National Center for Genome Analysis Support
Indiana University Pervasive Technology Institute

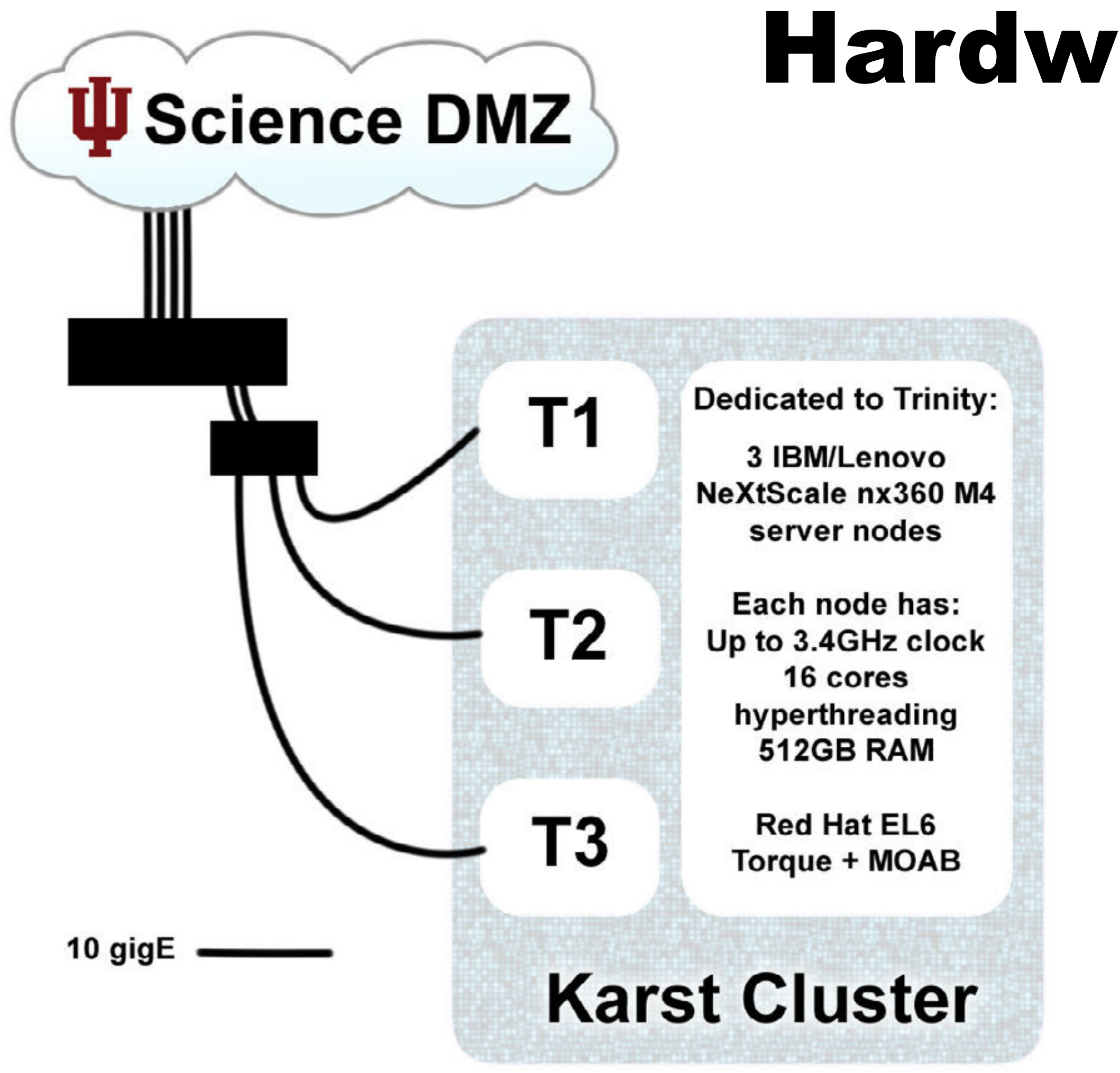
Trinity Galaxy Portal

Carrie Ganote ❄ Ben Fulton ❄ Brian Haas ❄ Timothy Tickle ❄ Le-Shin Wu ❄ Thomas Doak

The Trinity RNA-Seq software package is the leading tool for *de novo* assembly of transcriptomes [1]. Genes are transcribed into RNAs in the nucleus, processed, and exported to the cytoplasm as mature messenger RNA (mRNA). Here they program all the cell's protein production. mRNA can be specifically captured, converted into RNA-Seq libraries, and sequenced quickly and relatively cheaply with Next Generation Sequencing (NGS).

