

Developing Sequence Analysis Pipelines for Cores

April 22, 2020

Adrian Reich, PhD

■ My background


- RNA-seq (count, splice)
- DNA-seq (genome, CNV)
- xIP-seq (RIP, ChIP, CLIP)
- Small RNA (mi, pi)
- Amplicon
- *de novo* assembly (DNA, RNA)
- Single cell (manual, 10X)
- Illumina
- IonTorrent
- Nanopore
- BioNano Genomics
- GenapSys
- 454
- SOLiD
- Heliscope

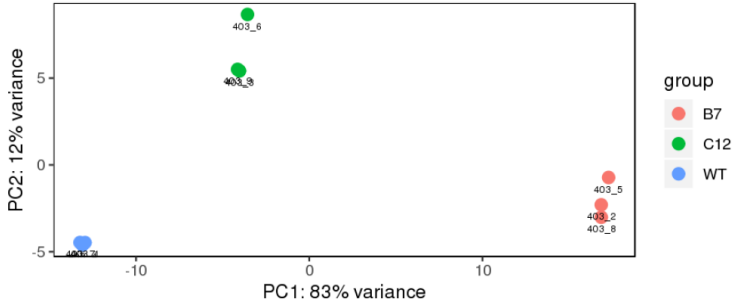
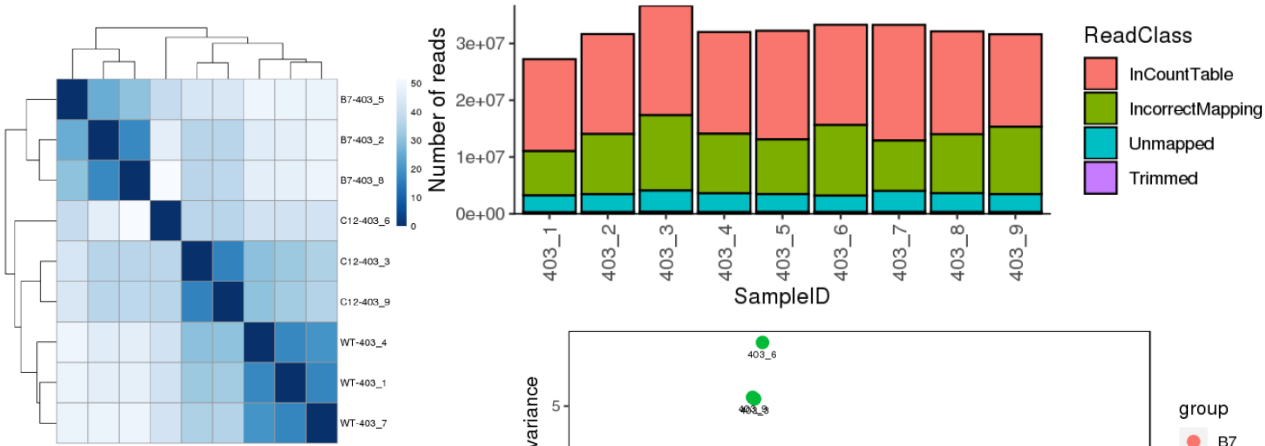
■ **Client service focus**

- **Support any project from the Genomics core (or outside)**
- **RNA-seq**
- **Amplicon sequencing**
- **ChIP-seq**
- **Single cell sequencing (10x)**
- **miRNA**

