UC DAVIS GENOME CENTER

SO YOU WANT TO DO A: RNA-SEQ EXPERIMENT

MATT SETTLES, PHD
UNIVERSITY OF CALIFORNIA, DAVIS
SETTLES@UCDAVIS.EDU

BIOINFORMATICS.UCDAVIS.EDU

Bioinformatics Core

Genome Center

UC Davis

DISCLAIMER

- This talk/workshop is full of opinion, there are as many different way to perform analysis as there are Bioinformaticians.
- My opinion is based on over a decade of experience and spending a considerable amount of time to understand the data and how it relates to the biological question.
- Each experiment is unique, this workshop is a starting place and should be adapted to the specific characteristics of your experiment.

ME

- B.S. Electrical Engineering
- M.S. Computer Science
- PhD Bioinformatics and Computational Biology

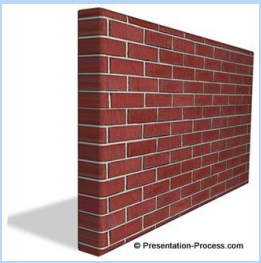
- Managed the Bioinformatics Core at Washington State University, 2007-2009
- Director of the Genomics Resources Core at the University of Idaho, 2009-2015
- Manager of the Bioinformatics Core at UC Davis

Perform Experiment

Sample and Extract RNA/DNA

Prepare Libraries

Sequence



Analyze

Interpret

OUTLINE

1. Introduction to High Throughput Sequencing and RNA-seq
   - High throughput sequencing
   - RNA-seq Experimental Design
   - RNA Sequencing and Sample Preparation
2. Workshop Dataset
3. Introduction to the Command Line
4. Overview of RNA-seq Data Analysis
   - Pipeline Workflow/Stages
   - Software
   - Metadata input
   - Files and Directory Structure

OUTLINE

5. Files and File Types
6. Read Preprocessing
   - Description of the expHTS preprocess pipeline
   - Parameters and what they mean
   - Preprocessing the Workshop Data
   - QA/QC
7. Read Mapping
   - Description of the expHTS mapping pipeline
   - Parameters and what they mean
   - Mapping the Workshop Data
   - QA/QC
8. Estimate known genes and transcripts expression – Counting
   - Description of the expHTS counting pipeline
   - Parameters and what they mean
   - Counting the Workshop Data
   - QA/QC

OUTLINE

9. Differential Expression Analysis using edgeR
   - Overview of Differential Expression Analysis
   - Models and model formulation
   - QA/QC
   - Perform Differential Expression the Workshop Data
10. Summarization and Visualization of Output

INTRODUCTION TO SEQUENCING

Section 1

HISTORY

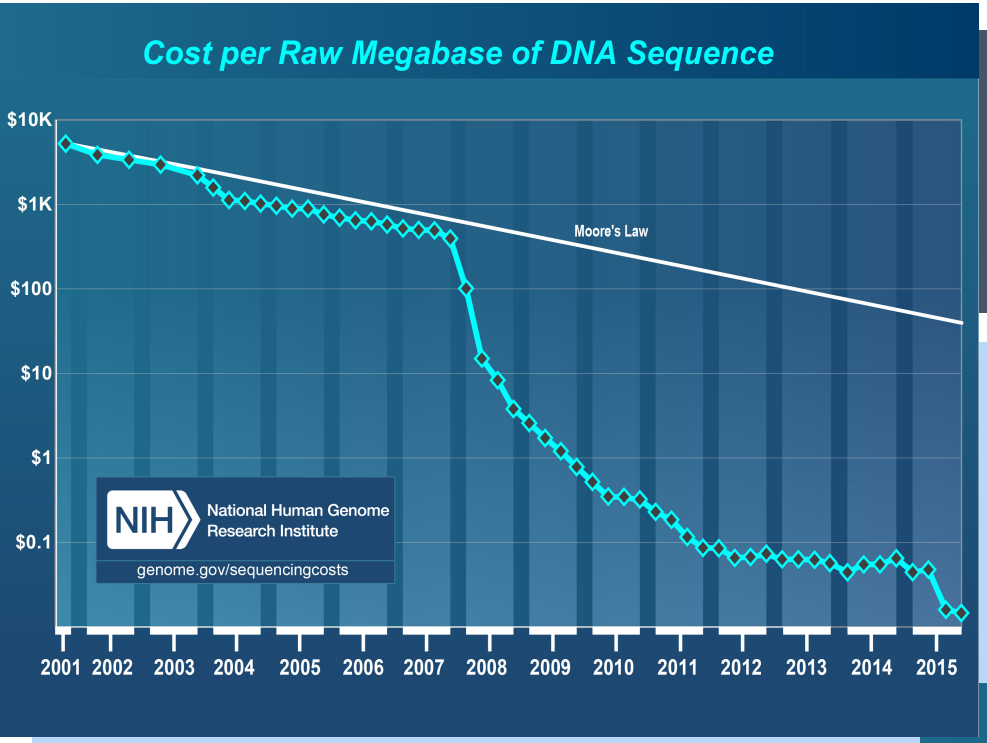
- It would take a few more decades after the discovery of the double helix in 1953 before we could readily analyze fragments of DNA.
- RNA sequencing actually preceded DNA sequencing when Walter Friers from the University of Ghent published the first complete gene and genome of Bacteriophage MS2 in 1972 and 1976 respectively.
- Location specific primer extension: Raw Wu (1970), using DNA polymerase catalysis and specific nucleotide labeling.
- Chain-terminating inhibitors: Frederick Sanger (1977), aided in speeding up the process

HISTORY

- Leroy E. Hood's laboratory at the California Institute of Technology announced the first semi-automated DNA sequencing machine in 1986.

- Applied Biosystems' produced the first fully automated sequencing machine, the ABI 370, in 1987, followed by the ABI Prism 373, (1990), ABI Prism 377 (1995), ABI Prism 310 (also 1995) represented the first capillary sequencer, ABI Prism 3700 (1999, the workhorse of the human genome project), ABI 3730xl DNA analyzer (2002) @ **2M bases per day**.

- Primer Walking
  - Design a primer that matches the sequence neighboring the unknown sequence
  - Sequence the short DNA strand using the Sanger method
  - The new sequenced portion is used to design a new primer and repeated
- *de novo* sequencing or “shotgun sequencing”
  - High-molecular weight DNA is sheared into random
  - fragments
  - Shorter fragments are cloned into a vectors
  - clones are sequenced from both ends, creating two ”reads”
  - original sequence is reconstitutes by ”assembling” the reads

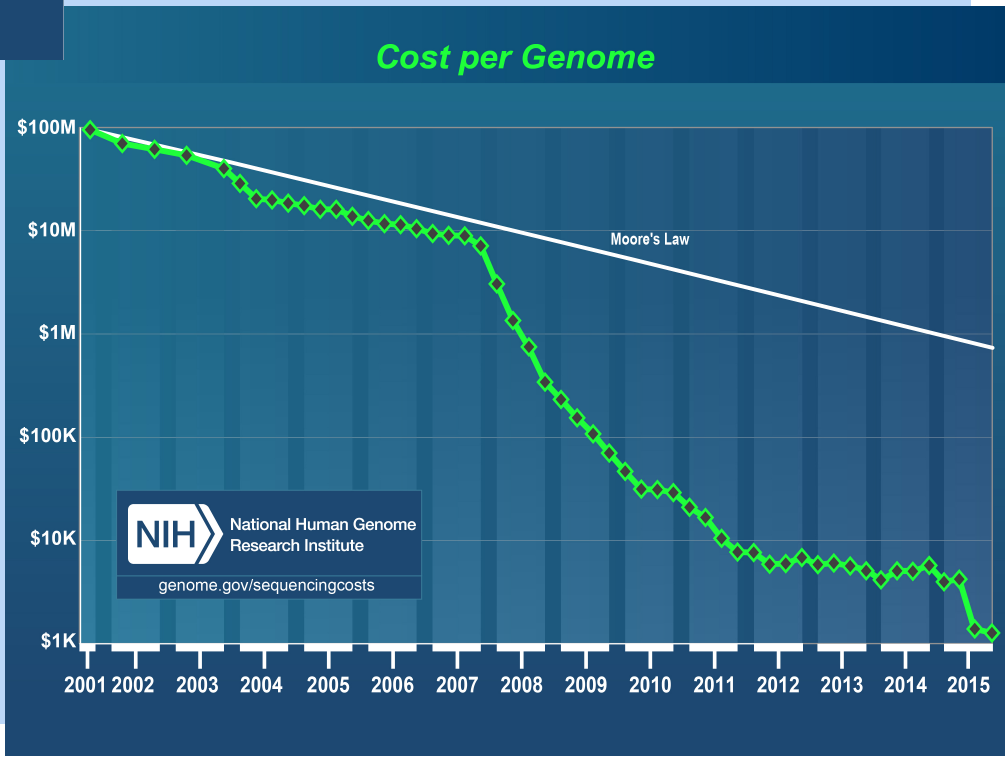


SEQUENCING COSTS

GROWTH IN SEQUENCING

- October – 2015
  - $0.014 per Megabase
  - $1,245 per Human Sized Genome (30x coverage)

Data Only!!
Does not include any analysis or bioinformatics



ROCHE 454

- The first massively parallel method (aka “next generation sequencing”) to become commercially available was developed by 454 Life Sciences in 2005 (acquired by Roche in 2007) and is based on the pyrosequencing technique. Similar to the Sanger method, sequencing is carried out using primed synthesis by DNA polymerase. However in the 454 pyrosequencing method, the DNA fragments are presented with each of the four dNTPs sequentially and without a dye-terminator, as is done with Sanger sequencing, allowing for multiple incorporation in the same flow. The amount of the incorporation is monitored by luminometric detection of the pyrophosphate released (hence the name ”pyrosequencing”).

