NGS Data Analysis and Galaxy

University of Cape Town
Cape Town, South Africa
21-25 October 2013

Dave Clements, Emory University
http://galaxyproject.org/

Gerrit Botha
Computational Biology
University of Cape Town
http://cbio.uct.ac.za/

CBIO
Computational Biology @ UCT

ACGT
AFRICAN CENTRE FOR GENE TECHNOLOGIES

Galaxy

amazon webservices™

UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

South Africa
Galaxy
Workshop Tour

This Week

Monday	Welcome, Project Intro, Basic Galaxy Usage NGS QualityControl
Tuesday	RNA-Seq - Mapping and Transcript Prediction RNA-Seq: Differential expression and Alternative Pipelines; SNP & Variant Analysis
Wednesday	SNP & Variant Analysis Chip-Seq Analysis
Thursday	Genome Assembly Install your own Galaxy on Amazon Cloud
Friday	Customizing Galaxy, Galaxy Tool Shed, and Wrapping Tools for Galaxy

Thursday Agenda

9:00	Welcome and Questions
9:15	*de novo* Genome Assembly, Part I
11:00	Break
11:30	*de novo* Genome Assembly, Part II
13:00	Lunch
14:00	Galaxy CloudMan on Amazon, Part I
15:30	Break
16:00	Galaxy CloudMan on Amazon, Part I
17:00	Done, Feedback

Acknowledgements

Cape Town

Gerrit Botha
Ayton Meintjes
Sumir Panji
Nikola Mulder
Cashifa Kerriem

Pretoria

Jessika Samuels
Fourie Joubert
John Becker
Burger van Jaarsveld
John Ambler

Atlanta

Dannon Baker

Tutorials & Datasets

RNA-Seq

Monica Britton, Nikhil Joshi, Joe Fass

ChIP-Seq

Shannan J. Ho Sui, Oliver Hoffman

Assembly

David Edwards, Kathryn Holt

Acknowledgements

ACGT

AFRICAN CENTRE FOR GENE TECHNOLOGIES

CBIO

Computational Biology @ UCT

NIH National Institutes of Health

Turning Discovery Into Health

AD DESTINATUM PERSEQUOR

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

amazon webservices™

SPES BONA

UNIVERSITY OF CAPE TOWN

IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

Thursday Agenda

9:00	Welcome and Questions
9:15	*de novo* Genome Assembly, Part I
11:00	Break
11:30	*de novo* Genome Assembly, Part II
13:00	Lunch
14:00	Galaxy CloudMan on Amazon, Part I
15:30	Break
16:00	Galaxy CloudMan on Amazon, Part I
17:00	Done, Feedback

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 Genomics Tutorial

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

FastQC Reports for both input datasets are in

Shared Data → Assembly

Note the very different results from RNA-Seq

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 Assembly Data Library

Shared Data → ...

But don’t do that either.

Collapse does not find any duplicates.

(Why? And why didn’t we do this with the RNA-Seq data?)

NGS Assembly: Velveth

NGS: Assembly → Velveth

Hash length?

Gives us choices from 11 to 29. But the tutorial says use 35: they have determined optimal value through experimentation.

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

Use 29. We will revisit this.

NGS Assembly: Velveth

Click on Add new Input Files

File format → FASTQ

Read type → shortPaired reads

Dataset → 1 (forward reads)

Repeat for 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?

