genBlastA: Enabling BLAST to identify homologous gene sequences

Rong She,[1,3] Jeffrey S.-C. Chu,[2,3] Ke Wang,[1] Jian Pei,[1] and Nansheng Chen[2,4]

[1] *School of Computing Science, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada;* [2] *Department of Molecular Biology and Biochemistry, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada*

BLAST is an extensively used local similarity search tool for identifying homologous sequences. When a gene sequence (either protein sequence or nucleotide sequence) is used as a query to search for homologous sequences in a genome, the search results, represented as a list of high-scoring pairs (HSPs), are fragments of candidate genes rather than full-length candidate genes. Relevant HSPs ("signals"), which represent candidate genes in the target genome sequences, are buried within a report that contains also hundreds to thousands of random HSPs ("noises"). Consequently, BLAST results are often overwhelming and confusing even to experienced users. For effective use of BLAST, a program is needed for extracting relevant HSPs that represent candidate homologous genes from the entire HSP report. To achieve this goal, we have designed a graph-based algorithm, genBlastA, which automatically filters HSPs into well-defined groups, each representing a candidate gene in the target genome. The novelty of genBlastA is an edge length metric that reflects a set of biologically motivated requirements so that each shortest path corresponds to an HSP group representing a homologous gene. We have demonstrated that this novel algorithm is both efficient and accurate for identifying homologous sequences, and that it outperforms existing approaches with similar functionalities.

[Supplemental material is available online at www.genome.org.]

Genome sequencing projects, including the human genome projects (Lander et al. 2001; Venter et al. 2001), have produced enormous amounts of nucleotide sequences. With recent advances in sequencing technologies (Margulies et al. 2005; Bentley 2006), the volume of the nucleotide sequences is expanding at an accelerating pace, further enriching genomic sequence resources. To effectively exploit these resources for biological and medical research, many homology-based similarity search and alignment tools have been developed over the past 20 yr. Representative similarity search and alignment tools include BLAST (Altschul et al. 1990), FASTA (Pearson and Lipman 1988), sim4 (Florea et al. 1998), WU-BLAST (Lopez et al. 2003), and BLAT (Kent 2002). These tools have been extremely useful, especially for comparative genomics, in which genomes of both closely and distantly related species are compared in order that knowledge of the genome of one species can be used to understand the genome of other species (Hardison 2003).

In general, these search tools work by identifying a list of sequence segments in a target genome sequence database that show similarity to a query sequence. For example, BLAST detects regions of similarity between the query sequence and target sequences in a database. As illustrated in Figure 1, each match between the query sequence fragment and the target sequence fragment is reported as a high-scoring pair (HSP), which consists of a pair of sequences: [Q,T], where Q is a segment from the query sequence (i.e., query segment) and T is the matching segment from a target sequence in the target database (i.e., target segment). When a BLAST search returns numerous HSPs for a query gene (a protein sequence or a cDNA sequence) in the target genome, it suggests the existence of one or more homologous genes in the genome (or nucleotide database), with each HSP usually corresponding to an exon. BLAST assigns each HSP a bit score, an expectation value (*E*-value), as well as a percentage of identity (PID) and similarity values. For example, when the protein encoded by the *Caenorhabditis elegans* gene C11G6.3 is used as a TBLASTN query for the *C. elegans* genome, many HSPs are reported. Each HSP is unique, with a corresponding *E*-value and PID. Among these HSPs, some may represent candidate bona fide genes and can provide biologists with a meaningful starting point for further research, while others are random hits. Thus, although BLAST and other similarity searching tools produce lists of HSPs, they do not reveal which HSPs represent candidate genes, let alone reveal how many homologous genes exist in the target genome.

Over the past years, ad hoc solutions have been developed to filter and group HSPs, which are produced using BLAST and other similarity-based searching tools, into groups representing genes. The problem is that these ad hoc solutions can resolve some genes but fail in many cases. The best-known program that provides the functionality of grouping HSPs is WU-BLAST (Lopez et al. 2003), a BLAST program derivative. It can categorize HSPs into groups when users enable the "topcomboE" option. Within each group produced by WU-BLAST, HSPs are usually adjacent and collinear. Although WU-BLAST can successfully group some HSPs into gene-like structures, for HSPs representing candidate genes within tandem clusters in the target genome, WU-BLAST inevitably fails. For these cases, WU-BLAST tends to group HSPs corresponding to different genes into the same group, as discussed later. A program based on the longest increasing subsequence algorithm (LIS) was developed to filter and group BLAST HSPs (Zhang 2003). Similar to the WU-BLAST program, it does not reliably interpret HSPs representing multiple paralogous genes. Another program, BLAST2GENE, was developed to specifi-

[3] **These authors contributed equally to this work.**
[4] **Corresponding author.**
E-mail chenn@sfu.ca; fax (778) 782-5583.

Article published online before print. Article and publication date are at http://www.genome.org/cgi/doi/10.1101/gr.082081.108.

