High Throughput Sequencing
Algorithms and Applications

A special track of the ISMB 2018 meeting
Chicago, Illinois, July 8-9, 2018

ISMB 2018 HiTSeq Track Proceedings

Chicago, IL, United States
July 8-9, 2018
http://www.hitseq.org

Organizers:

Can Alkan
Bilkent University, Bilkent, Ankara, Turkey
E-mail: calkan@gmail.com

Ana Conesa
University of Florida, Gainesville, Florida, USA
E-mail: vickycoce@gmail.com

Francisco M. De La Vega, D.Sc.
Stanford University, and TOMA Biosciences, USA.
E-mail: Francisco.DeLaVega@stanford.edu

Dirk Evers
Molecular Health GmbH, Heidelberg, Germany
E-mail: dirk.evers@gmail.com

Kjong Lehmann
ETH-Zürich, Zürich, Switzerland
E-mail: kjong.lehmann@inf.ethz.ch

Quaid Morris
University of Toronto, Toronto, ON, Canada E-mail: quaid.morris@utoronto.ca

Gunnar Rätsch
ETH-Zürich, Zürich, Switzerland
E-mail: raetsch@inf.ethz.ch

Hercules: a profile HMM-based hybrid error correction algorithm for long reads

Can Firtina, Ziv Bar-Joseph, Can Alkan*, A. Ercument Cicek*

Choosing whether to use second or third generation sequencing platforms can lead to trade-offs between accuracy and read length. Several studies require long and accurate reads including de novo assembly, fusion and structural variation detection. In such cases researchers often combine both technologies and the more erroneous long reads are corrected using the short reads. Current approaches rely on various graph based alignment techniques and do not take the error profile of the underlying technology into account. Memory- and time efficient machine learning algorithms that address these shortcomings have the potential to achieve better and more accurate integration of these two technologies.

We designed and developed Hercules, the first machine learning-based long read error correction algorithm. The algorithm models every long read as a profile Hidden Markov Model with respect to the underlying platform's error profile. The algorithm learns a posterior transition/emission probability distribution for each long read and uses this to correct errors in these reads.

The pipeline is shown in Figure 1 with a toy example. Initially (1), short reads are aligned to long reads using an external tool. Here, red bars on the reads correspond to erroneous locations. Then, for each long read Hercules creates a profile HMM with priors set according to the underlying technology as shown in (2). Using the starting positions of the aligned short reads, Forward-Backward algorithm learns posterior transition and emission probabilities. Finally, Viterbi algorithm finds the most likely path of transitions and emissions as highlighted with red colors in (3). The prefix and the suffix of the input long read in this example is "AGAACC...GCCT". After correction, substring "AT" inserted right after the first "A". Third "A" is changed to "T" and following two basepairs are deleted. Note that deletion transitions are omitted other than this arrow, and only two insertion states are shown for clarity of the figure. On the suffix, a "T" is inserted and second to last basespair is changed from "C" to "A".

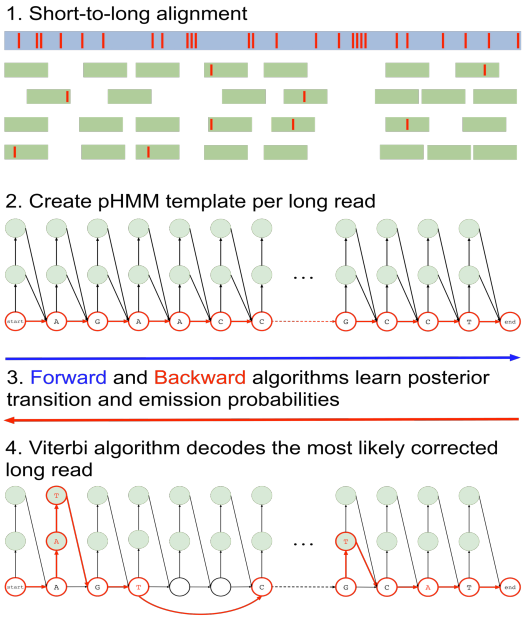


Figure 1. Overview of the Hercules algorithm.

Using datasets from two DNA-seq BAC clones (CH17-157L1 and CH17-227A2), and human brain cerebellum polyA RNA-seq, we show that Hercules-corrected reads have the highest mapping rate among all competing algorithms (Table 1) and highest accuracy when most of the basepairs of a long read are covered with short reads (Table 2). Full paper is available at https://www.biorxiv.org/content/early/2017/12/13/233080 and source code is available at https://github.com/BilkentCompGen/Hercules.

Table 1. Summary of error correction

Tool	Number of aligned reads											
	CH17-157L1 BAC clone				CH17-227A2 BAC clone				Human brain transcriptome			
	Mapped	80-90%	90-95%	>95%	Mapped	80-90%	90-95%	>95%	Mapped	80-90%	90-95%	>95%
Uncorrected	33,842	17,582	1,974	461	45,625	25,356	7,385	219	150,000	107,493	17,884	921
Colormap	36,391	13,586	7,904	6,235	46,209	18,398	12,364	4,715	150,663	49,803	35,412	50,518
HALC	35,220	17,867	2,974	2,249	46,680	23,969	9,662	2,356	150,970	93,500	29,822	6,682
LoRDEC	36,812	10,320	7,397	12,167	47,304	8,875	7,010	**25,844**	149,923	40,914	38,927	56,818
LSC	43,431	13,534	7,511	12,277	55,853	13,619	10,205	23,509	150,771	33,910	36,706	**66,726**
Hercules	**44,229**	13,609	7,569	**12,583**	**56,140**	13,433	9,809	24,269	**151,903**	34,360	38,352	65,880
proovread	38,962	14,918	6,714	7,956	47,344	17,460	9,233	11,140	150,235	82,344	21,891	25,944

Table 2. Correction accuracy given high breadth of coverage (>90%) in CH17-227A2.

Tool	No. of Aligned Reads			
	Mapped	80-90%	90-95%	>95%
Uncorrected	4,429	2,135	2,093	26
Colormap	4,432	668	2,345	1,271
HALC	4,433	1,789	2,436	58
LoRDEC	4,425	124	224	4,006
LSC	4,473	45	149	4,264
Hercules	**4,476**	43	142	**4,273**
proovread	4,434	1,722	1,665	880

Realignment of short reads around short tandem repeats significantly improves accuracy of genomic variants detection

Accurate detection and genotyping of genomic variants from short reads produced by high throughput sequencing technologies is a fundamental feature of any successful data analysis production pipeline for applications such as genetic diagnosis in medicine or genomic selection in plant and animal breeding. Our research group maintains a well established open-source software solution that tightly integrates algorithms for discovery of different types of genomic variants, which can be efficiently used from the command line, a rich graphical interface or a web environment. Understanding that incorrect alignments around small indels and especially short tandem repeats (STRs) are a main sources of false positives in variants discovery, we present our solution for realignment of short reads spanning these variants. Users can explicitly provide STRs predicted from other specialized tools such as tandem repeats finder (TRF) to realign consistently reads spanning these STRs and to genotype them as a single variation locus. Variable mononucleotide runs are also predicted directly from aligned reads. We performed extensive benchmark experiments comparing our solution to state-of-the-art software using both simulated datasets and real data from four species with different distributions of repetitive elements and varying conditions of ploidy, read length, average read depth and read alignment software. Figure 1 shows the accuracy of different tools in one of the evaluated scenarios. Our solution consistently shows equal or better accuracy and efficiency than the other solutions under different conditions. We expect that this work will contribute to the continuous improvement of quality in variant calling needed for applications such as personalized medicine.

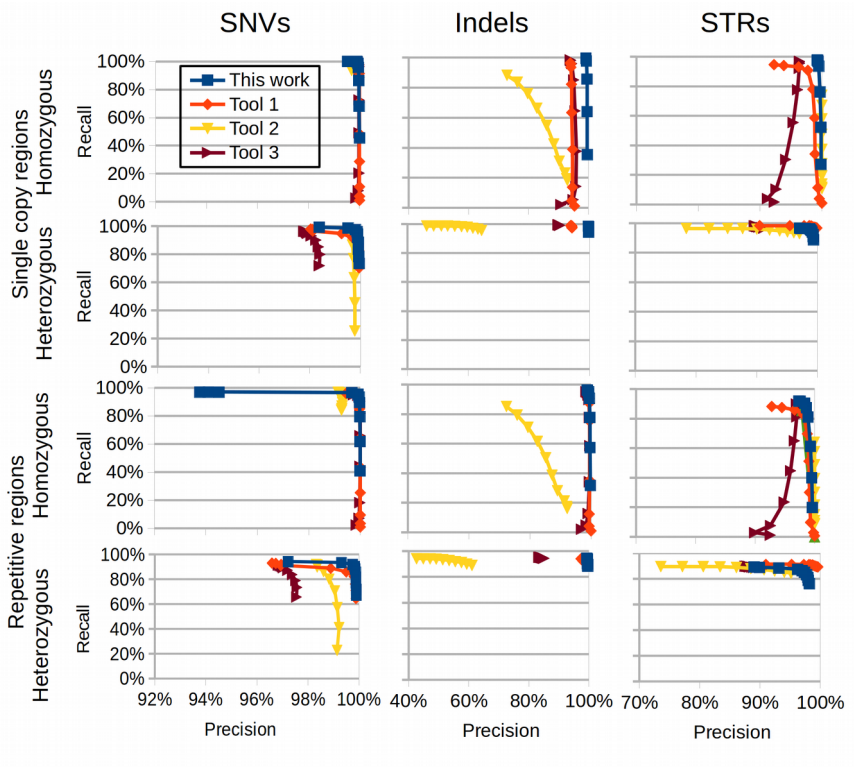


Figure 1. Precision and recall of different tools for detection of heterozygous and homozygous SNVs, indels and STRs. A diploid individual was simulated from a real 400Mbp genome with 50% repetitive content. Reads were simulated with length=200, error rate 0.01 and average RD=20x.

Convolutional filtering for mutation signature discovery

There are a number of computational methods for mutation signatures detection from genome sequencing of cancer. Most of these methods take both the base being mutated and also the bases immediate to the left and right of it into consideration (a *trinucleotide* sequence). In total, there are $4^3 \times (4-1) = 192$ different types of mutations (from $AAA \rightarrow ACA$ to $TTT \rightarrow TGT$). But in many cases, because of the reverse-complementary property of DNA, $AAC \rightarrow ACC$ on one strand is equivalent to $GTT \rightarrow GGT$ on the other, and therefore there are 96 different types of mutations in total, considered by most methods.

The standard approach for identifying mutation signatures has been popularized by [**?**] who used non-negative matrix factorization (NNMF) to decompose a $96 \times S$ count matrix (of all the trinucleotide mutations identified from the sequence data of S samples) into two low-rank $96 \times K$ and $K \times S$ matrices, where K is the number of mutation signatures. The first factor matrix contains the so-called mutation signatures, i.e. the patten of mutations, whilst the second contains the mutational rate/activity of each signature for each sample (Figure 1A). One limitation is that the trinucleotide context is chosen arbitrary and mutational processes might be related to more extended sequence contexts [**?**]. However, at present only one computational method, `pmsignature`, has addressed this problem. Furthermore, all methods do not implement model selection and estimate the number of signatures K. This parameter is usually pre-defined.

Here, we propose a novel approach for mutation signature identification based on convolutional filtering. The premise is that each mutational process can be described as a mutational filter which scans the genome. Each filter has coefficients which gives it more affinity to certain sequence types. When a filter encounters a sequence that it has high affinity for, there is an increase probability that a mutation will occur here. The inference task is to examine the set of mutations detected from genome sequencing of a cancer (and its sequence context) and to infer the set of mutational filters (the coefficients) that is likely to have given rise to that observed data (Figure 1B-D). This approach is therefore fundamentally very different to NNMF of mutation count matrices and has a qualitative link to the potential biochemical processes that might produce such mutations.

We phrase this problem within a Bayesian statistical inference framework called *convSig* and demonstrate performance on 3-bp and 5-bp windows that rivals pre-existing techniques. Furthermore, by embedding multiple layers of convolutional filters, we are able to extend to larger sequence contexts and learn novel mutation signatures associated with mutational clusters. Our framework allows for the rigorous assessment of model complexity (number of signatures) and convolutional filters are easily transferable allowing them to easily applied to novel genomes to identify previously discovered signatures. We believe this to be the first time that convolutional neural networks have been applied to mutation signature detection.

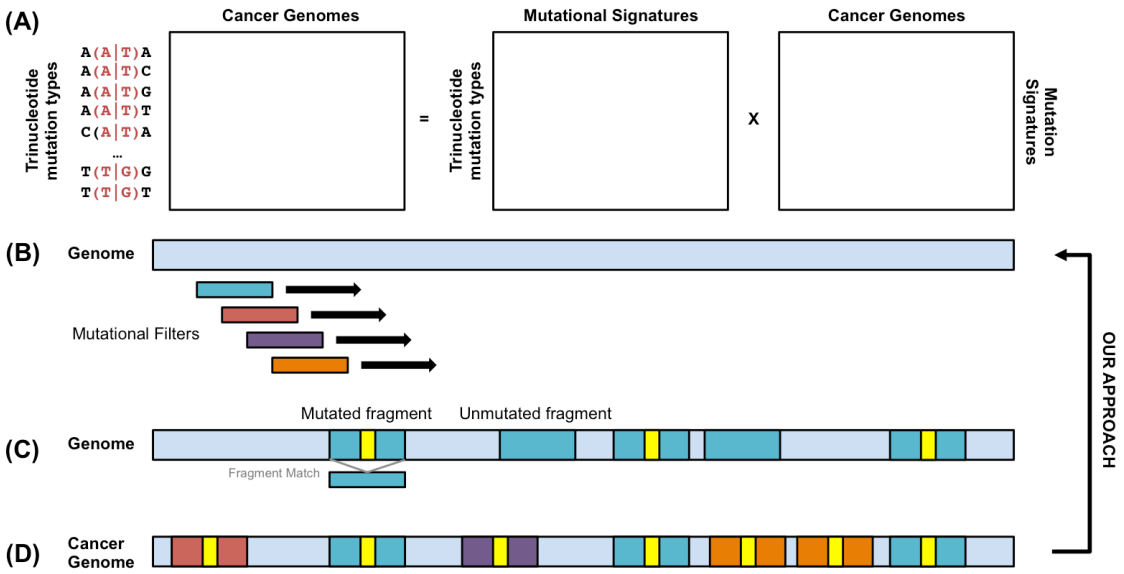


Figure 1: **Overview of convolutional filtering model for mutation signature identification.** (A) Schematic of standard matrix factorization approach for mutational signature discovery. (B) Mutation processes can be modelled as convolutional filters that screen the genome. (C) These convolution filters each has an affinity for certain sequence contexts based on their filtering coefficients. High affinity sub-sequences will lead to an increased probability that a mutation will occur. (D) A cancer genome can be considered as the output after a number of mutational filters have passed through the genome. The learning task is to take the set of observed mutations and their local sequence context and to learn the most plausible set of mutational filters that could have given rise to these observations.