This poster gives you an outline of the way we set up a Galaxy [3,4,5] instance specifically geared for this software package and also to provide some benchmarking of the kind of uptake and use we see on our resources.

Hardware Setup


Science DMZ

T1

T2

T3

Dedicated to Trinity:

3 IBM/Lenovo NeXtScale nx360 M4 server nodes

Each node has:
Up to 3.4GHz clock
16 cores
hyperthreading
512GB RAM

Red Hat EL6
Torque + MOAB

10 gigE

Karst Cluster

The work to deploy Trinity Galaxy, to develop new tools for cancer research, and to optimize Trinity further for HPC applications was funded by the National Institutes of Health (NIH), specifically the National Cancer Institute (NCI), award number U24CA180922. Included in this award was funding for hardware to set up the Galaxy front end for Trinity.

Three nodes were purchased to be added to Indiana University's Karst cluster - a general purpose Linux supercomputer for research and academic use. The Trinity software has high memory requirements and is not suitable for running with large input sizes on desktop-caliber machines. Due to this limitation, Trinity is not available on many public Galaxy instances. The half-terabyte nodes for the Trinity Galaxy should be sufficient for most Trinity runs.

About the National Center for Genome Analysis Support

NCGAS began as an informal group within Research Technologies at IU which focused on support for IU's biology department. The need for larger scale computation has grown with the advances in lab technology, especially with the explosion of throughput provided by NGS.

NCGAS was formalized as a genomics support group with the receipt of an Advances in Biological Informatics (ABI) award from the National Science Foundation (NSF). Funding began in late 2011 with award number 1062432. NCGAS offers services to any NSF-funded genomics project as well as supporting its IU user base. Services include:

- Galaxy support, of course!
- Genome browser setup and maintenance
- Data stewardship through the Scholarly Data Archive and IUScholarWorks
- Entire analysis - usually for coauthorship
- Advice on experimental setup
- Linux and software help
- Installation of bioinformatics software on NCGAS machines
- Analysing and interpreting results

Trinity Software

The Trinity software can be broken down into three major phases or componenents:

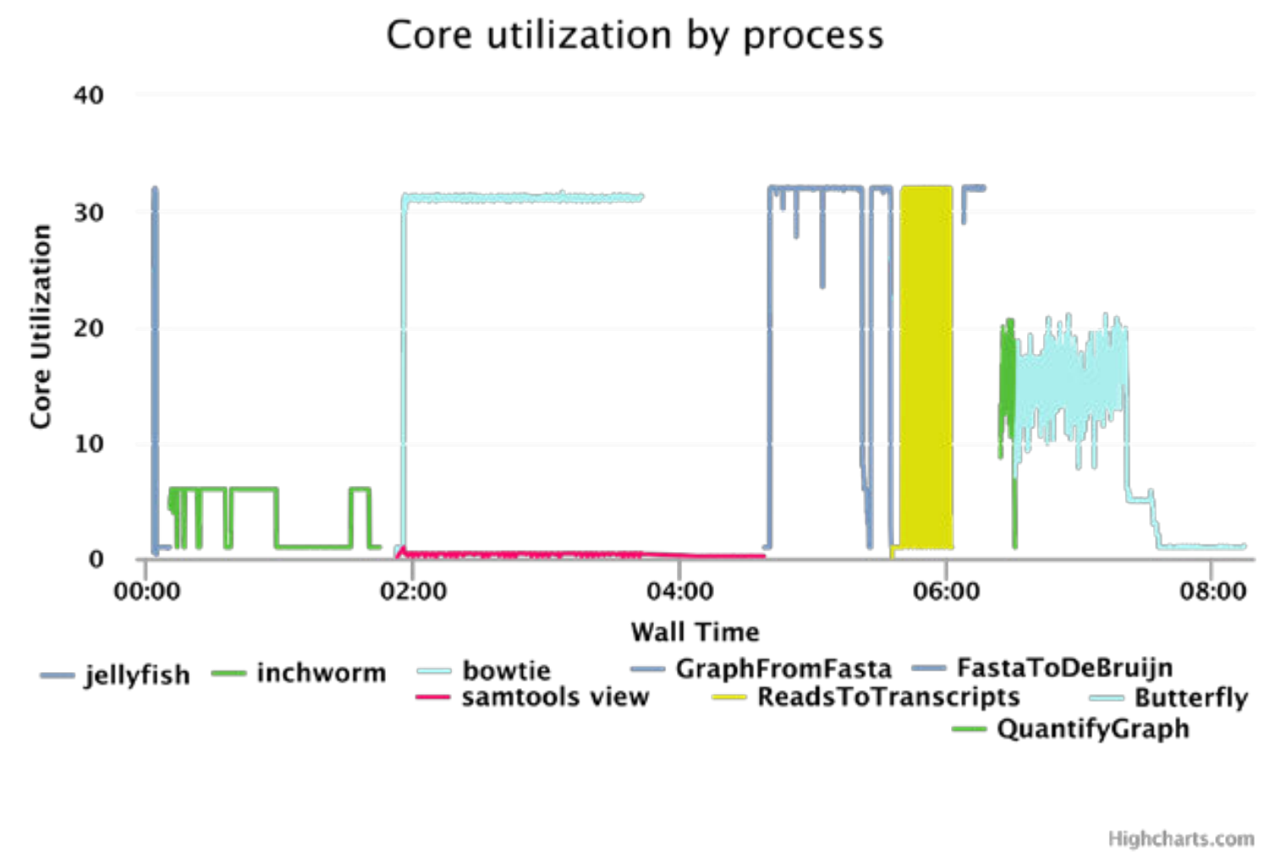


Figure [1]: Mus musculus data set core utilization in number of cores (32 total)
Command: Trinity --seqType fq --JM 20G --SS_lib_type RF --output ./trinity_out --CPU 32 --monitoring --left reads.left.fq --right reads.right.fq

• Inchworm • Chrysalis • Butterfly

Each of these steps performs a specific function which differs in resource requirements and in ability to run in parallel. The last step, for example, can take best advantage of CPU power by exhaustively multithreading. We have used the Detonate [1] software to benchmark different versions of Trinity - you may also see the performance pages at http://trinityrnaseq.github.io [2].

When gathering statistics on the memory consumption of Trinity, it is important to note that the statistics reported by the deploying machine might report the maximum memory usage - as shown on the right, this is not always a sustained level of memory use. It is important to know the behavior of a tool when optimizing Galaxy to run it.