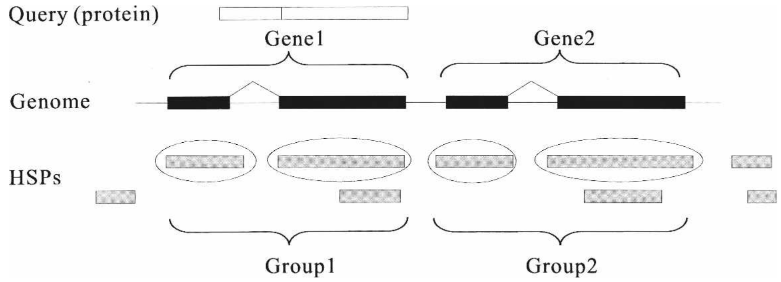

Figure 1. Grouping of HSPs into groups representing paralogs (Gene1 and Gene2) in tandem in the target genome. For simplicity, this figure shows only a small portion of the HSPs returned by BLAST. Each HSP may correspond to a coding segment (likely an exon) of a gene, thus a group of HSPs may collectively represent a full-length gene. Each shaded box at the *bottom* of the figures represents an HSP at its corresponding genomic position. Candidate genes are shown on the genome, with exons (black boxes) connected by introns (lines). The HSP groups that best represent the genes are shown under the corresponding genes, with relevant HSPs in the groups circled. Two paralogous genes in tandem (Gene1 and Gene2) are shown. The boundary of the two genes must be correctly resolved.

cally solve the multiple paralogous gene problem (Suyama et al. 2004); however, because it relies on many arbitrary thresholds and matrix usage, its application may be limited.

More recently, Cui et al. (2007) developed a new filtering and grouping algorithm that processes BLAST results, which were in turn used for identifying homologous genes. The investigators applied a three-step procedure to filter and group HSPs that represent candidate genes: (1) filter all HSPs by discarding HSPs with scores lower than a heuristic value; (2) group HSPs based on their physical distance along the chromosomes; and (3) further filter HSPs by estimating the genomic span of target regions. All HSPs that fall outside of the target regions are excluded from further analysis. Comparing to WU-BLAST, which fails in filtering and grouping HSPs representing all tandem homologous genes, this program correctly filters and groups HSPs representing some tandem homologous genes. However, this program has an important weakness, which is its dependence on the physical distances (step 2) between gene structures (groups of HSPs) to separate groups. It assumes that the distance between different genes are significantly larger than the distance between HSPs within a group, which is not true, especially for paralogous genes in tandem clusters. Due to the usage of ad hoc distance thresholds to separate adjacent genes, the program by Cui et al. (2007) fails to resolve individual paralogous genes within tandem clusters. On one hand, if the distance threshold value for separating genes is too large, HSPs corresponding to multiple genes will be lumped together into a large group. On the other hand, if the threshold value is too small, HSPs corresponding to a same gene could be divided into different HSP groups. In addition to this important weakness, the program by Cui et al. (2007) cannot be applied to filter HSPs that represent genes also because this program does not remove random HSPs that fall into the genomic region that contain the candidate gene.

The filtering and grouping task is particularly challenging when the query gene has a large number of paralogous genes in tandem in the target genome, as illustrated in Figure 1. Figure 1 shows that a query gene could have two (or more) homologous genes (Gene1 and Gene2) that are located in adjacent genomic regions. It is well known that a large number of genes in almost all sequenced genomes to date are parts of tandem homologous gene clusters. For example, in the nematode *C. elegans* genome, more than 1400 chemosensory genes form many tandem gene clusters, each of which contains two or more homologous genes (Robertson and Thomas 2006). Therefore, a program that is capable of filtering and assembling HSPs representing genes in tandem clusters is very important.

In this project, we developed a new graph-based algorithm, genBlastA, to directly address the above described challenge, among other issues, in filtering and assembling HSPs into genomic gene regions. A distinctive feature of genBlastA is that it does not rely on using ad hoc thresholds for filtering noise HSPs and on physical distance between target genes. Instead, genBlastA models the relationships and constraints among HSPs as a directed graph—designated the HSP graph—and models the HSP filtering and assembling problem as a search for the shortest paths in this graph. The novelty of this graph-based algorithm is an innovative edge length metric that reflects a set of biologically motivated requirements so that each shortest path corresponds to an HSP group representing a homologous gene. Unlike existing ad hoc grouping methods, this method filters and assembles HSPs on the basis of *optimizing* the path length to best capture the quality of a group of HSPs as a candidate gene. Consequently, our method is more robust, and it finds an optimal solution (with respect to a given length metric) without imposing a prior constraint (i.e., ad hoc thresholds) on gene structures.

We have tested the performance of genBlastA extensively in filtering and assembling HSPs found in the genomes of two closely related nematode species: *C. elegans* (Consortium 1998) and *Caenorhabditis briggsae* (Stein et al. 2003). These genomes were selected for testing because both have been extensively annotated. Our study shows that the performance of genBlastA is significantly better than that of WU-BLAST and the program by Cui et al. (2007).

Results

In this project, we developed the program genBlastA (described in Methods) that uses a novel graph-based algorithm that gives the program excellent capability for identifying HSP groups that represent orthologs (genes in different species but with same origin in evolution), paralogs (genes duplicated within a species), as well as novel genes (genes that have not yet been identified).

Test gene set preparation and test strategy

The data sets used for evaluation were obtained from WormBase (http://www.wormbase.org/), an integrated database for the biology and genomics of *C. elegans* and other nematode species including *C. briggsae* (Chen et al. 2005), release WS170. For testing the performance of genBlastA, we have selected a test gene set of 464 *C. elegans* genes that are representative of the *C. elegans* genome. To achieve this representation, the majority (300 genes) of these genes were taken from three representational contiguous regions of *C. elegans* chromosome I. These three regions are the left arm (containing 100 genes), the middle region (containing 100 genes), and the right arm (containing 100 genes) of chromosomal regions. To ensure that the test gene set contains representative genes of different complexities, we further included 164 additional genes, including genes with internal repetitive regions (Pfam domains) and genes that belong to large paralogous tandem clusters. The executable file and the test gene set can be downloaded from http://genome.sfu.ca/projects/genBlastA/.

