SciLifeLab

NBIS
NATIONAL BIOINFORMATICS INFRASTRUCTURE SWEDEN

Long read Assembly

elixir excelerate

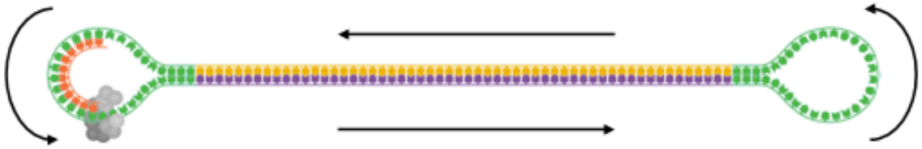
Outline

SciLifeLab

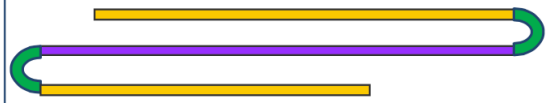
- The main protocols for long read technologies
- The error profile of long reads
- Methods of read correction
  - Correction with short reads
  - Correction using long reads
- Assembly Tools
- Assembly Diagnostics
- Assembly Polishing

Pacific Biosciences

SciLifeLab



SMRTbell™ Template



Polymerase Read

Definition:
- Sequence of nucleotides incorporated by polymerase while reading a template
- Includes adapters
- Often called “read”
- Includes adapters
- 1 molecule, 1 pol. read

Purpose:
- QC of instrument run
- Benchmarking

Subread

Definition:
- Single pass of template
- Adapters removed
- 1 molecule, ≥1 subreads

Unique data:
- Kinetic measurements
- Rich QVs

Purpose:
- For subsequent analysis

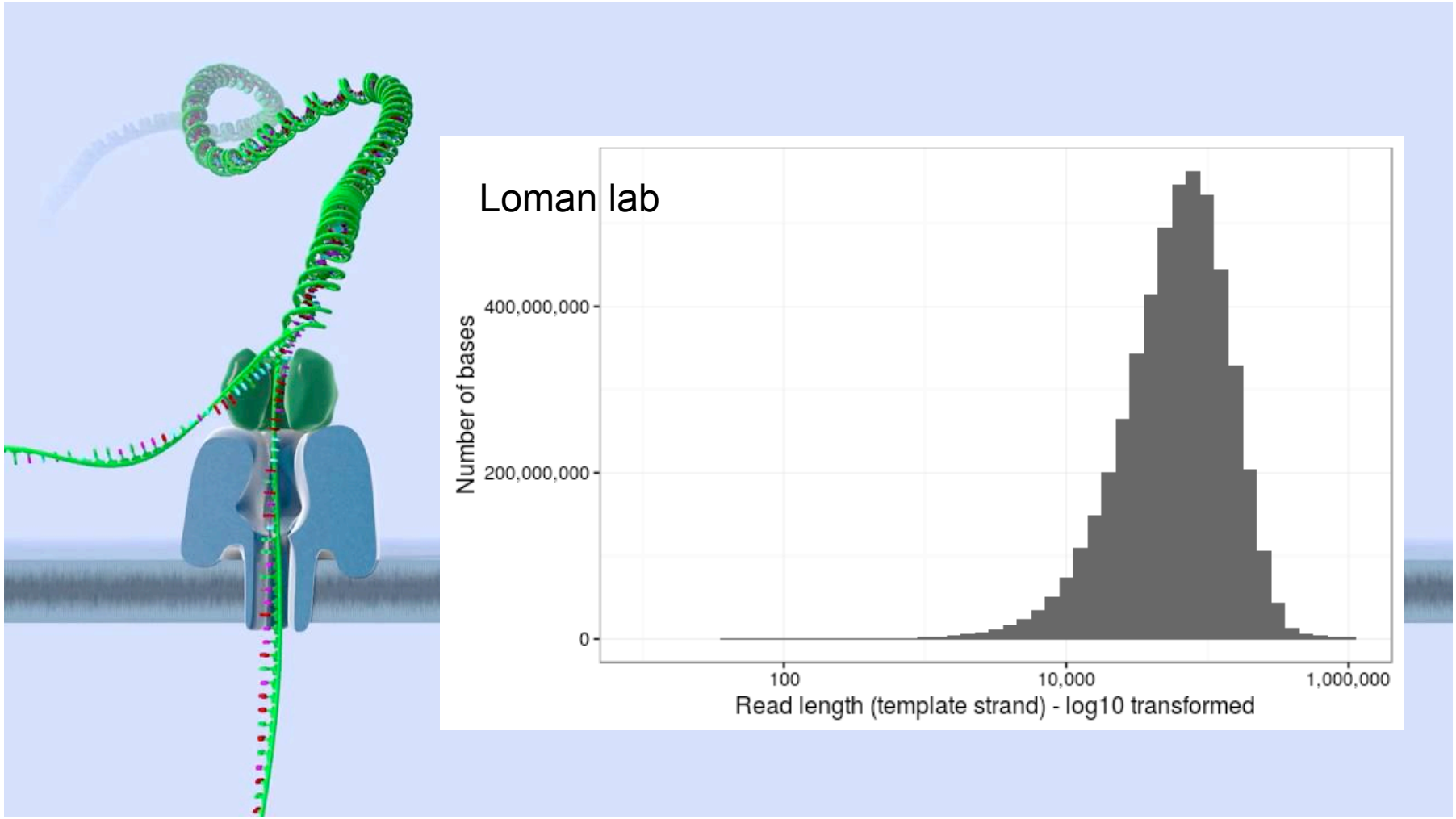
Read of Insert

Definition:
- Represents highest-quality single-sequence for an insert, regardless of number of passes
- Generalizes CCS for <2 passes and RQ <0.9
- 1 or more passes
- 1 molecule, 1 read

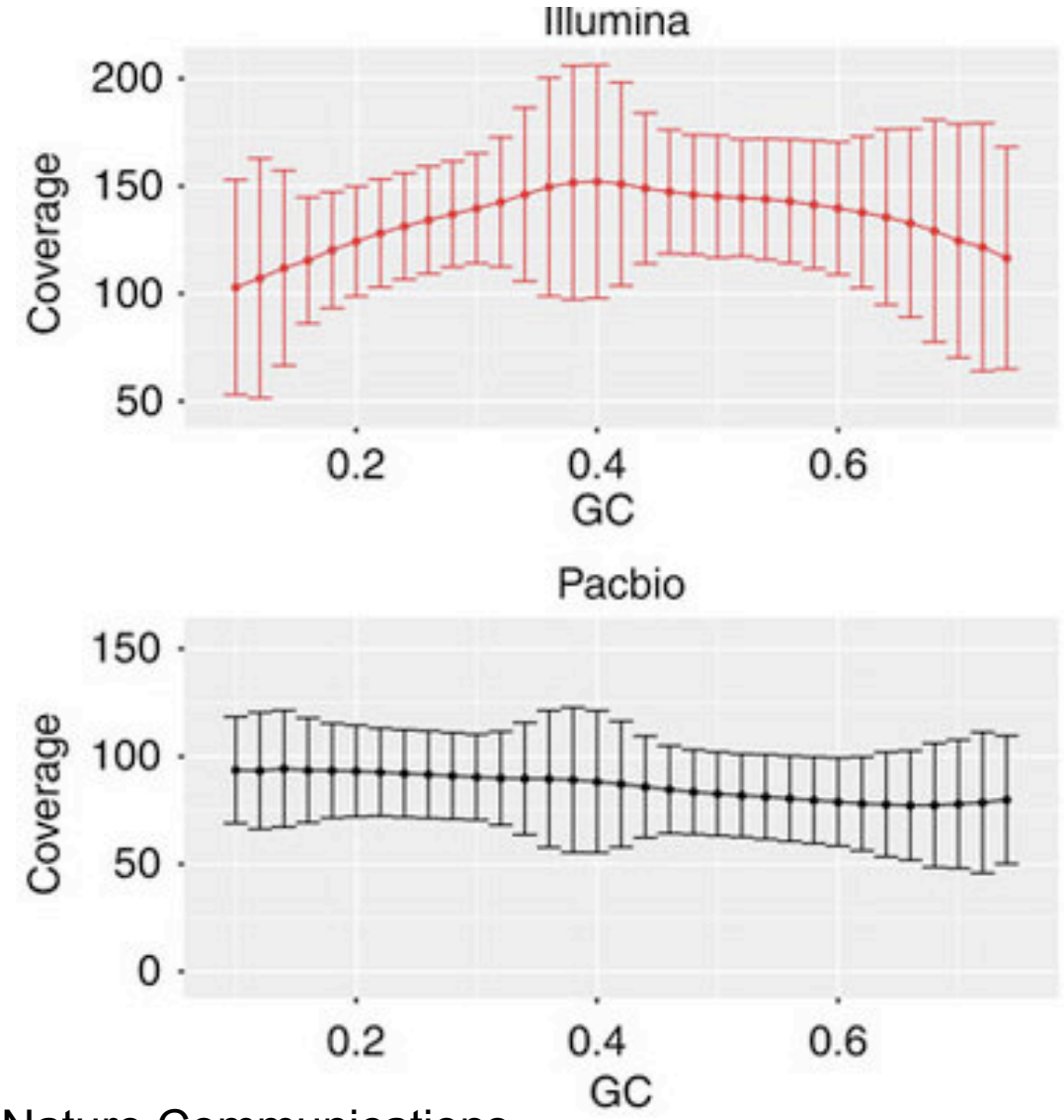
Purpose:
- For Library QC
- For subsequent analysis

Oxford Nanopore

SciLifeLab

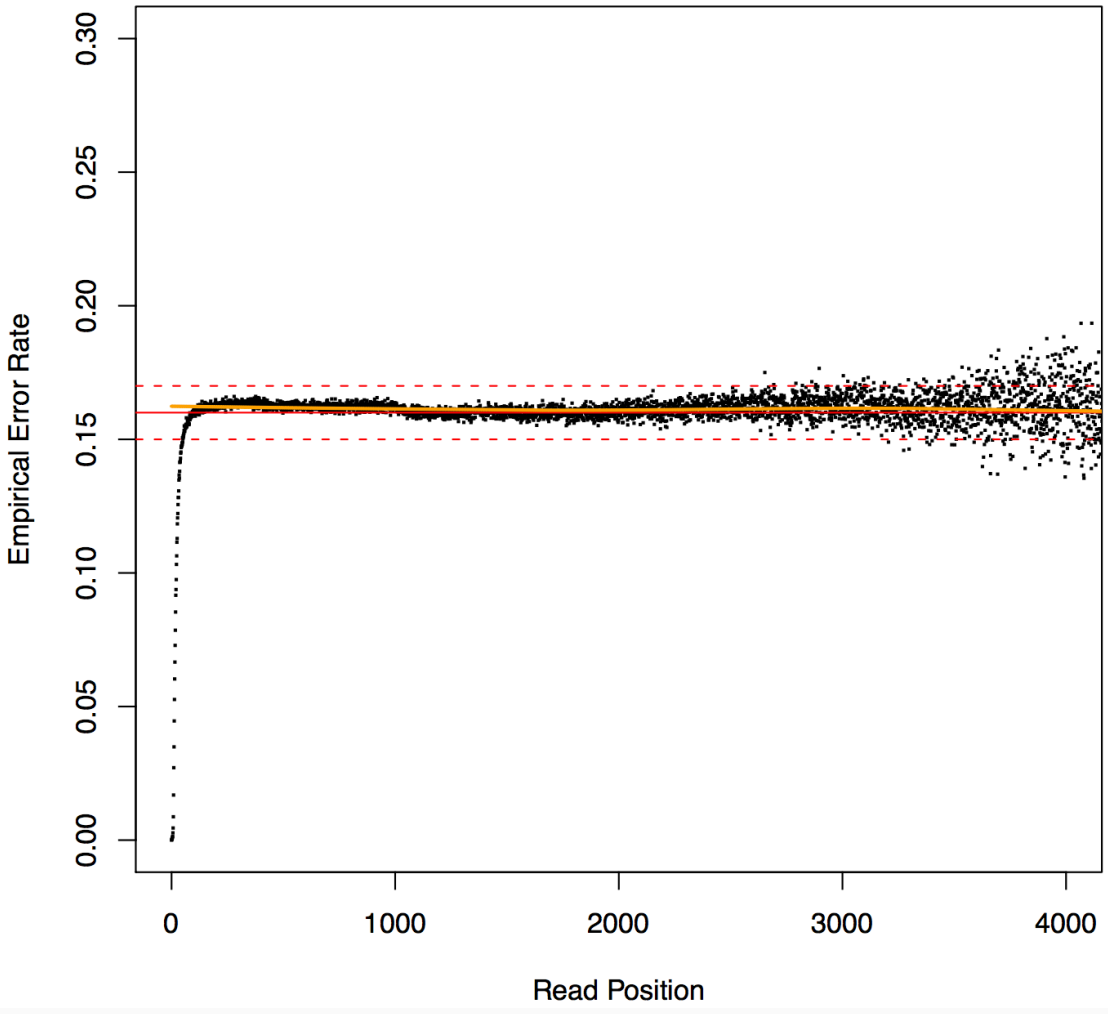


Short read GC Bias



Shi et al. 2016. Nature Communications.

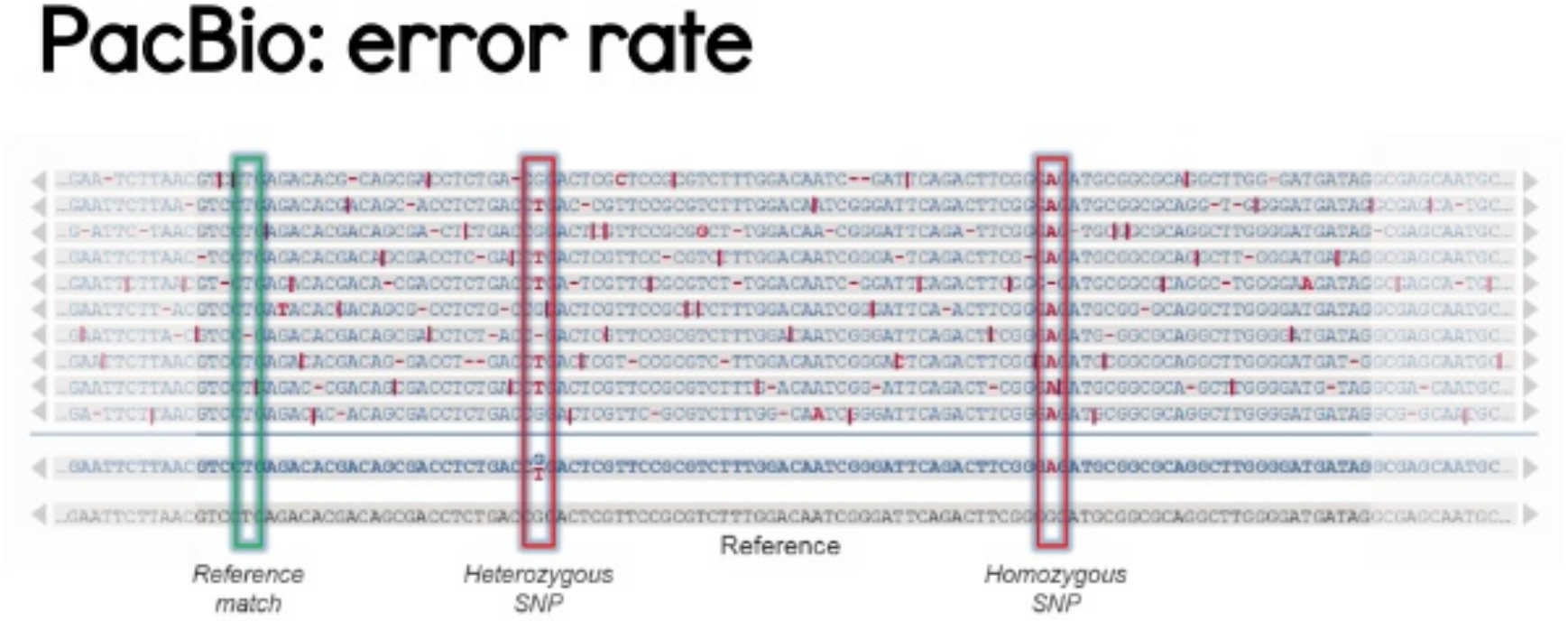
PacBio Error Profile



Koren et al. 2012. Nature Biotechnology.

