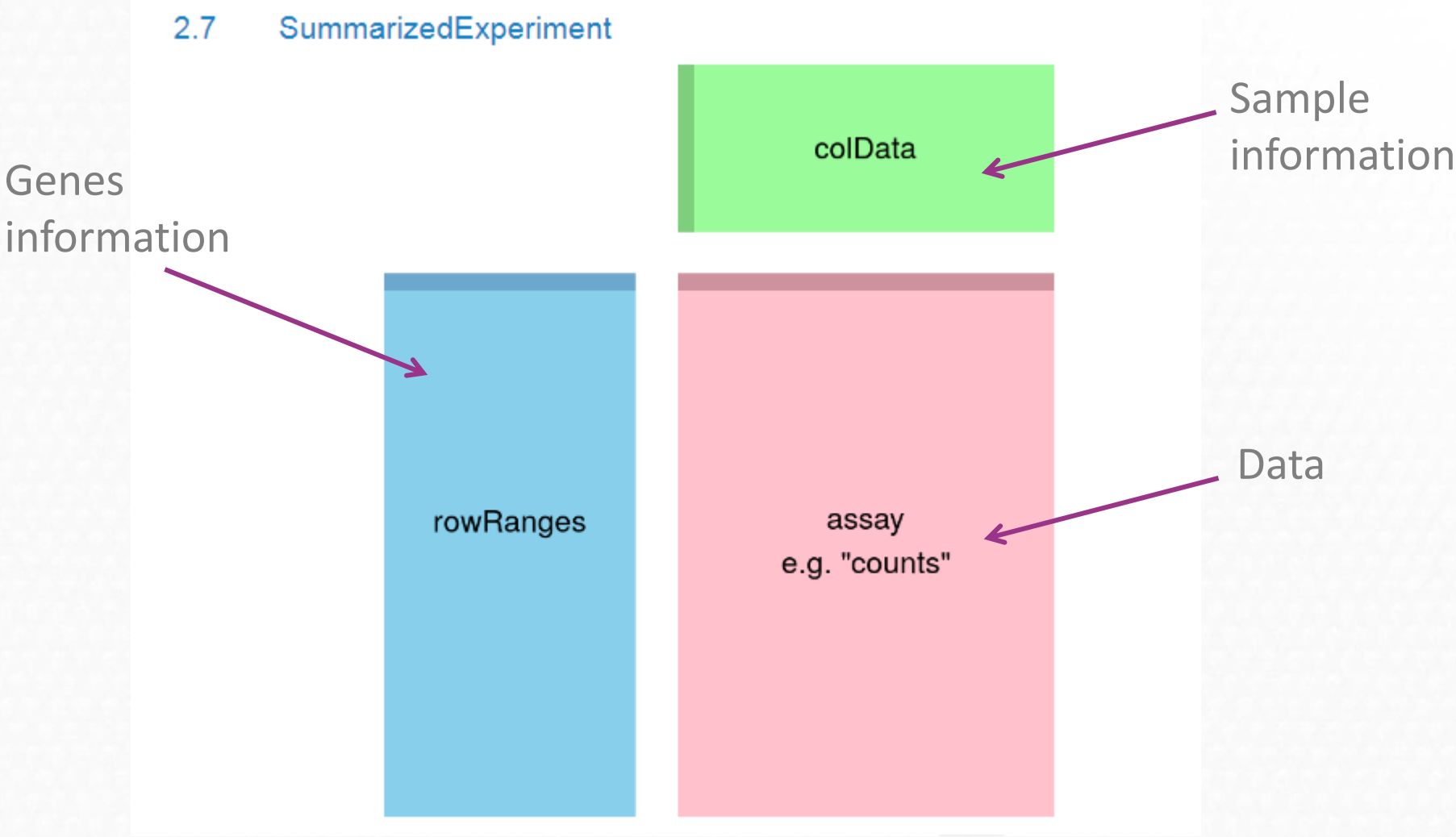
2.6 Read counting step

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.

```
35  ##Generating count Matrices
36  library("GenomicAlignments")
37
38  se <- summarizeOverlaps(features=ebg, reads=bamfiles,
39                          mode="Union",
40                          singleEnd=FALSE,
41                          ignore.strand=TRUE,
42                          fragments=TRUE )
43
```

4.7 RNA-seq pipeline with R



4.7 RNA-seq pipeline with R

```
44  ##See the count matrix
45  se
46  assay(se)
47  #dimensions of the count matrix
48  dim(se)
```

```
> 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
```

Analysis of differential expressed genes

```
50 ##############################################################################
51 ##START FROM THE COMPLETE DATA
52 ##############################################################################
53 data("airway")
54 se <- airway
55 dim(se)
```

```
dim(se)
[1] 64102 8
```

```
58 #reorder the levels
59 library(magrittr)
60 se$dex %<>% relevel("untrt")
61 se$dex
```

4.7 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(se)) / 1e6, 1 )
SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517 SRR1039520 SRR1039521
      20.6       18.8       25.3       15.2       24.4       30.8       19.1       21.2
```

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

```
> ##Check that the object is correct
> colData(se)
DataFrame with 8 rows and 9 columns
            SampleName     cell      dex    albut        Run avgLength Experiment    Sample    BioSample
              <factor> <factor> <factor> <factor>   <factor> <integer>   <factor>  <factor>     <factor>
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
```

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.

