bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

FALCON-Phase: Integrating PacBio and Hi-C data for phased diploid genomes

Zev N. Kronenberg[1◑], Richard J. Hall[2], Stefan Hiendleder[3,4], Timothy P. L. Smith[5], Shawn T. Sullivan[1], John L. Williams[3], and Sarah B. Kingan[2◑*]

1 Phase Genomics, Seattle, WA, USA
2 Pacific Biosciences, Menlo Park, CA, USA
3 Davies Research Centre, School of Animal and Veterinary Sciences, The University of Adelaide, Roseworthy, Australia
4 Robinson Research Institute, The University of Adelaide, Adelaide, Australia
5 US Meat Animal Research Center, ARS USDA, Clay Center, Nebraska, USA

◑These authors contributed equally to this work.
*skingan@pacificbiosciences.com

Abstract

De novo genome assembly of outbred diploid organisms remains a challenge in computational biology due to the difficulty of resolving similar haplotypes. FALCON-Unzip, a phased diploid genome assembler, separates PacBio long-reads by haplotype during assembly. The assembler outputs contiguous primary contigs, which are pseudohaplotypes containing phased haplotype regions and collapsed haplotypes. The ability to phase depends on the density of heterozygous variants, depth of coverage, and read length. As a result, haplotype phase information is lost when phase blocks are interrupted by regions of low heterozygosity, resulting in phase switches. Here, we present FALCON-Phase, a new method that resolves phase-switches by reconstructing contig-length phase blocks using Hi-C short-reads mapped to both homozygous regions and phase blocks. Such Hi-C data contain ultra-long-range phasing information (>1Mb). The novel FALCON-Phase algorithm is highly accurate (>96%) when benchmarked against a pedigree-based truth-set. The FALCON-Phase pipeline can also be extended to scaffolds to generate chromosome-scale phase blocks. The code is freely available (https://github.com/phasegenomics/FALCON-Phase) under a BSD and attribution license.

Author summary

FALCON-Phase combines PacBio long-read and Hi-C data for phased diploid genome assembly. The method builds upon FALCON-Unzip, which generates haplotype-resolved phase blocks limited by regions of low heterozysity along a genome. FALCON-Phase uses the ultra-long-range haplotype information in Hi-C data to resolve phase block switches and backfill homozygous regions, resulting in two phased haplotigs. The FALCON-Phase pipeline can also be applied to scaffolds to produce chromosome-scale phased diploid genome assemblies.

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Introduction

A high quality reference genome is an indispensable resource for basic and applied research in biology, genomics, agriculture, and medicine [1–3]. Fortunately, the cost of sequencing and computation for genome assembly projects is rapidly declining due to technological innovations in DNA sequencing, long-range genotyping, and assembly algorithms [4]. As researchers are now able to pursue genome projects for outbred, non-model, diploid and polyploid organisms, a major challenge in *de novo* assembly is accurate haplotype resolution. Most genome assemblers "collapse" multiple haplotypes into a single consensus sequence to generate a pseudo haploid reference. Unfortunately, this process can result in mosaic haplotypes with erroneously associated variants not present in either haplotype and concomitant effects on biological inference [5].

There are currently three approaches to haplotype resolution in long-read diploid genome assembly. The first is to utilize family trio information to separate long-reads from an F1 into maternal and paternal pools [6]. These parent-specific read pools are then simply assembled into two haploid references, thus avoiding the added complexity of diploid assembly. This method has been implemented in TrioCanu, and requires Illumina short-read data from both parents in order to identify parent-specific markers in the offspring long-read data [6]. Another approach uses a reference genome and read-backed phasing to infer both haplotypes followed by long-read partitioning and assembly [7–10]. Read-back phasing methods require single nucleotide variant (SNV) calling.

The diploid assembly approach is to separate haplotypes during the genome assembly process as implemented by FALCON-Unzip [11]. FALCON-Unzip outputs two types of genomic contigs: primary contigs, which are highly contiguous pseudo-haplotypes containing both phased and unphased haplotypes, and haplotigs, which are shorter phased contigs that represent the alternate alleles in heterozygous regions of the primary contig. The length of the phase-blocks (haplotigs) produced by FALCON-Unzip are limited by PacBio read lengths and the magnitude and distribution of heterozygosity in the diploid genome. Regions of low heterozysity are collapsed into non-haplotype-resolved regions because they contain insufficient information for read phasing. As a consequence, chromosome-specific information is lost between sequential phase blocks that flank a collapsed haplotype region.

