DNA Sequence Bioinformatics Analysis with the Galaxy Platform

University of São Paulo, Brazil
28 July - 1 August 2014

Dave Clements
Johns Hopkins University

Robson Francisco de Souza
University of São Paulo

José Belizario
University of São Paulo

The Week's Agenda

Mon	Introductions: Cloud Computing, Nuvem Cloud, Basic Analysis in Galaxy
Tues	Workflows, Sharing, Quality Control, ChIP-Seq
Wed	ChIP-Seq cont., Genome Assembly, RNA-Seq
Thur	RNA-Seq continued, SNP and Variant Calling
Fri	Intro to Command Line, Genome Annotation using MAKER, CloudMan and AWS

bit.ly/gxyusp2014

Wednesday's Agenda

9:00 ChIP-Seq, continued

9:50 Genome Assembly

20 minute Break at around 10:20

12:00 Lunch

2:00 Open Discussion and Q & A

2:15 RNA-Seq Read Mapping with Tuxedo Suite

20 minute Break at around 3:20

4:00 Differential Expression Analysis with Tuxedo Suite of Tools

5:00 Done

Where we left off:
ChIP-Seq Analysis: Visualize Results

Look at the HTML report dataset

Launch a Trackster visualization and bring in

- the called peaks
- the Treatment WIG
- the Control WIG
- the gene definitions

ChIP-Seq Analysis: Replicates

Shared Data → Data Libraries → ChIP-Seq Datasets → Peaks

Import Peaks files for

Nanog Rep 2

Pou5f1 Rep 1

Pou5f1 Rep 2

ChIP-Seq Analysis: Unify Replicates

Operate on Genomic Intervals → Concatenate

Concatenate Nanog Rep 1 and 2 peak files

Operate on Genomic Intervals → Cluster

Use default parameters

Rename the output dataset

Add the Nanog cluster output to your visualization

ChIP-Seq Analysis: Unify Replicates

Repeat for Pou5f1 replicates

Operate on Genomic Intervals → Concatenate

Concatenate Pou5f1 Rep 1 and 2 Peak files

Operate on Genomic Intervals → Cluster

Use default parameters

Rename the output dataset

Add the Pou5f1 cluster output to your visualization

ChIP-Seq Analysis: Differential binding

Operate on Genomic Intervals → Subtract

First dataset clustered → Pou5f1

Second dataset clustered → Nanog

Return → Intervals with no overlap

ChIP-Seq Mapping With MACS
Further reading & Resources

ChIP-Seq: FASTQ data and quality control
by Shannan Ho Sui

HAIB TFBS ENCODE collection

MACS Documentation

Model-based analysis of ChIP-Seq (MACS)
by Zhang *et al.*

Cistrome and Nebula Galaxy Servers

Nebula Tutorial
by Valentina Boeva

Wednesday's Agenda

9:00 ChIP-Seq, continued

9:50 Genome Assembly

20 minute Break at around 10:20

12:00 Lunch

2:00 Open Discussion and Q & A

2:15 RNA-Seq Read Mapping with Tuxedo Suite

20 minute Break at around 3:20

4:00 Differential Expression Analysis with Tuxedo Suite of Tools

5:00 Done

Evolutionary Applications

Open Access

Evolutionary Applications ISSN 1752-4571

REVIEWS AND SYNTHESIS

A field guide to whole-genome sequencing, assembly and annotation

Robert Ekblom and Jochen B. W. Wolf

Department of Evolutionary Biology, Uppsala University, Uppsala, Sweden

Box 2: Before you start

Some important points to consider

- Availability of appropriate computational resources
- Collaboration with sequencing facility and bioinformatics groups
- Plan for amount and type of sequencing data needed
- Does funding allow to produce sufficient sequence coverage? If not, alternative approaches should be considered rather than producing a poor, low coverage, assembly
- Familiarization with data handling pipelines and file formats (see below)
- High-quality DNA sample (with individual metadata)
- Plan for analyses and publication

