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

Subject Section

Lep-Anchor: Automated construction of linkage map anchored haploid genomes

Pasi Rastas (pasi.rastas@helsinki.fi)

Institute of Biotechnology, University of Helsinki, Finland

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Linkage mapping provides a practical way to anchor *de novo* genome assemblies into chromosomes and to detect chimeric or otherwise erroneous contigs. Such anchoring improves with higher numbers of markers and individuals, as long as the mapping software can handle all the information. Recent software Lep-MAP3 can robustly construct linkage maps for millions of genotyped markers and on thousands of individuals, providing optimal maps for genome anchoring. For such large data sets, automated and robust genome anchoring tool is especially valuable and can significantly reduce intensive computational and manual work involved.
Results: Here we present software Lep-Anchor to anchor genome assemblies automatically using dense linkage maps. As the main novelty, it takes into account the uncertainty of the linkage map positions caused by low recombination regions, cross type or poor mapping data quality. Furthermore, it can automatically detect and cut chimeric contigs, and use contig-contig, single read or alternative genome assembly alignments as additional information on contig order and orientations and to collapse haplotype contigs.
We demonstrate the performance of Lep-Anchor using real data and show that it outperforms ALLMAPS on anchoring completeness and speed. Accuracy-wise Lep-Anchor and ALLMAPS are about equal, but at the expense of lower completeness of ALLMAPS. The software Chromonomer was faster than the two other methods but has major limitations and is lower in accuracy. We also show that with additional information, such as contig-contig and read alignments, the anchoring completeness can be improved by up to 70% without significant loss in accuracy. Based on simulated data, we conclude that the anchoring accuracy can be improved by utilising information about map position uncertainty. Accuracy is the rate of contigs in correct orientation and completeness is the number contigs with inferred orientation.
Availability: Lep-Anchor is available with the source code under GNU general public license from http://sourceforge.net/projects/lep-anchor. All the scripts and code used to produce the reported results are included with Lep-Anchor.
Contact: pasi.rastas@helsinki.fi

1 Introduction

Advances in high-throughput sequencing and computational methods have made assembly of genome sequences *de novo* practical (Simpson and Pop, 2015). However, typically (*de novo*) assemblies contain assembly errors and are fragmented in many short contigs (or scaffolds) without information on how the sequences are physically located with respect to each other (Fierst, 2015; Simpson and Pop, 2015). When suitable mapping crosses and genetic marker data is available, linkage mapping provides independent information on locations and orientations of the sequences (Fierst, 2015).

Typical linkage map has a position in centiMorgans (cM, percentage of individuals recombining) for each genetic marker used to construct the map. This information can be used directly to position contigs into and within chromosome. If these map positions define a unique order and orientation of two adjacent contigs, these contigs can be scaffolded into longer sequence directly. If there is uncertainty in the local order of the contigs (e.g. all markers in both contigs have only one map position), the map information could be used as external evidence in local reassembly. The linkage map anchoring is a process where assembled genome sequences are put together by maximising the correlation of the physical (base pair) and the linkage map (cM) positions.

The number of individuals (offspring) in a mapping cross defines how many recombinations can be detected. To orient a contig, there must be at least two genotyped markers in it and at least one recombination between those markers. Moreover, each recombination can orient at most one contig. Even a mapping cross of less than 20 individuals can detect many assembly errors where distant parts are erroneously joined together (Rastas *et al.*, 2013). With more individuals, even more local errors can be detected and more contigs can be oriented and placed into chromosomes. As well as the number of individuals, the number of markers affects the map resolution. With too few markers, shorter contigs remain without any/proper linkage information, and some recombinations will be missed which reduces information on the contig orientation.

Low-coverage high-throughput whole genome sequencing has high potential in linkage mapping. It is a cost-efficient approach to obtain genotype information for millions of single nucleotide polymorphisms (SNPs) and thousands of individuals, pinpointing most recombinations within narrow regions in the genome, even for non-model species. However, the tools that are currently available for constructing linkage maps are not well suited for this many markers and even less so for low to medium coverage sequencing data. Recent linkage mapping software Lep-MAP3 (Rastas, 2017) can robustly construct linkage maps for millions of markers and thousands of individuals even from low coverage data. Moreover, Lep-MAP3 can output uncertainty in the linkage map positions.

