# ARCS: Assembly Roundup by Chromium Scaffolding

Sarah Yeo[1], Lauren Coombe[1], Justin Chu, René L Warren[1,2], and Inanç Birol

BC Cancer Agency, Genome Sciences Centre, Vancouver, BC V5Z 4S6, Canada
[1]Authors contributed equally
[2]corresponding author: rwarren@bcgsc.ca

**Abstract.** Sequencing of human genomes is now routine, and assembly of shotgun reads is increasingly feasible. However, assemblies often fail to inform about chromosome-scale structure due to lack of linkage information over long stretches of DNA – a shortcoming that is being addressed by new sequencing protocols, such as linked reads from 10X Genomics. Here we present ARCS, an application that utilizes the barcoding information contained in linked reads to further organize draft genomes into highly contiguous assemblies. We show how the contiguity of an ABySS *H. sapiens* genome assembly can be increased over six-fold using moderate coverage (25-fold) Chromium data. We expect ARCS to have broad utility in harnessing the barcoding information contained in Chromium data for connecting high-quality sequences in genome assembly drafts. Availability: http://www.bcgsc.ca/platform/bioinfo/software/arcs

**Keywords:** 10X Genomics Chromium, linked reads, ARCS, LINKS, ABySS, next-generation sequencing, *de novo* assembly, genome scaffolding, read clouds

**Supplementary information** available online.

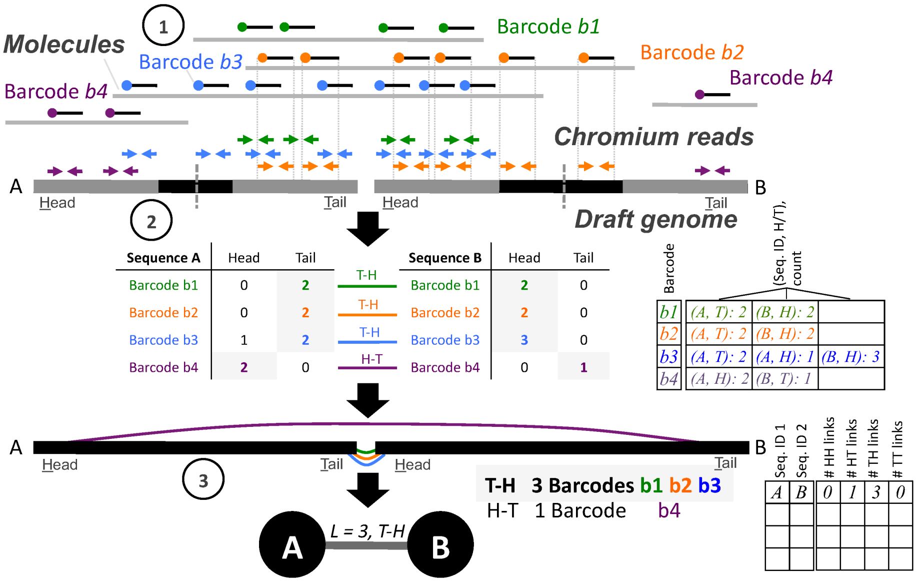
## 1 Introduction

The Chromium sequencing library preparation protocol from 10X Genomics (10XG, Pleasanton, CA) builds on the Illumina sequencing technology (San Diego, CA) to provide indexing/barcoding information along with short reads to localize the latter on long DNA fragments, thus benefiting the economies of scale of a high-throughput platform. As sequence reads from 20 to 200 kb molecules are barcoded/linked, applications of the technology has mainly focused on phasing variant bases in human genomes ([1] and [2]). The ability to generate linked reads with 10XG is akin to that of Illumina TruSeq [3]. The latter technology provides useful complementary information to whole genome shotgun assembly projects, as the pseudo-long reads it generates may help resolve long repeats. However, to generate pseudo-long reads, TruSeq requires high coverage data of the co-localized reads for a priori fragment assemblies (by default, transparent to the user), essentially generating low fragment coverage data for its target genome. Hence, TruSeq may be prohibitively expensive for providing mammalian-sized genomes with adequate fragment coverage. Conversely, the Chromium platform typically provides low-coverage for each single barcoded molecule, limiting its utility for individual fragment assembly. Though, it makes up for this limitation in throughput, providing higher fragment coverage.

Recently this data type has been utilized for scaffolding a draft genome assembly [4], using a software designed to scaffold sequences using contiguity preserving transposition sequencing (CTP-seq) and Hi-C data, another long-range information data source [5]. In their paper, Mostovoy and colleagues [4] showed 12-fold improvement in contiguity of a human genome assembly draft using GemCode sequencing (precursor to Chromium, from 10XG) at 97-fold coverage, demonstrating the potential of the technology for scaffolding draft genomes. Here we present ARCS, the Assembly Round-up by Chromium Scaffolding algorithm, a method that leverages the rich information content of high-volume long sequencing fragments to further organize draft genome sequences into contiguous assemblies that characterize large chromosome segments. We use the recent Genome In A Bottle (GIAB) human genome sequence data [6], and compare ARCS to fragScaff, the only other technology shown in a publication to utilize 10XG linked reads for scaffolding genome assembly drafts [4]. We also present similar benchmarks to Architect, a recently published scaffolder [7] shown to work on Illumina TruSeq synthetic long sequences' underlying short reads (read clouds) and suggested to be adaptable to Chromium data. We show how our implementation yields assemblies that are more contiguous and accurate than fragScaff and Architect over a wide range of parameters while using less time and compute resources. We note that ARCS scaffolding of pre-existing human genome drafts using two different linked reads datasets yields assemblies whose contiguity and correctness is on par with or better than those assembled with newly released 10XG Supernova *de novo* assembler [8].

## 2 Methods

We used two human Chromium read datasets for scaffolding, including one from individual NA12878 and GIAB HG004 NA24143. For the genome assembly contig and scaffold baselines, we first downloaded Illumina whole genome shotgun (WGS) 2x250 bp paired-end sequencing data and Illumina 6 kbp mate-pair data for an Ashkenazi female (NA24143). ABySS-2.0 [9] was used to assemble the whole-genome shotgun paired-end and mate-pair reads into contigs and scaffolds and Chromium reads were aligned to those sequences. Using these alignments as input, we ran and benchmarked ARCS (v1.0.0), Architect (v0.1) and fragScaff (v140324) to further scaffold contig and scaffold baseline sequences 3 kbp and longer, as recommended [4], while investigating the effects of multiple parameter combinations on scaffolding (Table S1), reporting contiguity length metrics and breakpoints from sequence alignments to the reference human genome, which serves as a proxy for counting large-scale misassemblies in both contigs and scaffolds.



