Towards Large-scale Genomics, Transcriptomics, and Metagenomics for All

Philip Blood
Pittsburgh Supercomputing Center
blood@psc.edu
Genomics, Metagenomics, Transcriptomics

- All DNA in a cell is the genome
- DNA is transcribed into corresponding molecules of RNA
- The transcriptome is all of the RNA transcripts of a particular cell

Courtesy Lenore Pipes, Cornell
The obligatory genomics big data graph

https://www.genome.gov/images/content/costpergenome2015_4.jpg
Facilitating Genomics

• Substantial and growing need for bioinformatics help
• Various resources available: XSEDE, CyVerse, NCGAS, Galaxy, GenePattern, etc…
• Key Challenges:
  – Data wrangling
  – Choosing best software tools/ bioinformatics know-how
  – Utilizing HPC systems needed for large analyses
  – Reproducibility
Installing Genomics Software!

hmmer/3.1b2 prokka/2.6.3 abyss/1.9.0 htseq/0.6.1 icc/16.0.2) python/2.7.11_gcc
annovar/2016.02.01 idba-tran/1.1.1 ansys/17.1 idba-tran/1.1.1_long anvio/2.0.2 idba-ud/1.1.1
augustus/3.2.2 bamtools/2.4.0 bcf tools/0.1.19 R/3.3.1-mkl bcf tools/1.3.1
kallisto/0.43.0 raxml/8.2.9 bedops/2.4.19 ray/2.3.1 bedtools/2.25.0 repeatmasker/4.0.6
blasr/1.3.1 rnammer/1.2 blast/2.2.31 rsem/1.2.21 blat/v35 macs/1.4.3
sailfish/0.9.2 macse/1.2 salmon/0.6.0 mafft/7.300 salmon/0.7.2 samtools/0.1.19
samtools/1.3 melt/0.3.8 bowtie/1.1.2 masurca/3.1.3 bowtie2/2.2.7 matlab/MCR_R2013a busco/1.22
scythe/0.981 bwa/0.7.13 seqtk/1.2-r94 maxbin/2.1.1 sickle/1.33 megan/5.11.3
kanu/1.3 signalp/4.1c cdfasta/2013 mothur/1.38.1 cd-hit/2016.06.21
snvmix/0.11.8-r5
soapdenovo2/2015-10-09 somaticsniper/1.0.5 spades/3.8.1 sra-toolkit/2.5.7 cufflinks/2.2.1
strelka/1.0.14 dammit/0.3 mummer/3.23 deeptools/2.3.5 detonate/1.10
diamond/0.7.11 discovar/52488 discovardenovo/52488 ectools/2014-12-01
ngscheckmate/2016.10.12 tmhmm/2.0c embo ss/6.6.0 tophat/2.1.1 trimmomatic/0.36 falcon/0.4.1
openslide/3.4.1 trinity/2.0.6 fasta-splitter/0.2.4 paml/4.9a trinity/2.1.1 fastq/0.11.3
trinity/2.2.0 fastq-splitter/0.1.2 trinotate/2.0.2fastx/0.0.14 trinotate_db/2.0
trinotate_db/2.0_pylon1 pbjelly/15.8.24 flash/1.2.11 perl/5.18.4-threads unceqr/2016-07-08
fraggenescan/1.20 varsca n/2.4.2 vcf tools/0.1.15 gatk/3.5 velvet/1.2.10-maxk63-big
gatk/3.6 velvet/1.2.10-maxk63-categ14-big phylosift/1.0.1 pindad/2.1.1
pilon/1.16
platanus/1.2.4 wgs/8.2 genome-music/0.4.1 plinkseq/0.10 wgs/8.3rc xhmm/1.0
primer3/1.1.4 primer3/2.2.3 hisat2/2.0.4 primer3/2.3.7 h mmer/2.3.2
prodigal/2.6.2
A Different Flavor of XSEDE ECSS

• Generally not interested in intensive optimization of a single code -- there are too many, and constantly changing!
  – Some major codes have been addressed through ECSS Community Codes (e.g. Trinity)

• Generally want to know:
  – What tools are best for the job?
  – Where can I run them?
  – How do I run them?
  – How do I write proposals for allocations on XSEDE?