The software Lep-Anchor has been developed to efficiently anchor genomes by using all the information provided by Lep-MAP3 and the additional information (for local reassembly) provided by read and contig-contig alignments.

1.1 Previous work

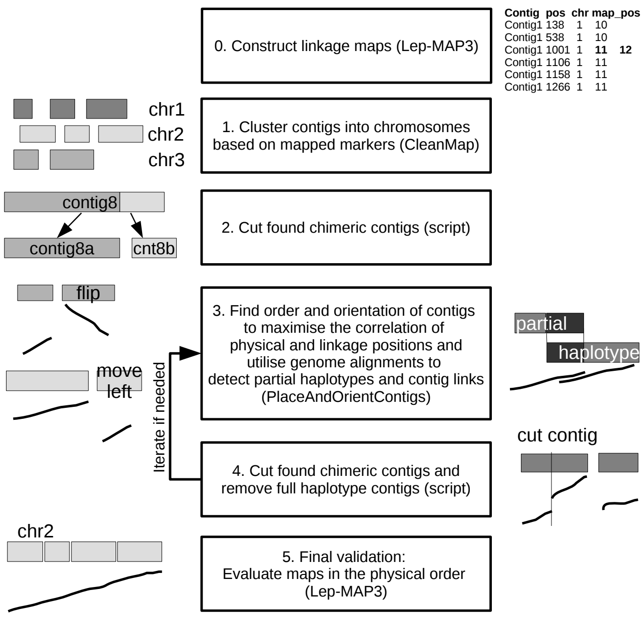
There are many genomes anchored into chromosomes using linkage maps, like yellow catfish (Tang *et al.*, 2015), red postman butterfly (Van Belleghem *et al.*, 2017) and the species mentioned in Fierst (2015).

There are also some software for integrating assemblies and linkage maps, such as ALLMAPS (Tang *et al.*, 2015), ArkMAP (Paterson and Law, 2013) and Chromonomer (http://catchenlab.life.illinois.edu/chromonomer/, Catchen (2015)). In this work, we compare the performance of Lep-Anchor, ALLMAPS (downloaded in August, 2019) and Chromonomer (version 1.08). The ArkMAP was not available for download (August, 2019).

Software ALLMAPS puts the contigs within each chromosome by maximising the number of supporting linkage map markers (length of the longest non-decreasing subset of markers). The same concept of marker support is used in Lep-Anchor by utilising map intervals describing the map uncertainty of each marker (see Fig. 3). The way Lep-Anchor puts the contigs into chromosomes is based on a hidden Markov model, similar to the model used in Lep-MAP (Rastas *et al.*, 2013). We do not know exactly how Chromonomer is achieving its anchoring. Based on its log files produced during our experiments, it seems to remove conflicting markers from the map until the map and genome are consistent.

We show in this article that Lep-Anchor improves upon these existing software. Based on our experiments, Lep-Anchor is very competitive on anchoring accuracy and completeness. Lep-Anchor is also fast, this computational efficiency is due to careful implementation of efficient algorithms.

As novel features, Lep-Anchor inputs map position intervals for each marker, taking into account the uncertainty in the map positions. Such uncertainty can occur due to low recombination regions (Fig. 2), cross type (e.g. multi-family data) or poor mapping data (genotype) quality. Moreover, Lep-Anchor also automatically finds and fixes chimeric contigs (locating in two or more chromosomes) and indicates possible within-chromosome chimerics for manual inspection and correction. Finally, Lep-Anchor can take as input contig-contig, related (species) assembly or single read alignments, and use them as additional evidence for contig order and orientation as well as to collapse haplotypic contigs. These haplotype contigs can be problematic in assemblies of highly heterozygous genomes (Huang *et al.*, 2017).