What does that entail?

```
@NS500704:697:HNTNCBGXC:1:11101:13432:1050 1:N:0:AGGCAGAA+NTTAGACG
TTTTANAAGATCGCCTTCAAATTATTTAATCACCTACAACCTTTAAACTAACTTTAAGCTGTTTAAGTCACCTT
+
AAAAA#AEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
@NS500704:697:HNTNCBGXC:1:11101:7509:1051 1:N:0:AGGCAGAA+NTTAGACG
CTGTANTTGAATACATCAGTCTTCAGCTGTGGTCTGCTGCACACGCCTCTTCCCTCACTACCTCTGGAGCACTC
+
AAAAA#EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
@NS500704:697:HNTNCBGXC:1:11101:8225:1052 1:N:0:AGGCAGAA+NTTAGACG
CAGTACGGAGGCGTCTTACAGCTCTCTTGTCTCACTGATGTCCTTCTTATGCTTGCCTTAAACTCAGCAATAAA
+
AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
@NS500704:697:HNTNCBGXC:1:11101:19019:1053 1:N:0:AGGCAGAA+NTTAGACG
GTCCTGCTATTGCATTATCATCTCAAGCTGTCACTCCAAGGGCCACCAGAAGCGCAAGGCCCTCAAGACCACA
+
AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
@NS500704:697:HNTNCBGXC:1:11101:4490:1054 1:N:0:AGGCAGAA+NTTAGACG
ATCATTATCTTAGCCACCAGAACACAGAATGTTCTTGGGAGAAGGGCCGGCGGATTTCGGGAAGTACTGCTGTAG
+
66AAEEEEEEEEEEAE/EEEA6E6EEEEEEAEAE/EE/EEE/AEEE6EEEEEEEEEEEEEEEEEEEE/EEE
@NS500704:697:HNTNCBGXC:1:11101:14432:1054 1:N:0:AGGCAGAA+NTTAGACG
GTGCAGGTGGCAGATTTTAACTATACTGAGTCAACGGTGCAGGTAACCACCCTCGGGCCAGTCTAGAGTAG
+
66AAAE6EEAEAEAE/EEEEEEEEEE/A/A/E//EEEEEEEEEEEEEEEEEEEEEEEE/6E<<EEEEEEA
```

 Scripps Research Scripps Florida 130 Scripps Way Jupiter, FL 33458 Scripps Florida Center for Computational Biology & Bioinformatics		Report	
		RNA-seq	
		Authors: Adrian Reich, PhD	
		Project number: Exp. 403; fl_ne_181_002	Date: March 6, 2020
		Client:	



Experimental	Control	Count of DEGs with p -value < 0.05	DEGs with adjusted p -value < 0.05
C12	B7	4213	2508
B7	WT	6002	4231
C12	WT	4619	2546

■ Pipelines for cores instead of labs

- **Labs – get data, research necessary tools, analyze, publish, repeat with new data and/or new tools**
 - Focus on single organism and can change tools between projects
 - Can have grad student or postdoc squeeze all the data from the project over 1-6 months
- **Cores – research client needs, research necessary tools, get data, analyze, repeat with new data and same tools**
 - Need to turn data around in less than 1 week for standard projects
 - Support everything, forever (tools, organisms, and sequencing platforms), but also add new functionalities



■ preprocess

- **How it's built**
- How it's executed
- Final output and data retention plan
- Resource usage
- Optimizations and tricks

■ How it's built

- The preprocess pipeline is an analysis framework that is designed to support most sequencing projects that come off an NGS sequencing platform
- Currently it supports most RNA sequencing applications and some ChIP projects
- Built as shell script that spawns and executes python and R scripts as needed as well as open source binaries

```
#!/bin/bash
```

```
VERSION=3.3.0  #preprocess version February 27, 2020  commit notes: force genome load hack for STAR, fix multi round trimming logic  
AUTHOR="Adrian Reich"
```

■ **How it's built**

- **Preprocess has a total of 25+ options, 3 are mandatory**
- **The pipeline requires at least 3 options and a specific directory structure**
- **The directory structure is easily generated by copying data from BaseSpace or from a single directory of reads using support scripts**

■ How it's built

- | | | |
|----------------------------|---|-----------------------------------|
| • 3 mandatory options | → | • Concatenation and trimming |
| • 3 mandatory + 1 optional | → | • ... and mapping |
| • 3 mandatory + 2 optional | → | • ... and gene counts |
| • 3 mandatory + 3 optional | → | • ... and differential expression |
| • 3 mandatory + 4 optional | → | • ... and sam or bam files |
| • 3 mandatory + 5 optional | → | • ... and bigwig files |
| • 3 mandatory + 6 optional | → | • ... and final report |
| • 3 mandatory + 7 optional | → | • ... change reference genome |
| • 3 mandatory + 8 optional | → | • ... change trimming tool |
| • 3 mandatory + 9 optional | → | • ... and exon usage differences |

■ preprocess

- How it's built
- **How it's executed**
- Final output and data retention plan
- Resource usage
- Optimizations and tricks

