Published in final edited form as:

Int Conf Sci Stat Database Manag. 2009 ; 5566: 200–216. doi:10.1007/978-3-642-02279-1_15.

Covariant Evolutionary Event Analysis for Base Interaction Prediction Using a Relational Database Management System for RNA

WeiJia Xu¹, Stuart Ozer², and Robin R. Gutell³

WeiJia Xu: xwj@tacc.utexas.edu; Stuart Ozer: stuarto@microsoft.com; Robin R. Gutell: robin.gutell@mail.utexas.edu

¹ Texas Advanced Computing Center, The University of Texas, Austin, Texas, USA

² Microsoft Corporation, One Microsoft Way, Redmond, WA., Seattle, Washington, USA

³ Center of Computational Biology and Bioinformatics, The University of Texas, Austin, Texas, USA

Abstract

With an increasingly large amount of sequences properly aligned, comparative sequence analysis can accurately identify not only common structures formed by standard base pairing but also new types of structural elements and constraints. However, traditional methods are too computationally expensive to perform well on large scale alignment and less effective with the sequences from diversified phylogenetic classifications. We propose a new approach that utilizes coevolutional rates among pairs of nucleotide positions using phylogenetic and evolutionary relationships of the organisms of aligned sequences. With a novel data schema to manage relevant information within a relational database, our method, implemented with a Microsoft SQL Server 2005, showed 90% sensitivity in identifying base pair interactions among 16S ribosomal RNA sequences from Bacteria, at a scale 40 times bigger and 50% better sensitivity than a previous study. The results also indicated covariation signals for a few sets of cross-strand base stacking pairs in secondary structure helices, and other subtle constraints in the RNA structure.

Keywords

Biological database; Bioinformatics; Sequence Analysis; RNA

1 Introduction

Comparative sequence analysis has been successfully utilized to identify RNA structures that are common to different families of properly aligned RNA sequences. Here we present enhance the capabilities of relational database management for comparative sequence analysis through extended data schema and integrative analysis routines. The novel data schema establishes the foundation that analyzes multiple dimensions of RNA sequence, sequence alignment, different aspects of 2D and 3D RNA structure information and phylogenetic/evolution information. The integrative analysis routines are unique, scale to large volumes of data, and provide better accuracy and performance. With these database enhancements, we details the computational approach to score the number of mutual or concurrent changes at two positions in a base pair during the evolution of Bacterial 16S rRNA molecules.

Since early studies on covariation analysis [1–4] analyzed the nucleotide and base pair frequencies, it has been appreciated that measuring the number of evolutionary events increases the accuracy and sensitivity of covariation analysis. Examples include prediction of a pseudoknot helix in the 16S rRNA [4] and determining divergent and convergent evolution of

the tetraloop [5]. More commonly, evolutionary events were found anecdotally, suggesting that many more can be identified from a more computationally intensive search for *evolutionary events*.

This particular problem requires the traversal of the phylogenetic tree hierarchy, potentially within only selected phylogenetic branches, joined with the existing multiple sequence alignment at large scale to identify sequence changes at paired positions. Over the years, the number of RNA sequences dramatically increased [6]. The number of sequences in ribosomal RNA alignments have grown from under 12, 000 sequences 5 years ago to nearly 200, 000 sequences today and the growth is accelerating[7]. The traditional flat file storage format for multiple sequence alignment [8] and the phylogenetic information from heterogenous data provenance not only requires custom data accessing methods but also pushes the limits of memory scale and affordability for available commodity hardware.

Our approach enhances the capabilities of a relational database management system (RDBMS), in this case Microsoft SQL Server 2005 [9] to provide a scalable solution for a class of problems in bioinformatics through data schema and integrated analysis routines. The value of integrating biological sequence analysis into relational databases has been highlighted by commercial offerings from Oracle [10] and IBM (DB2) [11]. A number of research projects have made broad fundamental attacks on this problem. Patel et al. have proposed a system extending existing relational database systems to support queries for the retrieval of sequences based on motifs [12]. Miranker et al. intended to support built-in database support for homology search [13–15]. Periscope/SQ system extends SQL to include data types to represent and manipulate local-alignments and together with operators that support both weighted matching of subsequences and, regular expression matching [16].

