

So you want to do a RNAseq Experiment Differential Expression Analysis

Last updated: Feb 2nd, 2017

Matt Settles

Director, Bioinformatics Core

Experimental Design

Differential Expression Analysis

Treating Bioinformatics as a Data Science

Seven stages to data science

1. Define the question of interest
2. Get the data
3. Clean the data
4. Explore the data
5. Fit statistical models
6. Communicate the results
7. Make your analysis reproducible

**Data science done well looks easy and
that's a big problem for data scientists**

**simplystatistics.org
March 3, 2015 by Jeff Leek**

What is Differential Expression

Differential expression analysis means taking the *normalised* sequencing fragment count data and performing statistical analysis to discover *quantitative* changes in expression levels between experimental groups.

For example, we use statistical testing to decide whether, for a given gene, an observed difference in fragment counts is significant, that is, whether it is greater than what would be expected just due to natural random variation.

Designing Experiments

Beginning with the question of interest (and working backwards)

- The final step of a DE analysis is the application of a linear model to each gene in your dataset.

Traditional statistical considerations and basic principals of statistical design of experiments apply.

- **Control** for effects of outside variables, avoid/consider possible biases, avoid confounding variables in sample preparation.
 - **Randomization** of samples, plots, etc.
 - **Replication** is essential (triplicates are THE minimum)
- You should know your final (DE) model and comparison contrasts before beginning your experiment.

Three outcomes

Goldilocks and the three bears

- Technical and/or biological variation exceeds that of experimental variation, results in 0 differentially expressed genes
- Experiment induces a significant phenotype with cascading effects and/or little to no biological variation between replicates (ala cell lines), results in 1000s of DE genes. Some of which are directly due to experiment; however, most due to cascading effects.
- Technical artifacts are controlled. Biological variation is induced in the experiment, and cascading effects are controlled, or accounted for, results in 100s of DE genes directly applicable to the question of interest.

General rules for preparing samples

- Prepare more samples than 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.

Sample preparation

In high throughput biological work (Microarrays, Sequencing, HT Genotyping, etc.), what may seem like small technical details 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 (seen on a global scale) and may cause significant issues during analysis.

Be Consistent

BE CONSISTENT ACROSS ALL SAMPLES!!!

Generating RNA-seq libraries

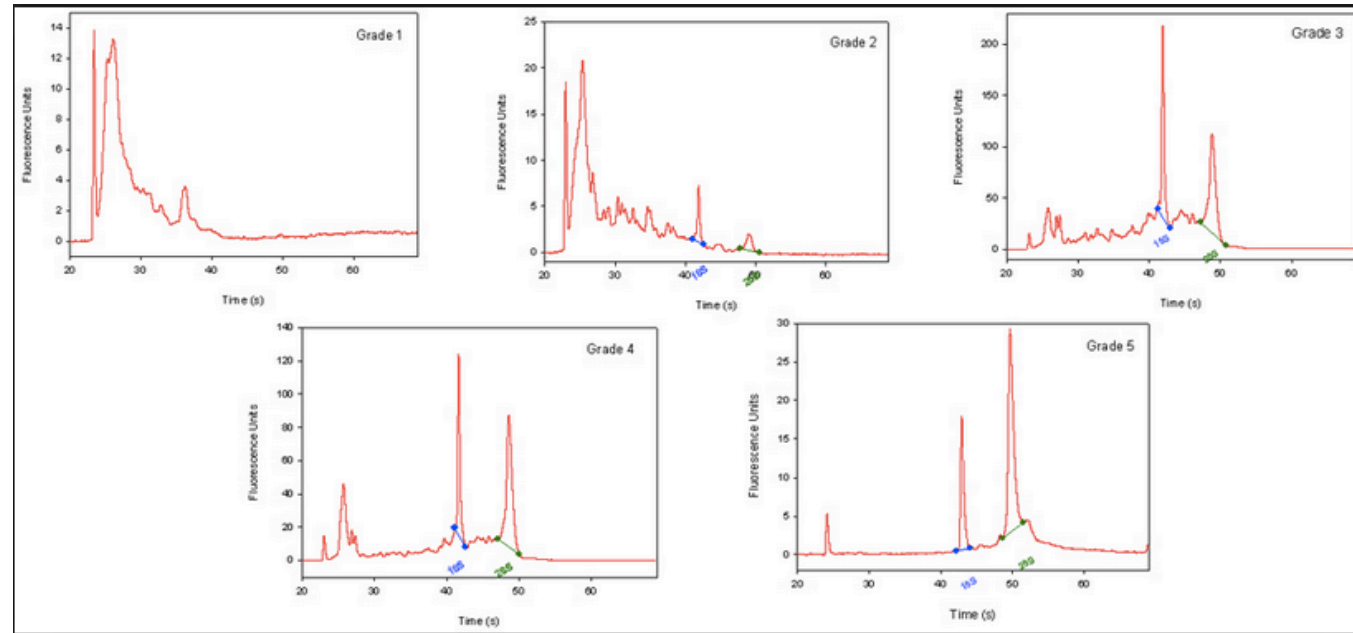
Considerations

- QA/QC of RNA samples
- What is the 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 Agilent 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 sample for “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!!!

[SUMMARY] Generating RNA-seq libraries

Considerations

- QA/QC of RNA samples [Consistency across samples is most important.]
- What is the RNA of interest [polyA extraction is recommended.]
- Library Preparation
 - Stranded Vs. Unstranded [Standard stranded library kits]
- Size Selection/Cleanup [Target mean 150bp or kit recommendation]
 - Final QA [Consistency across samples is most important.]

Sequencing Depth

- Coverage is determined differently for "Counting" based experiments (RNAseq, amplicons, etc.) where an expected number of reads per sample is typically more suitable.
- The first and most basic question is how many reads per sample will I get
Factors to consider are (per lane):
 1. Number of reads being sequenced
 2. Number of samples being sequenced
 3. Expected percentage of usable data

$$\frac{\text{reads}}{\text{sample}} = \frac{\text{reads.sequenced} * 0.8}{\text{samples.pooled}}$$

- Read length, or SE vs PE, does not factor into sequencing depth.

Sequencing

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 best)
- 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/experiment is determined by the goals of the experiment and the nature of the RNA sample.

