Pacific Symposium on Biocomputing 6:287-298 (2001)

Gene Verification and Discovery by Walking Tree Method

Tai Hsu, Paul Cull
Computer Science Department, Oregon State University
Corvallis, OR 97331
(hsuta@cs.orst.edu, pc@cs.orst.edu)

The Walking Tree Method [3, 4, 5, 18] is an approximate string alignment method that can handle insertions, deletions, substitutions, translocations, and more than one level of inversions all together. Moreover, it tends to highlight gene locations, and helps discover unknown genes. Its recent improvements in runtime and space use extends its capability in exploring large strings. We will briefly describe the Walking Tree Method with its recent improvements [18], and demonstrate its speed and ability to align real complete genomes such as Borrelia burgdorferi (910724 base pairs of its single chromosome) and Chlamydia trachomatis (1042519 base pairs) in reasonable time, and to locate and verify genes.

1. Introduction

Most biological string matching methods are based on the edit-distance model [15]. These methods assume that changes between strings occur locally. But, evidence shows that large scale changes are possible [7]. For example, large pieces of DNA can be moved from one location to another (translocations), or replaced by their reversed complements (inversions). Schöniger and Waterman [14] extended the edit-distance model to handle inversions, but their method handled only one level of inversion. Hannenhalli's algorithm [10] for the "translocation" problem runs in polynomial time, but it requires gene locations to be known. Furthermore, it seems unlikely that any simple model will be able to capture the minimum biologically correct distance between two strings. In all likelihood finding the fewest operations that have to be applied to one string to obtain another string will probably require trying all possible sequences of operations. Trying all possible sequences is computationally intractable. This intractability has been confirmed by a recent proof by Caprara [2] that determining the minimum number of flips needed to sort a sequence is an NP-complete problem. Although signed flips can be sorted in polynomial time [11], apparently, we need a method that can handle insertions, deletions, substitutions, translocations, and inversions together. The Walking Tree heuristic handles translocations and multi-level inversions well, and also tends to highlight genes [3, 4, 5, 18].

2. Walking Tree Method

2.1 The Method

The problem is to find an approximate biologically reasonable alignment between two strings, one called pattern P, and the other called text T. Our metaphor is to consider the data structure as a walking tree with |P| leaves, one for each characters in the pattern. When the walking tree is considering position $l + 1$, the internal nodes remember some of the information for the best alignment within the first l characters of the text (Figure 1). On the basis of this remembered information and the comparisons of the leaves with the text characters under them, the leaves update their information and pass this information to their parents. The data will percolate up to the root where a new best score is calculated. The tree can then walk to the next position by moving each of its leaves one character to the right. The whole text has been processed when the leftmost leaf of the walking tree has processed the rightmost character of the text.

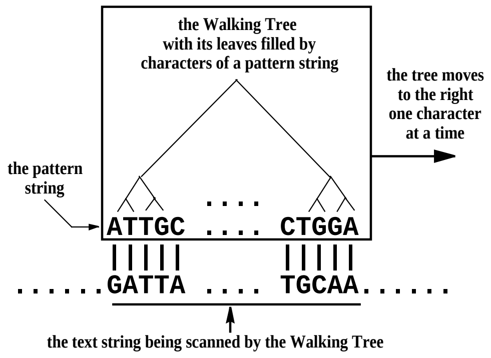

the Walking Tree with its leaves filled by characters of a pattern string
the tree moves to the right one character at a time
the pattern string
ATTGC .... CTGGA
......GATTA .... TGCAA......
the text string being scanned by the Walking Tree

Figure 1: This picture shows the walking tree's structure, a binary tree. Leaves of the tree contain the characters of the pattern string P. After comparing each leaf with a corresponding character of the text string, the walking tree updates its nodes with new scores, then moves to the next position by moving each of its leaves one character to the right. Then it repeats the leaf comparison, and updates its node scores until it reaches the end of the text string.

To define a scoring system that captures some biological intuitions, we use a function that gives a positive contribution based on the similarity between aligned characters, and a negative contribution that is related to the number and length of gaps, translocations, and inversions. A gap in an alignment occurs when adjacent characters in the pattern are aligned with non-adjacent characters in the text. The length of the gap is the number of characters between the non-adjacent characters in the text. The detailed description of the resource usage of the method can be found in Cull, Holloway and Hsu's papers [3, 4, 5, 18]

2. 2 Improvements in Speed and Space

The binary tree structure of the Walking Tree makes it extremely easy to implement a parallel version (Figure 2). Furthermore, inexpensive vector processors can be used because each node of the tree does the same operations at each scanning position. Each parent node of the walking tree simultaneously updates its score and position whenever it observes a better score.