A key contribution of our work is a data schema that unifies various types and dimensions of RNA information within a single relational database system. Data from various sources, including sequences and their annotations, sequence alignments, different types of higher-order structure associated with each position for each sequence and taxonomy for each of the rRNA sequences are maintained in our RNA Comparative Analysis Database (rCAD)[17]. The data schema of developed in rCAD closely ties together phylogenetic, sequence and structural information into a form that is readily analyzable using relatively simple SQL expressions -- dramatically simplifying a wide family of tasks that used to require complex custom programming against 'closed' data structures. Analysis of these multiple dimensions of data in rCAD is performed within this SQL-server system as stored procedures written in the C# language and T-SQL. With this integrated data storage and analysis system, significant reductions in the runtime were achieved by only analyzing those pairs of positions that might possibly have a significant concurrent change or covariation. Our approach is evaluated for both performance and accuracy of the result and compared with a previous study[18].

2 Background and Related Work of Evolutionary Covariation Analysis

2.1 Background for Covariation Analysis on Aligned Sequences

Covariation analysis, a specific form of comparative analysis, identifies positions with similar patterns of variation indicating that these sets of positions in a sequence evolve (ie. mutate) at similar times during their evolution, as determined from their phylogenetic relationships. Thus the positional variation is revealing a dependency in the patterns of variation for certain sets of positions in a multiple sequence alignment. And when this dependency is strong we have interpreted that the two positions that covary are base paired with one another. The accuracy of this method was rigorously tested and substantiated from a comparison of the covariation structure model and the high-resolution crystal structures for several different RNA molecules [19].

While the high accuracy of the predicted structure with covariation analysis and the elucidation of tertiary or tertiary-like base-base interactions substantiated the utility of the covariation analysis, previous analyses [2] has also revealed that this form for comparative analysis on a set of aligned sequences can reveal lower degrees of dependency between the patterns of variation for sets of positions that either indicate a base pair, a base triple, or a more complex structural unit that involves more than two positions, for example, hairpin loops with four nucleotides, tetraloops[5].

2.2 Related Work

Although the concept and importance of an evolutionary event has been stated in numerous publications [4,20–28], there are few sophisticated computational methods that can systematically identify them[18,29]. Both of these methods are recent. Yeang et. al. derive a probabilistic graphical model to detect the coevolution from a given multiple sequence alignment[18]. This approach extends the continuous-time Markov model for substitution at single position to that for substitutions at paired positions. In addition to the computationally expensive model-based computation steps, this approach requires the computation of evolutionary distance among sequences in preprocessing steps using an external program with a computational cost of $O(n^3)$ [30,31]. Therefore, this approach doesn't scale over large multiple sequence alignment and has limited applicability. The experiments reported in [18] are limited to only 146 16S ribosomal RNA sequences sampled from different phylogenetic branches within the Bacteria domain. Dutheil et al. used a Markov model approach to map substitution of a set of aligned sequences to the underlying phylogenetic tree. The approach applied to a bacterial rRNA sequences from 79 species[29].

In contrast to these model-based approaches, our method utilizes known phylogenetic relationships and identifies the minimal number of mutual changes for two positions during the evolution of the RNA under study. The major computational objective is to identify the pair of positions with similar patterns of variation and to increase the score for those covariations that have occurred more frequently during the evolution of the RNA sequences. Our method functions on any given multiple sequence alignment and existing phylogenetic classifications without additional preprocessing. With a novel data schema for storing multiple sequence alignments and phylogenetic tree reconstructions in rCAD, our approach exploits the efficiency of the database system and the effectiveness of C# running within the query engine to perform recursive computations on tree structures, thus enabling analysis of more than 4200 Bacterial 16S ribosomal RNA sequences that are distributed across the entire Bacterial phylogenetic domain (approximately 2337 different phylogenetic branches). To the best of our knowledge, this is the largest number of sequences analyzed with a phylogenetic-based covariation method.

3 Methods and Implementations

Central to our approach is a novel data schema for managing multiple dimensions of data using a relational database system and a coarse filter to facilitate the online computations. Due to the size of input data, two main challenges are the accessibility of the data and computational feasibility. Effectively accessing a large amount of various types of data is not a trivial step. In this case, each position of a multiple sequence alignment must be easily accessed by either row or column and any branch of the phylogenetic tree must be able to be hierarchically accessible. For computational efficiency and ease of development, our approach integrates the data analysis process directly into the database management system.

Our computational approach includes several steps as illustrated in Figure 1. For a given multiple sequence alignment, a candidate set of pairs of positions (columns) is first selected from all possible pairs. Since it is computationally expensive to compute mutual information

for every possible pair over a large alignment, we developed a coarse filter based on our empirical observations. The coarse filter effectively reduces the workload for computing mutual information. Filtered pairs are then evaluated for mutual information content, and candidate pairings are identified from the mutual information scores. Pairs are then predicted by gathering the number of mutation events in each candidate pairs' positions based on the phylogenetic tree. Pairs with potential interactions are then predicated based on the percentage of covariant events of total observed events and the number of total events observed.