| Sequence A | Head | Tail | | Sequence B | Head | Tail |
|---|---|---|---|---|---|---|
| Barcode b1 | 0 | 2 | T-H | Barcode b1 | 2 | 0 |
| Barcode b2 | 0 | 2 | T-H | Barcode b2 | 2 | 0 |
| Barcode b3 | 1 | 2 | T-H | Barcode b3 | 3 | 0 |
| Barcode b4 | 2 | 0 | H-T | Barcode b4 | 0 | 1 |

| Barcode | (Seq. ID, H/T), count | | |
|---|---|---|---|
| *b1* | *(A, T): 2* | *(B, H): 2* | |
| *b2* | *(A, T): 2* | *(B, H): 2* | |
| *b3* | *(A, T): 2* | *(A, H): 1* | *(B, H): 3* |
| *b4* | *(A, H): 2* | *(B, T): 1* | |

T-H 3 Barcodes b1 b2 b3
H-T 1 Barcode b4

| Seq. ID 1 | Seq. ID 2 | # HH links | # HT links | # TH links | # TT links |
|---|---|---|---|---|---|
| *A* | *B* | *0* | *1* | *3* | *0* |
| | | | | | |
| | | | | | |
| | | | | | |

**Fig. 1.** ARCS algorithm. 1) 10XG Chromium reads (blue, green, orange, and purple arrows) are aligned to the draft genome. 2) Sequences are split in half by length and the ends of each are considered the head (H) or tail (T) regions (represented with grey boxes, length of the ends controlled by ARCS parameter -*e*). The number of read pairs derived from the same barcode and aligning to the head (H) or tail (T) regions of the sequence are tallied. These tallies are stored in memory using a map data structure, where the key is the barcode sequence. The value maps a tuple of the sequence ID and H or T to the count of the number of read pairs of that barcode which map to the corresponding region of the sequence. 3) The number of barcodes supporting each link orientation (H-H, H-T, T-H, T-T) between sequence pairs is tallied. The tallies are stored in memory using a map data structure, where the key is a pair representing the two potentially linked sequences, and the value is a vector of integers representing the number of barcodes supporting each possible link orientation. For a given barcode to contribute linking evidence, the distribution of reads of that barcode aligning to the H or T regions of both sequences in the potential pair must significantly differ from a uniform distribution. A dot file is then generated which encodes the linkage evidence, where links (edges) between two sequences (nodes) are only added if the link orientation with the maximum support is predominant.

### 2.1 ARCS algorithm

The pipeline collectively referred to as ARCS first pairs sequences within a draft assembly, then lays out the pairing information for scaffolding. In the sequence pairing stage (Fig. 1), input alignments in BAM format are processed for sets of read pairs from the same barcode that align to different sequences. These form a link between the two sequences, provided that there is sufficient number of read pairs aligned (-*c*, set to 5 by default). Each link represents evidence that one barcode/molecule connects both sequences. To account for barcode sequencing errors, only barcodes within a specified multiplicity range (-*m*) are considered (default 50-10000 for Chromium). The multiplicity refers to the read frequency of each barcode and the range, a specific slice of reads that are considered by ARCS. In our experience, this distribution is wider for GemCode and should be set to 50-10000. As we are interested in ordering and orienting sequences, we consider reads that align to the 5' and 3' end (-*e*) bases of each sequence. This parameter effectively sets the maximum length of regions, at the end of sequences, where Chromium reads align. Reads whose alignments fall outside of these regions are not considered and thus, adjusting -*e* to a lower value to account for shorter contigs is important as it will have for effect to mitigate ambiguity when creating an edge between any two sequences. In addition, as the BAM file is read, only reads that align to a sequence with at least the specified sequence identity (-*s*, set to 98 by default), map in proper pairs and are unique alignments are considered. This ensures that only high-quality alignments provide evidence for the subsequent linking stages, as alignments involving reads with long repeat regions or chimeric reads will be skipped.

The relative orientations of sequences are inferred through the read alignment positions. First we determine, using read alignments, subsets of reads with the same barcode that co-locate within one end of the sequence (Fig. 1, step 1); Each sequence is split in half by length and one end region arbitrarily labeled head (H), the other tail (T). The number of read pairs of the same barcode aligning to the head or the tail of a sequence (within -*e* bp or less of the end) is tallied as the BAM file is read. A map data structure tracks, for a given barcode, the number of reads that map to the H or T of a sequence (Fig. 1, step 2). Once the alignment file is read into memory, every possible pair of sequences which have a sufficient number of aligned reads from a given barcode (-*c*) are considered. For each sequence in a potential pair, a binomial test is used to calculate whether the observed distribution of reads aligning to the 5 or 3 end of a sequence is significantly different from a uniform distribution (threshold $p$=0.05, parameter -*r*). Likewise, the number of linking barcodes/indices that support each of the four link orientations (H-H, H-T, T-H, T-T) is tallied in a map data structure for each potential sequence pair (Fig. 1, step 3). Using the link orientation tallies for each sequence pair, a graph data structure is constructed, where the nodes are sequences, and the edges represent links between them. An edge is formed only if the link orientation, defined by the order of sequence pairs head and tail regions, is the most represented combination across supporting barcodes.

After the sequence pairing stage completes, ARCS outputs a single file in the graph description language (gv) format. The gv file is converted using supplied python scripts (makeTSVfile.py), to a tab-separated value (tsv) file listing all possible oriented sequence pairs, the number of supporting barcodes with gap sizes arbitrarily set at 10 bp. Since positional information of reads within the molecule of origin is not known, estimates of gap sizes are not possible. At the layout stage, the latter tsv file is read and scaffolds constructed using the algorithm implemented in LINKS (v1.7 and later), as described [10]. Because linked sequence pairs may be ambiguous (a given sequence may link to multiple sequences), sequences are joined only if the number of links connecting a sequence pair is equal or greater than a minimum (LINKS parameter *-l*, default of 5). Ambiguous pairings are resolved when the ratio of barcode links of the second-most to top-most supported edge is equal or below a threshold (LINKS parameter *-a*, default of 0.3; we recommend higher values such as *-a* 0.7 and 0.9 when running LINKS within ARCS).

ARCS is implemented in C++ and runs on Unix.

### 2.2 Data sources

Two human Chromium datasets were downloaded, including one from individual NA12878 and GIAB HG004 NA24143 (Table S2). The Illumina WGS 2x250 bp paired-end sequencing data of an Ashkenazi mother (NA24143) was downloaded from https://github.com/genome-in-a-bottle/giab_data_indexes/blob/master/AshkenazimTrio/sequence.index.AJtrio_Illumina_2x250bps_06012016 under accession number NIST HG004 NA24143 SRS823307.

The Illumina 6 kbp mate-pair sequencing data from that individual was downloaded from https://github.com/genome-in-a-bottle/giab_data_indexes/blob/master/AshkenazimTrio/sequence.index.AJtrio_Illumina_6kb_matepair_wgs_08032015.