• Engaging users through:
  – ECSS Novel and Innovative Projects
  – “Light” ESRT projects
Genomics Community: Well-established in XSEDE?

Registered Users on Galaxy Main vs. Users on XSEDE
Genome Assembly Using Next-generation Sequencing

- Reconstruct genomes of millions to billions of nucleotide base pairs (bp)
- ...containing repetitive sequences thousands of bp long
- ...using random 100-250 bp fragments (reads)*
- ...which have systematic and random errors
- Doing this reliably requires deep coverage and often large shared memory

*Bigger fragments can be generated, but at higher cost

Kinghorn Centre for Clinical Genomics, Garvan Institute of Medical Research. Image credit: P. Morris/Garvan
N50 size

Def: 50% of the genome is in contigs as large as the N50 value

Example: 1 Mbp genome

N50 size = 30 kbp
(300k + 100k + 45k + 45k + 30k = 520k >= 500kbp)

Note:
A “good” N50 size is a moving target relative to other recent publications. 10-20kbp contig N50 is currently a typical value for most “simple” genomes.

Slide courtesy of Michael Schatz, CSHL
Improved SNP Detection in Metagenomic Populations
Ping Ma and Xin Xing, University of Georgia

• **Goal:** Develop statistical method that can distinguish closely related, unknown species in a metagenomic sample
• **Problem:** Didn’t know where to start with analyzing large metagenomic data sets

★ **ECSS Support**
  – Identified Ray MPI-based genome assembler capable of assembling large metagenomics data sets
  – Tested Ray on *Bridges* to guide user on core counts for massive metagenome assemblies
  – Helped user parallelize their R-based tool (MetaGen) on *Bridges*
  – Helped user distribute data parallel jobs across many Bridges nodes using Slurm

Environmental Shotgun Sequencing (ESS). (A) Sampling from habitat; (B) filtering particles, typically by size; (C) DNA extraction and lysis; (D) cloning and library; (E) sequence the clones; (F) sequence assembly. By John C. Wooley, Adam Godzik, Iddo Friedberg - http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.1000667, CC BY 2.5, https://commons.wikimedia.org/w/index.php?curid=17664682
Improved SNP Detection in Metagenomic Populations
Ping Ma and Xin Xing, University of Georgia

- Assembled 900 Gigabase pairs (Gbp) of gut microbial DNA from normal and diseased patients
  - Created metagenomes for both type-2 diabetes (T2D) and inflammatory bowel disease (IBD) patients
  - Identified important pathogenic or missing probiotic species
  - Massive metagenome assemblies took only hours using an MPI-based metagenome assembler, Ray, on dozens of Bridges RM nodes connected by Omnipath

- Improved characterization of composition of human gut microbiome
  - First use of unsupervised binning method provides more accurate estimate of number of microbial species (~2000)
  - Estimated abundance of known vs. unknown species
  - MetaGen binning software ran across 10 RM nodes, clustering 500,000 contiguous sequences in only 24 hours

- Identified important pathogenic or missing probiotic species in diseased patients
  - Eight pathogenic microbial species identified in IBD patients
  - Two probiotic strains identified with lower abundance in T2D patients
  - Working to predict genes from unknown microbial species and identify genes related to disease conditions
Factors affecting quality of transcriptome analysis

Raminder Singh
Indiana University

Jordi Abante, Noushin Ghaaffari and Charles D. Johnson
Texas A&M
What can the transcriptome tell us?

- The transcriptome shows when and where each gene is turned off or on in the cells and tissues of an organism.
- Counting the number of transcripts for a given gene can determine gene expression.

