

UTILISATION DES ALGORITHMES GENETIQUES POUR
L'ANALYSE DE SEQUENCES BIOLOGIQUES

Cédric Notredame

Doctorat en Bio-informatique

Février 1998

**Université Paul Sabatier
France**

Directeur De Thèse: Prof. François Amalric

Acknowledgement

I wish to thank the EMBL for their funding through an EMBL grant.

This work was carried out in the lab of Des Higgins, first at the European Molecular Biology Laboratory in Heidelberg, Germany and later at the EMBL outstation European Bioinformatics Institute, in Hinxton, U.K.

Des has been a constant source of support. I wish to thank him and express my gratitude for teaching me most of what I know in bioinformatics, and so much about being a scientist. I also wish to thank Ture Etzold who gave me a platform when Des had to leave for Ireland. Thanks to his expertise in the field, he had a lot of influence on my work.

A special thanks to the system managers, namely Roy Omond, Rodrigo Lopez Pettere Jokinen who have always been supportive, allowing me to overload machines any time I needed it. Without their help, this work could not have been achieved. I also wish to thank Miguel Andrade, Inge Jonassen and Burkhard for stimulating discussions and friendship. Chris Sander and Liisa Holm have always been available to share with me their extensive experience of the field. I wish to express my gratitude.

Michelle Magraine has proved an invaluable ally for fighting against my grammar weaknesses. I wish to thank her for that. I also wish to thank Rob H for joining me in my everyday fight against post script files. My stay at the EMBL was extremely enjoyable.

This work is dedicated to my friends and family for their constant support over the years.

RESUME DE THESE

Auteur: Cédric Notredame

Soutenance: Université Paul Sabatier, Salle des conférences du L.B.M.E. Le Lun Mars à 14 heures

Titre: Utilisation des Algorithmes Génétiques pour l'analyse de Séquences Biologiques

Directeur de Recherche: Prof. F. Amalric, L.B.M.E., Campus de l'université Paul Sabatier

Jury:

- Prof. F. Amalric
- Dr. Des Higgins
- Dr. William R. Taylor
- Dr. Manolo Gouy
- Prof. Jerzy Czaplicki

Mots Clés: Alignement de Séquences, Alignement Multiple, ARN, Protéines, SA RAGA, COFFEE.

Publications:

SAGA Sequence Alignment by Genetic Algorithm'
Notredame and Higgins
Nucleic Acids Research 1996, Vol 24, 8 1515-1524

RAGA: RNA Alignment by Genetic Algorithm
Notredame, O'Brien and Higgins
Nucleic Acides Research 1997 Vol 25, 22 4570-4580

COFFEE: an Objective Function for multiple sequence alignment evaluation
Notredame, Holm and Higgins
Bioinformatics, (in Press)

Optimisation of ribosomal profile alignments
O'Brien, Notredame and Higgins
Bioinformatics, (in Press)

Laboratoire d'accueil

European Institute of Bioinformatics
Genome Campus
Hinxton CB10 1RQ
United Kingdom

RESUME

Une grande partie de la recherche fondamentale en biologie moléculaire repose sur l'étude et des acides nucléiques. Ces molécules extrêmement complexes résultent de la combinaison plus simples: les acides aminés et les nucléotides. Vingt acides aminés constituent la grande des protéines, cinq nucléotides constituent la plupart des acides nucléiques. Le terme s'utilise pour désigner l'enchaînement de nucléotides constituant un acide nucléique ou l'enchaînement d'acides aminés constituant une protéine. La plupart des protéines sont codées par les gènes dans l'ADN des chromosomes.

Au cours de ces dernières années, de nombreux progrès techniques ont rendu possible le séquençage à grande échelle du génome de plusieurs espèces bactériennes ou eucaryotes. Ces données d'ADN sont entreposées dans des banques de données spécialisées (Swiss Prot, Gene Bank, nucléotides database...) dont la croissance (en taille) est aujourd'hui exponentielle.

La bio-informatique est une sous-discipline de la biologie ayant pour objet l'analyse de données par des moyens informatiques. Le principe de base d'une telle approche est la relation entre fonction et séquence. Le but est d'extrapoler des données obtenues de façon expérimentale sur certaines séquences à d'autres séquences pour lesquelles aucune donnée expérimentale n'est disponible.

Alors qu'il est clair que deux protéines (ou acide nucléiques) ayant la même séquence présentent probablement la même fonction, une corrélation devient plus difficile à établir lorsque les séquences ne présentent qu'une homologie partielle. La nécessité d'utiliser ce type d'information est la motivation derrière le développement des méthodes de comparaison aujourd'hui utilisées. Le thème présenté dans cette thèse est essentiellement consacré à cet aspect de la bio-informatique.

L'un des moyens les plus utilisés pour la comparaison de séquences est l'alignement. L'alignement permet d'identifier les zones conservées entre deux séquences. Ces zones correspondent à des motifs structuraux ou fonctionnels dont l'identification permet de faire des hypothèses quant à la fonction putative des séquences analysées. De façon plus générale, un alignement permet l'identification de régions sur lesquelles existent des contraintes diverses, imposant certaines propriétés. D'autre part, un alignement de qualité permet l'évaluation de la distance séparant deux organismes, ou deux protéines.

Cependant, dans les cas complexes, la quantité d'information contenue dans deux séquences n'est pas suffisante, et il devient nécessaire d'étendre la comparaison à plusieurs séquences. L'objet des alignements de séquences multiples. Leur problématique est double.

Il s'agit tout d'abord d'un problème biologique. Étant donné un groupe de séquences, les propriétés de l'alignement optimal doivent être définies. La règle la plus simple est de tenter d'obtenir autant d'identités que possible dans les colonnes tout en limitant le nombre d'insertions (gaps). En pratique néanmoins, les règles utilisées sont plus complexes et peuvent prendre en compte la nature des acides aminés alignés (protéines) ou la structure secondaire des séquences (ARN) de donner à cette liste de règles une forme mathématique associant un score à chaque alignement. On parle alors de fonction objective. Un nombre important de fonctions de ce type ont été proposées au cours de ces dernières années. Globalement, elles peuvent être divisées en deux groupes: les fonctions basées sur des matrices de substitutions et des pénalités d'insertion/délétion et les fonctions basées sur des modèles probabilistes tels que les HMM (Hidden Markov Models). Une des propriétés les plus importantes d'une fonction objective est sa signification biologique. De façon idéale, une fonction doit assigner à un alignement optimal un score traduisant l'intérêt biologique de l'information qu'il contient.

Le second aspect est purement informatique. Il ne suffit pas d'avoir une fonction objective, il faut aussi être capable d'optimiser le score de cette fonction (i.e. produire l'alignement ayant le meilleur score). Ce problème est loin d'être trivial. L'optimisation de la plupart des fonctions objectives appartient à la classe des problèmes dits NP-complets. En conséquence, l'optimisation n'est généralement pas réalisée qu'en utilisant des méthodes dites heuristiques qui ne garantissent pas une solution optimale.

Le travail présenté dans cette thèse englobe l'ensemble de ces problématiques. Dans la première partie, une méthode d'optimisation globale par algorithme génétique est proposée. Cette méthode est intégrée dans un logiciel nommé SAGA (Sequence Alignment by Genetic Algorithm). Les algorithmes génétiques sont des stratégies d'optimisation basées sur une analogie avec le processus de sélection naturelle. Cette méthode peut en théorie être appliquée à n'importe quel type de fonction objective.

Le second aspect du travail a consisté à définir une nouvelle fonction objective (Consistency based Objective Function for alignment Evaluation) et à optimiser cette fonction en utilisant SAGA de façon à prouver que COFFEE peut induire la création de meilleurs alignements que les méthodes alternatives.

La troisième application a été axée sur l'alignement d'ARNs ribosomiques avec détermination d'une fonction objective adaptée à la prise en compte des interactions secondaires. Ce programme adapté de SAGA a été nommé RAGA (RNA Alignment by Genetic Algorithm). L'une des principales limitations de RAGA réside dans la simplicité de la fonction objective utilisée. Afin de résoudre ce problème, un travail d'analyse a été réalisé sur des alignements de référence afin de déterminer des paramètres pouvant aider à la définition d'une fonction objective plus réaliste dans la prise en compte des contraintes à modéliser dans l'alignement. Ce travail constitue la quatrième application présentée dans cette thèse.

Dans l'ensemble, ce travail a permis d'établir l'utilité des algorithmes génétique contexte des problèmes d'alignement de séquences multiples. SAGA est à l'heure actuelle le plus performant pour l'optimisation des fonctions objectives couramment utilisées alignements de séquences multiples. Dans le cas de séquences protéiques, SAGA est le seul capable de réaliser l'alignement global de plus de dix séquences. Pour ce qui est de l'ARN r RAGA est le seul programme capable d'aligner des séquences ayant une longueur supérieure paires de bases, tout en prenant en compte les pseudo-noeux. D'autre part, la fonction C l'une des rares fonctions capable de permettre la génération d'alignements biologiquement que ceux obtenus par ClustalW (ClustalW est une des méthodes d'alignement les plus popul

RESUME DES ANNEXES

Document Numéro 1

SAGA: Sequence Alignment by Genetic Algorithm

Dans cet article, une nouvelle approche est proposée pour la résolution du problème des alignements de séquences multiples. Un algorithme génétique a été conçu et intégré dans un logiciel nommé SAGA. La méthode implique l'évolution d'une population d'alignements. Dans ce contexte, évolue la qualité des alignements est graduellement améliorée au gré d'une succession (générations) contenant des étapes de modifications aléatoires (opérateurs) ainsi que de sélection basée sur le score. Le degré d'amélioration est jugé par l'évaluation du score d'alignement à l'aide de la fonction objective. SAGA utilise une technique de contrôle automatique pour réguler l'utilisation simultanée de vingt opérateurs destinés à recombiner entre eux des alignements (crossing overs), ou bien à les modifier individuellement (mutation). Afin de tester SAGA, nous avons utilisé comme référence le programme M.S.A. (Multiple Sequence Alignment) capable d'optimiser des fonctions objectives le plus couramment utilisées (somme des paires avec pénalités de délétion/insertions).

Utilisé dans ce contexte, SAGA fournit de meilleurs résultats que M.S.A. en termes d'optimisation (score de l'alignement obtenu). De plus les alignements produits par SAGA sont biologiquement plus exacts s'il on en juge par leur similarité avec l'alignement des mêmes séquences réalisés par comparaison de structures. Au total, SAGA a été testé sur treize groupes de séquences pour lesquelles un alignement de référence basé sur les structures est disponible dans la banque de données 3D_ali.

Document Numéro 2

COFFEE: A New Objective Function for Multiple Sequence Alignments

Dans ce travail, nous présentons un nouveau mode d'évaluation des alignements de séquences multiples. Cette fonction est nommée COFFEE. COFFEE est une mesure du degré de consistance existant entre un alignement de séquences multiples et un alignement de référence de mêmes séquences alignées deux par deux. Il est montré que le score COFFEE peut être optimisé par SAGA. La fonction a été utilisée sur onze groupes de séquences pour lesquels un alignement de référence est disponible dans la banque de données 3D_ali. Dans neuf cas, SAGA, utilisé avec COFFEE, produit des alignements meilleurs que ceux obtenus avec ClustalW (on en juge par leur similarité avec les alignements de références). Nous avons aussi montré que COFFEE peut être utilisé pour évaluer la qualité d'un alignement multiple, de façon globale. Finalement, la bibliothèque de référence peut être constituée d'alignements en provenance de la banque de données 3D_ali par comparaison de structure (par exemple, des alignements extraits de FSSP). Dans ce cas, COFFEE est capable de produire des alignements structuraux multiple de très haute qualité. COFFEE devrait permettre d'appliquer aux alignements multiples n'importe quelle méthode d'alignement de paires de séquences.

Document Numéro 3

RAGA: RNA Sequence Alignment by Genetic Algorithm.

Cet article décrit une nouvelle approche pour aligner deux séquences d'ARN homologues dont la structure secondaire de l'une des deux molécules est connue. A cette fin, deux programmes ont été développés, RAGA (RNA séquence alignment by Genetic algorithm) et PRAGA (Parallel RAGA). Ces deux programmes sont essentiellement basés sur le programme SAGA. La parallélisation est réalisée par la synchronisation d'un nombre défini de copies actives de RAGA. Celles-ci évoluent indépendamment dans leur population suivant une topologie définie comme étant un arbre à branche de profondeur variable.

Cette méthode permet d'optimiser une fonction objective prenant en compte les inf primaires et secondaires contenues dans les deux séquences. Une des propriétés les plus int RAGA réside dans le fait qu'il est possible de prendre en compte aussi bien les tiges boucles que les pseudo-noeux présents dans l'ARN ribosomique.

RAGA à été testé à l'aide de neuf alignements de référence extraits à partir d'alig d'experts. Ces alignements, constitués d'ARNs de la petite sous unité ribosomique, ont référence. Dans chacun des cas, PRAGA est capable de surpasser en exactitude les méthodes basées sur la programmation dynamique. Ceci est vrai même lorsque la distance phylo séparant les deux séquences à aligner est très importante (comme entre l'humain et sac cerevisiae).

*Document Numéro 4
Optimisation of RNA Profile Alignments*

Ce projet fait pendant au projet numéro 3 et s'intègre dans un contexte plus large visant des outils nécessaires à la maintenance automatique des banques de données d'ARN riboso part le monde, plusieurs banques de données de ce type existent. Elles sont essentiellement de façon manuelle. A long terme, la création de méthodes automatiques va devenir un absolue. Dans le document numéro 3, nous avons proposé une procédure destinée à l'ali deux séquences. N'utiliser que deux séquences revient à ignorer la vaste quantité d'informat dans les alignements multiples établis par des groupes d'experts. Le but de ce projet a été certaines des modalités d'utilisation de cette information.

Différentes méthodes de pondérations ont été testées et implémentée dans un co programmation dynamique. L'évaluation des méthodes a été réalisée en testant la qualité obtenue lorsqu'une séquence est extraite puis réintroduite dans un alignement multiple. C ne prend en compte que les contraintes primaires.

Les résultats montrent que par l'utilisation d'un mode de pondération adéquat et d'un pénalités d'insertion/délétion adapté, il est possible d'améliorer considérablement la l'alignement entre un profil et une séquence.

TABLE OF CONTENTS

1-INTRODUCTION.....	9
1.1 BIOINFORMATICS AND BIOLOGY.....	9.....
1.2. COMPARING SEQUENCES.....	11.....
1.3 OUR APPROACH.....	12.....
2-THE SCOPE OF SEQUENCE ALIGNMENTS.....	14
2.1 WHAT IS A SEQUENCE ALIGNMENT?.....	14.....
2.2 WHAT IS THE USE OF A SEQUENCE ALIGNMENT?.....	14
2.3 WHAT IS A 'GOOD' ALIGNMENT?.....	15...
2.4 HOW TO BUILD A 'GOOD' SEQUENCE ALIGNMENT?.....	16.....
3-EVALUATING ALIGNMENTS.....	16
3.1 PROTEIN PAIRWISE ALIGNMENTS.....	16...
3.1.1 Substitution Matrices.....	16....
3.1.2 Gap penalties.....	19.....
3.1.3 Database Searches.....	20....
3.2 EVALUATING PROTEIN MULTIPLE SEQUENCE ALIGNMENTS.....	21.....
3.2.1 Using Substitution Matrices and Gap Penalties: SP Alignments.....	21.....
3.2.2 Weights.....	23.....
3.2.3 Profiles.....	25.....
3.2.4 hidden Markov models.....	26.....
3.3 RNA ALIGNMENTS: TAKING INTO ACCOUNT NON LOCAL INTERACTIONS.....	28.....
4-MAKING MULTIPLE SEQUENCE ALIGNMENT.....	30
4.1 COMPLEXITY OF THE PROBLEM.....	30...
4.2 DETERMINISTIC GREEDY APPROACHES.....	30.
4.2.1 Aligning Two Sequences.....	31...
4.2.2 Aligning Two Alignments : Progressive Alignment Methods.....	32.....
4.3 DETERMINISTIC APPROACHES FOR NON PROGRESSIVE MULTIPLE ALIGNMENTS.....	34.....
4.3.1 The Carrillo and Lipman Algorithm.....	34..
4.3.2 Other Approximation Techniques.....	34.....
4.4 STOCHASTIC HEURISTICS.....	35....
4.4.1 What is a stochastic Method?.....	35.....
4.4.2 Iterative alignments and Expectation-Maximization Strategies.....	35.....
4.4.3 Simulated Annealing.....	36....
4.5 GENETIC ALGORITHMS.....	37.....
4.5.1 What is a Genetic Algorithm?.....	37...
4.5.2 Applications of GAs in Sequence Analysis.....	39.....

5-COMPARISON OF THE METHODS 40

6-SUMMARY OF THE CONTRIBUTIONS..... 41

6.1 SAGA: MAKING MULTIPLE SEQUENCE ALIGNMENT BY GENETIC ALGORITHM.....41.....

6.2 COFFEE: IMPROVING ON EXISTING OBJECTIVE FUNCTIONS.....41.....

6.3 RAGA: THREADING RNA SECONDARY STRUCTURES.....42

6.4 OPTIMIZING RIBOSOMAL RNA PROFILE ALIGNMENTS.....43.....

CONCLUSION 43

REFERENCES..... 44

APPENDIX: RESEARCH PAPERS

- Annex 1 "SAGA: Sequence Alignment by Genetic Algorithm"
Notredame and Higgins, 1996
- Annex 2 "COFFEE: A new Objective Function for Multiple Sequence Alignments"
Notredame, Holm and Higgins, 1998
- Annex 3 "RAGA: RNA Alignment by Genetic Algorithm"
Notredame, O'Brien and Higgins, 1997
- Annex 4 "Optimisation of ribosomal RNA profile Alignments"
O'Brien, Notredame and Higgins, 1998

1-INTRODUCTION

1.1 BIOINFORMATICS AND BIOLOGY

Life as we know it is a complex arrangement of biological structures designed with each other. As complex as they may appear, even the most elaborated living can be described as arrangements of smaller less complex building blocks (some of which are themselves the result of the combination of even smaller basic biological metabolites, proteins, nucleic acids).