PacBio Error Profile

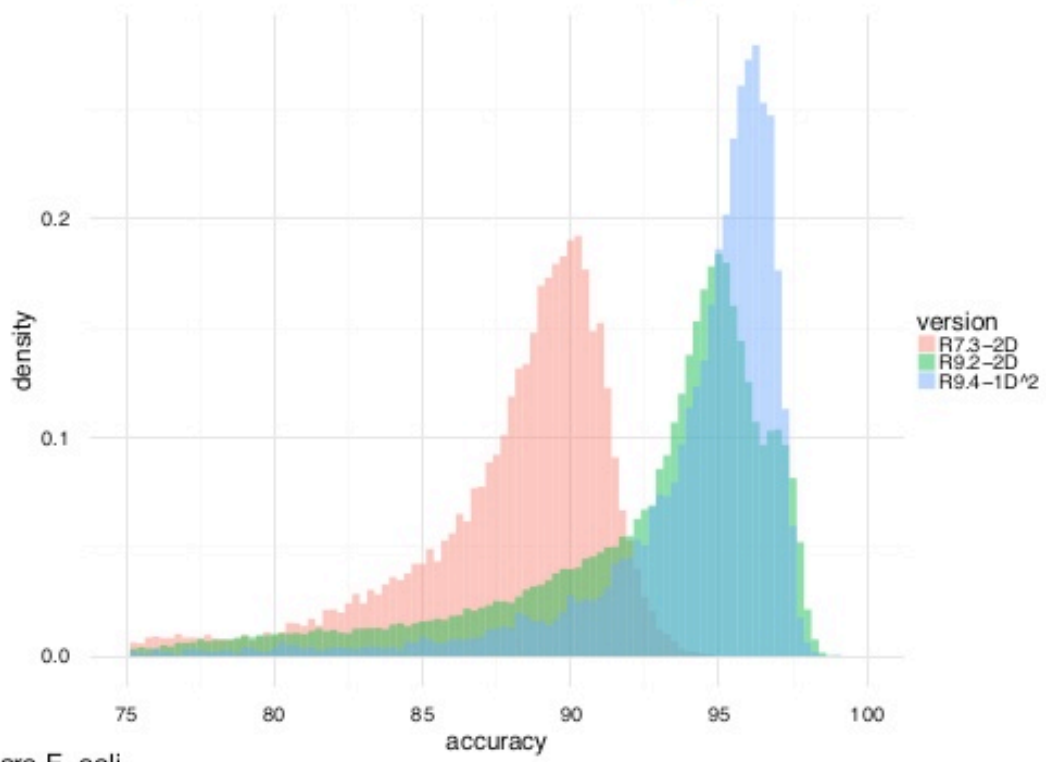
SciLifeLab



T. Seemann. 2015: slideshare.net: Long read sequencing

Nanopore error rate

SciLifeLab

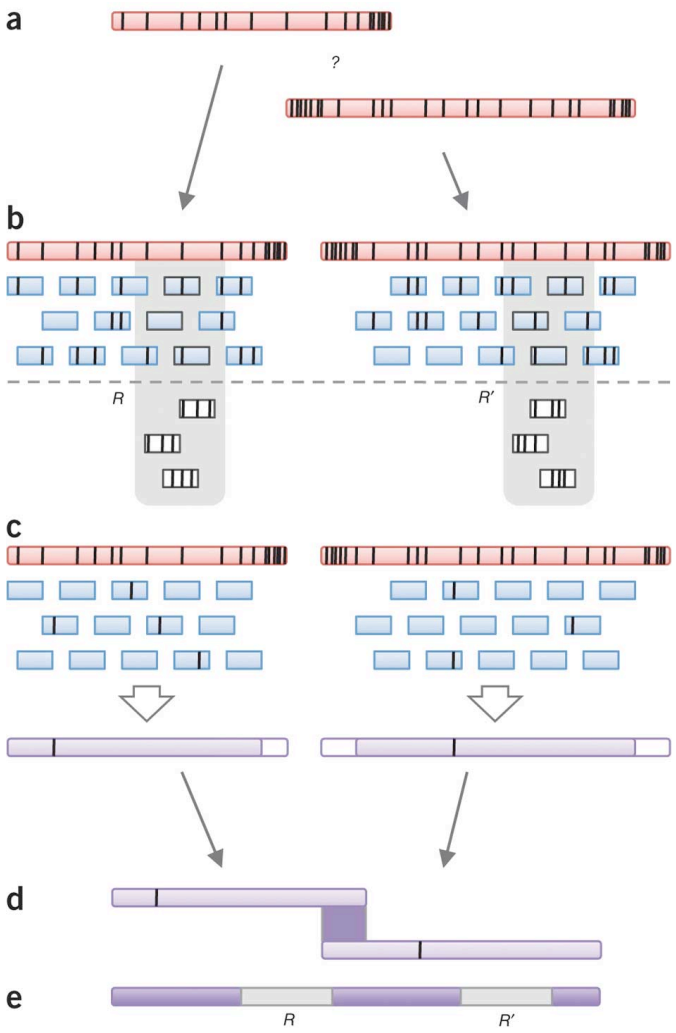


- All runs are E. coli
- R7.3-2D from Nick Loman
- R9.2-2D from OICR
- R9.4-1D² provided by ONT

26

Jared Simpson: AGBT presentation

Read Correction



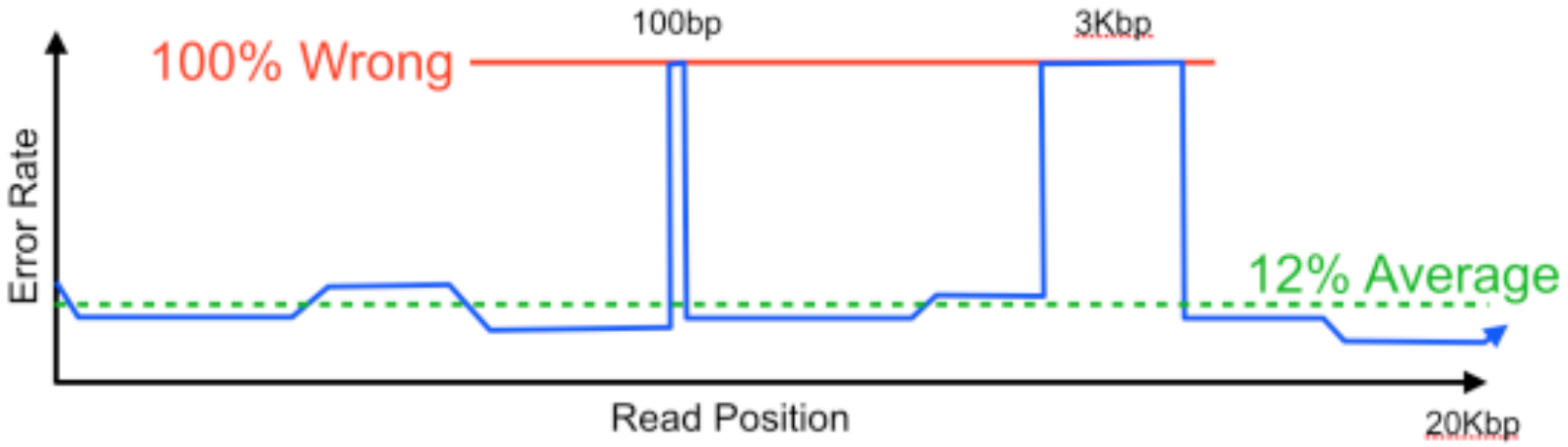
- Correction using Illumina reads
- Homopolymer correction, point mutations, and indels
- Doesn't correct structural errors
- Tools
  - PBcR / PacBioToCA
  - LSC / LSCplus
  - LoRDEC (de Bruijn graph)
  - Proovread
  - ECTools
  - Jabba (de Bruijn graph)

Koren et al. 2012. Nature Biotechnology.

Read Correction

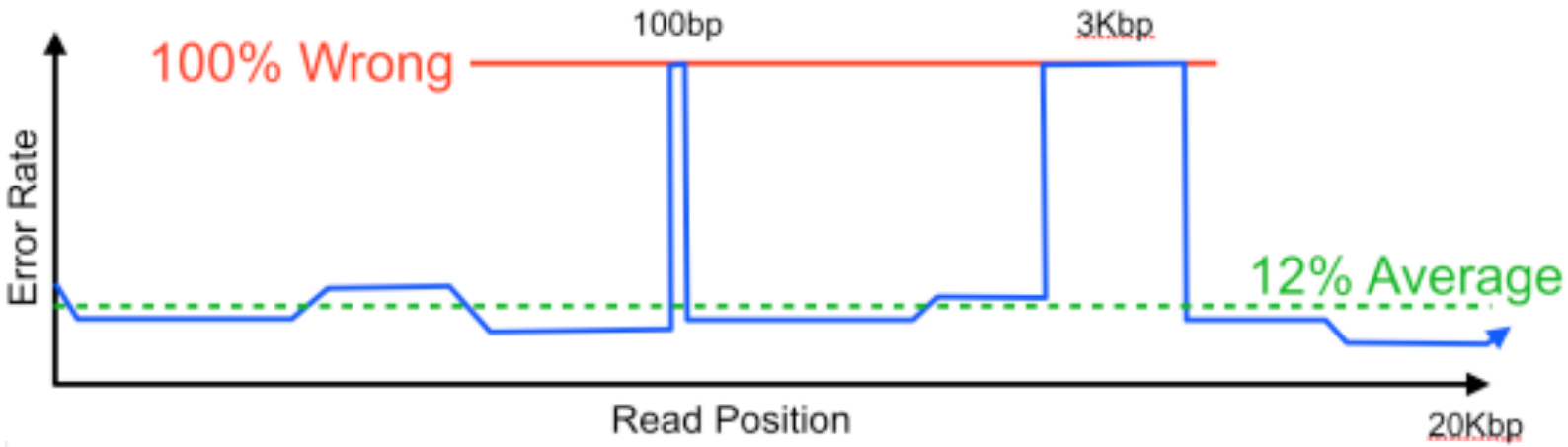
SciLifeLab

- Structural errors
  - Chimeric reads (see Tallon et al. 2014. BMC Genomics)
  - Missed or incorrectly inferred adapter
  - Interference from other molecules
  - …

