6338–6349 *Nucleic Acids Research, 2010, Vol. 38, No. 19* *Published online 8 June 2010*
doi:10.1093/nar/gkq526

AlexSys: a knowledge-based expert system for multiple sequence alignment construction and analysis

Mohamed Radhouene Aniba[1,2,3,4], Olivier Poch[1,2,3,4], Aron Marchler-Bauer[5] and Julie Dawn Thompson[1,2,3,4,*]

[1]Department of Structural Biology and Genomics, Institut de Génétique et de Biologie Moléculaire et Cellulaire (IGBMC), [2]Institut National de la Santé et de la Recherche Médicale (INSERM), [3]The Centre National de la Recherche Scientifique (CNRS), UMR7104, F-67400 Illkirch, [4]Université Louis Pasteur, F-67000 Strasbourg, France and [5]NCBI/NLM/NIH, 8600 Rockville Pike, Bldg. 38A, Bethesda, MD 20894, USA

Received January 29, 2010; Revised April 26, 2010; Accepted May 25, 2010

ABSTRACT

Multiple sequence alignment (MSA) is a cornerstone of modern molecular biology and represents a unique means of investigating the patterns of conservation and diversity in complex biological systems. Many different algorithms have been developed to construct MSAs, but previous studies have shown that no single aligner consistently outperforms the rest. This has led to the development of a number of 'meta-methods' that systematically run several aligners and merge the output into one single solution. Although these methods generally produce more accurate alignments, they are inefficient because all the aligners need to be run first and the choice of the best solution is made *a posteriori*. Here, we describe the development of a new expert system, AlexSys, for the multiple alignment of protein sequences. AlexSys incorporates an intelligent inference engine to automatically select an appropriate aligner a priori, depending only on the nature of the input sequences. The inference engine was trained on a large set of reference multiple alignments, using a novel machine learning approach. Applying AlexSys to a test set of 178 alignments, we show that the expert system represents a good compromise between alignment quality and running time, making it suitable for high throughput projects. AlexSys is freely available from http://alnitak.u-strasbg.fr/~aniba/alexsys.

INTRODUCTION

Comparative analyses of genetic sequences have become a cornerstone of modern genomics studies and represent a unique means of investigating the patterns of conservation and diversity in complex biological systems. Multiple sequence comparisons or alignments were originally used in evolutionary analyses to explore the phylogenetic relationships between organisms (1). More recently, new sequence database search methods have exploited multiple alignments to detect more and more distant homologues (2). Multiple sequence alignments of protein or nucleic acid sequences are also used to highlight conserved functional features and to identify major evolutionary events, such as duplications, recombinations or mutations. They have led to significant improvements in predictions of both 3D fold (3) and function (4). Of course, in the post-genomic era, it is also possible to perform comparative multiple sequence analysis at the genome level (5).

Such studies have important implications in numerous fields in biology. Nucleic acid divergence is used as a molecular clock to study organism divergence under the evolutionary forces of natural selection, genetic drift, mutation and migration, with applications from the scientific classification or taxonomy of species to genetic fingerprinting. Conserved sequence features or markers are used to characterise groups of individuals in population genetics (6). Genotype/phenotype correlations can reveal candidate genes associated with a particular trait (e.g. plant height) or inherited disease, such as schizophrenia (7). In drug discovery, a protein family perspective can identify specific structural or functional features that facilitate protein–ligand interaction studies for high-throughput virtual compound screening methods (8).

*To whom correspondence should be addressed. Tel: +33 388653200; Fax: +33 388653201; Email: julie.thompson@igbmc.fr

Thus, multiple alignments now play a fundamental role in most of the computational methods used in genomic or proteomic projects, ranging from gene identification and the functional characterisation of the gene products to genetics, human health and therapeutics.

Since the introduction of automatic methods for sequence alignment in the 1980s, a large number of studies have been performed and much progress has been achieved. The first algorithm for multiple sequence alignment (MSA; 9) was computationally expensive and consequently, most programs (known as 'aligners') in use today are based on some kind of heuristic approach that represents a compromise between reduced computation times and accurate solutions. As an example, the progressive alignment procedure, which exploits the fact that homologous sequences are evolutionarily related, consists of three main steps: (i) pairwise sequence alignment and distance matrix calculation, (ii) guide tree construction and (iii) multiple alignment following the branching order in the guide tree. The earliest aligners using the progressive approach incorporated either a global alignment method to construct an alignment of the complete sequences [e.g. ClustalW/X (10)], or a local algorithm to align only the most conserved segments of the sequences [e.g. Pima (11)].

More recently, MSA methods have evolved in response to the challenges posed by the post-genomic era, and numerous different alignment algorithms have been proposed. A comparison of many of these methods based on a widely used alignment benchmark dataset, BAliBASE (12), showed that no single algorithm was able to achieve high-quality alignments for a wide range of alignment problems and this led to the introduction of new alignment approaches, combining both global and local information in a single alignment program [e.g. DbClustal (13), TCoffee (14), MAFFT (15), Muscle (16)], or including a number of divergent algorithms, e.g. PipeAlign (17). Other approaches have also been developed that exploit other types of information to improve sequence alignments, e.g. 2D/3D structure in 3DCoffee (18) and PRALINE (19) or known domain organization in Refiner (20).

Today, next-generation sequencing technologies are further complicating the multiple alignment problem and it is now a routine task to align very large sets of sequences, containing hundreds or even thousands of sequences. The sequences, particularly those from eukaryotic organisms, often have complex domain organizations and natively disordered regions, which pose particular problems for multiple alignment programs. Furthermore, many of the alignments contain a high proportion of partial sequences, corresponding either to naturally occurring variants, or to artifacts, including sequences of proteins with a solved structure from the PDB (typically covering a single structural domain) and partially sequenced transcripts (for example from ESTs). The volume and complexity of the new data, combined with the wide variety of the available analysis tools, mean that it is often difficult for the non-specialist to choose an appropriate tool for his specific alignment problem and automatic processing by 'intelligent' computer systems is clearly required. One solution to this problem has been the development of meta-method approaches, that exploit information from multiple aligners, such as M-COFFEE (21), AQUA (22) or Mumsa (23). Although meta-methods have been shown to increase alignment accuracy, the fact that they require the computation of several alternative alignments for a single set of sequences limits their practical usage.

Here, we describe the development of a new alignment expert system, called AlexSys, whose main objective is to construct a high-quality MSA, as efficiently as possible. Specifically, the goal of the developments described here is to identify the most suitable aligners for a given alignment task, as early as possible in the alignment process. In this way, we can reduce the number of alternative alignments that need to be computed. AlexSys is designed to take advantage of the expert knowledge gained through decades of research in MSA algorithms, as well as more recent developments in the field of artificial intelligence and machine learning. The expert system exploits the advantages of the many different algorithms that have been developed over the years, by creating a model of their strengths and weaknesses, and by automatically selecting the most appropriate program, based on the input set of sequences and the intended use of the alignment.

An initial prototyping phase (24) allowed us to investigate the feasibility of such a system and we showed that a combination of different multiple alignment methods could improve the accuracy of existing MSA approaches. During this phase, we also established the suitability of UIMA (Unstructured Information Management Architecture: incubator.apache.org/uima/) for the development of expert systems in bioinformatics. UIMA is an open source Java platform that provides a general framework for the development of applications that incorporate many different types of data, including both structured and unstructured data. UIMA also supplies an execution environment in which individual computational modules can be integrated in order to build and run complex application pipelines. It has been widely used in the natural language processing field, for example to integrate and compare different text mining applications (25).