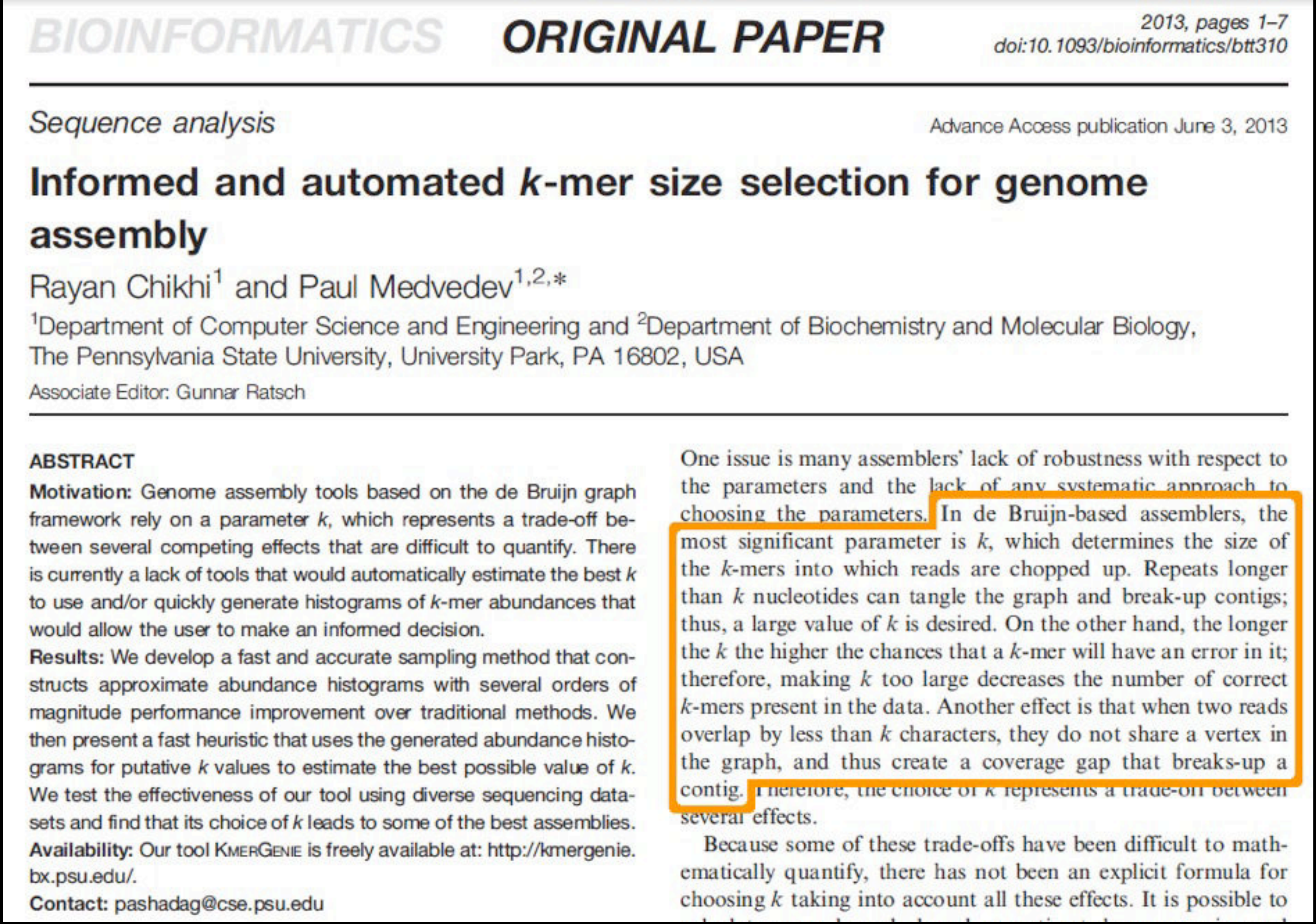
Hash size → 29

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

KmerGenie

Only takes one input. We have two inputs.

Concatenate them.

