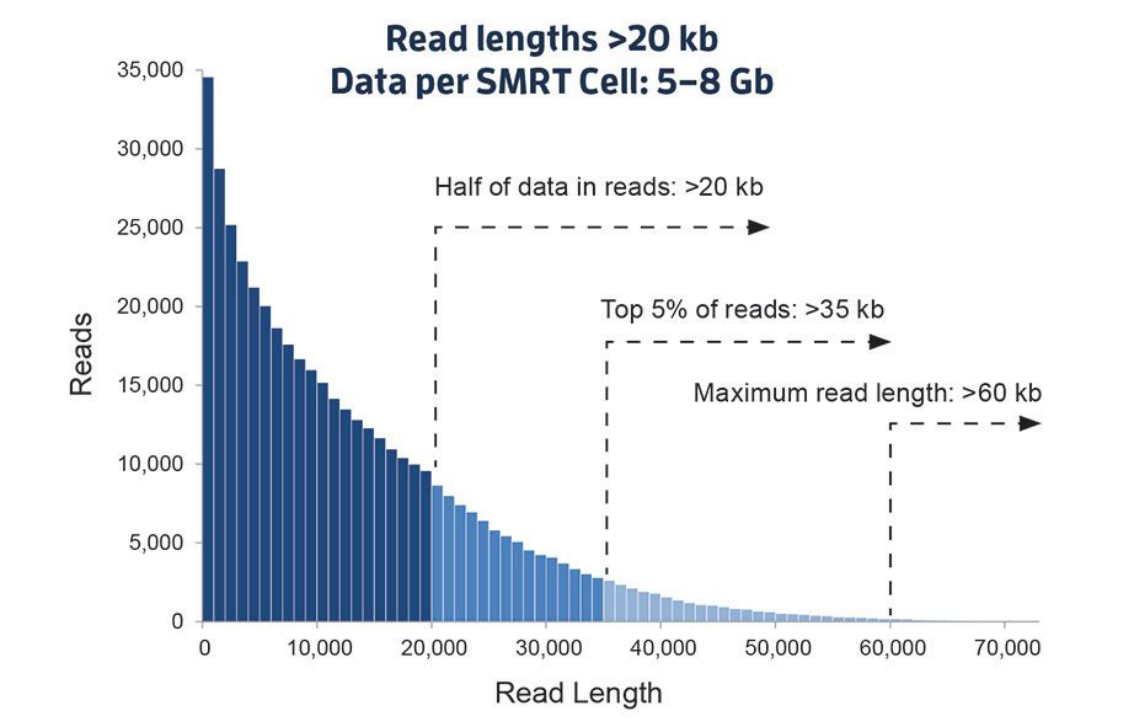
Long-Read Sequencing Analysis on the HPRC Clusters

