AGENDA

- SMRT Link Basics
- Resequencing
  - What’s new
  - Howto
- Barcoding
- CCS
  - What’s new
  - Howto
- Long Amplicon Analysis 2
- IsoSeq
- Assembly
  - HGAP.4
- Command Line Usage
SMRT Link
SMRT LINK – SEQUEL SYSTEM
AN INTEGRATED END-TO-END WORKFLOW MANAGER
Data Management

Workflow
Create a new data set  
View existing data set
Select between RSII and BAM data

Live search bar
Column Selector
Select data to include in Data Set
Choose the type of job you want to view

Create a new analysis job

View available data sets

Live search bar & column chooser
- Switch between Sequel data and imported PacBio RS II data
- Available data sets (down arrow reveals cell names)
- Live search bar & column chooser
- Add data to analysis
- Remove data from analysis
All workflows have been migrated to new BAM file format. You may still choose PacBio RS II data within the data selection area, but…
…but you will only be able to convert your data into BAM format.
Note: once you reach this screen, you cannot change the name of your dataset, or choose other data. You must click “Cancel” and restart the creating an analysis job.
Each analysis application has REQUIRED and ADVANCED parameters.

Note: Only REQUIRED parameters are revealed initially. Click on “Advanced Analysis Parameters” to reveal all available options.
Resequencing
RESEQUENCING

- What’s new
  - End to end use of BAM format
  - More scalable, quicker consensus results
  - New consensus calling / polishing algorithm – Arrow
NEW “ARROW” MODEL SUPERSEDES “QUIVER” MODEL

Simple HMM model, better grounded in statistical theory
- Operates in true probability space, enabling calibrated QV output and principled variant calling and phasing
- Runway for continued statistical insights
- More reliable to train, enabling better automation @ PacBio

Doesn’t require many basecaller “QV” tracks
- Smaller file sizes…

Applicable to Sequel and RS P6-C4 data
- CCS2
- GenomicConsensus
- LAA (HLA analysis)
CONSENSUS APPLICATION ARCHITECTURE

Core C++ libraries

ConsensusCore (quiver algorithm)

LAA (1) (“Long amplicon analysis”)

GenomicConsensus (“quiver” and “arrow” tools)

CCS (1) (“Circular consensus sequencing”)

SAv2

SAv3

CCS2

LAA2

t
Specify the reference genome and click “Start”

Note: Only REQUIRED parameters are revealed initially. Click on “Advanced Analysis Parameters” to reveal all available options.
- Live search bar & column chooser
- Reference names are hyperlinks, simply click the name and hit “Select”
SMRT Analysis

Analysis Results

- Analysis Overview
  - Status
  - Thumbnails
  - Display All
- Mapping Report
- Consensus Variants
- Top Variants
- Data

- Analysis: A7_Ecoli
- Analysis Id: 3783
- Status: SUCCESSFUL
- Created At: 2/29/2016, 8:46:15 PM
- Updated At: 2/29/2016, 8:46:15 PM
- Application: Resequencing
- Inputs: Reference, BAM Data
- Path: /pbi/dept/secondary
- Analysis Parameters

Navigation bar
One click access to all graphs produced by SMRT Analysis
### Analysis Results

**Analysis Overview**

- Analysis: A7_Ecoil
- Analysis Id: 3783
- Status: SUCCESSFUL
- Created At: 2/29/2016, 8:46:15 PM
- Updated At: 2/29/2016, 8:46:15 PM
- Application: Resequencing
- Inputs: Reference, BAM Data
- Path: /pbi/depdept/secondary

**Job specific reports**

- Mapping Report
- Consensus Variants
- Top Variants
- Analysis Parameters

---

**Dataset**

- Data: The page also contains a section for data, but the specific details are not visible in the image.
Download all available data
<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlignmentSet</td>
<td>84,861 bytes</td>
<td>AlignmentSet</td>
</tr>
<tr>
<td>PacBio Json Report</td>
<td>6,837 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>ContigSet</td>
<td>37,237 bytes</td>
<td>ContigSet</td>
</tr>
<tr>
<td>Gff</td>
<td>19,029 bytes</td>
<td>gff</td>
</tr>
<tr>
<td>Fastq Gathered</td>
<td>9,285,086 bytes</td>
<td>Fastq</td>
</tr>
<tr>
<td>VCF file</td>
<td>7,539 bytes</td>
<td>vcf</td>
</tr>
<tr>
<td>Loading report</td>
<td>27,224 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>BED file</td>
<td>7,642 bytes</td>
<td>bed</td>
</tr>
<tr>
<td>Alignment Summary GFF</td>
<td>89,128 bytes</td>
<td>gff</td>
</tr>
<tr>
<td>Output GFF file</td>
<td>118,519 bytes</td>
<td>gff</td>
</tr>
<tr>
<td>Coverage report</td>
<td>1,665 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>Variants report</td>
<td>3,152 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>AlignmentSet</td>
<td>50,708 bytes</td>
<td>AlignmentSet</td>
</tr>
<tr>
<td>pbsmrtpipe log</td>
<td>0 bytes</td>
<td>log</td>
</tr>
<tr>
<td>Master log file</td>
<td>633 bytes</td>
<td>log</td>
</tr>
</tbody>
</table>
### JSON FILE FORMAT

- **JavaScript Object Notation**
  - Open standard format
  - Human readable
  - Attribute value pairs
  - Use as a replacement for XML
  - Language independent

```json
{
    "firstName": "John",
    "lastName": "Smith",
    "isAlive": true,
    "age": 25,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021-3100"
    },
    "phoneNumbers": [
        {
            "type": "home",
            "number": "212 555-1234"
        },
        {
            "type": "office",
            "number": "646 555-4567"
        },
        {
            "type": "mobile",
            "number": "123 456-7890"
        }
    ],
    "children": [],
    "spouse": null
}
```
Barcoding Basics
Review
BARCODING BACKGROUND

