De novo meta-assembly of ultra-deep sequencing data

Hamid Mirebrahim[1], Timothy J. Close[2] and Stefano Lonardi[1]

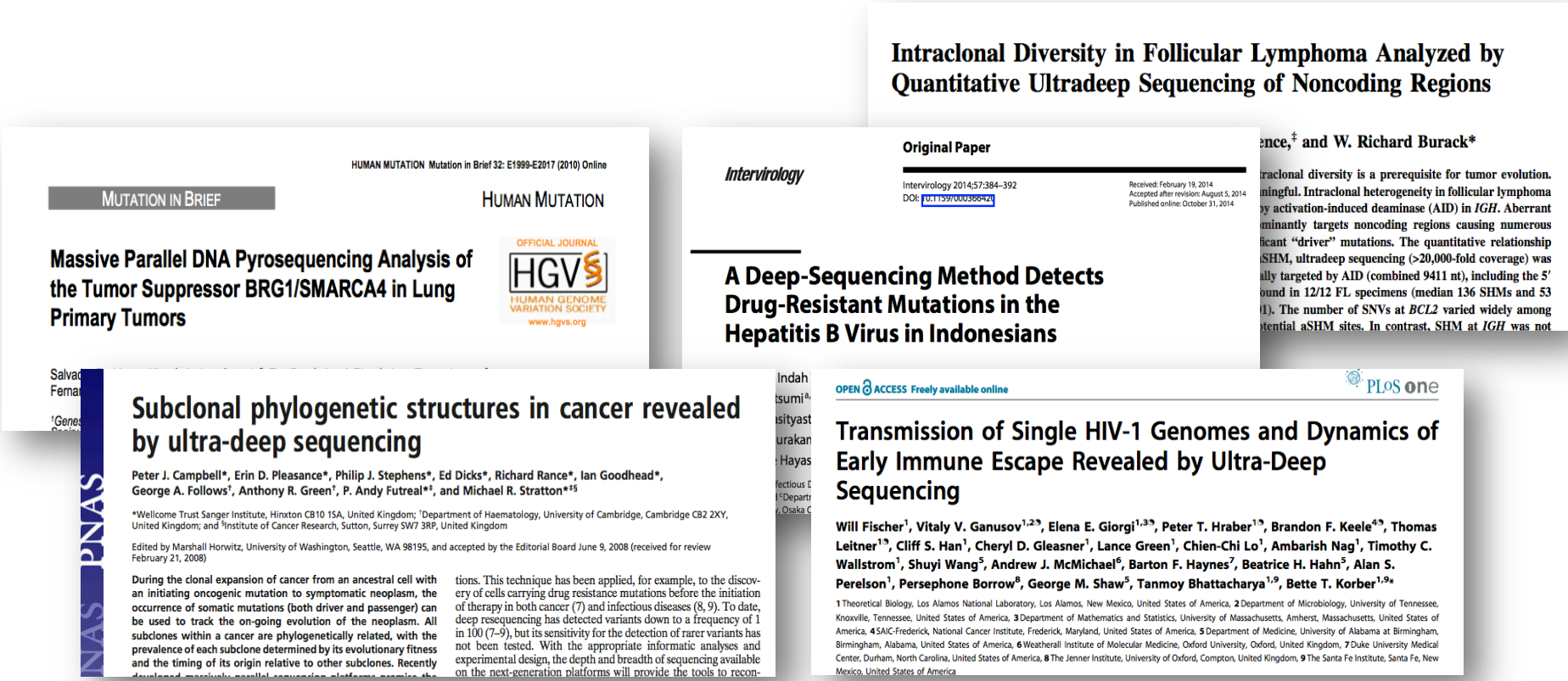
[1]Department of Computer Science and Engineering

[2]Department of Botany and Plant Sciences

UNIVERSITY OF CALIFORNIA
UC RIVERSIDE

DUBLIN
ISMB ECCB
2015
JULY 10-14

Ultra-deep sequencing (>1,000x coverage) is possible and feasible, expected to become more common



Intraclonal Diversity in Follicular Lymphoma Analyzed by Quantitative Ultradeep Sequencing of Noncoding Regions

ence,‡ and W. Richard Burack*

raclonal diversity is a prerequisite for tumor evolution. ningful. Intraclonal heterogeneity in follicular lymphoma y activation-induced deaminase (AID) in *IGH*. Aberrant minantly targets noncoding regions causing numerous ficant "driver" mutations. The quantitative relationship SHM, ultradeep sequencing (>20,000-fold coverage) was ally targeted by AID (combined 9411 nt), including the 5′ ound in 12/12 FL specimens (median 136 SHMs and 53 1). The number of SNVs at *BCL2* varied widely among otential aSHM sites. In contrast, SHM at *IGH* was not

HUMAN MUTATION Mutation in Brief 32: E1999-E2017 (2010) Online

MUTATION IN BRIEF

HUMAN MUTATION

Massive Parallel DNA Pyrosequencing Analysis of the Tumor Suppressor BRG1/SMARCA4 in Lung Primary Tumors

OFFICIAL JOURNAL HGVS HUMAN GENOME VARIATION SOCIETY www.hgvs.org

Original Paper

Intervirology

Intervirology 2014;57:384–392

DOI: 10.1159/000368420

Received: February 19, 2014
Accepted after revision: August 5, 2014
Published online: October 31, 2014

A Deep-Sequencing Method Detects Drug-Resistant Mutations in the Hepatitis B Virus in Indonesians

Subclonal phylogenetic structures in cancer revealed by ultra-deep sequencing

Peter J. Campbell*, Erin D. Pleasance*, Philip J. Stephens*, Ed Dicks*, Richard Rance*, Ian Goodhead*, George A. Follows†, Anthony R. Green†, P. Andy Futreal*‡, and Michael R. Stratton*‡§

*Wellcome Trust Sanger Institute, Hinxton CB10 1SA, United Kingdom; †Department of Haematology, University of Cambridge, Cambridge CB2 2XY, United Kingdom; and §Institute of Cancer Research, Sutton, Surrey SW7 3RP, United Kingdom

Edited by Marshall Horwitz, University of Washington, Seattle, WA 98195, and accepted by the Editorial Board June 9, 2008 (received for review February 21, 2008)

During the clonal expansion of cancer from an ancestral cell with an initiating oncogenic mutation to symptomatic neoplasm, the occurrence of somatic mutations (both driver and passenger) can be used to track the on-going evolution of the neoplasm. All subclones within a cancer are phylogenetically related, with the prevalence of each subclone determined by its evolutionary fitness and the timing of its origin relative to other subclones. Recently

tions. This technique has been applied, for example, to the discovery of cells carrying drug resistance mutations before the initiation of therapy in both cancer (7) and infectious diseases (8, 9). To date, deep resequencing has detected variants down to a frequency of 1 in 100 (7–9), but its sensitivity for the detection of rarer variants has not been tested. With the appropriate informatic analyses and experimental design, the depth and breadth of sequencing available on the next-generation platforms will provide the tools to recon-

OPEN ACCESS Freely available online

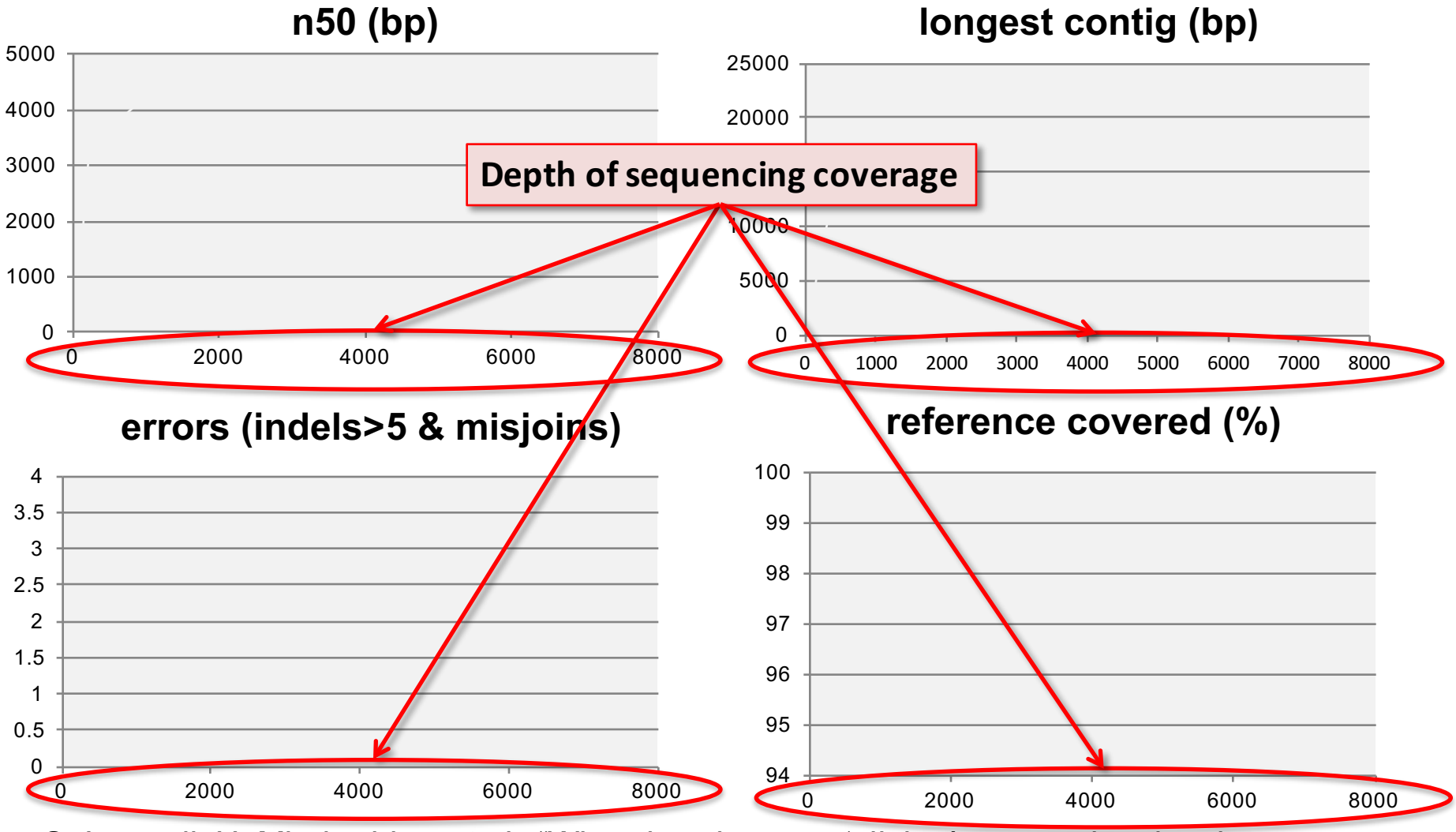
PLoS one

Transmission of Single HIV-1 Genomes and Dynamics of Early Immune Escape Revealed by Ultra-Deep Sequencing