```
> ##Filter out those rows without any count
> nrow(dds)
[1] 29391
> dds <- dds[ rowSums(counts(dds)) > 1, ]
> nrow(dds)
[1] 29391
```

Vall d'Hebron
Institut de Recerca

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)**

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

```
> ##variance stabilizing transformation (VST)
> vsd <- vst(dds, blind = FALSE)
> head(assay(vsd), 3)
                 SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517 SRR1039520 SRR1039521
ENSG00000000003   9.742340   9.430742   9.867872   9.646127  10.183344   9.880660  10.010591   9.640065
ENSG00000000419   9.334009   9.582000   9.486456   9.523397   9.427605   9.575154   9.326341   9.509553
ENSG00000000457   8.765748   8.698941   8.651978   8.732909   8.593308   8.703164   8.762420   8.724586
```

Vall d'Hebron
Institut de Recerca

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?

```
85  ##Sample distances.
86  sampleDists <- dist(t(assay(vsd)))
87  sampleDists
```

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Heatmap + Hierarchical Clustering

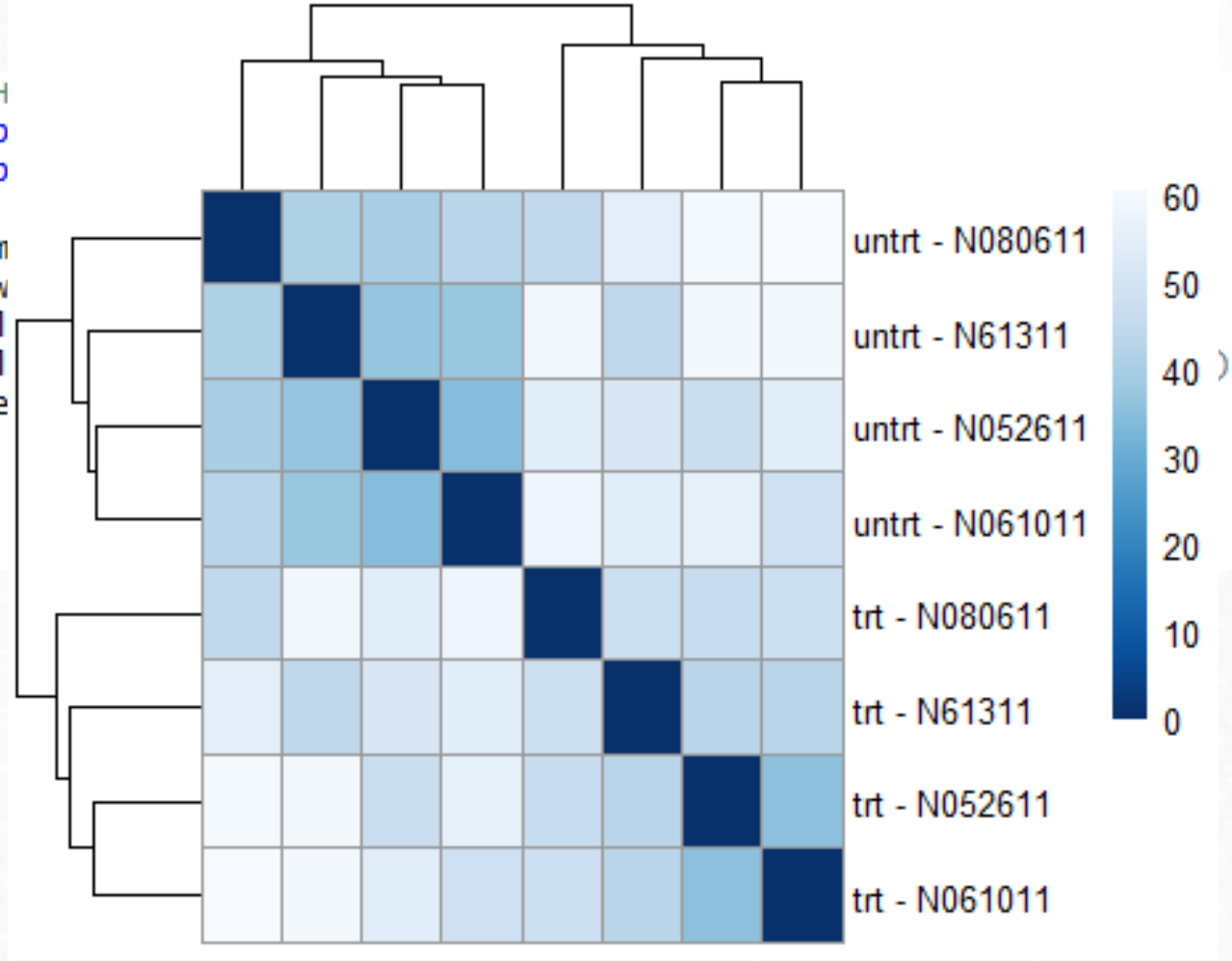
```
 89  ##Hierarchical clustering
 90  library("pheatmap")
 91  library("RColorBrewer")
 92
 93  sampleDistMatrix <- as.matrix( sampleDists )
 94  rownames(sampleDistMatrix) <- paste( vsd$dex, vsd$cell, sep = " - " )
 95  colnames(sampleDistMatrix) <- NULL
 96  colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
 97  pheatmap(sampleDistMatrix,
 98           clustering_distance_rows = sampleDists,
 99           clustering_distance_cols = sampleDists,
100           col = colors)
101
```

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Heatmap + Hierarchical Clustering

