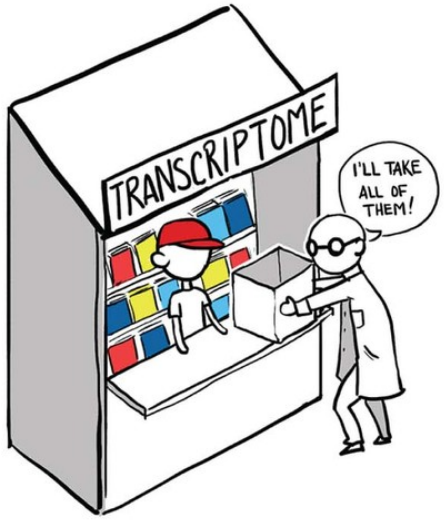
PacBio Genome Assembly using FALCON

Jie (Jessie) Li
PhD
Bioinformatics Analyst
Bioinformatics Core
UCD

ONE DOES NOT SIMPLY

ASSEMBLE A GENOME

Genome Assembly



Genome Assembly

Genome Assembly

Genome Assembly

Genome Assembly

Stages of Building a Genome

- Contigs
- Scaffolds
- Chromosomes

Stages of Building a Genome

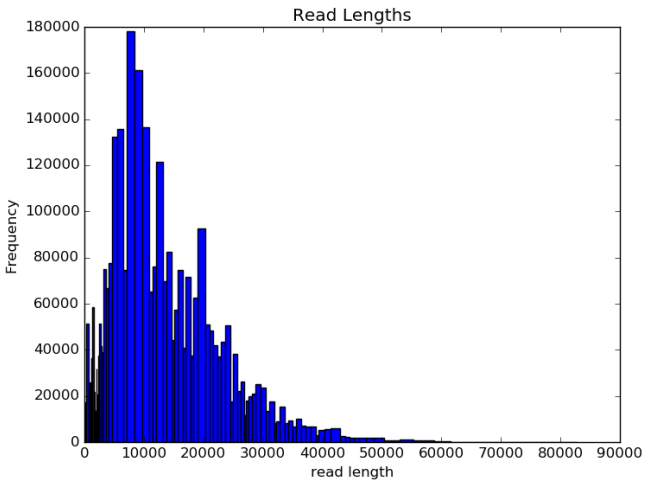
- Contigs
- Scaffolds
- Chromosomes

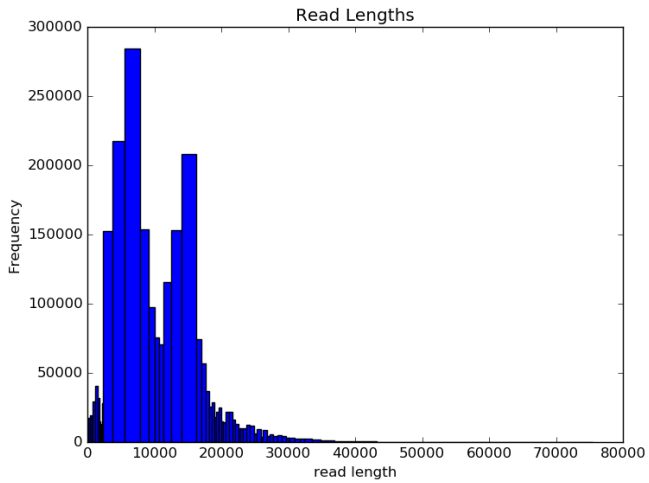
Difficulty in Genome Assembly

Difficulty in Genome Assembly

Strengths of PacBio Technology

- No base bias: GC content or sequence context
- Long read length
  - Averaging ~10kb
  - Some approaching 100kb





"Weakness" of PacBio Technology

- High error rate in individual single-pass sequence read

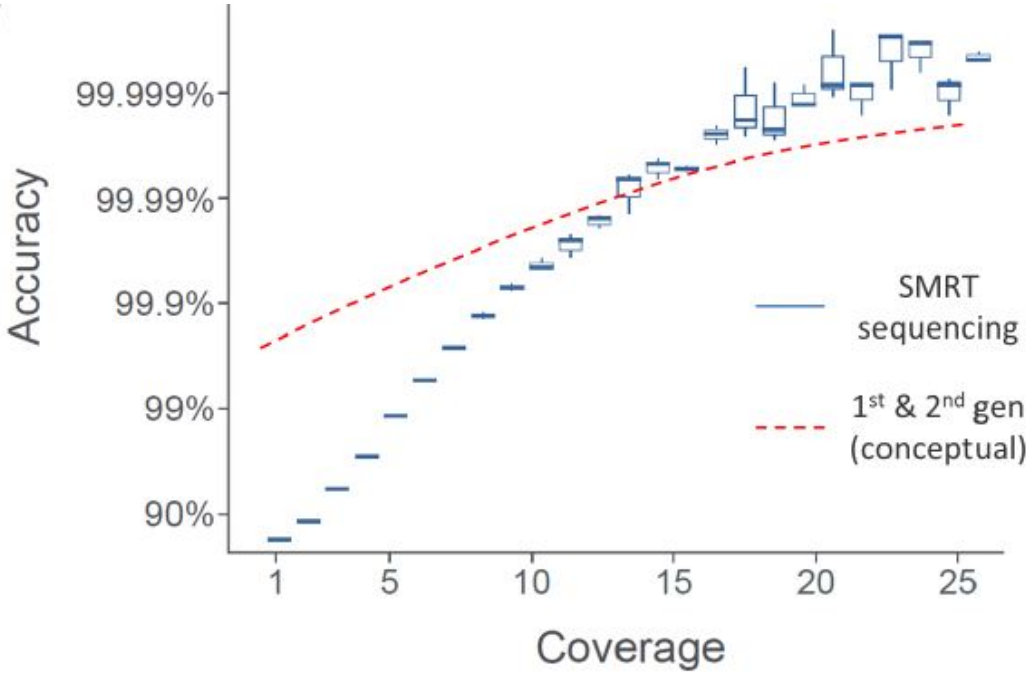
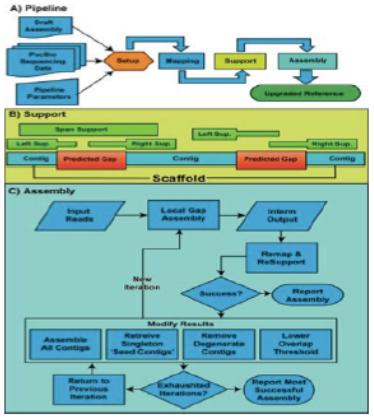


Figure 3. Consensus accuracy as a function of sequencing coverage for different sequencing technologies.

