

Solving the Problem of Trans-Genomic Query with Alignment Tables

D. Stott Parker, Ruey-Lung Hsiao, Yi Xing, Alissa Resch, Chris Lee

Abstract—The *trans-genomic query* (TGQ) problem — enabling the free query of biological information, even across genomes — is a central challenge facing bioinformatics. Solutions to this problem can alter the nature of the field, moving it beyond the jungle of data integration and expanding the number and scope of questions that can be answered.

An *alignment table* is a binary relationship on *locations* (sequence segments). An important special case of alignment tables are *hit tables* — tables of pairs of highly similar segments produced by alignment tools like BLAST. However, alignment tables also include general binary relationships, and can represent any useful connection between sequence locations. They can be curated, and provide a high-quality queryable backbone of connections between biological information. Alignment tables thus can be a natural foundation for TGQ, as they permit a central part of the TGQ problem to be reduced to purely technical problems involving tables of locations.

Key challenges in implementing alignment tables include efficient representation and indexing of sequence locations. We define a location datatype that can be incorporated naturally into common off-the-shelf database systems. We also describe an implementation of alignment tables in BLASTGRES, an extension of the open-source PostgreSQL database system that provides indexing and operators on locations required for querying alignment tables.

This paper also reviews several successful large-scale applications of alignment tables for Trans-Genomic Query. Tables with millions of alignments have been used in queries about *alternative splicing*, an area of genomic analysis concerning the way in which a single gene can yield multiple transcripts. Comparative genomics is a large potential application area for TGQ and alignment tables.

I. INTRODUCTION

Publication of the human genome has given scientists a digital blueprint for the human being. Complete genomes for other organisms — including mouse, rat, and chimp — have also been published recently. This data presents an unprecedented opportunity for medical research, but also a tremendous challenge demanding new information technology development. Specifically, it requires a database system that can connect biological information across genomes, and

D.S. Parker and R-L. Hsiao are with the Computer Science Dept. at UCLA. Y. Xing is with the Department of Internal Medicine, Roy, J and Lucille A. Carver College of Medicine, University of Iowa, Iowa City, IA 52242.

A. Resch is with the National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20894.

C. Lee is with the Department of Chemistry and Biochemistry at UCLA. This research was supported by NSF Grant IIS 0082964, NIH Grant 1P20MH065166, NIH Grant 1U54RR021813, and the UCLA *Center for Computational Biology (CCB)*, an NIH Center of Excellence.

answer queries about the diverse functional and structural relationships between sets of genes.

We call the problem of enabling the free query of these relationships the problem of *Trans-Genomic Query* (TGQ). The problem, and its lack of solutions, are familiar in bioinformatics research. This paper summarizes a technique for addressing TGQ that we have used successfully.

A. Trans-Genomic Query

The number and diversity of biological sequence information is growing rapidly, reflecting the increase in number of new genomes being sequenced. Sequence annotation comes in several flavors, depending on which sequence is being annotated. Annotation at the DNA sequence level reveals information about disease polymorphisms and homology relationships between different species, whereas annotation at the amino acid sequence level reveals functional insight about enzyme active sites, secondary structural elements and domain organization within certain protein families. Viewing this information as a kind of network or graph, as in Figure 1, highlights its diversity. Lack of a unifying information model has been a key challenge to solving the problem of TGQ.

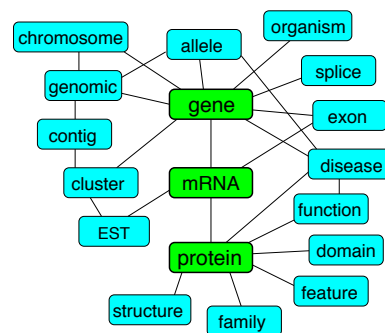


Fig. 1. The Trans-Genomic Query Problem requires the ability to connect and analyze biological information across genomes in an automated way. This diagram illustrates the diverse kinds of information that can be queried. This diversity has helped create the TGQ problem, by making it difficult to implement queries with existing database systems and information models. Sequence alignments will play a central role in solving the problem, since they define a hub around which other information is connected.

Sequence alignment will play an important, unifying role for TGQ, as a way to cope with the diversity of the information. The core of Figure 1 defines structural relationships between genomic, mRNA, and protein sequences. Since the human genome is the product of an evolutionary process of creating new genes through copying, incrementally mutating and recombining existing genes, there is a dense matrix of

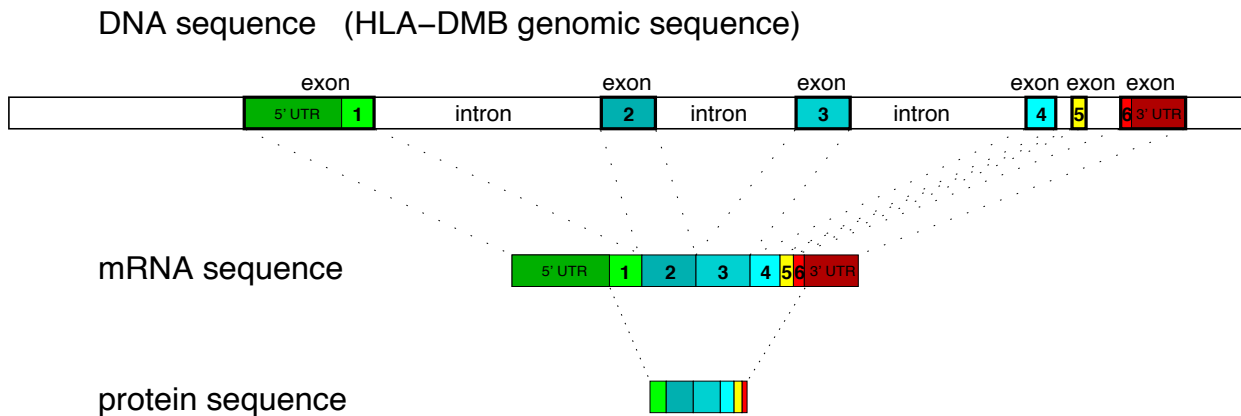


Fig. 2. Transcription of the human gene HLA-DMB, in human Chromosome 6. The HLA (Human Leukocyte Antigen) region of Chromosome 6 contains genes regulating the immune system, and is of great interest since it is involved in human tissue typing for organ transplants. When DNA is transcribed to mRNA sequences, introns are *spliced out* (removed), while exons are not. For this gene, there are six exons. The resulting mature mRNA sequence is translated to protein, omitting the UTRs (untranslated regions) at each end.

sequence similarity relationships across all genes in the human genome. These relationships connect them in rich detail to genes from other organisms (such as the mouse) for which biologists have powerful experimental tools and a wealth of data. Historically, as a result, the fastest route to a major discovery about human disease has often been finding a sequence similarity relationship between a human gene and a gene from a ‘model organism’ (such as mouse, chimp, *Drosophila*, *C. elegans*, or even *yeast*) where function is understood in detail. The ability to connect information across genomes is fundamental to answering biological questions. Sequence alignment is a foundation for expressing relationships among biological information, and a foundation for TGQ.

B. Reconciling Trans-Genomic Query and Sequence Alignment with modern Database Systems

Traditional relational database systems often seem ill-suited for managing the complexity, rapidity of evolution, and diversity of biological information. These systems were also designed partly to enforce a management discipline, and not to support the flexibility required by exploratory research. Biological information possesses an abundance of features, properties, and aggregations. Forcing this information into a limited set of datatypes (integer, char, ...) makes it harder to express questions that biologists would want to ask, ensures poor query performance and poor scalability, and limits the database’s ability to keep pace with the influx of new types of information. Although object-oriented approaches support new datatypes well, they can impose performance and style overhead. In fact, objects can make the information management problem worse, if simple descriptive values in tables are organized into complex class definitions.

Because of these problems, gene sequence ‘databases’ are often unable to answer even very basic queries, and as a result can become warehouses of ‘dead data’. To answer a query, all or part of the data must be extracted from the database, and a custom analysis program must be developed. Popular existing databases (e.g., Genbank) and query tools (e.g., BLAST [4] and Hidden-Markov Model applications) follow this pattern. This

process is inefficient, error-prone, and poses serious integration and collaboration problems for users of genome data. While there are many excellent individual analysis applications, each of which performs one distinct analysis function, the absence of a database capable of storing and querying the results makes it difficult to ask queries that relate two existing analyses. This impedes progress by preventing the research community from pooling and mining diverse analyses of genome data.

C. Objectives of this Work

This paper describes a technique that has been used to answer real TGQs, large-scale biological queries on databases involving many millions of records, and has been particularly useful in the analysis of SNPs (Single Nucleotide Polymorphisms) and Alternative Splicing [11]–[21]. Abstracts for these and some databases including alignment tables can be downloaded from www.bioinformatics.ucla.edu/ASAP/ (the ASAP project site).

After laying out the idea of alignment tables, we illustrate their use in TGQ involving Alternative Splicing. This paper first attempts to describe the idea formally, and explore its potential for addressing the problem of TGQ. Alignment tables give a way to represent essential information about sequence alignments in off-the-shelf database systems. They are simple, but the simplicity can pay off in scalability and broad applicability. We are unaware of previous work making systematic use of them.

The paper also describes an implementation using BLASTGRES, an extension of the PostgreSQL open-source object-relational database system that includes support for a location datatype and for alignment tables. This implementation avoids some of the frustration mentioned above, and suggests strategies for developing future bioinformatics database systems [23]. BLASTGRES is publicly available from www.BLASTgres.org, or from the *Center for Computational Biology (CCB)* at www.loni.ucla.edu/CCB/Software/. Output from the queries shown in this paper were produced by BLASTGRES, and they and the sample database are downloadable from the BLASTGRES site.