Barcodes and Pooling samples for sequencing

- Best to have as many barcodes as there are samples
 - Can purchase barcodes from vendor, generate them yourself from IDTdna (example), or consult with the DNA technologies core.
- Best to pool all samples into one large pool, then sequence multiple lanes
- IF you cannot generate enough barcodes, or pool into one large pool, RANDOMIZE samples into pools.
 - Bioinformatics core can produce a randomization scheme for you.
 - This must be consider/determined PRIOR to library preparation

Cost Estimation

- RNA extraction and QA/QC (Per sample)
- Enrichment of RNA of interest + library preparation (Per sample)
 - Library QA/QC (Bioanalyzer and Qubit)
 - Pooling (\$10/library) [If you do your own libraries]
- 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).

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

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 $\$2346 \times 2 = \4692
- Bioinformatics, simple pairwise comparison design, DE only \$2000
 - This is the most basic analysis, for in depth collaborative analysis double sequencing budget.

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

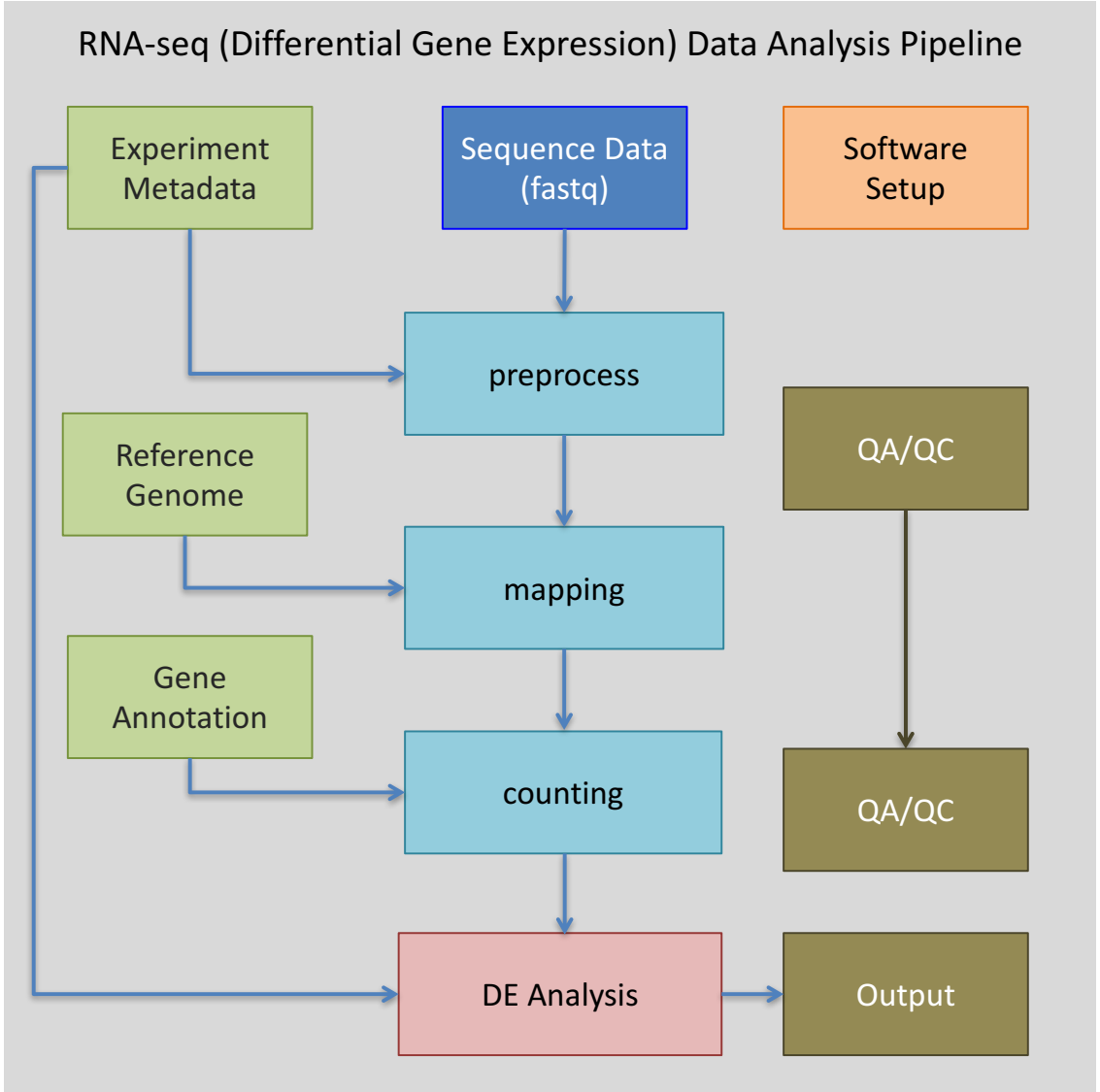
Approximately \$950 per sample @ 40M reads per sample

Overview of RNA-SEQ data analysis

Prerequisites

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

RNA-seq pipeline overview



Sequence Preprocessing

Why Preprocess reads

- We have found that aggressively “cleaning” and processing reads can make a large difference to the **speed** and **quality** of assembly and mapping results. Cleaning your reads means, removing reads/bases that are:
 - other unwanted sequence (polyA tails in RNA-seq data)
 - artificially added onto sequence of primary interest (vectors, adapters, primers)
 - join short overlapping paired-end reads
 - low quality bases
 - originate from PCR duplication
 - not of primary interest (contamination)
- Preprocessing also produces a number of statistics that are technical in nature that should be used to evaluate “experimental consistancy”

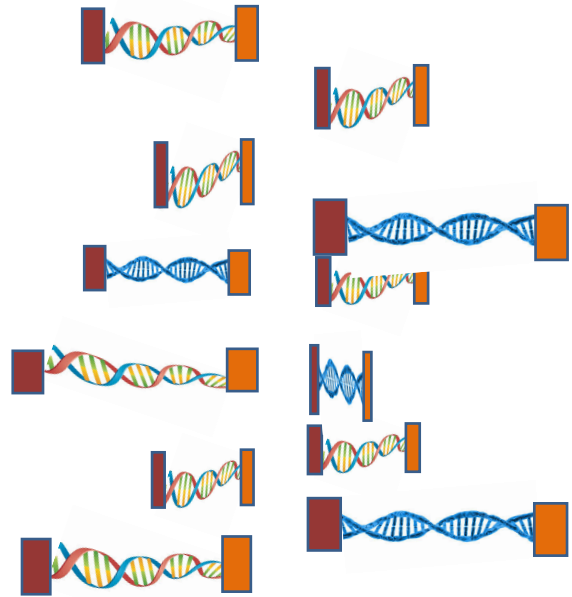
Read Preprocessing strategies, many over time

- 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 Illumina, either does not look for, or does a mediocre job of, identifying adapters and removing them.