ILLUMINA (SOLEXA)

- The second “next generation” sequencing technology to be released (in 2006) was Illumina Solexa sequencing. A key difference between Roche 454 and Illumina sequencing was the use of chain-terminating nucleotides. The fluorescent label on the terminating base can be removed to leave an unblocked 3’ terminus, mating the chain termination a reversible process. The method thus sequences one base at a time, rather than 0 or more bases as does Roche 454.

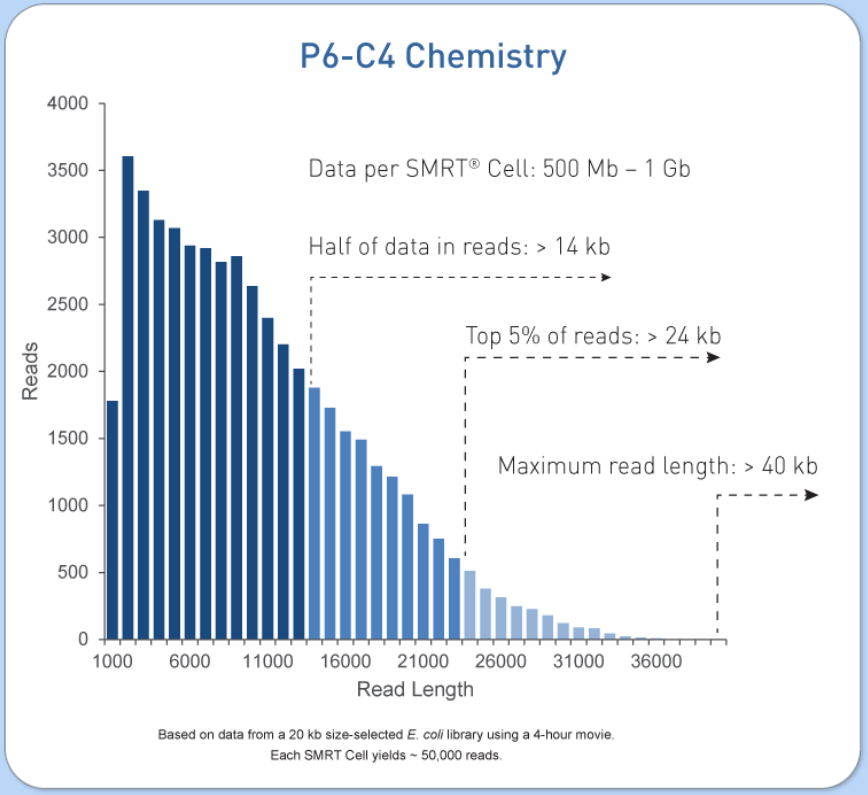
LIFE TECHNOLOGIES

- Ion Torrent PGM, first available in 2011, generates up to 400bp reads (reported) and up to 2Gb (5.5m reads) per run. Cheap fast runs. Ion Proton system can generate up to 10Gb per run. Generates flowgrams and SFF files similar to Roche 454 data as well as the standard fastq files. Ion Torrent is also considered a second generation sequencer.

PACIFIC BIOSYSTEMS

- Pacific Biosystems is so far the most successful third generation DNA sequencing system. Key differences are that its a single molecule, real time (SMRT) technology and capable of producing sequences of multi-kilobases.

Third generation sequencers are single molecule sequencers.



Iso-seq on Pac Bio possible, transcriptome without 'assembly'

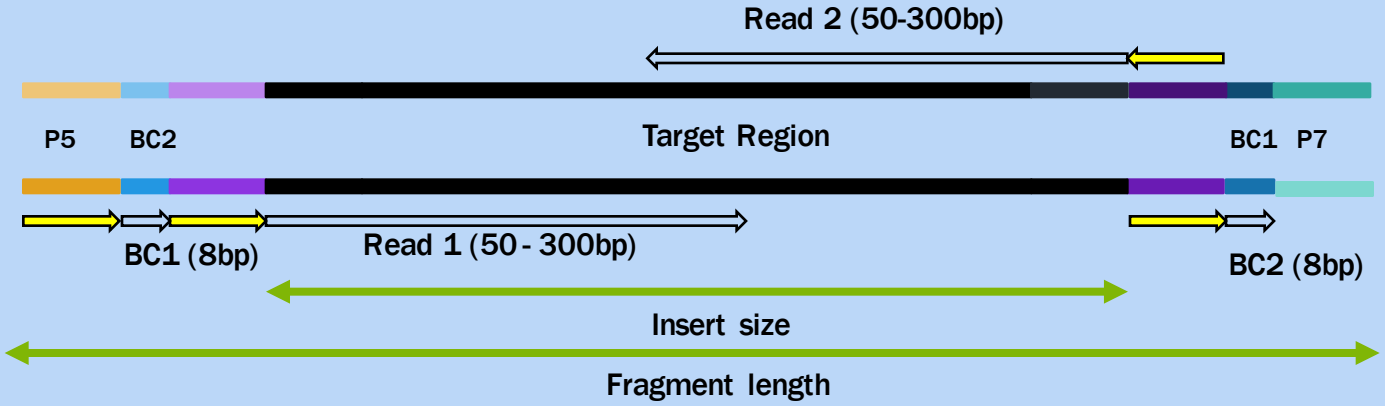
OXFORD NANOPORE

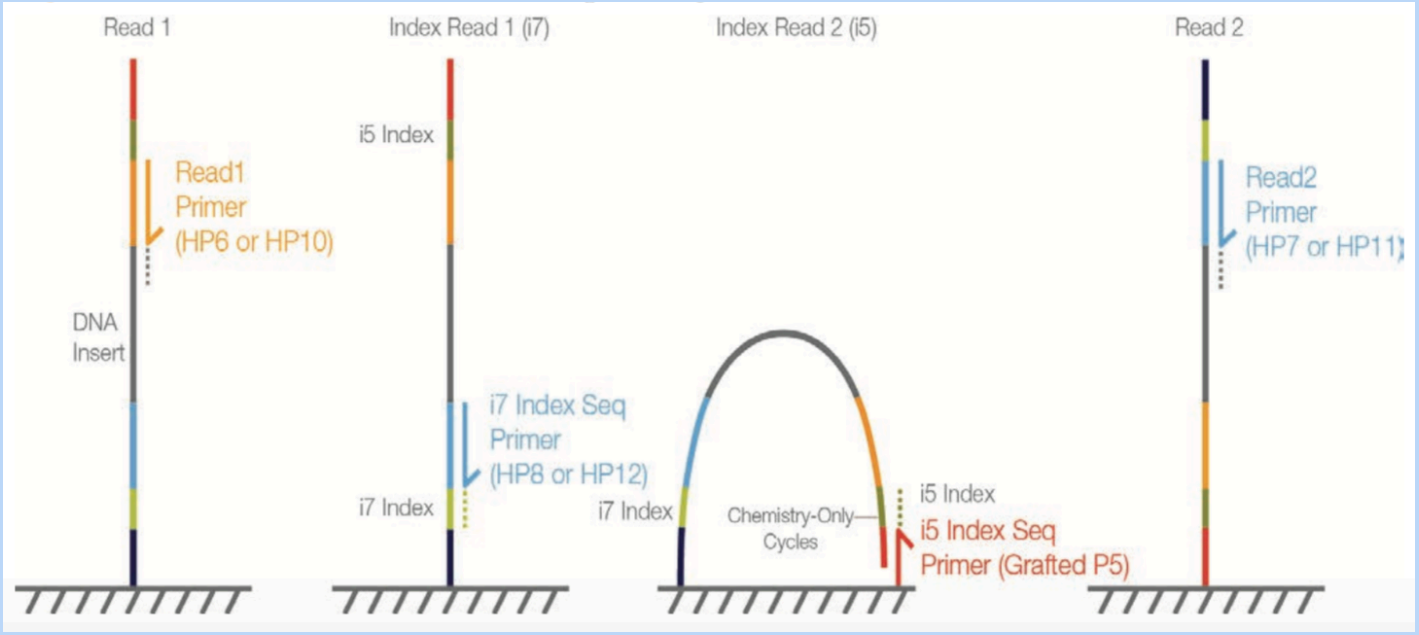
- Another 3rd generation sequencer, founded in 2005 and recently (May 2015) produced a commercial product, the MinION. The sequencer uses nanopore technology developed in the 90's to sequence single molecules. Throughput is about > 1Gb per flowcell.