The prototype system we developed previously used *a posteriori* knowledge to determine the quality of the MSAs produced by a variety of different algorithms and selected the best, most biologically meaningful alignment. Although this resulted in more accurate alignments, it was clearly inefficient as all the algorithms needed to be run in order to choose the best one. We have now introduced a novel inference engine that uses *a priori* information about the input sequences to guide the alignment procedure automatically. Thus, given a set of input sequences, AlexSys first predicts which aligner is likely to provide the best quality alignment. This single aligner is then used to construct the MSA, resulting in a more efficient alignment construction.

The rules used in the inference engine are deduced in a separate training phase based on a machine learning algorithm and a set of training alignments. Machine learning approaches have been widely used in bioinformatics (26) to analyze large data sets, in order to discover hidden

patterns and similarities. In particular, supervised learning provides techniques to learn predictive models from observations of a system and is thus particularly well suited to deal with large-scale biological data sets. Such approaches have found successful applications in a wide range of fields, including genome annotation (27), function prediction (28) or biomarker discovery (29). Tan and Gilbert (30) compared the accuracy of a number of different supervised learning methods, such as rule-based learning systems (decision trees, one rule, decision rules), statistical learning systems (naıve Bayes, SVM and artificial neural networks) and ensemble methods (stacking, bagging and boosting). They showed that, in general, statistical methods, such as SVM or neural networks tend to perform better for problems involving multi-dimensions and continuous attributes, while rule-based systems tend to perform better in cases with discrete or categorical attributes. Rule-based or decision tree methods have the additional advantage of combining interpretability, efficiency, and, when used in ensembles of trees, excellent accuracy (31).

In the work described here, we have chosen to use a decision tree based learning algorithm. Two complementary approaches have been developed which construct different rules for selecting an alignment program. The first approach is designed to rapidly construct multiple alignments even for large datasets, while the second approach is more time-consuming but results in more accurate alignments. The accuracy of the alignments produced by AlexSys is evaluated using the BAliBASE 12) and OXBench (32) benchmarks, and compared to six of the most widely used existing aligners.

MATERIALS AND METHODS

Training and test sets

The training and test sets were derived from the reference alignments in the BAliBASE and OXBench benchmarks. We used the 218 alignments in ref. (1–5), corresponding to (i) equidistant sequences with various levels of conservation, (ii) families aligned with a highly divergent 'orphan' sequence, (iii) subgroups with <25% residue identity between groups, (iv) sequences with N/C-terminal extensions and (v) internal insertions. These 218 alignments contain a total of 6222 protein sequences, including both full-length sequences and fragmentary sequences from the PDB database. In addition to the alignments from BAliBASE, we used the set of 672 extended alignments from OXBench, containing a total of 66 742 protein sequences. These alignments contain sequences corresponding to isolated structural domains. The combined data set was then divided into a training set of 712 alignments (80% of the alignments were selected at random) used to create the rules in the inference engine and a test set of 178 alignments (the remaining 20%) used for evaluation purposes.

To assess the performance of the aligners used in this study, we used the sum-of-pairs score (SP) (22) to compare the alignments produced by the aligner with the reference alignments. The SP score corresponds to the proportion of pairs of residues aligned the same in both alignments.

Running times for all programs were calculated on a Sun Enterprise server, with 8 Quad-Core AMD Opteron processors and 32 Gb of memory.

Selection of multiple alignment programs

Six of the most widely used aligners have been integrated in AlexSys, namely ClustalW, Dialign, Mafft, Muscle, Kalign and ProbCons. The algorithms implemented in each of the programs are described briefly below.

'ClustalW' (version 2.0) performs a traditional progressive alignment, by first comparing all pairs of sequences, then building a guide tree using the neighbour joining approach, and finally aligning all the sequences according to the branch order in the guide tree. For sequences that are globally related, ClustalW often provides accurate alignments, while in more complex cases it can be used as a good starting point for further refinement.

'Dialign' (34) (version 2.2.1) constructs multiple alignments by comparing segments of the sequences, rather than single residues. The main difference between Dialign and the other alignment approaches is the underlying scoring scheme or objective function. Instead of summing up substitution scores for aligned residues and subtracting gap penalties, the score of an alignment is based on *P*-values of local sequence similarities. Only those parts of the sequences are aligned that share some statistically significant similarity, unrelated parts of the sequences remain unaligned. This approach is particularly successful in situations where sequences share only local homologies.

'Mafft' (version 6.240) (MSA based on Fast Fourier Transform; option FFT-NS-i) is a fast aligner that builds an initial progressive alignment using an approximate measure based on shared 6-tuples to estimate the distance between pairs of sequences. A guide tree is then generated using the UPGMA algorithm with modified linkage and sequences are aligned following the branch order of the tree. The initial MSA is then improved by recalculating the distance matrix and repeating the progressive alignment steps. The final phase involves an iterative refinement to optimise a weighted sum of pairs (WSP) (35) score, using a group-to-group alignment and a tree-dependent restricted partitioning technique.

'Muscle' (version 3.7) (multiple sequence comparison by log-expectation) uses a three phase approach similar to the one implemented in Mafft. In the initial alignment phase, a k-mer distance is used to estimate the pairwise distances and the guide tree is built using the UPGMA algorithm. The initial MSA is then improved by calculating a more accurate Kimura distance (36) for aligned pairs, again repeating the progressive alignment steps. The final iterative refinement stage employs a variant of the tree dependent restricted partitioning algorithm.

'Kalign' (37; version 2.03) also uses a progressive alignment approach, the main difference being that it employs the Wu–Manber (38) approximate string matching algorithm when calculating the distances among sequences. This methodology allows for a fast, yet accurate distance

estimation. As in Mafft and Muscle, the UPGMA algorithm is used to build the guide tree. In addition, the program performs a consistency check in order to define the largest set of sequence matches that can be inserted in the alignment, using a modified version of the Needleman–Wunsch algorithm (39) to find the most consistent path through the dynamic programming matrix.

'ProbCons' (40; version 1.12) (Probabilistic Consistency-based MSA) incorporates a pair-hidden Markov model-based progressive alignment algorithm. The alignment procedure is divided into four steps, starting with a computation of posterior-probability matrices for every pair of sequences, followed by a dynamic programming calculation of the expected accuracy of every pairwise alignment. A probabilistic consistency transformation is then used to re-estimate the match accuracy scores. A guide tree is calculated with hierarchical clustering and the sequences are aligned using a progressive approach. In a post-processing phase, random bi-partitions of the generated alignment are realigned in order to check for better alignment regions.

AlexSys global architecture

The AlexSys expert system (Figure 1) is designed around a central core, containing the main data processing and alignment construction components. The information that drives the decisions made within the system is stored in a separate metadata layer. In addition, a knowledge base contains the necessary background information for the selection of the most appropriate aligner(s), based on the input data.