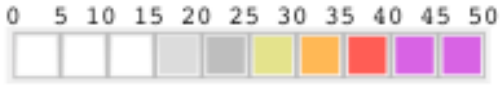


G. Myers. 2015. Dazzlerblog.wordpress.com. Intrinsic Quality Values

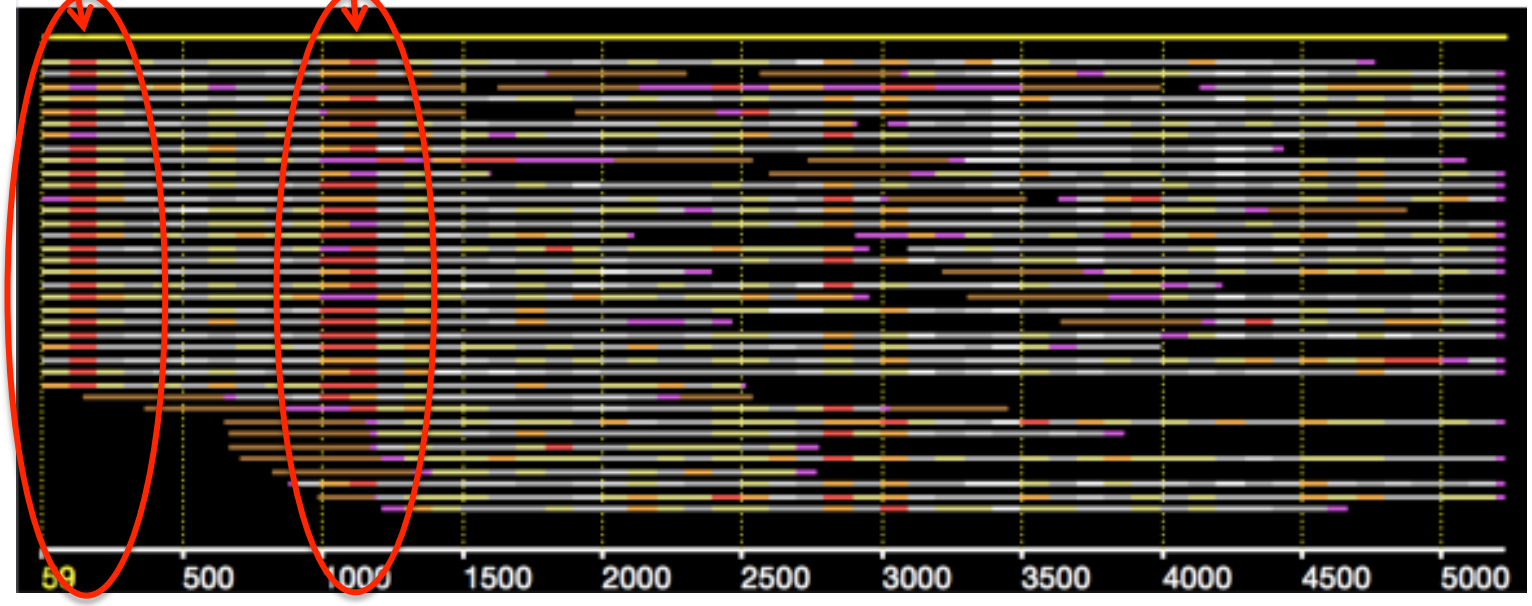
Read Correction



Likely wrong sequence



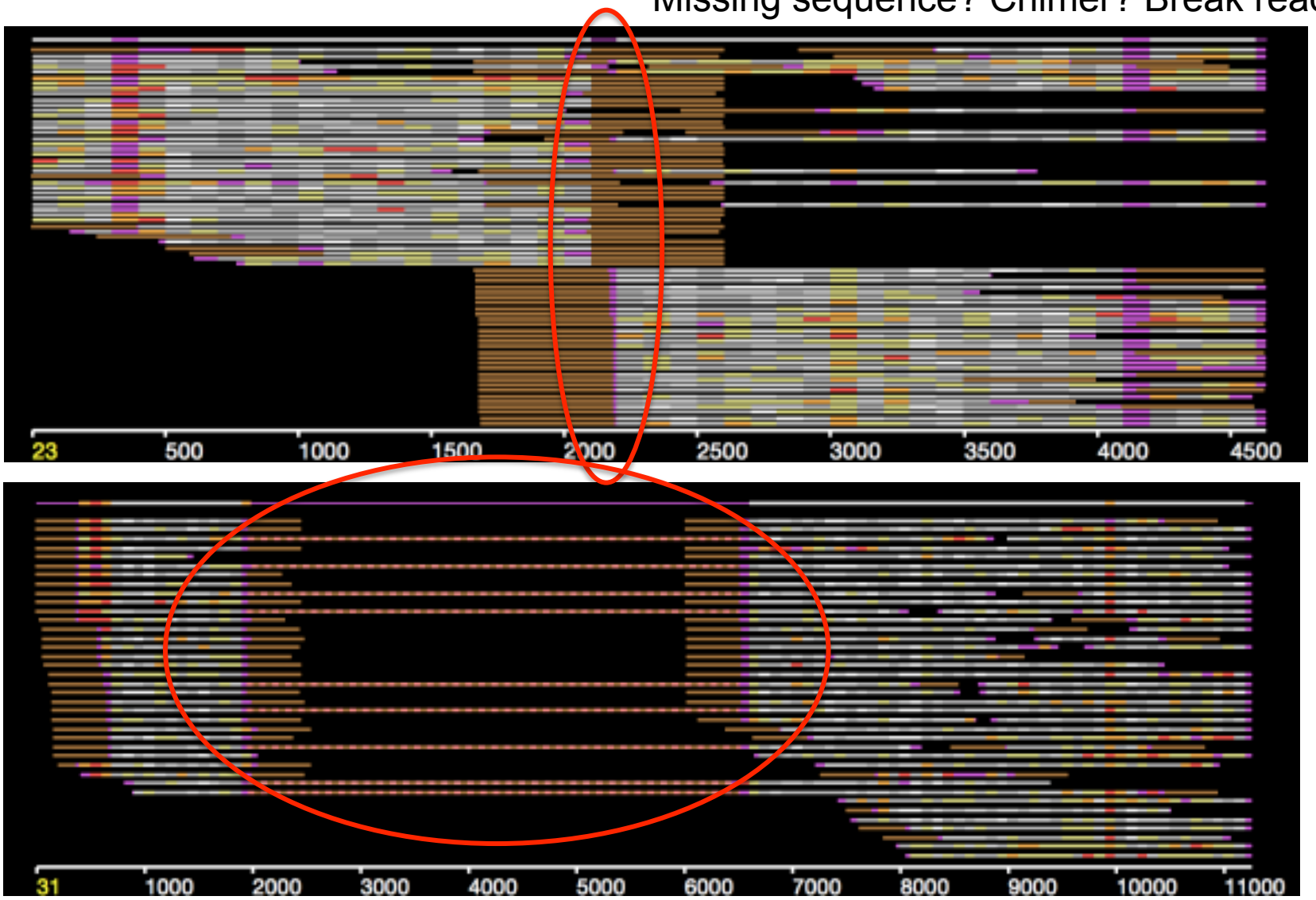
Differences per 100bp segment



Read Correction

SciLifeLab

Missing sequence? Chimer? Break read

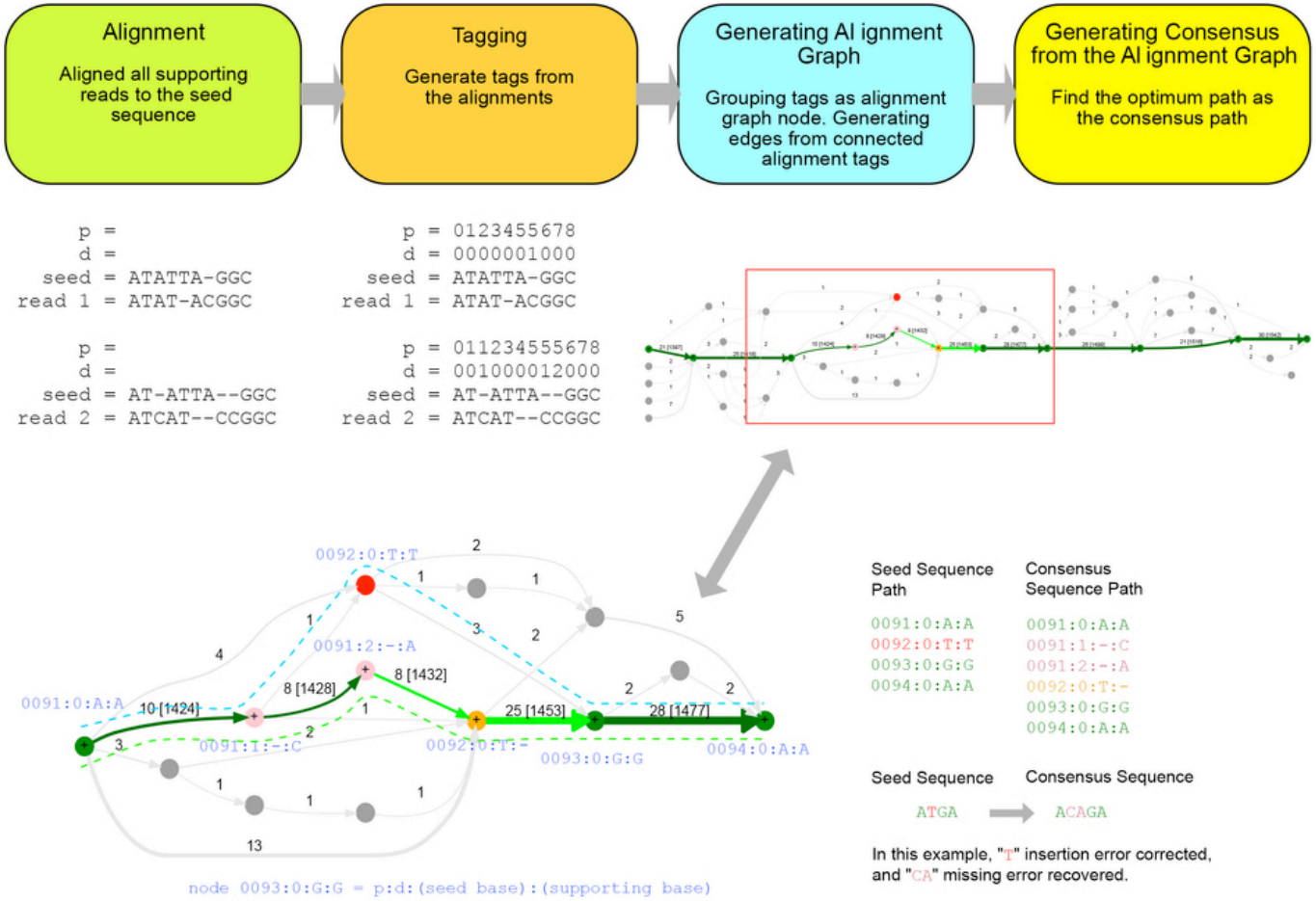


Inserted sequence from Chimer?

Read Correction

SciLifeLab

- Use a weighted directed acyclic graph to find consensus sequence



Supplementary Figure 12

An Example of how the FALCON-sense algorithm generates consensus sequence.

Chin et al. 2016. Nature Methods

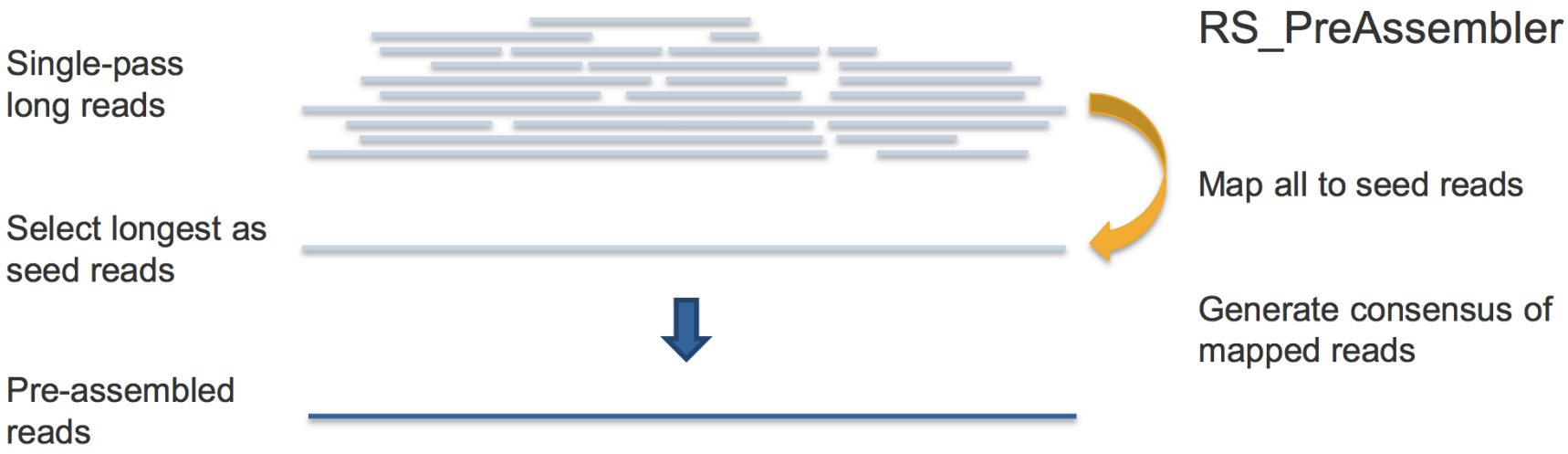
Read Assembly

SciLifeLab

- Popular long read assemblers:
  - HGAP (PacBio)
    - Limited to genomes < 200MB
    - http://www.pacb.com/support/software-downloads/
  - Canu (PacBio, Nanopore)
    - Any size genome
    - https://github.com/marbl/canu
  - Falcon (PacBio, Nanopore?)
    - Small and medium sized genomes, Diploid aware
    - http://pb-falcon.readthedocs.io/en/latest/
  - Miniasm (PacBio, Nanopore)
    - Any size genomes
    - https://github.com/lh3/miniasm

HGAP

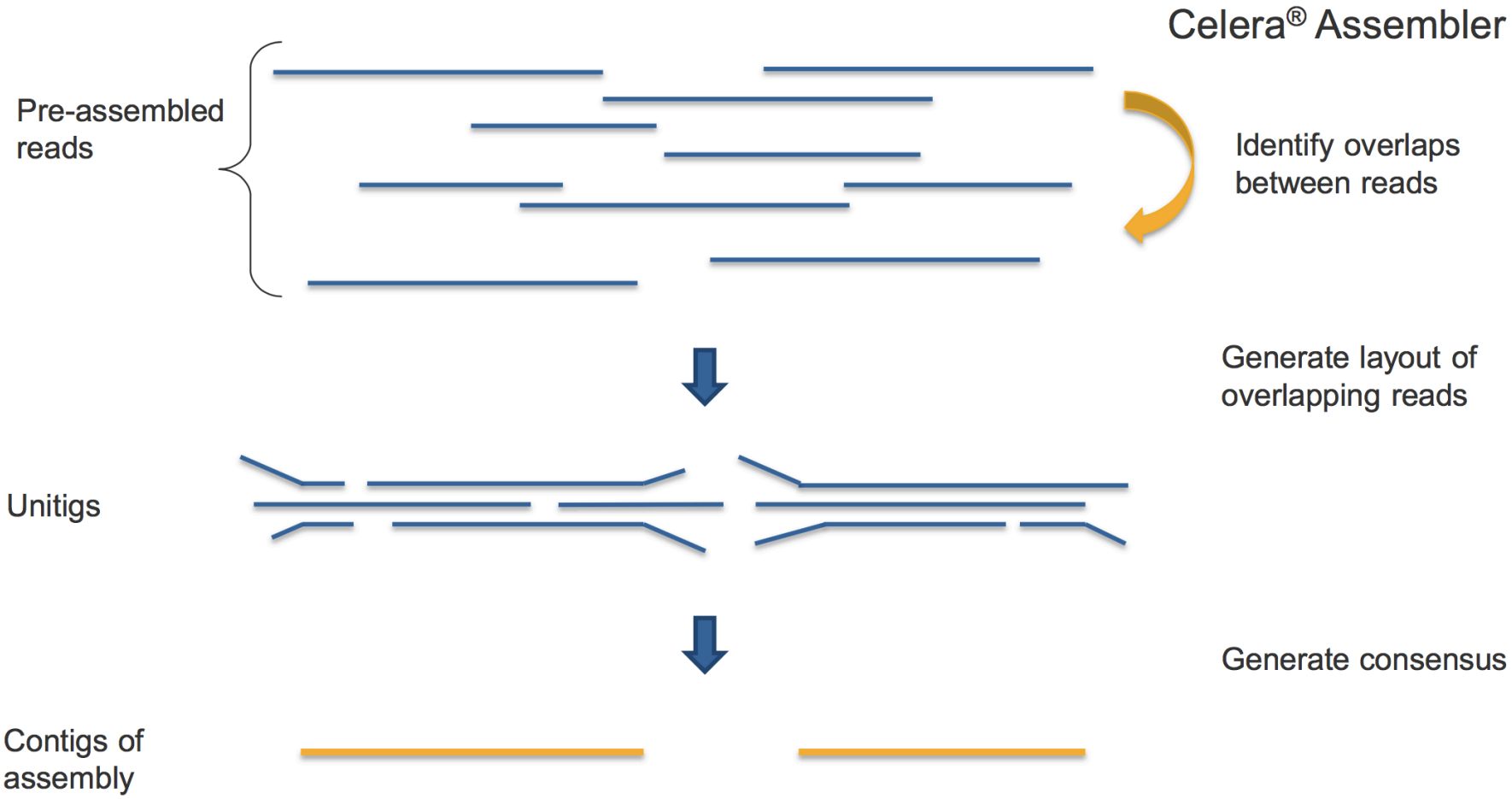
SciLifeLab



Slides from PacBio: BFX_Workshop_Oct_2014_Final-lh.pdf

HGAP

SciLifeLab



HGAP

SciLifeLab

Single-pass long reads

Contigs

High-quality consensus

RS_Resequencing

Map to *de novo*-assembled reference

Base-quality-aware consensus of uniquely mapped reads

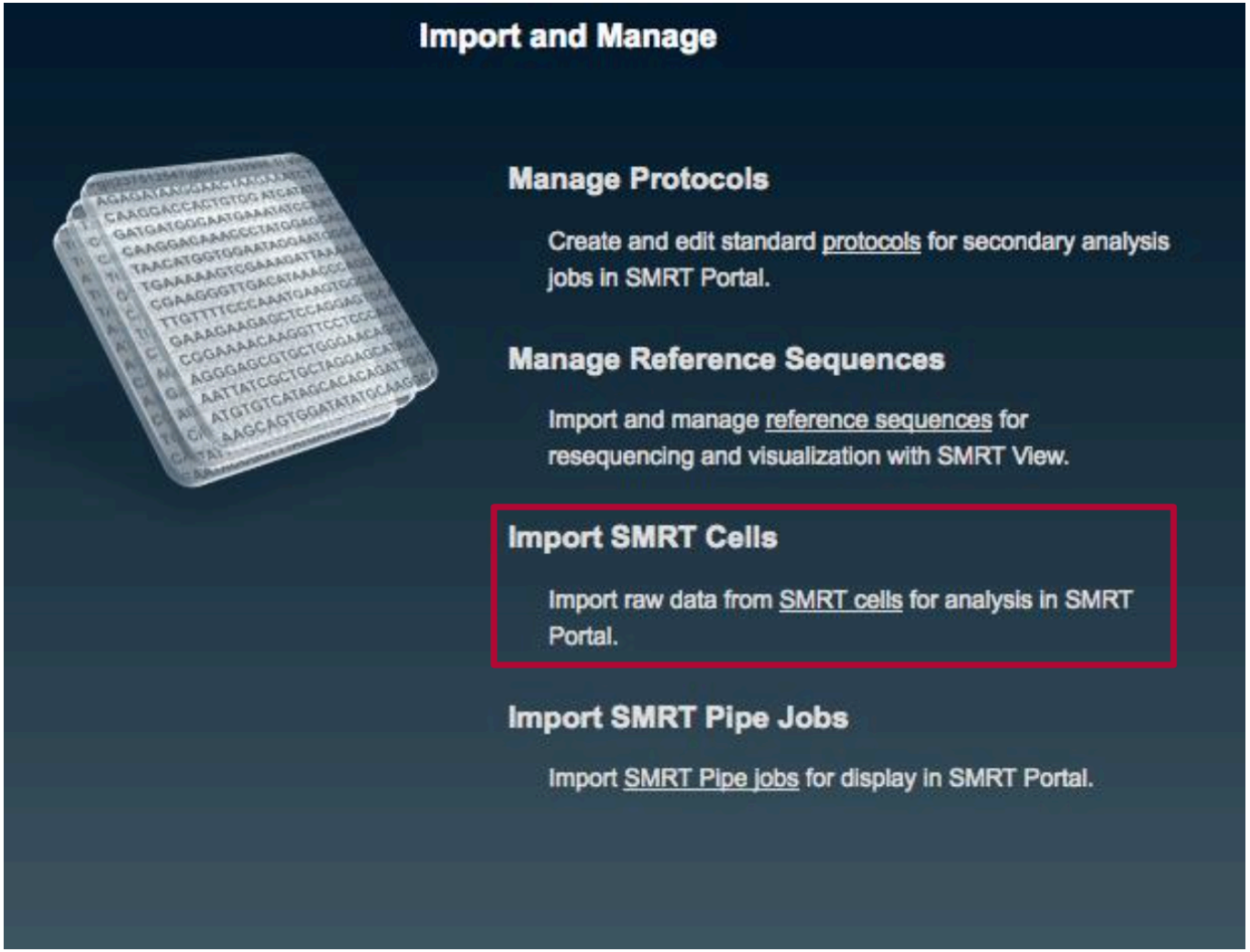
Quiver

HGAP


SMRT® Portal
Home
Admin
Help
About
Welcome, administrator!
Account
Log Off
DESIGN JOB
MONITOR JOBS
VIEW DATA
Open Existing
Create New
Import and Manage
RECENT JOBS
Job Name
Protocol
Reference Sequence
Started
Status
User
ugmExampleA_reseq
RS_Resequencing.1
ugmExampleA
Completed
kluong
NHGRI10_4cells_tandemArt
HGAP_Assembly_Artifact.1
Completed
kluong