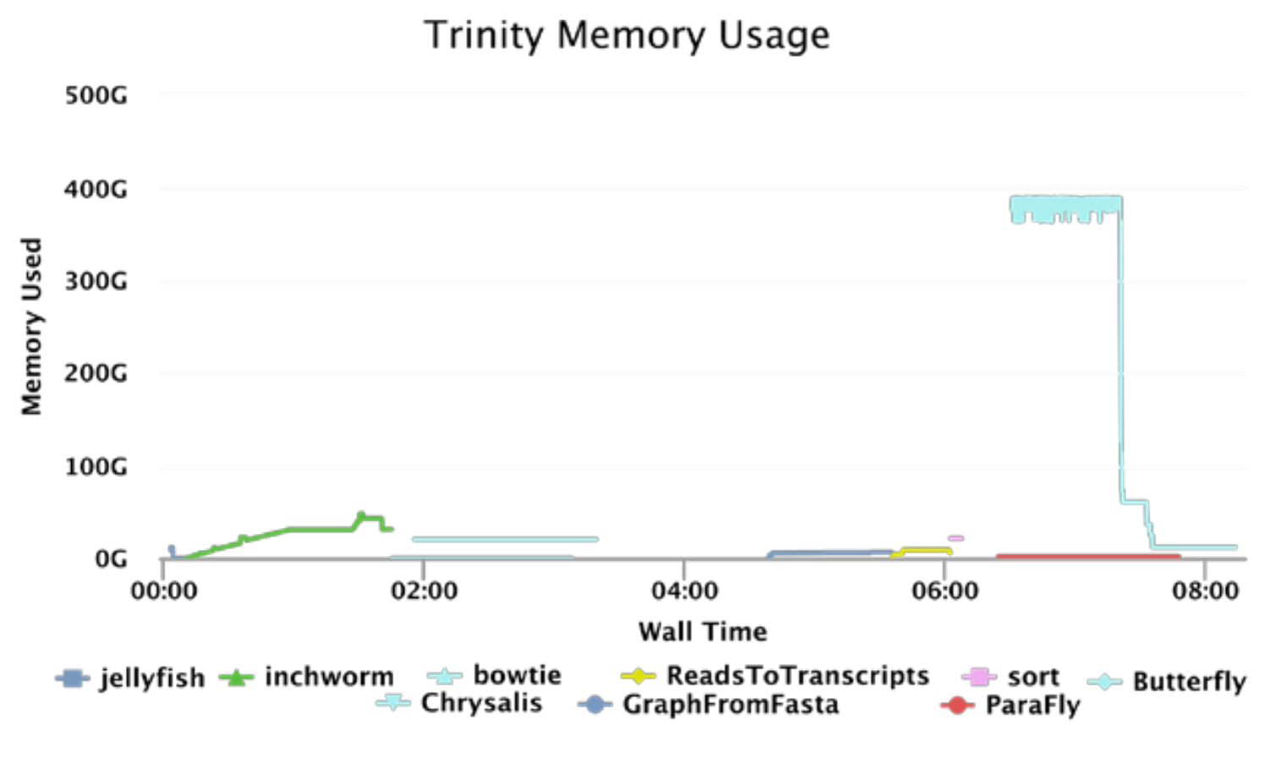


Figure [2]: Mus musculus data set memory usage
Command: Trinity --seqType fq --JM 20G --SS_lib_type RF --output ./trinity_out --CPU 32 --monitoring --left reads.left.fq --right reads.right.fq

Preprocessing

FastQ files typically contain millions of reads. Many of these reads do not add extra information - they are essentially redundant - but it takes extra time and resources downstream to deal with them. For this reason, digital normalization is recommended in order to make the job of assembly easier. Normalization removes reads that are likely to be sequencing error or are highly redundant.

Inchworm

This step starts the assembly process by first breaking the reads into smaller components called k-mers. In Trinity, a k-mer is a 25 base pair substring of the original read. The Jellyfish software package [6] is used to dissolve the reads into all possible 25 bp substring and count the occurance of each. Once the substrings are accounted for, Inchworm builds transcripts by joining k-mers back together. Alternatively spliced transcripts are not fully resolved at this step.