Quantification of private information leakage and privacy-preserving file formats for functional genomics data

Functional genomics experiments on human subjects present a privacy conundrum. On one hand, many of the conclusions we infer from these experiments are not tied to the identity of individuals but represent universal statements about disease and developmental stages. On the other hand, by virtue of the experimental procedures, the reads from them are tagged with small bits of patients' variant information, which presents privacy challenges in terms of data sharing. There are many benefits to sharing the data as broadly as possible. Measuring the amount of variant information leaked in a variety of experiments, particularly in relation to the amount of sequencing, will allow us to uncover ways of reducing information leakage and determine an appropriate set point for sharing information with minimal leakage. To this end, we aimed to derive information-theoretic measures for the private information leaked in experiments and develop various file format manipulations to reduce much of the leaked variants. We showed that high-depth experiments such as Hi-C provide accurate genotyping that can lead to large privacy leaks. Counter intuitively, noisy and partial genotypes from low-depth experiments such as ChIP-Seq and single-cell RNA-Seq, although not useful genotypes, can be used as strong quasi-identifiers for re-identification purposes through linking attacks. We showed that these incomplete genotypes could further be used to construct an individual's complete variant set and infer individual identifying phenotypes when combined with imputation. We then provide a proof-of-concept theoretical framework, in which the amount of leaked information can be estimated from the depth and breadth of the coverage as well as the sequencing bias of the functional genomics experiments. In order to solve the dilemma between data sharing and privacy leakage, we propose a file formatting system that enables the sharing of a large amount of data while protecting individuals' sensitive information and preserving the utility of the data. The proposed file format can achieve different levels of privacy and utility balance. At the highest level of privacy, our file format masks all the variant information leaked from reads, which can be used to calculate signal profiles with 99% recovery of the original profiles and 100% recovery of the original gene expression levels.

Allelic decomposition and exact genotyping of highly polymorphic and structurally variant genes

Ibrahim Numanagić[1,2], Salem Malikić[1], Michael Ford[1], Xiang Qin[3], Lorraine Toji[4], Milan Radovich[5], Todd C. Skaar[5], Victoria M. Pratt[5], Bonnie Berger[6], Steve Scherer[3], and S. Cenk Sahinalp[7]

[1]School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada
[2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge 02139, MA, USA
[3]Baylor College of Medicine Human Genome Sequencing Center, Houston 77030, TX, USA
[4]Coriell Institute for Medical Research, Camden 08103, NJ, USA
[5]Indiana University School of Medicine, Indianapolis 46202, IN, USA
[6]Department of Mathematics, Massachusetts Institute of Technology, Cambridge 02139, MA, USA
[7]Department of Computer Science, Indiana University, Bloomington 47405, IN, USA

Publication Details

Aldy has been published in Nature Communications in February 2018 (doi:10.1038/s41467-018-03273-1). The presenter will be Ibrahim Numanagić or Salem Malikić. The abstract is intended for ISMB HitSeq 2018.

Abstract

High-throughput sequencing provides the means to determine the allelic decomposition for any gene of interest— the number of copies and the exact sequence content of each copy of a gene. However, this is a challenging task, as many clinically and functionally important genes are highly polymorphic and have undergone structural alterations. Despite the clinical and scientific need, no high-throughput sequencing data analysis tool has yet been designed to effectively solve the full allelic decomposition problem.

Here we introduce a combinatorial optimization framework (Figure 1) that successfully resolves the number of copies and the exact sequence content of each copy of a gene, including for genes that have undergone structural alterations. We provide an associated computational tool Aldy that performs allelic decomposition of highly polymorphic, multi-copy genes through the use of whole or targeted genome sequencing data. For a large and diverse data set obtained through the use of various sequencing platforms, Aldy identifies multiple rare and novel alleles for several important pharmacogenes, significantly improving upon the accuracy and utility of current genotyping assays. Aldy has minimal impact on computational resources, and is capable of analyzing a high-coverage BAM file in less than a minute on a typical laptop machine. As more data sets become available, we expect Aldy to become an essential component of genotyping toolkits.

Aldy is available for download at `http://aldy.csail.mit.edu`.

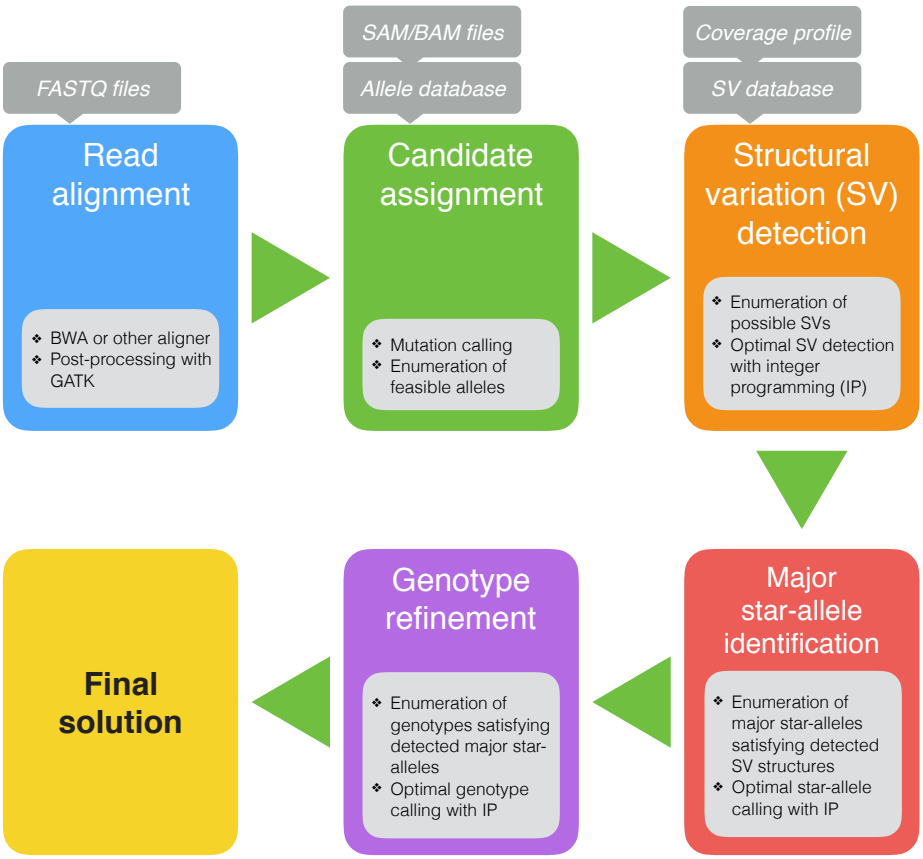


Figure 1: Graphical representation of steps performed by Aldy.

ISMB HiTSeq 2018 Committee
International Society for Computational Biology

Martin Steinegger
Am Faßberg 11
37077 Göttingen
martin.steinegger@mpibpc.mpg.de

Submission as Highlight-Talk 29. January 2018

Dear ISMB HiTSeq 2018 Committee,

Metagenomics is revolutionizing the study of microbes in their natural environments, such as the human gut, the oceans, or soil, and is revealing the enormous impact of microbes on our health, our climate, and ecology. In metagenomics, DNA or RNA of bacteria, archaea and viruses are sequenced directly, making the 99% of microbes that cannot be cultivated in the lab amenable to investigation. Combined with the enormous drop in sequencing costs by a factor of ten thousand in just ten years, this has led to an explosive growth in the amount of sequence data in public databases.

To predict functions for these new sequences, very fast sequence search tools have been developed in recent years, but the increase in speed was paid by lower search sensitivity. Yet many of the microbes investigated by metagenomics have no close relatives in the sequence databases, and current search tools are too insensitive to detect them. Consequently, for the large majority of metagenomic sequences no functions can be predicted.

To address the need for very fast yet sensitive sequence search, we developed the software MMseqs2 (Many-against-Many sequence searching). The most important distinction of MMseqs2 to previous fast search tools, is its ability to search with sequence profiles and not only with simple sequences. Since PSI-BLAST made its debut 20 years ago, sequence profiles have been known to improve search sensitivity enormously. But until now, no way had been found to drastically speed up sequence profile searches.

We developed a very fast, and sensitive sequence prefilter algorithm at the core of MMseqs2. It preselects the most promising database sequences for subsequent, slower, and more accurate comparison. Whereas all recent tools use *exact* matches between short words (*k*-mers), we extend a 27 year old idea from BLAST to detect *similar* instead of exact *k*-mer matches. This algorithm can generate lists of similar *k*-mers both for sequences and sequence profiles. To gain further sensitivity, we were able to increase the word length *k* from three to seven. We also developed a very efficient method to detect when two neighboring *k*-mer matches occur on the same diagonal, which excluded most chance matches.

MMseqs2 scales almost inversely with the number of used processor cores. It can automatically split and distribute query or target databases across several servers, allowing even users with relatively modest computing resources to cluster or search databases with billions of sequences. It also enables users to analyze jointly collections of datasets that could so far only be analyzed separately.

MMseqs2 improves on current search tools over the full range of speed-sensitivity trade-off, achieving sensitivities better than PSI-BLAST at more than 400 times its speed. Sensitive searches enabled us to annotate 1.1 billion sequences in 8.3 hours on 28 cores. MMseqs2 therefore offers great potential to increase the fraction of annotatable (meta)genomic sequences.

Best regards,
Martin Steinegger and Johannes Söding

Reference:

Steinegger, M., and Söding, J. (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnol*., 16 October 2017, doi: 10.1038/nbt.3988.

Probabilistic inference of clonal gene expression through integration of RNA & DNA-seq at single-cell resolution

Kieran R Campbell[1,2], Alexandre Bouchard-Côté[2], Sohrab P Shah[1,3]

1. *Department of Molecular Oncology, BC Cancer Agency*
2. *Department of Statistics, University of British Columbia*
3. *Department of Pathology and Laboratory Medicine, University of British Columbia*

Background Human cancers form clones - sets of cells that exhibit similar mutations and genomic rearrangements. As clones evolve to resist chemotherapy understanding their molecular properties is crucial to designing effective treatments. While it is possible to measure both the DNA (that defines clonal structure) and RNA (that defines cell state) in single-cells through assays such as G&T-seq (Macaulay, 2015), these assays are time consuming and hard-to-scale. In practice, it is far more common to have large datasets where DNA and RNA are measured in separate cells through scalable technologies such as DLP sequencing (Zahn, 2017) for DNA and 10x genomics single-cell RNA-seq (Zheng, 2017). Although the destructive nature of each measurement process means the same cell will never be observed twice, if such assays are applied to the same tumor samples we expect the same clones to be present in both data views. However, it remains an open problem to link data across the expression space and genomic space that would allow for clone-specific expression estimates.

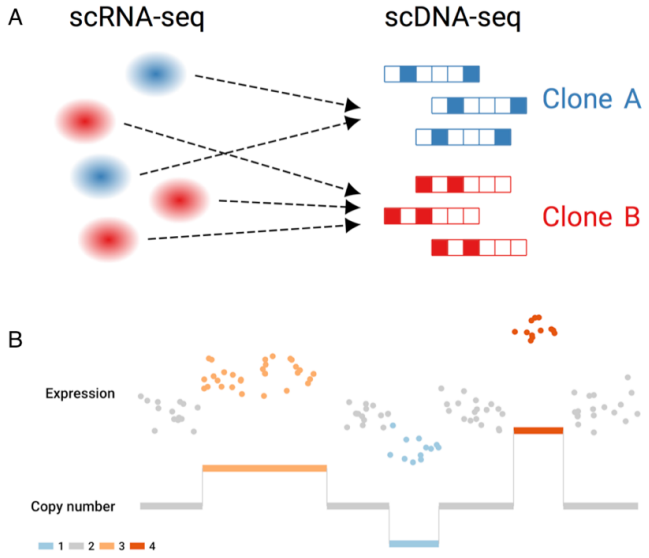


*Figure 1. (**A**) Single-cells measured in gene expression space to be assigned to cancer clones inferred from single-cell DNA-sequencing. (**B**) Assuming a noisy relationship between copy number and expression allows us to relate two distinct views.*

Results Here we present *clonealign*, a highly scalable statistical method to probabilistically assign each cell as measured in gene expression space (scRNA-seq) to a clone defined in copy number space (scDNA-seq) (figure 1A) by assuming a copy-number-dependent effect on expression (figure 1B). We derive an expectation-maximization (EM) algorithm that parallelizes across the genes present allowing thousands of cells to be assigned to clones in minutes on commodity hardware. Through simulations we

demonstrate that relatively few (<20%) genes must exhibit CNV-gene expression relationships for such assignment to be feasible and highly accurate.

We apply our method to independently generated whole genome scDNA-seq and 10x genomics scRNA-seq from a patient-derived breast cancer xenograft to characterize the gene expression of expanding clones over time (figure 2 A-B). We show the method infers expected clonal proportions and validate *clonealign's* clone assignments through held-out gene predictions (figure 2C) and loss-of-heterozygosity analyses. We also apply *clonealign* to a large dataset of 4000 cells from an ovarian cancer cell line, demonstrating gene expression assignment to clones defined by cutting the overall genomically inferred phylogenetic tree at different levels. Finally, we demonstrate how our framework serves as a basis for generalized multiview clustering from unpairable data sources for which we present a proof-of-concept.

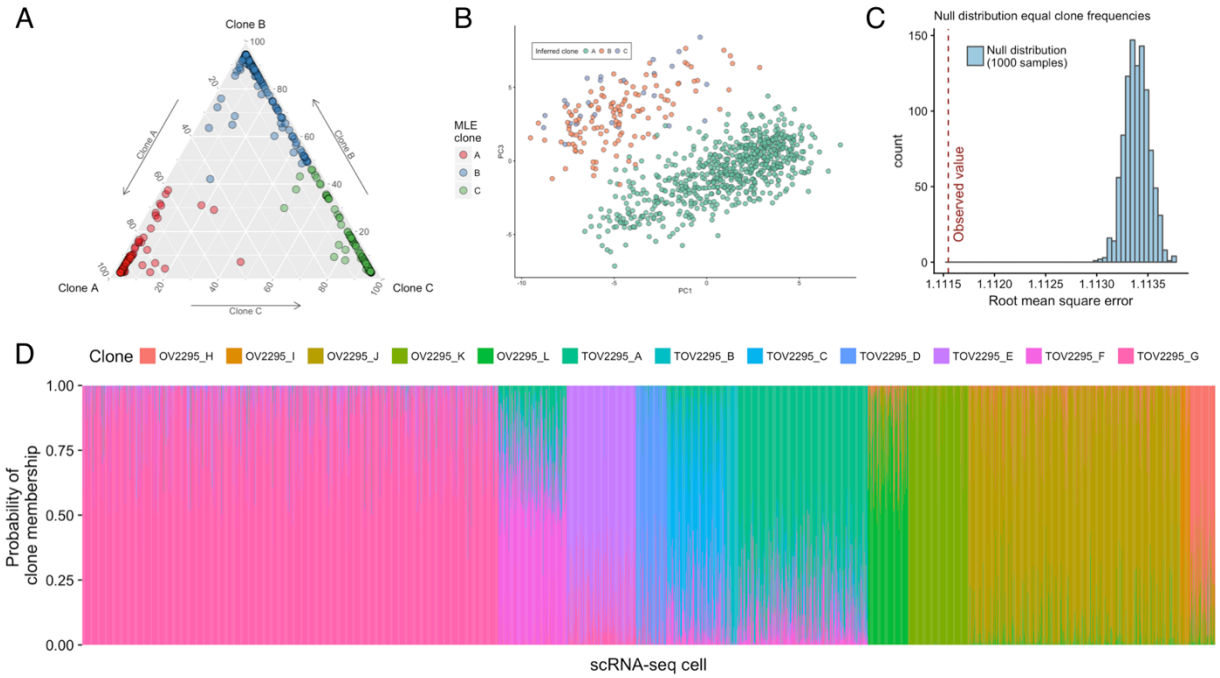


*Figure 2. (**A**) clonealign assigns single-cell RNA-seq to three distinct clones inferred from scDNA-seq data in the SA501 breast cancer cell line. (**B**) The cells separate by inferred clone along the first and third principal components in gene expression space. (**C**) Predicting the expression of held-out genes displays far higher accuracy than could be expected at random. (**D**) An example of a clonealign fit on the OV2295 ovarian cancer cell line containing over 4000 single-cells.*

Conclusions We describe *clonealign*, a method to assign gene expression states to cancer clones by aligning single-cell RNA-seq to copy number profiles measured using single-cell DNA-seq. We validate our method through multiply orthogonal analysis and demonstrate its utility on multiple datasets for which single-cell RNA and DNA-seq have been performed.

