# Computational Molecular Biology

Martin Vingron
*Deutsches Krebsforschungszentrum, Heidelberg*

Hans Peter Lenhof
*Max-Planck-Institut für Informatik, Saarbrücken*

Petra Mutzel
*Max-Planck-Institut für Informatik, Saarbrücken*

Contents



Computational Biology is a fairly new subject that arose in response to the computational problems posed by the analysis and the processing of biomolecular sequence and structure data. The field was initiated in the late 60's and early 70's largely by pioneers working in the life sciences. Physicists and mathematicians entered the field in the 70's and 80's, while Computer Science became involved with the new biological problems in the late 1980's. Computational problems have gained further importance in molecular biology through the various genome projects which produce enormous amounts of data.

For this bibliography we focus on those areas of computational molecular biology that involve discrete algorithms or discrete optimization. We thus neglect several other areas of computational molecular biology, like most of the literature on the protein folding problem, as well as databases for molecular and genetic data, and genetic mapping algorithms. Due to the availability of review papers and a bibliography from 1984 some older papers will not be explicitely mentioned in this bibliography.

# 1 Books and Surveys

In this section, we list some books and surveys that serve as a general introduction to the field.

M. S. Waterman. *Introduction to Computational Biology*. Chapman & Hall, 1995.
is a recent book on computational molecular biology containing a wealth of material on all the subjects dealt with in this bibliography.

J. R. Jungk and R. M. Friedman. *Bull. Math. Biol.*, 46:699–744, 1984.
is an annotated bibliography summarizing the literature up to 1984.

R. Doolittle. Molecular evolution: Computer analysis of protein and nucleic acid sequences. *Meth. Enzymol.*, 183, 1990.
is a collection of articles. This volume gives a good overview of the approaches and the software that are in use in molecular biology. A new volume by the same editor is due to appear in 1996.

P. A. Pevzner and M. S. Waterman. Open combinatorial problems in computational molecular biology. In *Proc. 3-rd Israel Symp. Theory of Comput. and Systems*, Tel Aviv, Israel, 1995. IEEE Computer Society Press.
is a collection of open problems and implicitely gives an excellent overview of the area.

The journal *Algorithmica* devoted a special issue to computational molecular biology: volume 13, numbers 1/2, 1995 (ed. by Myers). Similarly, *Discrete Applied Mathematics* is going to publish a special issue. A brief overview that gives a taste of the area is
E. S. Lander, R. Langridge, and D. M. Saccocio. A report on computing in molecular biology: Mapping and interpreting biological information. *Commun. ACM*, 34(11):33–39, 1991.

# 2 Sequence Alignment and Evolution

The sequence of a DNA molecule can be modeled as a string over a 4-letter alphabet, each letter representing one of the four nucleotides that make up DNA. Proteins, the other class of biological macromoleculas are linear chains of amino acids and are represented as strings over a 20-letter alphabet. Sequence alignment deals with comparing different DNA or different protein sequences. This is done by writing one on top of the other padding them with spaces ("indels", for insertion or deletion) to achieve identical length. In DNA, the criterion to distinguish among the many possibilities of this arrangment is the number of unequal letters ending up on top of each other minus the number of spaces that were introduced. For protein sequence comparison the pairs of matched letters are weighted and the adjacent spaces are summarized into blocks which receive a penalty. More formally, let $A$ be a finite alphabet and let $S = \{s_1, s_2, \cdots s_k\}$

be a set of finite strings over $A$. We define a new alphabet $\hat{A} = A \cup \{-\}$ by adding to $A$ the symbol dash "$-$" to represent indels. A set $\hat{S} = \{\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_k\}$ of strings over the alphabet $\hat{A}$ is called an alignment of the set $S$, if the following properties hold: (1) The strings in $\hat{S}$ have the same length. (2) Ignoring dashes, string $\hat{s}_i$ is identical with string $s_i$ (for all $i \in \{1, 2, \cdots, k\}$). Hence an alignment can be interpreted as an array with $k$ rows. The $i$th row contains string $\hat{s}_i$. Each column must contain at least one letter of a string in $S$ (columns filled only with dashes are forbidden). The score of an alignment is based on a distance function $d(\hat{s}_i, \hat{s}_j)$ for aligned sequences.

D. Sankoff and J. B. Kruskal. *Time Warps, String Edits and Macromolecules: the Theory and Practice of Sequence Comparison.* Addison Wesley, 1983.

is a book entirely dedicated to sequence comparison.

M. S. Waterman. General methods of sequence comparison. *Bull. Math. Biol.*, 46:473–500, 1984.

M. S. Waterman. Sequence alignments. In *Mathematical Methods for DNA Sequences.*, pages 53–92. CRC Press, Boca Raton, Fl, 1989.

E. Myers. An overview of sequence comparison algorithms in molecular biology. Technical Report 29, Department of Computer Science of the University of Arizona at Tucson Arizona, 1991.

are reviews on sequence alignment.

## 2.1 Pairwise Sequence Alignment

Assume that $k = 2$, i.e., only two strings $s_1$ and $s_2$ are given. The *Pairwise Sequence Alignment Problem* can be formulated as follows: Compute the alignment $\hat{S} = \{\hat{s}_1, \hat{s}_2\}$ of $s_1$ and $s_2$ that minimizes the distance $d(\hat{s}_1, \hat{s}_2)$. An overview of pairwise sequence alignment need not be given here since excellent reviews are available: The surveys listed above all cover pairwise sequence alignment thouroughly. Only certain special topics are not treated in depth in these general reviews.