0. Construct linkage maps (Lep-MAP3)
chr1
chr2
chr3
1. Cluster contigs into chromosomes based on mapped markers (CleanMap)
contig8
contig8a
cnt8b
2. Cut found chimeric contigs (script)
flip
move left
Iterate if needed
3. Find order and orientation of contigs to maximise the correlation of physical and linkage positions and utilise genome alignments to detect partial haplotypes and contig links (PlaceAndOrientContigs)
partial
haplotype
cut contig
4. Cut found chimeric contigs and remove full haplotype contigs (script)
chr2
5. Final validation: Evaluate maps in the physical order (Lep-MAP3)

Contig	**pos**	**chr**	**map_pos**	
Contig1	138	1	10	
Contig1	538	1	10	
Contig1	1001	1	**11**	**12**
Contig1	1106	1	11	
Contig1	1158	1	11	
Contig1	1266	1	11	

Fig. 1. The data processing pipeline for Lep-Anchor. The main modules are CleanMap that assigns contigs into chromosomes and PlaceAndOrientContigs that orders and orients the contigs within each chromosome.

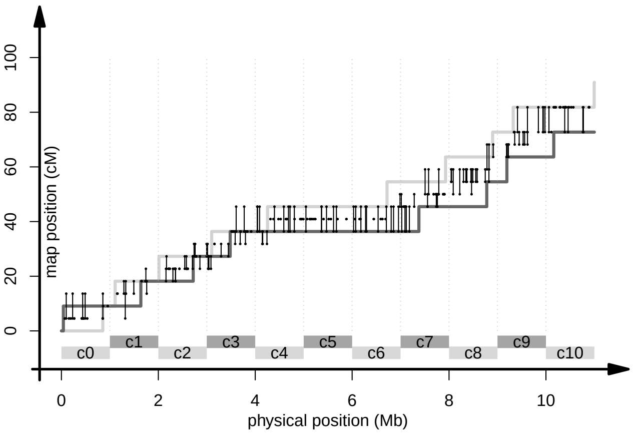


Fig. 2. Simulated data on F1 cross (two parents and their 11 offspring) and 200 markers. The piece-wise constant curves show the actual male (light grey) and female (dark grey) map positions as a function of the physical position (Marey map). The black vertical intervals show the marker positions and their uncertainty in the average map position based on simulated sequencing data with 20x average coverage. Due to markers informative differently in the parents, the intervals span several map positions. Such uncertainty cannot be reduced by improving genotype quality, and is more apparent in the regions of low recombination for one parent only (e.g. contigs c4-c6). The physical position consists of 11 contigs of length 1Mb (c0, . . . , c10) shown under the Marey map.

2 Methods

The Lep-Anchor (LA) workflow is illustrated in Fig. 1. This workflow consists of modules CleanMap and PlaceAndOrientContigs. CleanMap will only use the chromosome information of each marker, putting contigs into chromosomes, whereas PlaceAndOrientContigs puts the contigs into order and orientation within each chromosome.

2.1 CleanMap

CleanMap takes as input the genomic position (contig + position) and linkage group information of each marker. It uses an EM algorithm to find maximum likelihood parameters for a hidden Markov model on this input, in order to find likely (chimeric) contigs that belong to multiple chromosomes. The model has a state for each chromosome for every marker, and two parameters ϵ and τ for modelling errors in the map data (emission, accounting infrequent erroneous markers) and assembly errors (transition, change of chromosome), respectively. Thus, the model changes chromosome within a contig with probability τ and emits a different chromosome number from its state (label) with probability ϵ. The emission probability is scaled to cope with regions with high marker density (e.g. repeats or indels, on default regions with more than one marker per 100bp). Transition probability is defined on a fixed base-pair distance (default 1000bp), simply by adding states between adjacent markers with longer distance. The idea of CleanMap is similar to the ScaffoldHMM module in Lep-MAP (Rastas *et al.*, 2013), used to combine linkage groups of two linkage maps.