How it's executed

```
#!/bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l walltime=01:00:00
#PBS -l cput=8:00:00
#PBS -l mem=64gb
#PBS -q flits
#PBS -m bea
#PBS -M areich@scripps.edu

echo -n 'Begin '$PBS_JOBID' job at: ' && date

DIR=/gpfs/group/flits/areich/Exp999

/gpfs/home/areich/scripts/preprocess --in /gpfs/group/flits/ProjectData/Workshop/temp --out $DIR/preprocess_Exp999_new --kit NEBNext_Ultra_Direct_RNA_SI --species mouse --map star --count htseq --diff_expr deseq2 --exon_diff dexseq --final_report rnaseq --bigwig yes

echo -n 'Finish '$PBS_JOBID' job at: ' && date

exit

SAMPLE_SHEET_START
SampleID,Notes
999_1,Adult_double
999_2,Adult_double
999_3,Adult_double
999_4,Adult_double
999_5,Adult_flox
999_6,Adult_flox
999_7,Adult_flox
999_8,Adult_flox
999_9,Week2_double
999_10,Week2_double
999_11,Week2_double
999_12,Week2_double
999_13,Week2_flox
999_14,Week2_flox
999_15,Week2_flox
999_16,Week2_flox
SAMPLE_SHEET_END
```

1. Submit initial job
2. Wait 5-10 minutes

How it's executed

```
=====
Clobbering of data is in effect! Remove for production!
=====

The tool output from the initial test run can be found here: /home/areich/data/preprocess/TestOut/preprocessTest/preprocess.Tool_Output.txt
The initial report and final SLURM script can be found here: /home/areich/data/preprocess/TestOut/ReportAndSLURM.preprocess.array

Command used for the trimming of the test sample:
=====
module: loading 'trimmomatic/0.32'
trimmomatic PE -threads 8 -phred33 -trimlog /home/areich/data/preprocess/TestOut/preprocessTest/preprocess.Tool_Output.txt /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_R1.fastq.gz /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_R2.fastq.gz /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_R1.pair.trim.fastq.gz /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_R1.unpair.trim.fastq.gz /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_R2.pair.trim.fastq.gz /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_R2.unpair.trim.fastq.gz ILLUMINACLIP:/home/areich/data/preprocess/TestOut/Adapter.Nextera.fasta:2:30:10 LEADING:5 TRAILING:5 SLIDINGWINDOW:4:15 MINLEN:36
=====

Command used for the mapping of the test sample:
=====
module: loading 'STAR/2.4.0e'
STAR --genomeDir /media/HAL-FS-8B/preprocessDependencies/human/star/GENCODE/Overhang49 --genomeLoad NoSharedMemory --runThreadN 8 --readFilesCommand zcat --readFilesIn /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_R1.pair.trim.fastq.gz /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_R2.pair.trim.fastq.gz --outFileNamePrefix /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_Star
=====

Command used for the counting of the test sample:
=====
module: loading 'python/2.6.6'
module: loading 'htseq-count/0.6.1'
htseq-count --mode=union --stranded=no --idattr=gene_id /home/areich/data/preprocess/TestOut/preprocessTest/SubsetTest_StarAligned.out.sam /media/HAL-FS-8B/preprocessDependencies/human/star/GENCODE/Overhang49/Annotation.gtf
100000 GFF lines processed.
200000 GFF lines processed.
300000 GFF lines processed.
400000 GFF lines processed.
500000 GFF lines processed.
600000 GFF lines processed.
```

1. Submit initial job
2. Wait 5-10 minutes
3. Review output from initial run
4. Review report
5. Submit report to garibaldi to process all samples and all data

