

Decoding Synteny Blocks and Large-Scale Duplications in Mammalian and Plant Genomes

Qian Peng^{1,*}, Max A. Alekseyev³, Glenn Tesler², and Pavel A. Pevzner¹

¹ Department of Computer Science and Engineering
qpeng@cs.ucsd.edu

² Department of Mathematics
University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093

³ Department of Computer Science and Engineering
University of South Carolina, 315 Main St., Columbia, SC 29208

Abstract. The existing synteny block reconstruction algorithms use *anchors* (e.g., orthologous genes) shared over *all* genomes to construct the synteny blocks for multiple genomes. This approach, while efficient for a few genomes, cannot be scaled to address the need to construct synteny blocks in many mammalian genomes that are currently being sequenced. The problem is that the number of anchors shared among *all* genomes quickly decreases with the increase in the number of genomes. Another problem is that many genomes (plant genomes in particular) had extensive duplications, which makes decoding of genomic architecture and rearrangement analysis in plants difficult. The existing synteny block generation algorithms in plants do not address the issue of generating *non-overlapping* synteny blocks suitable for analyzing rearrangements and evolution history of duplications. We present a new algorithm based on the *A-Bruijn* graph framework that overcomes these difficulties and provides a unified approach to synteny block reconstruction for multiple genomes, and for genomes with large duplications.

Supplementary material: <http://grimm.ucsd.edu/ABS>

1 Introduction

Plant genomes exhibit an unusually large proportion of duplicated regions [1]. The large number of duplications makes decoding of genomic architecture and rearrangement analysis in plants difficult. In particular, segmental duplications represent a major obstacle to reconstruction of *synteny blocks* (i.e., conserved regions across the genomes), resulting in relatively few published results on synteny blocks in plant genomes as compared to vertebrate genomes (and especially to mammalian genomes) where segmental duplications are less prevalent and can therefore be largely ignored while constructing synteny blocks.¹ It is estimated that segmental duplications account only for less than 10% of the human

* Corresponding author.

¹ Segmental duplications in the human genome are usually represented as a set of pairwise alignments and are masked out by synteny block generation algorithms.

genome [2] and 2.9% of the mouse genome [3]. By contrast, in plant genomes, duplications are prevalent (they account for more than 70% of the *Arabidopsis thaliana* genome [4]), and ignoring duplicated regions would render a synteny block analysis meaningless. This represents an intrinsic difficulty in constructing syntenies in plant genomes.

From an algorithmic perspective, the problems of finding synteny blocks between two genomes and duplicated blocks are very similar. In fact, finding synteny blocks of multiple genomes can be converted into the problem of finding duplicated (or multi-copy) blocks within a single genome by concatenating the multiple genomes in an arbitrary order. In the past, this problem of reconstructing synteny blocks in k mammalian genomes was addressed by constructing k -way anchors shared between all genomes [5]. However, this approach is limited to small k since with the growing number of genomes, the number of k -way anchors sharply decreases. The disappearing k -way anchors may lead to disappearing synteny blocks. Short synteny blocks (which are important in studies of chromosome evolution [5,6]) are particularly vulnerable to this effect. In this paper, we propose a unified approach to synteny block reconstruction for two or multiple genomes, synteny block reconstruction for genomes with large duplications, and duplicated block reconstruction within a genome.

A typical synteny block generation algorithm takes as an input a set of *alignment anchors* (i.e., local alignments or pairs of similar genes) between two genomes (or two copies of the same genome) and outputs a set of synteny blocks (or duplicated blocks) that cover (without overlaps) most of each participating genome. As a result, each genome is represented as a shuffled sequence of the constructed synteny blocks that enables further rearrangement analysis of the genomes (e.g., computing the rearrangement distance between them). For two genomes, most existing synteny blocks generation algorithms employ a 2-dimensional *genomic dot-plot* where two genomes (or two copies of the same genome) are placed along the axes on the plane and their alignment anchors are represented as dots (Fig. 1(a)). These algorithms further decompose the dot-plot into a collection of “long” diagonal-like segments constituting *2-D synteny blocks* (Fig. 1(b)). The conventional (1-D) synteny blocks for each genome can be obtained as projections of the 2-D synteny blocks onto a corresponding axis (Fig. 1(b)). The notions of 2-dimensional dot-plots and synteny blocks generalize to k -dimensions when there are k genomes. This simple description hides a number of computational details that make the problem of synteny block generation non-trivial [7,8].