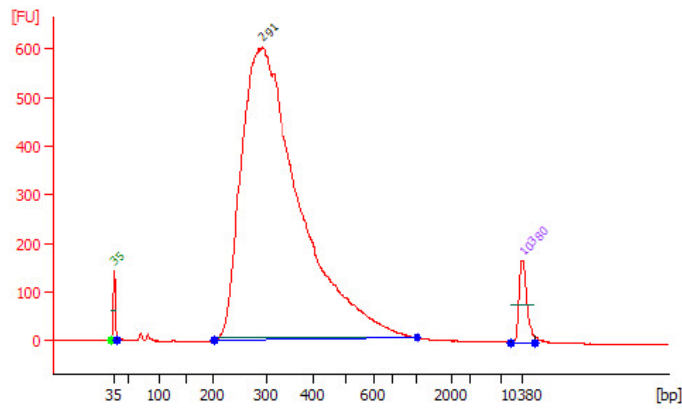
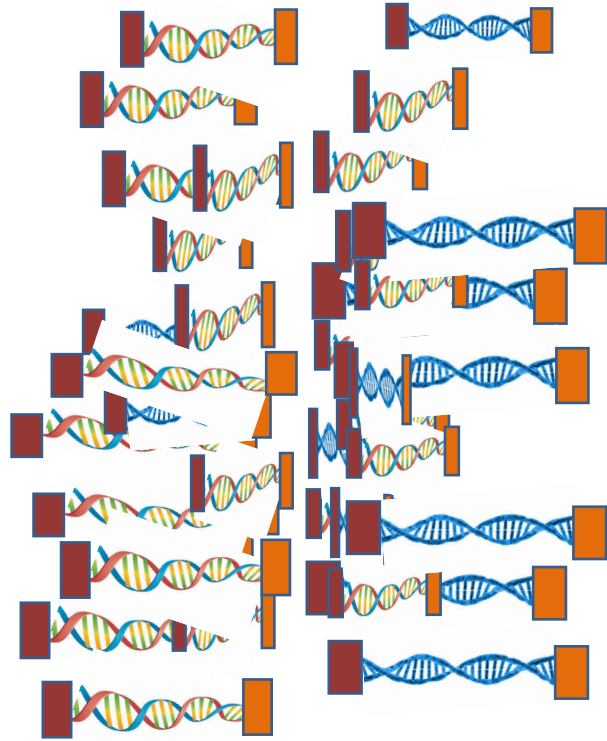
DNA/RNA, could contain 'contamination'



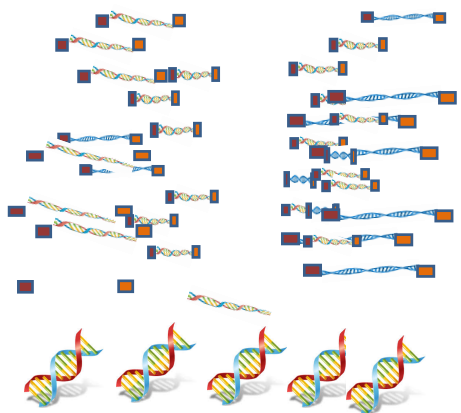
Library prep, fragmentation, adapter addition



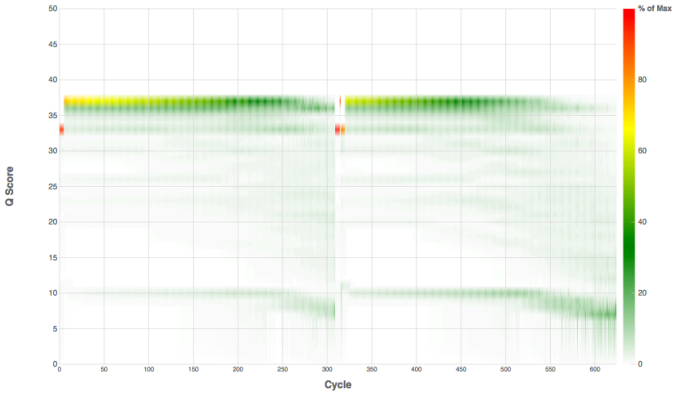
PCR enrichment



Final Library, size distribution



Possible addition of phiX



Sequencing Characteristics/ Quality

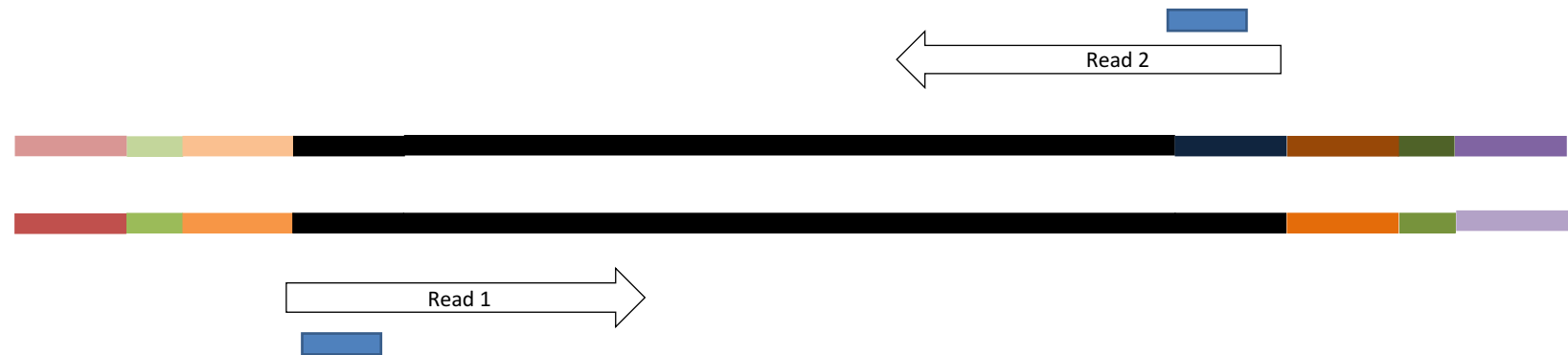
Preprocessing

- Map reads to contaminants/PhiX and extract unmapped reads [bowtie2 --local
 - Remove contaminants (at least PhiX), uses bowtie2 then extracts all reads (pairs) that are marked as unmapped.
- Super-Deduper [PE reads only]
 - Remove PCR duplicates (we use bases 10-35 of each paired read)
- FLASH2 [PE reads only]
 - Join and extend, overlapping paired end reads
 - If reads completely overlap they will contain adapter, remove adapters
 - Identify and remove any adapter dimers present
- Scythe [SE Reads only]
 - Identify and remove adapter sequence
- Sickle
 - Trim sequences (5' and 3') by quality score (I like Q20)
- cleanup
 - Run a polyA/T trimmer
 - Remove any reads that are less than the minimum length parameter
 - Produce preprocessing statistics