Short Insert

Polymerase will go around multiple times; multiple opportunities to view barcode

Long Insert

Few polymerases may make >1 pass; many polymerases may not see first barcode (or second one)
BARCODING SOLUTION WORK WELL FOR SHORT INSERTS
(TESTED TO 6 KB)

Barcode During Amplification

450, 16-bp barcodes can be synthesized into primers

Forward Primer

Forward Barcode

Reverse Barcode

Reverse Primer

PCR

Barcode

Adapter

Barcode After Amplification/Fragmentation

96 adapters with 16-bp barcodes in the stems*

*Adapters with barcodes are used to identify and distinguish different samples in the subsequent sequencing process.
BARCODE SCORING MODES

Symmetric Mode
- Barcode sequences and orientation are the **same** on both sides of the insert
- Recommended for all inserts, including inserts longer than 3 kb.

Asymmetric Mode
- **Different** barcode sequences or **different orientation** on either end of insert.
- Only recommended for inserts where both ends are sequenced.

RED circles are the sequences and orientation for barcode.fasta
BARCODE FASTA – SYMMETRIC MODE

Symmetric Mode
- Barcode sequences are the same on both sides of the insert
- Recommended for all inserts, including inserts longer than 3 kb.

- Each Barcode associated with one sample
- As read in FASTA, left end of barcode abuts adapter
BARCODE FASTA – ASYMMETRIC MODE

Asymmetric Mode

- **Different** barcode sequences on either end of the insert.
- Only recommended for inserts where both ends are sequenced.

- Each **sequential pair** of barcodes is associated with one sample
- As read in FASTA
  - Left end of forward barcode abuts adapter
  - Right end of reverse barcode abuts adapter
BARCODE FASTA – ASYMMETRIC MODE FOR BUP

Asymmetric (‘Tailed’) Mode

- **Same** barcode sequences on either end of the insert **but opposite orientation**

```
1 >FwdBC1
2 TCAGACGATGCGTCA
3 >FwdBC1_r
4 TCAGACGATGCGTCA
5 >FwdBC2
6 TATGACTAGTGTA
7 >FwdBC2_r
8 TATGACTAGTGTA
9 >FwdBC3
10 CAGATCTCTGTGATGT
11 >FwdBC3_r
12 CAGATCTCTGTGATGT
```

- Each **sequential pair** of barcodes are associated with one sample
- **Headers** must be **unique**
- As read in FASTA
  - Left end of forward barcode abuts adapter
  - Right end of reverse barcode abuts adapter
BARCOLDING – SMRT LINK (3.1)

- Barcoding is a separate protocol in SMRT Link

- Barcoded datasets get a label: “(barcoded)”
BARCODING – SMRT LINK (3.1)

- Set up barcoding protocol

1. Select Barcodes
2. Select Score Mode
3. Start Job
BARCODING – SMRT LINK (3.1) OUTPUTS

- Barcode Report – “Barcode Name” is the 0-indexed header in the BarcodeSet

<table>
<thead>
<tr>
<th>Barcode Name</th>
<th>Reads</th>
<th>Bases</th>
</tr>
</thead>
<tbody>
<tr>
<td>0--0</td>
<td>2449</td>
<td>4313370</td>
</tr>
<tr>
<td>1--1</td>
<td>2256</td>
<td>3997377</td>
</tr>
<tr>
<td>10--10</td>
<td>1788</td>
<td>3123661</td>
</tr>
<tr>
<td>100--100</td>
<td>2956</td>
<td>5259129</td>
</tr>
<tr>
<td>101--101</td>
<td>2374</td>
<td>4083767</td>
</tr>
</tbody>
</table>

- SubreadSet points to a BAM dataset with barcode tags

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubreadSet</td>
<td>8,429 bytes</td>
<td>SubreadSet</td>
</tr>
<tr>
<td>JSON report</td>
<td>40,035 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>Master log file</td>
<td>633 bytes</td>
<td>log</td>
</tr>
<tr>
<td>pbsmrtpipe log</td>
<td>0 bytes</td>
<td>log</td>
</tr>
</tbody>
</table>
BARCODING – SMRT LINK (3.1) ANALYSIS

- Currently supported protocols with barcoded datasets
  - Circular Consensus Sequences (CCS 2)
  - Long Amplicon Analysis (LAA 2)
  - Convert BAM to FASTX
BARCODING – SMRT ANALYSIS (3.1)

- Notable changes since SMRT Analysis v2.3
  - Barcode scoring is **orientation specific**
  - Use named BarcodeSets from SMRT Link according to sample prep
  - Barcode sequences automatically removed from SubreadSet (put into scraps.bam)

- BAM Alterations
  - Header
    - Metadata about barcode fasta, count, scoremode
  - Tags
    - bc: (added) **Barcode**, uint16[2] tuple [FwdIndex,RevIndex]
    - bq: (added) **BarcodeQual**, uint8 mean SW alignment score of called barcode
    - cx: (modified) **Read Context**, barcode_[before|after], pass direction* bitflags
  - Barcode Sequence
    - Moved from subreads.bam → scraps.bam
*NEW* BARCODE SCORING IN SMRTANALYSIS 3.X

- Old algorithm

  max barcode score = length of forward + reverse barcode

- New algorithm

  max barcode score = 2(length of forward and reverse barcode)

We have updated the Smith-Waterman algorithm matching barcodes next to the adapter regions to account for direction and # unique bases aligned, so every match has a +2 effect on score.
EXAMPLE SCORING

13/16 bp in forward match, + 14/16 bp reverse match

Old algorithm:
maxScore = 16+16 = 32
labelZmws = 27
Typical 95 Conf / 99 Accuracy threshold = 2 \times \frac{(13+14)}{32} = \sim 24