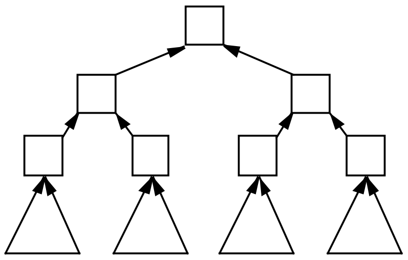


Figure 2: This picture shows the parallelization of the walking tree method. Given one processor per node of the tree, each child sends its current information to its parent; so a parent can update its best score and position by the information. Since the tree is $\log_2|P|$ high, $\Theta(\log_2|P|)$ startup time is needed for the root to receive its first information from leaves. After the startup time, all nodes work simultaneously; so, each text scan step takes $\Theta(1)$ time. The parallel runtime is $\Theta(\log_2|P| + |T|)$, i.e., $\Theta(|T|)$ because $|T| \geq |P|$.

We recognized that the alignment copying in the original design [3, 4, 5] was passively activated whenever a better score occurred. It's better to postpone the copying to allow faster scoring at the tree nodes. Based on this idea, we discovered improvements [18] for both the sequential and the parallel versions of the Walking Tree Method by using a state-caching technique similar to that used in recovering from program crashes (Figure 5.)

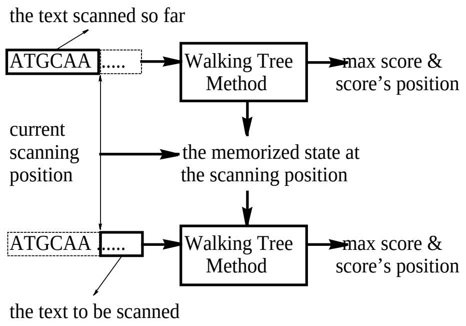
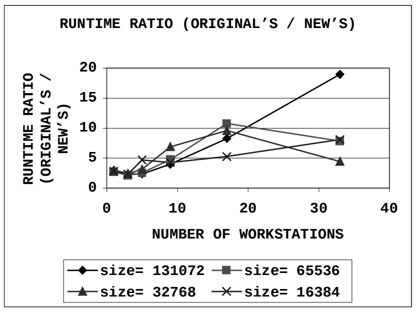

the text scanned so far
ATGCAA .....
Walking Tree Method
max score & score's position
current scanning position
the memorized state at the scanning position
ATGCAA .....
Walking Tree Method
max score & score's position
the text to be scanned

Figure 5: We use a technique similar to recovering a crashed program by saving its state before crashes. The memorized states record the states of the walking tree and the corresponding scanning positions of the text string. Once we have the recorded information, we can scan the text from the position we have memorized to avoid scanning from the first position of the text.

The improved sequential version [18] of the Walking Tree Method guarantees $\Theta(|P|*|T|*k)$ runtime using $\Theta(|P|*(\log_2|P|)^{1/k})$ space. With $\Theta(|P|)$ CPUs, the improved parallel version [18] guarantees $\Theta(|T|)$ runtime using $\Theta(|P|*\log_2|P|)$ space by reducing inter-processor communication to make CPUs spend more time on working rather than talking to each other. The improvements [18] also allows us to use a simpler implementation to overlap communication and computation in a shared memory model, e.g., a cluster of network computers. Exploring large strings becomes feasible. Fig. 14 shows the result of the new improvement versus the original method (the parallelization uses MPICH (version 1.1.0) [9, 12] and a cluster of Intel Pentium II 300 MHz machines (running Red Hat Linux 5.2) connected by a 100 Mbps switch). Our model doesn't consider the PRAM model [8] because the PRAM model [8] is considered unrealistic [1, 6], in that it assumes unlimited bandwidth and free interprocessor communication.



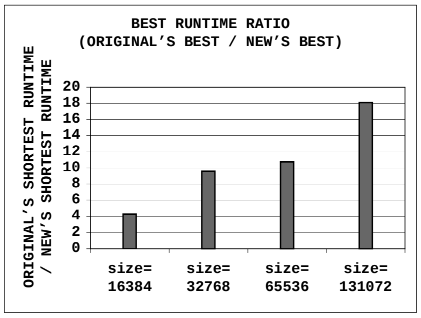


Figure 14: As the first picture shows, when only 1 workstation is used, the new method is about 2.7 times as fast as the original one; therefore, we should see the same speed ratio when using more workstations, if both methods are equally good in network parallelization. As the first picture shows, both are equally good when 5 workstations or less are used, but the new method prevails when 9 workstations or more are used. In addition, if both are equally good in network parallelization, the ratios of their best (i.e., the shortest) runtimes should be around 2.7 as well. That is, each method can use any number of the 33 workstations to get the best result for a particular input size. However, as the second picture shows, the new method prevails constantly with speed ratios better than 2.7, especially when the input size = 131072.