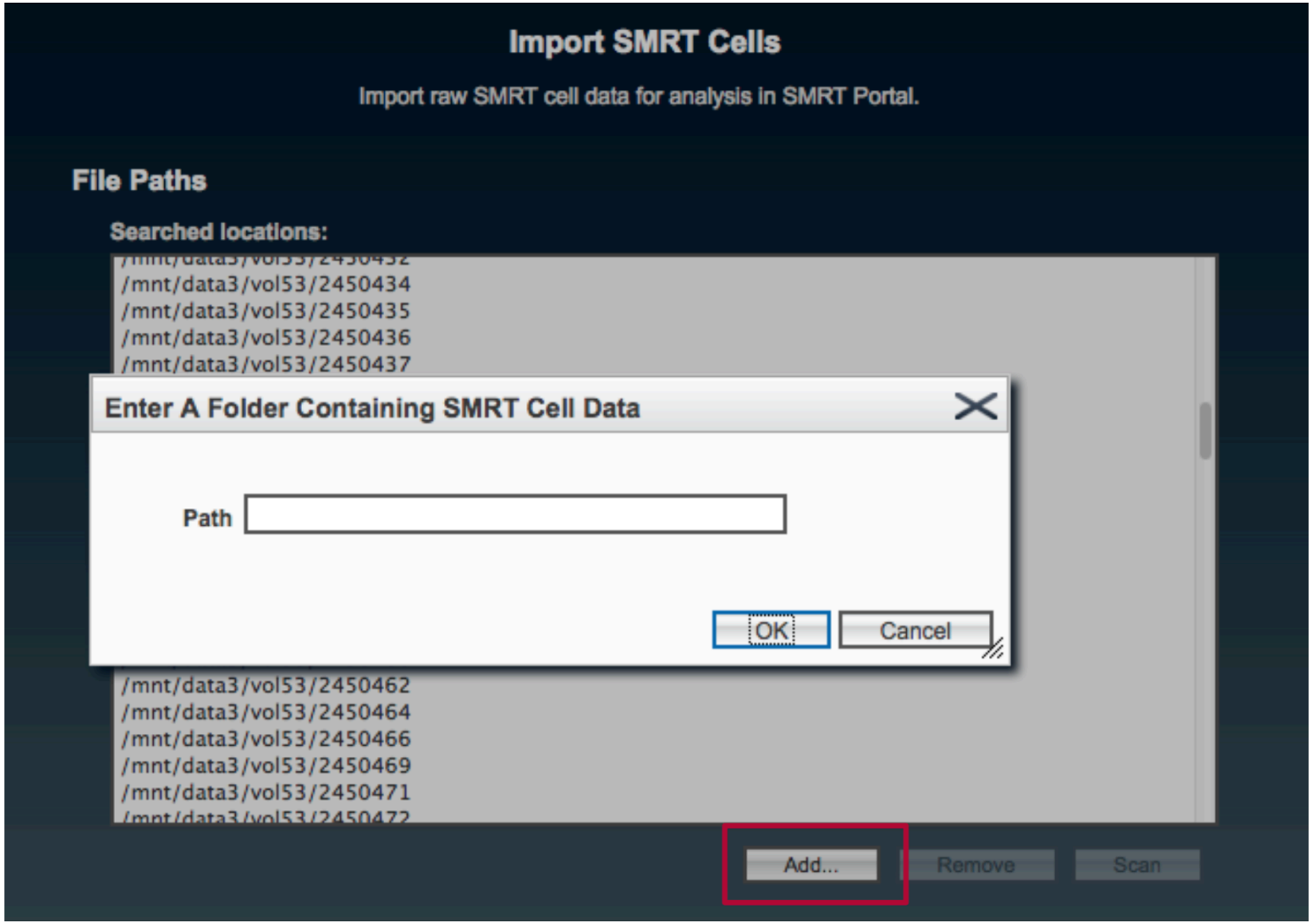
HGAP

SciLifeLab



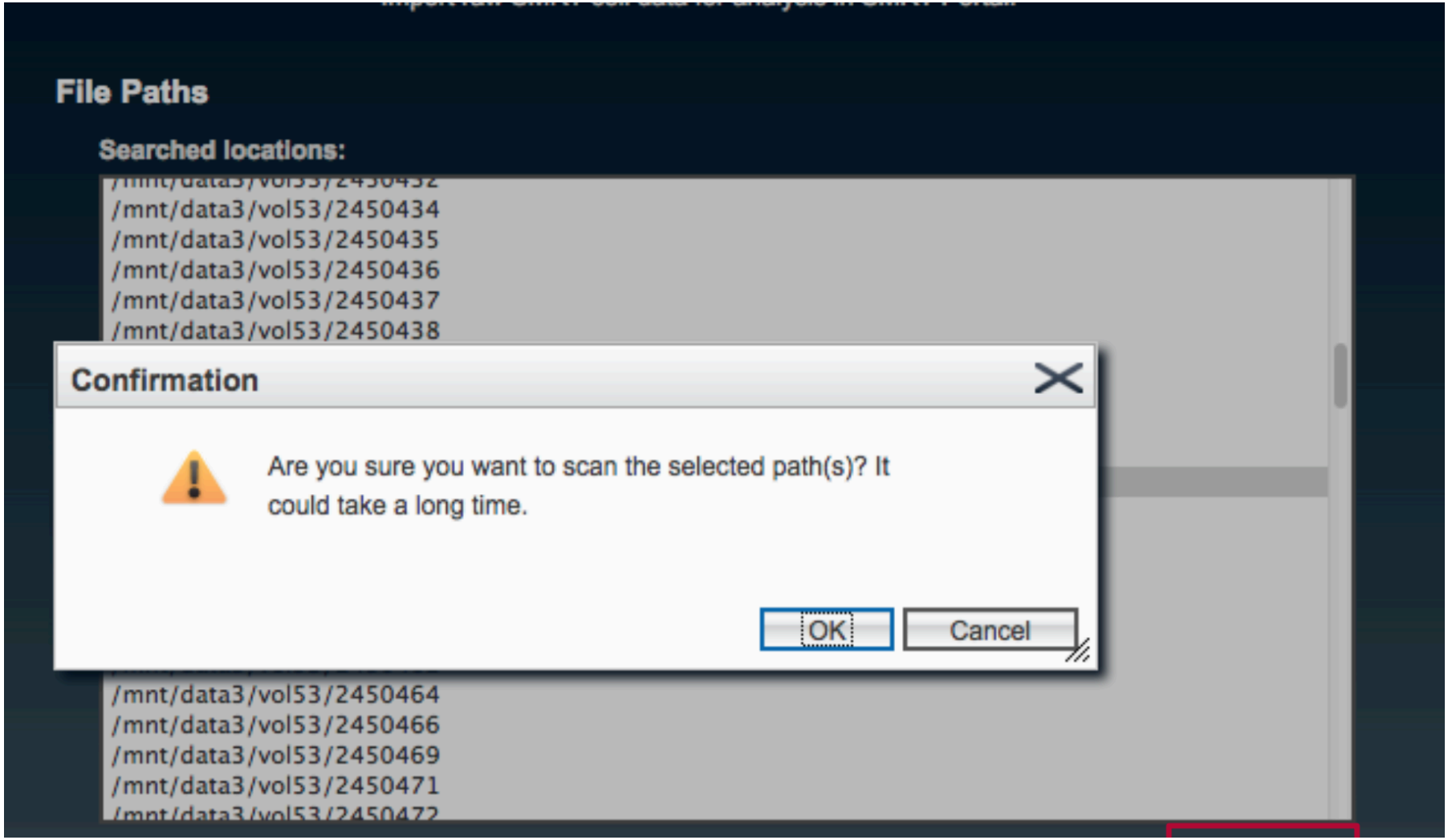
HGAP

SciLifeLab



HGAP

SciLifeLab

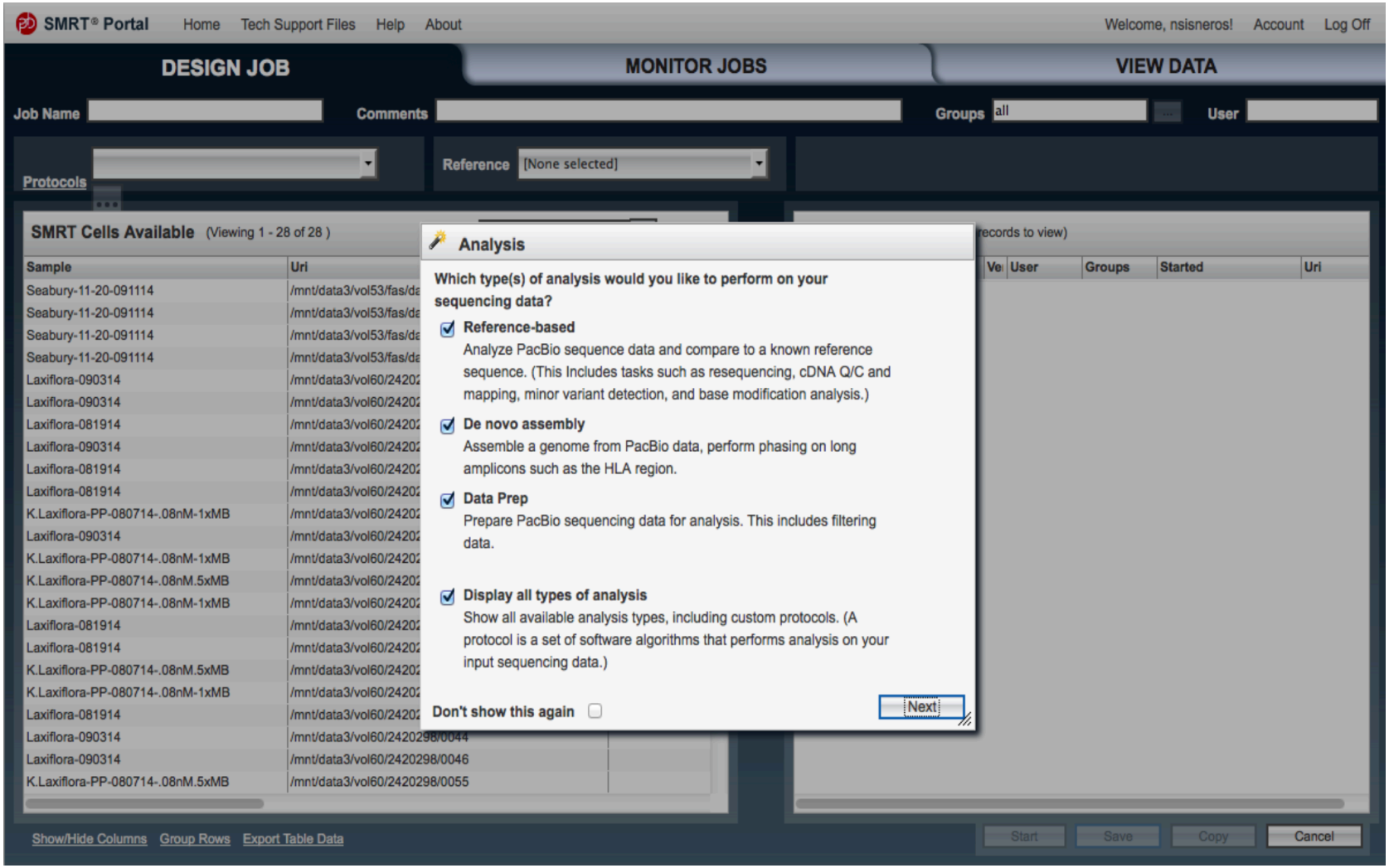


HGAP

SciLifeLab

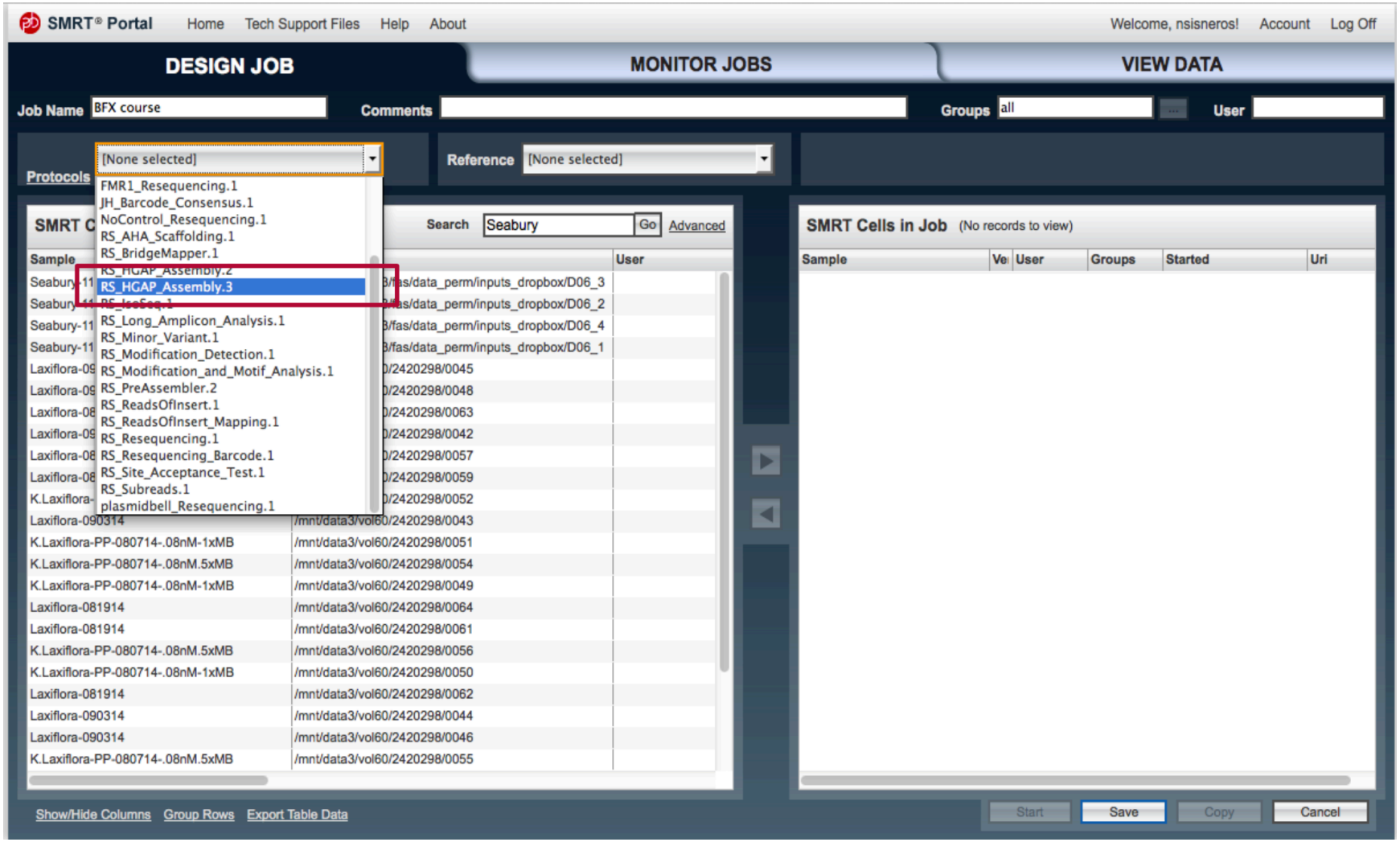
HGAP

SciLifeLab



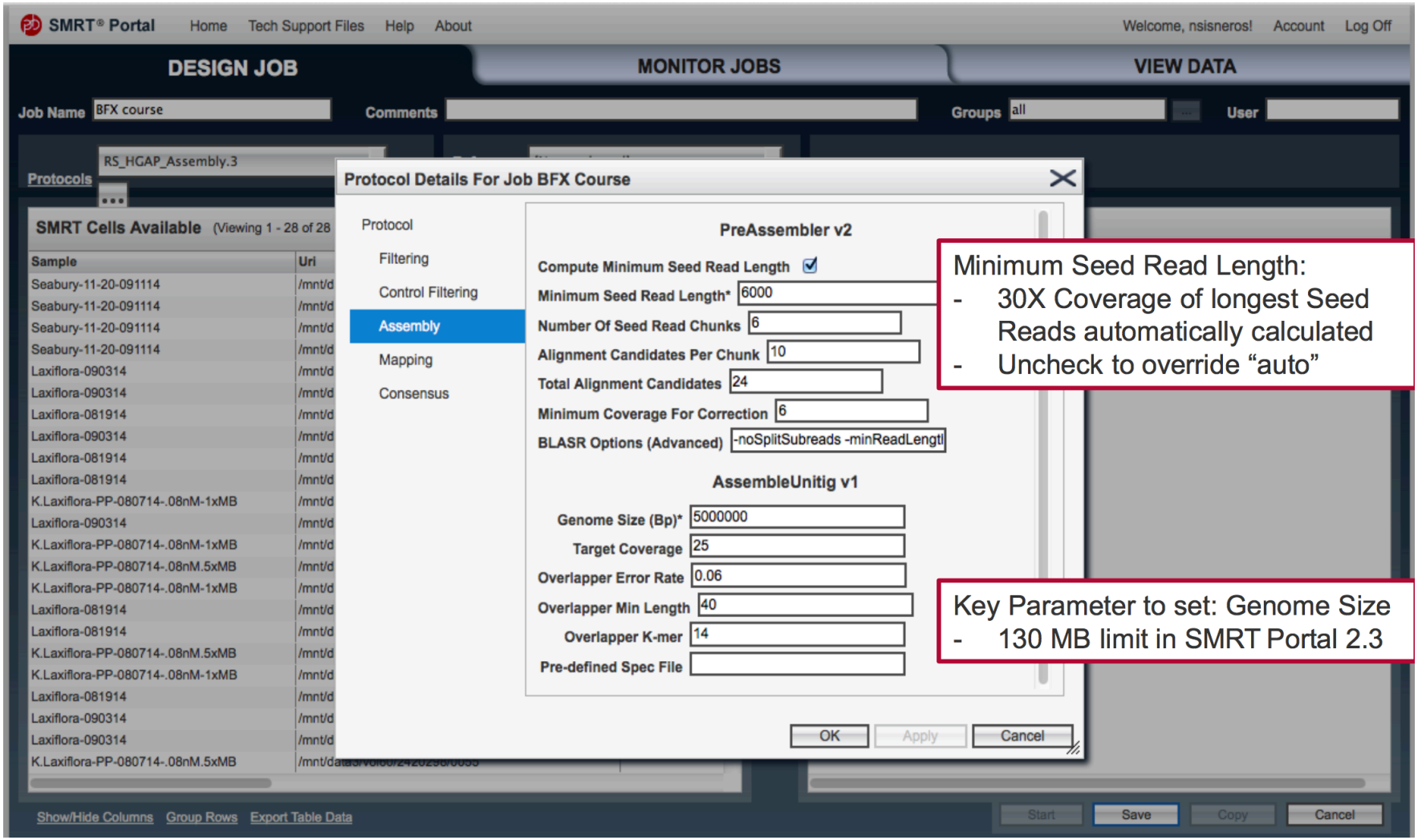
HGAP

SciLifeLab



HGAP

SciLifeLab

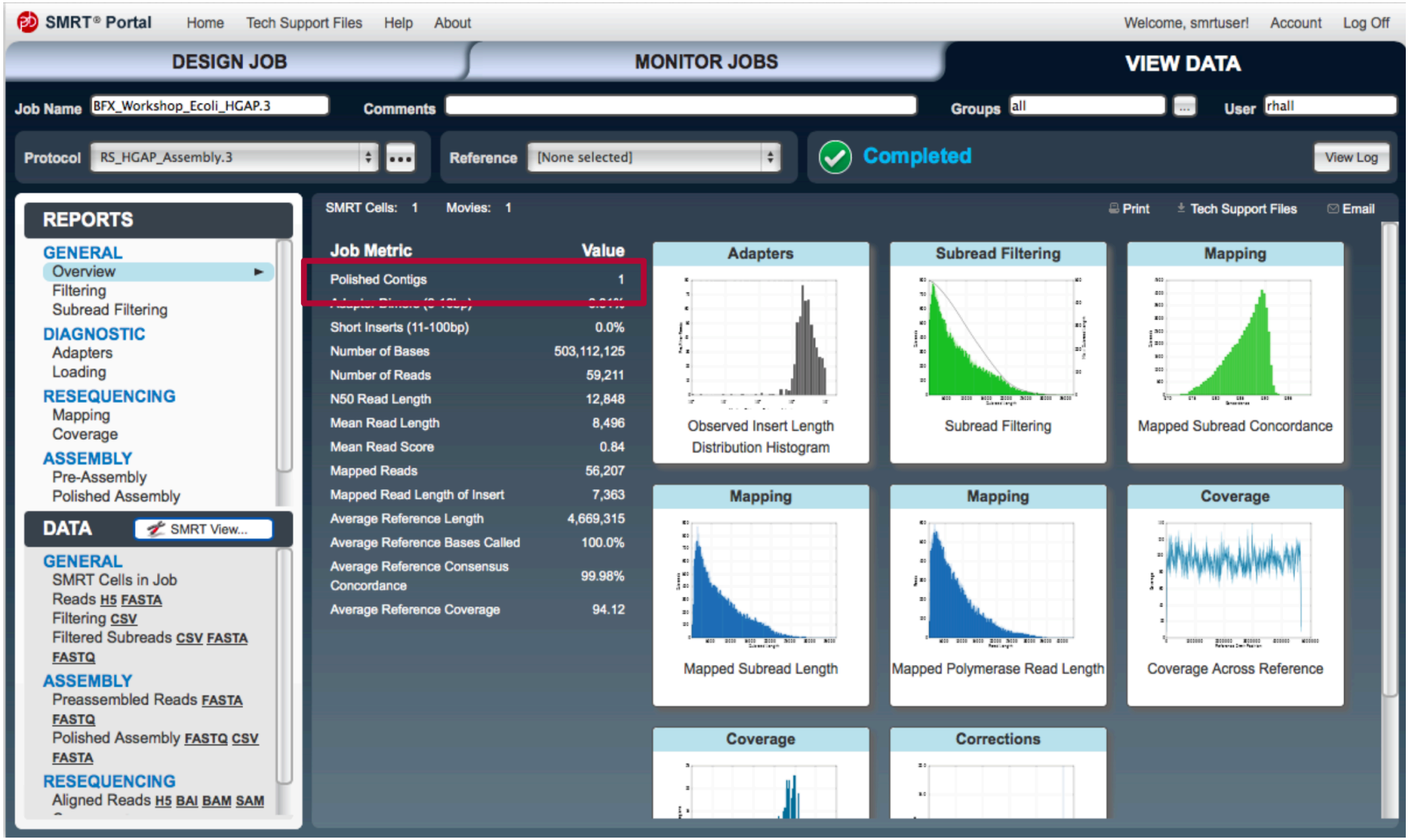


Minimum Seed Read Length:
- 30X Coverage of longest Seed Reads automatically calculated
- Uncheck to override “auto”