*New algorithm:
maxScore = 2 \times (16 + 16) = 64
labelZmws = 54
Typical 95 Conf / 99 Accuracy threshold = 2 \times \frac{2(13+14)}{3} = \sim 48

*Still testing this feature, may want to run a few different cutoffs and examine filtering yourself before deciding empirical cutoff for your data
BARCODING – SMRT ANALYSIS (3.1) (CONT’D)

- SMRT Link / pbsmrtpipe
  - Supported Pipelines automatically process per barcode
- Manual Demultiplexing
  - Barcoding protocol only **labels** BAM files
  - Per Barcode subreadset.[:name:].xml files
    - Use ‘dataset split –barcodes’ tool
      ```
      $ dataset split --barcodes --outdir myBarcodeXmls subreads_barcoded.subreadset.xml
      ```
    - Creates one XML per barcode with barcode filter
    - Chunk0 is the most abundant barcode.

- (Optional) Split into BAM files per barcode
  - Use ‘dataset consolidate’ per XML created by ‘dataset split’
CUSTOM BARCODING

- Maximum barcode score = 127

- $ dataset create --generateIndices --name my_symmetric_barcode --type BarcodeSet
  my_symmetric_barcode.barcodeset.xml
  my_symmetric_barcode.fasta

- $ pbservice import-dataset --host smrtlink-host --port 9091 my_symmetric_barcode.barcodeset.xml
2.3 CCS: PROBLEMS!

Old CCS algorithm had asymptotic plateau in accuracy.
CCS: A NEW BEGINNING

- CCS had problems!
  - Accuracy *saturation*
  - Residual homopolymer errors frustrated users

- **So we rebuilt CCS from scratch, replacing Quiver model with new *Arrow* model.**
  - True HMM model
  - Accounts for ZMW-to-ZMW variability
  - Training and testing model changes much easier
CCS: A NEW BEGINNING

Accuracy vs. Subreads

Empirical Mean Accuracy (Phred Scale)

Number of Subreads

Analysis
- Old
- New
Previously, residual errors chiefly resided in G/C homopolymers.

Not anymore.
Specify the minimum number of passes around a SMRTbell template necessary to generate a consensus sequence

<table>
<thead>
<tr>
<th>Sample Name</th>
<th>Run Name</th>
<th>Completed On</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>lambdaNEB_B01</td>
<td>Run_201511170019_A8</td>
<td>Invalid Date</td>
<td></td>
</tr>
</tbody>
</table>

Specify the minimum predicted consensus accuracy necessary to report a consensus sequence

- **Minimum Number of Passes**: 3
- **Minimum Predicted Accuracy**: 0.9

Note: Only REQUIRED parameters are revealed initially. Click on “Advanced Analysis Parameters” to reveal all available options.
<table>
<thead>
<tr>
<th>Value</th>
<th>Analysis Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>81463</td>
<td>CCS reads</td>
</tr>
<tr>
<td>128493310</td>
<td>Number of CCS bases</td>
</tr>
<tr>
<td>1552</td>
<td>CCS Read Length (mean)</td>
</tr>
<tr>
<td>0.9996</td>
<td>CCS Read Score (mean)</td>
</tr>
<tr>
<td>14</td>
<td>Number of Passes (mean)</td>
</tr>
</tbody>
</table>
CCS Read Length

![Graph showing CCS Read Length distribution.]
Analysis Results - RH_umd_bestCell

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>21,931 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>ConsensusReadSet</td>
<td>50,020 bytes</td>
<td>ConsensusReadSet</td>
</tr>
<tr>
<td>CCS report</td>
<td>4,327 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>GZIPed FASTA file(s)</td>
<td>12,820,827 bytes</td>
<td>gzip</td>
</tr>
<tr>
<td>GZIPed FASTQ file(s)</td>
<td>44,691,339 bytes</td>
<td>gzip</td>
</tr>
<tr>
<td>pbsmrtpipe log</td>
<td>0 bytes</td>
<td>log</td>
</tr>
<tr>
<td>Master log file</td>
<td>633 bytes</td>
<td>log</td>
</tr>
</tbody>
</table>
Long Amplicon Analysis
Review
LONG AMPLICON ANALYSIS

Generalized Amplicon Pipeline
- Customizable pipeline for de novo analysis of pooled amplicon datasets

Grouping Algorithms
- Barcodes (sample prep)
- Markov Graph Clustering (large-scale differences)
- Phasing (small-scale differences)

Polishing and Filtering
- Barcode Filter (pre-process)
- Length Filter (pre-process)
- Full-Pass Filter (pre-process)*
- Arrow* Polishing
- UCHIME Chimera Filter (post-process)
- Quality (Noise) Filter (post-process)

*New in LAA2
LONG AMPLICON ANALYSIS

- Notable changes since SMRT Analysis v2.3
  - Arrow Consensus Model
  - Full-Length filter (adapter-to-adapter subreads)
  - ‘Waterfall’ Clustering
    1. Cluster only up to Max Clustering Reads (-c)
       Generate POA consensuses per cluster
    2. Align/add additional subreads to each cluster up to a total of Maximum Number of Reads (-r) before phasing analysis
    3. Sort full-length sequences to top for better phasing
- More parameters exposed in SMRT Link
- Improved parallel processing for increased speed
  - SMRT Link & pbsmrtpipe automatically chunk and process barcodes in parallel
- Streamlined Outputs
  - Primary outputs unchanged (consensus sequences, summary)
  - Subread map files generated per-barcode
  - Coming soon: per-consensus BAM alignments
LONG AMPLICON ANALYSIS – PARAMETERS

- New Parameters (not available in SMRT Portal 2.3)
  - Full Length Subreads
  - Maximum Length
  - Max Phasing Reads
  - Minimum SNR
  - Min Split Reads
## LONG AMPLICON ANALYSIS – ADVANCED PARAMETERS