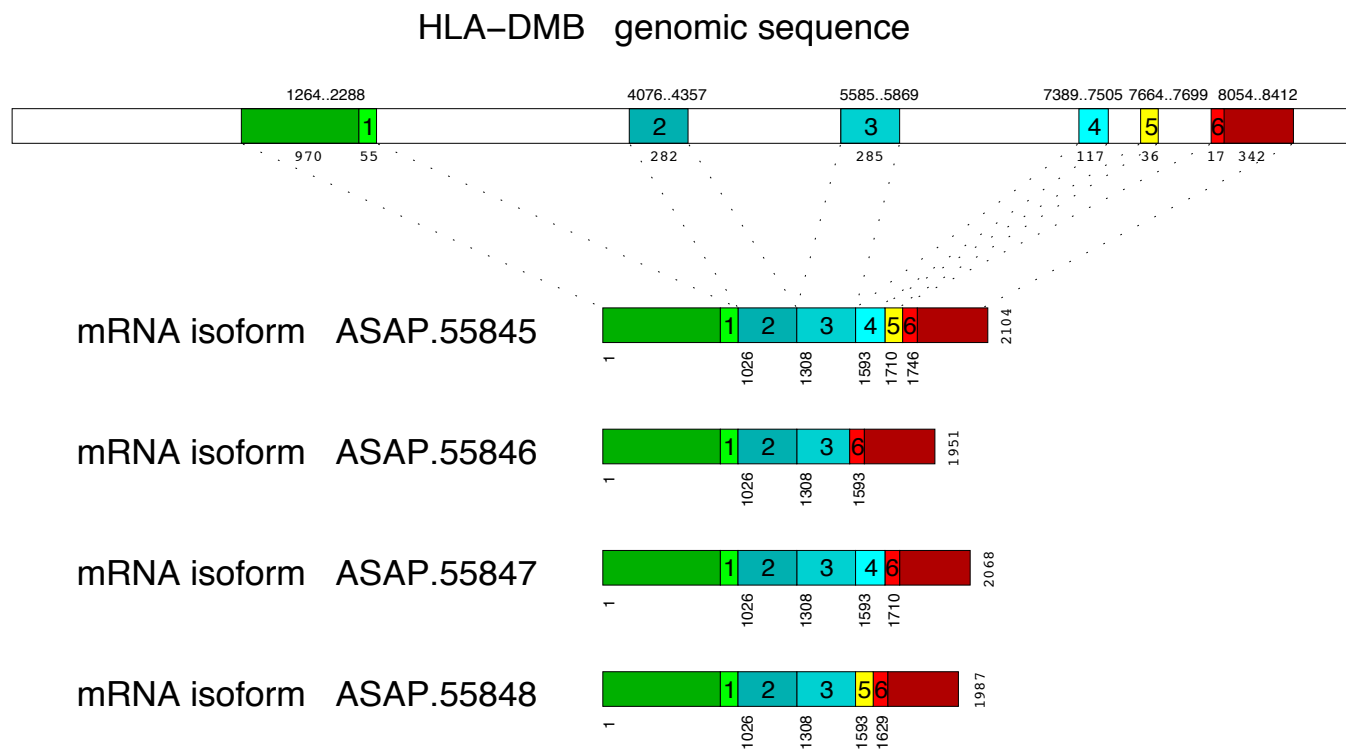


Fig. 3. Alignment of the human gene HLA-DMB with four of its mRNA isoforms, derived from sequences in the Unigene cluster Hs.1162, and each isoform consists of a subsequence of the exons in the genomic sequence. An alignment table corresponding to these isoforms is shown in Figure 4. Some explanation of the significance of the exonic structure of these isoforms is provided in [12]: *Analysis of these forms reveals a remarkably simple and intriguing functional effect. ... HLA-DM is normally targeted to lysosomes, and its beta chain contains a transmembrane domain anchoring its C-terminus ... Exon IV is short, and corresponds precisely to the transmembrane domain. Exon V is very short, and encodes the lysosomal targeting signal YTPL, whose first residue begins at the start of the exon. ... It appears that the gene structure of the HLA-DMB gene has been carefully 'designed' to enable control of HLA-DM function, by pulling out both the transmembrane helix and the lysosomal targeting signal into separate short exons (IV, V) that can be alternatively spliced in-frame (exon VI supplies the last 4 amino acids of the protein, identical in all forms).*

gene	genomic_start	genomic_end	isoform	mrna_start	mrna_end
HLA-DMB	1264	2288	ASAP.55845	1	1025
HLA-DMB	4076	4357	ASAP.55845	1026	1307
HLA-DMB	5585	5869	ASAP.55845	1308	1592
HLA-DMB	7389	7505	ASAP.55845	1593	1709
HLA-DMB	7664	7699	ASAP.55845	1710	1745
HLA-DMB	8054	8412	ASAP.55845	1746	2104
HLA-DMB	1264	2288	ASAP.55846	1	1025
HLA-DMB	4076	4357	ASAP.55846	1026	1307
HLA-DMB	5585	5869	ASAP.55846	1308	1592
HLA-DMB	8054	8412	ASAP.55846	1593	1951
HLA-DMB	1264	2288	ASAP.55847	1	1025
HLA-DMB	4076	4357	ASAP.55847	1026	1307
HLA-DMB	5585	5869	ASAP.55847	1308	1592
HLA-DMB	7389	7505	ASAP.55847	1593	1709
HLA-DMB	8054	8412	ASAP.55847	1710	2068
HLA-DMB	1264	2288	ASAP.55848	1	1025
HLA-DMB	4076	4357	ASAP.55848	1026	1307
HLA-DMB	5585	5869	ASAP.55848	1308	1592
HLA-DMB	7664	7699	ASAP.55848	1593	1628
HLA-DMB	8054	8412	ASAP.55848	1629	1987

Fig. 4. Part of an alignment table for the alignments of human gene HLA-DMB genomic sequence and mRNA isoform sequences in Figure 3. Generally, alignments are binary relationships on *locations* in sequences. In this table, locations are represented by a sequence identifier with an accompanying interval, giving a starting and ending position. For example, the genomic sequence referred to in this table is the complement in contig GI.17464666 of location 7815395..7821803; the isoforms referred to are sequences in the Human January 2002 version of the ASAP database at UCLA. Richer models for locations can be used, and further attributes such as the alignment *score* can be included, but this table shows the essence of the alignment table concept.

II. ALIGNMENT TABLES

Solving the problem of Trans-Genomic Query requires a general capability for joining information from different sources. If this information can be indexed by sequences, and retrieved when these sequences match a query sequence, then joining can be implemented via sequence alignment.

A. Sets of Alignments

TGQ differs from conventional sequence alignment in that it is necessary to manage *sets of alignments* — possibly very large sets. For example, the growing field of comparative genomics requires management of vast sets of alignments.

Traditional sequence alignment methods often focus on two problems: *pairwise sequence alignment*, in which alignment identifies pairs of sequences of high similarity, or segment pairs of high similarity between two sequences (local alignment), where similarity is generally measured by a score function; and *multiple sequence alignment*, in which alignment seeks a mapping of each member of a given set of sequences to segments of a consensus coordinate system. The consensus may be either given or derived, but the goal is to find a many-to-one alignment of the input sequences to locations in a single coordinate system.

This alignment is often chosen to minimize a global objective function, such as a sum of pairwise scores. Scores used include E-values, *p*-values, bit scores, and percent identity (percent of exact matches between a query and database segment); BLAST's high-scoring segment pairs are called *HSPs* or *hits* [4].

Trans-Genomic analysis presents a different environment for sequence alignment. There may be no consensus sequence, for one thing, and the issues raised by the scale of the analysis can differ significantly from those traditionally addressed by traditional alignment methods. Many such issues — including consistent ordering and orientation of aligned segments, as well as proper handling of repeats, paralogs, and regions of extremely biased nucleotide content — are addressed in [5], which discusses the selection of BLASTZ for developing whole-genome alignments of the human and mouse genomes. The family of BLAST tools implements multiple notions of alignment [1].

B. The Alignment Table Concept

An **alignment table** is a binary relationship on sequence locations. An example is shown in Figure 5, presenting a table that summarizes a set of three pairwise alignments of locations in sequences *S1* and *S2*. Both the diagram and the relational table give a useful summary of this set of alignments.

A sequence **location** in this table is a (*identifier, range*)-pair, giving a sequence identifier and an range specifying the location's [*start, end*] range. In this paper an *identifier* is a string specifying both a database and a particular sequence within that database, such as in the format *dbname.seqnumber*. Interesting properties of locations are explored later. The table can include other information, such as alignment score values or other information pertaining to the locations.

A more complex example is shown in Figure 4. Notice that the alignment table can be useful even without the actual sequences used to derive the table being present. Examples below show how the alignment table can be useful in its own right, and as backbones for joining biological information.

C. Hit Tables vs. Alignment Tables

Some popular tools for sequence alignment are able to produce a **hit table** as output. A hit table is an alignment table that lists all pairs of input segments aligned with a high score. Although hit tables and alignment tables have similar tabular structure, they differ in nature.

1) *Content*: A hit table is a catalog of segment pairs (HSPs) that have high score under some sequence similarity measure [4]. By contrast, an alignment table can be any binary relationship on segments. In principle it need not have anything directly to do with sequence similarity. For example, we can align sequences that are believed to have a common ancestor, even if their sequence similarity score is weak.

In this paper, alignment tables generally represent information about complete segments of larger biological significance (such as full genomic sequences, exons, or mRNA transcripts). Furthermore the notion of alignment can be more general than just sequence similarity.

2) *Scale*: Hit tables are not typically of large scale. Alignment tables are intended to be of potentially very large scale, however, with information about every sequence of interest. They thus can leverage the capabilities of database systems for large-scale data management.

3) *Integrity*: The alignments in a hit table may lack some forms of integrity. For example, as a set they may not be consistent in the sense that their hits cannot all be part of a single combined alignment. Figure 6 shows two pairwise alignments that cannot both be part of a single diagonal. BLASTZ [5] provides an option for requiring hits to occur in the same order and orientation. Another integrity criterion is that certain kinds of alignments be disjoint, or that overlapping alignments be coalesced into a single contiguous alignment. Integrity spans many issues, including sophistication of the alignment and strength of evidence. Alignment tables give a framework for managing some aspects of integrity, and incorporating ongoing maintenance by automated processes.

4) *Queryability*: Alignment tables are intended to be queried and used in answering larger queries; hit tables are not. Although they are a mainstay of bioinformatics research today, tools such as BLAST make little provision for interactive query of their output. Even when hit tables are stored in relational databases, there is no clear set of query operators for them that will make them useful as a backbone connecting biological information for TGQ. It is interesting to view the alignment table as a kind of *join index* [24] on locations. We present a set of query operators for alignment tables later.

5) *Foundation in Data management*: A theme in all of these differences, and perhaps the most fundamental difference, is that alignment tables are rooted in data management. Data management permits automation and transparency; for example, alignment tables might be automatically maintained

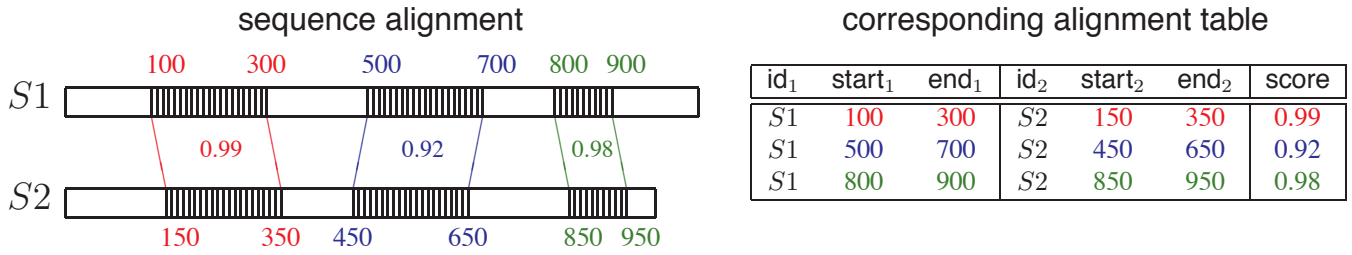


Fig. 5. An alignment of the sequences $S1$ and $S2$ with similarity scores can be digested into an alignment table, which summarizes the three segments of similarity. The table can include many pairwise sequence alignments. Although the alignment table loses information about the sequences, it captures the essence of their alignment similarity relationship. It also captures enough structure to permit connection of available information about the alignment.