2.2 PlaceAndOrientContigs

The main part of LA is PlaceAndOrientContigs module. Its input consists of one or more linkage maps and the chromosomal assignment of contigs from CleanMap. Linkage maps can be given by listing the genomic position and the linkage position or position interval(s) (e.g. position ∈ [20, 23] cM). The position intervals can be obtained from Lep-MAP3 (Rastas, 2017) and allow Lep-Anchor to utilise information on marker position uncertainty. PlaceAndOrientContigs tries to find an order and orientations of contigs (anchoring) that is supported by the most markers. The concept of marker support is illustrated in Fig. 3.

Given a fixed anchoring, the number of markers supporting it can be calculated by dynamic programming in $O(mn)$ time, where m is the number of markers and n is the number of unique map positions.

Let $M_1, \ldots, M_m$ be the markers in the anchored physical order and $S(i,p)$ be the number of supporting markers among $M_1, \ldots, M_i$ with marker M_i set to position p. Dynamic programming for this measure can be formulated as

$$\begin{aligned} S(0,p) &= 0 \\ S(i+1,p) &= max_{q \le p} S(i,q) + score(M_{i+1},p) \end{aligned} \tag{1}$$

, where $score(M,p)$ is the score of marker M put to position p. Any score function could be used here, but we only consider the aforementioned position intervals, i.e. score is 1 if the position p is within the marker interval(s) and 0 otherwise. Finally, $S = max_p S(m,p)$ is the number of markers supporting this marker order and defines the score we want to optimise. By backtracking the path obtaining this score, one would obtain the non-decreasing piece-wise constant function fitting to the intervals. This function could be used, e.g. to estimate recombination rate.

The algorithm for finding the orientation starts from some order and orientation of contigs (defines the marker order). Then each contig is tested (in random order) whether the score improves if the contig is reversed and/or moved within the current anchoring. Each such improvement is accepted and, when all contigs are tried without improvement, the algorithm terminates. All possible positions and orientations for k contigs can be tested in $O(kmn)$ time with a typical forward-backward type algorithm.

2.2.1 Handling multiple maps

In order to use multiple maps in LA, their relative orientation is required (so that linkage and physical positions are positively correlated). LA finds such orientation as follows. First the anchoring is done for the first map only in forward orientation starting from a random order of contigs (or from sorted order based on most abundant marker position of each contig). Then maps are added one by one in the input order by evaluating the current anchoring for the new map in both orientations and taking the orientation with higher score. Before adding the next map, the anchoring is improved by using the newly added map in the chosen orientation.

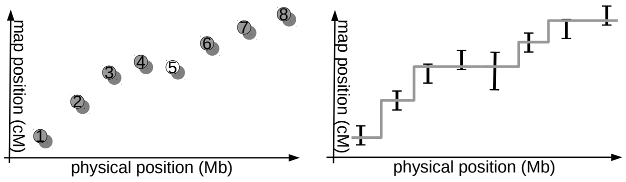


Fig. 3. The number of markers that are consistent with the physical and linkage positions defines the supporting score to optimise in anchoring. The largest non-decreasing subset of markers can be used when the map is given as plain cM positions (left, used in ALLMAPS). With position intervals (right, Lep-Anchor), a non-decreasing piece-wise constant function is fitted to the intervals. In this hypothetical case, the support score is 7 for positions but 8 for intervals. The longest non-decreasing subset of markers can be computed asymptotically in $O(m\log(m))$ time for m markers (Fredman, 1974). The algorithm that Lep-Anchor uses for intervals is typically somewhat slower, taking $O(mn)$ time where n is the number of unique map positions, see Section 2.2 for more details.

2.2.2 Using additional data to link contigs

PlaceAndOrientContigs can use contig-contig and read alignment data in addition to linkage maps to decide the orientation and placement of contigs. This is accomplished by defining the anchoring score as the supporting markers S plus the alignment score between each adjacent contigs in the anchoring.