```
89  ##H
90  lib
91  lib
92
93  sam
94  row
95  col
96  col
97  phe
98
99
100
101
```




```
" - " )
)
```

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Principal component Analysis Plot

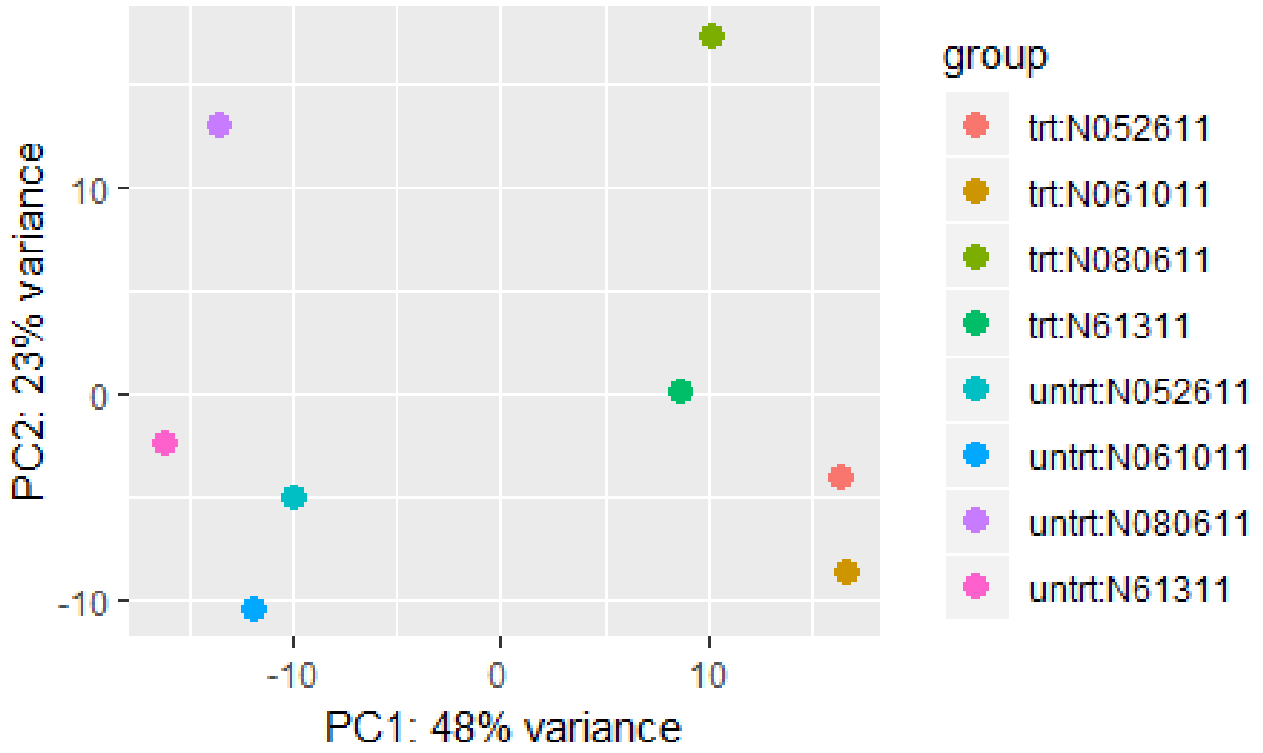
```
102  ##Principal Component Analysis Plot
103  plotPCA(vsd, intgroup = c("dex", "cell"))
```

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Principal