### Advanced Analysis Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LAA</strong></td>
<td></td>
</tr>
<tr>
<td>Min Split Score</td>
<td>500</td>
</tr>
<tr>
<td>Min Mutation Prevalence</td>
<td>0.09</td>
</tr>
<tr>
<td>Trim Ends</td>
<td>0</td>
</tr>
<tr>
<td>Min Predicted Accuracy</td>
<td>0.95</td>
</tr>
<tr>
<td>Cluster Inflation</td>
<td>2</td>
</tr>
<tr>
<td>Min Mutation Q-Score</td>
<td>20</td>
</tr>
<tr>
<td>Maximum number of reads to use in clustering</td>
<td>500</td>
</tr>
<tr>
<td>Chimera Score Threshold</td>
<td>1</td>
</tr>
<tr>
<td>Cluster Loop Weight</td>
<td>0.001</td>
</tr>
<tr>
<td>Min total barcode score</td>
<td>0</td>
</tr>
</tbody>
</table>

### New Cluster/Phase Parameters (cont’d)

- Min Split Score/Fraction
- Min Mutation Prevalence/Q-score
- Cluster Inflation/Loop Weight
- Max Reads in Clustering
- Chimera Score Threshold
- Min Total Barcode Score
- RNG Seed
LAA PROCESS

Optionally Separate by Barcode

Overlap

Cluster

Phasing

Arrow

Haplotype 1

Haplotype 2

Post-Processing Filters
### Long AmpliCon Analysis Output

#### Analysis Overview

#### AmpliCon Consensus

<table>
<thead>
<tr>
<th>Barcode</th>
<th>Sequence Cluster</th>
<th>Sequence Phase</th>
<th>Length (bp)</th>
<th>Estimated Accuracy</th>
<th>Subreads coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0--0</td>
<td>0</td>
<td>1</td>
<td>3138</td>
<td>100.0%</td>
<td>210</td>
</tr>
<tr>
<td>0--0</td>
<td>0</td>
<td>2</td>
<td>3138</td>
<td>100.0%</td>
<td>249</td>
</tr>
<tr>
<td>0--0</td>
<td>1</td>
<td>1</td>
<td>3384</td>
<td>100.0%</td>
<td>371</td>
</tr>
<tr>
<td>0--0</td>
<td>1</td>
<td>2</td>
<td>3395</td>
<td>99.9%</td>
<td>49</td>
</tr>
<tr>
<td>0--0</td>
<td>2</td>
<td>1</td>
<td>3394</td>
<td>100.0%</td>
<td>270</td>
</tr>
<tr>
<td>0--0</td>
<td>2</td>
<td>2</td>
<td>3384</td>
<td>100.0%</td>
<td>59</td>
</tr>
<tr>
<td>0--0</td>
<td>3</td>
<td>1</td>
<td>3430</td>
<td>99.9%</td>
<td>28</td>
</tr>
<tr>
<td>0--0</td>
<td>3</td>
<td>2</td>
<td>3430</td>
<td>100.0%</td>
<td>91</td>
</tr>
</tbody>
</table>
## LONG AMPLICON ANALYSIS OUTPUT

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fastq Gathered</td>
<td>189,944 bytes</td>
<td>Fastq</td>
</tr>
<tr>
<td>Fastq Gathered</td>
<td>143,864 bytes</td>
<td>Fastq</td>
</tr>
<tr>
<td>CSV</td>
<td>5,990 bytes</td>
<td>csv</td>
</tr>
<tr>
<td>Amplicon consensus report</td>
<td>10,780 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>Master log file</td>
<td>633 bytes</td>
<td>log</td>
</tr>
<tr>
<td>pbsmrtpipe log</td>
<td>0 bytes</td>
<td>log</td>
</tr>
</tbody>
</table>
IsoSeq Updates in SMRTAnalysis 3.x
ISOSEQ PIPELINE CHANGES MADE TO SMRTANALYSIS 3.X

- **BAMification**
  - ~30% boost in overall wall time
- Better memory management
  - Easier to implement on your cluster
  - Enables analysis of large datasets
- Enables new command line tools
  - `collapse_by_sam.py`: helps further collapse redundant transcripts for isoforms aligned to a reference (GMAP)
- “Count” information now provided
ISOSEQ BIOINFORMATICS PIPELINE

1. PacBio raw sequence reads
   - Remove adapters
   - Remove artifacts

2. Clean sequence reads
   - Reads clustering

3. Isoform clusters
   - Consensus calling

4. Nonredundant transcript isoforms
   - Quality filtering

5. Final isoforms
   - Map to reference genome (optional)

Evidenced-based gene models
IDENTIFY YOUR FULL LENGTH TRANSCRIPTS

Full-Length = 5’ primer seen, polyA tail seen, 3’ primer seen

- Identify and remove primers and polyA/T tail
- Identify read strandedness
ITERATIVE CLUSTERING FOR ERROR CORRECTION

1. Full-length, non-chimeric RoI
2. Build Similarity Graph using BLASR/DAiligner
3. Clique Finding
4. Fast Consensus Calling using DAGCon
5. Cluster Reassignment
6. Merge Clusters

Repeat steps 3-6 until convergence.
ARROW REPLACES QUIVER FOR CONSENSUS CALLING

- **Runs (.BAM)**
  - **IsoSeq classify**
    - **Reads Of Insert (Rol) (.fasta, .BAM)**
      - **Full-length, non-chimeric Rol**
        - **ICE**
          - **Cluster Consensus**
      - **Non-full-length, non-chimeric Rol**
        - **Arrow**
          - **HQ, Full-length, Polished Consensus**