http://www.pacb.com/uncategorized/a-closer-look-at-accuracy-in-pacbio/

Genome Assembly using PacBio Data

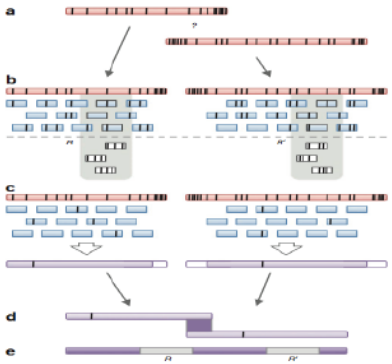
PBJelly



Gap Filling and Assembly Upgrade

English *et al* (2012)
PLOS One. 7(11): e47768

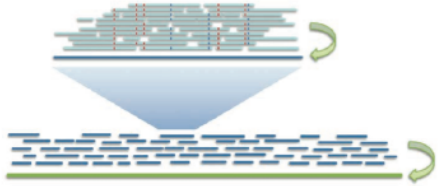
PacBioToCA & ECTools



Hybrid/PB-only Error Correction

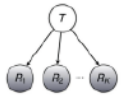
Koren, Schatz, *et al* (2012)
Nature Biotechnology. 30:693–700

HGAP & Quiver



$$\Pr(\mathbf{R} \mid T)$$

$$\Pr(\mathbf{R} \mid T) = \prod_k \Pr(R_k \mid T)$$



Quiver Performance Results
Comparison to Reference Genome
(*M. ruber*; 3.1 MB; SMRT® Cells)

	Initial Assembly	Quiver Consensus
QV	43.4	54.5
Accuracy	99.99540%	99.99964%
Differences	141	11

PB-only Correction & Polishing

Chin *et al* (2013)
Nature Methods. 10:563–569

< 5x PacBio Coverage > 50x

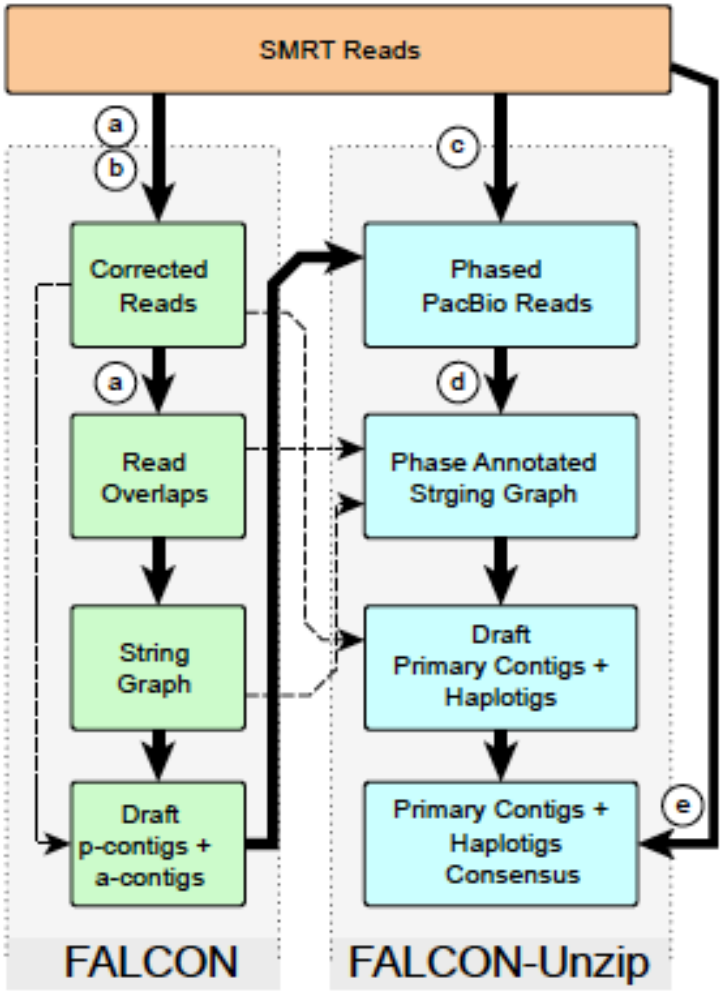
Computational Tools for PacBio-only Assembly

- Celera Assembler
- Canu
- HGAP
- FALCON + FALCON_unzip

Computational Tools for PacBio-only Assembly

- Celera Assembler
- Canu
- HGAP
- **FALCON + FALCON_unzip**
  - Available methods for diploid assembly tend to produce highly fragmented results, with contigs averaging just a few hundred bases to several kilobases in length.
  - Sequencing both parents and offsprings to infer haplotypes, but requires sequencing additional samples and is fundamentally limited in contiguity of the initial assemblies.
  - Pooled clonal fosmid sequencing produces diploid sequences but is expensive, labor intensive and the assembly contiguity is limited by the clonability of the source DNA, and the size and quality of the sequenced fosmids.