Any set of sequences can be selected from the existing alignment table based on various criteria with no need to repopulate the database or tables. Additionally, intermediate computing results are stored within the database as tables. Those tables can be used to trace back/repeat an analysis or for repurposing previous computations beyond the initial experimental design. For example, computations on a subset of sequences from a specific taxonomy group maybe derived from results of previous runs with larger scope

3.1 Data Schema and Database Design

The design of rCAD is fundamentally based on the idea that no data in the tables need ever be re-stated or re-organized. All access to the alignment data is typically driven by a sub-query that selects the sequence IDs of interest and places those values in a table variable or temp table as part of the analysis. Analysis queries then join these IDs to the sequence data itself present in the SEQUENCE table. Selection of sequences for display, analysis or export is typically specified using the sequences' location in the taxonomy hierarchy, combined with the ID of the type of molecule and alignment of interest. Other properties present in the Sequence Main table or related attributes can also be used for selection. The data structures are designed to hold not only a single alignment, but an entire library of aligned sequences spanning many RNA molecules of many types from many species. Figure 2 shows only tables used to store Sequence metadata, taxonomy data and alignment data that are essential for event counting studies.

Sequence Metadata—*Information about available RNA sequences is centered on the “SequenceMain” table, which links each sequence to its Genbank accession numbers, cell location (e.g. mitochondria, chloroplast or nucleus), NCBI Taxonomy assignment, raw (unaligned) sequence, and other source annotations. Each sequence is uniquely identified by a locally-assigned integer SeqID.*

Taxonomy Data—While the NCBI taxonomy assignment of each sequence is present in “SequenceMain” via a locally maintained integer *TaxID*, we need two tables “TaxonomyNames” and “AlternateNames” to associate each *TaxID* with the scientific and alternative names of the taxa. However, these two tables alone cannot provide any evolutionary relations among taxons. The “Taxonomy” table retains the basic parent-child relationships defining the phylogenetic tree. When joined with the “Sequence-Main” table, SQL’s ability for handling recursive queries makes it straightforward to perform selections of sequences that lie only on specific branches of the phylogeny tree. We also materialize a derived table, “Taxonomy Names Ordered”, that contains, for each taxon, a full description of its ancestral lineage.

Sequence Alignment—The bulk of the database records are contained in the tables representing the sequence alignments. Each alignment, identified by an integer *AlnID*, can be thought of conceptually as a 2-dimensional grid, with columns representing structurally aligned positions among sequences, and rows representing the distinct sequences from different species and cell locations. Each family of molecules, such as 16S rRNA, 23S rRNA, or tRNAs correspond to one or possibly more alignments. Our main design concern is to (1) avoid storing

data for alignment gaps and (2) minimize the impact of inserting a sequence that introduces new columns within an existing alignment.

The table “*Sequence*” contains data only for positions in each sequence that are populated with a nucleotide. Each entry is keyed by *SeqID* identifying the sequence, *AlnID* identifying the alignment. The field *Base* specifies the nucleotide present at that position of the sequence. The field *NucleotideNumber* is populated with the absolute position of that location within the sequence. The field *ColumnNumber* is a unique integer key assigned to every column of an alignment. With this schema, the *ColumnNumber* associated with an entry in “*Sequence*” never changes even when additional columns are inserted into the alignment. The table “*AlignmentColumn*” associates a *ColumnNumber* and its actual position in an alignment, which is noted as field *ColumnOrdinal*. Thus, if adding a new sequence requires inserting new columns into an alignment, only the values of *ColumnOrdinal* for a small number of entries in “*AlignmentColumn*” need to be updated to reflect the new ordering of columns. No data in the potentially huge and heavily indexed “*Sequence*” table need be changed. Gaps in the alignment are easily identified by the absence of data in the “*Sequence*” table. The table “*AlnSequence*” defines the first and last column number within an alignment for which sequence data exists for a given sequence.

Table indexes are created to facilitate data access. The queries that perform the Relative Entropy calculations and Mutual Information scoring of all candidate column pairs rely on the Clustered Index (alignment ID, Sequence ID, Column ID) to execute a high-performance sequential read covering exactly the range of sequence data of interest. Queries that perform the event counting of candidate column pairs use a covering secondary index (alignment ID, Column ID, Sequence ID) to quickly extract only the data from column pairs of interest within a specific alignment, avoiding unnecessary reads.

