RNA-seq with R-bioconductor Part 1.

- Maarten Leerkes PhD

Topics

- What is R
- What is Bioconductor
- What is RNAseq

What is R

- R is a programming language and software environment for statistical computing and graphics. The R language is widely used among statisticians and data miners for developing statistical software[2][3] and data analysis.

What is R

- R is an implementation of the S programming language combined with lexical scoping semantics inspired by Scheme. S was created by John Chambers while at Bell Labs. There are some important differences, but much of the code written for S runs unaltered.

What is R

- R is a GNU project. The source code for the R software environment is written primarily in C, Fortran, and R. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems. R uses a command line interface; there are also several graphical front-ends for it.

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

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2015-04-16, Full of Ingredients) R-3.2.0.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- 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.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the R project homepage for further information.

[Home]

Download

CRAN

R Project

About R
Contributors
What's New?
Mailing Lists
Bug Tracking
Conferences
Search

R Foundation

Foundation
Board
Members
Donors
Donate

Documentation

Manuals
FAQs
The R Journal
Books
Certification
Other

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

- **R 3.2.1 (World-Famous Astronaut) prerelease versions** will appear starting June 8. Final release is scheduled for 2015-06-18.
- **R version 3.2.0** (Full of Ingredients) has been released on 2015-04-16.
- **R version 3.1.3** (Smooth Sidewalk) has been released on 2015-03-09.
- **The R Journal Volume 6/2** is available.
- **useR! 2015**, will take place at the University of Aalborg, Denmark, June 30 - July 3, 2015.
- **useR! 2014**, took place at the University of California, Los Angeles, USA June 30 - July 3, 2014.

Topics

- What is R
- What is Bioconductor
- What is RNAseq

Rstudio

RStudio Products Resources

Download RStudio

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

If you run R on a Linux server and want to enable users to remotely access RStudio using a web browser please download RStudio Server.

Do you need support or a commercial license? Check out our commercial offerings

RStudio Desktop 0.99.489 — Release Notes

RStudio requires R 2.11.1 (or higher). If you don't already have R, you can download it here.

Rstudio

https://www.rstudio.com/products/rstudio/download/

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 0.99.489 - Windows Vista/7/8/10	73.9 MB	2015-11-05	`7ef8c00311d5c03b6c9abe22826497d6`
RStudio 0.99.489 - Mac OS X 10.6+ (64-bit)	56.2 MB	2015-11-05	`05cf866b07df6552583f98314ed09d38`
RStudio 0.99.489 - Ubuntu 12.04+/Debian 8+ (32-bit)	77.4 MB	2015-11-05	`1bf2997d91b6eaf0b483fbc52cca29b5`
RStudio 0.99.489 - Ubuntu 12.04+/Debian 8+ (64-bit)	83.9 MB	2015-11-05	`ed089d88cc2e5901e311c66f7b1ada8b`
RStudio 0.99.489 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	76.8 MB	2015-11-05	`642ede6193cc3ff24a55c3ffe20c31bc`
RStudio 0.99.489 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	77.7 MB	2015-11-05	`1a71fbfd49730695515d4f5343885d6b`

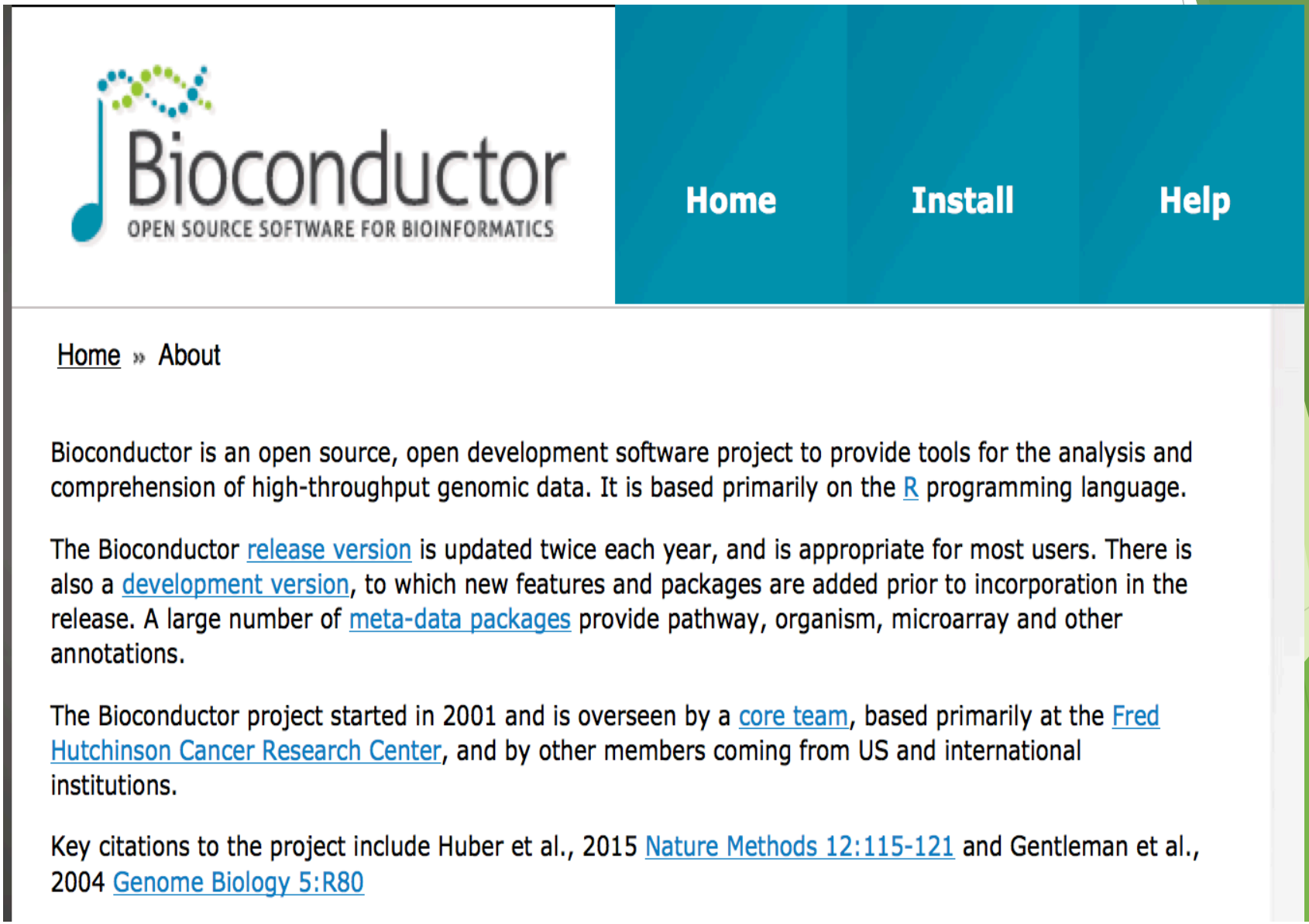
Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 0.99.489 - Windows Vista/7/8/10	105.5 MB	2015-11-05	`cb654d8480f6f740ad4a9e2bc56172a7`
RStudio 0.99.489 - Ubuntu 12.04+/Debian 8+ (32-bit)	78.1 MB	2015-11-05	`eb78f3e3c5af7146b70387d81ac0381e`
RStudio 0.99.489 - Ubuntu 12.04+/Debian 8+ (64-bit)	84.8 MB	2015-11-05	`ba5ae48bee96654e6f6ee4249bc2470b`
RStudio 0.99.489 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	77.4 MB	2015-11-05	`08d58c938fdaf4d761222ed8ffc48f7e`
RStudio 0.99.489 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	78.4 MB	2015-11-05	`f88b4d35b3b0e2cfe34b1a6aa85ca7e3`

Source Code

A tarball containing source code for RStudio v0.99.489 can be downloaded from here

What is bioconductor



Topics

- What is R
- What is Bioconductor
- What is RNAseq

What is RNAseq

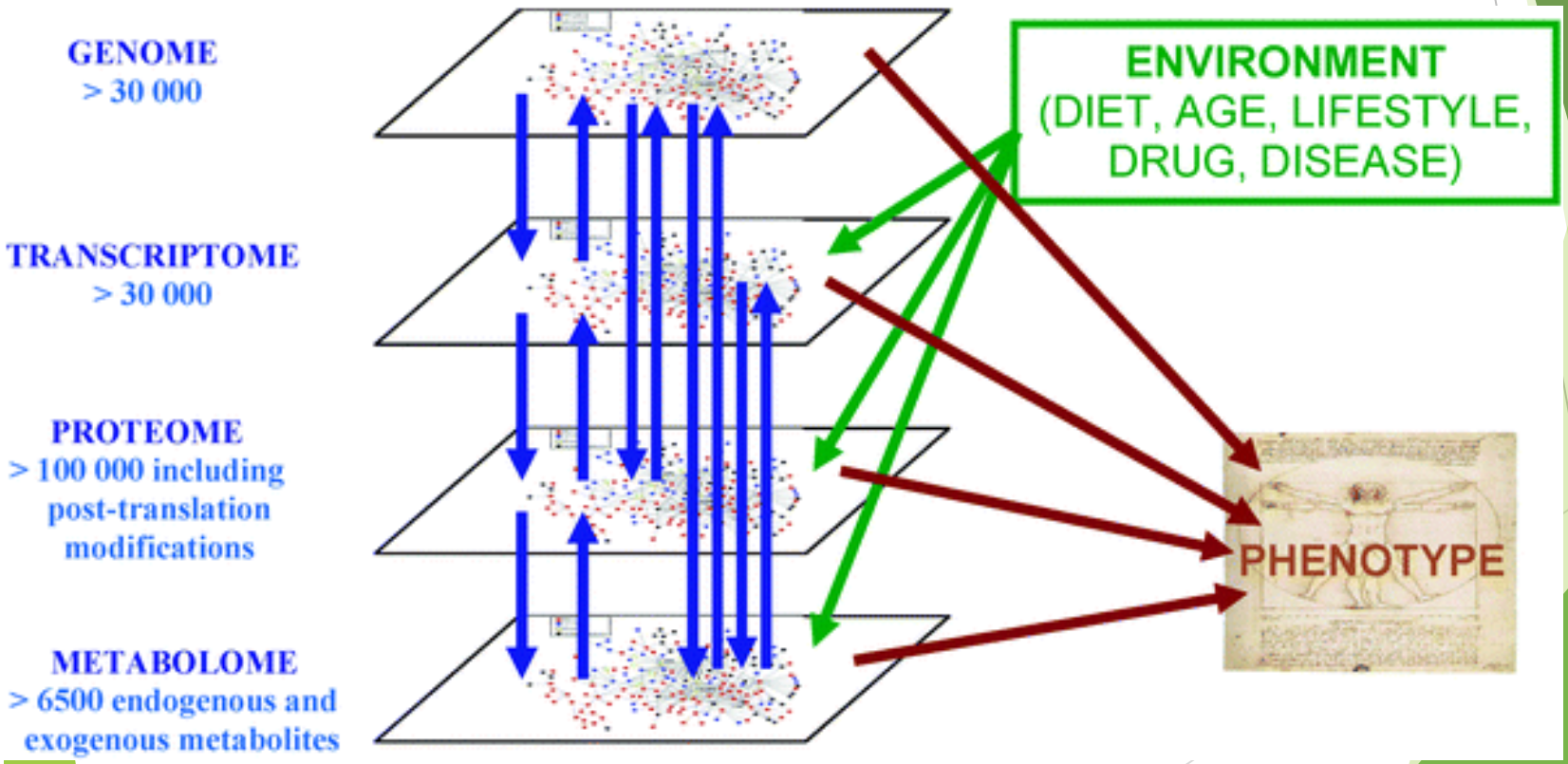
- **RNA-seq (RNA Sequencing), also called Whole Transcriptome Shotgun Sequencing (WTSS), is a technology that uses the capabilities of next-generation sequencing to reveal a snapshot of RNA presence and quantity from a genome at a given moment in time.**

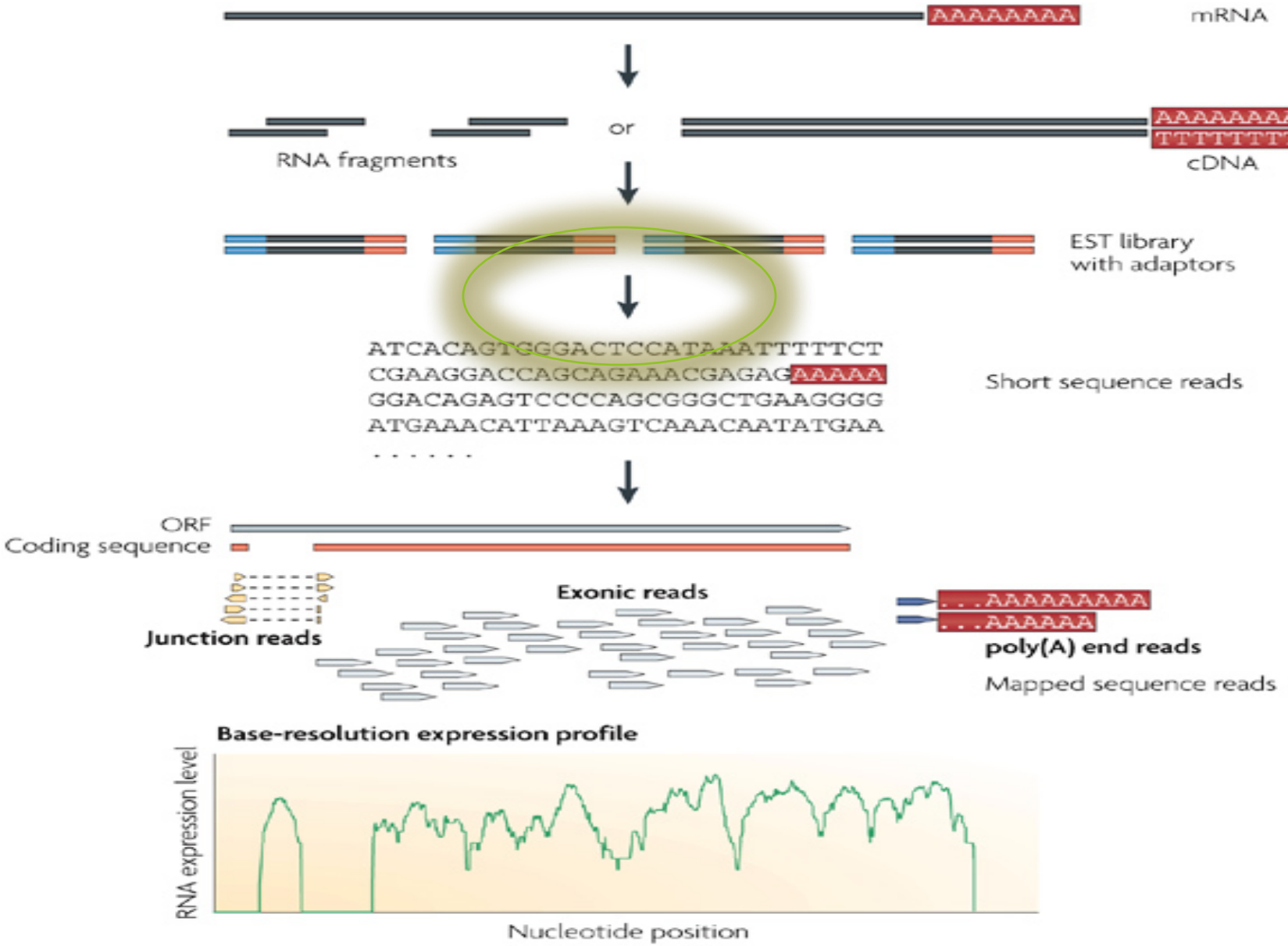
What is RNAseq

Whole Transcriptome Shotgun Sequencing hints to phenotyp

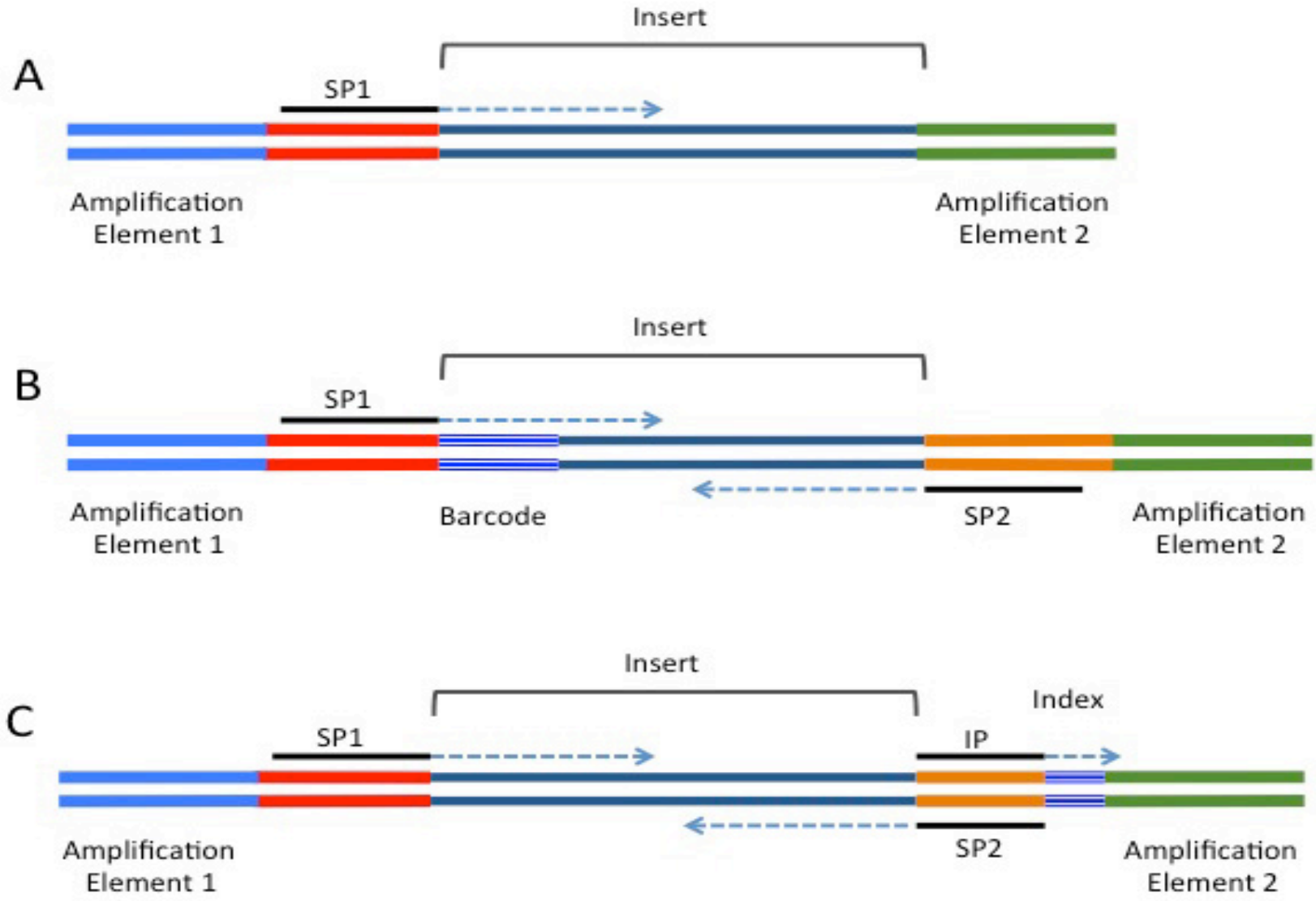
What is RNAseq

- Whole Transcriptome Shotgun Sequencing hints to phenotypes




AAAAAAAA
mRNA
or
RNA fragments
AAAAAAAA
TTTTTTTT
cDNA
EST library
with adaptors
ATCACAGTGGGACTCCATAAATTTTTCT
CGAAGGACCAGCAGAAACGAGAGAAAAA
GGACAGAGTCCCCAGCGGGCTGAAGGGG
ATGAAACATTAAAGTCAAACAATATGAA
. . . . . .
Short sequence reads
ORF
Coding sequence
Exonic reads
Junction reads
. . .AAAAAAAAA
. . .AAAAAA
poly(A) end reads
Mapped sequence reads
Base-resolution expression profile
RNA expression level
Nucleotide position
Nature Reviews | Genetics

Insert size


A
Insert
SP1
Amplification Element 1
Amplification Element 2
B
Insert
SP1
Amplification Element 1
Barcode
SP2
Amplification Element 2
C
Insert
Index
SP1
IP
SP2
Amplification Element 1
Amplification Element 2

Sequencing by synthesis

- Intro to Sequencing by Synthesis:
- https://www.youtube.com/watch?v=HMyCqWhwB8E
- http://rnaseq.uoregon.edu/
- https://www.youtube.com/watch?v=womKfikWlxM

Topics

- What is R
- What is Bioconductor
- What is RNAseq

- Comes together in: RNA-seq with R-bioconductor

Different kinds of objects in R

- **Objects.**
- The following data objects exist in R:
- vectors
- lists
- arrays
- matrices
- tables
- data frames
- Some of these are more important than others. And there are more.

A **matrix** is a collection of data elements arranged in a two-dimensional rectangular layout. The following is an example of a matrix with 2 rows and 3 columns.

$$A = \begin{bmatrix} 2 & 4 & 3 \\ 1 & 5 & 7 \end{bmatrix}$$

```
> A = matrix(
+   c(2, 4, 3, 1, 5, 7), # the data elements
+   nrow=2,              # number of rows
+   ncol=3,              # number of columns
+   byrow = TRUE)        # fill matrix by rows

> A                      # print the matrix
     [,1] [,2] [,3]
[1,]    2    4    3
[2,]    1    5    7
```

An element at the m^{th} row, n^{th} column of A can be accessed by the expression A[m, n].

```
> A[2, 3]      # element at 2nd row, 3rd column
[1] 7
```

The entire m^{th} row A can be extracted as A[m,].

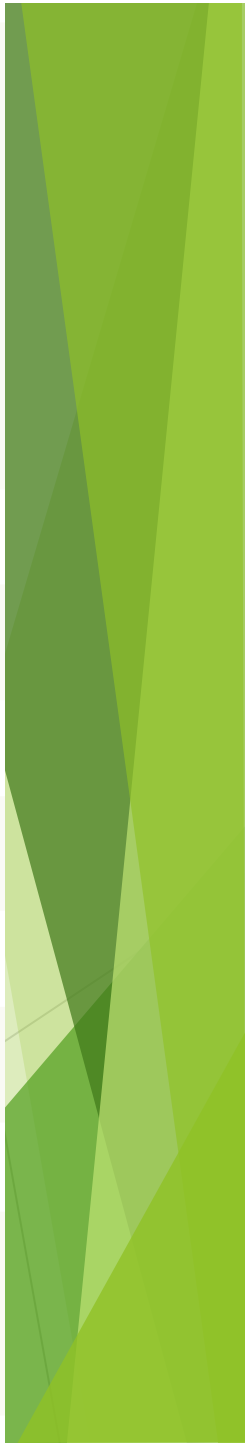
```
> A[2, ]       # the 2nd row
[1] 1 5 7
```

Similarly, the entire n^{th} column A can be extracted as A[,n].

```
> A[ ,3]       # the 3rd column
[1] 3 7
```

We can also extract more than one rows or columns at a time.

```
> A[ ,c(1,3)]  # the 1st and 3rd columns
     [,1] [,2]
[1,]    2    3
[2,]    1    7
```



A data frame is used for storing data tables. It is a list of vectors of equal length.

- A **data frame** is a table, or two-dimensional array-like structure, in which each column contains measurements on one variable, and each row contains one case. As we shall see, a "case" is not necessarily the same as an experimental subject or unit, although they are often the same.

	A	B	C	D
1	First Name	Last Name	Age	Salary
2	Jon	Smith	36	26500
3	Helen	Mirren	22	21000
4	David	Cameron	29	39000
5	Brad	Pitt	52	45000
6	Anna	Starolsky	41	22500
7	Peter	Piper	20	31500
8	David	Duck	19	15700
9	Julie	Walters	33	19000

Combine list of data frames into single data frame, add column with list index: list of vectors of equal length.

```
> dfList[[1]]
```

a	b	c
g	1.2724293	-0.005767173
j	0.4146414	2.404653389
o	-1.53995	0.763593461
x	-0.928567	-0.799009249
f	-0.2947204	-1.147657009

```
> dfList[[2]]
```

a	b	c
k	-0.04493361	0.91897737
a	-0.01619026	0.7821363
j	0.94383621	0.07456498
w	0.8212212	-1.9893517
i	0.59390132	0.61982575

```
> dfList[[3]]
```

a	b	c
m	-1.28459935	-0.6494716
w	0.04672617	0.7267507
l	-0.23570656	1.1519118
g	-0.54288826	0.9921604
b	-0.43331032	-0.4295131

index	a	b	c
1	g	1.2724293	-0.005767173
1	j	0.4146414	2.404653389
1	o	-1.53995	0.763593461
1	x	-0.928567	-0.799009249
1	f	-0.2947204	-1.147657009
2	k	-0.04493361	0.91897737
2	a	-0.01619026	0.7821363
2	j	0.94383621	0.07456498
2	w	0.8212212	-1.9893517
2	i	0.59390132	0.61982575
3	m	-1.28459935	-0.6494716
3	w	0.04672617	0.7267507
3	l	-0.23570656	1.1519118
3	g	-0.54288826	0.9921604
3	b	-0.43331032	-0.4295131

Methods: software carpentry:

http://swcarpentry.github.io/r-novice-inflammation/01-starting-with-data.html

software carpentry

Programming with R

Analyzing patient data

Learning Objectives

- Read tabular data from a file into a program.
- Assign values to variables.
- Select individual values and subsections from data.
- Perform operations on a data frame of data.
- Display simple graphs.

We are studying inflammation in patients who have been given a new treatment for arthritis, and need to analyze the first dozen data sets. The data sets are stored in comma-separated values (CSV) format. Each row holds the observations for just one patient. Each column holds the inflammation measured in a day, so we have a set of values in successive days. The first few rows of our first file look like this:

```
0,0,1,3,1,2,4,7,8,3,3,3,10,5,7,4,7,7,12,18,6,13,11,11,7,7,4,6,8,8,4,4,5,7,3,4,2,3,0,0
0,1,2,1,2,1,3,2,2,6,10,11,5,9,4,4,7,16,8,6,18,4,12,5,12,7,11,5,11,3,3,5,4,4,5,5,1,1,0,1
0,1,1,3,3,2,6,2,5,9,5,7,4,5,4,15,5,11,9,10,19,14,12,17,7,12,11,7,4,2,10,5,4,2,2,3,2,2,1,1
0,0,2,0,4,2,2,1,6,7,10,7,9,13,8,8,15,10,10,7,17,4,4,7,6,15,6,4,9,11,3,5,6,3,3,4,2,3,2,1
0,1,1,3,3,1,3,5,2,4,4,7,6,5,3,10,8,10,6,17,9,14,9,7,13,9,12,6,7,7,9,6,3,2,2,4,2,0,1,1
```

We want to:

- Load data into memory,
- Calculate the average value of inflammation per day across all patients, and
- Plot the results.