```
102
103
```

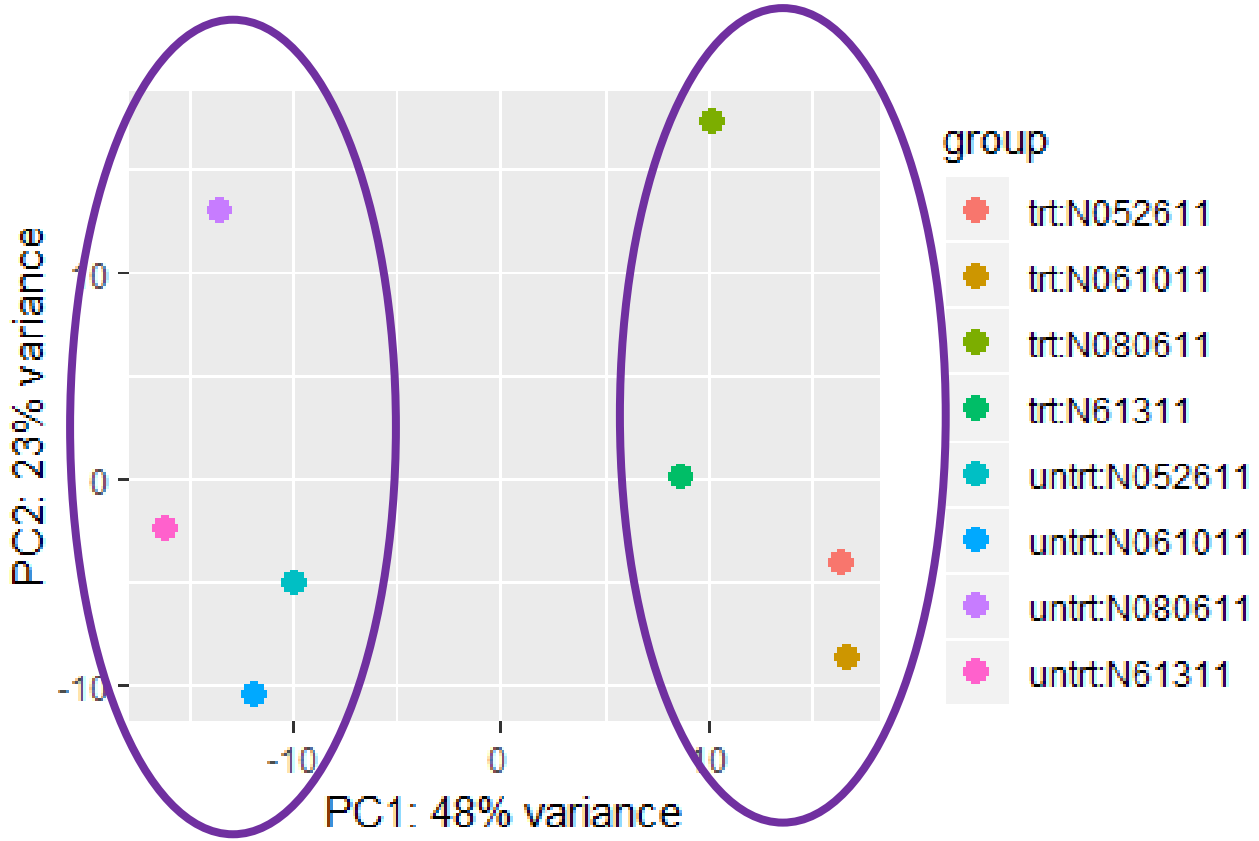


4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Principal

```
102
103
```



Vall d'Hebron
Institut de Recerca

5 Differential expression analysis

5.1 Running the differential expression pipeline

```
> ##Differential Expression Analysis
> dds <- DESeq(dds)
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
```

Vall d'Hebron
Institut de Recerca

Check the results

```
> ##building results table
> res <- results(dds, contrast=c("dex","trt","untrt"))
> ##Toptable
> head(res[order(res$padj),])
log2 fold change (MLE): dex trt vs untrt
Wald test p-value: dex trt vs untrt
DataFrame with 6 rows and 6 columns
                  baseMean log2FoldChange     lfcSE      stat        pvalue          padj
                 <numeric>      <numeric> <numeric> <numeric>     <numeric>     <numeric>
ENSG00000152583   997.4398       4.574919 0.1840564  24.85607 2.223017e-136 4.000096e-132
ENSG00000165995   495.0929       3.291062 0.1331768  24.71198 7.951622e-135 7.154075e-131
ENSG00000120129  3409.0294       2.947810 0.1214350  24.27480 3.619142e-130 2.170762e-126
ENSG00000101347 12703.3871       3.766996 0.1554341  24.23532 9.444883e-130 4.248781e-126
ENSG00000189221  2341.7673       3.353580 0.1417802  23.65337 1.089779e-123 3.921897e-120
ENSG00000211445 12285.6151       3.730403 0.1658267  22.49579 4.563626e-112 1.368631e-108
>
```

4.7 RNA-seq pipeline with R