K.-M. Chao, R. C. Hardison, and W. Miller. Recent developments in linear-space alignment methods: a mini survey. *J. Comput. Biol.*, 1:271–291, 1994.

A naive implementation of a sequence alignment algorithm requires space quadratic in the sequence length. This has been a major obstacle to the routine use of sequence alignment programs. Linear space alignment techniques have contributed greatly to the acceptance of alignment software by biologists.

M. Vingron. Near-optimal sequence alignment. *Curr. Opin. Struct. Biol.*, to appear, 1996.

gives an overview of techniques for computing suboptimal sequence alignments. Another special topic that has recently developed is parametric sequence comparison dealing with the computation of all solutions to the sequence alignment problem as scoring parameters are varied.

M. S. Waterman, M. Eggert, and E. Lander. Parametric sequence comparison. *Proc. Natl. Acad. Sci. USA*, 89:6090–6093, 1992.

D. Gusfield, K. Balasubramanian, and D. Naor. Parametric optimization of sequence alignment. *Algorithmica*, 12(4/5):312–326, 1994.

M. Waterman. Parametric and ensemble sequence alignment algorithms. *Bull. Math. Biol.*, 56(4):743–767, 1994.

describe algorithms for parametric sequence comparison.

D. Eppstein, Z. Galil, R. Giancarlo, and G. Italiano. Sparse dynamic programming I: Linear cost functions. *J. ACM*, 39:519–545, 1992.

D. Eppstein, Z. Galil, R. Giancarlo, and G. Italiano. Sparse dynamic programming II: Concave, convex cost functions. *J. ACM*, 39:546–567, 1992.

study sparse dynamic programming for sequence alignment.

The following books contain extensive material on approximate string matching and are thus in a wider sense related to our topic:

G. Stephen. *String Searching Algorithms.* World Scientific, 1994.

M. Crochemore and W. Rytter. *Text Algorithms.* Oxford University Press, 1994.

Much like sequence alignment, deriving the optimal fold of an RNA molecule is usually done by dynamic programming. This topic, too, is summarized in Waterman's "Introduction to computational biology".

## 2.2 Multiple Sequence Alignment

The biological task of comparing several sequences simultaneously has been formalized in different ways. An overview is given in

S. Chan, A. Wong, and D. Chiu. A survey of multiple sequence comparison methods. *Bull. Math. Biol.*, 54(4):563–598, 1992.

### 2.2.1 Sum-of-Pairs Multiple Alignment

The *Sum of Pairs Multiple Alignment Problem (SPMA)* is defined as follows: Compute the alignment $\hat{S}$ of $S$ that minimizes the sum of the distances of all pairs $\hat{s}_i, \hat{s}_j$:

$$\mathrm{SPMA}(S) := \min_{\hat{S}} \left[ \sum_{i,j} d(\hat{s}_i, \hat{s}_j) \right].$$

L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *J. Comput. Biol.*, 1:337–348, 1994.

prove that the SPMA-problem is NP-complete by reduction from the *shortest common supersequence problem* [5].

The SPMA-problem can be formulated as a shortest-path problem: Assume for simplicity that all strings in $S$ have length $n$. The set of possible alignments of $S$ can be represented by a $k$ dimensional mesh-shaped graph with $n^k$ vertices and $O(n^k 2^k)$ directed edges. Each path in the graph from the source to the sink represents a possible multiple alignment of $S$. The sequence of edges of a path represents the sequence of columns of the alignment, i.e., each edge codes for one column of the alignment. A vertex in the graph can be interpreted as a "frontier" in the string set $S$ and thus also represents a set of prefixes of $S$ ending at this frontier. Hence the set of all paths from the source to a vertex encode the set of all possible alignments of the prefixes represented by the vertex. Solving the SPMA-problem for $S$ means computing the (shortest) path that minimizes the cost function.

The SPMA-problem can be solved using dynamic programming:
M. Waterman, T. Smith, and W. Beyer. Some biological sequence metrics. *Adv. Math.*, 20:367–387, 1976.

For $k = 3$ see
R. Jue, N. Woodbury, and R. Doolittle. Sequence homologies among E. Coli ribosomal proteins: Evidence for evolutionary related groupings and internal duplications. *J. Mol. Evol.*, 15:129–148, 1980.

M. Fredman. Algorithms for computing evolutionary similarity measures with length independent gap penalties. *Bull. Math. Biol.*, 46:553–566, 1984.

M. Murata, J. Richardson, and J. Sussman. Simultaneous comparisons of three protein sequences. *Proc. Natl. Acad. Sci. U.S.A.*, 82:3037–3077, 1985.

O. Gotoh. Alignment of three biological sequences with an efficient traceback procedure. *J. Theor. Biol.*, 121:327–337, 1986.

Since the size of the graph grows exponentially with the number of strings $k$ ($O(n^k)$ vertices and $O(n^k 2^k)$ edges) the dynamic programming approach works only for very small $k$.
H. Carrillo and D. J. Lipman. The multiple sequence alignment problem in biology. *SIAM J. Appl. Math.*, 48(5):1073–1082, 1988.

present an elegant branch-and-bound approach. They describe a technique for reducing the part of the graph that has to be examined. Only the paths that are contained in a certain "polytope" around the shortest path will be explored by their algorithm.