Nadeau *et al.* [10] introduced the notion of *conserved segments*. Waterston *et al.* [11] and Pevzner *et al.* [7] described two approaches to synteny block generation, that produce similar results. There are many other studies describing different methods of “synteny block” generation [12,13,14,15,16,17,18]. While these approaches proved to be adequate for small sets of mammalian genomes,² and in some cases prokaryotic genomes, they do not particularly address issues that stem

² Since the number of duplications in mammalian genomes is small, the 2-D synteny blocks usually do not overlap in 1-D.

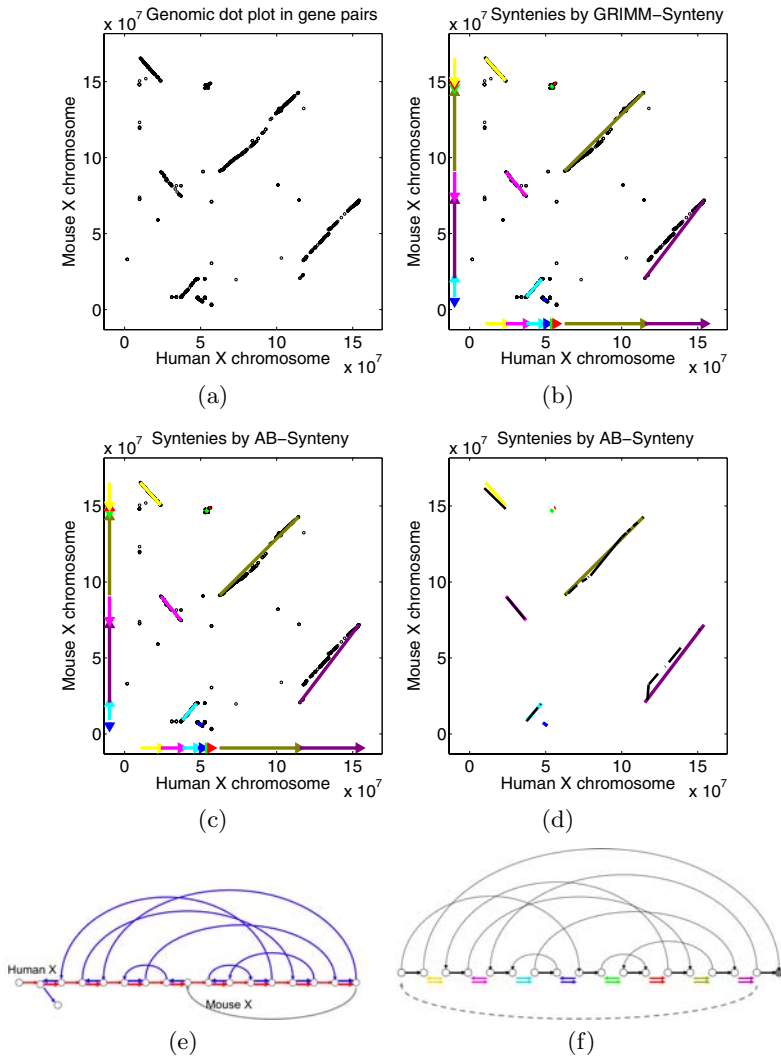


Fig. 1. (a) Genomic dot-plot between human (x axis) and mouse (y axis) X chromosomes for all 714 orthologous gene pairs in Ensembl. Each dot represents a pair of “similar” genes between the two species. (b,c) Synteny blocks in 2-D (diagonals) and 1-D (bar with arrow) produced by (b) GRIMM-Synteny [9]. (c) AB-Synteny. Each color represents a synteny (duplication on the concatenated genome). (d) Gene pairs removed. Black diagonals are the human-mouse part of the human-mouse-rat-dog-chicken 5-way synteny. (e) A-Bruijn graph after simplifications. The bold red and blue edges correspond to the synteny in (c). In the A-Bruijn graph, each pair of red/blue edges is a single edge of multiplicity 2 (illustrated as parallel edges). The black edge concatenates the human and mouse chromosomes. (f) Redraw of (e) with synteny represented in different colors. An end node (shaded) is added. The A-Bruijn graph is equivalent to the breakpoint graph if edges representing synteny (colored edges) and the edge transitioning from mouse to human (dashed arc) are removed.

from extensive duplications in plant genomes. In the presence of duplications, the 2-D synteny blocks may overlap in 1-D, i.e., along one of the genomes. There are a few previous efforts to generate synteny blocks for genomes with large duplications [4,19,20,21,22,23,24,25]. They do not directly address the issue of generating *non-overlapping synteny blocks in 1-D* either, which are more suitable for analyzing rearrangements and evolution history of duplications.