Identifying these structures and characterizing their functions is a major task. Questions can be addressed at any level of organization (one may wish to go from populations to atoms in fact). In such a top to bottom approach, molecular biology starts almost at the bottom. It deals with the biological structures at the molecular level to understand how these are created and interact with one another to perform functions of life.

The search for ordered systems is an important part of the biological method. Systems usually make it possible to establish general rules allowing a global understanding of otherwise disparate collections of facts. In this respect, the discovery of DNA and the understanding of RNA and protein synthesis have been two of the major milestones of modern molecular biology. They have allowed a deep understanding of some of the most central cellular mechanisms. We now know that proteins and RNA molecules are involved at virtually all the steps of biological processes. We also know that these key components of cellular life are coded in DNA sequences in almost universal manner. These DNA sequences are contained in the genomes of organisms.

At the lowest level, a genome can be described as a long string of nucleotide bases compared to a very long text made of four letters. As in a text, the letters are at random but organized in words. In the context of a genome, a word will be having a function. One of the difficulties when trying to identify these 'words' is the fact that nature uses spaces and punctuation in a very personal way. This is why we only do we ignore beforehand the function of the 'words', we also do not know where they start and finish. Add to this the fact that there are many different classes of words. Some allow the binding of other molecules on the DNA, others are translated into proteins. A protein or an RNA molecule may contain motifs that will have functions (binding, catalytic site...). The genome also contains very specific combinations of words such as genes (enhancer, promoter, exons...). Bioinformatics and molecular biology are two complementary techniques with similar aims: identification of biological structures and sub-structures at a molecular level and characterization of their function.

'Function' is a very general concept. If we look at it from an experimental point of view such as genetics, a function can be defined with respect to a gene and will refer to what happens when this gene is inactivated/modified by a mutation. Often it is not enough to gain a real deep understanding of the mechanisms involved. To do so we have to know whether this function is performed by a protein, an RNA, or a DNA sequence. If it is a protein then the next question is 'how does the protein perform its function?'. If it is an enzyme we will want to know where is the catalytic site like any other known site, what are the residues involved in the site and how do they perform their function? The protein (enzyme or not) may also interact with other proteins, acids or metabolites. Here again we will want to know what are the portions of the protein involved in these operations and what are the potential partners. In most cases, this information will be much easier to understand/predict when a 3D model is available for a protein. In a broad sense, the function of a protein (or a nucleic acid or a regulatory element) is defined by the sum of all these elements.

Until recently, the only way available for gathering together these pieces of was to use wet lab techniques. These involve genetic analysis, cloning, and interaction experiments..... Although the results obtained that way can usually be as strong biological evidence, they suffer from a major drawback. The cost of these techniques is extremely high, in terms of time and money. This means that the number of functions that can be thoroughly investigated through such techniques is limited. The problem has become especially severe now that the improvement of sequencing techniques gives us access to far more sequences than it will ever be possible to test in the wet lab. It is this situation that has promoted the massive development of computational techniques over the last few years.

Bioinformatics could be regarded as an approach diametrically opposed to experimental ones. Instead of starting from a phenotype, one will start with a sequence and try to gather as much information as possible by comparing this sequence with other sequences for which experimental evidence is available. But the difference between the two approaches is much less acute than it seems. Bioinformatics relies on the same basic assumptions as classic biology. It is a method of inquiry based on a series of comparisons and classifications/predictions. The main paradigm of bioinformatics is that sequence conservation is correlated to function conservation. Under this framework, one can extrapolate, as much as possible the information acquired experimentally to sequences that are not. This follows a traditional feedback scheme where models are built and validated by experiments made in wet lab or in silico.

Darwinian laws of evolution, and the notion of parsimony often underlie the bioinformatics approach. The assumption is that biological systems have evolved from the constantly reusing some basic building blocks (such as metabolic pathways) to new ones to respond to their environmental constraints. If each time a new constant a new biological system was created from scratch, the bioinformatics approach probably be bound to fail. Fortunately, in most of the cases, this is not the case. Through the cycles of mutation/selection that constitute evolution, new functions are created by reusing pieces of already existing machinery, and existing functions evolved to become more adapted to the environment in which they are found. If we consider this problem in terms of sequences, this means that two sequences that perform similar functions may be different, depending on how long they have been diverging, how long ago the original sequence was duplicated, or how long ago the two sequences containing these sequences started diverging). Nevertheless, if the distance between these sequences is small enough, an evolutionary scenario can be reconstructed that will show the relationship between these sequences. Depending on what is known for one of these sequences (the function of the sequences of the same category), it will then become possible to make assumptions about the function of the other. On the other hand, if the sequences are evolutionarily too far apart, analysing their relationship may prove difficult by simply comparing the sequences. The signal they contain may have to be enhanced using other techniques such as structural prediction.

Sequences are only conceptual objects. As such, they have no function in a vacuum. Even the distinction between RNA, proteins and DNA is artificial. As far as function is concerned, all these elements only exist as a complex 3D arrangement of atoms. It is the precise 3D structure that a molecule has that determines its mechanical and chemical needs to perform its function (catalytic activity, interactions...). The relationship between structure and function is probably one of the oldest paradigms of molecular biology. We also accept that broadly speaking, structures are induced by sequence although for a fact that very different amino acid sequences can code for similar 3D folds.

This last point helps understand why proteins with different sequences can perform similar functions and structures, since natural selection is applied on the active 3D structure rather than on the sequences (i.e. evolution gives more freedom to the sequence than to the structure). As a consequence, relationships between proteins (or RNAs) are usually easier to analyse when the structures are known. Unfortunately, structures are

determine experimentally and prediction from sequence alone (ab initio fold is still one of the main challenges of computational biology. It is true however tools exist that can help supplement weak sequence identity.

Developing new techniques for automatically analysing sequences is one of the purposes of research in bioinformatics. It is a point of crucial importance. The major databases of nucleotide or protein sequences are growing in size at a rapid rate (doubling every year or so). It means that the proportion of sequence data that are available is decreasing. For this reason, targeting the parameters of experiments are needed has become more important than ever. Such a goal is achieved by gaining some more understanding on the ways in which information is extrapolated from one sequence (or a set of sequences) to another. This is not available for making any use of the DNA sequencing results (at least in a reasonable amount of time). It is for this reason that sequence comparison tools are at the forefront of the bioinformatics approach.

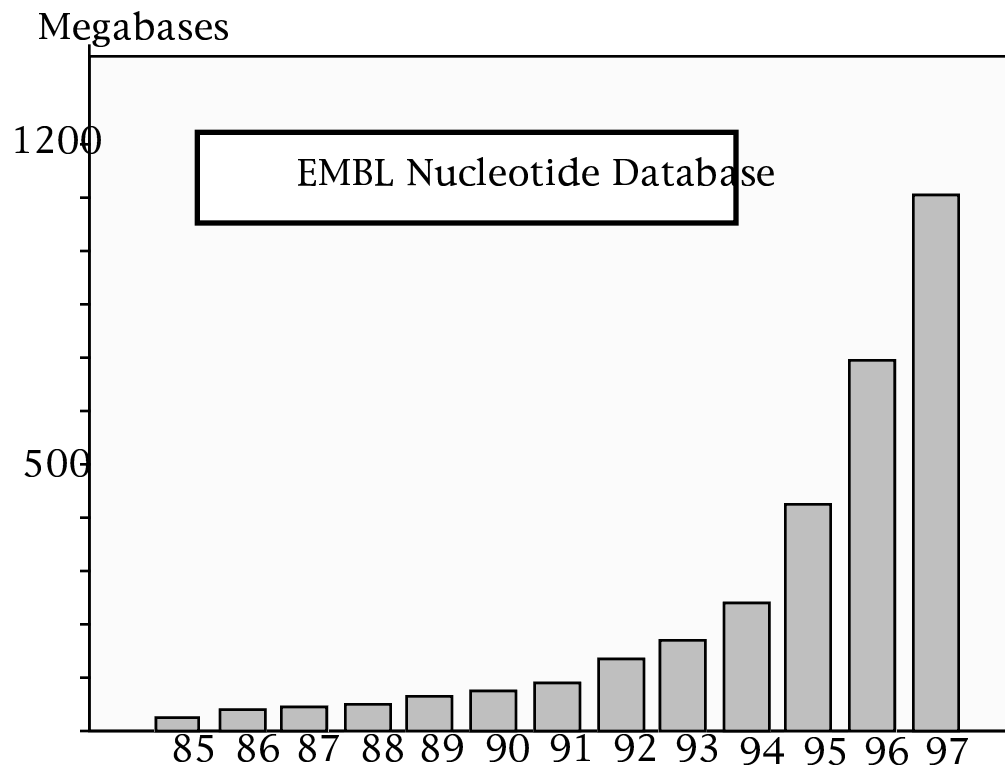


Figure 1. Growth of the EMBL Nucleotide database over the 1985-1997 period. The release of the EMBL nucleotide sequence database (Rel. 52, October 1997) contained 1,181,167,498 nucleotides. The last release of the Swiss-Prot Database (Rel. 32 October 1996) contained 21 210 389 amino acids in 59021 entries. For comparison, the release (proteins with known structures) of December 1997 contained a total of 6731

1.2. COMPARING SEQUENCES

In most cases the problem facing the user takes the following form: a new sequence is available and it is desirable to search the database and find out whether any relatives of this sequence have already been reported. If so, one may wish to use some of the experimental data gathered that way to the new sequence. In such a situation the solution is to compare the sequences of interest to all the sequences con

database, keeping track of the most similar. Two very popular tools are used for basic database similarity searches: FASTA(1) and BLAST(2).

Sequence comparison can also be much more complex. For instance, by analyzing experimental data contained in the databases and sequence analysis, one may find if a specific motif is sufficient for a protein to bind zinc. If several types of motifs emerge when doing such analysis, one may want to build a classification of experimental data that allows the establishment of differences between the zinc-binding motifs (some may be associated with RNA-binding proteins and others with DNA-binding proteins). This, for instance, is one of the motifs developed in the Prosite database(3). Once such results have been established, sequences can be scanned for the known motifs they contain.

As simple and trivial as they may seem, such strategies present various difficulties that need to be overcome. First of all, one has to define what is of interest. This is typically a biological problem. The features one is interested in looking at two sequences or more will obviously depend on the aim and the type of comparison. Do we want to compare two proteins for having the same sequence, for having related functions, for being expressed in the same cell, for having similar folds? Are these questions equivalent? When it comes to making a sequence analysis, sequence alignments appear as one of the most powerful solutions. In the simplest case, they only involve two sequences (pairwise alignments) but when extended to a larger number (multiple sequence alignments).

Having decided that we want to use sequence alignments, the problem of defining a 'good' sequence alignment remains. It is a difficult question that requires a deep understanding of the biological information one wishes to extract from such alignments. In most of the cases, the criterion that will allow evaluation of the quality of an alignment will take the form of a mathematical function (objective function) associating a score to an alignment. But even so, the problem is not solved. Having a criterion for alignment is not enough. One also needs to be able to build the best scoring alignment according to this quality criterion. In most of the cases, this is far from easy. Many of the algorithms developed in bioinformatics and more specifically in sequence alignment are said to be NP-complete. This means that the number of potential solutions rises exponentially with the number of sequences and their length (i.e. the solution cannot be found in polynomial time). The need to overcome such severe limitations requires the development of new algorithms.

1.3 OUR APPROACH

In the work presented here, the problem of sequence alignment was approached from the two aspects mentioned above:

- defining new objective functions for sequence alignment.
- developing new ways to optimize these functions.

One of the main concerns of the approach was the fact that there is no use in developing new sequence comparison schemes if no tool is available to use them and allow judgment on their potential relevance. To test an objective function, one must be able to optimize it and compare the quality of the alignments it provides with other methods.

The new optimization scheme proposed here is a genetic algorithm named SAGA (Sequence Alignment by Genetic Algorithm). This algorithm was used to evaluate the biological relevance of COFFEE (Consistency Based Objective Function for a Sequence Alignment Evaluation), a new objective function designed for protein multiple sequence alignment (Annex 2). SAGA was also adapted to RNA alignments (RAGA: RNA Alignment by Genetic Algorithm) using an objective function that takes into account the secondary structure of the RNA.

secondary structure interactions in RNA (Annex 3). In order to improve RAG function, a new function was designed for aligning an RNA sequence to a la RNA sequence alignment (Annex 4). This function was only tested using a optimization method (Dynamic Programming).

The following sections will deal with the three main concepts associated w alignment: what it is useful for, how to define a sequence alignment and fi build a sequence alignment. The last section will put the four contributions i context.

2-THE SCOPE OF SEQUENCE ALIGNMENTS

2.1 WHAT IS A SEQUENCE ALIGNMENT ?

A sequence alignment is the representation of two sequences in a way that shows their relationship. If the alignment is correct, two residues aligned with each other are homologous. The definition of homology depends on the criterion used for it. For instance, if the aim is to identify the relationship between two structures, two residues will be aligned because they are equivalent in the 3D structures. If the alignment is to reflect phylogenetic relationships, two residues will be aligned when they occupy the same residue in the common ancestor. The definition of a pairwise alignment can be extended to multiple sequence alignments. In this case, several sequences are aligned together and each column contains homologous residues. However, homologous residues do not necessarily exist in each sequence for each position of the alignment. If a sequence lacks one residue, a gap will be inserted in its place at the corresponding position. Gaps usually take the form of strings of nulls. In an evolutionary context, a gap means that a residue was inserted in one of the sequences or deleted in the other in the sequences were diverging from their common ancestor.

There are two types of alignments: global and local. In a local alignment, only the parts that are aligned are those which are clearly homologous. The rest of the sequences are ignored. In a global alignment, whole sequences are aligned, regardless of their similarity. The scope of global and local alignment is usually different. Local alignments are more appropriate when the sequences analysed are remotely related and consist of a few domains. Global alignments are mostly designed to analyse sequences known to be homologous to one another. In this thesis, I will mostly consider local sequence alignments.

It is also important to realize that, given a set of sequences, there are lots of possible alignments. For instance, given two sequences of 1000 residues each, there are 10^{764} different possible alignments. This rules out any naive enumeration of alignments to identify the correct one! Instead, we will see that several strategies have been developed that allow more or less efficient computation in polynomial time.

2.2 WHAT IS THE USE OF A SEQUENCE ALIGNMENT ?

Quantitatively, the most widely used application of sequence alignment is database searching. When doing so, the aim is to find for a given query sequence related sequences contained in a database. The principle is very straightforward: the query sequence is aligned in turn to all the members of the database and results are selected according to some similarity criterion. FASTA(1) and BLAST (2) are the most popular of this type. They rely on local alignments rather than global.

However, important as the results obtained in this way are, there is a clear limit on the alignments that can be deduced from these searches. Firstly, the sequence alignment algorithms implemented in these programs are only crude approximations of a sequence alignment algorithm. This is necessary in order to search very large databases in a reasonable amount of time. Secondly, in most of the cases, the searches are based on pairwise alignments. This means that they only contain a limited amount of information.

Although database searching is at the heart of many approaches, pushing it further may require important refinements. Such refinements can involve sequence prediction, identification of new motifs or domains, generalization of the family (i.e. combining the information contained in the known sequences in order to predict new ones).

distant members), phylogenetic analysis. For all these applications, pairwise alignment is of limited use. A way to simultaneously combine the information contained in multiple sequences is needed. Such a need is the main motivation for building multiple sequence alignments.

Multiple alignments are very important for phylogenetic analyses because they provide a way to compute evolutionary distances and phylogenetic trees. Trees are constructed from sets of pairwise distances using some clustering algorithms such as the neighbor-joining method(4). When computing a tree, it is very important to have accurate pairwise alignments, hence the use of a multiple alignment in which the pairwise alignment of two sequences depends on the information contained in all the sequences of the set.

Another fundamental application of multiple sequence alignments is the identification of motifs or domains. In a multiple alignment, these elements often appear as conserved regions in which constraints exist that limit divergence. If some of the sequences are experimentally characterized, these motifs can be used for function prediction. This is, for instance, the aim of the Prosite(3) or the ProDom databases(5). The information contained in a multiple sequence alignment can also be generalized in order to produce a hidden Markov model (7) that can be used for identifying new family members.

The other important use of multiple alignments is structure prediction. In a protein, residues do not evolve in the same fashion, depending on their role in the structure (buried/exposed, helix/beta strand/loop...). It is very hard to extract such information from a sequence alone while it can be accessed through analysis of multiple alignments by looking at the distribution of the substitutions. Using multiple sequence alignments for the prediction of protein secondary structure has had a dramatic effect on this area of sequence analysis, increasing the accuracy of protein secondary structure predictions from 55%(8) (9) to 75% (10). One can also go further and try to identify correlated mutations in multiple alignments. This has been done on many occasions in proteins with limited resolution (11-13). On the contrary, in RNA analysis, the identification of correlated mutations is of great help, allowing accurate prediction for secondary structure analysis and the identification of structures(14-16).

Finally, a less challenging but very important application of multiple sequence alignments is the localization of highly conserved areas for the design of efficient PCR primers to clone new members of a family. All these examples reflect the importance of multiple sequence alignments in the domain of sequence analysis. We will show here that finding good multiple sequence alignments is a multi-step task.