Overview

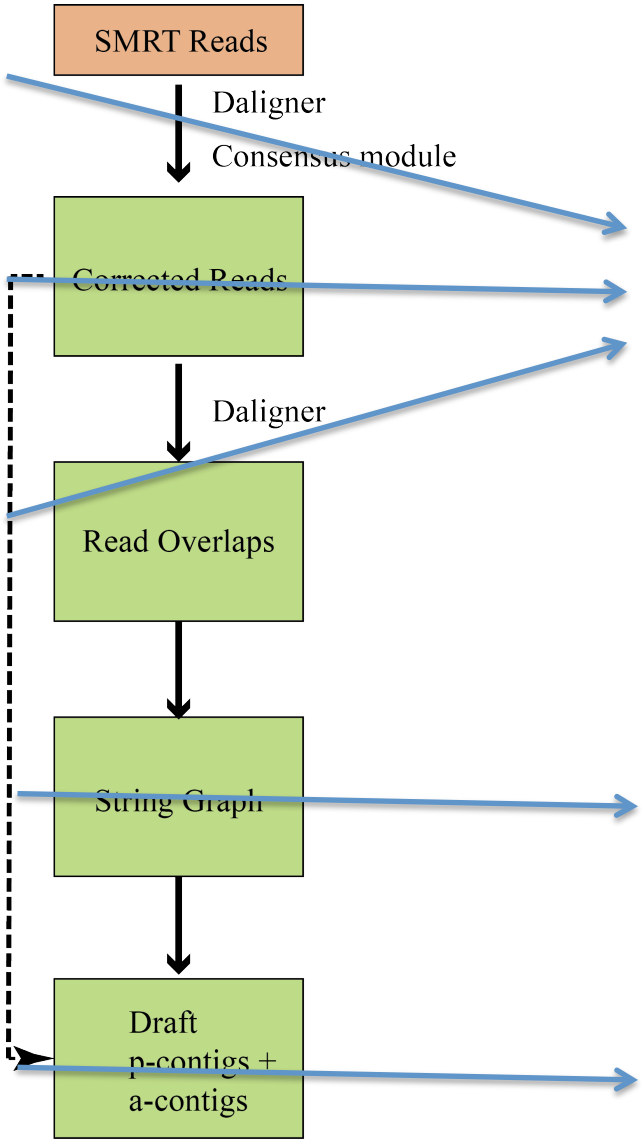


External code and internal modules used in FALCON and FALCON-Unzip

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

http://dx.doi.org/10.1101/056887

FALCON Overview

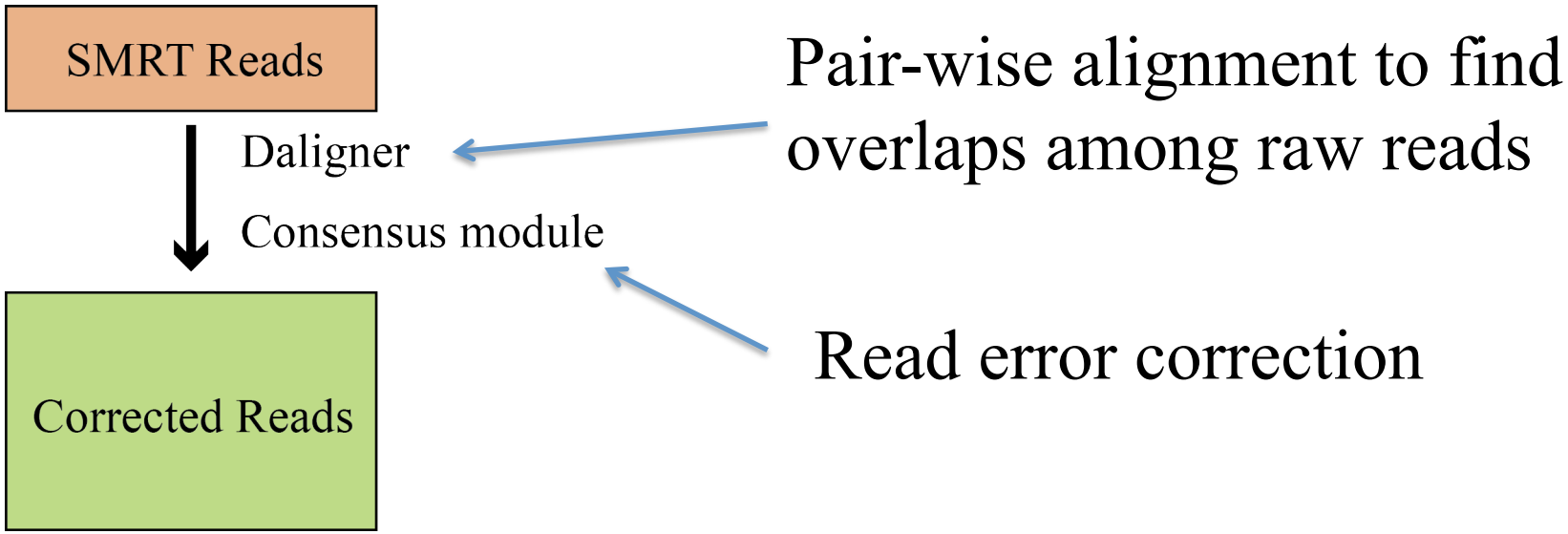


Overlap

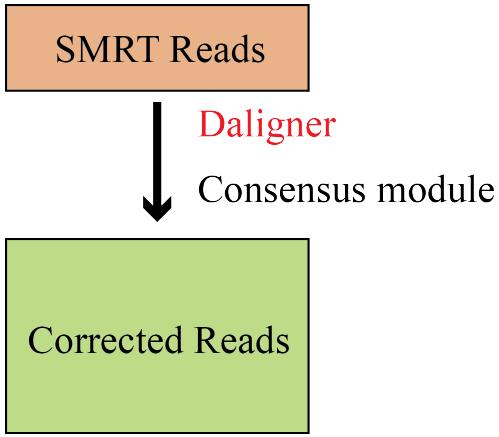
Building string graph

Resolve path

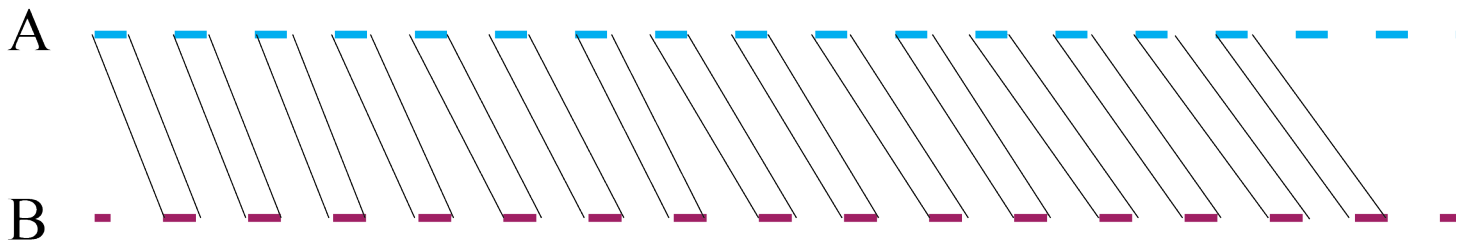
FALCON: Read Error Correction



DALIGNER



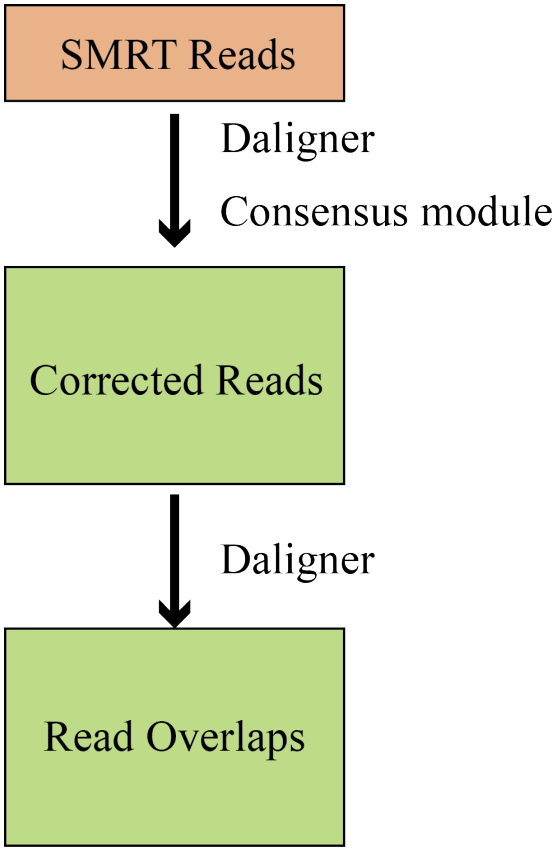
- Raw reads are split into blocks of data.
- Alignment is done using DALIGNER for each block of reads against all other blocks.