Parameters → Set manually

kmer range → 11-29

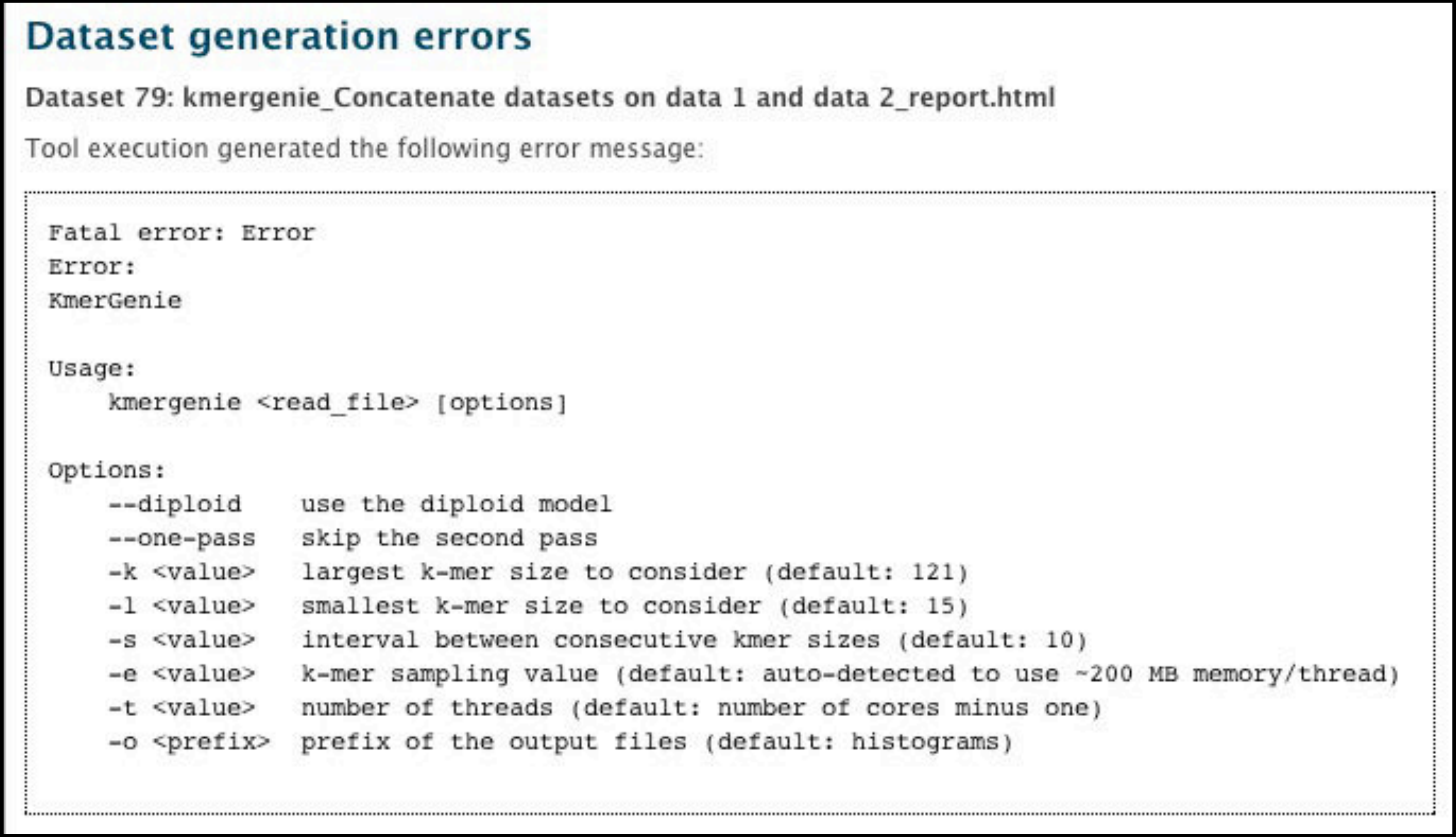
step size → 2

Kmer sampling → Automatic

Check Precise estimation of k

Execute it

KmerGenie gives us ... an error



Dataset generation errors

Dataset 79: kmergenie_Concatenate datasets on data 1 and data 2_report.html

Tool execution generated the following error message:

```
Fatal error: Error
Error:
KmerGenie

Usage:
    kmergenie <read_file> [options]

Options:
    --diploid    use the diploid model
    --one-pass   skip the second pass
    -k <value>   largest k-mer size to consider (default: 121)
    -l <value>   smallest k-mer size to consider (default: 15)
    -s <value>   interval between consecutive kmer sizes (default: 10)
    -e <value>   k-mer sampling value (default: auto-detected to use ~200 MB memory/thread)
    -t <value>   number of threads (default: number of cores minus one)
    -o <prefix>  prefix of the output files (default: histograms)
```

Is it giving us any idea what went wrong?

NGS Assembly: Parameters

KmerGenie

Rename the concatenated input dataset so there are no embedded spaces in the name.

Rerun with same parameters as before

NGS Assembly: Parameters

KmerGenie

KmerGenie offers us guidance on *one* of the key parameters to Velvet.

The k-mer length is a key input to any de Bruijn graph based assembler, of which there are several

However, we have Velvet, and velvet has a few other key parameters.

Is there any way we can estimate them?

NGS Assembly: Parameters

Velvet Optimiser

Explores a range of parameter values and combinations

kmer range → 11-29

step size → 2

Click Add new input read library

File Type → shortPaired

Check Are the reads paired ...