Michael Dickens, PhD - Research Scientist - HPRC

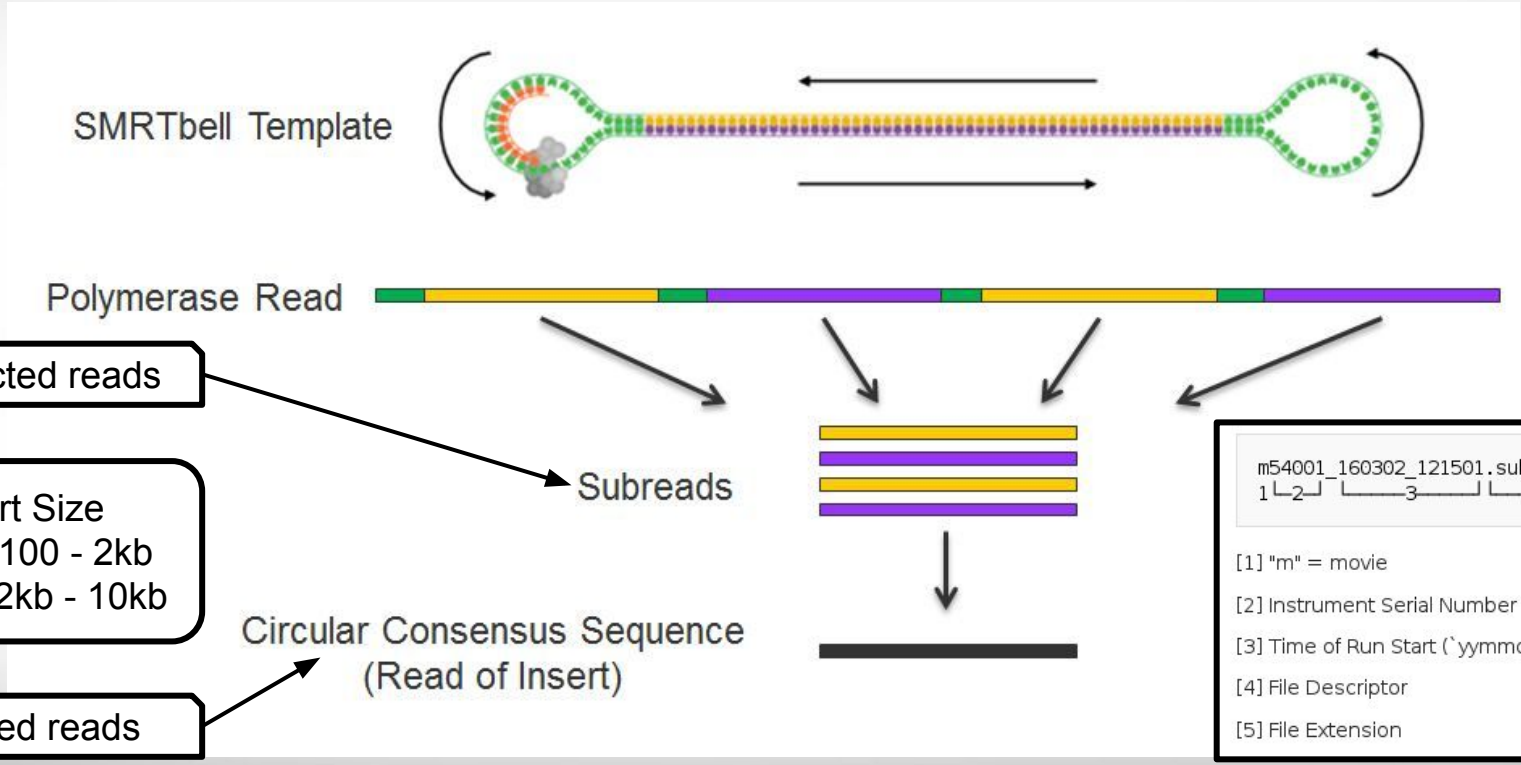
PacBio Long Read Sequencing

Sequel Sequencer



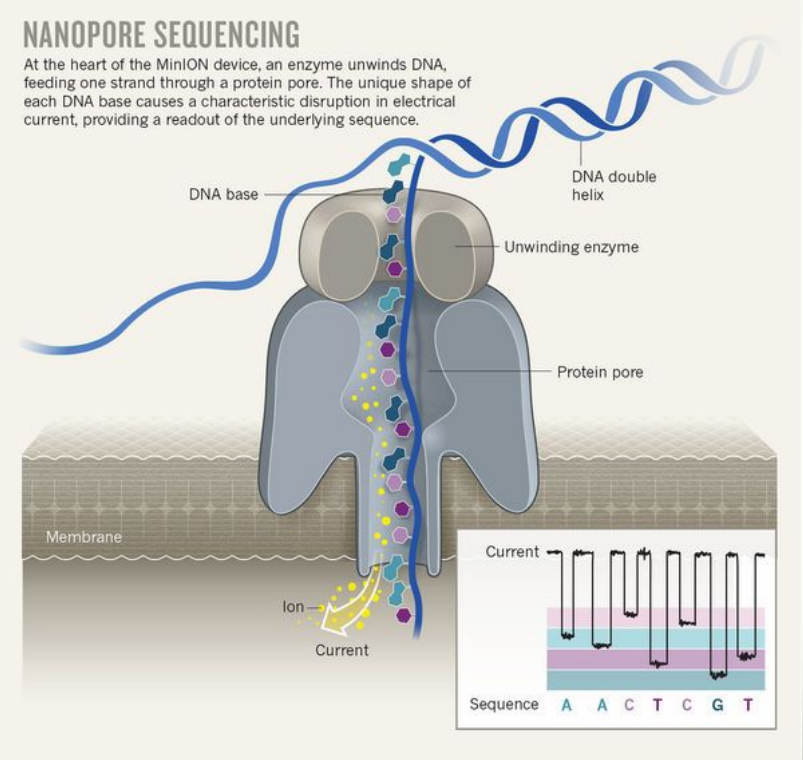
pacb.com

PacBio Long Read Sequencing



pacb.com

Oxford Nanopore Long Read Sequencing



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

Long-Read Analysis Tools

long-read-tools.org

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 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`

```
module load blasr/5.3.0
```

- **Poretools** Ada (ONT)
  - explore MinION datasets
- **Albacore** Ada (ONT)
  - provides the Oxford Nanopore basecalling algorithms