Basic considerations

Genome assembly is a challenging problem that requires time, resources and expertise. Before engaging in a genome sequencing project, it should thus be carefully considered whether a genome reference sequence is strictly necessary for the purpose in question.

it needs to be considered whether sufficient financial and computational resources are available to produce a genome of satisfactory quality. If funding is not available to obtain the appropriate read depth, it is advisable to utilize alternative approaches where possible (such as genotyping-by-sequencing or transcriptome sequencing), rather than settle for low-coverage whole-genome sequencing data. The latter would be a waste of funding, effort and time.

even more encouragement from Ekblom & Wolf

- it is essentially impossible to sequence and assemble all nucleotides in the genome (Ellengren 2014)
- there will also be some degree of error in the characterized genome sequence
- every genome assembly is the result of a series of assembly heuristics and should accordingly be treated as a working hypothesis
- it is often not realistic to aim for a chromosome level assembly

Best Practices

- Use several libraries covering different and longer insert sizes
- If using only short reads, ~100x coverage is needed. Suggested breakdown for mammals:
  - 45x coverage with short insert
  - 45x coverage with medium insert (3-10kb)
  - 1-5x coverage with long insert (10-40kb)
  - From Nagarajan and Pop, 2013

Best Practices

- Estimate genome size, sequencing error rates, repeat content and amount of genome duplication
- Can perform a pilot study to get these estimates.
- More repeats or duplication mean higher coverage
- Use inbred, parthenogenic or gynogenetic individuals. Heterozygosity is not your friend.

Beginner's guide to comparative bacterial genome analysis using next-generation sequence data

By David J Edwards and Kathryn E Holt

Microbial Informatics and Experimentation 2013, **3**:2

and the accompanying

Bacterial Comparative GenomicsTutorial

Create a new history

Shared Data → Data Libraries → **Assembly**

Select both FASTQ files

Illumina HiSeq paired-end reads
from *E. coli* O104:H4 strain TY-2482
(ENA accession SRR292770)

http://www.ebi.ac.uk/ena/data/view/SRR292770&display=html

http://www.ncbi.nlm.nih.gov/sra/SRX079805

NGS Assembly: Quality Control

Run FastQC Reports on both input datasets

NGS QC and Manipulation → Assembly

Only issue appears to be duplication

(How is it possible to *have* > 25% sequence duplication and then *not have any* overrepresented sequences?)

NGS Assembly: Quality Control

The duplication will affect the assembly.

The tutorial says you can use the FASTX Toolkit for this.

NGS: QC and Manipulation→ Collapse

Hmm, but

that will destroy our pairings

and

a pairing where only one end is a duplicate is not a duplicate

NGS Assembly: Quality Control

NGS: QC and Manipulation→ FASTQ Joiner
NGS: QC and Manipulation→ Collapse
NGS: QC and Manipulation→ FASTQ Splitter

But don't do this now. It is slow.

Just get the results from the ...

But don't do that either.

Collapse does not find any duplicates.

NGS Assembly: Velveth

NGS: Assembly → Velveth

Hash length?

Tutorial says use 35: authors have determined optimal value through experimentation.

The maximum k-mer-length Velvet can use is set at install/ compile time.

Use 35. We will revisit this, and other magic numbers

NGS Assembly: Velveth

Click on Add new Input Files

File format → FASTQ

Read type → shortPaired reads

Dataset → 1 (forward reads)

Dataset 2 (reverse reads)

Produces an index of the reads using the k-mer length.

Index is used by Velvetg to do actual mapping.

NGS Assembly: Velvetg

Velvetg does the actual assembly

Velvet Dataset → *Output dataset from velveth*

Check Generate unusedReads fasta file

The tutorial provides us with several "optimal" values to use.

Let's use them and then revisit them.

Coverage cutoff → Specify cutoff value → 2.81

Expected coverage of unique regions → Specify expected value → 21.0

Set minimum contig length → Yes → 200

Using paired end reads → Yes

NGS Assembly: Velvetg

Several output files

Unmapped Reads

Stats

Statistics about the graph nodes constructed during assembly.