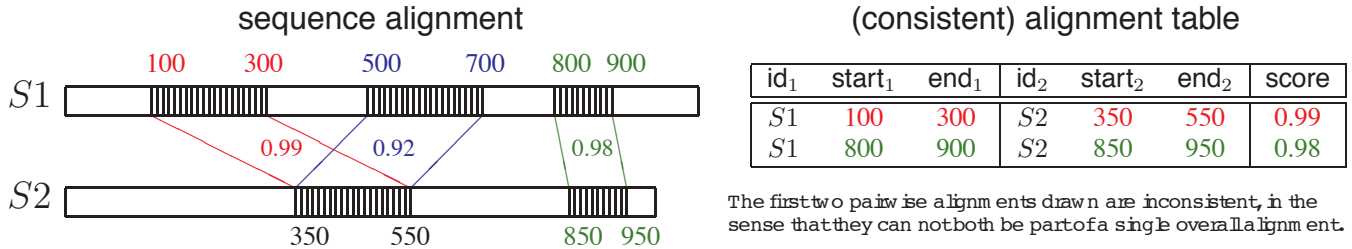


Fig. 6. Alignment tables can be held to quality standards defined by integrity constraints. The diagram shows an alignment of two sequences $S1$ and $S2$ for which there are three hits (segments of similarity). The first two hits are *inconsistent*, in the sense that they cannot both be part of a single overall alignment of the two sequences. An alignment table is shown for these sequences, giving a consistent subset of hits that maximizes overall score. Another alignment table could be created with the second and third hit (but with a lower total score). Consistency is a natural integrity constraint for alignment tables.

so as to track changes in underlying reference sequences, or to formalize important database integrity constraints such as referential integrity. Strengths of database systems can translate into strengths of alignment tables.

D. Significance for TGQ and Bioinformatics

The differences just listed between hit tables and alignment tables are significant, and all are important in TGQ. Generally speaking, trans-genomic analysis requires a flexible notion of alignment. For example, comparing genomes can require aligning pairs of sequences that are homologous (have a common evolutionary origin, and thus have similar sequences and similar structure, but possibly somewhat different function) even if they are not HSPs. Alignment tables permit creation and use of specialized alignments. We believe that the alignment table concept is useful, and that it merits the attention of the working bioinformatics researcher.

The remainder of this paper describes technical issues in the implementation of alignment tables, and then illustrates with examples how they can be useful for TGQ.

III. IMPLEMENTATION OF ALIGNMENT TABLES

An alignment table is a similarity relationship on locations, as we have illustrated in Figure 4. Because they define a basic form of biological information, with many nontrivial properties, the implementation has centered on the development of a *location datatype* and a *range datatype*. These datatypes are useful not only in expressing queries, but also in indexing.

This section describes an implementation of alignment tables developed in BLASTGRES [39], [40], and the problems of implementation. The primary point: alignment tables require sophisticated data management and query of locations, and BLASTGRES shows a way to provide this.

Similar implementations are possible in almost any database system. At the end of this section we discuss approaches successful in the past, and some directions for research.

A. Location and Range Datatypes

Bioinformatics make heavy use of the concept of a *location*. The location concept can be treated as similar to the idea of Cartesian coordinates, and location is a basis for ‘coordinatizing’ bioinformatics. Locations are essential for *annotation* (associations of features to parts of sequences), and more generally *maps* (associations of features to intervals in sequences). The ability to develop detailed maps has been essential to bioinformatics.

The notion of a Location is present in most data models for biology. The influential NCBI data model [9] defines a sequence (Bioseq) as having an integer coordinate system, with features being attached to sequences for at least one location (Seq-loc). Similarly, the core class Bio::Location in BioPerl [3] specifies integral ranges in sequences. Abstractions such as Bio::DB::GFF::Segment and sophisticated classes such as Bio::Coordinate are built directly upon it.

BLASTGRES provides both a *range* datatype and a *location* datatype, representing a location as an (*identifier*, *range*)-pair:

$$\begin{aligned} \text{range} &= \text{start}..\text{end} \\ \text{location} &= \text{identifier}[\text{start}..\text{end}]. \end{aligned}$$

Here the *identifier* is a sequence identifier, and the *range* is a pair of integers giving its coordinates (the integer starting and ending positions of the location). In BLASTGRES, ranges and locations appear (are read and printed) as composite strings:

```
'1308..1592'  
'ASAP_55845[1308..1592]'
```

BLASTGRES datatypes are implemented using POSTGRES [48], an object-relational database that supports user-defined datatypes. Whenever locations are read, these strings are parsed by a BLASTGRES-defined function into the *identifier*, *start*, and *end* values, and stored in an efficient data structure. Similarly, whenever locations are printed, a reverse transformation is used. The introduction of location datatype enables intuitive representation of the underlying data and the addition of necessary query operators (such as coordinate transformation and slicing operations). Development of a location datatype by itself is straightforward. The implementation becomes interesting in adding query operators and functions for this datatype, and in providing indexing on locations.

B. Query Operators on Locations

Both ranges and locations can be viewed abstractly as intervals, and a query language can be fashioned using known operators on intervals. Although intervals appear to be a simple form of information, they have surprisingly complex semantics. This is illustrated by Allen’s interval logic [34], a popular set of abstractions for intervals, adapted into the 13 relationships (predicates) on intervals shown in Figure 7.

Interval relationship	Predicate definition
$[a, b]$ after $[c, d]$	$(d < a)$
$[a, b]$ before $[c, d]$	$(b < c)$
$[a, b]$ contains $[c, d]$	$((c \geq a) \wedge (d \leq b))$
$[a, b]$ during $[c, d]$	$((a \geq c) \wedge (b \leq d))$
$[a, b]$ equals $[c, d]$	$((a = c) \wedge (b = d))$
$[a, b]$ finishes $[c, d]$	$((b = d) \wedge (a \geq c))$
$[a, b]$ finished_by $[c, d]$	$((d = b) \wedge (c \geq a))$
$[a, b]$ meets $[c, d]$	$(b = c)$
$[a, b]$ met_by $[c, d]$	$(d = a)$
$[a, b]$ overlaps $[c, d]$	$((a \leq c \leq b) \vee (a \leq d \leq b))$
$[a, b]$ overlapped_by $[c, d]$	$((c \leq a \leq d) \vee (c \leq b \leq d))$
$[a, b]$ starts $[c, d]$	$((a = c) \wedge (b \leq d))$
$[a, b]$ started_by $[c, d]$	$((c = a) \wedge (d \leq b))$

Fig. 7. BLASTGRES’ relationships on ranges. These differ from Allen’s 13 original relationships on intervals, aiming primarily at transparency and intuitiveness, where Allen’s predicates were aimed at completeness. Some aspects of these relationships are subtle, and direct use of inequalities instead of predicates like these can be error-prone.

Some of the definitions are both intuitive and easy to remember. Others, however, are not — and their use can be error-prone. To see this, consider which among the following possible definitions of an *overlap* relationship between intervals $[a, b]$ and $[c, d]$ are actually equivalent:

$$\begin{aligned}
 & ((a \leq c) \wedge (b \geq c) \wedge (b \leq d)) \\
 & ((c \leq b) \wedge (d \geq b) \wedge (a \leq c)) \\
 & \max(a, c) \leq \min(b, d) \\
 & [a, b] \cap [c, d] \neq \emptyset.
 \end{aligned}$$

The first definition is Allen’s, and it is equivalent to the second definition. The third definition is not equivalent unless we assume $a \leq b$ and $c \leq d$; under this assumption the first three can be re-expressed as $(a \leq c \leq b \leq d)$. Consequently, the fourth definition is inequivalent to the others. Naming

the fourth definition ‘ $[a, b]$ intersects $[c, d]$ ’ communicates its meaning more clearly.

The initial motivation for the work behind this paper stems from bad experiences accumulated while manipulating intervals with SQL. If queries are forced to work on *start* and *end* values directly (rather than on intervals) the SQL WHERE clauses become complex, and — as the *overlap* example shows — easy to get wrong.

The predicates defined by Allen are surprisingly useful for bioinformatics, but stem from a temporal view of intervals. For example, Allen’s predicates cannot express

$$[a, b] \text{ just_precedes } [c, d] \equiv (b = c - 1)$$

which is important in sequence alignment. Although an implementation based on intervals goes a long way toward providing the storage structures needed for this diversity of models, the need to provide natural query languages for them requires more than the traditional relational database. BLASTGRES addresses this need with the location datatype.

C. Location and Range Indexing

We have developed alignment tables with millions of pairs of sequences, where each sequence can possess as many as 100 locations. Depending on how the tables are used, it can be necessary to index on either one or both of the sequence identifiers, and for each indexed sequence identifier, also index the locations. Indexing can be vital for performance.

Relational database systems do not directly support indexing of locations, even when locations are implemented simply as intervals. Furthermore, relational query optimizers do not gracefully handle the kinds of WHERE clauses needed, which include clauses like

$$\begin{aligned}
 & \text{o.seq_start} > \text{d.query_start} \\
 & \text{and o.seq_start} < \text{d.query_end}
 \end{aligned}$$

Although these could be processed with a sequential scan, they are often compiled by query optimizers as ‘*less-than joins*’, resulting in very slow execution.

Interval indexing is often credited to Edelsbrunner [43]. Many implementation approaches have been developed since; surveys can be found in [44], and more recently [41], [42]. It is important to realize however that most of this past work has been restricted to *in-memory* data structures for interval computations. Interval indexes for database systems have been developed, such as the RI-Tree [41], [42], but database interval indexing is still young.

Successors to relational database systems — including object-relational, extensible, and component database systems — all have emphasized support for general datatypes, and for indexing of these types. Extensible indexing has been addressed explicitly in component and extensible database systems [45], and for example the Oracle Data Cartridge Interface permits user-defined types, including routines for index-definition, index-maintenance, and index-scan operations [50].

For indexing, the BLASTGRES implementation uses the native GiST (Generalized Search Tree) indexing, an elegant

genomic_location	isoform_location
HLA-DMB[1264..2288]	ASAP.55845[1..1025]
HLA-DMB[4076..4357]	ASAP.55845[1026..1307]
HLA-DMB[5585..5869]	ASAP.55845[1308..1592]
HLA-DMB[7389..7505]	ASAP.55845[1593..1709]
HLA-DMB[7664..7699]	ASAP.55845[1710..1745]
HLA-DMB[8054..8412]	ASAP.55845[1746..2104]
HLA-DMB[1264..2288]	ASAP.55846[1..1025]
HLA-DMB[4076..4357]	ASAP.55846[1026..1307]
HLA-DMB[5585..5869]	ASAP.55846[1308..1592]
HLA-DMB[8054..8412]	ASAP.55846[1593..1951]
HLA-DMB[1264..2288]	ASAP.55847[1..1025]
HLA-DMB[4076..4357]	ASAP.55847[1026..1307]
HLA-DMB[5585..5869]	ASAP.55847[1308..1592]
HLA-DMB[7389..7505]	ASAP.55847[1593..1709]
HLA-DMB[8054..8412]	ASAP.55847[1710..2068]
HLA-DMB[1264..2288]	ASAP.55848[1..1025]
HLA-DMB[4076..4357]	ASAP.55848[1026..1307]
HLA-DMB[5585..5869]	ASAP.55848[1308..1592]
HLA-DMB[7664..7699]	ASAP.55848[1593..1628]
HLA-DMB[8054..8412]	ASAP.55848[1629..1987]

Fig. 8. A BLASTGRES variant of the alignment table shown in Figure 4, containing pairs of location datatype values. Here the ASAP identifiers refer to sequences in the ASAP January 2002 Human alternative splicing database. HLA-DMB is a user-specified identifier, and actually refers to a location within a GenBank contig for HLA-DM – the reverse complement of 'GI.17464666[7813399..7823803]'.

scheme for extensibility of indexing in POSTGRES. Two kinds of indexing are needed for fast retrieval:

- 1) indexing on sequence identifier
- 2) indexing on range.