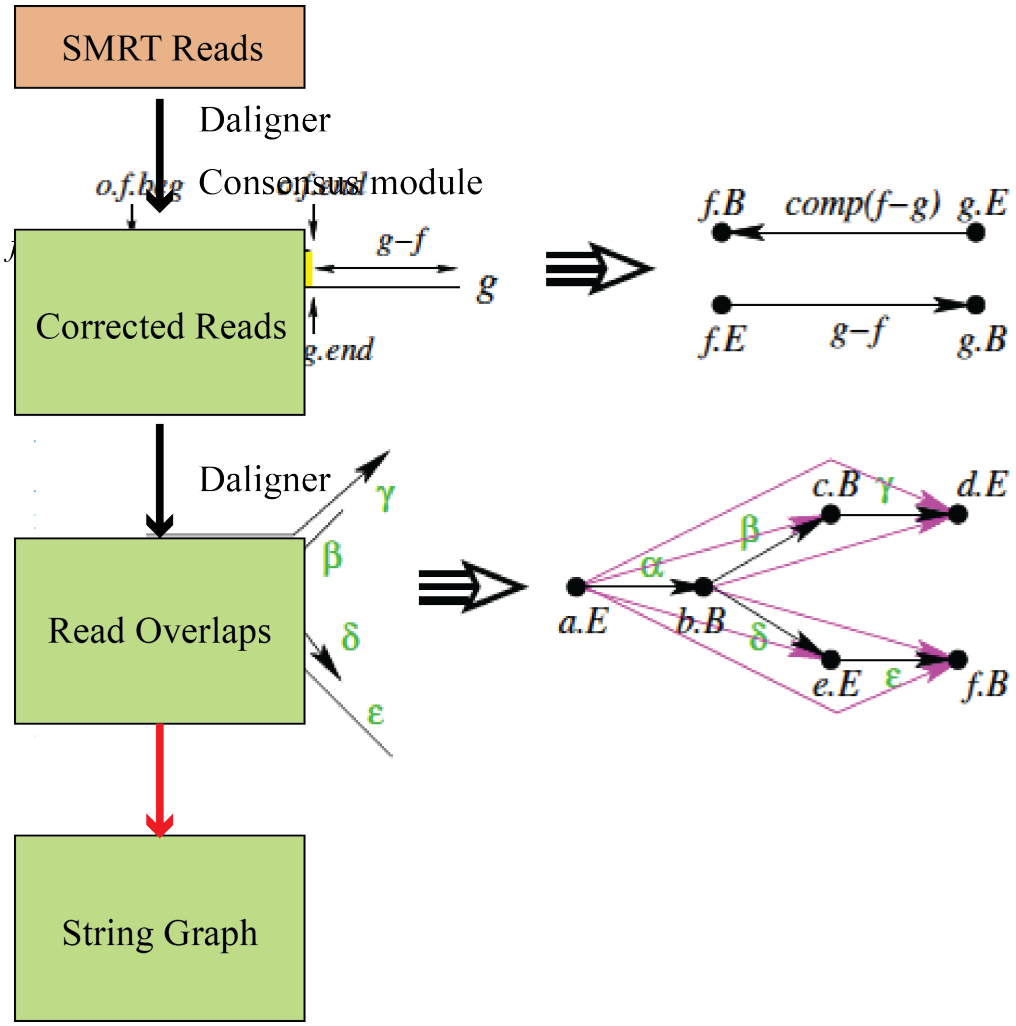
https://dazzlerblog.files.wordpress.com/2015/11/daligner.pdf

Chaisson MJ, Tesler G., *BMC Bioinformatics*, 13:238 (2012)

FALCON: Overlap of Error Corrected Reads

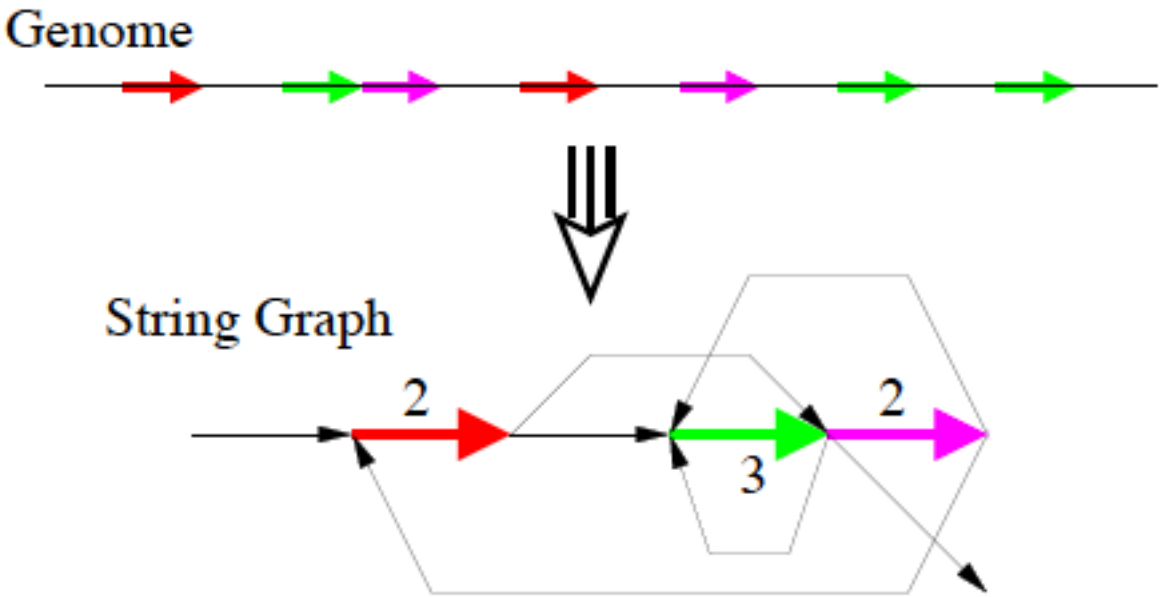


FALCON: Building String Graph



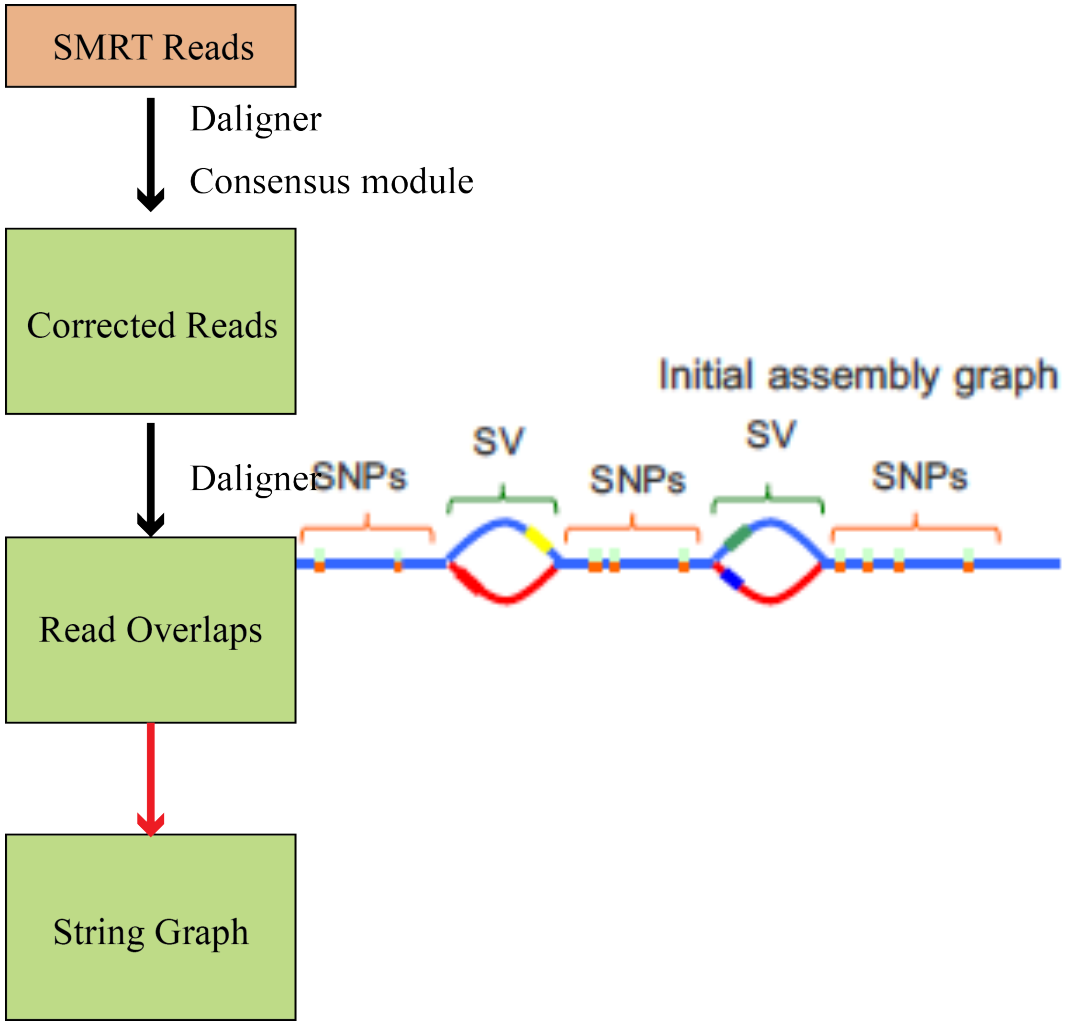
Myers, E.W. The fragment assembly string graph. Bioinformatics 21, ii79-ii85 (2005)