Information about the internals of Velvetg.

Contigs

The list of contigs produced by this assembly run.

Let’s take a look at the contigs

NGS Assembly: Velvetg

Contigs

FASTA Manipulation → Compute Sequence Lengths

Give it the contigs file

Filter and Sort → Sort

Column 2, descending

NGS Assembly: Parameters

Remember these?

Hash size → 35

Coverage cutoff → Specify cutoff value → 2.81

Expected coverage of unique regions → Specify expected value → 21.0

Not very often will someone tell you the optimal values.

NGS Assembly: Hash Size (k-mer)

BIOINFORMATICS ORIGINAL PAPER

2013, pages 1–7
doi:10.1093/bioinformatics/btt310

Sequence analysis

Advance Access publication June 3, 2013

Informed and automated *k*-mer size selection for genome assembly

Rayan Chikhi[1] and Paul Medvedev[1,2,*]

[1]Department of Computer Science and Engineering and [2]Department of Biochemistry and Molecular Biology, The Pennsylvania State University, University Park, PA 16802, USA

Associate Editor: Gunnar Ratsch

ABSTRACT

Motivation: Genome assembly tools based on the de Bruijn graph framework rely on a parameter *k*, which represents a trade-off between several competing effects that are difficult to quantify. There is currently a lack of tools that would automatically estimate the best *k* to use and/or quickly generate histograms of *k*-mer abundances that would allow the user to make an informed decision.

Results: We develop a fast and accurate sampling method that constructs approximate abundance histograms with several orders of magnitude performance improvement over traditional methods. We then present a fast heuristic that uses the generated abundance histograms for putative *k* values to estimate the best possible value of *k*. We test the effectiveness of our tool using diverse sequencing datasets and find that its choice of *k* leads to some of the best assemblies.

Availability: Our tool KmerGenie is freely available at: http://kmergenie.bx.psu.edu/.

Contact: pashadag@cse.psu.edu

One issue is many assemblers' lack of robustness with respect to the parameters and the lack of any systematic approach to choosing the parameters. In de Bruijn-based assemblers, the most significant parameter is *k*, which determines the size of the *k*-mers into which reads are chopped up. Repeats longer than *k* nucleotides can tangle the graph and break-up contigs; thus, a large value of *k* is desired. On the other hand, the longer the *k* the higher the chances that a *k*-mer will have an error in it; therefore, making *k* too large decreases the number of correct *k*-mers present in the data. Another effect is that when two reads overlap by less than *k* characters, they do not share a vertex in the graph, and thus create a coverage gap that breaks-up a contig. Therefore, the choice of *k* represents a trade-off between several effects.

Because some of these trade-offs have been difficult to mathematically quantify, there has not been an explicit formula for choosing *k* taking into account all these effects. It is possible to

NGS Assembly: Parameters

KmerGenie

Compute the k-mer abundance histogram for many values of k.

For each value of k, predict the number of distinct genomic k-mers in the dataset

Return the k-mer length which maximizes this number.

Velvet Optimiser

Explore a range of parameter values and combinations.

Specifically for Velvet.

Pick the best combination of parameters

NGS Assembly: Parameters

Velvet Optimiser

Explores a range of parameter values and combinations

kmer range → 11-47

step size → 2

Click Add new input read library

File Type → shortPaired

Check Are the reads paired ...

Select read files

and ...

NGS Assembly: Parameters

Velvet Optimiser

Does it work?

Velvet Optimiser

and wait 45 minutes....

NGS Assembly: Velvet Optimiser

	Paper	VOptimiser
Kmer size	35	35
Coverage cutoff	2.81	1.29
Expected coverage	21	21
Contigs	312	313
Bases on contigs	5318312	5318446
N50	45802	45802

NGS Assembly: What next?

Scaffolding

Want to tie together those contigs into larger units called scaffolds.

Some software solutions for this.

Can also use related genomes.

Get more reads, possibly on a different platform,

or different insert length.

