Approaches to Efficient Multiple Sequence Alignment and Protein Search

Thesis statements of the PhD dissertation

Adrienn Szabó

Supervisor: István Miklós

Eötvös Loránd University
Faculty of Informatics

Ph.D. School of Computer Science
Erzsébet Csuhaį-Varjú D.Sc.

Ph.D. Program: Basics and methodology of Informatics
János Demetrovics D.Sc.

Budapest, 2016.
Introduction

The science of biology offers many great opportunities to a data scientist and a software developer to take part in the challenges of making sense of the gargantuous amount of data being gathered constantly.

In my dissertation I focus on two topics: the NP-hard problem of aligning biological sequences \cite{aligning}, and searching protein families in a large database. The latter is more close to a traditional data mining task performed over biological data, while sequence alignment is a very challenging and active research subfield within bioinformatics in itself.

After brief introductions to bioinformatics and data mining of the first chapter, Chapter 2 explores previously existing solutions and techniques for aligning proteins and searching protein databases, by reviewing the related scientific literature.

Chapter 3 and Chapter 4 introduce new results in the topic of multiple sequence alignment: of the two, the first is based on \cite{RA1}, and it describes how we implemented a new variant of corner-cutting methods; while the second introduces a new representation for storing and carrying out statistical computations over a set of multiple sequence alignment paths. The related publications are \cite{ER1} and \cite{ER2}.

Chapter 5 shows how we tackled the problem of finding a needle in a haystack within a biological context: the task was to find protein families exhibiting some pre-defined complex features in a large protein database.

The main goal of the research-and-development projects behind my thesis was to discover new methodologies and techniques that could aid biologist researchers in their daily work in regards of acquiring new insights by automating the process of extracting useful information of the large amounts of data.

The developed solutions were tested on real datasets and validated in practice. All the related software codes are accessible as open source projects, and I put great emphasis on re-usability in designing the related codebases.

New Results

1. An efficient corner-cutting algorithm for multiple sequence alignment: Reticular Alignment

Usually, corner-cutting methods define a compact, mostly convex part of the dynamic programming table for narrowing down the search for the best scored multiple alignment. We introduced a new progressive alignment method called Reticular Alignment, which obtains a set of optimal and sub-optimal alignments at each step of the progressive alignment procedure. This set of alignments is
represented by a DAG network (a directed acyclic graph) of alignment columns.

The novelty of our algorithm, introduced in [RA 1] is that we do not restrict the search for solutions to a compact part of the dynamic programming table, but instead, we use a special data structure (see Fig. 1) for both representing the alignments and aligning the set of alignments against another set of alignments. The common parts of the alignments are represented only once, and aligned only once, thus saving a large amount of memory and running time.

Figure 1: A small example of a reticular alignment network: it shows three different alignments of the sequences ALLGVGQ and AVGQ

This novel corner-cutting approach allows the efficient search of the space of multiple sequence alignments for better alignments. The method has a parameter, \( t \), which affects how much of the alignment space is explored. The Reticular Alignment method can be combined with any scoring scheme, and in this way, we were able to infer the relative importance of sophisticated scoring schemes versus more exhaustive searches in the alignment space. The conclusion is that it is important to increase the search space for finding high-scored alignments.

The accuracy of the final alignments was tested on the BAliBASE database [?]. We compared our method with the most popular alignment tools: ClustalW, MAFFT, and FSA. For a comparison of alignment accuracy, see Figure 2, and for more details, refer to Figure 3.2 in Chapter 3 of the dissertation.

We found that combining sophisticated scoring schemes with the Reticular Alignment progressive alignment approach yielded a method whose accuracy is comparable to that of cutting-edge alignment methods, Clustal, MAFFT and FSA.

It should be noted that this procedure of explicitly tracing the possible alignment paths – as a means of a very strict and selective corner cutting method – can not be easily enhanced, unless some dependencies between the alignment columns would be taken into account, but that would require much more data and intelligent algorithms to be trained on it.
2. Increasing the “reticular threshold” parameter in the Reticular Alignment method does not guarantee a better alignment

The Reticular Alignment method usually could find more accurate alignments when its $t$ parameter was increased to explore more of the space of possible alignments; although we did find some interesting cases when the final MSA score decreased with a higher value of $t$, see Figure 3.

There can be be two fundamentally different reasons for why a widened search may result in worse final alignments:

- either the better scored alignments are less accurate (the wider search found better scored alignments, but these alignments happen to be differing more from the BAiLiBASE benchmark)

- or because of the local decisions at the guide tree nodes it may happen that more noise (meaning wrong sub-alignments) are propagated upwards, and a locally seemingly better alignment can be worse at the final alignment at the root. For a detailed explanation, see Fig. 4.
Alignment accuracy achieved by RetAlign for different reticular threshold values. Accuracy here is measured as the mean all-column SP score on BAliBASE v2.01 Reference sets 1–5. RetAlign was run with sequence weighting on and pairwise indel scoring.

3. A new representation for storing and carrying out statistical computations over a set of multiple sequence alignment paths

Representing a set of alignments as a DAG (directed acyclic graph) over alignment columns as nodes provides a way to quickly compute a distribution of alignment paths in the space of possible multiple sequence alignments, and other quantities of interest averaged over alignments [ER 1].

Joining a large collection of alignment paths – in our experiments, up to a couple of thousands of them – into a DAG network allows for downstream inference to be averaged over a substantive sample of alignments. Due to interchanges and crossovers at the common alignment columns, the number of alignments encoded in the network is usually a couple of orders of magnitude larger than the number of original alignment paths used to generate the DAG, so the effective sample size is greatly increased.