With a location datatype using the $(identifier, interval)$ representation mentioned above, GiST can provide both kinds of indexing together. The representation treats the sequence identifier as more significant, and for a given identifier provides interval indexing. With GiST, each index entry defining an index subtree has the form (p, ptr) , where p is a predicate that matches keys in the subtree, and ptr is a pointer to the subtree. For ranges contained within the interval $[a, b]$, the GiST entries have a predicate p of the form $[a, b]$. A query q has associated with it both an interval $[c, d]$, and a *strategy* value specifying which predicate relationship is needed. For ranges with endpoints x and y , the strategy s can range over predicates in the enumerated set $\{Left, Right, OverLeft, Overlap, OverRight, Right, Equal, Contains, Contained\}$ defined for R-trees [47]. In fact, more predicates could be defined over the relationships between intervals. For instance, Allen goes beyond this list (Allen's interval logic includes 13 predicates). One of the benefits of using GiST is that new user-defined predicates can be more easily added when needed. It is easy to support these and other predicates with an implementation like that for R-trees [46].

Alignment tables require sophisticated data management and query of locations, and BLASTGRES offers a general, extensible, and interesting way to provide this.

D. Different Representations of Locations, and Performance

There are a number of different ways to represent a location: (1) one field: a *location* value; (2) two fields: a string identifier and a *range* value; (3) three fields: a string identifier, start value, and end value; (4) any values that can be used to infer a location of form (1), (2), or (3). Each can lead to a

different implementation of alignment tables. An alignment table using the third implementation was shown earlier in Figure 4. An alternative implementation using the location datatype is shown in Figure 8. It can be important to allow different implementations of alignment tables.

The choice of representation for locations is akin to a choice of data structure; it influences both query structure and query performance. Important issues that affect performance include:

- the representation of identifiers
- the representation of ranges
- the kinds of queries anticipated; for example, queries can involve any of the Allen predicates mentioned earlier, and they can require a scan of a set of multiple range values.

Each of these issues influences the way in which information is stored on disk, and the number of disk reads is very important in determining performance. At some risk of oversimplifying, it is worth stressing this point.

Disk reads in database systems retrieve a *page* (a fixed-length block, typically with length 8K bytes). As a back-of-envelope rule, each read typically takes something like 5 msec to complete. Assuming this, a thousand disk reads will take about 5 sec, a million disk reads will take about 5000 sec, etc. As a result, for fast response to queries it is important to organize information so as to reduce the number of disk reads. A query that yields 1000 records, each from different disk pages, will require 5 sec to complete. If instead the database were organized so that each page contained 100 of these records, the query would complete 100 times faster. Storing information in an order that anticipates future queries can improve response time by orders of magnitude.

Indexing can be used to reduce the number of reads in finding specific records quickly. However if a query yields 1000 records, and indexing provides instantaneous access to them, whenever these records are located on 1000 different pages the query will still require 5 sec. Indexing alone cannot make a database fast. By contrast, if records are stored in anticipated-scan-order (such as increasing-range-start-order), with multiple scan-related records packed into a single page, scans will be faster. Clustered indexes (indexes of these packed pages, and not indexes of individual records) can then provide high performance for important classes of queries that require random access to sequences of related records. There are *many* other aspects of performance to consider [49], but this discussion gives a zeroth-order sketch of what indexing can provide.

An alignment table itself can be viewed as an index. Earlier we mentioned that it can be viewed as a join index [24] on locations. A single alignment relationship can actually be stored in multiple forms, each one anticipating a different kind of query. Therefore there should not be a single representation for alignment tables. Just as there can be different kinds of index, and different kinds of query, there can be different kinds of alignment table.

E. Implementation in almost any Database System

Essentially any database system can be used to implement alignment tables. Alignment tables and indexes for ranges and

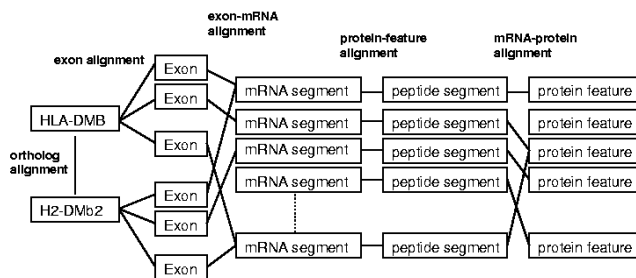


Fig. 9. Using alignment tables to provide TQG connectivity among genomic and proteomic information. Facts about exons in genomic sequences are ultimately connected to information about protein domains. This conceptual ‘information backbone’, implemented with the indicated alignment tables, runs through the examples in this paper.

locations have been implemented in many ways in the past, some in ingenious ways. For example, in systems like MySQL that support spatial datatypes and provide R-tree indexing [46], range retrievals have been implemented with minimum bounding rectangles. The UCSC Genome Browser [38] uses MySQL for its database and a spatial index on the tables. Numerous TQG analyses by authors of this paper were also developed with MySQL.

Furthermore, any sufficiently extensible database management system, such as Oracle and DB2, can be augmented to provide all the functionality provided by BLASTGRES just described. In fact, these systems could be augmented to provide even more powerful implementations of alignment tables and location indexing, and better support for sequence alignment. The GALA (Genome Alignment and Annotation) database [37] used DB2 to manage relational databases containing alignments and annotations of multiple genomes, including the UCSC 8-way alignment described later in Section IV-C.3. It supports complex queries on these databases, and was developed as an effective, user-friendly platform for TQG.

There is much more that can be done in database support of alignment tables and TQG. Both Oracle and DB2 have been extended specifically to integrate BLAST itself into the query language. In *Oracle 10g* [35], similarity search can be performed against sequences stored in a local database through a series of BLAST functions. In IBM’s *Information Integrator* [36], BLAST queries are sent to remote BLAST servers and BLAST results are imported through table functions. Advanced controls and features are needed to make sense of BLAST results, such as the annotational information for the sequences and clustering of the BLAST results. BLASTgres also supports direct invocation of BLAST in queries [39], [40], and has many other functions beyond the ones described earlier.

Summarizing: alignment tables are useful and interesting in their own right, and permit a wide variety of implementations. They can be implemented in ways that have been successful for answering large-scale ad hoc queries, such as Trans-Genomic Queries. There is much still to investigate in implementation of alignment tables and TQG.

IV. TRANS-GENOMIC QUERY WITH ALIGNMENT TABLES

Alternative splicing is a good example of a new research area requiring Trans-Genomic Query, as it requires the ability to connect biological information across genomes. After some background material, we show how it is possible to query alternative splicing relationships, exploiting the alignment table examples developed earlier. Specifically we show how queries can be developed that require joining genomic and proteomic information across genomes, following a general graphical view of connectivity between this information suggested by Figure 9.

A. Background on Alternative Splicing

For completeness, we summarize terminology relevant for the examples here. For more detail, there are many online tutorials (e.g., [7]).

An alignment among sequences can be represented as a binary relation among intervals (segments) in them. Figure 2 shows more details of this relationship: a gene (DNA sequence) contains exons which in turn contain coding regions, and introns, which are non-coding regions that are removed by splicing after the gene undergoes transcription. Below we refer to the genomic sequences ‘spliced out’ in this way as *splices*. The mRNA sequence then undergoes translation into protein. Successive 3-symbol blocks of mRNA symbols (codons) are translated to a single amino acid in the protein sequence.

During mRNA processing, introns are removed and exons are spliced together to produce a mature transcript. *Alternative splicing* is the controlled selection of individual exons for addition or removal from the transcript, so that a single gene can produce multiple gene products with differing selections of functional units [25], [26].

Study of alternative splicing has long been a valuable subfield of molecular biology, but until very recently has received less attention to major fields such as human gene identification or transcriptional regulation. Widely regarded as a less common form of functional regulation, it was much less studied than other mechanisms of regulation such as transcriptional control. A consensus estimate, found in textbooks prior to completion of the human genome, suggested that alternative splicing occurred in only 5–15% of genes in complex genomes. Whereas the study of transcription factors and mechanisms of transcriptional regulation is a vast and detailed literature, detailed mechanistic studies of splice regulation have been performed for only a small number of genes [26], [27].

Recently, however, genomics and bioinformatics analyses have indicated that alternative splicing is widespread and of enormous importance in the human genome and in particular for the brain and nervous system. Genome-wide analyses of mRNA sequence fragments (called expressed sequence tags, or *ESTs*) indicated that 40–60% of human genes have alternative splice forms [28], [29], [12], [13]. In the human genome alone, over 30,000 alternative splicing relationships have been identified [14] — effectively doubling the number of gene products for the estimated 25,000 genes in the human genome — and the number is constantly growing.

Genome-wide analyses indicate that alternative splicing is associated with many ‘information processing’ activities in cell biology and in the nervous system [12]. The predominant systemic functions associated with alternatively spliced genes involve nervous system development and immune-response. Indeed, for mammals the function and regulation of alternative splicing has been most extensively studied for neuronal splice forms [27]. Genomics studies of alternative splicing tissue specificity found that the largest group of tissue-specific alternative splice forms was detected in brain, retina, and nerve-derived tissue sources [14]. Indeed, there is currently an explosion of alternative splice form discovery throughout many areas of biomedical research, thanks in part to the avalanche of new data from genomics.

Study of alternative splicing has the potential to increase dramatically our understanding of the human genome if new tools for using this data are made easy to use and widely available. By comparison with the extensive databases and integrated tools that exist for genome maps (e.g. NCBI, Ensembl) and for gene expression data (e.g. Gene Expression Omnibus), alternative splicing databases are still relatively primitive and provide few tools for integrating this data with other biological information sources. These databases mainly let the user see what alternative splice forms exist for specific genes, following a ‘browsing’ model: users are shown graphical views of gene structure and splice forms, and navigate to more detailed information by clicking on specific items. While simple to use, these databases do not enable the user to ask penetrating queries about alternative splicing.

B. Alternative Splicing in HLA-DMB: a Sample Database

As an illustrative example we will use the human gene HLA-DMB, which plays a crucial function in the immune system distinguishing self from non-self [6]. It performs this function at a specific membrane compartment, and its targeting to this compartment and anchoring to the membrane are important for regulation of its function.

This gene is known to be alternatively spliced [10]. Figure 3 shows four alternative splice forms, or *isoforms*, for HLA-DMB. As explained in detail in [12], these isoforms were obtained from eighty ESTs in the Unigene cluster Hs.1162 obtained in January 2002 [8].

The database also contains information about H2-DMb2, which is a mouse ortholog of HLA-DMB. Its mRNA and protein sequences are highly similar to those of HLA-DMB. The relevant genomic sequences are included, showing the intron-exon structure, and providing alignment between orthologous genes in different genomes. Translated protein sequences for each isoform and protein feature information are essential for functional queries. All of these data constitute a multiple sequence alignment, and thus can be queried in a fully general way using alignment tables and TGQ.