Key Parameter to set: Genome Size
- 130 MB limit in SMRT Portal 2.3

HGAP

SciLifeLab



HGAP

SciLifeLab

- Running HGAP (Command line)
  - Install SMRT Analysis software
  - Make a HGAP assembly job using the SMRT portal and save.
  - Save the settings.xml file as HGAP_protocol.xml

- Every SMRT Portal job has the following structure. **Example:**

```
/path/to/smrtanalysis/userdata/jobs/016/016234
├── data/
├── results/
├── log/
├── workflow/
├── job.sh
├── input.xml
└── settings.xml
```

  - `data` is a **directory** that contains intermediate and final data files for the analysis job
  - `results` is a **directory** that contains summary statistics and plots for the analysis job
  - `log` is a **directory** that contains all log files for the analysis job
  - `workflow` is a **directory** that contains all the executables for the analysis job
  - `job.sh` is an executable file used by SMRT Portal to run the `smrtpipe.py` analysis job
  - `input.xml` is a .xml file containing a list of input `bax.h5` files used to run the analysis job
  - `settings.xml` is a .xml file containing the parameters needed to perform the analysis job

HGAP

SciLifeLab

- Running HGAP (Command line) cont'd.
  - Modify Genome size in HGAP_protocol.xml
    - <param name="genomeSize" label="Genome Size (bp)">
      <value>5000000</value>
  - Source the SMRT analysis environment
    - source /path/to/smrtanalysis/install/smrtanalysis_2.3.0.140936/etc/setup.sh
  - Add the full paths of your raw data (*.bax.h5) into an input.fofn
    - find <data_dir> -name "*.bax.h5" > input.fofn
  - Convert the input.fofn to an input.xml
    - fofnToSmrtpipeInput.py input.fofn > input.xml
  - Run SMRT pipe using the protocol and input xmls.
    - smrtpipe.py --params=HGAP_protocol.xml xml:input.xml
  - Results are found in index.html in the working directory
  - Assembly is in data/polished_assembly.fastq.gz

HGAP

SciLifeLab

```
#! /bin/bash
#SBATCH -A <your uppmax project>
#SBATCH -p core
#SBATCH -n 8
#SBATCH -t 1-00:00:00
#SBATCH -J run_smrt_assembly
#SBATCH -e run_smrt_assembly-%j.out
#SBATCH -o run_smrt_assembly-%j.out

module load bioinfo-tools SMRT/2.3.0
WORK_DIR=$SNIC_TMP/smrt_assembly_$(date +%Y_%m_%d-%H.%M)
PROJ_DIR=$PWD
PROTOCOL_XML=$PROJ_DIR/Settings/HGAP_protocol.xml
DATA_DIR=${PROJ_DIR}/00_RawData         # Use full path
GENOME_SIZE=5000000

# Modify Protocol xml to the correct genome size
perl -0777 -i.original -pe "s/<param name=\"genomeSize\" label=\"Genome Size \(bp\)\">\n\s+<value>\d
+<\/value>/<param name=\"genomeSize\" label=\"Genome Size (bp)\">\n\t\t<value>$GENOME_SIZE<\/value>/
igs" $PROTOCOL_XML

# Activate SMRT Analysis environment
source $SMRT_SETUP_SCRIPT
mkdir -p $WORK_DIR; cd $WORK_DIR

# Make input file
find ${DATA_DIR} -name "*.bax.h5" > input.fofn
fofnToSmrtpipeInput.py input.fofn > input.xml

smrtpipe.py --params=$PROTOCOL_XML xml:input.xml

cd $PROJ_DIR; rsync -av $WORK_DIR .
```

Canu

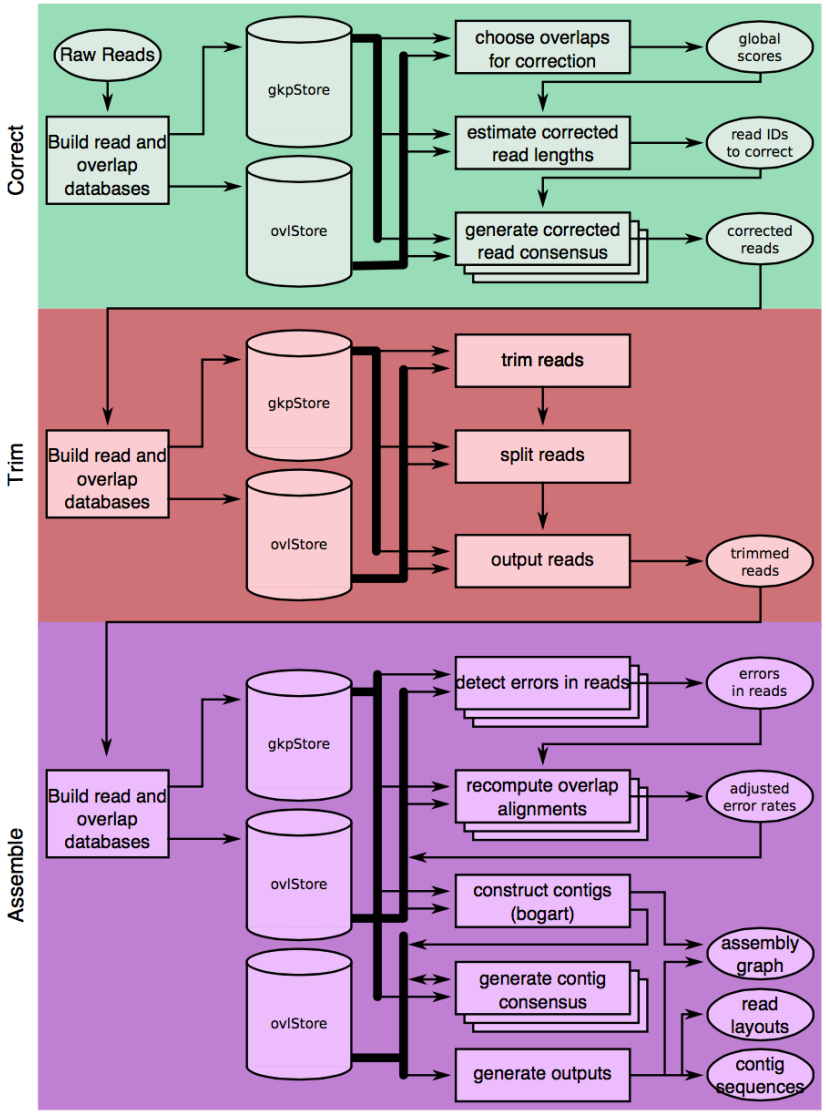


Figure 1. A full Canu run includes three stages: correction (green), trimming (red), and assembly (purple). Canu stages share an interface for binary on-disk stores (databases) as well as parallel store construction. In all stages, the first step constructs an indexed store of input sequences, generates a k-mer histogram, constructs an indexed store of all-vs-all overlaps, and collates summary statistics. The correction stage (green) selects the best overlaps to use for correction, estimates corrected read lengths, and generates corrected reads. The trimming stage (red) identifies unsupported regions in the input and trims or splits reads to their longest supported range. The assembly stage (purple) makes a final pass to identify sequencing errors; constructs the best overlap graph; and outputs contigs, an assembly graph, and summary statistics.

Koren et al. 2016. BioRxiv

Canu

SciLifeLab

- Running Canu
  - Can auto-detect cluster settings (not recommended for milou)
  - Run canu on a node
    - useGrid=false
    - maxThreads=$NPROCS

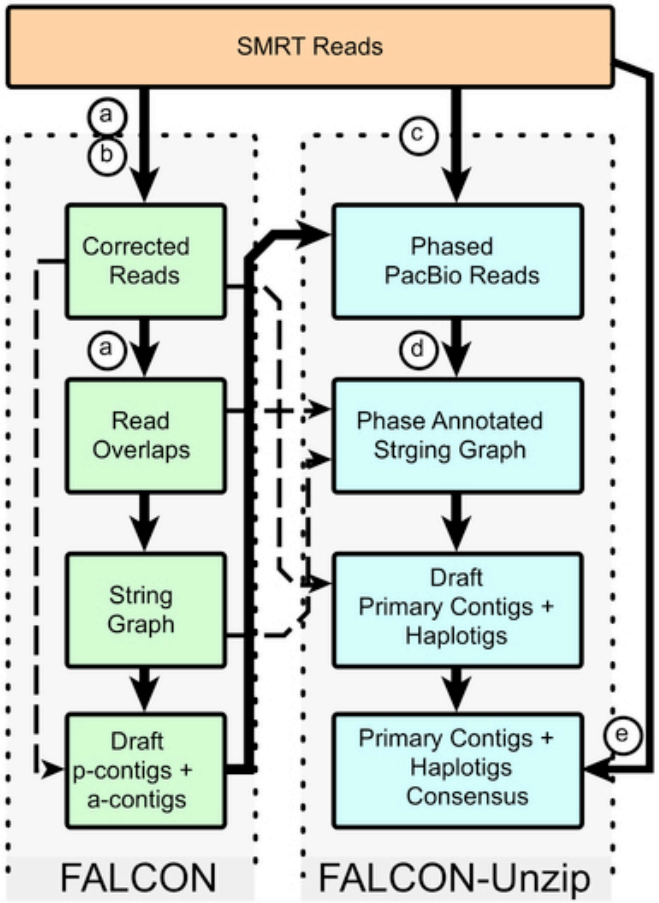
```
canu -p <file_prefix> -d <out_dir> genomeSize="18m"
   maxThreads=24 useGrid=false -pacbio-raw
   <filtered_subreads.fastq.gz>
```

- Results
  - Sequence is in file_prefix.contigs.fasta
  - Assembly graph is in file_prefix.gfa

Falcon

SciLifeLab

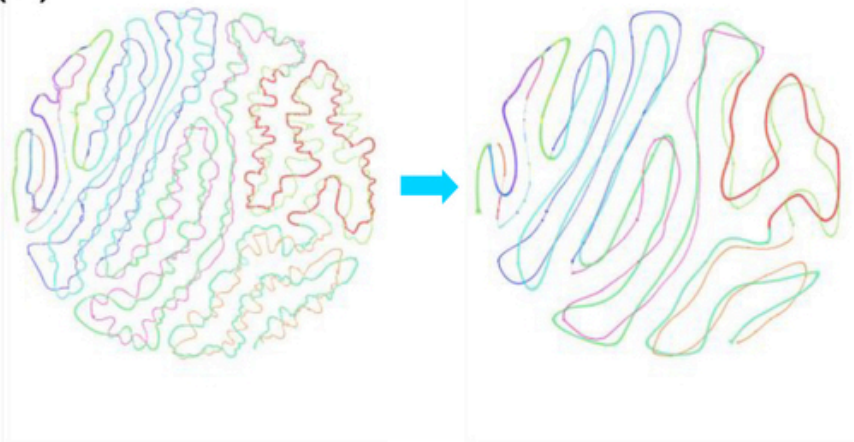
(a)



External code and internal modules used in FALCON and FALCON-Unzip

- (a) Daligner
- (b) Consensus Module (FALCON-sense)
- (c) Phasing Module (FALCON-phasing)
- (d) Graph "Unzip" Module
- (e) BLASR Alignment+ Quiver Consensus Module

(b)



Chin et al. 2016. Nature Methods

Falcon

SciLifeLab

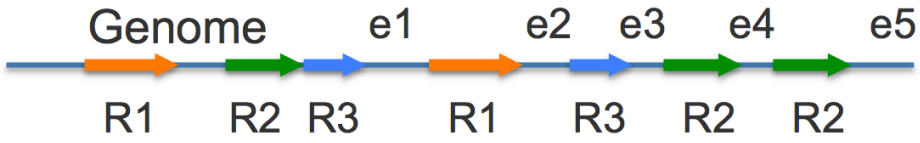
BIOINFORMATICS

Vol. 21 Suppl. 2 2005, pages ii79–ii85
doi:10.1093/bioinformatics/bti1114

Genes and Genomes

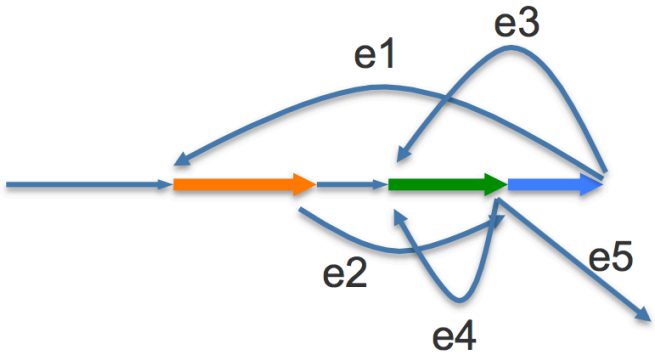
The fragment assembly string graph

Eugene W. Myers
Department of Computer Science, University of California, Berkeley, CA, USA



- String graph:
  - A graph structure that models a genome
- Nodes:
  - Particular positions (typically corresponding to the beginnings or endings of the read fragments) in the genome
- Edges:
  - The sequence between the vertices
- Any string from a path spell out a possible assembly from the reads

String Graph



Slides from Jason Chin:Diploid_Assembly_V1106_SpeakerDeck.pdf

Falcon

SciLifeLab

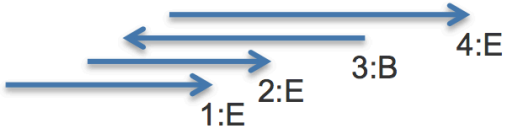
For each overlap, two edges are constructed.