We downloaded the corresponding 10X Genomics 2x128 bp barcoded paired-end sequencing data for that individual [6] from https://github.com/genome-in-a-bottle/giab_data_indexes/blob/master/AshkenazimTrio/alignment.index.AJtrio_10Xgenomics_ChromiumGenome_GRCh37_GRCh38_06202016.

The 156-fold coverage NA12878 10XG Chromium data was downloaded from: http://support.10xgenomics.com/de-novo-assembly/datasets/msNA12878, which we assembled with the 10XG Supernova v1.1 *de novo* assembler (available at http://support.10xgenomics.com/de-novo-assembly/software/downloads/latest), as described [8].

### 2.3 Data analysis

Prior to assembly, the adapters from the mate-pair reads were removed using NxTrim v0.4.0 [11] (with parameters *-norc -joinreads -preserve-mp*), which also classifies reads as mate-pair, paired-end, single-end or unknown. Only reads classified as mate-pair were used for scaffolding of contigs with ABySS. Both paired-end and mate-pair reads were corrected with BFC v181 [12] (with the parameter

-s3G). Contigs and scaffolds were assembled with ABySS v2.0 [9] with the command: abyss-pe *name*=hsapiens *np*=64 *k*=144 *q*=15 *v*=-v *l*=40 *s*=1000 *n*=10 *S*=1000-10000 *N*=7 *mp6k_de*=–mean *mp6k_n*=1 *lib*=pe400 *mp*=mp6k, where pe400 and mp6k are variables listing all files containing paired-end sequencing and MPET reads, respectively.

The 10XG Chromium sequencing data was converted from a BAM file to a FASTQ format. Briefly, the read barcodes were extracted from the RX tag in the BAM file and appended to the read name following an underscore character (eg. NAME_BARCODE). Read pairs were only added to the FASTQ file if the corresponding alignment had a sequence identity of at least 90% and were aligned in proper pairs. Chromium reads were then aligned to the contigs and scaffolds using BWA mem v0.7.13 (default values, adjusting *t* for multiple threads) [13] and sorted by name.

Based on recommendations, the input to fragScaff also included an N-base bed file, generated by the provided script fasta_make_Nbase_bed.pl, and a repeat bed file generated by performing a blastn v2.4.0 alignment [14] of the input assembly to itself (with parameters *word_size* 36, -*perc_identity* 95, -*outfmt* 6), which was then processed by the provided script blast_self_alignment_filter.pl. The scripts that ran on the data described above are available at ftp://ftp.bcgsc.ca/supplementary/ARCS, along with the commands and parameters we ran and set with each tool and corresponding assemblies.

The NG50 and NGA50 length metrics reported were calculated using a genome size of 3,088,269,832 bp.

Benchmarking was done on a DELL server with 128 Intel(R) Xeon(R) CPU E7-8867 v3, 2.50GHz with 2.6TB RAM.

## 3 Results

### 3.1 Scaffolding with the NA24143 GIAB Chromium data

We measured the contiguity (NG50 and NGA50 length metric) and correctness of resulting assemblies after ARCS, Architect and fragScaff scaffolding of baseline assemblies (Table S3). During this process, we tested the effect of various parameters, including the scaffolding-specific -*a*, -*u* and *–rc-rel-edge-thr* (abbreviated rel) parameters in the corresponding tools. Generally, these parameters affect scaffolding stringency by evaluating the validity of the linkages. Mostovoy *et al.* [4] reported their best assembly using fragScaff parameters -*j*1 and -*u*3, prompting us to explore similar values of -*j* and -*u* on our dataset. These parameters are described as *the mean number of passing hits per node to call the p-value cutoff* and *modifier to the score to consider the link reciprocated* in fragScaff and are the parameters previously optimized [5]. In Architect, the parameters -*rc-abs-thr*, -*rc-rel-edge-thr* and -*rc-rel-prun-thr* control the minimum number of reads from a given barcode aligning to sequences required to create an edge in the scaffold graph, the relative barcode support needed for creating edges and the relative barcode support needed for pruning edges, respectively.

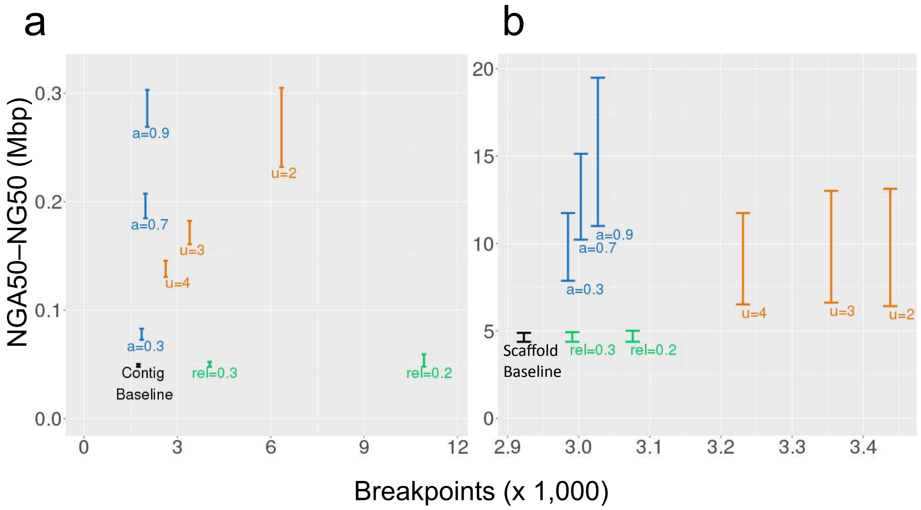


**Fig. 2.** Contiguity and correctness resulting from scaffolding (a) contig or (b) scaffold baseline assemblies with 10XG Chromium reads using fragScaff (orange), Architect (green), and ARCS (blue). We show the effect of the scaffolding parameters *-u* (fragScaff), *–rc-rel-edge-thr* (abbreviated rel, Architect) and *-a* (ARCS). The Y-axes show the range of NGA50 to NG50 lengths to indicate the uncertainty caused by real genomic variations (captured by breakpoints analysis) between individual NA24143 and the reference genome GRCh38. The X-axes show the number of breakpoints that occur when aligning the resulting assembly to the reference.

To assess correctness, we aligned the assemblies to the primary chromosome sequences of the human reference GRCh38 and counted the number of observed breakpoints using abyss-samtobreak [9]. At the contig level (Fig. 2a), we observe that while the ARCS and fragScaff assemblies (highest contiguity achieved at -*a*0.9 and -*u*2, in that order) have similar sequence contiguity (NG50 of 303,034 vs. 304,926 bp, respectively), the ARCS assembly has less than one third the number of breakpoints compared to fragScaff (2,030 vs. 6,345). In context, the corresponding ARCS and fragScaff assemblies have 16.3% and 263.4% more breakpoints than the baseline contig assembly, respectively (Tables S3-S5). This indicated that, while the resulting fragScaff assemblies were highly contiguous, they might harbor substantially more misassemblies. Architect scaffolding of the baseline contig assembly did not yield appreciable gains despite extensive parameter tuning (Fig. 2a and Table S6). Contig-level scaffolding took more than 5 hours and over 5 days for fragScaff and Architect, respectively (Table 1; complete benchmark results are available at: ftp://ftp.bcgsc.ca/supplementary/ARCS/benchmarks).