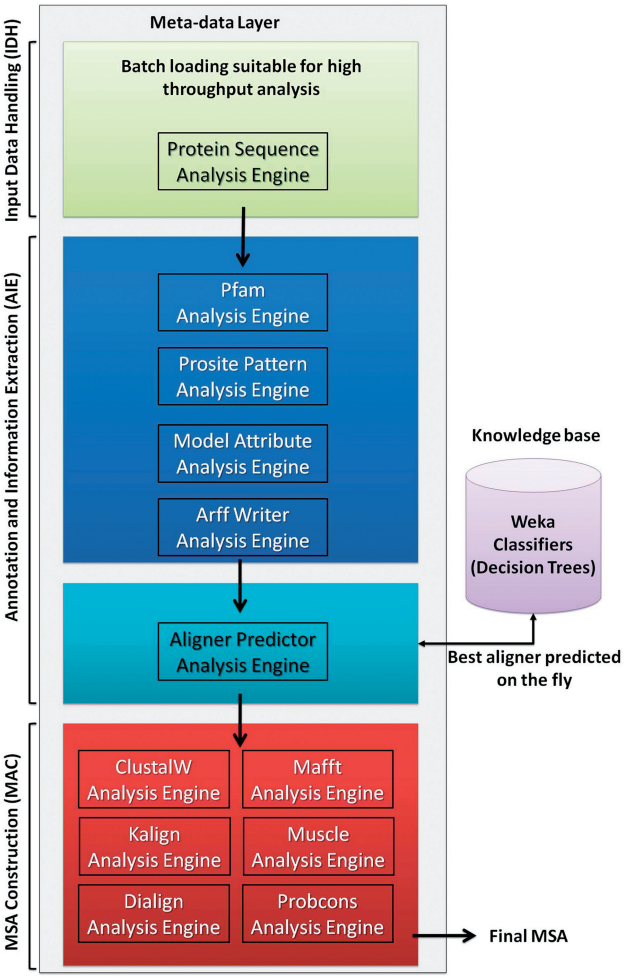

Figure 1. AlexSys global architecture. The core is divided into three main parts: IDH, AIE and MAC. Each part contains one or more Analysis Engines. A special Analysis Engine, the Aligner Predictor Analysis Engine, represents the intelligent inference engine for the whole system.

Knowledge base construction

The knowledge base in AlexSys contains 'alignment models' (Figure 2) that are used to predict the strengths and weaknesses of the individual aligners, given a specific set of input sequences. To achieve this, a supervised learning or classification algorithm is used. The alignment models are trained on a set of instances, corresponding to the multiple alignment sets in the training data described above, for which the performance of the aligners is already known.

Three problems needed to be addressed at this stage: (i) the characteristics used to describe the input sequences (the attributes), (ii) the structure of the predicted output classes, corresponding to the relative performance of the aligners and (iii) the learning algorithm used to predict the class for a given instance of input sequences.

The first problem concerns the selection of pertinent features or 'attributes' that adequately describe the input sequences. Based on our previous knowledge acquired working on multiple sequence alignments, we identified the following attributes:

- number of sequences in the dataset,
- average sequence length,
- average pairwise residue percent identity,
- number of sequences with known 3D structure, according to the PDB database (41)
- average number of residues found in α-helices per sequence,
- average number of residues found in β-strands per sequence,
- average number of functional domains per sequence, according to the Pfam database (42)
- number of sequences with low complexity regions,
- average number of regions with low complexity per sequence,
- average hydrophobicity of the sequences,
- average number of predicted transmembrane segments per sequence,
- average amino acid composition based on the six groups: [PAGST], [DEQN], [KRH], [LIVM], [FWY] and [C].

These attributes are then used to establish potential relationships between the input sequences and the performance of the individual aligners.

The second problem concerns the definition of the desired output classes. Given a set of input sequences,

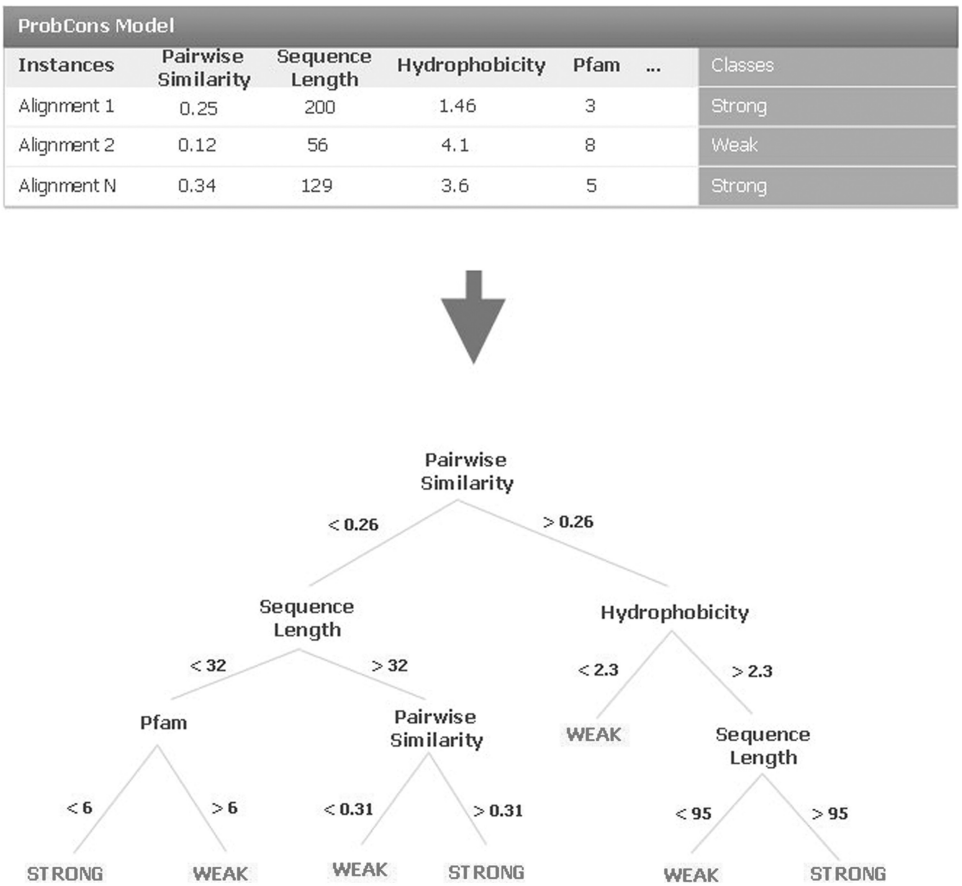

Figure 2. Alignment model creation. For each aligner, a decision tree (shown at the bottom) is generated from a set of training data (shown at the top). Each instance in the training set, corresponding to a set of sequences to be aligned, is associated with a vector of attributes and the desired classification. In the decision tree, numbers on the branches indicate the threshold value of the attribute that is used to select either the left or right branch.

the quality of the alignment produced by each aligner is measured using the SP score. We then define a binary classification, where an aligner is classified as Strong (if the SP score is >0.5) or Weak (if the SP score is <0.5).

The third problem concerns the choice of an appropriate supervised learning or classification algorithm. Here, we use a decision tree approach, where the leaves of the tree represent the output classifications, each node corresponds to a specific attribute and the branches correspond to a range of attribute values. We tested three widely used decision tree algorithms implemented in the Weka software (sourceforge.net/projects/weka/):

- The C4.5 (43) (known in Weka as J48) algorithm generates a classification or decision tree by recursive partitioning of the dataset. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits the samples into subsets enriched in one class or the other, based on a normalized information gain score.
- The Random Tree algorithm (44) is a fast decision tree learner that constructs a tree from random permutations. With k random features at each node, a tree is drawn at random from a set of possible trees and again, information gain is used as a selection criterion.
- The Random Forest algorithm (45) combines an ensemble classifier and the random selection of features, in order to construct a collection of decision trees with controlled variation. Each tree defines a classification, and is said to 'vote' for that classification. The forest algorithm then chooses the classification having the most votes (over all the trees in the forest).