2.3 WHAT IS A 'GOOD' ALIGNMENT ?

A scoring function associates a score to an alignment. Ideally, the better the alignment, the more biologically accurate the alignment. An alignment with the best possible score is said to be optimal, whether it is biologically relevant or not. An optimal alignment always exists. Being able to distinguish between biologically relevant and non-relevant alignments is an important issue, especially when analyzing databases. Powerful statistical methods have been developed for this purpose, allowing the discrimination of hits for their biological relevance(2). This problem remains when aligning sequences from different species. For instance, two homologous domains may surround a loop that is shared by both structures. In this case, any alignment of the residues contained by these two structures is meaningless even if a mathematical optimum exists.

Another important problem, common to many areas of computational biology, is the choice of parameters. Most of the objective functions come with a large number of parameters. In many cases, one has to rely on empirical values, known to be good (i.e. small changes of the parameter values may induce very different alignments). This may lead to inaccurate alignments. This explains the fact that a large amount of research is dedicated to objective function definition focused on parameter elimination.

much work has been done in this field that the choice of a scoring scheme regarded as one more parameter requiring optimization. Among the countless existing methods, we will only describe some of the most important schemes those related to the work carried out for this thesis.

There are two types of alignments: sequence and structure alignments. Sequence alignments do not require any non local interactions to be taken into account therefore less complex (algorithmically speaking) than structural alignments. In the problem of sequence alignments we will mostly talk about protein sequence alignments. The problem of structure alignment will be addressed through the example of RNA structures. The problem of protein structure alignment will not be analysed due to its complexity and the amount of literature available on this subject put it beyond the scope of this dissertation which is mostly oriented toward sequence analysis rather than structure.

2.4 HOW TO BUILD A 'GOOD' SEQUENCE ALIGNMENT

In many cases, building a sequence alignment takes the form of a compromise between biological relevance, mathematical optimality and efficiency. Given a scoring function, it may be very hard to produce the mathematically optimal alignment mentioned earlier than done in a naive way through enumeration, sequence alignment beyond the scope of any computer. For this reason, trade-offs need to be made. Objective functions need to be defined in such a way that they fit all optimization techniques, and optimization techniques need to be improved to accommodate the complexity of the problem.

For instance, in its more general form, the problem of aligning two sequences is NP complete(17). However, if formulated under certain constraints, it can be solved by a technique known as dynamic programming(18) but becomes NP complete a number of sequences (i.e. when trying to align more than two sequences simultaneously)(19). Structure alignments (i.e. sequence alignments taking into account non local interactions) are also NP complete, even for two sequences, and addressed in a simplified form(i.e. using an objective function that does not take into account known constraints)(20).

Because of this NP completeness most of the algorithms developed in the field of sequence alignments are heuristics(21-24). It means that they do not provide a mathematically optimal solution, but rather a good approximation. In many cases a trade-off is reasonable and allows the computation of multiple sequence alignments in a reasonable manner. In this thesis, I will describe some of the optimization methods currently used with a special emphasis on the genetic algorithms.

3 EVALUATING ALIGNMENTS

3.1 PROTEIN PAIRWISE ALIGNMENTS

3.1.1 Substitution Matrices

The twenty amino acids commonly found in proteins have very specific physico-chemical properties such as size, charge and hydrophobicity. The role of a residue in a protein mostly depends on these properties. For this reason, substitutions do not occur in a way that reflects physico-chemical constraints in the 3D structure. It is a very intuitive idea to try to associate with each possible substitution a cost or a probability. This information can be stored in what is known as a substitution matrix, a 20x20 table giving the relative cost or the probability of each possible substitution. Although many types of matrices have been proposed, the most commonly used are those derived empirically (25, 26). The principle often involves statistical

PAM matrices are computed in a similar fashion. The main difference is that the frequencies are measured in a way that takes into account sequence identity using the collection of blocks, several sets of matrices can be generated without doing any extrapolation. They range from 80 to 45 % average identity. BLOSUM have been shown on various occasions to outperform PAM or other matrices(

The main reproach that can be made to these two types of matrices, is that they are too general while only relying on alignments with a low information content (more than 85% identical), or domains that can be aligned without gaps (question of the existence of bias in these matrices has often been discussed let us consider alpha helices and beta strands. The type of substitutions observed for two types of structural elements are known to be slightly different (this is the efficient secondary structure prediction algorithms(29)). As a consequence, a matrix built using a data set that contains more helices than beta sheets, it will perform poorly when aligning portions of sequences coding for the beta-sheets. On the other hand, if the data set is equilibrated for the two types of structures, the matrix will simply be an average, and will not be as good as it could be in either of the two cases. To attempt to compensate for this type of potential problem, several alternative matrices have been built for helices(30), beta strands(30) or transmembrane domains(31). The way in which such matrices should be used is far from being obvious. The main problem often amounts to solving structure prediction, unless a structure is known for some of the homologous sequences in which one is interested.

Overall, about 40 different substitution matrices have been proposed, using different methods with different training sets in most cases. Recently, two groups have been made in an attempt to understand the fundamental differences between these schemes(27, 32). The main motivation in the work of Vogt et al. was to study the behavior of these matrices for the accuracy of the alignments they induce using computer programming (See Section 4.2.1). Correctness was judged using the structures contained in 3D_aln(33). Their work shows that there is very little difference between the best matrices (Gonnet(34), BLOSUM(26) and Benner(35)). They also compared matrices which are able to identify remote homologues in database searches leading to the more correct alignments. Finally, from an algorithmic point of view (4.2.1) they concluded that these matrices are more suited for global alignments than local alignments. The second study(32) focused on understanding the way these matrices are affected by amino acid properties. The authors found that PAM units are significantly more biased to volume and hydrophobicity while other matrices are much more biased to charge. Interestingly their results indicate that despite the different methods and data sets used for their construction, most of the substitution matrices based on sequence alignments are highly correlated with the Dayhoff's. When trying to group these matrices according to their level of correlation, the global matrices fall into the same cluster (i.e. are highly correlated) while structure-specific matrices fall into separate clusters. This is further supporting the idea that matrices should be applied in a way that takes into account structural information. We will see later that the Dirichlet mixtures(36) are an interesting alternative to this problem (see section 3.2.4).

The matrix comparison problem was also addressed in a different context using a smaller set of matrices, by Henikoff et al. who compared matrices for the discrimination of sequences in a database search using FASTA or BLAST (local alignments). The results obtained that way confirm that matrices based on the alignment of distantly related sequences (such as BLOSUM) or structures (37) perform much better than PAM matrices.

Finally, a point on which all these studies agree is the necessity of using appropriate gap penalties when scoring an alignment with a matrix. Most of the results observed with substitution matrices can be dramatically affected, depending on the set of gap penalties and score gaps. In the next section, we review some of the concepts underlying the scoring of gaps when aligning two sequences.

3.1.2 Gap penalties

Substitutions are not the only events affecting sequences while they diverge. deletions also occur. This means that two sequences may contain unrelated should not be aligned. Such an event is represented by a gap in the sequen receive the insertion, or where the deletion occurred.

Deletion	Terminal gap
XXXXXXX-----XXXXXXXXXX-----	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	
or Insertion	

It is obvious that a cost should be given to these events when scoring an Unfortunately the choice of a scoring scheme often implies some evolutionar we still lack a good understanding of the underlying biology of insertions/ instance, unless a very reliable phylogenetic scenario is available, it is often distinguish between insertion and deletion(38, 39), hence the word indel th position where an insertion OR a deletion occurred.

Some useful information can be gathered about indels by looking at alignments(39). As one would expect, indels do not happen at random, concentrate on the portions of the structure with less steric constraints such theory possible to extrapolate this information to sequences with unkn structure. For instance, Chou and Fassman(40) proposed using seconda propensity in order to derive some local gap penalties. Another method Pascarella and Argos (39) involves measuring, on reference structural alig probability of having a gap after a given residue. This scoring scheme implemented in ClustalW(21) where a gap can be given a cost depending o after which it occurs. Similarly, some heuristics have been implemented in C attempt to locate areas more prone to gap insertion such as stretches o residues, usually exposed in the loops. However, these methods are both e very general, they do not take into account the specificity of the sequences o in aligning (for instance due to the structural constraints, some positions conserved than others...). We will later discuss the way information der multiple sequence alignment can be used in order to established more reli propensity (cf. the Profile section, 3.2.3).

The question "where do gaps occur?" is only one part of the problem. When one also has to ask: "Is this gap long enough?". There is clear evidence that g penalized according to their sizes. Analyses made on mammals(41) sug logarithmic scheme could be quite appropriate, with a gap opening penalty an penalty that is a function of the length of the gap (i.e. a penalty per residue d the length of the gap). These results confirm those obtained by Benner et suggest that linear gap penalties are less than realistic (penalty cost increasin the gap length). Results also suggest that insertion and deletion should be from one another(38, 41). However, even if there is a wide agreement on the linear schemes would probably be biologically more relevant(42), alternat suffer from a major drawback: their implementation poses significant algorithm when it comes to optimizing alignments(43, 44). For this reason, in practice, are usually optimized under a simplified form known as "affine gap penal formalised as follow:

$$\text{cost} = \text{Gap Opening Penalty} + \text{Gap Extension Penalty} * \text{Length} \quad (1)$$

This gives penalties as a linear function of gap length and an efficient algorithm for optimizing this was introduced by Gotoh(45). There is no real justification for this type of model apart from the fact that it performs reasonably well, especially for gaps that are small (less than 20 residues). Since the size of indels is known to be a function of evolutionary distance(38), this means that linear gaps will be acceptable for closely related sequences. However, since affine gap penalties are not empirically derived, they raise the problem of defining the values of the two parameters: the gap opening penalty (GOP) and the gap extension penalty (GEP). There is no guaranteed way to choose values so that they fit the sequences one is interested in. A popular practice is to set the opening penalty a value equal to the average of the values contained in the substitution matrix used for comparison (excluding the main diagonal). The extension penalty is set to a tenth of the opening value. It is also common practice not to penalize gaps at least for opening.

When making an alignment there is a competition between gap insertion and gap deletion. To some extent, gap penalties can be regarded as thresholds used to decide whether a stretch of residues has a homologue or not in the other sequences. In this context, it is senseless to modulate the penalties with some local information (second-order statistics, propensity, profile information...). But even so, a major problem remains: when sequences aligned are only remotely related, the gap penalties lack robustness. As made by Vingron and Waterman showed that slight variations of values for the GEP can induce very different optimal alignments(46). Under such conditions, it can be hard to decide which alignment is biologically the most relevant. An attempt to improve the robustness of the penalty parameters has been proposed by Taylor: "Score Enhancement"(47). It originates from the observation that in biological alignments, gaps are usually clustered in a few parts of the sequences and long uninterrupted blocks. The technique proposed by Taylor involves enhancing the scores of long ungapped portions in order to avoid them being interrupted by gaps. It shows that under this scoring scheme, a correct guess for the penalty values is less critical than previously reported.

There is little doubt that the correct treatment of gaps and a deep understanding of sequence biology are critical parameters when making accurate sequence alignments. As shown here, the problem is mostly algorithmic. It is possible to define gap penalties that describe sequence relations in a realistic way, it is simply not practical to do so. The problem of practicality becomes even more acute when it comes to scanning large databases containing hundreds of thousands of sequences.

3.1.3 Database Searches

An exhaustive treatment of database search methods is beyond the scope of this book. The most commonly used methods will be briefly described here as they relate to the alignment problem. They involve specific scoring schemes designed for evaluating the statistical significance of an alignment. The principle of a database search is quite straightforward. A query sequence is aligned in turn with all the other sequences of the database. Depending on the score, alignments are kept as relevant or discarded.

BLAST(2) is probably one of the most popular methods for database searching. For proteins, the method involves finding the High Scoring Pairs of residues (stretches of aligned residues, with no gap, which have high scores). The scores are evaluated using a substitution matrix (PAM120 for instance). These segments are then used as seeds, looking for words of a specified size(48), and by extending these words. Since BLAST does not allow gaps, it will in many cases be restricted to more or less small segments. In such a context, the score alone will not be informative enough to decide whether a high scoring pair is significant. These scores will need to be normalized in order to be comparable from a statistical point of view. This normalization takes into account the size of the database and the size of the sequences and gives the probability that a hit occurs by chance. This score is called the E value and is used to rank the hits.

The other popular tool for database searches is FASTA(49). As in BLAST, FASTA by looking for high scoring segments using the Wilbur and Lipman method segments the method is interested in are those having a high proportion of identities are scored by using the main diagonal of a substitution matrix such as PAM considering identities, but giving them a score that depends on the amino acid best diagonals found that way are then re-scored using a full substitution second step, non collinear segments are joined by dynamic programming, using a joining penalty (analogous to a gap penalty). The resulting scores are then used to rank the matches. In order to prevent this chaining step from decreasing it is only applied when the best scoring segment is above some empirical threshold mean and the standard deviation of this distribution are then evaluated and on a final threshold that is used to separate spurious hits from real ones.

Of course, both these methods lead to false negatives and false positives, but they do not use the most accurate method for local alignments (50) that have significantly outperform FASTA(51), and also because some background noise to avoid especially when dealing with very large databases. The reason why behind FASTA are less evolved than for BLAST has to do with the fact that as statistical significance of a gapped alignment is much harder than for an ungapped as in BLAST. This may change soon. Vingron and Waterman have recently published a scheme that allows the estimation of probabilities for gapped alignments(52) very recently, a new version of BLAST, gBLAST(53), has been published which incorporates some of these results and allows statistical ranking of gapped alignments scanning a database. Other statistical scoring systems include the one developed by Bucher(54).

3.2 EVALUATING PROTEIN MULTIPLE SEQUENCE ALIGNMENT

3.2.1 Using Substitution Matrices and Gap Penalties: SP Alignments

Extending the definition of pairwise costs to multiple sequence alignments is complicated. To keep within the framework used for pairwise alignments, a multiple alignment cost which is the sum of the substitution costs (using the same substitution matrices and gap penalties). Nonetheless, based on different scenarios.

Because genetic mutations are binary events which change one protein or sequence into another, substitution costs for multiple alignments are generally terms of those for pairwise alignments(42). Two different approaches have been proposed. The first is to define the substitution cost for a set of elements as the sum of the cost for all pairs of elements chosen from the set(55-57). Thus for three sequences i, j, k the cost of the alignment i, j, k will be equal to the sum of the cost of each pairwise alignment induced by the multiple alignment ($\text{cost}(i,j) + \text{cost}(i,k) + \text{cost}(j,k)$). These pairwise alignments are also called pairwise projections. An alignment defined in this way is called SP for "sum-of-pairs" alignment. In such a context the evaluation of an alignment amounts to measuring the dissimilarities within a set of letters. This approach has the advantage of being straightforward and intuitive, its main drawback is that it has no clear foundation in the theory of molecular evolution. Sankoff proposed an approach in closer agreement with biological intuition. In his model, an evolutionary tree is assumed where each sequence is a leaf. The nodes are occupied by residues. If the tree has k nodes, then substitution costs are defined on k -tuples. The sum of the pairwise substitution costs associated with each edge of the tree defines a tree alignment. It must not be confused with multiple alignments (see Section 4.2.2) which often rely on estimated phylogenetic trees. A tree alignment computation of an approximated SP alignment. In the cases where the tree has only one central node, it is named star alignment, being based on star phylogeny.

Despite the fact that tree alignments are biologically more realistic than SP al have not become very popular, mostly because of the fact that their constru serious algorithmic difficulties. A majority of the multiple alignment meth pairwise substitutions attempt to produce optimal SP alignments. In an defining gap costs is not necessarily straightforward, as can be gathered from alternative schemes (55, 56, 59, 60).

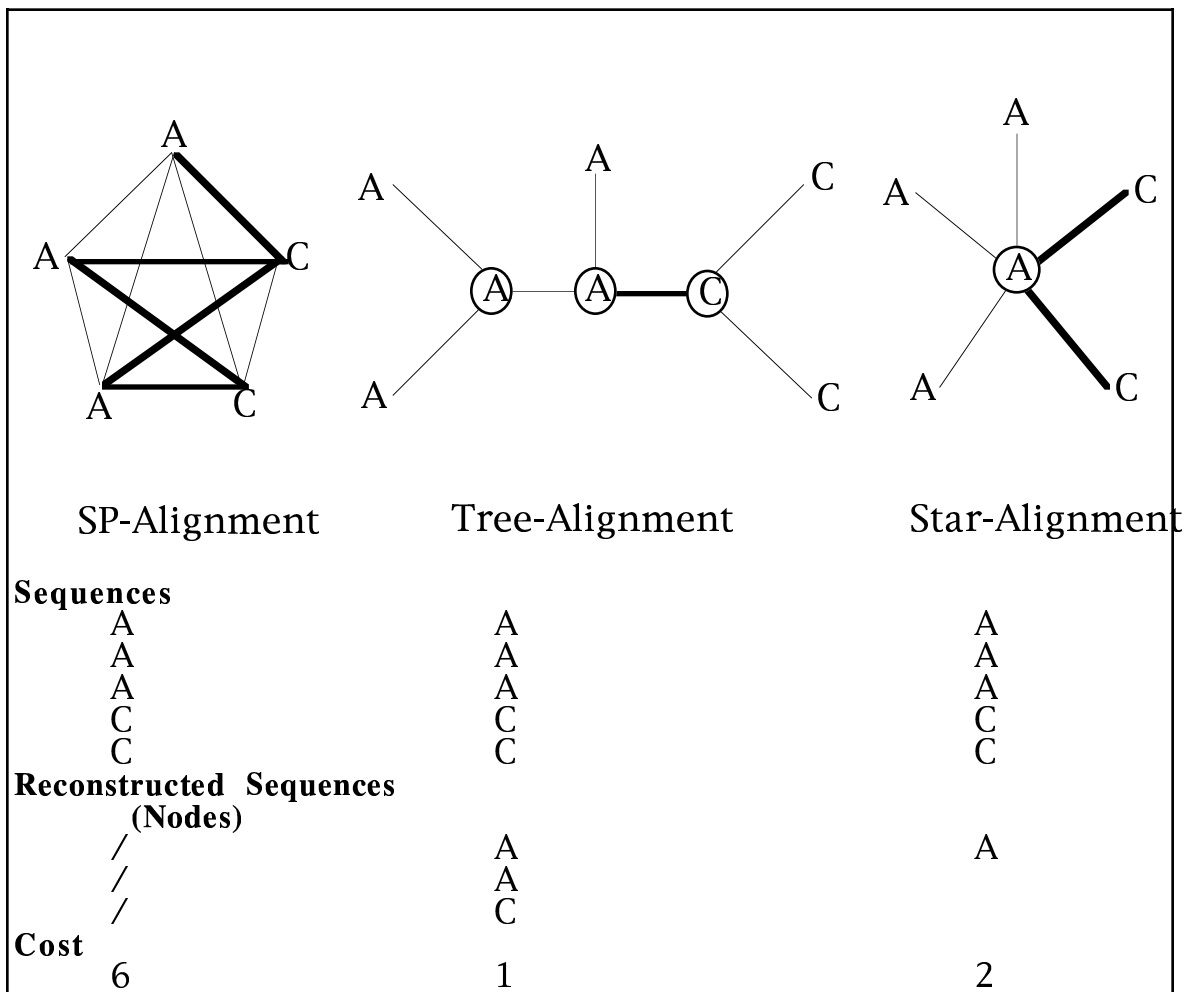


Figure 3. SP, Tree and Star alignment substitution costs for five 1 letter sequences (Altschul, (42)). The reconstructed sequences are indicated by circles at the tree nodes. Lines indicate substitution cost of 0 while plain lines indicate a cost of 1.