Example:

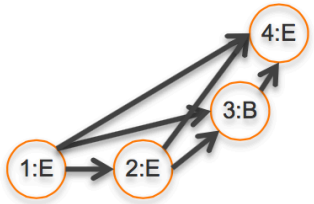
Overlapped reads

New edges

Add f:B, g:B, f:E, g:E as vertices
Add edges f:E ➔ g:B and g:E ➔ f:B

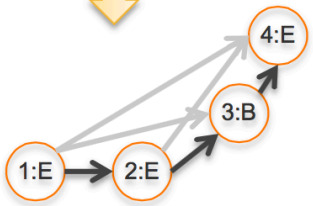


overlaps



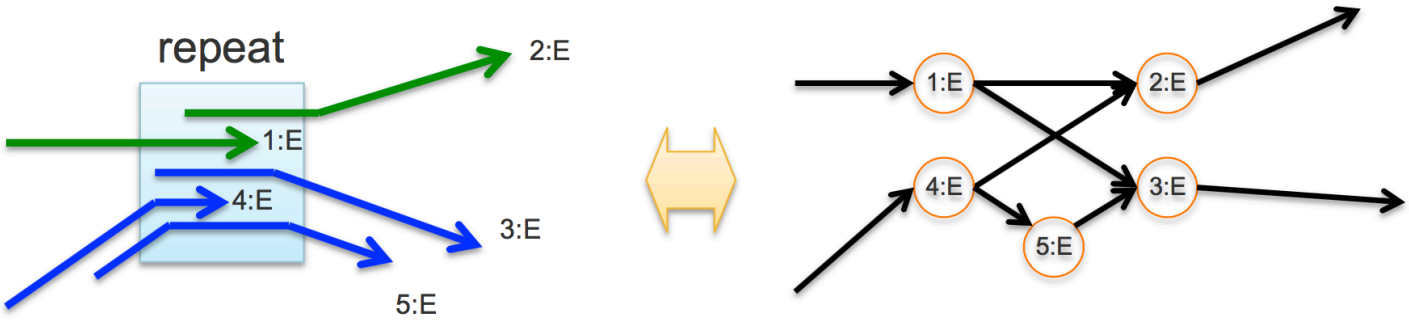
Initial graph

Transitive Reduction



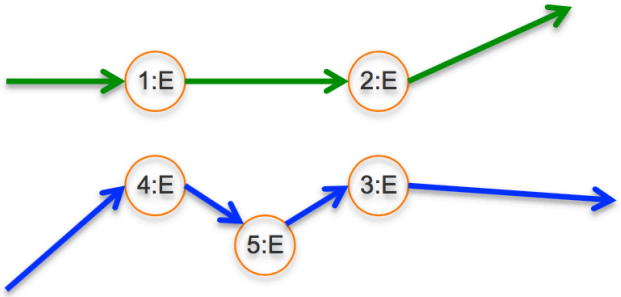
String graph

Falcon

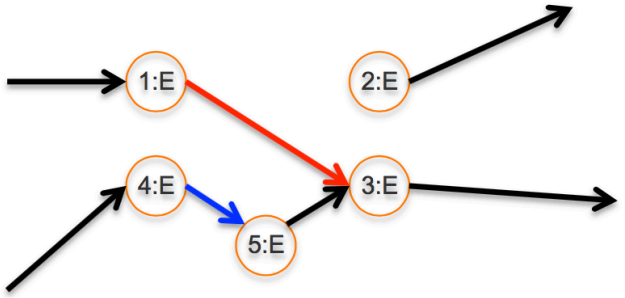


- Using a simple “best overlapping logic” to “untangle” the knots.

Desired final graph



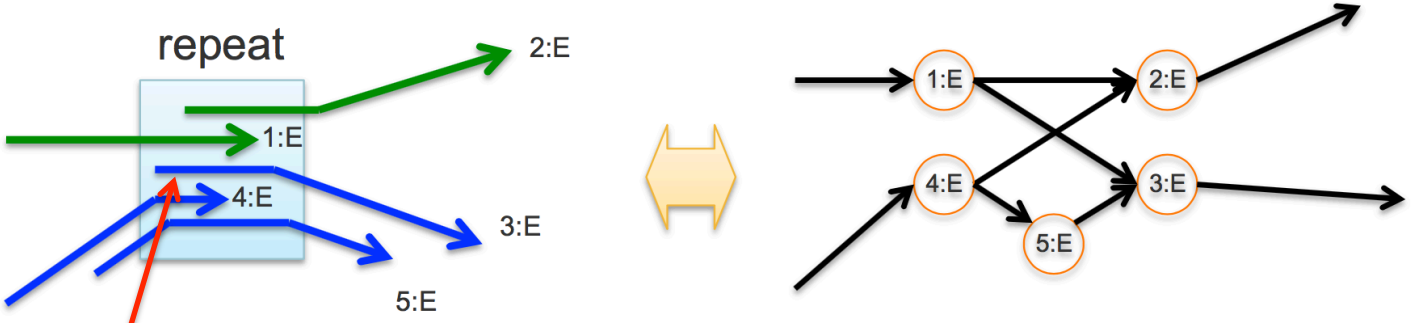
Best overlap string graph



The 4E➔5E edge is better than 4E ➔2E.
The 1E➔3E edge is better than 1E ➔2E.
(“wrong” edge)

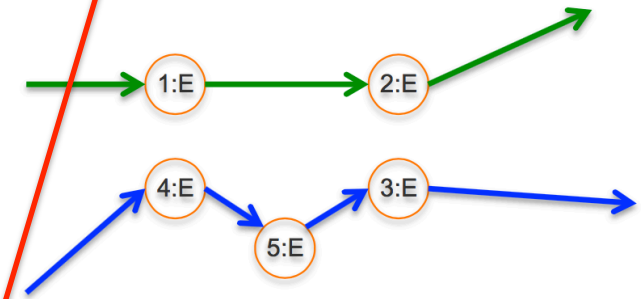
Falcon

SciLifeLab

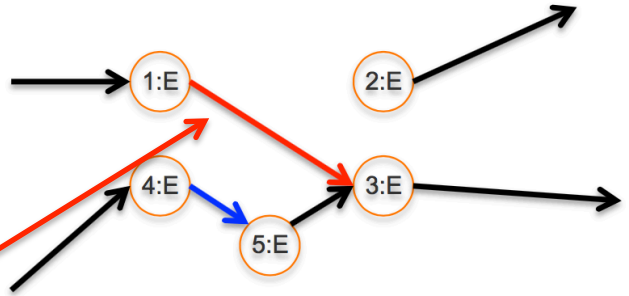


- Using a simple “best overlapping logic” to “untangle” the knots.

Desired final graph



Best overlap string graph

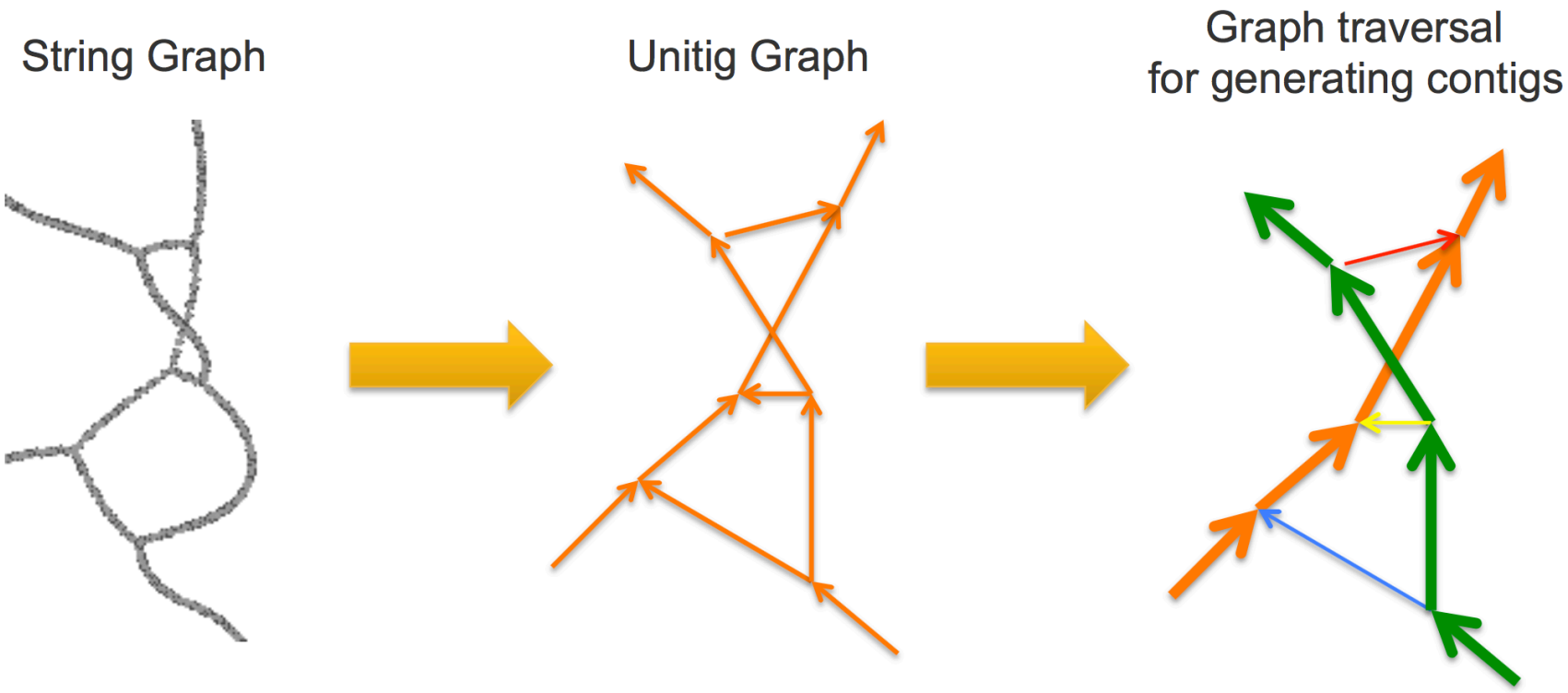


The 4E➔5E edge is better than 4E ➔2E.
The 1E➔3E edge is better than 1E ➔2E.
(“wrong” edge)