To address the problem of phase switching between FALCON-Unzip phase blocks, we have implemented a novel algorithm called FALCON-Phase which integrates ultra-long-range genotype (>1Mb) information in the form of Hi-C read pairs [12]. Different from read-back phasing methods, FALCON-Phase does not require SNV calling, but instead relies on the mapping pattern of Hi-C read pairs between contigs [8, 10, 13]. We validated the FALCON-Phase method against a highly heterozygous *Bos taurus*/*Bos indicus* F1 genome assembly that was phased using TrioCanu. We find that FALCON-Phase is >96% accurate, suggesting that Hi-C proximity information can be used to correct nearly all haplotype switches along FALCON-Unzip primary contigs and replace the requirement of parental genotype information.

Materials and methods

FALCON-Phase has three stages: processing FALCON-Unzip contigs and Hi-C data, application of the phasing algorithm, and emission of two phased haplotigs (Fig. 1). We implemented FALCON-Phase using the Snakemake language to provide flexibility and pipeline robustness [14]. The pipeline can be run interactively, on a single computer, or submitted to a cluster job scheduler.

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Data preprocessing

Haplotig placement

In stage one, a haplotig placement file is generated for the FALCON-Unzip assembly. This file uses the pairwise alignment format (PAF) [15] and specifies the map location of each haplotig on the primary contig (Fig. 1). Briefly, haplotigs are aligned, filtered, and processed with three utilities of the *mummer* v4 package [16]: *nucmer*, *delta-filter* and *show-coords*. Sub-alignments for each haplotig are chained in one dimension to find the optimal start and end of the placement using the *coords2hp.py* script. Finally, non-unique haplotig mappings and those fully contained by other haplotigs are removed with *filt_hp.py*.

Contig mincing

The haplotig placement file is used to generate three "minced" FASTA files (Fig. 1), "A_haplotigs.fasta", "B_haplotigs.fasta", and "collapsed_haplotypes.fasta". The "A" haplotigs are the original FALCON-Unzip haplotigs (red in Fig. 1), the "B" haplotigs are the corresponding phased region of the primary contigs (the alternate phased haplotype, blue in Fig. 1), and the "collapsed" haplotypes are the unphased regions of the assembly (grey in Fig. 1). The corresponding A and B minced haplotigs are paired ("A-B haplotig pairs" or "phase blocks") and tracked throughout phasing and emission of the final output (see Fig. 1). The three FASTA files are concatenated together into a single file for Hi-C read mapping.

Minced contig index file

A-B haplotig pairs are passed to the phasing algorithm in an index file, "ov_index.txt" generated by the *primary_contig_index.pl* script using the minced FASTA and haplotig placement file as inputs.

Minced contig index file format

The minced contig index file ("ov_index.txt") is a three column space-delimited file. There is one row for each FALCON-Unzip primary contig. Each number is the zero-based index of the minced contig in the concatenated FASTA. The columns consist of the following:

1. Primary contig ID.
2. Comma separated list of minced contig indices associated with the primary contig.
3. Comma separated list of tuples for A-B haplotig pairs. If there are no A-B pairings the field will be "NA." This last column is referenced as array C in the FALCON-Phase algorithm. For example, in the third row of the example below $C[0,1] = 5216$, $C[1,0] = 12$, and $C[1,1] = 900$.

Example rows from a contig index file:

```
000031F 7204 NA
000034F 7265,4310,7266,1195 1195:4310
000104F 8092,5216,2101 2101:5216,12:900
```

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Hi-C read mapping

The Hi-C reads are mapped to the minced contigs using *bwa-mem*, with the Hi-C option (-5) enabled [17]. The mapped reads are streamed to *samtools*, removing reads with the 2316 flag set, and are thereafter sorted by read name [18]. This operation ensures that each mate-pair only contains two entries. In the last step of read processing a map quality filter of 10 is applied removing reads without haplotype specific sequence. Both stringent (60) and relaxed (0) map quality filtering generated suboptimal-results.

