JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 11, Number 5, 2004

Pp. 800–811

ThurGood: Evaluating Assembly-to-Assembly Mapping

HAGIT SHATKAY,[1,5] JASON MILLER,[5] CLARK MOBARRY,[2,5] MICHAEL FLANIGAN,[3,5] SHIBU YOOSEPH,[4,5] and GRANGER SUTTON[4,5]

ABSTRACT

The alignment and mapping of large genomic sequences is the focus of much recent research. However, relatively little has been done so far about testing and validating alignment methods. We introduce criteria and new tools we have developed for alignment evaluation. These tools have already proved useful in the evaluation and ranking of several methods for assembly-to-assembly mapping, which were recently used to map multiple versions of the human genome to each other (Istrail *et al.*, 2004).

Key words: evaluation, assembly-to-assembly mapping, genome, alignment.

1. INTRODUCTION

SEQUENCING AND ASSEMBLY TECHNOLOGY has rapidly matured during the last decade (Myers, 1999b; Venter *et al.*, 2001; Waterston *et al.*, 2002), making complete genomes from a wide variety of organisms (see, e.g., TIGR) readily available. Moreover, multiple sequencing efforts using different assembly pipelines such as in the case of the human genome, have produced multiple assembled versions for same-species genomes. The many available genomic sequences are currently being studied, compared and contrasted for a variety of reasons, ranging from the study of the sequencing and assembly methods themselves to cross-species study of the genome in the context of comparative genomics, looking for synteny and difference among related species.

A first step in genome comparison is the *alignment* of two genomic sequences to each other. Alignment methods are commonly partitioned into two types: *global alignment* (Needleman and Wunsch, 1970; Delcher *et al.*, 1999; Bray *et al.*, 2003; Brudno *et al.*, 2003b), which aligns entire genomic sequences, without examining potential rearrangements, and *local alignment* (Smith and Waterman, 1981; Altschul *et al.*, 1990; Schwartz *et al.*, 2003), which finds relatively short regions of high similarity within two genomic sequences without considering the high-level correspondence (or evolutionary relationships) between the genomic sequences as a whole. Alignment of two complete genomes must combine multiple granularity levels, exposing orthologous regions through *local* alignment, while ignoring spurious, local

[1]School of Computing, Queen's University, Kingston, Ontario.
[2]White Oak Technologies, 1300 Spring St., Suite 320, Silver Spring, MD 20910.
[3]Steck Consulting LLC, 2121 K St., NW, Washington, DC 20037.
[4]J. Craig Venter Institute, 9704 Medical Center Dr., Rockville, MD 20850.
[5]This work was done while all authors were with the Informatics Research Group at Celera/Applied Biosystems.

similarities among repetitive short regions present on both genomes, and considering the *global* correspondence between large regions of the sequences. Recent tools that combine the global and local approach, supporting the alignment of whole genomes while taking into account possible rearrangement and shuffling (during evolution or during the assembly process), are Brudno *et al.*'s Glocal alignment suite (Brudno *et al.*, 2003a) and the A2Amapper (assembly-to-assembly mapper) pipeline (Istrail *et al.*, 2004; Mobarry and Sutton, 2003).

While much effort is devoted to the alignment process, methods for *evaluating* the various alignment techniques are still wanting. To quote Schwartz *et al.* (2003), "... it is an order of magnitude easier to design two good programs than to tell which one is better." For instance, in the context of mouse–human alignment (Schwartz *et al.*, 2003; Brudno *et al.*, 2003a), sensitivity is estimated as the percentage of aligned base-pairs (scoring above a certain similarity threshold). Specificity, on the other hand, is checked as done for BLASTZ (Schwartz *et al.*, 2003), relying on the known homology between chromosome 20 in human and 2 in mouse (Searle *et al.*, 1989). That is, if most of human chromosome 20 is aligned to mouse chromosome 2, the alignment is considered specific. Another specificity related test is conducted for BLASTZ by aligning the reversed (but not Watson–Crick-complement) mouse sequence, to the human, showing significantly fewer matches than those found between the actual mouse and human sequences. This method is used there to estimate spurious matches in the true assembly-to-assembly comparison.

These measures provide some insight into the alignment quality, but leave much to be desired. In terms of sensitivity, the above evaluation does not confirm that the regions mapped to each other as syntenic indeed evolutionarily correspond to each other. Moreover, an evaluation that measures the number of bases in matches exceeding a threshold score strongly depends on the specific scoring scheme used to calculate the matches in the alignment process itself. In addition, the specificity evaluation used for the case of mouse and man is obviously limited. For same-species alignment, we could extend it and look for same-chromosome alignment as a specificity check, but this again will not penalize misalignment within chromosomes, while unduly penalizing the alignment for correctly identifying genome rearrangements across chromosomes.

This paper presents a suite of methods, *ThurGood*, that explicitly addresses the evaluation of alignment programs which aim at finding corresponding regions between two assembled genomes, of either the same or different species. We view the alignment problem for two genomic sequences as that of creating a *one-to-one (1-1) mapping* between them. That is, *alignment* is the mapping of regions that are either evolutionarily related (when comparing different species) or constitute the same genomic region (when comparing assemblies or genome instances of the same species). We note that this view is naturally extended to cover many-to-one and many-to-many mappings, where multiple repeats of a genomic region on one sequence are mapped to the same region on another. A brief review of the evaluated mapping methods is provided in Section 2.[1] We examine two main evaluation directions:

- **Feature conservation.** The mapping should conserve biological features such as genes, exons, and regulatory regions. For instance, if a specific exon is located on a particular region of the genome and is present on both assemblies, the region containing the exon on one assembly should be 1-1-mapped to the region containing the exon on the other.
- **Simulation.** This kind of evaluation assumes that given two genomic sequences, a good 1-1-mapping algorithm should satisfy certain properties, which can be quantified in terms of *specificity* and *sensitivity*. The use of partly simulated data allows us to directly examine both measures.