Will Fischer[1], Vitaly V. Ganusov[1,2☯], Elena E. Giorgi[1,3☯], Peter T. Hraber[1☯], Brandon F. Keele[4☯], Thomas Leitner[1☯], Cliff S. Han[1], Cheryl D. Gleasner[1], Lance Green[1], Chien-Chi Lo[1], Ambarish Nag[1], Timothy C. Wallstrom[1], Shuyi Wang[5], Andrew J. McMichael[6], Barton F. Haynes[7], Beatrice H. Hahn[5], Alan S. Perelson[1], Persephone Borrow[8], George M. Shaw[5], Tanmoy Bhattacharya[1,9], Bette T. Korber[1,9]*

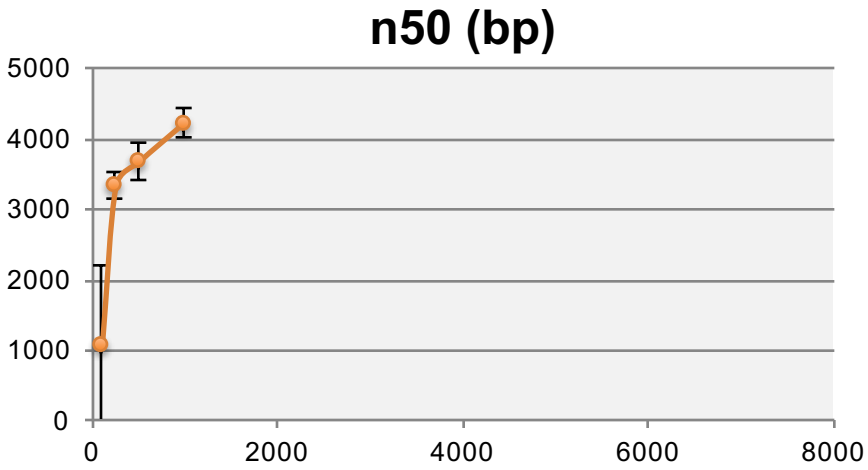
1 Theoretical Biology, Los Alamos National Laboratory, Los Alamos, New Mexico, United States of America, 2 Department of Microbiology, University of Tennessee, Knoxville, Tennessee, United States of America, 3 Department of Mathematics and Statistics, University of Massachusetts, Amherst, Massachusetts, United States of America, 4 SAIC-Frederick, National Cancer Institute, Frederick, Maryland, United States of America, 5 Department of Medicine, University of Alabama at Birmingham, Birmingham, Alabama, United States of America, 6 Weatherall Institute of Molecular Medicine, Oxford University, Oxford, United Kingdom, 7 Duke University Medical Center, Durham, North Carolina, United States of America, 8 The Jenner Institute, University of Oxford, Compton, United Kingdom, 9 The Santa Fe Institute, Santa Fe, New Mexico, United States of America

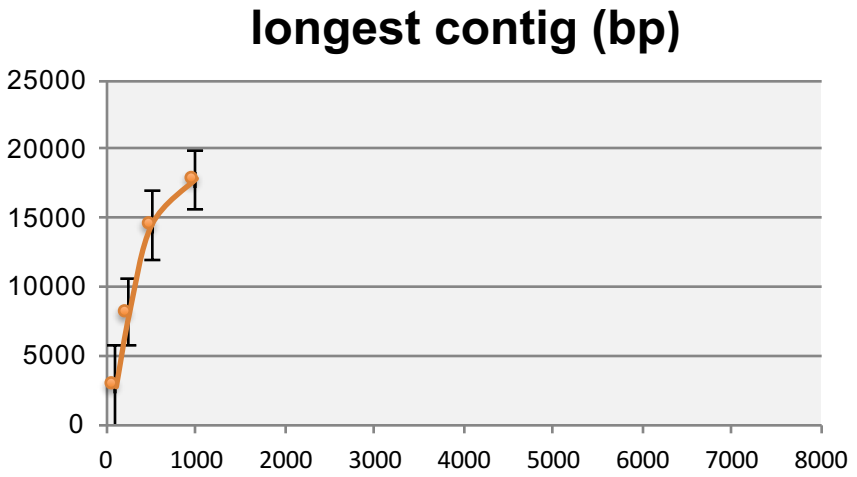
Expectation: more data → 'better' assemblies

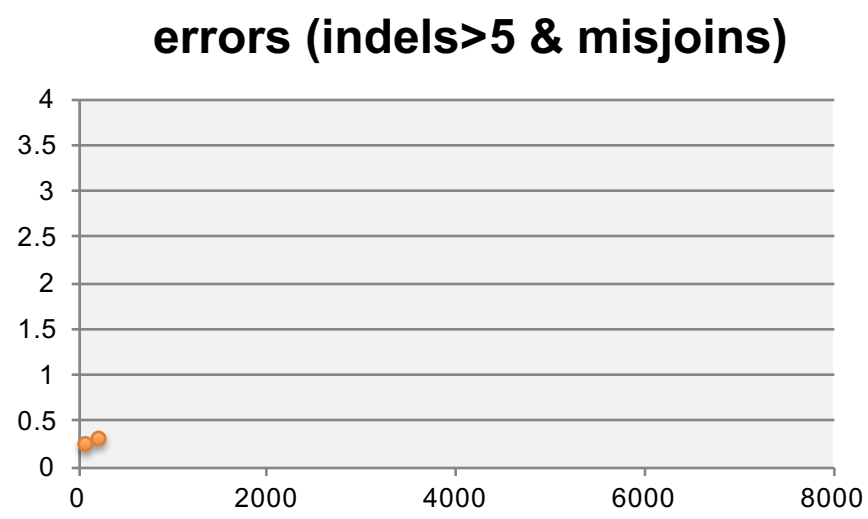


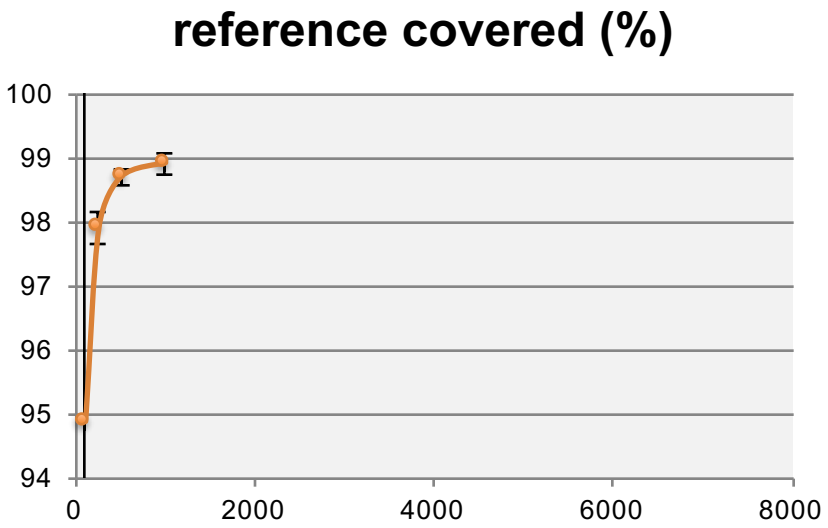
S. Lonardi, H. Mirebrahim, *et al.*, "When less is more: 'slicing' sequencing data improves read decoding accuracy and *de novo* assembly quality", *Bioinformatics*, 2015.