Contact matrix

The Hi-C mate-pair counts between minced contigs are enumerated into a contact matrix, M. Each element, $M_{i,j}$, in the matrix is later normalized by the number of restriction enzyme sites, z, in both the ith and j^{th} minced contigs as shown in Eq 1. The raw count matrix is encoded into a binary "binmat" format. The normalization step occurs in the second stage of the pipeline.

$$\hat{\mathrm{M}}_{\mathrm{i,j}} := \frac{M_{i,j}}{z_i + z_j} \tag{1}$$

Phasing algorithm

We designed an algorithm to phase the minced contigs for diploid genome assemblies based on Hi-C read pair mapping (see Al. 1). The algorithm stochastically searches for the optimum set of phase block configurations. The expected computational complexity of the algorithm is $O(n^2)$, where n is the number of phase blocks (e.g. "A-B haplotig pairs"). The algorithm is given a list, C, of tuples for the A-B haplotig pairs and their sequential ordering along each primary contig. During initialization each member of the A-B haplotig pair is randomly assigned one of the two possible phase configurations for a diploid organism $\in (\{[0,1],[1,0]\})$. The phase assignment is stored in array T, where 0 corresponds to phase configuration [0,1]. The algorithm sweeps along the A-B pairs of each primary contig, randomly assigning a phase for the haplotig pairs, conditioned on the phase assignment of the previous A-B pairs and the Hi-C links between them. The *phaseFreq* function (Eq. 2) calculates the frequency of Hi-C links from the current region, i, to all past regions, j, that have the same phase, i.e. $T[i] = T[j] = 1 = [1,0]$.

$$phaseFreq(i, T, \hat{M}, C) = \frac{\sum_{j=0}^{j<i} \gamma(i,j) * \alpha(i,j)}{\sum_{j=0}^{j<i} \alpha(i,j)} \tag{2}$$

The *phaseFreq* function takes the index of the current A-B haplotig pair, i, the phase assignment of all regions associated with a given primary contig, array T, the normalized Hi-C count matrix, $\hat{M}$, and the C array of the A-B haplotig pairs. The gamma function (Eq. 3) determines if two regions are in phase, and if so returns 1. The alpha function (Eq. 4) gives the normalized counts of Hi-C links between both sets of A-B haplotig pairs (see example box).

$$\gamma(\mathrm{i,j}) = \begin{cases} 1, & T[i] = T[j] \\ 0, & T[i] \neq T[j] \end{cases} \tag{3}$$

$$\alpha(i, j, \hat{M}, C) = \hat{M}[C[i,0], C[j,1]] + \hat{M}[C[i,1], C[j,0]] \tag{4}$$

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

```
Data: normalized Hi-C count matrix (M̂), contig overlap index array (C), number
      of permutations (n), burn-in (b)
Result: (R) array, the phase of the A-B haplotig pairs ∈ {0,1}
m ← length of C - 1;
R ≡ result array of length of C;
T ≡ temporary phase array of length of C;
P ≡ state count array (T[i] = 1) of length C;
if length of C == 1 then
  | return R[0] ← random(∈ {0,1})
end
for j ← 0 to m do
  | R[j] ← T[j] ← random(∈ {0,1});
  | P[j] ← 0;
end
for i ← 0 to n do
  | for j ← 1 to m do
  |   | T[j] ← 1;
  |   | if phaseFreq(j, T, M̂, C) < runif() then
  |   |   | T[j] ← 0;
  |   | end
  |   | if i > b and T[j] = 1 then
  |   |   | P[j] ← P[j] + 1;
  |   | end
  | end
end
for j ← 0 to m do
  | R[j] ← 1;
  | if P[j]/(n − b) < 0.5 then
  |   | R[j] ← 0;
  | end
end
return R
```

Algorithm 1: Phasing procedure

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

The process of phase assignment across a primary contig is iterated for a burn-in period followed by a scoring period. In the scoring period the number of iterations that each member of the A-B pair spends in phase 1 [1,0] is enumerated. The final phase assignment is the configuration in which each member of an A-B pair spent the most iterations. In practice, 10^7 iterations with $5 * 10^6$ burn-in period generated consistent results. The phasing algorithm requires nominal computational resources. The limiting computational resource is memory as $\hat{M}$ is not sparse.