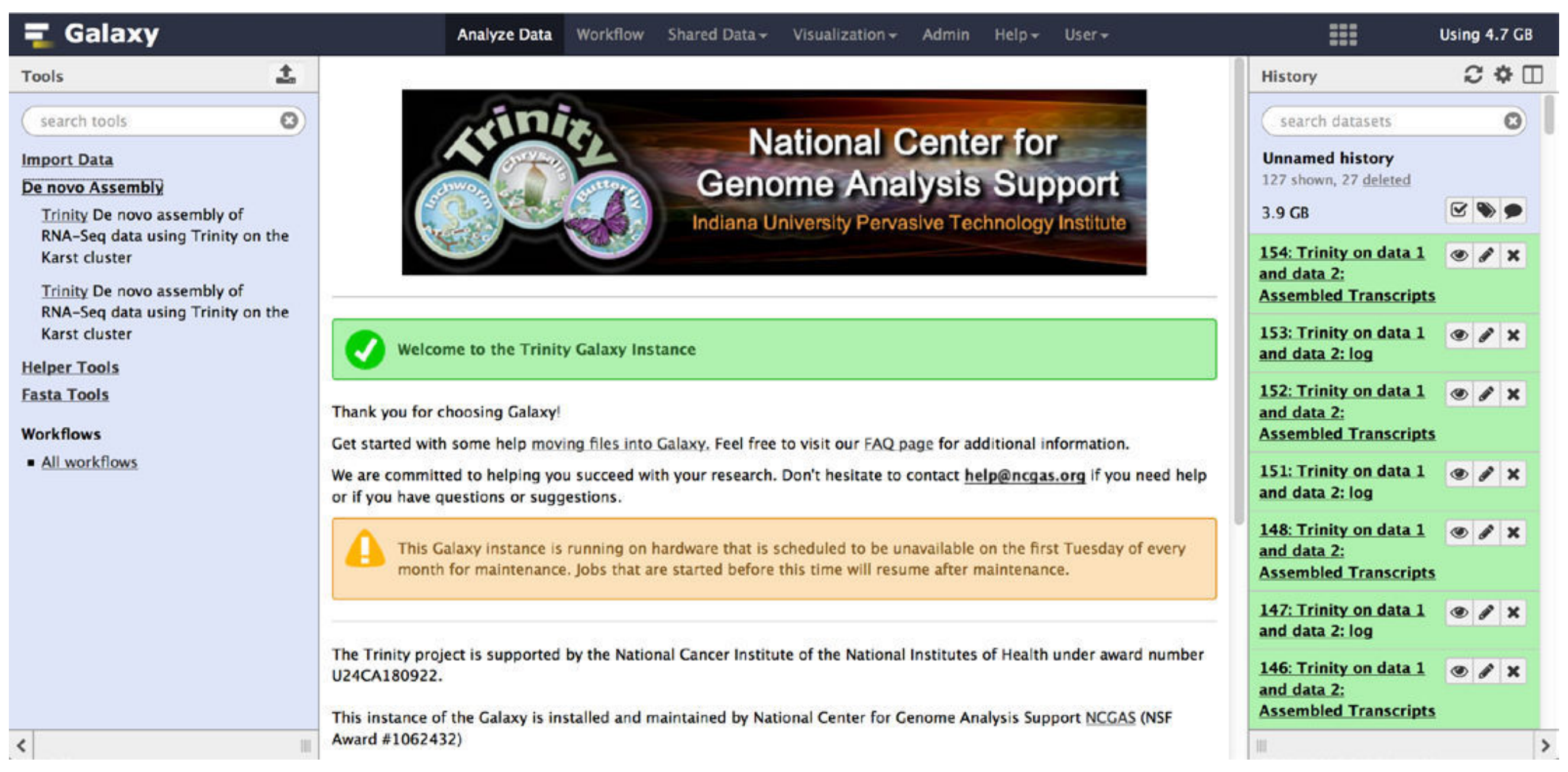
Chrysalis

At this step, alternative splicing is further explored - Inchworm transcripts that share regions are clustered and a De Bruijn graph is built for each cluster. The distribution of k-mers can inform decisions about clustering.

Butterfly

Often the most intense step, Butterfly resolves the De Bruijn graphs in parallel to decipher what reads belong to what isoforms or genes. This step builds a graph in memory for each thread and the underlying java software can allocate large amounts of memory.

The Trinity Galaxy Portal main screen: https://galaxy.ncgas-trinity.indiana.edu


Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Admin
Help
User
Using 4.7 GB
Tools
search tools
Import Data
De novo Assembly
Trinity De novo assembly of RNA-Seq data using Trinity on the Karst cluster
Trinity De novo assembly of RNA-Seq data using Trinity on the Karst cluster
Helper Tools
Fasta Tools
Workflows
All workflows
History
search datasets
Unnamed history
127 shown, 27 deleted
3.9 GB
154: Trinity on data 1 and data 2: Assembled Transcripts
153: Trinity on data 1 and data 2: log
152: Trinity on data 1 and data 2: Assembled Transcripts
151: Trinity on data 1 and data 2: log
148: Trinity on data 1 and data 2: Assembled Transcripts
147: Trinity on data 1 and data 2: log
146: Trinity on data 1 and data 2: Assembled Transcripts
National Center for Genome Analysis Support
Indiana University Pervasive Technology Institute

Welcome to the Trinity Galaxy Instance

Thank you for choosing Galaxy!

Get started with some help moving files into Galaxy. Feel free to visit our FAQ page for additional information.

We are committed to helping you succeed with your research. Don't hesitate to contact help@ncgas.org if you need help or if you have questions or suggestions.