3. Previous Result

Our previous result showed that the Walking Tree can detect unknown genes, and align translocations and inversions. In Figure 15 and Figure 16, we show two alignments of two pairs of real DNA sequences. They are identical to the alignments found in Cull *et al*'s paper [5, 18].

4. New Result

With our recent improved Walking Tree Method, we are now able to align two real complete genomes (Fig 17, Fig 18, Fig 19), Borrelia burgdorferi [16] (910724 base pairs of its single chromosome and Chlamydia trachomatis [17] (1042519 base pairs) in 24 hours using 65 Pentium II 300MHz PC's. We separate the new result into 3 categories:

1. Matched regions that have annotations on both DNAs (TABLE A, TABLE B, Fig 17)
2. Matched regions that have annotations on only one DNA (Fig 18)
3. Matched regions that have no annotations on either DNA (Fig 19)

There are 103 matches in category 1, i.e., 40 translocations and 63 inversions. There are 148 matches in category 2, i.e., 86 translocations and 62 inversions. There are 1367 matches in category 3, i.e., 700 translocations and 667 inversions.

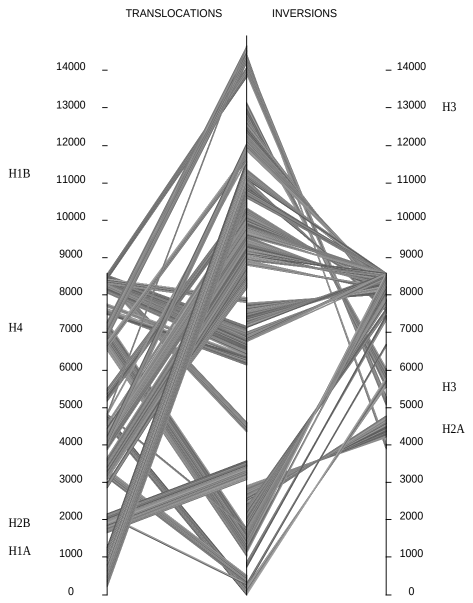
TABLE A: TRANSLOCATIONS

Borrelia burgdorferi		Chlamydia trachomatis	
Aligned positions by Walking Tree Method	Gene annotations & locations from Genbank	Aligned positions by Walking Tree Method	Gene annotations & locations from Genbank
88704 92799	89200 89814 BB0092	342373 346480	342872 343483 atpD
	89811 91115 BB0093		343468 344784 atpB
	91137 92792 BB0094		344787 346562 atpA
549888 551935	549642 551723 BB0540	505723 507776	505508 507592 fusA
454688 456702	454484 456403 BB0436	213601 215655	212937 215351 gyrB_1
			215354 215704 CT191
588032 589823	588066 589667 BB0575	204388 206112	204429 206048 pyrG
200960 202494	201052 202578 BB0201	299917 301427	300027 301478 murE
84672 86015	84041 85720 BB0088	75274 76600	74661 76469 lepA
456960 458239	456576 458036 BB0437	306732 307968	306433 307800 dnaA_2
114944 116223	114807 115508 BB0117	340937 342196	340917 342866 atpI
863488 864767	863636 865042 BB0817	897002 898255	897403 897822 CT763
686080 687103	685977 686507 BB0647	997281 998185	997122 997640 CT847
			997656 998162 CT848
326656 327678	326699 327757 BB0322	807860 808838	807691 808218 CT702
797696 798719	798057 799016 BB0755	82926 83736	82824 83780 ytgD
354816 355839	354648 355298 BB0346	716792 717677	717087 717770 cpxR
866304 867326	866494 866694 BB0820	898962 899997	898940 899272 rsbV_2
	866681 867601 BB0821		899276 900295 miaA
8704 9727	8412 9197 BB0008	514258 515301	514382 514606 CT444.1
673792 674815	673342 674778 BB0636	207123 208166	206802 208121 zwf
735744 736767	735343 736686 BB0694	30326 31339	29938 31284 ffh
526337 527359	526325 527305 BB0515	115999 117014	115919 116974 trxB
769537 770559	769547 771145 BB0730	430604 431626	430608 432185 pgi
345088 346111	345063 346364 BB0337	661878 662880	661850 663124 eno
752128 753151	752134 753219 BB0715	818326 819402	818358 819458 mreB
235520 236543	235595 237142 BB0230	566490 567573	566631 568025 rho
817153 818175	817393 817956 BB0776	997263 998285	997122 997640 CT847
			997656 998162 CT848
317440 318335	317247 318026 BB0309	306526 307394	306433 307800 dnaA_2
	318119 318256 BB0310		
586496 587390	586212 587024 BB0573	827709 828882	828429 828794 CT716
759296 760063	759586 760215 BB0721	987691 988495	987715 988779 CT839
528384 529151	528104 529198 BB0517	389889 390701	389567 390745 dnaJ
294912 295423	294785 295228 BB0284	690685 691202	690426 691121 CT610
60160 60670	60036 60554 BB0065	303646 304156	303731 304018 CT271
512513 513022	512393 513148 BB0507	828414 828982	828429 828794 CT716
459264 459775	459525 459824 BB0439	995630 996100	995570 996061 yfhC
823297 823807	823167 823811 BB0786	995596 996039	995570 996061 yfhC
738560 739006	738851 738964 BB0700	995494 996068	995570 996061 yfhC
531712 531967	531851 531967 BB0520	828452 828707	828429 828794 CT716