D. Lipman, S. Altschul, and J. Kececioglu. A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. U.S.A.*, 86:4412–4415, 1989.

implemented a slightly modified version of the algorithm of Carrillo and Lipman. Since the bounds of Carrillo and Lipman are not sufficiently tight for solving "real world" multiple sequence alignment instances, Lipman et al. propose heuristics to improve bounds.

S. Gupta, J. Kececioglu, and A. Schaeffer. Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. *J. Comput. Biol.*, 2:459–472, 1995.

achieve further, significant reduction in space requirements by introducing new algorithmic invariants determining when edges and vertices of the graph first need to be created and when they can be safely destroyed. Speedup results from the usage of more efficient data structures for the shortest-path problem.

This algorithm for computing optimal multiple sequence alignments can handle at most a dozen sequences of length $200 - 400$. Practical problem instances, however, may contain hundreds of sequences and sequence length may be above 1000. Such instances cannot be expected to be solved to optimality. Approximation algorithms and heuristics are required.

D. Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds. *Bull. Math. Biol.*, 55(1):141–154, 1993.

presents the first approximation algorithm for the SPMA-problem. The approximation factor of this algorithm is $2 - 2/k$, where $k$ is the number of sequences. The main idea of Gusfield's approach is to consider alignments that are derived from the *star trees* of $S$.

P. Pevzner. Multiple alignment, communication cost, and graph matching. *SIAM J. Appl. Math.*, 52(6):1763–1779, 1992.

improves the approximation bound to $2 - 3/k$ by deriving multiple alignments from so-called *3-stars*. Using the elegant concept of communication cost, Pevzner proved that there are 3-stars whose derived multiple alignments are at most a factor $2 - 3/k$ from optimal. Pevzner's algorithm computes an alignment in time $O(n^3k^3 + k^4)$, where $n$ is the (maximal) length of the sequences.

V. Bafna, E. Lawler, and P. Pevzner. Approximation algorithms for multiple sequence alignment. In *Proc. 5-th Annual Symp. Combinatorial Pattern Matching*, pages 43–53. Springer-Verlag, 1994.

By exploring cliques of $l$ sequences with one common center ($l$-stars instead of 3-stars) Bafna, Lawler and Pevzner achieve an approximation bound of $2 - l/k$.

### 2.2.2 Maximum Weight Trace

The input to the *Maximum Weight Trace (MWT)* alignment problem is a set $S$ of $k$ strings and a graph $G = (V, E)$. The letters of the strings $s_i$ of $S$ are the vertices $V$ of the graph. Every edge $e \in E$ of the graph has a positive weight $w_e$ and connects two vertices (letters) that belong to different strings (i.e., there are no edges that connect two letters of the same string). An alignment $\hat{S}$ realizes an edge $e$ if the two letters connected by the edge are placed in the same column of the alignment. The set of edges realized by an alignment $\hat{S}$ is called the *trace* (trace($\hat{S}$)) of $\hat{S}$. The MWT problem is defined as follows: Compute an alignment $\hat{S}$ that realizes a trace with maximal weight:

$$\mathrm{MWT}(S) := \max_{\hat{S}} \left[ \sum_{e \in \mathrm{trace}(\hat{S})} w_e \right].$$

J. Kececioglu. The maximum weight trace problem in multiple sequence alignment. In *Proc. 4-th Symp. Combinatorial Pattern Matching*, pages 106–119. Springer-Verlag, 1993.

introduced the MWT-problem. He proves that the MWT-problem contains the SPMA-problem under certain conditions, as a special case, and that the MWT-problem is NP-complete by reduction from the *Feedback Arc Set Problem* [5]. Kececioglu presents a branch-and-bound algorithm for the MWT-problem whose implementation could optimally align six sequences of length 250 in a few minutes. Here, a heuristic alignment of the $k$ sequences yields a lower bound. Upper bounds for the branch-and-bound approach are calculated by adding up the weights of all MWT-optimal pairwise alignments of suffix sets of $S$.

### 2.2.3 Chaining Multiple-Alignment Fragments

A fragment of a set $S$ of $k$ strings is a set $S' = \{s'_1, s'_2, \cdots, s'_k\}$, where $s'_i$ is a non-empty substring of $s_i$ (for all $i$). A fragment $f_1$ is smaller than a fragment $f_2$ ($f_1 < f_2$), if the substrings of $f_1$ and $f_2$ do not overlap and the substrings of $f_1$ are to the left of the substrings of $f_2$. The set of letters of $S$ that lie between the substrings of two fragments $f_1$ and $f_2$ with $f_1 < f_2$ are called the gap of $f_1$ and $f_2$. We call a set $\{f_1 < f_2 < \cdots < f_l\}$ of ordered fragments a chain of fragments. The *Chaining Multiple-Alignment Fragments Problem* is defined as follows: Let $F$ be a set of fragments of $S$, where each fragment $f \in F$ has a positive score (score($f$)). Let gap_cost($*, *$) be a "gap" penalty function that assigns a cost to a gap between two ordered fragments. Compute a chain of fragments $\hat{F}$ that maximizes the following function:

$$\mathrm{CMAF}(S) := \max_{\hat{F}} \left[ \sum_{f \in \hat{F}} \mathrm{score}(f) - \mathrm{gap\_cost}(f, \mathrm{successor}(f)) \right].$$

The CMAF-problem can be interpreted as an optimal-path problem: The fragments of $S$ are the vertices of the graph. Each vertex is labeled with the score of its fragment. For each ordered pair $f_1 < f_2$ of fragments a directed edge from $f_1$ to $f_2$ will be added to the graph. The edge will be labeled with the penalty "$-\mathrm{gap\_cost}(f_1, f_2)$" for the gap between $f_1$ and $f_2$. The CMAF-problem can now be formulated as follows: Compute the path in the directed graph that maximizes the sum of the vertex and edge labels.