**IsoSeq cluster**
ISOSEQ IN SMRTANALYSIS 3.1

Create New Analysis - Settings

Name *

Data Set

dry_C01

Collections in dry_C01

<table>
<thead>
<tr>
<th>Sample Name</th>
<th>Run Name</th>
<th>Completed On</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>dry_C01</td>
<td>Run_06.08.2018 16:24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Analysis Application *

- Assembly (HGAP 4)
- Barcoding
- CCS Mapping
- Circular Consensus Sequences (CCS 2)
- Convert BAM to FASTX
- IsoSeq
- IsoSeq Classify Only
- Long Amplicon Analysis (LAA 2)
- Resequencing
- Site Acceptance Test (SAT)
ISOSEQ IN SMRTANALYSIS 3.1

- This protocol is only for IsoSeq library QC
ISOSEQ IN SMRTANALYSIS 3.1

- Click on Advanced Analysis Parameters
ADVANCED ANALYSIS PARAMETERS

**CCS**
- Minimum Predicted Accuracy: 0.8
- Minimum Number of Passes: 1
- Minimum Z Score: -9999
- Maximum Subread Length: 15000
- By Strand CCS: OFF
- Minimum Subread Length: 300
- Minimum Read Score: 0.65
- Maximum Dropped Fraction: 0.8
- Minimum SNR: 3.75
- No Polish CCS

**IsoSeq**
- Bin by read length manually: OFF
- Bin by primer: OFF
- Minimum Accuracy of polished isoforms: 0.99
- Bin by read length in KB: 1
- Trim QVs 3': 30
- sample Name
- Ignore polyA: OFF
- Trim QVs 5': 100
- Minimum Sequence Length: 300

**CANCEL** **OK**
## ADVANCED ANALYSIS PARAMETERS

### CCS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Predicted Accuracy</td>
<td>0.8</td>
</tr>
<tr>
<td>Minimum Number of Passes</td>
<td>1</td>
</tr>
<tr>
<td>Minimum Z Score</td>
<td>-9999</td>
</tr>
<tr>
<td>Maximum Subread Length</td>
<td>15000</td>
</tr>
<tr>
<td>By Strand CCS</td>
<td>OFF</td>
</tr>
<tr>
<td>Minimum Subread Length</td>
<td>300</td>
</tr>
<tr>
<td>Minimum Read Score</td>
<td>0.65</td>
</tr>
<tr>
<td>Maximum Dropped Fraction</td>
<td>0.8</td>
</tr>
<tr>
<td>Minimum SNR</td>
<td>3.75</td>
</tr>
</tbody>
</table>

- Minimum accuracy for candidate FL
- Minimum # of passes around SMRTbell for candidate FL
- Identifies and removes non agreeing subreads
- Maximum size for candidate FL
- Toggle generating per strand CCS
- Minimum size for candidate FL
- Minimum individual subread accuracy for CCS inclusion
- Highest allowable portion of reads that can be dropped from ZMW
- Minimum acceptable level of signal to noise ratio for a candidate CCS
ADVANCED ANALYSIS PARAMETERS

- Can specify manual bins (min, max, int-int) eg (0,2,4,6,8) or (0,2,6,8)
- Toggle binning by primer sequence (for barcoded isoseq)
- Minimum post Arrow accuracy
- Bin automatically by specified interval size
- Trim bases < X QV from 3’ end of transcript
- Give sample unique name
- Toggle looking for polyA tail
- Trim bases < X QV from 5’ end of transcript
- Minimum length of candidate transcript
<table>
<thead>
<tr>
<th>Analysis Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Analysis</strong></td>
</tr>
<tr>
<td><strong>Analysis Id</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Created At</strong></td>
</tr>
<tr>
<td><strong>Updated At</strong></td>
</tr>
<tr>
<td><strong>Application</strong></td>
</tr>
<tr>
<td><strong>Inputs</strong></td>
</tr>
<tr>
<td><strong>Path</strong></td>
</tr>
</tbody>
</table>

**Analysis Parameters**
CCS Read Length

- Reads
- Mb > Read Length
- Read Length

Attributes
By Movie
CCS Read Length
CCS Read Score
Number of Passes
Number of Passes vs. Read Score
Transcript Classification
Transcript Clustering
Data
# Transcript Classification

<table>
<thead>
<tr>
<th>Value</th>
<th>Analysis Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>191936</td>
<td>Number of consensus reads</td>
</tr>
<tr>
<td>147608</td>
<td>Number of five prime reads</td>
</tr>
<tr>
<td>172365</td>
<td>Number of three prime reads</td>
</tr>
<tr>
<td>157807</td>
<td>Number of poly-A reads</td>
</tr>
<tr>
<td>50</td>
<td>Number of filtered short reads</td>
</tr>
<tr>
<td>57147</td>
<td>Number of non-full-length reads</td>
</tr>
<tr>
<td>134739</td>
<td>Number of full-length reads</td>
</tr>
<tr>
<td>134268</td>
<td>Number of full-length non-chimeric reads</td>
</tr>
<tr>
<td>263329893</td>
<td>Number of full-length non-chimeric bases</td>
</tr>
<tr>
<td>1961</td>
<td>Mean full-length non-chimeric read length</td>
</tr>
</tbody>
</table>
### IsoSeq Transcript Classification

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of consensus reads</td>
<td>191936</td>
</tr>
<tr>
<td>Number of five prime reads</td>
<td>147608</td>
</tr>
<tr>
<td>Number of three prime reads</td>
<td>172365</td>
</tr>
<tr>
<td>Number of poly-A reads</td>
<td>157807</td>
</tr>
<tr>
<td>Number of filtered short reads</td>
<td>50</td>
</tr>
<tr>
<td>Number of non-full-length reads</td>
<td>57147</td>
</tr>
<tr>
<td>Number of full-length non-chimeric reads</td>
<td>134739</td>
</tr>
<tr>
<td>Number of full-length non-chimeric bases</td>
<td>134268</td>
</tr>
<tr>
<td>Number of full-length non-chimeric reads</td>
<td>263329893</td>
</tr>
<tr>
<td>Mean full-length non-chimeric read length</td>
<td>1961</td>
</tr>
</tbody>
</table>
Read Length of Full-length Non-Chimeric Reads
## Transcript Clustering

