

Ontology Specific Data Mining Based on Dynamic Grammars

Daniel Quest and Hesham Ali
Department of Computer Science
College of Information Science and Technology
University of Nebraska at Omaha
Omaha, NE 68182-0116
daniel_quest@cox.net, hesham@unomaha.edu

Abstract

In this paper we introduce a new formal approach for mining biological data sets. The proposed grammar based approach provides a flexible and powerful tool for advanced sequence comparison and data mining. The approach benefits from the power of regular expressions in allowing the use of advanced queries in comparing sequences and searching for motifs or sequence attributes in biological databases. The formal grammar and the corresponding data mining engine is capable of extracting records from biological databases, filtering a subset of those records for mining, and then sorting those records based on similarity scheme designed by the user. This model is based on the objective (ontology) of the user and scoring is dynamic that is provided at runtime.

1. Introduction

A common hypothesis is that biological sequences contain elements or functional units that determine the interactions of the molecule. These elements may not be detectable by a homology search using simple alignment tools because of the interference and noise produced by mutations in the evolutionary process. However, these consensus subsequences or expressions are one key to the functionality of the sequence or to understanding the relationship between the sequence and other biological units.

Consequently, sequence alignment has been established as a dominant technique in establishing relationships between records and in searching databases for common elements. Blast [1] currently employs one successful heuristic for determining local alignment based homology between a sequence and elements in a database. Although Blast and other similar techniques have proved to be a significant tool, it has limitations (perhaps arising from applications for which not originally intended).

Some limitations include: (1) as such searching techniques are heuristics optimality is not guaranteed especially when considering statistically insignificant subsequences, (2) such tools must always consider sequence to database comparisons, in some cases we may wish to loosen or tighten alignment constraints based on expert knowledge, (3) properties of the records themselves can not be considered simultaneously to considering local alignment, and (4) regular expressions or other deterministic sequence criteria can not be used in the record extraction process.

The motivation of this work is to provide a tool that will accept either sequence or sequence model (in terms of a formal regular grammar) and provide an optimal alignment from each database element to the model. As computing power is limited, calculations should be order $O(mnd)$ where m is the length of each target in the database, n is the length of the grammar bounded by m and d is the number of elements in the database. We will also do the computations in linear space.

2. Data Mining Infrastructure

The Grammar Mining Engine has many components that are integrated together to extract records from a database (say Genbank) and evaluate the records based on static qualities found in the record and grammatical elements of sub-expressions in the sequence. The components interact as follows.

The record extraction engine is responsible for: (1) Obtaining records from a user defined location. (2) Parsing the records (sometimes eliminating records based on user criteria), (3) Stores appropriate records into the data repository.

The GUI accepts the grammar from the user, parses the grammar and dynamically constructs SQL statements based on deterministic record information and any regular expression found in the grammar as shown in figure 1.

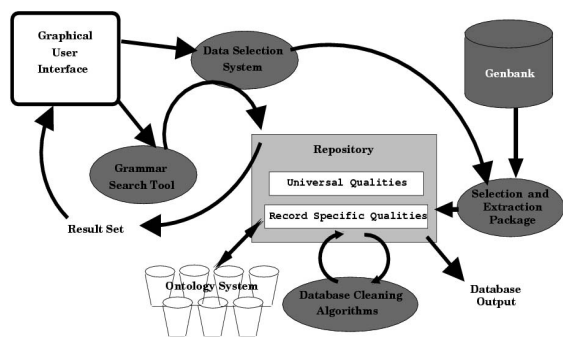


Figure 1 The Data Mining Engine

The grammar evaluation engine accepts the non-deterministic elements of the grammar and scores each sequence returned by the SQL query based on the optimum grammar-sequence path. If the score is above a threshold, the score is retained.

The ontology system records and indexes user defined relationships and allows the user to select records based off of previously established relationships. For instance, the user may define a database of proteins and then through queries discover a grammatical property of a subset of these proteins based on a binding site, the user may then label this as a separate relationship (ontology) in their database.

Database cleaning algorithms can be applied to the repository to select data no longer relevant for consideration.

3. Proposed Grammar

Given a grammar g and a target t that exists in the database, we wish to define a set of production rules and an associated cost for each production rule. The objective of the matching and scoring algorithm is to find an optimal ordered list (that is accumulate the least penalty) of grammar transformations that transform g into t . A careful selection of production rules will allow the engine to evaluate regular expressions or alignment by simply changing the scoring parameters.

Alignment algorithms contain three key production rules: insert, delete and match/mismatch. Each rule has an associated score. A scoring matrix p is defined for evaluating the score of each match or mismatch. Insert and delete have a default score to be used over the course of the algorithm. Linear gap alignment (affine gap) alternatives also call for an alternative scoring for consecutive gaps. Furthermore syntenic alignment methodologies call for alternative scoring regardless of operation but at a block penalty. The grammar may evaluate alternatives as if it were ends free or global

alignment. The grammar can also constrain such operations to a bound, and only accept operations that maintain a score bounded by a constant.

Expression	Expression Name	Function
Metacharacter Matches		
[...]	wildcard match	Match any one character
[. \ % . \ % . \ %]	character class	Match any character inside braces
[. \% . \% . \%]	percentage	Replace p with user defined match costs
[. \% . \% . \%]	character class	
Counting Modulators		
?	question	One subsequence allowed, optional
+	plus	One required, more optional
*	star	Any number allowed, but optional
{min, max}	specified range	min required, max allowed
Position Matchers		
^	caret	Matches position at start of line
\$	dollar	Matches position at end of line
Clarification and Flexibility Operators: Topology Line Operators		
(,)	parentheses	Allows for alternative subsequences
BioRegEx specific error and Gap Operators		
:	error	Verifies errors present is less than errors designated
{min,max}	sequence length	Size regulated subsequence match or variable block size
(open, extend)	constraint gap costs	Contains the affine gap costs to open and extend a gap

Table 1 The BioRegEx Grammar

Furthermore, it is sometimes desirable to score match/mismatch, insert and delete differently based on grammar specifications. In the case of match/mismatch we can simply replace p for another scoring matrix p' for all characters indicated. In the case of a specialized insert and or delete, the evaluation engine introduces another possible path with alternative scoring in addition to the conventional alignment paths.

Several alternative subsequences existing in g may also be considered by evaluating the best path over all such subsequences. The subsequences selected are chosen because they are on the best path between t and g . Alternatively, much as in regular expressions, subsequences can be repeated until the length of g is greater than or equal to the length of t .

References

- [1] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman, *Basic local alignment search tool*, Journal of Molecular Biology 215(3) (1990), 403-10.
- [2] Dan Gusfield, *Algorithms on strings trees and sequences*, τ -rst-with corrections ed., Cambridge University Press., 1999.