Tool Suites

- **pbbioconda** Terra (PacBio)
  - contains command line versions of a set of SMRT Analysis tools
  - pbbioconda tool list 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** Terra (PacBio)
  - 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	DB2fasta	DBdump
DBdust	DBrm	DBshow	DBsplit	DBstats	dexta	falconc	falconcpp
fasta2DB	fuse	gcpp	HPC.daligner	HPC.REPmask		HPC.TANmask	
ipdSummary	ipython	ipython2	isoseq3	juliet	julietflow	LA4Falcon	LA4Ice
laa	laagc	LAmerge	LAsort	lima	minimap2	motifMaker	pbalign
pbcromwell	pbdagcon	pbindex	pbmm2	pbservice	pbsv	pbvalidate	ra
REPmask	samtools	sawriter	summarizeModifications		TANmask	undexta	womtool

```
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/

Number of Reads	1,135,065
Number of Bases	10.9 Gb (80x)
Average Read Length	9,474 bp
Number of SMRT Cells	2

Assembled by PacBio using HGAP4 + Arrow polish

Comparing Assembler Resource Usage

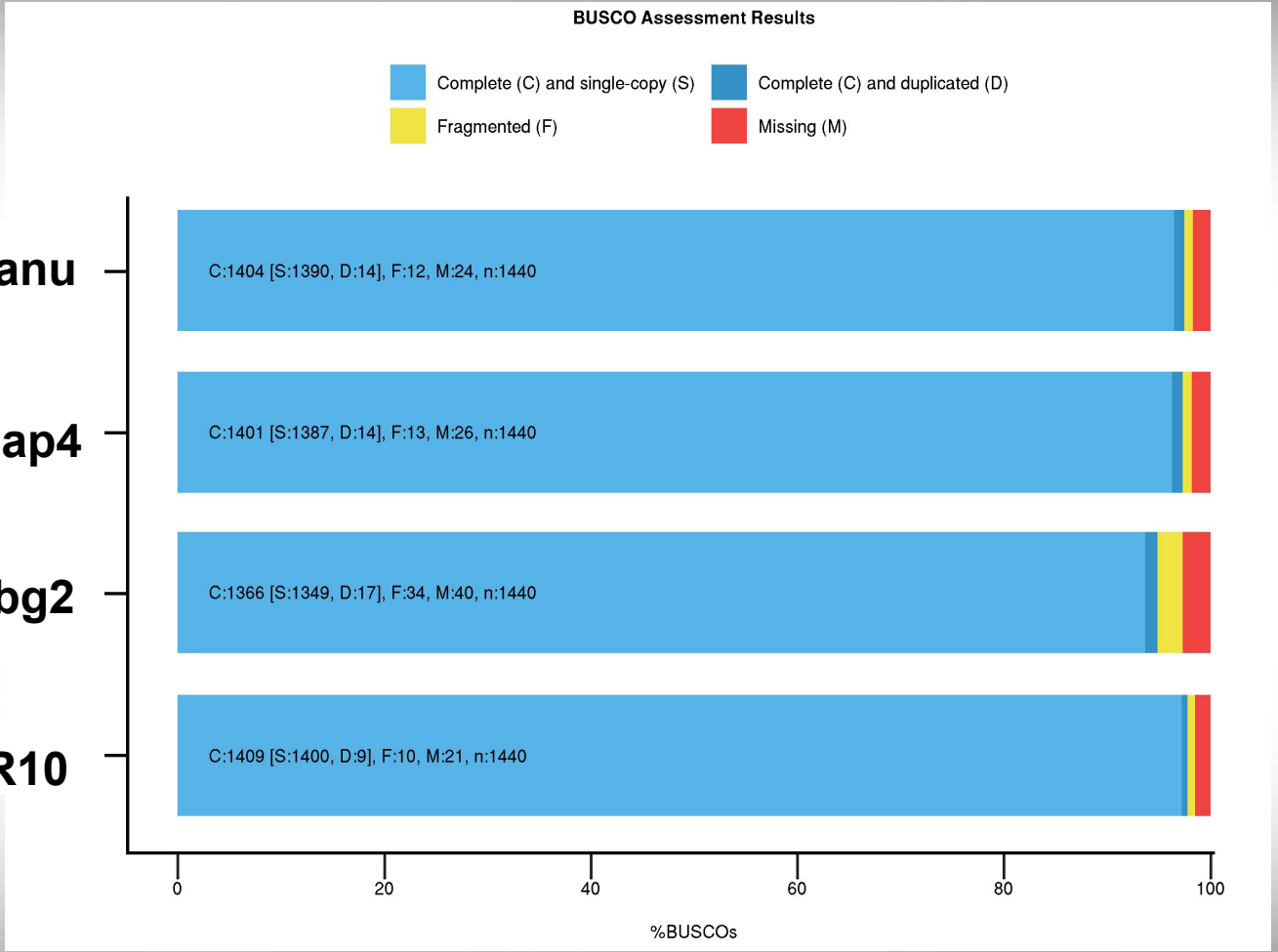
	Total Runtime	Total Memory	compute node
Canu v1.7.1 (Ada) + Arrow polish (Terra)	137 hours + 23 hours = 160 hours	35 GB : 20 GB	64 GB (1 job)
Canu v1.7.1 (Curie: grid) + Arrow polish (Terra)	29 hours + 23 hours = 53 hours	119 GB : 20 GB	multiple 256 GB (137 jobs)