Why Screen for PhiX

- PhiX is a common control in Illumina runs, facilities rarely tell you if/when PhiX has been spiked in
 - Does not have a barcode, so in theory should not be in your data
- **However**
 - When I know PhiX has been spiked in, I find sequence (many X coverage) every time
 - When I know PhiX has not been spiked in, I **do not** find sequence
- Better safe than sorry and screen for it.

Super Deduper



Data	Alignment Algorithm	MarkDuplicates	Rmdup	Super Deduper	FastUniq	Fulcrum	Total # of Reads
PhiX	BWA MEM	1,048,278 (0.25%)	1,011,145 (1.05%)	1,156,700 (13.7%)	4,202,526	3,092,155	4,750,299
	Bowtie 2 Local	1,054,725 (6.62%)	948,784 (10.2%)	1,166,936 (14.0%)	4,236,647	3,103,872	4,790,972
	Bowtie 2 Global	799,524 (0%)	800,868 (0.12%)	896,487 (9.92%)	3,768,641	2,704,114	4,293,787
Acropora digitifera	BWA MEM	5,132,111 (2.26%)	6,906,634 (44.5%)	5,133,339 (10.2%)	12,968,469	2,103,567	54,108,240
	Bowtie 2 Local	4,688,809 (4.03%)	5,931,862 (38.9%)	3,971,743 (9.32%)	9,893,903	4,259,619	41,728,154
	Bowtie 2 Global	1,457,865 (3.62%)	1,512,966 (24.2%)	1,185,838 (11.4%)	3,014,498	1,286,031	11,600,847