Bibliography

Macaulay, I. C. (2015). G&T-seq: parallel sequencing of single-cell genomes and transcriptomes. *Nature Methods*.

Zahn, H. a. (2017). Scalable whole-genome single-cell library preparation without preamplification. *Nature methods*.

Zheng, G. X. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature communications*.

IsoCon: Deciphering highly similar multigene family transcripts from Iso-Seq data

Background: A significant portion of genes in vertebrate genomes belongs to multigene families, with each family containing several gene copies that have arisen via duplication. Such copies vary in sequence identity and can produce different alternatively spliced forms (i.e. isoforms). Many duplicate genes have been associated with important human phenotypes, including a number of diseases, and in some of such cases, individual gene copies play different roles in disease etiology. The copy number of multigene families can be assayed using microarrays, quantitative PCR, droplet digital PCR, or DNA sequencing using Nanostring Technologies or Illumina platforms. The sequences of individual shorter exons can be obtained from Illumina DNA or RNA-seq data; however, alternative splicing and repetitive nature of duplicate gene copies complicates their *de novo* assemblies. Long PacBio reads from the Iso-Seq protocol overcome the assembly challenge by sequencing transcripts end to end, and has been successfully applied to reveal several complex isoform structures in, e.g., humans, plants, and fungi. However, the error rate in these reads makes it difficult, in the case of highly similar gene copies, to reconstruct end to end transcripts with nucleotide-level precision or assign alternatively spliced transcripts to their respective gene copies.

Results: We develop IsoCon, a *de novo* algorithm for error-correcting and removing redundancy of PacBio circular consensus sequence reads generated from targeted sequencing with the Iso-Seq protocol. Our algorithm allows one to decipher isoform sequences down to the nucleotide level and hypothesize how they are assigned to individual, highly similar gene copies of multigene families. IsoCon combines computational and statistical techniques to correct obvious errors and link variants across the transcript. Furthermore, IsoCon statistically integrates the large variability in read quality, which decreases as the transcript gets longer.

We evaluate IsoCon on simulated data and demonstrate that IsoCon has substantially higher precision and recall than its main competitor, ICE[1], across a wide range of sequencing depths, as well as of transcript lengths, similarities, and abundance levels. We also apply IsoCon to biological data from Y chromosome ampliconic gene families, a particularly interesting and challenging dataset to decipher, because each family contains several nearly identical (up to 99.99%) copies[2] with a potentially varying number of isoforms. We used a targeted design to isolate and sequence all nine Y chromosome ampliconic gene families from the testes of two men. Our validation using Illumina reads, previously annotated transcripts, and consistency in predictions between the two samples, shows that IsoCon drastically increases precision compared to both ICE and Illumina-based error correction with proovread [3] (Figure 1A) and has significantly higher recall than ICE (based on matches to database; Figure 1A). We show that IsoCon can detect rare transcripts that differ by as little as one base pair from dominant isoforms that have two orders of magnitude higher abundance. IsoCon's high sequence accuracy of the predicted transcripts enables us to further separate transcripts into putative gene copies and derive copy-specific exon sequences and splice variants. A demonstration of this is shown in Figure 1B-C, where IsoCon's predicted transcripts for the RBMY family is stacked on an artificially created reference for comparison.

Conclusions: We presented IsoCon, a method for deriving accurate transcript sequences from targeted Iso-Seq data. We showed that IsoCon *de novo* corrects reads and its accuracy allows us to phase highly similar transcripts on a SNV difference level. While demonstrating the applicability to decipher highly similar

transcripts from multi-gene families, we believe IsoCon will be useful also for phasing copies from diploid or polyploid organisms, or general error correction of Iso-Seq reads where high quality reference genome is not available.

A

	Predicted transcripts			
	reads (unique)	proovread	ICE	IsoCon
Sample1	11,413	6,183	281	406
Sample2	10,463	4,897	200	344

	Validation			
	reads (unique)	proovread	ICE	IsoCon
Illumina supported bases (%)	79	96	93	**99**
Transcripts with full Illumina support (%)	20	15	2	**70**
Matches to ENSEMBL	18	**23**	9	22
Concordance between samples (%)	1	10	2	30

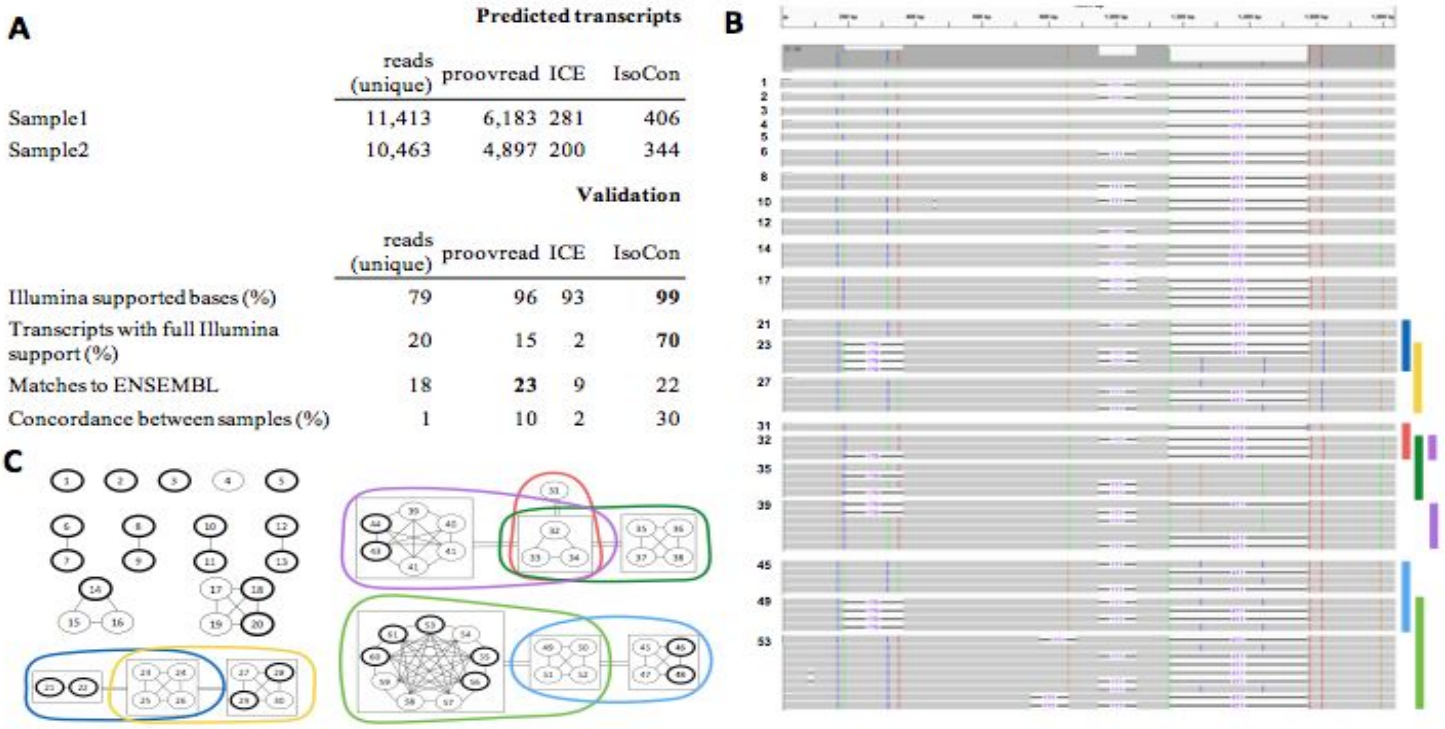


Figure 1. Summary of results for the 9 ampliconic gene families datasets. The upper table in panel A shows the number of unique predicted sequences in each sample for original reads, Illumina corrected reads (proovread), ICE, and IsoCon, respectively. The lower table shows summary metrics obtained from evaluating the predicted sequences using support from Illumina reads, exact matches to ENSEMBL database, and predictions shared between samples. Panel B: An IGV illustration of the multiple-alignment between the 61 RBMY transcripts predicted by IsoCon and shared by both samples. (C) Illustrates the relationship between the 61 transcripts as a graph. Vertices are transcripts and a vertex is boldfaced if it is predicted to be protein-coding. An edge between two transcripts means that they are potential isoforms from the same gene copy (*i.e.*, only exon presence/absence differences). To simplify the visualization, some of the vertices are surrounded by boxes, and a double-edge between two boxes indicates that all pairs of transcripts, between the two boxes, are potential isoforms from the same gene copy. Each maximal clique (*i.e. group* of vertices) greater than four vertices is shown as a colored circle and should be interpreted as transcripts potentially originating from the same gene copy.

1. Gordon, S. P. *et al.* Widespread Polycistronic Transcripts in Fungi Revealed by Single-Molecule mRNA Sequencing. *PLoS One* **10,** e0132628 (2015).
2. Bhowmick, B. K., Satta, Y. & Takahata, N. The origin and evolution of human ampliconic gene families and ampliconic structure. *Genome Res.* **17,** 441–450 (2007).
3. Hackl, T., Hedrich, R., Schultz, J. & Förster, F. proovread: large-scale high-accuracy PacBio correction through iterative short read consensus. *Bioinformatics* **30,** 3004–3011 (2014).

Bridging Linear to Graph-based Alignment with Whole Genome Population Reference Graphs

Recently, several attempts are devoted into building comprehensive catalogues of known genomic variants. However, read alignment approaches that efficiently utilize them are scarce. Since the catalogues contain hundreds of alleles which in general share most of their sequences except where the instant variations appear, that makes a graph of these alleles a reasonable and efficient representation of the data. Unfortunately, the lack of efficient implementations and algorithms for graph-based alignment makes graph-based approaches computationally expensive for practical application.

Our approach takes advantage of graph representation in obtaining prominent levels of data compressions, and efficiently linearizes the variants graph by sacrificing a portion of the compression ratio. Our model for linearizing the variants graph depends on our previous work in transcriptome segmentation for RNAseq. For each gene of interest, we start from the multiple sequence alignment (MSA) of the individual alleles (which can be already provided in the catalogue or derived from VCF files). Then we use Yanagi[1] to generate a set of *maximal L-disjoint* segments representing the linearized MSA graph (Figure 1). The segments library is then used by any alt-aware linear alignment tool.

The advantage of using our approach over the standard alt-aware aligners that uses a reference of the genome sequence appended by the population haplotypes is that segments sequences are highly compressed which is space efficient and speeds up the alignment process (Table 1). On the other hand, our approach is potentially flexible such that the generated segments can be used with most linear aligners rather than being limited to a specific graph model. Moreover, it avoids the expensive computational demands of aligning over graphs.

As a proof of concept, we test our approach on IPD-IMGT/HLA database to study six class I and class II HLA genes[2] with significant medical importance. In addition to testing using graph aligners (HISAT-genotype[3]), linear aligners (BWA-MEM), and linear aligners with segments (BWA-MEM), we also included a test for using fast and lightweight RNAseq aligners (RapMap) to examine the possibility of using fast RNAseq aligners for the task of read extraction. We simulated three datasets simulating three scenarios: 1) ClassI-Easy: reads are simulated from HLA class I alleles that are not very different from the reference genome, 2) ClassI-Hard: uses HLA class I alleles that are different from the reference, and 3) ClassII-Hard: uses HLA class II alleles that are much different from the reference. Preliminary results (Table 2) showed that the more divergent the samples are, the harder for linear aligners to correctly align reads. However, assisting linear aligners with the population segments can achieve comparable results to graph aligners without compromising the space and computational requirements (Table 3). Although RNAseq aligners with the reference alone performed the worst, adding segments elevated its performance back.

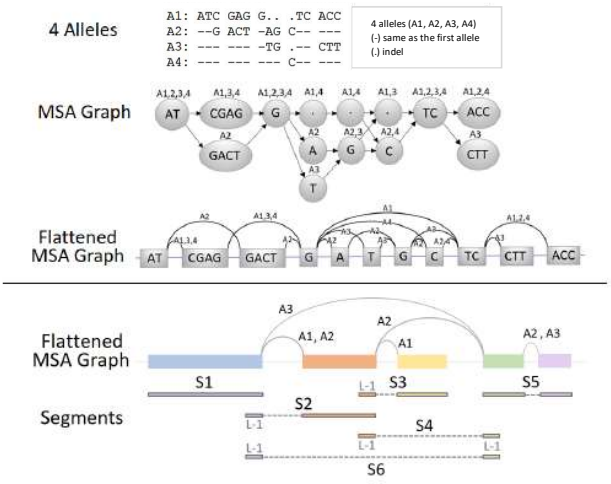


Figure 1: Illustrative examples for our segmentation model in two steps. (Top) Construct a flattened MSA graph of the gene's alleles. (Bottom) Create *maximal L-disjoint* segments of the population graph using Yanagi.

Table 1: Genome library size for the six HLA genes. In case of graph, number of bases is counted as the bases sum of the graph nodes.

	Reference	Ref+Alleles	Ref+Segments	Graph
Number of bases (Gb)	0.045	9.25	2.39	0.048
Number of sequences	6	2,094	45,609	2,094
FASTA file size (MB)	0.03	10	2.4	NA

Table 2: Number of correctly aligned reads from simulated reads using: HISAT-genotype (graph aligner), BWA-MEM (linear alt-aware aligner), and RapMap (RNAseq lightweight aligner). In case of both BWA-MEM and RapMap, results are shown either when using only the reference genome or the reference combined with yanagi's segments for the six HLA genes.

	Num.	HISAT-	BWA-MEM		RapMap	
	Reads	genotype	Ref	Ref+Segs	Ref	Ref+Segs
ClassI-Easy	6,000	5,900	6,000	6,000	4,163	5,990
ClassI-Hard	6,000	5,966	5,797	6,000	3,553	5,990
ClassII-Hard	14,000	13,844	12,232	13,997	7,628	13,975

Table 3: Running time for alignment of real sample NA12878.

	HISAT-genotype (Graph)	BWA-MEM (Ref+Segs)	RapMap (Ref+Segs)
Running Time	10 hours	3 hours	2 hours

[1] Gunady, Mohamed K., et al. "Yanagi: Transcript Segment Library Construction for RNA-Seq Quantification." LIPIcs-Leibniz International Proceedings in Informatics. Vol. 88. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017. (WABI-2017)

[2] The six genes: HLA-A, HLA-B, HLA-C, HLA-DQA1, HLA-DQB1, HLA-DRB1. We use L=150 for Yanagi in all experiments.

[3] Kim, Daehwan, Joseph M. Paggi, and Steven Salzberg. "HISAT-genotype: Next Generation Genomic Analysis Platform on a Personal Computer." bioRxiv (2018): 266197.

De novo single-cell transcript sequence reconstruction with Bloom filters

Background:

De novo transcript sequence reconstruction from RNA-seq data is a difficult problem due to the short read length and the wide dynamic range of transcript expression levels. Although more than 10 algorithms for bulk RNA-seq were published over the past decade, very limited effort was made for *de novo* single-cell transcript sequence reconstruction, likely due to the technical challenges in analyzing single-cell RNA-seq (scRNA-seq) data. Compared to bulk RNA-seq, scRNA-seq tend to yield more variable read depth across each transcript, lower transcript coverage, and lower overall signal-to-noise ratio. Therefore, isoform structure analysis at the single-cell level is almost non-existent and scRNA-seq is primarily used for gene expression analysis. Here, we present a fast and lightweight method for *de novo* single-cell transcript sequence reconstruction that leverages sequence reads across multiple cells.