Expectation: more data → 'better' assemblies



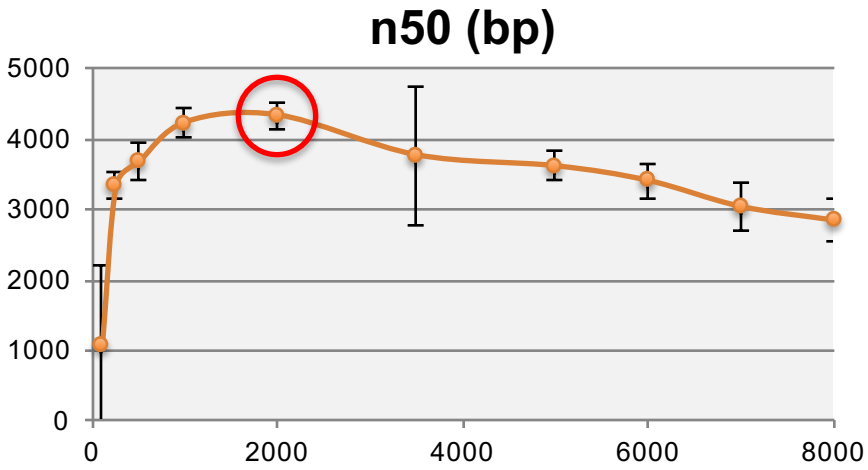


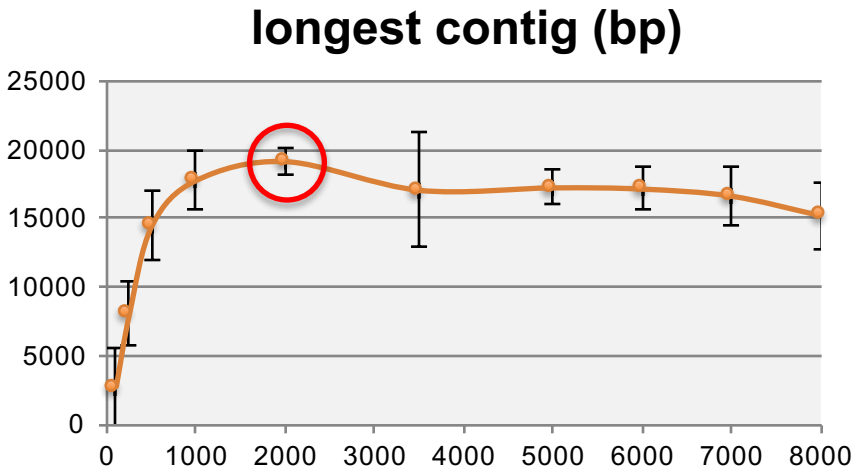


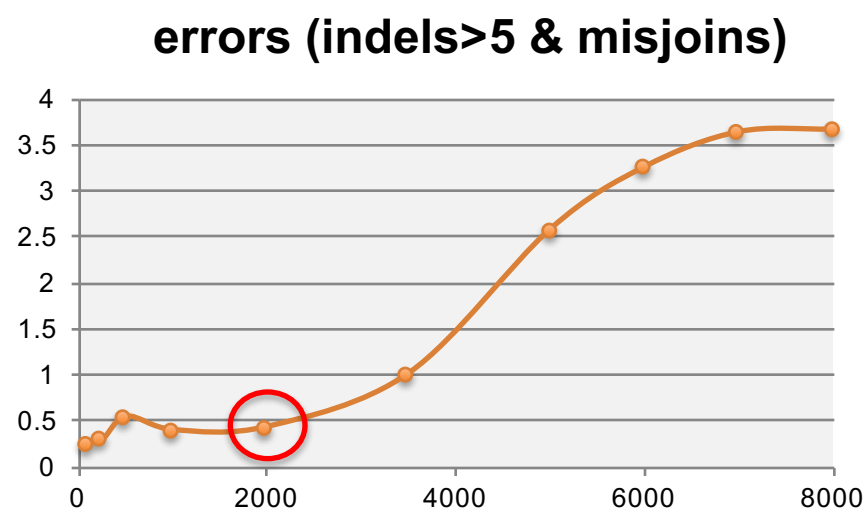


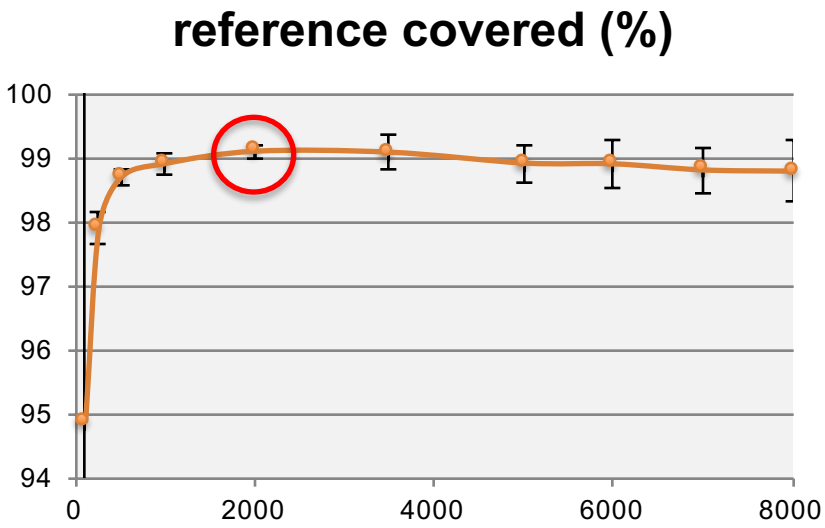
S. Lonardi, H. Mirebrahim, *et al.*, "When less is more: 'slicing' sequencing data improves read decoding accuracy and *de novo* assembly quality", *Bioinformatics*, 2015.

Reality: more data ↛ 'better' assemblies









S. Lonardi, H. Mirebrahim, *et al.*, "When less is more: 'slicing' sequencing data improves read decoding accuracy and *de novo* assembly quality", *Bioinformatics*, 2015.

More data are not necessarily better: why?

- Possible "suspects"
  - Sequencing errors
  - Highly uneven coverage
  - Read duplication / PCR amplification bias
  - Chimeric reads
  - "Imperfections" in the assembly algorithms

S. Lonardi, H. Mirebrahim, *et al.*, "When less is more: 'slicing' sequencing data improves read decoding accuracy and *de novo* assembly quality", *Bioinformatics*, 2015.

Possible solutions

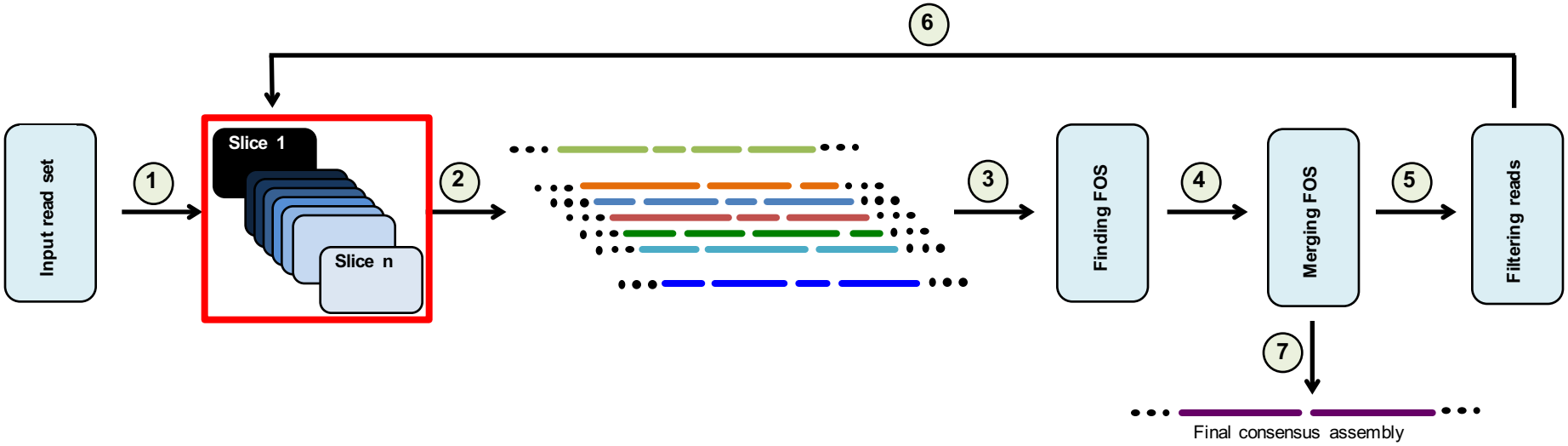
- "Classic" error correction
  - based on rare k-mers
  - ineffective for ultra-deep sequencing data
- Down-sampling
  - disregard a fraction of the input reads, according to some predetermined strategy
  - it may remove "critical" reads (i.e., rare error-free reads that can help bridge or fill assembly gaps)
  - not very effective
- SLICEMBLER (next)

S. Lonardi, H. Mirebrahim, *et al.*, "When less is more: 'slicing' sequencing data improves read decoding accuracy and *de novo* assembly quality", *Bioinformatics*, 2015.