This information is represented in the database shown in Figure 11, with the schema shown in Figure 10. This sample database is based on the content of real alternative splicing databases such as ASAP [15] and ASD [32]. It is detailed

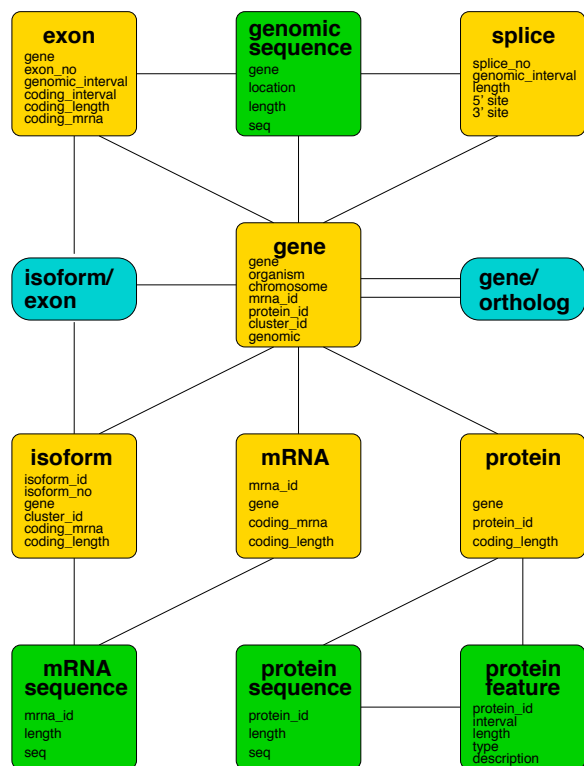


Fig. 10. Schema of the sample database shown in Figure 11 (including three ‘sequence’ tables, which were omitted from Figure 11). The larger boxes represent the indicated entities, while the smaller boxes relationships between these. These relationships are alignment tables. The exon and gene entities also are alignment tables in this schema — they include alignment relationships as attribute information.

enough to illustrate the usefulness of alignment tables in a concrete way, while still fitting in a page.

The database shown in Figure 11 covers only two genomes (Human and Mouse), but the schema is not specific to any genome, and information can be added about any species of interest. An expanded version of this database is downloadable as TGQ.sql from www.BLASTgres.org, including both Mouse orthologs (H2-DMb1 and H2-DMb2) for HLA-DMB. Analysis of multiple genomes is discussed in Section IV-C.3.

C. Applications in Genomic Analysis

How this kind of database produces useful results can be shown with several examples. The basic query shown in Figure 12 finds all (pairs of) alternative splices. This query illustrates how the range predicates described earlier can be used to produce an interesting alignment table that is not a hit table. The dual view of isoforms mentioned earlier, which treats an isoform as a sequence of splices, is effective both because splice observations can be tied directly to individual ESTs, and because determination of alternative splicing is natural in this view. A table obtained from splice information in this way plays a central role in the ASAP databases [15].

These query results show that exons 4 and 5 are *cassette exons* (exons that are skipped in some isoform). Alternative splicing of these exons will add or remove an individual peptide segment from the resulting protein.

gene						
gene	organism	chromosome	mrna_id	protein_id	cluster_id	genomic
HLA-DMB	Homo sapiens	6	GI.18641376	P28068	Hs.1162	GI.17464666[7813399..7823803]
H2-DMb2	Mus musculus	17	GI.6754123	P35737	Mm.195060	GI.28511250[394134..400213]

splice					
gene	splice_no	genomic_interval	length	_5prime_site	_3prime_site
HLA-DMB	1	2289..4075	1787	ACAGGAGCAGgtaaggacac	gctcctctagGTGGCTTCGT
HLA-DMB	2	4358..5584	1227	AACAGGACACgtgaggagag	tcctatgcagGGCCACCATC
HLA-DMB	3	5870..7388	1519	CGGGACTGGAgtaagtgtat	ttttttgcagCACCTGGGCT
HLA-DMB	4	5870..7663	1794	CGGGACTGGAgtaagtgtat	cctctcttagGTTACACTCC
HLA-DMB	5	7506..8053	548	GGCCACTCTAgtgagtgtact	tattctgtagGATGGCACAT
HLA-DMB	6	7506..7663	158	GGCCACTCTAgtgagtgtact	cctctcttagGTTACACTCC
HLA-DMB	7	7664..8053	390	CCTCTCTTAGgttacactcc	tattctgtagGATGGCACAT
HLA-DMB	8	7700..8053	354	TATTCAGAAGgtaaacatctc	tattctgtagGATGGCACAT
H2-DMb2	1	256..2435	2180	ATGGGGGCAGgtaagataac	cttccccagGTGGCTTTGT
H2-DMb2	2	2718..3335	618	CACAGAACGAgtgagcagag	tccccagcagGAGCGCCATC
H2-DMb2	3	3621..5290	1670	GGGGACTGGAgtaagtgtgt	tactttgtagCACCTGGGCT
H2-DMb2	4	5408..5574	167	CATTCTCCAgtgagtgtact	cttctcttagGTTACACTCC
H2-DMb2	5	5611..5869	259	TACCCAGAAGgtaaacatttg	gcttctgcagGACGGCCTCA
H2-DMb2	6	256..1680	1425	ATGGGGGCAGgtaagataac	taacatccagCCACATCTCT
H2-DMb2	7	1779..2435	657	CCCCAAAAGgtaggtgtgc	cttccccagGTGGCTTTGT

exon					
gene	exon_no	genomic_interval	coding_interval	coding_length	coding_mrna
HLA-DMB	1	1264..2288	2234..2288	55	GI.18641376[234..288]
HLA-DMB	2	4076..4357	4076..4357	282	GI.18641376[289..570]
HLA-DMB	3	5585..5869	5585..5869	285	GI.18641376[571..855]
HLA-DMB	4	7389..7505	7389..7505	117	GI.18641376[856..972]
HLA-DMB	5	7664..7699	7664..7699	36	GI.18641376[973..1008]
HLA-DMB	6	8054..8412	8054..8070	17	GI.18641376[1009..1025]
H2-DMb2	1	56..255	56..255	200	GI.31217351[158..212]
H2-DMb2	2	1681..1778	1681..1778	98	GI.31217351[213..229]
H2-DMb2	3	201..255	201..255	55	GI.6754123[1..55]
H2-DMb2	4	2436..2717	2436..2717	282	GI.6754123[56..337]
H2-DMb2	5	3336..3620	3336..3620	285	GI.6754123[338..622]
H2-DMb2	6	5291..5407	5291..5407	117	GI.6754123[623..739]
H2-DMb2	7	5575..5610	5575..5610	36	GI.6754123[740..775]
H2-DMb2	8	5870..5880	5870..5880	11	GI.6754123[776..786]

isoform					
isoform_id	isoform_no	gene	cluster_id	coding_mrna	coding_length
ASAP.55845	1	HLA-DMB	Hs.1162	ASAP.55845[971..1759]	789
ASAP.55846	2	HLA-DMB	Hs.1162	ASAP.55846[971..1606]	636
ASAP.55847	3	HLA-DMB	Hs.1162	ASAP.55847[971..1723]	753
ASAP.55848	4	HLA-DMB	Hs.1162	ASAP.55848[971..1642]	672
GI.6754123	1	H2-DMb2	Hs.1162	GI.6754123[1..786]	786
GI.31217351	2	H2-DMb2	Hs.1162	GI.31217351[158..229]	24

isoform_exon			
isoform_id	interval	gene	exon_no
ASAP.55845	1..1025	HLA-DMB	1
ASAP.55845	1026..1307	HLA-DMB	2
ASAP.55845	1308..1592	HLA-DMB	3
ASAP.55845	1593..1709	HLA-DMB	4
ASAP.55845	1710..1745	HLA-DMB	5
ASAP.55845	1746..2104	HLA-DMB	6
ASAP.55846	1..1025	HLA-DMB	1
ASAP.55846	1026..1307	HLA-DMB	2
ASAP.55846	1308..1592	HLA-DMB	3
ASAP.55846	1593..1951	HLA-DMB	6
ASAP.55847	1..1025	HLA-DMB	1
ASAP.55847	1026..1307	HLA-DMB	2
ASAP.55847	1308..1592	HLA-DMB	3
ASAP.55847	1593..1709	HLA-DMB	4
ASAP.55847	1710..2068	HLA-DMB	6
ASAP.55848	1..1025	HLA-DMB	1
ASAP.55848	1026..1307	HLA-DMB	2
ASAP.55848	1308..1592	HLA-DMB	3
ASAP.55848	1593..1628	HLA-DMB	5
ASAP.55848	1629..1987	HLA-DMB	6
GI.6754123	1..55	H2-DMb2	3
GI.6754123	56..337	H2-DMb2	4
GI.6754123	338..622	H2-DMb2	5
GI.6754123	623..739	H2-DMb2	6
GI.6754123	740..775	H2-DMb2	7
GI.6754123	776..786	H2-DMb2	8
GI.31217351	13..212	H2-DMb2	1
GI.31217351	213..310	H2-DMb2	2
GI.31217351	311..592	H2-DMb2	4
GI.31217351	593..877	H2-DMb2	5
GI.31217351	878..994	H2-DMb2	6
GI.31217351	995..1030	H2-DMb2	7
GI.31217351	1031..1041	H2-DMb2	8

gene_ortholog	
gene	ortholog
HLA-DMB	H2-DMb2

mrna		
gene	coding_mrna	coding_length
HLA-DMB	NM_002118.3[234..1025]	792
H2-DMb2	NM_010388.1[1..786]	786

protein_feature				
protein_id	interval	length	type	description
P28068	1..18	18	signal	potential
P28068	19..112	94	domain	luminal beta-1
P28068	110..110	1	glycosylation	N-linked (GlcNac...)
P28068	113..207	95	domain	luminal beta-2
P28068	114..208	95	domain	Ig-like
P28068	208..218	11	domain	connecting peptide
P28068	219..239	21	transmembrane	potential
P28068	240..263	24	domain	cytoplasmic
P28068	248..251	4	site	YXXZ motif
P35737	1..18	18	signal	potential
P35737	19..112	94	domain	luminal beta-1
P35737	75..75	1	glycosylation	N-linked (GlcNac...)
P35737	113..207	95	domain	luminal beta-2
P35737	114..204	91	domain	Ig-like
P35737	208..218	11	domain	connecting peptide
P35737	219..239	21	transmembrane	potential
P35737	240..261	22	domain	cytoplasmic
P35737	248..251	4	site	YXXZ motif

Fig. 11. Sample database illustrating how biological information, including alignment tables, can be represented in modern database management systems. This particular database supports TGQ concerning alternative splicing of the human gene HLA-DMB and its mouse ortholog H2-DMb2.

```
-- Find instances of alternative splicing

SELECT DISTINCT
  s1.gene,
  s1.splice_no as splice_no1,
  s1.genomic_interval as genomic1,
  upper(substr(s1._5prime_site,11,2)||'/'||substr(s1._3prime_site,9,2)) as type1,
  s2.splice_no as splice_no2,
  s2.genomic_interval as genomic2,
  upper(substr(s2._5prime_site,11,2)||'/'||substr(s2._3prime_site,9,2)) as type2,
  range_size(range_inter(s1.genomic_interval,s2.genomic_interval)) as overlap,
  CASE
    WHEN range_started_by(s1.genomic_interval,s2.genomic_interval) THEN 'same 5\'\'
    WHEN range_finished_by(s1.genomic_interval,s2.genomic_interval) THEN 'same 3\'\'
  END as kind
FROM splice s1, splice s2
WHERE s1.gene = s2.gene AND s1.splice_no <> s2.splice_no
      AND ( range_started_by(s1.genomic_interval,s2.genomic_interval)
          OR range_finished_by(s1.genomic_interval,s2.genomic_interval) )
;
```

gene	splice_no1	genomic1	type1	splice_no2	genomic2	type2	overlap	kind
H2-Dmb2	3	256..2435	GT/AG	1	256..1680	GT/AG	1425	same 5'
H2-Dmb2	3	256..2435	GT/AG	2	1779..2435	GT/AG	657	same 3'
HLA-DMB	4	5870..7663	GT/AG	3	5870..7388	GT/AG	1519	same 5'
HLA-DMB	4	5870..7663	GT/AG	6	7506..7663	GT/AG	158	same 3'
HLA-DMB	5	7506..8053	GT/AG	6	7506..7663	GT/AG	158	same 5'
HLA-DMB	5	7506..8053	GT/AG	7	7664..8053	GT/AG	390	same 3'
HLA-DMB	5	7506..8053	GT/AG	8	7700..8053	GT/AG	354	same 3'
HLA-DMB	7	7664..8053	GT/AG	8	7700..8053	GT/AG	354	same 3'