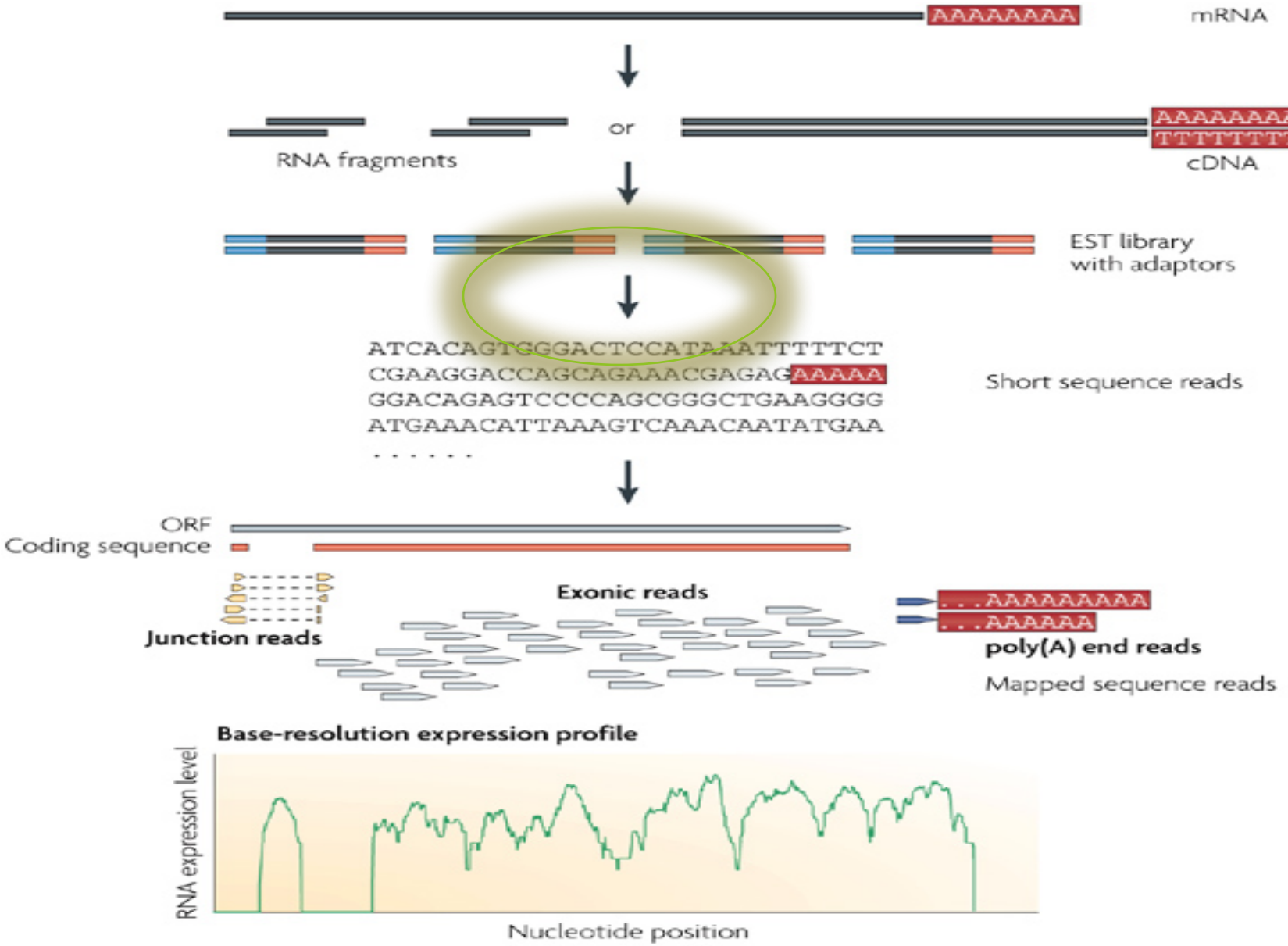
To do all that, we'll have to learn a little bit about programming.

Current working directory cwd

```
> getwd()
[1] "/Users/class06/Desktop/new"
>
```

Topics

- What is R
- What is Bioconductor
- What is RNAseq


AAAAAAAA
mRNA
or
RNA fragments
AAAAAAAA
TTTTTTTT
cDNA
EST library
with adaptors
ATCACAGTGGGACTCCATAAATTTTTCT
CGAAGGACCAGCAGAAACGAGAGAAAAA
GGACAGAGTCCCCAGCGGGCTGAAGGGG
ATGAAACATTAAAGTCAAACAATATGAA
. . . . . .
Short sequence reads
ORF
Coding sequence
Exonic reads
Junction reads
. . .AAAAAAAAA
. . .AAAAAA
poly(A) end reads
Mapped sequence reads
Base-resolution expression profile
RNA expression level
Nucleotide position
Nature Reviews | Genetics

Sequencing by synthesis

- Intro to Sequencing by Synthesis:
- https://www.youtube.com/watch?v=HMyCqWhwB8E

FASTQ read with 50nt in Illumina format (ASCII_BASE=33).

Information in ID line

Illumina FASTQ files

Divided into blocks of **4 lines**

Location of the cluster

Lane **Tile**

Machine ID | **Run ID** | | | **X pos** | **Y pos**

```
@ILMN-GA001_3_208HWAAXX_1_1_110_812
ATACAAGCAAGTATAAGTTCGTATGCCGTCTT
+ILMN-GA001_3_208HWAAXX_1_1_110_812
hhhYhh]NYhhhhhhYIhhaZT[hYHNSPKXR
@ILMN-GA001_3_208HWAAXX_1_1_111_879
GGAGGCTGGAGTTGGGGACGTATGCGGCATAG
+ILMN-GA001_3_208HWAAXX_1_1_111_879
hSWhRNJ\hFhLdhVOhAIB@NFKD@PAB?N?
```

1. ID
2. Sequence
4. Quality

Information in quality line

Illumina quality scores

```
@ILMN-GA001_3_208HWAAXX_1_1_111_879
GGAGGCTGGAGTTGGGGACGTATGCGGCATAG
+ILMN-GA001_3_208HWAAXX_1_1_111_879
hSWhRNJ\hFhLdhVOhAIB@NFKD@PAB?N?
```

String of same length as sequence: one score for each nucleotide

Quality scores represented as ASCII characters (ASCII-**64**)

http://en.wikipedia.org/wiki/ASCII

-> **64** is added to the Phred score

-> ASCII character that corresponds to sum is printed

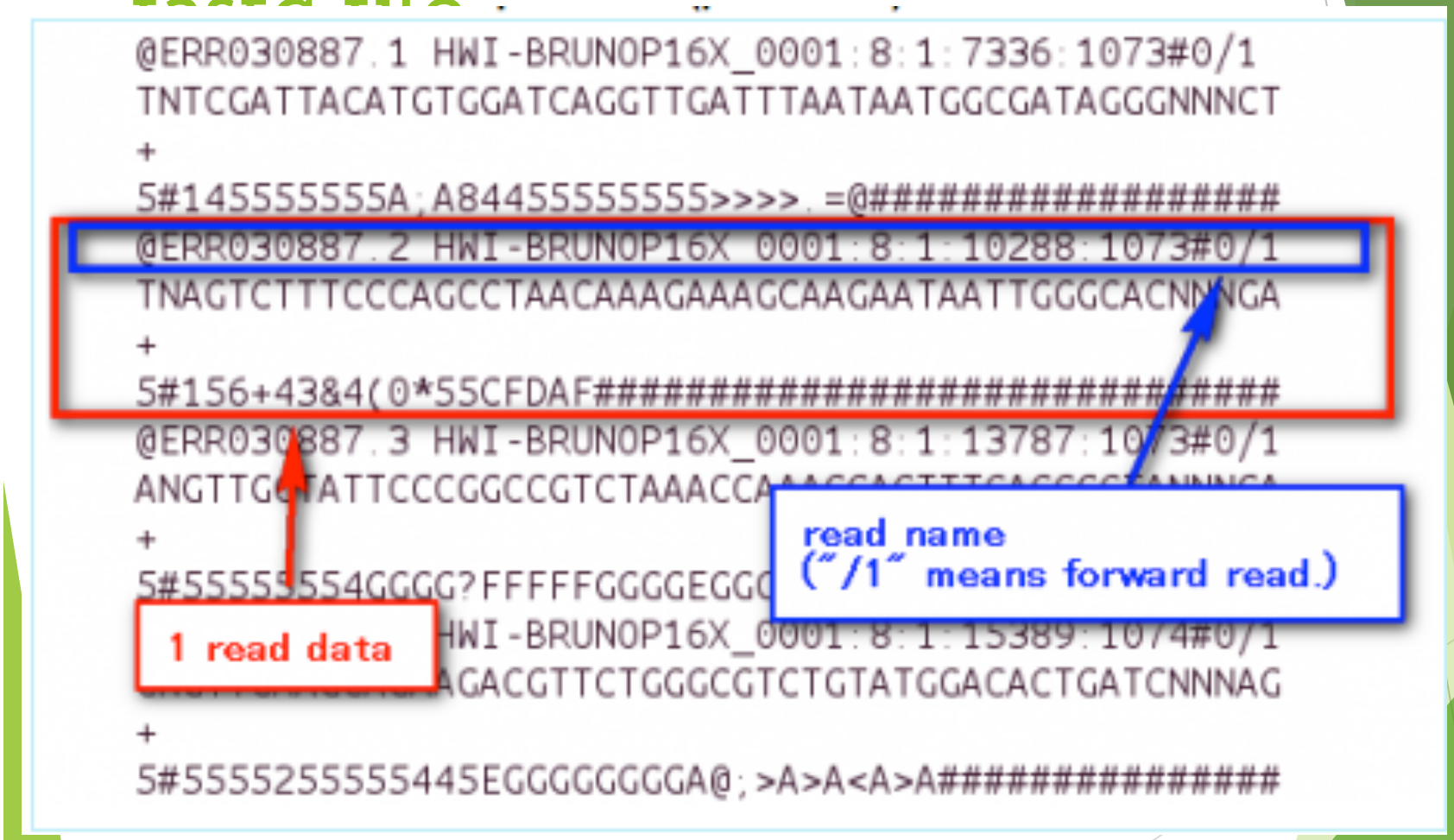
e.g. Phred score = 40 ; 40 + 64 = 104 ; character 104 in ASCII code = h

Illumina fastq

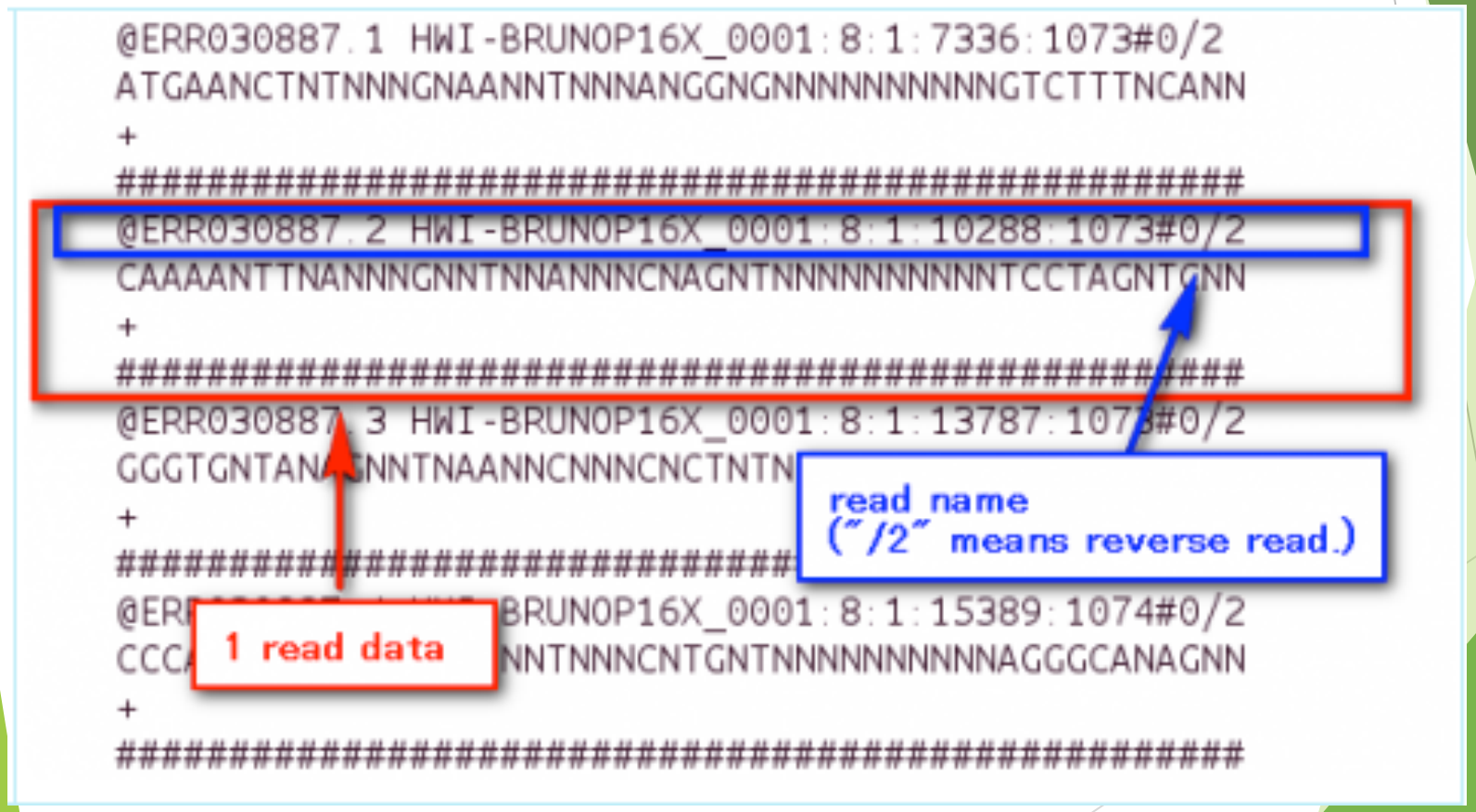
```
      1          2       3    4    5   6 7     8
@HWI-ST226:253:D14WFACXX:2:1101:2743:29814 1:N:0:ATCACG
TGCGGAAGGATCATTGTGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATGCCGTCTTCTGCTT
GAAAAAAAAAAAAAAAAATTA
+
B@CFFFFFHHFFHJIIGHIHIJJIJIIJJGDCHIIIJJJJJJJGJGIHHEH@)=F@EIGHHEHFFFFDCBBD:@CC@C
:<CDDDD50559<B########
```

1. unique instrument ID and run ID
2. Flow cell ID and lane
3. tile number within the flow cell lane
4. 'x'-coordinate of the cluster within the tile
5. 'y'-coordinate of the cluster within the tile
6. the member of a pair, /1 or /2 *(paired-end or mate-pair reads only)*
7. N if the read passes filter, Y if read fails filter otherwise
8. Index sequence

Paired end: read 1 in one fasta file



Paired end: read 2 in another fastq file



Intermezzo: quality scores

"Phred-score": used for sequence quality as well as mapping quality

Chance of 1/1000 that read is mapped at wrong position = 10-3 => phred-score = 30
Chance of 1/100 that read is mapped at wrong position = 10-2 => phred-score = 20

Sanger encoding: quality score 30 = ">"

```
  SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
  ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................
  ..............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII.....................
  ...................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.....................
  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
  |                         |    |        |                              |                     |
 33                        59   64       73                             104                   126

 S - Sanger        Phred+33,  raw reads typically (0, 40)
 X - Solexa        Solexa+64, raw reads typically (-5, 40)
 I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
 J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
    with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
    (Note: See discussion above).
```

Ascii off-sets

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

ASCII_BASE=64 Old Illumina

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	64 @	11	0.07943	75 K	22	0.00631	86 V	33	0.00050	97 a
1	0.79433	65 A	12	0.06310	76 L	23	0.00501	87 W	34	0.00040	98 b
2	0.63096	66 B	13	0.05012	77 M	24	0.00398	88 X	35	0.00032	99 c
3	0.50119	67 C	14	0.03981	78 N	25	0.00316	89 Y	36	0.00025	100 d
4	0.39811	68 D	15	0.03162	79 O	26	0.00251	90 Z	37	0.00020	101 e
5	0.31623	69 E	16	0.02512	80 P	27	0.00200	91 [	38	0.00016	102 f
6	0.25119	70 F	17	0.01995	81 Q	28	0.00158	92 \	39	0.00013	103 g
7	0.19953	71 G	18	0.01585	82 R	29	0.00126	93]	40	0.00010	104 h
8	0.15849	72 H	19	0.01259	83 S	30	0.00100	94 ^	41	0.00008	105 i
9	0.12589	73 I	20	0.01000	84 T	31	0.00079	95 _	42	0.00006	106 j
10	0.10000	74 J	21	0.00794	85 U	32	0.00063	96 `			

Numerous possible analysis strategies

- There is no one ‘correct’ way to analyze RNA-seq data
- Two major branches
  - Direct alignment of reads (spliced or unspliced) to genome or transcriptome
  - Assembly of reads followed by alignment*

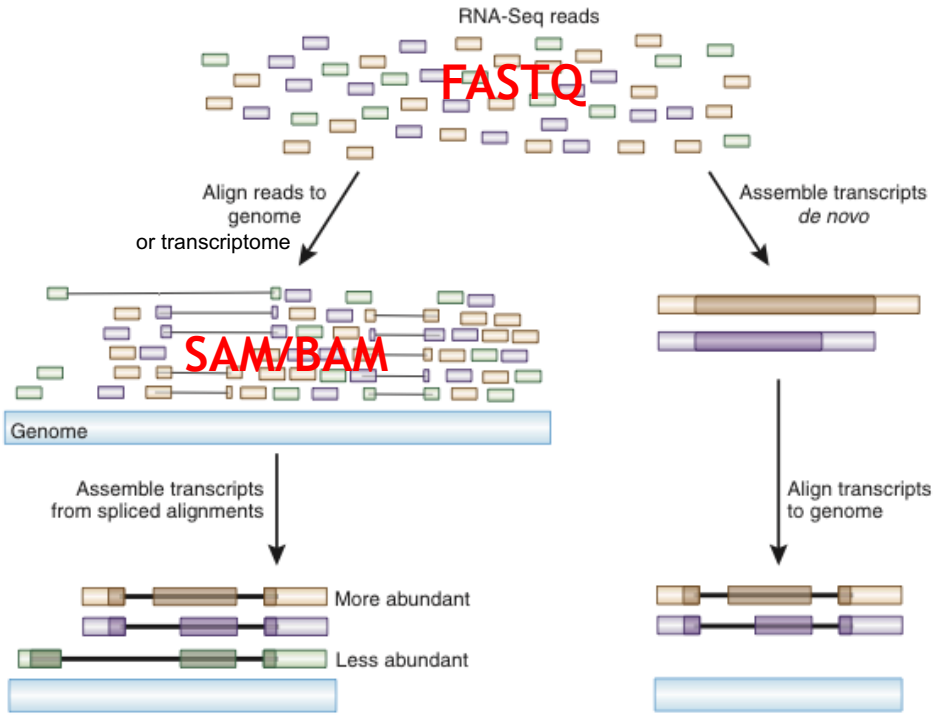


Image from Haas & Zody, 2010

*Assembly is the only option when working with a creature with no genome sequence, alignment of contigs may be to ESTs, cDNAs etc

What is an annotation file in GFF (General Feature Format) ?

- The evolution was something like this:
- GFF 2 -> GTF -> GFF 3

  conversions:
http://song.cvs.sourceforge.net/song/software/scripts/gtf2gff3/
- GFF/GTF File Format - Definition and supported options
- The GFF (General Feature Format) format consists of one line per feature, each containing 9 columns of data, plus optional track definition lines. The following documentation is based on the Version 2 specifications:
- http://useast.ensembl.org/info/website/upload/gff.html?redirect=no

GFF3 file example

```
##gff-version   3
##sequence-region   ctg123 1 1497228
ctg123 . gene            1000  9000  .  +  .  ID=gene00001;Name=EDEN
ctg123 . TF_binding_site 1000  1012  .  +  .  Parent=gene00001
ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00001;Parent=gene00001
ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00002;Parent=gene00001
ctg123 . mRNA            1300  9000  .  +  .  ID=mRNA00003;Parent=gene00001
ctg123 . exon            1300  1500  .  +  .  Parent=mRNA00003
ctg123 . exon            1050  1500  .  +  .  Parent=mRNA00001,mRNA00002
ctg123 . exon            3000  3902  .  +  .  Parent=mRNA00001,mRNA00003
ctg123 . exon            5000  5500  .  +  .  Parent=mRNA00001,mRNA00002,mRNA00003
ctg123 . exon            7000  9000  .  +  .  Parent=mRNA00001,mRNA00002,mRNA00003
ctg123 . CDS             1201  1500  .  +  0  ID=cds00001;Parent=mRNA00001
ctg123 . CDS             3000  3902  .  +  0  ID=cds00001;Parent=mRNA00001
ctg123 . CDS             5000  5500  .  +  0  ID=cds00001;Parent=mRNA00001
ctg123 . CDS             7000  7600  .  +  0  ID=cds00001;Parent=mRNA00001
ctg123 . CDS             1201  1500  .  +  0  ID=cds00002;Parent=mRNA00002
ctg123 . CDS             5000  5500  .  +  0  ID=cds00002;Parent=mRNA00002
ctg123 . CDS             7000  7600  .  +  0  ID=cds00002;Parent=mRNA00002
ctg123 . CDS             3301  3902  .  +  0  ID=cds00003;Parent=mRNA00003
ctg123 . CDS             5000  5500  .  +  2  ID=cds00003;Parent=mRNA00003
ctg123 . CDS             7000  7600  .  +  2  ID=cds00003;Parent=mRNA00003
ctg123 . CDS             3391  3902  .  +  0  ID=cds00004;Parent=mRNA00003
ctg123 . CDS             5000  5500  .  +  2  ID=cds00004;Parent=mRNA00003
ctg123 . CDS             7000  7600  .  +  2  ID=cds00004;Parent=mRNA00003
```

Column 1: "seqid"

Column 2: "source"

Column 3: "type"

Column 4: "start"

Column 5: "end"

Column 6: "score"

Column 7: "strand"

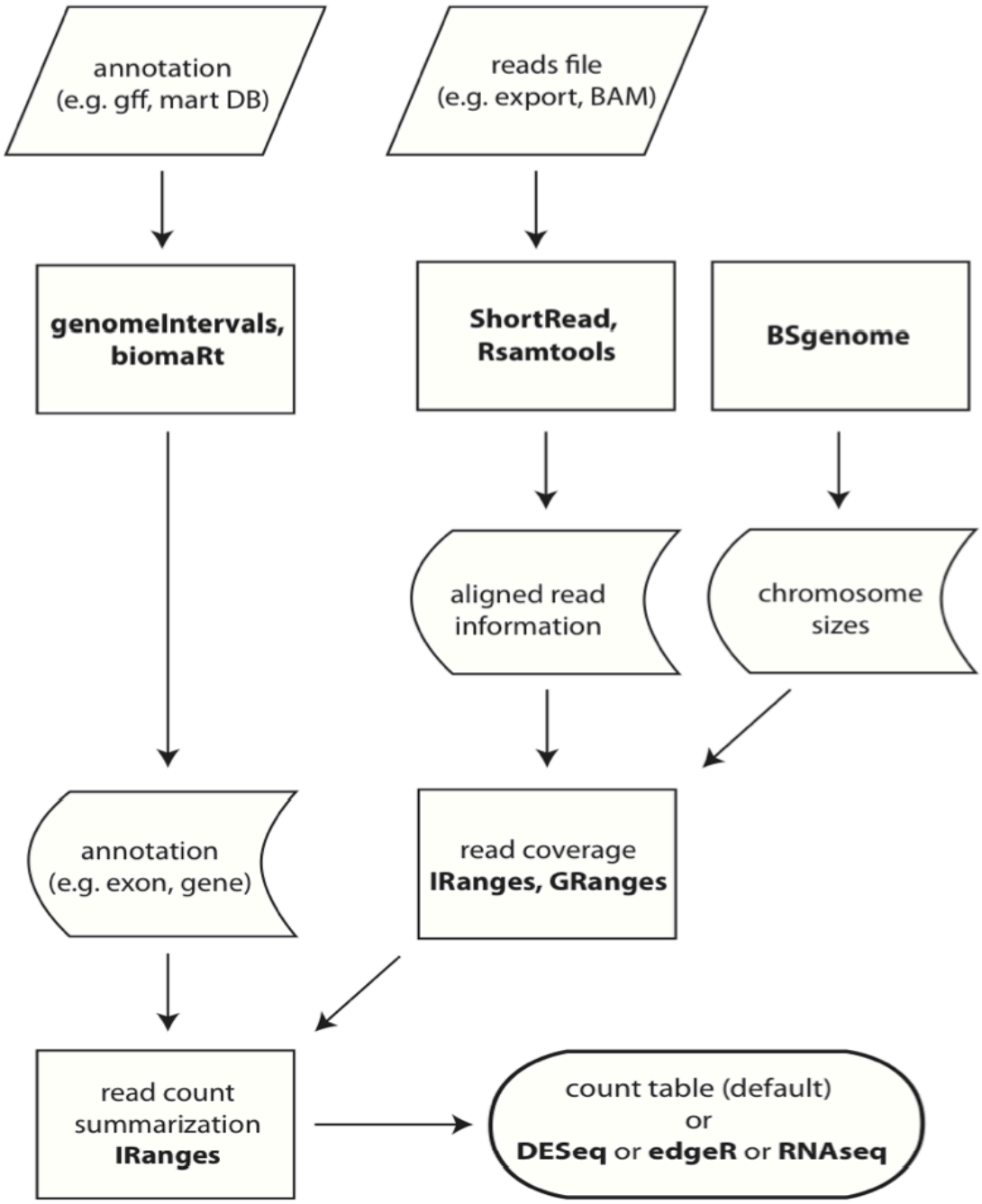
Column 8: "phase"

Column 9: "attributes"

UCSC Genome Browser

Checklist for mapping to reference genome

1. A **reference** genome sequence (fasta), to be indexed by the alignment software.
2. A genome **annotation** file (GFF3 or GTF), with indication of currently known annotations (optional, but highly recommended)
3. The cleaned (preprocessed) **reads** (fastq)


annotation
(e.g. gff, mart DB)
reads file
(e.g. export, BAM)
genomeIntervals,
biomaRt
ShortRead,
Rsamtools
BSgenome
aligned read
information
chromosome
sizes
annotation
(e.g. exon, gene)
read coverage
IRanges, GRanges
read count
summarization
IRanges
count table (default)
or
DESeq or edgeR or RNAseq

Alternative to count tables:

HTSeq: Analysing high-throughput sequencing data with Python

HTSeq is a Python package that provides infrastructure to process data from high-throughput sequencing assays.

- Please see the chapter *A tour through HTSeq* first for an overview on the kind of analysis you can do with HTSeq and the design of the package, and then look at the reference documentation.
- While the main purpose of HTSeq is to allow you to write your own analysis scripts, customized to your needs, there are also a couple of stand-alone scripts for common tasks that can be used without any Python knowledge. See the *Scripts* section in the overview below for what is available.
- For downloads and installation instructions, see *Prequisites and installation*.