SLICEMBLER algorithm

1: "Slice" the input

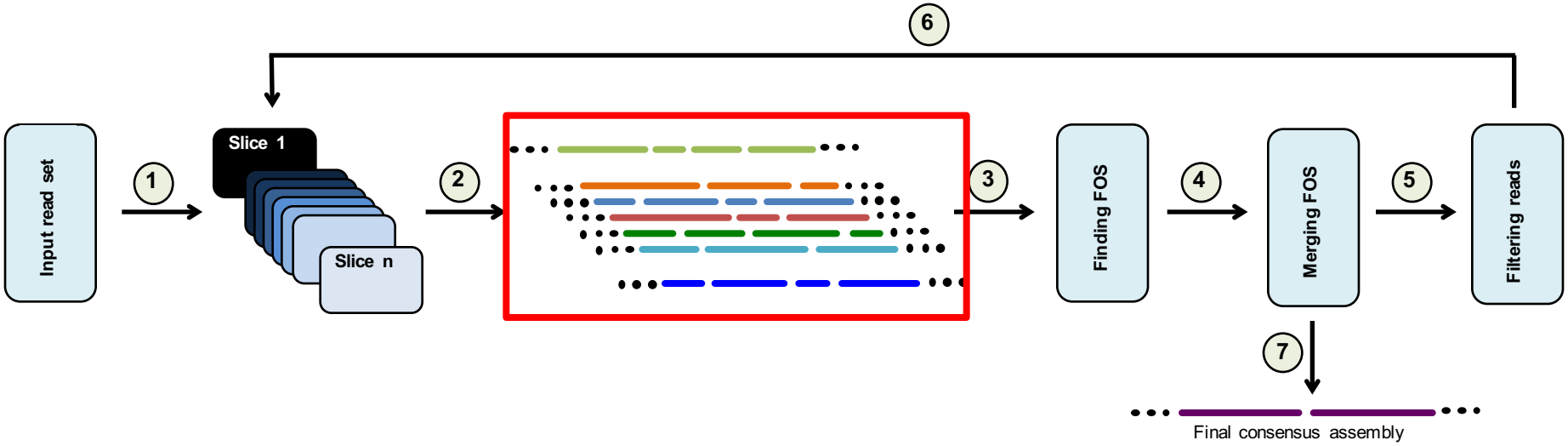
- The set of input reads is partitioned into *n* distinct slices, where *n* = the depth of coverage for the whole input read set / the desired depth of coverage for each slice
- Each slice contains approximately the same number of reads



SLICEMBLER algorithm

2: Assemble the reads in each slice

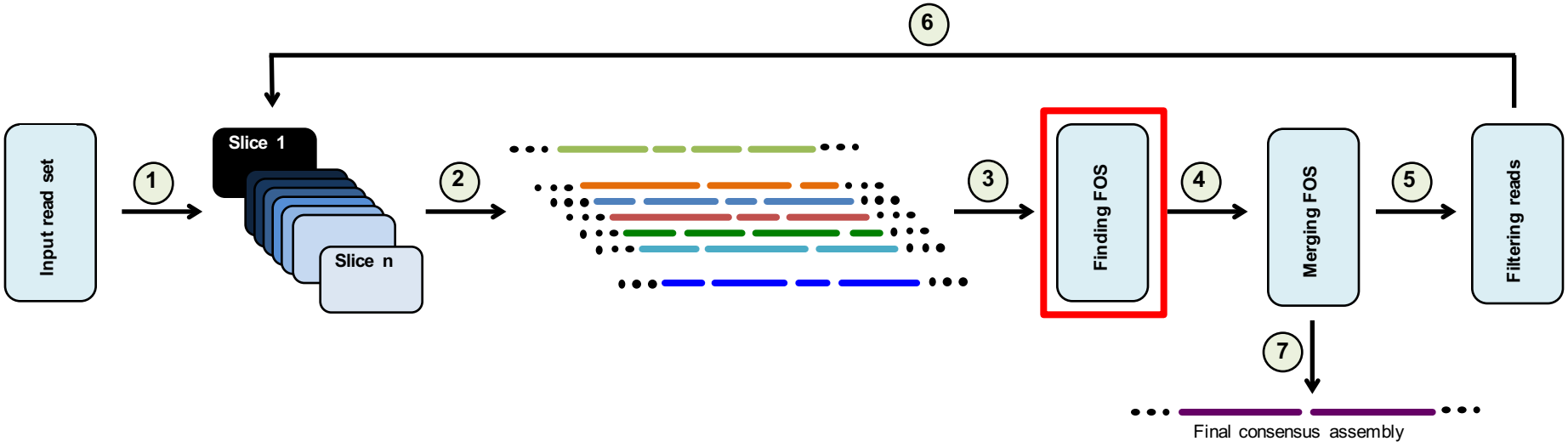
- Each of the *n* slices is assembled independently using a standard assembler (Velvet, SPAdes, IDBA, etc.)
- Each assembly is expected to contain a mix of high-quality and low-quality contigs



SLICEMBLER algorithm

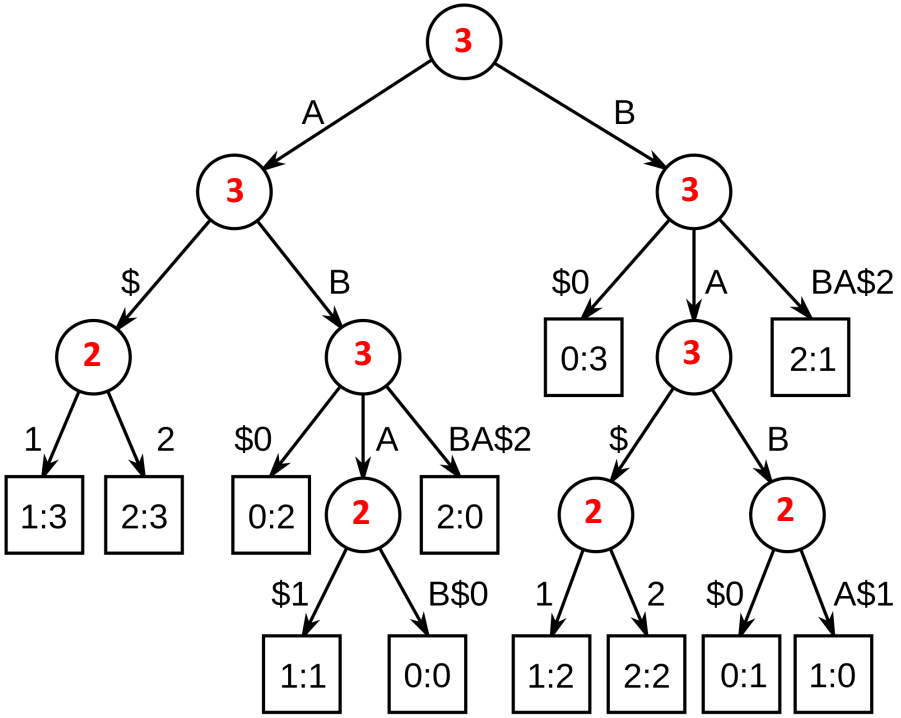
3: Find frequently occurring substrings (FOS)

- Identify high-quality contigs (or fraction thereof) or FOS
- Use a generalized suffix-tree for efficiency
- Remove tandem repeats at the end of FOS



Finding FOS efficiently

Definition: Given integers *k* and *l*, a *FOS* is a maximal substring *r* such that $|r| \geq l$ and it appears in at least *k* assemblies

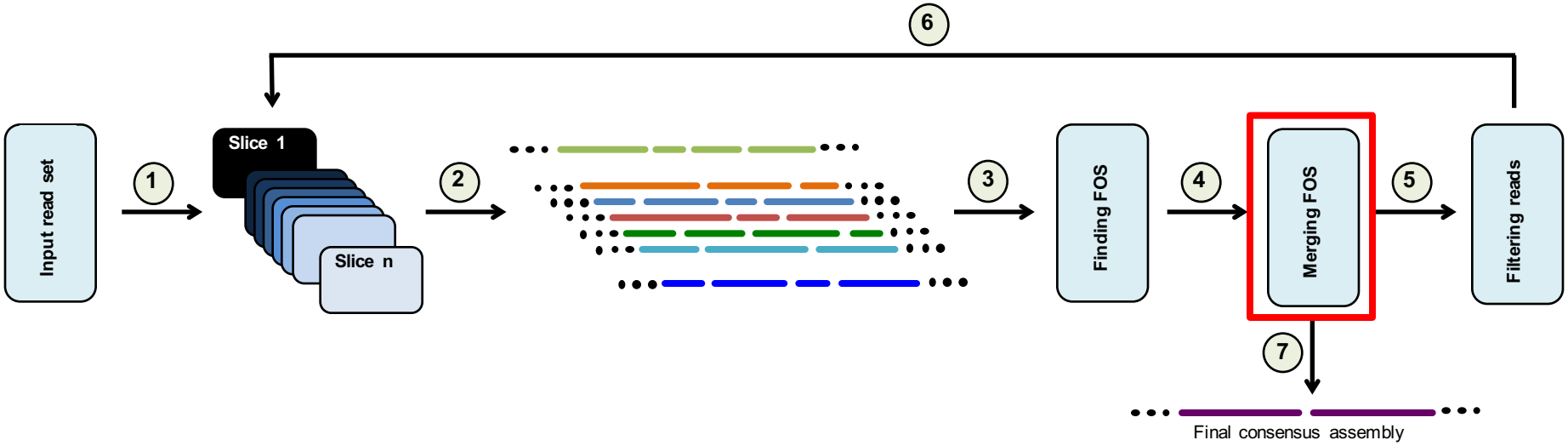


- Build a *generalized suffix tree* on the contigs of the *n* assemblies (and their reverse complement)
- Each input assembly is assigned a distinct "color" (Hui, CPM'92)
- Annotate each internal node *u* with the number of distinct colors in the subtree rooted at *u*
- In order to find FOS, determine all the deepest internal nodes (deeper than *l*) which have a color count of least *k*
- Building and annotating the suffix tree can be done in linear time