FYI: 4th generation sequencing is being described as "In-situ sequencing", for applications such as "spatial transcriptomics"

ILLUMINA SEQUENCING

- Illumina SBS





ILLUMINA SEQUENCING

- http://www.illumina.com/systems/hiseq-3000-4000/specifications.html

2500

MiSeq

	HISEQ 3000 SYSTEM	HISEQ 4000 SYSTEM
No. of Flow Cells per Run	1	1 or 2
Data Yield: 2 × 150 bp 2 × 75 bp 1 × 50 bp	 650-750 Gb 325-375 Gb 105-125 Gb	 1300-1500 Gb 650-750 Gb 210-250 Gb
Clusters Passing Filter (Single Reads) (8 lanes per flow cell)	2.1-2.5 billion	4.3-5 billion
Quality Scores: 2 × 50 bp 2 × 75 bp 2 × 150 bp	 ≥ 85% bases above Q30 ≥ 80% bases above Q30 ≥ 75% bases above Q30	 ≥ 85% bases above Q30 ≥ 80% bases above Q30 ≥ 75% bases above Q30
Daily Throughput	> 200 Gb	> 400 Gb
Run Time	< 1-3.5 days	< 1-3.5 days
Human Genomes per Run*	up to 6	up to 12
Exomes per Run**	up to 48	up to 96
Transcriptomes per Run***	up to 50	up to 100

Perform

Visit the Illumina website

- How many sequencing machines does Illumina Have?
- What are the differences between the HiSeq 2500 series and HiSeq 3000/4000 series machines?
- What are differences in read length options between the MiSeq and HiSeq?
- What are the differences between the HiSeq's High Output Run Mode and the Rapid Run Mode
- How many reads (and bp) do you get on one lane of a HiSeq 2500 in High Output Run Mode for 1x50 and 2x100, compare those to what you get on the HiSeq 3000

HOMEWORK 1

Getting to know Illumina

EXPERIMENTAL DESIGN

- In high throughput biological work (Microarrays, Sequencing, HT Genotyping, etc.), what may seem like small technical artifacts introduced during sample extraction/preparation can lead to large changes, or technical bias, in the data.
  - Not to say this doesn't occur with smaller scale analysis such as Sanger sequencing or qRT-PCR, but they do become more apparent and may cause significant issues during analysis.

GENERAL RULES FOR PREPARING SAMPLES

- Prepare more samples then you are going to need, i.e. expect some will be of poor quality, or fail
- Preparation stages should occur across all samples at the same time (or as close as possible) and by the same person
- Spend time practicing a new technique to produce the highest quality product you can, reliably
- Quality should be established using Fragment analysis traces (pseudo-gel images, RNA RIN > 7.0)
- DNA/RNA should not be degraded
  - 260/280 ratios for RNA should be approximately 2.0 and 260/230 should be between 2.0 and 2.2. Values over 1.8 are acceptable
- Quantity should be determined with a Fluorometer, such as a Qubit.

BE CONSISTENT ACROSS ALL SAMPLES!!!

SEQUENCING DEPTH

- The first and most basic question is how many base pairs of sequence data will I get
  Factors to consider are:
  - 1. Number of reads being sequenced
  - 2. Read length (if paired consider then as individuals)
  - 3. Number of samples being sequenced
  - 4. Expected percentage of usable data

$$bpPerSample = \frac{readLength * readCount}{sampleCount} * 0.8$$

- The number of reads and read length data are best obtained from the manufacturer's website (search for specifications) and always use the lower end of the estimate.

SEQUENCING COVERAGE

- Once you have the number of base pairs per sample you can then determine expected coverage
  Factors to consider are:
  - 1. Length of the genome (in bp)
  - 2. Any extra-genomic sequence, or contamination

$$expectedCoverage = \frac{bpPerSample}{totalGenomicContent}$$

- Coverage is determined differently for "Counting" based experiments (RNAseq, amplicons) where an expected number of reads per sample is typically more suitable.

RNA-SEQ

Characterization of transcripts or differential gene expression

Factors to consider are:

- Read length needed depends on likelihood of mapping uniqueness, but generally longer is better and paired-end is better than single-end. (2 x >75bp is recommended)
- Interest in measuring genes expressed at low levels (<< level, the >> the depth and necessary complexity of library)
- The fold change you want to be able to detect (< fold change more replicates, more depth)
- Detection of novel transcripts, or quantification of isoforms requires >> sequencing depth

The amount of sequencing needed for a given sample is determined by the goals of the experiment and the nature of the RNA sample.

GENERATING RNA-SEQ LIBRARIES

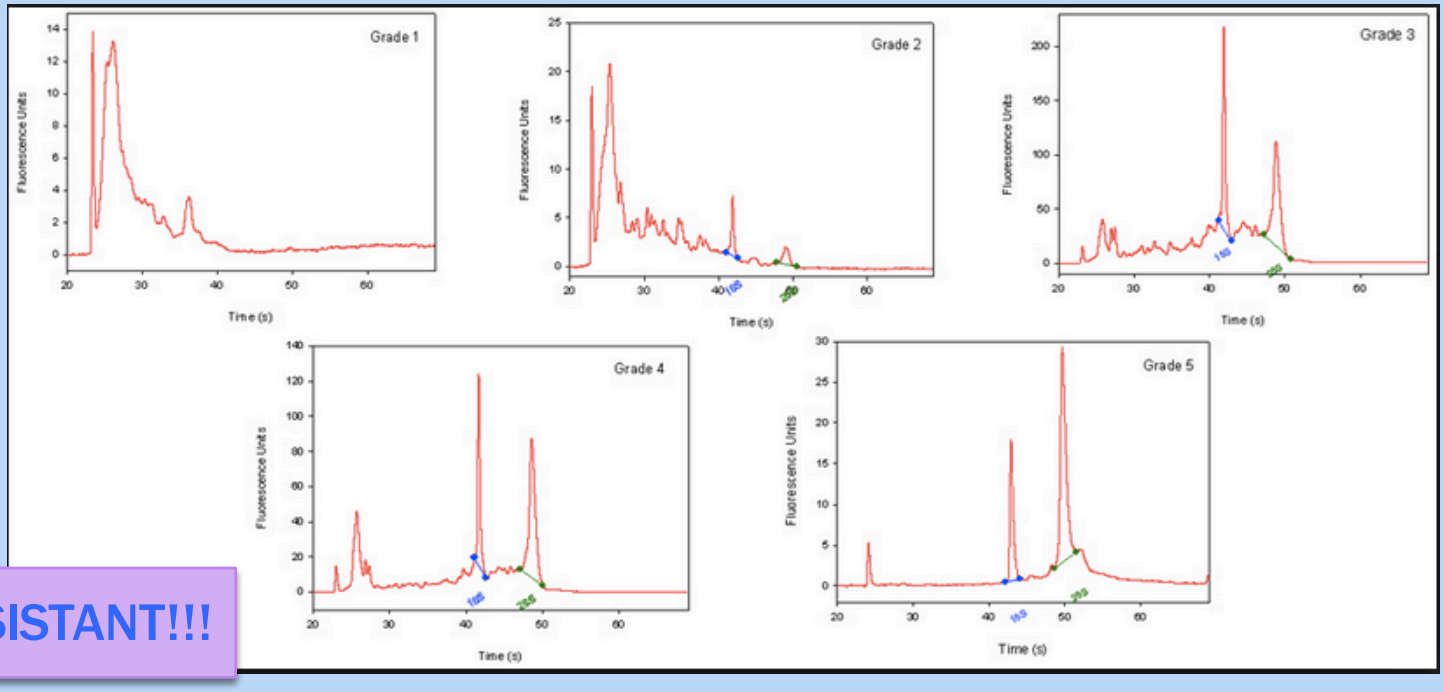
Considerations

- QA/QC of RNA samples
- RNA of interest
- Library Preparation
  - Stranded Vs. Unstranded
- Size Selection/Cleanup
  - Final QA

QA/QC OF RNA SAMPLES

RNA Quality and RIN (RQN on AATI Fragment Analyzer)

- RNA sequencing begins with high-quality total RNA, only an Agilant BioAnalyzer (or equivalent) can adequately determine the quality of total RNA samples. RIN values between 7 and 10 are desirable.



BE CONSISTANT!!!

RNA OF INTEREST

- From total RNA we extract "RNA of interest". Primary goal is to NOT sequence 90% (or more) ribosomal RNAs, which are the most abundant RNAs in the typical sample. there are two main strategies for enriching your "RNA of interest".
  - polyA selection. Enrich mRNA (those with polyA tails) from the sample by oligo dT affinity.
  - rRNA depletion. rRNA knockdown using RiboZero (or Ribominus) is mainly used when your experiment calls for sequencing non-polyA RNA transcripts and non-coding RNA (ncRNA) populations. This method is also usually more costly.

rRNA depletion will result in a much larger proportion of reads align to intergenic and intronic regions of the genome.