Courtesy Lenore Pipes, Cornell
SEQC RNA-Seq Data

• Sequence Quality Control (SEQC) Consortium
• Six sites generated RNA-Seq data from well-studied human samples
• Coordinated by US Food and Drug Administration
• Sample A: Ten pooled cancer cell lines
• Sample B: Multiple brain regions from 23 donors
Goal: Determine Best Practices of Transcriptome Assembly

- Use a well-defined, well-understood standard data set
- Use a state-of-the-art transcriptome assembly and quality assessment pipeline
- Explore factors that influence quality of assembly
- This work is ongoing
Factors Influencing Assembly Quality

• Some factors to consider:
  – Sample preparation sites/methods (Aspects 1 & 3)
  – Sequencing depths (Aspect 2)
  – Preprocessing measures (Aspect 4)
  – Potentially many others

• So far have run 2 studies to completion:
  – Aspect 1: Compare outcomes across all six SEQC sites (site-effect)
  – Aspect 3: Compare sequencing library quality of SEQC sites vs. vendor-prepared library (library-effect)
Workflow

- Pre-processing
  - Remove sequencing adapters, repetitive “tails”, rRNA and mitochondrial RNA, read error correction
- Assembly
  - Trinity transcriptome assembly pipeline
- Assembly Post Postprocessing and Quality Assessment
  - Statistical measures: N50
  - Assess completeness of transcriptome gene content
  - Map/Compare assembled transcriptome and original reads to reference genome
- **Problem**: Wrangling massive data sets through complex pipeline
Massive Transcriptome Assemblies with Trinity

Challenges:

- Run Trinity pipeline on unprecedented data set
- Optimized for Blacklight/Greenfield/Bridges

1.8 Billion RNA-seq reads from one primate species

- Remove adapters
- Quality filter
- Remove Poly A/T
- Remove mtDNA, rRNA
- Convert to fasta (performed in-house)

~1.4 Billion RNA-seq reads

Inchworm
Typical Run time: 100 hours
Cores used: 64

Chrysalis (Run on RAM disk)
Typical Run time: 400 hours
Cores used: 128 cores

Quantify Graph & Butterfly (Run on RAM disk)
Typical run time: 50 hours
Cores used: 64 cores

Trinity + files on RAM disk required
~1 TB RAM

On Bridges, use local disk

5x faster on RAM/local disk
Tools used at different stages of pipeline

### Pre-Processing

<table>
<thead>
<tr>
<th>Process/Tool</th>
<th>Purpose</th>
<th>CPU</th>
<th>Memory</th>
<th>Time (min.)</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cutadapt</td>
<td>Get rid of adapters.</td>
<td>16 CPUs</td>
<td>8 GB / CPU</td>
<td>50</td>
<td>python/2.7.9</td>
</tr>
<tr>
<td>Flexbar</td>
<td>Get rid of poly A/T tails.</td>
<td>15 CPUs</td>
<td>8 GB / CPU</td>
<td>90</td>
<td>samtools/1.2, flexbar/2.5</td>
</tr>
<tr>
<td>Bowtie</td>
<td>Remove reads coming from rRNA and chrM.</td>
<td>15 CPUs</td>
<td>8 GB / CPU</td>
<td>90</td>
<td>samtools/1.2, bowtie/1.1.1</td>
</tr>
<tr>
<td>SEECER</td>
<td>Correct errors in reads to improve transcriptome assembly.</td>
<td>15 CPUs</td>
<td>24 GB/CPU</td>
<td>60</td>
<td>seecker/0.0.1.3</td>
</tr>
</tbody>
</table>

