Assembly of *Ariolimax dolichophallus* using SOAPdenovo2

Charles Markello, Thomas Matthew, and Nedda Saremi

Image taken from Banana Slug Genome Project, S. Weber

SOAPdenovo Assembly Tool

- Short Oligonucleotide Analysis Package (SOAP) *de novo*
- de Bruijn based graph (DBG) assembler
- Collection of alignment and assembly programs developed at Beijing Genomics Institute (BGI)
- Has been applied to a number of genome sequencing projects including the Giant Panda

SOAPdenovo Data Flow

A) Library construction

B) Reads used to make de Bruijn graph

C) Removal of erroneous connections and tiny repeats

D) Break connections at repeat boundaries to output unambiguous sequence fragment as contigs

E) Use paired-end information to join unique contigs into scaffolds

F) Fill in intra-scaffolded gaps using paired-end extracted reads

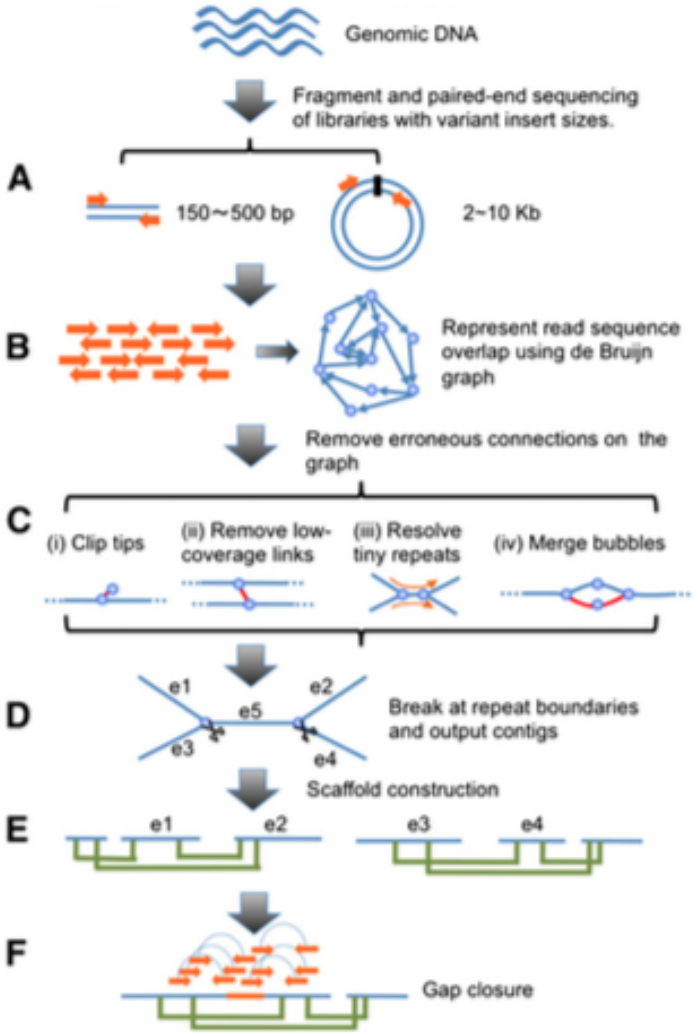


Image taken from Luo, R., et al

SOAP2.04 Updates

- Reduces memory consumption in de Bruijn graph construction
- Resolves more repeat regions in contig assembly
- Increases coverage and length in scaffold construction
- Improves gap closing
- Optimized for larger genomes and longer read datasets

Table 1 Evaluation of Assemblathon1 dataset assemblies

	Contig N50	Contig path NG50	Scaffold N50	Scaffold path NG50	Number of Structural Error	Substitution Error rate	Copy Number Error rate	Genome coverage (%)	Memory (G)	Run time (h)
V1	207,783	13,357	329,384	13,539	14,306	5.40E-05	9.14E-03	98.8	46	7
V1.05*	343,889	82,264	1,684,436	116,651	1,878	1.20E-05	6.75E-03	98.8	20	8
V2.0	357,238	111,365	15,077,357	170,432	1,414	4.25E-06	2.79E-03	98.8	20	10[§]
ALLPATHS-LG*	163,633	72,480	8,185,650	210,649	1,244	2.92E-06	6.71E-02	98.3	100	12

Contig and scaffold path NG50 were defined in Assemblathon1 [1].
*SOAPdenovo v1.05 and ALLPATHS-LG's evaluation result data were from [1].
§Time spent on filtering contamination was not included.

Arabidopsis thaliana sequencing project

Image taken from Luo, R. et al

SOAPdenovo employs multiple k-mers

- Similar to other DBG-based assemblers requiring k-mer selection, but can implement multiple k-mer strategy
- Selection is dependent on repetitiveness of genome, sequencing error, and heterozygosity
  - Smaller k-mer:
    - Minimizes sequencing errors and resolves heterozygotic regions
  - Larger k-mer:
    - Resolves short repeats

Multiple k-mers strategy combines range of k-mer lengths, resulting in longer contigs

de Bruijn Graph Assemblers

- de Bruijn graph assembly using k-mer specified
- Would in practice, give up on unresolvable repeats and yield fragmented assemblies
- Remove erroneous connections and solve short repeats
- Advantage is that:
  - O(N) work to build a de Bruijn graph, where N is the total length of all reads
  - Use sparse de Bruijn graph (DBG) to store only one out of every g (g<k) k-mers while trying to sub-sample evenly across the original DBG

SOAPdenovo2 Scaffolding

- Contigs will break at the repetitive sequences that can't be resolved with the chosen k-mer length
- 2 ideas were implemented to facilitate scaffolding:
  1) Build scaffolds heirarchically traversing from short insert size (200bp) to large insert sizes (10kbp)
  2) Repetitious contigs and contigs shorter than a threshold are masked before scaffolding to simplify contig graph
- Problem: heterozygous contigs influenced scaffold length
- Solution: use contig depth with location to keep only the heterozygous contig with the greatest depth