Pevzner *et al.* [26] introduced the *A-Bruijn graph* approach to repeat classification, and the approach was later found useful in other problems [27,28,29]. In this paper we demonstrate that the A-Bruijn graph framework can be also applied to the problem of synteny block generation for genomes with large duplications. Our algorithm produces non-overlapping synteny blocks in both 2-D and 1-D representations. By simply concatenating multiple genomes, we can generalize this approach to synteny block generation for multiple genomes. Previous efforts to generate synteny blocks for k genomes often required k -way alignment anchors, e.g., orthologous genes present in all k genomes [30]. While recent approaches [16,31] allow missing anchors, they pose other constraints. Our approach is not subject to constraints such as all-vs-all alignment as it uses pairwise anchors as an input, nor does it require a reference genome.

We benchmarked our algorithm on five vertebrate genomes to reconstruct 5-way syntenies, and on the plant genome *A. thaliana* to find duplicated blocks. We compared the results to the published syntenies or duplicated blocks. While there is no gold standard to what constitutes “correct” synteny blocks, all synteny block generation algorithms are parameter-dependent and may produce different synteny blocks on the same input data. To evaluate the performance of synteny block generation algorithms, we simulated genomes with large duplications and known synteny blocks and analyzed how well our algorithm reconstructs the underlined synteny blocks (see supplemental material).

2 Approach

Fig. 2(a) shows a hypothetical sequence of genes, resulting from multiple segmental duplications. In reality, we are given only the resulting genomic sequence and know nothing about the structure of its segments (marked by the colors in Fig. 2(a)). It is natural to ask what evolutionary events (including rearrangements and duplications) created the given genomic sequence. Before answering this question, we need to understand the duplication structure of the given genome, i.e., to represent it as a sequence of non-overlapping blocks, each of which may appear one or more times.

The diagonals in Fig. 2(b) are what conventional synteny block construction methods would produce as synteny blocks from the genomic dot-plot of a genome against itself. Since these blocks overlap along the sequence, the duplication structure is unclear. Ideally, we would like to see diagonals that do not overlap along the sequence (Fig. 2(c)). One natural approach is for every pair of partially overlapped blocks along each axis to cut the overlapping region off these blocks into two new entirely overlapping blocks. As newly created blocks

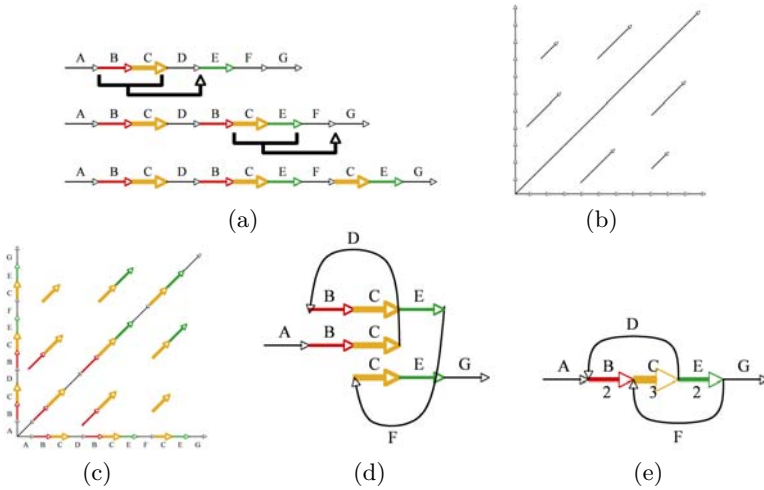


Fig. 2. (a) Hypothetical sequence with multiple duplications. (b,c) Genomic dot-plot and the resulting synteny blocks of the sequence. The 2-D representations overlap in 1-D. (d) Generate A-Bruijn graph of the sequence. (e) A-Bruijn graph of the sequence. Edges with multiplicity greater than 1 are synteny blocks. Blocks B, E each have two copies, and C has three copies. The algorithm outputs B,C,E as separate paths/blocks.

may partially overlap with other blocks, to eliminate all such partial overlaps a number of subsequent cuts may be required. The problem with such an approach, however, is that in some cases the initial synteny blocks might result in the iterative fragmenting and shrinking of synteny blocks. While this phenomenon is well known in repeat classification (e.g., RECON algorithm [32] follows a similar scheme), it has not been addressed yet in synteny block reconstruction. This simple and seemingly sensible approach does not work well in complex cases [26]. For example, early attempts to use a similar approach for constructing “duplication subunits” (analogs of synteny blocks for segmental duplications) failed and elaborate techniques were used to resolve this challenge [33]. While the complexity of synteny reconstruction so far is nowhere close to the complexity of the repeat analysis, the addition of every new species will soon make the synteny reconstruction more difficult, thus calling for techniques to overcome the limits of tools based on iterative splitting. In addition, synteny blocks (different from repeats) are subject to microrearrangements, further complicating the problem.