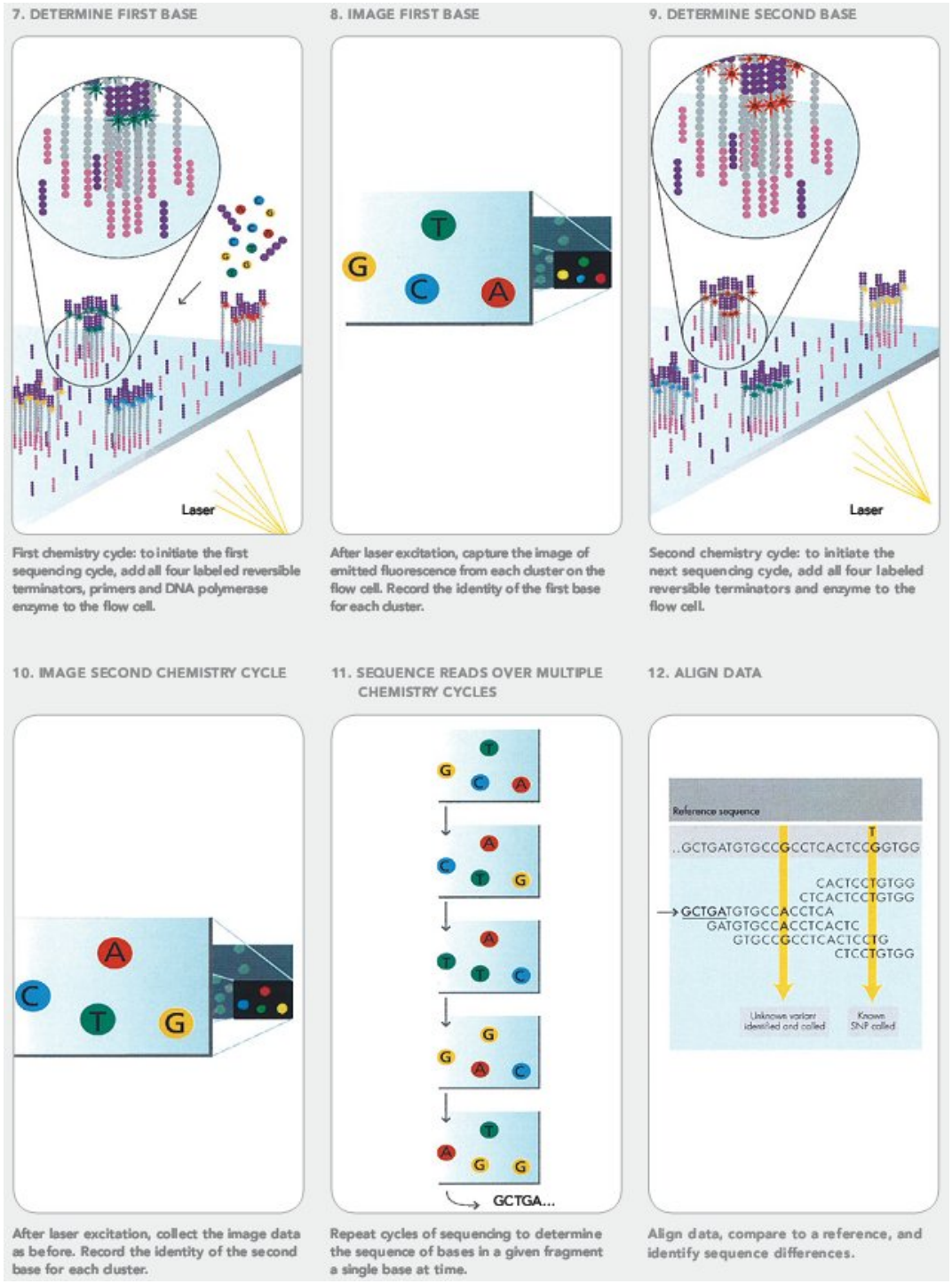
Paper

HTSeq is described in the following publication:

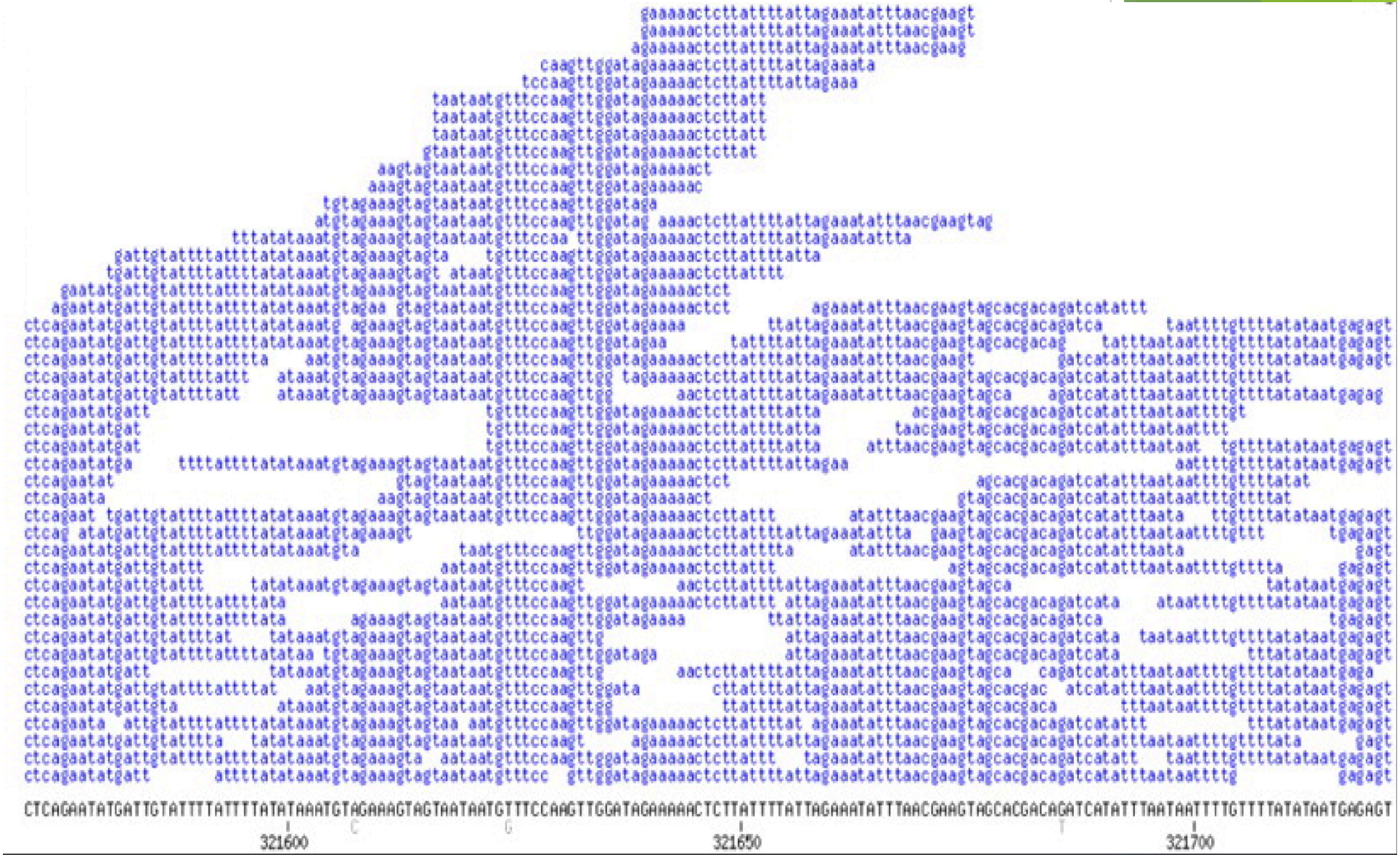
> Simon Anders, Paul Theodor Pyl, Wolfgang Huber
> *HTSeq – A Python framework to work with high-throughput sequencing data*
> Bioinformatics (2014), in print, online at doi:10.1093/bioinformatics/btu638

If you use HTSeq in research, please cite this paper in your publication.

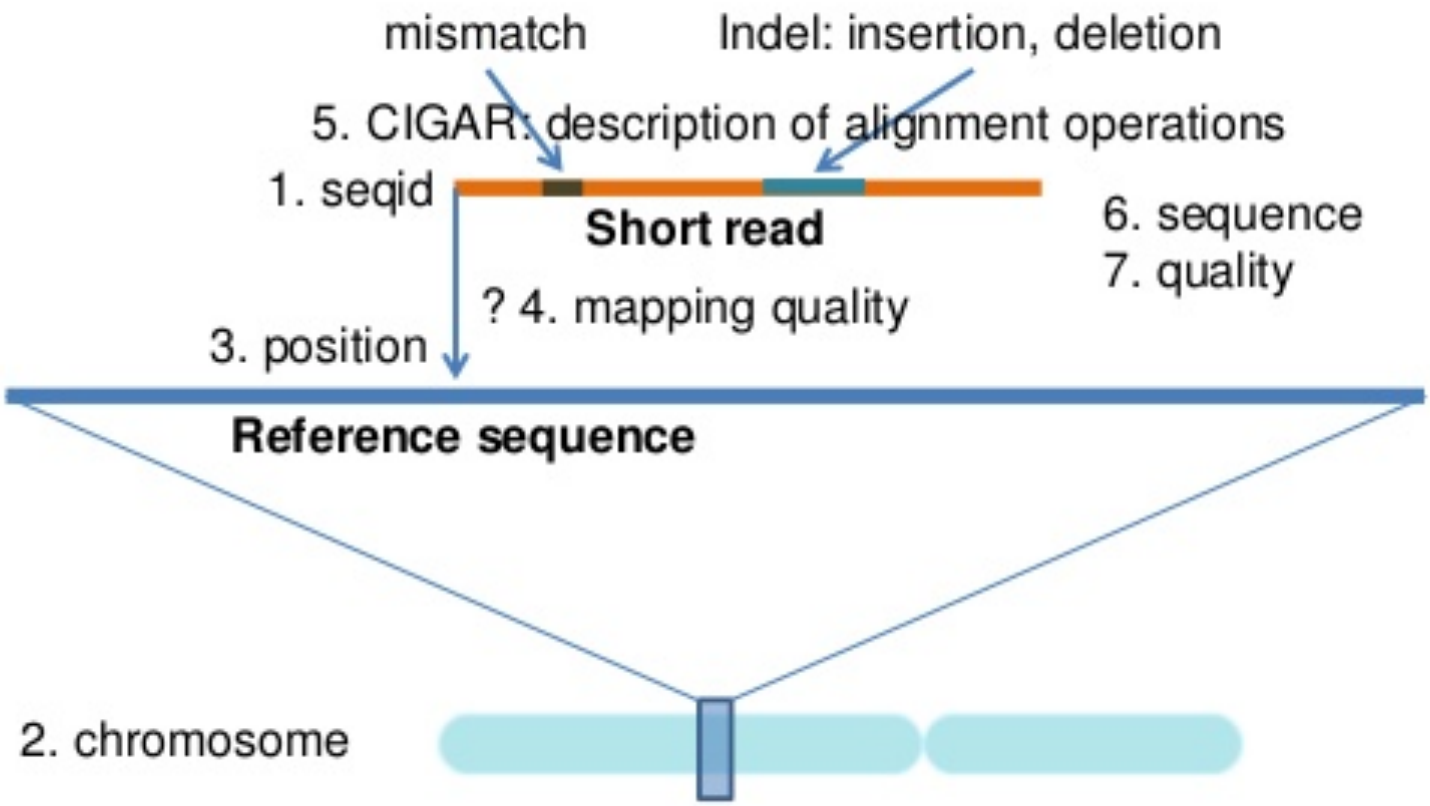
Illumina clonal expansion followed by image processing



Pile up sequences to



The SAM format



SAM format: what are sam/bam files

Headers

Alignments

```
1 @HD VN:1.0  SO:unsorted
2 @SQ SN:gi|110640213|ref|NC_008253.1|    LN:4938920
3 @PG ID:bowtie2  PN:bowtie2  VN:2.1.0
4 gi|110640213|ref|NC_008253.1|_418_952_1:0:0_1:0:0_0/1   0   gi|110640213|ref|NC_
  008253.1|   418 42  70M *   0   0   CCAGGCAGTGGCAGGTGGCCACCGTCCTCTCTGCCCCCGCCAAA
  ATCACCAACCATCTGGTAGCGATGAT  22222222222222222222222222222222222222222222222
  22222222222222222  AS:i:-3 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:8G61
  YT:Z:UU
5 gi|110640213|ref|NC_008253.1|_31_476_0:0:0_0:0:0_1/1    16  gi|110640213|ref|NC_
  008253.1|   407 42  70M *   0   0   GGAAAGCAATGCCAGGCAGGGGCAGGTGGCCACCGTCCTCTCTG
  CCCCCGCCAAAATCACCAACCATCTG  22222222222222222222222222222222222222222222222
  22222222222222222  AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:70 YT:Z
  :UU
6 gi|110640213|ref|NC_008253.1|_210_743_2:0:0_1:1:0_2/1   0   gi|110640213|ref|NC_
  008253.1|   210 42  70M *   0   0   CATTACCACCACCATCACCATTACCACAGGAAACGGTGCGGGCT
  GACGCGTACAGGAAACACCGAAAAAA  22222222222222222222222222222222222222222222222
  22222222222222222  AS:i:-6 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:30T31A7
      YT:Z:UU
```

Each row describes a single alignment of a raw read against the reference genome.
Each alignment has 11 mandatory fields, followed by any number of optional fields.

SAM FORMAT

SAM is TAB-delimited. Apart from the header lines, which are started with the '@' symbol, each alignment line consists of:

Col	Field	Description
1	QNAME	Query (pair) NAME
2	FLAG	bitwise FLAG
3	RNAME	Reference sequence NAME
4	POS	1-based leftmost POSition/coordinate of clipped sequence
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIAGR	extended CIGAR string
7	MRNM	Mate Reference sequence NaMe ('=' if same as RNAME)
8	MPOS	1-based Mate POSistion
9	ISIZE	Inferred insert SIZE
10	SEQ	query SEQuence on the same strand as the reference
11	QUAL	query QUALity (ASCII-33 gives the Phred base quality)
12	OPT	variable OPTional fields in the format TAG:VTYPE:VALUE

Each bit in the FLAG field is defined as:

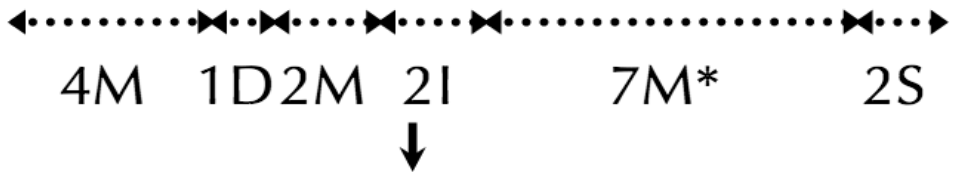
Flag	Description
0x0001	the read is paired in sequencing
0x0002	the read is mapped in a proper pair
0x0004	the query sequence itself is unmapped
0x0008	the mate is unmapped
0x0010	strand of the query (1 for reverse)
0x0020	strand of the mate
0x0040	the read is the first read in a pair
0x0080	the read is the second read in a pair
0x0100	the alignment is not primary
0x0200	the read fails platform/vendor quality checks
0x0400	the read is either a PCR or an optical duplicate

Meaning of fields: CIGAR

Operation	Meaning
M	Match*
D	Deletion w.r.t. the reference
I	Insertion w.r.t. the reference
N	Split or spliced alignment
S	Soft-clipping
H	Hard-clipping
P	Padding

Reference seq: ACCTGTC--TACCTTACG

Experimental seq: ACCT-TCCATACTTTATC



CIGAR string: 4M1D2M2I7M2S

Read CIGAR score

```
Alignment: 12345678901234  5678901234567890
Reference: AGCATGTTAGATAA**GATAGCTGTGCTAGTA
Read001+:        TTAGATAAAGGATA*CTG

Corresponding line in SAM file:
Read001  163   ref   7     30    8M2I4M1D3M  =     37
QNAME    FLAG  RNAME POS   MAPQ  CIGAR       MRNM  MPOS
```

CIGAR:

8M: first 8 bases of Read001 match reference

2I: then two insertions take place

4M: then again 4 matches

1D: then a deletion

3M: and finally again 3 matches

Read CIGAR score: A soft-clipped sequence is an unmatched fragment in a partially mapped read

```
Alignment: 12345678901234  5678901234567890
Reference: AGCATGTTAGATAA**GATAGCTGTGCTAGTA
Read002+:      aaaAGATAA*GGATA

Corresponding SAM file:
Read002  0     ref   9     30    3S6M1P1I4M  *     0
QNAME    FLAG  RNAME POS   MAPQ  CIGAR       MRNM  MPOS
```

CIGAR:

3S: first three bases are soft-clipped (= unaligned)

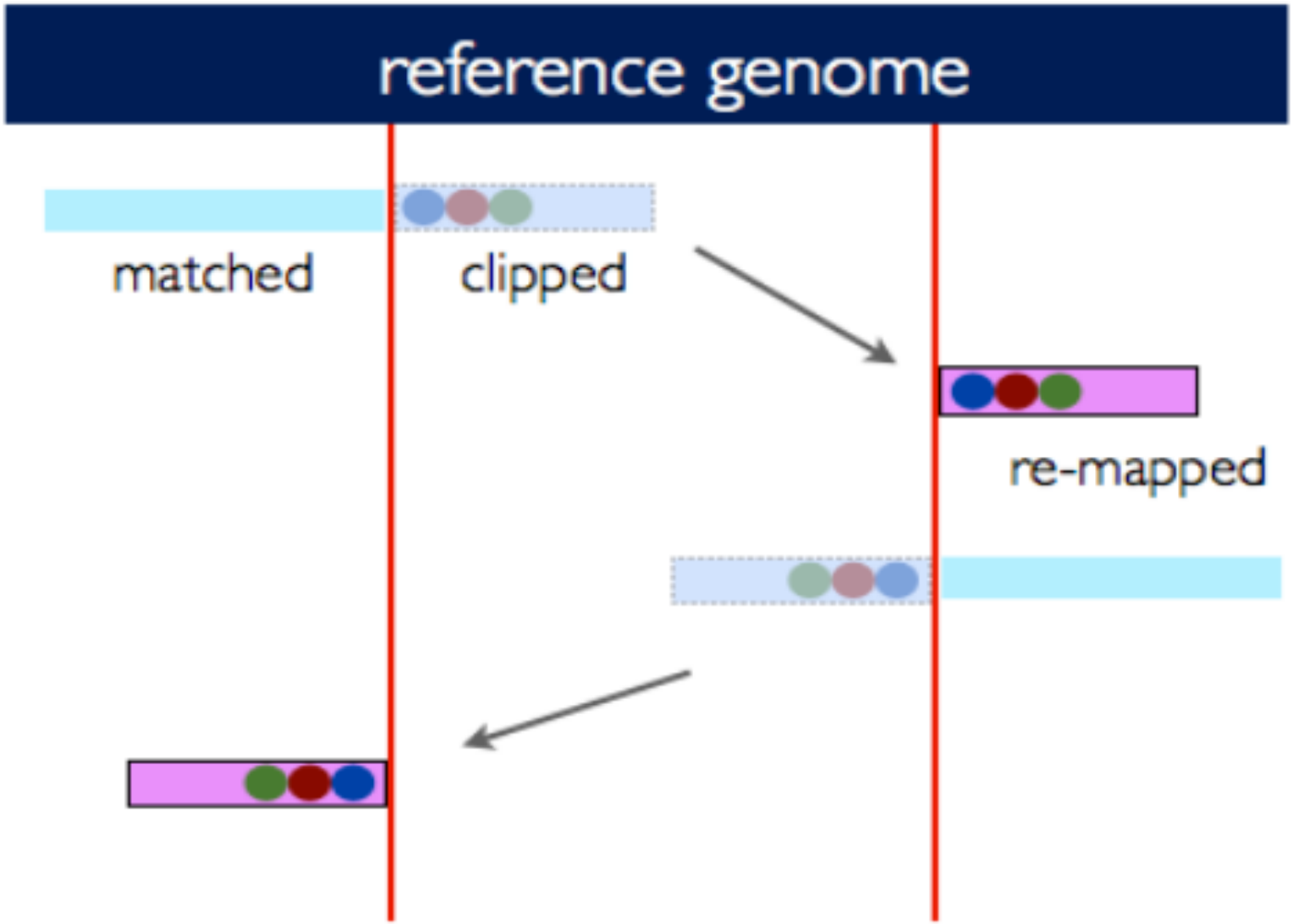
6M: next 6 bases of Read002 match reference

1P: then a padding (= addition of a gap to correctly align the rest of Read002)

1I: then an insertion takes place

4M: and finally again 4 matches

Gapped alignment calls



Read CIGAR score

```
Alignment: 12345678901234  5678901234567890
Reference: AGCATGTTAGATAA**GATAGCTGTGCTAGTA
Read003+:     gcctaAGATAA

Corresponding SAM file:
Read003  0     ref   9     30    5H6M  *     0
QNAME    FLAG  RNAME POS   MAPQ  CIGAR MRNM  MPOS
```

CIGAR:

5H: first five bases are aligned on reverse strand so impossible (= hard clipping)

6M: next 6 bases of Read003 match reference

Pair read analysis

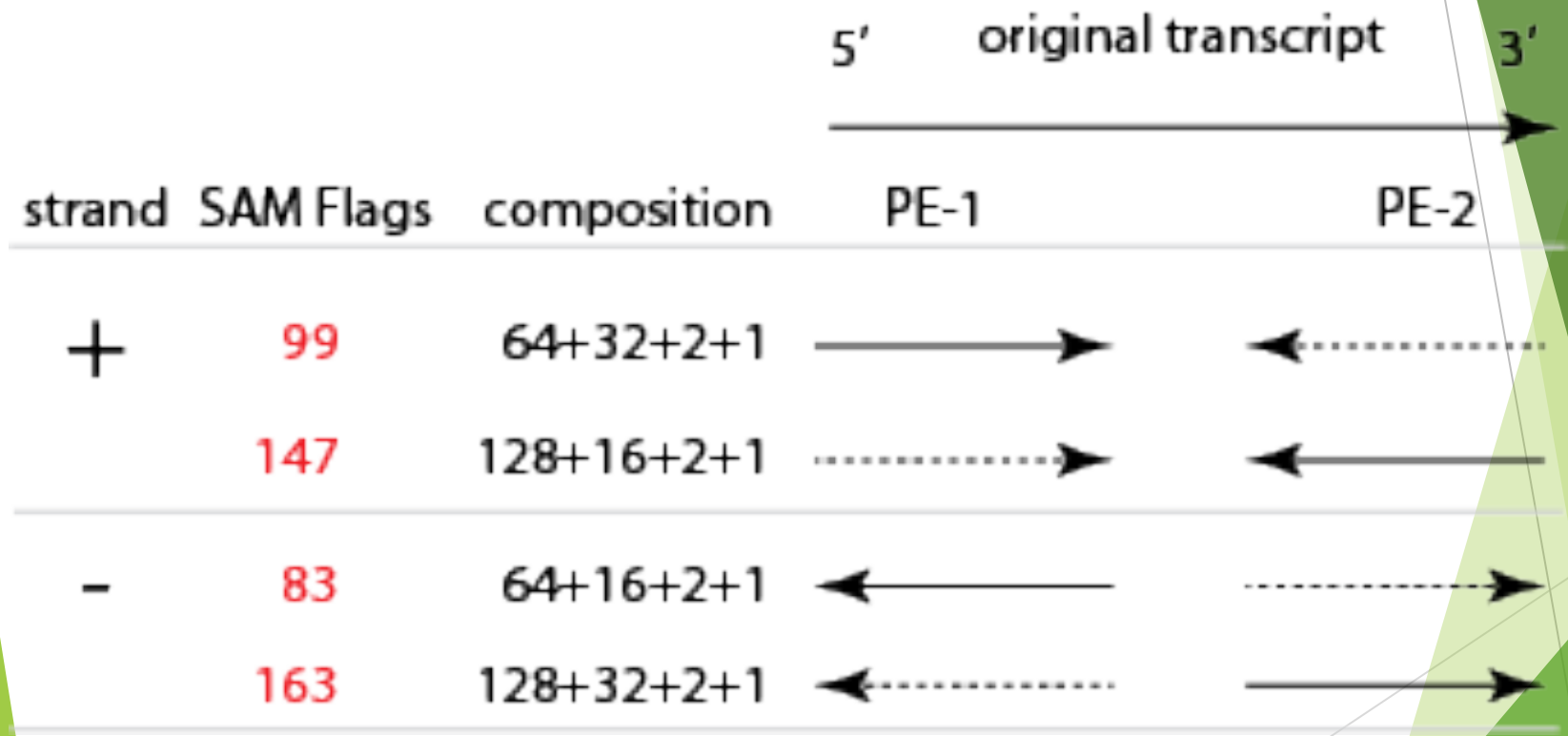
In a chromosome of a parasite genome

	Flag 1	Flag 2	Count	%	Average	Median	STD	Min	Max
← ←	65	129	4	0.000	278849	289087	262174.88	74557	703194
← ←	67	131	4	0.000	109	97	59.98	71	210
← →	81	161	224	0.001	18534.46	53	90016.41	28	1005063
← →	83	163	542	0.003	77.74	65	53.13	4	293
→ ←	97	145	1789	0.009	2320.61	410	29877.06	30	680974
→ ←	**99**	**147**	**99481**	**0.482**	**275.29**	**295**	**79.71**	**61**	**401**
→ →	113	177	7	0.000	306645.43	299601	182414.84	189196	681374
→ →	115	179	4	0.000	141.25	203	98.6	102	259
← ←	129	65	10	0.000	278402.3	237121	198856.09	128485	656117
→ →	131	67	6	0.000	194.67	178	79.93	137	321
← →	145	97	773	0.004	5837.39	52	52533.28	15	903807
← →	147	99	1128	0.005	73.06	68	34.43	4	286
→ ←	161	81	2286	0.011	1823.95	407	21527.69	15	597483
→ ←	**163**	**83**	**100010**	**0.485**	**273.92**	**295**	**80.98**	**59**	**401**
← ←	177	113	7	0.000	170902.43	102523	149875.07	44144	431897
← ←	179	115	12	0.000	221	255	108.48	92	378

Only 99, 147 and 163, 83 are properly mapped read pairs within a defined insert size

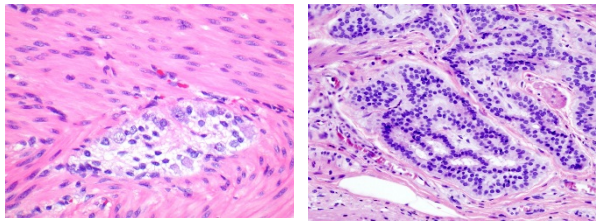
Single reads are not shown

Binary numbers: flags to orientation



RNA sequencing: abundance comparisons between two or more conditions / phenotypes