SOAPdenovo2 GapCloser

- In the scaffolds, regions between contigs are called gaps and represented by Ns
  - Most of gaps are repetitive patterns that were masked during scaffolding
- 2 step module in SOAPdenovo called GapCloser which fills gaps in the assembled scaffolds
  - Import and pre-process reads and scaffolds
  - Contigs are being extended to fill gaps iteratively

SOAPdenovo2 User Experience

- Configuration file needed to supply parameters to SOAP
  - Average insert size.
  - Paired end sequence orientation (forward-reverse or reverse-forward).
  - Assembly flags, indicates which parts of reads are used.
    - Contig assembly and/or Scaffold assembly.
    - Gap closure.
  - Read length cutoff.
  - Rank to determine order of read libraries to use for scaffold construction.
    - Ranking shorter insert length read data first is recommended.
  - Min # paired-end reads to connect 2 contigs or scaffolds.
  - Min alignment length between a read and contig for reliable read location.

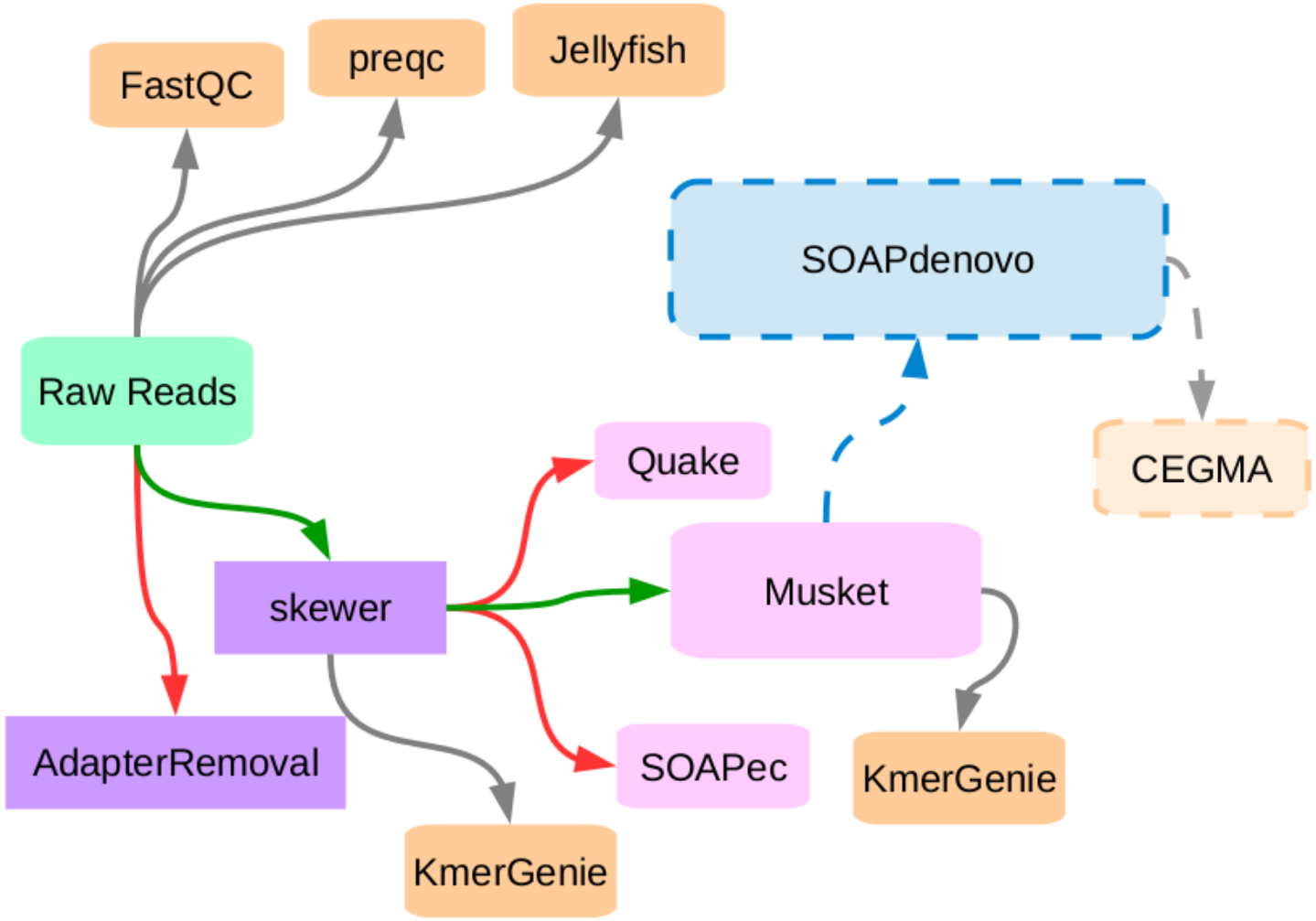
Sample config file

```
#maximal read length
max_rd_len=220
[LIB]
#average insert size
avg_ins=150
#if sequence needs to be reversed
reverse_seq=0
#in which part(s) the reads are used
asm_flags=3
#in which order the reads are used while scaffolding
rank=1
#fastq file for read 1
q1=/campusdata/BME235/data/slug/clean/run1_seqprep_quake/s_1_1_qseq_seqprep.cor.fastq.gz
#fastq file for read 2 always follows fastq file for read 1
q2=/campusdata/BME235/data/slug/clean/run1_seqprep_quake/s_1_2_qseq_seqprep.cor.fastq.gz
[LIB]
reverse_seq=0
asm_flags=3
rank=1
q=/campusdata/BME235/data/slug/clean/run1_seqprep_quake/
s_1_1_qseq_seqprep.cor_single.fastq.gz
[LIB]
reverse_seq=1
asm_flags=3
rank=1
q=/campusdata/BME235/data/slug/clean/run1_seqprep_quake/
s_1_2_qseq_seqprep.cor_single.fastq.gz
[LIB]
reverse_seq=0
asm_flags=3
```