These can be provided at initial assembly time.

Bradnam *et al. GigaScience* 2013, **2**:10
http://www.gigasciencejournal.com/content/2/1/10

(GIGA)n SCIENCE

RESEARCH Open Access

Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species

Keith R Bradnam[1*†], Joseph N Fass[1†], Anton Alexandrov[36], Paul Baranay[2], Michael Bechner[39], Inanç Birol[33], Sébastien Boisvert[10,11], Jarrod A Cha
Wen-Chi Chou[14,16], Jacques Corbeil
Scott Emrich[3], Pavel Fedotov[36], Nun
Sante Gnerre[22], Élénie Godzaridis[11],
Joseph B Hiatt[41], Isaac Y Ho[20], Jason
Huaiyang Jiang[32], Sergey Kazakov[36],
Tak-Wah Lam[29], Dominique Lavenie
Yue Liu[32], Ruibang Luo[28,29], Iain Mac
Delphine Naquin[8,9], Zemin Ning[34], T
Francisco Pina-Martins[31], Michael Pla
Stephen Richards[32], Daniel S Rokhsa
David C Schwartz[39], Alexey Sergushi
Jared T Simpson[34], Henry Song[32], Fe
Jun Wang[28], Kim C Worley[32], Shuan
Shiguo Zhou[39] and Ian F Korf[1*]

NGS Assembly: What's *better*?