LIBRARY PREPARATION

- Some library prep methods first require you to generate cDNA, in order to ligate on the Illumina barcodes and adapters.
  - cDNA generation using oligo dT (3' biased transcripts)
  - cDNA generation using random hexomers (less biased)
  - full-length cDNAs using SMART cDNA synthesis method
- Also, can generate strand specific libraries, which means you only sequence the strand that was transcribed.
  - This is most commonly performed using dUDP rather than dNTPs in cDNA generation and digesting the "rna" strand.
  - Can also use a RNA ligase to attach adapters and then PCR the second strand and remainder of adapters.

SIZE SELECTION/CLEANUP/QA

Final insert size optimal for DE are ~ 150bp

- Very important to be consistent across all samples in an experiment on how you size select your final libraries. You can size select by:
  - Fragmenting your RNA, prior to cDNA generation.
    - Chemically heat w/magnesium
    - Mechanically (ex. ultra-sonicator)
- Cleanup/Size select after library generation using SPRI beads or (gel cut)
- QA the samples using an electrophoretic method (Bioanalyzer) and quantify with qPCR.

Most important thing is to be consistent!!!

BE CONSISTANT!

- In high throughput biological work (Microarrays, Sequencing, HT Genotyping, etc.), what may seem like small technical artifacts introduced during sample extraction/preparation can lead to large changes, or bias, in the data.
- Not to say this doesn't occur with smaller scale analysis such as Sanger sequencing or qRT-PCR, but they do become more apparent and may cause significant issues during analysis.

COST ESTIMATION

- RNA extraction QA/QC (Bioanalyzer)
- Enrichment of RNA of interest + Library Preparation
  - Library QA/QC (Bioanalyzer and Qubit)
  - Pooling ($10/library)
- Sequencing (Number of Lanes)
- Bioinformatics (General rule is to estimate the same amount as data generation, i.e. double your budget)

http://dnatech.genomecenter.ucdavis.edu/prices/

Example: 12 samples, ribo-depletion libraries, target 30M reads per sample, Hiseq 3000 (2x100).

COST ESTIMATION

- 12 Samples
  - QA Bioanalyzer = $98 for all 12 samples
  - Library Preparation (ribo-depletion) = $383/sample = $4,596
- Sequencing = $2346 per lane
  - 2.1 - 2.5 Billion reads per run / 8 lanes = Approximately 300M reads per lane
  - Multiplied by a 0.8 buffer equals 240M expected good reads
  - Divided by 12 samples in the lane = 20M reads per sample per lane.
  - Target 30M reads means 2 lanes of sequencing $2346x2 = $4692
- Bioinformatics, simple pairwise comparison design, DE only
  - $2000

Total = $98 + $4596 + $4692 + $2000 = $11,386

Approximately $950 per sample

Perform

- Search the web for RNA-SEQ coverage recommendations.
- Cost estimate a project to perform a DE analysis on 8 samples, polyA enrichment, stranded libraries, 2x100 on a HiSeq 3000 targeting approximately 10M reads per sample

HOMEWORK 2

RNA-SEQ COVERAGE

PREREQUISITES

- Access to a multi-core (24 cpu or greater), ‘high’ memory 64Gb or greater Linux server.
- Familiarity with the ’command line’.
- Basic knowledge of how to install software
- Basic knowledge of R and statistical programming
- Basic knowledge of Statistics and model building

WORKSHOP DATASET

Section 2

WORKSHOP DATA

- Exophiala dermatitidis NIH/UT8656

Exophiala dermatitidis is a thermophilic black yeast, and a member of the Herpotrichiellaceae. While the species is only found at low abundance in nature, metabolically active strains are commonly isolated in saunas, steam baths, and dish washers. Exophiala dermatitidis only rarely causes infection in humans, however cases have been reported around the world. In East Asia, the species has caused lethal brain infections in young and otherwise healthy individuals. The fungus has been known to cause cutaneous and subcutaneous phaeohyphomycosis, and as a lung colonist in people with cystic fibrosis in Europe. In 2002, an outbreak of systemic E. dermatitidis infection occurred in women who had received contaminated steroid injections at North Carolina hospitals.

WORKSHOP DATA

Illumina sequencing of Exophiala dermatitidis NIH/UT8656

- Reference transcriptome for the Human Microbiome Project

- Data:
  - http://www.ncbi.nlm.nih.gov/Traces/sra/?study=SRP006291
- Article:
  - http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4059230
- References:
  - Broad - http://www.broadinstitute.org/annotation/genome/Black_Yeasts/MultiHome.html
  - Ensembl –
  - http://fungi.ensembl.org/Exophiala_dermatitidis_nih_ut8656/Info/Index

Perform

- Briefly scan the data publication
  - What is the experimental condition: what are we going to compare?
  - Scanning the document, what was their workflow
  - How many differentially expressed genes did they get (up regulated / down regulated)?
  - What was the main result of their experiment?

HOMEWORK 3

Workshop Data

THE COMMAND LINE

SECTION 3

THE COMMAND LINE

- The command line is powerful and the preferred way to run bioinformatics tools

BASICS:

Prompt msettles@MacBook-Pro:~$

~ is your home director

… $ command [parameters] [files] ENTER

parameters begin with a - short parameter or -- long parameter

Help … $ man command

… $ command –h

… $ command –help

Tab complete!!

THE COMMAND LINE

Command input/output:

- file/folder on the command line, either as a positional argument or a parameter, or defaults
- stdin aka 'standard in', input pipe
- stdout aka 'standard out', output pipe
- stderr aka 'standard error', pipe for error messages

Special characters:

|	vertical bar is the pipe, it pipes the stdout of one command to the stdin of another cmd
< > 2>	feeds stdin, stdout, stderr to the command
>>	append
&	at the end of a command will run the command in the background
.	Current directory
..	Up one folder
/	folder delimiter
*	wild character, match anything

When naming files/folders avoid special characters and spaces (use . and _)!!

THE COMMAND LINE

Basics Commands:

pwd	print the working directory (current dir)
ls [file/dir]	list the current contents of the directory
ls –lah [file/dir]	long list, human readable, show hidden of the directory
cd to	change directory
cp f1 f2	copy file f1 to file f2
mv from to	mv directory (also way to rename)
mkdir dir	make directory
rm file	remove file
rmdir dir	remove an empty directory
rm –r dir	recursively remove a non-empty directory
nano file	edit a document
cat f1 [f2]	print to stdout file f1 then if provided file f2 (concatenate)
less file	view file, one page at a time
head file	view the first 10 lines of a file
tail file	view the last 10 lines of a file

Perform

- Look at your home directory, what files are already there, what files are hidden
- Create a new folder
- Move into that folder and create a file (write anything)
- Show that file, using less and cat
- Go back to your home directory
- Create a second folder
- Copy the file from the first directory to the new directory
- Move to your home directory
- Remove both folders
- Play with the new commands