The input for contig-contig alignments is taken in as UCSC liftover chain file (Kent *et al.*, 2003). This chain can be generated by the first two step of HaploMerger2 (Huang *et al.*, 2017) pipeline (hm.batchA1.initiation_and_all_lastz + hm.batchA2.chainNet_and_netToMaf) on a repeat masked reference genome. The chain is used to lift the markers of a collapsed region to the remaining region. The additional score is defined by calculating the number of supporting markers after the liftover and adding one (+1) for each 1kb of aligning DNA and subtracting one (-1) for each kb in alignment gaps. If the score is positive between adjacent contigs, the partial haplotype region is automatically collapsed. At the end, PlaceAndOrientContigs also calculates and reports the score of full haplotypes and these contigs can be manually removed after the first run.

Raw sequencing reads (or contigs from a related species or individuals) can also be used to link contigs. Each read linking two adjacent contigs adds one to the anchoring score. Reads are only used if the contig-contig alignment score is not positive for the corresponding adjacent contigs. The input for read mappings is a minimap2 (Li, 2018) paf file.

The changes in the anchoring search algorithm are minimal when taking into account these additional scores; the linkage map and the alignment scores are added up and, instead of moving and orienting only individual contigs, the update step tries to move and orient chains of adjacent contigs with positive adjacent scores.

3 Results

We tested the performance of LA, ALLMAPS and Chromonomer using yellow catfish (*Tachysurus fulvidraco*) and red postman butterfly (*Heliconius erato*) data. The former consist of four linkage maps and a scaffold level assembly coming with ALLMAPS (Tang *et al.*, 2015). Recently, a more complete genome was published for this species (Gong *et al.*, 2018) and was considered here as the ground truth (GT). For the red postman, we used the original assembly (Van Belleghem *et al.*, 2017) as the GT and the corresponding map data from Rastas (2017). All experiments were run using a single core of a normal desktop computer with i7-7700 CPU @ 3.60GHz and 32Gb of memory.

We discuss in more detail the results of the yellow catfish analyses. However, Chromonomer can only utilise one of the four input maps and performed poorly. As its inferior results are mostly due to having too few markers, we do not discuss more on its performance on the yellow catfish data.

First, we notice that LA reports five chimeric scaffolds for the yellow catfish data (scaffolds 44, 75, 123, 165 and 230), whereas ALLMAPS on default puts all contigs into at most one chromosome. The mapping of the two genomes against each other with minimap2 (Li, 2018) supports all the chimerics found by LA. The start of scaffold 230 maps to unanchored part of GT (suggesting that GT could be improved with this linkage map) and the end to chromosome 23. The other four chimerics map to different chromosomes consistent with the marker positions in the LAs result. We tried the option split in ALLMAPS, meant to cut chimeric scaffolds: It reported 47 breakpoints within scaffolds. One of the ALLMAPS breakpoint-scaffolds, scaffold 44, was also reported by LA but the reported breakpoint was not the same. Thus, all the five verified chimerics were missed by ALLMAPS.

Second, we compared the score (number of supporting markers) of the two programs. LA reports the final result with the total of 6061 supporting markers summing up all 26 chromosomes, whereas ALLMAPS reports 5979 supporting markers. We verified the support by evaluating ALLMAPS results with LA. According to LA, ALLMAPS result had 6019 supporting markers counted by LA: in 10 chromosomes, the scores were equal and in the remaining 16 chromosomes, the score calculated by LA was greater. The difference is most likely due to ALLMAPS removing 225 markers (2.3%) as outliers.

Third, the number of scaffolds with known orientation were 520 and 493, with LA and ALLMAPS, respectively, whereas the total number of scaffolds were 940 and 929, respectively. We mapped the scaffolds to the GT and calculated orientation for 1724 scaffolds that had at least two mappings to a chromosome and at least 80% of mappings showed consistent orientation. In LA results, 69.4% of the comparable scaffolds (279 out of 402) had a consistent orientation, whereas in ALLMAPS results 68.3% (259 out of 379) were consistent. The scaffolds with consistent orientation accounted 82.8% (of 440 Mb) and 82.0% (of 426 Mb) of the total length of the scaffolds, respectively for LA and ALLMAPS (LA had 15Mb more in correct orientation). The higher correctness in the total length indicates that the shortest scaffolds were more likely to be in wrong orientation or that the mapping of short scaffolds is not reliable.