Super Deduper

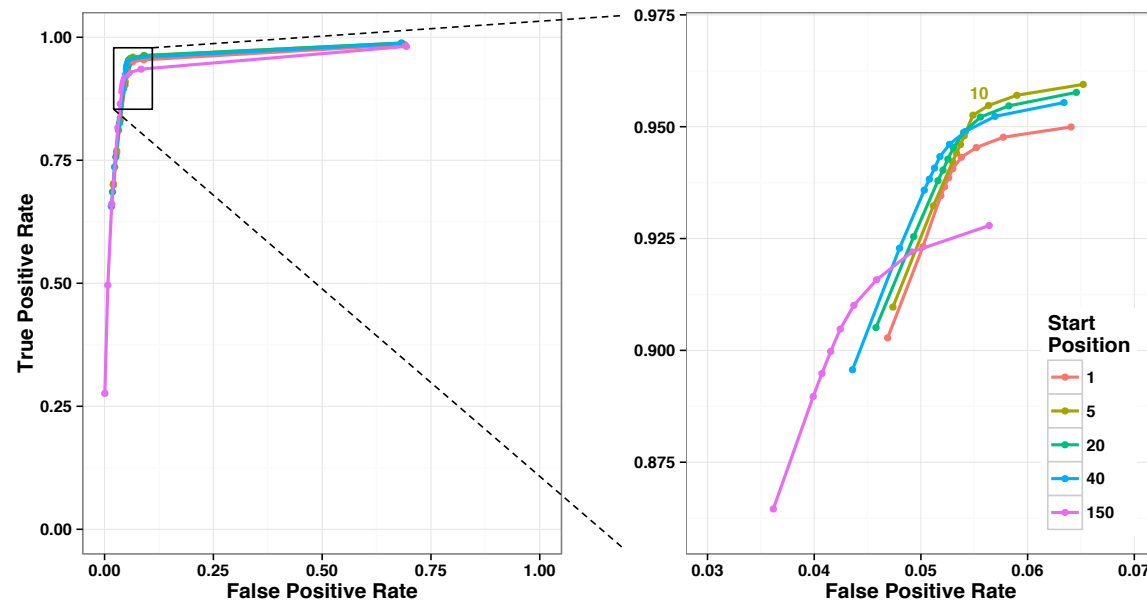
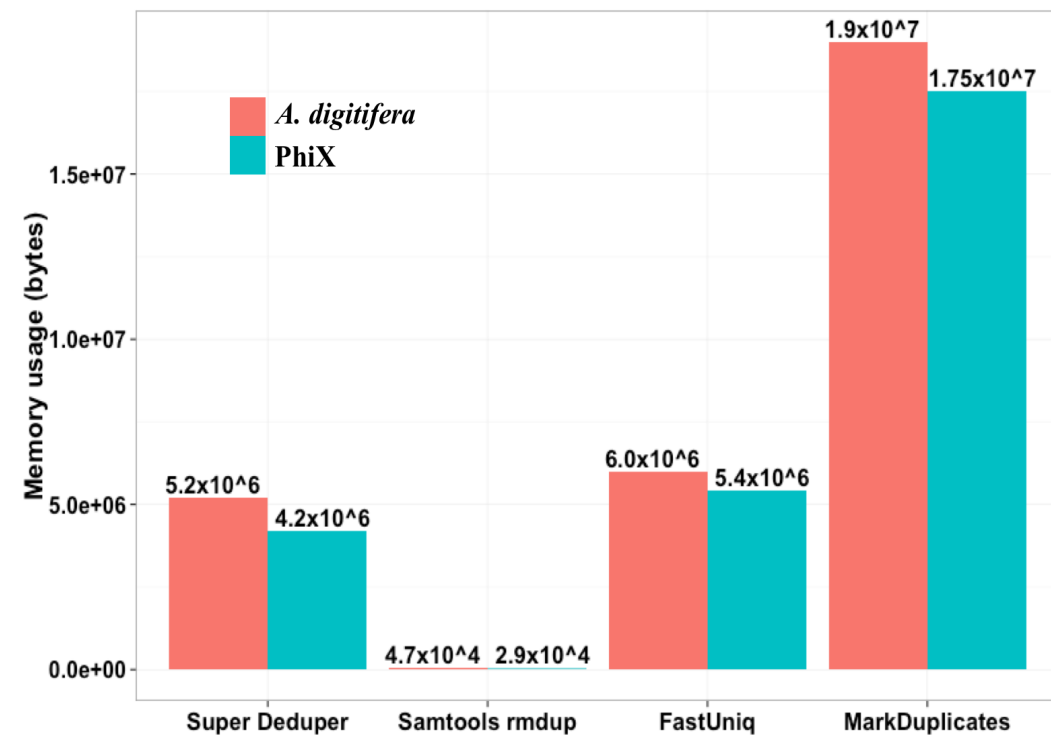
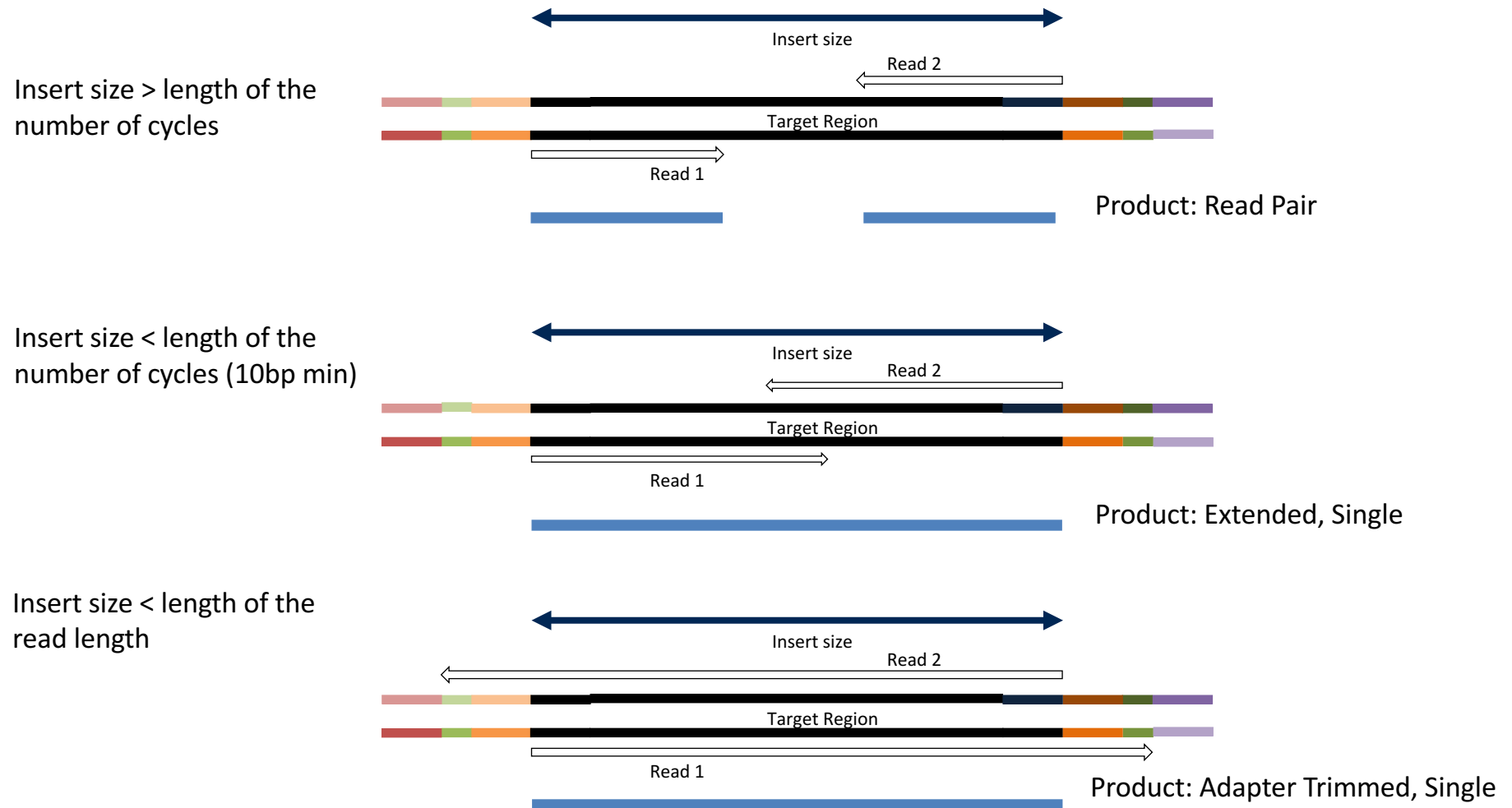


Figure 1: ROC curves. Only a representative subset of the different start positions is shown. The image on the left shows the full ROC curves and the image on the right is a zoomed in view of corner of the curves. Each curve represents a start position and each point represents a length. The labeled point in the image on the right is the default start and length for Super Deduper.



We calculated the Youden Index for every combination tested and the point that acquired the highest index value (as compared to Picard MarkDuplicates) occurred at a start position of 5bp and a length of 10bps (20bp total over both reads)