Cons of SOAPdenovo2

- Sensitive to sequencing errors
  - Must exclude data from poor libraries, filter low-quality reads and use high quality/coverage reads for *de novo* assembly
- Multiple-copy genes or genes containing repetitive sequences may be fragmented in assembly
- Large computational memory requirement for DBG
- DBG construction is order-dependent
  - Different input reads ordering results in different graph structure
- Must specify estimated genome size
  - Variation in estimate alters starting point of graph traversals
    - Some nodes visited more than once, increasing computation time
- Recommend using full DBG on small or repetitive genomes

Current SOAPdenovo Pipeline

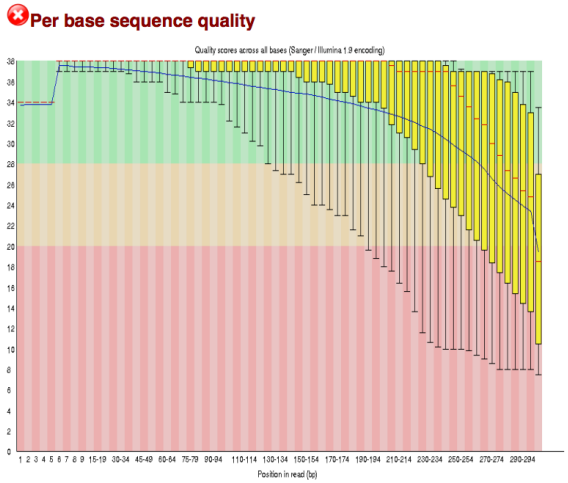


FastQC Results

- Per base sequence content skewed to Ts (all)
- Overrepresented sequences (possibly an adapter) (HiSeq SW018)

Overrepresented sequences

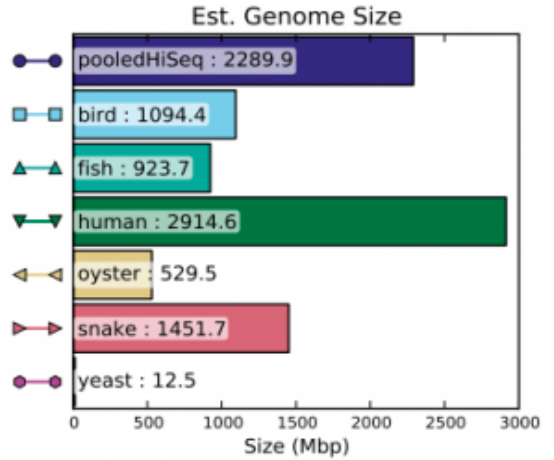
Sequence	Count	Percentage	Possible Source
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACTAGTTCCATCTCGTAT	2637049	1.8438408270948472	TruSeq Adapter, Index 10 (97% over 38bp)



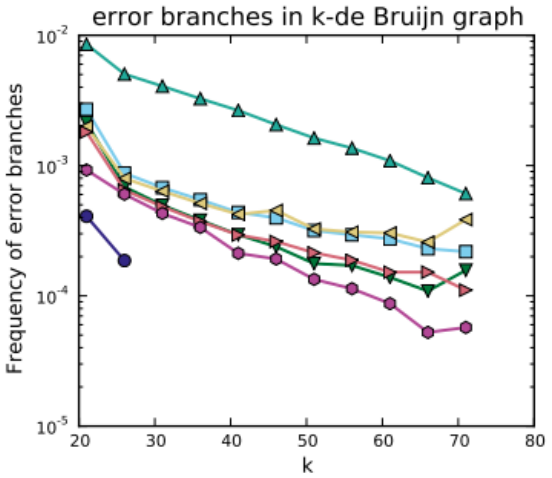
- Abnormal k-mers at start of reads (all)
- Base quality decreases at ends of reads (MiSeq)

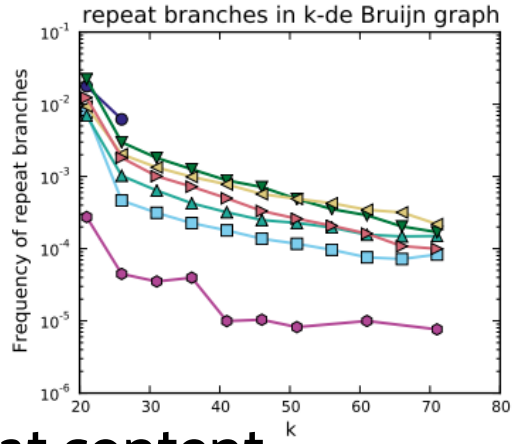
preqc Results

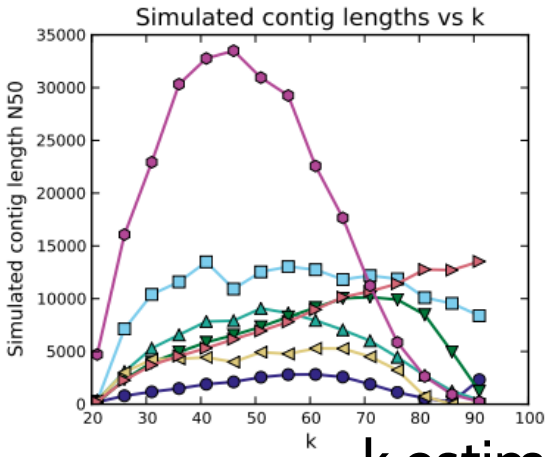
Genome size estimate 2.29Gb



Low error rate







High repeat content

k estimate

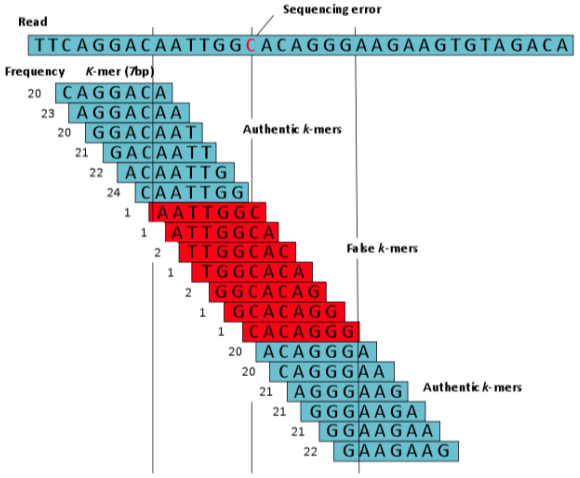
Skewer preferred over Adapter Trimmer

- Process of removing adapters used in sequencing

- AdapterRemoval
  - Incredibly slow
    - Single threaded ☹
    - Took about 12 hours to process ~4.5gb of read data from a single set of paired-end data.
- Skewer
  - Multithreaded ☺
  - User experience:
    - Easier to install and use
    - Much faster, takes about 3 hours using 32 threads.