Lastly, we used contig-contig alignments as input for the LA. We constructed the alignment chain file with the Haplomerger2 pipeline (Huang *et al.*, 2017) using the scaffold level assembly as input. With the chain we could find partial haplotypes and utilise unused scaffolds in the anchoring. Including this extra information, 563 scaffolds had an orientation (and 34 additional scaffolds had an orientation relative to some unoriented scaffold). From these, we could verify 431 scaffolds and 301 (69.8%) were in correct orientation, covering 83.5% (of 445Mb) of the total length.

The somewhat poor performance of both LA and ALLMAPS on the yellow catfish data might be due to the linkage map, especially due to the low marker density. Unfortunately, the raw linkage map data are not available so we cannot verify this. Instead, we did similar comparisons on the red postman maps generated with Lep-MAP3. We took all male-informative markers (3.2M) as input for LA and only markers without uncertainty in the position (position interval contains only one position) and consistent with the chromosomal assignment from CleanMap (3.0M markers, 4.3% discarded) for ALLMAPS and Chromonomer. We run LA with the map data only and then by including alternative genome assemblies (minimap2), contig-contig alignments (Haplomerger2) and raw PacBio reads (minimap2).

catfish	LA with chain	LA	ALLMAPS	Chromonomer
runtime (min)	2	**1**	24	**<1 ***
scaffolds	**940**	**940**	929	679*
in orientation	**563**	520	493	NA
consistent	**69.8%** **301** of 431	69.4% 279 of 402	68.3% 259 of 379	68.6%** 236 of 344
consistent (Mb)	**83.5%** **372** of 445	82.9% 364 of 440	82.0% 350 of 426	75.3%** 317 of 421
butterfly				
runtime (hours)	4	4	195	**0.04**
contigs	**1235**	**1235**	1216	1233
in orientation	**811**	474	116	NA
consistent	98.5% **764** of 776	98.9% 460 of 465	**100%** 111 of 111	97.8%** 437 of 447
consistent (Mb)	99.5% **379** of 382	99.6% 297 of 298	**100%** 109 of 109	98.1%** 270 of 276

Table 1. Comparison of Lep-Anchor, ALLMAPS and Chromonomer on yellow catfish and red postman butterfly data. The best result is in **bold**. * = Chromonomer only accepts one input map so it was run only on the first map of the yellow catfish data. ** = Chromonomer outputs orientation for all contigs even without any data to infer this, so we evaluated correctness only for the contigs oriented by LA. Note that only LA has split contigs into multiple chromosomes, other software have put each contig into at most one chromosome

We immediately notice that all the results contain about the same number of contigs, but ALLMAPS provides orientation information for far fewer contigs (about 25% of LA). Investigating this further, it seems that ALLMAPS rarely outputs an orientation for contigs with only one recombination (two map positions). However, there are contigs with one recombination that are orientated with ALLMAPS.

Chromonomer was much faster on this data (< 3 min) than LA or ALLMAPS. However, it gives an orientation for all contigs, not just ones for which there is map information. The orientation for the contigs without map information seemed random (about same number of contigs in + and - orientation). This makes it difficult to compare its results to the ones with ALLMAPS and LA as by random assignment 50% contigs will be in correct orientation. To obtain comparable results, we only took the contigs for which LA gave an orientation.

Finally, using additional alignments with LA, the number of oriented contigs almost doubles and about 180 contigs contain orientation information relative to some other unoriented contig. All the results are summarised in Table 1.