Fig. 12. A BLASTGRES query, and the output resulting when using the database shown in Figure 11. The query finds instances of alternative splicing, as well as other information, such as the size of the overlap of the splices and the 'type' of each splice (the splice's initial two and final two bases). The resulting table is also an alignment table (a binary relationship on sequence locations). The actual query — the five lines in the WHERE clause — is very concise.

```
-- Find protein feature differences of orthologous genes

SELECT go.gene, go.ortholog, p.protein_id, p.interval, p.type, p.description
FROM gene g1, gene g2, protein_feature p, gene_ortholog go
WHERE g1.protein_id = p.protein_id
      AND ((g1.gene = go.gene AND g2.gene = go.ortholog) OR (g2.gene = go.gene AND g1.gene = go.ortholog))
      AND ((p.type NOT IN (SELECT type from protein_feature WHERE protein_id=g2.protein_id))
          OR (p.interval NOT IN (SELECT interval from protein_feature WHERE protein_id=g2.protein_id))
          OR (p.description NOT IN (SELECT description from protein_feature WHERE protein_id=g2.protein_id)))
ORDER BY go.gene, go.ortholog
;
```

gene	ortholog	protein_id	interval	type	description
HLA-DMB	H2-Dmb2	P35737	75..75	glycosylation	N-linked (GlcNAc...)
HLA-DMB	H2-Dmb2	P35737	114..204	domain	Ig-like
HLA-DMB	H2-Dmb2	P35737	240..261	domain	cytoplasmic
HLA-DMB	H2-Dmb2	P28068	110..110	glycosylation	N-linked (GlcNAc...)
HLA-DMB	H2-Dmb2	P28068	114..208	domain	Ig-like
HLA-DMB	H2-Dmb2	P28068	240..263	domain	cytoplasmic

Fig. 13. BLASTGRES query determining differences in protein features coded for by pairs of orthologous genes.

The following section illustrates how knowledge about HLA-DMB, such as its alternative splicing patterns, can be obtained via TGQ, using alignment tables to connect it with information about the orthologous mouse gene H2-Dmb2.

1) *Exons coding for specific protein features:* As another example of a trans-genomic query, consider the problem of finding exons that code for specific protein features. That is, the range of each exon contains the (mapped) range of certain features. Thus their removal through alternative splicing can have the effect of removing a specific protein function. In [18] this kind of query sought to identify alternatively spliced exon intervals that also intersect with known protein intervals in the Pfam and Swissprot databases.

For HLA-DMB, the situation is illustrated in Figure 14.

The six exons of this gene overlap with specific protein features. Some exons correspond in a close way to protein domains, but others do not. For example, as mentioned earlier, omission of exon 4 in transcription results in omission of the transmembrane domain from the protein product.

Alignment tables permit this kind of query to be answered by reducing them to queries about locations, and then relying on indexing schemes for locations to process these efficiently.

A simple example is shown in Figure 13, comparing protein feature differences across genomes. These query results show that exon 5 contains a signal targeting HLA-DM to endosomal compartments, and removing this may redirect HLA-DM to the plasma membrane. Exon 4 codes for a transmembrane helix that anchors HLA-DM to the membrane, so splicing out

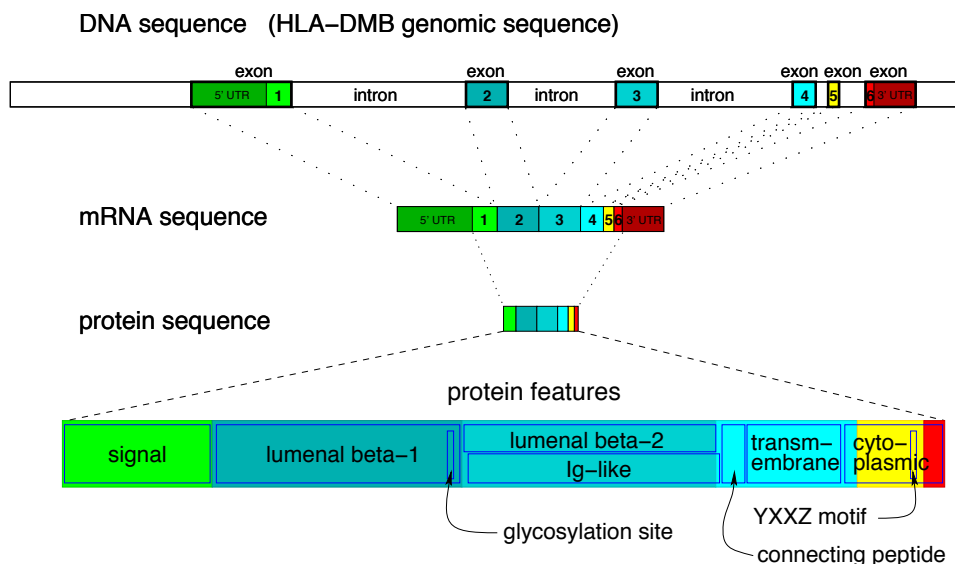


Fig. 14. Transcription and translation of the human gene HLA-DMB, showing the correspondence between the six exons and protein features.

-- Find exons that code for specific protein features

```

SELECT DISTINCT
  e.gene,
  e.exon_no,
  f.protein_id,
  f.interval,
  f.type,
  f.description
FROM
  gene g,
  exon e,
  mrna m,
  protein p,
  protein_feature f
WHERE
  g.gene = 'HLA-DMB' AND e.gene = g.gene AND e.gene = m.gene
  AND e.gene = p.gene AND p.protein_id = f.protein_id
  AND loc_range(e.coding_mrna)
    - loc_lower(m.coding_mrna) -- mrna segment
    + '-2..2':range          -- extended for reading frame
    @                          -- contains
    (f.interval - 1) * 3      -- feature interval
ORDER BY e.gene, e.exon_no
;
    
```

gene	exon_no	protein_id	interval	type	description
HLA-DMB	1	P28068	1..18	signal	potential
HLA-DMB	2	P28068	19..112	domain	luminal beta-1
HLA-DMB	2	P28068	110..110	glycosylation	N-linked (GlcNAc...)
HLA-DMB	3	P28068	113..207	domain	luminal beta-2
HLA-DMB	3	P28068	114..208	domain	Ig-like
HLA-DMB	4	P28068	208..218	domain	connecting peptide
HLA-DMB	4	P28068	219..239	transmembrane	potential
HLA-DMB	5	P28068	248..251	site	YXXZ motif

Fig. 15. This query determines which features are coded for (ultimately generated) by specific exons. It is an interesting query since it relates genomic sequence (DNA) information directly to biological features. For example, the output fact that exon 4 codes for the transmembrane feature is very significant: proteins obtained from isoforms of HLA-DMB that lack this exon (like the second and fourth isoforms in Figure 3) lose the ability to traverse cell membranes, significantly affecting their function. Similarly, exon 5 codes for the lysosomal targeting signal, and is removed in the second and third isoforms in Figure 3.

this exon removes the anchor. Isoforms of HLA-DMB that lack this exon lose the ability to traverse cell membranes.

2) *Effects of alternative splicing on function:* This transmembrane modification is very interesting from a functional point of view, so we have repeated this query over the genome [19]. We used alignment tables to investigate the effects of alternative splicing on transcripts encoding membrane proteins in 1,001 human genes. The example database in this paper, and specifically Figure 15, is a derivative of this work. Out of a total of 464 alternatively spliced genes encoding single-pass transmembrane proteins, in 188 we observed a splice form that specifically removed the transmembrane domain, producing a soluble protein isoform.

For example, in syndecan-4, the new alternative splice form closely parallels the proteolytic ectodomain shedding previ-

ously shown in this protein, and recognized as an important regulatory mechanism of receptor function. While many of the soluble isoforms produced by alternative splicing had already been validated, most were novel, and in 57 genes showed a statistically significant association (p -value < 0.01) with a specific tissue.

3) *Nonconservation of alternative splicing between man and mouse:* A great deal of attention has been given recently to how alternative splicing affects (orthologs of) a given gene in different organisms. Across the human and mouse genomes, there is disagreement as to whether alternative splicing is usually conserved [31] or usually not conserved [30]. Evidently both HLA-DMB and H2-DMb2 are alternatively spliced, but the patterns of alternative splicing of HLA-DMB differ from those of H2-DMb2. This is interesting, since the genes are

```
-- find cassette exons (exons that are skipped in some isoform)very similar.
```

```
CREATE TABLE cassette_exon
AS
SELECT DISTINCT
  e.gene,
  e.exon_no as cassette_exon,
  e.isoform_id
FROM
  isoform_exon e
WHERE EXISTS
  (SELECT * FROM isoform_exon other
   WHERE e.isoform_id <> other.isoform_id
    AND other.gene = e.gene
    AND NOT EXISTS
      (SELECT * FROM isoform_exon
       WHERE isoform_id = other.isoform_id
        AND gene = e.gene
        AND exon_no = e.exon_no))
;
```

cassette_exon	gene	cassette_exon	isoform_id
H2-DMb2		1	GI.31217351
H2-DMb2		2	GI.31217351
H2-DMb2		3	GI.6754123
HLA-DMB		4	ASAP.55845
HLA-DMB		4	ASAP.55847
HLA-DMB		5	ASAP.55845
HLA-DMB		5	ASAP.55848

```
CREATE TABLE peptide_exon_without_stop
AS
SELECT DISTINCT
  e.gene,
  e.exon_no,
  int4(range_size(coding_interval)) as exon_length
FROM
  exon e,
  genomic_sequence gs
WHERE
  e.gene = gs.gene
  AND (range_size(e.coding_interval) % 3) = 0
  AND substr(gs.seq,
            int4(range_lower(e.coding_interval)),
            int4(range_size(e.coding_interval))
            ) !~ '(taa|tag|tga)'
  -- there is no stop in the exon sequence
;
```

peptide_exon_without_stop	gene	exon_no	exon_length
H2-DMb2		7	36
HLA-DMB		5	36

```
-- Find peptide cassette exons without a stop codon
```

```
SELECT DISTINCT
  e.gene,
  e.exon_no,
  e.exon_length
FROM
  peptide_exon_without_stop e
WHERE
  EXISTS
  (SELECT * from cassette_exon
   WHERE gene=e.gene AND exon_no=e.exon_no)
;
```

gene	exon_no	exon_length
H2-DMb2	7	36
HLA-DMB	5	36

Fig. 16. A sequence of small BLASTGRES queries, finding peptide cassette exons (cassette exons whose length is a multiple of 3) that do not contain a STOP subsequence (taa, tag, tga). The `genomic_sequence` table, used in the second query but omitted here, contains genomic sequences as text.