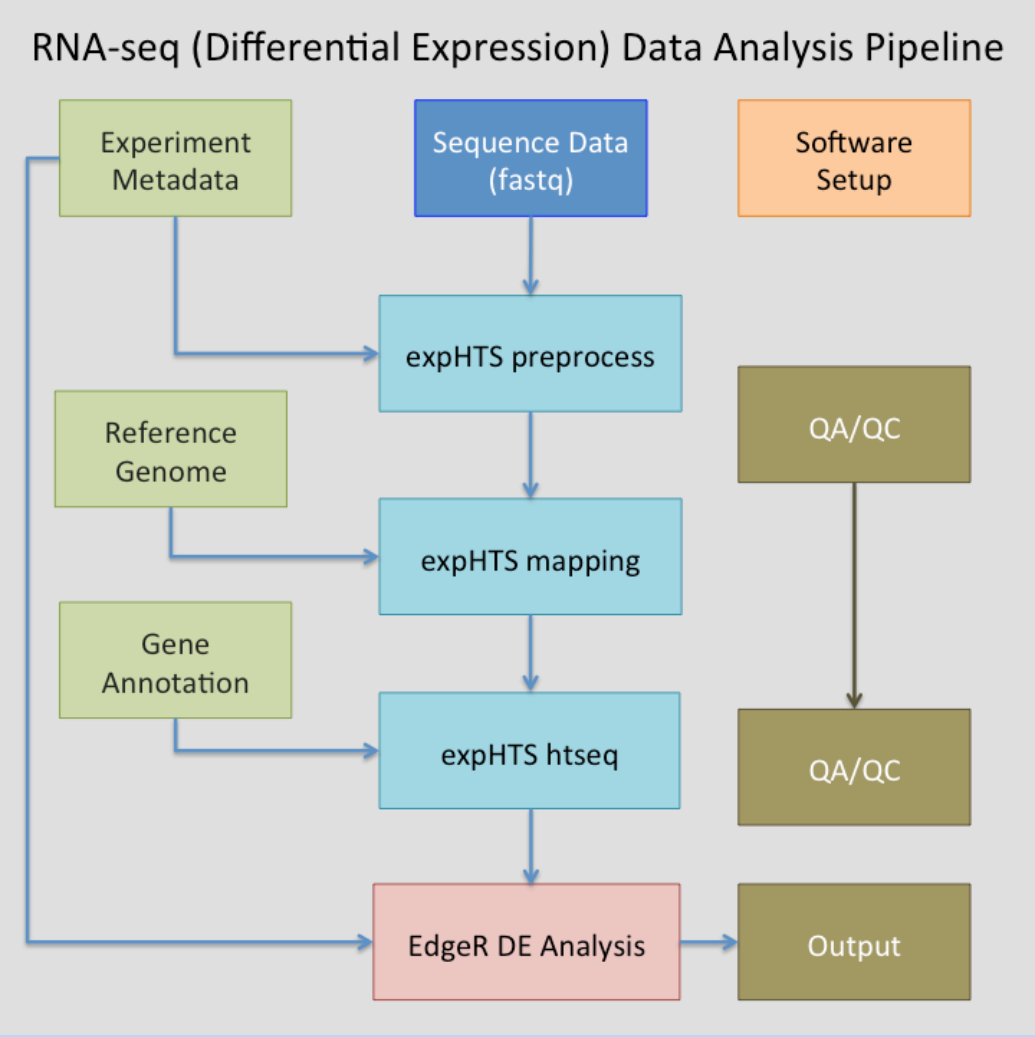
HOMEWORK 4

Playing with the command line

OVERVIEW OF RNA-SEQ DATA ANALYSIS

Section 4

RNA-SEQ PIPELINE OVERVIEW



SOFTWARE

Preprocessing:

- Python 2.7
  - Modules: argparse, optparse, distutils
- bowtie2 - contaminant screening
  - http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
- Super-Deduper – Identify and remove PCR duplicates
  - https://github.com/dstreett/Super-Deduper
- Sickle – Trim low quality regions
  - https://github.com/dstreett/sickle
- Scythe – Identify and remove adapters in SE reads
  - https://github.com/ucdavis-bioinformatics/scythe
- FLASH2 – Join overlapping reads, identify and remove adapter in PE reads
  - https://github.com/dstreett/FLASH2

SOFTWARE

Mapping:

- Python 2.7
- Bwa mem – map reads to a reference
  - http://sourceforge.net/projects/bio-bwa/files/
- samtools – processing of sam/bam file
  - http://www.htslib.org/

Read Counting:

- Python 2.7
- samtools – processing of sam/bam file
  - http://www.htslib.org/
- HTeq-0.6.1 htseq_count – count reads occurrences within genes
  - http://www-huber.embl.de/users/anders/HTSeq/

SOFTWARE

Analysis of differential expression:

- R 3.2.0 (or greater)
  - http://www.r-project.org/
  - R Package: optparse from cran
  - R Package: EdgeR from bioconductor – differential expression analysis
    - http://bioconductor.org/packages/release/bioc/html/edgeR.html
- RStudio
  - https://www.rstudio.com/

GIT

- Git is an open source program for tacking changes in text files. Its was written by the author of the Linux operating system, Linus Torvalds. Use git for:
  - Tracking software changes
  - Tracking notes changes
  - Tracking document changes
- Github is a web-based Git repository hosting service, free to use for open repositories
  - This workshop is available via git and github
  - https://github.com/msettles/Workshop_RNAseq

GIT

- git clone [repo] –to clone a copy of a repository from the remote server (github) onto your workstation. With your clone you can edit the documents as you please
- git status – displays the differences between your repositories and the current working head (changes you've made).
- git pull – when you fetch in changes to the repository and merging them in. for example if I edited the workshop repository, you can then pull in those changes to your workstation

Perform

- Find the expHTS repository on github
  - Using git clone the repository to your desktop
  - Go into the directory (cd)
  - Install expHTS with
    - python setup.py install

HOMEWORK 5

Install expHTS using git and python

METADATA FILE

- Variables that describe each sample, including those that will be used to compare samples to each other (experimental factors). Also good to include all technical factors that may influence experimental results (ex. Day of RNA isolation) in order to test for effect later.

- samples.txt – is a plain text tab delimited metadata file that will be used within the workshop to run expHTS and the R differential expression analysis. Rows are samples, columns are metadata
- Two REQUIRED columns, add more columns as you need
  - "SEQUENCE_ID" – folder name containing the sequences
  - "SAMPLE_ID" – Name in which to assign the sample

FILES AND DIRECTORY STRUCTURE

We use a strict directory structure to show the relationship between results and input, expHTS assumes this directory structure though it can be changed.

- PARENT folder, name of the experiment
  - 00-RawData
    - SEQUENCE_ID_1
      - Fastq Files
    - SEQUENCE_ID_2
      - Fastq Files
  - 02-Cleaned
    - SAMPLE_ID_1
      - Fastq Files
    - SAMPLE_ID_2
      - Fastq Files
    - Preprocessing_Summary.log

FILES AND DIRECTORY STRUCTURE

- 03-BWA
  - SAMPLE_ID_1
    - BAM Files
  - SAMPLE_ID_2
    - BAM Files
  - Mapping_Summary.log
- 04-HTseqCounts
  - SAMPLE_ID_1
    - Counts Files
  - SAMPLE_ID_2
    - Counts Files
  - Counts_Summary.log
- Reference
  - Reference fasta
  - Reference gtf file
- samples.txt

Perform

- Create the main experiment folder and 00-RawData folder
- Move the supplied fastq files to the 00-RawData folder
- Create the samples.txt text file for the experiment.
- Download the reference genome fasta file and corresponding gene annotation gtf/gff file

HOMEWORK 6

Lets Set up the Experiment

FILES AND FILE TYPES

Section 5

SEQUENCING READ FILES

fasta files

```
>sequence1
ACCCATGATTTGCGA
```

qual files

```
>sequence1
40 40 39 39 40 39 40 40 40 40 20 20 36 39 39
```

fastq files

```
@sequence1
ACCCATGATTTGCGA
+
IIHHIHIIII55EHH
```

QUALITY SCORES

$$Q = -10 log_{10} P$$

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	99.9%
40	1 in 10000	99.99%

QSCORE CONVERSION

$Q_{sanger} = -10log_{10}P$ - based on probability (aka phred)
$Q_{solexa} = -10log_{10}\frac{P}{1-P}$ - based on odds

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)
L - Illumina 1.8+	Phred+33,	raw reads typically (0, 41)

ILLUMINA READ NAMING CONVENTIONS

CASAVA 1.8 Read IDs

- @EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
  - EAS139 the unique instrument name
  - 136 the run id
  - FC706VJ the flowcell id
  - 2 flowcell lane
  - 2104 tile number within the flowcell lane
  - 15343 'x'-coordinate of the cluster within the tile
  - 197393 'y'-coordinate of the cluster within the tile
  - 1 the member of a pair, 1 or 2 (paired-end or mate-pair reads only)
  - Y Y if the read fails filter (read is bad), N otherwise
  - 18 0 when none of the control bits are on, otherwise it is an even number
  - ATCACG index sequence

SAM/BAM FILES

- SAM (Sequence Alignment/Map) format = unified format for storing read alignments to a reference sequence(Consistent since Sept. 2011).
  See http://samtools.sourceforge.net/SAM1.pdf
- BAM = binary version of SAM for fast querying

SAM/BAM FILES

SAM files contain two regions

- The header section
  - Each header line begins with character '@' followed by a two-letter record type code
- The alignment section
  - Each alignment line has 11 mandatory fields. These fields always appear in the same order and must be present, but their values can be '0' or '*', if the corresponding information if unavailable, or not applicable.

SAM COLUMNS

```
7172283    163    chr9    139389330    60    90M    =    139389482    242    TAGGAGG...    EHHHHHH...
7705896     83    chr9    139389513    60    90M    =    139389512    -91    GCTGGGG...    EBCHHFC...
7705896    163    chr9    139389512    60    90M    =    139389513     91    AGCTGGG...    HHHHHHH...
```

1	QNAME	query template name
2	FLAG	bitwise flag
3	RNAME	reference sequence name
4	POS	1-based leftmost mapping position
5	MAPQ	mapping quality
6	CIGAR	CIGAR string
7	RNEXT	reference name of mate
8	PNEXT	position of mate
9	TLEN	observed template length
10	SEQ	sequence
11	QUAL	ASCII of Phred-scaled base quality

SAM FLAGS

Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being reversed
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate
0x800	supplementary alignment

MAPQ EXPLAINED

- MAPQ, contains the "phred-scaled posterior probability that the mapping position" is wrong.
- In a probabilistic view, each read alignment is an estimate of the true alignment and is therefore also a random variable. It can be wrong. The error probability is scaled in the Phred. For example, given 1000 read alignments with mapping quality being 30, one of them will be incorrectly mapped to the wrong location on average.