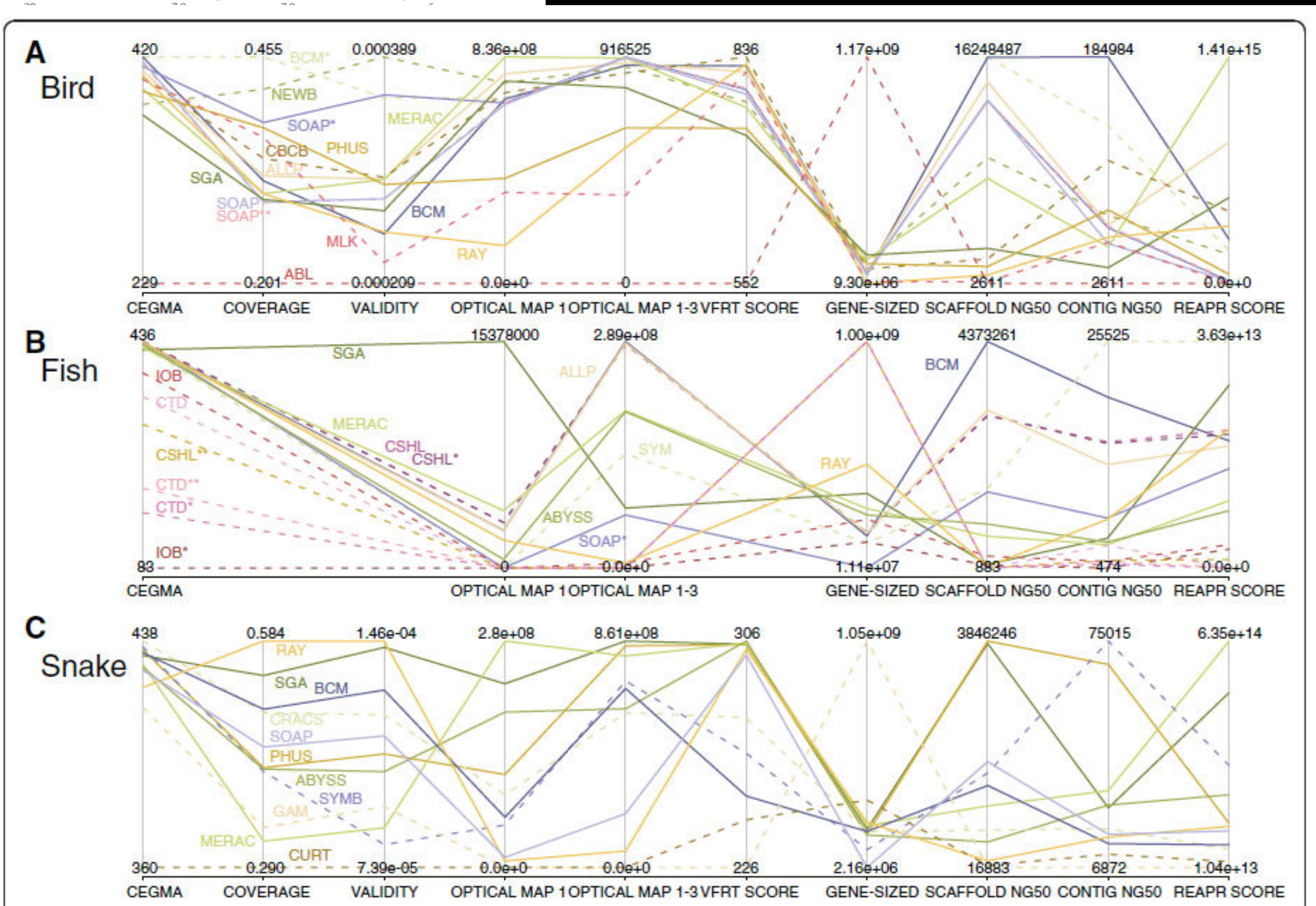


Figure 21 Parallel coordinate mosaic plot showing performance of all assemblies in each key metric. Performance of bird, fish, and snake

NGS Assembly: Resources and Reading

Beginner's guide to comparative bacterial genome analysis using next-generation sequence data

Bacterial Comparative Genomics Tutorial

By David J Edwards and Kathryn E Holt

Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species

Bradnam, *et al.*

Whole Genome Assembly and Alignment

Michael Schatz

Velvet Optimizer & Wrapper

Simon Gladman

Wednesday's Agenda

9:00 ChIP-Seq, continued

9:50 Genome Assembly Concepts

20 minute Break at around 10:20

11:10 Genome Assembly

12:00 Lunch

2:00 Open Discussion and Q & A

2:15 RNA-Seq Read Mapping with Tuxedo Suite

20 minute Break at around 3:20

4:00 Differential Expression Analysis with Tuxedo Suite of Tools

5:00 Done

Wednesday's Agenda

9:00 ChIP-Seq, continued

9:50 Genome Assembly Concepts

20 minute Break at around 10:20

11:10 Genome Assembly

12:00 Lunch

2:00 Open Discussion and Q & A

2:15 RNA-Seq Read Mapping with Tuxedo Suite

20 minute Break at around 3:20

4:00 Differential Expression Analysis with Tuxedo Suite of Tools

5:00 Done

Wednesday's Agenda

9:00 ChIP-Seq, continued

9:50 Genome Assembly Concepts

20 minute Break at around 10:20

11:10 Genome Assembly

12:00 Lunch

2:00 Open Discussion and Q & A

2:15 RNA-Seq Read Mapping with Tuxedo Suite

20 minute Break at around 3:20

4:00 Differential Expression Analysis with Tuxedo Suite of Tools

5:00 Done

RNA-Seq Quality Control

Run FastQC and review.

The 3 options introduced yesterday

- One preserves original read length, two don't
- One preserves number of reads, two don't
- Two keep/make every read the same length, one does not
- **One preserves pairings, two don't**

"Mixing paired- and single- end reads together is **not** supported."

Tophat Manual

"If you are performing RNA-seq analysis, there is no need to filter the data to ensure exact pairs before running Tophat."

Jen Jackson

Galaxy User Support Person Extraordinaire

"Dang."

Most of us

Running Tophat on *no-longer-cleanly-paired* data *does map the reads*, but, it no longer keeps track of read pairs in the SAM/BAM file.

Keeping paired ends paired: Options

- Don't bother.
- Run a workflow that removes any unpaired reads before mapping.
- Run the Picard Paired Read Mate Fixer after mapping reads.
- Use sliding windows for QC, but keep empty reads.