3.2 An Entropy Based Coarse Filter

The mutual information score is a measure of the reduction of randomness of one position given the knowledge of the other position. A high mutual information score indicates a strong correlation between two positions. The Mutual information score and its variations have been used in several studies [29,32,33]. Given a multiple sequence alignment, let x and y denote the two positions of interest, then mutual information (MIXY) between column x and column y is calculated as

$$MI(x;y)=\sum_{i,j}p_{12}(x_i,y_j)\ln\frac{p_{12}(x_i,y_j)}{p_1(x_i)p_2(y_j)}$$

where $p(x, y)$ is the joint probability function of x and y , and $p(x)$ and $p(y)$ is the marginal probability for position x and y .

With a multiple sequence alignment of m columns and n sequences, the time complexity for computing mutual information of each pair of columns is $O(n)$. The total possible column pairs to be considered is in the order of $O(m^2)$. Hence the total computational cost to computing mutual information among every possible pair is in the order of $O(m^2n)$. The multiple sequence alignment for 16S bacteria RNA sequence used here consists of 4,000 sequences and 3,237 columns. To speed up this computation, we developed a fast coarse filtering method based on empirical observations. Note that the compositional distribution between covariance columns always shows minimal difference. The compositional distribution difference between two positions x and y can be measured as

$$H(vx \parallel vy) = \sum_i p(vx_i) \log_2 \frac{p(vx_i)}{p(vy_i)}$$

Our approach uses modified relative entropy as a coarse filter. Only pairs with relative entropy score less than a predefined threshold are selected for mutual information computation. For each pair of positions, the computation of relative entropy is linear to the number of sequences. As shown in the result section, the relative entropy filter can significantly reduce the search space for mutual information computation and speed up the total running time. However, the probability distribution is easily skewed if the sample size is relative small. This is also the case that we observed here. A number of columns in the alignment contain a large portion of gap symbols rather than nucleotide bases. If there are very few bases in a column, the estimated results become unreliable. Therefore, we also prune out columns in which the number of bases are less than a predefined threshold.

With the filtered set of candidate column pairs, we then perform a mutual information calculation to compute the MIXY score. Candidate base pairs are identified by considering all column pairings (x,y) where the MIXY score for (x,y) is among the top N MIXY scores for all pairs (x,k), and the MIXY score for (x,y) is also among the top N MIXY scores for all pairs (k,y). This process is known as finding the Mutual N-Best MIXY scores, and can be executed as a single SQL query over the set of all scores for candidate pairs. N is typically set as 100~200 to identify a broad range of candidates for phylogenetic event counting analysis.

Sequences used in the input multiple sequence alignments are grouped according to their taxonomy classification. For each candidate pair, we accumulate the number of evolutionary events within each taxonomy group. There are two types of evolutionary events.

Definition 1 Evolutionary event—Given a pair of positions of a set of aligned sequences and their ancestral sequence, an evolutionary event is observed when there is at least one mutation from its ancestral sequence for each sequence.

Definition 2 Covariant evolutionary event—Given a pair of positions of a sequence, a covariant evolutionary event is observed when both positions contain mutations from its ancestral sequence for every sequence.

In practice, there are usually no actual ancestral sequences at every internal node of a phylogeny tree. Therefore, given an arbitrary set of sequences, we define an equality set for each position as candidates for ancestral sequences.

Definition 3 Equality set—Given a multiple sequence alignment, an equality set is defined as the set of base(s) with the maximum number of occurrences at those column(s) in the multiple sequence alignment.

Figure 3 shows pseudo code of the event counting algorithm. We use a variation of Fitch's maximum parsimony approach adapted for non-binary trees since any node in the phylogeny tree can contain an arbitrary number of sequences. We first determine the type of pair represented by a parent node, as the *equality set*, which may be a mixture of pair types (Lines 2–5). Based on the *equality set* of a node, we then compute the number of events and covariations exhibited by children at that node (Lines 7–11). We only count a pair type once per occurrence in a child equality set and do not count the number of underlying descendant pairs of that that type (Lines 12–16). The event count resulting from choosing any member of the equality set as the parent will be the same as the count from any other parent. However,

since covariant counts will differ based on parent selections; we only considered the minimum count in this case (Lines 17,18).

The algorithm is implemented as a C# stored procedure that bridges the ‘impedance mismatch’ between the types of queries that the relational model natively supports, and computations desired on more complex data structures such as graphs and trees. The Event Counting procedure initially takes the parent-child relationships expressed in the Taxonomy table and populates a high-performance recursive set of .Net Treenode data structures at query execution time -- which is then used to easily compute the event counts. This is an important illustration of how we can extend the expressiveness of SQL Queries to embody a rich computation close to the data.