wtdbg2 v2.3+ polish (uncorrected)	2 hours 22 min	34 GB	64 GB (1 job)
wtdbg2 v2.3 + polish (reads were corrected with canu prior to assembly)	20 hours 18 min	19.5 GB	64 GB (1 job)

Comparing Assembly Statistics

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

* 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

	minimap2 + samtools sort	pbmm2 align --sort
% mapped	95.0%	89.4%
secondary	554,423	0
supplementary	61,611	64,110
memory used	13.8 GB	9.9 GB
time	28 min	19 min
version	2.16 & 1.9	1.0.0
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)

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

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

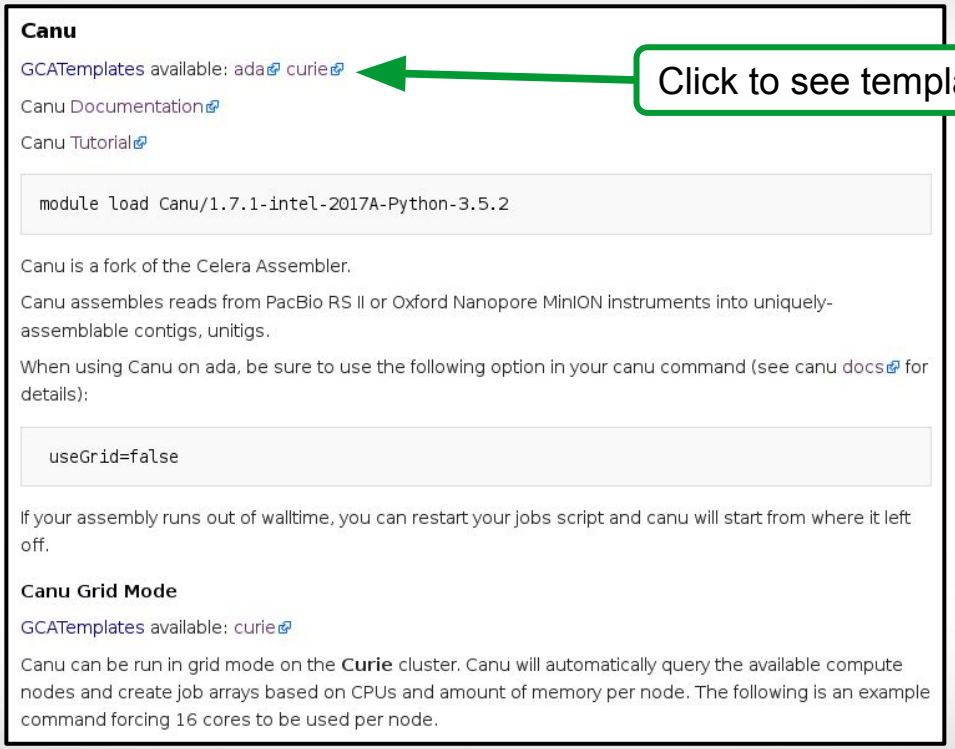
- **pbsv** Terra (PacBio)
  - call and analyze structural variants in diploid genomes
  - calls insertions, deletions, inversions, duplications, and translocations
  - installed in Anaconda pbbioconda-2019.4.29 environment
- **PBHoney** Ada, Terra (PacBio)
  - installed in the PBSuite module as Honey.py
  - 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**
  - https://github.com/PacificBiosciences/DevNet/wiki/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