Wrong edge selected because of greater overlap with other read

Falcon

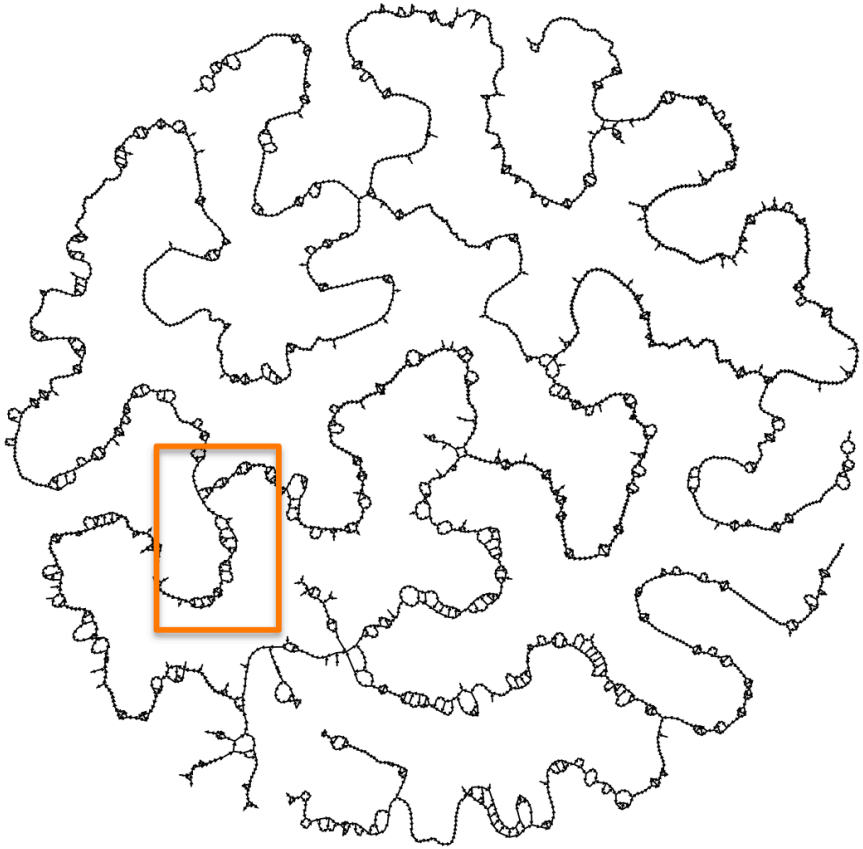
SciLifeLab



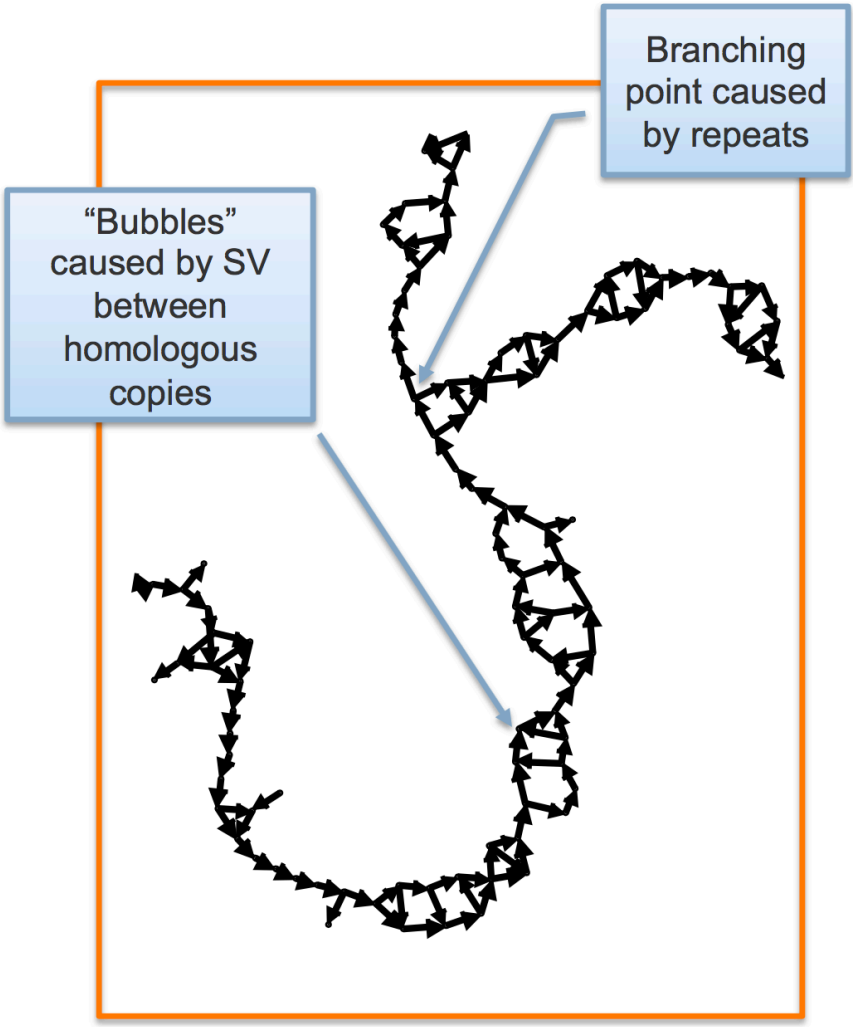
Falcon

SciLifeLab

An unitig graph from Ler-0 + Col-0 data

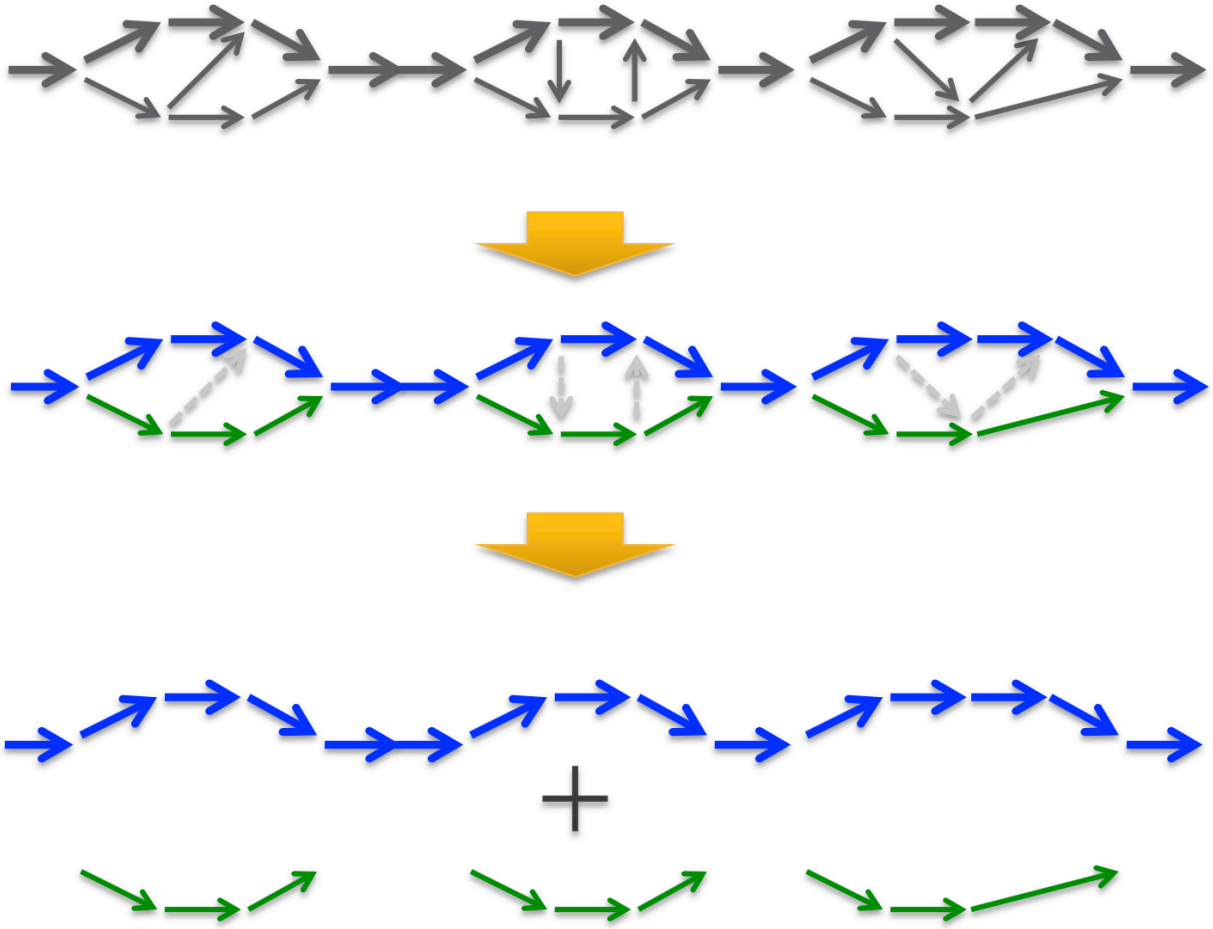


The graph “diameter” ~ 12 M bp
Mean edge size=17.4 k bp



Falcon

SciLifeLab

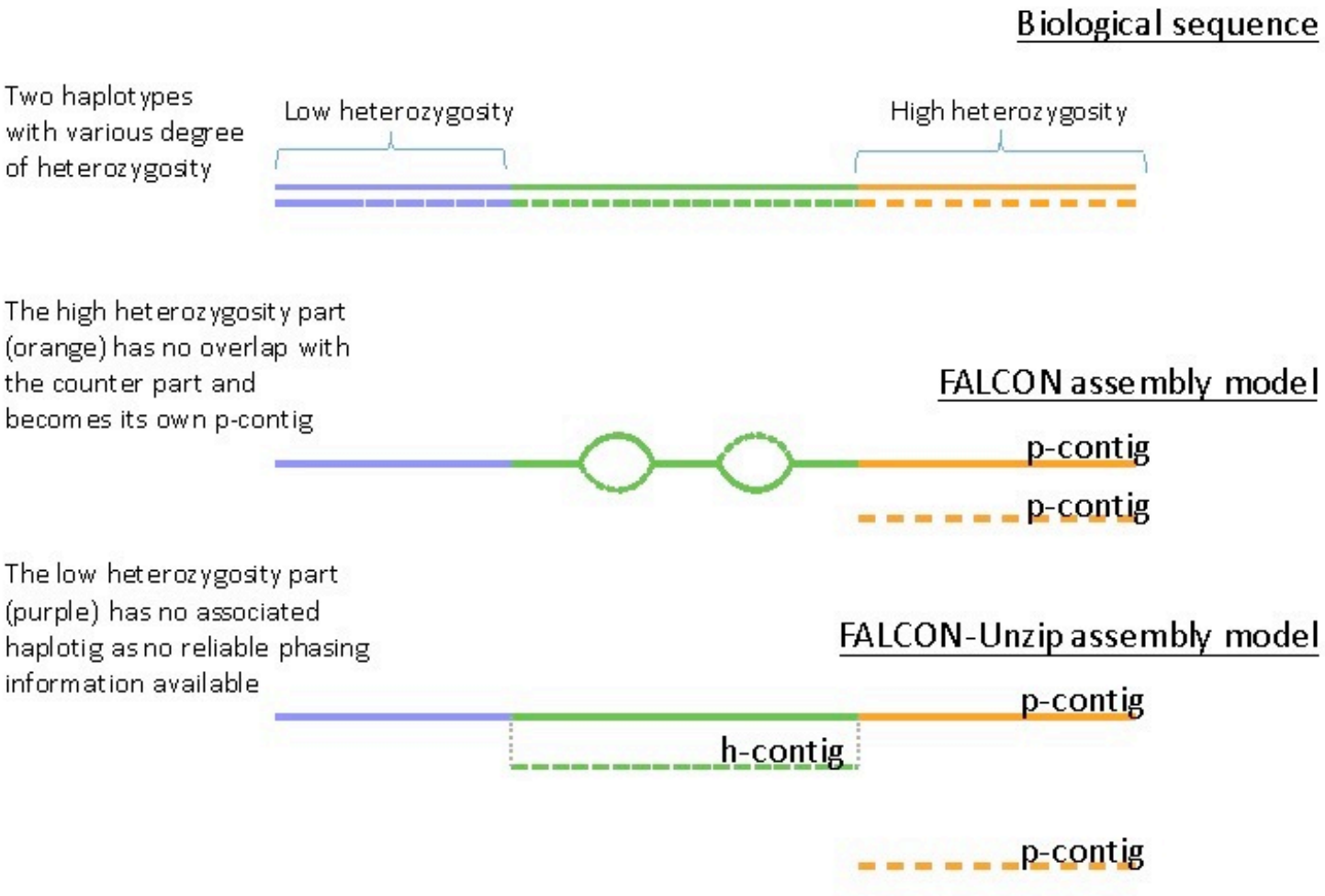


String Bundle

Choose a path
to be the
“primary contig”

Identify
“associated contigs”

Falcon


Biological sequence
Two haplotypes with various degree of heterozygosity
Low heterozygosity
High heterozygosity
The high heterozygosity part (orange) has no overlap with the counter part and becomes its own p-contig
FALCON assembly model
p-contig
p-contig
The low heterozygosity part (purple) has no associated haplotig as no reliable phasing information available
FALCON-Unzip assembly model
p-contig
h-contig
p-contig

Falcon

SciLifeLab

- Running Falcon
  - Make a configuration file
  - Can use SGE queuing manager
  - Run locally on a node
  - Separate filtered subreads into separate fasta's for each movie
    - `zcat *.fastq.gz | seqtk seq -l 5000 -A - | awk 'BEGIN { RS=">"; FS="/" } { print ">"substr($0,1,length($0)-1) > $1".fasta" }'`
  - Make an input fofn
    - `/bin/ls -1 *.fasta > input.fofn`
  - Run Falcon
    - `fc_run.py falcon.cfg`

Falcon

SciLifeLab

- Notes from the author
  - Falcon is limited by file i/o capabilities
    - Lustre file system recommended
    - NFS can handle 3-5 concurrent jobs during pre-assembly
    - Highly repetitive genomes require quadratically more storage space
  - Falcon scales quadratically
    - All-by-all comparison of raw subreads, with matches written to disk

Falcon

SciLifeLab

- The Falcon config file (parameter rich - explanation in the exercises)

```
[General]
jobtype = local                 # other values sge, slurm
input_fofn = input.fofn
input_type = raw                # uncorrected reads
#input_type = preads            # falcon corrected reads

# The length cutoff used for seed reads used in initial
mapping - these make the corrected reads
length_cutoff = 12000           # use longest 30X coverage

# The length cutoff used for seed reads used for pre-
assembly - the min length of corrected reads
length_cutoff_pr = 12000        # 0-5000 lower than above
```

Miniasm and Racon

SciLifeLab

- No read correction step
- Miniasm implements Overlap - Layout, Racon implements Consensus

```
# Overlap
minimap/minimap -Sw5 -L100 -m0 -t8 reads.fq reads.fq | \
gzip -1 > reads.paf.gz

# Layout
miniasm/miniasm -f reads.fq reads.paf.gz > contigs.gfa

# Get fasta
awk '/^S/{print ">"++seq"\n"$3}' contigs.gfa > contigs.fasta

# Consensus
miniasm/minimap -t 16 contigs.fasta reads.fq | \
racon -t 16 reads.fq - contigs.fasta consensus_contigs.fasta
```

Alternative PacBio Assemblers

SciLifeLab

- ABruijn
  - Uncorrected overlap assembly of long read sequences followed by polishing
  - https://github.com/fenderglass/ABruijn
- Ra
  - Uncorrected overlap assembly of long read sequences
  - https://github.com/mariokostelac/ra-integrate
- ARacon
  - Combination of GraphMap + Miniasm + Racon
  - https://github.com/isovic/aracon
- Hinge
  - Read filtering (but no correction) followed by overlap assembly of long read sequences
  - https://github.com/fxia22/HINGE
- SMARTdenovo
  - Uncorrected overlap assembly of long read sequences
  - https://github.com/ruanjue/smartdenovo

Preliminary Assembly Diagnostics

SciLifeLab

- Assembly Size
  - Assemblathon Script (https://github.com/KorfLab/Assemblathon)
  - Quast

Number of scaffolds	556	
Total size of scaffolds	31318563	← Expected genome size
Longest scaffold	447934	
Shortest scaffold	8580	
Number of scaffolds > 1K nt	556	100.0%
Number of scaffolds > 10K nt	555	99.8%
Number of scaffolds > 100K nt	38	6.8%
Number of scaffolds > 1M nt	0	0.0%
Number of scaffolds > 10M nt	0	0.0%
Mean scaffold size	56328	
Median scaffold size	43995	
N50 scaffold length	60037	
L50 scaffold count	152	

Preliminary Assembly Diagnostics

SciLifeLab

- Corrected Read Coverage
  - What happened in the correction process
  - High coverage? Use the ~100X longest subreads

```
--   Found 87386 reads.
--   Found 1654383605 bases (45.95 times coverage).
--
--   Read length histogram (one '*' equals 265.11 reads):
--         0    999      0
--      1000   1999      0
--      2000   2999      0
--      3000   3999      0
--      4000   4999      0
--      5000   5999      0
--      6000   6999      0
--      7000   7999      0
--      8000   8999      0
--      9000   9999      0
--     10000  10999      0
--     11000  11999      0
--     12000  12999      0
--     13000  13999      0
--     14000  14999      0
--     15000  15999  18558 **********************************************************************
--     16000  16999  15099 ********************************************************
--     17000  17999  11974 *********************************************
--     18000  18999   9486 ***********************************
--     19000  19999   7344 ***************************
--     20000  20999   5652 *********************
--     21000  21999   4328 ****************
--     22000  22999   3516 *************
--     23000  23999   2725 **********
--     24000  24999   2057 *******
--     25000  25999   1672 ******
--     26000  26999   1243 ****
--     27000  27999    920 ***
--     28000  28999    735 **
--     29000  29999    541 **
--     30000  30999    414 *
--     31000  31999    324 *
```

Preliminary Assembly Diagnostics

SciLifeLab

- Falcon: DBstats 1-preads_ovl/preads.db
  - Focus on % Bases column (multiply by read coverage to find cutoff).

```
Statistics for all wells of length 500 bases or more

         12,915 reads        out of          13,124  ( 98.4%)
    116,202,931 base pairs   out of     116,263,784  ( 99.9%)

          8,997 average read length
          6,983 standard deviation

  Base composition: 0.249(A) 0.239(C) 0.258(G) 0.255(T)

  Distribution of Read Lengths (Bin size = 1,000)

        Bin:       Count   % Reads   % Bases      Average
     42,000:           1       0.0       0.0        42279
     41,000:           2       0.0       0.1        41631
...
(more bin values)
...
      3,000:       1,065      75.5      95.0        11317
      2,000:       1,328      85.8      97.9        10259
      1,000:       1,444      97.0      99.7         9251
          0:         387     100.0     100.0         8997
```

Find the row that corresponds to 30x coverage.

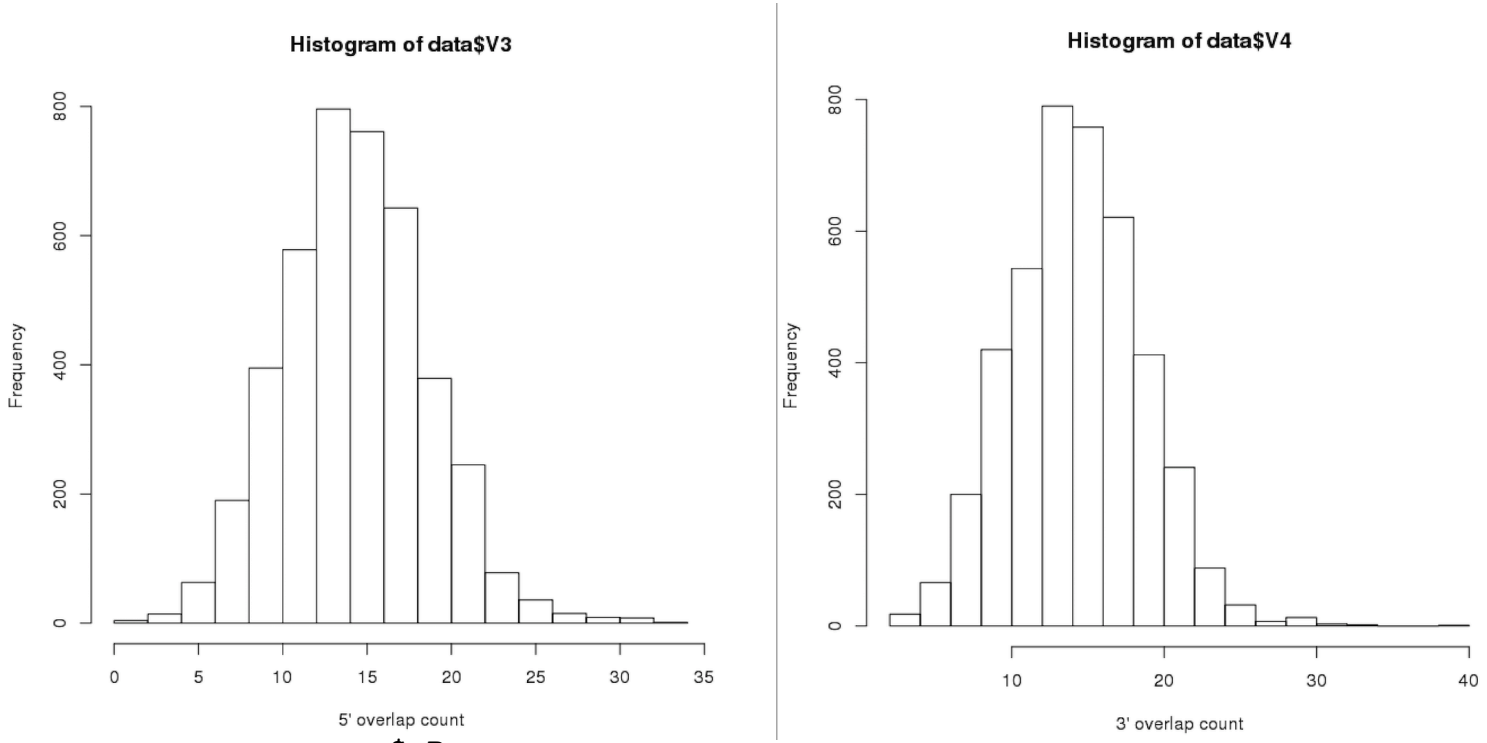
Total bases/ Est. genome size = Total coverage.
30x / Total coverage = % bases

Bin corresponding to 30x is the cutoff length.

Preliminary Assembly Diagnostics

SciLifeLab

- Falcon: Overlap statistics
  - `cd 1-preads_ovl/ ;`
    `fc_ovlp_stats --fofn merge-gather/las.fofn > ovlp_stats.txt`

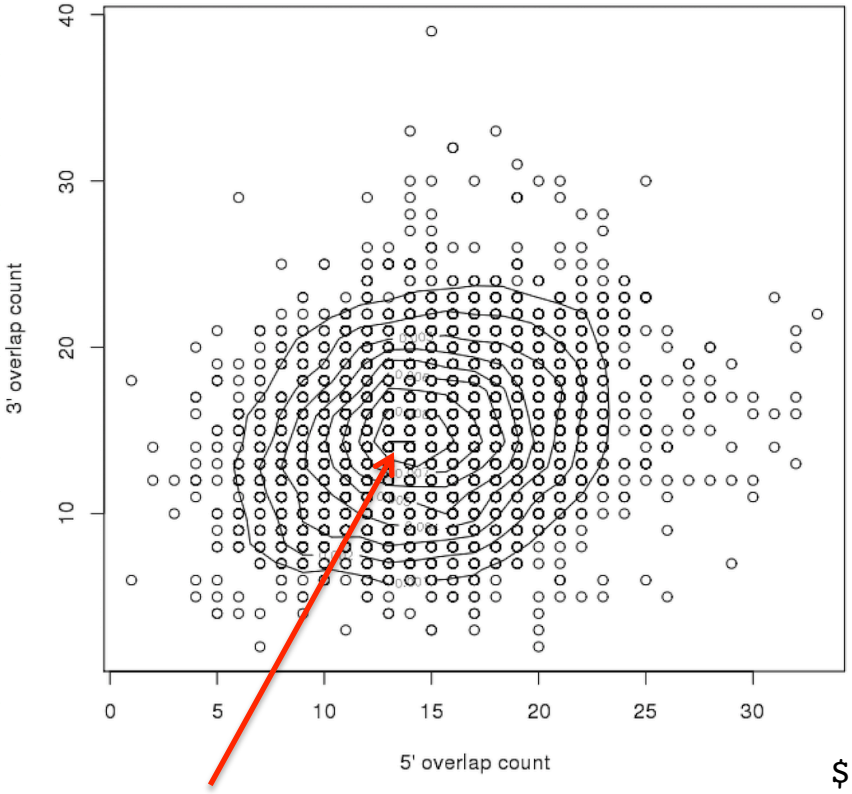



```
$ R
> data <- read.table("ovlp_stats.txt")
> hist(data$V3,xlab="5' overlap")
> hist(data$V4,xlab="3' overlap")
```

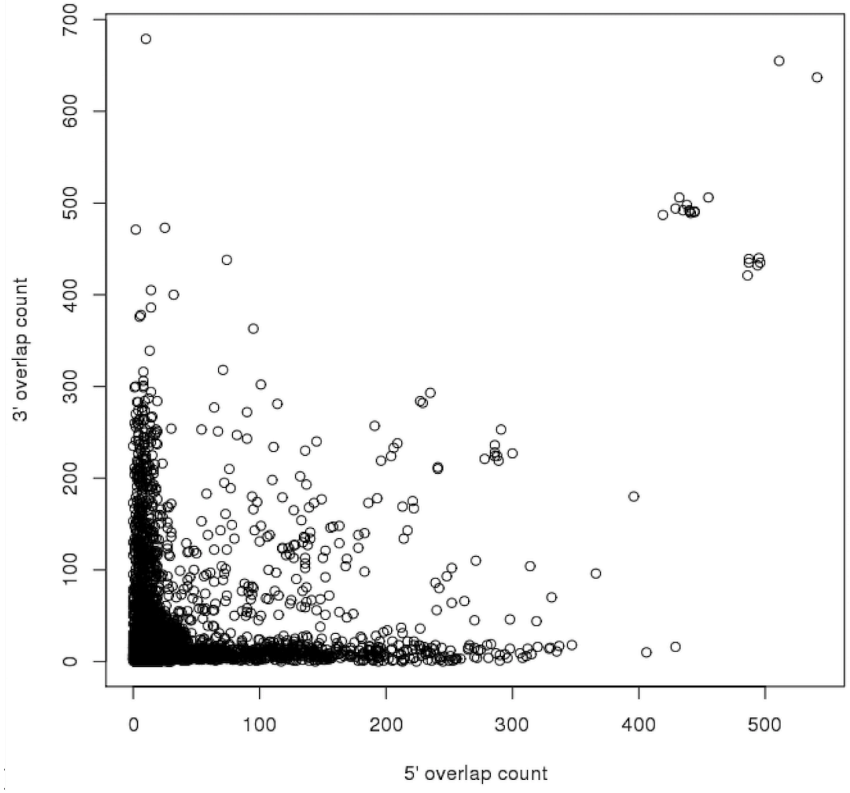
Preliminary Assembly Diagnostics

SciLifeLab

- Falcon: Overlap statistics



Central density and symmetry along x=y is expected.