Flash2 – overlapping of reads and adapter removal in paired end reads



QA/QC

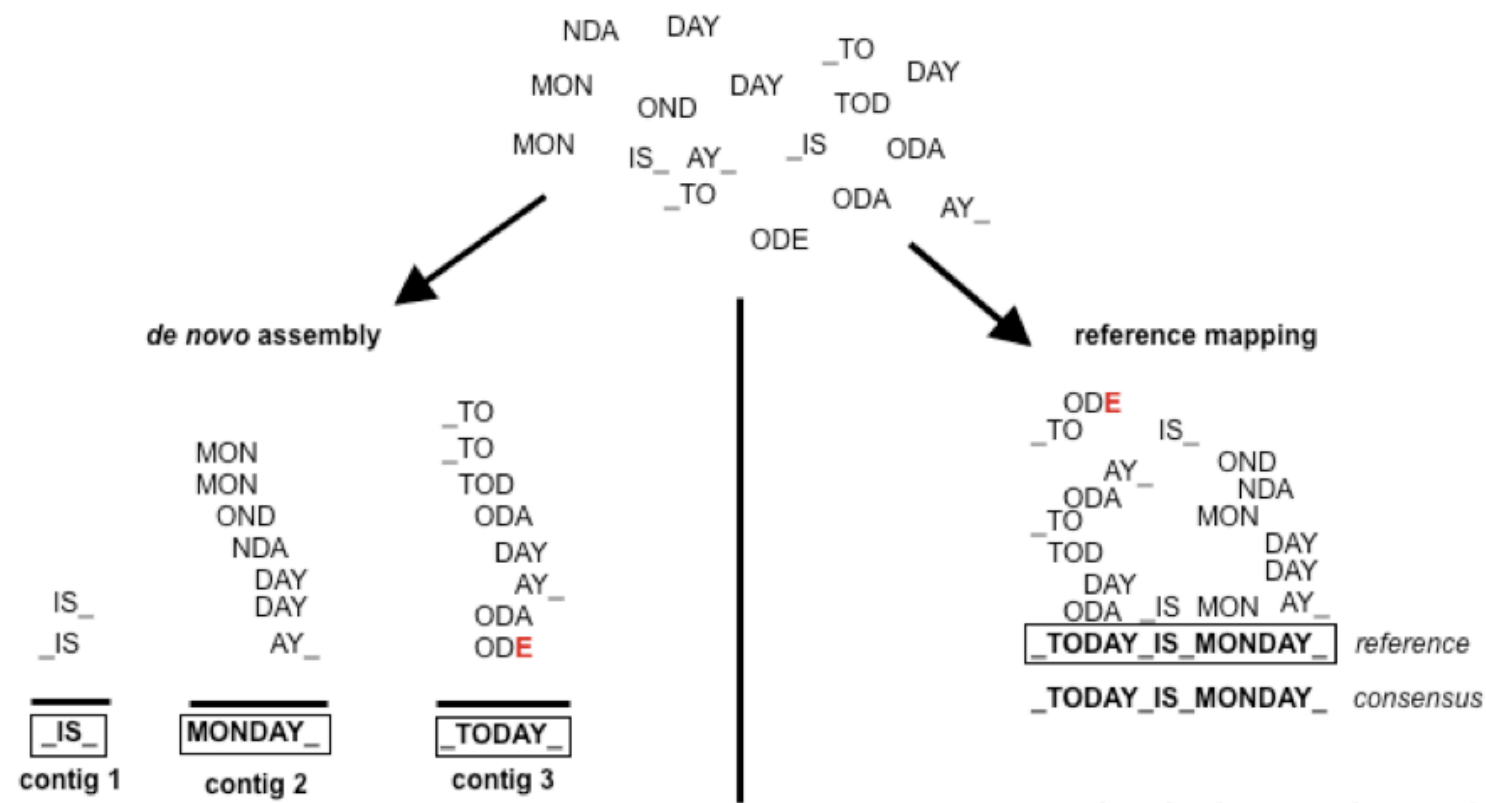
- Beyond generating better data for downstream analysis, cleaning statistics also give you an idea as to the quality of the sample, library generation, and sequencing quality used to generate the data.
- This can help inform you of what you might do in the future.
- I've found it best to perform QA/QC 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

Sequence Mapping

Mapping vs Assembly

- 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



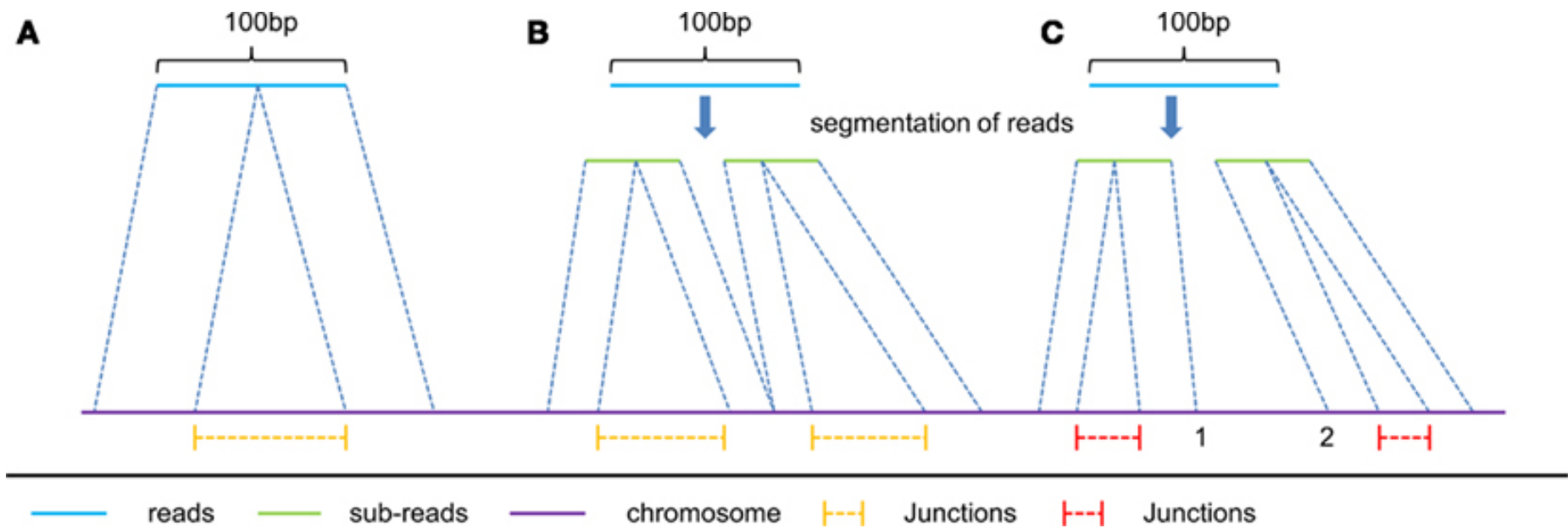
(modified from Panu Somervuo)

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 assumed 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
 - star
- Aligners that can 'clip'
 - Bowtie2 in local mode
 - bwa-mem

https://en.wikipedia.org/wiki/List_of_sequence_alignment_software

Genome vs Transcriptome Reference

- May seem intuitive to map RNAseq data to transcriptome, but its not that simple.
 - Transcriptomes are rarely complete,
 - Which transcript, canonical transcript? Shouldn't map to all splice variants as these would show up as multi-mappers
- More so, a mapper will do its damndest to map every read, somewhere, provided the result meets its minimum requirements.
 - Need to provide a mapper with all possible places the read could arisen from, which is best represented by the genome. Otherwise you get mismapping because its close enough.