Furthermore, the alternatively spliced exons of HLA-DMB — exons 4 and 5 — have the same lengths as exons 6 and 7 in H2-DMb2, and these lengths have desirable properties, such as that they are a multiple of 3 [33]. Figure 16 expands on this example, showing how both of these exons are cassette exons, and in fact both exon 5 in HLA-DMB and exon 7 in H2-DMb2 are modular in their impact on the protein reading frame. That is they do not alter the downstream protein reading frame, or introduce a STOP codon.

Interestingly, the length of exon 4 (always 117) is almost exactly the average length found in a recent study [33] of exons for which alternative splicing among species is not conserved, and the length of HLA-DMB exon 5 and H2-DMb2 exon 7 (both 36) is significantly shorter than the length found in this study of the average exon that is alternatively spliced in both. It also turns out that the intronic regions flanking these exons are highly conserved, a characteristic for a conserved alternative splicing pattern between human and mouse. Thus, although EST evidence is lacking, alternative splicing of these exons may be conserved among human and mouse [33].

Multiple genomes can be queried in the same general way. Many multi-genome alignments are available from UCSC [38]; the `multiz8way` alignment of 8 genomes (Human, Chimp, Mouse, Rat, Chicken, Fugu, Zebrafish) available at hgdownload.cse.ucsc.edu/goldenPath/hg17 is a good platform for TGQ. It is easily converted to an alignment table; a version converted for BLASTGRES is available at blastgres.cs.ucla.edu/demo/haussler/.

Figure 17 shows the result of a query seeking ranges in other genomes that align with exons in the Human genome. Continuing the analysis above, the query shows that four genomes — Dog, Mouse, Chimp, and Rat — align with HLA-DMB, and that in this case each of these possesses all six exons. Similar queries on the UCSC alignment table can find exons in Human that not aligned to anything in Mouse or Zebrafish, although their neighboring exons are — suggesting exon creation or loss. Related work by the authors on this subject is described in Section IV-D.2 below.

D. Past Success with Alignment Tables for Alternative Splicing

Various applications of TGQ with alignment tables are described in [12]–[22]. Highlights include the following:

1) *Alternative splicing and the human proteome [18]:*

Using alignment tables and a maximum likelihood based graph-traversal algorithm [21], [22], we constructed a database of Alternatively-Spliced Protein forms (ASP), consisting of 13,384 protein isoform sequences of 4,422 human genes. We used alignment tables to search for alternative splice events that removed known protein domains. We identified fifty protein domain types that were selectively removed by alternative splicing at much higher frequencies than average (p -value < 0.01). These include many well-known protein-interaction domains (e.g., KRAB; ankyrin repeats; Kelch) including some that have been previously shown to be regulated functionally by alternative splicing (e.g., collagen domain). The ASP database includes a number of novel examples

```
-- find information about HLA-DMB in the UCSC 8-way alignment

select gene, genomic
from gene
where gene = 'HLA-DMB'
;

-----
| gene | genomic |
-----
| HLA-DMB | hg17.chr6(33010393..33016795) |

select gene, exon_no, genomic_interval, coding_interval, coding_length
from exon
where gene = 'HLA-DMB'
;

-----
| gene | exon_no | genomic_interval | coding_interval | coding_length |
-----
| HLA-DMB | 1 | 33016508..33016795 | 33016508..33016562 | 55 |
| HLA-DMB | 2 | 33014439..33014720 | 33014439..33014720 | 282 |
| HLA-DMB | 3 | 33012927..33013211 | 33012927..33013211 | 285 |
| HLA-DMB | 4 | 33011291..33011407 | 33011291..33011407 | 117 |
| HLA-DMB | 5 | 33011097..33011132 | 33011097..33011132 | 36 |
| HLA-DMB | 6 | 33010393..33010742 | 33010026..33010742 | 17 |

-- get alignments of other genomes against the HLA-DMB gene

select a.src, a.dest, e.exon_no from multiz8way a, exon e, gene g
where g.gene = 'HLA-DMB'
and e.gene = g.gene
and g.genomic && a.src
and loc_range(a.src) && e.coding_interval
;

-----
| src | dest | exon_no |
-----
| hg17.chr6(33010156..33010582) | canFam1.chr12(5422975..5423404) | 6 |
| hg17.chr6(33010583..33012094) | canFam1.chr12(5423405..5424888) | 4 |
| hg17.chr6(33010583..33012094) | canFam1.chr12(5423405..5424888) | 5 |
| hg17.chr6(33010583..33012094) | canFam1.chr12(5423405..5424888) | 6 |
| hg17.chr6(33012796..33013245) | canFam1.chr12(5425488..5425980) | 3 |
| hg17.chr6(33014231..33014756) | canFam1.chr12(5427201..5427735) | 2 |
| hg17.chr6(33016152..33017593) | canFam1.chr12(5427996..5429416) | 1 |
| hg17.chr6(33010156..33010582) | mm5.chr17(60909449..60909818) | 6 |
| hg17.chr6(33010583..33012094) | mm5.chr17(60909965..60911234) | 4 |
| hg17.chr6(33010583..33012094) | mm5.chr17(60909965..60911234) | 5 |
| hg17.chr6(33010583..33012094) | mm5.chr17(60909965..60911234) | 6 |
| hg17.chr6(33012796..33013245) | mm5.chr17(60912215..60912681) | 3 |
| hg17.chr6(33014231..33014756) | mm5.chr17(60913090..60913594) | 2 |
| hg17.chr6(33016152..33017593) | mm5.chr17(60915394..60916733) | 1 |
| hg17.chr6(33010156..33010582) | panTro1.chr5(33450792..33451217) | 6 |
| hg17.chr6(33010583..33012094) | panTro1.chr5(33451218..33452729) | 4 |
| hg17.chr6(33010583..33012094) | panTro1.chr5(33451218..33452729) | 5 |
| hg17.chr6(33010583..33012094) | panTro1.chr5(33451218..33452729) | 6 |
| hg17.chr6(33012796..33013245) | panTro1.chr5(33453528..33453977) | 3 |
| hg17.chr6(33014231..33014756) | panTro1.chr5(33454963..33455488) | 2 |
| hg17.chr6(33016152..33017593) | panTro1.chr5(33456880..33458323) | 1 |
| hg17.chr6(33010156..33010582) | rn3.chr20(4829539..4829933) | 6 |
| hg17.chr6(33010583..33012094) | rn3.chr20(4829934..4831200) | 4 |
| hg17.chr6(33010583..33012094) | rn3.chr20(4829934..4831200) | 5 |
| hg17.chr6(33010583..33012094) | rn3.chr20(4829934..4831200) | 6 |
| hg17.chr6(33012796..33013245) | rn3.chr20(4832253..4832718) | 3 |
| hg17.chr6(33014231..33014756) | rn3.chr20(4834074..4834562) | 2 |
| hg17.chr6(33016152..33017593) | rn3.chr20(4836398..4837791) | 1 |
```

Fig. 17. A subset of the UCSC alignment of 8 genomes using the Human May 2004 (hg17) chromosome 6 as a reference sequence. This query finds all alignments that overlap exons in the hg17 HLA-DMB location (33010392..33016795) on Human chromosome 6; (Since this gene is reverse complemented, exon order is the reverse of sequence order.) Only the canFam1 (Dog), mm5 (Mouse), panTro1 (Chimp), and rn3 (Rat) genomes are recorded in the database as aligning with exons of HLA-DMB.

(Kruppel transcription factors; Pbx2; Enc1) illustrating how this pattern of alternative splicing changes the structure of a biological pathway, by redirecting protein interaction networks at key switch points. Our bioinformatics analysis indicates that a major impact of alternative splicing is removal of protein-protein interaction domains that mediate key linkages in protein interaction networks.

2) *Exon creation/loss and rate of evolution [16]:* We used the initial implementation of alignment query to search for exons that have been created recently during vertebrate evolution, by comparing aligned orthologous gene structures from the human, mouse, rat, and zebrafish genomes. This analysis revealed a very interesting pattern, namely that alternative splicing is strongly associated with a large increase in recent exon creation and loss. Whereas most exons in the mouse and human genomes are strongly conserved in both genomes,

exons that are only included in alternative splice forms (as opposed to the constitutive or major transcript form) are mostly not conserved, and thus the product of recent exon creation and loss events. A similar comparison of orthologous exons in rat and human validates this pattern, indicating that alternative splicing in these genomes has been associated with accelerated evolutionary change.

3) *Tissue-specific alternative splicing [13]:* We developed an automated method for discovering tissue-specific regulation of alternative splicing through a genome-wide analysis of expressed sequence tags (ESTs). Using this approach, we identified 667 tissue-specific alternative splice forms of human genes. We validated the muscle-specific and brain-specific splice forms for known genes. A high fraction (8/10) were reported to have a matching tissue-specificity by independent studies in the published literature. The number of tissue-specific alternative splice forms is highest in brain, while eye retina, muscle, skin, testis and lymph have the greatest enrichment of tissue-specific splicing. Overall, 10% to 30% of human alternatively spliced genes in the data show evidence of tissue-specific splice forms; 78% of the tissue-specific alternative splices appear to be novel discoveries.

4) *Alternative splicing associations with cancer [17]:* We performed a pilot-study identifying tumor-specific alternative splices within 120 genes above LOD score 2 (i.e. $p < 0.01$) in 18 out of 41 different human tissues. In the dataset there are 1768 human genes that have sufficient EST counts to achieve a LOD score equal to or greater than 2 (at least 7 EST observations for a pair of alternative splices). So by chance, only $1768 \times 0.01 = 18$ genes are expected to have a $LOD > 2$. Thus it is likely that the majority of these results are significant. The largest category by total number of tumor-specific splice forms was brain, which represents 48% of all tumor-specific alternative splicing events we observed. Brain was also the biggest category of tissue-specific alternative splicing, and all the tissues above also accounted for a bigger portion of tissue-specific alternative splicing in a previous genome-wide analysis [14].

V. CONCLUSION

The problem of Trans-Genomic Query (TGQ) is of central importance to bioinformatics. However, a common impression is that there is no general, scalable way to implement TGQ. In this paper we have presented a database representation of alignments called *alignment tables*. Alignment tables are binary relationships on locations, and exploit the central importance of sequence alignment to provide a high-quality, queryable backbone of connections between biological information. This method has been used successfully to answer trans-genomic queries on databases involving many millions of records [11]–[21].

We have shown that with a suitable implementation, alignment tables can scale for very large-scale analysis. The paper describes an implementation using BLASTGRES, an extension of the PostgreSQL open-source object-relational database system that includes support for a location datatype and for alignment tables. BLASTGRES is publicly available from

www.BLASTgres.org, or from the *Center for Computational Biology (CCB)* at www.loni.ucla.edu/CCB/Software/.

The paper has also tried to present the potential of alignment tables via a small sample database and a set of illustrative queries. The database relies on features of BLASTGRES, but can be easily transformed to work on any relational database system. There is a great deal more to say, but we have tried to show concretely how alignment tables can be a foundation for connecting biological information, and for solving the problem of Trans-Genomic Query.

ACKNOWLEDGEMENT

We are grateful to the referees for helpful comments that have substantially improved this paper.