The simple implementation of pairwise costs in SP evaluation is known as 'costs'. The idea is to consider the costs of the gaps in each pairwise projection alignment (any column of nulls is removed from the pairwise projection). A gap costs seem to be the obvious companions of an SP alignment, Altschul one to formally propose them(42). Most of the alternative schemes have been for aligning simultaneously three sequences. For instance, Gotoh(56) proposes gap as a set of columns having a null in identical positions. For Murata(55), of adjacent columns, each containing at least a null. The main motivation for these schemes was algorithmic. These approximations were made in order to make evaluation easier when computing an SP alignment. It is for this same reason that Gotoh proposed a simplified version of the natural gap cost named 'quasi natural gap cost'.

Quasi natural gap costs are very similar to natural ones. The main difference is that in pairwise projection is considered, columns of nulls are not removed, and a

counted as opening when a null run in one sequence starts after and finishes a run in the other sequence, such as:

```
Sequence 1 XXXXXX-----XXXXXXXXX
Sequence 2 XXXXXXXX-----XXXXXXXXXX
```

that will lead to considering two gaps when in practice there is only one between sequences. The motivations behind this approximation are purely algorithmic and do with efficiency requirements when implementing the Carillo and Lipman multiple sequence alignment(57) (see Section 4.3.1). This scheme induces alignments that favors similar gaps in aligned sequences, Nonetheless, this approximation because indels seem to be rare events that tend to be kept unchanged through alignments. As a consequence, in most multiple alignments, the number of times where the natural scheme induces alignments different from the natural one should be small. Natural and semi-natural gap costs can also be applied to tree or star alignments.

This type of gap penalties in the context of an SP alignment constitutes one of the most widely used objective functions for multiple sequence alignments. It is often called "sums of pairs with affine gap penalties". Some of the drawbacks of this method have already been discussed in the previous section. The main ones stem from the fact that substitution matrices are general descriptions that do not take into account local information. This is true as well of gap penalties that should incorporate local information. This has been made in ClustalW where penalties are locally reassessed, trying to take into account information from the other sequences being aligned(21). Another weakness of this function has to do with the fact that sequences are considered in pairs only. It makes more sense to consider a column in a multiple alignment as a distribution of amino acids generated by evolution. In the next sections about hidden Markov models (Sections 3.2.3 and 3.2.4), we will present some methods that attempt to take into account the local information when scoring multiple sequence alignments.

Finally, a potential weakness inherent in any scoring scheme is the problem of unrepresentative information. The sequences used for building an alignment are often not representative of the whole set. They are often biased by the composition of the database from which they were collected. In such a case, the alignment of the sequences from an isolated minority will suffer from the fact that the information they contain is not representative of the rest. We will see that several weighting schemes have been designed to overcome this problem.

3.2.2 Weights

With most of the multiple sequence scoring systems, weighting of the sequences is necessary. Weights are designed in order to correct for unequal representation of sequences. For instance, when aligning together globin sequences found in a database like Swiss-Prot(61), large numbers of sequences will be identical, while a few will be quite different from the rest of the set and will have no close relatives. If we want our alignment to be representative of the globin family in general, it must avoid a complete domination by the vertebrate myoglobin and hemoglobin sequences simply because the database contains far more of them. Such an alignment, where each sequence the same weight would be biased. Weights are used to avoid this bias. Several methods have been proposed that can be separated into two groups: they depend on an alignment or an estimated phylogenetic tree.

Alignment based weighting methods do not require the sequences to be representative. Therefore, complex issues of tree topology and root placement are avoided. They can be based on pairwise distances (62) or on the distances from some average sequence (63). Whatever the method, the general trend is similar and resists to the weighting of the sequences which are poorly related to the rest of the set when

which, on average, are more similar to the other sequences have their weight lowered.

The tree-based weights assume that sequences are related through evolution. A reasonably correct tree can be deduced from pairwise distances (64). Two schemes have been proposed: branch-length proportional weights (65), and the Lipman (ACL) weights that are based on a statistical analysis of the tree topology.

In the ACL scheme, a sequence receives a low weight if it is far from the tree and has close neighbors in the tree. ACL weights have the advantage of correcting information without biasing the alignment toward very divergent sequences. The underlying assumption is that although a very divergent sequence contains information, it is hard to exploit such information without bringing in too much noise. The main weakness of the ACL method is that when the topology of the tree is not well established, mistakes can be made regarding the position of the root. The weights by Thompson et al. (65) provide an alternative solution. They also rely on a tree, but under the scheme, sequences only get down weighted for having close neighbors.

In an SP context, applying pairwise or sequence weights to score a multiple alignment is straightforward. The weighted sums of pairwise alignment scores can be formulated. Given N sequences $S_1..S_N$, a weight can be estimated for each possible pair of sequences S_i, S_j . This pairwise weight will be obtained directly through comparison. It will result from the combination of individual sequence weights. Each pairwise alignment $A_{i,j}$ of the sequences S_i and S_j in the alignment has an associated cost calculated using a substitution matrix and a set of gap penalties. Given these definitions, the SP score is equal to:

$$\text{SCORE} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N W_{i,j} * \text{COST}(A_{i,j}) \quad (\text{eq. 2})$$

As with matrices, an important issue for weights is to decide which scheme is applied. Each of these weights has desirable properties and unwanted ones. Vingron and Sibbald proposed a systematic way of comparing five different schemes. Their conclusion was that when sequences are related through a robust phylogeny, the ACL does better than alignment based methods. On the other hand, when the relationship between the sequences is harder to estimate, leading to an inaccurate phylogeny, Sibbald and Argos(63) or related methods(67) are preferable. Similar results were recently established by Henikoff and Henikoff(68) using an empirical evaluation. These authors found that for phylogenetically related sequences, tree based methods are preferable and that the Thompson's scheme slightly outperforms ACL. It is, however, to what extent these findings are method dependent, especially if that most of these weights are used in different heuristic alignment making strategies.

Gotoh pointed out the fact that weighting schemes are very likely to be method dependent(69). For instance, an important difference between the ACL weights and Thompson's ones is that the ACL method produces pairwise weights while the other produces individual sequence weights. As a consequence, the ACL weights contain more information since the pairwise weights they depend on are not necessarily correlated. On the other hand, as we will see later, these two types of weights are used in different contexts. Thompson's weights are mostly used for progressive alignment, while ACL weights are probably very appropriate since remotely related sequences usually have a low weight on the overall multiple alignment(21). ACL weights are used in the program M

Sequence Alignment(22)) which does global simultaneous alignments where e is given a chance to affect the overall alignment.

We will now see that weights can also be useful for the construction and sequence profiles (i.e. generalized alignments used to describe a protein domain).

3.2.3 Profiles

Multiple sequence alignments can be used to provide position specific scores known as profiles(6). The procedure of turning an alignment into a profile is straightforward. It involves counting the residue frequencies in each column of the alignment, and deducing from these measures, a table of substitution costs for each position of the profile. A local cost is also evaluated for gap insertion and a profile refers to the collection of costs associated with each position of a profile. A profile can be treated as a single sequence and aligned to any other sequence using the profile substitution costs and penalties instead of a single matrix.

POS	ALN	A	C	D	E	F	G	H	I	K	L	M	P	Q	...	Gap Penalty
1	EGVL	3	-2	3	4	0	4	-1	3	-1	4	4	1	1	...	9
2	LLSP	2	-2	-2	-1	3	0	-1	3	-1	6	5	-1	3	...	9
3	VVVV	2	2	-2	-2	2	2	-3	11	-2	1	-2	-2	0	...	9
.
.
.
21	SS-D	3	2	5	4	-4	5	0	-1	2	-3	-2	4	3	...	4
22	S--S	2	3	1	1	-2	3	-1	0	1	-2	-1	2	2	...	4
.
.
.
49	SSNY	2	5	2	1	1	2	1	0	1	-2	-2	5	1	...	9

Figure 4. Example of profile (adapted from Gribskov et al.(6)). For each position (POS) a multiple alignment (ALN, presented in a vertical format with each line corresponding to a column) a substitution cost is calculated for any amino acid that would be aligned to that position. A gap penalty is also evaluated. Note that at positions 21 and 22 of the profile the gap penalty is lowered because the alignment used for the profile contains gap positions.

A profile is specific for a family (or a domain). One of the main uses of profiles is in sequence search databases for new members of a family. In such a context the desirable properties are sensitivity and selectivity (i.e. recognize very remote homologues and discriminate against non-homologues). Such a result can only be achieved if the profile induces very good search results. This in turn depends on the quality of the profile itself and can be affected by several factors: (i) choice of the sequences, (ii) method used for building the multiple alignment, (iii) method used to turn the multiple alignment into a profile (especially the treatment of gaps and the method used to describe background frequencies).

In many cases, the choice of the sequences is directly imposed by the database. The best way to remove this type of bias is to use a weighting scheme when building the multiple alignment (see previous section). However if one wishes to build some other type of profile, it is also possible to select the appropriate sequences, using techniques one described by Neuwald et al. (70). Weights also need to be applied when the multiple alignment is turned into a profile. On various occasions, it has been shown that many of the methods used for sequence alignments do as well when used to build profiles, and the level of generalization(65, 71). Accumulation of gaps is another side effect

when many sequences are used to build a profile. Because the number of increase with the number of sequences, schemes have to be used for down effect of their occurrence(65, 72, 73).

Profiles are not only important for database searches, their computation is step for some multiple sequence alignment strategies based on a progressive (74, 75). In this context, to build the full multiple sequence alignment, pairwise alignments need to be aligned in intermediate steps. The best way to do so is to align these alignments into profiles, and to align these profiles with one another. Doing so have been extensively described by Higgins(76, 77) and Gotoh(69,

Finally, since profiles are involved in database searches, some significant work has been done on establishing the statistical meaning of alignment scores(78, 79). This aspect of the problem has received much more attention in the context of the models based approaches that will now briefly be reviewed.

3.2.4 hidden Markov models

A hidden Markov model (HMM) describes a series of observations generated by a stochastic process (a Markov process)(80). They have been used extensively for sequence recognition. HMMs designed for sequence alignments are related to profiles to provide a statistical model representative of a given family of proteins(7). One of the main advantages of HMMs (as opposed to profiles) is that they provide a model directly from unaligned sequences. However, in practice, the methods available for HMM optimization require the computation of multiple alignments. Nevertheless, HMMs have some interesting features that distinguish them from profiles.

A key concept in HMMs is the notion of states. A HMM is a chain of elements representing different possible states. The number of possible states is arbitrarily defined. For instance, there are three states: align, insert and delete. When going through the model, probabilities are given to each possible transition. The values of these probabilities are evaluated by training the model using known members of a family. Sequences can be aligned to a trained model using a variant of dynamic programming known as the Viterby algorithm(80). An alignment between a sequence and a model is called a path, in the sense that it joins different states in order to produce the highest probability. To a large extent, aligning a sequence to a model can be considered equivalent to aligning a sequence to a profile. There is however a fundamental difference. A new sequence is not 'aligned' to a HMM. What is measured is the probability for a given HMM to generate the optimally aligned sequence (i.e. the sequence with the right pattern of gaps/unaligned residues/aligned residues).

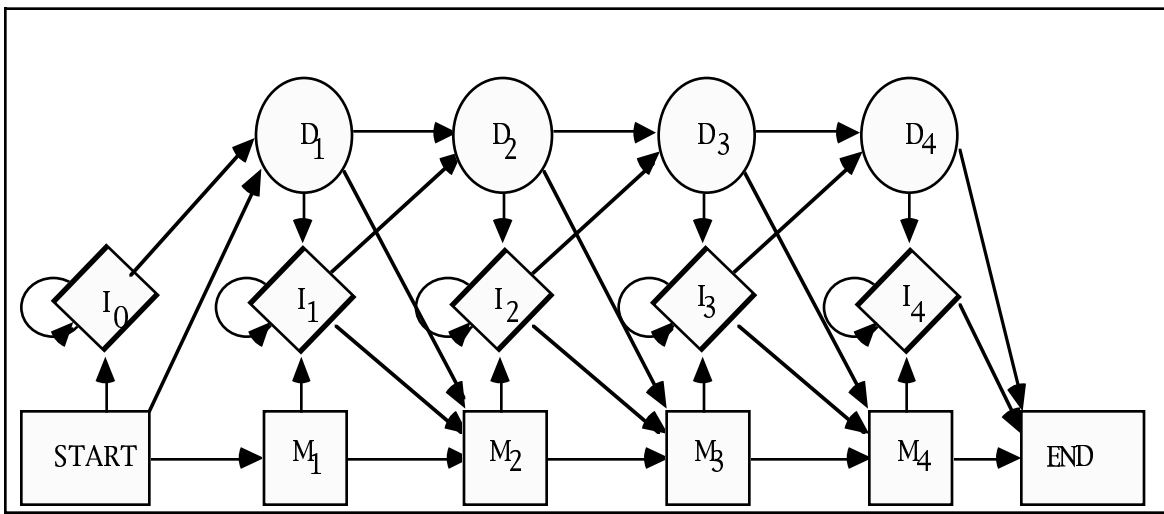


Figure 5. A linear hidden Markov model (from Hughey and Krogh, (82)). This model has three different states (M, I, D). Each state is connected to the others by a transition probability (arrows). Assigning the weights to each transition is the purpose of the

The number of sequences, and their range of identities are critical factors that affect the model. In their simplest expression, HMMs do not require any prior information, unlike profiles that required a multiple alignment made using a substitution matrix. In HMMs, residues are described as 'letters' and the training only relies on observed sequences to establish the parameters. If there are enough sequences in the training set, then a sensible model can be established since the substitution constraints will be 'discovered' by the model on a position per position basis. The accuracy and the sensitivity of a model trained on a small set will be highly dependent on the number of sequences and their information content. To overcome possible lack of information (missing data), pseudo counts are used in order to simulate background frequencies (82, 83). Their influence on the model is directly related to the amount of information present in the sequences.

The actual values of the pseudocounts can be estimated using various methods. One method is to measure the probability of each amino acid in the training set, or other methods use probabilities from a standard substitution matrix. Generally speaking, the more sequences used for the training, the more accurate the values of the parameters. In this context, Dirichlet mixtures (36) proved extremely useful. A Dirichlet distribution is a mathematical tool that, given an observed amino acid distribution and a set of expected distributions, allows the computation of a probability for the observed distribution in a hidden Markov model context, these mixtures can be regarded as the equivalent of a substitution matrix. They have been shown to be more sensitive to sequence variation than traditional substitution matrices (36).

As with profiles, HMMs can be used to generate multiple sequence alignments and scan databases. Scoring can be made by combining the probabilities of all possible alignments of a sequence to a model, which is equivalent to calculating the total probability of a sequence given the model. This can be done efficiently using the Viterbi algorithm (80). Such a score is called the NLL score for Negative Log Likelihood score. The NLL score measures how far a sequence is from its model (in other words, the standard deviation of a given model to produce an aligned sequence). The problem with NLL scores is that they depend on the size of both the sequence and the model. One way to normalize the NLL score is to measure the Z score, the number of standard deviations a NLL score is above an average NLL score of unrelated sequences of the same length. A complete algorithm for the computation of Z scores is described in (82).

This study of multiple sequence alignment scoring systems is far from exhaustive. A variety of alternative methods have been described that roughly fall into

categories: those relying on SP evaluation and those (like HMMs) that use distributions of amino acids rather than pairs. Although it seems that the latter is a more realistic approach, the main reason why the SP scheme has been more popular has mostly to do with algorithmic problems associated with sequence-based methods.

Both methods only deal with one aspect of the problem: the use of local information. We know that since proteins fold into active 3D structures, there must be more interactions between the sequences (i.e. tertiary structure interactions...) than what we discussed in an appropriate manner. There is no doubt it could help to improve the alignments. Few methods have been proposed to deal with these non-local interactions. There are two good reasons for that: this type of signal is usually very weak in the algorithmic problem is even more complex than when taking into account long-range sequences. We will now see that the problem is different with RNA hence the complex types of objective functions.