Although repeats and duplicated synteny blocks result from different biological events, they both represent sub-sequences appearing multiple times in the genomes. Repeats and duplicated synteny blocks differ mostly in length and in the number of occurrences in the genomes. Therefore, the problem of constructing non-overlapping synteny blocks for genomes with duplications is similar to the problem of *de novo* repeat classification and can be solved accordingly.

The same approach can be used for generation of synteny blocks across multiple genomes by simply concatenating them into a single genome. If there are no

duplications in the original genomes, a k -copy synteny block in the concatenated genome corresponds to a k -way synteny block of the original genomes.

AB-Synteny

Without loss of generality we will generate synteny blocks for a single genome. Suppose that the given genome is represented as a sequence of elements (base pairs or genes) v_1, v_2, \dots, v_n . These elements form the vertices of a *path-graph* P where every pair of consecutive vertices v_i and v_{i+1} (for $i = 1, \dots, n - 1$) are connected with a directed edge. To obtain an A-Bruijn graph [26] A from the graph P , one needs to “glue” all vertices of P belonging to the same anchor into a single vertex (Fig. 2(d)). The resulting A-Bruijn graph A inherits all edges from the path-graph P , counting multiplicity of each edge as its weight (hence, the edges in A are weighted and there are no parallel edges (Fig. 2(e)).

The A-Bruijn graph A has one source and one sink³ such that the original genome can be read along some path from the source to the sink. Every edge with weight greater than one corresponds to a *syntenic region* (i.e., a region that may belong to at most one synteny block), and its weight gives the number of copies of this syntenic region in the genome.

Unfortunately, such an interpretation of the A-Bruijn graph meets a number of obstacles. Inconsistencies in alignments and tandem duplications may create *whirls*⁴ in A , while gaps in alignments may create *bulges* in A [26]. As a result, the constructed A-Bruijn graph can be exceedingly complicated. For example, the A-Bruijn graph constructed from *Arabidopsis* gene pairs has 6394 vertices and 12761 edges. To overcome these difficulties, [26] suggested a heuristic routine simplifying an A-Bruijn graph, which we partially apply to the graph A . In the process, we simplify the A-Bruijn graph by substituting every simple path in the graph by a single edge with its length equal to the length of the path.

Overall, our synteny block generation algorithm AB-Synteny(G, C, g, B, L) has the following parameters: G and C are gap and block size thresholds in pre-processing to eliminate noisy anchors; the *girth* g specifies a distance threshold for removing whirls; B specifies the threshold for bulge removal; L is the block size threshold (minimum number of elements in the blocks):

1. For two or more genomes, concatenate all genomes forming a single genome.
2. Pre-processing: run GRIMM-Synteny(G, C) [9] to produce non-overlapping syntenic blocks in 2-D (blocks may overlap in 1-D). GRIMM-Synteny removes all anchors within “small” blocks (smaller than C).
3. Construct an A-Bruijn graph: run A-Bruijn(g, B)—a simpler version of the graph clean up routines detailed in [26]—on the remaining anchors, removing whirls shorter than g and simple bulges shorter than B .
4. Output non-overlapping paths whose multiplicities are greater than one and whose lengths are equal or greater than L . These are syntenic regions.

³ In practice, it is convenient to have also an inverted sequence of the same genome (for reverse DNA strand), in which case there are two sources and two sinks.

⁴ Whirls refer to short directed cycles and bulges short undirected cycles in A .

5. Post-processing: merge neighboring syntenic regions of same orientation interrupted by short gaps into syntenies. Assign each block a unique ID.

We remark that since the constructed paths (syntenic regions) do not overlap in the A-Bruijn graph, they also do not overlap in 1-D (both before and after the post-processing step). As a result, AB-Syntenity produces a number of syntenity blocks non-overlapping in 1-D and a representation of the given genome as a mosaic of these blocks (each block may appear in multiple copies). In other words, an entire genome is represented as a *word* over the alphabet of syntenity blocks, which facilitates further duplication and rearrangement studies.

3 Syntenic Analysis of Vertebrate Genomes

As an illustration and a validation of the AB-Syntenity algorithm, we extracted and analyzed 714 gene pairs between human and mouse X chromosomes from Ensembl database version 39. The gene pairs are described as “orthologs” by Ensembl. After highly repetitive gene pairs (present in ≥ 10 copies) are removed (as they do not normally contribute to syntenity blocks but would instead increase noises), 606 gene pairs remain. The human X chromosome has a total of 1360 genes and mouse 1267 genes. We further constructed 5-way syntenity blocks for human, mouse, rat, dog and chicken genomes using all available pairwise orthologous (one-to-one) genes from Ensembl 44 (supplementary Table S1).