<table>
<thead>
<tr>
<th>Value</th>
<th>Analysis Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>76340</td>
<td>Number of unpolished consensus isoforms</td>
</tr>
<tr>
<td>7298</td>
<td>Number of polished high-quality isoforms</td>
</tr>
<tr>
<td>69028</td>
<td>Number of polished low-quality isoforms</td>
</tr>
<tr>
<td>1990</td>
<td>Mean unpolished consensus isoforms read length</td>
</tr>
</tbody>
</table>

### Attributes

- Read Length of Consensus Isoforms Reads
<table>
<thead>
<tr>
<th>File Download</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConsensusReadSet</td>
<td>50,020 bytes</td>
<td>ConsensusReadSet</td>
</tr>
<tr>
<td>Text</td>
<td>21,700 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>CCS report</td>
<td>4,336 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>GZIPed FASTQ file(s)</td>
<td>64,658,355 bytes</td>
<td>gzip</td>
</tr>
<tr>
<td>GZIPed FASTA file(s)</td>
<td>64,045,593 bytes</td>
<td>gzip</td>
</tr>
<tr>
<td>JSON</td>
<td>10,804 bytes</td>
<td>json</td>
</tr>
<tr>
<td>CSV</td>
<td>7,338,897 bytes</td>
<td>csv</td>
</tr>
<tr>
<td>ContigSet</td>
<td>15,512,159 bytes</td>
<td>ContigSet</td>
</tr>
<tr>
<td>ContigSet</td>
<td>19,996,732 bytes</td>
<td>ContigSet</td>
</tr>
<tr>
<td>ContigSet</td>
<td>35,088,454 bytes</td>
<td>ContigSet</td>
</tr>
<tr>
<td>Iso-Seq classification report</td>
<td>5,335 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>Pickle file</td>
<td>3,826 bytes</td>
<td>pickle</td>
</tr>
<tr>
<td>ICE CLUSTER CHUNK PICKLE</td>
<td>6,171 bytes</td>
<td>pickle</td>
</tr>
<tr>
<td>ICE PARTIAL CHUNK PICKLE</td>
<td>41,083 bytes</td>
<td>pickle</td>
</tr>
<tr>
<td>ICE POLISH CHUNK PICKLE</td>
<td>7,668 bytes</td>
<td>pickle</td>
</tr>
<tr>
<td>Text</td>
<td>8,325 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>Text</td>
<td>161,142 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>Gather nfi pickles Done Text file</td>
<td>3,681 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>Text</td>
<td>11,130 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>Gather polished isoforms in each bin Done Text file</td>
<td>8,903 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>PacBio ContigSet file</td>
<td>4,551,082 bytes</td>
<td>ContigSet</td>
</tr>
</tbody>
</table>
Assembly with HGAP 4
DE NOVO ASSEMBLY: FINISHED GENOMES WITH > 99.999% ACCURACY ACHIEVED USING ONLY PACBIO® READS

Hierarchical Genome Assembly Process (HGAP)

- Utilizes all PacBio data from single, long-insert library
  - Longest reads for continuity
  - All reads for high consensus accuracy

HGAP.4 FEATURE SET IN SECONDARY 3.X

- FALCON-based, diploid aware
- Works on both haploid and diploid genomes
- Works with RSII or Sequel datasets
- Advanced FALCON settings available
- Automatic parameterization based on genome size bins
- Seed coverage now adjustable
- Faster (uses dazzler components DALIGNER and DAZZ_DB)
- New Pre-Assembly statistics
- Polishing with Arrow (Sequel) or Quiver (RSII)
PRE-ASSEMBLY STATISTICS

Tracking yield loss helps quickly diagnose where things may be going wrong. Use pre-assembly information to gain insight into what’s happening.

Raw Input
~60x

Seed Reads
30x (default)

Increase to get more pre-assembled reads for assembly, though too much can fragment assemblies.

Pre-Assembly Reads
15x (minimum)

25x

Lower coverage and/or short N50 values indicates poor input read quality.

Target coverage within this range usually gives the best results.
HGAP4 IS BASED ON PACBIO’S FALCON ASSEMBLER

- Employs Gene Meyers’ daligner, LASort, and LAMerge for rapid alignment of raw PacBio data

- Optionally preserves alternative haplotype information as associated contigs

- Streamlined execution compared to FALCON

- Optimized for large genome assembly

- Consensus polishing with Quiver or Arrow included!
FALCON ASSEMBLY

Total 7 contigs. Contigs break at the branch points. Long range information is lost.