SLICEMBLER algorithm

4: Merge FOS

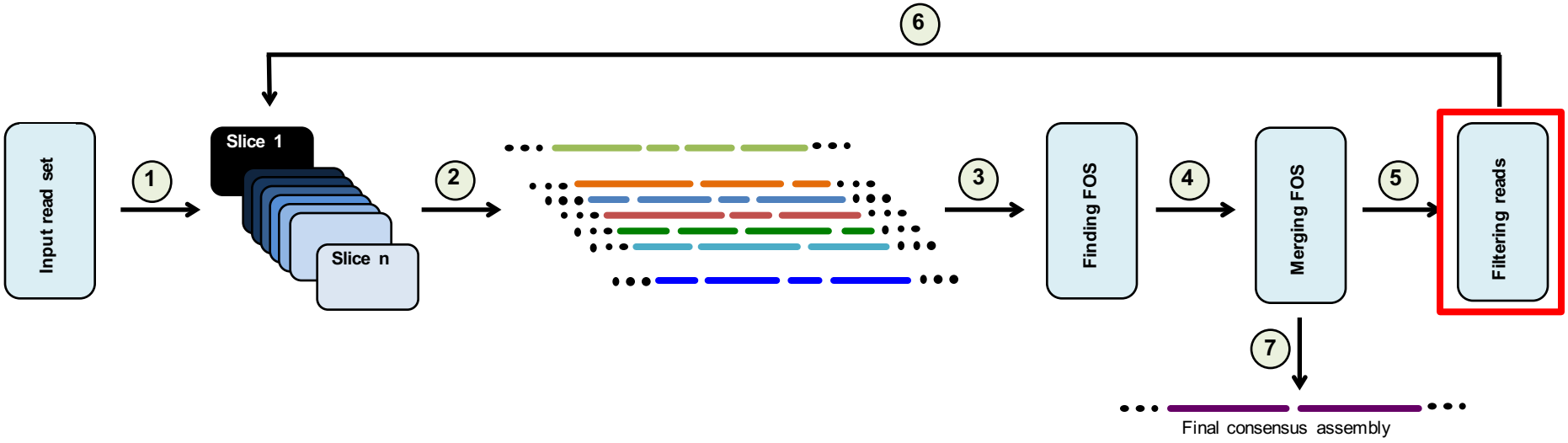
- When detected FOS are overlapping they can be merged to obtain longer FOS
- Merge based on exact suffix-prefix overlap and bridge reads (similar to scaffolding)



SLICEMBLER algorithm

5: Filter reads

- Input reads are mapped to FOS (e.g., BWA)
- Any read that maps to a contig in the current assembly is removed from input (unless it maps close to the end)
- Only the remaining reads are re-assembled in the next iteration

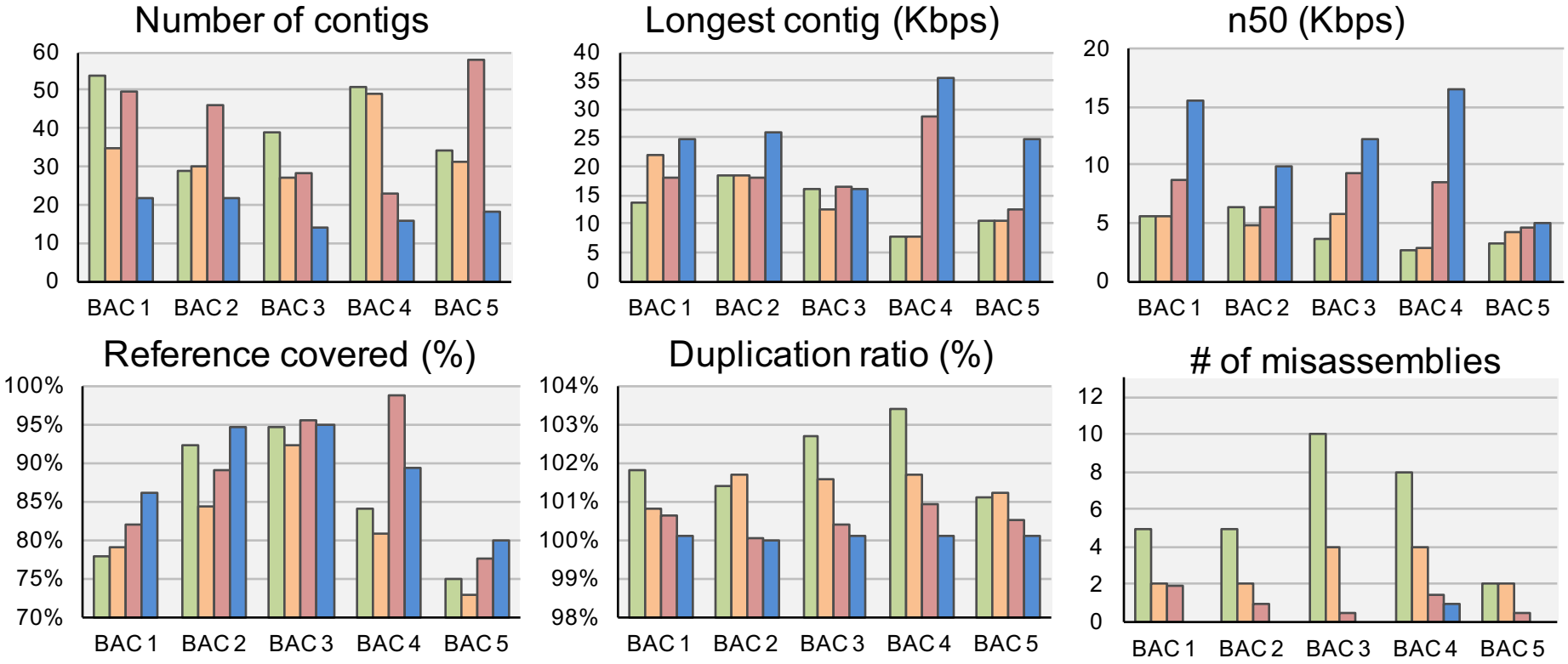


Experimental results

- Real ultra-deep sequencing data
  - Sequenced 16 barley BACs with Illumina HiSeq
  - Depth of coverage: 8,000x-15,000x
  - Paired-end reads (avg length ~88bp after trimming)
  - Selected 8,000x paired-end reads
  - High-quality references are available for five BACs

- Synthetic ultra-deep reads (*wgsim*)
  - Generated from the reference barley BACs
  - Paired-end reads (2x100 bp)
  - Various levels of coverage and error rates

Experimental results
(real barley BACs)



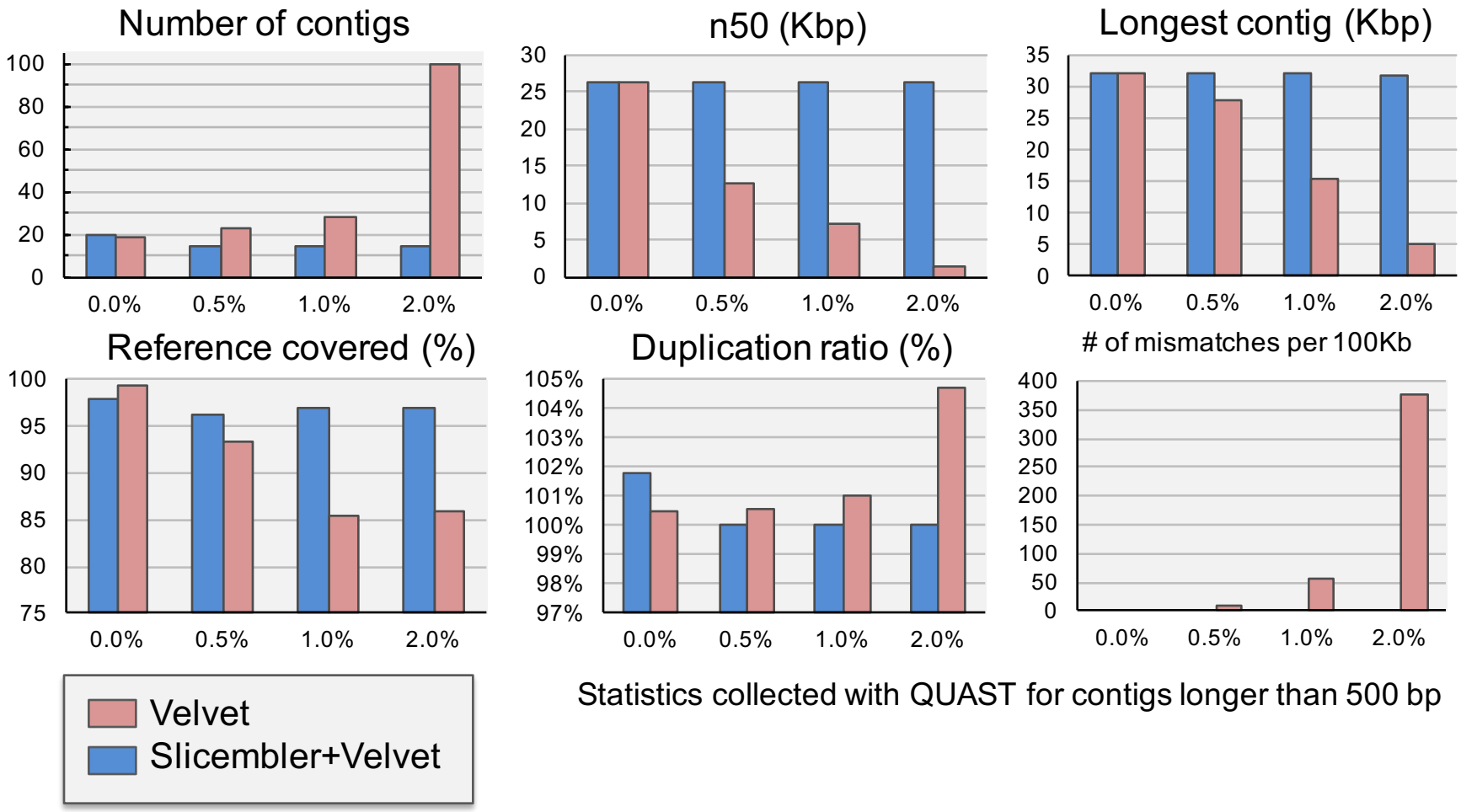
Statistics collected with QUAST for contigs longer than 500 bp