MAPQ EXPLAINED

- The calculation of mapping qualities is simple, but this simple calculation considers many of the factors below:
  - The repeat structure of the reference. Reads falling in repetitive regions usually get very low mapping quality.
  - The base quality of the read. Low quality means the observed read sequence is possibly wrong, and wrong sequence may lead to a wrong alignment.
  - The sensitivity of the alignment algorithm. The true hit is more likely to be missed by an algorithm with low sensitivity, which also causes mapping errors.
  - Paired end or not. Reads mapped in pairs are more likely to be correct.

MAPQ EXPLAINED

- When you see a read alignment with a mapping quality of 30 or greater, it usually implies:
  - The overall base quality of the read is good.
  - The best alignment has few mismatches.
  - The read has few or just one 'good' hit on the reference, which means the current alignment is still the best even if one or two bases are actually mutations, or sequencing errors.

In practice however, each mapper seems to compute the MAPQ in their own way.

SAM CIGAR

- Compact Idiosyncratic Gapped Alignment Report (CIGAR) SAM flag field:

Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
H	5	hard clipping (clipped sequences NOT present in SEQ)
P	6	padding (silent deletion from padded reference)
=	7	sequence match
X	8	sequence mismatch

CIGAR EXAMPLE

	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2
Ref Pos:	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
Reference:	C	C	A	T	A	C	T		G	A	A	**C**	**T**	G	A	C	T	A	A	C
Read:					A	C	T	A	G	A	A	**T**	**G**	G		C	T			

POS: 5

CIGAR: 3M1I6M1D2M

** mismatches are not considered in standard CIGAR

GFF/GTF FILES

- The GFF (General Feature Format) format consists of one line per feature, each containing 9 columns of data (fields). The GTF (General Transfer Format) is identical to GFF version 2.
- Fields must be tab-separated and all fields must contain a value; "empty" fields should be denoted with a '.'.
- Columns:
  - Seqname: Name of the sequence chromosome
  - Source: the program, or database, that generated the feature
  - Feature: feature type name, (e.g. gene, exon, cds, etc.)
  - Start: start position of the feature, sequences begin at 1
  - End: stop position of the feature, sequences begin at 1
  - Score: a floating point value (e.g. 0.01)
  - Strand: Defined as '+' (forward),or '-' (reverse)
  - Frame: One of '0', '1', '2', '0' represents the first base of a codon.
  - Attribute: A semicolon-separated list of tag-value pairs, providing additional information about each feature.

GFF/GTF FILES

Sample GTF output from Ensembl data dump:

```
1 transcribed_unprocessed_pseudogene  gene        11869 14409 . + . gene_id "ENSG00000223972"; gene_name "DDX11L1"; ge
1 processed_transcript                transcript  11869 14409 . + . gene_id "ENSG00000223972"; transcript_id "ENST000(
```