Primary contig

1 full length contig + 2 associated contigs. Maintain long range information at the expense of contig homogeneity.
Phased Diploid Genome Assembly with Single Molecule Real-Time Sequencing

Chen-Shan Chin1,4, Paul Pelosi1,2, Fritz J. Sedlazeck1, Maria Nattestad3, Gregory T. Concepcion4, Alicia Clum5, Christopher Dunn1, Ronan O’Malley5, Rosa Figueroa-Balderas6, Abraham Morales-Cruz6, Grant R. Cramer4, Massimo Delledonne4, Chongyuan Luo3, Joseph R. Ecker5, Dario Cantu7, David R. Rank1, Michael C. Schatz2,8

1 Pacific Biosciences, Menlo Park, CA 94025, USA
2 Departments of Computer Science and Biology, Johns Hopkins University, Baltimore, MD 21211, USA
3 Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724, USA
4 DOE Joint Genome Institute, Walnut Creek, California 94598, USA
5 Genomic Analysis Laboratory, The Salk Institute for Biological Studies, La Jolla, CA 92037, USA.
6 Department of Viticulture and Enology, University of California Davis, CA 95616, USA
7 Department of Biochemistry and Molecular Biology, University of Nevada, Reno, NV 89557, USA
8 Dipartimento di Biotecnologie, Università degli Studi di Verona, 37134 Verona, Italy

1 These authors contribute equally to this work. *Corresponding authors

Abstract

While genome assembly projects have been successful in a number of haploid or inbred species, one of the current main challenges is assembling non-Inbred or rearranged heterozygous genomes. To address this critical need, we introduce the open-source FALCON and FALCON-Unspl algorithms (https://github.com/PacificBiosciences/FALCON/) to assemble Single Molecule Real-Time SMRT Sequencing data into highly accurate, contiguous, and correctly phased diploid genomes. We demonstrate the quality of this approach by assembling new reference sequences for three heterozygous samples, including an F1 hybrid of the model species Arabidopsis thaliana, the widely cultivated V. vinifera cv. Cabernet Sauvignon, and the coral fungus Clavicipita pyxidata that have challenged short-read assembly approaches. The FALCON-based assemblies were substantially more contiguous and complete than alternate short or long-read approaches. The phased diploid assembly enabled the study of haplotype structures and heterozygosities between the homologous chromosomes, including identifying widespread heterozygous structural variations within the coding sequences.
Specify the approximate expected genome length, and click “Start”

Note: Only REQUIRED parameters are revealed initially. Click on “Advanced Analysis Parameters” to reveal all available options.
Filtering settings

BLASR settings (for resequencing)

Assembly settings (currently masked)

Resequencing settings
FILTER SETTINGS

- Minimum Subread Length
  - As before, self-explanatory

- Filter to add to the DataSet
  - May elect to add additional filters based on fields of the BAM file. See PacBio BAM file format specification for information on these fields.
PARAMETERIZATION

---

Access to lower-level configuration parameters

- Exposes power-users to any and all FALCON parameters via GUI
- Overrides are simple key/value pairs separated by semicolon, newlines ignored
- DALIGNER/DAZZ_DB options may also be passed
- Links to information and theory behind override options found in help section.
- Simple, high-level settings like seed length cutoff and coverage available separately (most people will use these)
INCREASING SPECIFICITY IN DALIGNER FOR FASTER ASSEMBLY

- Defaults: -l1000 -k14 -h35 -w6 -t14
- Specific: -l4800 -k18 -h480 -w8 -t16 -M32

- \( l \) specifies the minimum size of a match that gets recorded
- \( k \) specifies the minimum size of an exact-match \( k \)-mer
- \( h \) specifies the number of bases in the \( l \) window that must be covered by exact-matching \( k \)-mers
- \( w \) specifies the width of diagonal bands searched for exact-match \( k \)-mers as \( 2^w \)
- \( M \) specifies the memory usage per job
ALIGNMENT PARAMETERS

Defaults seldom need changing here, much like HGAP2 or HGAP3

Navy blue = selected

Concordant alignment aligns all subreads from one ZMW to the same position in the assembly

“Randombest” aligns reads to the best scoring position, and randomly between ties for the best score
CONSENSUS OPTIONS

- Minimum confidence is the QV score necessary to make a variant call.

- May use quiver algorithm only for RSII data.

- Diploid mode supplies Falcons alternate contigs.

- Other parameters referring to tracks and reports and such are placeholders for when additional reports are added to the output in future versions.
HGAP Reports
Overview
<table>
<thead>
<tr>
<th>Analysis</th>
<th>SMS_3150239_B01_2xSS_tc6_Sym_15kEcoli_HGAP-default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis Id</td>
<td>11400</td>
</tr>
<tr>
<td>Status</td>
<td>SUCCESSFUL</td>
</tr>
<tr>
<td>Created At</td>
<td>6/1/2016, 3:22:58 PM</td>
</tr>
<tr>
<td>Updated At</td>
<td>6/1/2016, 3:22:58 PM</td>
</tr>
<tr>
<td>Application</td>
<td>Assembly (HGAP 4)</td>
</tr>
<tr>
<td>Inputs</td>
<td>BAM Data</td>
</tr>
<tr>
<td>Path</td>
<td>/pbi/dep/secondary/siw/smrtlink/smrtlink-beta/smrtsuite_166987/usrdata/jobs_root/011/011400</td>
</tr>
</tbody>
</table>