How it's executed

```
#!/bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l walltime=36:00:00
#PBS -l cput=576:00:00
#PBS -l mem=128gb
#PBS -q flits
#PBS -t 1-4
#PBS -m bea
#PBS -M areich@scripps.edu
#=====
#preprocess Report
#
#
#The preprocessing script should complete successfully.
#Below is some summary information for you to review from the preprocess test.
#The test used a single sample prior but should apply to all of your samples.
#
#Version used of preprocess pipeline:      3.1.6
#
#Input directory for this project:         /gpfs/group/flits/ProjectData/Workshop/temp
#Output directory for this project:        /gpfs/group/flits/areich/Exp999/preprocess_Exp999_new
#Number of samples in this project:        16
#
#Random sample used for testing:           Sample_999_16
#Location of the output from testing:       /gpfs/group/flits/areich/Exp999/preprocess_Exp999_new/preprocessTest/preproces
s.Tool_Output.txt
#Location of this file:                    /gpfs/group/flits/areich/Exp999/preprocess_Exp999_new/ReportAndTORQUE.preproce
ss.array
#
#Tool used to trim reads:                  cutadapt version 1.18
#Forward read - 3'
#AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC
#Reverse read - 3'
#AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT
#Tool used for mapping:                   star version 2.5.2a
#Species used for mapping:                mouse
#Reference used for mapping:              ENSEMBL
#Basic information on genome:             mouse-ENSEMBL-grcm38.r91 : M.musculus-ENSEMBL-GRCm38.r91 : downloaded February
16, 2018
#Source of rRNA and mito contaminants:    mouse-UCSC : M.musculus-UCSC-mm10 : downloaded March 23, 2016
#Tool used to count gene features:         python version 2.7.11, and htseq version 0.11.0.
#Tool used to test for diff gene expr:     R version 3.5.1, and dseq2 version 1.20.0
#Tool used to test for diff exon usage:    R version 3.5.1, and dexseq version 1.26.0
#
#Kit used to prepare the samples:          NEBNext_Ultra_Direct_RNA_SI
#Source nucleotide:                       RNA
#Was it stranded sequencing:              Yes
#Length of a single read:                 40
#Paired or single reads:                  Paired
#
#Number of sampled reads:                  100000
#Number(%) of post-trim reads:             96078 (96.1%)
#Number(%) of uniquely mapped reads:       84138 (87.57%)
#Number(%) of reads in count table:        74274 (77.30%)
#Number(%) of reads excluded in count table: 18464 (19.21%)
#Number(%) of contaminating rRNA reads:    193 (0.20%)
```

- Analysis flat file reports which sample was randomly chosen for test run
- The adapters used for trimming
- Mapping tool and reference
- Most importantly allows the core to take the raw reads and this file to re-generate the completed analyses and final report

■ Mandatory options

- **--in**
 - Directory with reads in proper directory format
- **--out**
 - Directory where all analyses, results, and intermediate files are saved
- **--kit**
 - Necessary for trimming of adapters and allows preprocess to infer some run parameters
 - Is the sample DNA, RNA, or smallRNA?
 - Is the sample stranded?

■ Optional options

- **--map**
 - Identify the tool used for mapping
 - Currently: STAR, bowtie, bowtie2
- **--ref or --species**
 - If **--map** is specified, then one of these must also be specified
 - There are eight genomes currently available with pre-built indices: human, mouse, naked mole rat, seahare, zebrafish, rat, fly, and e. coli
 - The **--species** parameter is a shortcut in that it will select the default genome for that species
 - Current defaults:
 - ENSEMBL builds
 - Recent vs. old

■ Optional options

- **--count**
 - Can specify HTSeq to count features
 - Current default is “gene_id”
- **--final_report**
 - Specify if you want a final report generated, option currently include rna_seq and chip_seq
- **--diff_expr**
 - Do you want to run differential expression analysis for RNA-seq experiment using DESeq2, or macs2

■ Optional options

- **--exon_diff**
 - Run a differential exon usage analysis using DEXSeq
- **--keep_sams**
 - Optionally keep sam or bam files (not kept by default)
- **Other options are available**
 - Manual trimming including additional trimming of ends
 - Isoform analysis
 - Change trimming tool
 - Specify contamination source database

■ Options build-out

- **How is trimming done?**

- From **--kit**, 4 variables are specified:
 - ADAP_FILE_NAME = adapters used for cutadapt or path to adapter file
 - ADAP_FILE = nextera, small, HT, or LT
 - NUCLEOTIDE = DNA, RNA, or smallRNA
 - STRANDED = yes, no, or reverse
- Single vs. Paired end reads are detected automatically by script

■ Options build-out

- **How is genome specified?**
 - From **--species** or **--ref**, 3 variables are specified:
 - REFERENCE = GENCODE, ENSEMBL, or UCSC
 - SPECIES = mouse, rat, ecoli, fly, human, etc.
 - GENOME_INFO = genome annotations
 - From **--map**, 1 variable is specified:
 - MAPTOOL = star, bowtie, bowtie2
 - If **--combo** is also specified:
 - SPECIES = “combo-<species>-<combo input>”

■ **Execute analysis flat file**

- **Uses all the same options as the initial preprocess**
- **Submits array with requested nodes**
- **Remaining jobs go into queue if queue is full**
- **Last node to finish does all the wrap up work**
 - Final differential expression
 - Final exon usage
 - Final isoform splicing
 - Final report


■ preprocess

- How it's built
- How it's executed
- **Final output and data retention plan**
- Resource usage
- Optimizations and tricks

Data archiving

- Only the raw reads, final report and analysis flat file are retained
- All software versions and genome version and download date are saved in analysis flat file
- All versions of preprocess software are archived on Scripps git server