### Reads Post-Processing

<table>
<thead>
<tr>
<th>Process/Tool</th>
<th>Purpose</th>
<th>CPU</th>
<th>Memory</th>
<th>Time (min.)</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tophat</td>
<td>Map the reads to the reference genome.</td>
<td>15 CPUs</td>
<td>8 GB / CPU</td>
<td>2040</td>
<td>tophat/2.1.0-all</td>
</tr>
<tr>
<td>Genome coverage</td>
<td>Get coverage Tophat output.</td>
<td>4 CPUs</td>
<td>4 GB / CPU</td>
<td>20</td>
<td>Samtools</td>
</tr>
<tr>
<td>Samtools sort</td>
<td>Sort bam kariotypically.</td>
<td>5 CPUs</td>
<td>4 GB/CPU</td>
<td>250</td>
<td>Samtools</td>
</tr>
<tr>
<td>CreateSequenceDictionary</td>
<td>Generate dictionary from reference.</td>
<td>1 CPU</td>
<td>2 GB/CPU</td>
<td>10</td>
<td>Picard</td>
</tr>
<tr>
<td>GATK</td>
<td>Call SNPs.</td>
<td>4 CPUs</td>
<td>4 GB / CPU</td>
<td>240</td>
<td>Java, Samtools, Picard</td>
</tr>
</tbody>
</table>

### Assembly and Assembly Post-Processing

<table>
<thead>
<tr>
<th>Process/Tool</th>
<th>Purpose</th>
<th>CPU</th>
<th>Memory</th>
<th>Time (min.)</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trinity2</td>
<td>Assemble transcriptomes.</td>
<td>30 CPUs</td>
<td>66 GB/CPU</td>
<td>1500</td>
<td>trinity/2.0.6-all</td>
</tr>
<tr>
<td>BUSCO</td>
<td>Assess transcriptome completeness with single-copy orthologs.</td>
<td>8 CPUs</td>
<td>2 GB/CPU</td>
<td>80</td>
<td>emboss, hmmer, ncbi-blast, python</td>
</tr>
<tr>
<td>DETONATE</td>
<td>Evaluate transcriptome assemblies.</td>
<td>15 CPUs</td>
<td>8 GB/CPU</td>
<td>4320</td>
<td>detonate, bowtie2</td>
</tr>
<tr>
<td>gmap_build</td>
<td>Builds a gmap database for the reference genome.</td>
<td>4 CPUs</td>
<td>4 GB / CPU</td>
<td>20</td>
<td>gmap</td>
</tr>
<tr>
<td>gmap</td>
<td>Map the contigs to the reference genome</td>
<td>8 CPUs</td>
<td>4 GB/CPU</td>
<td>60</td>
<td>gmap</td>
</tr>
<tr>
<td>Genome coverage</td>
<td>Samtools</td>
<td>4 CPUs</td>
<td>4 GB / CPU</td>
<td>20</td>
<td>Samtools</td>
</tr>
<tr>
<td>Samtools sort</td>
<td>Sort bam kariotypically.</td>
<td>1 CPU</td>
<td>2 GB/CPU</td>
<td>300</td>
<td>Samtools</td>
</tr>
<tr>
<td>Samtools index</td>
<td>Generate indexes (.bai)</td>
<td>1 CPU</td>
<td>2 GB/CPU</td>
<td>10</td>
<td>Samtools</td>
</tr>
<tr>
<td>CreateSequenceDictionary</td>
<td>Generate dictionary from reference.</td>
<td>1 CPU</td>
<td>2 GB/CPU</td>
<td>10</td>
<td>Picard</td>
</tr>
<tr>
<td>Samtools faidx</td>
<td>Index reference.</td>
<td>1 CPU</td>
<td>2 GB/CPU</td>
<td>2</td>
<td>Samtools</td>
</tr>
<tr>
<td>GATK</td>
<td>Call single-nucleotide polymorphisms (SNPs).</td>
<td>4 CPUs</td>
<td>4 GB / CPU</td>
<td>240</td>
<td>Java, Samtools, Picard</td>
</tr>
</tbody>
</table>
• **Computational Heterogeneity**
  – Lots of different tools with different requirements
    • large memory thread parallel
    • regular memory thread parallel
    • small single core jobs (process placement)
  – Significant effort required to run each tool optimally
  – **Workflow tools challenged by this heterogeneity**