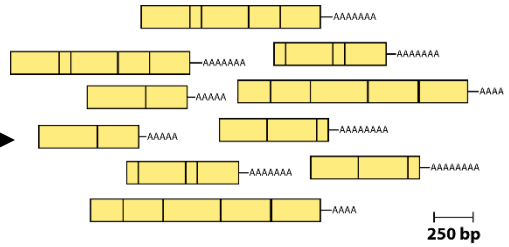
Samples of interest



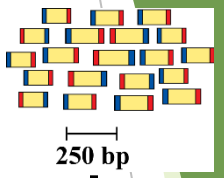
Condition 1 (normal tissue)

Condition 2 (diseased tissue)

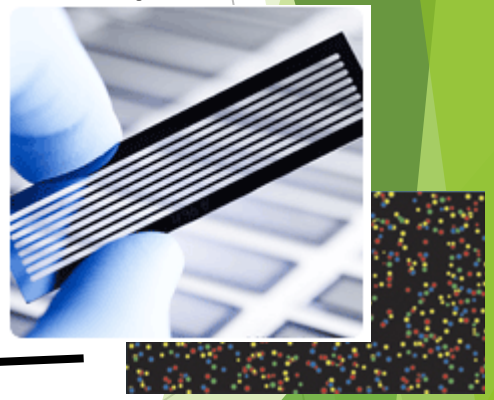
Isolate RNAs



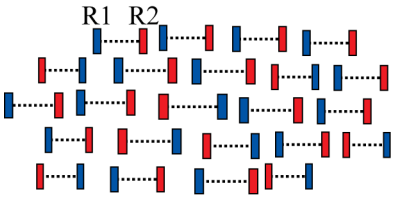
Generate cDNA, fragment, size select, add linkers



Sequence ends



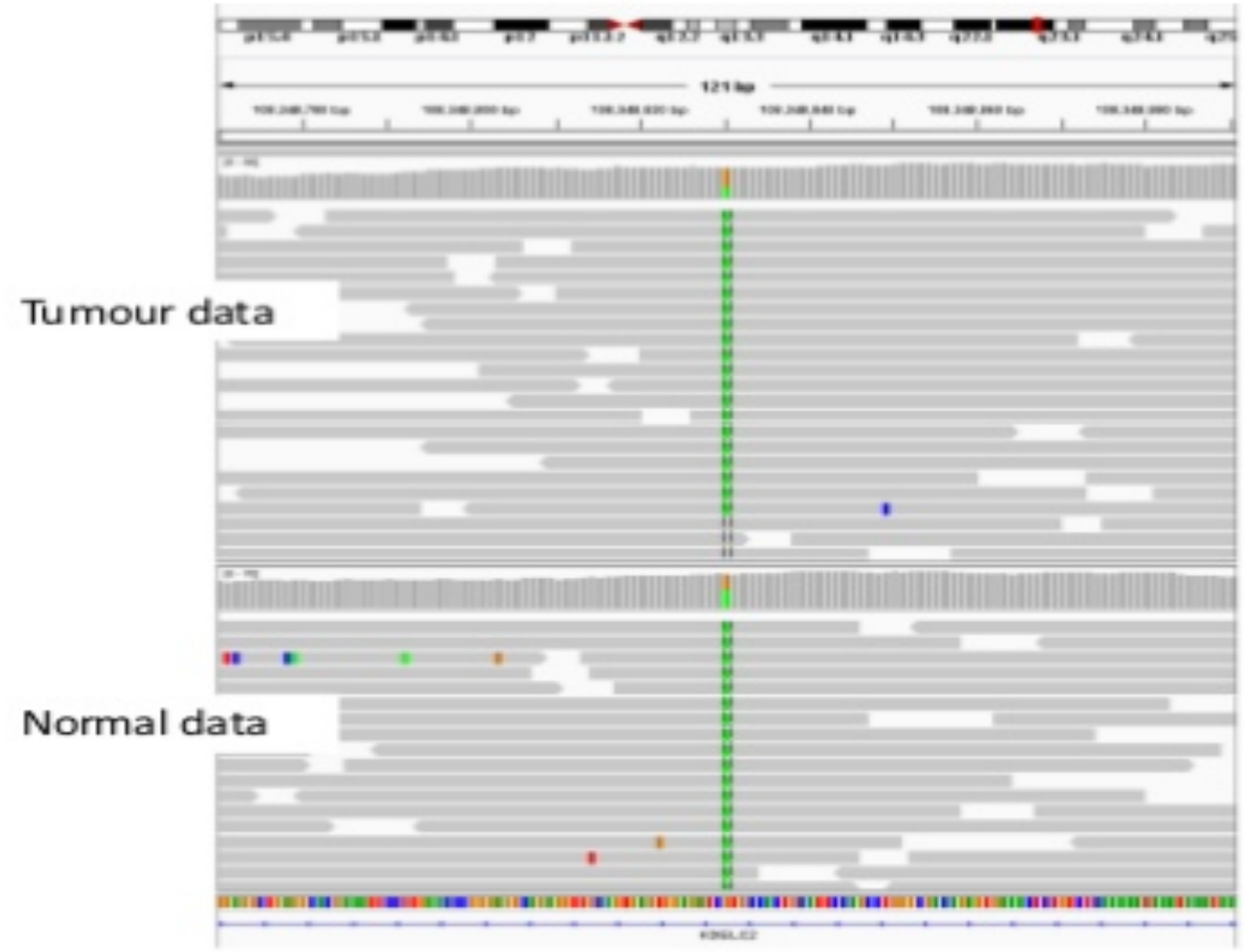
100s of millions of paired reads
10s of billions bases of sequence



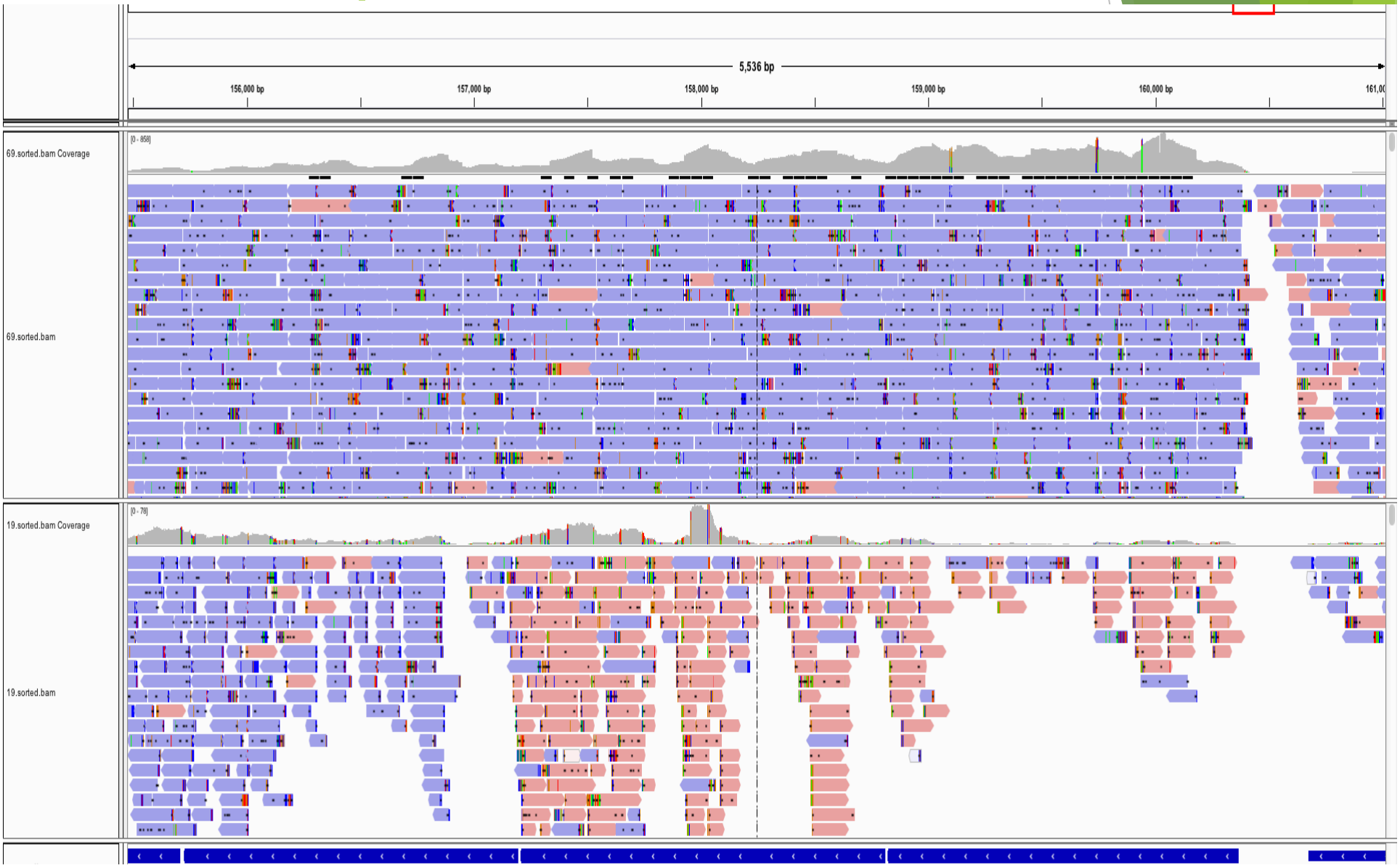
Map to genome, transcriptome, and predicted exon junctions

Downstream analysis

Compare two samples for



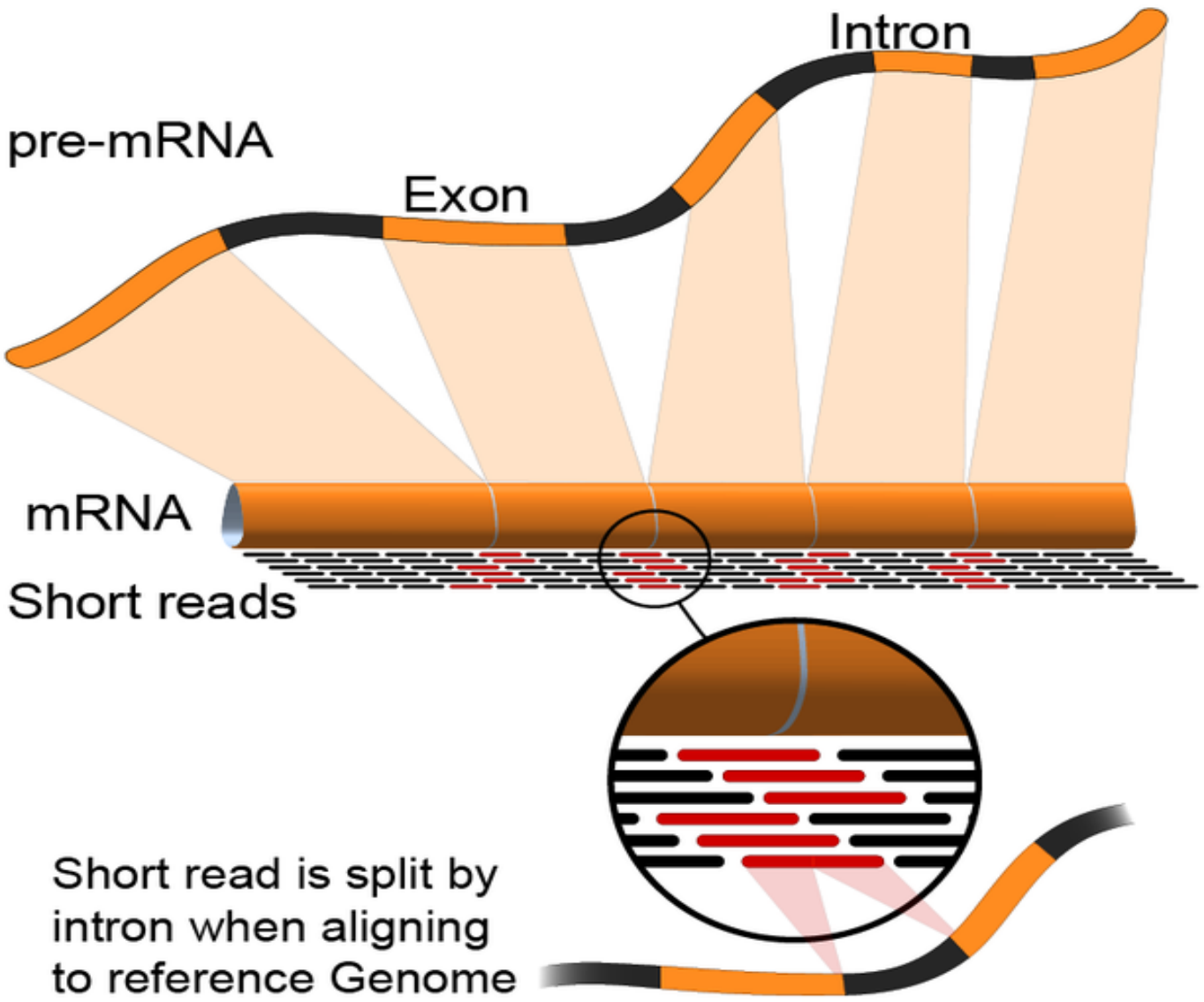
Transcript abundances differ

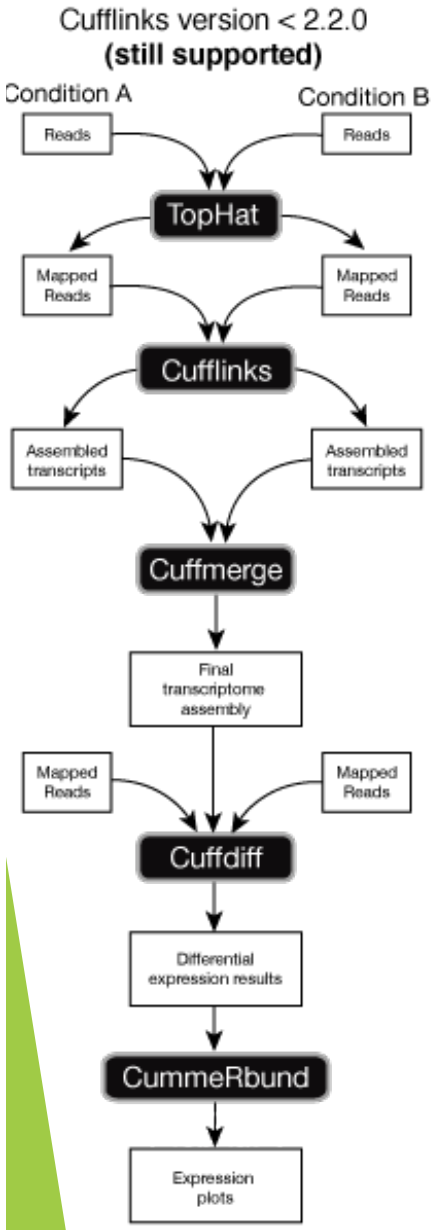


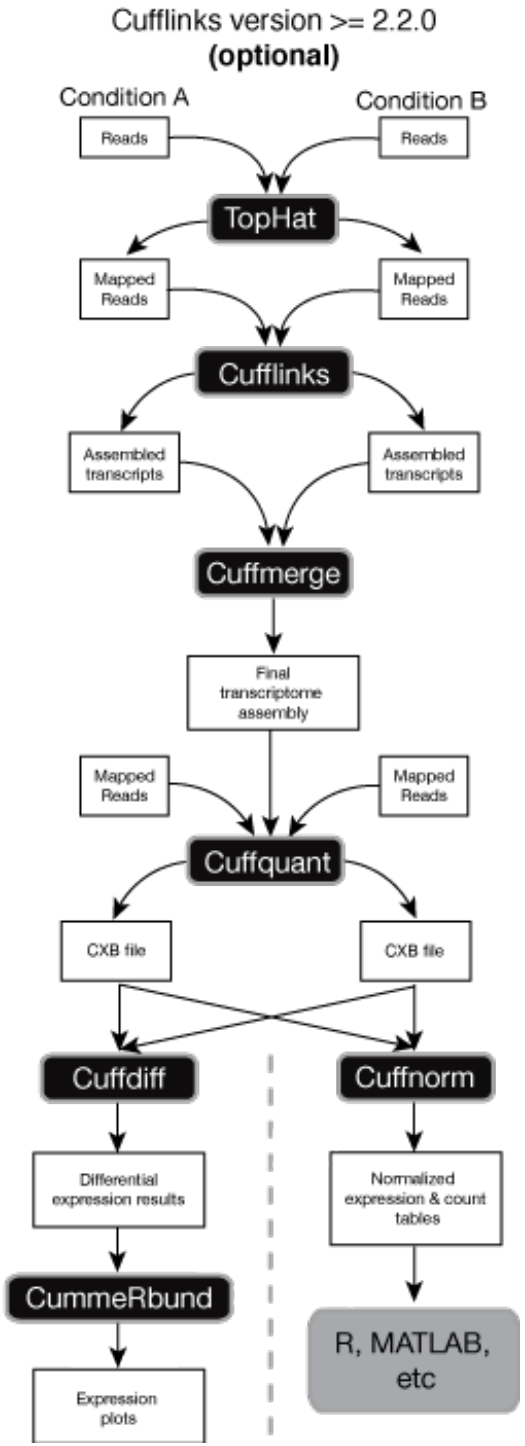
Genes have ‘structure’, solve by mapping

- This leads to for example analysis of intron-exon structure

Genes and transcripts







Currrent paradigm:

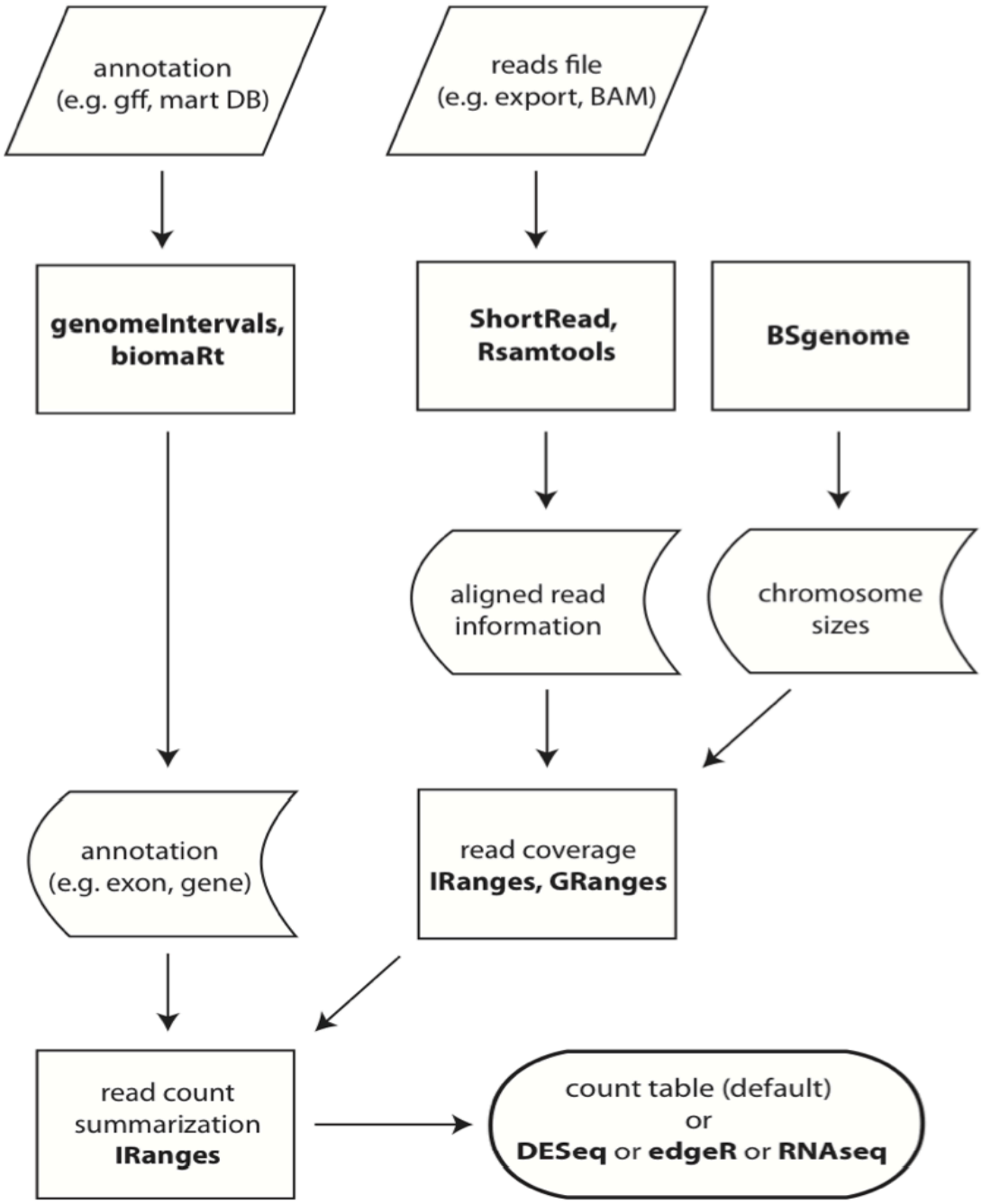
“cuff-suit”

Common analysis goals of RNA-Seq analysis (what can you ask of the data?)

- Gene expression and differential expression
- Alternative expression analysis
- Transcript discovery and annotation
- Allele specific expression
  - Relating to SNPs or mutations
- Mutation discovery
- Fusion detection
- RNA editing

Back to the demo

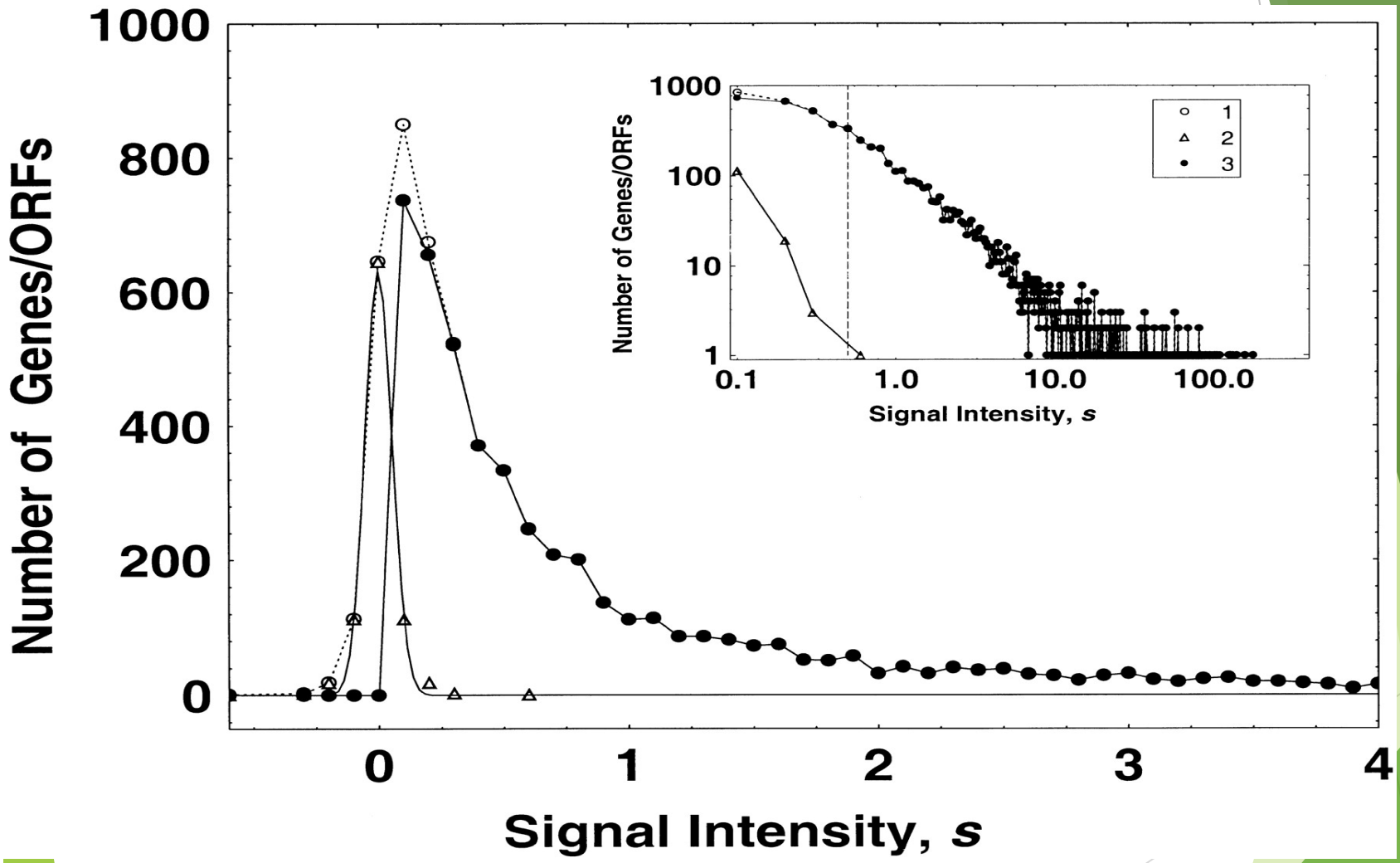
- Introduction to RNA sequencing
- Rationale for RNA sequencing (versus DNA sequencing)
- Hands on tutorial


annotation
(e.g. gff, mart DB)
reads file
(e.g. export, BAM)
genomeIntervals,
biomaRt
ShortRead,
Rsamtools
BSgenome
aligned read
information
chromosome
sizes
annotation
(e.g. exon, gene)
read coverage
IRanges, GRanges
read count
summarization
IRanges
count table (default)
or
DESeq or edgeR or RNAseq

Deseq and DEseq2

- method based on the negative binomial distribution, with variance and mean linked by local regression
- DEseq2:
- No demo scripts available yet:
- http://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.pdf

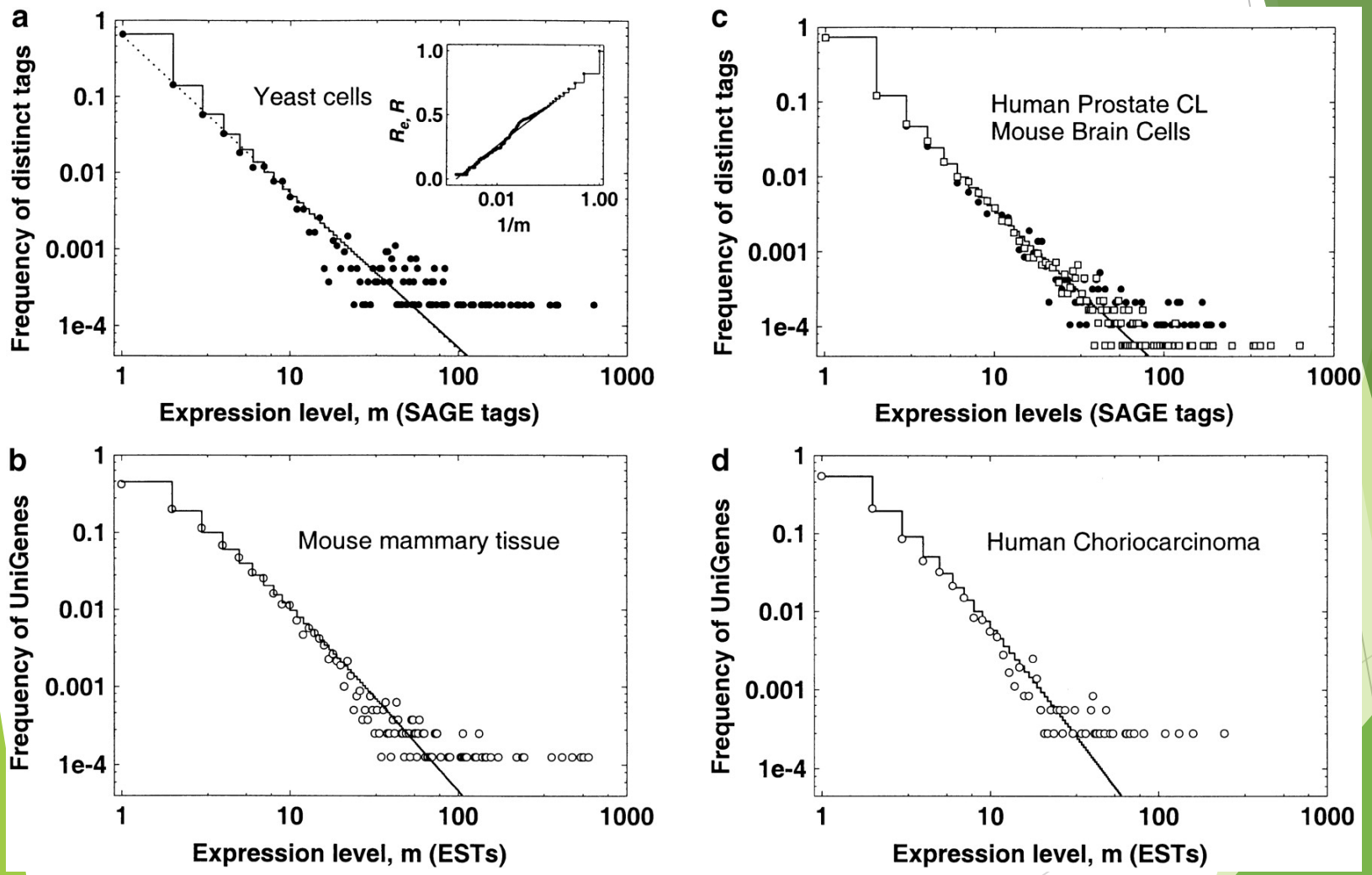
The empirical frequency distribution of the hybridization signal intensity values for Affymetrix microarray hybridization data for normal yeast cell genes/ORFs (Jelinsky and Samson 1999).



