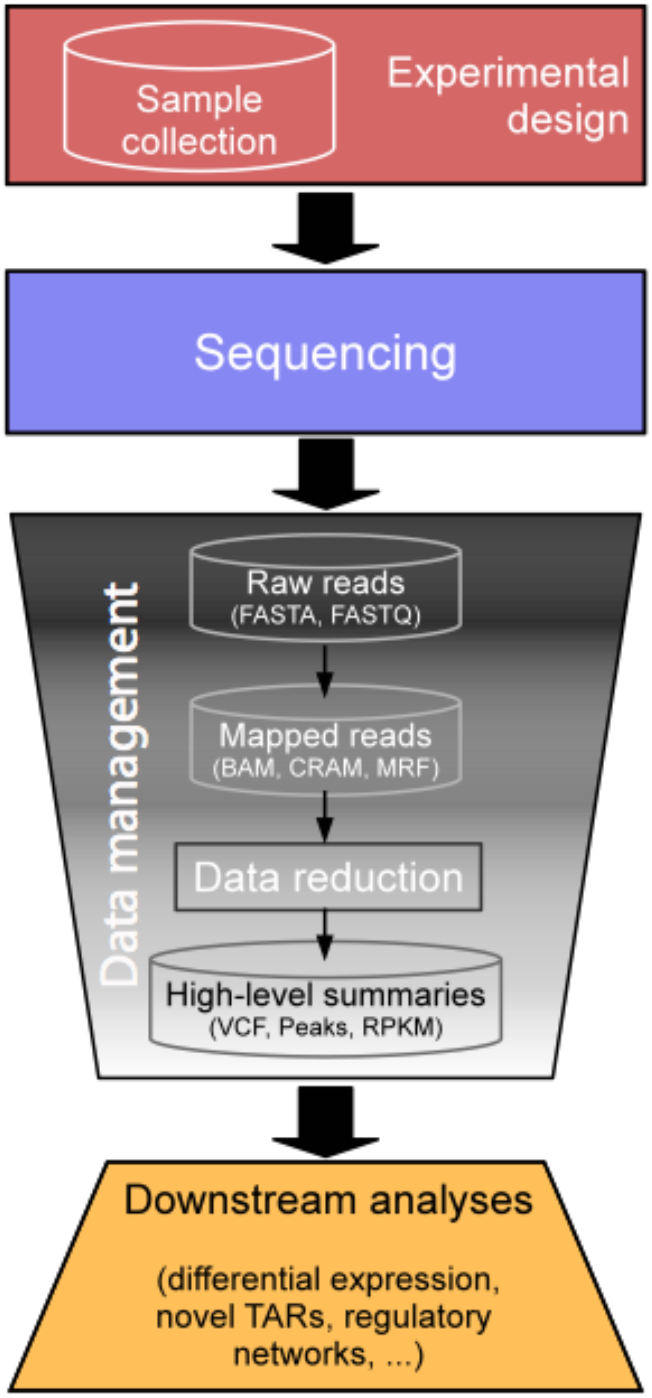


RNA-seq data analysis tutorial

Andrea Sboner

2015-05-21

NGS Experiment



Data management:

Mapping the reads
Creating summaries

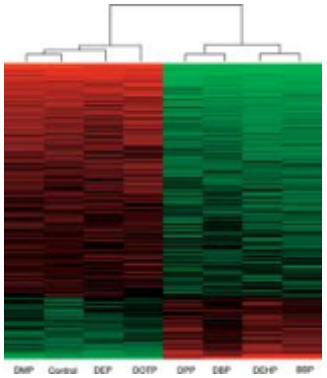
Downstream analysis: *the interesting stuff*
Differential expression, chimeric transcripts, novel transcribed regions, etc.

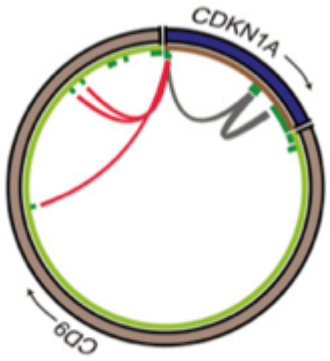
What is RNA-seq?

- Next-generation sequencing applied to the "transcriptome"

  Applications:

  - Gene (exon, isoform) expression estimation
  - *Differential gene (exon, isoform) expression analysis*
  - Discovery of novel transcribed regions
  - *Discovery/Detection of chimeric transcripts*
  - Allele specific expression
  - …



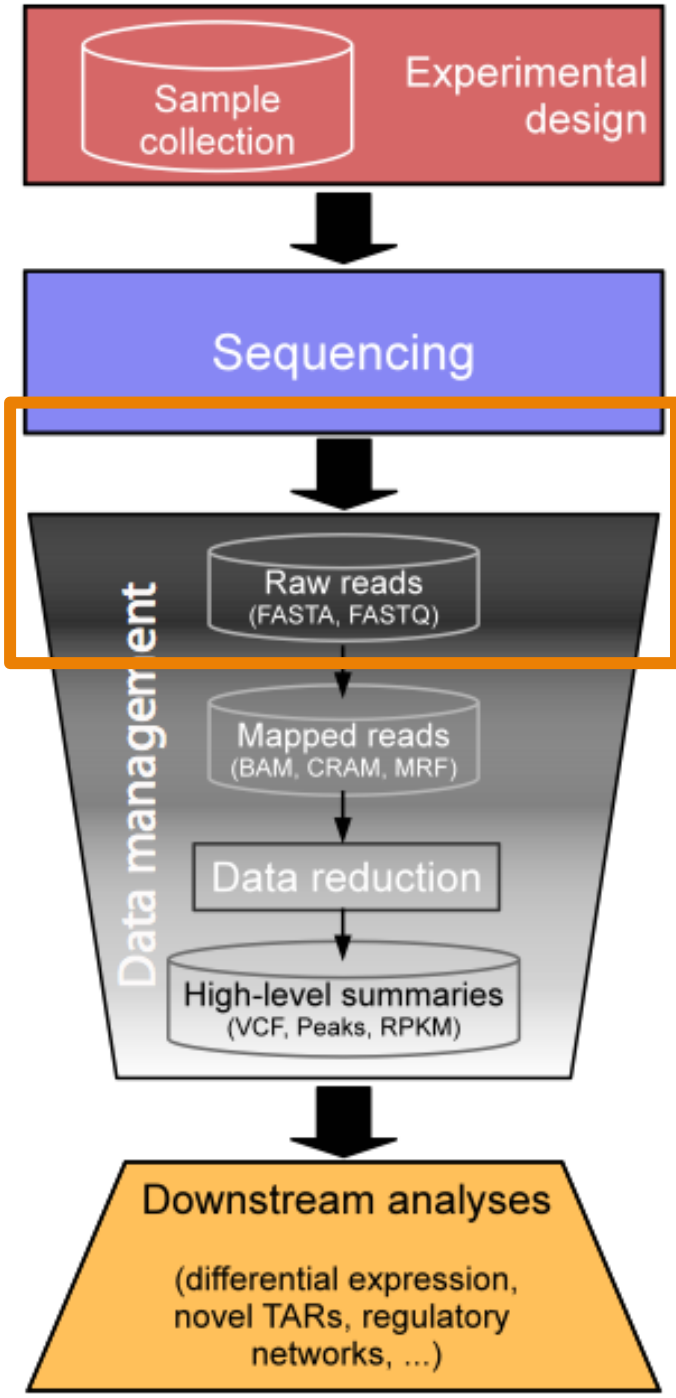


NGS Experiment

Data management:

Mapping the reads
Creating summaries

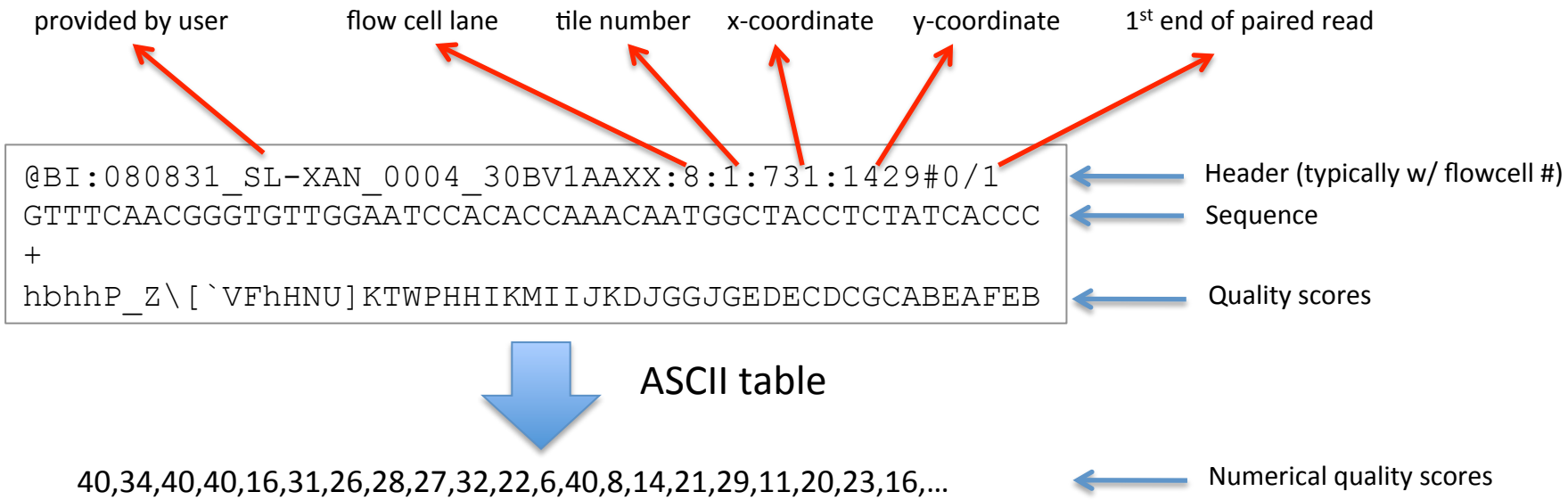
Downstream analysis: *the interesting stuff*
Differential expression, chimeric transcripts, novel transcribed regions, etc.



QC and pre-processing

- First step in QC:
  - Look at quality scores to see if sequencing was successful

- Sequence data usually stored in FASTQ format:



Phred Quality Score	Probability of Incorrect Base Call	Base Call Accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.90%
40	1 in 10000	99.99%

Freely available tools for QC

- FastQC
  - http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/
  - Nice GUI and command line interface

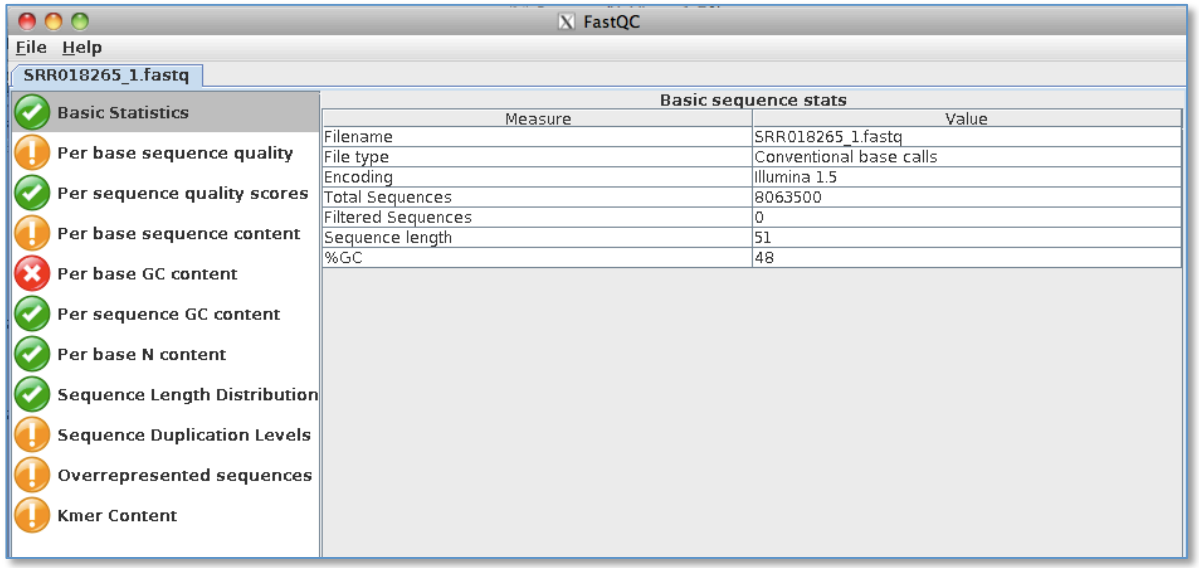
- FASTX-Toolkit
  - http://hannonlab.cshl.edu/fastx_toolkit/index.html
  - Tools for QC as well as trimming reads, removing adapters, filtering by read quality, etc.

- Galaxy
  - http://main.g2.bx.psu.edu/
  - Web interface
  - Many functions but analyses are done on remote server