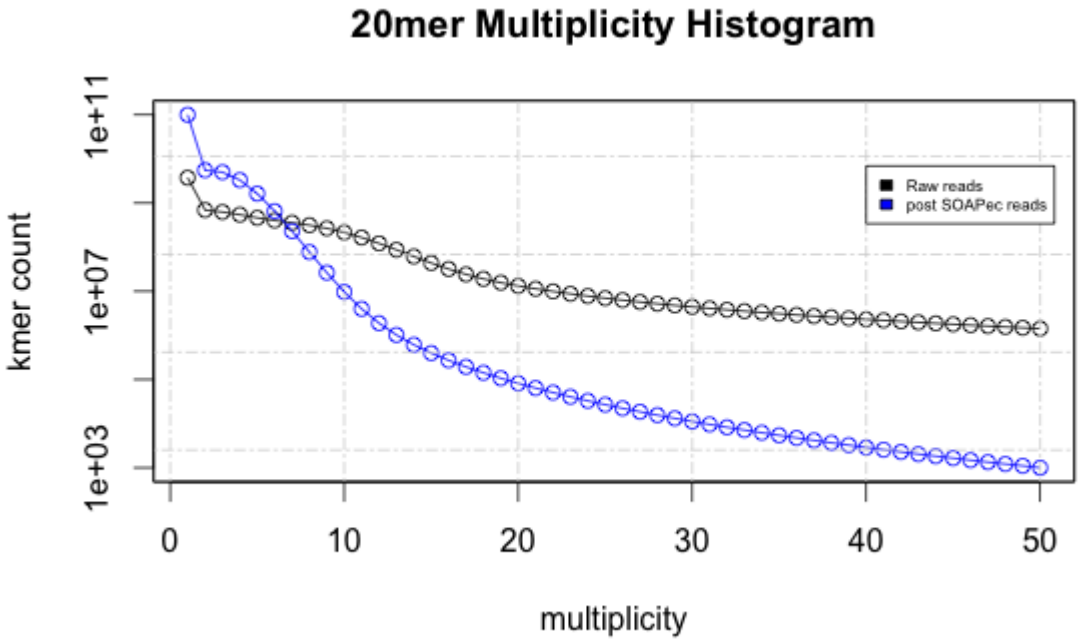
Error Correction with SOAPec

- Most low-frequency k-mers are generated by sequencing errors
- SOAPec corrects them based on k-mer frequency spectrums (KFS)
- In low frequency k-mers, determines which one base correction can transform false k-mers to authentic



SOAPec User Experience

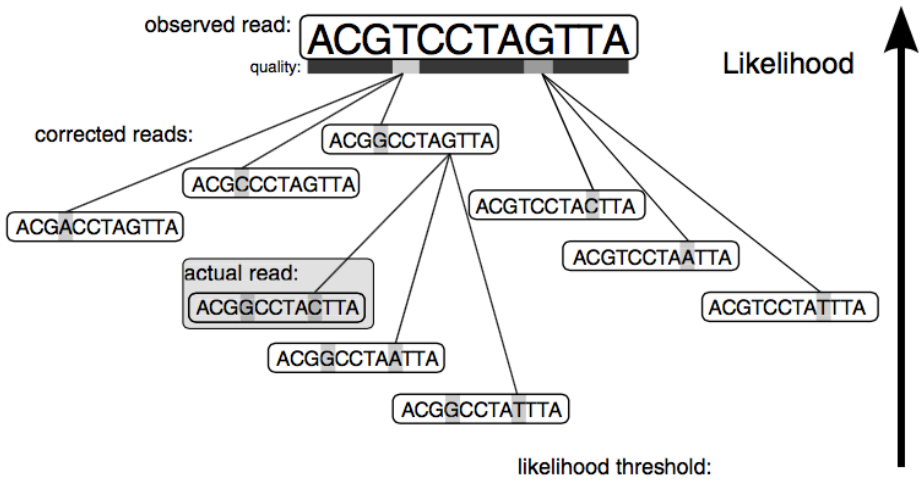
- Ran with adapter trimmed FASTQ files
- Each library run separately
- k-mer =20mer



- Small k-mers increased in count, attempted to use other EC tools

Error Correction with Quake

- Uses k-mer coverage and quality values to differentiate between trusted and untrusted k-mers
  - Untrusted k-mers have lower quality base calls
- Assigns cut-off to differentiate between trusted and untrusted k-mers based on distributions
- Reads containing untrusted k-mer are candidates for correction
- Find maximum likelihood set of corrections that makes all k-mers overlapping the region trusted