Adrian Reich > preprocess > Details


 preprocess




Star 0 Fork 0 HTTPS https://unknown.fl.ad.scripps.e


Files (266 KB) Commits (41) Branch (1) Tags (0) Readme

Add Changelog Add License Add Contribution guide Add Kubernetes cluster Set up CI/CD

master preprocess / History Find file

 Update README.md
Adrian Reich authored about a minute ago 585bdc85

Name	Last commit	Last update
 README.md	Update README.md	about a minute ago
 preprocess	v3.0.0 - Huge overdue update including DEXSeq exon usa...	2 weeks ago
 preprocess.1.html	v2.3.4 - fix sample sheet option and requirements, re-do ...	10 months ago

 README.md

Running the Scripps Florida NGS Analysis Pipeline

This document lays out everything you need to run the Scripps processing pipeline from data transfer to report generation.

Documentation

preprocess

[NAME](#)
[SYNOPSIS](#)
[DESCRIPTION](#)
[OPTIONS](#)
[SEE ALSO](#)
[BUGS](#)
[AUTHOR](#)

NAME

preprocess – Generate TORQUE script for Illumina read processing

SYNOPSIS

preprocess -i *DIRECTORY* -o *DIRECTORY* --kit *TYPE* [*OPTIONS*]

DESCRIPTION

Generate a TORQUE script in order to process reads from an Illumina machine by filtering out low quality reads, trimming adapters, and optionally mapping reads to a reference and counting gene features. The script will also test if the specified pipeline will successfully execute when submitted to garibaldi and estimate various run parameters for the pipeline. All estimations are documented in a simple report that also serves as the TORQUE script to be submitted to garibaldi after it has been carefully reviewed.

The in (-i, --in), out (-o, --out), and sample kit (--kit) flags are required in order to generate the TORQUE script and estimate run parameters. If the out directory does not yet exist, it will be created.

Additional options are available to map the reads against a reference with a specified tool.

OPTIONS

Mandatory

-i, --in

The absolute path to the directory containing all of the read files as delivered by Genomics. Should be in the format: Project_NAME

-o, --out

The absolute path to the directory where all results will be output including the report/TORQUE script and all intermediate files.

--kit

Nextera|*tsDNA_HT*|*tsDNA_LT*|*tsDNA_ChIP*|*tsDNA_PCRfree*|*tsRNA_LT*|*tsRNA_Strand_HT*|*tsRNA_Strand_LT*|*tsRNA_Small*|*tsRNA_Access*|*ctSMART_Nextera*|*ctSMART_tsRNA_HT*|*ctPico_Mammal*|*NuGEN_univ_RNAseq*|*NuGEN_SoLo_RNAseq*|*NEBNext_Ultra_Direct_RNA_DI*|*NEI*
The abbreviation 'ts', 'ct', 'HT', 'LT', 'SI', and 'DI' stand for 'TruSeq', 'Clontech' 'High Throughput', 'Low Throughput', 'Single Index', and 'Dual Index', respectively. The sample kit used to prepare the libraries as reported by Genomics. If they report a kit that you do not see here, please inform the author so that it can be added or use the options --manual_strand, --manual_kit, and --manual_nucleotide. If the option "no_trim" is specified then no trimming or quality filtering of the reads will be done. Please note that the entries do not have spaces, but have underscores.

Optional

--trim *cutadapt*|*trimmomatic*

Specify the trimming tool to use. Cutadapt is chosen by default.

--map *star*|*bowtie*|*bowtie2*|*tophat*|*tophat2*|*bwa*

To map the reads, a tool must be specified. The difference between **tophat** or **tophat2** is the use of bowtie or bowtie2 for mapping the reads, respectively. If a tool is not listed, please contact the author so it can be added. If --map is specified, --ref or --species must also be used.