To evaluate the capability of genBlastA to identify and group HSPs into gene-like structures and the capability of identifying novel genes, we used *C. elegans* genome as the target database for *C. elegans* query genes (called EvsE test). To evaluate the performance of genBlastA in identifying homologous sequences in genomes of different but related species, we used *C. briggsae* genome as the target database for the same set of *C. elegans* query genes (called EvsB test). These two species split ~80–120 million yr ago (Coghlan and Wolfe 2002; Stein et al. 2003), around the same time as the human/mouse split (Waterston et al. 2002).

The query for genBlastA can be either protein sequences or cDNA sequences. Details for using genBlastA are described in the README file included in the software package. In our experiments, genBlastA was able to process all 464 test genes (with over 43,000 HSPs reported by BLAST in EvsE test) within only 1 min on a medium-speed PC (with a Pentium-IV 2.6-GHz CPU). Since these 464 genes are representative of the entire *C. elegans* genome and comprise 2% of the genome, we calculate that it would take less than 1 h to process the entire genome (which contains ~20,000 genes).

We compared the performance of genBlastA with two existing programs with similar functionalities—WU-BLAST (Lopez et al. 2003) and the program by Cui et al. (2007). WU-BLAST is available by an academic license. Since the HSP grouping functionality of the program by Cui et al. (2007) is not readily available, we implemented this program, called ML in the following text, based on their publication (Cui et al. 2007). ML requires a distance threshold to resolve different HSP groups. This threshold is not described in detail in their publication; therefore, we derived an optimal distance value based on simulation results. In our experiments, we found that ML performs best when the distance threshold is set to 1000 bp for our test cases described below (Supplemental Figs. 1–3). Therefore this distance was used for ML throughout our analysis.

For each query gene in the test gene set, we first ran TBLASTN against the *C. elegans* genome (for EvsE test) and the *C. briggsae* genome (for EvsB test) with two different BLAST settings: "ungapped" and "gapped," while the gapped HSPs are generally longer with more gaps and mismatches and ungapped HSPs are generally shorter with much higher PIDs. We then carried out three sets of experiments, each with a different purpose.

1. Resolving paralogous genes in tandem clusters: This first experiment was designed to test the capability of these programs in addressing the major challenge that we have identified—resolving HSP groups that correspond to target gene families in the target genome. For this purpose, we selected 30 genes from the test gene set that belong to large gene families and these family members form tandem gene clusters.
2. Searching for orthologous groups: In this test, each gene in the test gene set was used as a query to identify the top-ranked HSP group, i.e., the candidate ortholog of the query gene. Since the top-ranked group is expected to be the most similar to the query gene, in the EvsE test, it is expected to map to the query gene itself; in the EvsB test, it should map to its *C. briggsae* ortholog.
3. Identifying novel genes: In the third experiment, we explored the utility of genBlastA for identifying novel (paralogous) genes, i.e., the genomic regions that show high similarity to known genes but have no gene annotations.

Resolving paralogous genes in tandem clusters

To test the three programs' abilities to resolve tandem duplicate genes, we examined the HSP groups produced for 30 query genes in the test gene set that are members of large gene families. For our comparison, after we identified HSP groups using genBlastA, WU-BLAST, and ML, we retained all candidate regions with query coverage ≥50%. The HSP groups were then examined and divided into two categories: "specific" and "nonspecific" groups. An HSP group is called specific if the corresponding genomic region contains only one annotated gene and is called nonspecific if the region has multiple annotated genes. HSP groups with high similarity to the query and containing only single genes are likely to be true paralogs. The programs' performance in resolving multiple paralogous genes is evaluated by comparing the ratio of specific groups over the total number of HSP groups examined. Figure 2 illustrates an example, in which there are five paralogous genes in a tandem gene cluster. As expected, WU-BLAST correctly identified only one target gene and failed to produce HSP groups corresponding to the rest of the four genes. ML produced three groups, two of which erroneously contain HSPs corresponding to other adjacent genes. ML missed groups for two target genes (T27B7.4 [*nhr-115*] and T27B7.6a [*nhr-228*]), and mistakenly grouped HSPs corresponding to T27B7.6a to the HSP group corresponding to T27B7.5 (*nhr-227*) (Fig. 2). In contrast, genBlastA successfully resolved all five genes, producing five groups of HSPs.

In summary, when BLAST was executed with the ungapped setting in the EvsE sets, the average ratio of specific HSP groups by genBlastA is ~80%, which is significantly higher than that produced by WU-BLAST (~20%) or ML (~40%) (Fig. 3). Similar results were observed when BLAST was performed with the gapped setting. Thus, in all cases, genBlastA was able to resolve more specific HSP groups in tandem duplicates compared to either WU-BLAST or ML. WU-BLAST usually generated numerous HSP groups, but they usually spanned regions with multiple genes (therefore nonspecific). Consequently, WU-BLAST groups together tandem paralogous genes, leading to poor performance in resolving tandem paralogous genes. ML had poor performance due to its use of a distance threshold. In particular, as the distance threshold increases, the ability of ML to resolve closely spaced paralogous groups decreases.