3.3 RNA ALIGNMENTS: TAKING INTO ACCOUNT NON LOCAL INTERACTIONS

When reliable non local interaction information is available, it makes sense to include it into the scoring scheme. This is rarely the case with proteins, hence the problem is encountered when doing structure threading (threading a protein sequence onto a structure)(84). Fortunately, in the case of RNA the problem is different and the formation of non local interactions are better understood(85, 86). The secondary structures encountered in RNAs are due to Watson and Crick base pairing. The existence of mathematical models that describe these interactions on a thermodynamic basis make RNAs good candidates for 'ab-initio' folding predictions. However, the parameters influence the structure of an RNA molecule (local conditions, unknown tertiary interactions...) that govern accurate predictions. Thermodynamic models are still very imperfect despite some recent improvements. This does not mean that the thermodynamic approach is wrong, but it may lack information it uses is not sufficiently detailed considering our knowledge of the folding process. However when combined with phylogenetic information (such as taken from multiple sequence alignments) predictions can be realized to a high accuracy(16).

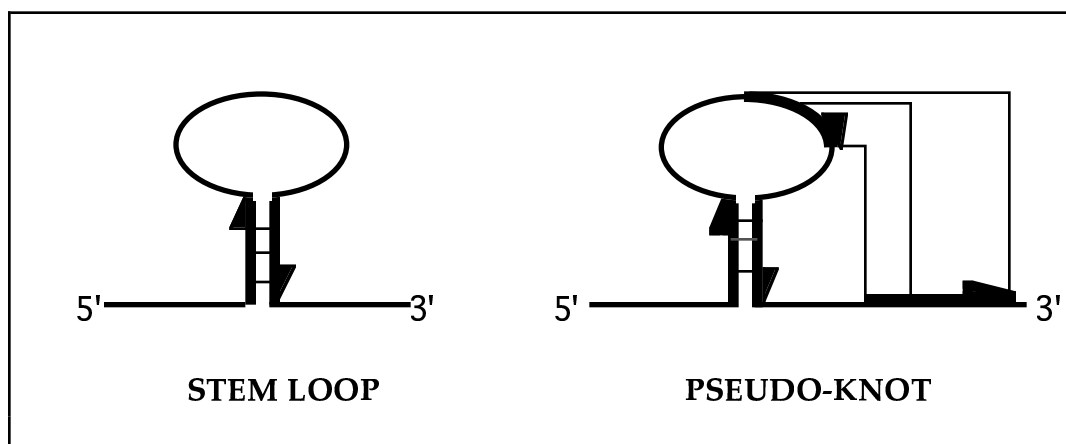


Figure 6. Some of the motifs commonly found in RNA secondary structure. Base pairs are usually made through Watson and Crick interaction although non canonical pairs often been reported.

Multiple sequence alignments have the advantage of revealing constraints with respect to hypotheses on their origin. Such analysis, performed on ribosomal RNA sequences,

made it possible to predict the secondary structures of these molecules with outperforming traditional energy minimization schemes like Zuker(89). Unfortunately these types of RNA alignments are difficult to build automatically due to algorithmic problems.

Several functions have been proposed that incorporate RNA secondary structure information into their evaluation of multiple sequence alignment quality. One approach, Kim et al. (90) proposed a function that computes the probability of potential secondary structures contained in a multiple sequence alignment taken into account. It is a scheme that has the advantage of being very flexible (for instance pseudo-knots) and not requiring the actual computation of a secondary structure (structures are not assessed for compatibility). Its main drawback is that evaluation time increases with the length of the alignment. This can prove a severe limit when dealing with long sequences (like ribosomal RNAs) while using an iterative or stochastic optimization such as simulated annealing. Eddy and Durbin(91) proposed a different type of approach. In their method, the secondary structure is expressed as a binary tree in which each node stands for a column in a multiple sequence alignment. This tree can be searched through a generalized HMM (HMM with bifurcation) named Covariance Model (CM). CM can be trained like an HMM. Once this has been done, the sequences are then aligned to the model in order to produce the multiple sequence alignment. This approach is similar to the stochastic context free grammar (SCFG) methods(93, 94), which use regular expressions to express the structure using a special type of regular expression. The alignment is done by parsing the sequences through the proper expression that has been obtained from the model. CM and SCFG methods suffer from the same drawback. They do not allow nested structures and cannot take into account pseudoknots(95, 96) as the method proposed by Kim. Furthermore, their computation is very expensive for long sequences (>200 nucleotides). An option for decreasing the complexity of the problem is to do threading: assume that some master structure is known and thread the sequences onto it. In several important cases, like ribosomal RNA, this is a realistic assumption.

This approach can of course be taken using a SCFG based objective function. Alternatively, one can consider a simpler function such as the one described by Michot(92). In this case, the evaluation of the alignment is split in two steps: (i) the primary sequence alignment (ii) evaluating the quality of the fold in the sequence of known structure onto the sequence of unknown structure. To obtain an overall score, these two terms are combined with one another. Although this approach lacks a real theoretical justification as opposed to those previously described, it has the advantage of being conceptually simple. It can also accommodate a range of interactions such as pseudoknots and other non nested structures. In this case, the alignment problem is NP-complete(17). In order to provide a reasonable heuristic solution for long sequences (>1000 nucleotides) that we developed the package RAGA(97) (See Appendix 3).

4 MAKING MULTIPLE SEQUENCE ALIGNMENT

4.1 COMPLEXITY OF THE PROBLEM

So far, we have focused on reviewing some of the objective functions evaluating the quality of sequence alignments. However, as pointed out earlier one side of the coin. The other one, that constitutes in fact the main problem of optimization. In other words, given an objective function, is it possible to produce the best scoring alignment? There are at least two good reasons why efficient and accurate optimization strategies. The first one is obvious: alignments that are needed for whatever purpose.... The second reason is less extreme importance. The evaluation schemes described above are only theoretical using phylogenetic or structural criteria. As such, they do not constitute a must therefore be validated through empirical analysis (i.e. how well do they

The optimization methods required for these two reasons do not necessarily equivalent. One can, for instance, use a very robust but expensive (compute memory) method to compare and validate alternative scoring schemes. If a scheme proves useful it may later become appropriate to develop a very simple method that approximates the optimization reasonably well while being efficient for production purpose. Needless to say, whatever direction one wishes to take, an optimization technique will always prove to be a very demanding problem

We already mentioned that even for two sequences of moderate length, a naive alignment computation can lead to impractical enumeration problems. Fortunately, this situation does not have to be that bad and will depend on the scoring scheme used to optimize. In many cases, there are short cuts that allow efficient computation of multiple sequence alignment, given some specific objective functions. We will see in section 4.2 that dynamic programming (18) is one of these techniques that allows the computation of pairwise alignments in time proportional to the product of the length of the two sequences. This essential technique constitutes the core of many alignment methods. It is not restricted to two sequences, but since its complexity is a function of the length of the sequences to align, it can hardly be used for more than three sequences (55). This does not mean multiple sequence alignments cannot be computed automatically with dynamic programming, but it means that to do so, one must use heuristic algorithms.

Heuristic methods do not guarantee an optimal solution but may sometimes even guarantee the solution to be inside given boundaries. Generally, multiple sequence alignment algorithms can be divided into two classes: (i) algorithms that usually rely on sequence clustering algorithms and dynamic programming for making progressive alignments (ii) the non-progressive algorithms that simultaneously align all the sequences. These non-progressive algorithms can be divided into two distinct sub-categories:

- deterministic heuristics.
- stochastic heuristics.

In the following sections, the underlying principles of these algorithms and the differences will be briefly explained. More emphasis will be given to the general techniques on which the SAGA package is based (98) (see section 5.1 and appendix

4.2 DETERMINISTIC GREEDY APPROACHES

4.2.1 Aligning Two Sequences

The main algorithm for aligning two sequences, often referred to as the Needleman and Wunsch(18) or dynamic programming (DP) algorithm, is one of the oldest important tools in bioinformatics. Over the last 30 years, it has been used in another in most of the methods developed for sequence comparison.

When applying dynamic programming to two sequences, it is possible to compare scoring alignment between these sequences using an amount of memory proportional to the product of the lengths of the two sequences to align. This is a dramatic improvement over the naive approach that would require enumerating all possible alignments. An important advantage of dynamic programming is the general scheme. Given substitution costs (e.g. matrix or profile...) and a scoring gaps, the algorithm can compute the alignment with the best score. It can accommodate any context-independent scoring scheme.

The algorithm is based on extending recursively the best scoring alignment. When residues of each sequence have been aligned. In practice, this means finding the best path through a matrix constructed from the scores of all pairs of elements between sequences. Let us consider two sequences, A of length m and B of length n, assigns a score $d_{i,j}$ to the substitution of residue i in A by the residue j in B with a penalty g. Computation of the optimal score is achieved by incrementally extending the best path with a locally optimal step. For example; element $d_{i,j}$ can extend any path in the preceding row ($i-1, m:m < j$) and/or the preceding column ($n, j-1:n-1$) with a penalty. If the extension that produces the highest new path score is always chosen, applying the condition repeatedly (to every element in the sequence matrix) can be expressed more formally by the recursive expression:

$$S_{ij} = d_{ij} + \max \{ (S_{i-1,j-1}), (S_{i-1,m} - g, (m < j-1)), (S_{n,j-n} - g, (n < i-1)) \} \quad (3)$$

Once the global score S_{ij} has been found, the optimal alignment can be retrieved by a traceback from the cell with the highest score. This algorithm is slightly more complex than the original Needleman and Wunsch that did which used simpler gap penalties. Computation is quadratic in time and space for the score. Gotoh propose a version of this algorithm that allows computation of score in linear space using affine gap penalties(45). Finally, a recursive algorithm originally described by Hirschberg(99) and refined by Myers and Millers(100) allows computation and alignment in linear space, using affine gap penalties. A vast number of variations have been proposed around the original algorithm. For instance, since it is known that a mathematically optimal alignment is rarely biologically optimal methods have been described for the computation of sub optimal alignments(101-103). Such computations can be useful when trying to assess the reliability of a sequence alignment(104). Another variation proposed by Taylor, involves biasing the DP toward motifs, in order to minimize the gap penalty choice(47).

DP is not restricted to global alignments but can also be used for finding the longest common subsequence shared between two otherwise unrelated sequences. This algorithm is extremely important for database searches. It is mostly an adaptation of the Needleman and Wunsch algorithm and involves extending a path as long as the alignment is improving (hence the 0 in the following equation):

$$S_{ij} = \max \{ (S_{i-1,j-1}) + d_{ij}, (S_{i-1,m} - g, (m < j-1)) + d_{ij}, (S_{n,j-n} - g, (n < i-1)) + d_{ij}, 0 \}$$

The score of a path is set to 0 if this path gets a negative score (the substitution needs to give negative values to unfavored substitutions). Best segments are

finding the best scores in the score matrix and making tracebacks, using 0 for the path.

The use of DP is not restricted to sequences. For instance, the Viterby algorithm aligning a sequence to a HMM(80) is an adaptation of the classic DP algorithm. Modified versions of DP can be used to align structures. This was shown by Michot who described a DP strategy for aligning two RNA molecules taking into account potential secondary structures in one of the sequences while knowing the structure of the other(92). Taylor and Orengo(105) also described a heuristic based on dynamic programming designed to align two protein structures (double dynamic programming). Finally, the algorithm used to align an RNA molecule to a covariance model is on DP and is very similar to the one described by Zuker for predicting RNA secondary structure. Most of these algorithms are computationally very demanding (higher complexity than regular DP). For this reason, alternative techniques (like stochastic optimization) can be sensible alternatives or even necessities when dynamic programming is used.

Nonetheless, in the case of pairwise sequence alignments DP may provide an efficient and accurate way to compute a solution, as long as traditional gap penalties are used. We will now see that how this pairwise method can be extended to multiple sequence alignments.

4.2.2 Aligning Two Alignments : Progressive Alignment Methods

Although it is hard to align more than two sequences at a time by regular dynamic programming, it is possible to align two alignments (i.e. two sets of pairwise alignments) with a pairwise algorithm. This is due to the fact that an alignment (pairwise) can be regarded as a generalized sequence where instead of having one position, one has a residue vector. Such a definition allows the alignment of two alignments, treating them like normal sequences(69). It makes sense when doing each alignment as a profile.

These methods for aligning multiple alignments are very important because they lead to one of the most popular multiple alignment strategies: the progressive alignment algorithms described by Feng and Dolittle(74) and Taylor(75). The principle of progressive alignment is quite straightforward. Since it is impossible to perform multiple sequence alignment by DP, using all the sequences at the same time, one begins by aligning pairs of closely related sequences. Such pairwise alignments with very similar sequences are likely to be correct if the sequences are similar. In the second stage, pairs of closely related pairwise alignments will then be aligned together, and so on until the whole multiple sequence alignment has been produced. The order in which the sequences are aligned depends on their phylogenetic relationship (phylogenetic tree) and the procedure will follow the topology of some estimated phylogenetic tree. Several methods exist that allow the computation of such trees using various algorithms that rely on an estimate of the pairwise distances between the sequences(106).

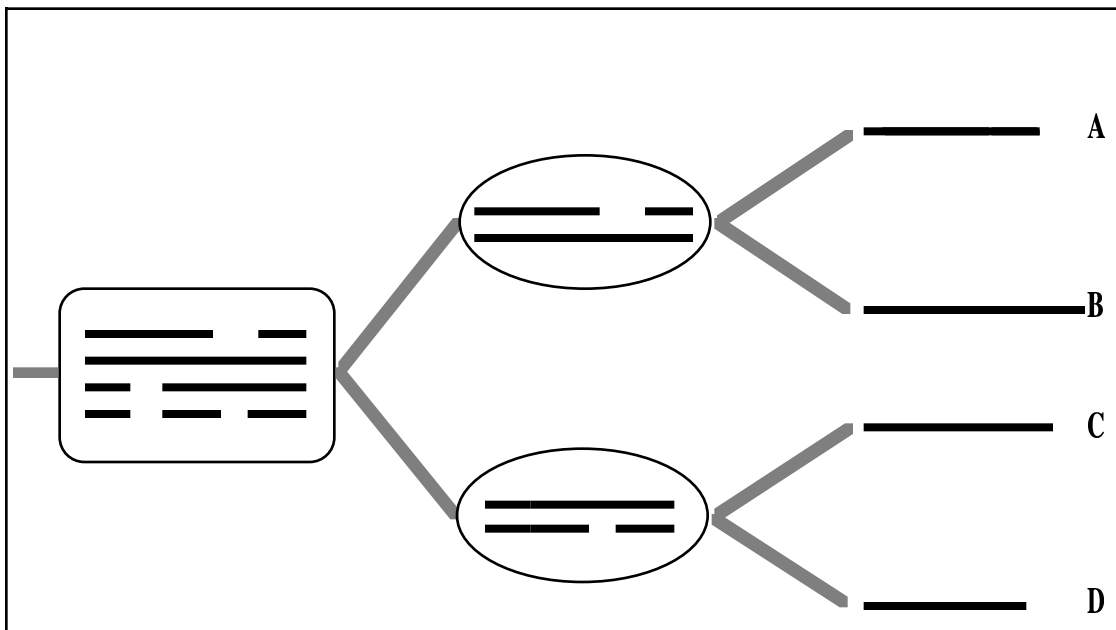


Figure 7. Progressive alignment strategy(74, 75). A phylogenetic tree is estimated for four sequences A, B, C and D (gray lines). Its topology is used to decide the order for the sequences by dynamic programming.

The advantage of progressive alignments is that they provide an efficient way of overcoming the problem of computational complexity posed by global alignment techniques. However, if the aim is, for instance, to optimize the sums of substitution matrix and a pair of gap penalties, one has to be aware that this method does not offer any guarantee. Its performances may be unpredictable and depend on the quality of the intermediate alignments. These will depend on the relations between the sequences and the accuracy and the density of the tree. Alignments are not necessarily correct (in terms of optimization) because they use only a subset of the information available. This may have serious consequences: early mistakes will never be corrected ('once a gap always a gap') and may be compounded by inducing more mistakes in further intermediate alignments. This problem is referred to as a 'local minimum problem', in the sense that the method by being pushed toward satisfying short term constraints that may lead to a non global

Nevertheless, in many cases, this progressive strategy leads to convincing alignments at a low computational cost. There are two good reasons for that. First of all, in many cases the sequences are well fitted to a progressive approach, and secondly because local optimality does not necessarily mean better biological alignments. This explains why the most popular multiple sequence alignment packages are based on this strategy. They include Clustal V(77), Multal(75), ClustalW(21), Barton and Stenberg(10). In a wide variety of cases these methods have been shown to do well when their alignments are assessed using sets of reference alignments. Many of the techniques used for multiple sequence alignments such as position specific gap penalties, tree based gap penalties, secondary or tertiary structure information can be implemented in a progressive strategy as shown in ClustalW(21) and in the work of Barton and Stenberg(10).

One of the problems of progressive alignments is that their accuracy will clearly depend on the relation between the sequences aligned. If these are closely related, progressive alignments are likely to be accurate and to induce a correct overall alignment. On the other hand, if the sequences are only remotely related, it may be necessary to simulate a global alignment. All the information they contain to build a realistic model (i.e. most of the information needed to build the intermediate alignments will not contain enough information for inducing a correct pairwise alignment). The only way to improve the relation

the sequences in a set is to incorporate some more sequences into this set. In some cases this is not an option, simply because the sequences are not present in the set and may not even exist. Simultaneous sequence alignment methods are there. We will now describe methods that attempt to produce this type of multiple alignment.