FastQC

- GUI mode

  fastqc



- Command line mode

  fastqc *fastq_files* –o *output_directory*

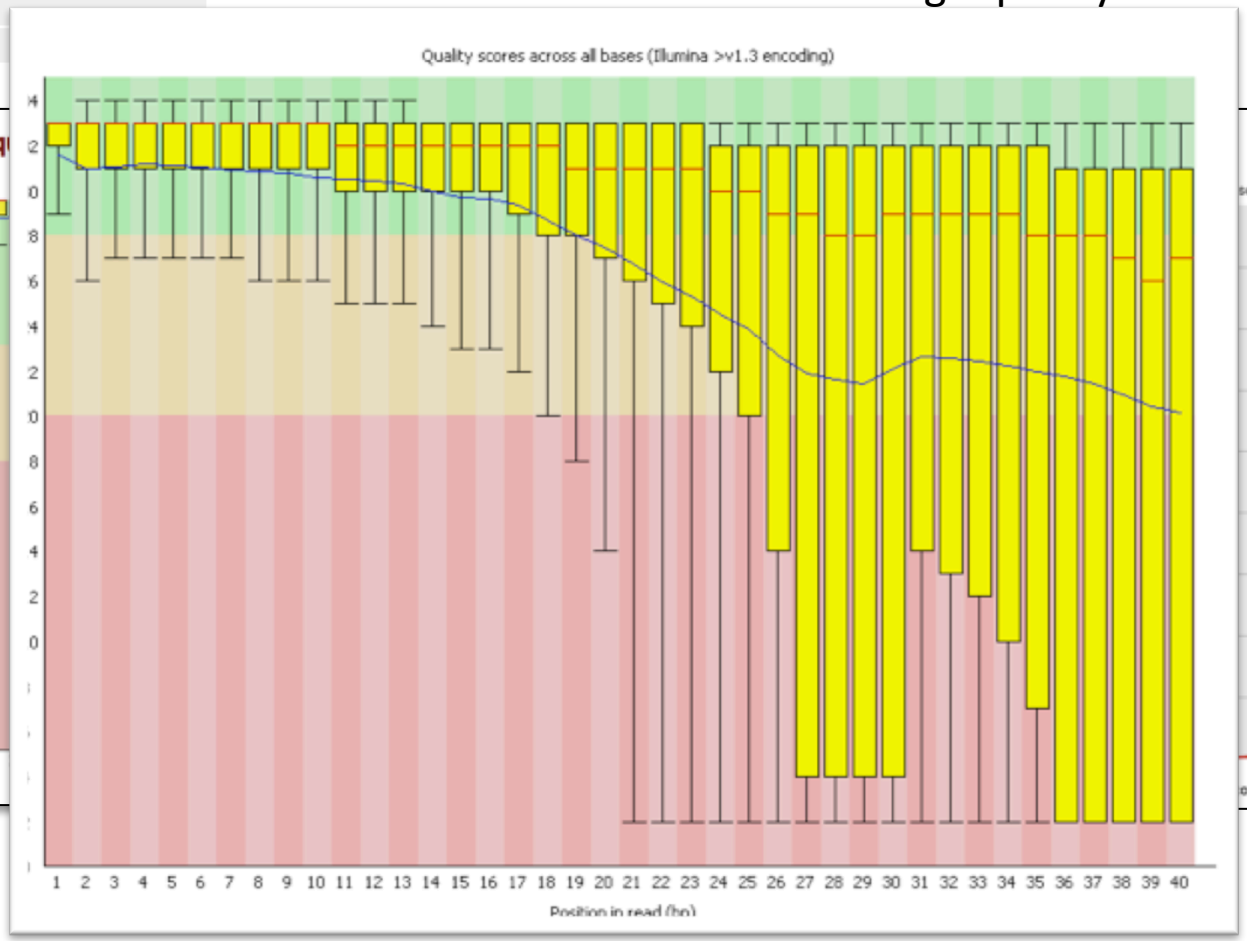
  – will create fastq_file_fastqc.zip in output directory

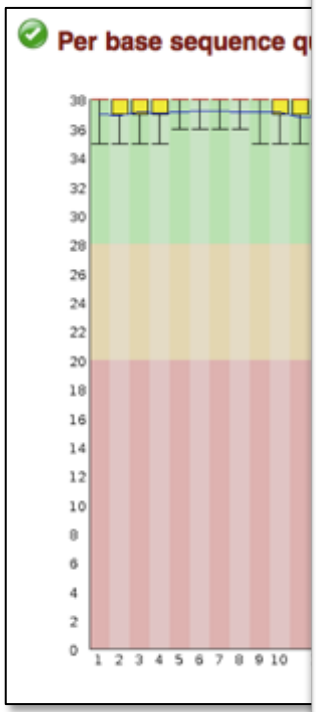
FastQC

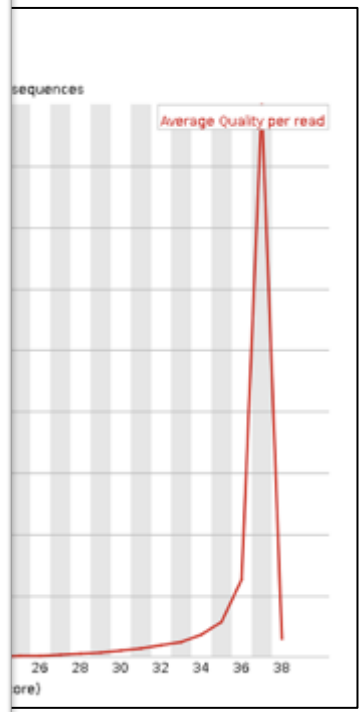
Basic Statistics

Measure	Value
Filename	sample_1.fastq.gz
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	255665
Filtered Sequences	0
Sequence length	51
%GC	45

rule of thumb: average quality > 20 for the first 36bp

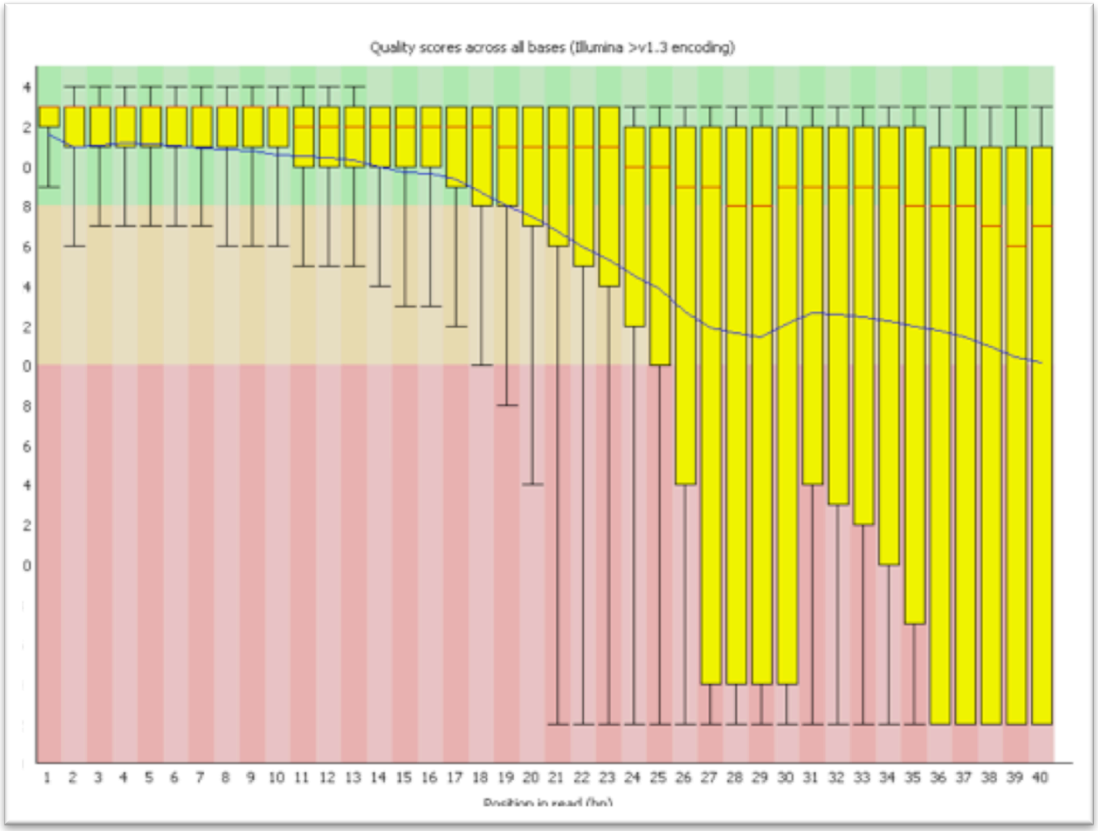






What to do when quality is poor?