Quake User Experience

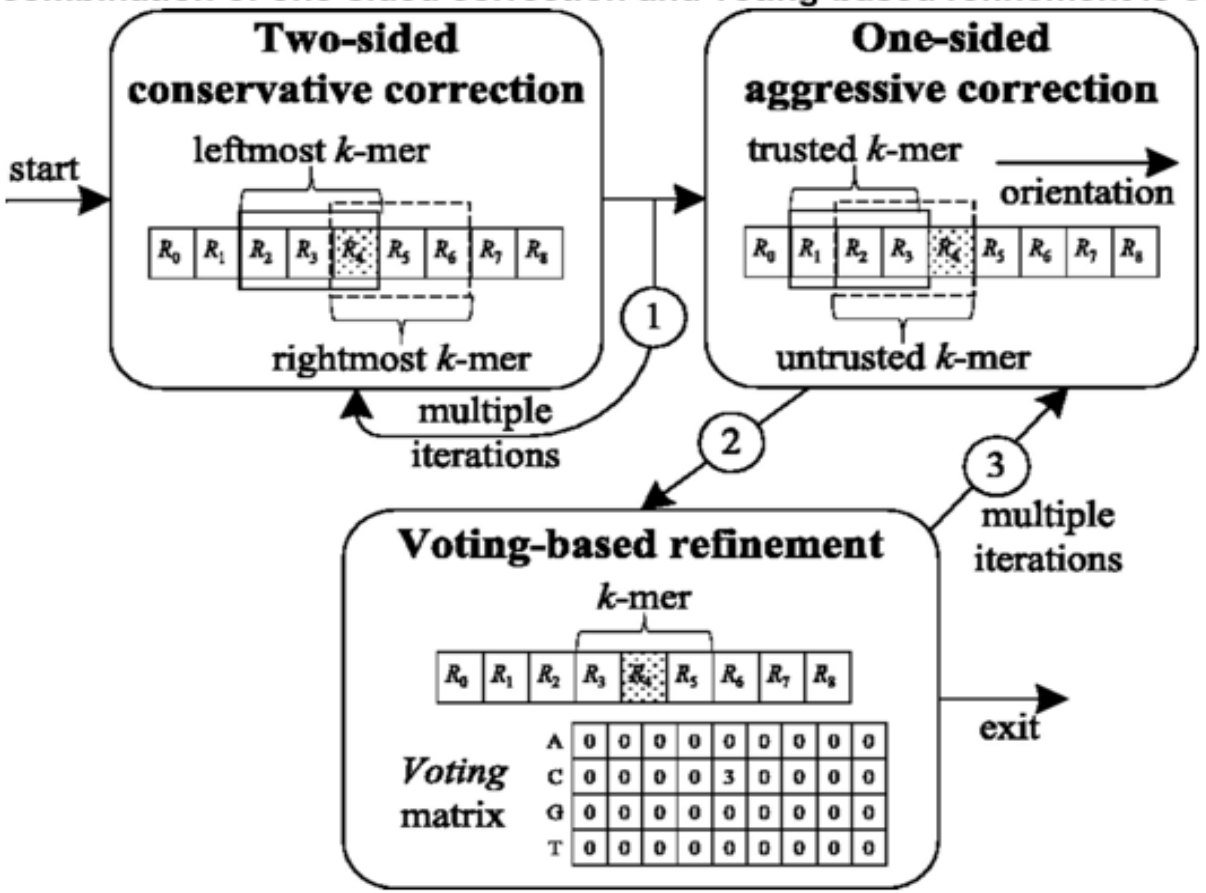
- Running each adapter trimmed library separately
- k-mer = 20
- Trial history:
  - Failed due to missing R package ‘VGAM’
  - Installed VGAM
  - Re-ran
  - Second run failed, no data output

Musket

- 2 stages:
  - K-mer spectrum construction
  - Error correction
- K-mer spectrum construction:
  - Counts # of non-unique k-mers using Bloom filter and hash table.
- Estimates coverage cut-off from the lowest density of the left valley.
  - Classifies trusted and untrusted kmers
- Error Correction in 3 techniques
  - two-sided conservative correction
  - one-sided aggressive correction
  - voting-based refinement

Musket

Error correction workflow: (i) two-sided conservative correction is performed using multiple iterations; (ii) one-sided aggressive correction is directly followed by voting-based refinement; and (iii) the combination of one-sided correction and voting-based refinement is conducted in