W. J. Wilbur and D. J. Lipman. The context dependent comparison of biological sequences. *SIAM J. Appl. Math.*, 44:557–567, 1984.

introduced this viewpoint in the context of pairwise alignment.

E. Sobel and R. Martinez. A multiple sequence alignment program. *Nucl. Acids Res.*, 14:363–374, 1986.

apply it to multiple sequence alignment. Assume that computing the gap cost takes time $O(g)$, then the CMAF-problem can be solved in time $O(|F|^2 g)$ by dynamic programming. Here, $|F|$ is the number of fragments in $F$. The algorithm requires $O(k|F|)$ space and works for any arbitrary gap cost function.

Z. Zhang, B. Raghavachari, R. Hardison, and W. Miller. Chaining multiple-alignment blocks. *J. Comput. Biol.*, 1:217–226, 1994.

implemented a practical algorithm for the CMAF-problem that uses $kD$-trees to compute the optimal chain.

E. Myers and W. Miller. Chaining multiple-alignment fragments in sub-quadratic time. In *Proc. 6-th Annual ACM-SIAM Symp. Discr. Algorithms*, pages 38–47, 1995.

presented the first sub-quadratic time algorithm for special gap cost functions. Myers and Miller prove that the maximal fragment chain can be computed in time $O(|F|(\log|F|)^k)$ with $O(k|F|(\log|F|)^{k-1})$ space.

## 2.3 Evolutionary Trees

Molecular sequences are used to reconstruct the course of evolution. Since evolution is assumed to have proceeded from a common ancestral species in a tree-like branching of species (molecules), this process is generally modeled by a tree. When the most ancestral species is known, the model will be a rooted tree. The leaves of the tree are labeled with contemporary species while the inner nodes correspond to hypothetical ancestors. The key question is the reconstruction of this tree based on contemporary data. These data may come in one of two forms: As a multiple alignment with the sequences corresponding to leaves or as a matrix of distances between leaf-labels. Methods are thus divided into character-based methods and distance-based methods.

### 2.3.1 Character-based methods

F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem.* North-Holland, 1992.

contains a good introduction to character-based methods.

An idealized but interesting model of evolution is embodied in the *Perfect Phylogeny Problem.* Let the number of species be $k$. Let a set of $m$ characters (e.g., the columns of the multiple alignment), each character having $r$ possible states (e.g., the four nucleotides A, C, G, and T), be given. We say that a character is compatible with a tree when the inner nodes of the tree can be labeled such that each character state induces a subtree. A tree is said to be a perfect phylogeny when all characters are compatible with it. The perfect phylogeny problem is to decide whether a given set of characters has a perfect phylogeny and if so construct it.

H. Bodlaender, M. Fellows, and T. Warnow. Two strikes against perfect phylogeny. In *Proc. 19-th Int. Coll. Automata, Lang. and Program.*, pages 273–283. Lecture Notes Comp. Sci., 1992.

M. A. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *J. Classification*, 9:91–116, 1992.

both show NP-completeness for arbitrary number of character states.

D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21:19–28, 1991.

solves the perfect phylogeny problem for binary characters ($r = 2$) in linear time.

A. Dress and M. Steel. Convex tree realizations. *Appl. Math. Lett.*, 5:3–6, 1992.

present a solution for $r = 3$ in $O(km^2)$.

S. K. Kannan and T. J. Warnow. Inferring evolutionary history from DNA sequences. *SIAM J. Comput.*, 23(4):713–737, 1994.

This paper presents an $O(k^2m)$ algorithm for $r \leq 4$ which is especially important because it allows the modeling of evolution of DNA sequences.

F. R. McMorris, T. Warnow, and T. Wimer. Triangulating vertex-colored graphs. *SIAM J. Discr. Math.*, 7:296–306, 1993.

show that the perfect phylogeny problem is polynomially equivalent to coloring triangulated graphs and use this to design a perfect phylogeny algorithm for arbitrary $r$ running in $O((rm)^{m+1} + km^2)$.

R. Agarwala and D. Fernández-Baca. A polynomial time algorithm for the perfect phylogeny problem when the number of character states is fixed. In *Proc. 34-th Annual IEEE Symp. Found. Comput. Sci.*, 1993. Also to appear in SIAM J. Comp.

improve this result by giving a dynamic programing algorithm for the perfect phylogeny problem. Based on their ideas

S. Kannan and T. Warnow. A fast algorithm for the computation and enumeration of perfect phylogenies when the number of character states is fixed. pages 595–603, 1995.

give an $O(2^{2r}km^2)$ algorithm.

T. Warnow. Tree compatibility and inferring evolutionary history. *J. Algorithms*, 16:388–407, 1994.

For a given alignment, the question of identifying the maximal number of alignment columns that allow a perfect phylogeny is addressed. It is mapped to a maximum clique problem.

The *Parsimony Problem*, famous in molecular biology, can be thought of as a relaxation of the perfect phylogeny problem. When a character state cannot be mapped to a subtree, it will induce a forest. For a given tree, finding the inner node assignment such that the number of trees in all the forests induced by the states of the characters is minimized is the parsimony problem. This number equals the minimal number of mutations required to explain the leaf-labeling of a given tree.