Metadata layer

The metadata layer in AlexSys contains the 'on-the-fly' data that will drive the multiple alignment construction process. It consists of a set of UIMA 'type systems', which are the equivalent of structures or objects in a traditional programming language. There are currently six type systems (TSs) designed to represent the major data structures used:

- the Protein Sequence TS contains the basic sequence information;

- the Model Attributes TS stores the attribute data associated with the alignment models;
- the Pfam TS and Prosite Pattern TS contain information about function domains mapped on the sequences, obtained from the InterPro database (46);
- the Arff Analyzed TS contains information about the sequence attributes in Arff (Attribute-Relation File Format);
- the Aligner Predictor TS collects information concerning the aligners, such as the predicted strengths and weaknesses for each aligner and the final choice of an aligner.

AlexSys Core

The central core of AlexSys was designed and built using the UIMA development toolkit. UIMA provides an ideal framework for the construction of applications that integrate different applications and heterogeneous data types. UIMA applications can easily be decomposed into modules or components, that handle different aspects of complex process, such as data input, quality control, data analysis, results output and visualization. UIMA then provides the facilities required to manage these components and the data flow between them. Components can be written in Java or C++ and the data that flows between components is designed for efficient mapping between these languages.

The primitive processing units in UIMA applications, called Analysis Engines (AEs), can be combined in order to analyze data containing structured or unstructured information. The AE's core is called an Annotator and contains the actual analysis software. The AEs can then be organized using Flow Controllers (FC) inside more complex structures called Aggregate Analysis Engines (AAEs). The AEs share data via the TSs in the metadata layer. Figure 4 shows the overall architecture of the central core, which is divided into three main AAE, described in detail below.

Input data handling

When a set of sequences is input to AlexSys, they are transferred to the metadata layer (Protein Sequence TS), using the Protein Sequence AE. This AE uses the framework of the Biojava sequence input/output API (47) to provide access to sequences from a number of common file formats such as FASTA, GenBank and EMBL. Thus, regardless of the input format used, sequences can be simply transformed into UIMA TSs, making them easily available to the other analysis engines.

Annotation and information extraction

This AAE contains a number of AE that are used to obtain pertinent information associated with the set of input sequences. When new data is stored in the Protein Sequence TS, the Model Attributes AE calculates the sequence attributes required for the selection of an appropriate aligner and stores the information in the Model Attributes TS. The attributes are also read by the Arff Writer AE, which transforms them into a special format called Arff (Attribute-Relation File Format) used by the Aligner Predictor AE to select one or more appropriate aligners for the input sequences.

In addition to the attributes that can be calculated directly from the sequence data, two AEs have been defined that extract additional information from external databases. The Pfam AE uses the WSInterProScan (48) web service (www.ebi.ac.uk/Tools/webservices/services/interproscan) to retrieve the associated Pfam domains from the InterPro database and maps them to the sequences. The additional information generated is then stored in the Pfam TS. In a similar way, the Prosite Pattern AE maps patterns from the Prosite database (49) to the input sequences.

Multiple alignment construction

The first task in the multiple alignment process is the selection of an appropriate aligner to use. This is performed by the Aligner Predictor AE, which represents the AlexSys inference engine. Based on the attributes associated with the input sequences, the inference engine uses the alignment models in the knowledge base to predict the class (Strong or Weak) of each aligner. Two alternative methods have been developed to make the final selection of the most suitable aligner.

- The first method is based on the probability scores (provided by the Weka software. For each of the five aligners, the probability associated with a Strong prediction is obtained and the aligner with the highest probability is then selected.
- The second method builds a set of IF-THEN rules. Each of the five aligners incorporated in AlexSys is classified as either Strong or Weak. In the case where more than one aligner is classified as Strong, we select the one that requires the least CPU time.

Once an aligner has been selected, a UIMA Flow Controller is used to call the appropriate alignment AE. These AE encapsulate the actual alignment program, accessible via JNI (Java Native Interface). AlexSys requires that these programs are already installed on the user's platform.

Availability

The source code and help for the complete AlexSys system are available from http://alnitak.u-strasbg.fr/~aniba/alexsys.

RESULTS AND DISCUSSION

In this article, we describe the development of an expert system, AlexSys, for the construction of MSAs. The MSA field is a highly active one and numerous alignment methods have been developed, based on a wide variety of different algorithms. Unfortunately, there is no single algorithm that works best on all problems (33), due to the high complexity of today's sequence alignment tasks. AlexSys is therefore designed to combine the power of the existing approaches in a single system which is both efficient and easy to use for the biologist.

One of our main objectives in developing AlexSys was to improve the efficiency of the alignment construction, by selecting the most suitable aligners as early as possible in the alignment process. In this way, we avoid running aligners that are unlikely to provide useful information. To achieve this, we have introduced an 'intelligent' inference engine that predicts a priori the performance of the different aligners. Based only on specific attributes of the set of sequences to be aligned, we then choose the most suitable aligner that is most likely to produce a high quality alignment.

Supervised learning

The rules used in the inference engine were trained using a supervised learning approach. We tested a number of different approaches based on decision tree algorithms, implemented in the Weka machine learning software. Weka is freely available and is written in Java, which means that it can be easily integrated in the UIMA environment. It provides easy access to a wide range of state-of-the-art machine learning algorithms and is supported by a large developer community.

Supervised learning, or classification, techniques are used in a wide range of applications in bioinformatics. However, as far as we are aware, this is the first direct application of such algorithms to try to solve the multiple sequence alignment problem. A wide range of learning algorithms are available, including neural networks, support vector machines and decision trees, and their performance largely depends on the data to be classified and the model used to represent them. We decided to base our studies on the decision tree algorithms since they provide predictions based on simple rules that are comprehensible to both humans and computers. Regardless of the learning algorithm used, two factors are known to play an important role in determining the accuracy of the resulting classifications: (i) the characteristics used to describe the input data and (ii) the form of the classes to be learned.

(i) The sequences input to AlexSys are characterized by a set of attributes that were initially selected based on our previous experience of constructing multiple alignments and a number of previous studies to evaluate the performance of aligners (50–52). The attributes were then refined in a number of preliminary experiments (data not shown) to determine an adequate set for use in this work. As might be expected, the average pairwise residue percent identity is a crucial factor that is found in the decision trees of all the aligners and confirms previous observations that the similarity of the sequences significantly affects alignment quality (33). The average sequence length is another important attribute for some of the aligners, including ClustalW, Dialign and Muscle. A related factor is the average number of Pfam domains per sequence, where high values indicate the presence of multi-domain proteins. This attribute is found in the decision trees for ClustalW and Dialign, which might be explained by the fact that these programs are exclusively based on either a global or local algorithm. The other aligners include both local and global information. More surprisingly, the residue composition of the sequences also affects the accuracy of all the aligners. In the case of ClustalW, Mafft, Muscle and Probcons the amino acid group [KRH] is determinant, while for Dialign and Kalign the most important group is [PAGST]. Nevertheless, it should be noted that this attribute set is clearly not definitive and work is still on-going to investigate other attributes and to evaluate their usefulness.

(ii) The second issue proved to be more problematic. Our initial design of the learning process involved a single model, where the class of an instance in the training data was defined to be the best aligner for this set of sequences. We then evaluated the performance of various learning algorithms, but the resulting prediction accuracy was low, due to the relatively small number of instances in the training set, the high dimensionality of the data and the difficulties associated with unambiguously selecting the best aligner among several high scoring ones. As a consequence, we redesigned the problem as a binary classification, with a separate model for each aligner, where the class of an instance corresponds to the strength of the aligner, defined as either strong or weak.