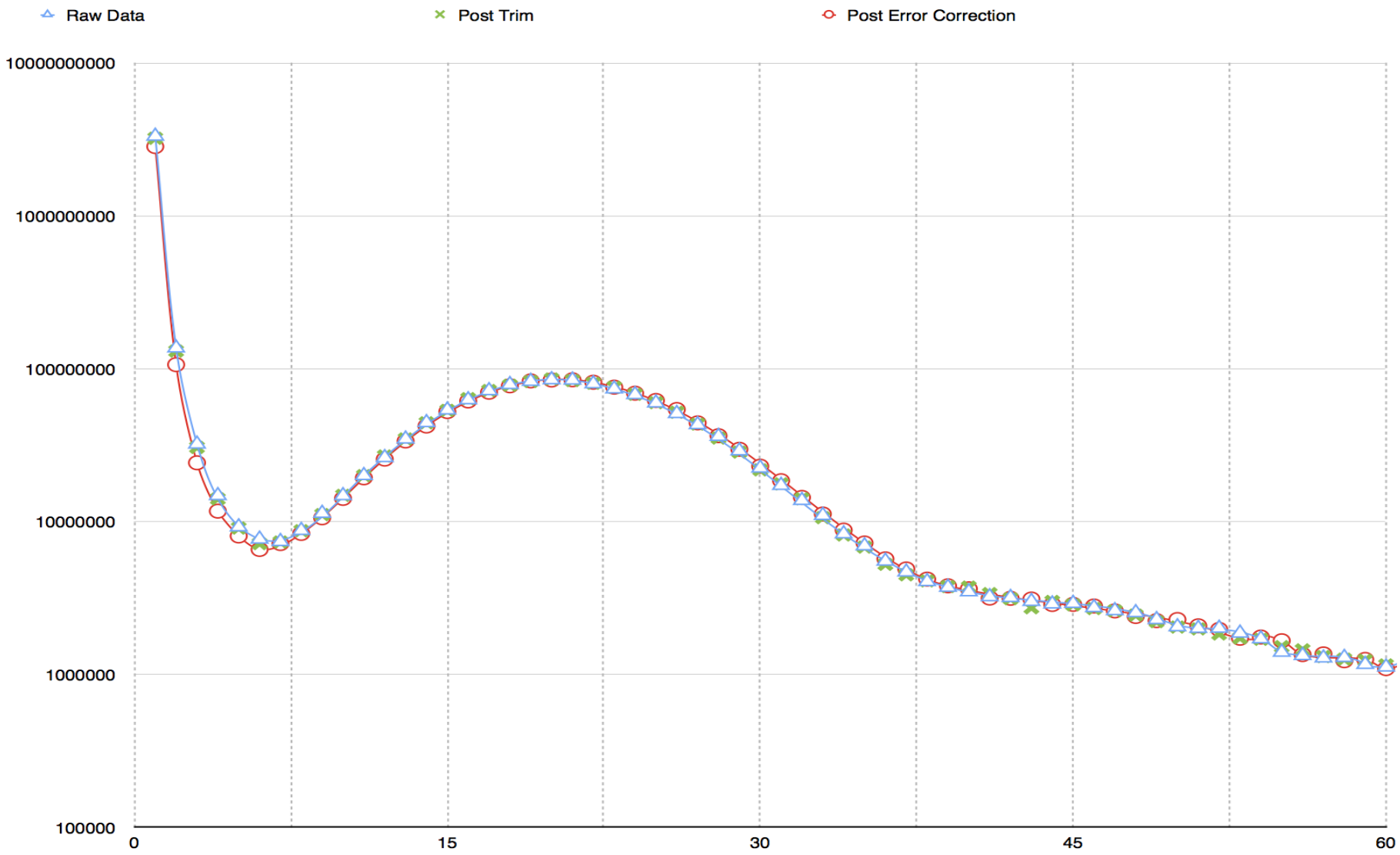


Yongchao Liu et al. Bioinformatics 2013;29:308-315

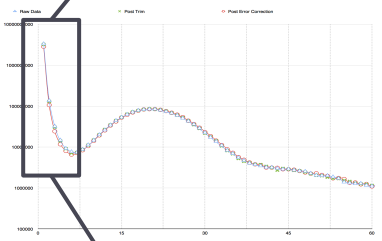
Musket User Experience

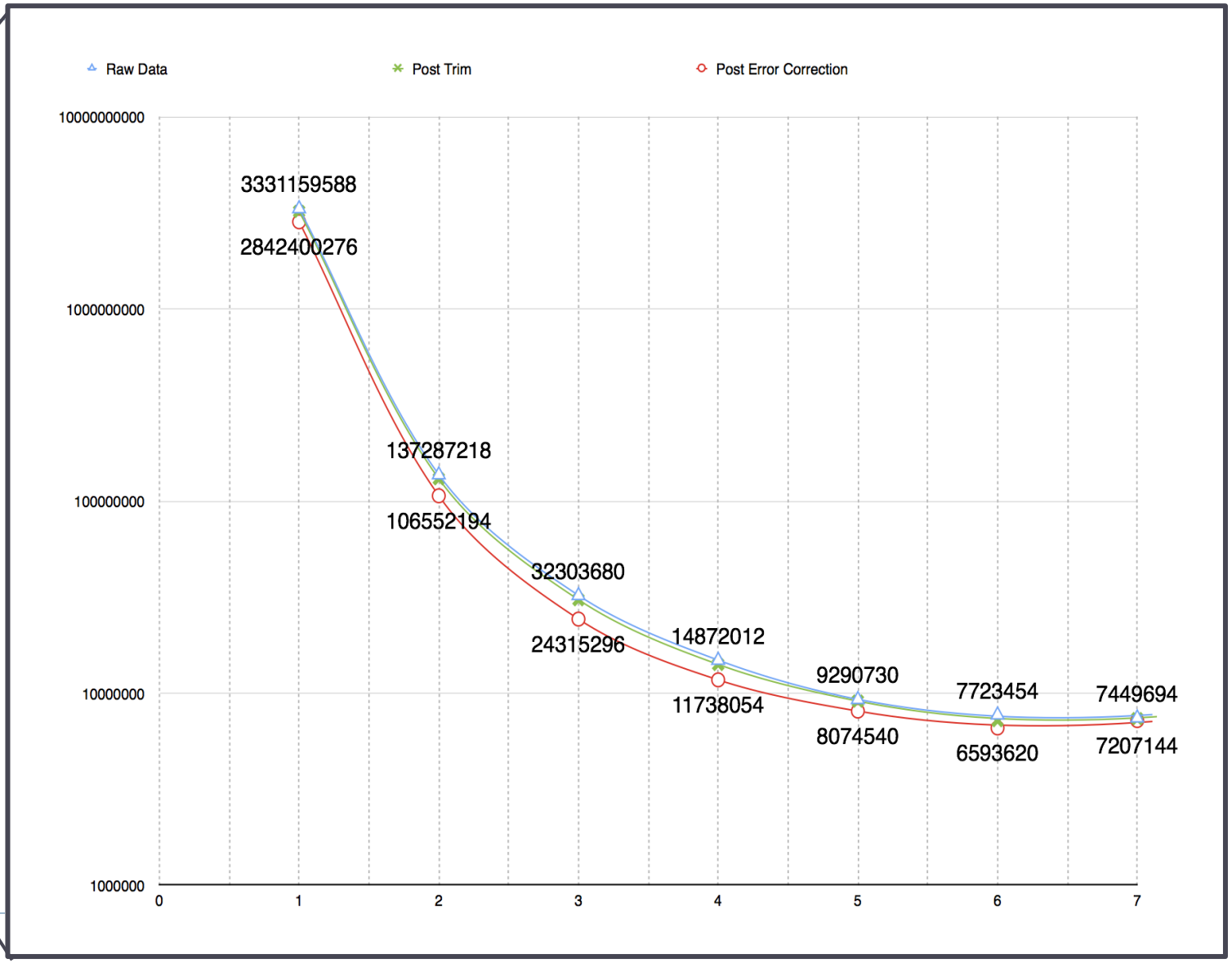
- Easy to setup and install
  - Only required setting max sequence/read length
- Chose to run algorithm with default 21-mer analysis
- Ran once to get accurate 21-mer total count.
  - Useful for setting specific parameter for balancing memory consumption between Bloom filters and hash tables.
- Ran again to get 21-mer multiplicity-by-frequency histogram for estimating max multiplicity cutoff filter post error correction.
- Takes about 8 hours running on 30 threads on 3 pairs of Illumina read data.