W. Fitch. Toward defining the course of evolution: Minimum change for specific tree topology. *Systematic Zoology*, 20:406–416, 1971.

gave a linear time, dynamic programming algorithm for the parsimonious inner node assignment.

J. A. Hartigan. Minimum mutation fits to a given tree. *Biometrics*, 29:53–65, 1973.

prove correctness of Fitch's algorithm. After minimizing the number of necessary mutations over all trees one finds the "most parsimonious tree".

L. R. Foulds and R. Graham. The Steiner problem in phylogeny is NP-complete. *Adv. Appl. Math.*, 3:43–49, 1982.

show NP-completeness of finding the most parsimonious tree. This is based on an analogy to Steiner trees: The most parsimonious tree is the minimal Steiner tree linking the given sequences.

L. R. Foulds, M. D. Hendy, and D. Penny. A graph theoretic approach to the development of minimal phylogenetic trees. *J. Mol. Evol.*, 13:127–149, 1979.

give an approximation algorithm for the most parsimonious tree based on the minimum spanning tree heuristic for Steiner trees.

M. D. Hendy and D. Penny. Branch and bound algorithms to determine minimal evolutionary trees. *Math. Biosci.*, 59:277–290, 1982.

apply branch-and-bound algorithms for calculating the most parsimonious tree.

### 2.3.2 Distance based methods

Distance-based methods attempt to approximate a given set of distances on the leaf-labels of the tree by the path-metric of an edge-weighted tree. A distance matrix that coincide with the path-metric of a tree is called an additive matrix. A characterization of such matrices is given in

P. Bunemann. The recovery of trees from measures of dissimilarity. In F. Hodson, D. Kendall, and P. Tautu, editors, *Mathematics in the archaeological and historical sciences*, pages 387–395. Edinburgh University Press, 1971.

M. S. Waterman, T. F. Smith, M. Singh, and W. A. Beyer. Additive evolutionary trees. *J. Theoret. Biol.*, 64:199–213, 1977.

Let an additive matrix be given. This paper presents an algorithm to compute the tree whose path-metric coincides with the given matrix. Furthermore it proves uniqueness of the tree.

Other algorithms for this purpose are given in:

H.-J. Bandelt and A. Dress. Reconstructing the shape of a tree from observed dissimilarity data. *Adv. Appl Math.*, 7:309–343, 1986.

N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406–425, 1987.

J. Culberson and P. Rudnicki. A fast algorithm for constructing trees from distance matrices. *Inform. Process. Lett.*, 30:215–220, 1989.

H.-J. Bandelt. Recognition of tree metrics. *SIAM J. Discr. Math.*, 3:1–6, 1990.

The idea that for all species the same amount of time has passed since the existence of some common ancestor has led to the study of rooted, edge-weighted trees with all leaves being the same distance away from the root. Such a tree is called an ultrametric tree and its corresponding path-metric is called an ultrametric. A simple clustering method like single linkage clustering [3] suffices to recognize such metrics and compute the tree.

Let $(A)_{i,j}$ be the (additive) path-metric of a tree and let $(B)_{i,j}$ be an arbitrary distance matrix. To judge how well the additive matrix (and thus the tree that goes with it) approximates the distance matrix, a distance between matrices is used. Usually it is defined as $\sum_{i,j} |A_{ij} - B_{ij}|^\alpha$, with $\alpha = 1$ corresponding to the $L_1$-norm, $\alpha = 2$ corresponding to the $L_2$-norm. Some authors use the $L_\infty$-norm ($\max_{i,j} |A_{ij} - B_{ij}|$). The problem of finding the closest tree under either an $L_1$ or and $L_2$-norm has been proven NP-complete in

W. H. E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bull. Math. Biol.*, 49:461–467, 1987.

using results from

M. Krivánek and J. Morávek. NP-hard problems in hierarchical-tree clustering. *Acta Inform.*, 23:311–323, 1986.

Recently, guaranteed error bound algorithms for approximation of a distance matrix by ultrametric and additive trees have been developed:

M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13(1/2):155–179, 1995.

R. Agarwala, V. Bafna, M. Farach, B. Naryanan, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). In *Proc. 7-th Annual ACM-SIAM Symp. Discr. Algorithms*, pages 365–372, 1996.

In this paper the approximation is measured in $L_\infty$-norm.

The reader may have noted a certain abundance of algorithms to construct trees from additive matrices. The importance of having several algorithms on hand for this purpose lies in the fact that they also constitute a source of ideas for heuristics. The review

D. L. Swofford and G. J. Olsen. Phylogeny Reconstruction. In D. M. Hillis and C. Moritz, editors, *Molecular Systematics*, pages 411–501. Sinauer Associates, Sunderland, Massachusetts, 1990.

give an overview of heuristics for tree approximation as well as many other approaches to phylogeny reconstruction that are in practical use.

A prominent method applies maximum likelihood estimation to judge the quality of a tree:

J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, 17:368–376, 1981.

In the last few years, several other interesting approaches to phylogeny reconstruction have emerged which are not based on discrete optimization. Some key papers are

L. Székely, M. Steel, and P. Erdös. Fourier calculus on evolutionary trees. *Adv. Appl. Math.*, 14:200–216, 1993.

S. N. Evans and T. P. Speed. Invariants of some probability models used in phylogenetic inference. *Ann. Statist.*, 21:355–377, 1993.

H. Bandelt and A. Dress. A canonical decomposition theory for metrics on a finite set. *Adv. Appl. Math.*, 92:47–105, 1992.