Select read files

and ...

NGS Assembly: Velvet Optimiser

... and

Click Execute *and then wait several hours,*

or

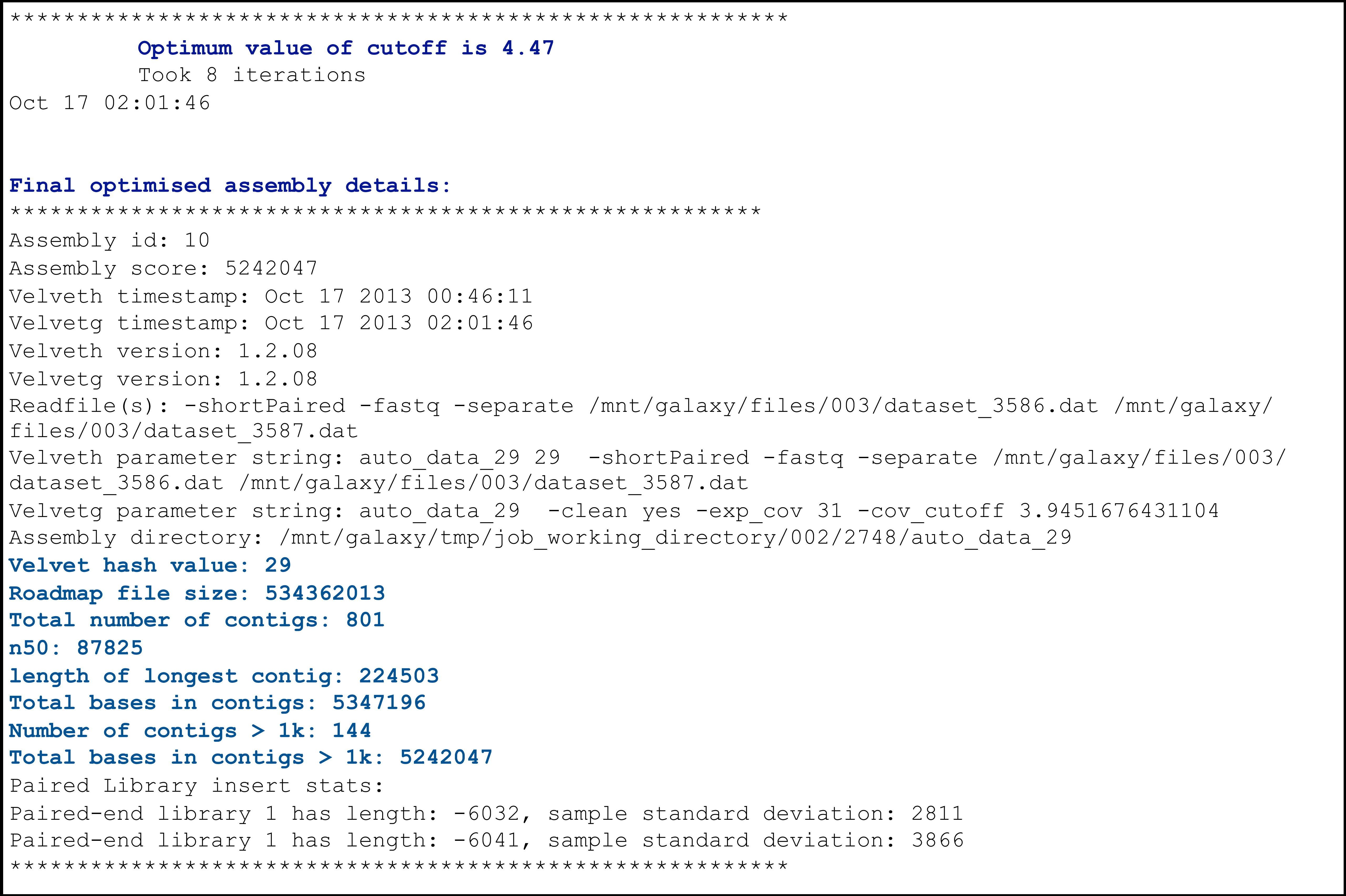
Get the results from the data library

Shared Data → Data Libraries → Assembly

Import Velvet Optimiser: Contigs, Velvet Optimiser: Logfile

NGS Assembly: Velvet Optimiser

Let’s look at the log file (well, I’ll look at the logfile, its big)



```
**********************************************************
          Optimum value of cutoff is 4.47
          Took 8 iterations
Oct 17 02:01:46


Final optimised assembly details:
********************************************************
Assembly id: 10
Assembly score: 5242047
Velveth timestamp: Oct 17 2013 00:46:11
Velvetg timestamp: Oct 17 2013 02:01:46
Velveth version: 1.2.08
Velvetg version: 1.2.08
Readfile(s): -shortPaired -fastq -separate /mnt/galaxy/files/003/dataset_3586.dat /mnt/galaxy/
files/003/dataset_3587.dat
Velveth parameter string: auto_data_29 29  -shortPaired -fastq -separate /mnt/galaxy/files/003/
dataset_3586.dat /mnt/galaxy/files/003/dataset_3587.dat
Velvetg parameter string: auto_data_29  -clean yes -exp_cov 31 -cov_cutoff 3.9451676431104
Assembly directory: /mnt/galaxy/tmp/job_working_directory/002/2748/auto_data_29
Velvet hash value: 29
Roadmap file size: 534362013
Total number of contigs: 801
n50: 87825
length of longest contig: 224503
Total bases in contigs: 5347196
Number of contigs > 1k: 144
Total bases in contigs > 1k: 5242047
Paired Library insert stats:
Paired-end library 1 has length: -6032, sample standard deviation: 2811
Paired-end library 1 has length: -6041, sample standard deviation: 3866
**********************************************************
```

NGS Assembly: Velvet Optimzer

Contigs

FASTA Manipulation → Compute Sequence Lengths

Give it the contigs file

Filter and Sort → Sort

Column 2, descending

Compare with the hand-tuned parameter run

NGS Assembly: What's *better*?

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[1
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*]