Kuznetsov V A et al. Genetics 2002;161:1321-1332

GENETICS

Empirical relative frequency distributions of the gene expression levels.



Kuznetsov V A et al. Genetics 2002;161:1321-1332

GENETICS

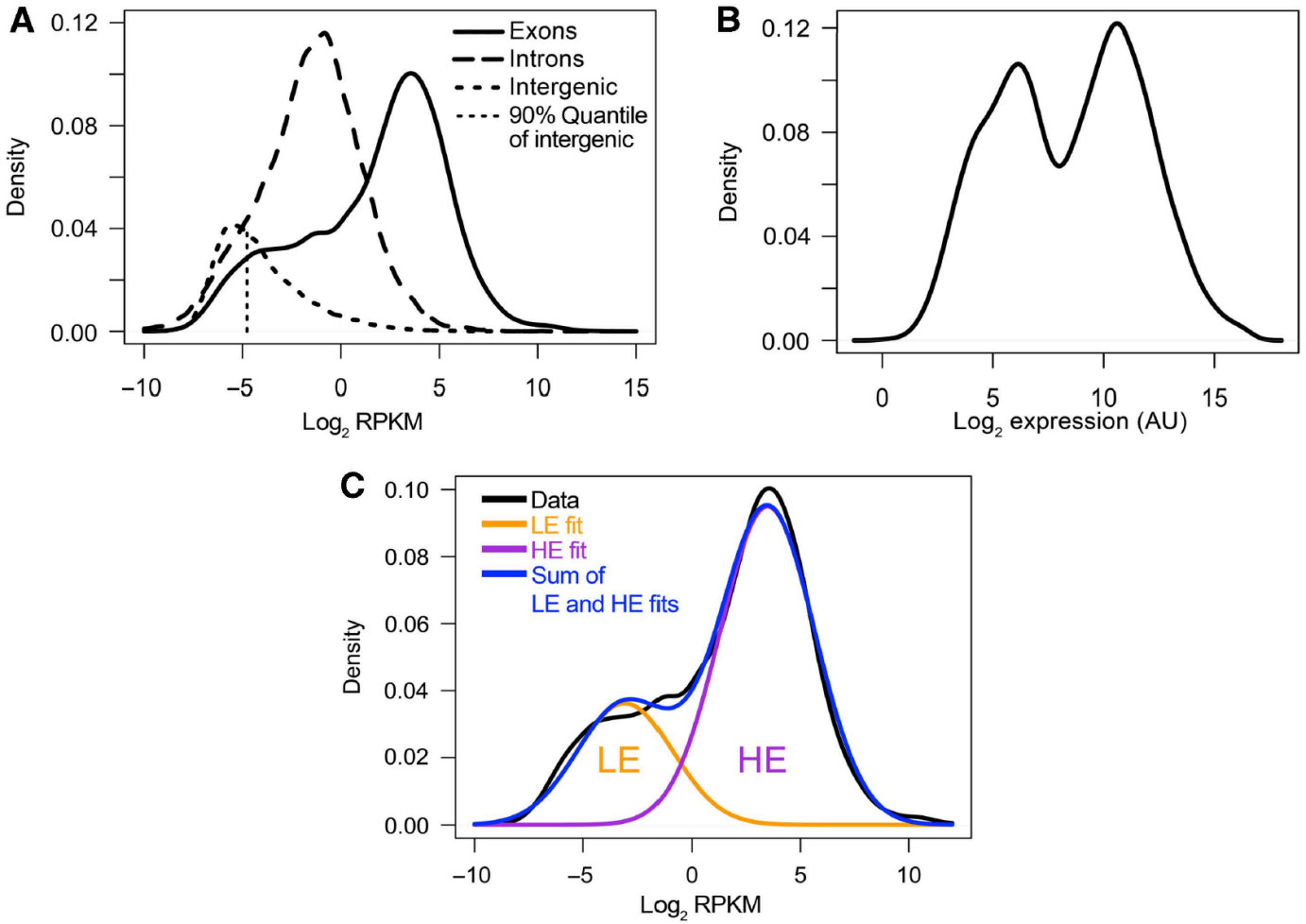
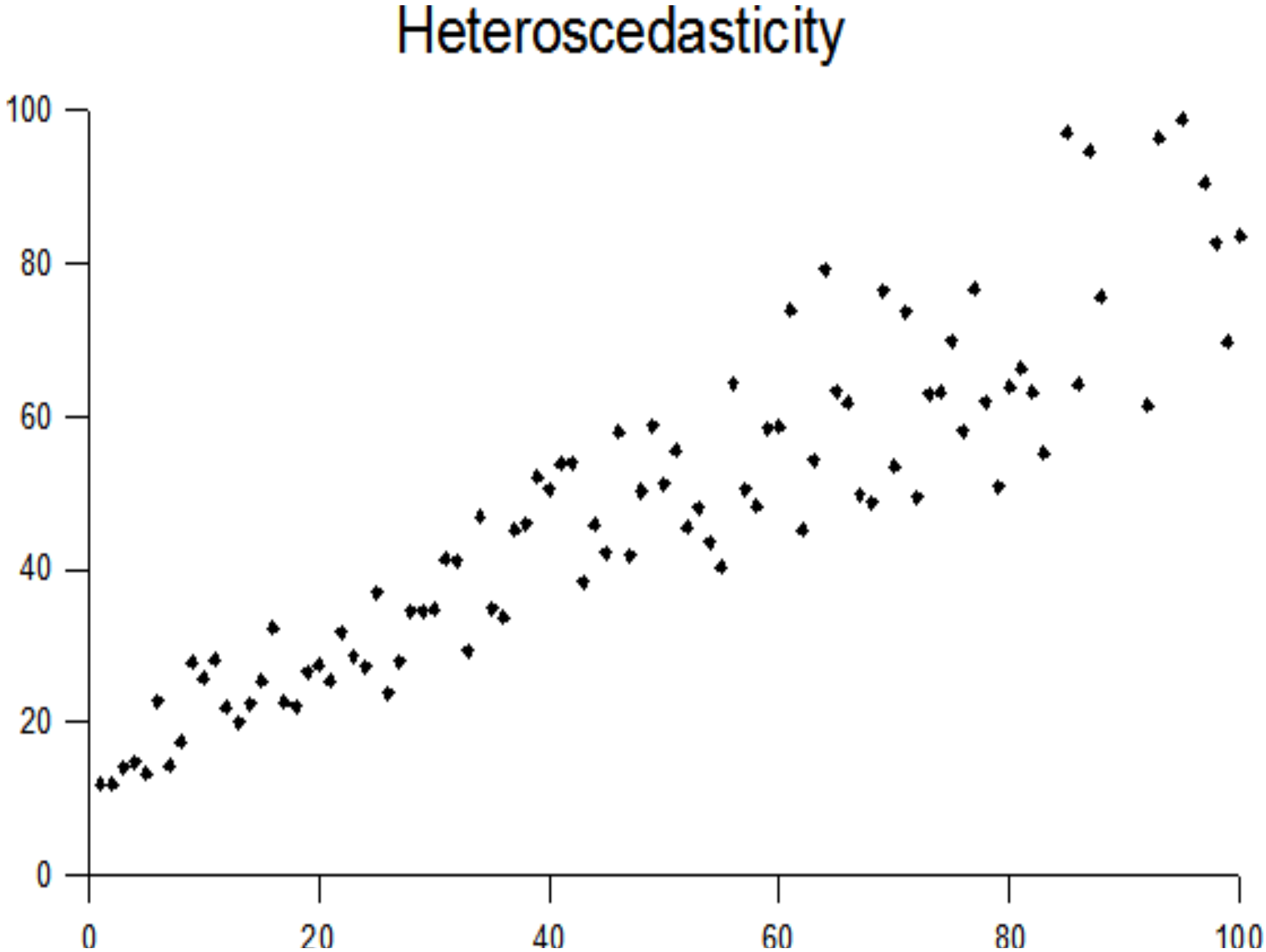
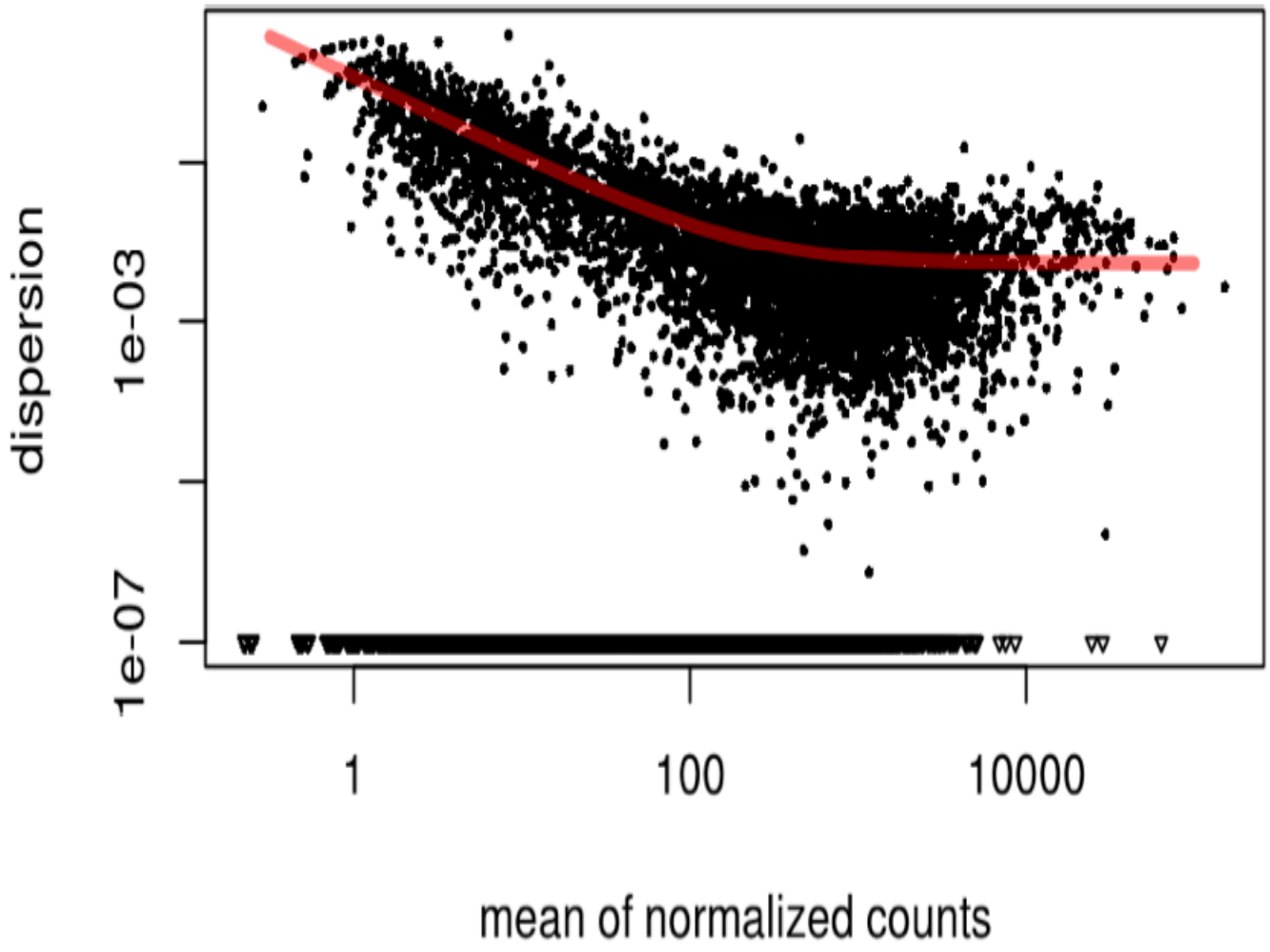


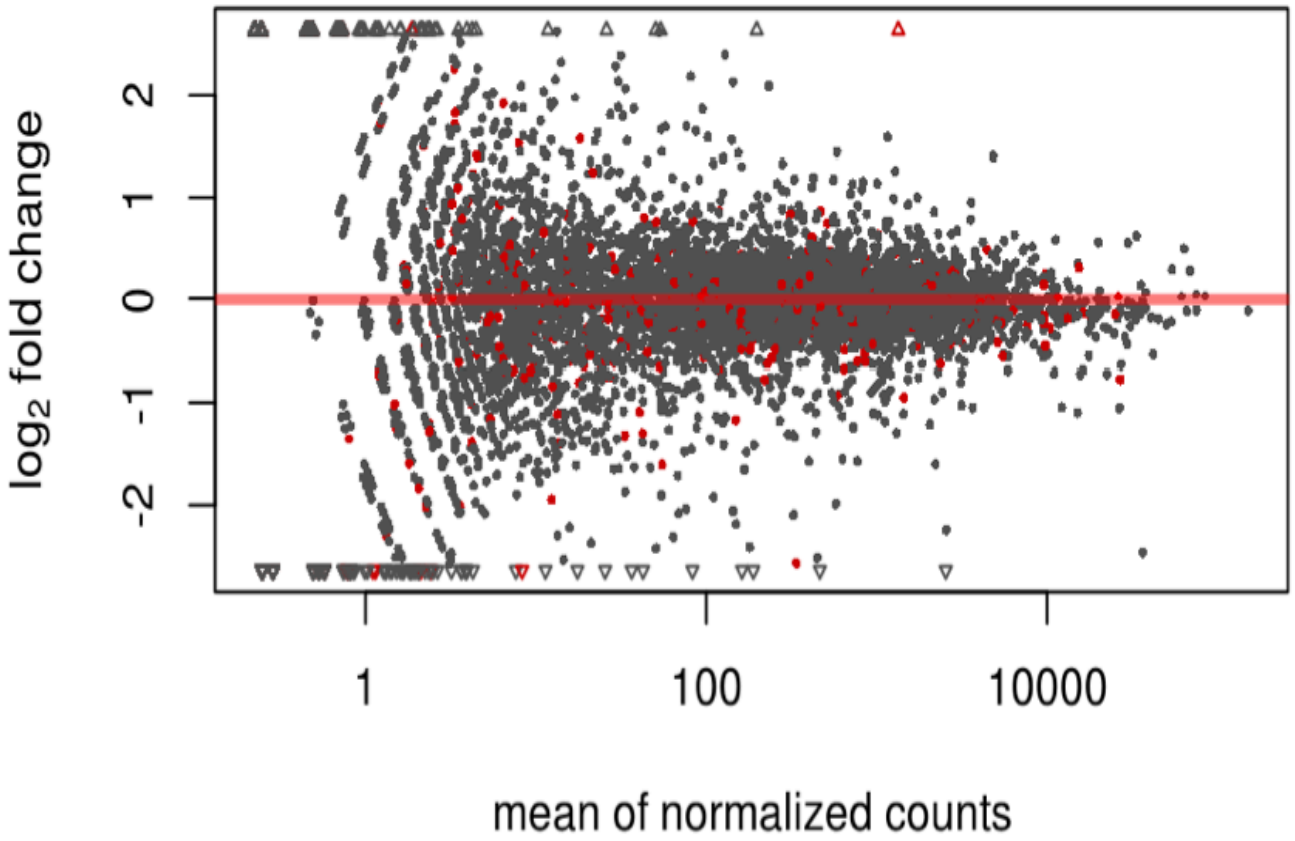
Figure 1 Distribution of gene expression levels. (**A**) Kernel density estimates of RPKM distributions of RNA-seq data within exons, introns and intergenic regions as indicated. The fragments used to estimate intron and intergenic RPKM were based on randomizations using the same length distribution as the exonic parts of genes. The 90% quantile of the intergenic distribution is indicated. (**B**) Kernel density estimate of expression level distribution of microarray data (Wei *et al*, 2009). (**C**) Expectation-maximization-based curve fitting of RNA-seq data of A.


Heteroscedasticity
100
80
60
40
20
0
0
20
40
60
80
100

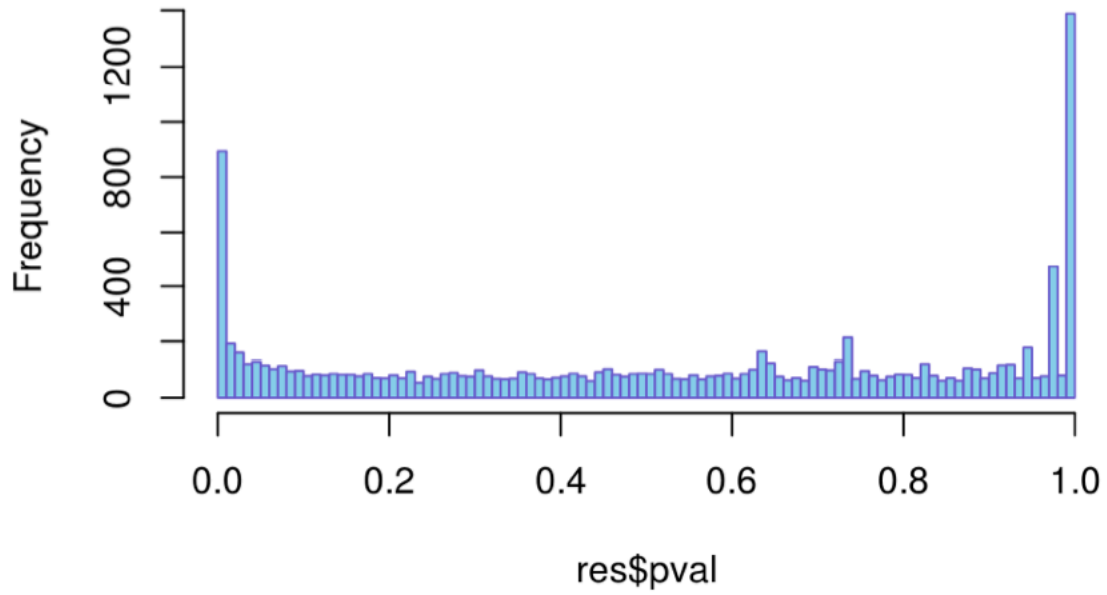
Empirical (black dots) and fitted (red lines) dispersion values plotted against the mean of the normalised counts.



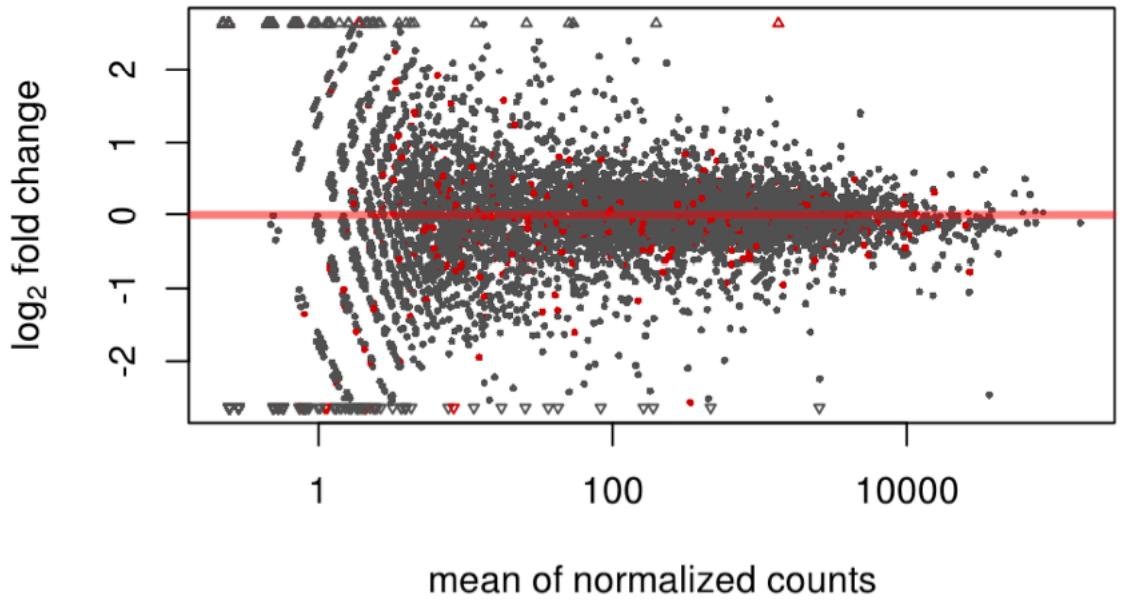
Plot of normalised mean versus log2 fold change for the contrast untreated versus treated.



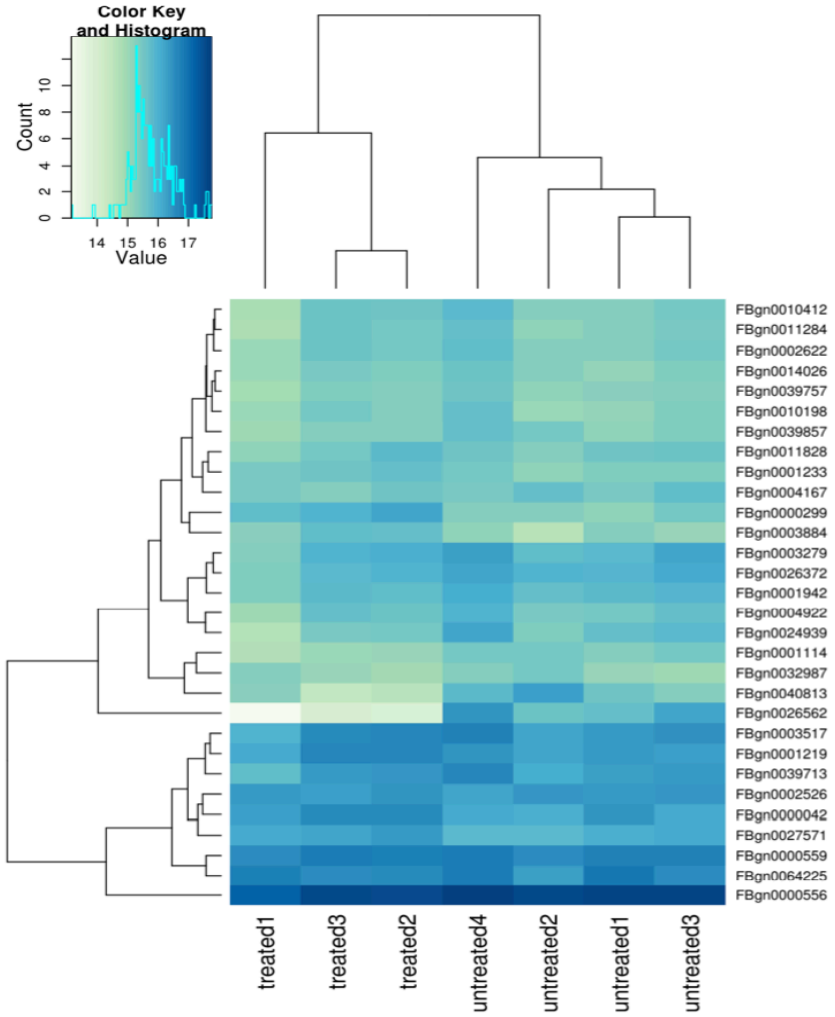
Histogram of p-values from the call to nbinomTest.