Evaluation of decision tree algorithms

We compared the predictive performance of three different decision tree algorithms, namely Random Tree, Random Forest and J48 with default parameters. Table 1 shows the accuracy of each method, estimated using 10-fold

Table 1. Correctly and incorrectly classified instances for each aligner

	ClustalW		Dialign		Mafft		Muscle		Kalign		ProbCons		Average (%)	
	CCI	ICI	CCI	ICI	CCI	ICI	CCI	ICI	CCI	ICI	CCI	ICI	ACCI	AICI
J48	812	74	825	61	**838**	**48**	**842**	**44**	822	64	845	41	93.8%	6.2%
RandomTree	806	80	810	76	816	78	822	64	805	81	839	47	92.1%	7.9%
RandomForest	**825**	**61**	**828**	**58**	835	51	839	47	**823**	**63**	**846**	**40**	**94**	**6**

CCI, correctly classified instances; ICI, incorrectly classified instances; ACCI, average CCI; AICI, average ICI. Numbers shown in bold indicate the best scores for each aligner.

cross-validation, which reduces the problems of overfitting. Cross-validation is one of several approaches that can be used to estimate how well the model will perform on future as-yet-unseen data. The Random Forest algorithm is the most accurate predictor for all aligners, except Mafft and Muscle where the Random Tree method performs slightly better. With an average correct classification rate of 94%, this algorithm seems to be the most appropriate for our purposes. Nevertheless, Random Tree and J48 also performed well, with an average correct classification rate of around 92% and 93.2%, respectively.

A more detailed study of the performance of the Random Forest algorithm is shown in Figure 3. The results confirm that the classification is highly accurate for all five aligners used here. The true positive (TP) rates range from 0.97 to 0.99 for high scoring multiple alignments (class = Strong), whereas for low scoring alignments (class = Weak) the TP rates range from 0.72 to 0.87. The *F*-measure, defined as:

$$F-\text{measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall+precision}} = \frac{2TP}{2TP+FN+FP}$$

is a widely used score in the information retrieval and natural language processing communities and combines measures of precision (also called positive predictive value = TP/TP+FP) and recall (also called sensitivity = TP/TP+FN). The *F*-measure score ranges from 0.0 to 1.0, with 0.0 indicating the poorest result and 1.0 a perfect retrieval. In these tests, the *F*-measures for the Random Forest algorithm range from 0.96 to 0.98 for Strong class alignments and from 0.77 to 0.90 for Weak class alignments.

Based on these results, we concluded that the Random Forest approach was the most appropriate for our purposes. This was then used to build the inference engine used by the AlexSys to select the most appropriate aligner for a given set of sequences.

Choice of aligners

There are now hundreds of different programs available for the construction of multiple sequence alignments and it is clearly impossible to incorporate all of these in AlexSys. We therefore selected a small number of aligners, representing different alignment approaches. ClustalW is a global alignment method, while Dialign uses a local alignment algorithm. Mafft and Muscle were developed more recently and use both local and global information to construct the alignment. Kalign and Mafft are very fast aligners, while ProbCons is less efficient but often produces a higher quality final alignment.

Based on the Random Forest machine learning algorithm and the sequence attributes described above, the inference engine in AlexSys predicts which of these six aligners should be used to align a given set of sequences. Using a test set of 178 reference alignments, we compared AlexSys' prediction of the best aligner to the 'ideal' aligner, which achieves the highest score. In 80 (45%) of

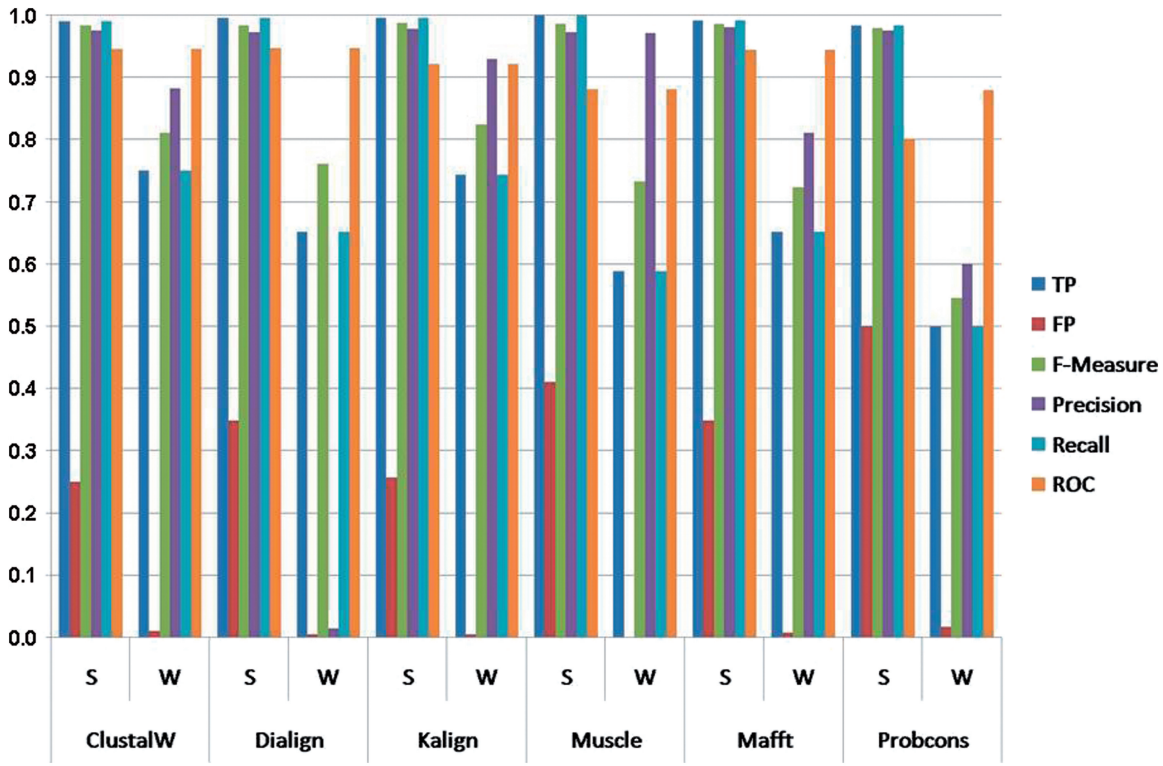

Figure 3. Evaluation of the Random Forest algorithm for the classification of aligner performance as S, strong or W, weak. For each aligner, the TP (true positive rate, proportion of correctly classified instances); FP (false positive rate, proportion of wrongly classified instances); Precision (=TP/TP+FP); Recall (=TP/TP+FN); F-measure (combines recall and precision scores into a single measure of performance) and ROC area (or the area under the receiver operating characteristic curve, the probability that when we randomly pick one positive and one negative example, the classifier will assign a higher score to the positive example than to the negative) are indicated.

the alignment tests, AlexSys accurately predicted the highest scoring program. In another 81 (45.5%) of the tests, the prediction made by AlexSys corresponded to the second highest scoring program. In general, when AlexSys did not choose the best aligner, the difference in the scores obtained by the different aligners was generally small. Thus, the total root mean square error (RMSE) of the SP scores obtained by AlexSys and the 'ideal' aligner is 0.006, where the values of the SP score can range from 0 to 1.

Thanks to the modular design of AlexSys, it is easy to incorporate other aligners and we will evaluate the use of more specialized algorithms, such as POA (53) or PRANK (54), in the future.