4 Performance and Experiments Results

The size of alignment considered here is significantly larger than other related works and poses a major computational challenge. The alignment data used in this paper are publicly available at the CRW website[34]. The alignment is generated and curated by human experts over the past 20 years. New sequences have been periodically added into this alignment through a progressive alignment approach. The Bacteria 16S rRNA alignment consists of 4200 sequences, 3237 columns, and 6,103,090 bases. The phylogeny tree is downloaded from NCBI and stored into relational tables as described in the section 2. The phylogeny tree data is also updated periodically. At the time of this experiment, the sequences were from 2337 different classification groups.

4.1 Database Performance Evaluations

Figure 4 shows an overview of the execution time of the entire computational process. The entire computing process took over 28 hours. The evolutionary event counting step accounts for 98.9% computing time. The covariation computation and filtering steps account for 77.90% and 19.77% of the rest of the computations. Based on the computational complexity, the entire computation consists of two parts. The first part is to create a list of candidate pairs based on an existing multiple sequence alignment. The second part is to calculate the evolutionary events for the candidate pairs. The computational complexity of the first part depends on the number of sequences and the length of the aligned sequences. The computational complexity of the second part depends on the size of the candidate pairs and number of taxons in the taxonomy tree. In practice, the second part of the computation accounts for 99% of the computations.

Our approach enables a flexible way to select multiple sequence alignments from existing data. Using the SQL selection query, an arbitrary ‘window’ view of an alignment can be created from a subset of alignments or a range of columns. The selection query can be further combined with information stored in the other tables to construct a customized data set such as one based on composition statistics or phylogenetic information. With these features, we created a series of subsets of alignments to study the database performance.

Figure 5 (left) shows the evolutionary counting part of its performance with regard to the number of candidate pairs. In this test, we selected a subset of sequences which are covered by a taxonomy tree with 496 taxons. We then only select a fixed number of pairs from the set of candidate pairs. The CPU time is reported through a feature of the MS SQL server. Figure 5 (right) shows the evolutionary counting performance with regard to the number of taxons considered. For each testing subset of sequences, the number of taxons required various from 496 to 1450. The number of candidate pairs used in the computations is limited at 4000 for each test set.

4.2 Effectiveness of Coarse Filtering Methods

To speed up covariation computation, we implemented a coarse filter to select pair candidates. In the coarse filtering step, each column in a multiple sequence alignment is inspected with two criteria: the number of bases contained within each column and the relative entropy of that column.

A multiple sequence alignment can be viewed as a matrix with a fixed number of columns. Each sequence, as a row in this matrix, is filled with gaps in order to properly align conserved bases. As more diversified sequences across the phylogeny tree are aligned, the number of gaps is expected to increase. From a practical consideration, a position with a large number of gaps, i.e. a small number of bases, is unlikely to be found in a covariant pair.

Figure 6 shows the number of possible pairs grouped by the minimum number of bases contained in each position for each five hundred interval. For example, the group labeled as 2500 shows that there are 79,255 possible pairs in which the minimum number of bases of the two positions is between 2000 and 2500. Note that the column for the first 500-group is truncated for presentation. There are 3,924,099 pairs which contain positions with less than 500 non-gap bases, and this accounts for about 75% of the total 5,182,590 possible pairs. For each pair in the first 500-group, each pair contains at least one column filled with more than an 88% gap. From practical experience, we pruned those pairs for further consideration.

The second score used in coarse filtering is the relative entropy score described in the previous section. For each column, the relative entropy is computed. Pairs with a maximum relative entropy value higher than a given threshold are pruned.

Figure 7 plots 51,820 pairs from all possible pairwise comparisons for the Bacteria 16S rRNA alignment. The mutual information for each selected pair is higher than 0.1. The plot shows a good correspondence between maximum relative entropy score and mutual information scores with only a handful of outliers. From practical experience, we determined a threshold of 0.5 for relative entropy score computations (Table 1). Figure 8 shows the comparison of total running time for covariant analysis with and without coarse filters.

4.3 Performance Comparison with CO Model

Since our approach is fully integrated within the relational database system and the scale of data considered here, a precise performance comparison with related work which is implemented in functional programming language is not straightforward. Yeang et al. implemented the model computation in C and reported over 5 hours of running time for 146 sequences[18]. However, this running time does not include the computation of evolutionary distance among sequences using DNAPARS in the PHYLIP package[35]. In our test using the same code and data set provided by the authors, the same computation takes 6 hours and 1 minute to run in command prompt using our sql server machine. However, for the 4200 16S rRNA sequences used in our study, both preprocessing and model computation are failed to finish within the 10 days time frame. Since our approach was finished within 28-hour time frames, we conclude our method is more efficient than related work due to the fact the other method failed at running on the scale of our dataset.