At the scaffold level (Fig. 2b), we observe that ARCS achieves a greater sequence contiguity and correctness than Architect and fragScaff (NG50 (Mbp) / breakpoints, 19.5 / 3,027 vs. 5.0 / 3,076 vs. 13.1 / 3,438 in that order) when comparing amongst the most contiguous assemblies for each tool (Tables S4-S6). This is despite there being roughly one order of magnitude fewer misassemblies between fragScaff and ARCS assemblies that used the scaffold as opposed to contig baseline sequences for scaffolding (411 vs. 4,315 breakpoints, respectively). These assemblies respectively harbor 3.6%, 5.2% and 17.6% more breakpoints than the baseline scaffold assembly, which suggests that ARCS and other scaffolders for 10XG data work best when the draft to re-scaffold is more contiguous. To see whether these 411 additional breakpoints in the fragScaff vs. ARCS assemblies are large-scale misassemblies, we aligned the corresponding assemblies to the reference human genome and plotted the alignments (Fig. 3). As observed, fragScaff scaffolding of the baseline scaffold sequences yields more interchromosomal translocation misassemblies (Fig. 3a) when compared to ARCS (Fig. 3b). We note that increasing the fragScaff -*j* parameter (mean passing links across nodes) while relaxing -*u* (score cut-off multiplier) yields assemblies whose contiguity rival that of ARCS (16.9 vs 19.5 Mbp NG50, respectively), but at the cost of increased misassemblies (Tables S4 and S5). Architect scaffolding of both contig and scaffold baseline assemblies yielded a marginal increase in contiguity figures (Fig. 2 and Table S6), which we can only speculate on. Apart from the tool design, which is intended for Illumina TruSeq read clouds [7], it is possible that the sequencing depth of the 10X Genomics linked reads used in our study (25-fold) may not be adequate to observe appreciable gains with this tool.

**Table 1.** Total wall-clock time and peak memory usage for ARCS (-*c*5 -*e*30000 -*r*0.05 -*l*5 -*a*0.3), Architect (-*t*10 or -*t*5 –*rc-abs-thr*5 –*rc-rel-edge-thr*0.4 –*rc-rel-prun-thr*0.2) and fragScaff (-*C*5 -*E*30000 -*j*1 -*u*4) scaffolding applied to baseline contig and scaffold assemblies.

| Scaffolder | Baseline assembly | Number of threads | Wall-clock time (h:mm) | Peak memory (GB) |
|---|---|---|---|---|
| ARCS | contig | 1 | 1:12 | 9.4 |
| Architect | contig | 1 | 141:18 | 11.1 |
| fragScaff | contig | 64 | 6:37 | 8.1 |
| ARCS | scaffold | 1 | 0:55 | 3.4 |
| Architect | scaffold | 1 | 6:07 | 9.6 |
| fragScaff | scaffold | 64 | 1:59 | 14.1 |

We also compared the resource efficiency of all three tools over the parameter range tested (See ftp://ftp.bcgsc.ca/supplementary/ARCS/benchmarks) and report its runtime and memory usage on the most contiguous assemblies of baseline scaffold sequences (Table 1). ARCS outperforms Architect and fragScaff for run time on both contigs and scaffolds (average 3 fold faster than fragScaff) and memory usage on scaffolds (4 times less memory when compared to fragScaff). It should be noted that the run time of Architect and fragScaff increases quadrat-

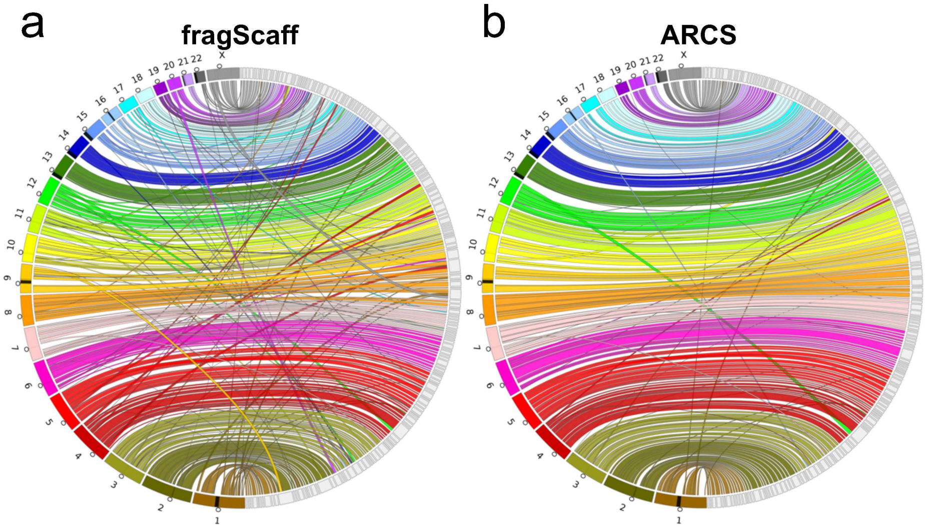


**Fig. 3.** (a) A Circos [15] assembly consistency plot of conservative fragScaff (-$C$5 -$E$30000 -$j$1 -$u$4) and (b) ARCS (-$c$5 -$e$30000 -$r$0.05 -$l$5 -$a$0.3) scaffolding of the baseline scaffold assembly. Scaftigs from the largest 175 (fragScaff) and 177 (ARCS) scaffolds, consisting of 75% of the genome are aligned to GRCh38 with BWA mem. GRCh38 chromosomes are displayed on the left while scaffolds are displayed on the right. Connections show the aligned regions between the genome and scaffolds. The open circles along chromosomes indicate the centromeres, while the black regions on chromosomes indicate gaps in the reference.

ically with the number of input sequences, making them inefficient choices for assemblies with a large number of input sequences (more than 250,000). Running Architect on the baseline contig assemblies took roughly 6 days (141 h) for most parameter combinations. In contrast, equivalent runs of this tool on the scaffold baseline assemblies were faster (6 h) due to having 20 times less sequences to process (Table S3). The execution speed of ARCS on the contig and scaffold baseline assemblies was consistent, both finishing in approximately 1 hour (1 h 12 m and 55 m, respectively).