AlexSys multiple alignment performance

The efficiency and accuracy of the multiple alignment construction process in AlexSys were evaluated using a test set of 178 multiple alignments (see 'Materials and Methods' section). Alignment accuracy was estimated by comparing the results obtained with AlexSys to the reference alignments in both BAliBASE and OXBench benchmarks. Two alternative approaches, using probability- and rule-based methods, for selecting the most suitable aligner in the AlexSys inference engine were tested here. The probability-based inference engine results in higher accuracy, with an average score of 0.891, compared to a score of 0.888 obtained by the rule-based system. The difference in alignment accuracy can be explained by the background knowledge built into the rules, which favours a shorter running time when more than one aligner is predicted to give a strong performance. In contrast, the probability-based implementation systematically selects the aligner with the highest probability of a strong performance. The performance of these alternative methods was also compared to the five existing aligners run independently (Figure 4). In terms of alignment accuracy, both methods implemented in AlexSys (probability- and rule-based) achieved higher scores than five of the independent aligners. The differences between AlexSys (probability) and ClustalW, Dialign, Kalign, Muscle are statistically significant with P-values of 3.783×10^{-7}, 4×10^{-2}, 3.13×10^{-5}, 7.1×10^{-3}, respectively, based on the non parametric Wilcoxon signed rank test. The only non significant comparison concerns Mafft with a P-value of 0.552. The only aligner that scores higher than the probability-based AlexSys is ProbCons, with an average SP score of 0.903 and the difference is statistically significant ($P = 3.15 \times 10^{-6}$). However, AlexSys only requires 180 minutes to align all 178 test alignments, while ProbCons takes almost 480 min. AlexSys thus represents a good compromise between alignment quality and the computational time needed to produce the alignments.

A more detailed comparison of the quality of the alignments produced by AlexSys and the other aligners was also performed (Figure 5). The distributions of the alignment scores obtained for the 178 test alignments shows that AlexSys (probability-based) generally results in less low scoring alignments than the other aligners, with the

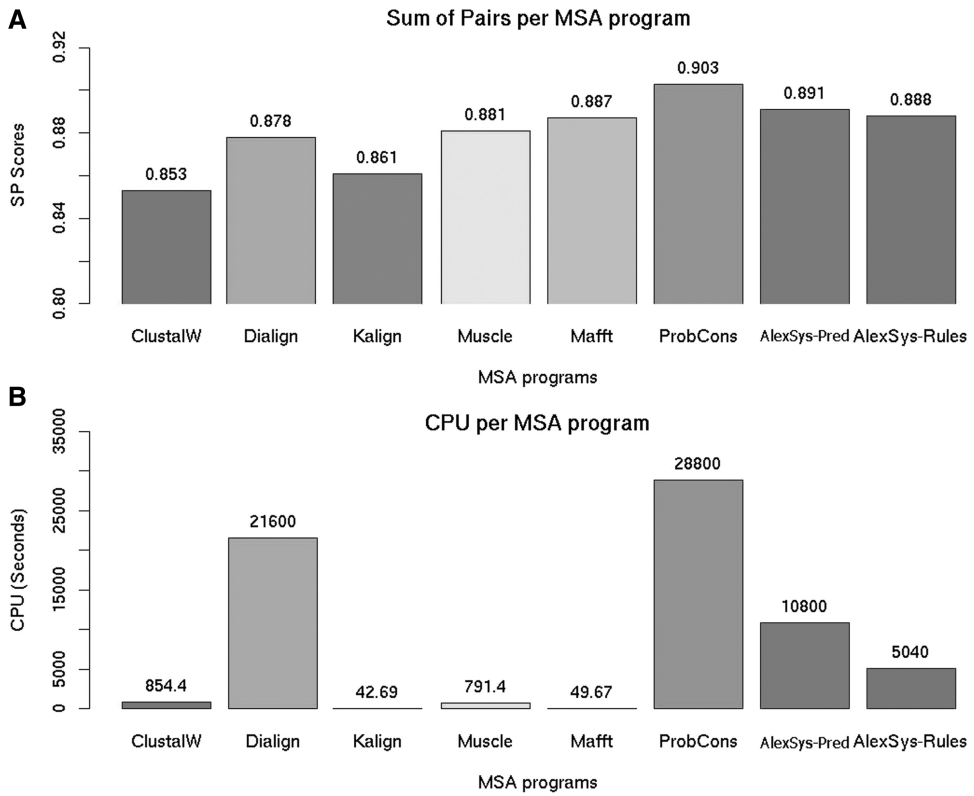

Figure 4. Evaluation of alignment accuracy and efficiency for AlexSys and the six existing aligners. (**A**) Average alignment accuracy for a test set of 178 multiple alignments, measured using the SP score. (**B**) The total CPU time required to construct the 178 multiple alignments.

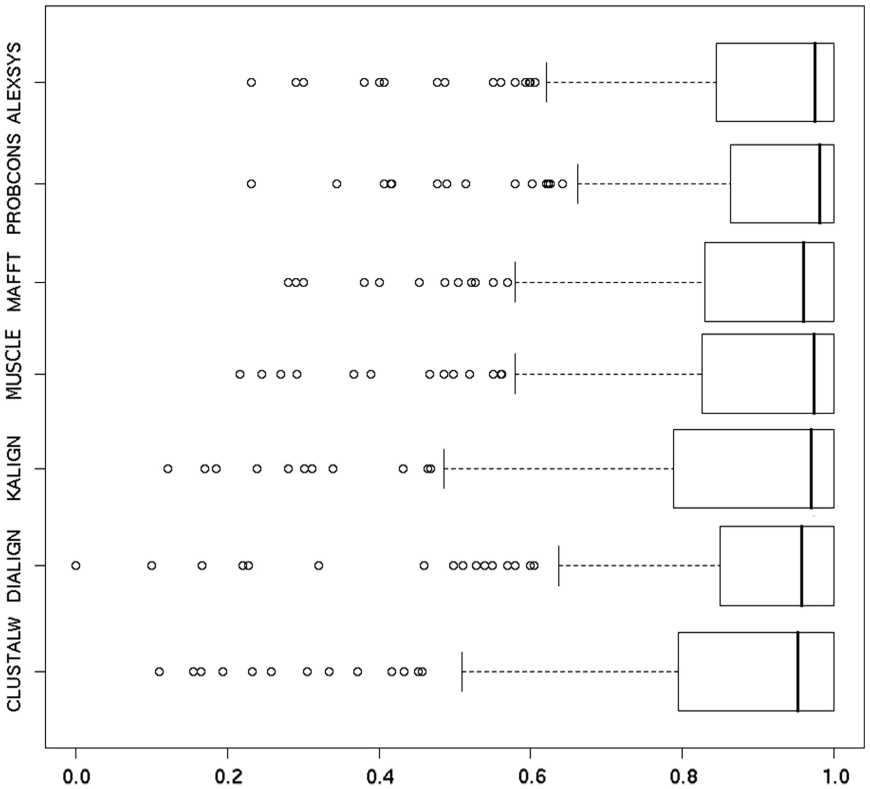

Figure 5. Distribution of alignment accuracy scores for AlexSys (probability based) and the six existing aligners. The grey boxes represent the lower and upper quartiles of the SP scores obtained by each aligner for a test set of 178 multiple alignments. Black lines in the boxes indicate the median score. Black circles represent outliers, corresponding to low scoring alignments.

exception of Mafft and ProbCons. Furthermore, the median score for AlexSys is higher than for all other aligners except ProbCons. Taken together, these results demonstrate that our intelligent platform is able to produce more reliable alignments within a reasonable time scale, which makes it suitable for high-throughput applications.