■ Report output comparison

■ preprocess

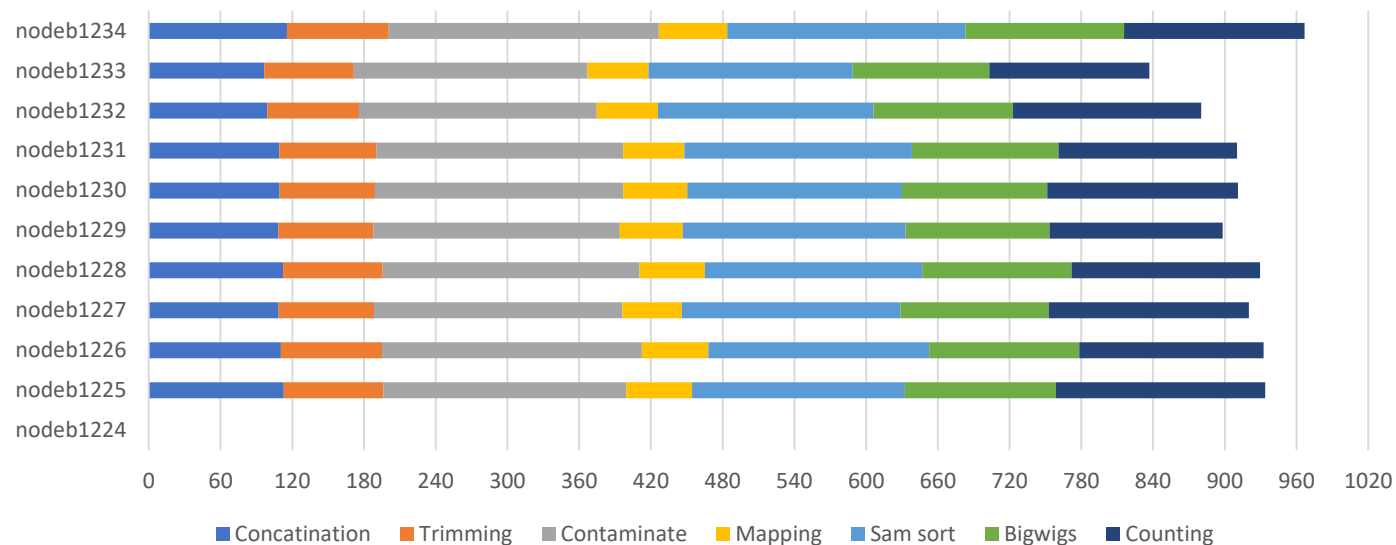
- How it's built
- How it's executed
- Final output and data retention plan
- **Resource usage**
- Optimizations and tricks

■ Resource calculation

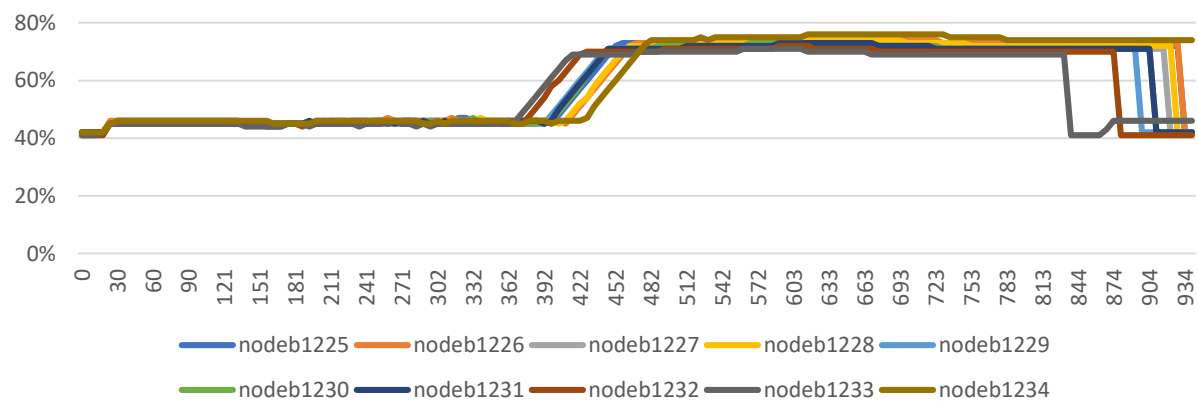
- **Total number of samples counted**
- **Determine how many 16-core nodes are needed for 4 samples per node**
- **Reset number to 11 nodes if needed (size of flits queue)**
- **Divide all remaining samples across available nodes**
- **Standard analyses typically finish in 2-8 hours from initial submission to final report generation**

Resource usage

Pipeline time per stage for TGI mega project (320 samples)



Data saved to scratch drives



■ preprocess

- How it's built
- How it's executed
- Final output and data retention plan
- Resource usage
- **Optimizations and tricks**

■ Optimizations and tricks

- **Always treat the sysadmins nicely**
- **Identify the bottleneck on the software side AND the hardware side**
- **Use torque arrays**

More questions?