## 2.4 Trees and Alignment

Let $T$ be a tree whose $k$ leaves are labeled with the $k$ sequences of $S$. The $m$ internal nodes represent sequences of hypothetical ancestral species and are not labeled. A tree alignment $\hat{S}(T) = \{\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_k, s_1^a, \cdots, s_m^a\}$ is a set of $k+m$ strings over the alphabet $\hat{A}$ with the following properties: (1) All $k+m$ strings have the same length. (2) Ignoring dashes, the first $k$ strings are identical with the $k$ strings in $S$. The $m$ last strings represent possible ancestral sequences, the labels for the internal nodes. The cost "$\text{cost}_{\hat{S}}(e)$" of an edge $e$ in a tree alignment $\hat{S}$ is defined as the cost of the "projected" pairwise alignment of the two sequences that are stored in the nodes connected by the edge. The cost of a tree alignment is the sum of the costs of all edges in the tree. The *Multiple Sequence Tree Alignment Problem* is defined as follows: Compute the tree alignment $\hat{S}(T)$ that minimizes the total sum of the edge costs:

$$\text{MSTA}(S) := \min_{\hat{S}(T)} \left[ \sum_{e \in T} \text{cost}_{\hat{S}(T)}(e) \right].$$

There is a more general variant of the MSTA-problem that is called *Generalized Multiple Sequence Tree Alignment GMSTA*. In this more difficult variant of the tree alignment problem, only the $k$ sequences are given and the tree as well as the hypothetical ancestral sequences have to be constructed. Note that for a given alignment, finding the tree that minimizes Hamming distance along the tree edges is the parsimony problem of Section 2.3.

T. Jiang, E. L. Lawler, and L. Wang. Aligning sequences via an evolutionary tree: complexity and approximation. In *Proc. 26-th Annual ACM Symp. Theory of Comput.*, pages 760–769, 1994.

show that the MSTA-problem is NP-hard and that the GMSTA-problem is MAX SNP-hard.

L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *J. Comput. Biol.*, 1:337–348, 1994.

prove that the MSTA-problem is MAX SNP-hard if the given phylogeny (tree) is a star tree.

H. T. Wareham. A simplified proof of the NP- and MAX SNP-hardness of multiple sequence tree alignment. *J. Comput. Biol.*, 2(4):509–514, 1995.

Wareham offers an alternative proof that the GMSTA-problem is both NP-complete and MAX SNP-hard.

D. Sankoff. Minimal mutation trees of sequences. *SIAM J. Appl. Math.*, 28:35–42, 1975.

D. Sankoff and R. J. Cedergreen. Simultaneous comparison of three or more sequences related by a tree. In D. Sankoff and J. B. Kruskal, editors, *Time Warps, String Edits* *and Macromolecules: The Theory and Practice of Sequence Comparison.*, pages 93–120. Addison-Wesley, 1983.

Sankoff raised the question of multiple sequence alignment and introduced the MSTA-problem. It can be formulated as a shortest-path problem in a $k$-dimensional mesh-shaped graph and can be optimally solved using dynamic programming. The edges of the graph represent possible columns of alignments. We have to assign letters to the lower $m$ rows of each column (letters for the inner vertices), such that the cost of each column is minimized. The $m$ lower letters of a column can be computed using a dynamic programming approach (the Fitch-Hartigan algorithm to construct a parsimonious tree, see Section 2.3). This "inner minimization" has to be carried out for each edge (possible column) of the graph. The shortest-path from the source to the sink codes the optimal tree alignment.

S. Altschul and D. J. Lipman. Tree, stars, and multiple biological sequence alignment. *SIAM J. Appl. Math.*, 49(1):179–209, 1989.

present a branch-and-bound algorithm for the MSTA-problem that is an extension of Carrillo and Lipman's algorithm ([Carrillo and Lipman 1988], see Subsection 2.2). Altschul and Lipman describe a new approach to compute suitable bounds for the branch-and-bound algorithm by solving optimization problems that are "almost" classic linear programming problems.

D. Sankoff, R. J. Cedergren, and G. Lapalme. Frequency of insertion-deletion, transversion, and transition in evolution of 5S ribosomal RNA. *J. Mol. Evol.*, 7:133–149, 1976.

designed an iterative procedure for local optimization in the tree. Sankoff et al. decompose the tree into small overlapping subtrees (star trees). In each iteration step all subtrees will be locally optimized starting with the more "peripheral" subtrees. The algorithm stops when an iteration step has been performed without change in the subtree cost.

J. Hein. A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given. *Mol. Biol. Evol.*, 6:649–668, 1989.

J. Hein. A tree reconstruction method that is economical in the number of pairwise comparisons used. *Mol. Biol. Evol.*, 6:669–684, 1989.

Hein designed and implemented a heuristic method that yields good approximations for tree alignment and suggests an algorithm for the GMSTA-problem. Hein introduced the concept of *sequence graphs* for storing large sets of sequences. A sequence graph is a directed, acyclic and connected graph with a source and a sink. Each edge represents a letter or even a subsequence of a sequence. Each path from the source to the sink codes a sequence. A sequence graph represents the set of sequences that is coded by all source-to-sink paths.

D. Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds. *Bull. Math. Biol.*, 55(1):141–154, 1993.

Since the MSTA-problem can be formulated as a Steiner tree problem on graphs, approximation approaches for Steiner minimal trees have been successfully applied to the MSTA-problem. Gusfield has shown that the minimum spanning tree approach yields an approximation ratio of 2. By using better approximation techniques for the Steiner tree problem, for instance the approach of Zelikovsky [8] or of Berman and Ramaiyer [1], better approximation ratios can be obtained.