- Trim the reads
- FASTX-toolkit
  - **fastx_trimmer**
    –f N –l N
  - **fastq_quality_filter**
    -q N –p N
  - **Fastx_clipper**
    -a ADAPTER

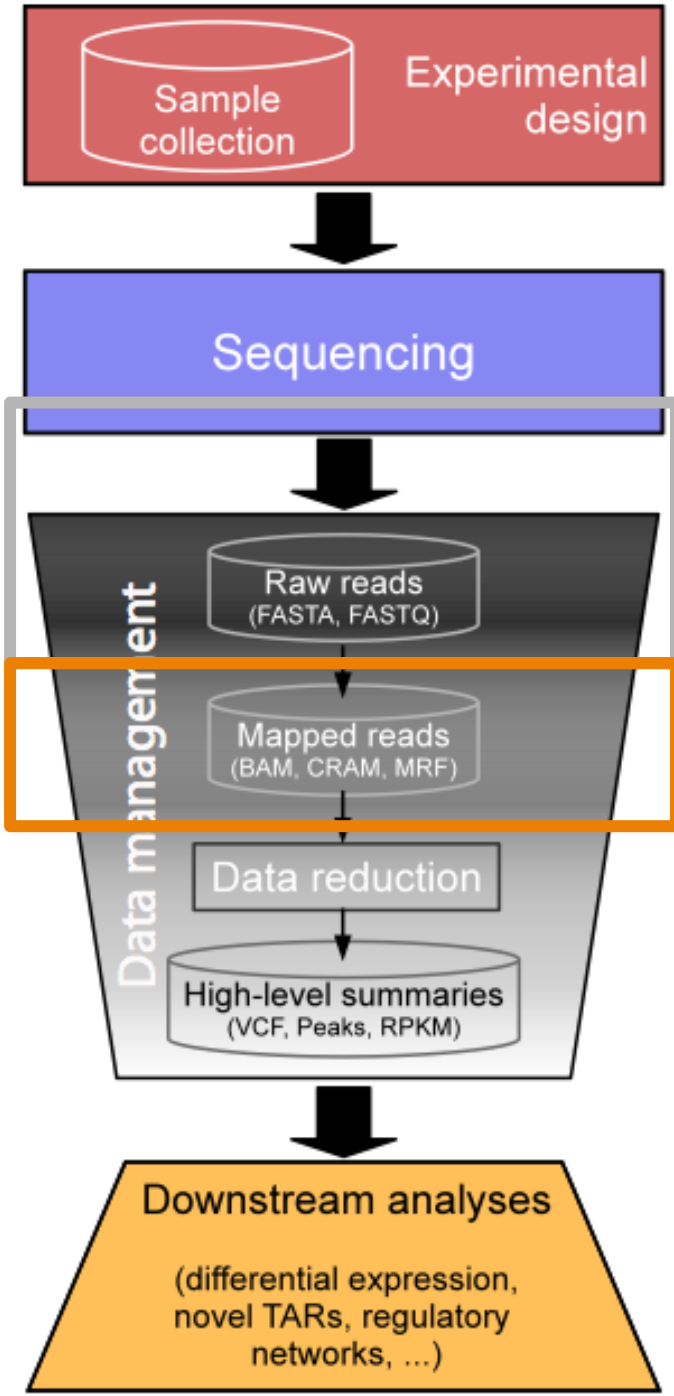


NGS Experiment

Data management:

Mapping the reads
Creating summaries

Downstream analysis: *the interesting stuff*
Differential expression, chimeric transcripts, novel transcribed regions, etc.

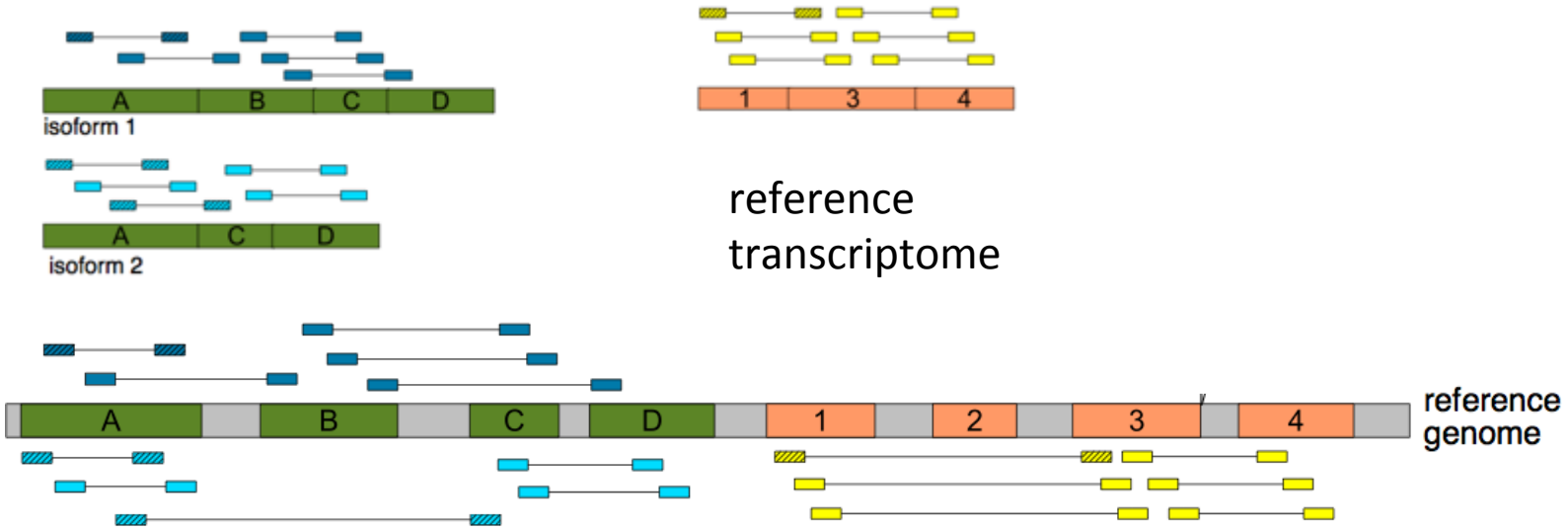


Mapping

Mapping


Google
ATCCAGCATTCGCGAAGTCGTA
Get directions
My places
1305 York Ave
New York, NY 10021
Directions
Search nearby
Save to map
more
Selected businesses at this address:
Ahmed Shakil MD
Borden William MD
Cardiology: Kutler David MD
Center For Reproductive Medicine: Rauch Eden MD
Cholst Ina MD
Cornell IVF Program
Dr. Jonathan H. Zippin, MD
Dr. Samuel H. Selesnick, MD
Dr. Shari Lipner, MD
Ent: Kacker Ashutosh MD
Gauthier Susan A DO
Jacobson Ira M MD
Kang Hey-Joo MD
Kim Alyn MD
Levinger Joshua I MD
Modi Vikash K MD
Neurology Clinic: Winterkorn Jacqueline MD
Prasad Mukesh MD
Sarkaria Sawreet MD
Voigt Erich P MD
All businesses at this address »
Ad
NY ear nose throat doctor
Locate New York sinus specialists
skilled in latest office procedures
www.sinussurgeryoptions.com/
See your ad here »