```
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-assemblable contigs, unitigs.

When using Canu on ada, be sure to use the following option in your canu command (see canu docs for details):

```
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.

https://hprc.tamu.edu/wiki/Bioinformatics:PacBio_tools#Canu

Example of a GCATemplates script for Ada on github.tamu.edu

```
#BSUB -L /bin/bash              # use the bash login shell to initialize environment.
#BSUB -J canu_ada_westmere      # job name
#BSUB -n 40                     # assigns 40 cores for execution
#BSUB -R "span[ptile=40]"       # assigns 40 cores per node
#BSUB -R "select[mem2tb]"       # request 1TB memory node
#BSUB -R "rusage[mem=49750]"    # reserves 49GB memory per core
#BSUB -M 49750                  # sets to 49GB per process enforceable memory limit. (M * n)
#BSUB -W 48:00                  # sets to 48 hour the job's runtime wall-clock limit.
#BSUB -q xlarge                 # xlarge queue required for Westmere nodes
#BSUB -o stdout.%J              # directs the job's standard output to stdout.jobid
#BSUB -e stderr.%J              # directs the job's standard error to stderr.jobid

module load Westmere
module load Canu/1.8-intel-2017A-Python-3.5.2

<<'README'
    - Canu Tutorial: http://canu.readthedocs.io/en/latest/tutorial.html
README

###############################################################################
# TODO Edit these variables as needed:
# PARAMETERS
genome_size='40m'               # supported units: g, m, k
stop_on_low_coverage="stopOnLowCoverage=4"  # default 10, using 4 for sample dataset, adjust as needed

# INPUTS
pacbio_raw_reads="/scratch/datasets/GCATemplates/data/pacbio/OR74A_filtered_subreads.fastq"

# OUTPUTS
prefix="n_crassa"
assembly_directory="build_1.8_out"

###############################################################################
# command to run pipeline with -pacbio-raw option
canu useGrid=false -p $prefix -d $assembly_directory genomeSize=$genome_size \
 -pacbio-raw $pacbio_raw_reads $stop_on_low_coverage

<<CITATION
    - Acknowledge TAMU HPRC: https://hprc.tamu.edu/research/citations.html

    - Canu: Koren S, Walenz BP, Berlin K, Miller JR, Phillippy AM.
            Canu: scalable and accurate long-read assembly via adaptive
            k-mer weighting and repeat separation. Genome Research. (2017).
CITATION
```

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, Circlator
  - LSC, proovread
  - Purge_Haplotigs
  - ArrowGrid_HPRC
  - pbmm2, pbalign, minimap2
  - NanoSim

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 jobfile, input/output files
- Application used if any
- Module(s) loaded if any
- Error messages
- Steps you have taken, so we can reproduce the problem