Assembly quality vs. base assembler (real barley BAC)

Method	Number of contigs	% ref covered	Duplication ratio	Mismatches per 100Kbp	N50	Longest contig
IDBA (8,000x)	34	97.0%	**1.010**	**0.93**	7,335	13,889
SLICEMBLER + IDBA (10 slices of 800x)	**13**	**97.0%**	**1.010**	1.1	**16,121**	**31,161**
Velvet (8,000x)	39	94.7%	1.027	20.0	3,649	16,048
SLICEMBLER + Velvet (10 slices of 800x)	**14**	**95.1%**	**1.001**	**0**	**12,178**	**16,128**
SPAdes (8,000x)	49	95.7%	**1.006**	**0.94**	9,129	21,872
SLICEMBLER + SPAdes (10 slices of 800x)	**11**	**96.9%**	1.024	1.2	**27,685**	**31,158**
Ray (8,000x)	35	80.0%	1.003	**0**	3,996	7,186
SLICEMBLER + Ray (10 slices of 800x)	**24**	**88.0%**	**1.000**	**0**	**7,192**	**12,842**

Statistics collected with QUAST for contigs longer than 500 bp

Assembly quality vs. sequencing error rate (simulated barley BACs)



Statistics collected with QUAST for contigs longer than 500 bp

Conclusions

- Modern *de novo* genome assemblers seem unable to take advantage of ultra-deep coverage
- SLICEMBLER is an iterative meta-assembler that takes advantage of the whole dataset and due to its "majority voting" scheme
  - Is more resilient to sequencing errors than its base assemblers
  - Almost never incorporates misassemblies in the consensus assembly
- SLICEMBLER is available at www.slicembler.cs.ucr.edu
- SLICEMBLER is slow, but a C++ implementation called SLICEMBLER++, will be available soon

Thank you

DBI-1062301 and IIS-1302134

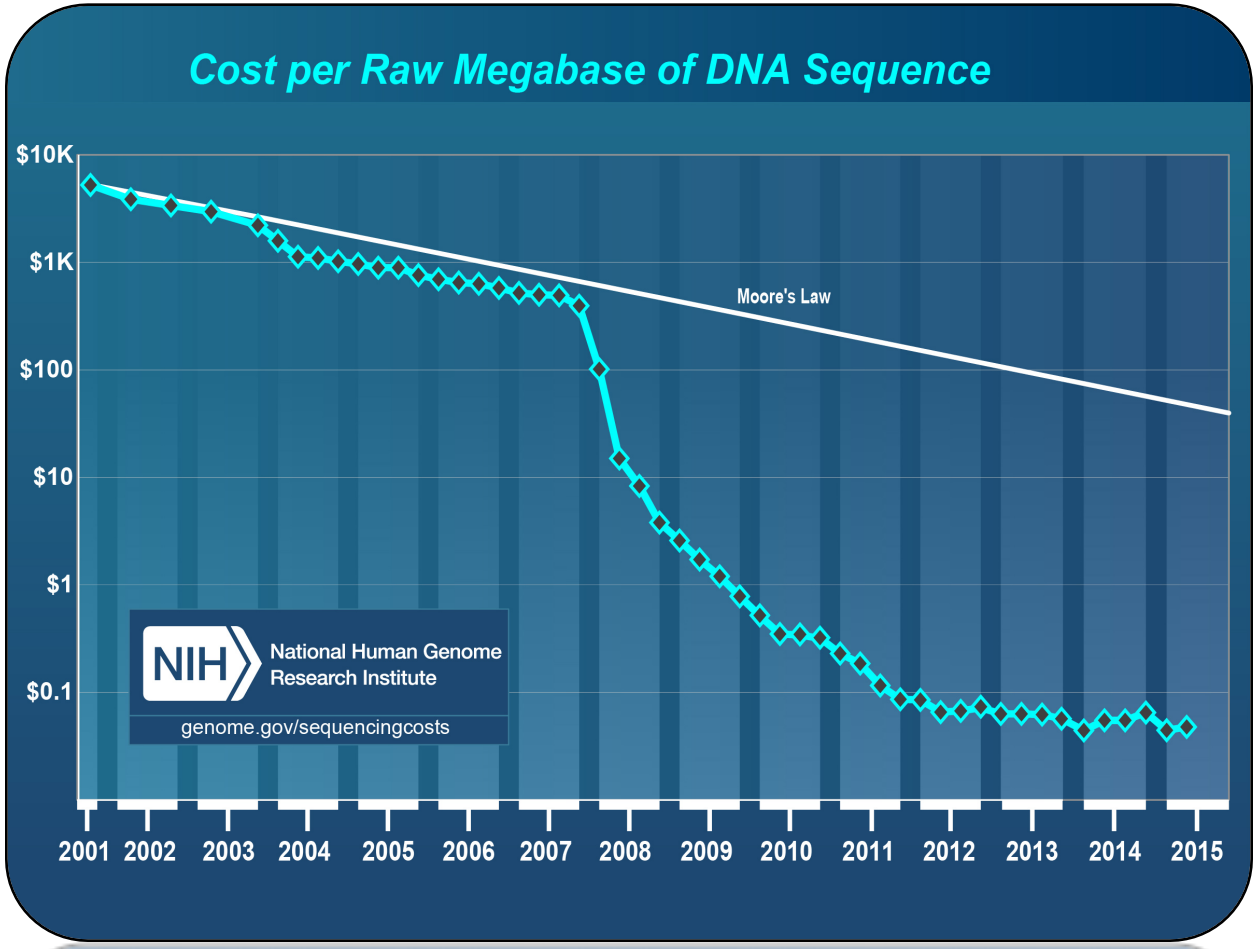
NIFA 2009-65300-05645

AID-OAA-A-13-00070

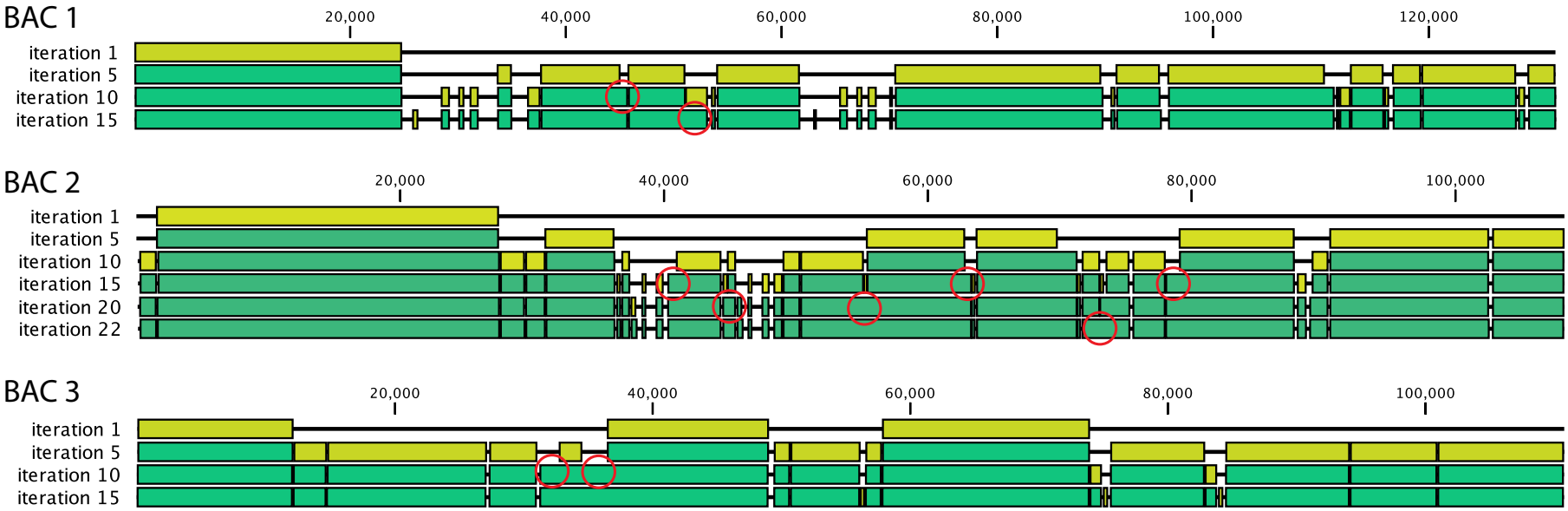
www.slicembler.cs.ucr.edu

Will sequencing cost continue to decrease?