Emission of Phased Haplotigs

Once the phase of each A-B haplotig pair (phase block) is determined, the minced FASTA sequences are joined into two phased, full-length haplotig sequences (phase0 and phase1) per primary contig (Fig. 1). The ordering of minced contigs (A-B haplotig pairs and collapsed haplotypes) along each FALCON-Unzip primary contig is known from the haplotig placement file. The emission step involves joining the correctly ordered minced contigs and choosing between A and B minced haplotig to produce "phase0" and "phase1" full-length haplotigs.

Evaluation of Method

F1 Bull Genome Dataset

We evaluated FALCON-Phase assembly (phase0 and phase1) of an F1 bull which is a cross between the two cattle subspecies, *Bos taurus indicus* and *Bos taurus taurus* against the dam and sire TrioCanu assembly [6]. The TrioCanu haploid assembly lengths were 2.57Gb and 2.68Gb, for the dam (*B. t. indicus*) and sire (*B. t. taurus*) derived contigs respectively. The ultra-long-range Hi-C data was generated by Phase Genomics.

Phase Assignment

Phase assignment of the minced A-B haplotig pairs (phase blocks) was evaluated through minimap2 [15] alignments to the TrioCanu assemblies. We determined the parent by the higher pairwise identity of the longest haploid alignment and required that the parental assignment of each A-B pair be concordant. For example, in a case where both members of the A-B pair are assigned to the same parent, the phase block would receive no parental assignment.

We were able to assign 2.32 Gb to dam and 2.32 Gb to sire (see Tab. 1) and evaluated FALCON-Phase using only these regions. In total, 1,704 out of the 3,128 A-B pairs could not be unambiguously assigned to parents. These unassigned pairs were largely small fragments, comprising a total of 116 Mb, (5%), of the phased, or "unzipped" portion of the assembly. In addition, 374 Mb of of the primary contigs were not unzipped, and remained as collapsed haplotypes.

Full-length Haplotig Production

Errors in alignment-based haplotig placement can introduce errors in the join sites during the full-length haplotig reconstruction. We tested the accuracy of the FALCON-phase junctions by aligning them to the TrioCanu dam assembly. We aligned 1,000bp around each of the 5,346 FALCON-Phase join sites using *bwa mem* (bwa-0.7.17-r1188) [17]. A junction was scored as concordant if greater than 750bp aligned in a single segment with no chimeric alignments.

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Table 1. Amount of the FALCON-Unzip assembly assayed.

Contig Assignment	Count	Length	Mean length
Dam - unzipped	2,305	2.32 Gb	1.01 Mb
Sire - unzipped	2,305	2.32 Gb	1.01 Mb
NA - unzipped	1,704	116 Mb	68.1 Kb
Collapsed	3,934	374 Mb	88.2 Kb

Accounting of the phased ("unzipped") and collapsed regions of the FALCON-Unzip assembly and the parental assignment of the Unzipped regions. using TrioCanu dam and sire contigs. "NA" denotes pairs that were unassigned due to either low differentiation or no orthogonal sequence in the TrioCanu assemblies.

Scaffolding and Evaluation

We scaffolded the phase0 full length haplotigs from FALCON-Phase using default *Proximo* (Phase Genomics, WA) settings [19, 20]. *Proximo* generated chromosome-scale scaffolds. The pairing of the phase0 and phase1 full length haplotigs is known and their order along the bull chromosomes was determined by scaffolding. This enabled us to apply FALCON-Phase a second time, resulting in phased, chromosome-scale scaffolds.

Scaffold-level chromosome phasing was evaluated against inheritance patterns of short-read genotypes. Standard paired-end Illumina libraries from dam (biosample:SAMN08473804) and sire (biosample:SAMN08473804) were aligned to FALCON-Phase haplotigs from phase0 using *bwa mem* (bwa-0.7.17-r1188) [17]. Reads were sorted with *samtools* [18]. Single Nucleotide Variants (SNVs) were called using *Freebayes* [21]. The VCF file was manually filtered, keeping homozygous sites where the dam and sire had different genotypes. Phase0 haplotigs were classified as dam or sire given the pattern of inheritance along each contig. The fraction of SNVs that did not match the haplotig parental assignment were used to measure accuracy.