4.3 DETERMINISTIC APPROACHES FOR NON PROGRESSIVE MULTIPLE ALIGNMENTS

4.3.1 The Carrillo and Lipman Algorithm

The basic DP algorithm is hard to apply to more than two sequences as the computation of an SP multiple alignment using DP has been shown to be NP complete(19). In order to compensate for these limitations, an algorithm was developed by Carrillo and Lipman(57) to perform such operations on a larger number of sequences (up to 10). It relies on the idea that the whole alignment space does not necessarily need to be explored when aligning sequences. Bounds can be derived that guarantee optimal alignment inside a portion of the hyper space defined by the sequence similarity. Several methods have been described for deriving guaranteed boundaries. Unfortunately, these boundaries are often too loose to be practical (i.e. they are too large to be explored). To be of any use, the Carrillo and Lipman algorithm has to rely on tighter heuristic boundaries that do not guarantee optimality but are implemented in the package MSA(22, 109). Their main property is that they are based on the level of similarity of the sequences. For very closely related sequences, the boundaries are extremely tight and define a very small portion of the hyper space. If the similarity decreases, the boundaries become looser, until the computation requires the exploration of a space beyond computational ability.

The limit on the number of sequences and on the level of similarity that a program can handle constitutes the major drawback of MSA. It means that in most cases only alignments that are within the scope of the program are those for which progressive alignment can be made because the sequences are related enough. An important drawback of MSA is that it is restricted to a very specific type of gap penalty function. This is especially drastic with regard to the type of gap penalties used ('semi natural gap penalties' instead of 'natural gap penalties', see section 4.3.2).

4.3.2 Other Approximation Techniques

More recently, non-dynamic programming based methods have been proposed to generate multiple sequence alignments by solving graph theory problems. One technique is known as the MWT (Maximum Weight Trace) formulation, introduced in 1993 by Kececioğlu(110), (111). Like dynamic programming, the solving of MWT is known to be NP complete. But even so, using a branch and bound algorithm based on MWT has been described(111) that manages to align up to 15 sequences in a reasonable amount of time. Unfortunately, this approach suffers from the same problems as MSA: it requires tight boundaries to be established, which is not always possible, especially when the sequences have very low similarity.

Considering all this evidence, it is clear that in the years to come, our ability to handle multiple sequence alignments will increase along with computer power. The methods that are able to handle more and more sequences. On the other hand, the problem, that in most cases has been shown to be NP complete is likely to remain unsolvable in polynomial time. This implies that the number of sequences a method is able to handle will always remain severely limited. Another serious problem is the fact that in many cases these algorithms are very specific for a given objective and difficult to adapt. We will now focus this review on stochastic techniques that have been proven to be much more flexible and versatile tools.

4.4 STOCHASTIC HEURISTICS

4.4.1 What is a stochastic Method ?

A stochastic optimization strategy (also known as Monte Carlo simulation) finding a solution to a problem through some form of random sampling, which the search is random mostly depends on the heuristic one uses. The randomness is that there is a non null probability for any potential solution regardless of the solution space size. Of course, the randomness also implies solutions may not be looked at (including the global optimum). In order to solve a problem, a large number of heuristics have been designed that attempt to explore the solution space. The aim is to improve the chances of finding an optimal solution. As a consequence, most stochastic strategies can be regarded as a trade-off between greediness and randomness.

This can best be explained considering a solution landscape with hills and valleys. The optimal solution is the top of the highest hill (or the bottom of the deepest valley). If there is only one hill, one can find this point by simply climbing along the gradient. Such a method does not have to be stochastic and is known as hill climbing. However, if there are several small hills and a high one, the outcome of hill climbing will be restricted by the point where it was started. The action of climbing the hill will not necessarily lead to the optimum, but to the first local optimum. This is the problem of the excessive greediness of a hill climbing strategy that always goes up. If an element of randomness that decreases the greediness of the strategy, it may be possible to take alternative routes (going down, for instance, which in turn leads to search to encounter the highest hill). This less greedy technique is known as Stochastic Hill Climbing. Ultimately, its success (as for any stochastic strategy) will mostly depend on the fitness landscape. The more complex this landscape, the harder it is to find the global optimum.

Stochastic methods are quite well suited for solving sequence alignment problems where the landscape is very complex (113) and the number of solutions is too large to be exhaustively analysed (19). Many of these techniques are inspired from physics (Simulated Annealing), biological (Genetic Algorithms, Evolutionary strategies), or even social (Ant Colony search) phenomena, others are simply based on iterative statistical analyses (stochastic expectation-maximization strategy). In this section, we will discuss some of the applications of stochastic heuristics to the problem of sequence alignment. Since one of these techniques (Genetic Algorithm) is central to the work presented here, it will be given a more detailed presentation in the next section.

A very important advantage of many Monte Carlo strategies over more traditional hill climbing is the fact that they often allow conceptual separation between optimization and evaluation. Ideally, the optimization strategy will play the role of a black box in which the objective function can theoretically be plugged. The main drawback of these methods is that they rarely give any indication of how close to the optimum they are. In fact, if these methods are iterative, it is usual practice to stop them if they reach a specific number of iterations. The general lack of guaranteed criteria for stopping is usually the main problem. It can sometimes be overcome if the score obtained from the objective function contains some information regarding the completeness of the optimization.

4.4.2 Iterative alignments and Expectation-Maximization Strategies

Iterative strategies are heuristic approaches, very similar in spirit to Expectation-Maximization (EM). Expectation-Maximization is a Hill climbing strategy that involves two steps: (i) given a model and some data, estimation of new parameters

model, (ii) given the new parameters and the ancient model, estimation of The procedure is carried on iteratively until the model and the parameters st between the model and the parameters). Depending on the implementat incorporate some stochastic steps as do iterative alignment strategies. The strategies is quite large. The optimization of RNA covariance models is m EM(91), as well as the training of HMMs in SAM(81). Several multiple s alignments strategies have also been described based on EM(114, 115)

To perform an iterative multiple sequence alignment, two components are n prealigned sequences and an algorithm for aligning pairs of sequences alignments) such as DP using Gotoh's operators for aligning profiles and al 73). An iterative alignment algorithm usually takes the following steps:

- 1-split the multiple sequence alignment into two groups of prealigned s
- 2-apply the DP algorithm to the two multiple alignments
- 3-if the score is not stabilized, go to 1

Since the number of possible partitioning in step 1 increases factorially with sequences, exhaustive analysis cannot be done, and one has to use random other types of partitions that use for instance the phylogenetic tree. Sever schemes of this type have been described (for a review, see (116)) These better on average than progressive strategy (in terms of scores) but are also (100 times on average). Recently, Gotoh presented such a technique named randomized iterative strategy(117). He uses it to optimize the weighted sums reevaluation of the weights after each iteration. According to the author, a reference structural alignments, this technique is significantly more ac progressive alignments strategies.

Gibbs sampling is a more sophisticated example of stochastic EM(118). In Gib applied to multiple sequence alignment(83), a random solution is generate guide the generation of the next solution (ungapped multiple alignmen probabilities to all the neighboring solutions, and randomly choosing one of exactly, choosing a new solution according to their probabilities, given the cu The search is iterative and has been successfully implemented for identify motifs conserved among a set of sequences. In such a context, given a set of sequences, the aim is to generate the more informative model (the less likely obtained by chance). The objective function described in that context by Law purely based on statistical analysis of the data and can be made fully indep prior information. In many respects it is very similar to HMMs.

4.4.3 Simulated Annealing

One of the first stochastic techniques described was simulated annealing (SA SA relies on an analogy with physics. The idea is to compare the solving of an problem to the cooling of a metal (i.e. finding the best position of each atom metal is equivalent to solving a multi constraints problem). The princi straightforward. Given a function, a random solution will be generated, and chosen. Every iteration, a new solution is created based on the previous one random modification algorithm). If the new solution is better than the old accepted. Otherwise, it will be accepted/rejected with a probability that de temperature and the level of disimprovement (i.e. the higher the tempera likely a solution is to be accepted, regardless of its quality). Temperature wil cooled down. This means that at the beginning, almost any new solution is ac in the last phases (low temperature) only the solutions bringing an impr accepted. The SA algorithm can be summarized as follow given that S_c is solution at the n^{th} iteration:

- 1) Generate an initial solution $S_c=S_0$ and an initial Temperature (T_0), T
- 2) Modify the solution S_c into S_n
- 3) IF $F(S_n)$ is better than $F(S_c)$ then keep the new solution: $S_c=S_n$
ELSE
keep S_c with a probability $P=e^{-(|F(S_n)-F(S_c)|)/T_c}$
- 4) Change the value of T_c according to the annealing schedule
- 5) Go to 2 as long as the termination criterion has not been met.

The strategy depends on three parameters: the initial temperature, the cooling function, and the acceptance function. The cooling and the acceptance function define the SA. The most common are those using Boltzmann distributions, but one can use Gaussian or other ad-hoc distributions(121). A remarkable property of SA is that if the cooling function is slow enough, the search is guaranteed to reach the optimum. However, this argument remains theoretical since in most of the cases an adequate cooling schedule would require too much time to be of any practical use (hence the name Simulated Annealing).

Despite this intrinsic limitation, SA has been applied to multiple sequence alignment on several occasions(122-124). The conclusion of these studies has mostly been that it does reasonably well, because of its slowness, SA has to be restricted to a local alignment improver method rather than a full alignment method. In all these applications, SA was applied to sequence alignment in a similar fashion: a multiple alignment was generated (randomly or using some heuristic) and modified using specific sub-routines to delete gaps around or insert them. We will later see that these 'modification procedures' are in many ways similar to the mutations used in a genetic algorithm strategy.

As opposed to the iterative methods previously discussed, SA is not restricted to a single SP multiple alignments using DP. On the contrary, the method displays the properties of a black box. If it was not so slow, SA would possibly be the idea for solving any optimization problem. It has successfully been applied to the alignment of RNA molecules using potential secondary structure information(90). Similarly, SA-based methods(125) have been described for predicting the fold of very long RNA molecules, which are beyond the scope of traditional methods like Zuker's. Finally, a variant of SA for sequence maximization based on SA is used for the training of HMMs(126). In the work presented here (SAGA and RAGA), SA has been an important source of inspiration, because of the similarities that exist between SA and GAs.

GAs are probably some of the most interesting stochastic optimization tools available today. They can best be described as a very flexible framework in which any available methods (deterministic or not) can be integrated in order to construct a powerful tool. One of the reasons why GAs have received so little attention in the context of sequence alignment is probably due to the fact that the implementation of a GA for sequence alignment is much less straightforward than for simulated annealing. In other areas of computational biology, GAs have been established as powerful tools. These include protein 3D structure prediction and secondary structure prediction.

4.5 GENETIC ALGORITHMS

4.5.1 What is a Genetic Algorithm?

Genetic algorithms are based on a loose analogy with the phenomenon of natural selection. They have existed in one form or another for quite a while, but were first introduced by Holland in 1975(127). The principle is quite straightforward: for a given problem, a set of potential solutions (population) compete with one another.

These solutions can be modified (mutations), or combined with one another. The idea is that acting together, selection and evolution will lead to an overall improvement of the population. Most of the ideas developed here about GAs are taken from

There are two essential concepts at the heart of the GA strategy. The first one is Selection, which is established in order to lead the search toward improvement. The best solutions to the problem (as judged with the objective function) are to be selected according to their quality. If one was to do so in a deterministic instance, by only selecting the best solution every generation, the search would rapidly converge toward the first local optimum it encounters. In such a case, algorithms mostly behave like Hill Climbing. To avoid this major problem (called premature convergence), the selection procedure in a GA is not absolute but statistical. Statistical selection is a very straightforward process. In a first step, each individual is evaluated using the objective function. The score obtained that way is turned into a fitness measure. In the selection round, an imaginary wheel is spun where each slot has a number of slots proportional to its fitness. This means that individuals with a high score are the most likely to be selected, while those with a low score are less likely. However, everything is possible and it is this uncertainty that prevents the search from converging prematurely to the closest local minimum.

The second key concept in a GA is the concept of evolution. The aim of evolution is to create new solutions based on those present in the population. By analogy to biological evolution, there are two sets of operators: the crossover (combining two individuals) and the mutation. Since crossovers can be regarded as a very primitive interpretation of sexual reproduction, it is current practice to apply selection when choosing the two individuals to be combined. When doing so, one hopes that the child produced that way will inherit the best qualities of both parents.

In practice, a genetic algorithm follows a series of cycles known as generations. In each cycle, individuals are evaluated and used to create the next generation through selection and operations. The variations one can put inside such a scheme are infinite. Different schemes can be used for turning a score into a fitness measure. Parameters can be made stronger or weaker. Many of the effects of such parameters have been studied in detail. However, there is no solid argument for one model of GA to outperform others. In fact, the choice of a model seems to be mostly problem dependent. Our experiments show that if the problem is suited to combinatorial optimization, a GA does reasonably well. However, some do better than others and the appropriate GA needs to be determined in what is mostly a trial-and-error strategy.

There is very little theory around the reasons why GAs perform efficient optimization in some instances, as compared to SA, there is no formal proof that a GA will reach a global solution given enough time. The most popular concept to explain the performance of GAs is the notion of building blocks. The concept is best understood if we consider a coding system that consists of binary strings (i.e. each individual is a chromosome where each gene can have two allelic values 0 or 1). In such a context, a block is defined as a stretch of 0s and 1s present in a chromosome. The GA strategy through selection and combination makes it possible for important blocks to be selected (i.e. increase the number of their copies in the population). This in turn will increase the number of their copies through mutations and crossovers. The more fit a block is (i.e. the better the score it induces), the more likely it is to spread in the population. This would be the equivalent of the fixation of a group of alleles in an animal population. In a population context, mutations help create new blocks, or restore blocks that may have been lost. In fact, the number of existing blocks is much larger than the population itself, a phenomenon known as implicit parallelism.

This GA procedure can also be looked at through another angle. If we consider a solution as a vector in a hyperspace, with each piece of the chromosome being a coordinate, we can view the search as a form of multivariate analysis. In

building blocks are sets of coordinates that restrict the search to some small hyperspace that has a higher concentration of good solutions than the rest. When it goes on, this hyperspace becomes smaller and smaller until it cannot be reduced.

The theory of the building blocks was initially proposed for the simple Genetic Algorithm developed by Goldberg (128). For this reason, it is restricted to binary chromosomes where each gene only has two allelic values (1 or 0). It is hard to extrapolate the more complex representations required by sequence alignment problems. Empirical evidences suggest that the building block theory constitutes a good approximation when analysing the behavior of more complex representations. It should also be stressed that although it provides an elegant way to answer the question of how a GA works, the theory of the building blocks is useless for predicting the type of problem the GA approach will fail or succeed.

Deceptivity is an area of GA research that has received some significant attention in the last few years. It amounts to defining the conditions in which a GA will fail to find a correct solution. What these studies have shown is that like any optimization algorithm, a GA is sensitive to the fitness landscape of the function one is interested in. It is not what is established what would intuitively be assumed. When there is very little complexity in the function one is interested in, the optimization becomes much harder or impossible. On the other hand, given a complex problem, there are often alternative ways to explore the landscape. In fact, the landscape will depend on the representation of a solution and the means one uses to walk along the landscape (operators in our case). When using binary coding, methods that attempt to reshape the fitness landscape in order to make it more continuous are known as gray coding. More generally, whatever the problem representation, efforts should be made so that neighboring solutions have comparable fitness. Defining the neighborhood of two solutions is a complex task that depends on two factors: the representation system and the operators used. The challenge is to have a representation that induces a very smooth fitness landscape if the operators are as well designed to sample close neighbors...

For instance, in the case of a multiple sequence alignment, there are many ways to define such neighbors. The neighbor of a given alignment will be another alignment created when inserting a gap or when shifting an already existing gap. The complexity of the landscape explored during the search will depend on the shape of the fitness function and the operators these operators define (i.e. do they induce any gradient that can be followed?). For example, a mutation that inserts a gap at a random position (thus potentially changing a whole sequence) does not explore the same neighborhood as a mutation that shifts a gap by one residue. This is a serious problem, and in practice defining the neighborhood operators constitutes the main difficulty when designing a Genetic Algorithm.

4.5.2 Applications of GAs in Sequence Analysis

Although the concept of GA is relatively new in the context of sequence alignment, it has already been applied to sequence analysis on several occasions, mostly in the context of protein structure predictions for DNA, RNA and proteins. This makes sequence alignment problems fit the concept of GAs very well. A protein structure can be described as a list of amino acids associated with some conformation values. These conformation values may be angles of chemical bonds. In such a context, the representation can be the list of the bonds that need to be characterized (allelic values being the actual angles in a given conformation). These are the most difficult problems, mostly because it is hard to create an objective function that accurately describes the protein folding process. Several attempts have been made that suggest GAs as the best suited optimization strategies for *ab initio* folding (130-136). Recently, GAs have been proposed for docking analysis using a GA (137). Finally, RNA folding has also received some attention (87, 88, 138, 139). Generally speaking, the use of GAs in sequence and structure analysis is rapidly expanding. For instance,

records about 110 publications describing GA based methods for sequence alignment a period of 8 years [1989-1997]. Out of these 110 publications, 45 appeared alone.

In terms of sequence alignment, the only work we are aware of is an attempt at iterative alignments using a genetic algorithm in order to speed up the process from SAGA, no method has been described which is able to directly perform the alignment and to accommodate non-standard objective functions. We will review the motivations behind SAGA as well as the work done to validate and extend its scope to a larger variety of objective functions and problems.

5 COMPARISON OF THE METHODS

We are only aware of two general studies made to compare systematical alignment methods (117, 140). The work by McClure et al. was made by comparing performances of seven different methods (22, 23, 74, 77, 107, 110, 141) with four different protein families (Hemoglobin, RibonucleaseH, Kinases and proteases). The methods were evaluated for their ability to correctly align sequences of known structural or functional importance in each family. The conclusion is that proteins with more than 50% identity can usually be correctly aligned, regardless of the methods. When identity decreases, all the methods become affected by the same sequences (more sequences do not necessarily improve the results). The conclusion is that progressive methods doing global alignments are usually better. However, the test sets were not large enough to really allow strong distinction between the methods ability and the non progressive methods tested were at an early development stage.