We present several evaluation methods that we have devised and the tools we implemented based on both types of criteria. Specifically, Section 3 describes the $M4$ pipeline for exon based evaluation. Section 4 introduces *Mutagen*, a program that mutates a genome according to certain parameters while keeping a log of the true matches between the mutated and the original genome. The log constitutes a gold standard that allows specificity and sensitivity evaluation and along with other material is available as a benchmark (*www.cs.queensu.ca/BioInfo/MutaData*). That section also briefly discusses two simulated noise models that

[1]Note that this paper is not concerned with mapping methods or their quality, but rather with *tools applied to their evaluation.* See Istrail *et al.* (2004) and Mobarry and Sutton (2003) for more detail on the mapping methods.

are useful for the evaluation process. These techniques and tools were used to select the mapping method for the first comparison of human genome assemblies (Istrail *et al.*, 2004), and Section 5 demonstrates the results of their application to several candidate mapping methods.

2. ASSEMBLY TO ASSEMBLY MAPPING

When discussing the mapping of two assemblies, we distinguish between three related concepts: *genome*, a genomic sequence representing the consensus among many individual haplotypes of a common species; *genome instance*, the actual DNA sequence of a single individual (haplotype); and *assembly*, the genomic sequence resulting from a sequencing effort applied to either a particular species or an individual. In either case, the assembly is an *approximation* of the true sequence—be it a haplotype or a consensus sequence—and suffers various errors such as inversion, substitution, deletion, and insertion. These and other variations can be introduced through errors in any phase of the sequencing and assembly process. However, similar types of variation also naturally occur, due to evolution, in actual DNA sequences of related and even of the same species.

Assembly-to-assembly mapping is well motivated by a variety of applications: *variation studies* between individuals of the same species; *comparative genomics*, in which multiple instances of the same species, as well as assemblies of different species, are compared for studying evolutionary events and trends; and *sequence validation*, where the mapping is between two assembled versions of the same species, obtained through different methods or based on different data. The motivation for the latter is to compare the two versions by investigating their similarities and differences. Additionally, in the current era in which many assemblies are still *drafts* rather than confirmed and completed sequences, the mapping allows one to *track biological features* from older assembly versions to newer ones, maintaining backward compatibility as versions are updated.

The methods we compare here combine local and global alignment, in the context of sequence validation and feature-tracking across assemblies (same species analysis), and are all based on a 1-1-mapping approach (Mobarry and Sutton, 2003; Istrail *et al.*, 2004; source: *https://panther.appliedbiosystems.com/pub/genealtsplicingair*) which we briefly review. We note that these methods are the only ones available to us that are fast enough (finish within hours or at most days) to allow multiple comparative runs. No other available alignment method that we know of can currently handle full sized human assembly within a reasonable time (although collaboration is ongoing to apply our benchmark to another published alignment technique).

The five methods, dubbed *He*, *Ne*, *Ca*, *Hg*, and *Cs*, differ from each other in the specific choices made on each of the following steps, as summarized in Table 1.

1. *Putative anchors* which are *unique, maximal, and exact matches* are identified using two alternative techniques, differing in their uniqueness measure. The first, dubbed *K-mer*, uses coalesced unique k-mers—sequences of length k, that are identical and unique on both assemblies. The second, dubbed *MIUM*, uses coalesced *maximal inexactly unique matches* (introduced by Ross Lippert), which are identical subsequences present on both assemblies, such that all their subwords above a certain length, l, are also unique on both assemblies, but any extension of them is either not present on both assemblies or violates the unique-subword property.
2. Putative anchors are either all retained—this option is denoted as *none* for *no-filtering*—or are *filtered* so that only matches with a potential to participate in a high-coverage alignment are retained. Filtering

TABLE 1. SUMMARY OF THE MAPPING METHODS USED

Method	*Putative anchors*	*Filtering/ chaining*	*Match extension*	*Unique filtering*
He	K-mer	None	N	N
Ne	K-mer	Local Chaining	Y	N
Ca	K-mer	Global Chaining	Y	Y
Hg	K-mer	Local Chaining	Y	Y
Cs	MIUM	Local Chaining	Y	N

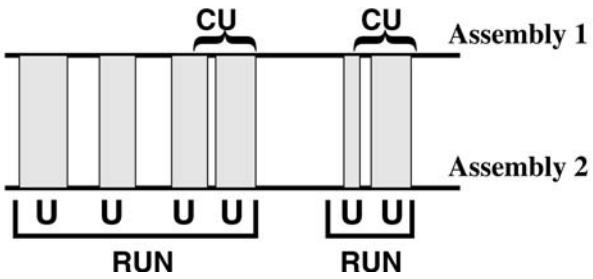

FIG. 1. Three levels of 1-1-mapped regions between assemblies: Ungapped matches (U), Chained ungapped matches (CU), and Runs.

is based on either *global chaining* or *local chaining* as follows. Global chaining groups together matches while allowing translocations and reversals along an entire chromosome. Local chaining does not allow gaps of over 100 Kbp between putative anchors in the same group. Each anchor in a chain is assigned a score which is the total length of the putative anchors in its chain. Chained putative anchors whose score is above a certain threshold are retained while the others are discarded.