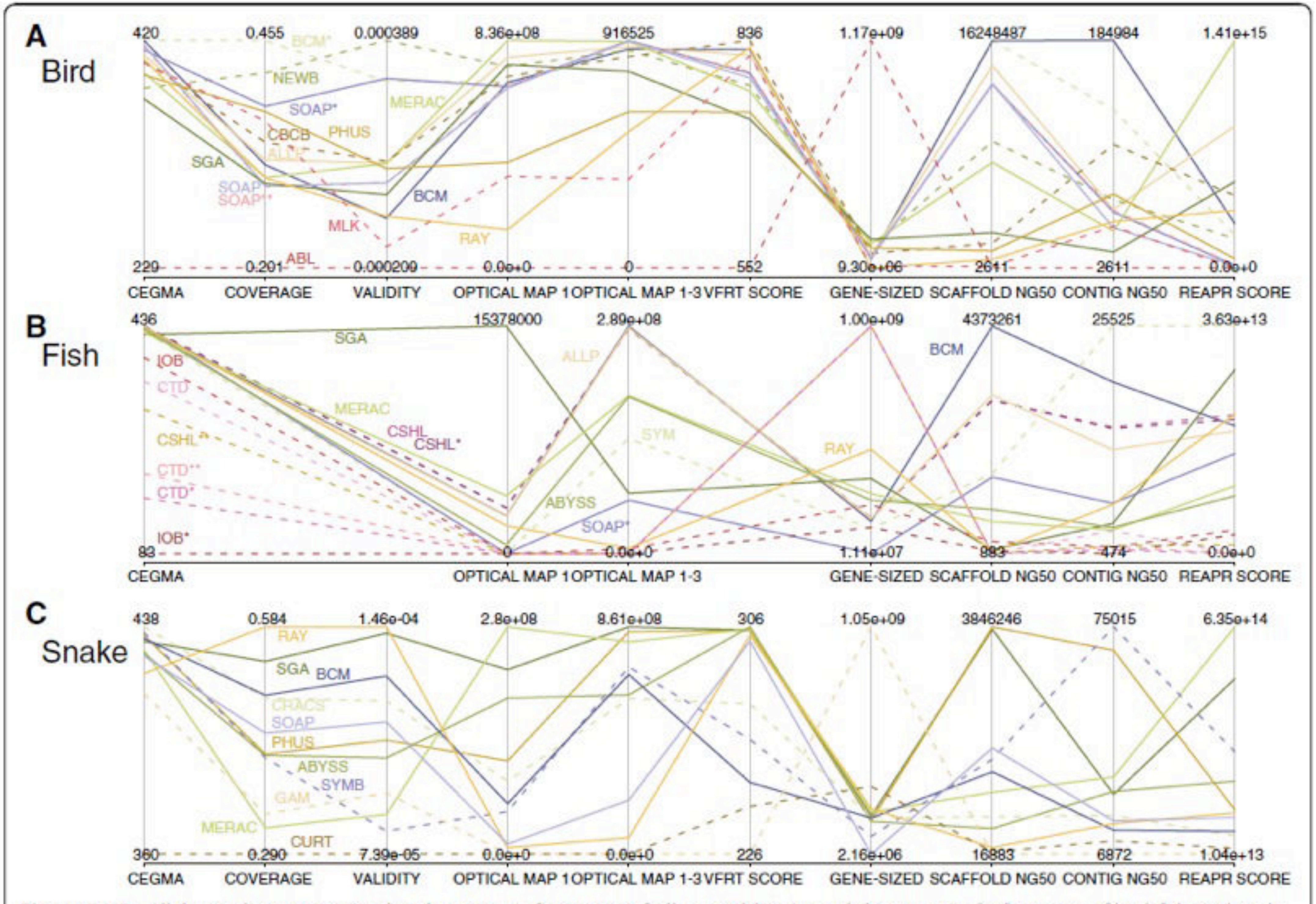


Figure 21 Parallel coordinate mosaic plot showing performance of all assemblies in each key metric. Performance of bird, fish, and snake

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.

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

KmerGenie Wrapper

Rayan Chikhi

Velvet Optimizer & Wrapper

Simon Gladman

Thursday Agenda

9:00	Welcome and Questions
9:15	*de novo* Genome Assembly, Part I
11:00	Break
11:30	*de novo* Genome Assembly, Part II
13:00	Lunch
14:00	Galaxy CloudMan on Amazon, Part I
15:30	Break
16:00	Galaxy CloudMan on Amazon, Part I
17:00	Done, Feedback

Thursday Agenda

9:00	Welcome and Questions
9:15	*de novo* Genome Assembly, Part I
11:00	Break
11:30	*de novo* Genome Assembly, Part II
13:00	Lunch
14:00	Galaxy CloudMan on Amazon, Part I
15:30	Break
16:00	Galaxy CloudMan on Amazon, Part I
17:00	Done, Feedback