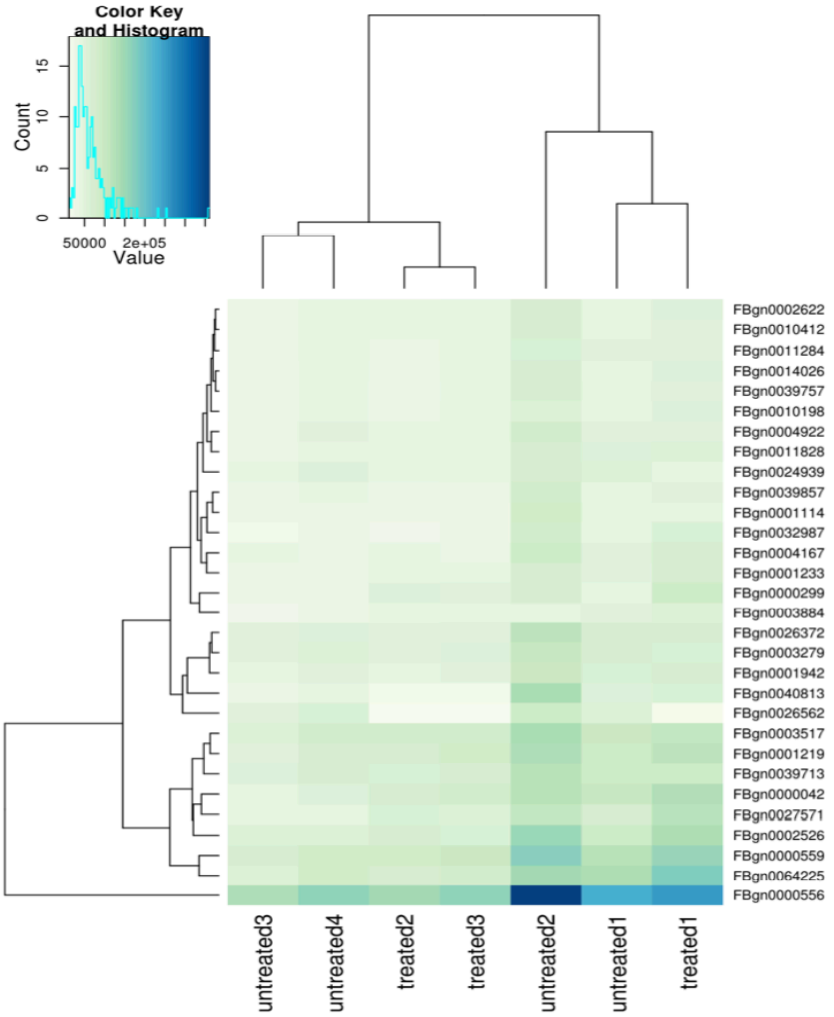


MvA plot for the contrast“treated”vs.“untreated”, using two treated and only one untreated

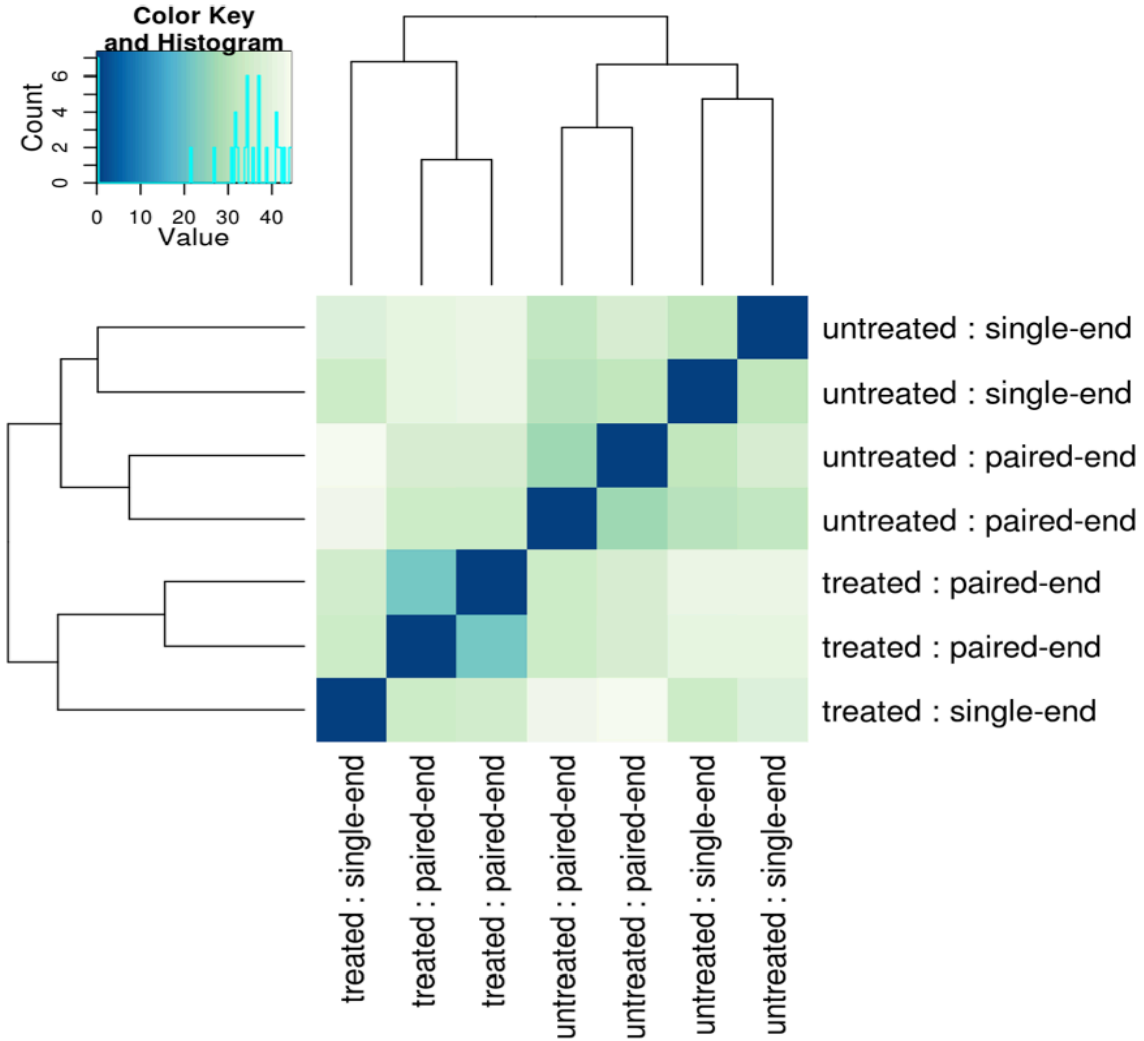


Heatmaps showing the expression

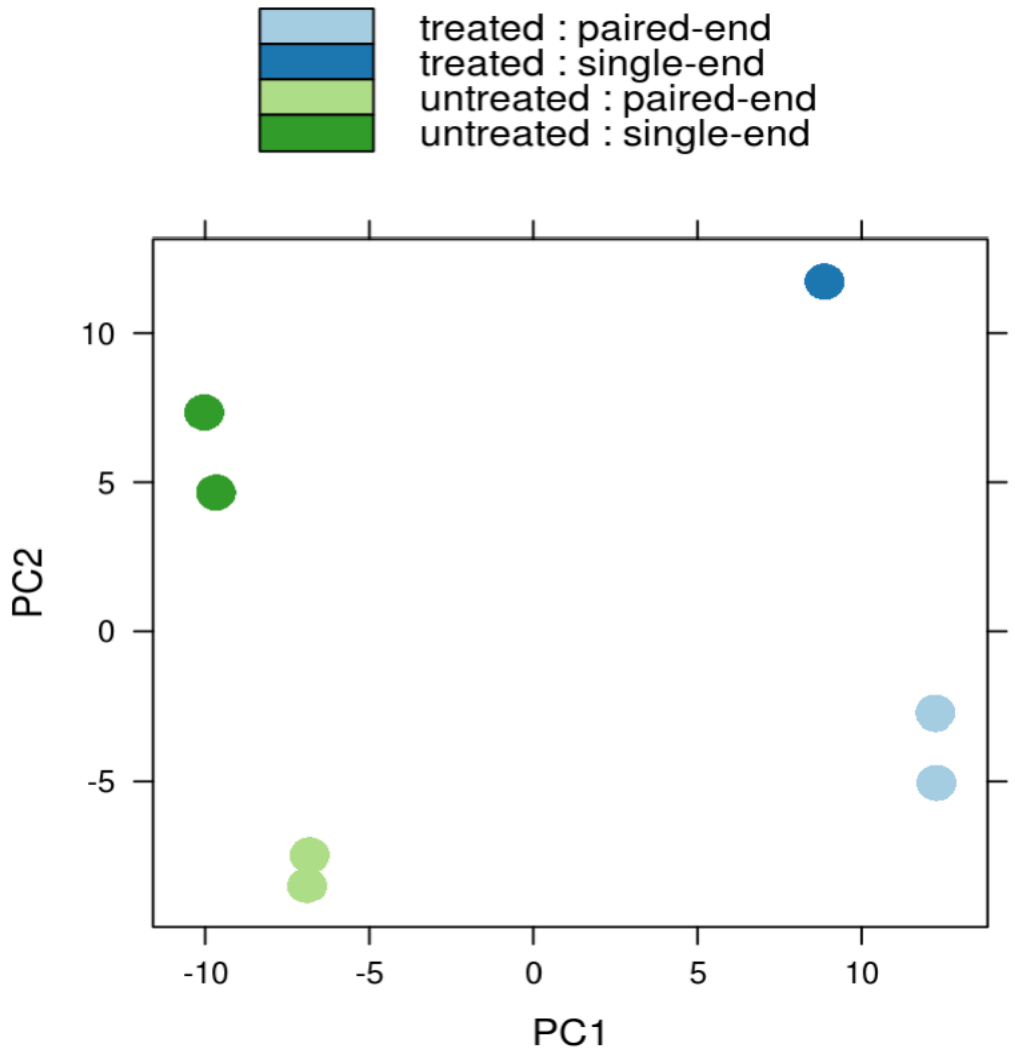




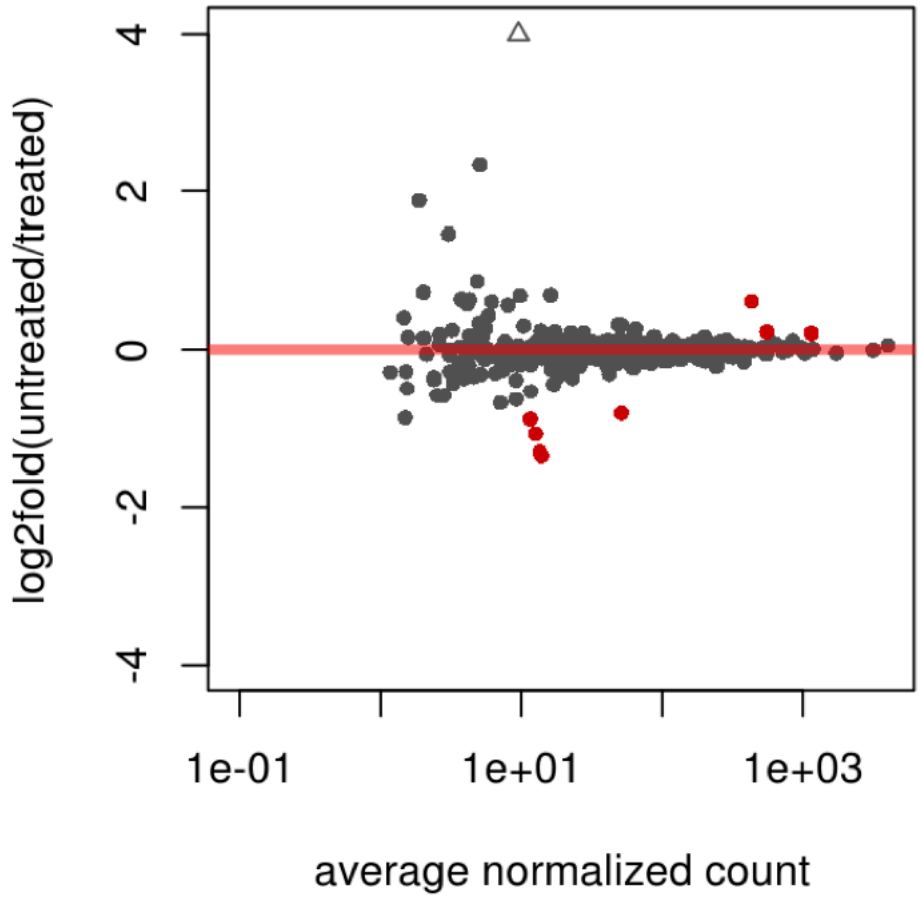
Heatmap showing the Euclidean distances between the samples as calculated from the variance stabilising transformation of the count data.



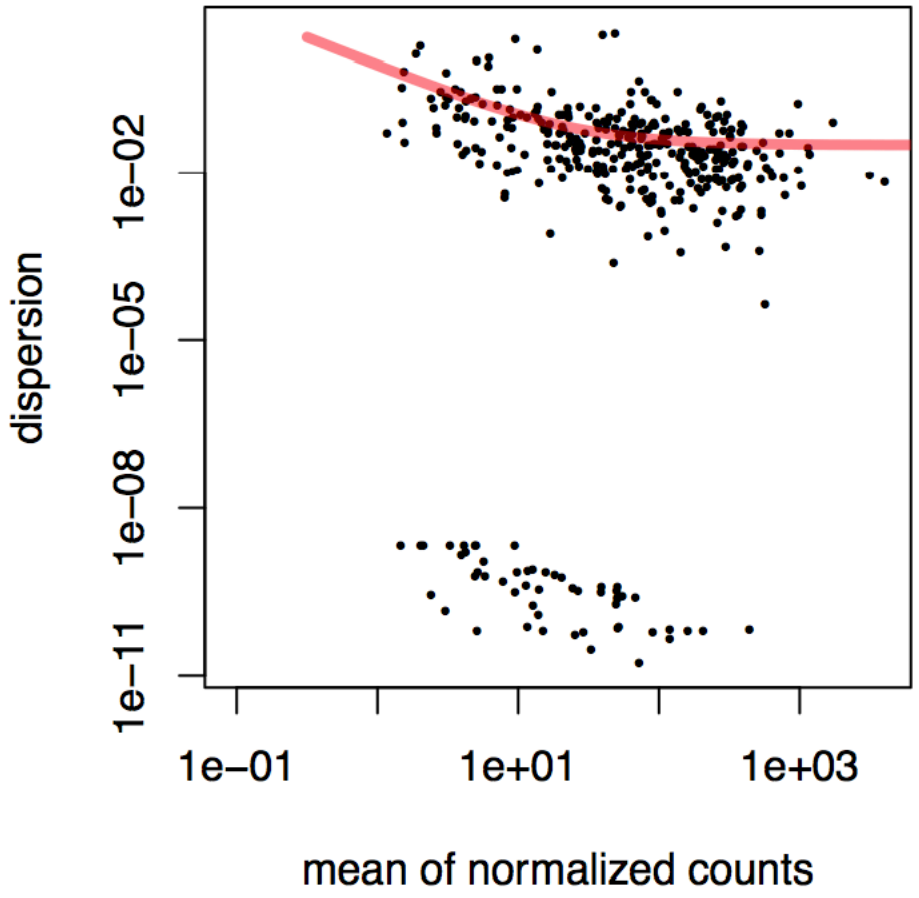
Biological effects of condition and libType



Mean expression versus log2 fold change plot. Significant hits (at p



Per-gene dispersion estimates (shown by points) and the fitted mean-



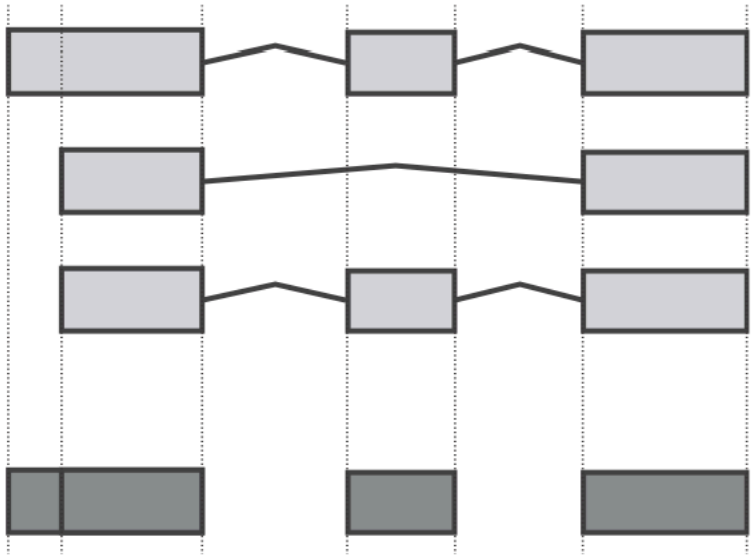
Differential exon usage

- Detecting spliced isoform usage by exon-level expression analysis

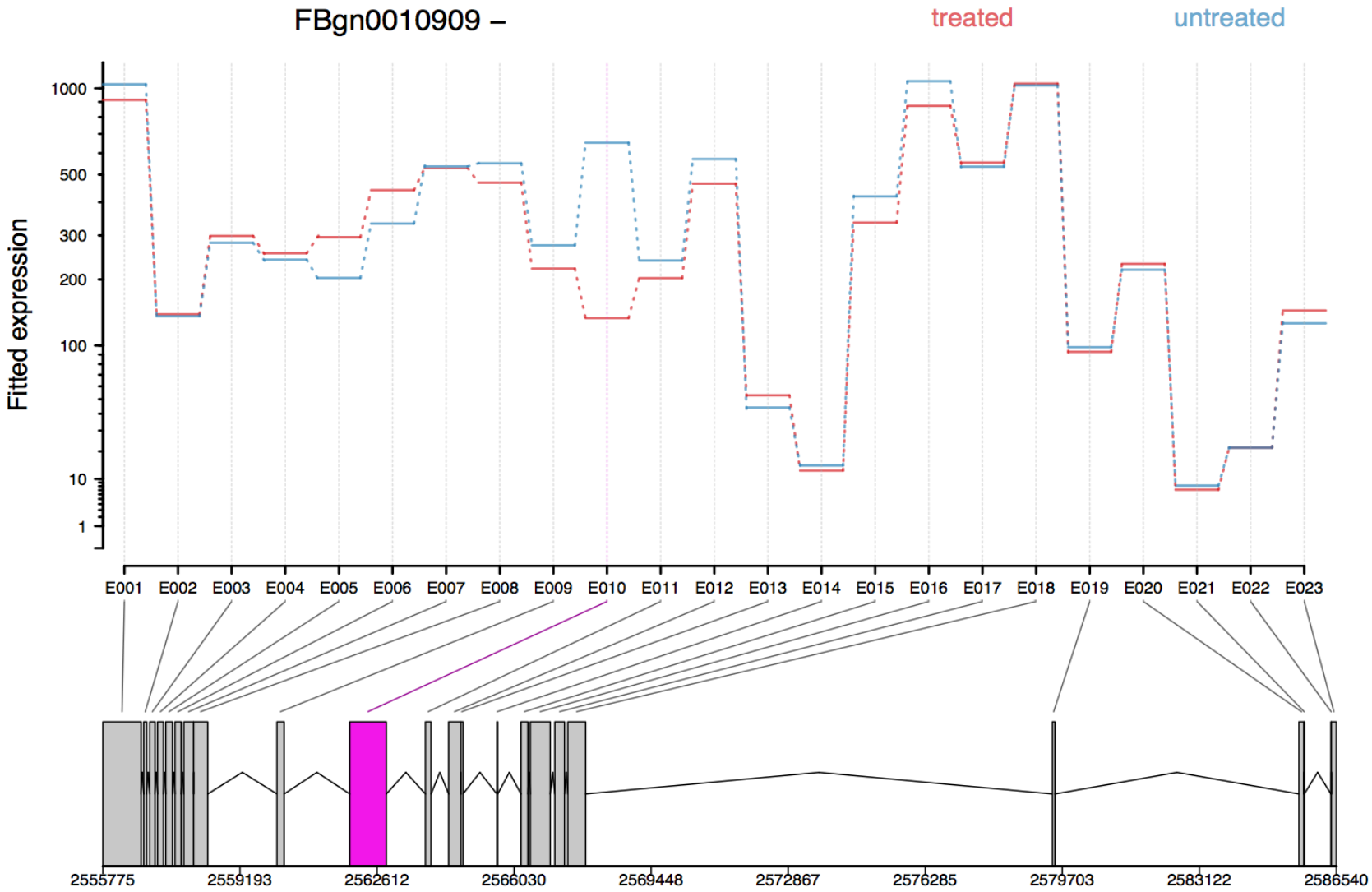
pasilla data

The splicing factor pasilla (NOVA1 and NOVA2 ortholog) was knocked-down in *Drosophila* cell cultures.

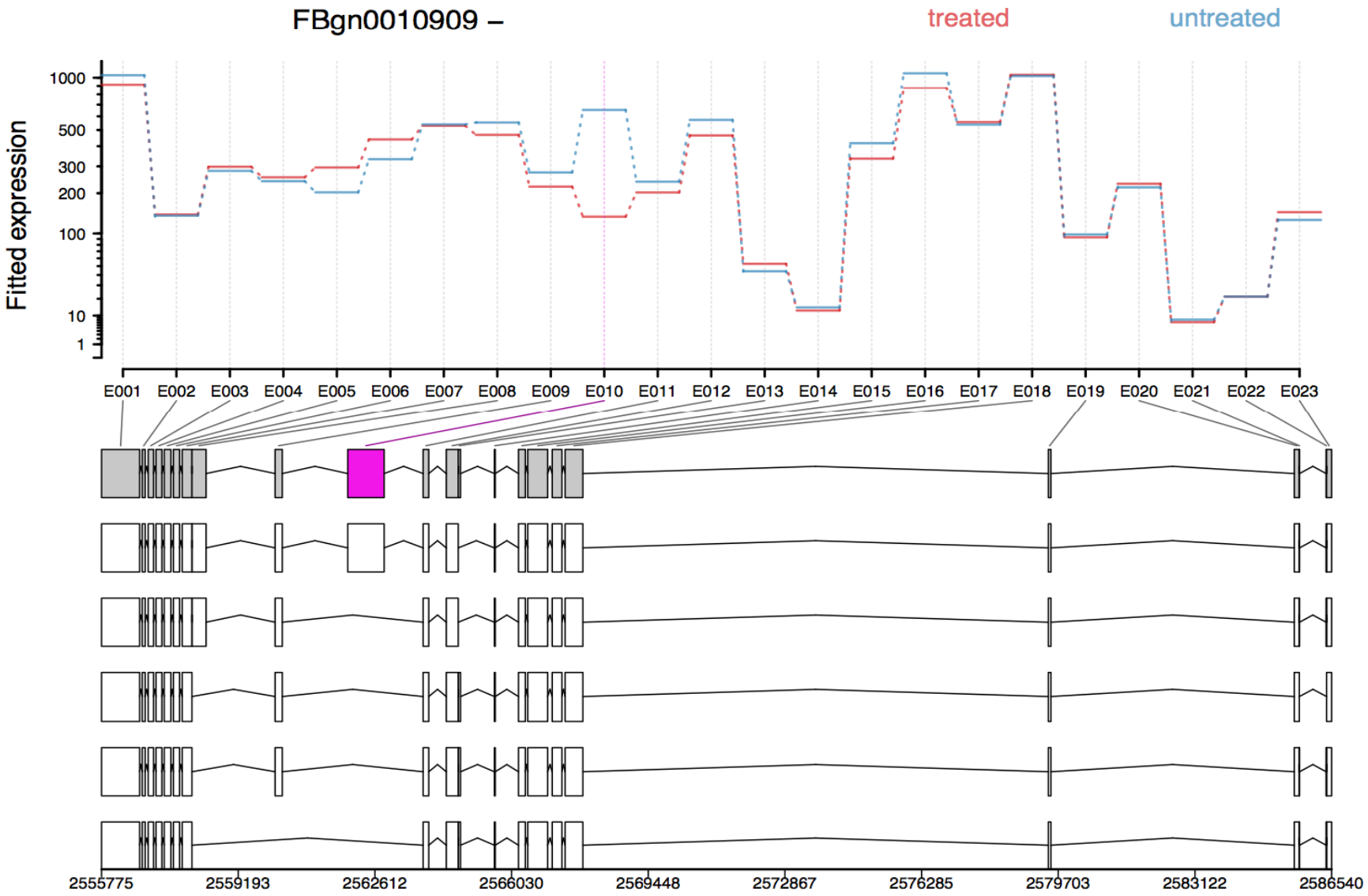
Types of splicing



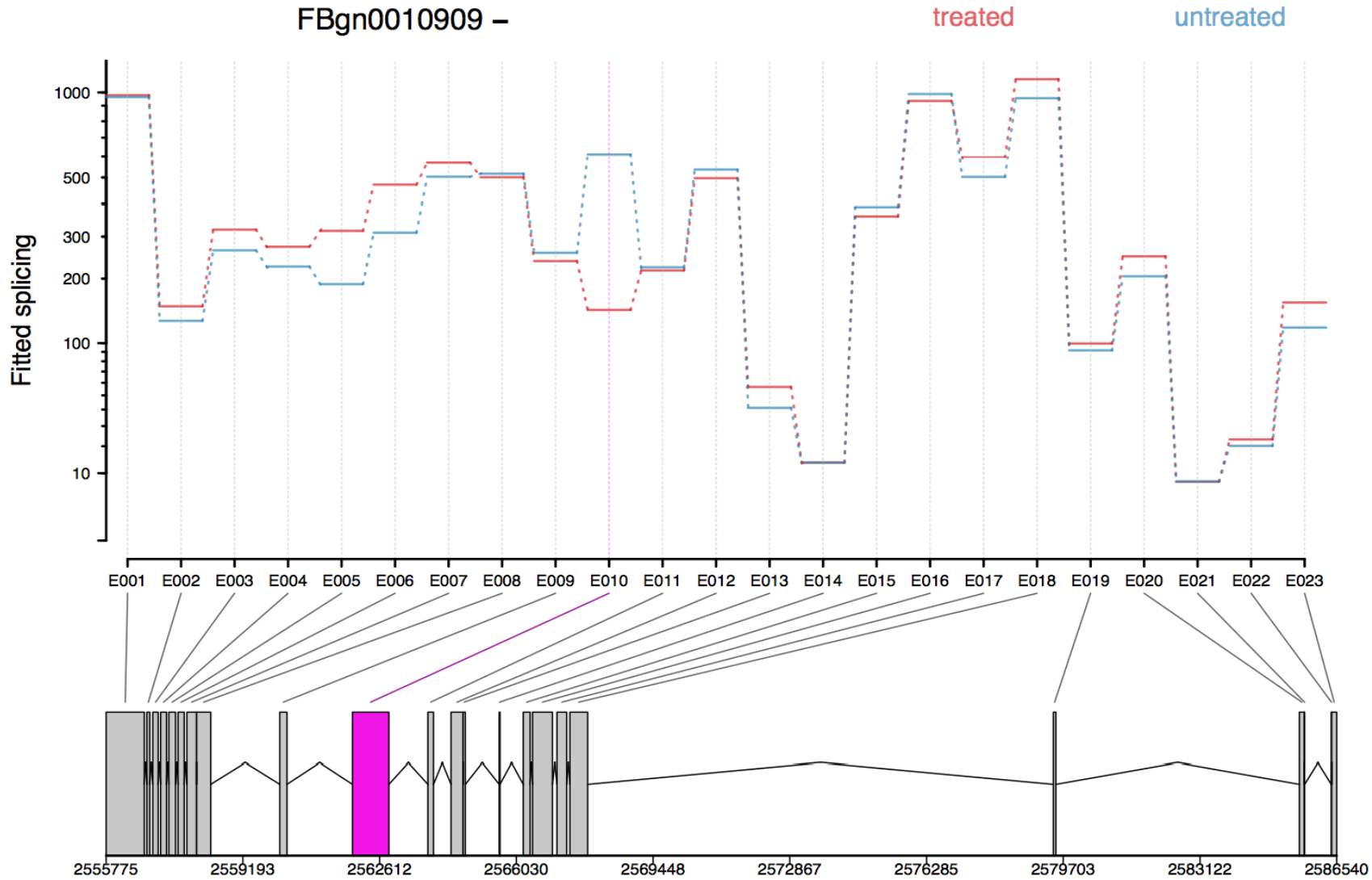
expression estimates from a call to testForDEU. Shown in red is the exon that showed significant differential exon usage