Searching for orthologous groups

In this test, the top-ranked HSP group corresponding to each query gene is evaluated by comparing to the expected gene as annotated in WormBase (WS170). First, we compared the accuracy rates of three programs when *C. elegans* genes were used as query genes to search for top-ranked genes in *C. elegans* genome. The accuracy rate is defined as the percentage of correctly assembled HSP groups. The accuracy rate for genBlastA is 97.2%, much higher than those of WU-BLAST and ML, which are 67.0% and 82.8%, respectively. For more accurate comparisons, the similarity or overlap between the HSP group and the expected gene were quantified. We used the following two criteria to evaluate the top-ranked HSP groups: (1) query coverage and (2) genomic span. Query coverage measures the similarity between the HSP group and the query gene. It is defined as the proportion of the query sequence covered by the HSPs in the HSP group identified by each of the three programs. A program should identity the HSP group that best covers the query gene. Genomic span measures the extent of overlap between the genomic region

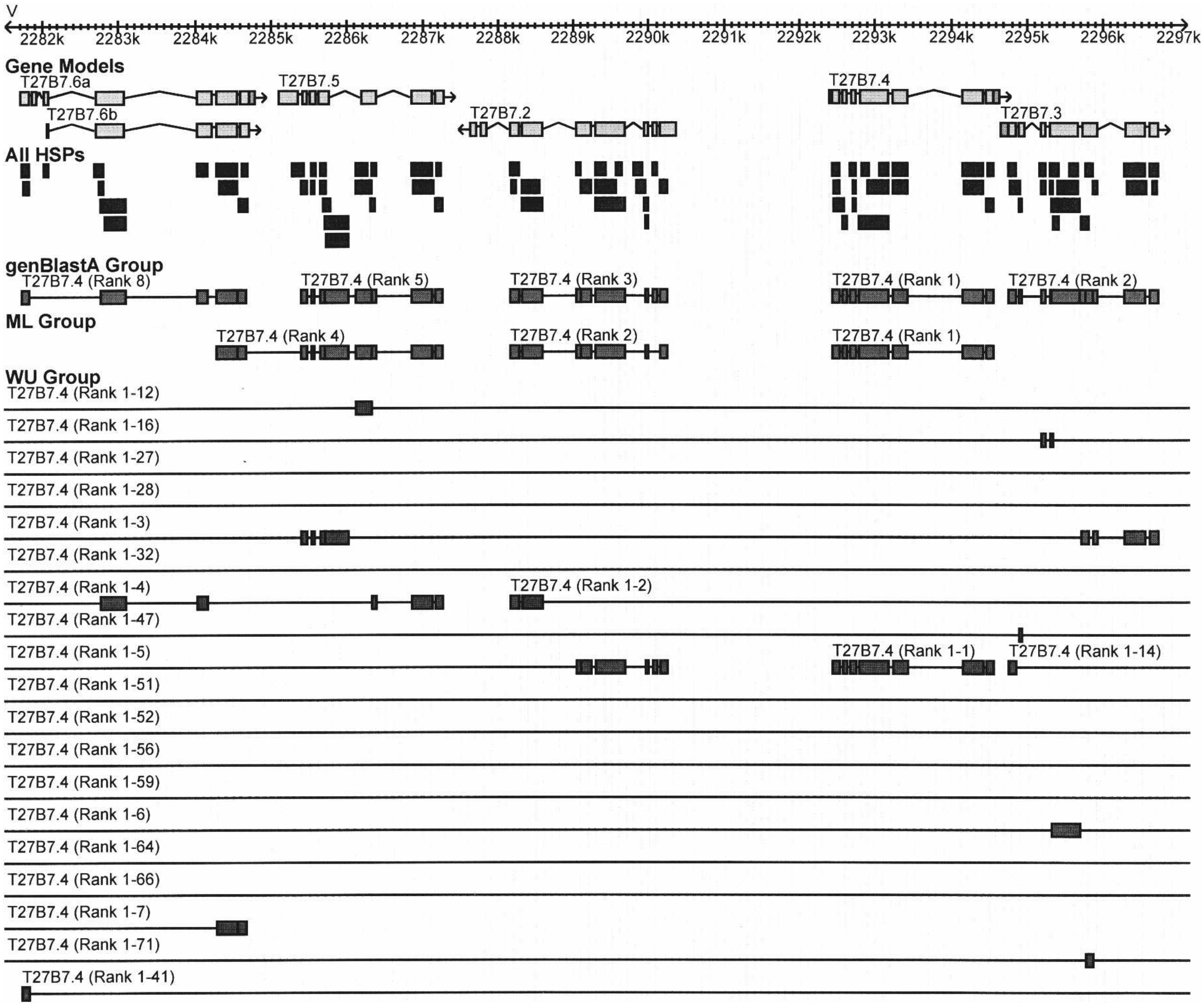

Figure 2. Grouping HSPs into groups representing individual genes. genBlastA was able to resolve all five members, while ML resolved only two and WU only one. Gene models are shown in the Gene Models track. HSPs are shown as blue boxes in the All HSPs track. The color indicates different PIDs for the HSPs. Darker color indicates higher PID. The genBlastA Group, ML Group, and WU Group tracks show HSPs groupings that are returned by genBlastA, ML, and WU-BLAST, respectively.

given by the HSP group and the expected gene region in the target genome. We evaluated this using the Jaccard similarity: For the annotated target gene region R_A and the reported gene region R_R, their similarity is $(|R_A \cap R_R|/|R_A \cup R_R|)$. This result is zero when two regions do not overlap.

