Long-Read Sequencing

Analysis on the HPRC Clusters

Michael Dickens, PhD - Research Scientist - HPRC
PacBio Long Read Sequencing

Sequel Sequencer

Read lengths >20 kb
Data per SMRT Cell: 5–8 Gb

- Half of data in reads: >20 kb
- Top 5% of reads: >35 kb
- Maximum read length: >60 kb

pacb.com
PacBio Long Read Sequencing

Insert Size
- CCS = 100 - 2kb
- CLR = 2kb - 10kb

uncorrected reads

corrected reads

Circular Consensus Sequence (Read of Insert)

Subreads

m54601_160302_121501.subreads.bam

[1] "m" = movie
[2] instrument Serial Number

pacb.com
Oxford Nanopore Long Read Sequencing

NANOPORE SEQUENCING
At the heart of the MinION device, an enzyme unwinds DNA, feeding one strand through a protein pore. The unique shape of each DNA base causes a characteristic disruption in electrical current, providing a readout of the underlying sequence.

http://blogs.nature.com
TechBlog: The nanopore toolbox
16 Oct 2017 | 12:00 GMT | Posted by Jeffrey Perkel
Long-Read Analysis Tools

A catalogue of long read sequencing data analysis tools

- 290+ tools listed
- 29 analysis type categories
- 5 sequencing technologies
  - Bio-Nano Genomics
  - Hi-C
  - Oxford Nanopore
  - PacBio
  - 10X Genomics

long-read-tools.org
Long-Read Tools on HPRC Clusters

- Sequence Data Processing
- Tool Suites
- QC and Read Correction
- Genome Assembly
- Polishing Assemblies
- Sequence Alignments
- Obtaining Example Reads

hprc.tamu.edu/wiki/Bioinformatics:PacBio_tools

hprc.tamu.edu/wiki/Bioinformatics:OxfordNanopore_tools
Sequence Data Processing

- **pls2fasta** Ada (PacBio)
  - Convert PacBio RSII plx.h5, bax.h5 or fofn (file of file names) to fasta or fastq format
    - trim low quality reads: `-trimByRegion`
    - filter on minimum subread length `-minSubreadLength`

- **Poretools** Ada (ONT)
  - explore MinION datasets

- **Albacore** Ada (ONT)
  - provides the Oxford Nanopore basecalling algorithms

module load blasr/5.3.0
Tool Suites

- **pbbioconda** Terra (PacBio)
  - contains command line versions of a set of SMRT Analysis tools
  - pbbioconda tool list [https://github.com/PacificBiosciences/pbbioconda](https://github.com/PacificBiosciences/pbbioconda)
    - blasr - the long read aligner
    - pbalign - Python wrapper for BLASR and associated tools
      - slower than minimap2 but output is coordinate sorted bam
    - pbsv - Structural variant analysis
    - bax2bam - convert legacy PacBio bax.h5, bas.h5 and ccs.h5 to BAM
    - bam2fastq, bam2fasta - convert multiple bam files to one fastq/a.gz
      - pbindex - create a .pbi index file for a .bam file
    - genomicconsensus - for polishing genome assemblies (used by Arrow)

```
module load Anaconda/2-5.0.1
module load BamTools/2.5.1-GCCcore-7.3.0
source activate pbbioconda-2019.4.29
```
Tool Suites

- **SMRT-Link**
  - contains command line versions of additional SMRT Analysis tools
  - PacBio’s open-source SMRT Analysis software suite is designed for use with Single Molecule, Real-Time (SMRT) Sequencing data.

```
bam2fasta  bam2fastq  bamsieve  bax2bam  blasr  ccs  cleric  cromwell
DALigner   DALigner_p datander  dataset  dazcon DB2Falcon dexta
DBDust     DBRm      DBshow   DBsplit DBstats dexta
FastA2DB   fuse      gcpp     HPC.daliner HPC.REPmask
IPDSummary ipython   ipython2 isoseq3 juliet julietflow
Iaa        laagc     LAMerge  LAsort  lima  minimap2
PB Cromwell pbdagcon pbindex  pbmm2  pbservice pbsv
repmask    samtools  sawriter  summarizeModifications TANmask
```

module load SMRT-Link/8.0.0.80529-cli-tools-only
QC and Read Correction

- Filter reads on quality scores, error rate or kmers
  - Filtlong  Ada
    - trim both ends of reads based on reference genome k-mers
  - MiniScrub  Ada

- Correct assemblies
  - Racon  Ada
    - Consensus module for raw de novo DNA assembly of long uncorrected reads. (integrated into Unicycler workflow)