We concatenated 1360 genes from the human X chromosome and 1267 genes from the mouse X chromosome, forming a genome of 2627 genes. During concatenation, a number of elements (larger than the gap threshold) were inserted between two chromosomes to prevent syntenity blocks from forming across boundaries of chromosomes or genomes. An A-Bruijn graph was constructed on the concatenated genomes using the 606 gene pairs between human and mouse X chromosomes as gluing instructions. The A-Bruijn graph has 906 vertices and 1636 edges (not shown). The graph was further simplified with the parameters $(g, B, L) = (10, 20, 4)$. The remaining graph has 46 vertices and 64 edges, from which syntenity blocks were extracted as shown in Fig. 1(e).⁵ Fig. 1(f) illustrates that the A-Bruijn graph is actually equivalent to the breakpoint graph for analyzing rearrangement scenarios [34]. After joining the neighboring syntenies of the same orientation, a total of 8 strips of syntenies emerged (Fig. 1(c)), covering 85.64% of human and 89.72% of mouse X chromosomes. The syntenies are similar to the published results [7] with small differences mainly caused by correcting fragment assembly errors in the latest versions of the human and mouse genomic sequences. The GRIMM-Syntenity results on this dataset are shown in 1(b). The blocks from the two algorithms largely coincide.

For human-mouse-rat-dog-chicken, we did two sets of syntenity block generations. In the first set, we concatenated all genes (31101 in human, 28157 in

⁵ The numbers of vertices and edges reflect the final A-Bruijn graph including both the forward and inverted sequences. The final A-Bruijn graph shown however only includes forward sequence for illustration purpose.

mouse, 27264 in rat, 22602 in dog, and 15936 in chicken) into a single genome, and applied AB-Synteny ($G, C, g, B, L = 30, 3, 10, 20, 4$) to the resulting genome using total of 125347 available gene pairs (1-to-1 orthologs) between any two genomes as gluing instructions. In the second set, we removed all genes that do not belong to any gene pair, and concatenated the remaining genes (16196 in human, 17196 in mouse, 16464 in rat, 15792 in dog, and 10908 in chicken) into a single genome, and applied AB-Synteny with the same parameters. The results are very similar and we report the results from the second set. After the vertices with 1-in and 1-out edges are merged, the A-Brujn graph has 23228 vertices and 41833 edges. After simplification, 3564 vertices and 6814 edges remain, resulting in 666 5-way synteny blocks. We also extracted 8735 5-way orthologous genes from the gene pairs and applied GRIMM-Synteny ($G, C = 100, 4$), where G is the total gap threshold. The results from the two algorithms are compared in Table 1. The shared coverage refers to regions of a genome that belong to synteny blocks reconstructed by both algorithms. Fig. 1(d) compares the human-mouse X chromosome portion of the 5-way synteny blocks to those synteny blocks derived from human-mouse data alone as shown in Fig. 1(c). As expected, the blocks from 5-way syntenies are shorter and more fragmented.

GRIMM-Synteny requires k -way anchors when there are k species. Some of the synteny blocks recovered by AB-Synteny but missed by GRIMM-Synteny

Table 1. 5-way synteny blocks constructed by AB-Synteny and GRIMM-Synteny

	Genome	AB-Synteny (666)		GRIMM-Synteny (466)		Shared coverage	
	length (Mb)	length (Mb)	%	length (Mb)	%	length (Mb)	%
human	3080	2017	65.47	2091	67.90	1837	59.63
mouse	2644	1792	67.79	1810	68.45	1604	60.66
rat	2719	1884	69.30	1915	70.43	1692	62.24
dog	2445	1669	68.26	1776	72.62	1527	62.44
chicken	1032	790	76.52	790	76.57	702	68.03