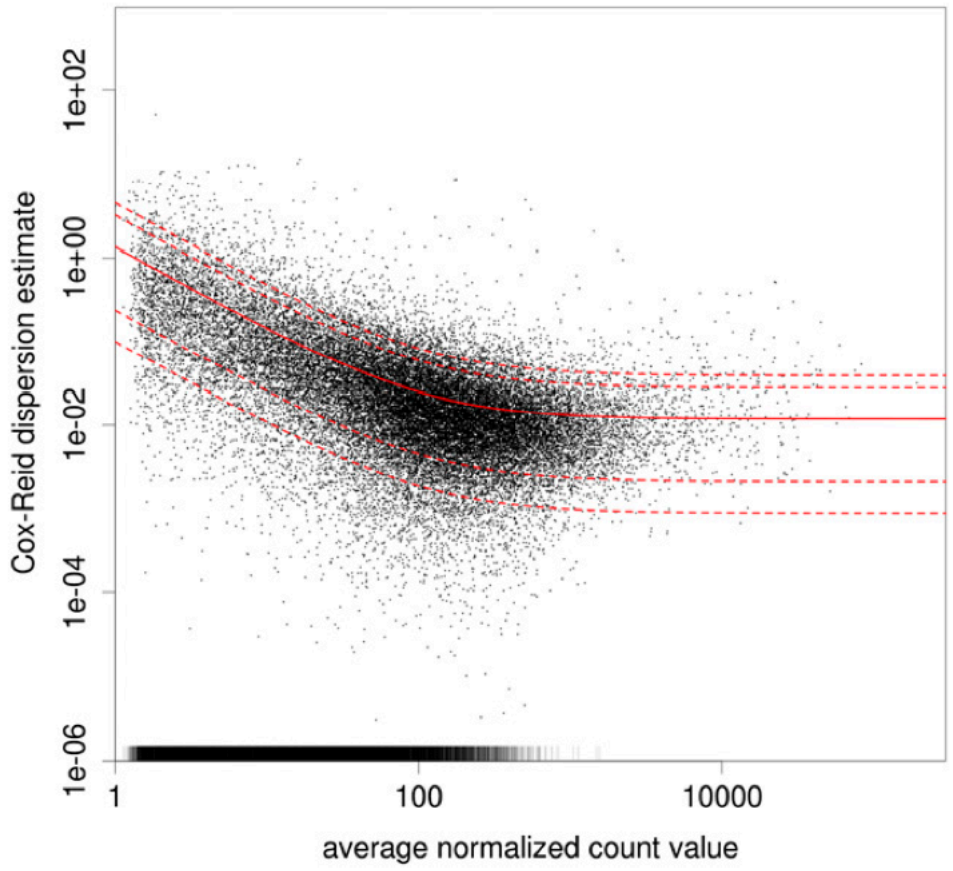
Normalized counts. As in previous Figure. with normalized count values of

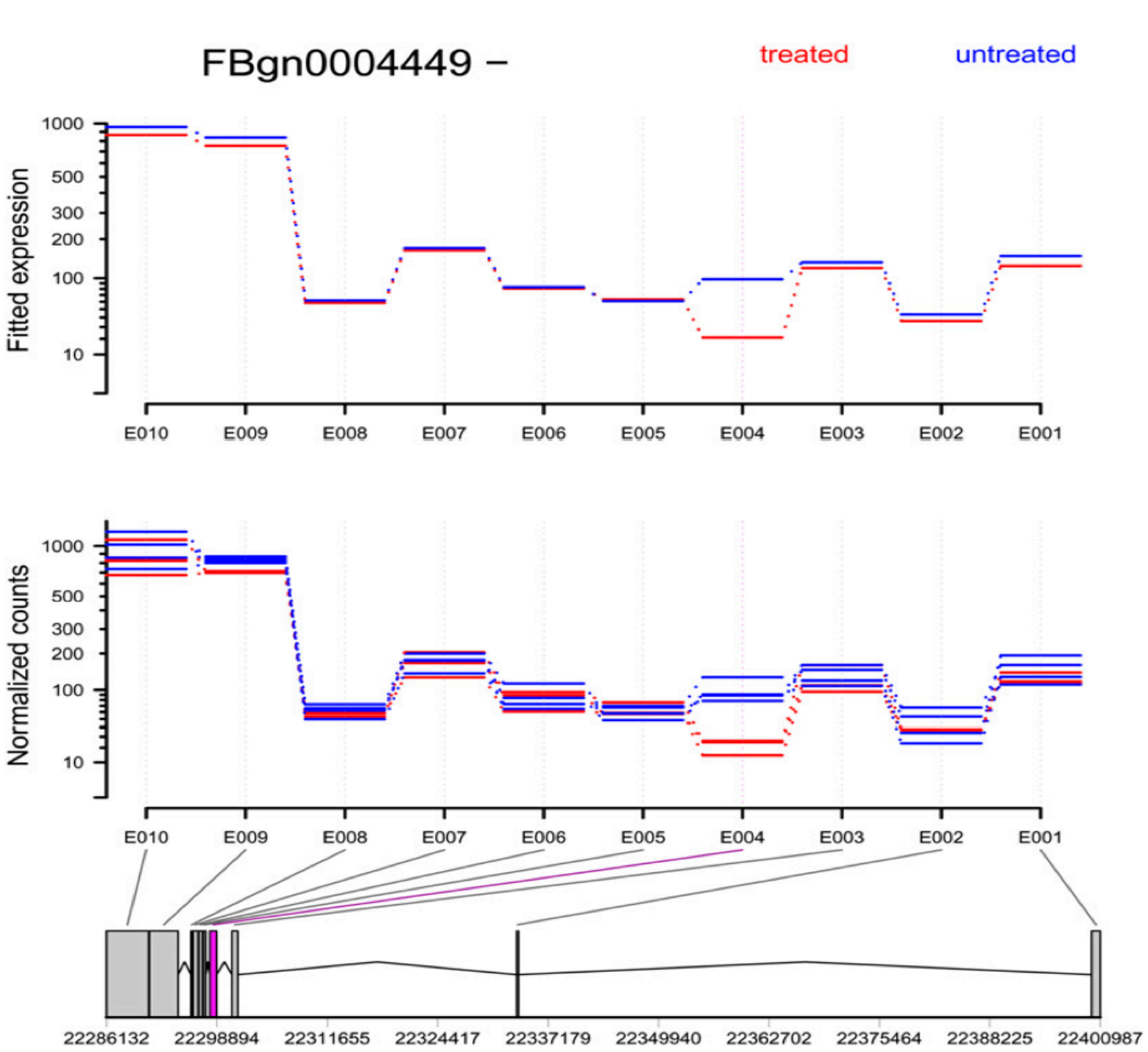


estimated effects, but after

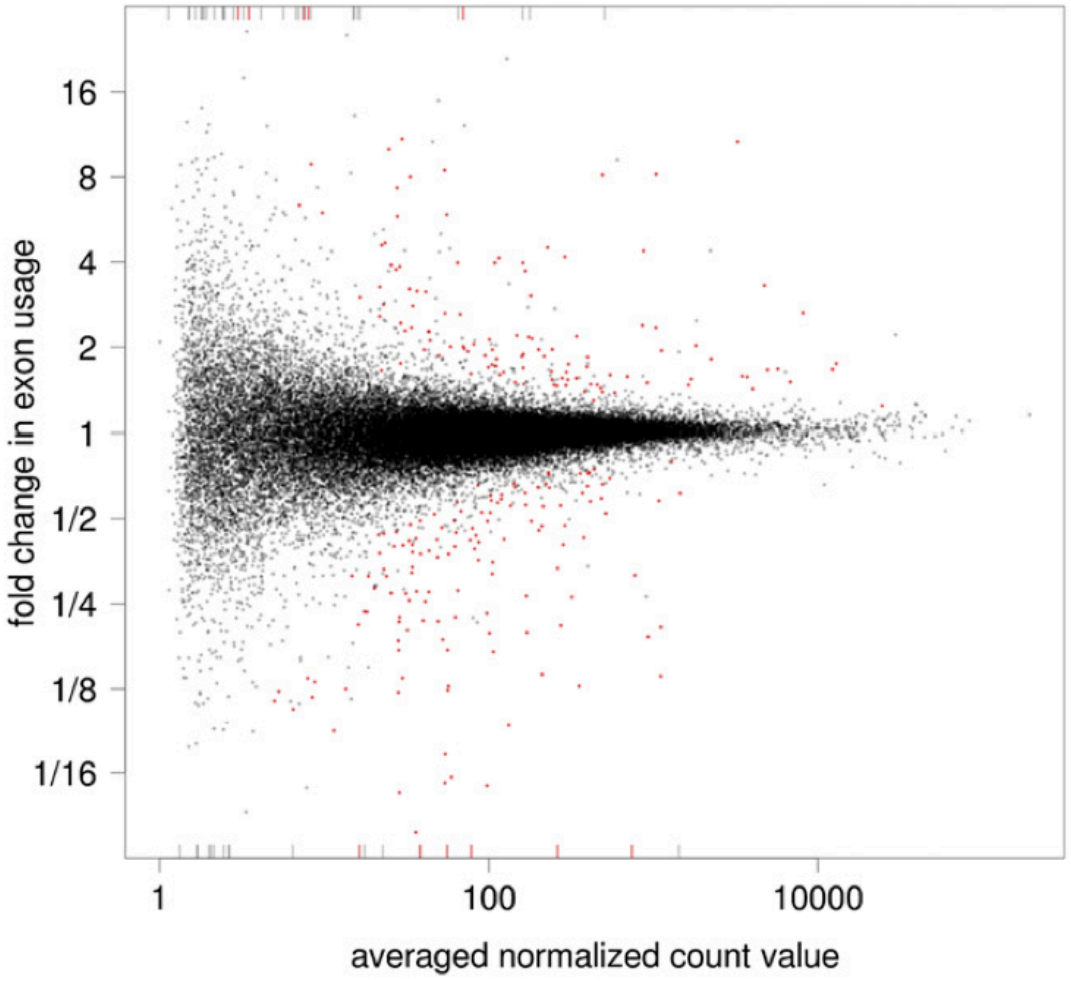


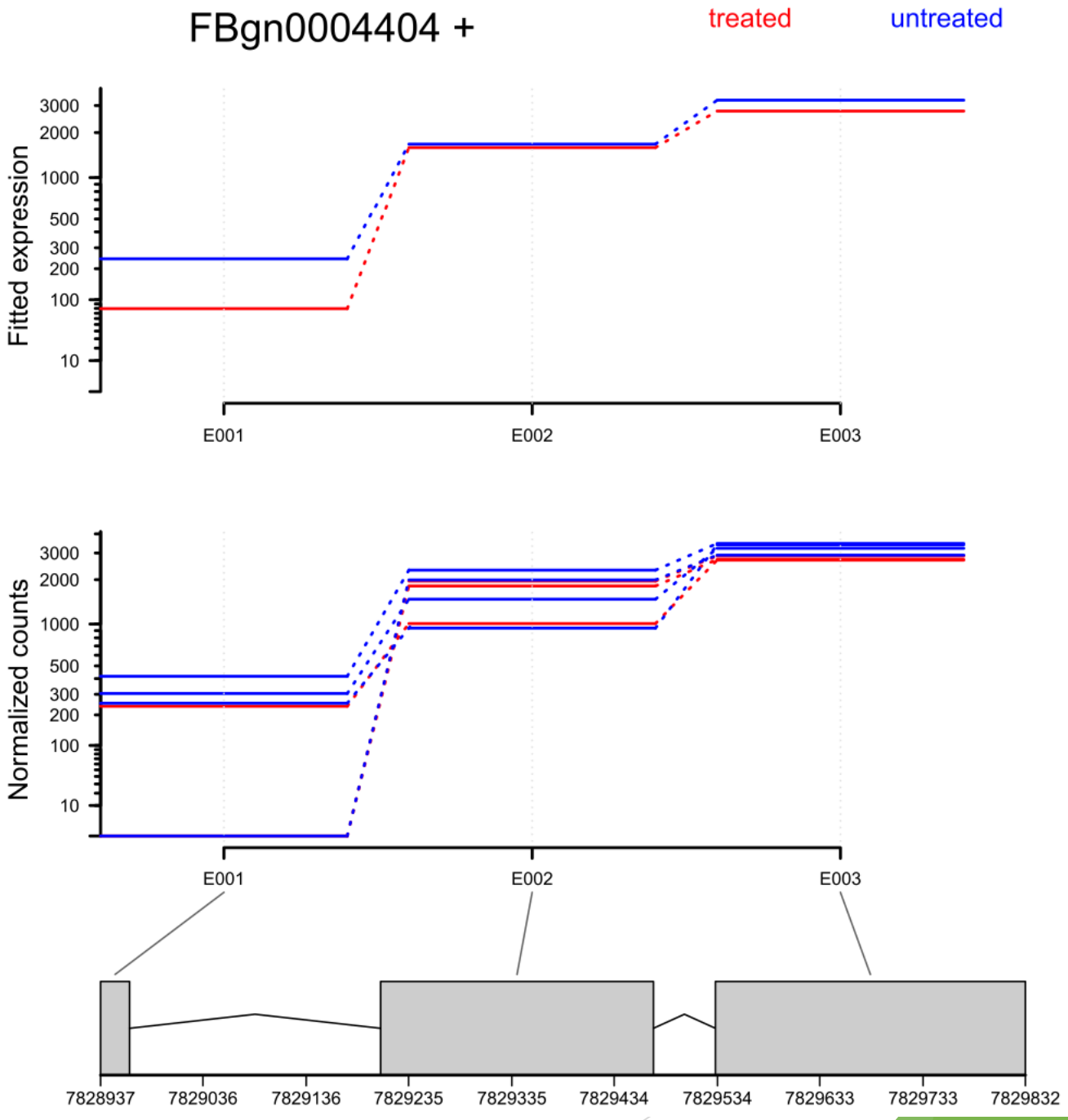
Dependence of dispersion on the mean




FBgn0004449 –
treated
untreated
Fitted expression
1000
500
300
200
100
10
E010
E009
E008
E007
E006
E005
E004
E003
E002
E001
Normalized counts
1000
500
300
200
100
10
E010
E009
E008
E007
E006
E005
E004
E003
E002
E001
22286132
22298894
22311655
22324417
22337179
22349940
22362702
22375464
22388225
22400987

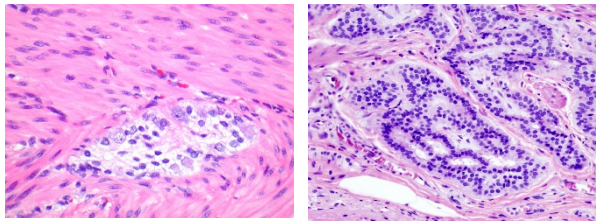
Distributions of Fold changes of exon usage




FBgn0004404 +
treated
untreated
Fitted expression
3000
2000
1000
500
300
200
100
10
E001
E002
E003
Normalized counts
3000
2000
1000
500
300
200
100
10
E001
E002
E003
7828937
7829036
7829136
7829235
7829335
7829434
7829534
7829633
7829733
7829832

RNA sequencing: abundance comparisons between two or more conditions / phenotypes

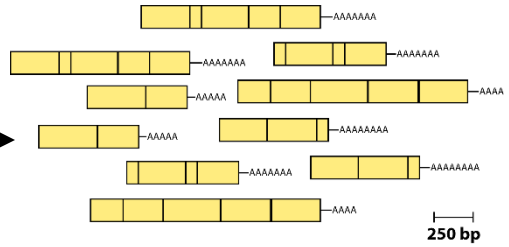
Samples of interest



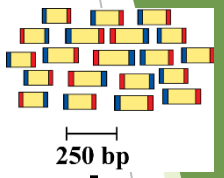
Condition 1 (normal tissue)

Condition 2 (diseased tissue)

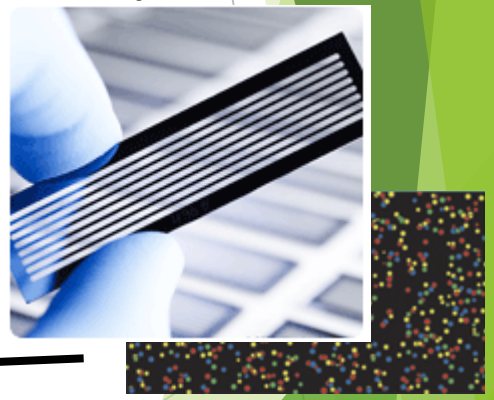
Isolate RNAs



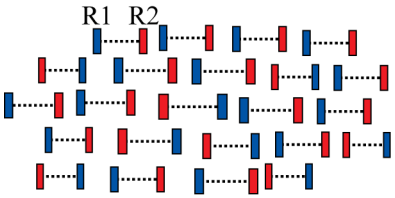
Generate cDNA, fragment, size select, add linkers



Sequence ends



100s of millions of paired reads
10s of billions bases of sequence



Map to genome, transcriptome, and predicted exon junctions

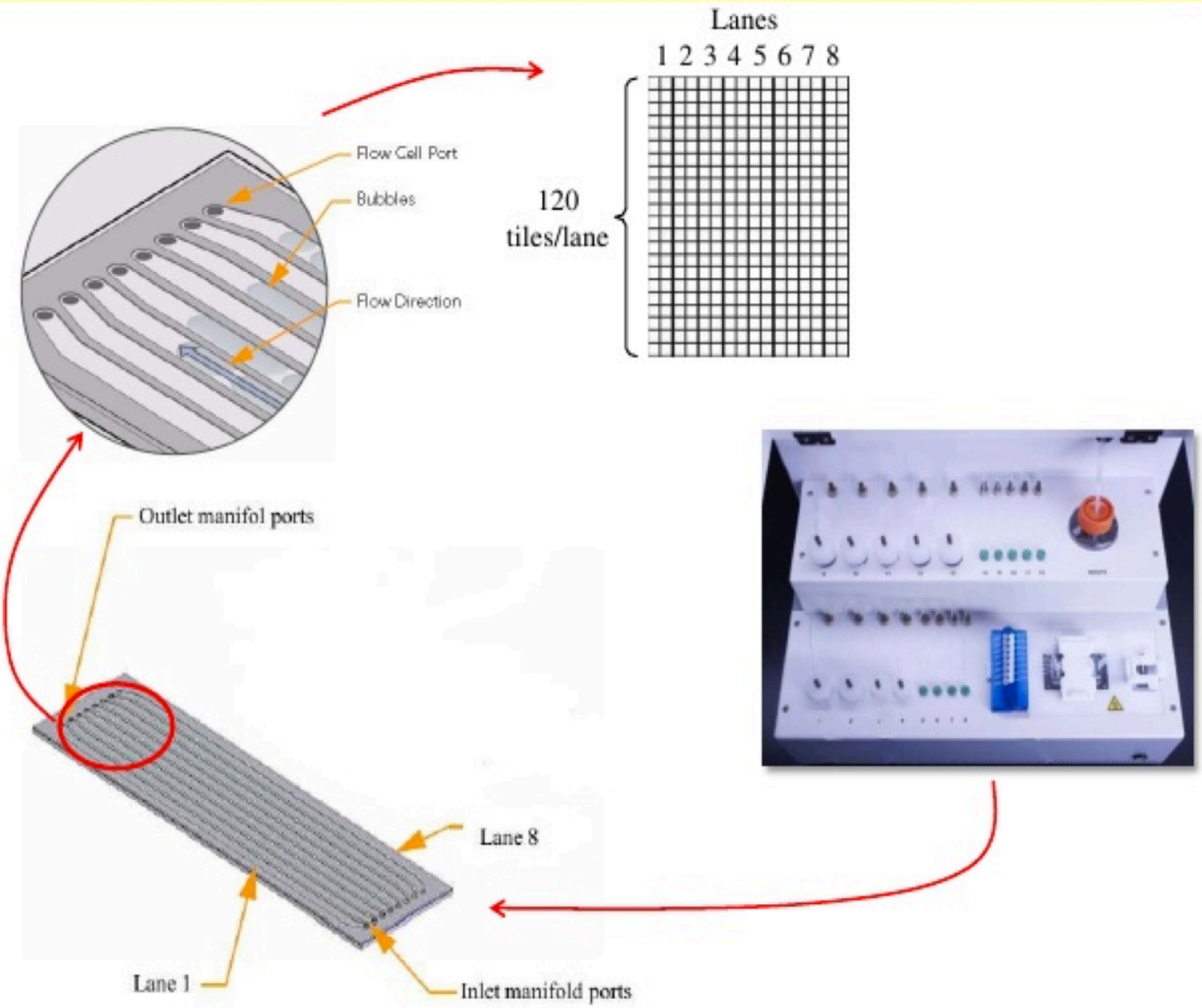
Downstream analysis

Sample preparation

Clusters amplification

Sequencing by synthesis

Analysis pipeline



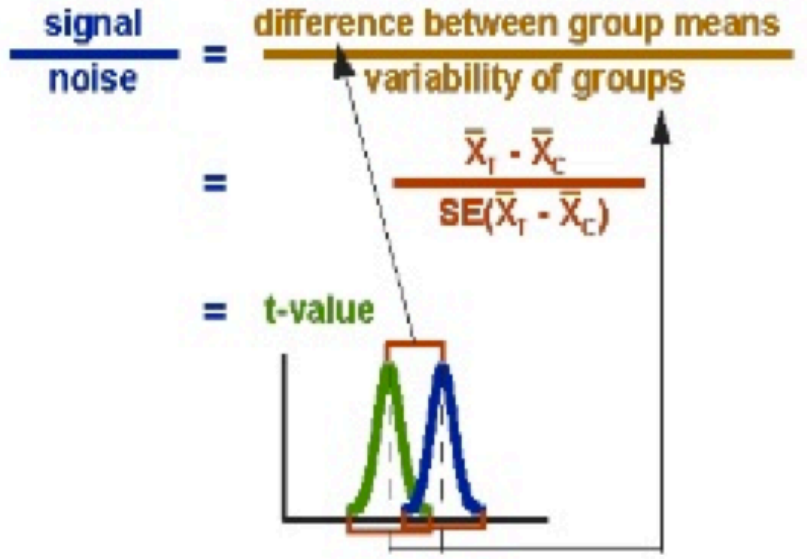
Count-based statistics

People often use discrete distributions (Poisson, negative binomial etc.) rather than continuous (e g normal) distributions for modeling RNA-seq data.

This is natural when you consider the way data are generated.

Problems associated with a t test

Couldn't we just use a Student's t test for each gene?



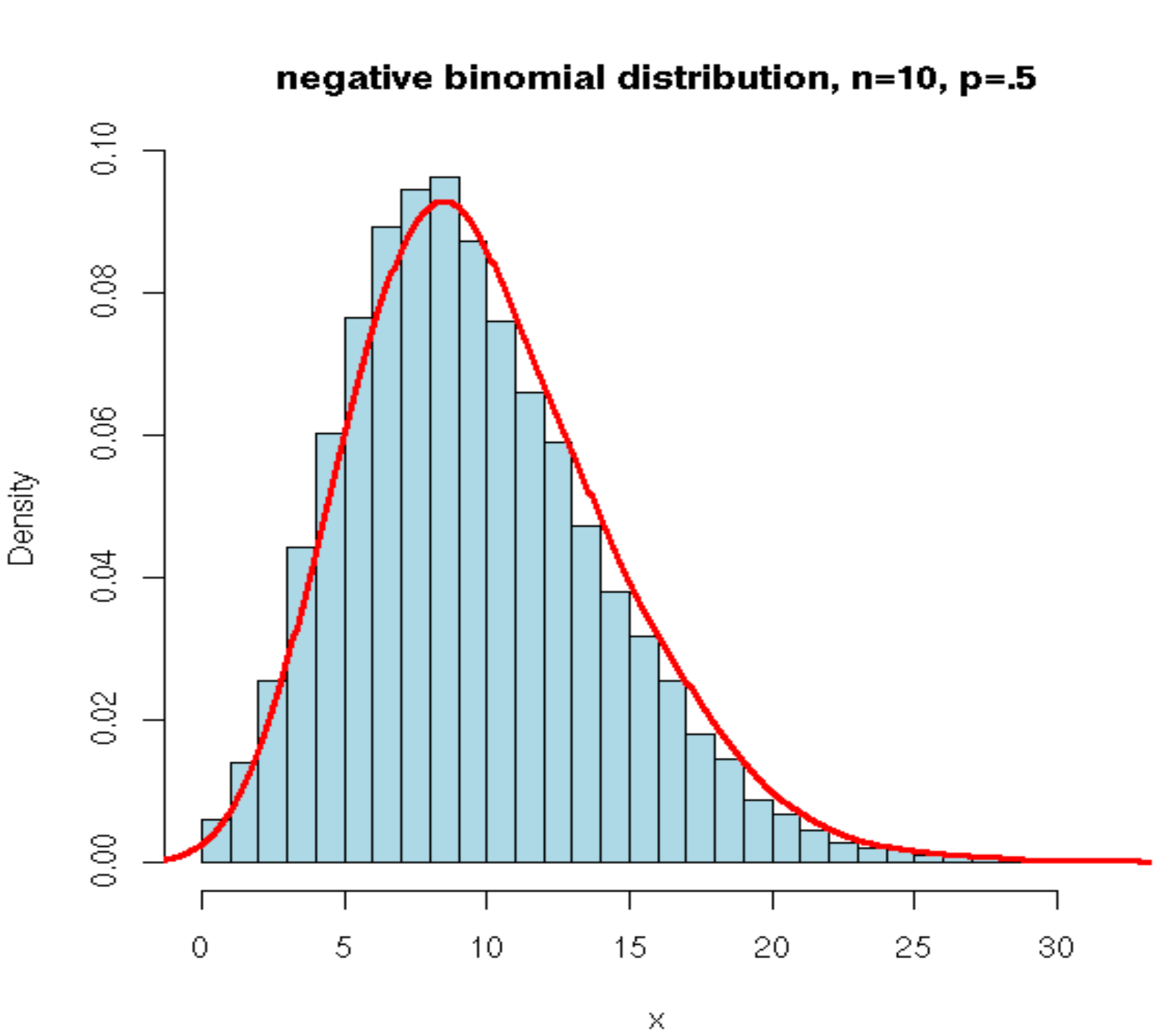
http://www.socialresearchmethods.net/kb/stat_t.php

Problems with this approach:

- May have **few replicates**

- Distribution is **not normal**

- **Multiple testing** issues

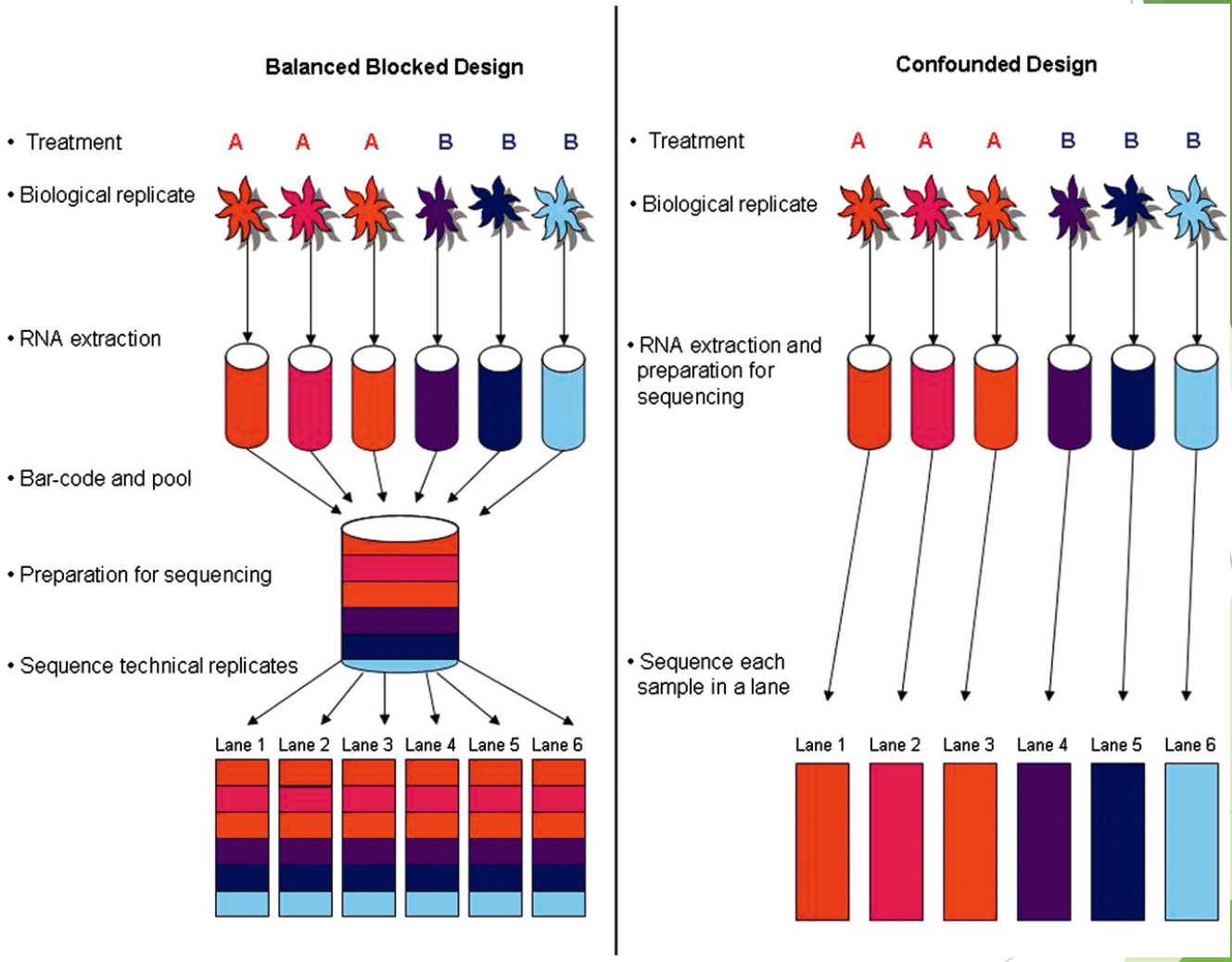

negative binomial distribution, n=10, p=.5
Density
0.00
0.02
0.04
0.06
0.08
0.10
0
5
10
15
20
25
30
x

Experimental design: best softwares

http://arxiv.org/ftp/arxiv/papers/1505/1505.02017.pdf

	Agreement with other tools[1]	WT v WT FPR[2]	Fold-change Threshold (T)[3]	Tool recommended for: (# good replicates per condition) [4]		
				<=3	<=12	>12
BaySeq	inconsistent	Pass				
cuffdiff	consistent	Fail				
DEGSeq	inconsistent	Fail				
DESeq	consistent	Pass	0			Yes
			0.5		Yes	Yes
			2	Yes	Yes	Yes
edgeR	consistent	Pass	0			Yes
			0.5	Yes	Yes	Yes
			2	Yes	Yes	Yes
Limma	consistent	Pass	0			Yes
			0.5		Yes	Yes
			2	Yes	Yes	Yes
NOISeq	inconsistent	Pass				
PoissonSeq	inconsistent	Fail				
SAMSeq	inconsistent	Fail				

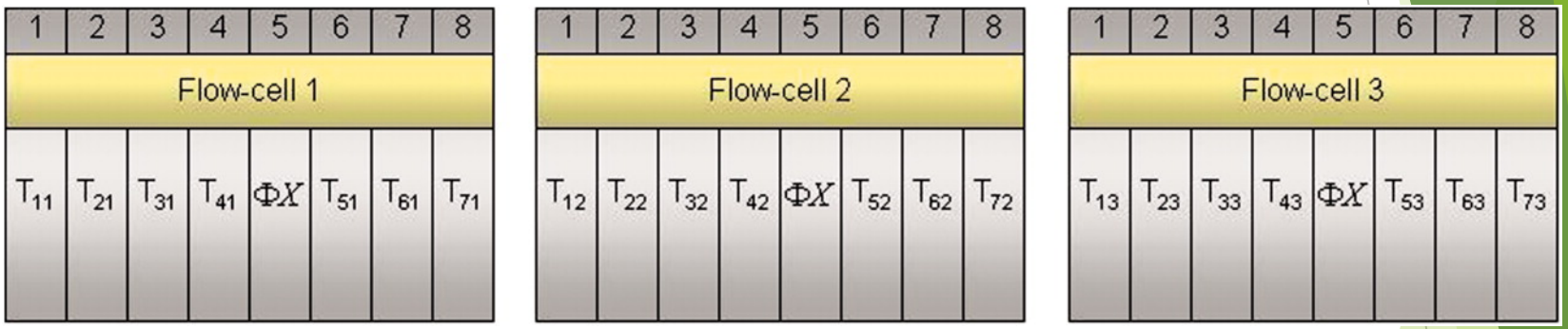
Comparison of two designs for testing differential expression between treatments A and B. Treatment A is denoted by red tones and treatment B by blue tones.



Paul L. Auer, and R. W. Doerge Genetics 2010;185:405-416

GENETICS

A multiple flow-cell design based on three biological replicates within seven treatment groups.



Paul L. Auer, and R. W. Doerge Genetics 2010;185:405-416

GENETICS

Technical vs biological replicates

Technical replicates:

Assess variability of measurement technique
Typically low for bulk RNA-seq (not necessarily single-cell RNA-seq)
Poisson distribution can model variability between RNA-seq technical replicates rather well

Biological replicates:

Assess variability between individuals / "normal" biological variation
Necessary for drawing conclusions about biology
Variability across RNA-seq biological replicates not well modelled by Poisson – usually negative binomial ("overdispersed Poisson") is used

Replicates and differential expression

Intuitively, the variation **between** the groups that you want to compare should be large compared to the variation **within** each group to be able to say that we have differential expression.

The more biological replicates, the better you can estimate the variation. But how many replicates are needed?

Depends:

Homogeneous cell lines, inbred mice etc: maybe 3 samples / group enough.
Clinical case-control studies on patients: can need a dozen, hundreds or thousands, depending on the specifics ....

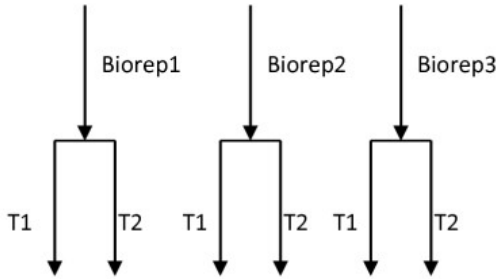
7 Recommendations for RNA-seq experiment design

The results of this study suggest the following should be considered when designing an RNA-seq experiment for DGE:

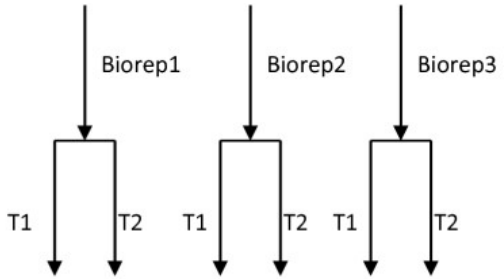
1) At least 6 replicates per condition for all experiments.
2) At least 12 replicates per condition for experiments where identifying the majority of all DE genes is important.
3) For experiments with < 12 replicates per condition; use *edgeR*.
4) For experiments with > 12 replicates per condition; use *DESeq*.

5) Apply a fold-change threshold appropriate to the number of replicates per condition between $0.1 \leq T \leq 0.5$ (see Figure 2 and the discussion of tool performance as a function of replication).

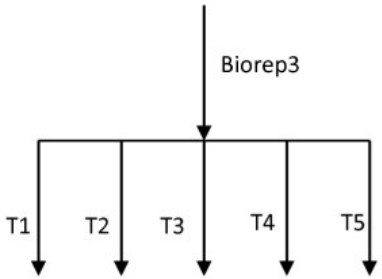
D. melanogaser



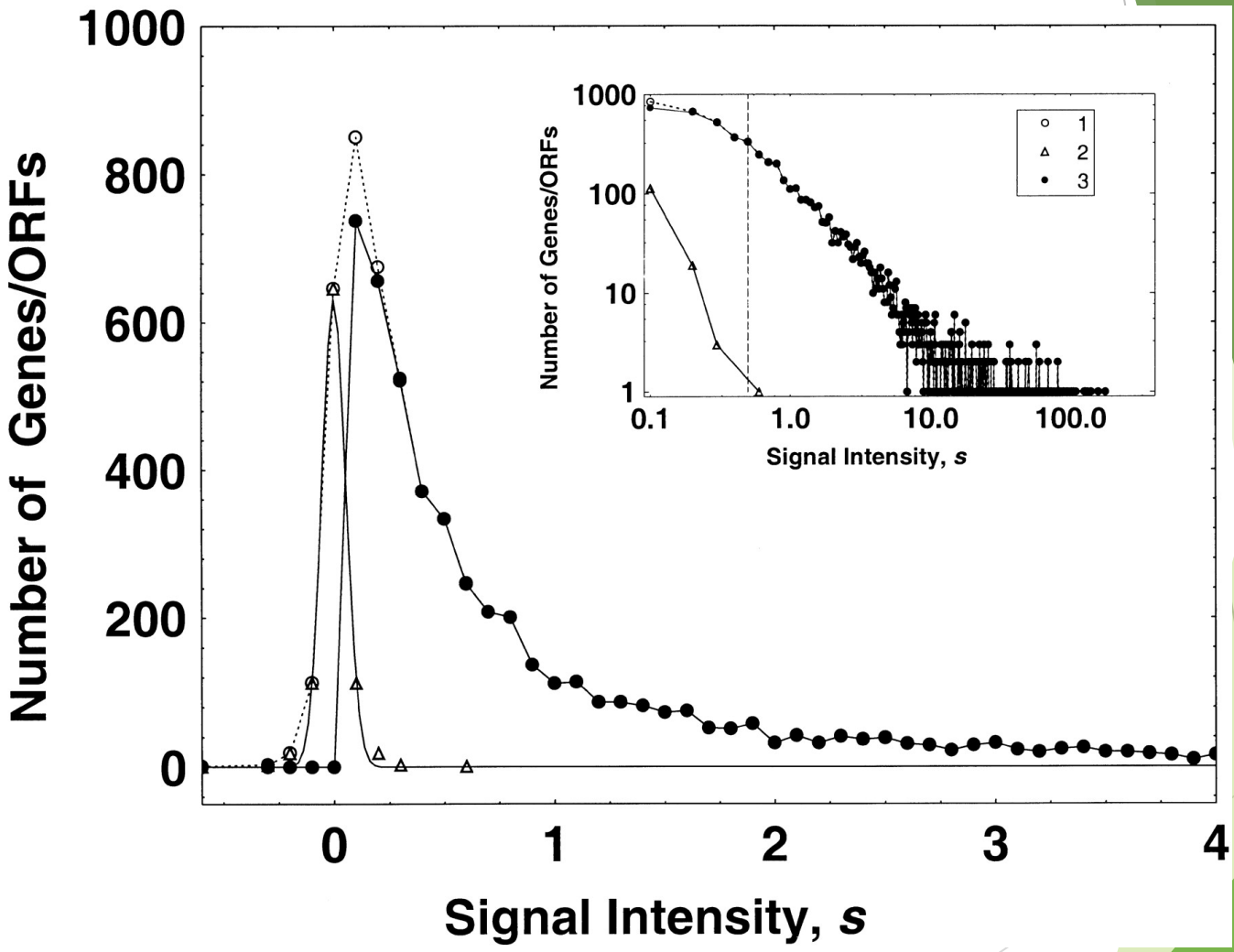
D. simulans



Cell line “c167”



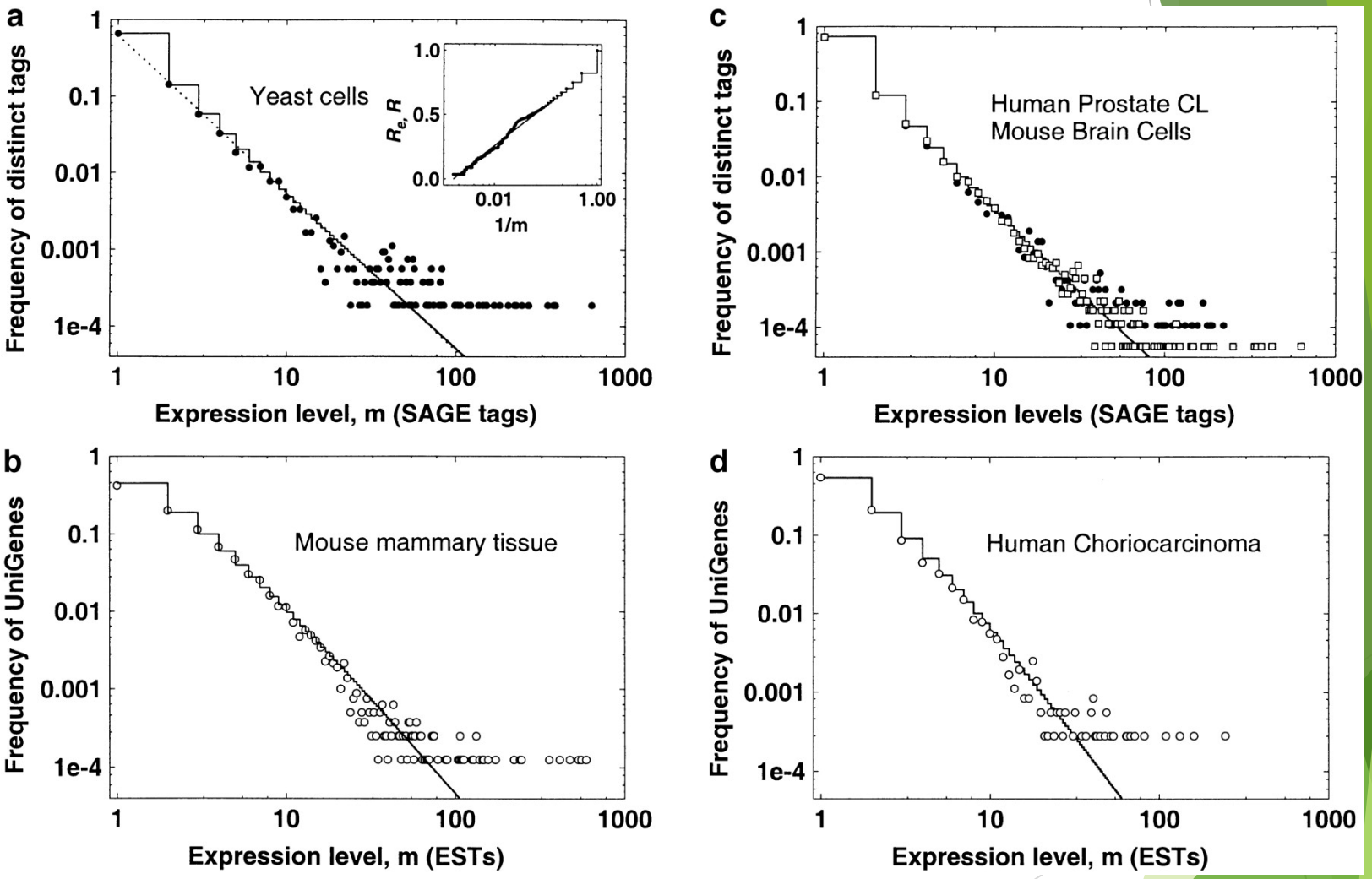
The empirical frequency distribution of the hybridization signal intensity values for Affymetrix microarray hybridization data for normal yeast cell genes/ORFs (Jelinsky and Samson 1999).



Kuznetsov V A et al. Genetics 2002;161:1321-1332

GENETICS

Empirical relative frequency distributions of the gene expression levels.



Kuznetsov V A et al. Genetics 2002;161:1321-1332

GENETICS

Resources: RNA-Seq workflow, gene-level exploratory analysis and differential

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Search:

Home | Install | Help | Developers | About

Home » Help » Workflows » RNA-Seq workflow at the gene level

About This Document »

Package: rnaseqGene
Built with Bioconductor (R): 3.1 (3.2.0)

Author	Michael Love
Date	March 27, 2015
Source Package	rnaseqGene_0.99.104324.tar.gz
Windows Binary	rnaseqGene_0.99.104324.zip
Mac OS 10.6 (Snow Leopard)	rnaseqGene_0.99.104324.tgz

R Script
Last Built At: Thu May 28 08:14:17 PDT 2015
First Committed: Thu Sep 04 09:51:18 PDT 2014
SVN Revision: 104324
Install with (under BioC 3.1):

```
source("http://bioconductor.org/workflows.R")
workflowInstall("rnaseqGene")
```

Workflows »

Common Bioconductor workflows include:

- Oligonucleotide Arrays
- High-throughput Sequencing
- Counting Reads for Differential Expression (parathyroideSE vignette)
- Annotation Resources
- Annotating Genomic Ranges
- Annotating Genomic Variants
- Flow Cytometry and other assays
- Candidate Binding Sites for Known Transcription Factors
- Cloud-enabled cis-eQTL search and annotation
- RNA-Seq workflow: gene-level exploratory analysis and differential expression
- Changing genomic coordinate systems with rtracklayer::liftOver
- Mass spectrometry and proteomics data analysis

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

Aligning reads to a reference

The computational analysis of an RNA-Seq experiment begins earlier: what we get from the sequencing machine is a set of 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. 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.

A number of software programs exist to align reads to the reference genome, and the development is too rapid for this document to provide an up-to-date list. We recommend consulting benchmarking papers that discuss the advantages and disadvantages of each software, which include accuracy, ability to align reads over splice junctions, speed, memory footprint, and many other features.

The reads for this experiment were aligned to the Ensembl release 75 human reference genome using the STAR read aligner:

```
for f in `cat files`; do STAR --genomeDir ../STAR/ENSEMBL.homo_sapiens.release-75 \
--readFilesIn fastq/$f\_1.fastq fastq/$f\_2.fastq \
--runThreadN 12 --outFileNamePrefix aligned/$f.; done
```

SAMtools was used to generate BAM files.

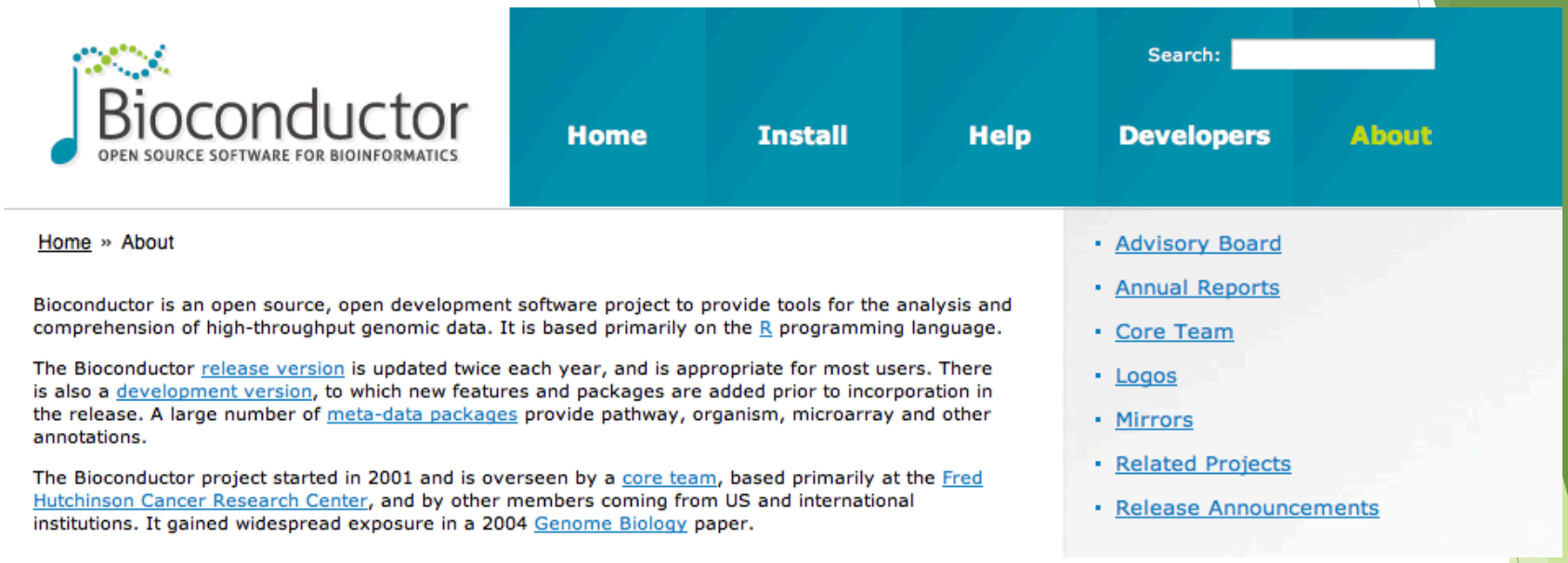
```
cat files | parallel -j 7 samtools view -bS aligned/{}.Aligned.out.sam -o aligned/{}.ba
```

The BAM files for a number of sequencing runs can then be used to generate count matrices, as described in the following section.

Outline

- Introduction to RNA sequencing
- Rationale for RNA sequencing (versus DNA sequencing)
- Hands on tutorial
- http://swcarpentry.github.io/r-novice-inflammation/
- http://swcarpentry.github.io/r-novice-inflammation/02-func-R.html
- http://www.bioconductor.org/help/workflows/
- http://www.bioconductor.org/packages/release/data/experiment/html/parathyroidSE.html
- http://www.bioconductor.org/help/workflows/rnaseqGene/

About bioconductor



High-throughput sequence analysis with R and Bioconductor: http://www.bioconductor.org/help/course-materials/2013/useR2013/Bioconductor-tutorial.pdf * http://bioconductor.org/packages/2.13/data/experiment/vignettes/RnaSeqTutorial/inst/doc/RnaSeqTutorial.pdf

Also helpful: http://www.bioconductor.org/help/course-materials/2002/Summer02Course/Labs/basics.pdf

http://www.nature.com/nprot/j

PROTOCOL

Count-based differential expression analysis of RNA sequencing data using R and Bioconductor

Simon Anders[1], Davis J McCarthy[2,3], Yunshun Chen[4,5], Michal Okoniewski[6], Gordon K Smyth[4,7], Wolfgang Huber[1] & Mark D Robinson[8,9]

[1]Genome Biology Unit, European Molecular Biology Laboratory, Heidelberg, Germany. [2]Department of Statistics, University of Oxford, Oxford, UK. [3]Wellcome Trust Centre for Human Genetics, University of Oxford, Oxford, UK. [4]Bioinformatics Division, Walter and Eliza Hall Institute, Parkville, Victoria, Australia. [5]Department of Medical Biology, University of Melbourne, Melbourne, Victoria, Australia. [6]Functional Genomics Center UNI ETH, Zurich, Switzerland. [7]Department of Mathematics and Statistics, University of Melbourne, Melbourne, Victoria, Australia. [8]Institute of Molecular Life Sciences, University of Zurich, Zurich, Switzerland. [9]SIB Swiss Institute of Bioinformatics, University of Zurich, Zurich, Switzerland. Correspondence should be addressed to M.D.R. (mark.robinson@imls.uzh.ch) or W.H. (whuber@embl.de).

Published online 22 August 2013; doi:10.1038/nprot.2013.099

RNA sequencing (RNA-seq) has been rapidly adopted for the profiling of transcriptomes in many areas of biology, including studies into gene regulation, development and disease. Of particular interest is the discovery of differentially expressed genes across different conditions (e.g., tissues, perturbations) while optionally adjusting for other systematic factors that affect the data-collection process. There are a number of subtle yet crucial aspects of these analyses, such as read counting, appropriate treatment of biological variability, quality control checks and appropriate setup of statistical modeling. Several variations have been presented in the literature, and there is a need for guidance on current best practices. This protocol presents a state-of-the-art computational and statistical RNA-seq differential expression analysis workflow largely based on the free open-source R language and Bioconductor software and, in particular, on two widely used tools, DESeq and edgeR. Hands-on time for typical small experiments (e.g., 4–10 samples) can be <1 h, with computation time <1 d using a standard desktop PC.

RnaSeqTutorial

RNA-Seq Tutorial (EBI Cambridge UK, October 2011)

Bioconductor version: Release (2.12)

A selection of RNA-Seq data to get familiar with the related Bioconductor core packages and the easyRNASeq package.

Author: Nicolas Delhomme, Ismael Padioleau

Maintainer: Nicolas Delhomme <delhomme at embl.de>

To install this package, start R and enter:

```
source("http://bioconductor.org/biocLite.R")
biocLite("RnaSeqTutorial")
```

To cite this package in a publication, start R and enter:

```
citation("RnaSeqTutorial")
```

Documentation

PDF	R Script	RNA-Seq Tutorial
PDF		Reference Manual
Text		NEWS

Details

biocViews	ExperimentData, RNAseqData
Version	0.0.12
License	Artistic-2.0
Depends	R (>= 2.15.0), methods, easyRNASeq
Imports	
Suggests	Rsamtools, ShortRead, BSgenome.Dmelanogaster.UCSC.dm3, GenomicRanges, biomaRt, genomeIntervals
System Requirements	
URL	
Depends On Me	
Imports Me	
Suggests Me	easyRNASeq

Package Downloads

Package Source	RnaSeqTutorial_0.0.12.tar.gz
Windows Binary	RnaSeqTutorial_0.0.12.zip (32- & 64-bit)
Mac OS X 10.6 (Snow Leopard)	RnaSeqTutorial_0.0.12.tgz
Package Downloads Report	Download Stats

Workflows »

Common Bioconductor workflows include:

- Oligonucleotide Arrays
- High-throughput Sequencing
- Annotation
- Variants
- Flow Cytometry and other assays
- Finding Candidate Binding Sites for Known Transcription Factors via Sequence Matching

Mailing Lists »

Post questions about Bioconductor packages to our mailing lists. Read the posting guide before posting!

- bioconductor
- bioc-devel

Downloads / tutorial

- http://www.bioconductor.org/help/workflows/rnaseqGene/
- http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE52778
- ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByStudy/sra/SRP%2FSRP033%2FSRP033351/SRR1039508/
- http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE52778

The End