Results

The FALCON-Unzip assembler generated 2.7 Gb of primary contig sequence for the F1 bull, of which 2.45 Gb (90%) was "unzipped." The N50 for the primary contig set was 31.4 Mb while the N50 for the haplotig set was 2.48 Mb, reflecting the short phase blocks of the initial assembly. The FALCON-Phase pipeline correctly phased a total of 2.1 Gb of the unzipped sequence. This represents accuracy of 96.72% when measured against the TrioCanu parental assignment. Using the same parental assignment approach on the unphased assembly, we find 63% phasing accuracy of the FALCON-Unzip primary contigs. Therefore, FALCON-Phase provides an increase by nearly 34% percentage points in phasing accuracy along primary contigs. Visualizing the haplotype phase along primary contigs (e.g. Fig. 2A) revealed low rates of false haplotype switches.

A second iteration of FALCON-Phase, run on scaffolds, reconstructed a dam and sire haplotype for each chromosome-scale scaffold. We measured the accuracy of the first scaffold, Scaffold1. Scaffold1 was 157Mb and highly syntenic to the bovine reference chromosome one (see Fig. 2B) [22]. As an orthogonal form of phasing validation we used dam and sire SNVs discovered against phase0 contigs. Before scaffold-level phasing, we found nearly equal number of dam and sire specific SNVs: 110,146 and 109,021, respectively. After scaffold-level phasing the first haplotype (phase0) contained 95.4% dam specific SNVs (209,147:10,020) (Fig. 2C). We examined the SNVs that were discordant against the phase0. We found a few phase switching errors that were propagated from the initial round of contig phasing. Overall, the SNV inheritance

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

pattern shows that the FALCON-Phase algorithm is effective at long-range phasing.

We investigated the mincing and in-phase joining of haplotigs on the accuracy of the phased, full-length haplotig pairs by mapping 5,346 dam FALCON-Phase join sites to the TrioCanu dam assembly (see Methods). We note that 88% (4,709) of the sites are highly concordant with the TrioCanu dam assembly. There were 641 discordant alignments (12%). Of the discordant alignments, 333 were split-read mappings and 116 had trans-contig mappings. Most of the discordant events occurred on smaller contigs, likely in difficult-to-assemble regions of the genome. These results are consistent with mapping errors and differences in assemblies rather than issues with the join sites. Furthermore, when manually checking FALCON-Phase haplotype alignments we see only a few gross errors (incorrect parental assignment), likely due to haplotig placement via alignment. In the next version of FALCON-Unzip the haplotig placement files will be generated directly from the assembly graph, relieving errors caused by mapping haplotigs to the primary contigs.

Discussion

FALCON-Phase is a novel method that extends haplotype phasing to the chromosome level in diploid genome assemblies. FALCON-Phase relies on two data types which are commonly used to generate high quality reference genomes: PacBio long reads and Hi-C read pairs. Because it does not require family-trio data, it can be applied to wild-caught samples or organisms lacking pedigree information.

FALCON-Phase builds upon the functionality of the FALCON-Unzip assembler, which generates haplotype blocks in regions of the genome exhibiting high levels of heterozygosity. FALCON-Phase integrates ulta-long-range Hi-C data to phase these haplotype blocks along a contig. In this study, we applied FALCON-Phase in two iterations to an F1 bull genome. First, we generated pairs of high contiguity haplotigs representing maternal and paternal haplotypes. We scaffolded these contigs using the Hi-C data, and performed a second round of FALCON-Phase, resulting in phased chromosome sequences.

Our method is optimized for outbred diploid organisms with moderate levels of heterozygosity. Typical parameter settings for FALCON-Unzip genome assembly allow haplotypes up to 5% divergent from each other to be "unzipped." Haplotypes with higher divergence may be assembled as separate primary contigs and could not be paired with the current haplotig placement functionality. Future development work can focus on broadening the application to high heterozygosity samples and organisms with higher ploidy.

Future work will also involve the integration of the phasing algorithm into the assembly process itself. During the layout phase of the genome assembly process, "bubbles" in the overlap graph capture divergent haplotypes. The Hi-C data can be integrated into the graph data structure in order to determine the haplotype-specific path through multiple bubble regions. Integration at this stage will remove the need for haplotig placement information, which has the potential to introduce errors. Moreover, it will simplify the workflow such that extended phasing is not required as an add-on module.

While FALCON-Phase corrects phase-switch errors between phase blocks in the FALCON-Unzip primary contigs, the current implementation of the method has some limitations. First, haplotypes in collapsed regions remain unresolved in the emitted haplotigs. These regions contain SNVs and structural variation of insufficient density to assign raw PacBio reads to a haplotype during the assembly process. Integration of Hi-C data during assembly (see above) can enhance phasing of PacBio reads in collapsed regions so that collapsed regions may be "unzipped" as well.