A main drawback of the approach taken by McClure was the limited set of alignments on which their comparison was based, and the fact that these alignments have been established by structural analysis. It has now become standard procedure to compare new methods by comparing their output with reference alignments available in structural databases (33, 142-144). Gotoh recently presented such a comparison (117) based on structural databases (33, 142). In his study, he compared four methods based on progressive alignments (21), two iterative alignment strategies previously described (73, 145) and a new one that involves iterative alignment with dynamic reevaluation of the weights and the phylogenetic tree for optimizing the sums of pairs with affine gap penalties. The methods were applied to 54 protein families. The conclusion was that iterative alignments methods do on average a bit better than ClustalW (21). It should be noted that the study did not make any use of the previous results obtained with the MSA program (22) which indicate that MSA outperforms ClustalW (98), (Barton, personal communication). This suggests that the quality of the sums of pair optimization provided by the iterative alignment (i.e. global optimization) the improvement is probably due to the dynamic scheme used by Gotoh that allows a better use of the information. The weighting strategy are similar to those described for the MSA program (66). In MSA, the weights are computed from an initial progressive alignment which is probably less accurate than those obtained by Gotoh on optimized alignments and used for reevaluation. It may also be that the iterative strategy used by Gotoh does not lead to maximum optimality, but to sub-optimal alignments that are biologically more accurate.

6 SUMMARY OF THE CONTRIBUTIONS

6.1 SAGA: MAKING MULTIPLE SEQUENCE ALIGNMENT BY GENETIC ALGORITHM

SAGA is a package designed to perform multiple sequence alignments using a genetic algorithm. The method involves evolving a population of alignments in an evolutionary manner and gradually improving the fitness of the population using an objective function which measures multiple alignment quality. SAGA uses a scheduling scheme to control the usage of 22 different operators for combining or mutating them between generations. When used to optimize the sums of pairwise alignment function, SAGA performs better than some of the widely used alternative programs like MSA. This is seen with respect to the ability to achieve an optimal solution. One of the attractions of the approach is the ability to optimize any objective function.

This last point was the main motivation behind the development of SAGA. Many alternative packages exist that can provide reasonable optimizations of multiple sequence alignments, for instance. However, it is known that in cases where the sequences are remotely related, these methods fail, not necessarily because of an optimization problem but also because traditional SP objective functions may not be well adapted for multiple alignments. Another potential application for a good quality optimization is hidden Markov model training. The EM based algorithms used at present are very sensitive to local minimum problems. This problem could easily be overcome using a GA based training scheme.

The validation of SAGA as an optimization tool was made using the program *msa*. Remarkably, starting from unaligned sequences, SAGA was always able to achieve results comparable to MSA results or to outperform them. As far as we know, SAGA is the only optimization method that can achieve this result without using dynamic programming. It is fair to say that SAGA is not yet ready for systematic use and does not pose a real challenge to programs like ClustalW or other progressive alignment packages. On the other hand, SAGA is a powerful tool for analysing new objective functions and for the development of faster heuristic methods. In the long term, we hope that SAGA will become much more practical, thanks to the increasing power of the computers and the improvements made in the algorithm (parallelisation, better seeding, optimization).

The original version of SAGA has been available for just over 10 years (http://www.ebi.ac.uk/~cedric/saga_hp.html). The package now regroups a large number of known users of about 30 people. The software is supported and a new release is planned for the beginning of 1998.

6.2 COFFEE: IMPROVING ON EXISTING OBJECTIVE FUNCTIONS

COFFEE is a natural extension of SAGA. It is an attempt to design a new type of objective function for evaluating the quality of multiple sequence alignments. There are a large number of objective functions described for evaluating multiple sequence alignments, all of which have qualities and drawbacks. With COFFEE, the aim is not to add one more objective function to the list, but to propose a scheme that makes it possible to combine different scoring schemes. The COFFEE score reflects the level of consistency between multiple sequence alignments and a library containing pairwise alignments of the same sequences. We show that multiple sequence alignments can be optimized for their COFFEE score using the genetic algorithm package SAGA. The function is tested on 11 test cases, including structural alignments extracted from 3D_ali(33). On 9 of these test cases, SA

COFFEE function is able to outperform ClustalW (progressive multiple alignments) as judged by comparison with the structural references. We also show that given a library of structure based pairwise sequence alignments from FSSP, SAGA can produce high quality multiple sequence alignments.

An important issue in COFFEE is the validation of an objective function. This should not be confused with the evaluation of an optimization strategy, as in SAGA. An objective function may be optimized correctly but lead to biologically incorrect alignments. Verifying the correctness of an alignment is a complex process, the use of biologically correct references. Since the closest thing to a biological alignment is a structural alignment, we used such alignments to validate our approach (3D_ali).

COFFEE is powerful mostly because of its openness. It can accommodate various types of modification, including position specific weights and different type weights. Furthermore any alignment making technique (pairwise, multiple, etc.) can be integrated into the COFFEE framework and several otherwise incompatible techniques can be combined together.

6.3 RAGA: THREADING RNA SECONDARY STRUCTURES

The distinction between sequence and structure alignments has already been made here. It seemed like an interesting question to ask whether SAGA would accommodate a structure based objective function. The fitness landscape of sequence alignment is likely to be very different from those used for protein alignments. We adapted our technique to RNA secondary structure because this is a much simpler problem than threading. Even if the algorithmic complexity is the same, the problem can be formulated in a fairly accurate way, thanks to the fact that RNA secondary structure is mostly based on Watson and Crick base pairing.

In RAGA, we describe a new approach for accurately aligning two homologous sequences, when the secondary structure of one of them is known. To do so we developed two software packages called RAGA and PRAGA which use a genetic algorithm to optimize the alignments. RAGA is mainly an extension of SAGA. In PRAGA genetic algorithms run in parallel and exchange individual solutions. This method aims to optimize an objective function that describes the quality of a RNA pairwise alignment taking into account both primary and secondary structure, including pseudoknots. We report the results obtained using PRAGA on nine test cases of pairs of eukaryotic subunit ribosomal RNA sequence (nuclear and mitochondrial).

The parallel implementation is described in detail in the corresponding paper. It involves a set of synchronized RAGA processes running on different machines over a set of sequences. Every N generations (typically 5), the processes exchange their fittest individuals. The main originality of this implementation is its relative simplicity: it only involves a population sharing scheme between several GAs and does not require any low level modification of the GAs. Of course, the parallel program is very different from the original GAs (i.e. it induces a very different population structure). Despite this, the parallel version has properties very close to those that would have been obtained if RAGA was parallelized at a lower level (speed and accuracy). The parallelization is made very general. It can be used with SAGA and does not depend on any specific objective function. We plan to maintain this module so that it remains compatible with the developments made on SAGA or RAGA. This type of parallelisation (known as island parallelisation) is not completely new, but we are not aware of any previous model strictly identical to ours. Although no systematic work has yet been done for an accurate characterization, we found that on the RAGA objective function, this approach is much more significant than when the parallel module is applied to SAGA. This is due to the fact that our parallelisation makes it easier for a GA to analyse

fitness landscape, such as the one required by a structure based objective function and PRAGA have been made available over the WWW (http://www.ebi.ac.uk/~cedric/raga_hp.html)

In its present form, RAGA is mostly a prototype. There is little interest in aligning two ribosomal RNA sequences at a time. Large accurate alignments exist and can be used in order to guide the characterization of a new sequence. This has been the motivation for the last project presented here: analysing large multiple alignments in order to optimally use the information they contain. However, in many of the interesting cases, such as the requirement of knowing the guide structure of the sequences to align is not available, and will only be extractable when sequence identity is very low. For this reason, we plan to develop an algorithm in order to use an objective function similar to the one described in this paper that allows the alignment of two RNA sequences using primary and secondary structure. Finally, applying RAGA to the problem of Protein Threading is a natural continuation of this work. We plan to do so in the context of the phDthreader(20).

6.4 OPTIMIZING RIBOSOMAL RNA PROFILE ALIGNMENTS

The purpose of this project is to show that the use of the information contained in an alignment can be optimized when trying to introduce (align) a new sequence. The main question is to decide how each sequence of the alignment should contribute to this new alignment.

It is quite obvious that sequences closely related to the one of interest should be given a higher weight. For this reason, we developed a weighting scheme based on the distances between the new sequence and the rest of the sequences. This is a pairwise weighting scheme (cf. section 3.2.3) quite similar in fact to the one used in COFFEE. On the other hand, there is also a need to avoid bias due to the representation of different taxa. This effect is achieved by using a tree based approach to correct for unequal representation.

We found that the ideal weighting scheme, in a dynamic programming context, should be a combination of these two. Attempts were also made to use position independent penalties. In a DP context, it is hard to take into account secondary structure. This can be done in a second step that will involve implementing a profile based object oriented framework. In the long term the alignment method is meant to be available over the WWW through a JAVA-based server.

CONCLUSION

This work provides further evidence for the usefulness of genetic algorithms in an analysis context. GAs mimic very efficiently the human approach that is made up of a continual attempt to combine partial results in order to obtain a global solution. The work done here is only preliminary, and there is still a lot to be done before SAGA or RAGA come into everyday use. However, this may happen if the algorithm is improved in terms of speed and if the developments of COFFEE keep the preliminary results.

Putting aside the slowness that constitutes their main drawback, GAs have a number of properties. They can accommodate a large variety of problems and are easier to use than other alternative methods. They also allow a conceptual barrier to be broken between biological and computational problems. Optimizing a function is purely a formal exercise that does not teach us much about biology. On the other hand, designing an objective function can be extremely informative about the problem analysed and about the understanding of the type of constraints that occur during evolution.

It is along these lines that I plan to extend the genetic algorithm strategy to a range of problems, namely genome alignments and motif discovery. In a period when complete genome sequences are published every month or so, there is a surprising number of tools allowing one to compare these genomes. It is unfortunate that such comparisons can be extremely informative in understanding the way genomes evolve and how way functionally related sequences get clustered or not. In terms of computational difficulty, this is a difficult problem, because of the nature of the events that occur during evolution (inversion, transpositions, deletions, insertion) that are almost impossible to handle with traditional techniques. I believe a GA approach is very likely to provide a solution to such a problem.

REFERENCES

1. Lipman, D. J. and Pearson, W. R., *Rapid and sensitive protein similarity searches*. Science, 1985. 227: p. 1435-1441.
2. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J., *local alignment search tool*. Journal of Molecular Biology, 1990. 215: p. 403-410.
3. Bairoch, A., Bucher, P., and Hofmann, K., *The PROSITE database, its status i 1997*. Nucleic Acids Research, 1997. 25: p. 217-221.
4. Saitou, N. and Nei, M., *The neighbor-joining method: a new method for reconstructing phylogenetic trees*. Mol. Biol. Evol., 1987. 4: p. 406-425.
5. Sonnhammer, E. L. L. and Kahn, D., *Modular Arrangement of Proteins as Inferred from Analysis of Homology*. Protein Science, 1994. 3: p. 482-492.
6. Gribskov, M., McLachlan, M., and Eisenberg, D., *Profile analysis: Detection distantly related proteins*. Proceedings of the National Academy of Sciences, 1987. 4355-5358.
7. Haussler, D., Krogh, A., Mian, I. S., and Sjölander, K. *Protein Modeling us Hidden Markov Models: Analysis of Globins*. in *Proceedings for the 26th Hawaii International Conference on Systems Sciences*. 1993. Wailea, HI, U.S.A.: Los Alamitos, CA: IEEE Computer Society Press.
8. Garnier, J., Gibrat, J.-F., and Robson, B., *GOR method for predicting protei secondary structure from amino acid sequence*. Methods in Enzymology, 1996. 266: p 540-553.
9. Garnier, J. and Robson, B., *The GOR method for predicting secondary structure in proteins*, in *Prediction of protein structure and the principles of protein conformation*, F.G. D., Editor. 1989, Plenum Press: New York. p. 417-465.
10. Rost, B., Sander, C., and Schneider, R., *PHD - an automatic server for protein secondary structure prediction*. CABIOS, 1994. 10: p. 53-60.
11. Göbel, U., Sander, C., Schneider, R., and Valencia, A., *Correlated mutations residue contacts in proteins*. Proteins, 1994. in press: p. 000-000.
12. Neher, E., *How frequent are correlated changes in families of protein sequences?* Proceedings of the National Academy of Sciences, 1994. 91: p. 98-102.
13. Shindyalov, I. N., Kolchanov, N. A., and Sander, C., *Can three-dimensio contacts in protein structures be predicted by analysis of correlated mutations?* Protein Engineering, 1994. 7: p. 349-358.
14. Woese, C. R., Gutell, R., Gupta, R., and Noller, H. F., *Detailed analysis of higher-order of 16S-like ribosomal ribonucleic acids*. Microbiol. Rev., 1983. 47: p. 621-669.
15. Gautheret, D. and Gutell, R. R., *Inferring the conformation of RNA base pairs and triples from patterns of sequence variation*. Nucleic Acids Res, 1997. 25(8): p. 1559-64.
16. Gutell, R. R., Weiser, B., Woese, C. R., and Noller, H. F., *Comparative anatoto of 16S-like ribosomal RNA*. Prog. Nucleic Acid Res. Mol. Biol., 1985. 32: p. 155-21

17. Lathrop, R. H., *The protein threading problem with sequence amino acid interaction preferences is NP-complete*. Protein Engineering, 1994. 7: p. 1059-1068.
18. Needleman, S. B. and Wunsch, C. D., *A general method applicable to the search for similarities in the amino acid sequence of two proteins*. J. Mol. Biol., 1970. 48: p. 443-53.
19. Wang, L. and Jiang, T., *On the complexity of multiple sequence alignment*. Journal of computational biology, 1994. 1(4): p. 337-348.
20. Rost, B., Schneider, R., and Sander, C., *Protein fold recognition by prediction based threading*. Journal of Molecular Biology, 1996. : p. in press.
21. Thompson, J., Higgins, D., and Gibson, T., *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice*. Nucleic Acids Res., 1994. 22: p. 4673-4690.
22. Lipman, D. J., Altschul, S. F., and Kececioglu, J. D., *A tool for multiple sequence alignment*. Proc. Natl. Acad. Sci. USA, 1989. 86: p. 4412-4415.
23. Taylor, W. R., *Multiple sequence alignment by a pairwise algorithm*. Computer Applications in Biological Science, 1987. 3: p. 81-87.
24. Corpet, F., *Multiple sequence alignment with hierarchical clustering*. Nucleic Acids Res., 1988. 16: p. 10881-10890.
25. Dayhoff, M. O., Schwarz, R. M., and Orcutt, B. C., *A model of evolutionary change in proteins. Detecting distant relationships: computer methods and results*, in *Atlas of Protein Sequence and Structure*, M.O. Dayhoff, Editor. 1979, National Biomedical Research Foundation: Washington, D.C. p. 353-358.
26. Henikoff, S. and Henikoff, J. G., *Amino acid substitution matrices from protein blocks*. Proc. Natl. Acad. Sci., 1992. 89: p. 10915-10919.
27. Vogt, G., Etzold, T., and Argos, P., *An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited*. J. Mol. Biol. 1995, 1995. 299(4): p. 816-831.
28. Henikoff, S. and Henikoff, J. G., *Performance evaluation of amino acid substitution matrices*. Proteins: Structure, Function, and Genetics, 1993. 17: p. 49
29. Rost, B. and Sander, C., *Prediction of protein secondary structure at better than 70% accuracy*. Journal of Molecular Biology, 1993. 232: p. 584-599.
30. Lüthy, R., McLachlan, A. D., and Eisenberg, D., *Secondary structure-based profiles: use of structure-conserving scoring tables in searching protein sequence databases for structural similarities*. Proteins: Structure, Function, and Genetics, 1991. 10: p. 239.
31. Jones, D. T., Taylor, W. R., and Thornton, J. M., *A mutation data matrix transmembrane proteins*. FEBS Letter, 1994. 339: p. 269-275.
32. Tomii, K. and Kanehisa, M., *Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins*. Protein Engineering, 1996. 9: p. 27-36.