Results:

Our method is implemented in a program called "RNA-Bloom," which utilizes lightweight probabilistic data structures based on Bloom filter for the in-memory storage of (i) de Bruijn graph (DBG), (ii) k-mer counts, (iii) k-mer pairs in input reads, and (iv) k-mer pairs in reconstructed read fragments. The overall workflow of RNA-Bloom for scRNA-seq data is summarized in Figure 1. To alleviate the detrimental effects of low transcript coverage and variable read-depth of scRNA-seq, a shared DBG, generated by pooling the input reads from all cells, is used for correcting errors in reads and reconstructing the read fragments for individual cells. To maintain cell-specificity during transcript reconstruction for each cell, the shared DBG is discarded and a new DBG is generated using only the reconstructed fragments of the corresponding cell.

RNA-Bloom was benchmarked against Trans-ABySS and Trinity, two state-of-the-art *de novo* assemblers for bulk RNA-seq, for the overall performance and accuracy using a 2x150-bp Illumina scRNA-seq dataset of 7 mouse B cells. This dataset was selected because it has complementary long reads from Oxford Nanopore Technologies, which are valuable for assessing the accuracy of the assembled transcripts. Using 12 CPUs, RNA-Bloom has a peak memory usage of 1.64 GB and a total runtime of 3.85 minutes. Trans-ABySS has a peak-memory usage of 0.71 GB and total runtime of 13.9 minutes. Trinity has a peak-memory usage of 5.38 GB and total runtime of 80.6 minutes. As shown in Figure 2, RNA-Bloom has better overall transcript reconstruction than Trans-ABySS and Trinity.

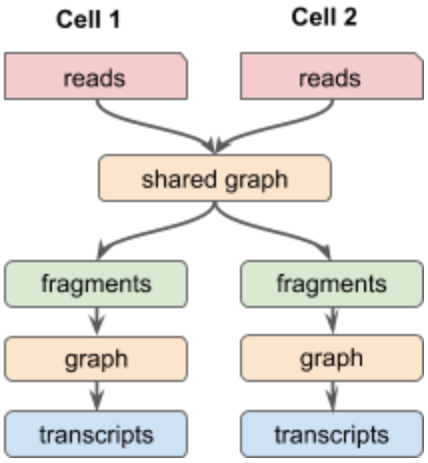


Figure 1. Workflow of RNA-Bloom.

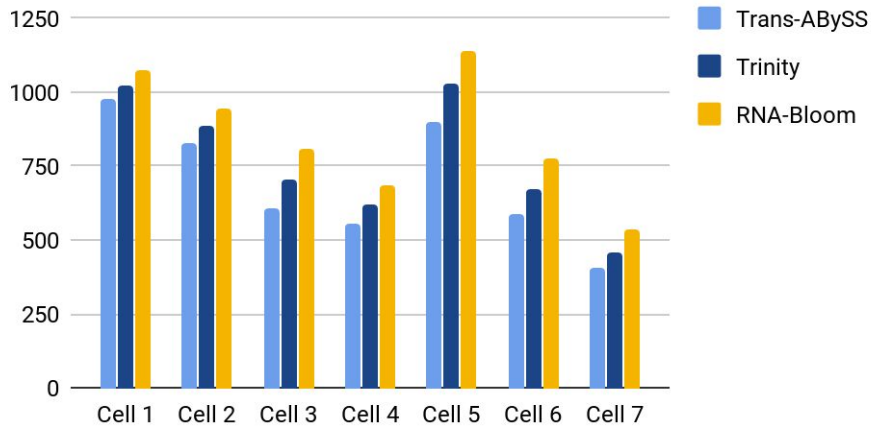


Figure 2. Isoforms ≥ 50% assembled by a single contig.

Conclusions:

In summary, RNA-Bloom is a lightweight method for transcript reconstruction from scRNA-seq data. RNA-Bloom's performance and accuracy surpasses state-of-the-art methods that were designed for bulk RNA-seq data. While scRNA-seq has traditionally been used for gene expression analysis, this work unlocks new territory for identifying unique isoform structures at the single-cell level.

Jointly aligning a group of DNA reads improves accuracy of identifying large deletions

Split-alignments, which are alignments where two different portions of a read align to disjoint genomic locations on the reference, are direct evidence of structural variants (SV), and a crucial step in the analysis of high throughput sequencing assays. Accurately computing split-alignments remains a very challenging problem [1, 2], owing to the highly repetitive nature of genomes and the propensity of genomic rearrangements to accumulate in the vicinity of repeats [3]. This affects the ambiguity of reported pairwise alignments in a severe way. Indeed, by analysing the variants reported in the Venter genome [4], we showed that this issue cannot be ignored, as **even under ideal conditions** of no sequencing errors and very high coverage, **40% of deletions** $\geq$ 32bp **cannot be identified with certainty** by pairwise alignments of 100bp-long reads (13% for paired-end reads).

In this regard, current techniques based on split-alignment have two major shortcoming: (1) they do not account for alignment uncertainties that could be computed using probabilistic models [5], and (2) reads are aligned *independently* of each other. Since those reads are in fact highly correlated, utilizing information from the group as a whole, can mitigate misalignment issues arising due to repeat-rich genomic context of SVs or due to sequencing errors.

Here we take a slight departure from the conventional align-and-call workflow, and propose a new framework of *jointly* aligning a group of reads identifying a common genomic structural variant. Our method measures the uncertainty of each reported joint split-alignment, based on a probabilistic model of sequence alignment. Additionally, we show how to incorporate paired-end reads in our workflow by using pairing information to improve the confidence value of a prediction.

We demonstrated the advantages of our method, JRA, over other split-aligners, by applying it to the problem of identifying medium and large deletions ($\geq$ 20bp) from typical human genome resequencing datasets, both simulated (Fig. 1A) and real (Fig. 1B, [6]).

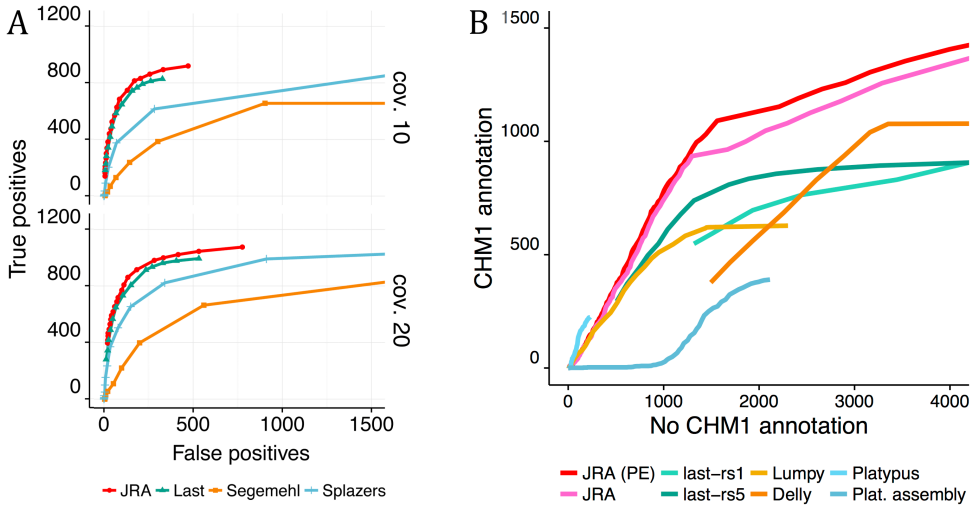


Figure 1: Performance of split alignment for simulated and real data (A) Performance on simulated reads (100bp) from chr. 1+2 of the C. Venter genome, for two values of expected coverage, with JRA (our method), LAST, Splazers, and Segemehl (B) Performance for Illumina paired-end reads (101bp, 41x coverage) from the CHM1 cell line (SRX652547). The deletions were annotated from long read PacBio sequencing of the same cell line [6]. Comparison of JRA (our method paired-end and default mode), LAST (min. read support: 1, 5), Lumpy, Delly, and Platypus (default, assembly option activated).

We also contrasted our technique of bolstering otherwise ambiguous split alignments by combining read group and paired-end information to the conventional method of detecting deletions through discordant alignments of paired-end reads. Quite surprisingly, we found that these conventional methods have lower sensitivity in practice, as they miss a majority of deletions in their attempt to account for variability in fragment size. Finally, we showed that the computational overhead of our method is small with an overall running time well within the range of contemporary methods.

[1] Alkan, C. *et al.* Genome structural variation discovery and genotyping. *Nature reviews. Genetics*, 2011.

[2] Medvedev, P. *et al.* Computational methods for discovering structural variation with next-generation sequencing. *Nature methods*, 2009.

[3] Treangen, TJ and Salzberg, SL. Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nature Reviews Genetics*, 2012.

[4] Levy, S. *et al.* The diploid genome sequence of an individual human. *PLoS Biol*, 2007.

[5] Durbin, R. *et al. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[6] Chaisson, M. *et al.* Resolving the complexity of the human genome using single-molecule sequencing. *Nature*, 2015. Letter.

ABySS-LR: *de novo* Assembly Pipeline for Linked Reads

Background

Recently, 10x Genomics introduced the Chromium library preparation protocol for augmenting Illumina paired-end reads with long range linkage information ("linked reads"). Under the Chromium protocol, each read pair is tagged with a 16 bp barcode that associates it to one or more long DNA molecule(s) up to 100 kbp in length, providing invaluable information for resolving genomic repeat structures during *de novo* assembly.

Results

Here we present ABySS-LR, a linked read assembly pipeline that uses the Chromium barcode information to resolve repeat components, detect and cut misassemblies, and build long-range scaffolds. In the first step (Fig. 1A), we perform an ordinary de Bruijn graph assembly of the linked reads, without using barcode information. In the second step (Fig. 1B), we link unitigs across unresolved repeat components based on unitig-to-barcode associations. In particular, we test the validity of connecting paths between unitigs using a statistical model that relates the fraction of shared barcodes between two sequences to their distance. In the third step (Fig. 1C), we use the Tigmint misassembly detection tool identify and cut probable misassemblies in contigs. Tigmint calculates the physical coverage profile of the Chromium long molecules and locates gaps in coverage to identify probable misassemblies. In the final step (Fig. 1D), we use ARCS to build a scaffold graph in which nodes represent contigs, and edge weights represent the number of Chromium barcodes shared between contig heads/tails. We then traverse high-confidence paths within the ARCS graph to generate the output scaffold sequences.

ABySS-LR is being developed for assembly of large genomes with multiple Chromium libraries, in conjunction with other sequencing data types such as paired-end reads, mate pair reads, and long reads. On a linked reads data set for human chromosome 21, ABySS-LR yields an NA50 length of 5.1 Mbp, which represents a 50X improvement over a standard ABySS v2.0 assembly.

Conclusion

ABySS-LR leverages the long-range information provided by Chromium linked reads to substantially improve the contiguity and correctness of *de novo* genome assemblies.

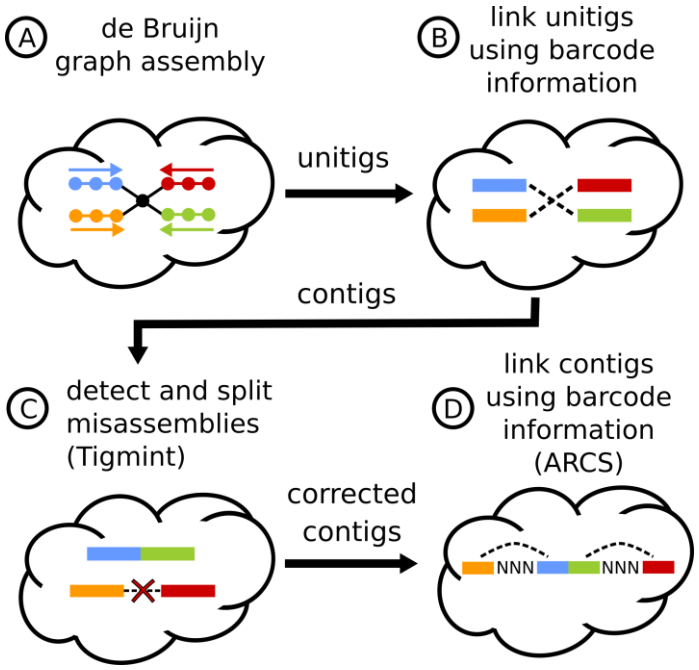


Figure 1: ABySS-LR linked reads assembly pipeline. (A) Unitigs are assembled using a standard de Bruijn graph assembly. (B) Unitigs are linked across unresolved repeat components using barcode information. (C) Tigmint detects and cuts misassemblies based on gaps in the physical coverage of Chromium long molecules. (D) ARCS links contigs into scaffolds using shared barcode information.

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

Integrating Hi-C links with assembly graphs for chromosome-scale assembly

Jay Ghurye [1,2], Arang Rhie [2], Brian P. Walenz [2], Anthony Schmitt [3], Siddarth Selvaraj [3], Mihai Pop [1], Adam M. Phillippy [2,*], and Sergey Koren [2,*]

[1] Department of Computer Science, University of Maryland, College Park, USA
[2] Genome Informatics Section, Computational and Statistical Genomics Branch, National Human Genome Research Institute, National Institutes of Health, Bethesda, USA and
[3] Arima Genomics, Inc, San Diego, USA

* To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Long-read sequencing and novel long-range assays have revolutionized *de novo* genome assembly by automating the reconstruction of reference-quality genomes. In particular, Hi-C sequencing is becoming an economical method for generating chromosome-scale scaffolds. Despite its increasing popularity, there are limited open-source tools available. Errors, particularly inversions and fusions across chromosomes, remain higher than alternate scaffolding technologies.
Results: We present a novel open-source Hi-C scaffolder that does not require an *a priori* estimate of chromosome number and minimizes errors by scaffolding with the assistance of an assembly graph. We demonstrate higher accuracy than the state-of-the-art methods across a variety of Hi-C library preparations and input assembly sizes.
Availability and Implementation: The Python and C++ code for our method is openly available at https://github.com/machinegun/SALSA
Contact: sergey.koren@nih.gov, adam.phillippy@nih.gov
Supplementary information: Not available online.

1 Introduction

Genome assembly is the process of reconstructing a complete genome sequence from significantly shorter sequencing reads. Most genome projects rely on whole genome shotgun sequencing which yields an oversampling of each genomic locus. Reads originating from the same locus are identified using assembly software, which can use these overlaps to reconstruct the genome sequence (Nagarajan and Pop, 2013; Miller *et al.*, 2010). Most approaches are based on either a de Bruijn (Pevzner *et al.*, 2001) or a string graph (Myers, 2005) formulation. Repetitive sequences exceeding the sequencing read length (Nagarajan and Pop, 2009) introduce ambiguity and prevent complete reconstruction. Unambiguous reconstructions of the sequence are output as "unitigs" (or often "contigs"). Ambiguous reconstructions are output as edges linking unitigs. Scaffolding utilizes long-range linking information such as BAC or fosmid clones (Venter *et al.*, 1996; Gnerre *et al.*, 2011), optical maps (Schwartz *et al.*, 1993; Dong *et al.*, 2013; Shelton *et al.*, 2015), linked reads (Zheng *et al.*, 2016; Weisenfeld *et al.*, 2017; Yeo *et al.*, 2017), or chromosomal conformation capture (Simonis *et al.*, 2006) to order and orient unitigs. If the linking information spans large distances on the chromosome, the resulting scaffolds can span entire chromosomes or chromosome arms.