### 3.2 Scaffolding with the NA12878 Chromium data

Recently, 10XG released their *de novo* assembly software called Supernova, which implements a scaffolding stage and is developed specifically for assembling Chromium data [8]. The authors presented a variety of human genome assemblies, each yielding N50 contiguity lengths 15 Mbp or higher, factoring in scaffolds 10 kbp and larger. We re-capitulated the Supernova experiment on 156-fold Chromium sequencing data for the NA12878 individual (Table S2), and corroborate their results (Table S7). When applying a scaffold sequence length cut-off on par with that used in our study (500 bp), we report N50 length metrics corrected for genome size (NG50) in the megabase range (14.7e6 bp), which is consistent with what was maximally achievable with ARCS using either the same dataset (NG50=18.3e6 bp) or the lower-coverage GIAB Chromium dataset (NG50=19.5e6 bp) that used the same scaffolding parameters. A Supernova assembly of 51-fold raw GIAB chromium reads produced a similarly contiguous assembly (NG50=13.5e6 bp).

Despite the NA12878 Chromium read data used with ARCS having substantially deeper coverage (5x deeper, Table S7 datasets 5 vs. 3), we observe that ARCS performs consistently across both human NA12878 and NA24143 datasets. Perhaps more interesting is the observation that there are only marginal gains in N50 length contiguity when using the higher coverage Chromium dataset (21.8e6 vs. 22.2e6 bp when using NA24143 GIAB vs. NA12878 data, with parameters -*e* 30,000 -*r* 0.05 -*c* 5 -*l* 5 -*a* 0.9) in spite of making 406 additional merges (Tables S2 and S7). This indicates that, under the conditions tested herein, with the draft assembly utilized and parameters set, the solution may work optimally with less data. We do stress the importance of knowing the distribution of reads within each barcode as it may vary between datasets, which is what is observed for NA24143 and NA12878 (Fig. S1). In our experience, this distribution is wider for GemCode and should be set to 50-10000 to include the bulk of the data while discarding outliers.

While we show that the contiguity of a *H. sapiens* contig assembly can be increased over six-fold with the use of 10XG data with only a marginal increase in probable errors with an average ± S.D. of 196 ± 77 total breakpoints compared to the baseline contig assembly, there are limitations to the method. As mentioned by Adey *et al.* [5], when using a barcode-based approach, it is difficult to confidently place short input sequences due to a lower number of indexed read pools aligning to the sequence. In addition, as the barcoded molecules may be over 100 kbp in length [16], it is possible that they span several entire short input sequences, preventing ARCS from extracting orientation information from read alignment positions, as they do not preferentially align to one end. Barcode reuse across molecules or incorrect alignment of linked reads due to repeats can also introduce false linkages at the sequence pairing stage, resulting in incorrect merges during the scaffolding stage.

With Chromium, 10X Genomics improved upon the GemCode protocol by increasing the number of fragment partitions and curbing barcode reuse. While the positional information of linked reads within a given fragment is not known, making it challenging for estimating gap or overlap sizes in genome assemblies, it remains an attractive technology for scaffolding draft genomes. This is especially true when the technique is applied to later stages of scaffolding, when the contiguity of the draft sequence assembly is high. To our knowledge, ARCS is the first publicly available stand-alone application for scaffolding draft genomes that is designed specifically for 10X Genomics Chromium reads.

## Acknowledgements

This work has been supported by the National Human Genome Research Institute of the National Institutes of Health (under award number R01HG007182), with additional support provided by Genome Canada, Genome British Columbia, and the British Columbia Cancer Foundation. The content is solely the responsibility of the authors, and does not necessarily represent the official views of the National Institutes of Health or other funding organizations.

## References

1. Narasimhan, V.M., Hunt, K.A., Mason, D., Baker, C.L., Karczewski, K.J., Barnes, M.R., Barnett, A.H., Bates, C., Bellary, S., Bockett, N.A., Giorda, K., Griffiths, C.J., Hemingway, H., Jia, Z., Kelly, M.A., Khawaja, H.A., Lek, M., McCarthy, S., McEachan, R., O'Donnell-Luria, A., Paigen, K., Parisinos, C.A., Sheridan, E., Southgate, L., Tee, L., Thomas, M., Xue, Y., Schnall-Levin, M., Petkov, P.M., Tyler-Smith, C., Maher, E.R., Trembath, R.C., MacArthur, D.G., Wright, J., Durbin, R., van Heel, D.A.: Health and population effects of rare gene knock-outs in adult humans with related parents. Science. 352, 474477 (2016). doi:10.1126/science.aac8624
2. Zheng, G.X., Lau, B.T., Schnall-Levin, M., Jarosz, M., Bell, J.M., Hindson, C.M., Kyriazopoulou-Panagiotopoulou, S., Masquelier, D.A., Merrill, L., Terry, J.M., Mudivarti, P.A., Wyatt, P.W., Bharadwaj, R., Makarewicz, A.J., Li, Y., Belgrader, P., Price, A.D., Lowe, A.J., Marks, P., Vurens, G.M., Hardenbol, P., Montesclaros, L., Luo, M., Greenfield, L., Wong, A., Birch, D.E., Short, S.W., Bjornson, K.P., Patel, P., Hopmans, E.S., Wood, C., Kaur, S., Lockwood, G.K., Stafford, D., Delaney, J.P., Wu, I., Ordonez, H.S., Grimes, S.M., Greer, S., Lee, J.Y., Belhocine, K., Giorda, K.M., Heaton, W.H., McDermott, G.P., Bent, Z.W., Meschi, F., Kondov, N.O., Wilson, R., Bernate, J.A., Gauby, S., Kindwall, A., Bermejo, C., Fehr, A.N., Chan, A., Saxonov, S., Ness, K.D., Hindson, B.J., Ji, H.P.: Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. Nat. Biotechnol. 34, 303311 (2016). doi:10.1038/nbt.3432
3. Kuleshov, V., Xie, D., Chen, R., Pushkarev, D., Ma, Z., Blauwkamp, T., Kertesz, M., Snyder, M.: Whole-genome haplotyping using long reads and statistical methods. Nat. Biotechnol. 32, 261266 (2014). doi:10.1038/nbt.2833
4. Mostovoy, Y., Levy-Sakin, M., Lam, J., Lam, E.T., Hastie, A.R., Marks, P., Lee, J., Chu, C., Lin, C., Dzakula, Z., Cao, H., Schlebusch, S.A., Giorda, K., Schnall-Levin, M., Wall, J.D., Kwok, P.Y.: A hybrid approach for de novo human genome sequence assembly and phasing. Nat. Methods. 13, 587590 (2016). doi:10.1038/nmeth.3865
5. Adey, A., Kitzman, J.O., Burton, J.N., Daza, R., Kumar, A., Christiansen, L., Ronaghi, M., Amini, S., Gunderson, K.L., Steemers, F.J., Shendure, J.: In vitro, long-range sequence information for de novo genome assembly via transposase contiguity. Genome Res. 24, 20412049 (2014). doi:10.1101/gr.178319.114
6. Zook, J.M., Catoe, D., McDaniel, J., Vang, L., Spies, N., Sidow, A., Weng, Z., Liu, Y., Mason, C.E., Alexander, N., Henaff, E., McIntyre, A.B., Chandramohan, D., Chen, F., Jaeger, E., Moshrefi, A., Pham, K., Stedman, W., Liang, T., Saghbini, M., Dzakula, Z., Hastie, A., Cao, H., Deikus, G., Schadt, E., Sebra, R., Bashir, A., Truty, R.M., Chang, C.C., Gulbahce, N., Zhao, K., Ghosh, S., Hyland, F., Fu, Y., Chaisson, M., Xiao, C., Trow, J., Sherry, S.T., Zaranek, A.W., Ball, M., Bobe, J., Estep, P., Church, G.M., Marks, P., Kyriazopoulou-Panagiotopoulou, S., Zheng, G.X., Schnall-Levin, M., Ordonez, H.S., Mudivarti, P.A., Giorda, K., Sheng, Y., Rypdal, K.B., Salit, M.: Extensive sequencing of seven human genomes to characterize benchmark reference materials. Sci Data. 3, 160025 (2016). doi: 10.1038/sdata.2016.25
7. Kuleshov, V., Snyder, M.P., Batzoglou, S.: Genome assembly from synthetic long read clouds. Bioinformatics. 32, i216-i224 (2016). doi:10.1093/bioinformatics/btw267
8. Weisenfeld, N.I., Kumar, V., Shah, P., Church, D., Jaffe, D.B.: Direct determination of diploid genome sequences. bioRxiv. 070425 (2016). doi:http:10.1101/070425