Table B: INVERSIONS

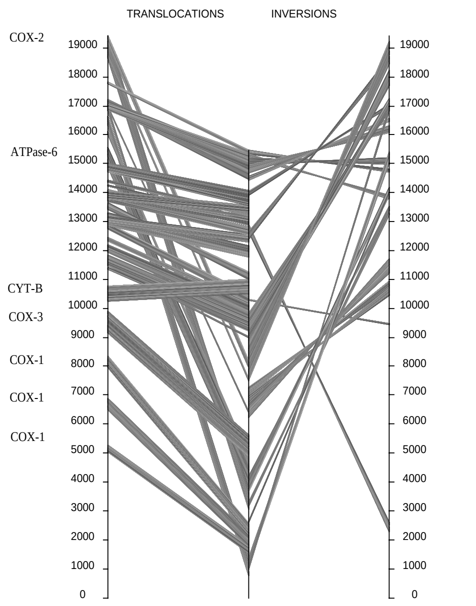
Borrelia burgdorferi		Chlamydia trachomatis	
Aligned positions by Walking Tree Method	Gene annotations & locations from Genbank	Aligned positions by Walking Tree Method	Gene annotations & locations from Genbank

500355 497028	497245 497874 BB0479	592489 595884	592462 593136 rs3
	497880 498191 BB0480		593146 593481 rl22
	498213 499046 BB0481		593500 593766 rs19
	499056 499334 BB0482		593772 594626 rl2
	499341 499703 BB0483		594650 594985 rl23
	499707 500588 BB0484		595001 595669 rl4
438403 435204	435201 435312 5S_rrlA	877808 881136	878039 880902 23SrRNA_2
	435334 438267 23S_rrlA		881027 881143 5SrRNA_2
441731 438660	438590 441508 23S_rrlB	877746 880813	878039 880902 23SrRNA_2
884099 882052	881085 884213 BB0833	21536 23573	21432 24542 ileS
258434 256452	256463 258985 BB0251	236302 238242	235766 238225 leuS
531331 529540	529198 531105 BB0518	451380 453186	451614 453596 dnaK
639875 638085	637963 638556 BB0611	811081 812898	811130 812389 clpX
	638580 639872 BB0612		812399 813010 clpP_2
446083 444548	444581 446118 16S	876201 877753	876174 877723 16SrRNA_2
467330 465796	465518 467038 BB0446	986947 988305	986612 987712 CT838
502659 501124	501215 501469 BB0487	590356 591861	590272 590814 rl5
	501491 501865 BB0488		590816 591151 rl24
	501880 502185 BB0489		591164 591532 rl14
	502191 502739 BB0490		591549 591800 rs17
690051 688516	688490 690127 BB0649	126399 127939	126336 127970 groEL_1
536963 535621	535704 537527 BB0526	340342 341750	340429 340875 atpK
496259 494980	495012 496217 BB0476	362055 363207	361980 363164 tufA
350850 349573	349600 351090 BB0342	2295 3565	2108 3583 gatA
180867 179588	179540 181423 BB0178	577486 578744	576941 578773 gidA
370051 369028	368885 370027 BB0361	828101 828995	828429 828794 CT716
803715 802692	802838 803212 BB0760	325245 326318	325478 325954 ptsN_2
			325956 326393 dut
311682 310660	310559 311653 BB0302	892795 893807	892826 893983 ftsW
52610 51588	51253 52434 BB0056	794746 795789	794941 796152 pgk
28035 27140	27434 27865 BB0029	997284 998154	997122 997640 CT847
			997656 998162 CT848
298371 297476	297466 298776 BB0288	765474 766365	765053 766381 yscN
297347 296580	296428 297051 BB0286	989003 989618	988877 989842 mesJ
	297038 297469 BB0287		
490626 489861	489733 490554 BB0471	540112 540904	540292 540933 CT465
504707 503941	503926 504285 BB0494	588442 589308	588369 588866 rs5
	504298 504795 BB0495		588881 589252 rl18
662403 661637	661606 662529 BB0630	690639 691280	690426 691121 CT610
6274 5508	5251 6312 BB0005	658647 659408	658617 659657 trpS
473539 472836	472566 473408 BB0453	541799 542519	541534 542592 atoS
505859 505220	505104 505541 BB0497	587606 588244	587942 588376 rl15
179587 178948	178917 179543 BB0177	995073 995719	995075 995413 rs1
695299 694660	694693 695523 BB0655	384856 385501	385149 385610 CT338
343043 342404	342335 343207 BB0334	791980 792623	791730 792695 dppD
238978 238468	238301 239128 BB0234	997811 998377	997656 998162 CT848
365443 364932	365115 365603 BB0355	830284 830887	830165 830689 CT718
347011 346500	346431 346841 BB0338	141786 142289	141972 142361 rs9
189826 189317	189299 189859 BB0190	982158 982667	982118 982699 infC
50050 49540	49341 50012 BB0053	686297 686809	686330 687019 ung
411491 411015	410787 411446 BB0399	903475 903960	903584 903943 ybeB
690563 690180	690151 690489 BB0650	385115 385495	385149 385610 CT338
500995 500613	500593 501009 BB0485	592026 592407	592013 592429 rl16
505027 504708	504799 505104 BB0496	589905 590193	589853 590254 rs8
438659 438404	438446 438557 5S_rrlB	858780 859118	858982 859098 5SrRNA_1
42883 42628	42480 42881 BB0044	898943 899200	898940 899272 rsbV_2
189059 188804	188708 189055 BB0188	982917 983169	982923 983294 rl20
482307 482180	482222 482308 tRNA-Ser-3	485243 485361	485247 485330 tRNASer_3



max_score = 11108; max_pos = 18914; |P| = 8592; |T| = 14942; c = 2; IZ = 9; updates = 2907
Pattern: (in the middle) X. laevis gene cluster X03017
Text : (at both sides) X. laevis gene cluster X03018

Figure 15: An alignment of two histone gene clusters from Xenopus laevis, GenBank accession number: X03017 (in the middle) and X03018 (at both sides). Note that genes H2A, H2B, H3, and H4 are marked on both sequences. The alignment shows that the orientation of H2A and H3 are reversed in the two sequences. This picture shows the Walking Tree Method is capable of finding inversions and translocations of genes.



max_score = 19871; max_pos = 31102; |P| = 15455; |T| = 19431; c = 2; IZ = 9; updates = 5847
Pattern: (in the middle) Anopheles Quadrimaculatus mitochrondrial genome MSQNCATR
Text : (at both sides) Schizosaccharomyces pombe mitochrondrial genome, MISPCG

Figure 16: An alignment of the mitochondrial genomes of Anopheles quadrimaculatus, GenBank locus MSQNCATR (in the middle), and Schizosaccharomyces pombe, GenBank locus MISPCG (at both sides). The previously unrecognized Cytochrome c oxidase 3 (COX-3) region in this map is identified by the Walking Tree Method.