Hi-C is a sequencing-based assay originally designed to interrogate the 3D structure of the genome inside a cell nucleus by measuring the contact frequency between all pairs of loci in the genome (Lieberman-Aiden *et al.*, 2009). The contact frequency between a pair of loci strongly correlates with the one-dimensional distance between them. Hi-C data can provide linkage information across a variety of length scales, spanning tens of megabases. As a result, Hi-C data can be used for genome scaffolding. Shortly after its introduction, Hi-C was used to generate chromosome-scale scaffolds (Burton *et al.*, 2013; Kaplan and Dekker, 2013; Marie-Nelly *et al.*, 2014; Bickhart *et al.*, 2017; Dudchenko *et al.*, 2017).

LACHESIS (Burton *et al.*, 2013) is an early method for Hi-C scaffolding which first clusters unitigs into a user-specified number of chromosome groups and then orients and orders the unitigs in each group independently to generate scaffolds. Thus, the scaffolds inherit any assembly errors present in the unitigs. The original SALSA (Ghurye *et al.*, 2017) method first corrects the input assembly, using a lack of Hi-C coverage as evidence of error. It then orients and orders the corrected unitigs to generate scaffolds. However, SALSA requires manual parameter tuning for each dataset which affects the contiguity and correctness of the final scaffolds. Recently, the 3D-DNA (Dudchenko *et al.*, 2017) method was introduced and demonstrated on a draft assembly of the *Aedes aegypti* genome. 3D-DNA also corrects the errors in the input assembly and then iteratively orients and orders unitigs into a single megascaffold. This megascaffold is then broken into a user-specified number of chromosomes, identifying chromosomal ends based the on Hi-C contact map.

There are several shortcomings common across currently available tools. They require the user to specify the number of chromosomes *a priori*. This can be challenging in novel genomes where no karyotype is available. An incorrect guess often leads to mis-joins that fuse chromosomes. They are also sensitive to input assembly contiguity and Hi-C library variations and require tuning of parameters for each dataset. Inversions are common when the input unitigs are short, as orientation is determined by maximizing the interaction frequency between unitig ends across all possible orientations (Burton *et al.*, 2013). When unitigs are long, there are few interactions spanning the full length of the unitig, making the true orientation apparent from the higher weight of links. However, in the case of short unitigs, there are interactions spanning the full length of the unitig, making the true orientation have a similar weight to incorrect orientations. Biological factors, such as topologically associated domains (TADs) also confound this analysis (Dixon *et al.*, 2012).

In this work, we introduce SALSA2 – an open source software that combines Hi-C linkage information with the ambiguous-edge information from a genome assembly graph to better resolve unitig orientations. We also propose a novel stopping condition, which does not require an *a priori* estimate of chromosome count, as it naturally stops when the Hi-C information is exhausted. We show that SALSA2 has fewer orientation, ordering, and chimeric errors across a wide range of assembly contiguities. We also demonstrate robustness to different Hi-C libraries with varying intra-chromosomal contact frequencies. When compared to 3D-DNA, SALSA2 generates more accurate scaffolds across all conditions tested. To our knowledge, this is the first method to leverage assembly graph information for scaffolding Hi-C data.

2 Methods

Figure 1(A) shows the overview of the SALSA2 pipeline. A draft assembly is generated from long reads such as Pacific Biosciences (Eid *et al.*, 2009) or Oxford Nanopore (Jain *et al.*, 2016). SALSA2 requires the unitig sequences and, optionally, a GFA-format graph (Li, 2016) representing the ambiguous reconstructions. Hi-C reads are aligned to the unitig sequences, and unitigs are optionally split in regions lacking Hi-C coverage. A hybrid scaffold graph is constructed using both ambiguous edges from the GFA and edges from the Hi-C reads, scoring edges according to a "best buddy" scheme. Scaffolds are iteratively constructed from this graph using a greedy weighted maximum matching. A mis-join detection step is performed after each iteration to check if any of the joins made during this round are incorrect. Incorrect joins are broken and the edges blacklisted during subsequent iterations. This process continues until the majority of joins made in the prior iteration are incorrect. This provides a natural stopping condition, when accurate Hi-C links have been exhausted. Below, we describe each of the steps in detail.

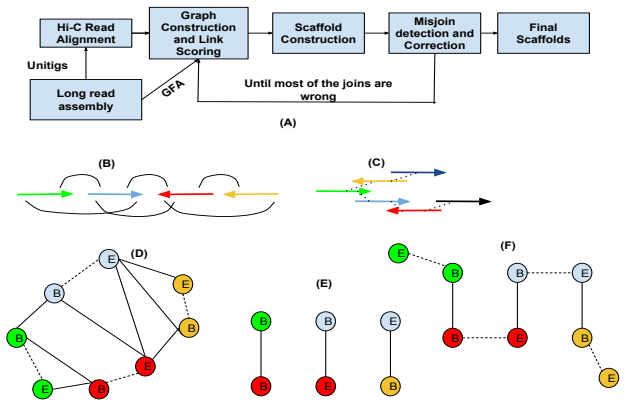


Fig. 1. (A) Overview of the SALSA2 scaffolding algorithm. (B) Linkage information obtained from the alignment of Hi-C reads to the assembly. (C) The assembly graph obtained from the assembler. (D) A hybrid scaffold graph constructed from the links obtained from the Hi-C read alignments and the overlap graph. Solid edges indicate the linkages between different unitigs and dotted edges indicate the links between the ends of the same unitig. (E) Maximal matching obtained from the graph using a greedy weighted maximum matching algorithm. (F) Edges between the ends of same unitigs are added back to the matching.

2.1 Read alignment

Hi-C paired end reads are aligned to unitigs using the BWA aligner (Li and Durbin, 2009)(parameters: -t 12 -B 8) as single end reads. Reads which align across ligation junctions are chimeric and are trimmed to retain only the start of the read which aligns prior to the ligation junction. After filtering the chimeric reads, the pairing information is restored. Any PCR duplicates in the paired-end alignments are removed using Picard tools (Wysoker *et al.*, 2013). Read pairs aligned to different unitigs are used to construct the initial scaffold graph. The suggested mapping pipeline is available at http://github.com/ArimaGenomics/mapping_pipeline.

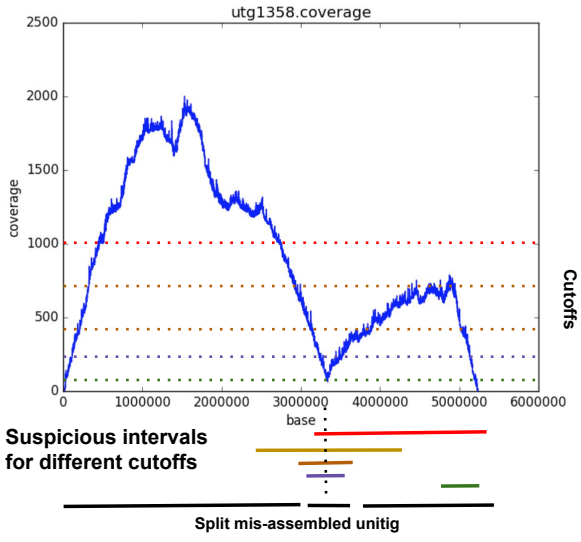


Fig. 2. Example of the mis-assembly detection algorithm in SALSA2. The plot shows the position on x-axis and the physical coverage on the y-axis. The dotted horizontal lines show the different thresholds tested to find low physical coverage intervals. The lines at the bottom show the suspicious intervals identified by the algorithm. The dotted line through the intervals shows the maximal clique. The smallest interval (purple) in the clique is identified as mis-assembly and the unitig is broken in three parts at its boundaries.

2.2 Unitig correction

As any assembly is likely to contain mis-assembled sequences, SALSA2 uses the physical coverage of Hi-C pairs to identify suspicious regions and break the sequence at the likely point of mis-assembly. We define the physical coverage of a Hi-C read pair as the region on the unitig spanned by the start of the leftmost fragment and the end of the rightmost fragment. A drop in physical coverage indicates a likely assembly error. We extend the mis-assembly detection algorithm from SALSA which split a unitig when a fixed minimum coverage threshold was not met. A drawback of this approach is that coverage can vary, both due to sequencing depth and variation in Hi-C link density.

Figure 2 sketches the new unitig correction algorithm implemented in SALSA2. Instead of a single coverage threshold, a set of suspicious intervals is found with a sweep of thresholds. Using the collection of intervals as an interval graph, we find the maximal clique. This can be done in $O(NlogN)$ time, where N is the number of intervals. For any clique of a minimum size, the region between the start and end of the smallest interval in the clique is flagged as a mis-assembly and the unitig is split into three pieces — the sequence to the left of the region, the junction region itself, and the sequence to the right of the region.

2.3 Assembly graph construction

For our experiments, we use the unitig assembly graph produced by Canu (Koren *et al.*, 2017) (Figure 1(C)), as this is the more conservative graph output. SALSA2 requires only a GFA format (Li, 2016) representation of the assembly. Since most long read genome assemblers such as FALCON (Chin *et al.*, 2016), miniasm (Li, 2016), Canu (Koren *et al.*, 2017), and Flye (Kolmogorov *et al.*, 2018) provide assembly graphs in GFA format, their output is compatible with SALSA2 for scaffolding.

2.4 Scaffold graph construction

The scaffold graph is defined as $G(V, E)$, where nodes V are the ends of unitigs and edges E are derived from the Hi-C read mapping (Figure 1B). The idea of using unitig ends as nodes is similar to that used by the string graph formulation (Myers, 2005).

Modeling each unitig as two nodes allows a pair of unitigs to have multiple edges in any of the four possible orientations (forward-forward, forward-reverse, reverse-forward, and reverse-reverse). The graph then contains two edge types - one explicitly connects two different unitigs based on Hi-C data, while the other implicitly connects the two ends of the same unitig.

We normalize the Hi-C read counts by the frequency of restriction enzyme cut sites in each unitig. This normalization reduces the bias in the number of shared read pairs due to the unitig length as the number of Hi-C reads sequenced from a particular region are proportional to the number of restriction enzyme cut sites in that region. For each unitig, we denote the number of times a cut site appears as $C(V)$. We define edges weights of G as:

$$W(u,v) = \frac{N(u,v)}{C(u)+C(v)}$$

where $N(u, v)$ is the number of Hi-C read pairs mapped to the ends of the unitigs u and v.

We observed that the globally highest edge weight does not always capture the correct orientation and ordering information due to variations in Hi-C interaction frequencies within a genome. To address this, we defined a modified edge ratio, similar to the one described in (Dudchenko *et al.*, 2017), which captures the relative weights of all the neighboring edges for a particular node.

The best buddy weight $BB(u, v)$ is the weight $W(u, v)$ divided by the maximal weight of any edge incident upon nodes u or v, excluding the (u, v) edge itself. Computing best buddy weight naively would take $O(|E|^2)$ time. This is computationally prohibitive since the graph, G, is usually dense. If the maximum weighted edge incident on each node is stored with the node, the running time for the computation becomes $O(|E|)$. We retain only edges where $BB(u, v) > 1$. This keeps only the edges which are the best incident edge on both u and v. Once used, the edges are removed from subsequent iterations. Thus, the most confident edges are used first but initially low scoring edges can become best in subsequent iterations.

For the assembly graph, we define a similar ratio. Since the edge weights are optional in the GFA specification and do not directly relate to the proximity of two unitigs on the chromosome, we use the graph topology to establish this relationship. Let $\bar{u}$ denote the reverse complement of the unitig u. Let $\sigma(u, v)$ denote the length of shortest path between u and v. For each edge (u, v) in the scaffold graph, we find the shortest path between unitigs u and v in every possible orientation, that is, $\sigma(u, v)$, $\sigma(u, \bar{v})$, $\sigma(\bar{u}, v)$ and $\sigma(\bar{u}, \bar{v})$. With this, the score for a pair of unitigs is defined as follows:

$$Score(u,v) = \frac{\min\limits_{x' \in \{u,\bar{u}\}-\{x\}, y' \in \{v,\bar{v}\}-\{y\}} \sigma(x',y')}{\min\limits_{x \in \{u,\bar{u}\}, y \in \{v,\bar{v}\}} \sigma(x,y)}$$

where x and y are the orientations in which u and v are connected by a shortest path in the assembly graph. Essentially, $Score(u, v)$ is the ratio of the length of the second shortest path to the length of the shortest path in all possible orientations. Once again, we retain edges where $Score(u, v) > 1$. If the orientation implied by the assembly graph differs from the orientation implied by the Hi-C data, we remove the Hi-C edge and retain the assembly graph edge (Figure 1D). Computing the score graph requires $|E|$ shortest path queries, yielding total runtime of $O(|E| * (|V| + |E|))$ since we do not use the edge weights.

2.5 Unitig layout

Once we have the hybrid graph, we lay out the unitigs to generate scaffolds. Since there are implicit edges in the graph G between the beginning and end of each unitig, the problem of computing a scaffold layout can be modeled as finding a weighted maximum matching in a general graph, with edge weights being our ratio weights. If we find the weighted maximum matching of the non-implicit edges (that is, edges between different unitigs) in the graph, adding the implicit edges to this matching would yield a complete traversal. However, adding implicit edges to the matching can introduce a cycle. Such cycles are removed by removing the lowest weight non-implicit edge. Computing a maximal matching takes $O(|E||V|^2)$ time (Edmonds, 1965). We iteratively find a maximum matching in the graph by removing nodes found in the previous iteration. Using the optimal maximum matching algorithm this would take $O(|E||V|^3)$ time, which would be extremely slow for large graphs. Instead, we use a greedy maximal matching algorithm which is guaranteed to find a matching within 1/2-approximation of the optimum (Poloczek and Szegedy, 2012). The greedy matching algorithm takes $O(|E|)$ time, thereby making the total runtime $O(|V||E|)$. The algorithm for unitig layout is sketched in Algorithm 1. Figure 1(D - F) show the layout on an example graph.

Junctions in the graph can prevent some nodes from being included in larger scaffolds. At a junction, only one of the possible unitigs can be included in the matching, demoting the other unitigs at the junction to alternate matchings. To account for this, we try to insert unitigs from small scaffolds (less than five unitigs) into all possible positions in the large scaffolds in all possible orientations. A unitig is inserted into the scaffold at the position and orientation which maximizes the sum of edge weights between it and all adjacent unitigs at that location. If the gain in the sum

of edge weights is not sufficient, the unitig is not inserted into any of the existing scaffolds but can be scaffolded in subsequent iterations.

Algorithm 1 Unitig Layout Algorithm

E : Edges sorted by the best buddy weight
M : Set to store maximal matchings
G : The scaffold graph
while all nodes in G are not matched **do**
  $M^* = \{\}$
  for $e \in E$ sorted by best buddy weights **do**
    if e can be added to M^* **then**
      $M^* = M^* \cup e$
    end if
  end for
  $M = M \cup M^*$
  Remove nodes and edges which are part of M^* from G
end while

Algorithm 2 Misjoin detection and correction algorithm

Cov : Physical coverage array for a window size w around a scaffold join at position p on a scaffold
A : Auxiliary array
I : Maximum sum subarray intervals
for $\delta \in$ {min_coverage, max_coverage} **do**
  if $Cov[i] \leq \delta$ **then**
    $A[i] = 1$
  else
    $A[i] = -1$
  end if
  $s_\delta, e_\delta =$ maximum_sum_subarray(A)
  $I = I \cup \{s_\delta, e_\delta\}$
end for
$s, e =$maximal_clique_interval(I)
if $p \in \{s, e\}$ **then**
  Break the scaffold at position p
end if

2.6 Iterative mis-join correction

Since the unitig layout is greedy, it can introduce errors by selecting a false Hi-C link which was not eliminated by our ratio scoring. These errors propagate downstream, causing large chimeric scaffolds and chromosomal fusions. We examine each join made within all the scaffolds in the last iteration for correctness. Any join with low spanning Hi-C support relative to the rest of the scaffold is broken and the links are blacklisted for further iterations.