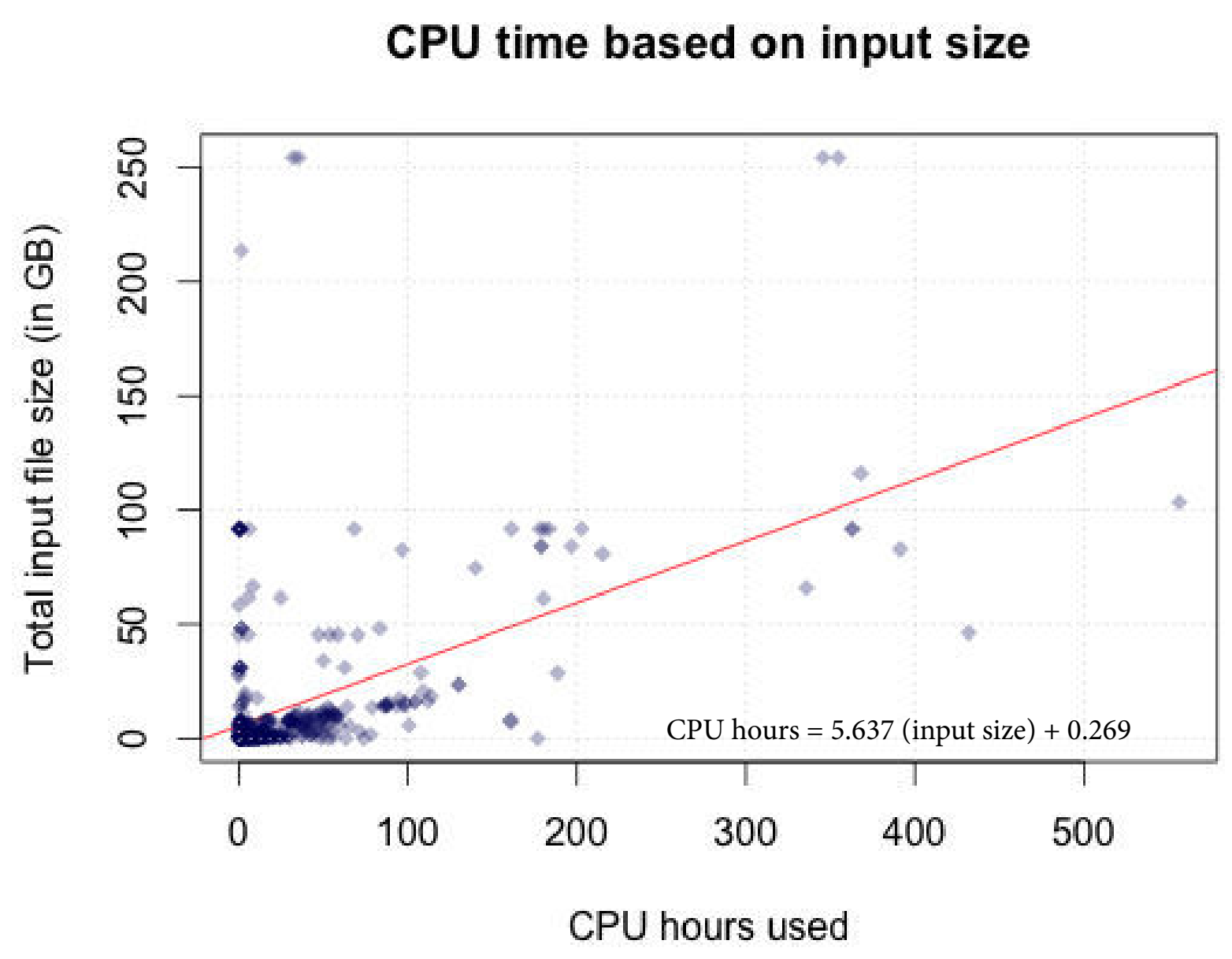
This Galaxy instance is running on hardware that is scheduled to be unavailable on the first Tuesday of every month for maintenance. Jobs that are started before this time will resume after maintenance.

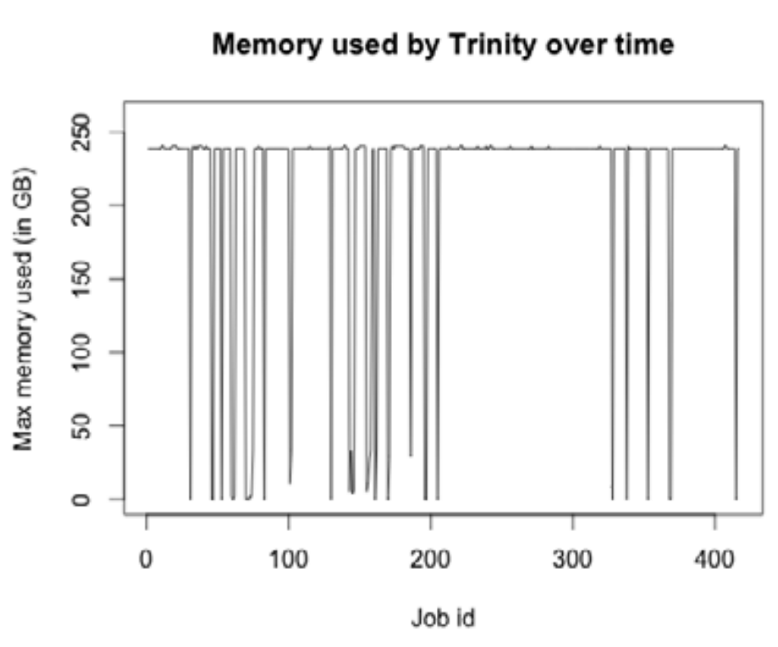
The Trinity project is supported by the National Cancer Institute of the National Institutes of Health under award number U24CA180922.

This instance of the Galaxy is installed and maintained by National Center for Genome Analysis Support NCGAS (NSF Award #1062432)

Benchmarks

The best way to inform decisions about setting up a Galaxy tool to consume certain resources is to see how the tool behaves on the system in question. In this case, 417 jobs were used as a small test of how the system behaves. Galaxy launched these Trinity jobs with 8 cores and 250GB of RAM:
The correlation is not as strongly linear as expected, based on previous results.