Query coverage test

Figure 4, A and C, shows the average query coverage for 464 query genes in the test gene set. When BLAST was executed using the ungapped setting in the EvsE test (Fig. 4A) and the EvsB test (Fig. 4C), genBlastA identifies HSP groups with close to 100% query coverage and significantly outperformed both WU-BLAST and ML. Similarly, when BLAST was executed using the gapped setting, genBlastA significantly outperformed both WU-BLAST and ML in the EvsE test (Fig. 4A) and the EvsB test (Fig. 4C).

Genomic span test

As shown in Figure 4B, when BLAST was run using the ungapped setting, for both EvsE and EvsB tests, genBlastA significantly outperformed both WU-BLAST and ML by large margins, suggesting that genomic regions predicted by WU-BLAST and ML are dramatically different from the real genomic regions. Similarly, when BLAST was run using the gapped setting, for both EvsE and EvsB tests, genBlastA outperformed both WU-BLAST and ML significantly, while WU-BLAST outperformed ML.

Taken together, genBlastA outperformed both WU-BLAST and ML in identifying orthologous HSP groups.

Identifying novel genes

Since genBlastA can be applied to effectively identify homologous genomic regions in a target genome, we reasoned that it can be used for identifying novel paralogous genes that have been missed by other approaches. To demonstrate this, we examined whether genBlastA can be used to identify HSP groups in the *C. elegans* genome that are homologous to the test genes and that do not overlap with any existing gene annotation, therefore, identifying putative novel genes or novel pseudogenes.

We evaluated all candidate homologous gene regions for the 464 query genes for ones that show both significant query gene

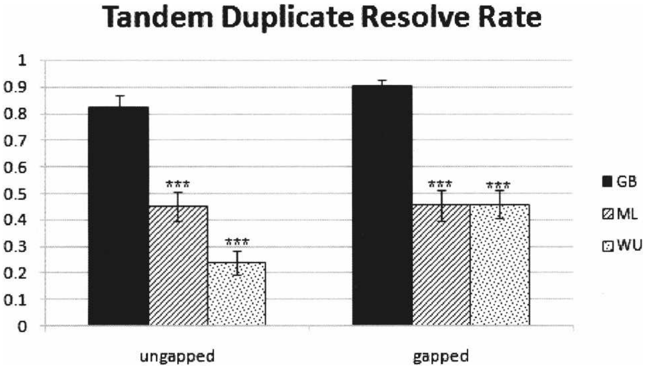

Figure 3. Grouping of HSPs to represent individual homologous genes in tandem clusters. This figure shows average resolve rate for a total of 30 tandem duplicated gene clusters in the EvsE data set for genBlastA (GB), Cui et al. (2007) (ML), and WU-Blast (WU). Ratio of specific groups was calculated as the number of genes resolved over the total number of genes in each tandem gene cluster. A gene is considered resolved if the HSP group overlaps with only one single gene in WormBase and the span similarity is ≥50%. Gapped and ungapped represent two independent BLAST results using either gapped setting or ungapped setting. GB alpha value is 0.5. ML distance threshold is 1000. Error bars, SE. (***) Statistical significance ($P < 0.001$) by paired Student's *t*-test.

coverage (>80%) and do not correspond to known genes. We found eight candidates. In particular, four of them contain putative novel genes that are relatively long (>300 amino acids) (Supplemental Table 1; Fig. 5). These putative novel genes will be tested in the laboratory to examine if they are real genes. Since the test gene set represents ~2% of *C. elegans* genome, we estimate that genBlastA will identify hundreds of novel homologous genes (including hundreds of long genes) in the entire genome. Our finding has thus demonstrated that genBlastA has the potential to identify novel paralogous genes.

Discussion

BLAST and related search programs have been widely used for identifying homologous sequences since they are sensitive and effective in finding homologous fragments for query genes. However, BLAST results often contain a large number of HSPs and can be challenging if not overwhelming for the end users. Our program genBlastA provides an effective way to interpret the large list of HSPs reported by BLAST in order to allow users to focus on targets they find interesting. genBlastA enables users to effectively identify homologous genomic regions that represent full-length candidate genes, rather than fragments of a gene (HSPs). Thus, genBlastA empowers users by allowing them to effectively identify candidate genes in target genomes. This will make BLAST and related programs even more useful.

Our analysis has clearly shown that genBlastA outperforms existing programs developed previously with similar objectives. In particular, genBlastA is very effective in grouping HSPs corresponding individual genes within tandem clusters of homologous genes. Both WU-BLAST and the program developed by Cui et al. (2007) failed in this task. Additionally, although ML performs better than WU-BLAST in resolving multiple paralogous genes in tandem clusters, the current ML program is not ready for