Galaxy CloudMan

http://usegalaxy.org/cloud

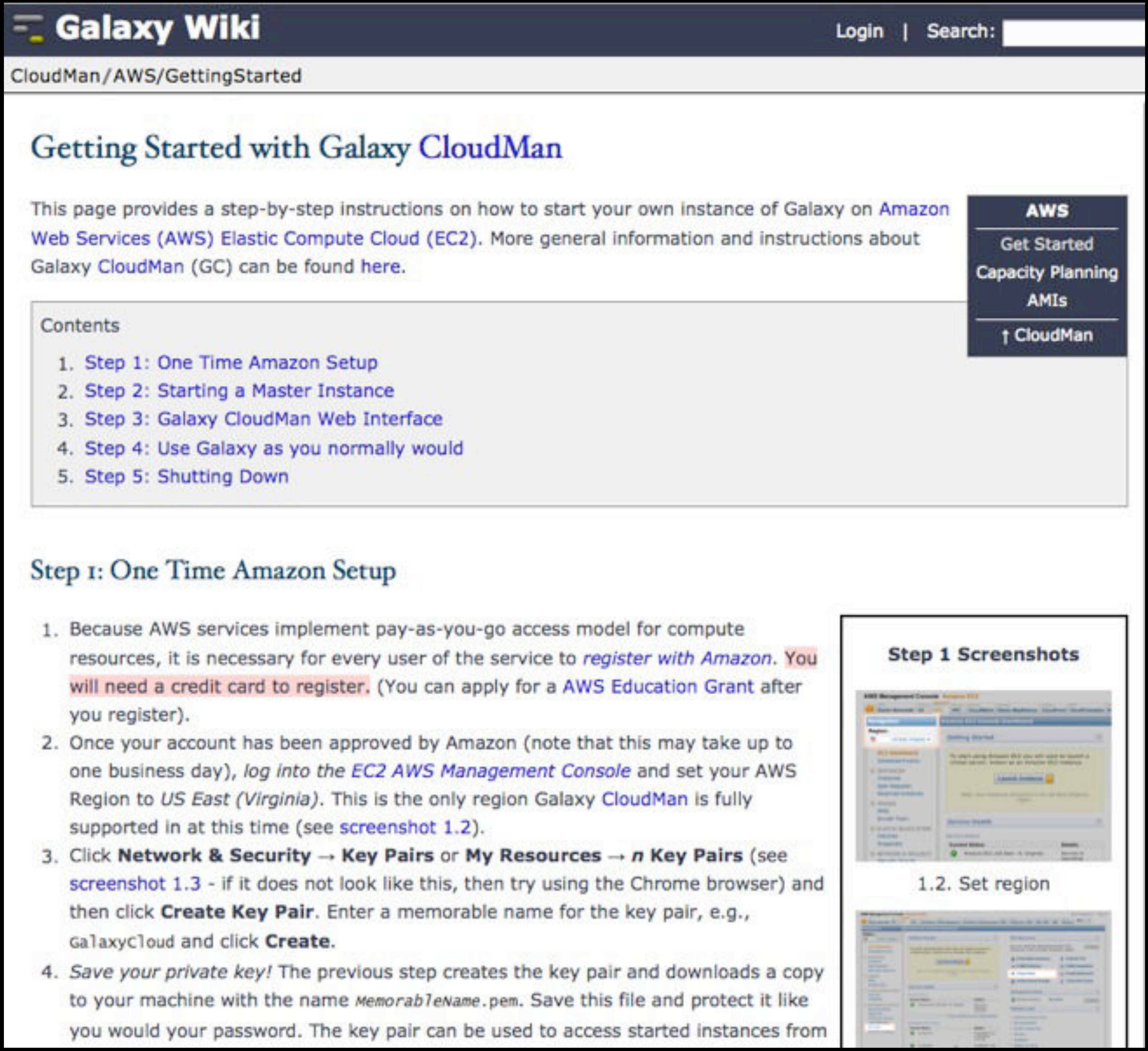
- Start with a **fully configured and populated** (tools and data) Galaxy instance.
- Allows you to scale up and down your compute assets as needed.
- Someone else manages the data center.
- **We are using this today.**

amazon web services™

- **You will set up an instance now**

http://aws.amazon.com/education

Could follow the step by step instructions on the wiki, but ... AWS just revamped it's interface.



Getting Started with Galaxy CloudMan

This page provides a step-by-step instructions on how to start your own instance of Galaxy on Amazon Web Services (AWS) Elastic Compute Cloud (EC2). More general information and instructions about Galaxy CloudMan (GC) can be found here.

Contents

Step 1: One Time Amazon Setup

1. Because AWS services implement pay-as-you-go access model for compute resources, it is necessary for every user of the service to *register with Amazon*. You will need a credit card to register. (You can apply for a AWS Education Grant after you register).
2. Once your account has been approved by Amazon (note that this may take up to one business day), *log into the EC2 AWS Management Console* and set your AWS Region to *US East (Virginia)*. This is the only region Galaxy CloudMan is fully supported in at this time (see screenshot 1.2).
3. Click **Network & Security** → **Key Pairs** or **My Resources** → ***n* Key Pairs** (see screenshot 1.3 - if it does not look like this, then try using the Chrome browser) and then click **Create Key Pair**. Enter a memorable name for the key pair, e.g., `GalaxyCloud` and click **Create**.
4. *Save your private key!* The previous step creates the key pair and downloads a copy to your machine with the name `MemorableName.pem`. Save this file and protect it like you would your password. The key pair can be used to access started instances from

Step 1 Screenshots

1.2. Set region

AWS Credentials

http://bit.ly/??????/

Be and Admin!

Create an account
Use CloudMan to make it an admin

Add some tools!

htseq-count
KmerGenie

Be and Admin!

Get some data!

Copy datasets FROM the published history TestYourToolsData (on your cloud1/2/3 instance!)