Genome and Annotation

- Genome fasta files and Annotation files go together! Should be identified before beginning any analysis
 - Genome fasta files should include all primary chromosomes, unplaced sequences and un-localized sequences, as well as any organelles. Should not contain any contigs that represent patches or alternative haplotypes.
 - Annotation file should be GTF (preferred), and should be the most comprehensive you can find.
 - Chromosome names in the GTF should match those in the fasta, they don't always do.
 - Star recommends the Gencode annotations for mouse/human

Preparing a sam file for counting and stats

- Samtools is used to manipulate mapping files for counting, common steps include:
 - samtools view [to convert from sam to bam]
 - samtools sort [possibly by read and not by position, htseq-count requirement]
 - samtools index
 - samtools idxstats
 - samtools flagstat
 - samtools stats
- Check with the counting application as to its input requirements.

QA/QC

- Mapper produce summary statistics, view the summary report (in a text editor) and compare across samples.
 - Other additional summary statistics can be produced with:
samtools flagstat
samtools idxstats
samtools stats
- Produce a 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 artifacts

Estimate known genes and transcripts expression –
Counting

Counting as a measure of expression

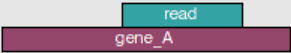
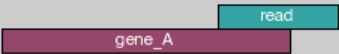


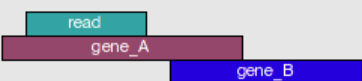
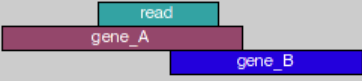
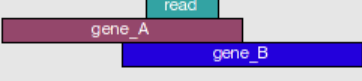
- The more you can count (and HTS sequencing systems can count a lot) the better the measure of copy number for even rare transcripts in a population.
 - Most RNA-seq techniques deal with count data. Reads are mapped to a reference genome, transcripts are detected, and the number of reads that map to 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
 - mapability (uniqueness), the read is not ambiguous

Read Counting with HTSEQ-COUNT

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) that 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

Counting genes -- STAR

- Counts coincide with Htseq-counts under default parameters. Need to specify GTF file at genome generation step or during mapping.
- Output, 4 columns
 - GeneID
 - Counts for unstranded
 - Counts for first read strand
 - Counts for the second read strand
- Chose the columns that makes sense and generate a matrix table, columns are sample, rows are genes.

QA/QC

- View summary report (in a text editor)
- Produce a 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 means.
- Statistics such as:
 - % Multimapped reads
 - % Uniquely mapped reads
 - Splice sites
 - Unmapped
 - Chimeric
 - Etc.

Differential Expression Analysis using edgeR/Limma Voom

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/Voom (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

Basic steps procedure – edger/limma voom

1. Read the count data in
2. Filter genes(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

Filtering genes

- Most common filter is to **remove** genes that are less than X reads counts across a certain number of samples. EX.
 - `rowSums(cpms <= 1) < 3` , require at least 1 cpm in at least 3 samples to keep
- A second less used filter to is minimum variance across all samples, so if a gene isn't changing (constant expression) its not interesting no need to test.

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/Voom is model-based, you calculate normalizing factors using the function `calcNormFactors` function which by default uses TMM (trimmed means of M values). **Assumes most genes are not DE.**

RPKM vs FPKM vs CPM vs model based

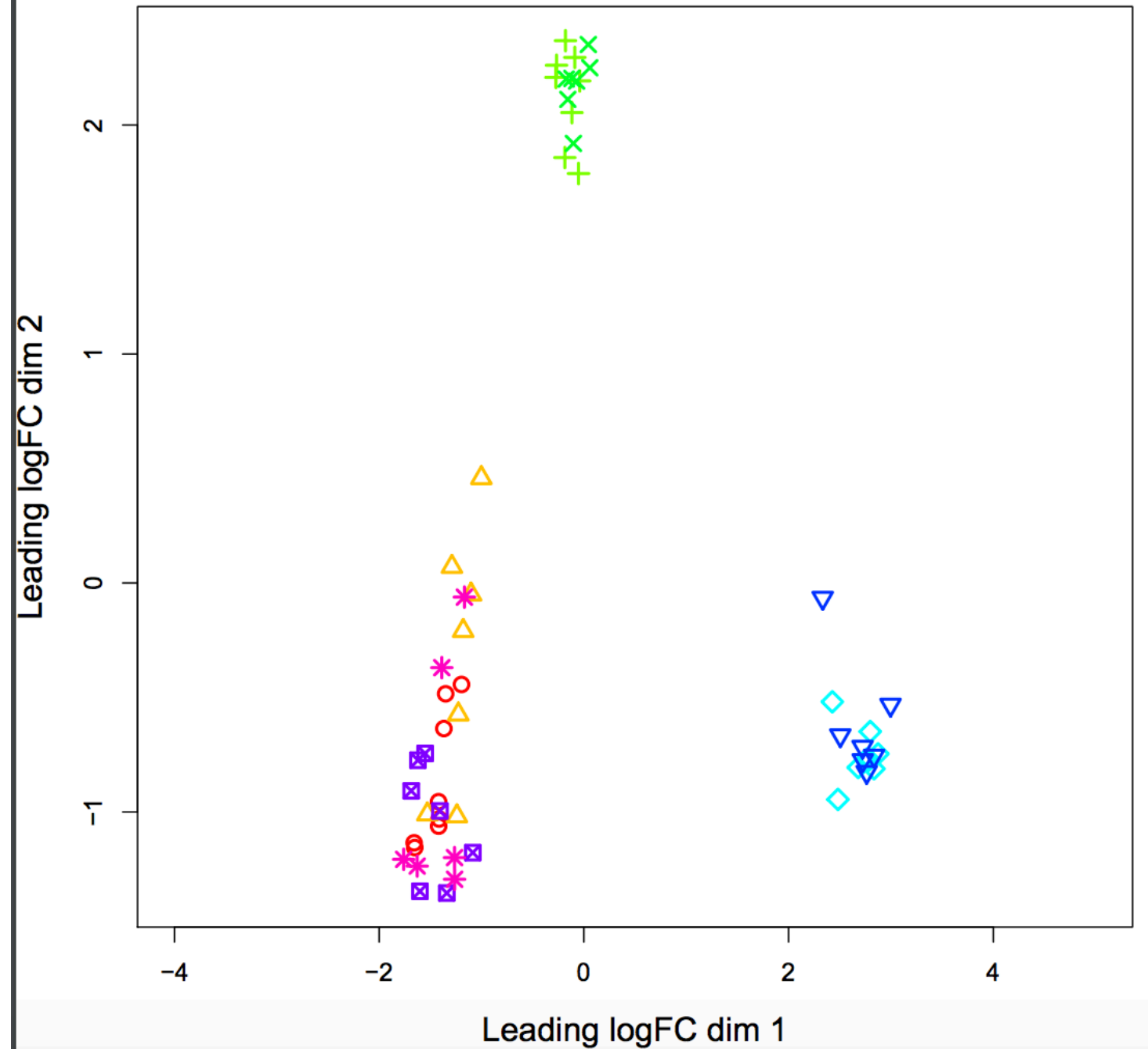
- RPKM - Reads per kilobase per million mapped reads
- FPKM - Fragments per kilobase per million mapped reads
- logCPM – log Counts per million [good for producing MDS plots, estimation of normalized values in model based]
- Model based - original read counts are not themselves transformed, but rather correction factors are used in the DE model itself.