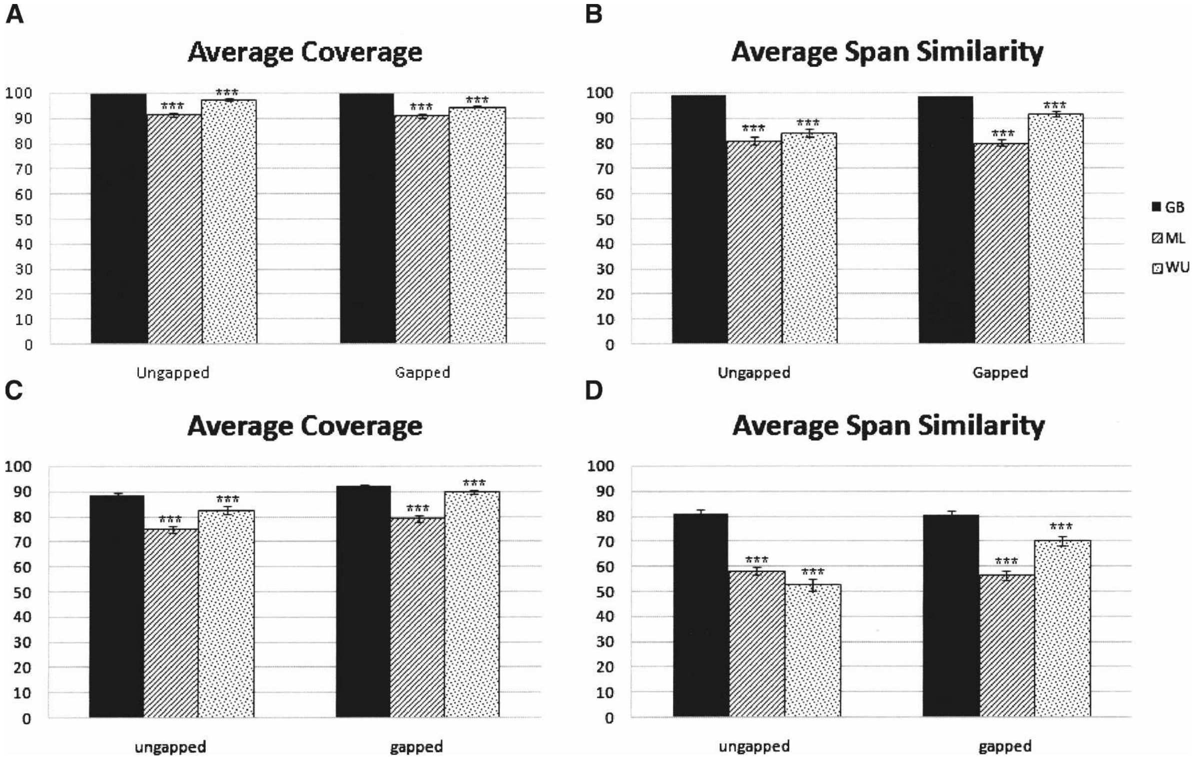

Figure 4. (*A*) Average coverage for EvsE data set. (*B*) Average span similarity for EvsE data set. (*C*) Average coverage for EvsB data set. (*D*) Average span similarity for EvsB data set. In all cases, figures represent averaged results over 464 test genes for three different programs genBlastA (GB), Cui et al. (2007) (ML), and WU-Blast (WU). Gapped and ungapped represent two independent BLAST results using either gapped setting or ungapped setting. Span similarity is calculated by Jaccard similarity. GB alpha value is 0.5. ML distance threshold is 1000. Error bars, SE. (***) Statistical significance ($P < 0.001$) by paired Student's *t*-test.

this job because the current ML program is not capable of removing random HSPs in the genomic regions.

The ability of effectively resolving HSP groups by genBlastA will enable users to take advantage of HSP groups, which are useful in several ways. First, genBlastA can be used by researchers to quickly locate candidate gene structures in the identified homologous genomic regions in the target genomes. Compared with the large collection of HSPs reported by BLAST and similar programs, ranked HSP groups provide much more useful information relevant to full-length target gene structures, instead of fragments of target genes. Since end users such as experimental biologists are usually more interested in genes, genBlastA makes search results more accessible and meaningful to them.

Second, genBlastA can be used to preprocess genomic DNA sequences for gene finding programs, including genewise (Birney et al. 2004) and exonerate (Slater and Birney 2005). Both genewise and exonerate are widely used for homology-based gene prediction. However, both programs, especially genewise, are computationally expensive when used to search for candidate genes in entire genomes. Their performance can be dramatically enhanced if their genomic search space is reduced. genBlastA, which is capable of identifying candidate genomic regions, can be used effectively to preprocess the genomic sequences in order to reduce search spaces. It can also be integrated into the program by Cui et al. (2007) to identify homologous genes.

Third, these HSPs can be used to resolve gene structures, either manually or computationally. Candidate gene models can be accurately defined by HSPs in each HSP group, intron–exon splicing information at the edges of HSPs, as well as the similarity between query and candidate genes. A gene prediction program based on this is being developed and will be reported separately.