String Graph



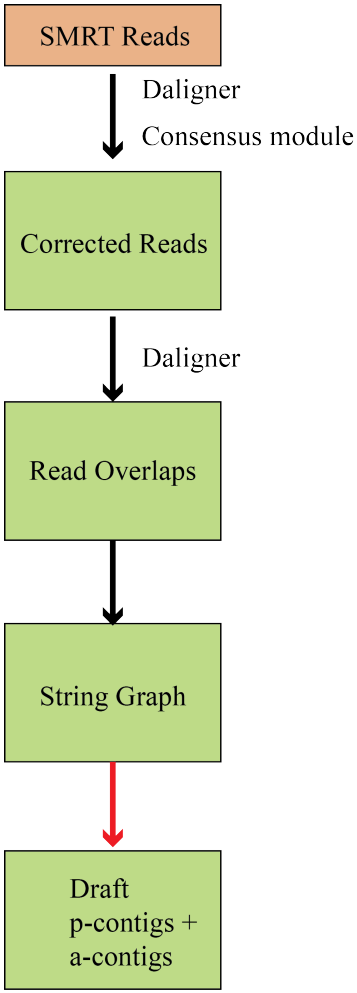
Myers, E.W. The fragment assembly string graph. Bioinformatics 21, ii79-ii85 (2005)