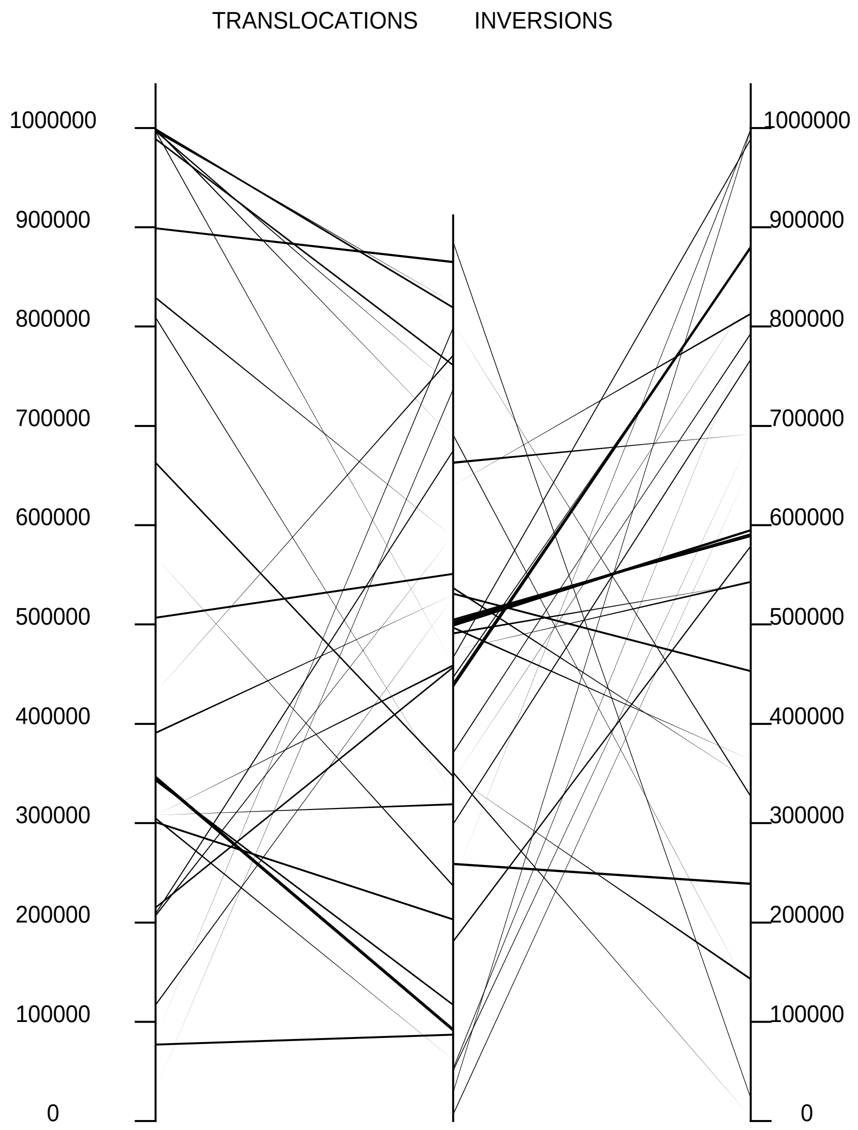


Fig 17. This picture shows that the Walking Tree Method reveals the matched genes that are labeled (annotated) on both DNAs (total DNA sequence of Borrelia burgdorferi aligned with the total DNA sequence of Chlamydia trachomatis). There are 40 translocations and 63 inversions in this picture. Again, this picture shows the Walking Tree Method is capable of finding inversions and translocations of genes.