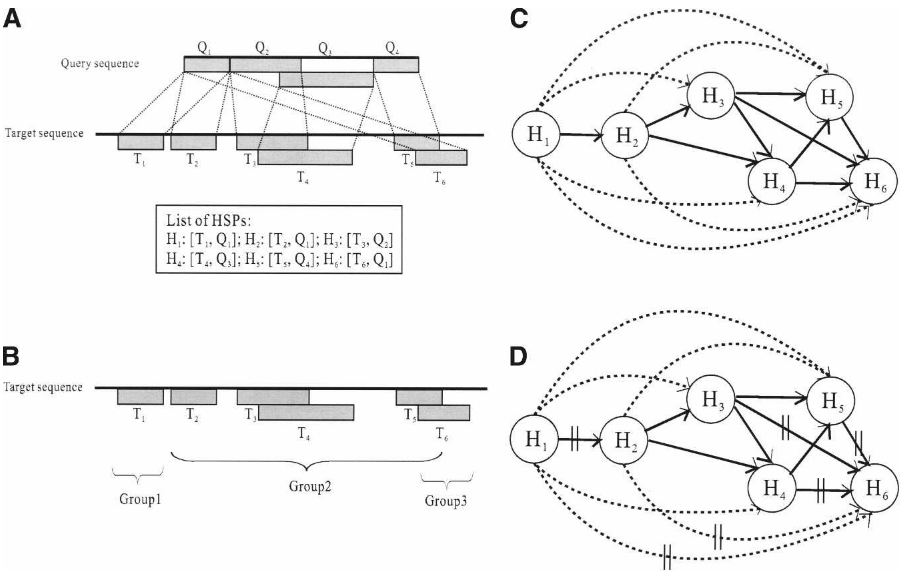

Figure 5. (*A*) HSPs returned by BLAST. Q_1, Q_2, Q_3, and Q_4 represent query segments, while T_1, T_2, T_3, T_4, T_5, and T_6 represent target segments. (*B*) Example of groups of HSPs. (*C*) The HSP graph, with solid lines representing edges and dotted edges indicating skip edges. (*D*) The HSP graph, with vertical bars indicating separating edges.

Methods

Problem definition

In this work, we study the following problem: given a query (gene) sequence, which is a protein (gene product), and a database of target genomic sequences, we want to identify all homologous genomic regions containing target genes (genes in the target sequences that are homologous to the query gene). First, as a preprocessing step, we apply BLAST to find local alignments between the query sequence and the target sequences. This step produces a list of HSPs, with each HSP containing the following information: (1) the target segment T and its location in the target sequence, and the corresponding query segment Q and its location in the query sequence, (2) an *E*-value, and (3) a PID value. In the second step, we filter and group the HSPs such that each group of HSPs forms a candidate region containing the target gene, called candidate gene region. genBlastA focuses on the second step.

An example of a list of HSPs is shown in Figure 5A, where the correspondence between the target segment (T) and query segment (Q) in an HSP is illustrated by dotted lines. For example, $[Q_1,T_1]$ and $[Q_1,T_2]$ represent two different HSPs. HSPs may overlap in terms of their genomic positions and/or their query correspondences. Note the HSPs shown in this figure are only for illustration purposes, although our algorithm is able to properly handle HSPs with various kinds of relationships.

Each genomic sequence has two strands—positive and negative. Each strand is considered a separate target sequence by genBlastA. Their only difference is the direction of alignment between the target gene and the query gene. Because each target sequence is independent and has its own list of HSPs, we process each target sequence separately to obtain the candidate gene regions for that sequence. Finally, all candidates for all target sequences are ranked into a single ranked list by their score as computed by our algorithm (discussed later). From now on, for brevity, all discussions will be based on a query sequence and a single positive-strand target sequence.

In this report, due to space limitation, we briefly present a novel graph-based method genBlastA to model the best grouping of HSPs as the problem of searching for shortest paths in a graph. Details of genBlastA algorithm are described in the Supplemental Data.

HSP groups

With each HSP target segment that matches a query segment, a sequential group of HSP target segments can collectively match a larger piece of the query sequence. We are interested in those groups of HSPs, which correspond to genes that are homologous to the query gene. Such groups are termed HSP groups. In general, there are different numbers of HSP groups in the target sequence for each query gene. If the query gene is not conserved in the target genome, then no HSP group can be found. If the query gene belongs to a multigene family (or the query gene has many paralogous genes), there will be multiple HSP groups in the target sequence, each representing a candidate region encoding a paralogous gene.

Consider the example in Figure 5A. T_3 and T_4 are in the same order as their query segments. So [Q_3,T_4] can be in the same group as [Q_2,T_3]. In fact, by merging T_3 and T_4 into one continuous target region, and merging their query segments into one continuous query region, we have a larger, thus better alignment. Figure 5B shows a possible grouping of HSPs that satisfies the sequential ordering and co-linearity requirements. Note that Group 1 and Group 3 have incomplete query gene coverage because a large portion of the query sequence is not covered by their query segments. In contrast, Group 2 covers the entire query sequence. A good HSP group should have large query coverage.

For a group of HSPs, the combined region of their query segments should cover the query sequence as much as possible. In Figure 5B, Group 2 is better than either Group 1 or Group 3 because it covers a larger region of the query sequence.

Graph modeling

An HSP graph is a graph representation that captures the above requirements on HSP groups. Each HSP is represented by a node, with edges that model the sequential ordering of the HSP target segments and edges that skip HSPs. An HSP grouping is modeled by grouping the nodes on a path, such that each group covers as many query segments as possible while preserving colinearity. By using a length metric (Supplemental Data), we will show that an optimal HSP group is a shortest path in the HSP graph.

Figure 5C shows the HSP graph for the HSPs in Figure 5A. The dotted edges are skip edges. Each path in the graph represents a way of selecting HSPs along the path. With skip edges, the HSP graph provides a complete search space for all possible groupings of HSPs. The number of skip edges can be very large. However, after introducing a length metric on edges (Supplemental Data), we will show that many skip edges can be removed without affecting the result. Our program genBlastA will not construct such skip edges, thus dramatically increasing the efficiency of genBlastA.

In Figure 5D, to distinguish these two types of edges, we add a vertical bar to each separating edge. For example, $H_1 \rightarrow H_2$ is a separating edge, which means that its source node and destination node should belong to different HSP groups. The skip edge $H_1 \rightarrow H_3$ is an extension edge, and the skip edge $H_1 \rightarrow H_6$ is a separating edge.