RNA-Seq Exercise

Create new history


(cog) → Create New

Get some data

Shared Data → Data Libraries

→ RNA-Seq Example*

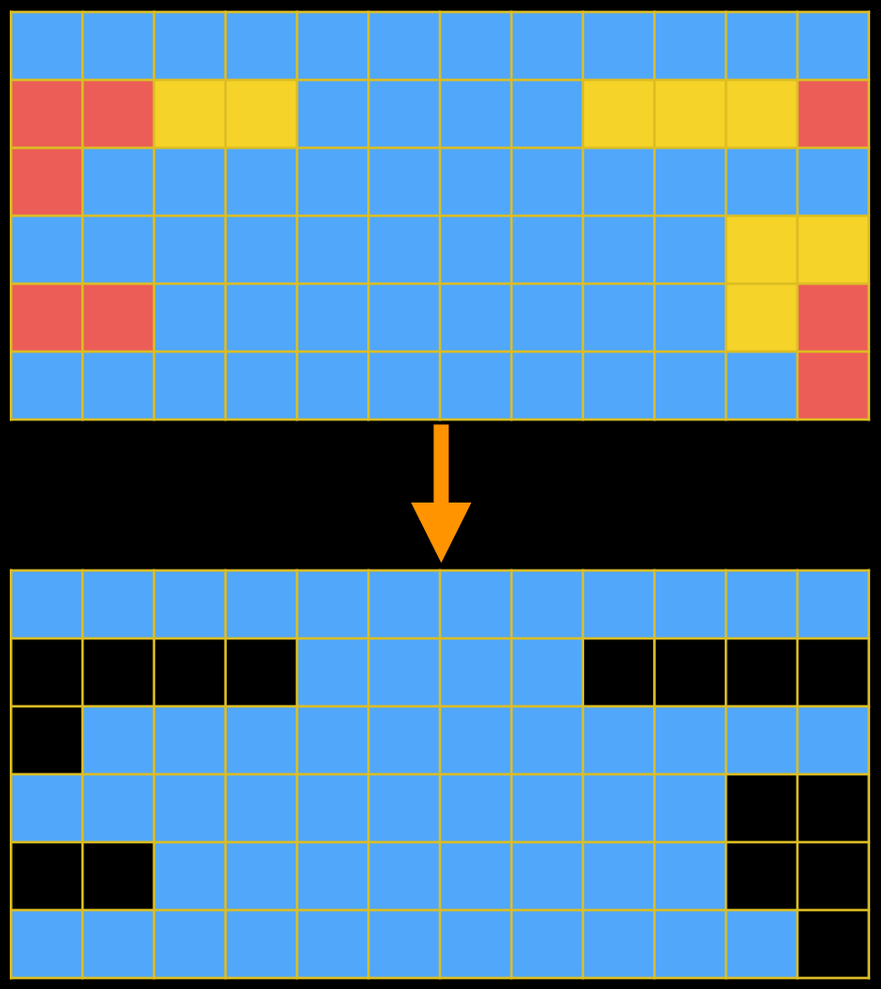
→ Untrimmed FASTQ

→ Select MeOH_REP1_R1, MeOH_REP1_R2
and then Import to current history

UCDAVIS Genome Center Bioinformatics Core

* RNA-Seq example datasets from the 2013 UC Davis Bioinformatics Short Course. http://bit.ly/ucdbsc2013

NGS Data Quality: Base Quality Trimming



I'll use Option 3 (*but ...):*

- NGS QC and Manipulation → **FASTQ Quality Trimmer by sliding window**

Check **"Keep reads with zero length"**

Run again:

- NGS QC and Manipulation → **FastQC** on trimmed dataset

NGS Data Quality: Base Quality Trimming

New Problem?

Now some reads are so short they are just noise and can't be meaningfully mapped

Option 2 can fix this (but break pairings).