- PoreChop  Ada, Terra (ONT)
  - finding and removing adapters

- Correct long-reads with Illumina reads (computationally intensive)
  - LSC  Ada, Terra, Curie
    - version 2.0 supports a parallelization step (use with TAMULauncher)
  - Proovread  Ada
    - recommended to split input file and run a job array
Genome Assemblers

- **Canu** Ada, Curie
  - need to convert input .bam file to .fasta/q (quality scores not used)
  - for large genomes, use grid mode on Curie which detects resources and configures array jobs based on genome size, sequence coverage (20x recommended) and available resources
    - currently no SUs charged for running Canu on Curie

- **wtdbg2** Ada, Terra, Curie
  - does not correct reads and has its own assembly polishing option
  - claims 10 times faster than Canu with comparable base accuracy

- **Unicycler** Ada
  - assembly pipeline for bacterial genomes (uses Racon)
  - circularises replicons without needing a separate tool like Circulator

- **miniasm** Ada
  - small genomes
Sample Dataset for Comparing Canu and wtdbg2 Assemblers

*Arabidopsis thaliana* PacBio Sequel data

https://downloads.pacbcloud.com/public/SequelData/ArabidopsisDemoData/Assembly/

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Reads</td>
<td>1,135,065</td>
</tr>
<tr>
<td>Number of Bases</td>
<td>10.9 Gb (80x)</td>
</tr>
<tr>
<td>Average Read Length</td>
<td>9,474 bp</td>
</tr>
<tr>
<td>Number of SMRT Cells</td>
<td>2</td>
</tr>
</tbody>
</table>

Assembled by PacBio using HGAP4 + Arrow polish
### Comparing Assembler Resource Usage

<table>
<thead>
<tr>
<th>Assembler Configuration</th>
<th>Total Runtime</th>
<th>Total Memory</th>
<th>Compute Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canu v1.7.1 (Ada) + Arrow polish (Terra)</td>
<td>137 hours + 23 hours = 160 hours</td>
<td>35 GB : 20 GB</td>
<td>64 GB (1 job)</td>
</tr>
<tr>
<td>Canu v1.7.1 (Curie: grid) + Arrow polish (Terra)</td>
<td>29 hours + 23 hours = 53 hours</td>
<td>119 GB : 20 GB</td>
<td>multiple 256 GB (137 jobs)</td>
</tr>
<tr>
<td>wtdbg2 v2.3+ polish (uncorrected)</td>
<td>2 hours 22 min</td>
<td>34 GB</td>
<td>64 GB (1 job)</td>
</tr>
<tr>
<td>wtdbg2 v2.3 + polish (reads were corrected with canu prior to assembly)</td>
<td>20 hours 18 min</td>
<td>19.5 GB</td>
<td>64 GB (1 job)</td>
</tr>
</tbody>
</table>
Comparing Assembly Statistics

<table>
<thead>
<tr>
<th></th>
<th>Canu + Arrow</th>
<th>wtdbg2 + polish</th>
<th>correct reads + wtdbg2 + polish</th>
<th>HGAP4 * + Arrow *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total size of contigs</td>
<td>123 Mb</td>
<td>119.9 Mb</td>
<td>112 Mb</td>
<td>122.8 Mb</td>
</tr>
<tr>
<td># Contigs</td>
<td>121</td>
<td>243</td>
<td>480</td>
<td>238</td>
</tr>
<tr>
<td># Contigs &gt; 10K</td>
<td>118</td>
<td>126</td>
<td>349</td>
<td>214</td>
</tr>
<tr>
<td># Contigs &gt; 100K</td>
<td>31</td>
<td>45</td>
<td>155</td>
<td>38</td>
</tr>
<tr>
<td># Contigs &gt; 1M</td>
<td>14</td>
<td>21</td>
<td>31</td>
<td>15</td>
</tr>
<tr>
<td># Contigs &gt; 10M</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Longest contig</td>
<td>15,951,152</td>
<td>14,639,931</td>
<td>5,195,794</td>
<td>14,996,695</td>
</tr>
<tr>
<td>NG50</td>
<td>8,759,240</td>
<td>6,448,251</td>
<td>847,497</td>
<td>9,655,093</td>
</tr>
</tbody>
</table>

* performed by PacBio
Comparing Assemblies with BUSCO

All four assemblies:
BUSCO version is: 3.0.2
embryophyta_odb9
Assembly Comparison Summary