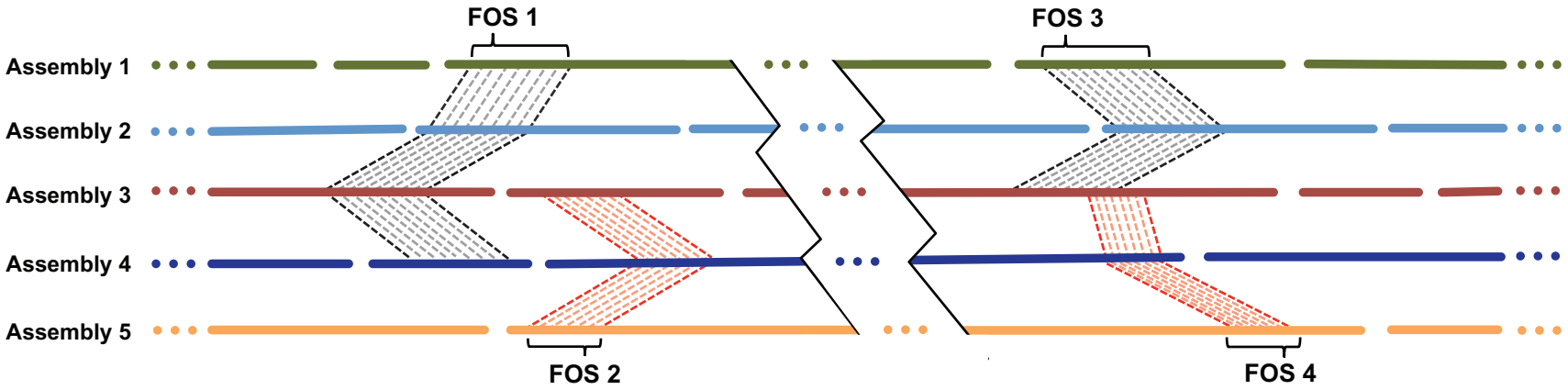
"It's difficult to make predictions, especially about the future"



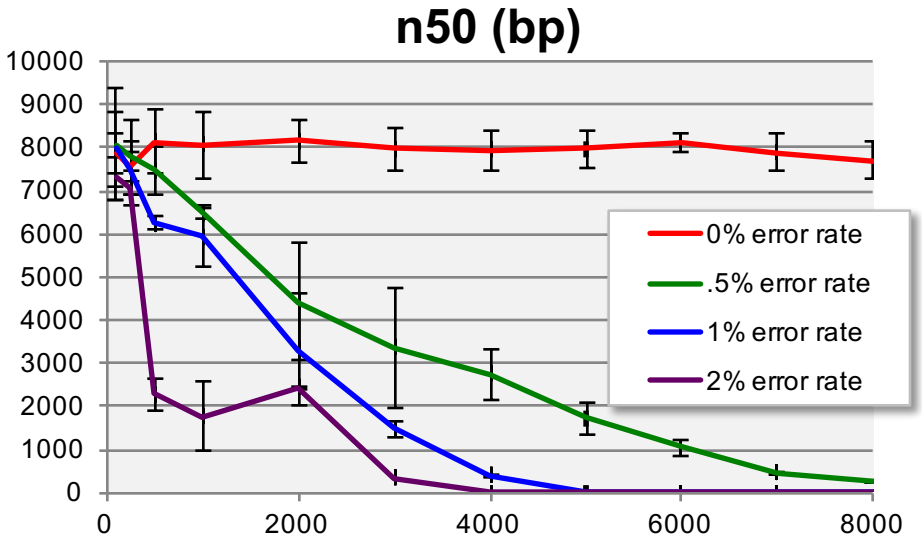
Experimental results
(real barley BACs)

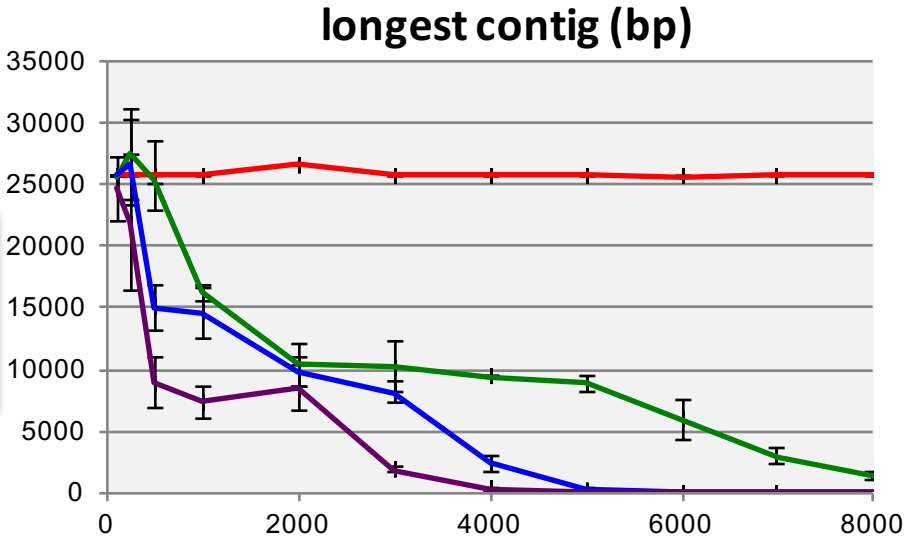


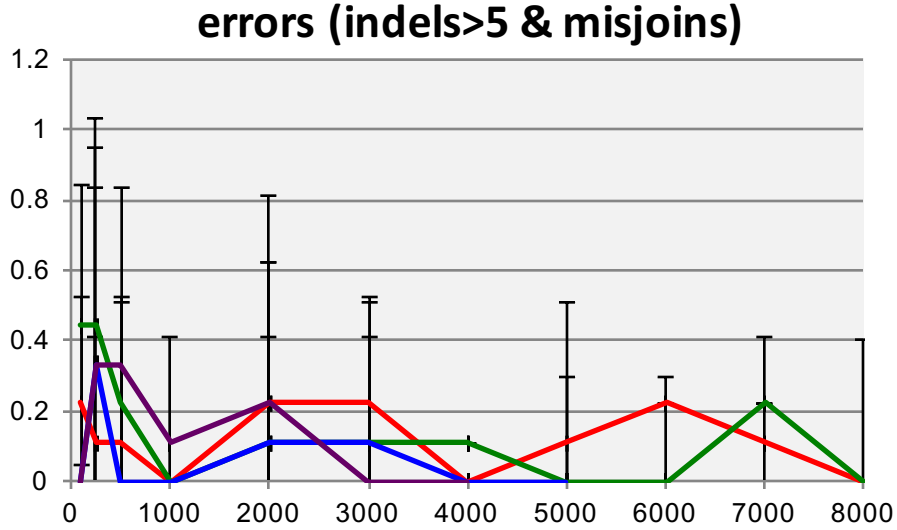
Frequently occurring substrings (FOS)

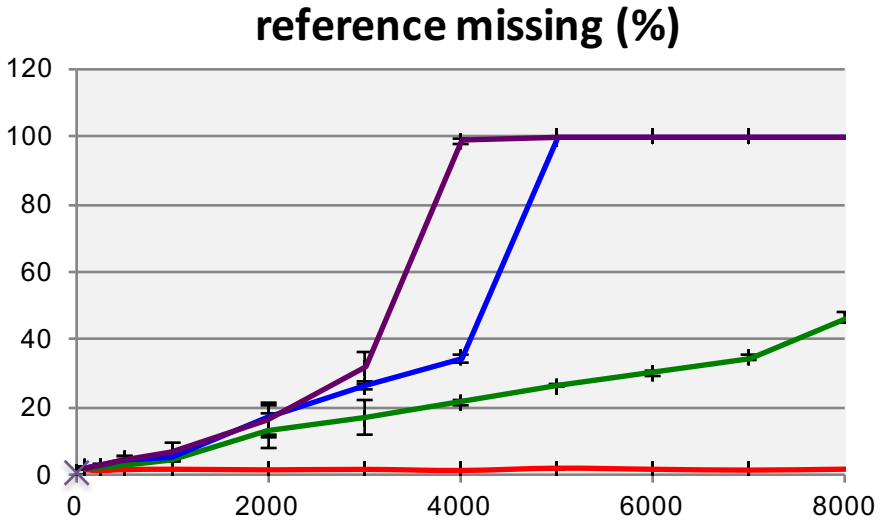


Varying sequencing error rate (Velvet)