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 .       +       .
```

Perform

- Explore the fastq files in the experiment
  - Use less (or gzip –d –c) to view gz files
  - What are the read lengths?
  - Barcode?
  - Sequencer ID?
- Explore the reference fasta and annotation file, do the two appear concordant?

HOMEWORK 7

Files types

READ PREPROCESSING

Section 6

WHY PREPROCESS READS

- We have found that aggressively ”cleaning” and processing reads can make a large difference to speed and quality of assembly and mapping results. Cleaning your reads means, removing reads/bases

  that are:

  - not of primary interest (contamination)
  - originate from PCR duplication
  - artificially added onto sequence of primary interest (vectors, adapters, primers)
  - low quality bases
  - other unwanted sequence (polyA tails in RNA-seq data)
  - join short overlapping paired-end reads

READ PREPROCESSING STRATEGIES

- Identity and remove contaminant and vector reads
  - Reads which appear to fully come from extraneous sequence should be removed.
- Quality trim/cut
  - "end" trim a read until the average quality > Q (Lucy)
  - remove any read with average quality < Q
- eliminate singletons/duplicates
  - If you have excess depth of coverage, and particularly if you have at least x-fold coverage where x is the read length, then eliminating singletons is a nice way of dramatically reducing the number of error-prone reads.
  - Read which appear the same (particularly paired-end) are often more likely PCR duplicates and therefor redundant reads.
- eliminate all reads (pairs) containing an "N" character
  - If you can afford the loss of coverage, you might throw away all reads containing Ns.
- Identity and trim off adapter and barcodes if present
  - Believe it or not, the software provided by Roche or Illumina, either does not look for, or does a mediocre job of, identifying adapters and removing them.

READ PREPROCESSING PIPELINE

- The pipeline we've developed for read preprocessing employs the following steps for each read:

  Contaminant screening (phiX minimum) → PCR duplicate detection/removal → 5' and 3' end quality trimming, polyA/T removal → Join (when possible) and remove adapters from paired end reads → Final cleanup, remove too short sequence, second polyA/T removal. Finally generate statistic for each sample

- This pipeline is realized using the following applications

  screen.py (part of expHTS), extract_unmapped_reads.py (part of expHTS), Super Deduper, Sickle, Flash, cleanupWrapper.py (part of expHTS)

- screen.py and extract_unmapped_reads.py
  - Remove contaminants (at least PhiX), uses bowtie2 then extracts all reads (pairs) that are marked as unmapped.
- Super-Deduper
  - Remove PCR duplicates (we use bases 10-35 of each paired read)
- Sickle
  - Trim sequences (Left and Right) by quality score (I like Q20)
  - Remove any polyA/T tails (if RNA)
- FLASH2
  - Join and extend, overlapping paired end reads
  - If reads completely overlap they will contain adapter, remove adapters
  - Identify and remove any primer dimers present
- cleanupWrapper.py
  - Remove any reads that are less then the minimum length parameter
  - Run a second polyA/T screen (if RNA)
  - Produce run statistics

QA/QC

- Beyond generating better data for downstream analysis, cleaning statistics also give you an idea as to the quality of the sample, library generation, and sequencing technique used to generate the data.
- This can help inform you of what you might do in the future.
- I've found it best to perform QA/QC on both the run as a whole (poor samples can affect other samples) and on the samples themselves as they compare to other samples (REMEMBER, BE CONSISTANT).
  - Reports such as Basespace for Illumina, are great ways to evaluate the runs as a whole.
  - PCA/MDS plots of the preprocessing summary are a great way to look for technical bias across your experiment

EXPHTS PREPROCESS

- To run preprocessing across all your samples using the raw fastq files stored in 00-RawData folder, use **expHTS preprocess**.
- Parameter considerations (for RNA-seq):
  - Remove polyA/T (-a, --polyA): turn on detection and removal of polyA/T tails.
  - Skip overlapping of PE reads (-O, ----skip-overlap): Currently htseq-count cannot handle a sam/bam file with both single and paired-end reads, so for now we skip overlapping of reads (this also turns 'off' detection of adapters in Illumina data)
  - Force split of PE reads (-S, --forcePairs in mapping): EXPERIMENTAL, to handle the situation of htseq not handling both PE and SE we are playing with the idea of splitting resulting SE reads (in half) to generate pairs.

expHTS is defaulted for 2x100bp paired-end DNA samples

QA/QC

- View the summary report (in a text editor)
- In the workshop git repository is a script
  - expHTS_Processing_QAQC.R
  - Will produce multi-dimensional scaling (MDS) plots of the summary files, the purpose is to look for patterns in the plot that are non-random, and may be influenced by technical
  - Need to provide the script with
    - The input summary file
    - The output file to write the pdf to

Perform

- Launch read preprocessing via
  - expHTS preprocess
- When complete perform QA/QC

HOMEWORK 8

Running preprocessing

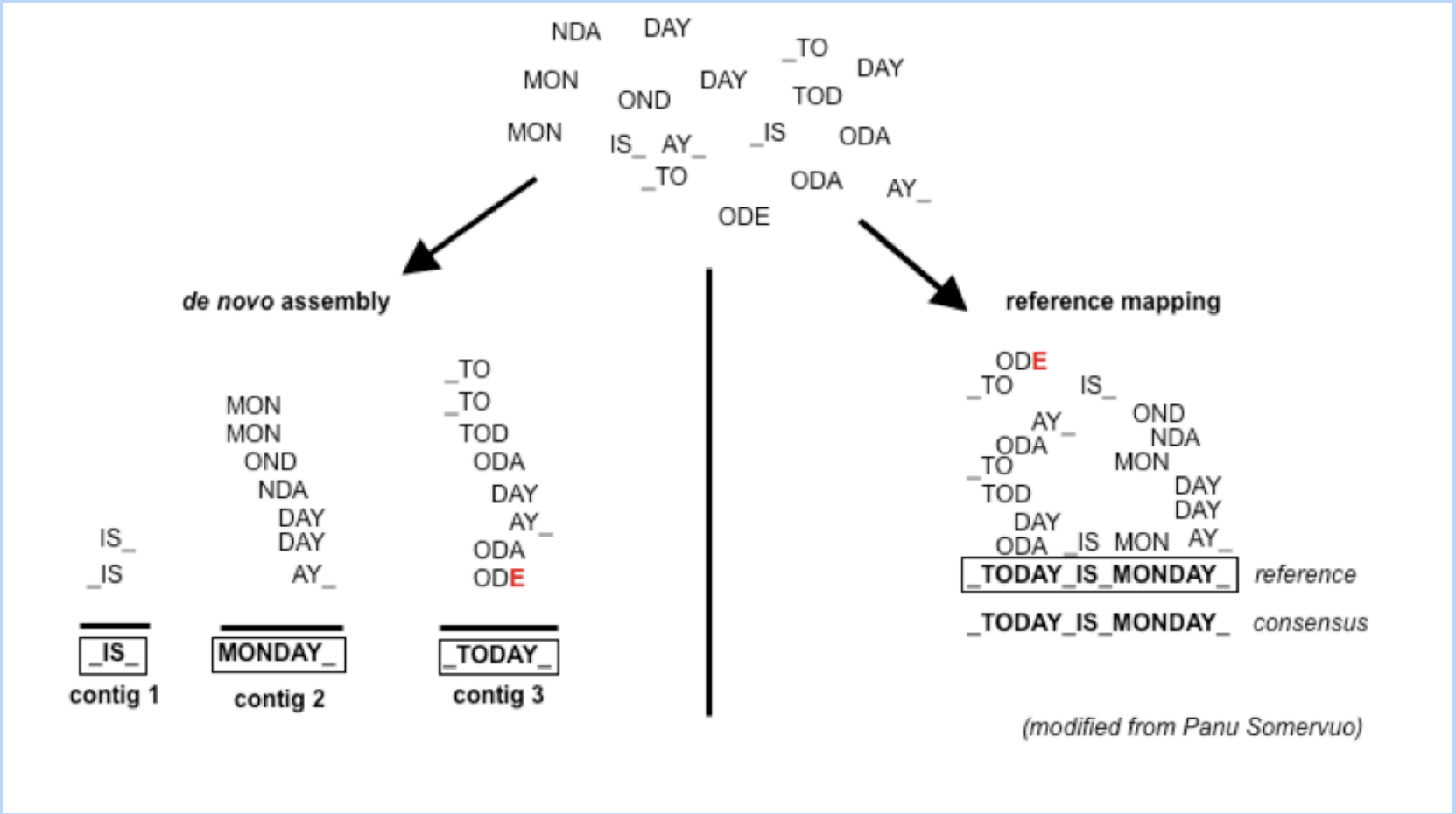
READ MAPPING

Section 7

MAPPING VS ASSEMLY

- Given sequence data,
  - Assembly seeks to put together the puzzle without knowing what the picture is
  - Mapping tries to put together the puzzle pieces directly onto an image of the picture
- In mapping the question is more, given a small chunk of sequence, where in the genome did this piece most likely come from.
- The goal then is to find the match(es) with either the "best" edit distance (smallest), or all matches with edit distance less than max edit dist. Main issues are:
  - Large search space
  - Regions of similarity (aka repeats)
  - Gaps (INDELS)
  - Complexity (RNA, transcripts)

EXAMPLE


NDA
DAY
_TO
DAY
MON
DAY
TOD
OND
MON
IS_
AY_
_IS
ODA
_TO
ODA
AY_
ODE
de novo assembly
reference mapping
_TO
_TO
TOD
ODA
DAY
AY_
ODA
ODE
MON
MON
OND
NDA
DAY
DAY
AY_
IS_
_IS
_IS_
contig 1
MONDAY_
contig 2
TODAY
contig 3
ODE
_TO
IS_
AY_
OND
ODA
NDA
_TO
MON
TOD
DAY
DAY
DAY
ODA
_IS
MON
AY_
_TODAY_IS_MONDAY_
reference
_TODAY_IS_MONDAY_
consensus
(modified from Panu Somervuo)

BLAST

- Some say the first bioinformatics tool, developed at NIH and published in 1990.
  Problem:
- - Exact algorithms like Smith-Waterman and Needleman-Wunsch (dynamic programming) are slow, when the search space becomes large.
- - With the advent of automated DNA sequencing technology, the database of possible matches was becoming increasingly larger.
- the BLAST algorithm emphasizes speed over sensitivity, and does not guarantee an optimal alignment.

BLAST is a few-to-many - performs gapped alignment

BLAST LIKE ALIGNMENT TOOL (BLAT)

- Blat (Jim Kent, UCSC, 2002) was designed to solve the problem of performing comparisons between large genomes and was one of the first algorithms to efficiently search many query sequences against a large database (a genome). Blat also performs a gapped-alignment for searching RNA sequences against a genome and handling splice junctions.
- gapped-alignment alignment allowing for insertions and deletions greater than a few base pairs. Gapped alignment are
- less efficient, but more accurate.

BLAT is a many-to-many algorithm - performs gapped alignments

HIGH THROUGHPUT MAPPING

- Many additional algorithms have been developed since BLAST and BLAT, mainly improving on either speed or accuracy, or both.

and then came Illumina data

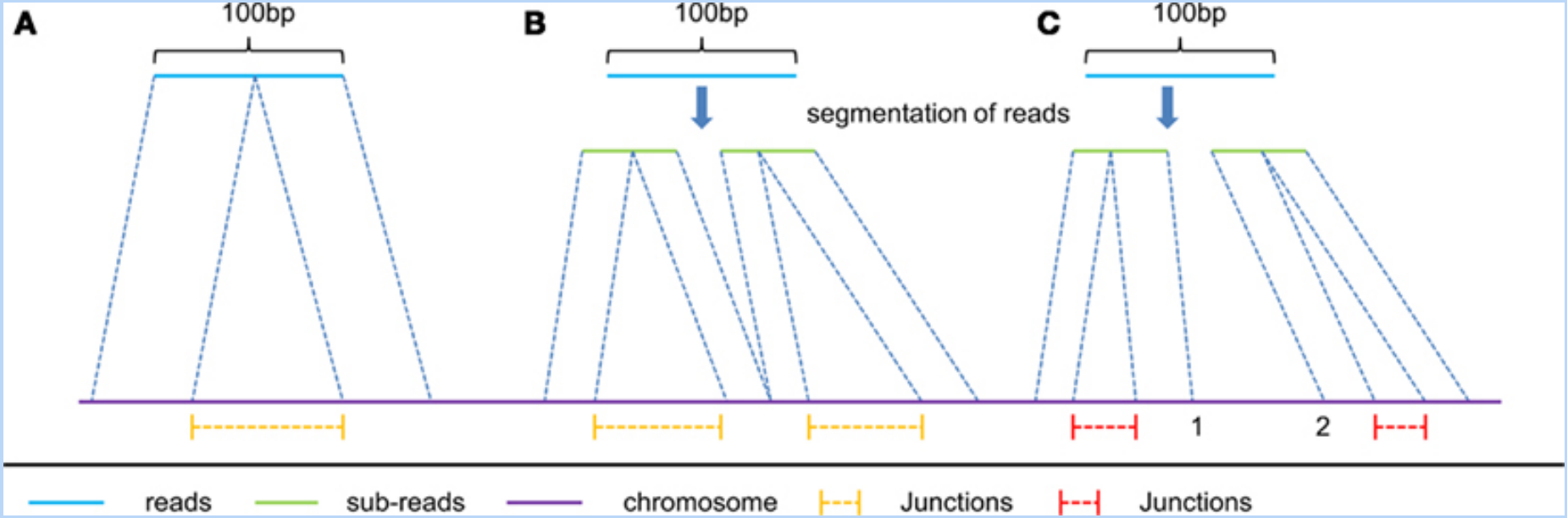
- New Problem:
  - We still have a large search space (aka genome)
  - Small fragments with with to map with possibly many possible close matches
  - Millions or Billions of query sequences

CONSIDERATION

- Placing reads in regions that do not exist in the reference genome (reads extend off the end) [mitochondrial, plasmids, structural variants, etc.].
- Sequencing errors and variations: alignment between read and true source in genome may have more differences than alignment with some other copy of repeat.
- What if the closest fully sequenced genome is too divergent? (3% is a common alignment capability)
- Placing reads in repetitive regions: Some algorithms only return 1 mapping; If multiple: map quality = 0
- Algorithms that use paired-end information => might prefer correct distance over correct alignment.

INTRON/EXON JUNCTIONS

- In RNA-seq data, you must also consider splice junctions, reads may span an intron



SOME ALIGNERS

- Spliced Aligners
  - Tophat (Bowtie2)
  - GSNAP
  - SOAPsplice
  - MapSplice
  - TrueSite
  - rna-star
- Aligners that can 'clip'
  - Bowtie2 in local
  - bwa-mem
- http://en.wikipedia.org/wiki/List_of_sequence_alignment_software