**Analysis Parameters**
### Analysis Results - SMS_3150239_B01_2xSS_tcs6_Sym_15kEcoli_HGAP-default

#### Preassembly

<table>
<thead>
<tr>
<th>Value</th>
<th>Analysis Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000000</td>
<td>Genome Length (user input)</td>
</tr>
<tr>
<td>56207</td>
<td>Number of Raw Reads</td>
</tr>
<tr>
<td>5327.367086432903</td>
<td>Raw Read Length Mean</td>
</tr>
<tr>
<td>9067</td>
<td>Raw Read Length (N50)</td>
</tr>
<tr>
<td>16546</td>
<td>Raw Read Length 95%</td>
</tr>
<tr>
<td>310900050</td>
<td>Number of Raw Bases (total)</td>
</tr>
<tr>
<td>62.0180112</td>
<td>Raw Coverage (bases/genome_size)</td>
</tr>
<tr>
<td>9102</td>
<td>Length Cutoff (user input or auto-calc)</td>
</tr>
<tr>
<td>10380</td>
<td>Number of Seed Reads</td>
</tr>
<tr>
<td>14451.192581986246</td>
<td>Seed Read Length Mean</td>
</tr>
<tr>
<td>15109</td>
<td>Seed Read Length (N50)</td>
</tr>
<tr>
<td>22507</td>
<td>Seed Read Length 95%</td>
</tr>
<tr>
<td>150003379</td>
<td>Number of Seed Bases (total)</td>
</tr>
<tr>
<td>30.0006758</td>
<td>Seed Coverage (bases/genome_size)</td>
</tr>
<tr>
<td>18324</td>
<td>Number of Pre-Assembled Reads</td>
</tr>
<tr>
<td>5104.426544422615</td>
<td>Pre-Assembled Read Length Mean</td>
</tr>
<tr>
<td>7262</td>
<td>Pre-Assembled Read Length (N50)</td>
</tr>
<tr>
<td>12458</td>
<td>Pre-Assembled Read Length 95%</td>
</tr>
<tr>
<td>93533512</td>
<td>Number of Pre-Assembled Bases (total)</td>
</tr>
<tr>
<td>18.7067024</td>
<td>Pre-Assembled Coverage (bases/genome_size)</td>
</tr>
<tr>
<td>62.354%</td>
<td>Pre-Assembled Yield (bases/seed_bases)</td>
</tr>
</tbody>
</table>
Contig Coverage vs Confidence

![Contig Coverage vs Confidence graph]

- Mean Confidence (QV) vs Mean Coverage Depth

Attributes
- Contig Coverage vs Confidence
- Data
### Analysis Results - SMS_3150239_B01_2xSS_tc6_Sym_15kEcoli_HGAP-default

#### File Downloads

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FASTA file</td>
<td>317,601,628 bytes</td>
<td>Fasta</td>
</tr>
<tr>
<td>FOFN file</td>
<td>151 bytes</td>
<td>generic_fofn</td>
</tr>
<tr>
<td>INIT file</td>
<td>605 bytes</td>
<td>cfg</td>
</tr>
<tr>
<td>JSON file</td>
<td>1,822 bytes</td>
<td>json</td>
</tr>
<tr>
<td>Input file list</td>
<td>151 bytes</td>
<td>generic_fofn</td>
</tr>
<tr>
<td>Shell script</td>
<td>3,331 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>Status file</td>
<td>0 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>FOFN file</td>
<td>6,004 bytes</td>
<td>generic_fofn</td>
</tr>
<tr>
<td>Input file list</td>
<td>306 bytes</td>
<td>generic_fofn</td>
</tr>
<tr>
<td>Shell script</td>
<td>704 bytes</td>
<td>txt</td>
</tr>
<tr>
<td>Preassembly report</td>
<td>3,468 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>FASTA sequences</td>
<td>4,717,719 bytes</td>
<td>Fasta</td>
</tr>
<tr>
<td>ReferenceSet XML</td>
<td>1,700 bytes</td>
<td>ReferenceSet</td>
</tr>
<tr>
<td>AlignmentSet</td>
<td>40,472 bytes</td>
<td>AlignmentSet</td>
</tr>
<tr>
<td>Alignment Summary GFF</td>
<td>76,237 bytes</td>
<td>gff</td>
</tr>
<tr>
<td>Fastq Gathered</td>
<td>9,287,125 bytes</td>
<td>Fastq</td>
</tr>
<tr>
<td>Gff</td>
<td>701,195 bytes</td>
<td>gff</td>
</tr>
<tr>
<td>ContigSet</td>
<td>16,316 bytes</td>
<td>ContigSet</td>
</tr>
<tr>
<td>Polished Assembly Report</td>
<td>1,368 bytes</td>
<td>JsonReport</td>
</tr>
<tr>
<td>BED file</td>
<td>370,666 bytes</td>
<td>bed</td>
</tr>
<tr>
<td>VCF file</td>
<td>2,443 bytes</td>
<td>vcf</td>
</tr>
</tbody>
</table>
HGAP4 on the Command Line

Out with smrtpipe.py, in with pbsmrtpipe
The pbsmrtpipe command is designed to be more-or-less self-documenting. It is normally run in one of several modes, which is specified as a positional argument. To get the full overview of functionality, run this:

```
$ pbsmrtpipe --help
```

and you can get further help for individual modes, for example:

```
$ pbsmrtpipe pipeline-id --help
```

To display a list of available pipelines, use `show-templates`:

```
$ pbsmrtpipe show-templates
```

For details about a specific pipeline, specify the ID (the last field in each item in the output of `show-templates`) with `show-template-details`:

```
$ pbsmrtpipe show-template-details
  pbsmrtpipe.pipelines.polished_falcon_fat
```
OUT WITH THE INPUT.XML, IN WITH THE ENTRY POINTS

pbsmrtpipe operates in terms of “entry points.” Each task in the pipeline has its own entry points, and its outputs are bound to the entry points of the next task.

```
$ pbsmrtpipe pipeline-id \
pbsmrtpipe.pipelines.polished_falcon_fat \
-e eid_subread:subreads_bam/subreadset.xml \
--preset-xml=preset.xml
```

Entry points are most typically the xml files of DataSets, such as SubreadSets or ReferenceSets
OUT WITH THE SETTINGS.XML, IN WITH THE PRESET.XML

The preset.xml structure is quite a bit simpler than settings.xml, but the simplest route to a reusable template XML file is still from a job in SMRT Link.
DIRECTORY STRUCTURE HAS CHANGED FROM 2.3

- logs/ is as before

- html/ has replaced reports/

- workflow/ now only contains the workflow diagrams, not any scripts

- tasks/ now contains all scripts and data output
Falcon directories are chunks of 0-rawreads/ 1-preads-ovlp/ and 2-asm-falcon/ like a Falcon job.

pbalign and genomic_consensus directories are from the subsequent Quiver polishing.

EACH TASK GIVEN ITS OWN DIRECTORY FOR SCRIPTS AND OUTPUT