9. Jackman, S.D., Vandervalk, B.P., Mohamadi, H., Chu, J., Yeo, S., Hammond, S.A., Jahesh, G., Khan, H., Coombe, L., Warren, R.L., Birol, I.: ABySS 2.0: Resource-Efficient Assembly of Large Genomes using a Bloom Filter. bioRxiv. 068338 (2016). doi:10.1101/068338
10. Warren, R.L., Yang, C., Vandervalk, B.P., Behsaz, B., Lagman, A., Jones, S.J., Birol, I.: LINKS: Scalable, alignment-free scaffolding of draft genomes with long reads. Gigascience. 4, 35 (2015). doi:10.1186/s13742-015-0076-3
11. O'Connell, J., Schulz-Trieglaff, O., Carlson, E., Hims, M.M., Gormley, N.A., Cox, A.J.: NxTrim: optimized trimming of Illumina mate pair reads. Bioinformatics. 31, 2035-2037 (2015). doi:10.1093/bioinformatics/btv057
12. Li, H.: BFC: correcting Illumina sequencing errors. Bioinformatics. 31, 2885-2887 (2015). doi:10.1093/bioinformatics/btv290
13. Li, H., Durbin, R.: Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics. 26, 589-595 (2010). doi:10.1093/bioinformatics/btp698
14. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. 215, 403-410 (1990).
15. Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., Jones, S.J., Marra, M.A.: Circos: an information aesthetic for comparative genomics. Genome Res. 19, 1639-1645 (2009). doi:10.1101/gr.092759.109
16. Goodwin, S., McPherson, J.D., McCombie, W.R.: Coming of age: ten years of next-generation sequencing technologies. Nat. Rev. Genet. 17, 333351 (2016). doi: 10.1038/nrg.2016.49

## Supplementary information

**Table S1.** ARCS parameters and pertinent LINKS parameters for building the scaffold layout.

| Module | Parameter | Description | Recommended range/value |
|---|---|---|---|
| ARCS | -f | Genome seq. assembly draft file (Multi-FASTA) | NA |
| ARCS | -a | File of file names listing BAM alignment files | NA |
| ARCS | -s | Min. percent sequence identity to consider reads | 90-100, default: 98 |
| ARCS | -c | Min. number of mapping read pairs/barcode and seq. | 3-5, default: 5 |
| ARCS | -l | Min. number of barcode links to create graph edge[1] | 0-5, default: 0 |
| ARCS | -z | Minimum seq. length to consider | 250-5000, default: 500 |
| ARCS | -m | Barcode read frequency range (min-max) | 25-100000, default: 50-10000 |
| ARCS | -d | Max. degree of nodes in graph | typically set to 0 |
| ARCS | -e | Length to consider in 5' and 3' of seq. | 10000-60000, default: 30000 |
| ARCS | -r | Max. p-val. head/tail and orientation assignments | 0.05-0.1, default: 0.05 |
| LINKS | -l | Min. number of links to consider an edge | 3-5, default: 5 |
| LINKS | -a | Max. barcode link ratio between two edges at fork | 0.3-0.9, default: 0.3 |
| LINKS | -z | Min. seq. length to consider | 250-1000, default: 500 |

[1] Best handled in LINKS

**Table S2.** Datasets used in our study.

| Dataset | Individual | Processing | Number of | Read length | Fold |
|---|---|---|---|---|---|
| | | step | read pairs | (bp) | coverage |
| 1 | NA24143 | Raw reads sequenced | 523,746,206 | 151 | 51.2 |
| 2 | NA24143 | Reads from BAM | 420,496,741 | 128 | 34.9 |
| 3 | NA24143 | Filtered from BAM | 305,846,648 | 128 | 25.3 |
| 4 | NA12878 | Raw reads sequenced | 1,598,106,419 | 151 | 156.3 |
| 5 | NA12878 | Post Long Ranger | 1,514,291,941 | 128/151 | 136.8 |

**Table S3.** Contiguity length metrics and number of sequence alignment breakpoints for the baseline ABySS v2.0 contigs and scaffolds obtained from assembling GIAB NA24143 Illumina WGS 2x250 bp paired-end and 6 kbp mate-pair sequence data.

| Assembly | Sequences | NG50 | NGA50 | Number of |
|---|---:|---:|---:|---:|
| Stage | $\geq 3kbp$ | (bp) | (bp) | breakpoints |
| contig | 80,910 | 50,351 | 47,878 | 1,746 |
| scaffold | 4,037 | 4,889,645 | 4,377,837 | 2,923 |

**Table S4.** ARCS contiguity length metrics and breakpoints obtained from scaffolding contigs and scaffolds greater than 3kbp with various parameterizations. The NG50 and NGA50 lengths were calculated for scaffolds 500 bp and longer. Values in bold are plotted in the manuscript, Fig. 2. In ARCS, We consider (-*c* or more) reads that align to the 5' and 3' end (-*e* or less) bases of each sequences. The number of read pairs of the same barcode aligning to the head or the tail of a scaffold is tallied, and a binomial test is used to calculate whether the observed distribution is significantly different from a uniform distribution (threshold p=0.05, parameter -*r*). Once oriented relative to each other, pairs of sequence IDs are passed onto LINKS for generating the scaffold layout. Edges in the graph are considered with sufficient (-*l* or more) barcode links. Forks in the graph are resolved by choosing the edge with the most support, and when the ratio of barcode links of the second most supported edge relative to it is equal or below a threshold (-*a*).