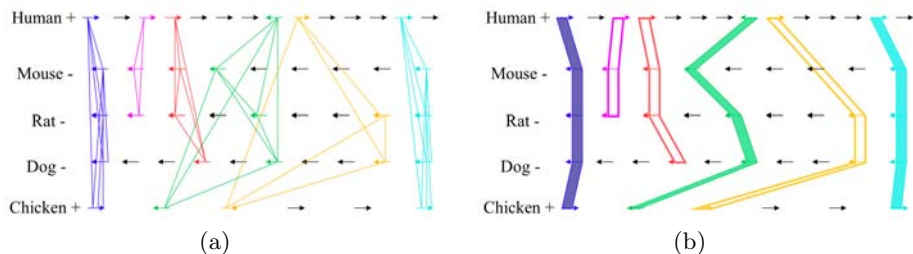


Fig. 3. A region of chromosome 1 of five species that is recovered by AB-Synteny but missed by GRIMM-Synteny due to the small number of 5-way anchors. (a) one-to-one orthologous gene pairs from Ensembl database. (b) 5-way anchors input to GRIMM-Synteny (3 filled): block missed; and all available anchors used by AB-Synteny (+3 unfilled): block recovered.

are due to a reduced number of k -way (5-way) anchors as the number of species increases. Fig. 3 illustrates such an example. The region is on chromosome 1 of the five species and consists of 13 genes from human, 9 from mouse, 9 from rat, 10 from dog and 5 from chicken. There are three 5-way anchors, two 4-way anchors and one 3-way anchor. Only the 5-way anchors can be used as inputs to GRIMM-Synteny or any other algorithms that require k -way anchors for k genomes. Since the number of such anchors is below the block threshold, the syntenic region is missed by GRIMM-Synteny. On the other hand, AB-Synteny requires only pairwise anchors between any two genomes. All six anchors therefore can be used as inputs. With equivalent parameter settings, AB-Synteny is able to recover the block as a result of more supporting anchors. This feature of AB-Synteny allows the algorithm to scale more easily to a large number of genomes.

4 Duplication Analysis of Plant Genome

We analyzed 5700 paralogous gene pairs in *A. thaliana* from [20], selected from about 30503 *A. thaliana* genes, and compared the results to published duplicated blocks generally accepted by the plant research community.

We applied AB-Synteny ($G, C, g, B, L = 20, 6, 10, 100, 4$) to the *A. thaliana* genome with 30503 genes and 5700 anchors (gene pairs). It generated 223 non-overlapping segments in 1-D, making up 103 syntenic blocks. Tables 2 and 3 compare AB-Synteny results with the syntenic blocks from [20] and [4]. Almost all our syntenic blocks are inside blocks of Bowers *et al.*, and about 2.66% of our blocks (covering 812 genes) are outside blocks of Blanc *et al.*

Fig. 4 shows several syntenic blocks generated by AB-Synteny for the *A. thaliana* genome. Notice that the blocks appearing in more than two copies (blue colored blocks) are delineated from the 2-copy blocks (magenta blocks). The single red block is one of the syntenic blocks (referred to as *chromosomal segment pairs*) reported in [20]. Careful inspection of the genomic segment shown in Fig. 4 reveals a large gap of 499 genes in chromosome 3 (x axis) and a corresponding gap of 12 genes in chromosome 4 (y axis). We argue that the AB-Synteny representation provides a more accurate view of the *A. thaliana* genomic architecture.

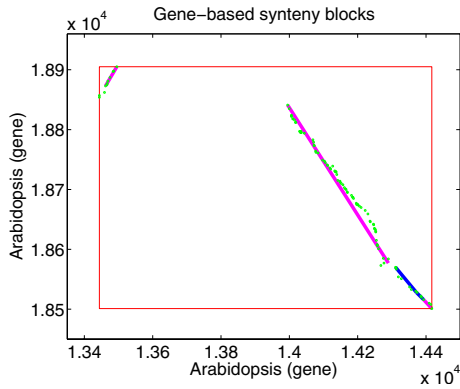
The last step in the syntenic block generation algorithm in [20] combines adjacent syntenic regions with opposite orientation and order that may be explained by local inversions, although it is not clear which inversions are considered local. The separation of segments in such cases can partially explain the comparatively low coverage of AB-Synteny as shown in Table 2.

Table 2. Comparison of AB-Synteny results to published *A. thaliana* syntenic blocks

Methods	# of Syntenic blocks	Coverage		Overlap in 1-D	
		# genes	%	# genes	%
1. [20]	34	26034	85.35	5069	16.62
2. [4]	91	24370	79.89	7118	23.34
3. AB-Synteny	103	21862	71.67	0	0

Table 3. Synteny block coverage shared between methods in Table 2

Methods	# genes	%
1 & 2	24089	78.97
1 & 3	21847	71.62
2 & 3	21050	69.01

**Fig. 4.** A local view of synteny blocks of *A. thaliana* generated by AB-Synteny. the synteny block from [20] is a red box; and other colored diagonals are synteny blocks generated by AB-Synteny: magenta: 2 copies, blue: 3 copies (extra copy not shown).