3. Retained putative anchors are either extended in a *match-extension* step (Y in the table), by appending surrounding bases while allowing 5% substitution errors, or remain as they are (N in the table). The putative anchors are then merged into larger ungapped matches.
4. The merged putative anchors are either reexamined, filtered, and modified to ensure their uniqueness in both assemblies, in a step called *unique filtering* (Y), or left as they are (N).

Putative anchors extending at least a minimum length (40 bp) are called *anchors*, while the rest are discarded. We then chain together anchors consistent in order and orientation that are separated by no more than 100 Kbp, into local chains that exclusively span regions of both assemblies. These chains are denoted *Runs*, and define a global mapping between the assemblies. Runs containing fewer than 100 bp anchor coverage within their whole span are discarded. From the remaining runs, those that are within 100 Kbp apart are merged into longer runs when possible. The gap regions between consecutive anchors within a run are then locally aligned, producing ungapped matches which must satisfy 90% identity. The latter matches along with the anchors are denoted by U (the total set of ungapped matches). The notions U and *Runs* are demonstrated in Fig. 1 and are further used in Sections 3 and 5 (where the notation CU is explained as well).

3. M4: BIOLOGICAL FEATURES CONSERVATION

Biological features are any genomic regions corresponding to functional units. Some examples are genes, exons, introns, regulatory regions, etc. We limit our discussion to exons, since a large body of cDNA sequences is available from RefSeq (Pruitt and Maglott, 2001), making them readily available. Additionally, exons are a "classical" feature due to their fundamental role in protein coding. When mapping between two genomic sequences, one expects that a feature that is present on both will be present within regions that are mapped to each other. This is of utmost importance when using assembly-to-assembly mapping for *tracking* annotated biological features from one assembly version to the next, or from the genome of a well-studied species to a related newly sequenced one.

Therefore, our underlying hypothesis is that a good 1-1-mapping of genomic sequences should conserve exons; if an exon is present on two assemblies, the region containing it on one, should be 1-1-mapped to the corresponding region on the other. An ideal setting is shown in Fig. 2.

The figure depicts two genome assemblies, A_1 and A_2, a region that is 1-1-mapped between them and a RefSeq exon that is found on both assemblies within the 1-1-mapped region.

A preliminary step is to find the respective positions of exons on both assemblies (*human assemblies* for the results discussed here) and on the RefSeq cDNA. As RefSeq cDNAs are not annotated with exons, these positions are *inferred* by utilizing a set of 19,667 human cDNA reference sequences (all *homo sapiens* entries from the RefSeq database, prefixed by "NM"), along with a program based on the *sim4* cDNA

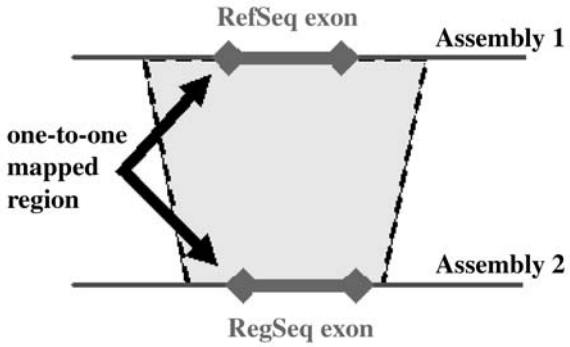

FIG. 2. An exon (bold fragment) is contained within the 1-1-mapped region (between dashed lines).

aligner (Florea *et al.*, 1998). Each of the cDNA sequences is aligned to each assembly separately and is fragmented into exons around canonical splice sites (AG, GT) on the assembly, while preserving the order imposed by the cDNA. Each RefSeq fragment bears high similarity to the assembly-fragment to which it is aligned by the program. We can denote each resulting *inferred exon* as a quadruple $\langle A^1{}_s, A^1{}_e, R^1{}_s, R^1{}_e \rangle$ where $A^1{}_s$, $A^1{}_e$ are the exon's respective start and end positions on the assembly A^1, and $R^1{}_s$, $R^1{}_e$ are its start and end coordinates on the RefSeq cDNA as found by alignment to A^1.

Following this process, we apply a pipeline we devised, dubbed $M4$, that measures how many inferred RefSeq exons, of all those found on both assemblies, are indeed located within 1-1-mapped regions. Given two assemblies that have been 1-1-mapped by a candidate alignment algorithm, along with exons mapped to each assembly as described above, we check whether each exon that was mapped to both assemblies is indeed contained within a 1-1-mapped region.

Two issues must be addressed to conduct such a check. First, what does it mean for an exon to be mapped on *both* assemblies, given that exons are inferred by cDNA alignment to each assembly separately? Second, what is the extent or the granularity of a 1-1-mapped region?

We address the first issue as depicted in Fig. 3. If exon E has the coordinates $\langle A^1{}_s, A^1{}_e, R^1{}_s, R^1{}_e \rangle$ when a cDNA is aligned to assembly A^1, while Exon E' has coordinates $\langle A^2{}_s, A^2{}_e, R^2{}_s, R^2{}_e \rangle$ by aligning the same cDNA to assembly A^2, E and E' are viewed as the *same exon* on the two assemblies if and only if $R^1{}_s$ and $R^2{}_s$ are within a few bases of each other (5 bases in our case) and the same holds for $R^1{}_e$ and $R^2{}_e$.

As for the second issue, we look at three levels of 1-1-mapped regions (see Fig. 1): The fine-granularity ungapped matches, U, the coarse chained and gapped matches, *Runs*, both of which were defined in Section 2, as well as an intermediate level of *chained-U's*, denoted CU. Those are sequences of close U-regions with a gap of at most X bases between them ($X = 5$ bp is the threshold we use here).