Or, your mapper may have an option to ignore shorter reads

NGS Data Quality: Sequencing Artifacts

Repeat this process with MeOH Rep1 R2 (the reverse reads)

... and there's a problem in Overrepresented sequences:

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CTGTGTATTTGTCAATTTTCTTCTCCACGTTCTTCTCGGCCTGTTTCCGTAGCCT	590	0.3541692929220167	No Hit
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	342	0.2052981325073385	No Hit
CGGCCACAAATAAACACAGAAATAGTCCAGAATGTCACAGGTCCAGGGCAGAGGA	325	0.1950932545756871	No Hit
CTGCATTATAAAAAGGACAGCCAGATATCAACTGTTACAGAAATGAAATAAGACG	230	0.1380659955458709	No Hit
CGGCCGCAAATAAACACAGAAATAGTCCAGAATGTCACAGGTCCAGGGCAGAGGA	199	0.1194571004940361	No Hit
GTCAGCTCAACTTGTAGGCCCCAAAAGAAAACAGCGTCTTACTGGGGAGGGATAT	197	0.1182565266197242	No Hit

NGS QC and Manipulation → **Remove sequencing artifacts**

But this will break pairings.

NGS Data Quality: Done with 1st Replicate!

Now, only 5 more to go!

Workflows?

Create a QC workflow that does the trimming

Or, cheat and just import the already trimmed datasets from the RNA-Seq Example → Trimmed FASTQ shared data library

RNA-seq Exercise: Mapping with Tophat

Create a new history

Import all datasets from library:

RNA-Seq Example → Trimmed FASTQ

and genes_chr12.gtf

RNA-seq Exercise: Mapping with Tophat

- Tophat looks for best place(s) to map reads, and best places to insert introns
- *Imagine pages and pages of discussion on the intricacies and pitfalls of RNA-seq mapping here.*

Mapping with Tophat: mean inner distance

Expected distance between paired end reads

- Determined by sample prep
- We'll use 90* for mean inner distance
- We'll use 50 for standard deviation

✳ The library was constructed with the typical Illumina TruSeq protocol, which is supposed to have an average insert size of 200 bases. Our reads are 55 bases (R1) plus 55 bases (R2). So, the Inner Distance is estimated to be 200 - 55 - 55 = 90

From the 2013 UC Davis Bioinformatics Short Course

Mapping with Tophat: Use Existing Annotations?

You can bias Tophat towards known annotations

- Use Own Junctions → Yes
  - Use Gene Annotation → Yes
    - Gene Model Annotation → genes_chr12.gtf
- Use Raw Junctions → Yes (tab delimited file)
- Only look for supplied junctions → Yes

Mapping with Tophat: Make it quicker?

Warning: Here be dragons!

- Allow indel search → No
- Use Coverage Search → No (wee dragons)

TopHat generates its database of possible splice junctions from two sources of evidence. The first and strongest source of evidence for a splice junction is when two segments from the same read (for reads of at least 45bp) are mapped at a certain distance on the same genomic sequence or when an internal segment fails to map - again suggesting that such reads are spanning multiple exons. With this approach, "GT-AG", "GC-AG" and "AT-AC" introns will be found *ab initio*. The second source is pairings of "coverage islands", which are distinct regions of piled up reads in the initial mapping. Neighboring islands are often spliced together in the transcriptome, so TopHat looks for ways to join these with an intron. We only suggest users use this second option (--coverage-search) for short reads (< 45bp) and with a small number of reads (<= 10 million). This latter option will only report alignments across "GT-AG" introns

TopHat Manual

Mapping with Tophat: Max # of Alignments Allowed

Some reads align to more than one place equally well.

For such reads, how many should Tophat include?

If more than the specified number, Tophat will pick those with the best mapping score.

Tophat breaks ties randomly.

Tophat assigns equal fractional credit to all *n*

Instructs TopHat to allow up to this many alignments to the reference for a given read, and choose the alignments based on their alignment scores if there are more than this number. The default is 20 for read mapping. Unless you use --report-secondary-alignments, TopHat will report the alignments with the best alignment score. If there are more alignments with the same score than this number, TopHat will randomly report only this many alignments. In case of using --report-secondary-alignments, TopHat will try to report alignments up to this option value, and TopHat may randomly output some of the alignments with the same score to meet this number.