• **Checkpoint and Recovery**
  – Long-running processes (days to weeks!)
  – Periodically take snapshots, especially when using local disk that is purged upon job failure/completion
  – To do: look at automated checkpointing
ECSS Support: Wrangling Data Through Pipelines

• **Data movement/management**
  – Local disk/ RAM disk needed for performance
  – Scratch → Local disk → Long term storage

• **Data verification**
  – Lots of data movement steps, with lots of files
  – Verify at each step or confusing issues arise downstream
Effect of Sequencing Site

Sample A, Replicates 1 to 4, 6 Centers

- N50
- Median contig length
- Average contig

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>N50</td>
<td>1,285</td>
<td>1,535</td>
<td>964</td>
<td>1,528</td>
<td>1,281</td>
<td>1,208</td>
</tr>
<tr>
<td>Median contig length</td>
<td>966.34</td>
<td>1,036.37</td>
<td>828.27</td>
<td>1,043.91</td>
<td>959.06</td>
<td>933.78</td>
</tr>
<tr>
<td>Average contig</td>
<td>581</td>
<td>580</td>
<td>547</td>
<td>591</td>
<td>570</td>
<td>570</td>
</tr>
</tbody>
</table>
Effect of Library Prep

Aspect 3, Random R1-4 SamA

- N50
- Median contig length
- Average contig

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>N50</td>
<td>2,109</td>
<td>1,197.15</td>
<td>615</td>
</tr>
<tr>
<td>Median contig length</td>
<td>1,321</td>
<td>938.39</td>
<td>544</td>
</tr>
<tr>
<td>Average contig</td>
<td>2,160</td>
<td>1,215.99</td>
<td>620</td>
</tr>
</tbody>
</table>

Aspect 3, R5, SamA

- N50
- Median contig length
- Average contig

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>N50</td>
<td>2,021</td>
<td>1,164.12</td>
<td>602</td>
</tr>
<tr>
<td>Median contig length</td>
<td>2,033</td>
<td>1,168.44</td>
<td>606</td>
</tr>
<tr>
<td>Average contig</td>
<td>2,056</td>
<td>1,183.75</td>
<td>617</td>
</tr>
</tbody>
</table>
Ongoing work

• Address other aspects affecting transcriptome quality
  – Sequencing depths (Aspect 2)
  – Preprocessing measures (Aspect 4)
  – Potentially many others (different tools, tool options, etc.)

• Make transcriptome analysis pipeline available through a gateway so others can benefit
Galaxy XSEDE Gateway  
James Taylor, Johns Hopkins  
Anton Nekrutenko and Nate Coraor, Penn State

• **Goal:** Enable large jobs on Galaxy Main to run on XSEDE
• **Problem:** Remote job submission in Galaxy; optimized tools for XSEDE

★ ECSS Support
  – Helped with process of becoming a gateway and writing Gateway proposals
  – Jobs being submitted to Stampede for over a year
  – Created an optimized Trinity Galaxy tool to run on Bridges - now active!
Critical Assessment of Metagenomic Interpretation
http://www.cami-challenge.org

- interpretation of metagenomes relies on computational approaches
  - short read assembly
  - taxonomic binning/classification
  - taxonomic profiling
- CAMI aims at independent, comprehensive and bias-free evaluation of methods
- extensive high-quality unpublished metagenomic data sets
- results will provide reproducible and quantitative measurements of tool performance
- will serve as
  - guide to users
  - help developers identify directions for future work
CAMI Evaluation Metrics