Mapping to a reference



- **Genome**
- Transcriptome
- Genome + Transcriptome
- Transcriptome + Genome
- Genome + splice junction library

Alignment tools

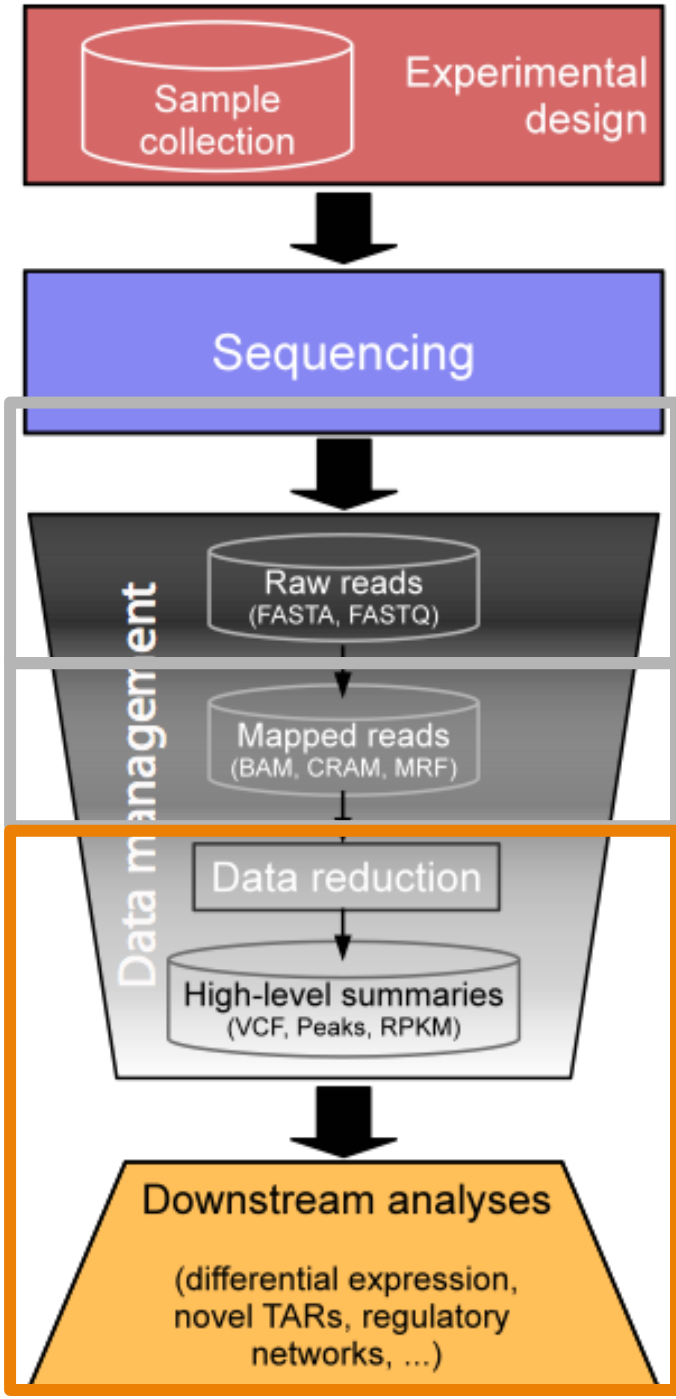
- BWA
  - http://bio-bwa.sourceforge.net/bwa.shtml
  - Gapped alignments (good for indel detection)
- Bowtie
  - http://bowtie-bio.sourceforge.net/index.shtml
  - Supports gapped alignments in latest version (bowtie 2)
- TopHat
  - http://tophat.cbcb.umd.edu/
  - Good for discovering novel transcripts in RNA-seq data
  - Builds exon models and splice junctions *de novo.*
  - Requires more CPU time and disk space
- STAR
  - https://code.google.com/p/rna-star/
  - Detects splice junctions *de novo*
  - Super fast: ~10min for 200M reads but
  - Requires 21Gb of memory
- More than 70 short-read aligners:
  - http://en.wikipedia.org/wiki/List_of_sequence_alignment_software

NGS Experiment

Data management:

Mapping the reads
Creating summaries

Downstream analysis: *the interesting stuff*
Differential expression, chimeric transcripts, novel transcribed regions, etc.



Analyzing RNA-Seq experiments

- How many molecules of $mRNA_1$ are in my sample?
  - Estimating expression

- Is the amount or $mRNA_1$ in sample/group A different from sample/group B ?
  - Differential analysis

Estimating expression: counting how many RNA-seq reads map to genes

- Using R
  - summarizeOverlaps in GenomicRanges
  - easyRNASeq
- Using Python
  - htseq-count
- How it works:
  - SAM/BAM files (TopHat2, STAR, …)
  - Gene annotation (GFF, GTF format)

GFF/GTF file format:

http://en.wikipedia.org/wiki/General_feature_format
http://useast.ensembl.org/info/website/upload/gff.html
http://www.sanger.ac.uk/resources/software/gff/
http://www.sequenceontology.org/gff3.shtml

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.

The GTF (General Transfer Format) is identical to GFF version 2.

- Fields
- Track lines
- More information

Fields

Fields **must** be tab-separated. Also, all but the final field in each feature line must contain a value; "empty" columns should be denoted with a '.'

1. **seqname** - name of the chromosome or scaffold; chromosome names can be given with or without the 'chr' prefix.
2. **source** - name of the program that generated this feature, or the data source (database or project name)
3. **feature** - feature type name, e.g. Gene, Variation, Similarity
4. **start** - Start position of the feature, with sequence numbering starting at 1.
5. **end** - End position of the feature, with sequence numbering starting at 1.
6. **score** - A floating point value.
7. **strand** - defined as + (forward) or - (reverse).
8. **frame** - One of '0', '1' or '2'. '0' indicates that the first base of the feature is the first base of a codon, '1' that the second base is the first base of a codon, and so on..
9. **attribute** - A semicolon-separated list of tag-value pairs, providing additional information about each feature.

Sample GFF output from Ensembl export:

```
X       Ensembl Repeat  2419108 2419128 42      .       .       hid=trf; hstart=1; hend=21
X       Ensembl Repeat  2419108 2419410 2502    -       .       hid=AluSx; hstart=1; hend=303
X       Ensembl Repeat  2419108 2419128 0       .       .       hid=dust; hstart=2419108; hend=2419128
X       Ensembl Pred.trans.     2416676 2418760 450.19  -       2       genscan=GENSCAN00000019335
X       Ensembl Variation       2413425 2413425 .       +       .
X       Ensembl Variation       2413805 2413805 .       +       .
```

Track lines

Although not part of the formal GFF specification, Ensembl will use track lines to further configure sets of features. Track lines should be placed at the beginning of the list of features they are to affect.