TopHat Manual

RNA-Seq Mapping With Tophat: Resources

RNA-Seq Concepts, Terminology, and Work Flows
by Monica Britton

Aligning PE RNA-Seq Reads to a Genome
by Monica Britton

both from the UC Davis 2013 Bioinformatics Short Course

RNA-Seq Analysis with Galaxy
by Jeroen F.J. Laros, Wibowo Arindrarto, Leon Mei

from the GCC2013 Training Day

RNA-Seq Analysis with Galaxy
by Curtis Hendrickson, David Crossman, Jeremy Goecks

from the GCC2012 Training Day

Wednesday's Agenda

9:00 ChIP-Seq, continued

9:50 Genome Assembly Concepts

20 minute Break at around 10:20

11:10 Genome Assembly

12:00 Lunch

2:00 Open Discussion and Q & A

2:15 RNA-Seq Read Mapping with Tuxedo Suite

20 minute Break at around 3:20

4:00 Differential Expression Analysis with Tuxedo Suite of Tools

5:00 Done

Cuffdiff?

- Part of the Tuxedo RNA-Seq Suite (as are Tophat and Bowtie)
- Identifies differential expression between multiple datasets
- Widely used and widely installed on Galaxy instances

NGS: RNA Analysis → Cuffdiff

Cuffdiff?

Cuffdiff uses FPKM/RPKM as a central statistic.
Total # mapped reads heavily influences FPKM/RPKM.
Can lead to challenges when you have very highly expressed genes in the mix.

Cuffdiff

- Running with 2 Groups: MeOH and R3G
- Each group has 2 replicates each

Cuffdiff

- Which Transcript definitions to use?
  - Official
  - MeOH or R3G Cufflinks transcripts
  - Results of Cuffmerge on MeOH & R3G Cufflinks transcripts
- Depends on what you care about

NGS: RNA Analysis → Cuffdiff

Cuffdiff

- Produces many output files, all explained in doc
- We'll focus on gene differential expression testing files (also care about gene FPKM files)
- Column 7 ("status") can be FAIL, NOTEST, LOWDATA or OK
- Filter and Sort → Filter
  - c7 == 'OK'
  - Column 14 ("significant") can be yes or no
- c14 == 'yes'

Thanks

Dave Clements

Galaxy Project

Johns Hopkins University

outreach@galaxyproject.org

Cuffdiff Alternatives

Rapaport, *et al*., "Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data."
Genome Biology 2013, 14:R95 doi:10.1186/gb-2013-14-9-r95

Reviews 7 packages

Each tool has it's own strengths and weaknesses.

What's a biologist to do?

Alternatives: What's a biologist to do?

Learn the strengths and weaknesses of the tools you have ready access to. Are they a good match for the questions you are asking?

If not, then research alternatives, identify good options and then work with your bioinformatics/systems people to get access to those tools.

Cuffdiff Alternatives: DESeq

DESeq is an R based differential expression analysis package where expression analysis is much more effectively isolated between features.

Cuffdiff Alternatives: DESeq

Takes a simple, tab delimited list of features and read counts across different samples.

First, have to create that list.

htseq-count

Is a tool that walks BAM files producing these lists

Cuffdiff Alternatives: DESeq

NGS: SAM Tools → htseq-count
once for each BAM file

Join the HTSeq datasets together on gene name

Cut out the duplicate gene name columns

OR, just use the 6x DESeq Prep workflow

NGS: RNA Analysis → DE Seq

Cuffdiff Alternatives: DESeq

DESeq output is a list of genes,
sorted by adjusted P value,
with lowest P values listed first

How many genes have an adjusted P value < 0.05 ?

Differential Expression: Reading & Resources

Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data
by Rapaport, *et al.*

DESeq Reference Manual

DESeq Galaxy Wrapper
by Nikhil Joshi

htseq-count Galaxy Wrapper
by Lance Parsons