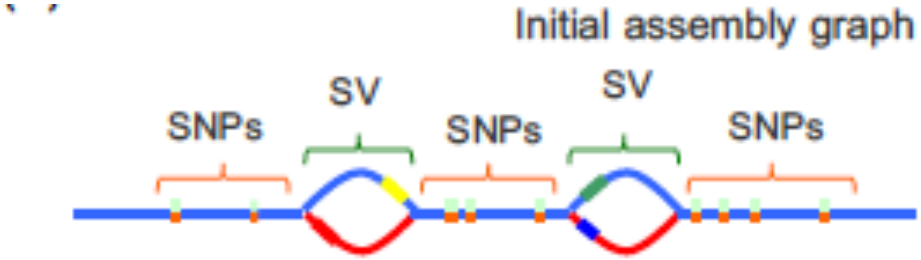
String Graph



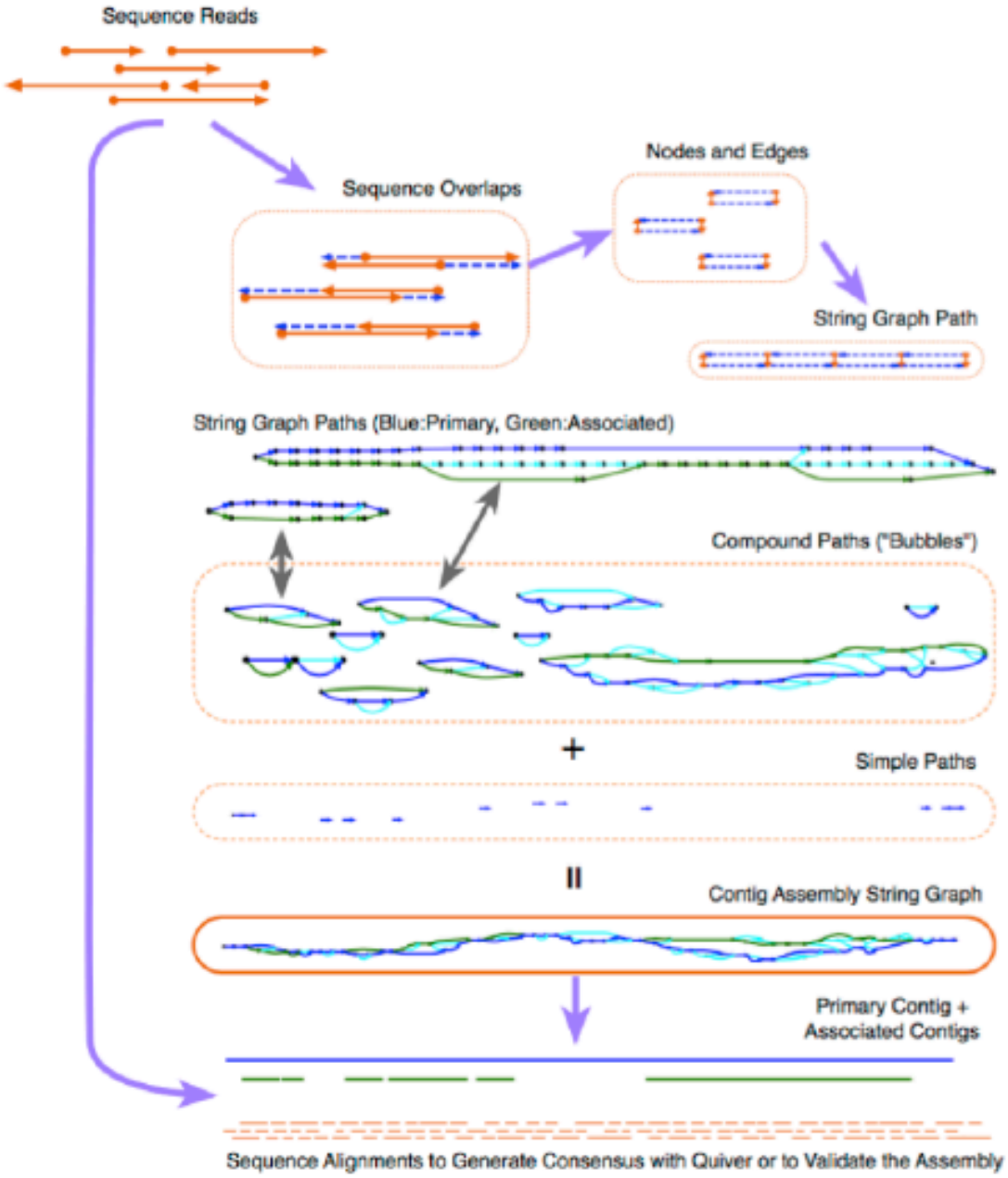
http://dx.doi.org/10.1101/056887

String Graph





http://dx.doi.org/10.1101/056887



http://dx.doi.org/10.1101/056887

Polish Draft Assembly: Quiver

- Finds the maximum quasi-likelihood template sequence given PacBio reads of the template
  - PacBio reads are modeled using a conditional random field approach that scores the quasi-likelihood of a read given a template sequence
  - Uses several additional QV covariates that the basecaller provides
- For long references, uses tiling windows across the reference to limit memory usage
- Uses the alignment provided by mapper (BLASR) to determine the grouping of reads
- Implicitly performs its own realignment, so it is highly sensitive to all variant types

Chin, C.-S. et al. *Nature methods* 10, 563-569 (2013)

Genome Assembly

After FALCON

- Haplotype genome
  - Assembly assessment, comparative genomics
- Diploid genome
  - FALCON_Unzip
  - Assembly assessment, comparative genomics

FALCON in Practice

Jie (Jessie) Li
PhD
Bioinformatics Analyst
Bioinformatics Core
UCD

Running FALCON --- Overview

- FALCON requires its specific python environment
- Required input files:
  - List of sequencing reads fasta
  - One configuration file

Running FALCON --- Python virtualenv

- srun --reservation=workshop --pty /bin/bash
- source $FALCON/bin/activate

Running FALCON --- Configuration File

```
[General]
  # list of files of the initial subread fasta files
  input_fofn = input.fofn

  input_type = raw
  #input_type = preads

  # The length cutoff used for seed reads used for initial mapping
  length_cutoff = 12000

  # The length cutoff used for seed reads usef for pre-assembly
  length_cutoff_pr = 12000

  # Cluster queue setting
  sge_option_da = -pe smp 8 -q jobqueue
  sge_option_la = -pe smp 2 -q jobqueue
  sge_option_pda = -pe smp 8 -q jobqueue
  sge_option_pla = -pe smp 2 -q jobqueue
  sge_option_fc = -pe smp 24 -q jobqueue
  sge_option_cns = -pe smp 8 -q jobqueue

  # concurrency settgin
  pa_concurrent_jobs = 32
  cns_concurrent_jobs = 32
  ovlp_concurrent_jobs = 32

  # overlapping options for Daligner
  pa_HPCdaligner_option =  -v -dal4 -t16 -e.70 -l1000 -s1000
  ovlp_HPCdaligner_option = -v -dal4 -t32 -h60 -e.96 -l500 -s1000

  pa_DBsplit_option = -x500 -s50
  ovlp_DBsplit_option = -x500 -s50

  # error correction consensus optione
  falcon_sense_option = --output_multi --min_idt 0.70 --min_cov 4 --local_match_count_threshold 2 --max_n_read 200 --n_core 6

  # overlap filtering options
  overlap_filtering_setting = --max_diff 100 --max_cov 100 --min_cov 20 --bestn 10
```

Running FALCON --- Configuration File

```
[General]
  # list of files of the initial subread fasta files
  input_fofn = input.fofn

  input_type = raw
  #input_type = preads
```

Input sequencing data file: input.fofn

```
ecoli.1.subreads.fasta
ecoli.2.subreads.fasta
ecoli.3.subreads.fasta
```

Running FALCON --- Configuration File

```
# The length cutoff used for seed reads used for initial mapping
  length_cutoff = 12000

# The length cutoff used for seed reads used for pre-assembly
  length_cutoff_pr = 12000
```

Running FALCON --- Configuration File

```
# Cluster queue setting
   sge_option_da = -pe smp 8 -q jobqueue
   sge_option_la = -pe smp 2 -q jobqueue
   sge_option_pda = -pe smp 8 -q jobqueue
   sge_option_pla = -pe smp 2 -q jobqueue
   sge_option_fc = -pe smp 24 -q jobqueue
   sge_option_cns = -pe smp 8 -q jobqueue
```

```
# Cluster queue setting
job_type = SLURM
sge_option_da = -c8 -N1 --mem-per-cpu 5000
sge_option_la = -c8 -N1 --mem-per-cpu 5000
sge_option_pda = -c12 -N1 --mem-per-cpu 6000
sge_option_pla = -c12 -N1 --mem-per-cpu 6000
sge_option_fc = -c12 -N1 --mem-per-cpu 6000
sge_option_cns = -c12 -N1 --mem-per-cpu 6000
```

Running FALCON --- Configuration File

```
# concurrency settgin
  pa_concurrent_jobs = 32
  cns_concurrent_jobs = 32
  ovlp_concurrent_jobs = 32
```

Running FALCON --- Configuration File

```
# overlapping options for Daligner
  pa_HPCdaligner_option =  -v -dal4 -t16 -e.70 -l1000 -s1000
  ovlp_HPCdaligner_option = -v -dal4 -t32 -h60 -e.96 -l500 -s1000
```

- Parameters for DALIGNER
- -dal4 specifies 4 serial calls to daligner command per job submission
- -l and –s set how many base pairs constitute the minimum local alignment and how frequently (in base pairs) these are recorded
- -t suppresses the use of any k-mer that occurs more than t times in either the subject or target block. It suppresses the memory useage by significantly over-reapresented k-mers.
- A better way to handle this is to use –M parameter, that sets the memory limit and asks the program to self adjust k-mer choice.

Running FALCON --- Configuration File

```
# overlapping options for Daligner
   pa_HPCdaligner_option =  -v -dal4 -t16 -e.70 -l1000 -s1000
   ovlp_HPCdaligner_option = -v -dal4 -t32 -h60 -e.96 -l500 -s1000
```

```
# overlapping options for Daligner
pa_HPCdaligner_option =  -v -dal128 -M32 -e.70 -l4800 -h480 -k18 -w8 -s1000
ovlp_HPCdaligner_option = -v -dal128 -M64 -e.96 -l2400 -h240 -k24 -s1000
```

- -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 window l that must be covered by exact-match k-mers
- -w specifies the width of diagonal bands searched for exact-match k-mers as 2^w (default w=6)
- -M specifies memory limit for each job
- https://dazzlerblog.wordpress.com/

Running FALCON --- Configuration File

```
pa_DBsplit_option = -x500 -s50
ovlp_DBsplit_option = -x500 -s50

pa_DBsplit_option = -x500 -s400
ovlp_DBsplit_option = -x500 -s400
```

- Parameters for how to split reads into database blocks
- -s sets the number of megabytes in each DB block: larger number generates a smaller number of longer jobs
- -x ignors all reads shorter than x
- https://dazzlerblog.wordpress.com/