We also tested the software using simulated data. We simulated small linkage map data according to F1 cross with 11 offspring over 11Mb chromosome consisting of 11 contigs of length 1Mb in forward (+) orientation. There was exactly one male and female recombination within each contig except for contigs c4, c5 and c6. Contig c5 had no recombinations and contig c4 and c6 had no female recombinations. We simulated 50, 100 and 200 genetic markers from these contigs with varying average sequencing depth of 5, 10 or 20 per individual and each marker being informative in male, female or both with equal likely. Simulated reads were converted to genotype likelihoods by assuming a fixed read-error rate of 1%. The Marey map, contigs and 200 markers with position intervals from this simulation are shown in Fig. 2. Note that this simulation does not contain any information about the orientation of c5, other contigs can be orientated if there are enough markers (informative markers at informative positions).

markers (coverage)	LAi +/-/?	LA+ +/-/?	LAs +/-/?	AM +/-/?	AM+ +/-/?	Chromonomer +/-/?
50 (5x)	**3/0/8**	**4/1/6**	4.2/1.9/4.9	2.2/1.6/7.2	3/2/6	4.3/1.8/4.9**
100 (5x)	**8/0/3**	7/0/4	5.4/1.1/4.5	5.1/1.0/4.9	9/2/0	4.9/1.6/4.5**
200 (5x)	9/0/2	**10/0/1**	8.7/0.6/1.7	5.0/0.6/5.4	8/0/3	9.0/0.3/1.7**
50 (10x)	3/0/8	4/0/7	4.4/0.7/5.9	3.9/0.2/6.9	**7/0/4**	4.5/0.6/5.9**
100 (10x)	6/0/5	**8/0/3**	7.2/0.8/3.0	3.1/0.6/7.3	**8/0/3**	7.1/0.9/3.0**
200 (10x)	**10/0/1**	9/0/2	8.4/0.2/2.4	3.6/0.0/7.4	7/0/4	8.5/0.1/2.4**
50 (20x)	4/0/7	5/0/6	4.8/1.3/4.9	4.2/1.0/5.8	**7/1/3**	4.7/1.4/4.9**
100 (20x)	6/0/5	**9/0/2**	6.8/0.6/3.6	3.9/0.4/6.7	8/0/3	7.1/0.3/3.6**
200 (20x)	**10/0/1**	**10/0/1**	8.7/0.4/1.9	4.3/0.1/6.6	7/0/4	8.9/0.2/1.9**

Table 2. Comparison of Lep-Anchor, ALLMAPS (AM) and Chromonomer on simulated data. Results for LAs, AM and Chromonomer are averages over 10 independent samplings (explained in the text). Lep-Anchor was run on linkage position intervals (LAi), with sampled map positions (LAs) and with all 10 sampled datasets together (LA+). ALLMAPS was run on sampled positions and with all 10 samples together (AM+). Chromonomer was run on the 10 samples only (due to lack of multi-map support). The numbers are contigs in forward (+), backward (-) and unknown (?) orientation. All contigs should be in forward orientation, but the data is not sufficient to orient contig c5, thus the best result is 10/0/1. We rank the solutions by the highest number of "correct - incorrect", e.g. 10/0/1 yields 10-0=10 and 8/2/1 8-2=6. For LA+ we required a support of 5 markers as each marker is multiple times in the dataset, for ALLMAPS we could not control its sensitivity in this way. The best result is in **bold**. ** = Chromonomer outputs orientation for all contigs even without any data to infer this, so we evaluated correctness only for the contigs oriented by LAs (supported by one or more markers).

The results on simulated data are shown in Table 2. To obtain the map, we run Lep-MAP3 on the data in the correct marker order and took out the linkage position intervals for each marker. We run LA directly with these intervals (LAi in Table 2), and in order to test other software and the effect of using intervals, we sampled one map position from each interval randomly 10 times. We also tried to use only markers without any uncertainty in the map position, but then many contigs would be left without any markers (over 50% of markers were discarded). Each of the sampled dataset was run with LA (LAs in Table 2), ALLMAPS (AM) and Chromonomer. With LA and ALLMAPS we could also run all 10 sampled datasets together as 10 families (LA+ and ALLMAPS+).