We compute the physical coverage spanned by all read pairs aligned in a window of size w around each join. For each window, w, we create an auxiliary array, which stores -1 at position i if the physical coverage is greater than some cutoff δ and 1, otherwise. We then find the maximum sum subarray in this auxiliary array, since it captures the longest stretch of low physical coverage. If the position being tested for a mis-join lies within the region spanned by the maximal clique generated with the maximum sum subarray intervals for different cutoffs (Figure 2), the join is marked as incorrect. The physical coverage can be computed in $O(w + N)$ time, where N is the number of read pairs aligned in window w. The maximum sum subarray computation takes $O(w)$ time. If K is the number of cutoffs(δ) tested for the suspicious join finding, then the total runtime of mis-assembly detection becomes $O(K(N + 2 * w))$. The parameter K controls the specificity of the mis-assembly detection, thereby avoiding false positives. The algorithm for mis-join detection is sketched in Algorithm 2. When the majority of joins made in a particular iteration are flagged as incorrect by the algorithm, SASLA2 stops scaffolding and reports the scaffolds generated in the penultimate iteration as the final result.

3 Results

3.1 Dataset description

We created artificial assemblies, each containing unitigs of same size, by splitting the GRCh38 (Schneider *et al.*, 2017) reference into fixed sized unitigs of 200 to 900 kbp. This gave us eight assemblies. The assembly graph for each input is built by adding edges for any adjacent unitigs in the genome.

For real data, we use the recently published NA12878 human dataset sequenced with Oxford Nanopore (Jain *et al.*, 2017) and assembled with Canu (Koren *et al.*, 2017). We use a Hi-C library from Arima Genomics (Arima Genomics, San Diego, CA) sequenced to 40x coverage (SRSXXX). We compare results with the original SALSA, SALSA2 without the assembly graph input, and 3D-DNA. We did not compare our results with LACHESIS because it is no longer supported and is outperformed by 3D-DNA (Dudchenko *et al.*, 2017). SALSA2 was run using default parameters, with the exception of graph incorporation, as listed. For 3D-DNA, alignments were generated using the Juicer alignment pipeline (Durand *et al.*, 2016b) with defaults (-m haploid -t 15000 -s 2), except for mis-assembly detection, as listed. The chromosome number was set to 23 for all experiments. A genome size of 3.2 Gbp was used for contiguity statistics for all assemblies.

For evaluation, we also used the GRCh38 reference to define a set of true and false links from the Hi-C graph. We mapped the assembly to the reference with MUMmer3.23 (nucmer -c 500 -l 20) (Kurtz *et al.*, 2004) and generated a tiling using MUMmer's show-tiling utility. For this "true link" dataset, any link joining unitigs in the same chromosome in the correct orientation was marked as true. This also gives the true unitig position, orientation, and chromosome assignment. We masked sequences in GRCh38 which matched known structural variants from a previous assembly of NA12878 (Pendleton *et al.*, 2015) to avoid counting true variations as scaffolding errors.

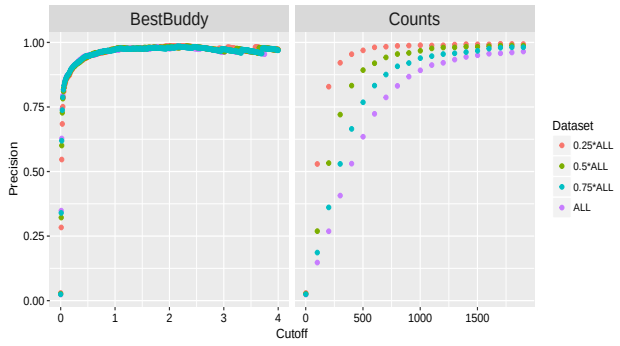


Fig. 3. Precision at different cutoffs for Hi-C links. The plot on the left shows the curve for the SALSA2 best buddy weight cutoffs and the plot on the right shows the curve for a fixed Hi-C pair count cutoff, used in SALSA1, across changing coverage.

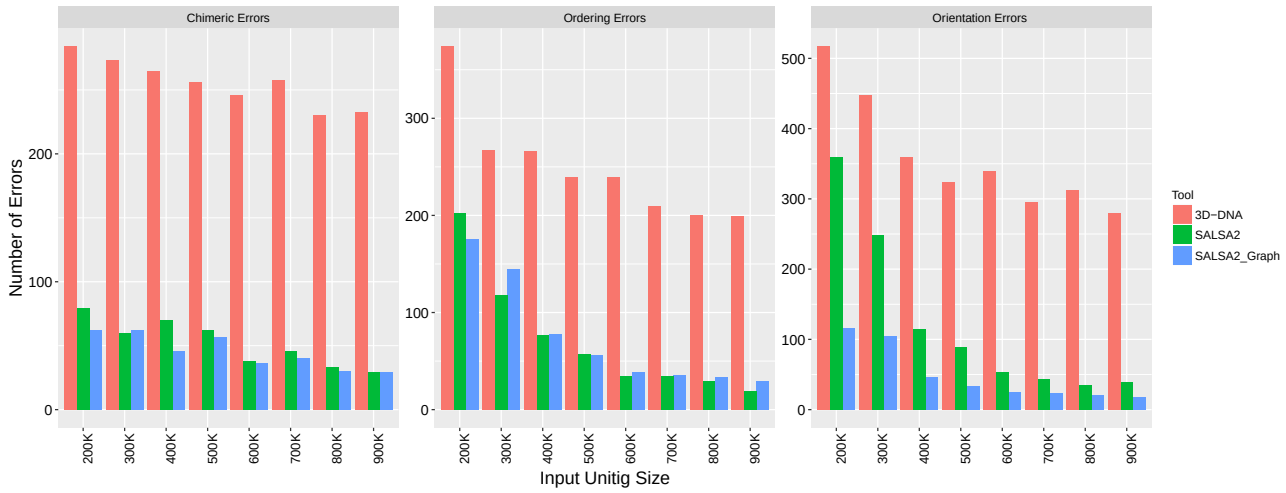


Fig. 4. Comparison of orientation, ordering, and chimeric errors in the scaffolds produced by SALSA2 and 3D-DNA on the simulated data. As expected, the number of errors for all error types decrease with increasing input unitig size. Incorporating the assembly graph reduces error across all categories and most assembly sizes, with the largest decrease seen in orientation errors. SALSA2 utilizing the graph has 2-4 fold fewer errors than 3D-DNA.

3.2 Scoring effectiveness

For correct scaffolding, we want to filter false edges and retain only the correct linkage information between pairs of unitigs. Our previous algorithm used a fixed, user-defined minimum for edges connecting a pair of unitigs. The drawback of a fixed cutoff is that it cannot handle variations in coverage within the assembly and varies between any pair of sequencing datasets. To compare the scoring methods, we down-sample the alignments into three different sets with 0.25, 0.5 and 0.75 of the original coverage and computed the precision of filtering based on the ratio score and a fixed threshold. The precision remained almost constant for the ratio cutoff on all datasets, whereas the precision changes rapidly for different coverages and a fixed threshold (Figure 3).

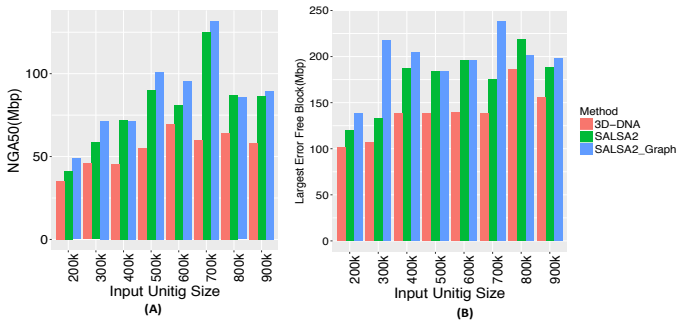


Fig. 5. (A) NGA50 statistic for different input unitig sizes and (B) The length of longest error-free block for different input unitig sizes. Once again, the assembly graph typically increases both the NGA50 and the largest correct block.

3.3 Evaluation on simulated unitigs

3.3.1 Assembly correction

We simulated assembly error by randomly joining 200 pairs of unitigs from each simulated assembly. All erroneous joins were made between unitigs that are more than 10 Mbp apart or were assigned to different chromosomes in the reference. The remaining unitigs were unaltered. We then aligned the Arima-HiC data and ran our assembly correction algorithm. When the algorithm marked a mis-join within 20 kbp of a true error we called it a true positive, otherwise we called it a false positive. Any unmarked error was called a false negative. The average sensitivity over all simulated assemblies was 77.62% and the specificity was 86.13%. The sensitivity was highest for larger unitigs (50% for 200 kbp versus >90% for untigs greater than 500 kbp) implying that our algorithm is able to accurately identify errors in large unitigs, which can have a negative impact on the final scaffolds if not corrected.

3.3.2 Scaffold mis-join validation

As before, we simulated erroneous scaffolds by joining unitigs which were not within 10 Mbp in the reference or were assigned to different chromosomes. Rather than pairs of unitigs, each erroneous scaffold joined 10 unitigs and we generated 200 such erroneous scaffolds. The remaining unitigs were correctly scaffolded (ten unitigs per scaffold) based on their location in the reference. The average sensitivity was 68.89% and specificity was 100% (no correct scaffolds were broken). Most of the un-flagged joins occurred near the ends of scaffolds and could be captured by decreasing the window size. Similar to assembly correction, we observed that sensitivity was highest with larger input unitigs. This evaluation highlights the accuracy of the mis-join detection algorithm to avoid over-scaffolding and provide a suitable stopping condition.

3.3.3 Scaffold accuracy

We evaluated scaffolds across three categories of error: orientation, order, and chimera. An orientation error occurs whenever the orientation of a unitig in a scaffold differs from that of the scaffold in the reference. An ordering error occurs when a set of three unitigs adjacent in a scaffold have non-monotonic coordinates in the reference. A chimera error occurs when any pair of unitigs adjacent in a scaffold align to different chromosomes in the reference. We broke the assembly at these errors and computed corrected scaffold lengths and NGA50 (analogous to the NGA50 defined by Salzberg et al. (Salzberg *et al.*, 2012)). This statistic corrects for large but incorrect scaffolds which have a high NG50 but are not useful for downstream analysis because of errors.

Hi-C scaffolding errors, particularly orientation errors, increased with decreasing assembly contiguity. We evaluated scaffolding methods across a variety of simulated unitig sizes. Figure 4 shows the comparison of these errors for 3D-DNA, SALSA2 without the assembly graph, and SALSA2 with the graph. SALSA2 produced fewer errors than 3D-DNA across

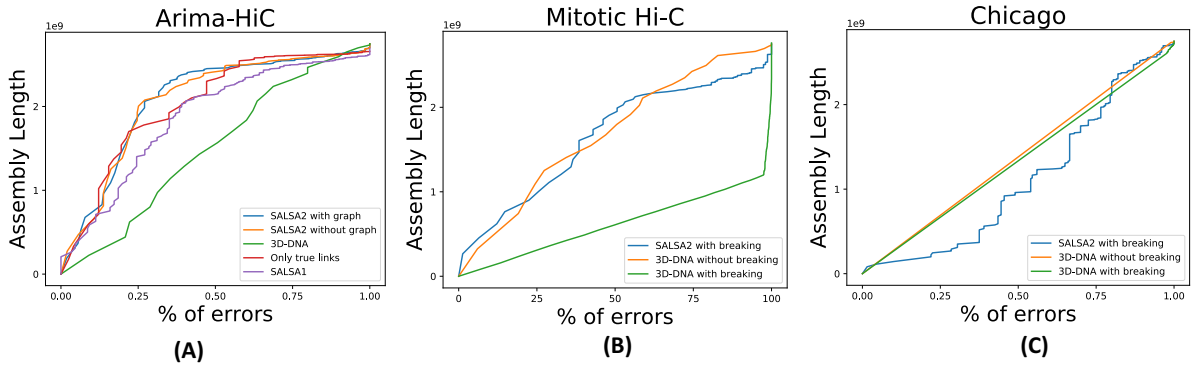


Fig. 6. Feature Response Curve for (A) assemblies obtained from unitigs as input (B) assemblies obtained from mitotic Hi-C data and (C) assemblies obtained using Dovetail Chicago data. The best assemblies lie near the top left of the plot, with the largest area under the curve. The FRC for 3D-DNA scaffolds with Chicago input is a straight line because 3D-DNA generated a single 2.7 Gbp super-scaffold which contained the majority of the genome sequence.

Dataset	Method	NG50(Mbp)	NGA50(Mbp)	Longest Chunk (Mbp)	Orientation Errors	Ordering Errors	Chimeric Errors
Arima-HiC	SALSA2 true links	83.31	79.48	172.19	78	101	0
	SALSA2 w graph	125.34	57.20	165.11	156	289	142
	SALSA2 wo graph	101.96	56.84	155.68	168	302	152
	3D-DNA	137.88	28.61	130.88	233	405	178
	SALSA1	19.09	14.81	73.14	99	176	96
Mitotic Hi-C	SALSA2 w graph	69.23	26.46	145.53	117	98	58
	3D-DNA w correction	16.34	0.064	0.96	12017	11687	7217
	3D-DNA wo correction	141.18	21.47	84.00	345	320	163
Chicago	SALSA2 w graph	6.15	4.63	34.60	59	72	128
	3D-DNA w correction	2,641.31	2.62	12.76	244	186	1550
	3D-DNA wo correction	1,648.92	4.52	34.60	119	100	711

Table 1. Assembly scaffold and correctness statistics for NA12878 assemblies scaffolded with different Hi-C libraries. The NG50 of human reference GRCh38 is 145 Mbp. The ratio between NG50 and NGA50 represents how many erroneous joins affect large scaffolds in the assembly. A high ratio between NGA50 and NG50 indicates a more accurate assembly. We observe that 3D-DNA mis-assembly detections shears the input with both the mitotic Hi-C and Chicago data so we include results both with and without this assembly correction. In case of Chicago data, 3D-DNA generates a large super-scaffold containing more than 50% of the genome, giving a very high NG50 but a poor NGA50 and ratio.

all error types and input sizes. The number of correctly oriented unitigs increased significantly when assembly graph information was integrated with the scaffolding, particularly for lower input unitig sizes (Figure 4). For example, at 400 kbp, the orientation errors with the graph were comparable to the orientation errors of the graph-less approach at 900 kbp. The NGA50 for SALSA2 also increased when assembly graph information was included (Figure 5). This highlights the power of the assembly graph to improve scaffolding and correct errors, especially on lower contiguity assemblies. This also indicates that generating a conservative assembly, rather than maximizing contiguity, can be preferable for input to Hi-C scaffolding.

3.4 Evaluation on NA12878

Table 1 lists the metrics for NA12878 scaffolds. We include an idealized scenario, using only reference-filtered Hi-C edges for comparison. As expected, the scaffolds generated using only true links had the highest NGA50 value and longest error-free scaffold block. SALSA2 scaffolds were more accurate and contiguous than the scaffolds generated by SALSA1 and 3D-DNA, even without use of the assembly graph. The addition of the graph further improved the NGA50 and longest error-free scaffold length.