```
> ##Information about the columns of the results table
> 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
```

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca

Subset the results table to these genes and then sort it by the log2 fold change estimate to get the **significant genes with the strongest down-regulation**:

```
> ##Subset the significant genes with strongest downregulation
> resSig <- subset(res, padj < 0.1)
> head(resSig[ order(resSig$log2FoldChange), ])
log2 fold change (MLE): dex trt vs untrt
Wald test p-value: dex trt vs untrt
DataFrame with 6 rows and 6 columns
                 baseMean log2FoldChange     lfcSE      stat       pvalue         padj
                <numeric>      <numeric> <numeric> <numeric>    <numeric>    <numeric>
ENSG00000128285  6.624741      -5.325912 1.2578863 -4.234017 2.295537e-05 2.378002e-04
ENSG00000267339 26.233573      -4.611553 0.6731316 -6.850894 7.338997e-12 2.053778e-10
ENSG00000019186 14.087605      -4.325920 0.8578247 -5.042895 4.585398e-07 6.611351e-06
ENSG00000183454  5.804171      -4.264087 1.1669498 -3.654045 2.581412e-04 2.048057e-03
ENSG00000146006 46.807597      -4.211875 0.5288797 -7.963767 1.668799e-15 7.166675e-14
ENSG00000141469 53.436528      -4.124784 1.1297977 -3.650905 2.613174e-04 2.069606e-03
```

Vall d'Hebron
Institut de Recerca

Subset the results table to these genes and then sort it by the log2 fold change estimate to get the **significant genes with the strongest up-regulation**:

```
> ##Subset the significant genes with strongest upregulation
> head(ressig[ order(ressig$log2FoldChange, decreasing = TRUE), ])
log2 fold change (MLE): dex trt vs untrt
Wald test p-value: dex trt vs untrt
DataFrame with 6 rows and 6 columns
                  baseMean log2FoldChange     lfcSE      stat       pvalue         padj
                 <numeric>      <numeric> <numeric> <numeric>    <numeric>    <numeric>
ENSG00000179593  67.243048       9.505972 1.0545111  9.014578 1.976299e-19 1.252166e-17
ENSG00000109906 385.071029       7.352628 0.5363902 13.707610 9.141988e-43 2.253437e-40
ENSG00000250978  56.318194       6.327393 0.6778153  9.334981 1.010098e-20 7.212582e-19
ENSG00000132518   5.654654       5.885113 1.3241367  4.444491 8.810031e-06 1.000175e-04
ENSG00000127954 286.384119       5.207160 0.4930828 10.560419 4.546302e-26 5.049763e-24
ENSG00000249364   8.839061       5.098168 1.1596852  4.396166 1.101798e-05 1.223812e-04
```

4.7 RNA-seq pipeline with R

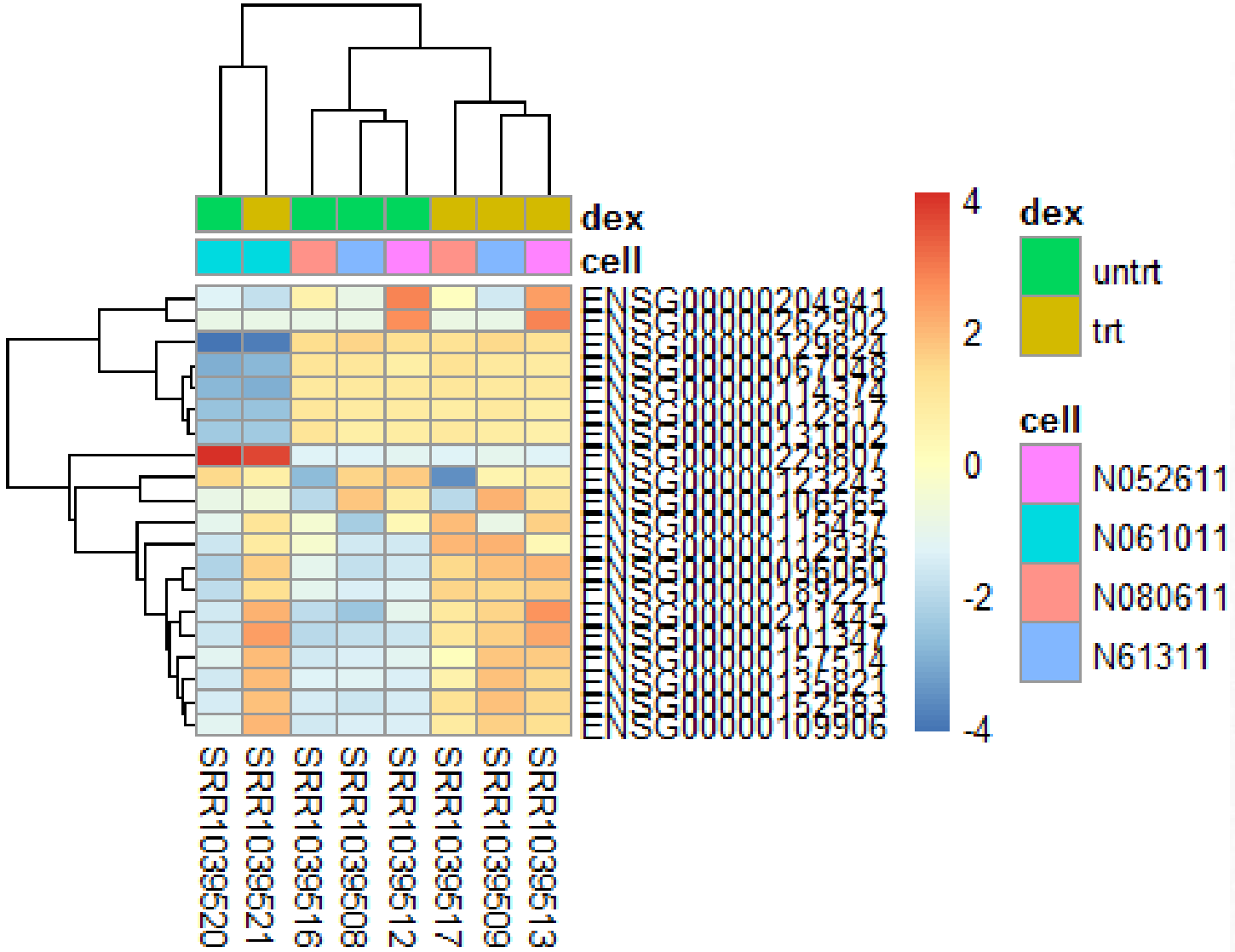
Vall d'Hebron
Institut de Recerca

6.3 Gene clustering

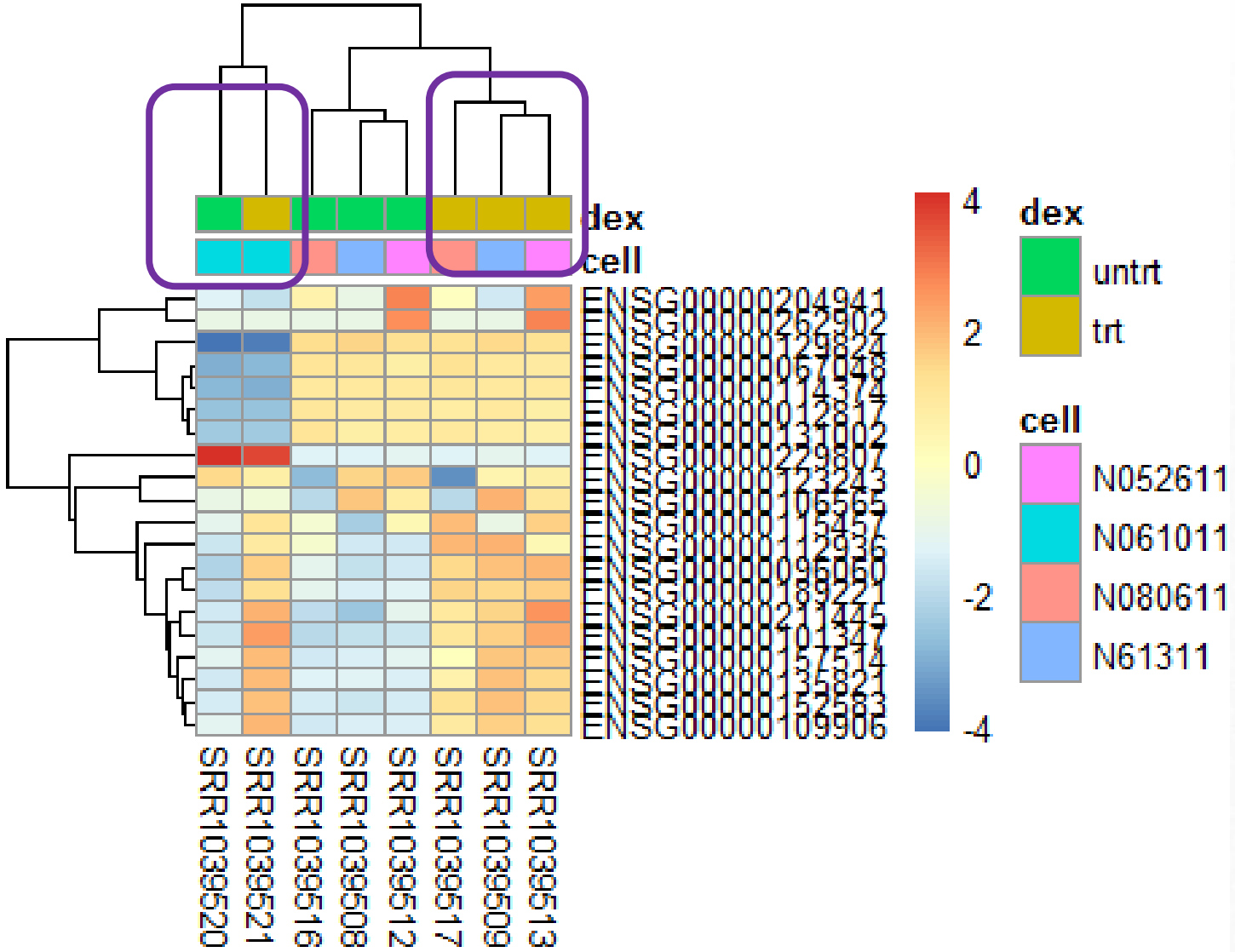
In the sample distance heatmap made previously, the dendrogram at the side shows us a hierarchical clustering of the samples. Such a clustering can also be performed for the genes. Since the clustering is only relevant for genes that actually carry a signal, one usually would only cluster a subset of 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.