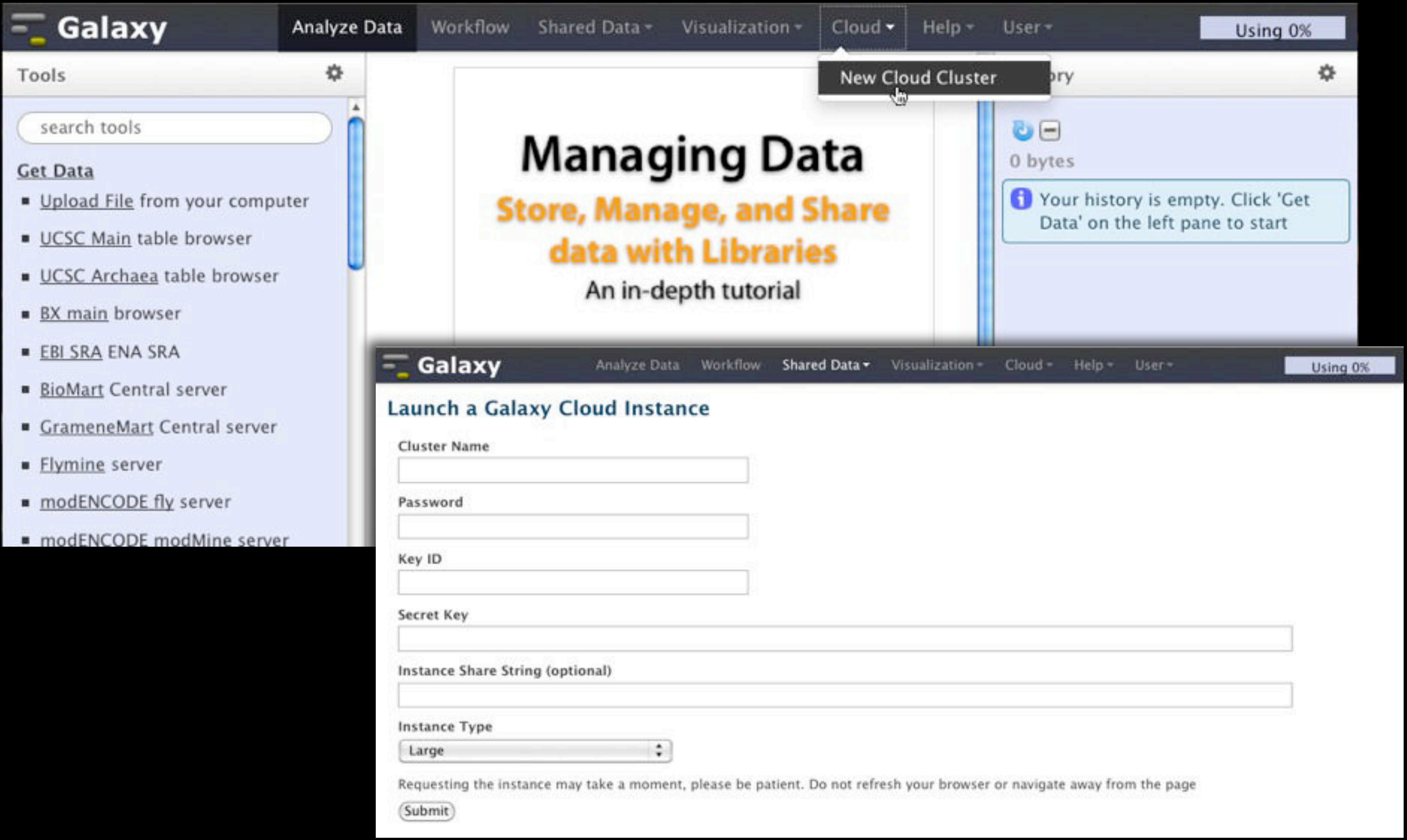
Test those tools!

htseq-count

KmerGenie

Instant CloudMan

http://usegalaxy.org/cloudlaunch



Your history is empty. Click 'Get Data' on the left pane to start

Managing Data

Store, Manage, and Share data with Libraries

An in-depth tutorial

Launch a Galaxy Cloud Instance

Cluster Name

Password

Key ID

Secret Key

Instance Share String (optional)

Instance Type

Large

Requesting the instance may take a moment, please be patient. Do not refresh your browser or navigate away from the page

Feedback

http://bit.ly/
gxyuctfeed

Feedback

http://bit.ly/gxyuctfeed

Thanks

Dave Clements

Galaxy Project
Emory University

clements@galaxyproject.org