The track line consists of the word 'track' followed by space-separated key=value pairs - see the example below. Valid parameters used by Ensembl are:

- **name** - unique name to identify this track when parsing the file
- **description** - Label to be displayed under the track in Region in Detail
- **priority** - integer defining the order in which to display tracks, if multiple tracks are defined.

More information

For more information about this file format, see the documentation on the Sanger Institute website.

GFF/GTF file format:

http://en.wikipedia.org/wiki/General_feature_format
http://useast.ensembl.org/info/website/upload/gff.html
http://www.sanger.ac.uk/resources/software/gff/
http://www.sequenceontology.org/gff3.shtml

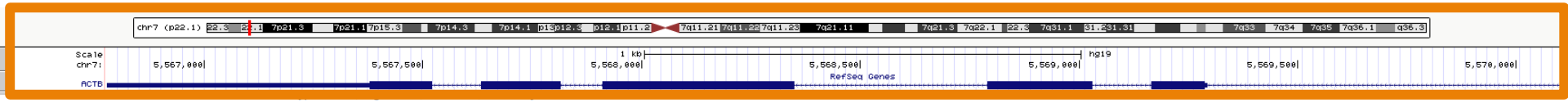
GFF/GTF File Format - Definition and supported options

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.

The GTF (General Transfer Format) is identical to GFF version 2.

- Fields
- Track lines
- More information

Fields



4. **start** - Start position of the feature, with sequence numbering starting at 1.
5. **end** - End position of the feature, with sequence numbering starting at 1.
6. **score** - A floating point value.
7. **strand** - defined as + (forward) or - (reverse).
8. **frame** - One of '0', '1' or '2'. '0' indicates that the first base of the feature is the first base of a codon, '1' that the second base is the first base of a codon, and so on..
9. **attribute** - A semicolon-separated list of tag-value pairs, providing additional information about each feature.

Sample GFF output from Ensembl export:

```
X       Ensembl Repeat  2419108 2419128 42      .       .       hid=trf; hstart=1; hend=21
X       Ensembl Repeat  2419108 2419410 2502    -       .       hid=AluSx; hstart=1; hend=303
X       Ensembl Repeat  2419108 2419128 0       .       .       hid=dust; hstart=2419108; hend=2419128
X       Ensembl Pred.trans.     2416676 2418760 450.19  -       2       genscan=GENSCAN00000019335
X       Ensembl Variation       2413425 2413425 .       +       .
X       Ensembl Variation       2413805 2413805 .       +       .
```

Track lines

Although not part of the formal GFF specification, Ensembl will use track lines to further configure sets of features. Track lines should be placed at the beginning of the list of features they are to affect.

The track line consists of the word 'track' followed by space-separated key=value pairs - see the example below. Valid parameters used by Ensembl are:

- **name** - unique name to identify this track when parsing the file
- **description** - Label to be displayed under the track in Region in Detail
- **priority** - integer defining the order in which to display tracks, if multiple tracks are defined.

More information

For more information about this file format, see the documentation on the Sanger Institute website.

Tutorial: RNA-seq count matrix

- Download
  - http://icb.med.cornell.edu/faculty/sboner/lab/EpigenomicsWorkshop/count_matrix.txt

- Load into R, inspect

Tutorial: RNA-seq count matrix

```
# working directory
getwd()

# read in count matrix
countData  <- read.csv("count_matrix.txt",
header=T, row.names=1, sep="\t")

dim(countData)

head(countData)
```

Read counts

GENE	ctrl1	ctrl2	ctrl3	treat1	treat2	treat3
0610005C13Rik	1438	1104	1825	1348	1154	1005
0610007N19Rik	1012	1152	1139	878	885	835
0610007P14Rik	704	796	881	826	865	929
0610009B22Rik	757	802	780	885	853	987
0610009D07Rik	1107	1183	1220	1258	1221	1428
…	…	…	…	…	…	…

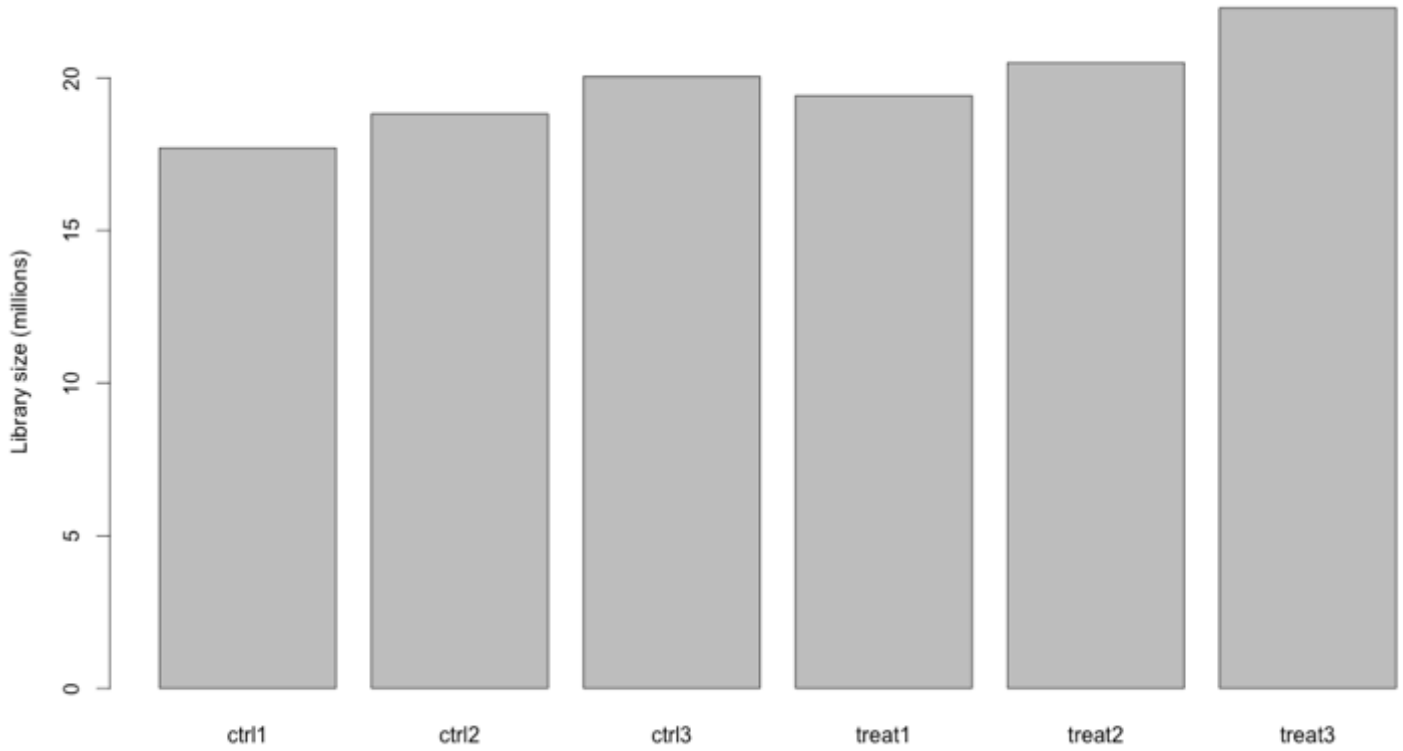
24009 rows, i.e. genes
6 columns, i.e. samples