- Based on the PacBio *Arabidopsis thaliana* long read data
  - wtdbg2 was 10x faster than Canu (grid_mode)
  - wtdbg2 + polish was 20x faster than Canu (grid_mode) + Arrow step
  - wtdbg2 produced a comparable build to Canu but the Canu build is more complete based on BUSCO results
    - wtdbg2 had 15 of 1440 fewer BUSCOs than Canu assembly
  - Canu and HGAP4 produced similar quality builds
    - Canu had 3 more complete BUSCOs than HGAP4
  - Canu had the fewest number of contigs and the longest contig
- Canu is free to use (no SUs charged) on Curie and is much quicker than running on a single Ada node
Genome Hybrid Assemblers
(Long-reads + Illumina reads)

- **SPAdes** Ada, Terra
  - subreads as input files
  - no need to correct subreads with short reads prior to assembly
  - uses long reads for gap closure and repeat resolution

- **MaSuRCA** Ada
  - All long-reads must be in a single fasta file

- **Unicycler** Ada
  - assembly pipeline for bacterial genomes
  - circularises replicons without the need for a separate tool like Circulator
Polishing Assemblies

- **ArrowGrid_HPRC**  Terra only  (PacBio)
  - a parallel wrapper around ArrowGrid and the Arrow consensus SMRT Analysis software
  - running arrow alone can take weeks/months to finish on a single node
  - ArrowGrid_HPRC runs job arrays: Ex. 10 subreads files = 10 compute nodes used
- **Purge_Haplotigs**  Terra only
  - separates haplotigs from primary contigs in heterozygous diploid assemblies
- **wtdbg2  (wtpoa-cns)**  Ada, Terra, Curie
  - recommends minimap2 for alignments of reads to assembly
- **Racon**  Ada
  - can use Illumina reads for polishing an assembly; part of the Unicycler workflow
- **Circlator**  Ada
  - circularizes bacterial genomes and plasmids by joining contigs
    - uses assembly .fasta with corrected PacBio reads
- **PBJelly**  Ada  (PacBio)
  - aligns long reads to assemblies for filling or reducing gaps (in PBsuite module as Jelly.py)
- **Nanopolish**  Ada, Terra  (ONT)
  - calculate an improved consensus sequence for a draft genome assembly
Polishing Diploid Genome Assemblies

● **Purge_Haplotigs**  Terra, Ada, Curie
  ○ separates haplotigs from primary contigs in highly heterozygous diploid assemblies
  ○ uses mapped read coverage and minimap2 alignments to determine which contigs to keep for the haploid assembly.

● **Redundans**  Ada
  ○ program takes as input assembled contigs, sequencing libraries and/or reference sequence and returns scaffolded homozygous genome assembly.
  ○ final assembly should be less fragmented and with total size smaller than the input contigs.
  ○ will automatically close the gaps resulting from genome assembly or scaffolding.
Sequence Alignments

- **minimap2**  
  - Ada, Terra, Curie  
  - DNA or mRNA sequences

- **pbalign**  
  - Terra  
  - (PacBio)  
  - installed in Anaconda pbbioconda-2019.4.29 environment  
  - Python wrapper for BLASR and associated tools  
  - much slower than minimap2 but output is coordinate sorted bam

- **blasr**  
  - Ada, Terra*  
  - (PacBio)  
  - * found in Anaconda pbbioconda-2019.4.29 environment

- **pbmm2**  
  - Terra  
  - (PacBio)  
  - A minimap2 SMRT wrapper for PacBio data  
  - installed in Anaconda pbbioconda-2019.4.29 environment  
  - official replacement for BLASR  
  - can align and sort with one command (--sort)  
  - by default all threads on compute node will be used unless specified
## Sequence Aligner Comparison

### aligning one subreads.bam file from the PacBio A. thaliana dataset to the PacBio HGAP4 assembly

<table>
<thead>
<tr>
<th></th>
<th>minimap2 + samtools sort</th>
<th>pbmm2 align --sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>% mapped</td>
<td>95.0%</td>
<td>89.4%</td>
</tr>
<tr>
<td>secondary</td>
<td>554,423</td>
<td>0</td>
</tr>
<tr>
<td>supplementary</td>
<td>61,611</td>
<td>64,110</td>
</tr>
<tr>
<td>memory used</td>
<td>13.8 GB</td>
<td>9.9 GB</td>
</tr>
<tr>
<td>time</td>
<td>28 min</td>
<td>19 min</td>
</tr>
<tr>
<td>version</td>
<td>2.16 &amp; 1.9</td>
<td>1.0.0</td>
</tr>
</tbody>
</table>
| notes                |                          | 1. unmapped reads are not saved in bam output file by default  
                      |                          | 2. secondary reads are not reported and no option available (as with blasr) |