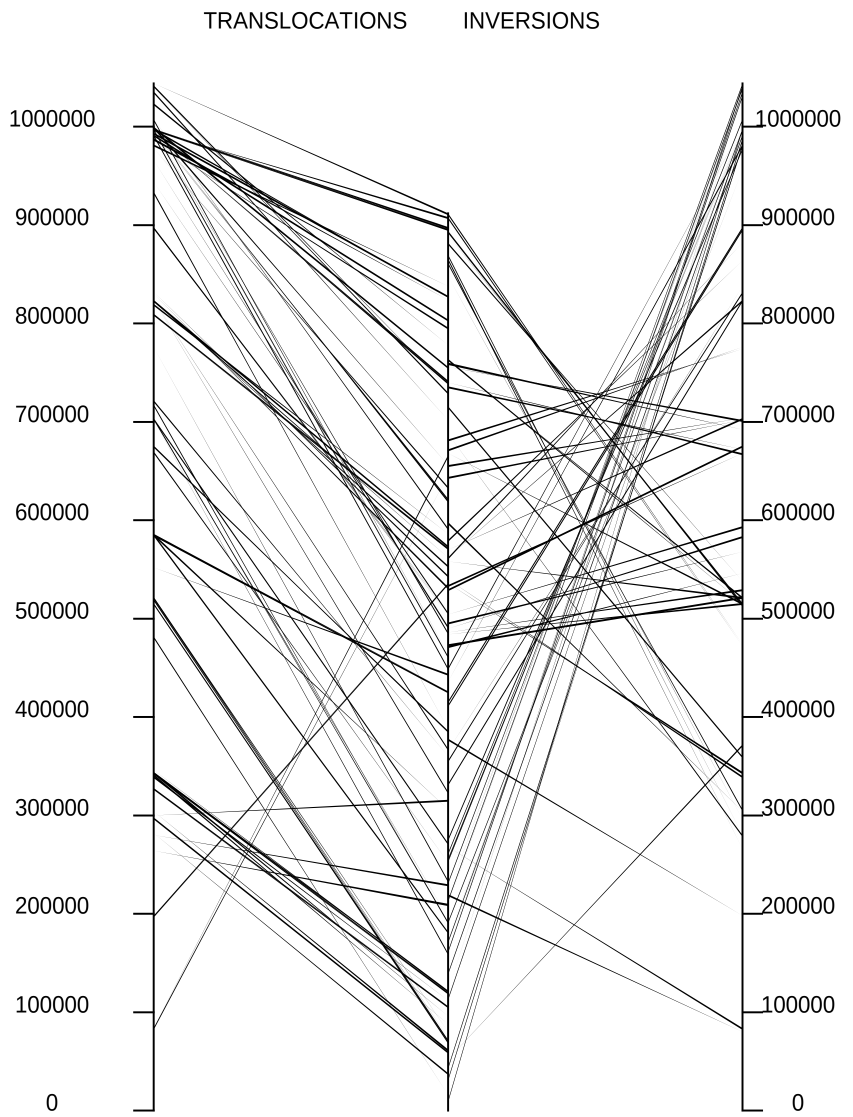


Fig 18. This picture shows that Walking Tree Method reveals the matched genes that are labeled on only one DNA, i.e., genes can be located in one sequence if the aligned portion of the other sequence is known to be a gene. There are 86 translocations and 62 inversions in this picture. This picture shows potential gene locations that are not annotated in one DNA, but annotated in another.

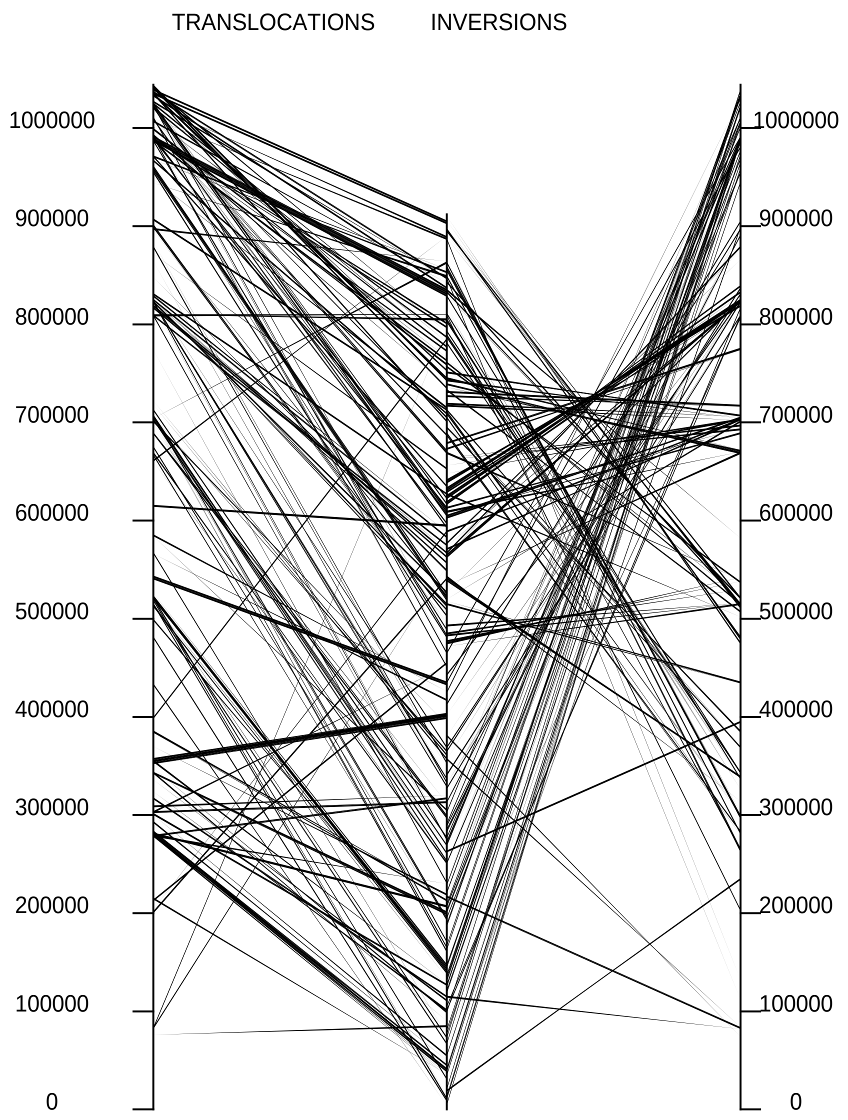


Fig 19. This picture shows that the Walking Tree Method reveals potential genes that are unlabeled on both DNAs. There are 700 translocations and 667 inversions in this picture. What interests us is the big match (Chlamidia: 352764 to 357294 and Borrelia: 399872 to 403967) which only covers 50% of the locus BORRPOB annotated in the GenBank database, but is found on both DNAs. This implies that Borrelia's BORRPOB annotation in Genbank may need to be reinvestigated.