EXPHTS MAPPING

- To run mapping across all your samples using the cleaned fastq files created with expHTS preprocess, use **expHTS mapping**.
- Parameter considerations (for RNA-seq):
  - Sort by Read ID (-n, --sortByReadID): turns on bam resorting by read id instead of position, for compatibility with htseq-count
  - Ignore singles (-s, --ignoreSingles): Ignore all SE read files present in the cleaned folder, for compatibility with htseq-count. If you turned off overlapping of reads there shouldn't be, but this parameters makes 'sure' they aren't included.
  - Force split of PE reads (-S, --forcePairs): EXPERIMENTAL, to handle the situation of htseq not handling both PE and SE we are playing with the idea of splitting resulting SE reads (in half) to generate pairs.

QA/QC

- View the summary report (in a text editor)
- In the workshop git repository is a script
  - expHTS_Processing_QAQC.R
  - Will produce multi-dimensional scaling (MDS) plots of the summary files, the purpose is to look for patterns in the plot that are non-random, and may be influenced by technical
  - Need to provide the script with
    - The input summary file
    - The output file to write the pdf to

Perform

- Launch read mapping via
  - expHTS mapping
- When complete perform QA/QC

HOMEWORK 9

Mapping

ESTIMATE KNOWN GENES AND TRANSCRIPTS EXPRESSION – COUNTING

Section 8

COUNTING AS A MEASURE OF EXPRESSION

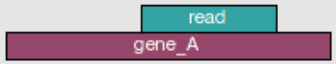
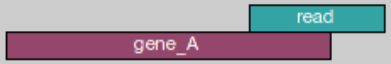
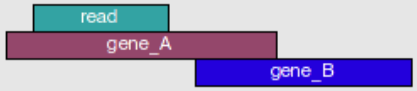
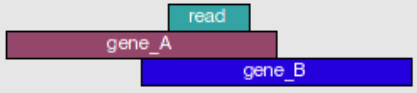
- The more you can count (and HTS sequencing systems can count a lot) the better the measure of copy number for even rare transcripts in a population.
  - Most RNA-seq techniques deal with count data. Reads are mapped to a reference genome, transcripts are detected, and the number of reads that map to a transcript (or gene) are counted.
  - Read counts for a transcript are roughly proportional to the gene's length and transcript abundance.
- technical artifacts should be considered during counting
  - mapping quality
  - mappability (uniqueness), the read is not ambiguous

READ COUNTING WITH HTSEQ

Problem:

- Given a sam/bam file with aligned sequence reads and a list of genomic feature (genes locations), we wish to count the number of reads (fragments) than overlap each feature.
  - Features are defined by intervals, they have a start and stop position on a chromosome.
  - For this workshop and analysis, features are genes which are the union of all its exons. You could consider each exon as a feature, for alternative splicing.
- Htseq-count has three overlapping modes
  - union:
  - intersection-strict
  - intersection-nonempty

HTSEQ-COUNT

	union	intersection _strict	intersection _nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous	gene_A	gene_A
	ambiguous	ambiguous	ambiguous

EXPHTS HTSEQ

- To run htseq-count across all your samples using bam files created with expHTS mapping, use **expHTS htseq**.
- Parameter considerations:
  - Stranded/unstranded (-s, --stranded): 'yes' (stranded, same strand as reference), 'reverse' (stranded, reverse strand as reference), 'no' (unstranded libraries).
  - Minimum alignment quality (-a, --minaqual): skip all reads with lower than this minimum mapping quality value, default is 10.
  - Feature type (-y, --type): The feature type to use as reference (3rd column of the gtf/gff file, defines the interval to use, default is 'exon'
  - Feature id (-i, --idattr): The feature id to use (union of feature type), defines the final count 'groups', default is 'gene_id'.
  - Overlapping mode (-m, --mode): The mode to use to define overlaps, 'union', 'intersection-strict', 'intersection-nonempty', default is 'union'.

QA/QC

- View the summary report (in a text editor)
- In the workshop git repository is a script
  - expHTS_Processing_QAQC.R
  - Will produce multi-dimensional scaling (MDS) plots of the summary files, the purpose is to look for patterns in the plot that are non-random, and may be influenced by technical
  - Need to provide the script with
    - The input summary file
    - The output file to write the pdf to

Perform

- Launch read mapping via
  - expHTS htseq –s reverse
- When complete perform QA/QC

- View the top (head) and bottom (tail) portion of the counts file, see what the the data looks like.

HOMEWORK 10

Read Counting

DIFFERENTIAL EXPRESSION ANALYSIS USING EDGER

Section 9

DIFFERENTIAL EXPRESSION ANALYSIS

- Differential Expression between conditions is determined from count data, which is modeled by a distribution (ie. Negative Binomial Distribution, Poisson, etc.)
- Generally speaking differential expression analysis is performed in a very similar manner to DNA microarrays, once and normalization have been performed.
- A lot of RNA-seq analysis has been done in R and so there are many packages available to analyze and view this data. Two of the best are:
  - DESeq, developed by Simon Anders (also created htseq) in Wolfgang Huber's group at EMBL
  - edgeR (extension to Limma [microarrays] for RNA-seq), developed out of Gordon Smyth's group from the Walter and Eliza Hall Institute of Medical Research in Australia
  - http://bioconductor.org/packages/release/BiocViews.html#___RNASeq

NORMALIZATION

- In differential expression analysis, only sample-specific effects need to be normalized, NOT concerned with comparisons and quantification of absolute expression.
  - Sequence depth – is a sample specific effect and needs to be adjusted for.
  - RNA composition - finding a set of scaling factors for the library sizes that minimize the log-fold changes between the samples for most genes (uses a trimmed mean of M-values between each pair of sample)
  - GC content – is NOT sample-specific (except when it is)
  - Gene Length – is NOT sample-specific (except when it is)

- Normalization in edgeR is model-based

RPKM VS FPKM VS MODEL BASED

- RPKM - Reads per kilobase per million mapped reads
- FPKM - Fragments per kilobase per million mapped reads
- Model based - original read counts are not themselves transformed, but rather correction factors are used in the DE model itself.
  - Produces CPM – Counts per million

BASIC STEPS PROCEDURE - EDGER

1. Read the count data in
2. Remove (uninteresting genes, e.g. unexpressed)
3. Calculate normalizing factors (sample-specific adjustment)
4. Calculate dispersion (gene-gene variance-stabilizing transformation.)
5. Fit a model of your experiment
6. Perform likelihood ratio tests on comparisons of interest (using contrasts)
7. Adjust for multiple testing, Benjamini-Hochberg (BH) is the defaults.
8. Check results for confidence
9. Attach annotation if available and write tables

Perform

- Using the R-script, Analysis_EdgeR_RNAseq.R on the workshop repository, perform Differential expression analysis on the dataset.

HOMEWORK 11

Differential Expression Analysis

SUMMARIZATION AND VISUALIZATION

Section 10

THE TOP TABLE

- The basic table
  - Gene_ID: The Gene Id from the GTF file
  - logFC: log fold change, positive values indicate up-regulation, negative numbers indicate down-regulation
  - logCPM: log counts per million, average 'expression' value of the gene
  - LR: log ratio of the test (ignore)
  - Pvalue: raw p-value for that gene (best to sort on)
  - FDR: false discover rate for that gene
- Annotation is added in additional columns (must first uncomment the line to do so in the R script

VISUALIZATION AND NEXT STEP TOOLS

Visualization

1. Integrated Genome Viewer (https://www.broadinstitute.org/igv/)

Further Annotation of Genes

1. DAVID (http://david.abcc.ncifcrf.gov/tools.jsp)
2. ConsensusPathdb (http://cpdb.molgen.mpg.de/)
3. NetGestalt (http://www.netgestalt.org/)
4. Molecular Signatures Database (http://www.netgestalt.org/)
5. PANTHER (http://www.pantherdb.org/)
6. Cognoscente (http://vanburenlab.medicine.tamhsc.edu/cognoscente.shtml)
7. Pathway Commons (http://www.pathwaycommons.org/)
8. Readctome (http://www.reactome.org/)
9. PathVisio (http://www.pathvisio.org/)
10. Moksiskaan (http://csbi.ltdk.helsinki.fi/moksiskaan/)
11. Weighed Gene Co-Expression Network Analysis (WGCNA)s
12. More tools in R Bioconductor

GENE SET ENRICHMENT ANALYSIS (GSEA) AND GO/PATHWAY ENRICHMENT

Gene set enrichment analysis

- A computational method that determines whether an a priori set of genes (e.g. gene ontology group, or pathway) shows statistically significant, concordant differences between two biological states (e.g. phenotypes)

Gene Ontology/Pathways enrichment analysis

- Given a set of genes that are up-regulated, which gene ontologies or pathways are over-represented (or under-represented) using annotations for that gene set.

Perform

- View the output table
- Compare our results to those of the paper (PMC4059230)

HOMEWORK 12

Summarization