| Baseline assembly | *e* | *r* | *c* | *l* | *a* | NG50 (bp) | NGA50 (bp) | Breakpoints |
|---|---:|---:|---:|---:|---:|---:|---:|---:|
| **contig**[1] | 30,000 | 0.05 | 5 | 5 | 0.3 | **82,979** | **72,782** | **1,851** |
| contig | 30,000 | 0.05 | 5 | 5 | 0.5 | 142,140 | 127,239 | 1,915 |
| **contig** | 30,000 | 0.05 | 5 | 5 | 0.7 | **207,455** | **184,753** | **1,972** |
| **contig** | 30,000 | 0.05 | 5 | 5 | 0.9 | **303,034** | **268,962** | **2,030** |
| **scaffold**[1,2] | 30,000 | 0.05 | 5 | 5 | 0.3 | **11.74e6** | **7.87e6** | **2,985** |
| scaffold | 30,000 | 0.05 | 5 | 5 | 0.5 | 13.81e6 | 9.05e6 | 2,999 |
| **scaffold** | 30,000 | 0.05 | 5 | 5 | 0.7 | **15.13e6** | **10.22e6** | **3,003** |
| **scaffold** | 30,000 | 0.05 | 5 | 5 | 0.9 | **19.48e6** | **11.00e6** | **3,027** |
| scaffold | 60,000 | 0.05 | 5 | 5 | 0.3 | 13.24e6 | 8.07e6 | 3,016 |
| scaffold | 60,000 | 0.05 | 5 | 5 | 0.5 | 15.69e6 | 9.38e6 | 3,033 |

[1] Benchmarking results for the corresponding assemblies are reported in the manuscript, Table 1

[2] The corresponding assembly is depicted in the manuscript, Fig. 3b

**Table S5.** fragScaff contiguity length metrics and breakpoints obtained from scaffolding contigs and scaffolds greater than 3kbp with various parameterizations. The NG50 and NGA50 lengths were calculated for scaffolds 500 bp and longer. Values in bold are plotted in the manuscript, Fig. 2. The parameters -*E*, -*C*, -*j*, and -*u* respectively control the sequence end node size, the minimum number of reads required to align to a node, the mean number of passing links across nodes and link validity.

| Baseline assembly | *E* | *C* | *j* | *u* | NG50 (bp) | NGA50 (bp) | Breakpoints |
|---|---|---|---|---|---|---|---|
| contig | 5,000 | 5 | 1 | 2 | 313,774 | 253,880 | 5,651 |
| contig | 5,000 | 5 | 1 | 3 | 193,266 | 171,334 | 3,263 |
| contig | 5,000 | 5 | 1 | 4 | 148,482 | 112,428 | 2,270 |
| contig | 5,000 | 5 | 1 | 5 | 85,861 | 76,430 | 2,017 |
| contig | 5,000 | 10 | 1 | 2 | 176,775 | 144,624 | 7,592 |
| contig | 5,000 | 10 | 1 | 3 | 141,559 | 122,812 | 5,180 |
| contig | 5,000 | 10 | 1 | 4 | 102,642 | 92,638 | 3,149 |
| contig | 5,000 | 10 | 1 | 5 | 77,145 | 70,639 | 2,301 |
| **contig** | 30,000 | 5 | 1 | 2 | **304,926** | **231,937** | **6,345** |
| **contig** | 30,000 | 5 | 1 | 3 | **182,369** | **160,833** | **3,393** |
| **contig**[1] | 30,000 | 5 | 1 | 4 | **145,539** | **130,710** | **2,622** |
| contig | 30,000 | 5 | 1 | 5 | 145,539 | 130,710 | 2,622 |
| contig | 30,000 | 5 | 2 | 2 | 673,216 | 314,033 | 13,191 |
| contig | 30,000 | 5 | 3 | 2 | 1.23e6 | 330,317 | 17,376 |
| scaffold | 5,000 | 1 | 1.25 | 2 | 14.13e6 | 6.41e6 | 3,575 |
| scaffold | 5,000 | 3 | 1.25 | 2 | 13.98e6 | 6.44e6 | 3,492 |
| scaffold | 5,000 | 5 | 1.25 | 2 | 11.85e6 | 6.10e6 | 3,495 |
| scaffold | 5,000 | 5 | 1 | 2 | 11.85e6 | 6.10e6 | 3,495 |
| scaffold | 5,000 | 5 | 1 | 3 | 11.20e6 | 6.10e6 | 3,435 |
| scaffold | 5,000 | 5 | 1 | 4 | 9.57e6 | 5.87e6 | 3,331 |
| scaffold | 5,000 | 5 | 2 | 2 | 11.85e6 | 6.10e6 | 3,495 |
| **scaffold** | 30,000 | 5 | 1 | 2 | **13.13e6** | **6.41e6** | **3,438** |
| **scaffold** | 30,000 | 5 | 1 | 3 | **13.01e6** | **6.62e6** | **3,355** |
| **scaffold**[1,2] | 30,000 | 5 | 1 | 4 | **11.74e6** | **6.52e6** | **3,231** |
| scaffold | 30,000 | 5 | 1 | 5 | 10.55e6 | 6.30e6 | 3,151 |
| scaffold | 30,000 | 5 | 2 | 2 | 16.93e6 | 6.52e6 | 3,813 |
| scaffold | 30,000 | 5 | 3 | 2 | 16.93e6 | 6.52e6 | 3,813 |

[1] Benchmarking results for the corresponding assemblies are reported in the manuscript, Table 1

[2] The corresponding assembly is depicted in the manuscript, Fig. 3a

**Table S6.** Architect contiguity length metrics and breakpoints obtained from scaffolding contigs and scaffolds greater than 3kbp with various parameterizations. The NG50 and NGA50 lengths were calculated for scaffolds 500 bp and longer. Values in bold are plotted in the manuscript, Fig. 2. The parameters *-t*, *-abs*, *-rel* and *-prun* in Architect control the minimum number of aligned reads from a barcode required for a sequence hit, the minimum number of aligning reads from a given barcode required to create a graph edge, the relative barcode support needed for creating edges and the relative barcode support needed for pruning edges, respectively.