REFERENCES

- [1] S.F. Altschul et al., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", *Nucleic Acids Research* 25(17): 3389–3402, 1997.
- [2] A.D. Baxeavanis and B.F. Ouellette, eds., *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins (2nd ed.)*, Wiley Interscience, 2001.
- [3] BioPerl release 1.2.3 documentation, 2003. docs.bioperl.org/releases/bioperl-1.2.3/
- [4] NCBI BLAST: www.ncbi.nlm.nih.gov/BLAST/
- [5] S. Schwartz, W.J. Kent, A. Smit, Z. Zhang, R. Baertsch, R.C. Hardison, D. Haussler, W. Miller, "Human-mouse alignments with BLASTZ", *Genome Res.* 13(1), 103–107, 2003.
- [6] National Cancer Institute, "Understanding the Immune System: Human Tissue Typing for Organ Transplants", press2.nci.nih.gov/sciencebehind/immune/immune29.htm
- [7] NCBI, "What Is a Genome? A Basic Introduction". www.ncbi.nlm.nih.gov/About/primer/genetics_genome.html
- [8] Unigene cluster Hs.1162: HLA-DMB: Major histocompatibility complex, class II, DM beta. www.ncbi.nlm.nih.gov/UniGene/clust.cgi?ORG=Hs&CID=1162
- [9] J.M. Ostell, S.J. Wheelan, J.A. Kans, "The NCBI Data Model", Ch.2 in *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins (2nd ed.)*, A.D. Baxeavanis and B.F. Ouellette, eds., Wiley, 2001.
- [10] J. Shaman, E. von Scheven, P. Morris, M.Y. Chang, E. Mellins, "Analysis of HLA-DMB mutants and -DMB genomic structure", *Immunogenetics* 41: 117–124, 1995.
- [11] K. Irizarry, et al., C. Lee, "Genome-wide analysis of single-nucleotide polymorphisms in human expressed sequences", *Nature Genetics* 26: 233–236, October 2000.
- [12] B. Modrek, A. Resch, C. Grasso, C. Lee, "Genome-wide detection of alternative splicing expressed sequences of human genes", *Nucleic Acids Research* 29(13): 2850–2859, October 2001.
- [13] B. Modrek, C. Lee, "A Genomic view of alternative splicing," *Nature Genetics* 30: 13–19, 2002.
- [14] Q. Xu, B. Modrek, C. Lee, "Genome-wide detection of tissue-specific alternative splicing in the human transcriptome," *Nucleic Acids Res.* 30: 3754–66, 2002.
- [15] C. Lee, L. Atanelov, B. Modrek, Y. Xing, "ASAP: The Alternative Splicing Annotation Project," *Nucleic Acids Res.* 31: 101–105, 2003. www.bioinformatics.ucla.edu/ASAP/
- [16] B. Modrek, C. Lee, "Alternative splicing in the human, mouse and rat genomes is associated with an increased frequency of exon creation and/or loss," *Nature Genetics* 34: 177–180, 2003.
- [17] Q. Xu, C. Lee, "Discovery of novel splice forms and functional analysis of cancer-specific alternative splicing in human expressed sequences," *Nucleic Acids Res.* 31: 5635–5643, 2003.
- [18] A. Resch, Y. Xing, B. Modrek, M. Gorlick, R. Riley, C. Lee, "Assessing the Impact of Alternative Splicing on Domain Interactions in the Human Proteome," *J. Proteome Res.* 3(1): 76–83, 2003.
- [19] Y. Xing, Q. Xu, C. Lee, "Widespread production of novel soluble protein isoforms by alternative splicing removal of transmembrane anchoring domains," *FEBS Lett.* 555: 572–578, 2003.
- [20] A. Resch, Y. Xing, A. Alekseyenko, B. Modrek, C. Lee, "Evidence for a subpopulation of conserved alternative splicing events under selection pressure for protein reading frame preservation," *Nucleic Acids Research* 32(4): 1261–1269, 2004.
- [21] Y. Xing, A. Resch, C. Lee, "The Multiassembly Problem: reconstructing multiple transcript isoforms from EST fragment mixtures." *Genome Res.* 14(3): 426–441, 2004.
- [22] C. Lee, "Generating consensus sequences from partial order multiple sequence alignment graphs". *Bioinformatics* 19: 999–1008, 2003.
- [23] D.S. Parker, M. Gorlick, C. Lee, "Evolving from Bioinformatics in the Small to Bioinformatics in the Large", *OMICS Journal* 7(1): 37–48, 2003.
- [24] P. Valduriez, "Join Indexes", *ACM TODS* 12(2): 218–246, June 1987.
- [25] T. Maniatis, B. Tanis, "Alternative pre-mRNA splicing and proteome expansion in metazoans", *Nature* 418: 236–243, 2002.
- [26] C.W.J. Smith, J. Valcarcel, "Alternative pre-mRNA splicing: the logic of combinatorial control", *TIBS* 25: 381–388, 2000.
- [27] P.J. Grabowski, D.L. Black, "Alternative RNA splicing in the nervous system", *Prog. Neurobiol.* 65: 289–308, 2001.
- [28] A.A. Mironov, J.W. Fickett, M.S. Gelfand, "Frequent alternative splicing of human genes", *Genome Res.* 9: 1288–1293, 1999.
- [29] D. Brett et al., "EST comparison indicates 38% of human mRNAs contain possible alternative splice forms", *FEBS Letters* 474: 83–86, 2000.
- [30] R.N. Nurtdinov, I.I. Artamonova, A.A. Mironov, M.S. Gelfand, "Low conservation of alternative splicing patterns in the human and mouse genomes". *Human Molecular Genetics* 12(11): 1313–1320, 2003.
- [31] T.A. Thanaraj, F. Clark, J. Muilu, "Conservation of human alternative splice events in mouse", *Nucleic Acids Res.* 31(10): 2544–2552, 2003.
- [32] T.A. Thanaraj et al., "ASD: The Alternative Splicing Database", *Nucleic Acids Res.* 32: D64–D69, 2004.
- [33] R. Sorek, R. Shamir, G. Ast, "How prevalent is functional alternative splicing in the human genome?", *Trends in Genetics* 20(2): 68–71, 2004.
- [34] James F. Allen, "Maintaining knowledge about temporal intervals", *Communications of the ACM* 26(11): 832–843, 1983.
- [35] S.M. Stephens, J.Y. Chen, M.G. Davidson, S. Thomas, B.M. Trute, "Oracle Database 10g: a platform for BLAST search and Regular Expression pattern matching in life sciences", *Nucleic Acids Res.*, 2005.
- [36] B. Eckman, D.D. Prete, "Efficient Access to BLAST using IBM DB2 Information Integrator", *IBM Healthcare and Life Sciences*, 2004.
- [37] B. Giardine, L. Eltnitski, C. Riemer, I. Makalowska, S. Schwartz, W. Miller, R.C. Hardison, "GALA, a database for genomic sequence alignments and annotations", *Genome Research* 13: 732–741, 2003.
- [38] W.J. Kent, C.W. Sugnet, T.S. Furey, K.M. Roskin, T.H. Pringle, A.M. Zahler, and D. Haussler, "The Human Genome Browser at UCSC", *Genome Res.* 12: 996–1006, 2002. hgdownload.cse.ucsc.edu/downloads.html
- [39] R.L. Hsiao, D.S. Parker, "BLASTgres (an extension of the Postgres database system for BLAST and large-scale bioinformatics)", demonstration session, *Intl. Symp. on Molecular Biology (ISMB 2005)*, 2005.
- [40] R.L. Hsiao, D.S. Parker, "The BLASTgres Database System", *Proc. Data Integration in the Life Sciences*, Springer-Verlag, San Diego, 2005.
- [41] J. Enderle, M. Hampel, T. Seidl, "Joining Interval Data in Relational Databases", *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, 683–694, 2004.
- [42] H.-P. Kriegel, M. Poetke, T. Seidl, "Managing Intervals Efficiently in Object-Relational Databases", *Proc. 26th Intl. Conf. on Very Large Data Bases (VLDB)*, 407–418, 2000.
- [43] H. Edelsbrunner, "Dynamic Rectangle Intersection Searching", Institute for Information Processing, Report 47, Tech. U. Graz, Austria, 1980.
- [44] Preparata F. P., Shamos M. I., *Computational Geometry: An Introduction (5th ed.)*, Springer-Verlag, 1993.
- [45] K.R. Dittrich, A. Geppert, *Component Database Systems*, Morgan-Kaufmann, 2000.
- [46] A. Guttman, "R-Trees: A Dynamic Index Structure For Spatial Searching", *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, Boston, 47–57, 1984.
- [47] J. Hellerstein, J.F. Naughton, A. Pfeffer, "Generalized Search Trees for Database Systems", *Proc. VLDB*, 1995. Available from the GIST site: gist.cs.berkeley.edu:8000/gist/gist1.html
- [48] PostgreSQL documentation (including version 7.4 Reference Manual, 2003). www.postgresql.org/docs/
- [49] D.E. Shasha, P. Bonnet, *Database Tuning: Principles, Experiments, and Troubleshooting Techniques*, revised ed., Morgan-Kaufmann, 2002.
- [50] J. Srinivasan et al., "Extensible Indexing: a Framework for Integrating Domain-Specific Indexing Schemes into Oracle8i", *Proc. ICDE*, San Diego, 2003.



D. Stott Parker received an A.B. in Mathematics from Princeton in 1974, and a Ph.D. in Computer Science from the University of Illinois in Urbana-Champaign in 1978. He joined the Faculty of the UCLA Computer Science Department in 1979. His current interests center around data mining in general, and scientific databases in particular. He is currently a co-director of the *Center for Computational Biology* at UCLA, and is involved in a number of projects involving bioinformatics, computational biology, and scientific data mining.



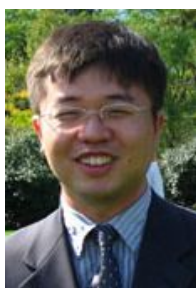
Alissa M. Resch received a Ph.D. in Biochemistry from the University of California, Los Angeles in 2004. She is currently a Postdoctoral Fellow with the Koonin Group at the National Center for Biotechnology Information (NCBI). Her research interests include large-scale comparative genomics, evolution of gene structure and alternative splicing.



Ruey-Lung Hsiao is currently a graduate student at UCLA Computer Science Department. His research interests include scientific data management, bioinformatics and data mining. He also works on some open-source development projects, including *BLASTgres* and *BioPostgres*. He is currently working on models for scientific databases and frameworks for knowledge discovery.



Christopher J. Lee received a B.A. summa cum laude in Biochemistry and Molecular Biology from Harvard College in 1988. He received a Ph.D. in Structural Biology from Stanford University in 1993. With Michael Levitt of Stanford he co-founded Molecular Applications Group, an early bioinformatics software company, in Palo Alto, CA, and served as Vice-President of Research and Development, as well as lead developer of its LOOK / GeneMine software products. After the acquisition of MAG by Celera in 1998, Dr. Lee joined the faculty of the Department of Chemistry and Biochemistry, University of California at Los Angeles, where he is currently Associate Professor. His research focuses on bioinformatics and computational genome analysis. Recent computational projects include graph algorithms for multiple sequence alignment; graph database query of genomics data; new query algorithms for multigenome alignment databases; and graph models of evolutionary kinetics. Dr. Lee has published approximately 70 articles in scientific journals.



Yi Xing is an Assistant Professor in the Department of Internal Medicine, Roy J. and Lucille A. Carver College of Medicine, University of Iowa. He was a Ph.D. student with Christopher Lee at University of California, Los Angeles (2001-2006), and a visiting scientist with Wing Hung Wong and Matthew Scott at Stanford University (2006-2007). His major research interest is genomics and bioinformatics of eukaryotic gene regulation.