4.4 Experimental Results Evaluations

From the annotations maintained at CRW, we expect to find 357 secondary base pairing interactions and 17 tertiary pairing interactions in our predicated data set. In our approach, a pair of positions is considered as coevolved if there are more than 50% of co-evolutionary events among all accumulated evolutionary events. Figure 9 shows the percentage of annotated base pairs when using the percentage of covariation events. The figure shows very high

selectivity. For example, there are 244 out of 271 (or 90%) of pairs with more than 50% of covariation events are known base pairs. Therefore, our approach can achieve 90% accuracy in identifying known interacting pairs. By comparison, Yeang et al. reported the 57% sensitivity of using a proposed CO model to identify 251 secondary interactions with 150 false positives and much lower sensitivities with other models including the Watson–Crick co-evolution model and using mutual information.

In addition to the results that are consistent to pairs with known base pairing interactions, our results also indicate candidate pairs that have lower constraints (or non-independence) on the evolution of RNA sequences. While the vast majority of the covariations have been interpreted correctly as a base pair interaction between the two positions with similar patterns of variation [19], exemplar interactions between non-base paired nucleotides have been identified in the literature such as base triple tertiary interactions in helix and tetraloops [5,36]. In those situations, a number of nucleotides within the same helix or loop also show a tendency of coevolution over time but are not base paired with one another. Instead of forming hydrogen bonding, the bases of these nucleotides are partially stacked onto one another or same strand or cross strand. However, those interacting nucleotide sites are often filtered out by traditional covariation analysis due to a lack of contrast from the background.

5 Conclusions and Discussions

While the significant increase in the number of nucleic acid sequences creates the opportunity to decipher very subtle patterns in biology, this overwhelming amount of information is creating new challenges in the management and analysis with computational methods. Our approach enhances the capability of a modern relational database management system for comparative sequence analysis in bioinformatics through extended data schema and integrated analysis routines. Consequently, the features of RDBMS also improve the solutions of a set of analysis tasks. For example with dynamic memory resources management, alignment sizes are no longer limited by the amount of memory on a server. And analysis routines no longer need to incur the runtime overhead of loading entire alignments into RAM before calculations can begin. As a result alignments can now be analyzed in the order of thousands of columns and sequences to support comparative analysis.

The extended data schema presented here is not designed specifically for the covariant evolutionary event analysis. It is designed to facilitate a class of sequence analysis tasks based on multiple sequence alignment such as computing template multiple sequence alignments, and performing conventional covariation analysis. It is the data schema like this that enables us to conduct innovative analysis on tasks such as covariant evolutionary event counting presented in this paper. Several other innovative analysis methods enabled by this data schema are being developed by our research group, including statistical analysis on sequence compositions and structures. However, we won't be able to present all of those related research tasks in complete detail in this paper.

Furthermore, integrating analysis within the relational database environment has enabled simpler, faster, more flexible and scalable analytics than were possible before. Now concise SQL queries can substitute for pages of custom C++ or script programming. To store analytical results in relational tables simplifies the process of forming an analysis pipeline. New analysis opportunities also emerge when results from different query, parameters can be joined together or with updated data. We expect the database management system to play a key role in automating comparative sequence analysis.

While the computational improvements noted above make it possible to analyze significantly larger datasets and integrate different dimensions of information, our analysis reveals that this

newer comparative analysis system has increased sensitivity and accuracy. Our preliminary results are identifying base pairs with higher accuracy, and with higher sensitivity we are finding that a larger set of nucleotides are coordinated in their evolution. Based on our previous analysis of base triples and tetraloop hairpin loops[5,36], these larger sets of coordinated base changes are forming more complex and dynamic structures. This approach also has the potential to identify different base pairs and other structural constraints in the same region of the RNAs in organisms on different branches of the phylogenetic tree. Previously, a few examples of this from have been identified through anecdotal visual analysis of RNA sequence alignments. For example positions 1357:1365 in the *Escherichia coli* 16S rRNA covary between A:G and G:A in bacteria, and G:C and A:U pairings in Archaea and Eucarya [37,38]. In conclusion, with significant increases in the number of available RNA sequences and these new computational methods, we are confident in our ability to increase the accuracy of base pair prediction and identify new structural elements.