The $M4$ pipeline takes a 1-1-mapping of any two assemblies, along with the exons mapping to each of them, and produces statistics such as the number and percentage of exons that are within 1-1-mapped

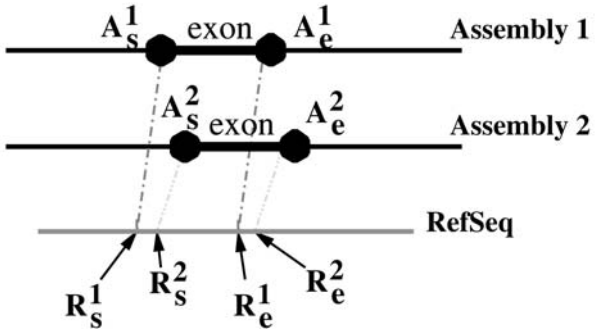

FIG. 3. The mapped exons on assemblies A_1 and A_2 have similar boundaries on the RefSeq, and hence are viewed as "same exon."

regions on each assembly and the number of bases within such exons. The pipeline itself requires several minutes (wall-clock, single alpha processor, 1.5 Mb memory) to evaluate the mapping of the complete human genome assembly with respect to exons derived from 19,667 RefSeq transcripts mapped to the whole genome. We note that the preceding step of mapping the entire transcripts set to each assembly is also not prohibitively intensive and requires about three CPU hours (single alpha processor). Experiments and results running this pipeline are provided in Section 5.1.

4. CONTROLLED EVALUATION: MUTAGEN AND MARKOV

While it is important for a 1-1-mapping to respect biological features, and while such a check validates to some extent the mapping's *sensitivity* (in terms of the percentage of homologous exons that the mapper matches to each other), curated biological features are currently too few and too sparse with respect to the whole genome to support the evaluation of the mapping's specificity. Moreover, the function of most of the genome is yet to be determined, relying on comparative studies, which in turn are based on assembly-alignment. Thus, controlled evaluation independent of the current status of biological databases is needed.

We examined several controlled ways to evaluate assembly-to-assembly mappings, most significant of which is the introduction of Mutagen: a mutation-simulator that modifies a genomic sequence according to certain parameters provided to it, while keeping a record of the true mappings between the corresponding regions of the original and the mutated sequences. Mutagen's parameters can be set to simulate assembly errors as well as evolution based mutation. The idea of simulation has already been used within genomics. Examples include Rose[2] (Stoye *et al.*, 1998), a generator of evolutionary-like mutations, and Myers' celsim (Myers, 1999a), a generator of data for validating the genome-assembly process. The set of mutations we use (given in Section 4.1) is more extensive than any of the above. It is designed specifically for supporting assembly-to-assembly comparison, where in addition to the evolutionary and haplotypical differences, errors in assembly—including inversions and repetitions—need to be accounted for. Moreover, while simulation was used to a limited extent, on sequences of 1,000 bases with isolated types of mutation, when aligning human and mouse genomes (Waterston *et al.*, 2002, supplementary material), none of the simulators mentioned above was applied to the validation of alignment over complete genomic sequences or supported such a validation.

Besides Mutagen's output, we use two additional synthetic genomic sequences to assist in evaluating the quality of 1-1-mappings. One is simply the reverse of the assembly, as used for testing BLASTZ (Schwartz *et al.*, 2003). The other is obtained by training a 10-order Markov model from a true assembly, sampling from it a genome-like sequence, and mapping the true assembly to the genome-like noise. The mapping to the reverse and to the Markov sequences allows us to estimate and characterize noisy chance-matches in mappings and therefore to estimate the significance of all identified matches.

4.1. Mutagen

Mutagen's goal is to produce a gold-standard mapping between a true assembly G and another assembly G' that deviates from G due to assembly errors or to evolution. The first type of deviation mostly supports the assessment of assembly-to-assembly and same-species mapping, while the second one (combined with the first) is appropriate when evaluating cross-species mapping.

Mutagen thus takes an assembly, G, and produces as output a mutated assembly G'. The mutant G' is the result of applying a series of *transformations* from a predefined set of both single-nucleotide polymorphisms (SNPs, specifically, point substitutions and deletions) and multinucleotide mutations (region duplication, region deletion, translocation, inversion, and the insertion of Alu repeats for a couple of Alu sequences; the latter two mutations are unique to Mutagen). The transformations' extent and application rate is based on a set of parameters that can be set according to the mapping problem at hand (assembly to assembly, two species and the evolutionary distance between them, etc.) We distinguish between two types of translocation:

[2]While this manuscript was under revision, a study actually using Rose to compare alignment methods over relatively short (10 Kbp) noncoding Drosophila-like DNA was published (Pollard *et al.*, 2004).

local, where the translocated fragment is inserted within the same chromosome, and *global* (potentially interchromosome), which can reinsert a region anywhere on the assembly. The parameterization used for the experiments reported here is described in Section 5. These parameters do not model evolution, and the operators do not currently include all evolutionary events such as crossovers and tandem repeat expansions, as we concentrate here on same-species assembly comparison.

Mutagen, like celsim (Myers, 1999a), implements deletions but not arbitrary insertions, since deletions in the mutated sequence can be viewed as insertions in the original sequence. Also, like celsim, Mutagen first applies all multinucleotide operators, while avoiding overlaps between deletions and any of the other operators (inversion and translocation), and follows those by the SNP operators. The repeat insertion that Mutagen supports, as well as the inversion, are specific to our goal of assembly-to-assembly comparison. Moreover, Mutagen, unlike any earlier simulators, produces an alignment between two sequences of which one is a *true assembly*, G, while the other, G', is the result of simulating mutations over the true assembly G. While mutating G to create G', Mutagen logs the transformations it applied, and unlike earlier simulators, it also produces a complete fine-granularity mapping of each base in G to its (possibly mutated) image on G'.