Tutorial: Basic QC

```
barplot(colSums(countData)*1e-6,
    names=colnames(countData),
    ylab="Library size (millions)")
```

Tutorial: Basic QC

```
barplot(colSums(countData)*1e-6,
    names=colnames(countData),
    ylab="Library size (millions)")
```



Analyzing expression

- How many molecules of $mRNA_1$ are in my sample?
  - Estimating expression

- Is the amount or $mRNA_1$ in sample/group A different from sample/group B ?
  - Differential analysis

Tutorial: Installing BioConductor packages

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

biocLite("DESeq2")
```

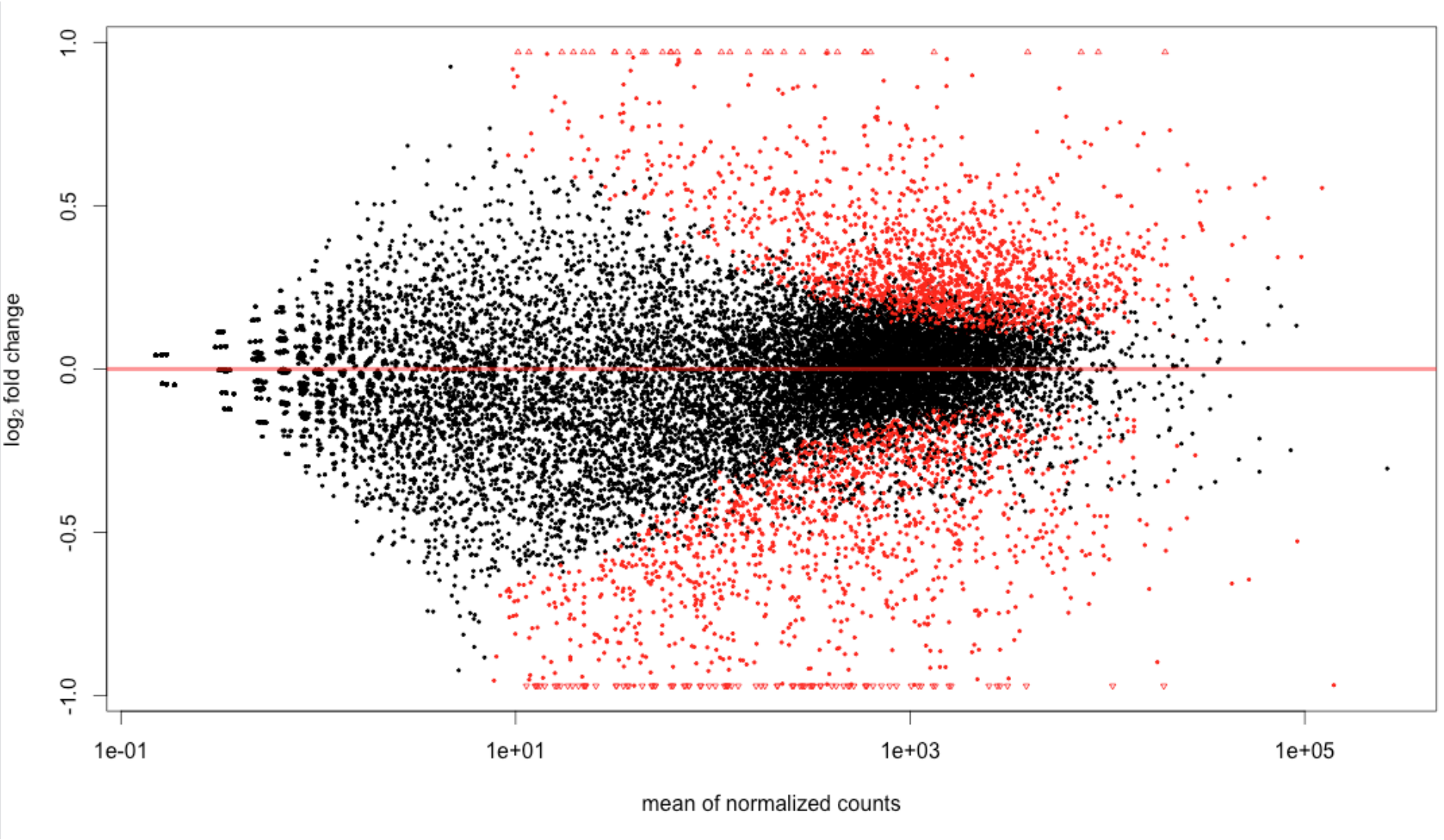
http://www.bioconductor.org/

M. I. Love, W. Huber, S. Anders: Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. bioRxiv (2014). doi: 10.1101/002832 [1]

Tutorial: DESeq2 analysis

```
# load library
library(DESeq2)

# create experiment labels (two conditions)
colData <- DataFrame(condition=factor(c("ctrl","ctrl",
"ctrl", "treat", "treat", "treat")))

# create DESeq input matrix
dds <- DESeqDataSetFromMatrix(countData, colData,
formula(~ condition))

# run DEseq
dds <- DESeq(dds)

# visualize differentially expressed genes
plotMA(dds)
```

Tutorial: DESeq2 analysis



Tutorial: DESeq2 analysis

```
# load library
library(DESeq2)

# create experiment labels (two conditions)
colData <- DataFrame(condition=factor(c("ctrl","ctrl", "ctrl", "treat", "treat", "treat")))

# create DESeq input matrix
dds <- DESeqDataSetFromMatrix(countData, colData, formula(~ condition))

# run DEseq
dds <- DESeq(dds)

# visualize differentially expressed genes
plotMA(dds)

# get differentially expressed genes
res <- results(dds)

# order by BH adjusted p-value
resOrdered <- res[order(res$padj),]

# top of ordered matrix
head(resOrdered)
```

Tutorial: DESeq2 analysis

```
# get differentially expressed genes
res <- results(dds)

# order by BH adjusted p-value
resOrdered <- res[order(res$padj),]

# top of ordered matrix
head(resOrdered)

DataFrame with 6 rows and 6 columns
          baseMean log2FoldChange      lfcSE      stat       pvalue         padj
         <numeric>      <numeric>  <numeric> <numeric>    <numeric>    <numeric>
Pck1    19300.0081     -2.3329116 0.16519373 -14.12228 2.768978e-45 3.986497e-41
Fras1    1202.1842     -0.8469410 0.06499738 -13.03039 8.219001e-39 5.916448e-35
S100a14   590.6305      2.1903041 0.17608923  12.43860 1.612985e-35 7.740716e-32
Ugt1a2   2759.7012     -1.7037495 0.15339576 -11.10689 1.161372e-28 4.180067e-25
Crip1     681.0106      0.7717364 0.07264577  10.62328 2.322502e-26 5.572844e-23
Smpdl3a 11152.4458      0.3398371 0.03195000  10.63653 2.014913e-26 5.572844e-23
```