Transformation

- Transformation turn the gene count data into a distribution more suitable for statistical analysis (more "normal" like).
- Most common
 - Limma-trended transformation, logCPM, best when total counts per sample are relatively close to each other (~3-fold)
 - Voom, is best used when library sizes are quite variable across samples

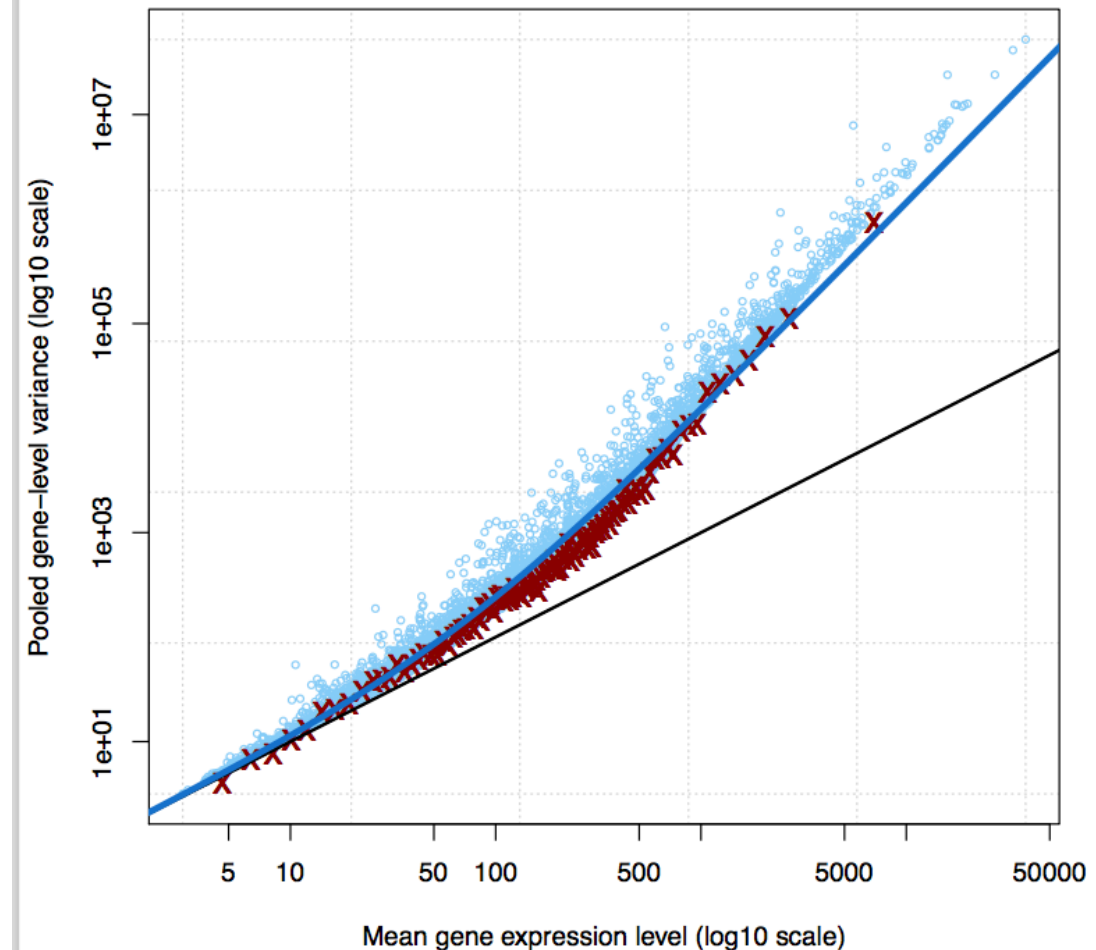
QA/QC

- MDS plots of logCPM



Variance Stabilization - eBayes

- The variance characteristics of low expressed genes are different from high expressed genes, if treated the same, the effect is to over represent low expressed genes in the DE list.



Multiple testing correct

- Simply a must! Best choices are
 - FDR (false discovery rate)
 - qvalue
- The FDR (or qvalue) is a statement about the list and no longer about the gene. So a FDR 0.05, says you expect 5% false positives in the list of genes with an FDR of 0.05 and less.
- The statement “Statistically significant” **means** FDR of 0.05 or less.
 - My opinion is these genes do not require further validation (eg qrtPCR)
 - You can dip below FDR 0.05, but in my opinion you then need to validate those genes.

EdgeR/Limma Manual

- Both edgeR and limma voom have VERY comprehensive user manuals
 - Limma voom
<https://bioconductor.org/packages/release/bioc/vignettes/limma/inst/doc/usersguide.pdf>
 - edgeR
<http://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf>

Summarization and Visualization

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.

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:

- 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:

- 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/>

OR simultaneous read mapping and counting:

- Star
 - <https://github.com/alexdobin/STAR> [performs both alignment and counting]

Software

Analysis of differential expression:

- R <http://www.r-project.org/>
 - R Packages: EdgeR, limma from bioconductor – differential expression analysis
 - <http://bioconductor.org/packages/release/bioc/html/edgeR.html>
 - <http://bioconductor.org/packages/release/bioc/html/limma.html>
 - <https://genomebiology.biomedcentral.com/articles/10.1186/gb-2014-15-2-r29>
- RStudio
 - <https://www.rstudio.com/>

Files and file types

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

+

IIHHIIIIII55EHH

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

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

$Q_{solexa} = -10\log_{10}\frac{P}{1-P}$ - 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).
 - <http://samtools.github.io/hts-specs/SAMv1.pdf>
 - <http://samtools.github.io/hts-specs/SAMtags.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 is 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.
- A value 255 indicates that the mapping quality is not available.

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	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 "ENST0000
```

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=GENSCAN000000019335
X	Ensembl Variation		2413425	2413425	.	+	.
X	Ensembl Variation		2413805	2413805	.	+	.