<table>
<thead>
<tr>
<th>Anonymous Name</th>
<th># contigs</th>
<th>Largest contig</th>
<th>Total length</th>
<th>N50</th>
<th>GC (%)</th>
<th># misassemblies</th>
<th># relocations</th>
<th># translocations</th>
<th># inversions</th>
<th># misassembled contigs</th>
<th># indels</th>
</tr>
</thead>
<tbody>
<tr>
<td>pensive_babbage</td>
<td>10.0</td>
<td>6503724.0</td>
<td>8361599.0</td>
<td>6503724.0</td>
<td>46.96</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>focused_bardeen</td>
<td>8864.0</td>
<td>487875.0</td>
<td>3.680766E7</td>
<td>23804.0</td>
<td>54.17</td>
<td>28.0</td>
<td>5.0</td>
<td>23.0</td>
<td>0.0</td>
<td>27.0</td>
<td>-</td>
</tr>
<tr>
<td>sharp_perman</td>
<td>17911.0</td>
<td>888870.0</td>
<td>7.15542E7</td>
<td>19216.0</td>
<td>54.92</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>adoring_jones</td>
<td>16018.0</td>
<td>457213.0</td>
<td>4.718912E7</td>
<td>13601.0</td>
<td>54.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>goofy_darwin</td>
<td>15795.0</td>
<td>888811.0</td>
<td>4.6721356E7</td>
<td>28403.0</td>
<td>53.75</td>
<td>172.0</td>
<td>33.0</td>
<td>130.0</td>
<td>9.0</td>
<td>125.0</td>
<td>93.0</td>
</tr>
<tr>
<td>elegant_kowalevski</td>
<td>20004.0</td>
<td>2780101.0</td>
<td>8.807272E7</td>
<td>24752.0</td>
<td>54.62</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>trusting_colden</td>
<td>184.0</td>
<td>8.0</td>
<td>133.0</td>
<td>816.0</td>
<td>0.18</td>
<td>67.0</td>
<td>14.0</td>
<td>4775.0</td>
<td>9.77</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>lonely_franklin</td>
<td>32148.0</td>
<td>10239.0</td>
<td>3.469251E7</td>
<td>1183.0</td>
<td>56.54</td>
<td>45.0</td>
<td>6.0</td>
<td>37.0</td>
<td>2.0</td>
<td>45.0</td>
<td>-</td>
</tr>
<tr>
<td>drunk_galileo</td>
<td>8216.0</td>
<td>307410.0</td>
<td>3.558718E7</td>
<td>23934.0</td>
<td>54.41</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>hungry_jones</td>
<td>13325.0</td>
<td>15618.0</td>
<td>1.184191E7</td>
<td>889.0</td>
<td>55.27</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
</tr>
<tr>
<td>elated_wright</td>
<td>1957.0</td>
<td>515047.0</td>
<td>2.156639E7</td>
<td>65409.0</td>
<td>57.52</td>
<td>80.0</td>
<td>24.0</td>
<td>49.0</td>
<td>7.0</td>
<td>65.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

![Contig Length vs. GC](image.png)

7035:
- **elegant_kowalevski**: 77801086
Custom PSC topology for data-intensive HPC

20 Storage Building Blocks, implementing the parallel Pylon filesystem (~10PB) using PSC’s SLASH2 filesystem

6 “core” Intel OPA edge switches: fully interconnected, 2 links per switch

Intel OPA cables

20 “leaf” Intel OPA edge switches

42 LSM (3TB) compute nodes

4 ESM (12TB) compute nodes

2 gateways per ESM

4 ESM nodes
2 front-end nodes
2 boot nodes
8 management nodes

32 RSM nodes with NVIDIA next-generation GPUs

16 RSM nodes with NVIDIA K80 GPUs

800 RSM (128GB) compute nodes, 48 with GPUs

4 MDS nodes
2 front-end nodes
2 boot nodes
8 management nodes

20 gateways per ESM

12 database nodes

6 web server nodes

https://www.psc.edu/index.php/bridges-virtual-tour