From these results we conclude that using marker intervals reduces errors in the anchoring. Sometimes the sampled map positions yield better result but this could be due to stochasticity of the sampling. On the sampled datasets, the results of Chromonomer and LAs are about equal in accuracy, whereas ALLMAPS performs somewhat worse. By combining all 10 samples, the performance of ALLMAPS gets better and the error rate is reduced for LA. However, the conclusions is that the use of marker intervals in LA is a reasonable approach, and only small improvements could be achieved by studying the optimal combination of sampling and the marker interval approach in more detail. Note that the map uncertainty is higher in these simulations compared to the real maps used in this article, the latter containing only male or female informative markers.

4 Discussion and Conclusion

By using two real data sets, we have demonstrated that Lep-Anchor outperforms ALLMAPS and Chromonomer in genome assembly anchoring. Lep-Anchor produces greater numbers of contigs/scaffolds with accurate orientation information than the other available software. Moreover, Lep-Anchor is able to incorporate additional alignment data with the linkage map and thus obtain even more complete anchoring.

Lep-Achor is almost 50 times faster than ALLMAPS, making it more practical for larger datasets. Chromonomer is even faster but is more limited in its input and output.

Acknowledgements

The author would like to thank Juhana Kammonen, Panu Somervuo and Ari Löytynoja for valuable comments on this manuscript.

Funding

The author has been funded by the Jane and Aatos Erkko Foundation.

Conflict of Interest: none declared.

References

Catchen, J. (2015). Chromonomer. Available online. Accessed: 2019-08-01.

Fierst, J. (2015). Using linkage maps to correct and scaffold de novo genome assemblies: methods, challenges, and computational tools. *Frontiers in Genetics*, **6**(220).

Fredman, M. (1975). On computing the length of longest increasing subsequences. *Discrete Mathematics*, **11**, 29–35.

Gong, G., Dan, C., Xiao, S., Guo, W., Huang, P., Xiong, Y., Wu, J., He, Y., Zhang, J., Li, X., Chen, N., Gui, J.-F., and Mei, J. (2018). Chromosomal-level assembly of yellow catfish genome using third-generation DNA sequencing and Hi-C analysis. *GigaScience*, **7**(11).

Huang, S., Kang, M., and Xu, A. (2017). HaploMerger2: rebuilding both haploid sub-assemblies from high-heterozygosity diploid genome assembly. *Bioinformatics*, **33**(16), 2577–2579.

Kent, W. J., Baertsch, R., Hinrichs, A., Miller, W., and Haussler, D. (2003). Evolution's cauldron: Duplication, deletion, and rearrangement in the mouse and human genomes. *Proceedings of the National Academy of Sciences*, **100**(20), 11484–11489.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, **34**(18), 3094–3100.

Paterson, T. and Law, A. (2013). Arkmap: integrating genomic maps across species and data sources. *BMC Bioinformatics*, **14**(1), 1–10.

Rastas, P. (2017). Lep-MAP3: robust linkage mapping even for low-coverage whole genome sequencing data. *Bioinformatics*, **33**(23), 3726–3732.

Rastas, P., Paulin, L., Hanski, I., Lehtonen, R., and Auvinen, P. (2013). Lep-map: fast and accurate linkage map construction for large snp datasets. *Bioinformatics*, **29**(24), 3128–3134.

Simpson, J. T. and Pop, M. (2015). The theory and practice of genome sequence assembly. *Annual Review of Genomics and Human Genetics*, **16**(1), 153–172.

Tang, H., Zhang, X., Miao, C., Zhang, J., Ming, R., Schnable, J. C., Schnable, P. S., Lyons, E., and Lu, J. (2015). Allmaps: robust scaffold ordering based on multiple maps. *Genome Biology*, **16**(1).

Van Belleghem, S., Rastas, P., and *et al.* (2017). Complex modular architecture around a simple toolkit of wing pattern genes. *Nature Ecology & Evolution*, **1**, 0052.