Nevertheless, as shown in Figure 6, some test cases were not well aligned by any of the aligners currently incorporated in AlexSys. For example, for the test alignments 8, 91, 139 or 159, none of the programs tested achieved an SP score higher than 0.3. In the future, these difficult cases will be identified automatically by the expert system and a warning can be produced to indicate that the resulting alignment may not be of very high quality. In these cases, where the sequences are highly divergent, additional information will be needed in order to build biologically meaningful alignments.

CONCLUSIONS

We have shown that the 'intelligent' inference engine in AlexSys can be used to select a priori an appropriate aligner for a given alignment problem. Reliable alignments can then be produced in a time scale suitable for high-throughput projects. The architecture used to build the expert system is highly modular and flexible, allowing AlexSys to evolve as new alignment algorithms are made available. In the future, we plan to extend the inference engine to identify multiple algorithms that could potentially provide complementary information about the input sequences. For example, well aligned regions from different aligners will be identified and combined into a single consensus alignment. Additional information such as structural and functional data will also be exploited to improve the final alignment accuracy. Finally, a crucial aspect of any bioinformatics tool is its accessibility and usability. Therefore, we are currently developing a web server, and a web services based distributed system. We will also design a novel visualization module that will provide an intuitive, user-friendly interface to all the information retrieved and constructed by AlexSys.

ACKNOWLEDGEMENTS

We would like to thank Luc Moulinier for many fruitful discussions, Nicolas Lachiche, Pierre Gancarski for timely advice on machine learning and the members of the Strasbourg Bioinformatics Platform (BIPS) for their support. We also thank the UIMA and Weka communities for their ongoing help and support.

FUNDING

Institute funds from the Centre National de la Recherche Scientifique (CNRS); Institut National de la Santé et de la

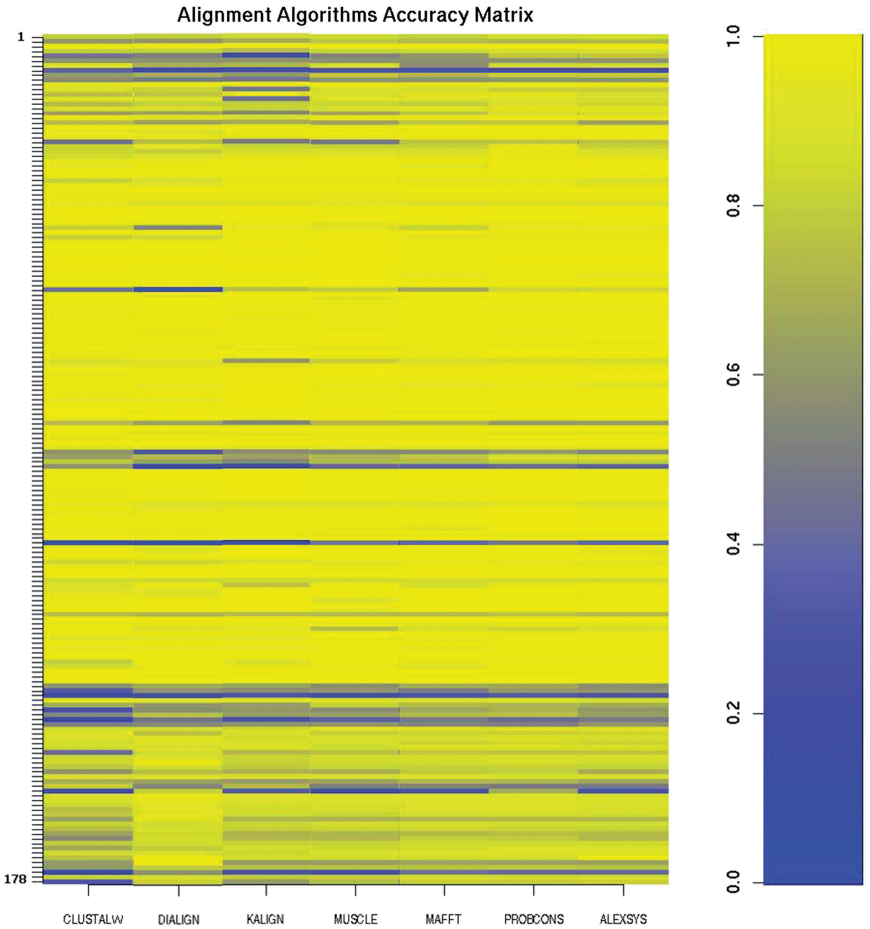

Figure 6. Alignment accuracy matrix. For each alignment in the test set, the SP scores obtained by the different alignment programs are indicated using a colour ranging from blue (low score) to yellow (high score). Rows that are predominantly blue highlight the alignments that could not be aligned accurately by any of the aligners tested.

Recherche Médicale (INSERM); Université de Strasbourg. Funding for open access charge: CNRS (Centre National de la Recherche Scientifique).

Conflict of interest statement. None declared.

REFERENCES

1. Phillips,A., Janies,D. and Wheeler,W. (2000) Multiple sequence alignment in phylogenetic analysis. *Mol. Phylogenet. Evol.*, **16**, 317–330.
2. Yu,Y.K., Gertz,E.M., Agarwala,R., Schäffer,A.A. and Altschul,S.F. (2006) Retrieval accuracy, statistical significance and compositional similarity in protein sequence database searches. *Nucleic Acids Res.*, **34**, 5966–5973.
3. Moult,J.A. (2005) Decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Curr. Opin. Struct. Biol.*, **15**, 285–289.
4. Watson,J.D., Laskowski,R.A. and Thornton,J.M. (2005) Predicting protein function from sequence and structural data. *Curr. Opin. Struct. Biol.*, **15**, 275–284.
5. Margulies,E.H., Chen,C.W. and Green,E.D. (2006) Differences between pair-wise and multi-sequence alignment methods affect vertebrate genome comparisons. *Trends Genet.*, **22**, 187–193.
6. Kidd,K.K., Pakstis,A.J., Speed,W.C. and Kidd,J.R. (2004) Understanding human DNA sequence variation. *J. Hered.*, **95**, 406–420.
7. Owen,M.J., Craddock,N. and O'Donovan,M.C. (2005) Schizophrenia: genes at last? *Trends Genet.*, **21**, 518–525.
8. Lenz,G.R., Nash,H.M. and Jindal,S. (2000) Chemical ligands, genomics and drug discovery. *Drug Discov. Today*, **5**, 145–156.
9. Myers,E.W. and Miller,W. (1988) Optimal alignments in linear space. *Comput. Appl. Biosci.*, **4**, 11–17.
10. Larkin,M.A., Blackshields,G., Brown,N.P., Chenna,R., McGettigan,P.A., McWilliam,H., Valentin,F., Wallace,I.M., Wilm,A., Lopez,R. *et al.* (2007) Clustal W and Clustal X version 2.0. *Bioinformatics*, **23**, 2947–2948.
11. Smith,R.F. and Smith,T.F. (1992) Pattern-induced multi-sequence alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modelling. *Protein Eng.*, **5**, 35–41.
12. Thompson,J.D., Koehl,P., Ripp,R. and Poch,O. (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins*, **61**, 127–136.
13. Thompson,J.D., Plewniak,F., Thierry,J.C. and Poch,O. (2000) Rapid and reliable global multiple alignments of protein sequences detected by database searches. *Nucleic Acid Res.*, **28**, 2919–2926.
14. Notredame,C., Higgins,D.G. and Heringa,J. (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.
15. Katoh,K., Misawa,K., Kuma,K. and Miyata,T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.*, **30**, 3059–3066.
16. Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, **32**, 1792–1797.