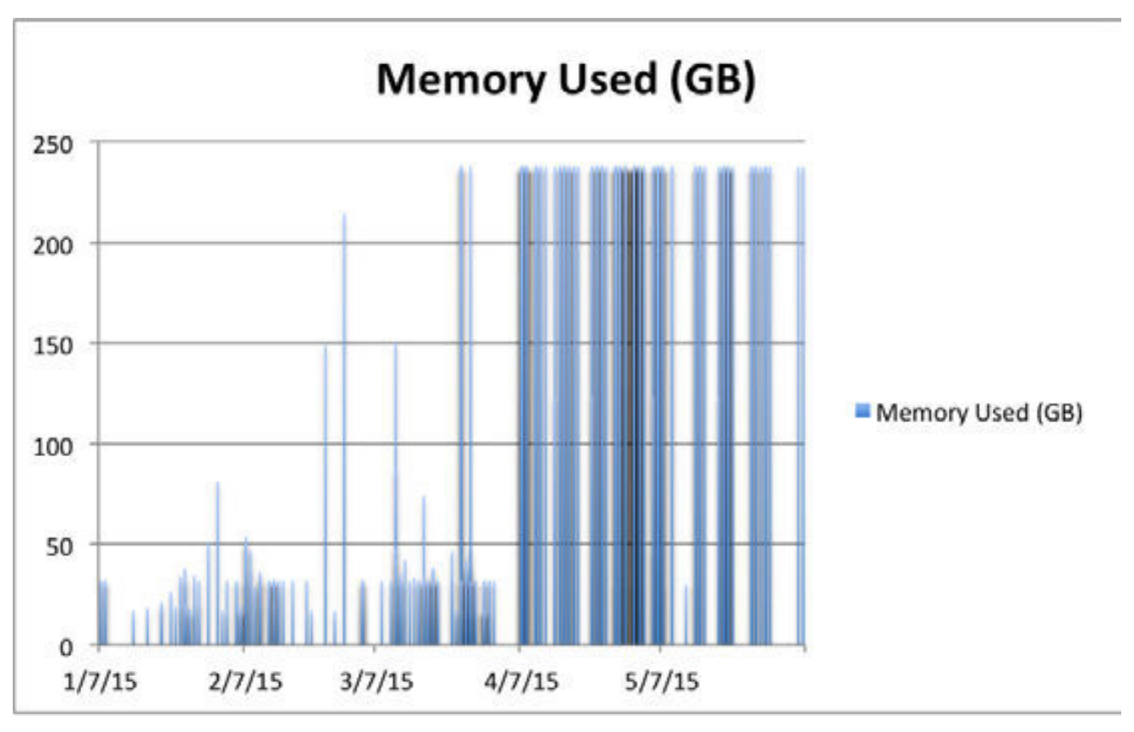


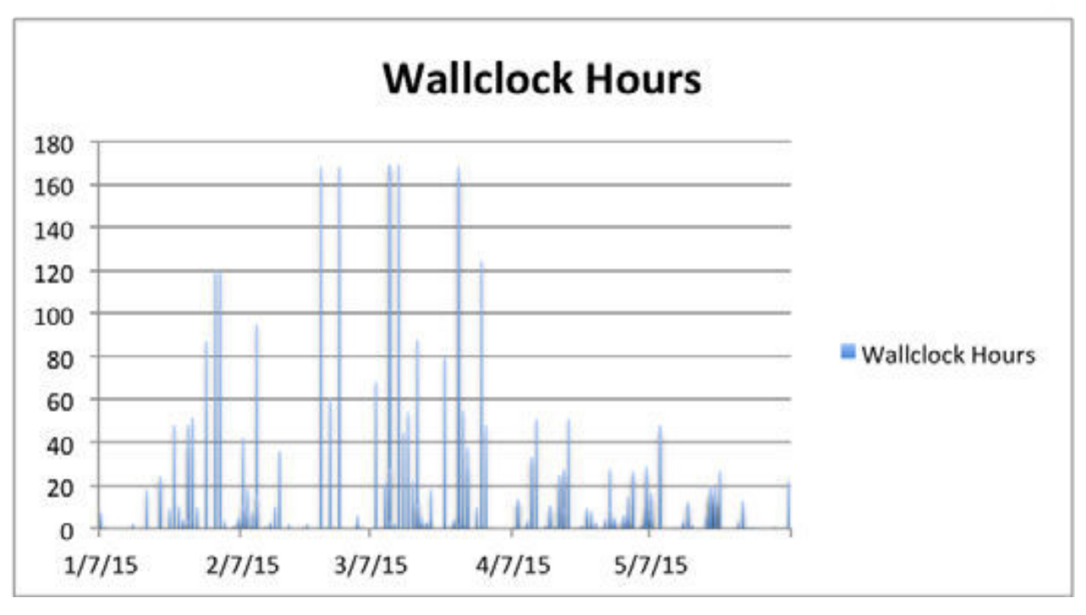
Note that the memory consumed in these runs is reported as a maximum of 250GB.