There is partial agreement of our synteny blocks with those generated by LineUp [21] (data not shown). While the syntenic regions reported by LineUp do in general overlap with the regions generated by AB-Synteny, LineUp reports all statistically significant syntenic regions without trying to define the boundaries of the regions. As these regions overlap significantly, they cannot be used for reconstruction of rearrangement and duplication scenarios.

5 Discussion

The uniqueness of our new synteny block generation algorithm AB-Synteny stems from the fact that it produces synteny blocks that do not overlap in 1-D representations, an essential property for further analysis of the synteny blocks to study rearrangement and duplication history. AB-Synteny can also be used for generation of synteny blocks across multiple genomes. Given k genomes, one simply concatenates them into a single genome. If there are no duplications in the original genomes, then the edges with multiplicity k in the A-Bruijn graph correspond to synteny blocks shared by all k genomes.

Our AB-Synteny algorithm constructs A-Bruijn graphs using the the RepeatGluer code initially developed for repeat classification and DNA fragment assembly. Extending RepeatGluer to new research domains typically requires

new application-specific algorithmic developments (e.g., constructing A-Bruijn graphs in mass spectrometry applications [29]). Similarly, the synteny block reconstruction may benefit from the modifications of the A-Bruijn graph approach that take into account the specific challenges of analyzing large highly duplicated genomes. We found that while most RepeatGluer steps (e.g., bulge removal) work well for synteny block generation, some steps need to be further optimized for the new application domain. In particular, we found that the *threading* heuristic from [26] (which worked well for fragment assembly) may lead to suboptimal results in synteny block reconstruction. Optimizing the A-Bruijn graph approach for synteny block generation represents the next challenge in analyzing the genomic architecture of the quickly increasing set of mammalian and plant genomes that are being sequenced using next generation sequencing technologies.

Acknowledgment

We thank H.X. Tang for providing the RepeatGluer code for A-Bruijn graph construction. We thank J.R. Ecker for insightful comments. QP was supported by NSF Plant System Biology IGERT Training Grant DGE0504645. GT was supported by NSF Grant DMS-0718810.

References

1. Vision, T.J., Brown, D.G., Tanksley, S.D.: The Origins of Genomic Duplications in Arabidopsis. *Science* 290(5499), 2114–2117 (2000)
2. Lander, E., Linton, L., Birren, B., Nusbaum, C., et al.: Initial sequencing and analysis of the human genome. *Nature* 409, 860–921 (2001)
3. Bailey, J., Baertsch, R., Kent, W., Haussler, D., Eichler, E.: Hotspots of mammalian chromosomal evolution. *Genome Biol.* 5(4), R23 (2004)
4. Blanc, G., Hokamp, K., Wolfe, K.H.: A recent polyploidy superimposed on older large-scale duplications in the arabidopsis genome. *Genome Res.* 13(2), 137–144 (2003)
5. Bourque, G., Pevzner, P.A., Tesler, G.: Reconstructing the Genomic Architecture of Ancestral Mammals: Lessons From Human, Mouse, and Rat Genomes. *Genome Res.* 14(4), 507–516 (2004)
6. Pevzner, P., Tesler, G.: Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution. *PNAS* 100(13), 7672–7677 (2003)
7. Pevzner, P., Tesler, G.: Genome rearrangements in mammalian evolution: Lessons from human and mouse genomes. *Genome Res.* 13, 37–45 (2002)
8. Peng, Q., Pevzner, P., Tesler, G.: The fragile breakage versus random breakage models of chromosome evolution. *PLoS Comput. Biol.* 2(2), e14 (2006)
9. Tesler, G.: Grimm: genome rearrangements web server. *Bioinf.* 18(3), 492–493 (2002)
10. Nadeau, J., Taylor, B.: Lengths of chromosomal segments conserved since divergence of man and mouse. *PNAS* 81, 814–818 (1984)
11. Waterston, R., Lindblad-Toh, K., Birney, E., Rogers, J., Abril, J., Agarwal, P., Agarwala, R., Ainscough, R., Alexanderson, M., An, P., et al.: Initial sequencing and comparative analysis of the mouse genome. *Nature* 420, 520–562 (2002)