21-mer multiplicity histogram



21-mer multiplicity histogram zoom



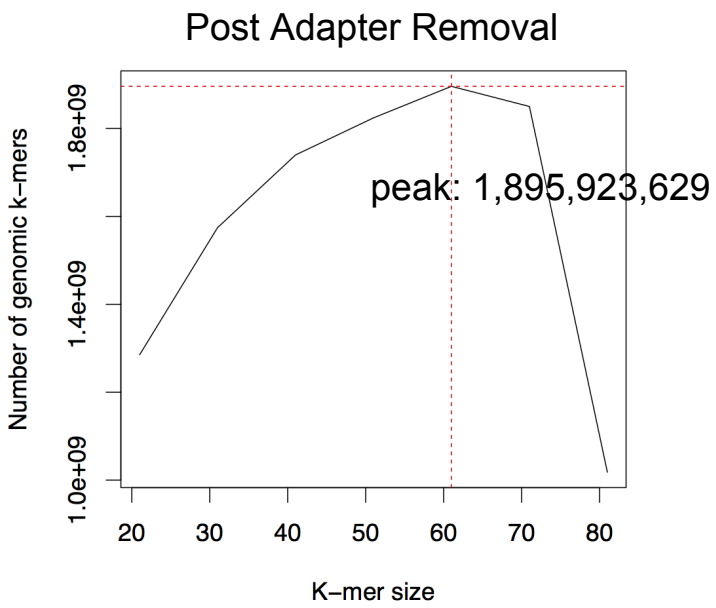


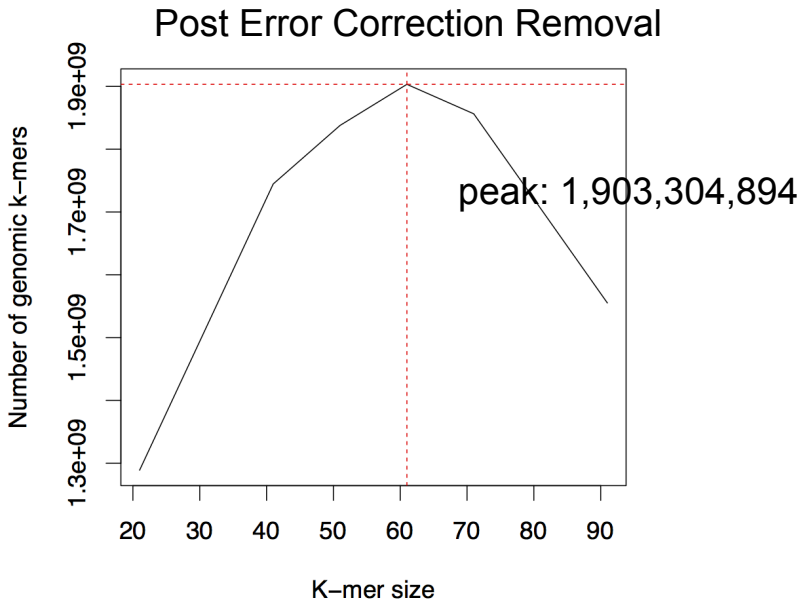
KmerGenie Analysis

- In de Bruijn-based assemblers, the most significant parameter is k
  - Choice of k is a trade-off between several effects
    - If selecting a short k, repeats longer than k can tangle the graph and break-up contigs
    - However, the longer k is, the higher the chance the k-mer will have an error in it
- KmerGenie constructs approximate abundance histograms to determine optimal k
- Best choice of k is one that provides the most distinct non-erroneous k-mers

KmerGenie Results

- Pooled all libraries of adapter trimmed and error corrected reads from skewer and musket, respectively
- Created histograms for k-mers in range 21 to 121 by 10
- Best k-mer = 61





Next Steps

- 61mer -- SOAPdenovo2
- optional multi-kmer selection:
  - range 51, 63 -- increased contig N50 (compute resources permitting)
- REAPR -- Evaluate assembly accuracy
- CEGMA -- Search for genes found in all eukaryotes
- Meta-assembly
- Re-map all read data to merged assembly

References

- Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z., … Wang, J. (2010). De novo assembly of human genomes with massively parallel short read sequencing. Genome Research, 20(2), 265–272. http://doi.org/10.1101/gr.097261.109
- Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., … Wang, J. (2012). SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience*, *1*(1), 18. http://doi.org/10.1186/2047-217X-1-18
- Chikhi, R., & Medvedev, P. (2014). Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, *30*(1), 31–37. http://doi.org/10.1093/bioinformatics/btt310
- Yang, X., Chockalingam, S. P., & Aluru, S. (2012). A survey of error-correction methods for next-generation sequencing. *Briefings in Bioinformatics*, bbs015. http://doi.org/10.1093/bib/bbs015
- Lee, H. C., Lai, K., Lorenc, M. T., Imelfort, M., Duran, C., & Edwards, D. (2011). Bioinformatics tools and databases for analysis of next-generation sequence data. *Briefings in Functional Genomics*, elr037. http://doi.org/10.1093/bfgp/elr037

Supplement Section

SOAPdenovo uses de Bruijn Alignment

- SOAPdenovo based on the de Bruijn graph structure
  - Nodes to represent all possible k-mers
  - Edges to represent perfect overlap of heads and tails of length k-1