One can thus evaluate any alignment algorithm by applying it to the pair G and G' and checking whether each base on G is mapped to its corresponding base on G' as it appears in the Mutagen log.

4.1.1. Tofumi: Obtaining summary statistics. Once a mutation G' is produced, we use each candidate mapping program to map between G and G' and calculate specificity and sensitivity with respect to the true mapping as recorded in Mutagen's log. This evaluation requires checking for overlaps between the 1-1-mapped regions produced by the mapper and comparing those against the ones produced by Mutagen. To do this, we used an in-memory data management system with an advanced querying capability called Tofumi, which is an extension of Celamy (Turner *et al.*, 2001).

Tofumi's query language is optimized to perform queries over features that have genomic coordinates. It includes the basic Boolean operators, as well as specialized operators for handling interval length and feature depth, and a visualization capability. The results of Tofumi queries can be named and used in subsequent queries. We can thus load a set of intervals representing the 1-1-mapped regions according to Mutagen, as well as according to any candidate mapping program, and form queries to investigate these regions and obtain their statistical characterization, including how many of the intervals agree with each other and how many base pairs are included in the agreement. Using this capability, we compared the results of mapping programs against the gold standard in Mutagen's log as discussed in Section 5.2.

4.2. Assembly-to-noise mapping: Markov and reverse

As another controlled quality check, we validate that a mapper is discriminating and does not find significant 1-1-matches where there should not be any. This is done more as a sanity check than as a formal method. Note that while this check does not produce exact specificity results (as measured in the Mutagen analysis), it is a form of a high-level specificity validation, reassuring that a mapping program does not find significant alignment between an assembly and a nongenomic sequence of a similar base composition. Thus, this section addresses the need for mapping a genome-like sequence, which is not a result of sequencing and assembly, to a true genome assembly. One expects a mapper to find short sporadic matches, when mapping a genomic sequence to a nongenomic one, without contiguous homologous regions.

We used two methods to produce genome-like nongenomic sequences, where by "genome-like" we mean that the sequence has base composition and distribution similar to a true genome but no biological meaning. First, we simply reversed (without complementing) the genomic sequence, as was done before (Schwartz *et al.*, 2003). This method indeed conserves the base composition, but does not conserve the distribution of subsequences longer than 1 bp.

As an alternative, by taking statistics from the whole Celera *WGSA*[3] assembly (submitted to GenBank, account number *AADD00000000*), we created a 10-order Markov model (Billingsley, 1959). Essentially, this is a collection of 5^{11} conditional probabilities that for every 10-mer $x_1, \ldots, x_{10}$ and a base x_{11} where $x_1, \ldots, x_{11} \in \{A, C, G, T, N\}$ (N is an unidentified base), specifies the probability $Pr(x_{11}|x_1, \ldots, x_{10})$.

[3] *WGSA* is the Whole Genome Shotgun Assembly, produced from shotgun data alone without any shredded pre-sequenced data.

To create the model, we calculated maximum-likelihood estimates for all these probabilities based on the frequencies of occurrence in the assembly. For every chromosome in the assembly, we recorded its length and also saved its prefix, consisting of its first 10 bases, to be a starting point. Starting every chromosome with its original prefix in the true assembly sequence and using Monte Carlo sampling to generate the rest of the sequence from the model, we created "chromosome-like" sequences that had base and subsequence distributions that are similar to those of the true assembly, but are obviously not a biologically-valid genome.

Both the reverse and the Markov-sampled sequence were mapped to the true assembly from which they were derived to show that the mapper produces short and sporadic matches between a true assembly and a mockup. Any mapper that produces similar distributions of matches (in terms of length and frequency) for mapping two true genome assemblies to each other, as when mapping a mockup version to a true genomic sequence, is obviously flawed.

5. EXPERIMENTS AND RESULTS

We applied our evaluation to several assembly-to-assembly mapping methods as described in Section 2.

5.1. Exon conservation

Our first analysis examines, for several assembly-to-assembly mappers, how many exons of those present on both of the compared assemblies are indeed enclosed within regions mapped to each other by the mapper. We conducted experiments using three granularity levels for defining *mapped regions*, as described before, namely ungapped matches (U), chained-U-matches (CU with gaps that do not exceed 5 bases), and *Runs*.

The assemblies compared were NCBI's Build 34 ($B34$), and Celera's *WGSA*. Table 2 summarizes the results. The number of exons found by a version of *sim4* (Florea *et al.*, 1998) on *WGSA* is 195,145 while the number found on $B34$ is 202,026.

Of those, 185,847 were found on both assemblies. Each column states the number of exons found within 1-1-mapped regions using each mapping method, as well as the percentage with respect to the total number of exons found on both. As demonstrated by the results, the vast majority of the exons that are present on both assemblies are indeed contained within regions that are mapped to each other, even when we examine the mapping in the finest granularity level (U). While the differences between the methods are small, the *Ne* method has the highest number and percentage of exons that are contained within 1-1-mapped regions. The second best mapper under this criterion is *Hg*. Note that this agrees with the results from the next validation method, which show that *Ne* is indeed more sensitive than *Hg*, as well as with our notion that exon conservation across 1-1-mapped regions correlates with sensitivity.