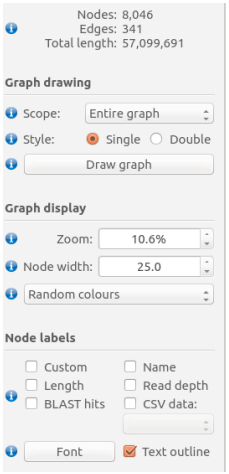


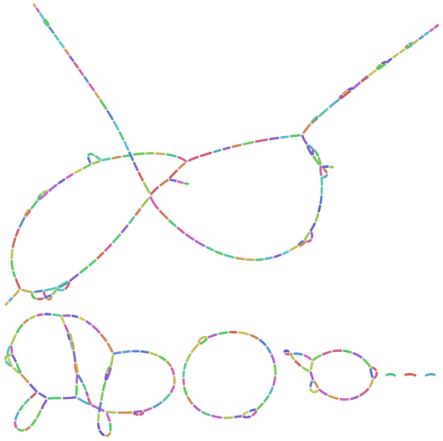

```
$
> library(MASS)
> data <- read.table("ovlp_stats.txt")
> plot(data$V3,data$v4)
> z <- kde2d(data$V3,data$v4)
> contour(z,add=TRUE)
```

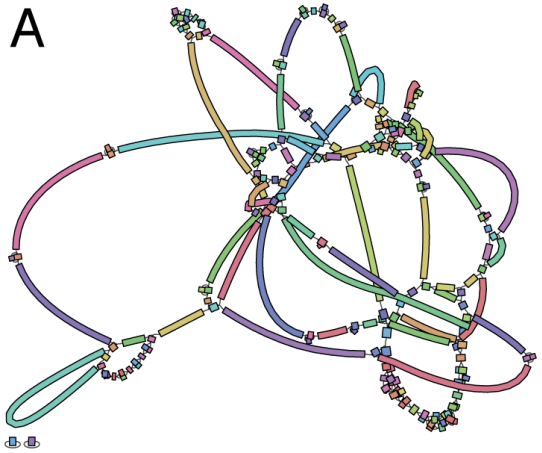
Preliminary Assembly Diagnostics

SciLifeLab

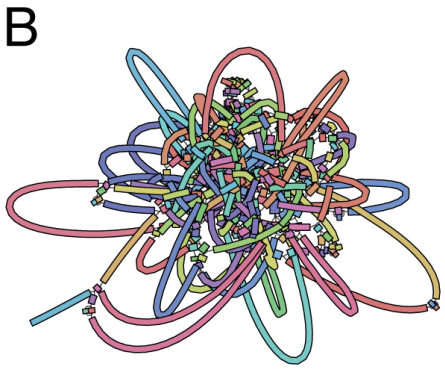
- Assembly Graph
  - Check connectedness of contigs
    - Is longer range information needed?
      - Higher quality sequence material
      - BioNano
      - Chicago / Dovetail



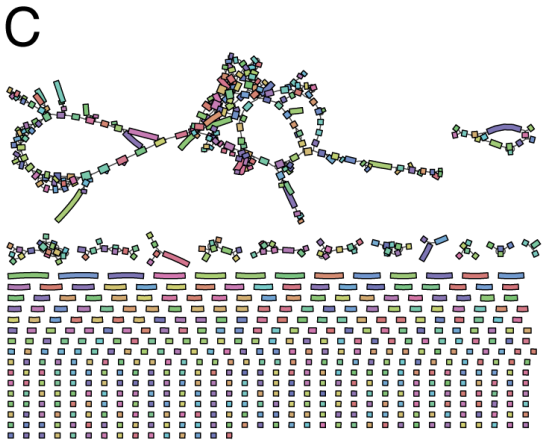




Good



OK

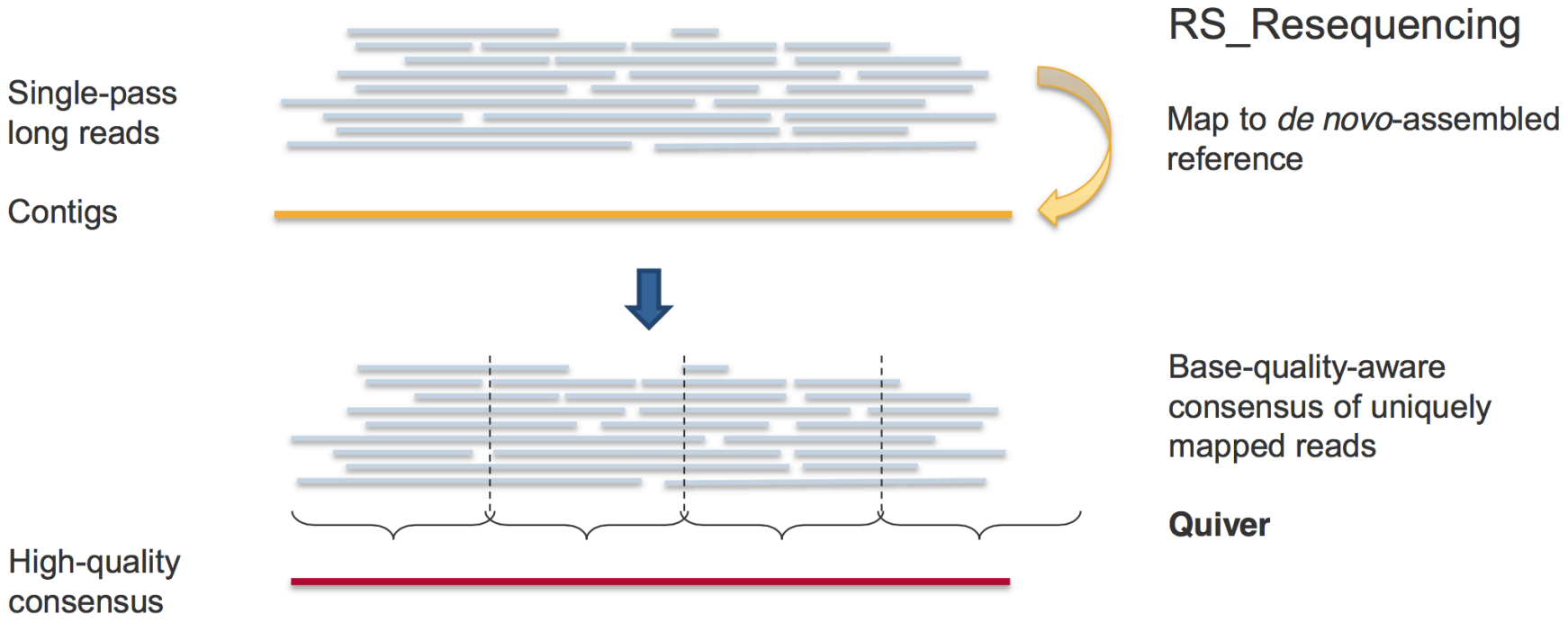


Bad

Polishing assemblies

SciLifeLab

- Draft assemblies still contain many InDel and base substitution errors.
  - Correction using Quiver / Arrow and PacBio reads



Polishing assemblies

SciLifeLab

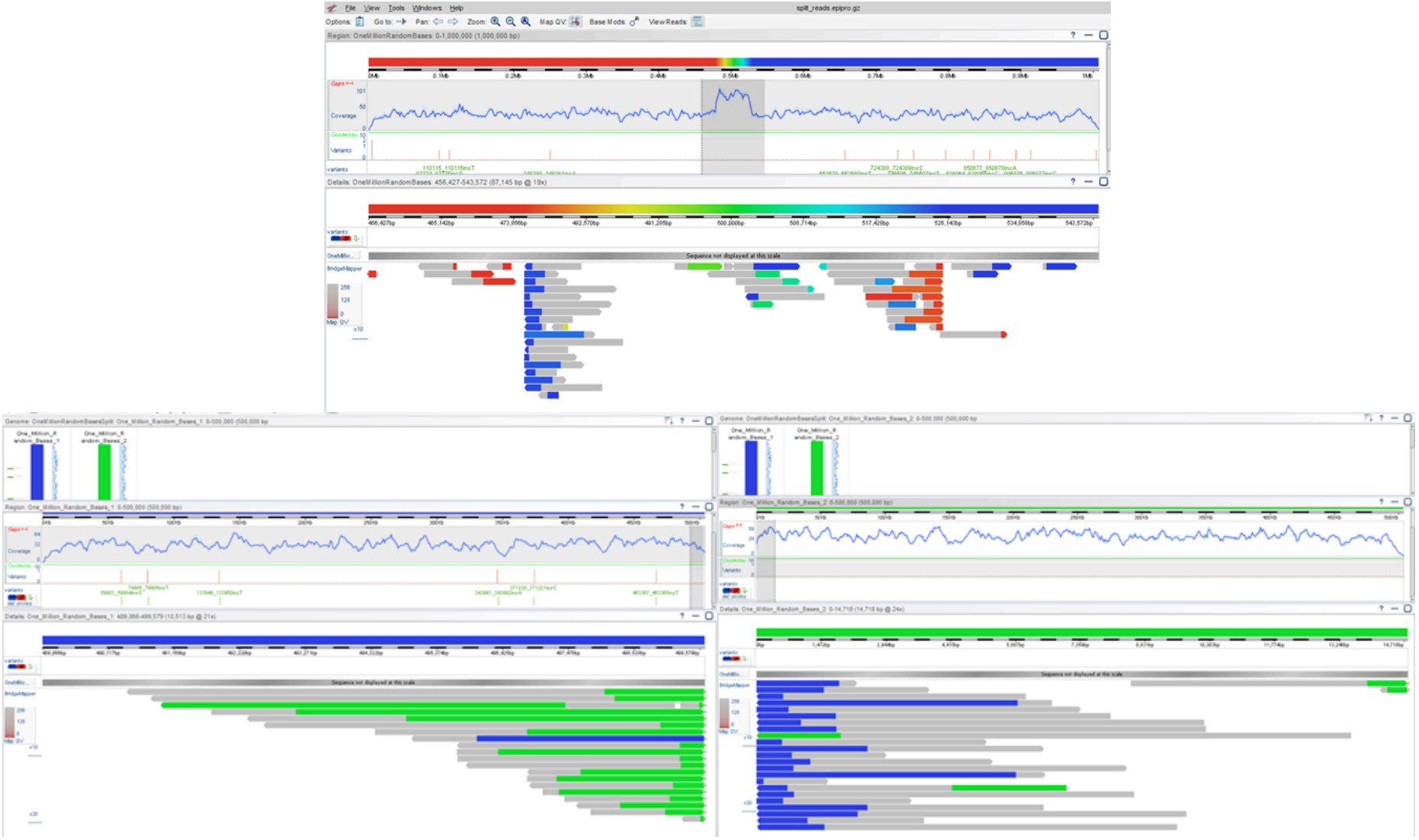
- In the SMRT portal select a protocol that includes the following modules
  - P_Filter
  - P_Mapping
  - P_GenomicConsensus
    - RS_Resequencing
    - RS_BridgeMapper
      - P_BridgeMapper

- To run via SMRT pipe
  - Copy the settings.xml of the dummy job
  - Make the draft assembly a reference using ReferenceUploader
  - Change reference value in settings.xml
  - Run

Polishing assemblies

SciLifeLab

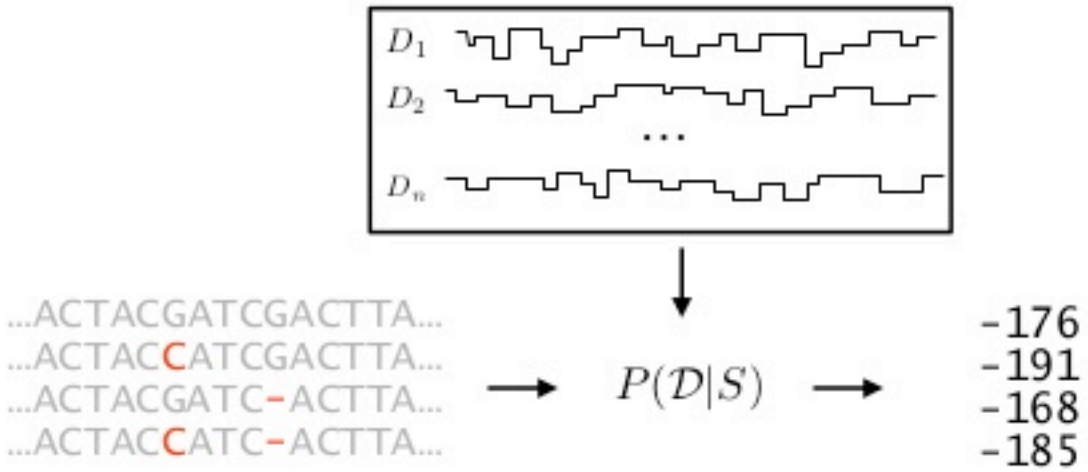
- Bridge Mapper results opened with SMRTview



Polishing assemblies

SciLifeLab

- Draft assemblies still contain many InDel and base substitution errors.
  - Correction using Nanopolish and nanopore reads
    - Toolkit for working with signal-level data
    - Originally designed for improving a consensus sequence using the signals from multiple reads



Jared Simpson: AGBT presentation

Polishing assemblies

SciLifeLab

Human Assembly Consensus

OICR

Assembly	Percent Identity
canu	94.8%
canu+racon	96.5%
canu+racon+nanopolish (30X)	99.1%
canu+racon+nanopolish (60X)	99.4%
canu+racon+nanopolish (60X) + pilon	99.6%

- This is a work in progress
- Polishing a single 6 Mbp chr20 contig
- 30X data set is NA12878 consortium data only
- 60X data set includes 30X PCR-amplified NA12878 provided by ONT
- stats calculated from bwa mem alignments to GRCh38
- differences that matched an NA12878 variant were not consider an error

21

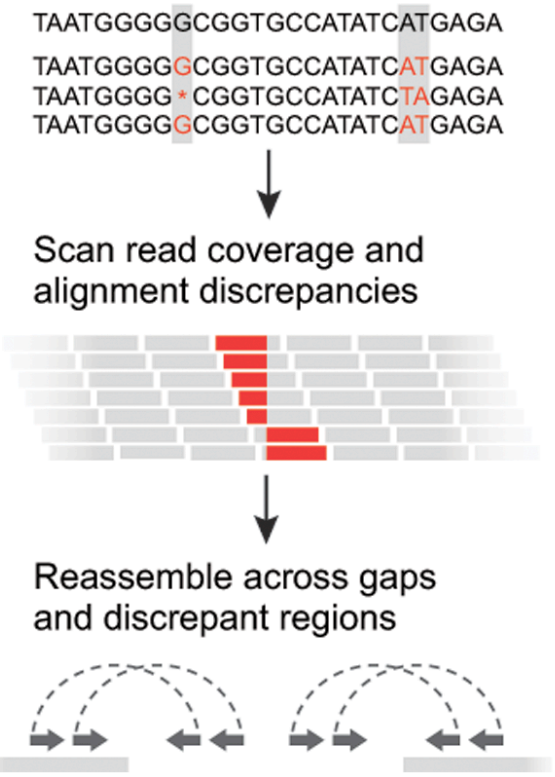
Jared Simpson: AGBT 2016 Presentation

Polishing assemblies

SciLifeLab

- Assembly Polishing
  - Can also be performed using Pilon and Illumina reads

PROCESS	RESULT	
Pilon protocol	Assembly improvement (Fasta)	Variation detection (VCF)
Evaluate alignment pileups	Identify and fix base errors	Identify SNPs and small indels
Scan read coverage and alignment discrepancies	Identify potential local misassemblies	Identify larger insertions and deletions
Reassemble across gaps and discrepant regions	Attempt to fill gaps and fix local misassemblies	Attempt to build out the full sequence of larger insertions



Walker et al. 2014. PLOS One

Summary

SciLifeLab

- Long read sequencing is very dependent on sample DNA quality
- The longest reads are targeted for correction
- Correction with Illumina only does part of the read correction job
- Check basic stats
- Use the assembler output to diagnose major issues.
- Select your best assemblies
- Polish!
- Assess correctness.