R. Ravi and J. Kececioglu. Approximation algorithms for multiple sequence alignment. In Z. Galil and E. Ukkonen, editors, *Proc. 6-th Symp. Combinatorial Pattern Matching*, pages 330–339, 1995.

propose an approximation algorithm for regular deg-ary trees (each internal node has exactly deg children) that finds solutions with an approximation ratio of $\frac{\deg+1}{\deg-1}$.

T. Jiang, E. L. Lawler, and L. Wang. Aligning sequences via an evolutionary tree: complexity and approximation. In *Proc. 26-th Annual ACM Symp. Theory of Comput.*, pages 760–769, 1994.

present the first polynomial time approximation scheme (PTAS) for the MSTA-problem that yields approximation ratios arbitrarily close to 1. The algorithm of Jiang et al. computes for any given $t > 1$ an approximation with ratio smaller than $1 + 3/t$. The running time of the algorithm is exponential in $\deg^{t-1}$.

L. Wang and D. Gusfield. Improved approximation algorithms for tree alignment. *to be published*, 1996.

designed a polynomial time approximation scheme (PTAS) for regular deg-ary trees (each internal node has exactly deg children). For a fixed $t > 1$, the approximation ratio of the PTAS is $1 + \frac{2}{t} - \frac{2}{t2^t}$.

# 3 Tree Comparison

Given two or more evolutionary trees computed from different gene families, the problem of comparing these phylogenies arises. Several notions of similarity between trees have been suggested, some of which are merely a similarity measure while others compute a consensus tree. We give only a few references to some prominent approaches.

M. S. Waterman and T. F. Smith. On the similarity of dendrograms. *J. Theor. Biol.*, 73:784–900, 1978.

and

K. Culik II and D. Wood. A note on some tree similarity measures. *Inform. Process. Lett.*, 15:39–42, 1982.

introduce the Nearest Neighbor Interchange (NNI) metric on trees. The metric counts the number of elementary operations, the NNI's, required to transform one tree into the other. For an edge with two subtrees at either node, two NNIs are possible, representing the two alternative topologies.

E. K. Brown and W. H. E. Day. A computationally efficient approximation to the nearest neighbor interchange metric. *J. Classification*, 1:93–124, 1984.

give a heuristic approximation algorithm for calculation of NNI distance.

The existence of a polynomial time algorithm for NNI is still open.

M. Krivánek. Computing the nearest neighbor interchange metric for unlabeled binary trees is NP-complete. *J. Classification*, 3:55–60, 1986.

has shown NP-completeness only for unlabeled trees.

C. R. Finden and A. D. Gordon. Obtaining common pruned trees. *J. Classification*, 2:255–276, 1985.

introduced *Maximum Agreement Subtree* as the homeomorphous subtree common to two trees spanning the maximum number of leaves.

M. Steel and T. Warnow. Kaikoura tree theorems: Computing the maximum agreement subtree. *Inform. Process. Lett.*, 48:77–82, 1993.

M. Farach and M. Thorup. Fast comparison of evolutionary trees. In *Proc. 5-th Annual ACM-SIAM Symp. Discr. Algorithms*, pages 481–488, 1994.

have successively improved the corresponding algorithms.

Maximum agreement subtrees for more than two trees are considered in

A. Amir and D. Keselman. Maximum agreement subtree in a set of evolutionary trees - metrics and efficient algorithms. In *Proc. 35-th Annual IEEE Symp. Found. Comput. Sci.*, pages 758–769, 1994.

J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. In *Proc. 6-th Annual Symp. Combinatorial Pattern Matching*, pages 177–190, 1995.

# 4 Genome Rearrangements

The order in which genes are arranged on the DNA molecule is the result of an evolutionary process. Over time, a gene order formerly present in an ancient species may, due to certain rearrangements in the genome, have evolved into a gene order we can observe today. The computational task lies in reconstructing the changes that may have occured to transform one gene order into another. A much studied elementary operation in these transformations is, e.g., the reversal of subsets of genes. The choice of elementary operation depends on the organism or cell organelle under study. More sophisticated scenarios model the evolution of sets of chromosomes which can exchange genes among each other.

S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Proc. 27-th Annual ACM Symp. Theory of Comput.*, 1995.

is an excellent overview of the field of genome rearrangements.

Let an ordered set of $n$ genes be given. A *reversal* is the operation which cuts out a contiguous subset, inverts the order of genes in this subset, and reinserts it again in its original position. *Reversal distance* between two permutations is the minimal number of reversals required to transform one permutation into another. Since genes also have a direction, a more accurate model introduces signs for the genes. A given set of genes is represented by a permutation with signs on each entry. Reversing a subset of genes then has the additional effect of changing all signs of the affected genes. The problem is to calculate the minimal number of *signed reversals* necessary to transform one signed permutation into another.

G. A. Watterson, W. J. Ewens, T. E. Hall, and A. Morgan. The chromosome inversion problem. *J. Theor. Biol.*, 99:1–7, 1982.

was the first paper to raise the formal problems of comparing gene orders.

D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B. F. Lang, and R. Cedergren. Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome. *Proc. Natl. Acad. Sci. USA*, 89:6575–6579, 1992.

use heuristics to estimate the number of rearrangements that occured between two contemporary species. The resulting distance measure is used to reconstruct an evolutionary tree.