5.2. Mutagen results

Using Mutagen, we altered NCBI's Build 33 sequence, ($B33$), introducing mutations according to rates defined by a parameter setting given in Table 3. The mutated result is denoted by *BMuta*33. The parameters shown are based on rough overestimates of the differences between Celera's *WGSA* assembly and UCSC's *hg6* [UCSC(hg6)]. Cursory visual inspection of dot plots and various alignments suggest that these two

TABLE 2. EXON CONTAINMENT IN MAPPED REGIONS OF THREE GRANULARITIES ACROSS VARIOUS MAPPING METHODS

Mapping method	*U* *# of exons (%)*	*CU* *# of exons (%)*	*Runs* *# of exons (%)*
He	178947 (96.29%)	183287 (98.62%)	184174 (99.10%)
Ne	179313 (96.48%)	183636 (98.81%)	184505 (99.28%)
Ca	179117 (96.38%)	183451 (98.71%)	184337 (99.19%)
Hg	179138 (96.39%)	183473 (98.72%)	184359 (99.20%)

TABLE 3. MUTAGEN PARAMETER SETTING FOR CREATING *BMuta*33

Mutation type	*Rate (per Gbp)*	*Size range (in Kbp)*	*Other information*
Point deletion	500,000	NA	NA
Point substitution	2,000,000	NA	NA
Large deletion	10,000	0.002–0.5	NA
Repeat	1,000	1–50	At most 2 copies
Alu	2,000	NA	L35531, U67801[a]
Local translocation	10,000	1–30	Distance: 1-250 Kbp
Global translocation	100	1–300	NA
Inversion	1,000	1–20	NA

[a]GenBank accession numbers for the Alus used.

sequences are the furthest apart among about 14 different versions of the human genome we have examined (Istrail *et al.*, 2004). These two assemblies differ in assembly algorithms, DNA sequencing means, and donor source. Note that our parameters do not accurately model the differences between *hg6* and *WGSA*, but rather are an *overestimate* based on them, producing a "tough" test case—though still grounded in reality—for the various mapping procedures. Specifically, our local chromosome translocation rate is much higher than expected between any two of the human assemblies we examined.

We report the results of using five candidate algorithms, described in Section 2, to map between *B*33 and *BMuta*33. As stated before, Mutagen produces a log which records for each base on the original assembly (*B*33 in our case) its counterpart on the other assembly (*BMuta*33). Note that in the case of deletion, some bases of *B*33 remain without a counterpart on *BMuta*33. Moreover, duplications and repeated translocations may mutate some regions beyond recognition. This causes several potential matches between a single region on one assembly to several counterparts on the other to seem equally good, while Mutagen still logs a single "true" match based on the operators it applied. The evaluation criteria is thus strict and does not give the mapper the "benefit-of-the-doubt" in such cases. Additional analysis of the mapping output helps us isolate these artifacts. Note though that this stringency does not bias the comparative study, as all mapping algorithms are evaluated against the same dataset using the same stringent criteria.

We evaluate the mapping results against Mutagen's log, in terms of both sensitivity and specificity. *Sensitivity* is the percentage of bases whose mapping by the algorithm agrees with the Mutagen log, with respect to all 1-1-mapped bases *recorded by Mutagen*. It is calculated as $(100 \cdot Agree/Mu)$, where *Agree* is the number of bases whose mapping by the algorithm agrees with the correct mapping according to the Mutagen log, and *Mu* is the total number of bases mapped by Mutagen. *Specificity* is the percentage of bases whose mapping by the algorithm agrees with the Mutagen log, with respect to all the bases that were *1-1-mapped by the algorithm*, calculated as $(100 \cdot Agree/Mapped)$, where *Mapped* is the total number of bases 1-1-mapped by the algorithm. To accentuate the differences, we examine the *error rates* corresponding to *lack* of sensitivity and specificity. Namely, we denote by $ER_1 = 100 \cdot (1 - Agree/Mu)$ the percentage of bases that are mapped by Mutagen but *not* by the algorithm (lack of sensitivity), while $ER_2 = 100 \cdot (1 - Agree/Mapped)$ similarly expresses lack of specificity. We calculated these two kinds of errors, as well as their sum (*Total ER*) for the mapping obtained from each of the algorithms. The results are listed in Table 4, where the semantics of the *Mapped* and the *Agree* columns is the same as explained above. The number of bases that are 1-1-mapped according to Mutagen (Mu) is 2,849,749,309.

For comparison, we used as a baseline a simple sensitivity and specificity evaluation similar to that used for SLAGAN (Brudno *et al.*, 2003a), where sensitivity is the percentage of bases in high scoring matches, while specificity is the percentage of bases matched on the *same chromosome* on both assemblies. As before, we list the error rate rather than the success rate and denote it by SER_1 (*lack of sensitivity*) and SER_2 (*lack of specificity*) and the total error ($SER_1 + SER_2$) by *Total SER*.