| Baseline assembly | *t* | *abs* | *rel* | *prun* | NG50 (bp) | NGA50 (bp) | Breakpoints |
|---|---|---|---|---|---|---|---|
| **contig** | 5 | 3 | 0.2 | 0.2 | 59,442 | **48,048** | **10,922** |
| **contig** | 5 | 3 | 0.3 | 0.2 | 52,502 | **47,887** | **4,035** |
| contig | 5 | 3 | 0.4 | 0.2 | 50,689 | 47,876 | 2,113 |
| contig | 5 | 5 | 0.2 | 0.2 | 59,428 | 48,044 | 10,900 |
| contig | 5 | 5 | 0.3 | 0.2 | 52,499 | 47,887 | 4,030 |
| contig | 5 | 5 | 0.4 | 0.2 | 50,689 | 47,876 | 2,110 |
| contig | 10 | 3 | 0.2 | 0.2 | 58,171 | 48,026 | 9,105 |
| contig | 10 | 3 | 0.3 | 0.2 | 51,951 | 47,880 | 3,297 |
| contig | 10 | 3 | 0.4 | 0.2 | 50,577 | 47,876 | 1,995 |
| contig | 10 | 5 | 0.2 | 0.2 | 58,170 | 48,026 | 9,083 |
| contig | 10 | 5 | 0.3 | 0.2 | 51,948 | 47,880 | 3,292 |
| contig[1] | 10 | 5 | 0.4 | 0.2 | 50,570 | 47,876 | 1,991 |
| scaffold | 5 | 3 | 0.1 | 0.1 | 5.48e6 | 4.38e6 | 3,293 |
| scaffold | 5 | 3 | 0.2 | 0.1 | 5.01e6 | 4.38e6 | 3,076 |
| **scaffold** | 5 | 3 | 0.2 | 0.2 | **5.01e6** | **4.38e6** | **3,076** |
| **scaffold** | 5 | 3 | 0.3 | 0.2 | **4.93e6** | **4.38e6** | **2,991** |
| scaffold | 5 | 3 | 0.4 | 0.2 | 4.93e6 | 4.38e6 | 2,974 |
| scaffold | 5 | 5 | 0.2 | 0.2 | 5.01e6 | 4.38e6 | 3,076 |
| scaffold | 5 | 5 | 0.3 | 0.2 | 4.93e6 | 4.38e6 | 2,991 |
| scaffold[1] | 5 | 5 | 0.4 | 0.2 | 4.93e6 | 4.38e6 | 2,974 |
| scaffold | 10 | 3 | 0.1 | 0.1 | 5.36e6 | 4.38e6 | 3,216 |
| scaffold | 10 | 3 | 0.2 | 0.1 | 5.01e6 | 4.38e6 | 3,060 |
| scaffold | 10 | 3 | 0.2 | 0.2 | 5.01e6 | 4.38e6 | 3,060 |
| scaffold | 10 | 3 | 0.3 | 0.2 | 4.93e6 | 4.38e6 | 2,981 |
| scaffold | 10 | 3 | 0.4 | 0.2 | 4.89e6 | 4.38e6 | 2,973 |
| scaffold | 10 | 5 | 0.2 | 0.2 | 5.01e6 | 4.38e6 | 3,056 |
| scaffold | 10 | 5 | 0.3 | 0.2 | 4.93e6 | 4.38e6 | 2,981 |
| scaffold | 10 | 5 | 0.4 | 0.2 | 4.89e6 | 4.38e6 | 2,973 |

[1] Benchmarking results for the corresponding assemblies are reported in the manuscript, Table 1. The parameters were abbreviated to fit the table: *abs, rel* and *prun* correspond to *–rc-abs-thr, –rc-rel-edge-thr* and *–rc-rel-prun-thr*, respectively

**Table S7.** Supernova (SN) assemblies of a human Chromium datasets and comparison to ARCS scaffolding of a human ABySS scaffold assembly.

| Dataset | Individual | Assembly | Cut-off[1] size (kbp) | n | NG50 (bp) | N50 (bp) | Largest (bp) | Breakpoints |
|---|---|---|---|---|---|---|---|---|
| 4 | NA12878 | 10XG SN v1.0 | 10 | 1,231 | 14.66e6 | 16.40e6 | 68.87e6 | 3,737 |
| 4 | NA12878 | Local SN v1.1 | 10 | 1,341 | 14.74e6 | 16.22e6 | 57.01e6 | 3,782 |
| 4 | NA12878 | Local SN v1.1 | 0.5 | 21,774 | 14.74e6 | 16.10e6 | 57.01e6 | 3,782 |
| 1 | NA24143 | Local SN v1.1 | 0.5 | 23,693 | 13.47e6 | 15.03e6 | 95.16e6 | 3,879 |
| 3 | NA24143 | ARCS v1.0[2] | 0.5 | 64,922 | 19.48e6 | 21.82e6 | 97.86e6 | 3,027 |
| 5 | NA12878 | ARCS v1.0[3] | 0.5 | 64,516 | 18.34e6 | 22.16e6 | 111.6e6 | 3,225 |

[1] Cut-off size for reporting the assembly length metrics

[2] Parameters: -*m* 50-1000 -*s* 98 -*z* 3000 -*e* 30,000 -*r* 0.05 -*c* 5 -*l* 5 -*a* 0.9

[3] Parameters: -*m* 50-6000 -*s* 98 -*z* 3000 -*e* 30,000 -*r* 0.05 -*c* 5 -*l* 5 -*a* 0.9

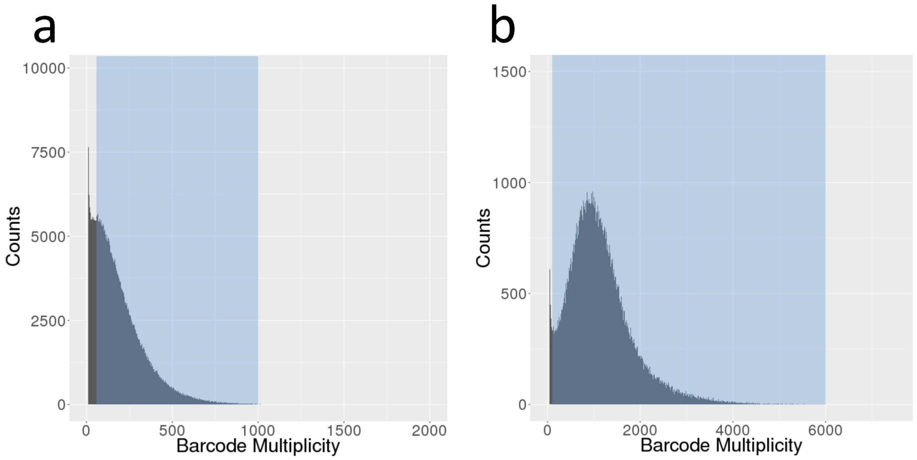


**Fig. S1.** Distributions of barcode-read multiplicities (read frequency per index) in human (a) NA24143 and (b) NA12878 Chromium datasets. Blue shades show the multiplicity range we set in ARCS as -*m* 50-1000 and -*m* 50-6000 for the NA24143 and NA12878 Chromium sequence data, respectively.