17. Plewniak,F., Bianchetti,L., Brelivet,Y., Carles,A., Chalmel,F., Lecompte,O., Mochel,T., Moulinier,L., Muller,A., Muller,J. *et al.* (2003) PipeAlign: a new toolkit for protein family analysis. *Nucleic Acids Res.*, **31**, 3829–3832.
18. O'Sullivan,O., Suhre,K., Abergel,C., Higgins,D.G. and Notredame,C. (2004) 3DCoffee: combining protein sequences and structures within multiple sequence alignments. *J. Mol. Biol.*, **340**, 385–395.
19. Simossis,V.A. and Heringa,J. (2005) PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. *Nucleic Acids Res.*, **33**, W289–W294.
20. Chakrabarti,S., Lanczycki,C.J., Panchenko,A.R., Przytycka,T.M., Thiessen,P.A. and Bryant,S.H. (2006) Refining multiple sequence alignments with conserved core regions. *Nucleic Acids Res.*, **34**, 2598–2606.
21. Wallace,I.M., O'Sullivan,O., Higgins,D.G. and Notredame,C. (2006) M-Coffee: combining multiple sequence alignment methods with T-Coffee. *Nucleic Acids Res.*, **34**, 1692–1699.
22. Muller,J., Creevey,C.J., Thompson,J.D., Arendt,D. and Bork,P. (2010) AQUA: automated quality improvement for multiple sequence alignments. *Bioinformatics*, **26**, 263–265.
23. Lassmann,T. and Sonnhammer,E.L. (2006) Kalign, Kalignvu and Mumsa: web servers for multiple sequence alignment. *Nucleic Acids Res.*, **34**, W596–W599.
24. Aniba,M.R., Siguenza,S., Friedrich,A., Plewniak,F., Poch,O., Marchler-Bauer,A. and Thompson,J.D. (2009) Knowledge-based expert systems and a proof-of-concept case study for multiple sequence alignment construction and analysis. *Brief Bioinform.*, **10**, 11–23.
25. Kano,Y., Baumgartner,W.A. Jr, McCrohon,L., Ananiadou,S., Cohen,K.B., Hunter,L. and Tsujii,J. (2009) U-Compare: share and compare text mining tools with UIMA. *Bioinformatics*, **25**, 1997–1998.
26. Inza,I., Calvo,B., Armañanzas,R., Bengoetxea,E., Larrañaga,P. and Lozano,J.A. (2010) Machine learning: an indispensable tool in bioinformatics. *Methods Mol Biol.*, **593**, 25–48.
27. Hoff,K.J., Tech,M., Lingner,T., Daniel,R., Morgenstern,B. and Meinicke,P. (2008) Gene prediction in metagenomic fragments: a large scale machine learning approach. *BMC Bioinformatics*, **9**, 217.
28. Schietgat,L., Vens,C., Struyf,J., Blockeel,H., Kocev,D. and Dzeroski,S. (2010) Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics*, **11**, 2.
29. Azuaje,F., Devaux,Y. and Wagner,D. (2009) Computational biology for cardiovascular biomarker discovery. *Brief Bioinform.*, **10**, 367–377.
30. Tan,A.C. and Gilbert,D. (2003) An empirical comparison of supervised machine learning techniques in bioinformatics. *Proc. First Asia-Pacific Bioinformatics Conf Bioinformatics*, **19**, 219–222.
31. Geurts,P., Irrthum,A. and Wehenkel,L. (2009) Supervised learning with decision tree-based methods in computational and systems biology. *Mol. Biosyst.*, **5**, 1593–1605.
32. Raghava,G.P., Searle,S.M., Audley,P.C., Barber,J.D. and Barton,G.J. (2003) OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics*, **4**, 47.
33. Thompson,J.D., Plewniak,F. and Poch,O. (1999) A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.*, **27**, 2682–2690.
34. Subramanian,A.R., Kaufmann,M. and Morgenstern,B. (2008) DIALIGN-TX: greedy and progressive approaches for segment-based multiple sequence alignment. *Algorithms Mol. Biol.*, **3**, 6.
35. Wallace,I.M., O'Sullivan,O. and Higgins,D.G. (2005) Evaluation of iterative alignment algorithms for multiple alignment. *Bioinformatics*, **21**, 1408–1414.
36. Kimura,M. (1983) *The Neutral Theory of Molecular Evolution*. Cambridge University Press, Cambridge.
37. Lassmann,T., Frings,O. and Sonnhammer,E.L. (2009) Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features. *Nucleic Acids Res.*, **37**, 858–865.
38. Wu,S. and Manber,U. (1992) Fast text searching allowing errors. *Commun. ACM*, **35**, 83–91.
39. Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.
40. Do,C.B., Mahabhashyam,M.S., Brudno,M. and Batzoglou,S. (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res.*, **15**, 330–340.
41. Berman,H.M. (2008) The Protein Data Bank: a historical perspective. *Acta Crystallogr A.*, **64**, 88–95.
42. Finn,R.D., Mistry,J., Tate,J., Coggill,P., Heger,A., Pollington,J.E., Gavin,O.L., Gunasekaran,P., Ceric,G., Forslund,K. *et al.* (2010) The Pfam protein families database. *Nucleic Acids Res.*, **38**, D211–D222.
43. Quinlan,J.R. (1986) Induction of decision trees. *Mach. Learn.*, **1**, 81–106.
44. Fan,W., Wang,H., Yu,P.S. and Ma,S. (2003) Is random model better? On its accuracy and efficiency. *Proceedings of the Third IEEE International Conference on Data Mining,* pp. 51–58.
45. Breiman,L. (2001) Random forests. *Mach. Learning*, **45**, 5–32.
46. Hunter,S., Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Binns,D., Bork,P., Das,U., Daugherty,L., Duquenne,L. *et al.* (2009) InterPro: the integrative protein signature database. *Nucleic Acids Res.*, **37**, D211–D215.
47. Holland,R.C., Down,T.A., Pocock,M., Prlić,A., Huen,D., James,K., Foisy,S., Dräger,A., Yates,A., Heuer,M. *et al.* (2008) BioJava: an open-source framework for bioinformatics. *Bioinformatics*, **24**, 2096–2097.
48. Quevillon,E., Silventoinen,V., Pillai,S., Harte,N., Mulder,N., Apweiler,R. and Lopez,R. (2005) InterProScan: protein domains identifier. *Nucleic Acids Res.*, **33**, W116–W120.
49. Sigrist,C.J., Cerutti,L., de Castro,E., Langendijk-Genevaux,P.S., Bulliard,V., Bairoch,A. and Hulo,N. (2010) PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Res.*, **38**, D161–D166.
50. Nuin,P.A., Wang,Z. and Tillier,E.R. (2006) The accuracy of several multiple sequence alignment programs for proteins. *BMC Bioinformatics*, **7**, 471.
51. Kjer,K.M., Gillespie,J.J. and Ober,K.A. (2007) Opinions on multiple sequence alignment, and an empirical comparison of repeatability and accuracy between POY and structural alignment. *Syst. Biol.*, **56**, 133–146.
52. Ogdenw,T.H. and Rosenberg,M.S. (2006) Multiple sequence alignment accuracy and phylogenetic inference. *Syst. Biol.*, **55**, 314–328.
53. Lee,C., Grasso,C. and Sharlow,M. (2002) Multiple sequence alignment using partial order graphs. *Bioinformatics*, **18**, 452–464.
54. Löytynoja,A. and Goldman,N. (2008) Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, **320**, 1632–1635.