Central to both biological analysis and computational algorithm development is the issue of how various types of information can be used most effectively. The increasing complexity of analyzing biological data is not only caused by the exponential growth of raw data but also due to the demand of analyzing diverse types of information for a particular problem. For *in silico* explorations, this usually means manually marshalling large volumes of related information and developing a new application program for each problem or even problem instance (or at least the *ad hoc* scripting and custom parameterization and integration of existing tools). The size of the data also makes everything, from retrieving relevant information to analyzing large result sets, no longer trivial tasks. Thus biologists will benefit from an integrated data management and analysis environment that can simplify the process of biological discovery. As the growth of biological data is outpacing that of computer power, disk-based solutions are inevitable (i.e. the Moore's law doubling constant for biological data is smaller than the constant for computer chips) [6,39]. With mature disk-based data access machineries and a well designed data schema, a relational database management system (DBMS) can be an effective tool for bioinformatics.

Acknowledgments

This work is support with grants from the NIH (GM085337 and GM067317) and from Microsoft Research. The database servers are maintained by Texas Advanced Computing Center, a TeraGrid resource provider.

References

1. Cannone JJ, et al. The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics* 2002;3(1):2. [PubMed: 11869452]
2. Gutell RR, et al. Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. *Nucleic Acids Res* 1992;20(21):5785–5795. [PubMed: 1454539]
3. Gutell RR, et al. Comparative anatomy of 16-S-like ribosomal RNA. *Prog Nucleic Acid Res Mol Biol* 1985;32:155–216. [PubMed: 3911275]
4. Gutell RR, Noller HF, Woese CR. Higher order structure in ribosomal RNA. *EMBO J* 1986;5(5):1111–1113. [PubMed: 3720727]
5. Woese CR, Winker S, Gutell RR. Architecture of ribosomal RNA: constraints on the sequence of tetraloops. *Proceedings of the National Academy of Sciences of the United States of America* 1990;87(21):8467–8471. [PubMed: 2236056]
6. Benson DA, et al. GenBank. *Nucleic acids research* 2007;35(Database issue):D21–D25. [PubMed: 17202161]

7. Walker, E. A Distributed File System for a Wide-area High Performance Computing Infrastructure. Proceedings of the 3rd conference on USENIX Workshop on Real, Large Distributed Systems; Seattle: USENIX Association; 2006.
8. Macke, TJ. The Ae2 Alignment Editor. 1992.
9. Walker, E. Creating Private Network Overlays for High Performance Scientific Computing. ACM/IFIP/USENIX International Middleware Conference; ACM: Newport Beach; 2007.
10. Stephens SM, et al. Oracle Database 10g: a platform for BLAST search and Regular Expression pattern matching in life sciences. Nucl Acids Res 2005;33:675–679.
11. Eckman B. Efficient Access to BLAST using IBM DB2 Information Integrator. IBM health care & life sciences. September;2004
12. Tata, S.; Patel, JM. PiQA: an algebra for querying protein data sets. Proceedings of 15th International Conference on Scientific and Statistical Database Management; 2003. p. 141-150.
13. Miranker, DP.; Xu, W.; Mao, R. MoBioS: a Metric-Space DBMS to Support Biological Discovery. 15th International Conference on Scientific and Statistical Database Management (SSDBM 2003); Cambridge, Massachusetts, USA. Los Alamitos: IEEE Computer Society; 2003.
14. Xu W, et al. Using MoBioS' Scalable Genome Joins to Find Conserved Primer Pair Candidates Between Two Genomes. Bioinformatics 2004;20:i371–i378. [PubMed: 15262822]
15. Xu, W., et al. On integrating peptide sequence analysis and relational distance-based indexing. IEEE 6th Symposium on Bioinformatics and Bioengineering (BIBE 2006); Arlington, VA, USA. 2006. accepted
16. Sandeep, T.; James, SF.; Anand, S. Declarative Querying for Biological Sequences. Proceedings of the 22nd International Conference on Data Engineering (ICDE 2006); IEEE Computer Society; Los Alamitos. 2006.
17. rCAD. Comparative RNA Analysis Database. (manuscripts in preparation)
18. Yeang CH, et al. Detecting the coevolution of biosequences—an example of RNA interaction prediction. Molecular biology and evolution 2007;24(9):2119–2131. [PubMed: 17636042]
19. Gutell RR, Lee JC, Cannone JJ. The accuracy of ribosomal RNA comparative structure models. Curr Opin Struct Biol 2002;12(3):301–310. [PubMed: 12127448]
20. Noller HF, et al. Secondary structure model for 23S ribosomal RNA. Nucleic Acids Res 1981;9(22):6167–6189. [PubMed: 7031608]
21. Rzhetsky A. Estimating substitution rates in ribosomal RNA genes. Genetics 1995;141(2):771–783. [PubMed: 8647409]
22. Hofacker IL, et al. Automatic detection of conserved RNA structure elements in complete RNA virus genomes. Nucleic acids research 1998;26(16):3825–3836. [PubMed: 9685502]
23. Knudsen B, Hein J. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. Bioinformatics 1999;15(6):446–454. [PubMed: 10383470]
24. Rivas E, et al. Computational identification of noncoding RNAs in E. coli by comparative genomics. Current biology 2001;11(17):1369–1373. [PubMed: 11553332]
25. di Bernardo D, Down T, Hubbard T. ddbna: detection of conserved secondary structures in multiple alignments. Bioinformatics 2003;19(13):1606–1611. [PubMed: 12967955]
26. Coventry A, Kleitman DJ, Berger B. MSARI: multiple sequence alignments for statistical detection of RNA secondary structure. Proceedings of the National Academy of Sciences of the United States of America 2004;101(33):12102–12107. [PubMed: 15304649]
27. Washietl S, et al. Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome. Nature biotechnology 2005;23(11):1383–1390.
28. Pedersen JS, et al. Identification and classification of conserved RNA secondary structures in the human genome. PLoS computational biology 2006;2(4):e33. [PubMed: 16628248]
29. Dutheil J, et al. A model-based approach for detecting coevolving positions in a molecule. Molecular biology and evolution 2005;22(9):1919–1928. [PubMed: 15944445]
30. Felsenstein J. Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 1981;17(6):368–376. [PubMed: 7288891]
31. Ranwez V, Gascuel O. Quartet-Based Phylogenetic Inference: Improvements and Limits. Molecular biology and evolution 2001;18(6):1103–1116. [PubMed: 11371598]