We also evaluated the assemblies using Feature Response Curves (FRC) based on scaffolding errors (Vezzi *et al.*, 2012). An assembly can have a high raw error count but still be of high quality if the errors are restricted to only short scaffolds. FRC captures this by showing how quickly error is accumulated, starting from the largest scaffolds. Figure 6(A) shows the FRC for different assemblies, where the X-axis

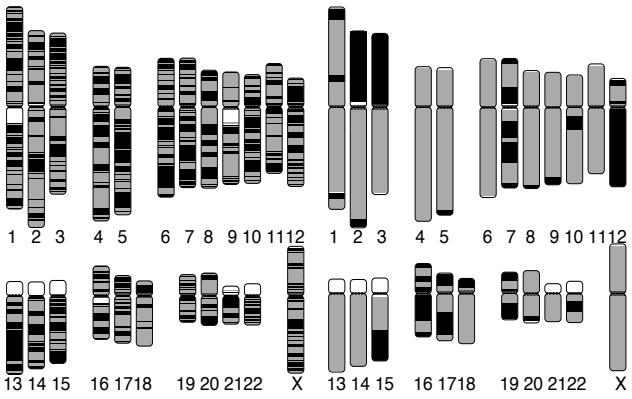


Fig. 7. Chromosome ideogram generated using the coloredChromosomes (Böhringer et al., 2002) package. Each color switch denotes a change in the aligned sequence, either due to large structural error or the end of a unitig/scaffold. Left: input unitigs aligned to the GRCh38 reference genome. Right: SALSA2 scaffolds aligned to the GRCh38 reference genome. More than ten chromosomes are in a single scaffold. Chromosomes 1 and 7 are more fragmented due to scaffolding errors which break the alignment.

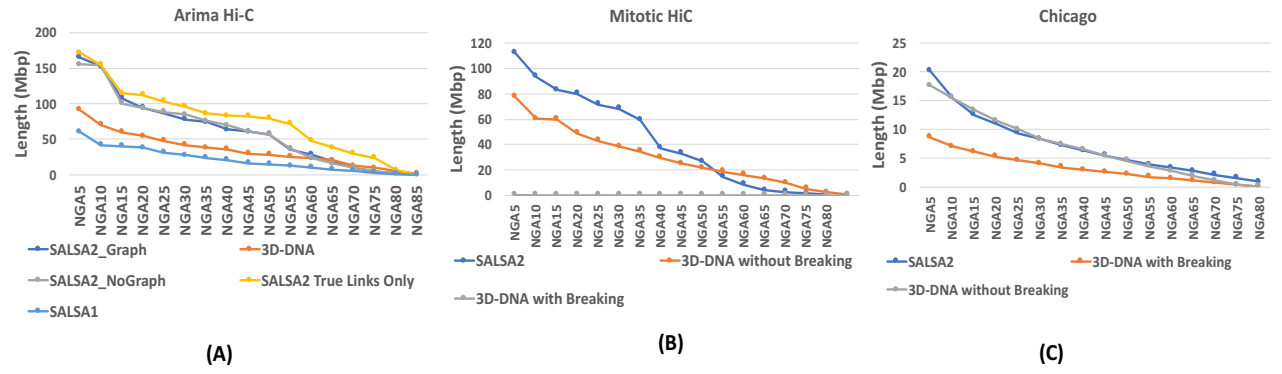


Fig. 8. Contiguity plot for scaffolds generated with (A) standard Arima-HiC data (B) mitotic Hi-C data and (C) Chicago data. The X-axis denotes the NGAX statistic and the Y-axis denotes the corrected block length to reach the NGAX value. SALSA2 results were generated using the assembly graph, unless otherwise noted.

denotes the cumulative % of assembly errors and the Y-axis denotes the cumulative assembly size. The assemblies with more area under the curve accumulate fewer errors in larger scaffolds and hence are more accurate. SALSA2 scaffolds with and without the graph have similar areas under the curve and closely match the curve of the assembly using only true links. The 3D-DNA scaffolds have the lowest area under the curve, implying that most errors in the assembly occur in the long scaffolds. This is confirmed by the lower NGA50 value for the 3D-DNA assembly (Table 1).

Apart from the correctness, SALSA2 scaffolds were highly contiguous and reached an NG50 of 125 Mbp (cf. GRCh38 NG50 of 145 Mbp). Figure 7 shows the alignment ideogram for the input unitigs as well as the SALSA2 assembly. Every color change indicates an alignment break, either due to error or due to the end of a sequence. The input unitigs are fragmented with multiple unitigs aligning to the same chromosome, while the SALSA2 scaffolds are highly contiguous and span entire chromosomes in many cases. Figure 8(A) shows the contiguity plot with corrected NG stats. As expected, the assembly generated with only true links has the highest values for all NGA stats. The curve for SALSA2 assemblies with and without the assembly graph closely matches this curve, implying that the scaffolds generated with SALSA2 are approaching the optimal assembly of this Arima-HiC data.

3.5 Robustness to input library

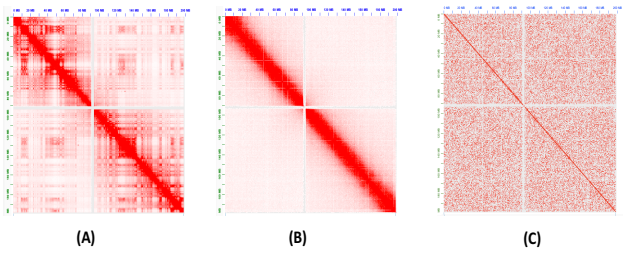


Fig. 9. Contact map of Hi-C interactions on Chromosome 3 generated by the Juicebox software (Durand et al., 2016a). The cells sequenced in (A) normal conditions, (B) during mitosis, and (C) Dovetail Chicago

We next tested scaffolding using two libraries with different Hi-C contact patterns. The first, from (Naumova *et al.*, 2013), is sequenced during mitosis. This removes the topological domains and generates fewer off-diagonal interactions. The second, the L1 library from (Putnam *et al.*, 2016), is an *in vitro* chromatin sequencing library (Chicago) generated by Dovetail Genomics. It also removes off-diagonal matches but has shorter-range interactions, limited by the size of the input molecules. As seen from the contact map in Figure 9, both the mitotic Hi-C and Chicago libraries follow different interaction distributions than the standard Hi-C (Arima-HiC in this case). We ran SALSA2 with defaults and 3D-DNA with both the assembly correction turned on and off.

For mitotic Hi-C data, we observed that the 3D-DNA mis-assembly correction algorithm sheared the input assembly into small pieces, which resulted in more than 12,000 errors and more than half of the unitigs incorrectly oriented or ordered. Without mis-assembly correction, the 3D-DNA assembly has a higher number of orientation (345 vs. 117) and ordering (320 vs. 98) errors compared to SALSA2. The feature response curve for the 3D-DNA assembly with breaking is almost a diagonal (Figure 6(B)) because the sheared unitigs appeared to be randomly joined. SALSA2 scaffolds contain longer stretches of correct scaffolds compared to 3D-DNA with and without mis-assembly correction (Figure 8(B)).

For the Chicago libraries, 3D-DNA mis-assembly detection once again sheared the input unitigs. It generated a single 2.7 Gbp scaffold and was unable to split it into the requested number of chromosomes. 3D-DNA uses signatures of chromosome ends (Dudchenko *et al.*, 2017) to identify break positions which are not present in Chicago data. As a result, it generated more chimeric joins compared to SALSA2 (1,550 vs. 128 errors). However, the number of order and orientation errors was similar across the methods. Even in the large single scaffold generated by 3D-DNA, the sizes of the correctly oriented and ordered blocks were smaller than SALSA2 (Figure 8(C)). Since Chicago libraries do not provide chromosome-spanning contact information for scaffolding, the NG50 value for SALSA is 6.15 Mbp, comparable to the equivalent coverage assembly (50% L1+L2) in (Putnam *et al.*, 2016) but much smaller than Hi-C libraries. SALSA2 is robust to changing contact distributions. In the case of Chicago data it produced a less contiguous assembly due to the shorter interaction distance. However, it avoids introducing false joins, unlike 3D-DNA, which appears tuned for a specific contact model.

4 Conclusion

In this work, we present the first Hi-C scaffolding method that integrates an assembly graph to produce high-accuracy, chromosome-scale assemblies. Our experiments on both simulated and real sequencing data for the human genome demonstrate the benefits of using an assembly graph to guide scaffolding. We also show that SALSA2 outperforms alternative Hi-C

scaffolding tools on assemblies of varied contiguity, using multiple Hi-C library preparations.

Hi-C scaffolding has been historically prone to inversion errors when the input assembly is highly fragmented. The integration of the assembly graph with the scaffolding process can overcome this limitation. Existing Hi-C scaffolding methods also require an estimate for the number of chromosomes in the genome. Since SALSA2's mis-join correction algorithm stops scaffolding after the useful linking information in a dataset is exhausted, no chromosome count is needed as input. As the Genome10K consortium (Koepfli *et al.*, 2015) and independent scientists begin to sequence novel lineages in the tree of life, it may be impractical to generate physical or genetics maps for every organism. Thus, Hi-C sequencing combined with SALSA2 presents an economical alternative for the reconstruction of chromosome-scale assemblies.

Acknowledgements

AS and SS were funded by generous support from NHGRI (grant# 1R44HG009584). JG and MP were supported by NIH grant R01-AI-100947 to MP. SK, AR, BPW, and AMP were supported by the Intramural Research Program of the National Human Genome Research Institute, National Institutes of Health. AR was also supported by a grant from the Korean Visiting Scientist Training Award (KVSTA) through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number: HI17C2098). This work utilized the computational resources of the NIH HPC Biowulf cluster (http://hpc.nih.gov).

References

Bickhart, D. M. *et al.* (2017). Single-molecule sequencing and chromatin conformation capture enable de novo reference assembly of the domestic goat genome. *Nature Genetics*, **49**(4), 643–650.

Böhringer, S. *et al.* (2002). A software package for drawing ideograms automatically. *Online J Bioinformatics*, **1**, 51–61.

Burton, J. N. *et al.* (2013). Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions. *Nature biotechnology*, **31**(12), 1119–1125.

Chin, C.-S. *et al.* (2016). Phased diploid genome assembly with single molecule real-time sequencing. *bioRxiv*.

Dixon, J. R. *et al.* (2012). Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, **485**(7398), 376–380.

Dong, Y. *et al.* (2013). Sequencing and automated whole-genome optical mapping of the genome of a domestic goat (capra hircus). *Nature biotechnology*, **31**(2), 135–141.

Dudchenko, O. *et al.* (2017). De novo assembly of the aedes aegypti genome using hi-c yields chromosome-length scaffolds. *Science*, **356**(6333), 92–95.

Durand, N. C. *et al.* (2016a). Juicebox provides a visualization system for hi-c contact maps with unlimited zoom. *Cell systems*, **3**(1), 99–101.

Durand, N. C. *et al.* (2016b). Juicer provides a one-click system for analyzing loop-resolution hi-c experiments. *Cell systems*, **3**(1), 95–98.

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of mathematics*, **17**(3), 449–467.

Eid, J. *et al.* (2009). Real-time dna sequencing from single polymerase molecules. *Science*, **323**(5910), 133–138.

Ghurye, J. *et al.* (2017). Scaffolding of long read assemblies using long range contact information. *BMC genomics*, **18**(1), 527.

Gnerre, S. *et al.* (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences*, **108**(4), 1513–1518.

Jain, M. *et al.* (2016). The oxford nanopore minion: delivery of nanopore sequencing to the genomics community. *Genome biology*, **17**(1), 239.

Jain, M. *et al.* (2017). Nanopore sequencing and assembly of a human genome with ultra-long reads. *bioRxiv*, page 128835.

Kaplan, N. and Dekker, J. (2013). High-throughput genome scaffolding from in vivo dna interaction frequency. *Nature biotechnology*, **31**(12), 1143–1147.

Koepfli, K.-P. *et al.* (2015). The genome 10k project: a way forward. *Annu. Rev. Anim. Biosci.*, **3**(1), 57–111.

Kolmogorov, M. *et al.* (2018). Assembly of long error-prone reads using repeat graphs. *bioRxiv*.

Koren, S. *et al.* (2017). Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome research*, **27**(5), 722–736.

Kurtz, S. *et al.* (2004). Versatile and open software for comparing large genomes. *Genome biology*, **5**(2), R12.

Li, H. (2016). Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, **32**(14), 2103–2110.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, **25**(14), 1754–1760.

Lieberman-Aiden, E. *et al.* (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *science*, **326**(5950), 289–293.

Marie-Nelly, H. *et al.* (2014). High-quality genome (re) assembly using chromosomal contact data. *Nature communications*, **5**.

Miller, J. R. *et al.* (2010). Assembly algorithms for next-generation sequencing data. *Genomics*, **95**(6), 315–327.

Myers, E. W. (2005). The fragment assembly string graph. *Bioinformatics*, **21**(suppl 2), ii79–ii85.

Nagarajan, N. and Pop, M. (2009). Parametric complexity of sequence assembly: theory and applications to next generation sequencing. *Journal of computational biology*, **16**(7), 897–908.

Nagarajan, N. and Pop, M. (2013). Sequence assembly demystified. *Nature Reviews Genetics*, **14**(3), 157–167.

Naumova, N. *et al.* (2013). Organization of the mitotic chromosome. *Science*, **342**(6161), 948–953.

Pendleton, M. *et al.* (2015). Assembly and diploid architecture of an individual human genome via single-molecule technologies. *Nature methods*.

Pevzner, P. A. *et al.* (2001). An eulerian path approach to dna fragment assembly. *Proceedings of the National Academy of Sciences*, **98**(17), 9748–9753.

Poloczek, M. and Szegedy, M. (2012). Randomized greedy algorithms for the maximum matching problem with new analysis. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 708–717. IEEE.

Putnam, N. H. *et al.* (2016). Chromosome-scale shotgun assembly using an in vitro method for long-range linkage. *Genome research*, **26**(3), 342–350.

Salzberg, S. L. *et al.* (2012). Gage: A critical evaluation of genome assemblies and assembly algorithms. *Genome research*, **22**(3), 557–567.

Schneider, V. A. *et al.* (2017). Evaluation of grch38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome research*, **27**(5), 849–864.

Schwartz, D. C. *et al.* (1993). Ordered restriction maps of saccharomyces cerevisiae chromosomes constructed by optical mapping. *Science*, **262**(5130), 110–114.

Shelton, J. M. *et al.* (2015). Tools and pipelines for bionano data: molecule assembly pipeline and fasta super scaffolding tool. *BMC genomics*, **16**(1), 734.

Simonis, M. *et al.* (2006). Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture–on-chip (4c). *Nature genetics*, **38**(11), 1348–1354.

Venter, J. C. *et al.* (1996). A new strategy for genome sequencing. *Nature*, **381**(6581), 364.

Vezzi, F. *et al.* (2012). Feature-by-feature–evaluating de novo sequence assembly. *PloS one*, **7**(2), e31002.

Weisenfeld, N. I. *et al.* (2017). Direct determination of diploid genome sequences. *Genome research*, **27**(5), 757–767.

Wysoker, A. *et al.* (2013). Picard tools version 1.90.

Yeo, S. *et al.* (2017). Arcs: Scaffolding genome drafts with linked reads. *Bioinformatics*.

Zheng, G. X. *et al.* (2016). Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nature biotechnology*.

CliqueSNV: Scalable Reconstruction of Intra-Host Viral Populations from NGS Reads

S. Knyazev[1,2], V. Tsyvina[1], A. Melnyk[1], A. Artyomenko[3], T. Malygina[4], Y. B. Porozov[4,5], E. Campbell[2], W. M. Switzer[2], P. Skums[1,2], and A. Zelikovsky[1,5]

[1]Georgia State University, Atlanta, GA, USA, [2]Centers for Disease Control and Prevention, Atlanta, GA, USA, [3]Guardant Health Inc., Redwood City, CA, USA [4]ITMO University, St. Petersburg, Russia, [5]I.M. Sechenov First Moscow State Medical University, Moscow, Russia

1 Background

Highly mutable RNA viruses such as influenza A virus, human immunodeficiency virus and hepatitis C virus exist in infected hosts as highly heterogeneous populations of closely related genomic variants. The presence of low-frequency variants with few mutations with respect to major strains may result in an immune escape, emergence of drug resistance, and an increase of virulence and infectivity. Next-generation sequencing technologies permit detection of sample intra-host viral population at extremely great depth, thus providing an opportunity to access low-frequency variants. Long read lengths offered by single-molecule sequencing technologies allow all viral variants to be sequenced in a single pass. However, high sequencing error rates limit the ability to study heterogeneous viral populations composed of rare, closely related variants.