The lines on this graph represent where the Trinity tool did not use the full allotment of memory.

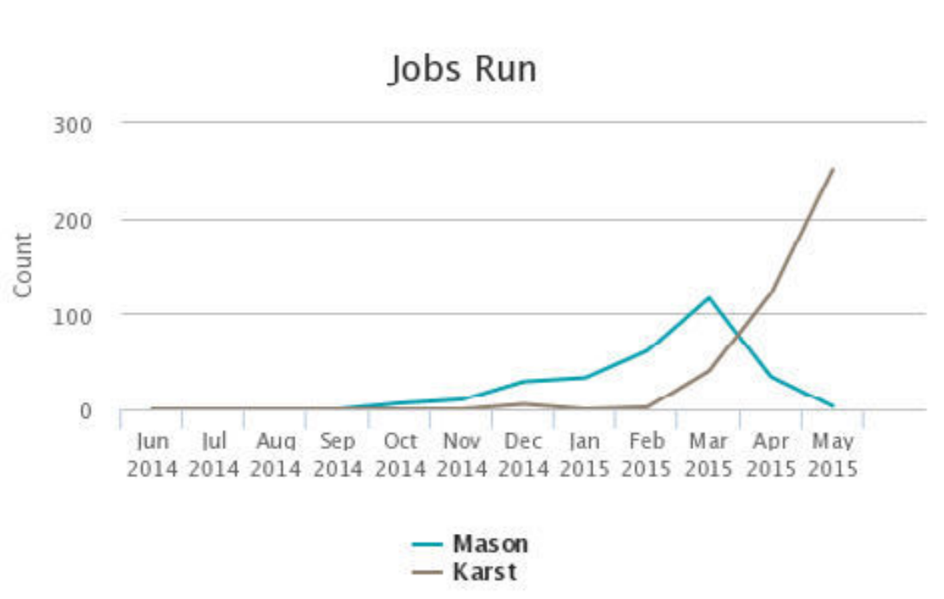
This may be due to higher volumes of testing the system, but there is also a change in implementation that may have contributed. At the end of March all jobs were required to use the --normalize_reads flag, which changes the performance of Trinity. The previous analysis was done using only successful jobs - the effect appears stronger if failed jobs are included.

The memory used and wall hours change noticeably after the end of March.





The final goal of this work is to optimize the way that jobs are launched through Galaxy to get the best throughput for Trinity assemblies. The number of users and jobs submitted is climbing:



With three nodes and a growing number of jobs, the only way forward is to make sure the cluster stays as utilized as possible. Next steps will involve varying the amount of memory given to a job, carefully tuning the number of cores, and more accurately guessing walltime based on the input sizes of the data.

This data was collected using Galaxy's logs, Torque's logs, and preliminary collectl data - more sophisticated methods benchmarking is a future plan.

PERVASIVE TECHNOLOGY INSTITUTE
INDIANA UNIVERSITY
PITTSBURGH SUPERCOMPUTING CENTER
TACC
TEXAS ADVANCED COMPUTING CENTER
BROAD INSTITUTE
ITCR Informatics Technology for Cancer Research
XSEDE
Extreme Science and Engineering Discovery Environment
Open Science Grid
NSF
NIH

References
[1] Bo Li, Nathanael Fillmore, Yongsheng Bai, Mike Collins, James A. Thompson, Ron Stewart, Colin N. Dewey. Evaluation of de novo transcriptome assemblies from RNA-Seq data.
[2] Marçais, G. and Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics, 27(6):764-770.
[3] Goecks, J, Nekrutenko, A, Taylor, J and The Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol. 2010 Aug 25;11(8):R86.
[4] Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, Nekrutenko A, Taylor J. "Galaxy: a web-based genome analysis tool for experimentalists". Current Protocols in Molecular Biology. 2010 Jan; Chapter 19:Unit 19.10.1-21.
[5] Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J, Miller W, Kent WJ, Nekrutenko A. "Galaxy: a platform for interactive large-scale genome analysis." Genome Research. 2005 Oct; 15(10):1451-5.
[6] Guillaume Marcais and Carl Kingsford, A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics (2011) 27(6): 764-770 (first published online January 7, 2011) doi:10.1093/bioinformatics/btr011