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

FALCON-Phase requires haplotig placement files to delimit phase blocks on the FALCON-Unzip primary contigs. Currently, haplotig placement file generation relies on sequence alignment whereas future releases of FALCON-Unzip will automatically generate haplotig placement files directly from the overlap graph. Errors in haplotig placement decrease the power of FALCON-Phase by combining phase blocks together in the same minced haplotig. This can result in conflicting signals in the Hi-C contact matrix. We anticipate performance improvements with the assembly-graph based haplotig placement file.

Acknowledgments

The authors would like to thank Mark Chaisson, Jason Chin, Luke Harmon, Jonas Korlach, Ryan Layer, Ivan Liachko, Ivan Sovic, Max Press, Billy Rowell, and Elizabeth Tseng for thoughtful discussion of the method. We are grateful to Kaylee Mueller for assistance with figures. Greg Concepcion tested the pipeline and provided useful feedback.

References

1. English AC, Richards S, Han Y, Wang M, Vee V, Qu J, et al. Mind the gap: upgrading genomes with Pacific Biosciences RS long-read sequencing technology. PloS one. 2012;7(11):e47768.

2. Gordon D, Huddleston J, Chaisson MJP, Hill CM, Kronenberg ZN, Munson KM, et al. Long-read sequence assembly of the gorilla genome. Science (New York, NY). 2016;352(6281):aae0344–aae0344.

3. Merker JD, Wenger AM, Sneddon T, Grove M, Zappala Z, Fresard L, et al. Long-read genome sequencing identifies causal structural variation in a Mendelian disease. Genetics in Medicine. 2018;20(1):159–163.

4. Sedlazeck FJ, Lee H, Darby CA, Schatz MC. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. Nature reviews Genetics. 2018;19(6):329–346.

5. Korlach J, Gedman G, Kingan SB, Chin CS, Howard JT, Audet JN, et al. De novo PacBio long-read and phased avian genome assemblies correct and add to reference genes generated with intermediate and short reads. GigaScience. 2017;6(10):1–16.

6. Koren S, Rhie A, Walenz BP, Dilthey AT, Bickhart DM, Kingan SB, et al. Complete assembly of parental haplotypes with trio binning. bioRxiv. 2018; p. 271486.

7. Chaisson MJP, Sanders AD, Zhao X, Malhotra A, Porubský D, Rausch T, et al. Multi-platform discovery of haplotype-resolved structural variation in human genomes. bioRxiv. 2017; p. 193144.

8. Bansal V, Bafna V. HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. Bioinformatics (Oxford, England). 2008;24(16):i153–i159.

9. Selvaraj S, Dixon JR, Bansal V, Ren B. Whole-genome haplotype reconstruction using proximity-ligation and shotgun sequencing. Nature biotechnology. 2013;31(12):1111–1118.

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

10. Patterson M, Marschall T, Pisanti N, van Iersel L, Stougie L, Klau GW, et al. WhatsHap: Weighted Haplotype Assembly for Future-Generation Sequencing Reads. Journal of computational biology : a journal of computational molecular cell biology. 2015;22(6):498–509.

11. Chin CS, Peluso P, Sedlazeck FJ, Nattestad M, Concepcion GT, Clum A, et al. Phased diploid genome assembly with single-molecule real-time sequencing. Nature methods. 2016;13(12):1050–1054.

12. Lieberman-Aiden E, van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, et al. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science (New York, NY). 2009;326(5950):289–293.

13. Edge P, Bafna V, Bansal V. HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Genome research. 2017;27(5):801–812.

14. Köster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. Bioinformatics (Oxford, England). 2012;28(19):2520–2522.

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

16. Marçais G, Delcher AL, Phillippy AM, Coston R, Salzberg SL, Zimin A. MUMmer4: A fast and versatile genome alignment system. PLoS computational biology. 2018;14(1):e1005944.

17. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. bioRxiv. 2013;.

18. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics (Oxford, England). 2009;25(16):2078–2079.

19. Bickhart DM, Rosen BD, Koren S, Sayre BL, Hastie AR, Chan S, et al. Single-molecule sequencing and chromatin conformation capture enable de novo reference assembly of the domestic goat genome. Nature genetics. 2017;49(4):643–650.