It is also possible to weight each alignment according to a more reliable estimate of the posterior probability, rather than analysing only a small set of individual samples.

Note, that many standard algorithms that were designed to work on single alignments – for
Figure 4: *Explaining how the internal score might decrease with the reticular threshold.* In this particular example $s_a$ is the score of the best alignment at internal node $v_a$, with $t_1$ as the threshold value. At an upper level, namely, at node $v_b$, an $x_{b,1}$-network is generated (score: $s_{b,1}$) based on the $x_{a,1}$-network. Then the best alignment at the root has score $s_{r,1}$. After increasing the value of $t$ to $t_2$, the best alignment at node $v_a$ is still the same, but the network of suboptimal alignments to consider at higher levels is the wider $x_{a,2} > x_{a,1}$ network. From this network, it is possible to find a better scored alignment at node $v_b$, with score $s_{b,2} > s_{b,1}$. If $s_{b,2} - x_{b,2} > s_{b,1}$ then the $x_{b,2}$-network might be so different from $x_{b,1}$, that they do not contain any common alignments. Therefore, the best alignment at the root obtained from the $x_{b,2}$-network might have a score $s_{r,2} < s_{r,1}$.

example, forward-backward algorithms for HMMs – can be fairly easily adapted so that they can handle alignment DAGs as well.

In [ER 1] and [ER 2], we presented a general framework for dealing with alignment uncertainty: on the one hand we explained the statistical background, while on the other hand, we tested our ideas on real data, by implementing the framework in Java as part of the *WeaveAlign* package.

### 4. A new protocol for finding re-occurring motifs in disordered regions of proteins

Suspecting a role in the development of Alzheimer's disease, we were looking for examples of “non-position specific conservation” of an amino acid pattern (HD) within a certain distance of
transmembrane (TM) domains of TM proteins.

I worked with the UniProt/SwissProt database, containing over half a million of protein sequences and their annotations.

The proposed protocol consist of the following, individually parameterizable steps, or levels:

1. Filtering transmembrane proteins
2. Cutting TM and extracellular fragments
3. Clustering protein fragments
4. Multiple sequence alignment of protein fragments and tree building for each cluster
5. Finding subtrees of the cluster-trees that include the HD pattern
6. Visualization and analysis of results

I wrote two Java packages, the first of which (ProteinSearch) contains classes to handle the SwissProt files and perform the filtering, cutting and clustering of proteins; while the second (PhyTreeSearch) gives an easy-to-use solution to the problem of searching subtrees of a large evolutionary tree that adhere to some pre-defined properties (in terms of containing an amino acid pattern in the sequences, for example).

Scripts and configuration files for running the Java programs according to the above pipeline can be found in a separate, third code repository (APP-HD-pattern-runner).

By running the pipeline, I managed to find a handful of small sets of proteins that could be interesting to look at – from this point on, a biologist needs to decide if they are worth to be further analyzed.

This project is still work-in-progress – after proving the biological relevance of the findings, we plan to submit a methodology paper to BMC Bioinformatics in 2016.

**Summary, availability of software**

Ideally, all software tools used and developed during a research process should be open software. Usually at least some new code fragments or scripts have to be written to carry out new research experiments.

Our Java software packages – runnable jar files and java sources – for multiple sequence alignment methods of Chapters 3 and 4 are openly available at the following URLs:

http://phylogeny-cafe.elte.hu/RetAlign/retalign-0.22a.zip
http://statalign.github.io/WeaveAlign/downloads.html
https://github.com/statalign/WeaveAlign

In Chapter 5, while implementing a pipeline for filtering protein fragments, I used two complementing methodologies / tools to ensure easy reproducibility of the final results:
• I shared my working environment through a Docker virtual image

• I published all the software codes (java program sources and scripts, along with example configuration files) on GitHub, as open source projects under Apache License 2.0\(^1\) and Unlicense\(^2\):

  https://github.com/ador/APP-HD-pattern-runner
  https://github.com/ador/ProteinPatternSearch
  https://github.com/ador/PhyTreeSearch

\(^1\)http://www.apache.org/licenses/LICENSE-2.0
\(^2\)http://unlicense.org
Author’s related publications

[RA 1] Adrienn Szabó, Ádám Novák, István Miklós, and Jotun Hein:

Reticular alignment: A progressive corner-cutting method for multiple sequence alignment

*BMC Bioinformatics* (IF: 3.028), vol 11, page 570, 2010

URL: [http://www.biomedcentral.com/1471-2105/11/570](http://www.biomedcentral.com/1471-2105/11/570)

DOI: 10.1186/1471-2105-11-570

[ER 1] Joseph L. Herman, Ádám Novák, Rune Lyngsø, Adrienn Szabó, István Miklós, and Jotun Hein:

Efficient representation of uncertainty in multiple sequence alignments using directed acyclic graphs

*BMC Bioinformatics* (IF: 2.67), vol 16, number 1, 2015

URL: [http://dx.doi.org/10.1186/s12859-015-0516-1](http://dx.doi.org/10.1186/s12859-015-0516-1)

DOI: 10.1186/s12859-015-0516-1

[ER 2] Joseph L. Herman, Adrienn Szabó, István Miklós, and Jotun Hein:

Approximate statistical alignment by iterative sampling of substitution matrices

*arXiv*, 2015