Running FALCON --- Configuration File

```
# error correction consensus optione
   falcon_sense_option = --output_multi --min_idt 0.70 --min_cov 4
--local_match_count_threshold 2 --max_n_read 200 --n_core 6
   falcon_sense_skip_contained = true
```

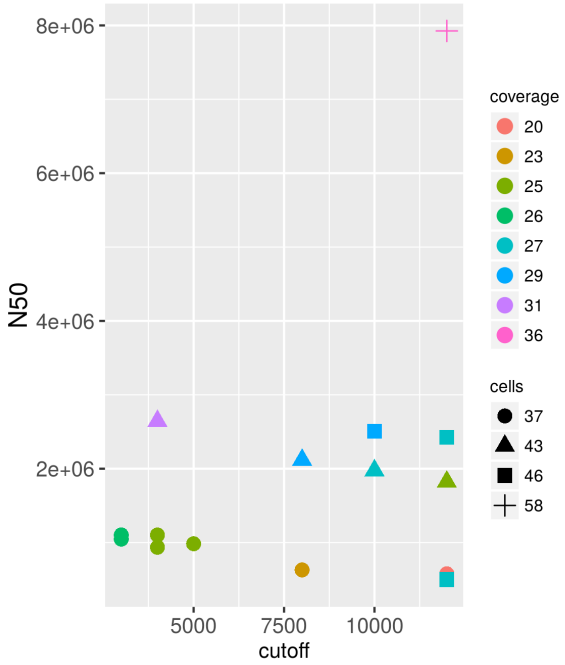
- Parameters for error correction of raw reads
- These settings work fine with many assemblies, but max_n_read may need to be lowered in highly repetitive genomes.

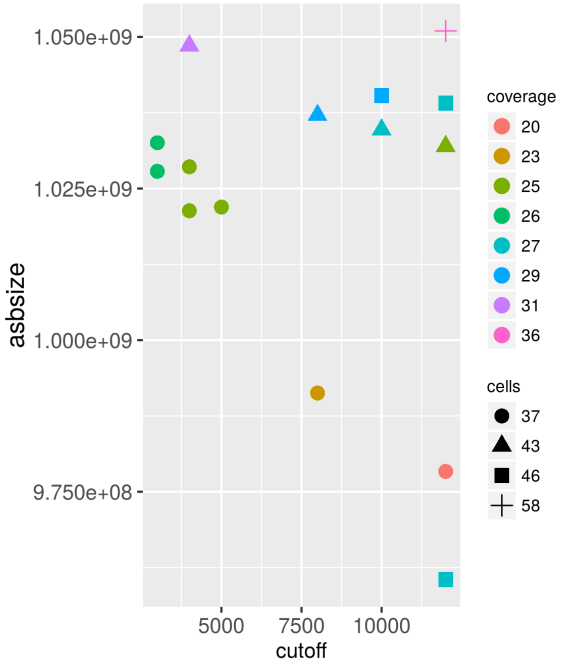
Running FALCON --- Configuration File

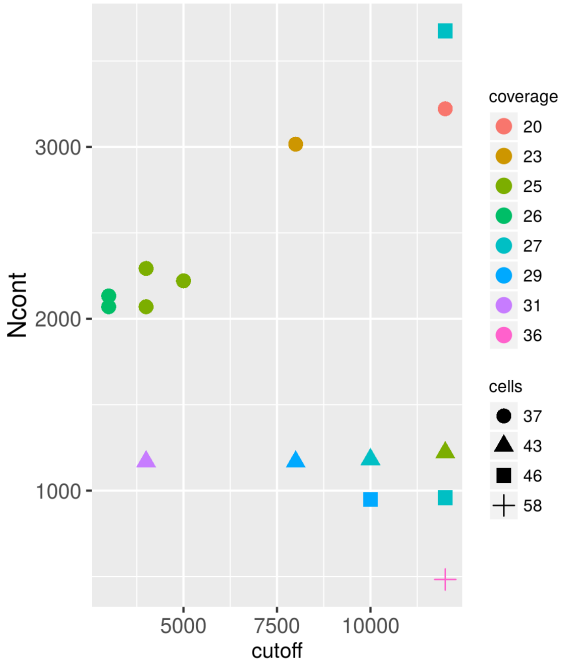
```
# overlap filtering options
  overlap_filtering_setting = --max_diff 100 --max_cov 100 --min_cov 2 --bestn 10
```

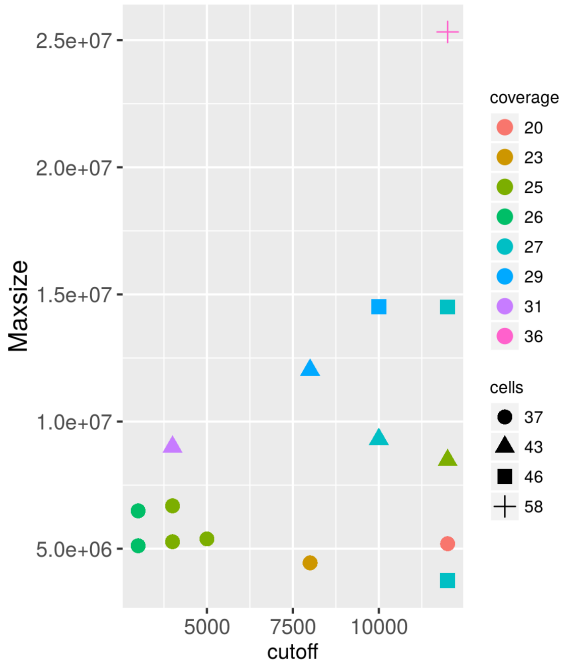
- Parameters for overlap of error-corrected reads
- min_cov specifies the minimum coverage in error-corrected reads in order to continue to extend contig. Low values promote contiguity at expense of misassembly
- max_diff specifies the maximum difference in coverage between ends of error-corrected reads, which usually indicates a repeat