33. Pascarella, S. and Argos, P., *A data bank merging related protein structures and sequences*. Protein Eng., 1992. 5: p. 121-137.
34. Gonnet, G. H., Cohen, M. A., and Benner, S. A., *Exhaustive matching of the entire protein sequence database*. Science, 1992. 256: p. 1443-1445.
35. Benner, S. A., Cohen, M. A., and Gonnet, G. H., *amino acid substitution during functionally constrained evolution of protein sequences*. Protein Engineering, 1994. 7(11): p. 1323-1332.
36. Sjolander, K., Karplus, K., Brown, M., Huguey, R., Krogh, A., Saira, M. Haussler, D., *Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology*. Comp. App. in Biosci., 1996. 12(4): p. 327-345.
37. Overington, J., Donnelly, D., Johnson, M. S., Sali, A., and Blundell *Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds*. Protein Science, 1992. 1: p. 216-226.
38. Benner, S. A., Cohen, M. A., and Gonnet, G. H., *Empirical and structural models for insertions and deletions in the divergent evolution of proteins*. J. Mol. Biol., 1993. 229: p. 1065-1082.
39. Pascarella, S. and Argos, P., *Analysis of insertions/deletions in protein structures*. J. Mol. Biol., 1992. 224: p. 461-471.
40. Chou, P. Y. and Fasman, G. D., *Empirical predictions of protein conformation*. Ann. Rev. Biochem., 1978. 47: p. 251-276.
41. Gu, X. and Wen-Hsiung, L., *The size distribution of insertions and deletions in human and roent pseudogenes suggests the logarithmic gap penalty for sequence alignment*. J. Mol. Evol., 1995. 40: p. 464-473.
42. Altschul, S. F., *Gap costs for multiple sequence alignment*. J. Theor. Biol., 1989. 138: p. 297-309.
43. Waterman, M. S., *Efficient sequence sequence alignment*. Journal of Theoretic Biology, 1984. 108: p. 333-337.
44. Miller, W. and Myers, E. W., *Sequence comparison with concave weighting functions*. Bull. Math. Biol., 1988. 50: p. 97-120.
45. Gotoh, O., *An improved algorithm for matching biological sequences*. J. Mol. Biol., 1982. 162: p. 705-708.
46. Vingron, M. and Waterman, M. S., *Sequence alignment and penalty choice*. Journal of Molecular Biology, 1994. 235: p. 1-12.
47. Taylor, W. R., *Motif-Biased Protein Sequence Alignment*. Journal of Computational Biology, 1994. .
48. Wilbur, W. J. and Lipman, D. J., *Rapid similarity searches of nucleic acid and protein data banks*. Proceedings of the National Academy of Sciences, 1983. 80: 730.
49. Pearson, W. R. and Lipman, D. J., *Improved tools for biological sequence comparison*. Proceedings of the National Academy of Sciences, 1988. 85: p. 244

50. Smith, T. F. and Waterman, M. S., *Identification of common molecular subsequences*. J. Mol. Biol., 1981. 147: p. 195-197.
51. Pearson, W. R., *Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms*. Genomics, 1991. 11: p. 635-650.
52. Waterman, M. S. and Vingron, M., *Rapid and accurate estimates of statistical significance for sequence database searches*. Proc. Natl. Acad. Sci. USA, 1994. 91: p. 4625-4628.
53. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., M W., and Lipman, D., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*. Nucleic Acids Research, 1997. .
54. Bucher, P. and Hofmann, K., *A sequence similarity search algorithm based on a probabilistic interpretation of an alignment scoring system*. Ismb, 1996. 4(44): p. 44-51.
55. Murata, M., Richardson, J. S., and Sussman, J. L., *Simultaneous comparison three protein sequences*. Proceedings of the National Academy of Sciences, 1985. 3073-3077.
56. Gotoh, O., *Alignment of three biological sequences with an efficient traceback procedure*. J. Theor. Biol., 1986. 121: p. 327-337.
57. Carrillo, H. and Lipman, D. J., *The multiple sequence alignment problem in biology*. SIAM J. Appl. Math., 1988. 48: p. 1073-1082.
58. Sankoff, D., *Minimal mutation trees of sequences*. SIAM J. Appl. Math., 1975. 28: p. 35-42.
59. Fredman, M. L., *Algorithms for computing evolutionary similarity measures with length-independent gap penalties*. Bull. Math. Biol., 1984. 46: p. 553-566.
60. Altschul, S. F. and Erickson, B. W., *Optimal sequence alignment using affine gap costs*. Bull. Math. Biol., 1986. 48: p. 603-616.
61. Bairoch, A. and Apweiler, R., *The SWISS-PROT protein sequence data bank and its new supplement TrEMBL*. Nucleic Acids Research, 1997. 25: p. 31-36.
62. Vingron, M. and Sibbald, P., *Weighting in sequence space: a comparison of methods in terms of generalized sequences*. Proceedings of the National Academy of Sciences, 1993. 90: p. 8777-8781.
63. Sibbald, P. R. and Argos, P., *Weighting aligned protein or nucleic acid sequences to correct for unequal representation*. Journal of Molecular Biology, 1990. 216: p. 8 818.
64. Felsenstein, J., *Inferring evolutionary trees from DNA sequences*, in *Statistical analysis of DNA sequences*, B.S. Weir, Editor. 1983, Marcel Dekker Inc.: New York 133-150.
65. Thompson, J. D., Higgins, D. G., and Gibson, T. J., *Improved sensitivity profile searches through the use of sequence weights and gap excision*. Computer Applications in Biological Science, 1994. 10: p. 19-29.
66. Altschul, S. F., Carroll, R. J., and Lipman, D. J., *Weights for data related by tree*. Journal of Molecular Biology, 1989. 207: p. 647-653.

67. Sander, C. and Schneider, R., *Database of homology-derived structures and the structural meaning of sequence alignment*. Proteins: Structure, Function, and Genet 1991. 9: p. 56-68.
68. Henikoff, S. and Henikoff, J. G., *Position-based sequence weights*. Journal of Molecular Biology, 1994. 243: p. 574-578.
69. Gotoh, O., *Optimal alignment between groups of sequences and its application to multiple sequence alignment*. Comput Appl Biosci, 1993. 9(3): p. 361-70.
70. Neuwald, A. F., Liu, J. S., Lipman, D. J., and Lawrence, C. E., *Extrac protein alignment models from the sequence database*. Nucleic Acids Res, 1997. 25(9): p. 1665-77.
71. Gibson, T. J., Thompson, J. D., and Heringa, J., *The KH domain occurs i diverse set of RNA-binding proteins that include the antiterminator NusA and is probably involved in binding nucleic acid*. FEBS Letters, 1993. 324: p. 361-366.
72. Taylor, W. R., *A non-local gap-penalty for profile alignment*. Bull Math Biol, 1996. 58(1): p. 1-18.
73. Gotoh, O., *Further improvement in methods of group-to-group sequence alignment with generalized profile operations*. Comput Appl Biosci, 1994. 10(4): p. 379-87.
74. Feng, D.-F. and Doolittle, R. F., *Progressive sequence alignment as a prerequisite to correct phylogenetic trees*. Journal of Molecular Evolution, 1987. 25: p. 351-360.
75. Taylor, W. R., *A flexible method to align large numbers of biological sequences*. Journal of Molecular Evolution, 1988. 28: p. 161-169.
76. Higgins, D. G. and Sharp, P. M., *CLUSTAL: a package for performing multiple sequence alignment on a microcomputer*. Gene, 1988. 73: p. 237-244.
77. Higgins, D. G., Bleasby, A. J., and Fuchs, R., *CLUSTAL V: improved softwar for multiple sequence alignment*. Computer Applications in Biological Science, 1992. 189-191.
78. Bucher, P., Karplus, K., Moeri, N., and Hofmann, K., *A flexible motif sea technique based on generalized profiles*. Comput Chem, 1996. 20(1): p. 3-23.
79. Luthy, R., Xenarios, I., and Bucher, P., *Improving the sensitivity of the sequence profile method*. Protein Sci, 1994. 3(1): p. 139-46.
80. Rabiner, L. R., *A tutorial on hidden Markov Models and selected applications in speech recognition*. Proc. IEEE, 1989. 77: p. 257-286.
81. Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D., *H Markov Models in Computational Biology: Applications to Protein Modeling*. J. Mol. Biol., 1994. 235: p. 1501-1531.
82. Hughey, R. and Krogh, A., *Hidden Markov models for sequence analysis: extension and analysis of the basic method*. Computer Applications in Biological Scienc 1996. 12: p. 95-107.
83. Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A and Wootton, J. C., *Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment*. Science, 1993. 262: p. 208-214.

84. Bryant, S. H. and Altschul, S. F., *Statistics of sequence-structure threading*. Current Opinion in Structural Biology, 1995. 5: p. 236-244.
85. Crick, F. H. C., *Central dogma of molecular biology*. Nature, 1970. 227: p. 561-563.
86. Richards, R. G., *5 S RNA An analysis of Possible Base Pairing Schemes*. Eur J Biochem, 1969. 10: p. 36-42.
87. Gultayayev, A. P., van Batenburg, F. D. H., and Pleij, C. W. A., *The comp Simulation of RNA Folding Pathways Using a Genetic Algorithm*. J. Mol. Biol., 1995. 250: p. 37-51.
88. Shapiro, B. A. and Wu, J. C., *Predicting RNA HType pseudoknots with the massively parallel genetic algorithm*. Comp. Applic. in Biosci., 1997. 13(4): p. 459-47
89. Zuker, M., *Computer prediction of RNA structure*. Meth. Enzymol., 1989. 180: p 262-288.
90. Kim, J., Cole, J. R., and Pramanik, S., *Alignment of possible secondary structures in multiple RNA sequences using simulated annealing*. Comp. Applic. Biosci., 1996. 12(4): p. 259-267.
91. Eddy, S. R. and Durbin, R., *RNA alignment using covariance models*. Nucleic Acid Res., 1994. 22(11): p. 2079-2088.
92. Corpet, F. and Michot, B., *RNAlign program: alignment of RNA sequences using both primary and secondary structures*. Comp. Applic. Biosci., 1994. 10(4): p. 389-99.
93. Sakakibara, Y., Brown, M., Underwood, R. C., Mian, I. S., and Hauss *Stochastic Context-Free Grammars for Modeling RNA*. in *27th Hawaii International Conference on System Sciences*. 1994. Wailea, HI, U.S.A.: Los Alamitos, CA: IEEE Computer Society Press.
94. Lefebvre, F. *An optimized parsing algorithm well suited to RNA folding*. in *ISMB-95*. 1995. Cambridge, England: AAAI Press.
95. Pleij, C., *Pseudoknots: a new motif in the RNA GAME*. Trends Biochem. Sci., 1990. 15: p. 143-147.
96. Westhof, E. and Jaeger, L., *RNA pseudoknots*. Current Opinion In Struc Biology, 1992. 2: p. 327- 333.
97. Notredame, C., O'Brien, E. A., and Higgins, D. G., *RAGA: RNA Sequen Alignment by Genetic Algorithm*. Nucleic Acids Res.(in press), 1997. .
98. Notredame, C. and Higgins, D. G., *SAGA: sequence alignment by genetic algorithm*. Nucleic Acids Res., 1996. 24: p. 1515-1524.
99. Hirschberg, D. S., *A linear space algorithm for computing maximal common subsequences*. Comm. ACM, 1975. 18: p. 341-343.
100. Myers, E. W. and Miller, W., *Optimal alignments in linear space*. Comput. Applic Biosci., 1988. 4: p. 11-17.
101. Naor, D. and Brutlag, D. L., *On Near-Optimal Alignmnets of Biological Sequences*. Journal of Computational Biology, 1994. 1(4): p. 349-366.

102. Saqi, M. A. S. and Sternberg, M. J. E., *A simple method to generate non-trivial alternate alignments of protein sequences*. Journal of Molecular Biology, 1991. 219: 727-732.
103. Chao, K. M., *Computing all the suboptimal alignments in linear space*. Lecture Notes In Computer Science, 1994. 807: p. 31-42.
104. Vingron, M. and Argos, P., *Determination of reliable regions in protein sequence alignment*. Protein Eng., 1990. 3: p. 565-569.
105. Taylor, W. R. and Orengo, C. A., *Protein structure alignment*. Journal of Molecular Biology, 1989. 208: p. 1-22.
106. Sneath, P. H. A. and Sokal, R. R., *Numerical Taxonomy*. 1973, San Francisco: Freeman, W.H.
107. Barton, G. J. and Sternberg, M. J. E., *A strategy for the rapid multiple alignment of protein sequences: confidence levels from tertiary structure comparisons*. Journal of Molecular Biology, 1987. 198: p. 327-337.
108. Gusfield, D., *Efficient methods for multiple sequence alignment with guaranteed error bounds*. Bull. Math. Biol., 1993. 55: p. 141-154.
109. Gupta, S. K., Kececioğlu, J. D., and Schaffer, A. A., *Improving the space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment*. Journal of Computational Biology, 1995. 2(3): p. 459-472.
110. Kececioğlu, J. D., *The maximum weight trace problem in multiple sequence alignment*. Lecture Notes in Computer Science, 1993. 684: p. 106-119.
111. Reinert, K., Lenhof, H. P., Mutzel, P., Melhorn, K., and Kececioğlu, J., *branch-and-cut Algorithm for multiple sequence alignment*. Recomb97, 1997. : p. 241-249.
112. Kauffman, S. and Levin, S., *Toward a General Theory of Adaptive Walks on Rugged Landscapes*. J. Theor Biol, 1987. 128(1): p. 11-45.
113. Charleston, M. A., *Toward a characterisation of landscapes of combinatorial optimisation problems, with special attention to the phylogeny problem*. Journal of Computational Biology, 1995. 2(3): p. 439-450.
114. Berger, M. P. and Munson, P. J., *A novel randomized iterative strategy for aligning multiple protein sequences*. Comput. Appl. Biosci., 1991. 7: p. 479-484.
115. Ishikawa, M., Toya, T., and Tokoti, Y., *Parallel Iterative Aligner with Genetic Algorithm*. in *Artificial Intelligence and Genome Workshop, 13th International Conference on Artificial Intelligence*. 1993. Chambery, France:
116. Hirosawa, M., Tokoti, Y., Hoshida, M., and Ishikawa, M., *Comprehensive study on iterative algorithms of multiple sequence alignments*. Comp. App. Biosci., 1995. 11(1): p. 13-18.
117. Gotoh, O., *Significant Improvement in Accuracy of Multiple Protein Sequence Alignments by Iterative Refinements as Assessed by Reference to Structural Alignments*. J. Mol. Biol., 1996. 264(4): p. 823-838.

118. Geman, D. and Geman, D., *Gibbs Sampling*. Trans. Pattern Anal. Mach. I 1984. 6: p. 721.
119. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and E., *Equation of state calculations by fast computing machines*. J. Chem. Phys., 1953. 21: p. 1087-1092.
120. Kirkpatrick, S., Gelatt, C. D. J., and Vecchi, M. P., *Optimization by Simulated Annealing*. Science, 1983. 220: p. 671-680.
121. Ingber, L. and Rosen, B., *Genetic Algorithm and Very Fast simulated Reannealing: a comparison*. Mathematical Computer Modeling, 1993. 16: p. 87-100.
122. Ishikawa, M., Toya, T., Hoshida, M., Nitta, K., Ogiwara, A., and Kaneh *Multiple sequence alignment by parallel simulated annealing*. Comp. Applic. Biosci., 1993. 9: p. 267-273.
123. Kim, J., Pramanik, S., and Chung, M. J., *Multiple Sequence Alignment using Simulated Annealing*. Comp. Applic. Biosci., 1994. 10(4): p. 419-426.
124. Godzik, A. and Sander, C., *Conservation of residue interactions in a family of Ca-binding proteins*. Prot. Eng., 1989. 2: p. 589-96.
125. Schmitz, M. and Steger, G., *Description of RNA Folding by Simulated Annealing*. J. Mol. Biol, 1996. 255: p. 254-266.
126. Eddy, S. R. *Multiple alignment using hidden Markov models*. in *Third International conference on Intelligent Systems for Molecular Biology (ISMB)*. 1995. Cambridge, England: Menlo Park, CA: AAAI Press.
127. Holland, J. H., *Adaptation in natural and artificial systems*. 1975, Ann Arbor, MI University of Michigan Press.
128. Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. ed. D.E. Goldberg. 1989, New York: Addison-Wesley.
129. Davis, L., *The handbook of Genetic Algorithms*. ed. L. Davis. 1991, New York: Van Nostrand Reinhold.
130. Rabow, A. A. and Scheraga, H. A., *Improved genetic algorithm for the protein folding problem by use of a cartesian combination operator*. Protein Science, 1996. 5: p. 1800-1815.
131. Pedersen, J. T. and Moult, J., *Ab initio structure prediction for small polypeptides and protein fragments using genetic algorithms*. Proteins: Structure, Function, and Genetics, 1995. 23: p. 454-460.
132. Legrand, S. M. and Merz, K. M., *The genetic Algorithm and the conformational search of polypeptides and proteins*. Molecular Simulation, 1994. 13: p. 299-320.
133. Unger, R. and Moult, J., *Genetic Algorithms for Protein Folding Simulations*. J. Mol. Biol., 1993. 231: p. 75-81.
134. Schulze-Kremer, S., *Genetic algorithms for protein tertiary structure prediction*, in *Proceedings of the 2nd Conference on Parallel Problem Solving from Nature*, R. Männer and B. Manderick, Editor. 1992, Elsevier Science Publishers: Amsterdam. p. 3

135. Sun, S., *Reduced representation model of protein structure prediction: statistical potential and genetic algorithms*. Protein Science, 1993. 2(5): p. 762-785.
136. Dandekar, T. and Argos, P., *Potential of genetic algorithms in protein folding and engineering simulations*. Protein Eng., 1992. 5: p. 637-645.
137. Verkhivler, G. M., Rejto, P. A., Gelhaar, D. K., and Freer, S. T., *Exploring energy landscape of molecular recognition by a genetic algorithm: analysis of the requirement for robust docking of HIV-1 Protease and FKBP-2 Complexes*. Proteins:Structure, Function and Genetics, 1996. 250: p. 342-353.
138. Shapiro, B. A. and Wu, J. C., *An annealing mutation operator in the genetic algorithm for RNA folding*. Comp. Applic. Biosci., 1996. 12(3): p. 171-180.
139. Ogata, H., Yutaka, A., and Minoru, K., *A genetic algorithm based molecular modeling technique for RNA stem-loop structures*. Nucleic Acids res., 1995. 23(3): p. 419-426.
140. McClure, M. A., Vasi, T. K., and Fitch, W. M., *Comparative analysis of multiple protein-sequence alignment methods*. Molecular Biology Evolution, 1994. 11(4): p. 559-592.
141. Subbiah, S. and Harrison, S. C., *A method for multiple sequence alignments with gaps*. J. Mol. Biol., 1989. 209: p. 539-548.
142. Sali, A. and Overington, J. P., *Derivation of Rules for comparative protein modeling from a database of protein structure alignments*. Protein Sci., 1994. 3: p. 1582-1596.
143. Holm, L. and Sander, C., *The FSSP database: fold classification based on structure-structure alignment of proteins*. Nucleic Acids Res., 1996. 24: p. 206-210.
144. Barton, G. J., *Scop: structural classification of proteins*. Trends Biochem Sci, 1994. 19(12): p. 554-5.
145. Gotoh, O., *A weighting System and Algorithm for aligning many phylogenetically related Sequences*. Comput. App. in Biosci., 1995. 11(543-551).
146. Tabaska, E. J. and D., S. G. *Automated alignment of RNA sequences to pseudoknotted structures*. in ISMB-97. 1997. Halkidiki, Greece: AAAI Press.