In this article, we present CliqueSNV, a novel reference-based method for reconstruction of viral variants from NGS data. It efficiently constructs an allele graph based on linkage between single nucleotide variations and identifies true viral variants by merging cliques of that graph using combinatorial optimization techniques. The full paper text is available at `https://www.biorxiv.org/content/early/2018/03/31/264242`

2 Results

CliqueSNV is designed to accurately reconstructintra-host viral variants from noisy next-generation and third-generation sequencing data. A novel method eliminates the need for preliminary error correction and assembly and infers haplotypes from patterns in distributions of SNVs in sequencing reads. It is applicable to both long single-molecule reads (e.g., PacBio) as well as high volume short paired reads (e.g., Illumina). CliqueSNV uses linkage between single nucleotide variations (SNVs) to accurately and effciently distinguish them from sequencing errors. It constructs an allele graph with edges connecting linked SNVs and finds all cliques as well as unlikely linked SNV pairs referred as forbidden SNV pairs. CliqueSNV reports viral variants corresponding to maximal connected subsets of cliques without forbidden SNV pairs.

Validation of different haplotype reconstruction methods should report similarity between reconstructed and true variants by simultaneously taking into account sequences and frequencies. We propose to use the Earth Mover's Distance (EMD) [1] as a distance measure between populations, which generalizes edit distance between genomes of individual variants taking into account theyr frequencies.

We have compared four haplotyping tools CliqueSNV, 2SNV [2], PredictHaplo [3], and aBayesQR [4] on 4 benchmarks. The first benchmark IAV_PacBio represent PacBio sequencing data from 10 similar Iinfluenza A virus (IAV) [2] and the second benchmark Reduced_labmix represent MiSeq sequencing data from 5 different HIV subtypes [5]. The remaining 2 benchmarks IAV_MiSeq and HIV_MiSeq represent simulated MiSeq reads from 10 similar IAV variants and 7 HIV variants from the same subtype accordingly.

The results of comparison are presented in Table 1. CliqueSNV outperforms all other methods on all benchmarks. For IAV_PacBio and HIV_MiSeq, it reconstruct all haplotypes without mismatches. For

Table 1: Comparison of four haplotype reconstruction methods on simulated and real datasets

Dataset	EMD	# variants	CliqueSNV		2SNV		PredictHaplo		aBayesQR	
			TP	EMD	TP	EMD	TP	EMD	TP	EMD
IAV_Pacbio	4.22	10	**10**	**0.22**	9	0.23	7	0.38	-	-
Reduced_labmix	19.4	5	**3**	**6.52**	-	-	**3**	6.8	0	19.2
IAV_MiSeq	4.22	10	**7**	**0.0939**	-	-	1	3.03	0	3.64
HIV_MiSeq	11	7	**7**	**0.018**	-	-	0	5.84	3	0.84

TP = the number of variants predicted without errors

Reduced_labmix and IAV_MiSeq it reconstruct 3 and 7 haplotypes with a single mismatch, and accurately identifies all haplotypes for the HIV_Sim_MiSeq dataset. CliqueSNV is significantly faster than the other tools in our study. For example, the Reduced_labmix benchmark the runtimes of aBayesQR and SAVAGE were more than 10h, PhedictHaplo's runtime was 24 min, and CliqueSNV took 79 seconds.

3 Conclusions

We developed CliqueSNV, a new method for inference of rare genetically-related viral variants, which allows for accurate haplotyping in the presence of high sequencing error rates and which is also suitable for both single-molecule and short-read sequencing. CliqueSNV infers viral haplotypes by detection of clusters of statistically linked SNVs rather than through assembly of overlapping reads. Using experimental data, we demonstrate that CliqueSNV can detect haplotypes with frequencies as low as 0.1%, which is comparable to the sensitivity of many deep sequencing-based point mutation detection methods [6, 7]. Furthermore, CliqueSNV can successfully infer viral variants, which differ by only a few mutations, thus demonstrating the high sensitivity of identifying closely related variants. Another significant advantage of CliqueSNV is its low computation time, which is achieved by fast searching of linked pairs of SNVs and the application of the special graph-theoretical approach to SNV clustering.

The ability to accurately infer the structure of intra-host viral populations makes CliqueSNV applicable for studying evolution and examining genomic compositions in RNA viruses. However, we envision that the application of our method can be extended to other highly heterogeneous genomic populations, such as metagenomes, immune repertoires, and cancer cells. The open source implementation of CliqueSNV is freely available for download at `https://github.com/vyacheslav-tsivina/CliqueSNV`

References

[1] Peleg, S., Werman, M., Rom, H.: A unified approach to the change of resolution: Space and gray-level. IEEE Transactions on Pattern Analysis and Machine Intelligence **11**(7), 739–742 (1989)

[2] Artyomenko, A., Wu, N.C., Mangul, S., Eskin, E., Sun, R., Zelikovsky, A.: Long single-molecule reads can resolve the complexity of the influenza virus composed of rare, closely related mutant variants. In: International Conference on Research in Computational Molecular Biology, pp. 164–175 (2016). Springer International Publishing

[3] Prabhakaran, S., Rey, M., Zagordi, O., Beerenwinkel, N., Roth, V.: HIV haplotype inference using a propagating Dirichlet process mixture model. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **11**(1), 182–191 (2014)

[4] Ahn, S., Vikalo, H.: abayesqr: A bayesian method for reconstruction of viral populations characterized by low diversity. In: International Conference on Research in Computational Molecular Biology, pp. 353–369 (2017). Springer

[5] Giallonardo, F.D., Töpfer, A., Rey, M., Prabhakaran, S., Duport, Y., Leemann, C., Schmutz, S., Campbell, N.K., Joos, B., Lecca, M.R., Patrignani, A., Däumer, M., Beisel, C., Rusert, P., Trkola, A., Günthard, H.F., Roth, V., Beerenwinkel, N., Metzner, K.J.: Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations. Nucleic Acids Research **42**(14), 115 (2014)

[6] Flaherty, P., Natsoulis, G., Muralidharan, O., Winters, M., Buenrostro, J., Bell, J., Brown, S., Holodniy, M., Zhang, N., Ji, H.P.: Ultrasensitive detection of rare mutations using next-generation targeted resequencing. Nucleic Acids Res. **40**(1), 2 (2012)

[7] Harismendy, O., Schwab, R.B., Bao, L., Olson, J., Rozenzhak, S., Kotsopoulos, S.K., Pond, S., Crain, B., Chee, M.S., Messer, K., Link, D.R., Frazer, K.A.: Detection of low prevalence somatic mutations in solid tumors with ultra-deep targeted sequencing. Genome Biol. **12**(12), 124 (2011)

Graph-guided assembly for novel HLA allele discovery

Accurate typing of human leukocyte antigen (HLA), a histocompatibility test, is important because HLA genes are crucial to the regulation of immune system. Also, they play various roles in transplant rejection as well as infectious and autoimmune diseases. The current gold standard for HLA typing uses DNA sequencing technology combined with sequence enrichment techniques using specially designed primers or probes, requiring additional experiments. Although there exist enrichment-free computational methods that use various types of sequencing data, hyper-polymorphism found in the HLA region makes it challenging to type HLA genes with high accuracy from whole genome sequencing (WGS) data. Furthermore, WGS-based methods developed up to this point are *database-matching* approaches where their output is inherently limited by the incompleteness of already known types, forcing them to find the best matching known alleles from a database, thereby causing them to be unsuitable for discovery of rare or novel alleles.

In order to ensure both high accuracy as well as the ability to recover novel alleles (HLA gene sequences), we developed a graph-guided HLA assembler called `Kourami`, which is capable of assembling phased, full-length haplotype sequences of typing exons given high-coverage ($>$ 30-fold) WGS data (an overall workflow of `Kourami` is shown in Figure 1). `Kourami` first uses partial order graphs to compactly capture variant regions among related sequences to fully take advantage of known alleles. Then read alignments are projected onto the graphs so that each read alignment is stored as a path in the graphs and read depths on edges naturally become edge weights. During this step, the graphs are modified by adding nodes and edges to incorporate differences found by alignment such as substitutions and indels. Finally, with the weighted graphs with alignment paths, we formulate the problem of constructing the best pair (diploid, therefore 2 alleles per loci) of HLA allele sequences as finding the pair of paths through the graph, which explains the read mapping data best. When finding the pair, we select the pair that maximizes the adjusted coverage (with a use of base quality scores) and the consistency of phasing information.

`Kourami` can type with high accuracy (>98%), comparable to that of the gold-standard typing assays, when tested across various WGS datasets such as simulation, Illumina Platinum Genomes and 1000 Genomes. At the same time, `Kourami` only takes a fraction of time compared to other available methods with a moderate use of memory. Additionally, `Kourami` is the first method that directly assembles both haplotypes of HLA genes, capable of discovering novel alleles rather than inferring the best matching alleles in the database.

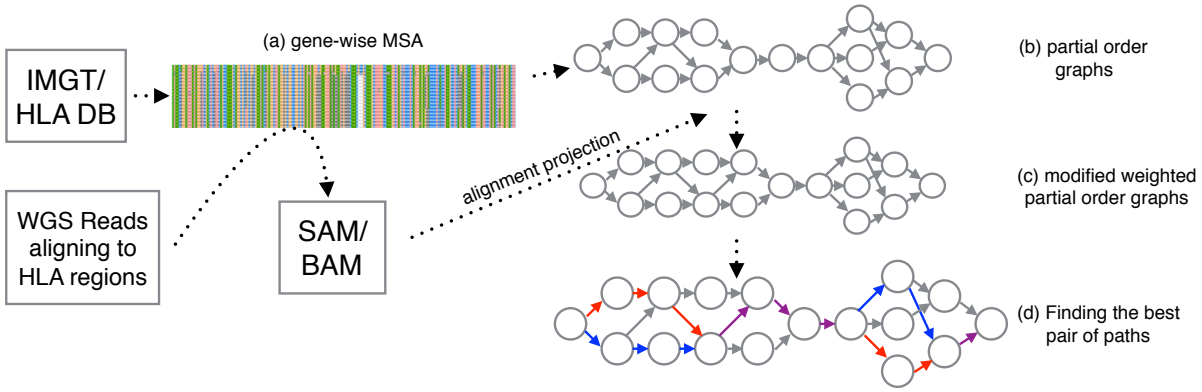


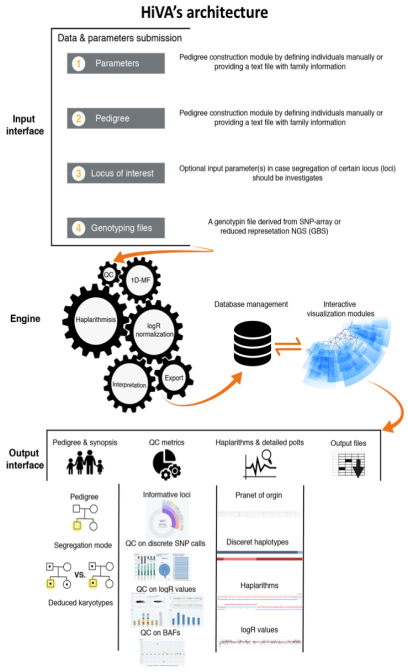
Figure 1: **An overall workflow of** `Kourami`. The haplotype assembly of two solution alleles is obtained by finding two paths (shown in (d) – drawn in red and blue; overlap in purple) through the graph.

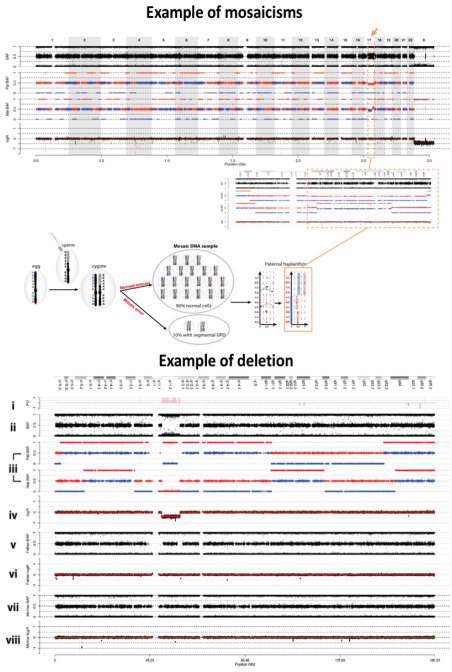
HiVA: a web platform for haplotyping and copy number analysis of single-cell genomes and mosaicism detection in bulk DNA

Single-cell sequencing offers numerous research and clinical opportunities in reproductive genomics, cancer, etc. Genetic mosaicism can affect diverse organs at different developmental stages during the course of a human life-time, such that it may cause miscarriages, birth defects, developmental disorders, cancer or simply contribute to the normal spectrum of phenotypic variation. Genetic variation arises through a diversity of mechanisms. Identifying the nature and origin of acquired numerical and structural chromosome aberrations in healthy or diseased tissues via single-cell or bulk DNA analyses are imperative for understanding mutational processes and elucidating their impact on phenotypes and diseases. This is more challenging when complex genomes should be characterized, including single-cell genomes or mosaic/chimeric genome. Given all of these, analyzing data towards the interpretation need a well-designed algorithm. We named our algorithm Haplarithmisis. While the Haplarithmisis algorithm is promising in both research and clinics, needs of an interactive visualization is crucial. Therefore, we chose to develop a user-friendly web tool with a rich user interface to let users analyze their data easily without being worried about the infrastructure and expertise needed to run the analysis.

We prototyped HiVA (Haplarithm inference of Variant Alleles), an interactive web application that determines genome-wide haplotypes, the copy number of those haplotypes, the level of genetic mosaicism/chimaerism, and the parental and segregational origin of haplotype aberrations in DNA samples derived from a large number of cells down to a single cell. This method can be carried out using data from SNP arrays or from single-cell sequencing. It provides a novel approach for reduced-representation genome sequencing of single cells and bulk DNA, as well as novel insight into genomic composition that are missed by conventional bulk analysis methods.

Users can submit their data analysis request to the system after registration. Input parameters are divided into four categories: (1) analysis parameters, (2) family structure, (3) list of chromosome loci on which the user wants to focus, (4) and genotype file. As soon as the results gets ready HiVA integrates data into a visualization tools which allow users to interact with the results. HiVA result's explorer is an interactive visualization to observe the results of Haplarithmisis. The visual output illustrates haplotype blocks, the paternal and maternal origin, relative copy number (logR values), SNP BAFs, and haplarithms across the entire genome for a single or multiple samples simultaneously. Additionally, HiVA provide several quality control measurements that help the users to discover the underlying basis of the analysis.





We are showing that HiVA enables concurrent haplotyping and copy-number profiling of single cells. Haplarithmisis decodes the number of haplotypes for genomic regions across the genome. In contrast to conventional family-based haplotyping methods that make use of discrete bi-allelic SNP genotypes (AA, AB and BB) to reconstruct haplotypes, haplarithmisis uses continuous SNP genotypes values (i.e., SNP B allele fractions (BAFs)), which potentially harbor quantitative (haplotype) and qualitative (copy number) assessment of genomes. Haplarithmisis enables blueprinting these information into parental haplarithms (i.e., paternal haplarithm and maternal haplarithm). Parental haplarithms harbor haplotype and copy number state of genomic regions and reveal the segregational origin of aberrations, such that an aberration can be traced back to meiosis I, meiosis II or mitosis.