With extension edges and separating edges, each path in the HSP graph represents a way of filtering and grouping HSPs: As we traverse a path, following an extension edge extends the current HSP group to include the destination node, and following a separating edge ends the current HSP group at its source node and starts a new HSP group at its destination node. If an extension edge is a skip edge, following the edge will skip over the nodes on the paths that are shortcut by the edge. In this sense, the HSP graph provides a complete search space for filtering and grouping HSPs.

The single-source shortest path algorithm for a directed acyclic graph can be done efficiently in $O(E)$ time, where E is the number of edges (Manber 1989). Executing this algorithm once for each possible starting node H_1, the total running time is $O(E \cdot V)$, where V is the number of end nodes of separating edges and is bounded by the number of HSPs.

Acknowledgments

We thank Fiona Brinkman and Robert Johnsen for critical reading of the manuscript and constructive comments, and three anonymous reviewers for constructive input. K.W. and N.C. are supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants. J.S.-C.C. is supported by a NSERC Postgraduate Scholarship. Some support was also provided by the SFU Community Trust funded BCID Project. N.C. is also a Michael Smith Foundation for Health Research (MSFHR) Scholar.

References

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J. 1990. Basic local alignment search tool. *J. Mol. Biol.* **215:** 403–410.

Bentley, D.R. 2006. Whole-genome re-sequencing. *Curr. Opin. Genet. Dev.* **16:** 545–552.

Birney, E., Clamp, M., and Durbin, R. 2004. GeneWise and Genomewise. *Genome Res.* **14:** 988–995.

Chen, N., Harris, T.W., Antoshechkin, I., Bastiani, C., Bieri, T., Blasiar, D., Bradnam, K., Canaran, P., Chan, J., Chen, C.K., et al. 2005. WormBase: A comprehensive data resource for *Caenorhabditis* biology and genomics. *Nucleic Acids Res.* **33:** D383–D389.

Coghlan, A. and Wolfe, K.H. 2002. Fourfold faster rate of genome rearrangement in nematodes than in *Drosophila*. *Genome Res.* **12:** 857–867.

Consortium. 1998. Genome sequence of the nematode *C. elegans*: A platform for investigating biology. *Science* **282:** 2012-2018.

Cui, X., Vinar, T., Brejova, B., Shasha, D., and Li, M. 2007. Homology search for genes. *Bioinformatics* **23:** i97–i103.

Florea, L., Hartzell, G., Zhang, Z., Rubin, G.M., and Miller, W. 1998. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res.* **8:** 967–974.

Hardison, R.C. 2003. Comparative genomics. *PLoS Biol.* **1:** e58. doi: 10.1371/journal.pbio.0000058.

Kent, W.J. 2002. BLAT—The BLAST-like alignment tool. *Genome Res.* **12:** 656–664.

Lander, E.S., Linton, L.M., Birren, B., Nusbaum, C., Zody, M.C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., et al. 2001. Initial sequencing and analysis of the human genome. *Nature* **409:** 860–921.

Lopez, R., Silventoinen, V., Robinson, S., Kibria, A., and Gish, W. 2003. WU-Blast2 server at the European Bioinformatics Institute. *Nucleic Acids Res.* **31:** 3795–3798.

Manber, U. 1989. *Introduction to algorithms: A creative approach*. Addison-Wesley, Reading, MA.

Margulies, M., Egholm, M., Altman, W.E., Attiya, S., Bader, J.S., Bemben, L.A., Berka, J., Braverman, M.S., Chen, Y.J., Chen, Z., et al. 2005. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **437:** 376–380.

Pearson, W.R. and Lipman, D.J. 1988. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci.* **85:** 2444–2448.

Robertson, H.M. and Thomas, J.H. 2006. The putative chemoreceptor families of C. elegans. In *WormBook* (ed. The *C. elegans* Research Community, WormBook). doi: 10.1895/wormbook.1.66.1, http://www.wormbook.org.

Slater, G.S. and Birney, E. 2005. Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics* **6:** 31. doi: 10.1186/1471-2105-6-31.

Stein, L.D., Bao, Z., Blasiar, D., Blumenthal, T., Brent, M.R., Chen, N., Chinwalla, A., Clarke, L., Clee, C., Coghlan, A., et al. 2003. The genome sequence of *Caenohabditis briggsae*: A platform for comparative genomics. *PLoS Biol.* **1:** e45. doi. 10.1371/journal.pbio.0000045.

Suyama, M., Torrents, D., and Bork, P. 2004. BLAST2GENE, a comprehensive conversion of BLAST output into independent genes and gene fragments. *Bioinformatics* **20:** 1968–1970.

Venter, J.C., Adams, M.D., Myers, E.W., Li, P.W., Mural, R.J., Sutton, G.G., Smith, H.O., Yandell, M., Evans, C.A., Holt, R.A., et al. 2001. The sequence of the human genome. *Science* **291:** 1304–1351.

Waterston, R.H., Lindblad-Toh, K., Birney, E., Rogers, J., Abril, J.F., Agarwal, P., Agarwala, R., Ainscough, R., Alexandersson, M., An, P., et al. 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature* **420:** 520–562.

Zhang, H. 2003. Alignment of BLAST high-scoring segment pairs based on the longest increasing subsequence algorithm. *Bioinformatics* **19:** 1391–1396.

Received June 9, 2008; accepted in revised form September 29, 2008.