Notes:

- unmapped reads are not saved in bam output file by default
- secondary reads are not reported and no option available (as with blasr)
RNA-Seq Sequence Alignments

- **minimap2** Ada, Terra, Curie
  - DNA or mRNA sequences
- **Star** Ada, Terra, Curie
  - requires more memory than GMAP
- **GMAP** Ada
Structural Variant Analysis

- **pbsv**
  - Terra (PacBio)
  - Call and analyze structural variants in diploid genomes
  - Calls insertions, deletions, inversions, duplications, and translocations
  - Installed in Anaconda pbbioconda-2019.4.29 environment

- **PBHoney**
  - Ada, Terra (PacBio)
  - Installed in the PBSuite module as Honey.py
  - Good for small structural variants

- **Sniffles**
  - Ada
  - Detects all types of SVs (10bp+) using evidence from split-read alignments, high-mismatch regions, and coverage analysis.
Obtaining Example Long-Reads

- **PacBio** provides sample datasets
    - RS II and Sequel (human 10x & 60x) datasets

- **Simulate Reads**
  - **BBMap** Ada, Terra (PacBio)
    - use PacBio error model and set min/max lengths
  - **NanoSim** Ada, Terra (ONT)
  - **DeepSimulator** Terra (Anaconda) (ONT)
    - deep learning based simulator to mimic the entire pipeline of Nanopore sequencing
Genomic Computational Analysis Templates (GCATemplates) HPRC Template Job Scripts
See GCATemplates Availability on the HPRC wiki

**Canu**

- GCATemplates available: ada & curie
- Canu Documentation
- Canu Tutorial

```bash
module load Canu/1.7.1-Intel-2017A-Python-3.5.2
```

Canu is a fork of the Celera Assembler.
Canu assembles reads from PacBio RS II or Oxford Nanopore MinION instruments into uniquely-assembleable contigs. unitigs.
When using Canu on ada, be sure to use the following option in your canu command (see canu docs for details):

```bash
useGrid=false
```

If your assembly runs out of walltime, you can restart your jobs script and canu will start from where it left off.

**Canu Grid Mode**

- GCATemplates available: curie

Canu can be run in grid mode on the Curie cluster. Canu will automatically query the available compute nodes and create job arrays based on CPUs and amount of memory per node. The following is an example command forcing 16 cores to be used per node.

[Click to see template script on github]

[https://hprc.tamu.edu/wiki/Bioinformatics:PacBio_tools#Canu](https://hprc.tamu.edu/wiki/Bioinformatics:PacBio_tools#Canu)
Example of a GCATemplates script for Ada on github.tamu.edu
GCATemplates Command Line Tool

- This HPRC resource provides numerous template job scripts to help you get started with your bioinformatics project
- Template scripts use existing software modules on the HPRC cluster
- Many have sample small datasets that you can run immediately for testing
- All template job scripts linked on the HPRC wiki are available using the GCATemplates command line tool

```
module load GCATemplates
gcatemplates
```
GCATemplates Command Line Tool

- Select **CATEGORY, TASK, TOOL, OPTIONS** to find a template **SCRIPT**
- Copy the **SCRIPT** to your current directory
- Edit the selected job script template with your input files and your job specific parameters
- Long-read sequencing templates:
  - Canu, wtdbg2, Circulator
  - LSC, proovread
  - Purge_Haplotigs
  - ArrowGrid_HPRC
  - pbmm2, pbalign, minimap2
  - NanoSim

```
BIOINFORMATICS GCATemplates [ada]

CATEGORY -------------- PacBio tools
TASK ----------- assemble and polish
TOOL ------------ wtdbg2.2.3
OPTIONS ------------- merge subreads.bam inputs, assemble, polish
SCRIPT ----------------- run wtdbg2_2.3 merge subreads assemble polish ada.sh

Copy SCRIPT to current directory?

y yes
b back
h home
s search
q quit

Select:
```
For More Help...

Website: hprc.tamu.edu
Email: help@hprc.tamu.edu
Telephone: (979) 845-0219
Visit us in person: Henderson Hall, Room 114A

Help us, help you -- we need more info
- Which Cluster
- UserID/NetID
- Job id(s) if any
- Location of your job file, input/output files
- Application used if any
- Module(s) loaded if any
- Error messages
- Steps you have taken, so we can reproduce the problem