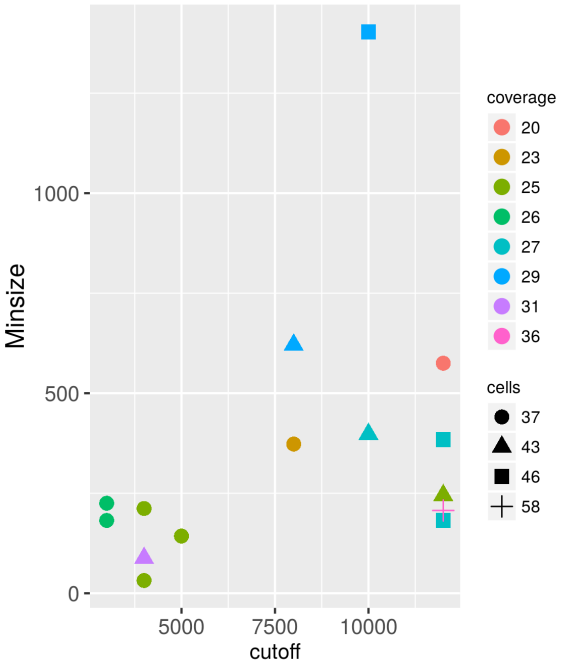
```
[General]
  # list of files of the initial subread fasta files
  input_fofn = input.fofn

  input_type = raw
  #input_type = preads

  # The length cutoff used for seed reads used for initial mapping
  length_cutoff = 12000

  # The length cutoff used for seed reads usef for pre-assembly
  length_cutoff_pr = 12000

  # Cluster queue setting
  sge_option_da = -pe smp 8 -q jobqueue
  sge_option_la = -pe smp 2 -q jobqueue
  sge_option_pda = -pe smp 8 -q jobqueue
  sge_option_pla = -pe smp 2 -q jobqueue
  sge_option_fc = -pe smp 24 -q jobqueue
  sge_option_cns = -pe smp 8 -q jobqueue

  # concurrency settgin
  pa_concurrent_jobs = 32
  cns_concurrent_jobs = 32
  ovlp_concurrent_jobs = 32

  # overlapping options for Daligner
  pa_HPCdaligner_option =  -v -dal4 -t16 -e.70 -l1000 -s1000
  ovlp_HPCdaligner_option = -v -dal4 -t32 -h60 -e.96 -l500 -s1000

  pa_DBsplit_option = -x500 -s50
  ovlp_DBsplit_option = -x500 -s50

  # error correction consensus optione
  falcon_sense_option = --output_multi --min_idt 0.70 --min_cov 4 --local_match_count_threshold 2 --max_n_read 200 --n_core 6

  # overlap filtering options
  overlap_filtering_setting = --max_diff 100 --max_cov 100 --min_cov 20 --bestn 10
```


N50
8e+06
6e+06
4e+06
2e+06
5000
7500
10000
cutoff
coverage
20
23
25
26
27
29
31
36
cells
37
43
46
58


asbsize
1.050e+09
1.025e+09
1.000e+09
9.750e+08
5000
7500
10000
cutoff
coverage
20
23
25
26
27
29
31
36
cells
37
43
46
58


Ncont
3000
2000
1000
5000
7500
10000
cutoff
coverage
20
23
25
26
27
29
31
36
cells
37
43
46
58


Maxsize
2.5e+07
2.0e+07
1.5e+07
1.0e+07
5.0e+06
5000
7500
10000
cutoff
coverage
20
23
25
26
27
29
31
36
cells
37
43
46
58


Minsize
1000
500
0
5000
7500
10000
cutoff
coverage
20
23
25
26
27
29
31
36
cells
37
43
46
58

FALCON --- OUTPUT

- 0-rawreads
  - job_* contains all DALIGNER output
  - m_* contains all merge jobs
  - Main output is inside preads. The out.*.fasta are the error corrected reads
- 1-preads_ovl
  - Overlaps among error-corrected reads generated from previous step
  - Similar to 0-rawreads setup, except for no consensus step
- 2-asm-falcon
  - The final output directory.
  - Information of the assembly graph and draft contigs (p_ctg.fa, a_ctg.fa)

FALCON --- OUTPUT

- Each finished job is associated with one *_done file.
- Time stamp of files are used to track the stage of the workflow.

FALCON --- COMMANDS

- srun --reservation=workshop --pty /bin/bash
- cp /share/biocore/workshops/Genome-Assembly-Workshop/examples/falcon/fc_run.cfg .
- cp /share/biocore/workshops/Genome-Assembly-Workshop/examples/falcon/input.fofn .
- module load falcon/0.4.2
- source $FALCON/bin/activate
- fc_run.py fc_run.cfg >& out.log &

Alignment

Aligned all supporting reads to the seed sequence

Tagging

Generate tags from the alignments

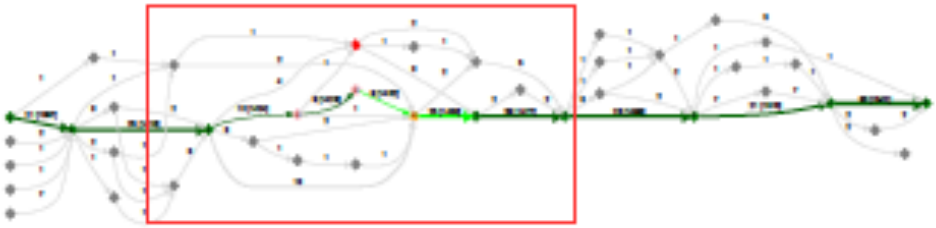
Generating Al ignment Graph

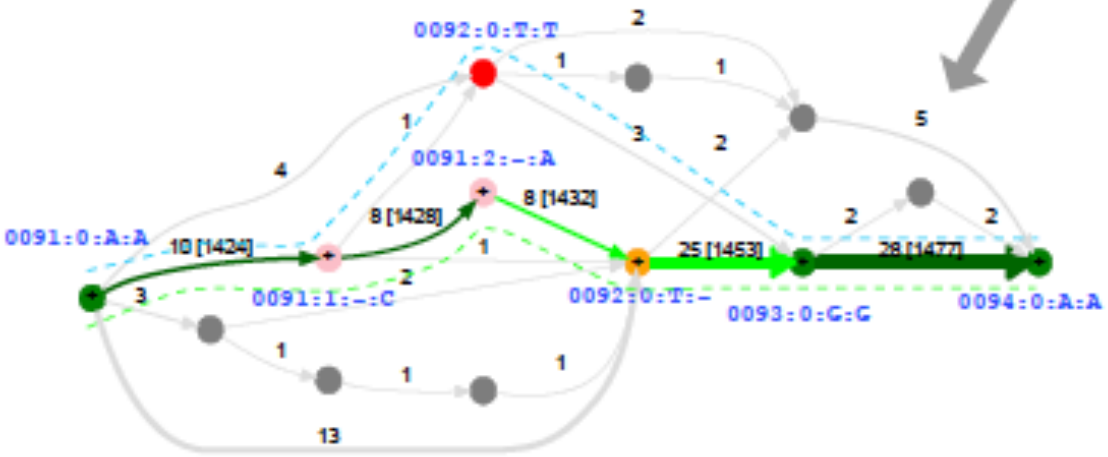
Grouping tags as alignment graph node. Generating edges from connected alignment tags

Generating Consensus from the Al ignment Graph

Find the optimum path as the consensus path

```
    p =                    p = 0123455678
    d =                    d = 0000001000
 seed = ATATTA-GGC      seed = ATATTA-GGC
read 1 = ATAT-ACGGC    read 1 = ATAT-ACGGC

    p =                    p = 011234555678
    d =                    d = 001000012000
 seed = AT-ATTA--GGC    seed = AT-ATTA--GGC
read 2 = ATCAT--CCGGC  read 2 = ATCAT--CCGGC
```





node 0093:0:G:G = p:d:(seed base):(supporting base)

Seed Sequence Path	Consensus Sequence Path
0091:0:A:A	0091:0:A:A
0092:0:T:T	0091:1:-:C
0093:0:G:G	0091:2:-:A
0094:0:A:A	0092:0:T:-
	0093:0:G:G
	0094:0:A:A

Seed Sequence	Consensus Sequence
ATGA	ACAGA

In this example, "T" insertion error corrected, and "CA" missing error recovered.

http://dx.doi.org/10.1101/056887