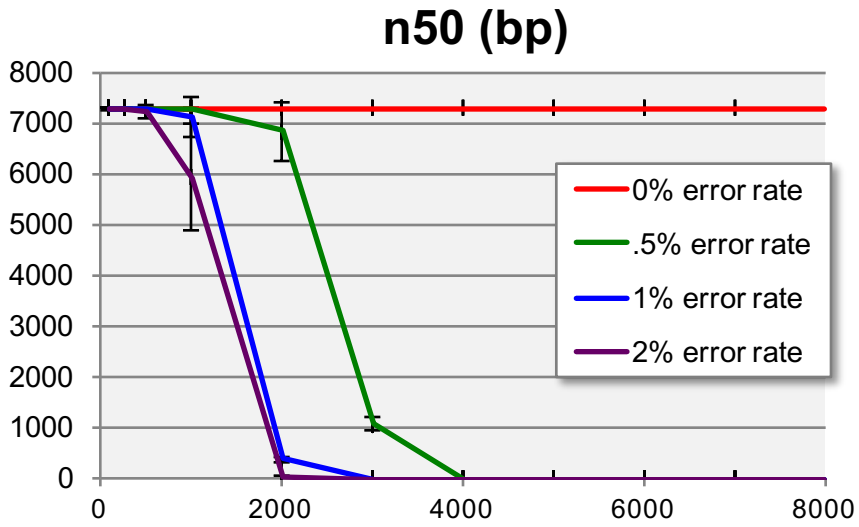


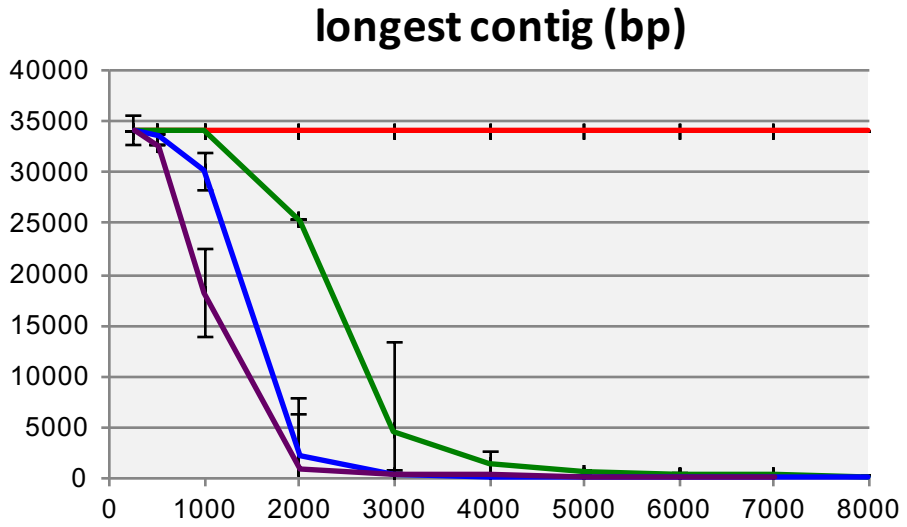


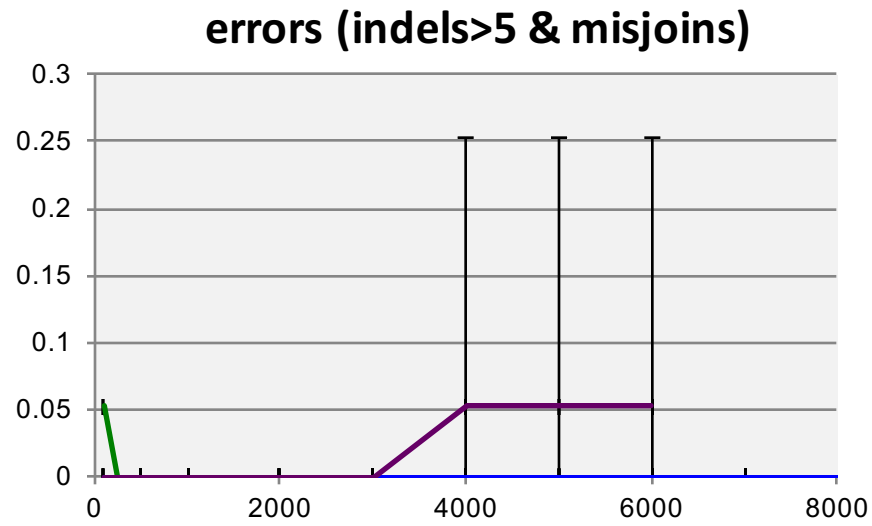


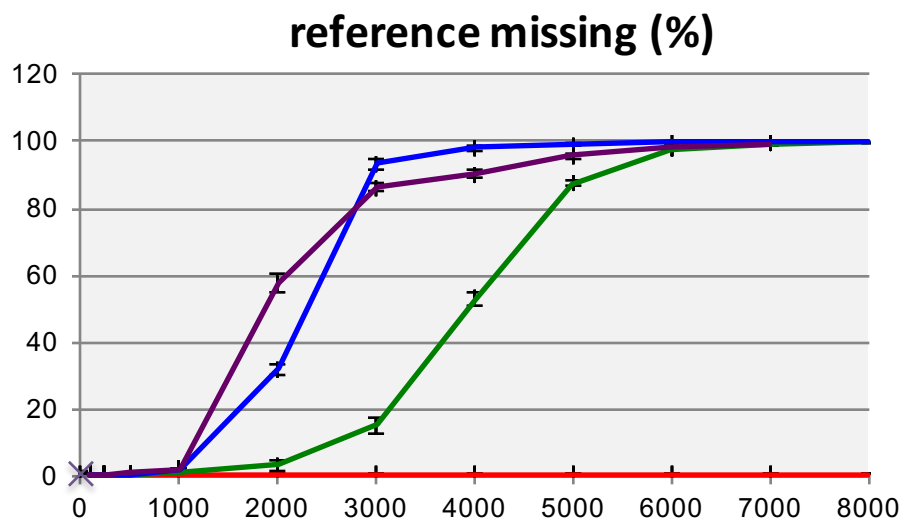
S. Lonardi, H. Mirebrahim, *et al.*, "When less is more: 'slicing' sequencing data improves read decoding accuracy and *de novo* assembly quality", *Bioinformatics*, 2015.

Varying sequencing error rate (IDBA)







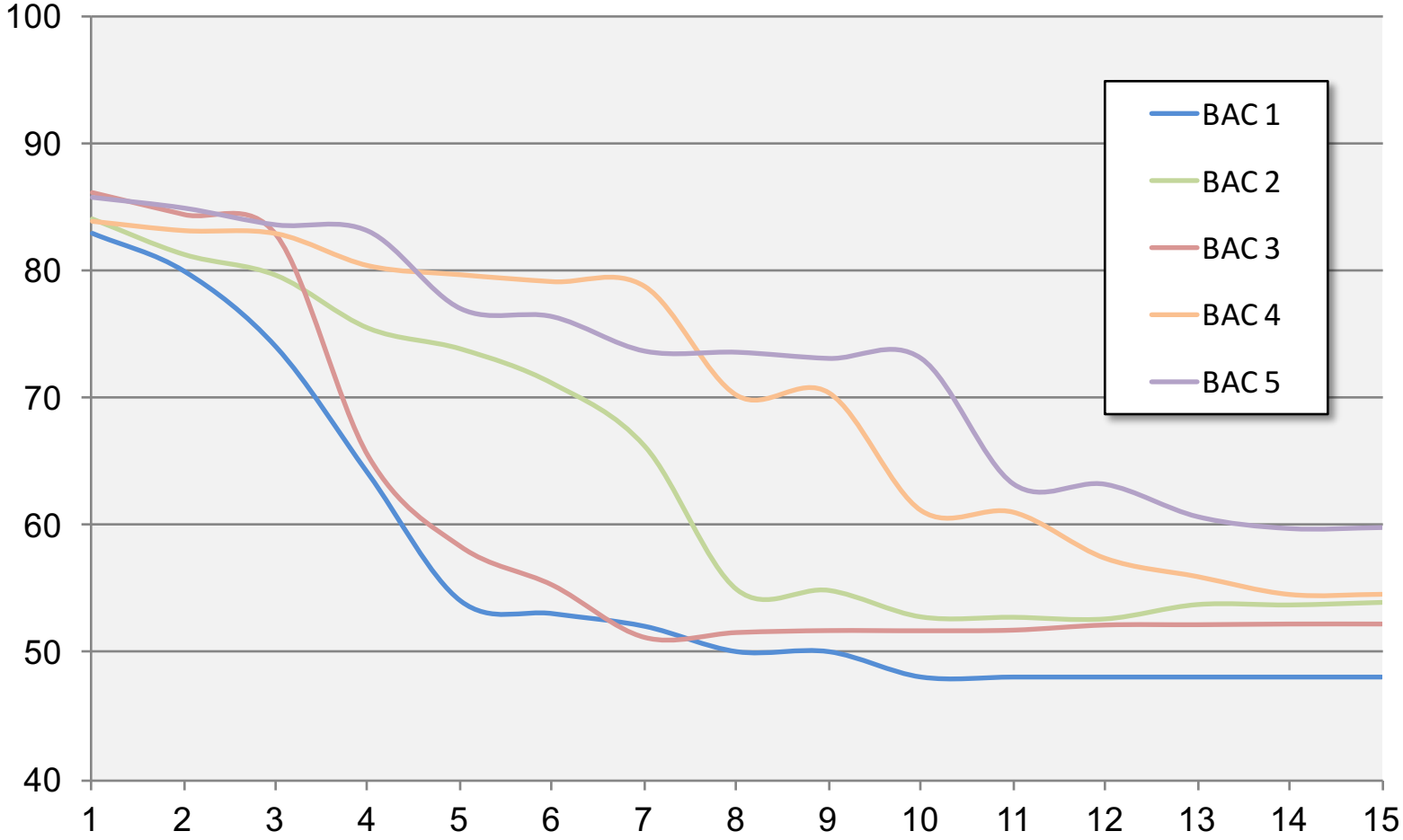


S. Lonardi, H. Mirebrahim, *et al.*, "When less is more: 'slicing' sequencing data improves read decoding accuracy and *de novo* assembly quality", *Bioinformatics*, 2015.

Sketch of the algorithm

- **partition** the input read set to *n* slices, each of which has D_s coverage
- **while** the cumulative length of FOS is less than the length of genome
  - **assemble** the reads in all *n* slices individually
  - **create** a suffix-tree from the *n* assemblies and their reverse compl
  - **assign** $k = n$, $l = l_{target}/5$
  - **while** ($l > l_{min}$)
    - **find** FOS longer than *l*, appearing in at least *k* assemblies
    - **if** FOS found **then merge** them with the previous FOS, **break**
    - **else if** k > n/2 **then assign** k = k-1
      - **else assign** l = l/2 , k = n
  - **if** (l <= l_{min}) **and** (no FOS were found) **then break**
  - **map** reads to FOS and **eliminate** mapped reads from the input
- **report** FOS

Percentage of reads that map exactly to the reference, iteration by iteration



Assembly quality vs. slice coverage

	500x	1,000x	2,500x	5,000x	7,500x	10,000x
Number of contigs	20	12	11	**10**	18	38
Longest contig	27,364	31,823	31,946	**31,950**	21,865	9,425
N50	6,707	26,275	**26,288**	26,267	12,428	3,643
Percent Refer. Covered	90.6%	88.7%	**94%**	93.9%	92.9%	84.7%
Duplication ratio	1	1	1	1	1	1
Mismatches per 100kbp	0	0	0	0	0	0

Another application for SLICEMBLER:
co-assembly of single cell sequencing data