The table demonstrates, for instance, that method *Hg* has a lower total error, *Total ER*, than all other methods, while method *Cs* is the most sensitive (lowest ER_1) and method *Ca* the most specific (lowest ER_2). We used the Wilcoxon Rank Sum test to check the significance of these differences, and verified that almost all differences are indeed highly statistically significant. The differences in ER_1 between almost

TABLE 4. ERROR WITH RESPECT TO MUTAGEN LOG (ER), AND WITH RESPECT TO A BASELINE APPROACH (SER), ACROSS SEVERAL MAPPING METHODS

Method	*Mapped (bp)*	*Agree (bp)*	ER_1	ER_2	*Total ER*	SER_1	SER_2	*Total SER*
He	2,781,301,874	2,775,032,219	2.62%	0.225%	2.845%	10.18%	0.88%	11.06%
Ne	2,835,611,100	2,811,498,154	1.34%	0.850%	2.190%	8.43%	1.38%	9.81%
Ca	2,791,260,340	2,786,806,754	2.21%	0.159%	2.369%	9.86%	0.72%	10.58%
Hg	2,817,492,167	2,804,942,064	1.57%	0.445%	2.015%	9.01%	0.99%	10.0%
Cs	2,839,953,362	2,811,995,958	1.32%	0.984%	2.304%	8.28%	1.51%	9.79%

every two methods are highly statistically significant with $p < 0.002$, with the exception of the pairs *Ne–Cs* and *He–Ca*, which have (pairwise) about the same error rate ER_1 (same sensitivity). The differences in ER_2 between every two methods are highly statistically significant, with $p < 0.0001$. The differences in total error are not as significant, but still *Hg* has the lowest total error and the difference is statically significant with respect to *He, Ne*, and *Cs* ($p < 0.025$) while somewhat less significant ($p < 0.075$) with respect to *Ca*.

The error rates produced by the baseline approach, SER_1 and SER_2, are quantitatively different from those produced by using Mutagen. However, the ordering between the methods remains the same. The particular mapping methods used gain sensitivity from longer matches, which implies agreement between the simple sensitivity measure, (SER_1), and true sensitivity. Moreover, there are very few interchromosome translocations—the only mutation that affects the simple specificity scheme. By examining Mutagen's interchromosome translocations, we note that there are 16,934,040 bases on *B*33, whose correct mapping according to Mutagen on *BMuta*33 is to a different chromosome from the one on B33. Thus, for instance, the algorithm *Hg*, which reports 28,085,449 interchromosome base maps, *correctly maps 16,793,720 of these*, but is still penalized by the baseline specificity measure for all 28,085,449 mapped bases, because they do not respect chromosome boundaries, regardless of the fact that more than half of these bases do agree with the true (Mutagen) mapping. It is important to note that interchromosome mappings are biologically valid and not an arbitrary feature of Mutagen. For instance, Robertsonian chromosomal translocations occur in about 1 in 1,000 people, and are linked to infertility (e.g., Luciani and Guichaoua, 1985). Correctly aligning a "Robertsonian translocated" genome to a "normal" one thus requires interchromosome mapping.

We also note a shuffle in the order of the total error under the baseline setting, (*Total SER*), with respect to the Mutagen based error (*Total ER*). The choice of the denominator to be the total size of the assembly rather than the total number of true matches (which is unknown in the baseline setting) makes the sensitivity absolute value arbitrary and thus the total error value arbitrary as well. Finally, the sensitivity value (SER_1) depends on the threshold set for the similarity measure and is thus not a robust measure of performance.

5.3. Assembly-to-noise mapping

An additional way to validate the specificity of an assembly-to-assembly mapping is to ascertain that given an assembly of a true genome along with a noise sequence with a similar base distribution, the matches detected would be much shorter and more sporadic, as expected by chance, than those obtained from mapping two assemblies of truly related genomes.

We experimented with two models of genome-like noise, as discussed in Section 4.2: one sampled from a 10-order Markov model trained on a true genome assembly, and the other simply reversing the same true assembly. Figures 4 and 5 plot the distribution of the exact and unique match length for mapping the Markov sequence and the reverse sequence, respectively, to the true assembly. The *X* axis shows the match length measured in base pairs, while the *Y* axis shows the number of matches of this length. Both distributions are characterized by a multitude of short matches. There are no matches longer than 34 bases found when aligning *WGSA* to Markov, and no matches longer than 69 bases when aligning it to the reverse sequence. In both cases, the rest of the pipeline could not further extend the matches into ungapped regions (U's) or into longer Runs. These results are contrasted with mapping NCBI's *B33* to Celera's *WGSA*, where the mean match length is 245 bp (standard deviation 469), the maximum is 16,258 bp, and the median is 64 bp.

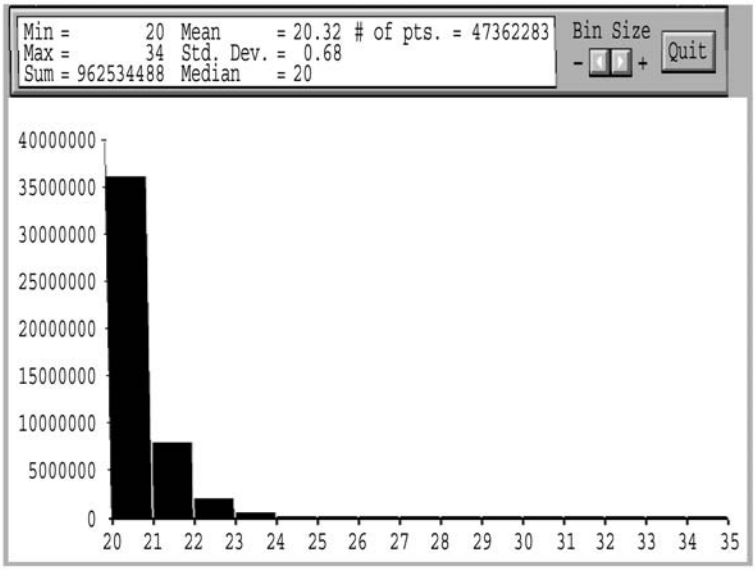

FIG. 4. Distribution of match length, *WGSA* versus Markov. *X*: Match length (in bases); *Y*: number of matches with this length.