5. Conclusion

The Walking Tree Method is a powerful tool for gene finding. The technique works by finding a "best" alignment between sequences. In common with other techniques, the Walking Tree can use a known gene in one genome to find a corresponding gene in another genome.

The real power of the technique is to find corresponding but unannotated regions in different genomes. Preservation of regions across separated species is strong evidence of biological function. We gave several examples of the locations of genes or interesting regions in a variety of organisms. Our improved parallelization technique makes alignment of million base sequences a one day operation.

6. Reference

1. A. Alexandrov, M. Ionescu, E. Schauser, C. Scheiman, "LogGP: Incorporating Long Messages into the LogP Model - One Step Closer Towards a Realistic Model for Parallel Computation," 7th Annual Symposium on Parallel Algorithms and Architectures, July, 1995.
2. A. Caprara, "Sorting by reversals is difficult," RECOMB 1997, pp75-83. ACM Press, New York.
3. P. Cull and J. L. Holloway, "Divide and Conquer Approximate String Matching: When Dynamic Programming is not Powerful Enough," Technical Report 92-20-06, Computer Science Department, Oregon State University, 1992.
4. P. Cull and J. L. Holloway, "Aligning genomes with inversion and swaps," Second International Conference on Intelligent Systems for Molecular Biology. Proceedings of ISBM '94, 195-202. AAAI Press, Menlo Park CA 1994.
5. P. Cull, J. L. Holloway, and J. D. Cavener, "Walking Tree Heuristics for Biological String Alignment, Gene Location, and Phylogenies," CASYS'98, Computing Anticipatory Systems (D. M. Dubois, editor), American Institute of Physics, Woodbury, New York, pp201-215, 1999.
6. D. E. Culler, R. M. Karp, D. A. Patterson, A. Sahay, K. E. Schauser, E. Sabtos, R. Subramonian, and T. von Eicken; "LogP: Towards a Realistic Model of Parallel Computation," Proceedings of the 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, San Diego, California. May 1993.
7. K. M. Devos, M. D. Atkinsoon, C. N. Chinoy, H. A. Francis, R. L. Harcourt, R. M. D. Koebner; C. J. Liu, P. Masojc, D. X. Xie, and M. D. Gale, "Chromosomal rearrangements in the rye genome relative to that of wheat," Theoretical and Applied Genetics 85:673-680, 1993.

8. S. Fortune and J. Wyllie, “Parallelism in Random Access Machines,” Proceedings of the 10th Annual Symposium on Theory of Computing, pp114-118, 1978.
9. W. Gropp, E. Lusk, N. Doss, and Skjellum A., “A high-performance, portable implementation of the MPI message passing interface standard,” Parallel Computing, vol. 22, no. 6, p.789-828, September 1996.
10. S. Hannenhalli, “Polynomial Algorithm for Computing Translocation Distance between Genomes,” Combinatorial Pattern Matching, pp162-176, 1995.
11. S. Hannenhalli and P. Pevzner, “Transforming Cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals,” Proceedings of the 27th ACM Symposium on the Theory of Computing, pp178-189, 1995.
12. Message Passing Interface Forum. MPI: A message-passing interface standard. International Journal of Supercomputer Applications, 8(3/4):165-414, 1994.
13. M. Python, “Just the Words.” Methuen, London, 1989.
14. M. Schöniger and M. S. Waterman, “A local algorithm for DNA sequence alignment with inversions,” Bulletin of Mathematical Biology, 54:521-536, 1992.
15. J. Setubal and J. Meidanis, “Introduction to Computational Molecular Biology,” PWS Publishing, Boston, MA, 1996.
16. Fraser, C.M., Casjens, S., Huang, W.M., et al. Genomic sequence of a Lyme disease spirochaete, Borrelia burgdorferi. Nature 1997 Dec; 390(6660):580-586.
17. R. S. Stephens, S. Kalman, C. J. Lammel and colleagues "Genome Sequence of an Obligate Intracellular Pathogen of Humans: Chlamydia Trachomatis", Science 282:754-759, 1998
18. Paul Cull and Tai Hsu. “Improved Parallel and Sequential Walking Tree Algorithms for Biological String Alignments”, Supercomputing Conference, 1999