```
# how many differentially expressed genes ? FDR=10%, |fold-change|>2 (up and down)
```

Tutorial: DESeq2 analysis

```
# how many differentially expressed genes ? FDR=10%, |fold-change|>2 (up and down)

# get differentially expressed gene matrix
sig <- resOrdered[!is.na(resOrdered$padj) &
      resOrdered$padj<0.10 &
      abs(resOrdered$log2FoldChange)>=1,]
```

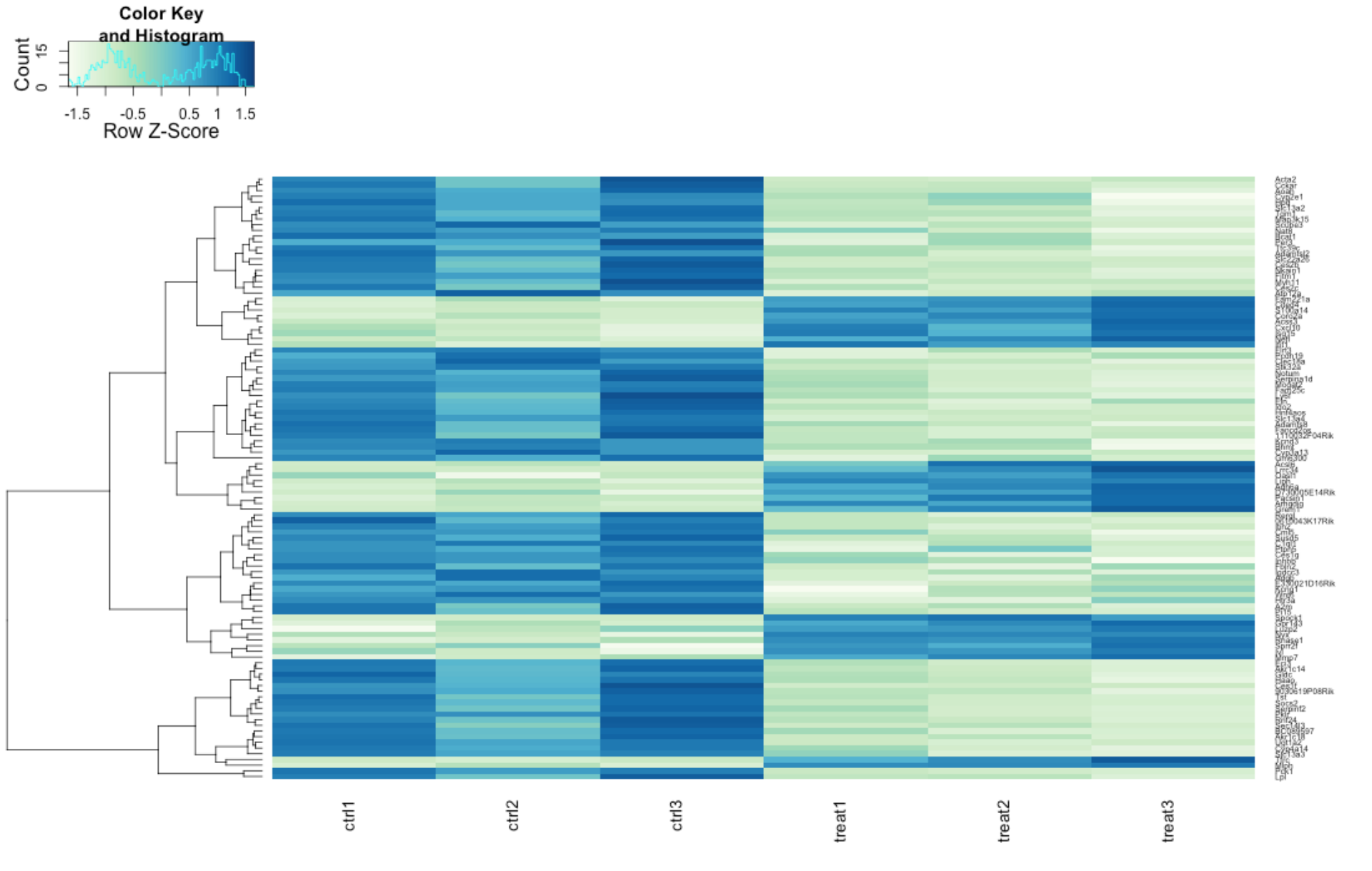
Tutorial: DESeq2 analysis

```
# how many differentially expressed genes ? FDR=10%, |fold-change|>2 (up and down)

# get differentially expressed gene matrix
sig <- resOrdered[!is.na(resOrdered$padj) &
     resOrdered$padj<0.10 &
     abs(resOrdered$log2FoldChange)>=1,]

head(sig)
DataFrame with 6 rows and 6 columns
         baseMean log2FoldChange     lfcSE      stat    pvalue      padj
        <numeric>      <numeric> <numeric> <numeric> <numeric> <numeric>
Pck1        19300          -2.33     0.165    -14.12  2.77e-45  3.99e-41
S100a14       591           2.19     0.176     12.44  1.61e-35  7.74e-32
Ugt1a2       2760          -1.70     0.153    -11.11  1.16e-28  4.18e-25
Pklr          787          -1.00     0.097    -10.34  4.62e-25  9.49e-22
Mlph         1321           1.20     0.117     10.20  1.90e-24  3.42e-21
Ifit1         285           1.39     0.156      8.94  3.76e-19  3.38e-16



dim(sig)

# how to create a heat map
```

Tutorial: Heat Map

```
# how to create a heat map

# select genes
selected <- rownames(sig);selected

## load libraries for the heat map
library("RColorBrewer")


source("http://bioconductor.org/biocLite.R")
biocLite("gplots")
library("gplots")

# colors of the heat map
hmcol <- colorRampPalette(brewer.pal(9, "GnBu"))(100) ## hmcol <- heat.colors

heatmap.2( log2(counts(dds,normalized=TRUE)[rownames(dds) %in% selected,]),
     col = hmcol, scale="row",
     Rowv = TRUE, Colv = FALSE,
     dendrogram="row",
     trace="none",
     margin=c(4,6), cexRow=0.5, cexCol=1, keysize=1 )
```

Tutorial: Heat Map



Selecting the most differentially expressed genes and run GO analysis

```
# universe
universe <- rownames(resOrdered)

# load mouse annotation and ID library
biocLite("org.Mm.eg.db")
library(org.Mm.eg.db)

# convert gene names to Entrez ID
genemap <- select(org.Mm.eg.db, selected, "ENTREZID", "SYMBOL")
univmap <- select(org.Mm.eg.db, universe, "ENTREZID", "SYMBOL")

# load GO scoring package
biocLite("GOstats")
library(GOstats)

# set up analysis
param<- new ("GOHyperGParams", geneIds = genemap, universeGeneIds=univmap, annotation="org.Mm.eg.db",
ontology="BP",pvalueCutoff=0.01, conditional=FALSE, testDirection="over")

# run analysis
hyp<-hyperGTest(param)

# visualize
summary(hyp)


## Select/sort on Pvalue, Count, etc.
```

Summary

- Intro of RNA-seq
- Estimating expression levels
- Differential expression analysis with DESeq2

- Andrea Sboner: ans2077@med.cornell.edu