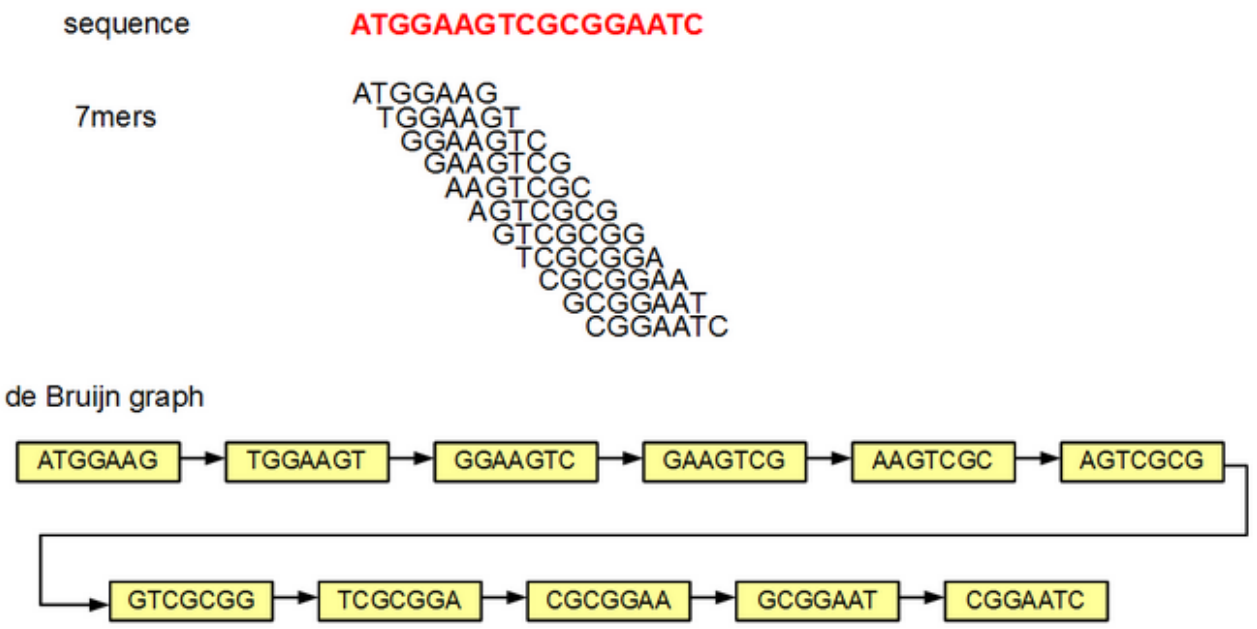


Image taken from http://www.homolog.us/Tutorials/index.php?p=2.1&s=1

SOAPdenovo2 Updates

- Use sparse de Bruijn graph (DBG) to store only one out of every g (g<k) k-mers while trying to sub-sample evenly across the original DBG
  - DBG reduced in size by factor of g
    - Reduced memory consumption 2-5 times in DBG construction step
- Allows for parallelization
  - Contig construction is dependent on number of threads specified
- Recognizes heterozygous contig pairs that resulted in two separate contigs in original SOAPdenovo
- Chimeric scaffolds incorrectly built are examined and fixed before extension with libraries of larger insert sizes

SOAPdenovo k-mer selection

Possible Run options

1) 63-mer

2) 127-mer

3) range(63, 127) `-m 127 -K 63`

4) range(13, 63) `-m 63 -K 13`

Output files

- *.contig
  - contig sequences without using mate pair information
- *.scafSeq
  - scaffold sequences

Compute time and Memory Requirements

- Contig N50 improves linearly from 10X to 30X coverage
- 150GB memory required for human genome assembly
-
-
-
-

SOAPdenovo Conditions

Possible Run options

1) 63-mer

2) 127-mer

3) range(63, 127) `-m 127 -K 63`

4) range(13, 63) `-m 63 -K 13`

Error Correction

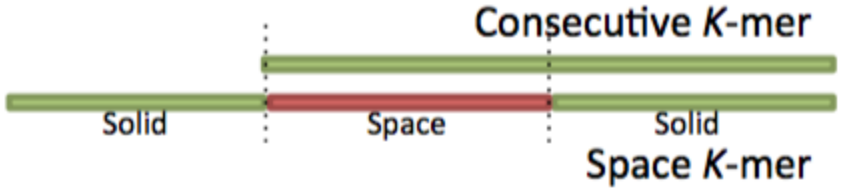
- 3 types:
  1. K-spectrum based
  2. suffix tree/array based
  3. MSA-based

K-spectrum Error Correction

- A k-mer occurring at least M times is termed solid, and is termed insolid otherwise
- Reads containing insolid k-mers are converted to solid ones with a minimum number of edit operations so that they contain only solid k-mers post-correction
  - Similar idea is used in SOAPec

SOAPec KFS Technique

- Define two kinds of k-mers
  1. consecutive k-mer [i to i+k] k bp in length
  2. space k-mer with gap s [i to i+s+k] k bp with gap s



- 1st approach (k<17) uses index table using 4^n bytes
- 2nd approach (k>17) stores k-mers and frequencies in hash table using G*2k

SOAPec ec Technique

- Import k-mer frequency tables into memory
- Divide k-mers into low and high frequency
- Reads with low frequency are considered possible errors and passed to next correction stage
- Aim of error correction is to convert min false k-mers to authentic k-mers with one correction
-

EC with Quake in-depth

- Increment k-mer's coverage by the product of the probabilities that the base calls in the k-mer are correct as defined by the quality values (q-mer counting)
  - better differentiates between true k-mers sequenced to low coverage and error k-mers that occurred multiple times due to bias or repetitive sequence
- Histogram of two distributions, true and error k-mers
  - must choose cut-off to differentiate between
    - trusted k-mers as a mixture of Gaussian and Zeta distributions
    - untrusted k-mers as Gamma distribution
  - Convert each read to be free of untrusted k-mers
    - Heuristically locate erroneous region in r ising insolid k-mers, if cover 3' end trimming is applied
    - Greedily correct bases with low quality scores until all k-mers are solid

slide assignments

Charlie: 15, 21-25, 28

Nedda: 13,14,16-20, 26-27

Thomas: 1-12