```
125 ##Heatmap of genes
126 library("genefilter")
127 topVarGenes <- head(order(rowVars(assay(vsd)), decreasing = TRUE), 20)
128
129 mat  <- assay(vsd)[topVarGenes, ]
130 mat  <- mat - rowMeans(mat)
131 anno <- as.data.frame(colData(vsd)[, c("cell","dex")])
132 pheatmap(mat, annotation_col = anno)
133
```

4.7 RNA-seq pipeline with R



4.7 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*:

```
135  ##Results annotation
136  library("org.Hs.eg.db")
137  library("AnnotationDbi")
138
139  res$symbol <- mapIds(org.Hs.eg.db,
140                       keys=row.names(res),
141                       column="SYMBOL",
142                       keytype="ENSEMBL",
143                       multiVals="first")
144
145  res$entrez <- mapIds(org.Hs.eg.db,
146                       keys=row.names(res),
147                       column="ENTREZID",
148                       keytype="ENSEMBL",
149                       multiVals="first")
150
151
152  resOrdered <- res[order(res$pvalue),]
153  head(resOrdered)
```

4.7 RNA-seq pipeline with R

```
> head(resOrdered)
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      entrez
               <numeric>      <numeric> <numeric> <numeric>    <numeric>    <numeric> <character> <character>
ENSG00000152583   997.4398       4.574919 0.1840564  24.85607 2.223017e-136 4.000096e-132     SPARCL1        8404
ENSG00000165995   495.0929       3.291062 0.1331768  24.71198 7.951622e-135 7.154075e-131      CACNB2         783
ENSG00000120129  3409.0294       2.947810 0.1214350  24.27480 3.619142e-130 2.170762e-126       DUSP1        1843
ENSG00000101347 12703.3871       3.766996 0.1554341  24.23532 9.444883e-130 4.248781e-126      SAMHD1       25939
ENSG00000189221  2341.7673       3.353580 0.1417802  23.65337 1.089779e-123 3.921897e-120        MAOA        4128
ENSG00000211445 12285.6151       3.730403 0.1658267  22.49579 4.563626e-112 1.368631e-108        GPX3        2878
```

4.7 RNA-seq pipeline with R

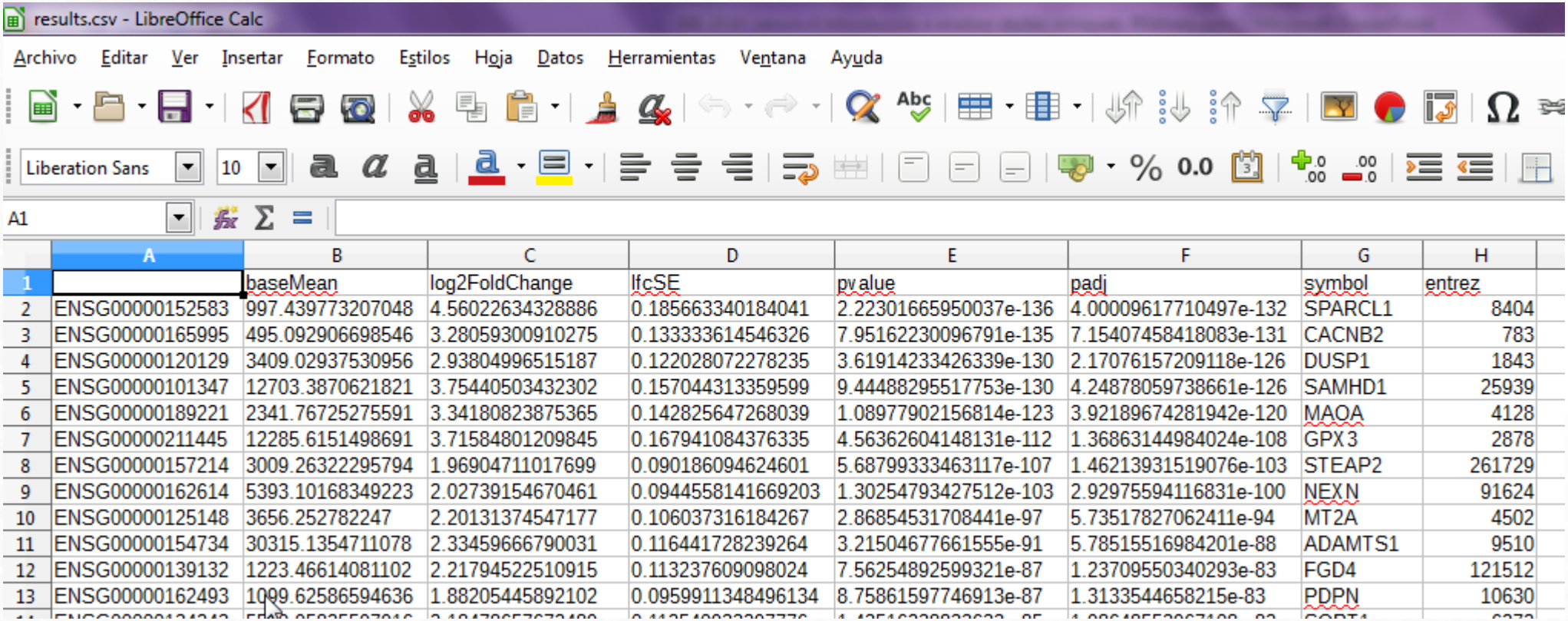
7.1 Exporting results

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.

```
157  ##Exporting the results
158  resOrderedDF <- as.data.frame(resOrdered)[1:100, ]
159  write.csv(resOrderedDF, file = "results.csv")
```

4.7 RNA-seq pipeline with R

Vall d'Hebron
Institut de Recerca



	A	B	C	D	E	F	G	H
1		baseMean	log2FoldChange	lfcSE	pvalue	padj	symbol	entrez
2	ENSG00000152583	997.439773207048	4.56022634328886	0.185663340184041	2.22301665950037e-136	4.00009617710497e-132	SPARCL1	8404
3	ENSG00000165995	495.092906698546	3.28059300910275	0.133333614546326	7.95162230096791e-135	7.15407458418083e-131	CACNB2	783
4	ENSG00000120129	3409.02937530956	2.93804996515187	0.122028072278235	3.61914233426339e-130	2.17076157209118e-126	DUSP1	1843
5	ENSG00000101347	12703.3870621821	3.75440503432302	0.157044313359599	9.44488295517753e-130	4.24878059738661e-126	SAMHD1	25939
6	ENSG00000189221	2341.76725275591	3.34180823875365	0.142825647268039	1.08977902156814e-123	3.92189674281942e-120	MAOA	4128
7	ENSG00000211445	12285.6151498691	3.71584801209845	0.167941084376335	4.56362604148131e-112	1.36863144984024e-108	GPX3	2878
8	ENSG00000157214	3009.26322295794	1.96904711017699	0.090186094624601	5.68799333463117e-107	1.46213931519076e-103	STEAP2	261729
9	ENSG00000162614	5393.10168349223	2.02739154670461	0.0944558141669203	1.30254793427512e-103	2.92975594116831e-100	NEXN	91624
10	ENSG00000125148	3656.252782247	2.20131374547177	0.106037316184267	2.86854531708441e-97	5.73517827062411e-94	MT2A	4502
11	ENSG00000154734	30315.1354711078	2.33459666790031	0.116441728239264	3.21504677661555e-91	5.78515516984201e-88	ADAMTS1	9510
12	ENSG00000139132	1223.46614081102	2.21794522510915	0.113237609098024	7.56254892599321e-87	1.23709550340293e-83	FGD4	121512
13	ENSG00000162493	1099.62586594636	1.88205445892102	0.0959911348496134	8.75861597746913e-87	1.3133544658215e-83	PDPN	10630