20. Burton JN, Adey A, Patwardhan RP, Qiu R, Kitzman JO, Shendure J. Chromosome-scale scaffolding of *de novo* genome assemblies based on chromatin interactions. Nature biotechnology. 2013;31(12):1119–1125.

21. Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing. bioRxiv. 2012;.

22. Merchant S, Wood DE, Salzberg SL. Unexpected cross-species contamination in genome sequencing projects. PeerJ. 2014;2(7):e675.

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

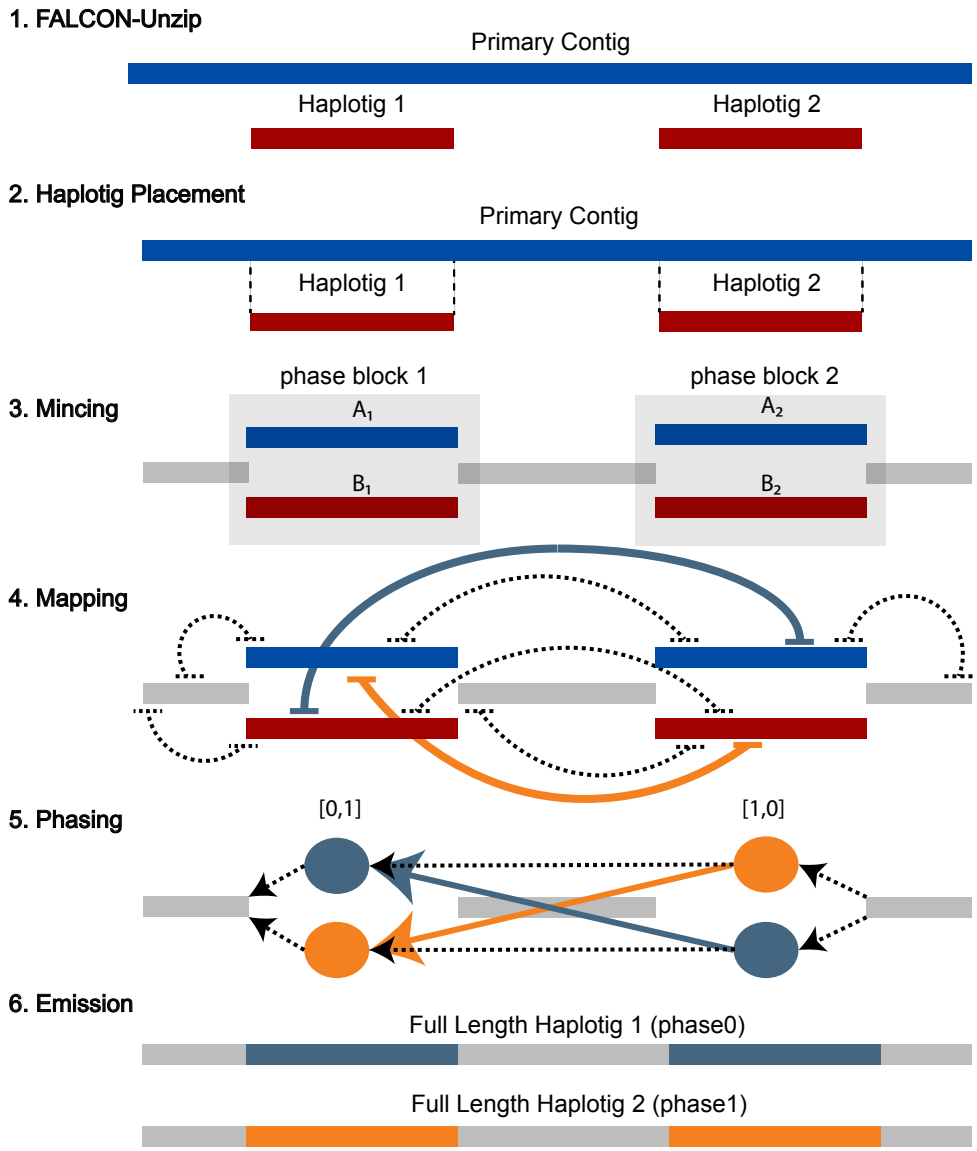

Fig 1. The FALCON-Phase Pipeline. Workflow example for a single FALCON-Unzip primary contig. **Step 1.** FALCON-Unzip assembly consisting of long primary contigs (blue) and shorter associated haplotigs (red). The region where a haplotig overlaps a primary contig is a phase block and is referred to as being "unzipped" because both haplotypes are resolved. Regions of the primary contig without associated haplotigs are referred to as "collapsed" haplotypes and have lower heterozygosity than unzipped regions. **Step 2.** A haplotig placement file specifies primary contig coordinates where the haplotigs align. **Step 3.** This placement file is used to "mince" the primary contigs at the haplotig alignment start and end coordinates. Mincing defines the "phase blocks" (A-B haplotig pairs, blue and red) and collapsed haplotypes (grey). **Step 4.