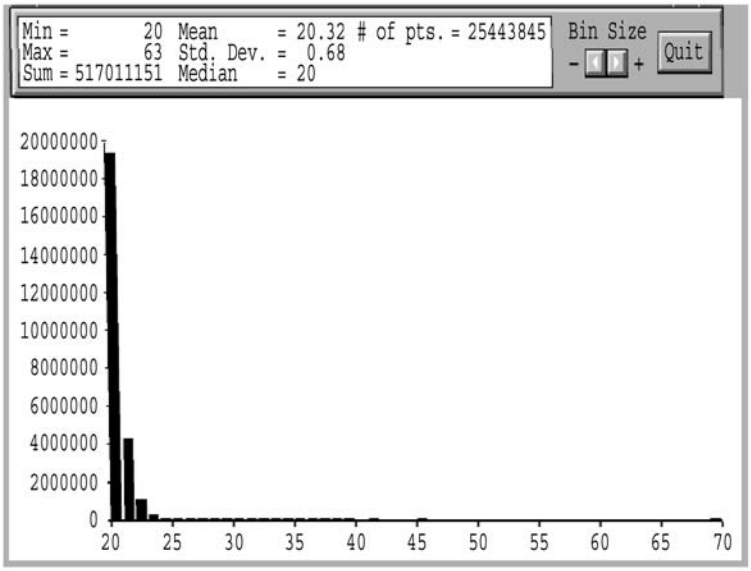

FIG. 5. Distribution of match length, *WGSA* versus Reverse. *X*: Match length (in bases); *Y*: number of matches with this length.

6. CONCLUSIONS

We introduced *ThurGood*, a suite of methods for evaluating assembly-to-assembly mapping, the most prominent of which is based on Mutagen. The latter provides the means to produce a ground-truth map, which allows the calculation of sensitivity and specificity of the alignment process with respect to it. The results in Section 5 demonstrate the objective evaluation process by which mapping methods can be compared against each other using meaningful metrics. The Mutagen-based data used in this evaluation is publicly available at *www.cs.queensu/BioInfo/MutaData*, as a benchmark to support the evaluation of various whole-genome alignment tools.

ACKNOWLEDGMENTS

We thank Liliana Florea and Brian Walenz for the support and tools for RefSeq mapping, Michael Waterman for his comments on statistical significance, and Sorin Istrail for his advice throughout.

REFERENCES

Altschul, S.F. *et al.* 1990. Basic Local Alignment Search Tool. *J. Mol. Biol*. 215, 403–410.

Billingsley, P. 1959. Statistical methods in Markov chains. *Ann. Math. Statist*. 32, 12–40.

Bray, N. *et al.* 2003. AVID: A global alignment program. *Genome Res*. 13(1), 97–102.

Brudno, M. *et al.* 2003a. Glocal alignment: Finding rearrangements during alignment. *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, 54–62.

Brudno, M. *et al.* 2003b. LAGAN and Multi-LAGAN: Efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res*. 13(4), 721–731.

Delcher, A.L. *et al.* 1999. Alignment of whole genomes. *Nucl. Acids Res*. 27(11), 2369–2376.

Florea, L. *et al.* 1998. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res*. 8(9), 967–974.

Istrail, S. *et al.* 2004. Whole-genome shotgun assembly and comparison of human genome assemblies. *Proc. Natl. Acad. Sci*. 101(7), 1916–1921.

Luciani, J.M., and Guichaoua, M.R. 1985. Chromosomal abnormality in male infertility. *Ann. Biol. Clin.* 43(1).

Mobarry, C., and Sutton, G. 2003. An assembly to assembly comparison tool. *RECOMB Satellite Conference on DNA Sequencing Technologies and Computation.*

Myers, E. 1999a. A dataset generator for whole genome shotgun sequencing. *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, 202–210.

Myers, E. 1999b. Whole-genome DNA sequencing. *IEEE Comp. Eng. Sci*. 3(1), 33–43.

Needleman, S.B., and Wunsch, C.D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol*. 48, 443–453.

Pollard, D.A. *et al.* 2004. Benchmarking tools for the alignment of functional noncoding DNA. *BMC Bioinformatics* 5(1), 6.

Pruitt, K.D., and Maglott, D.R. 2001. RefSeq and LocusLink: NCBI gene-centered resources. *Nucl. Acids Res*. 29(1), 137–140. *www.ncbi.nlm.nih.gov/LocusLink*.

Schwartz, S. *et al.* 2003. Human–mouse alignment with BLASTZ. *Genome Res*. 13, 103–107.

Searle, A.G. *et al.* 1989. Chromosome maps of man and mouse, IV. *Ann. Human Genet*. 53, 89–140.

Smith, T.F., and Waterman, M.S. 1981. Identification of common molecular subsequences. *J. Mol. Biol*. 147, 195–197.

Stoye, J. *et al.* 1998. Rose: Generating sequence families. *Bioinformatics* 14(2), 157–163.

TIGR Genome Database. *www.tigr.org/tdb*.

Turner, R. *et al.* 2001. Visualization challenges for a new cyberpharmaceutical computing paradigm. *Proc. IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, 7–18.

UCSC. hg6. *www.genome-hg8.cse.ucsc.edu/cgi-bin/hgGateway?db=hg6*.

Venter, J.C. *et al.* 2001. The sequence of the human genome. *Science* 291, 1304–1351.

Waterston, R.H. *et al.* 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature* 420, 520–562.

Address correspondence to:
Hagit Shatkay
School of Computing
Queen's University
Kingston, Ontario
Canada K7L 3N6

E-mail: shatkay@cs.queensu.ca