32. Atchley WR, et al. Correlations among amino acid sites in bHLH protein domains: an information theoretic analysis. *Molecular biology and evolution* 2000;17(1):164–178. [PubMed: 10666716]
33. Tillier ER, Lui TW. Using multiple interdependency to separate functional from phylogenetic correlations in protein alignments. *Bioinformatics* 2003;19(6):750–755. [PubMed: 12691987]
34. Comparative RNA Website. <http://www.rna.cccb.utexas.edu/>
35. PHYLIP. <http://evolution.genetics.washington.edu/phyliip.html>
36. Gautheret D, Damberger SH, Gutell RR. Identification of base-triples in RNA using comparative sequence analysis. *J Mol Biol* 1995;248(1):27–43. [PubMed: 7537339]
37. Gutell RR. Collection of small subunit (16S- and 16S-like) ribosomal RNA structures. *Nucleic Acids Res* 1993;21(13):3051–3054. [PubMed: 8332526]
38. Woese CR, et al. Detailed analysis of the higher-order structure of 16S-like ribosomal ribonucleic acids. *Microbiol Rev* 1983;47(4):621–669. [PubMed: 6363901]
39. Patterson, DA.; Hennessy, JL. Computer architecture: a quantitative approach. Vol. xxviii. Morgan Kaufman Publishers; San Mateo: 1990. p. 594p. 160

Fig. 1.
Overview of evolutionary event analysis

Fig. 2.
Database tables related with event counting


```

1. Start with root
2. If a node is a leaf
3.   Compute the equality set for the node as the set of pair
   types having the maximum count among the child sequences
   at this node
4. else
5.   Compute the equality set for the node as the set of pair
   types having the maximum of (count of pairs of this
   type among child sequences at this node + count of
   pairs of this type among child equality sets).
6. For each member of equality set as candidate parent
7.   For each child sequence.
8.     If the child is different from the candidate parent,
9.       Count an evolutionary event.
10.    If the child covaries with the candidate parent,
11.      Count a covariation event.
12.   For each child equality set.
13.     If every member of the set differs from the parent,
14.       Count an evolutionary event.
15.     If every member covaries with the candidate parent
16.       Count a covariation event.
17. EventCount := minimum event count among the tested parents.
18. CovaryCount := minimum test covariation count among the
   parents.

```

Fig. 3.
Pseudo code of evolutionary event counting algorithm

Fig. 4.
Execution time breaks down

Fig. 5. Evolutionary counting performances vs. size of candidate pairs (left) and number of Taxons (right) considered in computations

Fig. 6.
Number of possible pairs at each position grouped by the minimum number of bases

Fig. 7.
Relative entropy score vs. mutual information score

Fig. 8.
Comparison of running time for filtered method and complete method

Fig. 9.
Accuracy of using percentage of covariation events for identify nucleotide interaction

Table 1

The number of pairs selected by the filters for Bacteria 16S rRNA dataset

Total possible pairs	5,234,230
Pairs selected with more than 500 minimum number of base occurrences	1,258,491
Pairs selected based on combined base occurrences and relative entropy less than 0.5	587,747