12. Kent, W.J., Baertsch, R., Hinrichs, A., Miller, W., Haussler, D.: Evolution's cauldron: Duplication, deletion, and rearrangement in the mouse and human genomes. *PNAS* 100(20), 11484–11489 (2003)
13. Brudno, M., Malde, S., Poliakov, A., Do, C., Couronne, O., et al.: Glocal alignment: Finding rearrangements during alignment. *Bioinf.* 19, i54–i62 (2003)
14. Darling, A., Mau, B., Blattner, F., Perna, N.T.: Mauve: Multiple alignment of conserved genomic sequence with rearrangements. *Genome Res.* 14, 1394–1403 (2004)
15. Bourque, G., Yacef, Y., El-Mabrouk, N.: Maximizing synteny blocks to identify ancestral homologs. In: McLysaght, A., Huson, D.H. (eds.) RECOMB 2005. LNCS (LNBI), vol. 3678, pp. 21–34. Springer, Heidelberg (2005)
16. Ma, J., Zhang, L., Suh, B.B., Raney, B.J., Burhans, R.C., Kent, W.J., Blanchette, M.: Reconstructing contiguous regions of an ancestral genome. *Genome Res.* 16, 1557–1565 (2006)
17. Sinha, A., Meller, J.: Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. *BMC Bioinf.* 8(1), 82 (2007)
18. Hachiya, T., Osana, Y., Pependorf, K., Sakakibara, Y.: Accurate identification of orthologous segments among multiple genomes. *Bioinf.* 25(7), 853–860 (2009)
19. Kellis, M., Birren, B.W., Lander, E.S.: Proof and evolutionary analysis of ancient genome duplication in the yeast *saccharomyces cerevisiae*. *Nature* 428(6983), 617–624 (2004)
20. Bowers, J.E., Chapman, B.A., Rong, J., Paterson, A.H.: Unravelling angiosperm genome evolution by phylogenetic analysis of chromosomal duplication events. *Nature* 422, 433–438 (2003)
21. Hampson, S., McLysaght, A., Gaut, B., Baldi, P.: LineUp: Statistical Detection of Chromosomal Homology With Application to Plant Comparative Genomics. *Genome Res.* 13(5), 999–1010 (2003)
22. Haas, B.J., Delcher, A.L., Wortman, J.R., Salzberg, S.L.: DAGchainer: a tool for mining segmental genome duplications and synteny. *Bioinf.* 20(18), 3643–3646 (2004)
23. Vandepoele, K., Saeys, Y., Simillion, C., Raes, J., Van de Peer, Y.: The Automatic Detection of Homologous Regions (ADHoRe) and Its Application to Microcolinearity between *Arabidopsis* and Rice. *Genome Res.* 12(11), 1792–1801 (2002)
24. Simillion, C., Janssens, K., Sterck, L., Van de Peer, Y.: i-ADHoRe 2.0: an improved tool to detect degenerated genomic homology using genomic profiles. *Bioinf.* 24(1), 127–138 (2008)
25. Soderlund, C., Nelson, W., Shoemaker, A., Paterson, A.: SyMAP: A system for discovering and viewing syntenic regions of FPC maps. *Genome Res.* 16(9), 1159–1168 (2006)
26. Pevzner, P.A., Tang, H., Tesler, G.: De Novo Repeat Classification and Fragment Assembly. *Genome Res.* 14(9), 1786–1796 (2004)
27. Raphael, B., Zhi, D., Tang, H., Pevzner, P.: A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Res.* 14(11), 2336–2346 (2004)
28. Zhi, D., Raphael, B., Price, A., Tang, H., Pevzner, P.: Identifying repeat domains in large genomes. *Genome Biol.* 7(1), R7 (2006)
29. Bandeira, N., Clauser, K.R., Pevzner, P.A.: Shotgun Protein Sequencing: Assembly of Peptide Tandem Mass Spectra from Mixtures of Modified Proteins. *Mol. Cell Proteomics* 6(7), 1123–1134 (2007)
30. Bourque, G., Zdobnov, E.M., Bork, P., Pevzner, P.A., Tesler, G.: Comparative architectures of mammalian and chicken genomes reveal highly variable rates of genomic rearrangements across different lineages. *Genome Res.* 15(1), 98–110 (2005)

31. Dewey, C.N., Pachter, L.: Mercator: Multiple whole-genome-orthology map construction (2006), <http://bio.math.berkeley.edu/mercator>
32. Bao, Z., Eddy, S.R.: Automated De Novo Identification of Repeat Sequence Families in Sequenced Genomes. *Genome Res.* 12(8), 1269–1276 (2002)
33. Jiang, Z., Tang, H., Ventura, M., Cardone, M.F., Marques-Bonet, T., She, X., Pevzner, P.A., Eichler, E.E.: Ancestral reconstruction of segmental duplications reveals punctuated cores of human genome evolution. *Nat. Genet.* 11, 1361–1368 (2007)
34. Hannenhalli, S., Pevzner, P.: Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *J ACM* 46, 1–27 (1999)