** Hi-C pairs are mapped to the minced contigs and alignments are filtered. **Step 5.** Phase blocks are assigned to state 0 or 1 using algorithm 1. **Step 6.** The output of FALCON-Phase is two full length haplotigs for phase 0 and 1 of comparable length to the original FALCON-Unzip primary contig.

bioRxiv preprint doi: https://doi.org/10.1101/327064; this version posted May 21, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

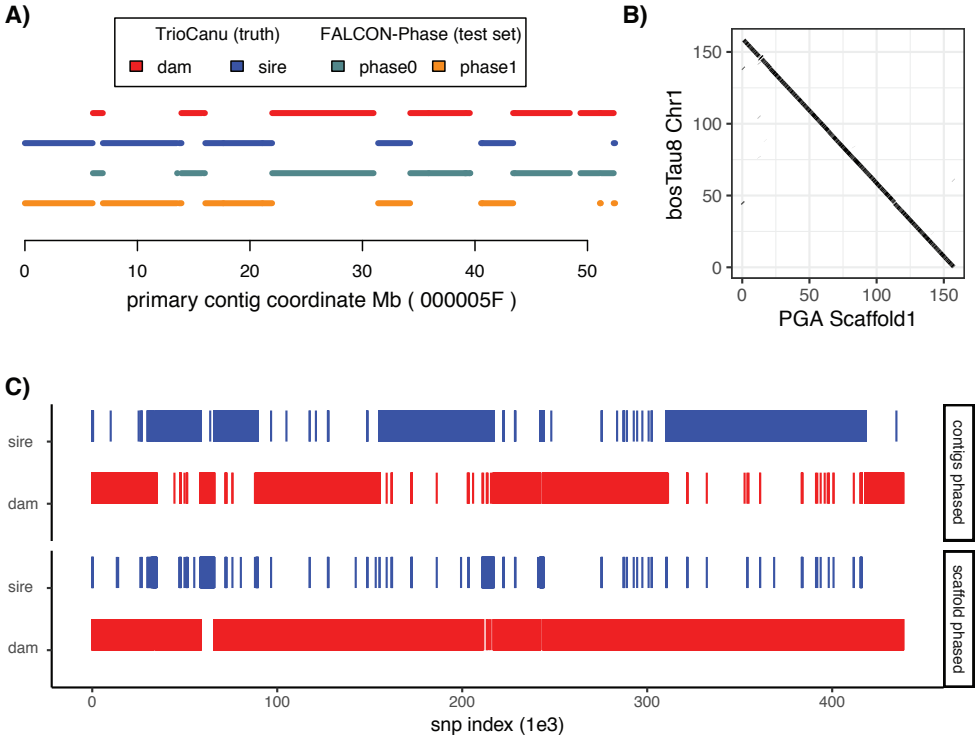

Fig 2. FALCON-Phase performance. A) Example of parent- (red and blue segments) and phase-assignment (orange and grey segments) for minced haplotigs along FALCON-Unzip primary contig 000005F. Phase0 corresponds to dam and phase1 corresponds to sire. This example shows the common phase switches between haplotype blocks of the FALCON-Unzip primary contigs as well as how phasing errors are more common for small haplotigs. **B)** Alignment plot showing the agreement between Proximity Guided Assembly (PGA), Hi-C based, scaffold one and chromosome one of bosTau8 [22]. **C)** Concordance between parental single nucleotide variants (SNV) and PGA scaffold1. Blue SNVs support dam and red SNVs support sire. Before phasing, Scaffold1 was a mixture of both dam and sire contigs. After phasing the majority of SNVs support the dam haplotype.