J. Kececioglu and D. Sankoff. Exact and approximation algorithms for the inversion distance between two permutations. *Algorithmica*, 13:180–210, 1995.

give an approximation algorithm for the number of (unsigned) reversals with a guaranteed error bound of 2 and devise a branch-and-bound algorithm. They speculate that the problem is NP-complete.

V. Bafna and P. Pevzner. Genome rearrangements and sorting by reversals. In *Proc. 34-th Annual IEEE Symp. Found. Comput. Sci.*, pages 148–157, 1993. To appear in SIAM J. Comp., vol 25(2), April 1996.

improve the error bound to 1.75. They further show that it may take up to $n-1$ reversals to transform one permutation into another and also devise a factor 1.5 approximation algorithm for signed reversals.

J. Kececioglu and D. Sankoff. Efficient bounds for oriented chromosome inversion distance. In *Proc 5-th Annual Symp. Combinatorial Pattern Matching*, pages 307–325, 1994.

give a branch-and-bound algorithm for signed reversals.

S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Proc. 27-th Annual ACM Symp. Theory of Comput.*, 1995.

found a duality theorem for the number of signed reversals. Based on this theorem they devised an $O(n^4)$ algorithm.

S. Hannenhalli and P. Pevzner. To cut ... or not to cut. In *Proc. 7-th Annual ACM-SIAM Symp. Discr. Algorithms*, pages 304–313, 1996.

use the improved algorithm for signed permutations and connections between the two problems to devise a practically efficient algorithm for unsigned reversals.

V. Bafna and P. Pevzner. Sorting permutations by transpositions. In *Proc. 6-th Annual ACM-SIAM Symp. Discr. Algorithms*, pages 614–623, 1995.

study another operation on permutations, so-called transpositions. In this context a transposition does not exchange two single elements but two adjacent stretches from a permutation.

Genome rearrangements are more complicated when genes are distributed over different chromosomes. Chromosomes can exchange genetic material. A *translocation* is the process where a contiguous set of genes from an end of a chromosome is exchanged with a contiguous set of genes from an end of another chromosome. Fusion, the combination of two chromosomes, and fission, the breakage of one chromosome into two new ones, are other relevant processes.

J. Kececioglu and R. Ravi. Of mice and men: Evolutionary distances between genomes under translocation. In *Proc. 6-th Annual ACM-SIAM Symp. Discr. Algorithms*, pages 604–613, 1995.

S. Hannenhalli. Polynomial-time algorithm for computing translocation distance between genomes. pages 162–176.

study distances between genomes based on the above operations.

J. Kececioglu and D. Gusfield. Reconstructing a history of recombinations from a set of sequences. In *Proc. 5-th Annual ACM-SIAM Symp. Discr. Algorithms*, pages 471–480, 1994.

study related operations in the context of reconstructing evolutionary history.

# 5 Sequencing and Mapping

The goal of the Human Genome Project is the determination of the location of the human genes on the DNA and the sequencing (the determination of the sequence of nucleotides A,C,G, and T) of the entire human genome. In total, the human genome contains about 3 billion letters distributed over 23 chromosomes. Biochemical techniques, however, only allow the researcher to handle comparatively small segments of DNA at a time. Thus, a DNA molecule is broken into smaller pieces. The information one obtains on small pieces needs to be put into a larger context again. To this end one has to infer the order in which the segments occur on the underlying DNA. This task is called mapping. Depending on the scale at which it is done the techniques and the corresponding computational problems vary.

An overview of the combinatorial problems arising in DNA mapping and sequencing is given in

R. M. Karp. Mapping of the genome: Some combinatorial problems arising in molecular biol. In *Proc. 25-th Annual ACM Symp. Theory of Comput.*, pages 278–285, 1993.

P. A. Pevzner, editor. *Combinatorial Methods for DNA Mapping and Sequencing*, volume 2, 2 of *Special Issue of J. Comput. Biol.* 1995.

This volume contains eleven presentations of the DIMACS workshop "Combinatorial Methods in DNA Mapping and Sequencing", that opened the DIMACS computational molecular biology year 1994/1995. The articles give a strong emphasis on applications of combinatorial methods in molecular biology.

See also [Special Issue of Algorithmica, ed. by Myers] and [Open combinatorial problems in computational molecular biology, by Pevzner and Waterman] (see Section 1).

## 5.1 Sequence Assembly

Current technology permits experimentalists to directly determine the sequence of a DNA strand of up to 800 nucleotides in length. To sequence a long piece of DNA many such reads are taken and subsequently re-assembled to produce the original sequence. Computationally, this gives rise to the problem of assembling the fragments using the overlap information among them. Overlaps are deduced from sequence similarity. Since 1–10% of the nucleotides in the fragment data are missing or incorrect, and since a fragment's sequence can be reversed with respect to the others these overlaps cannot be perfectly determined. Thus, the formal *Sequence Reconstruction Problem*, first formalized in

H. Peltola, H. Söderland, J. Tarhio, and E. Ukkonen. Algorithms for some string matching problems arising in molecular genetics. In *Proc. 9-th IFIP World Computer Congress*, pages 59–64, 1983.

can be stated as follows: Given a collection of fragment sequences $\mathcal{F}$ and an error rate $0 \leq \epsilon < 1$, find a shortest sequence $S$ such that every fragment $F \in \mathcal{F}$, or its reverse complement, matches a substring of $S$ with at most $\epsilon|F|$ errors.

H. Peltola, H. Söderland, and E. Ukkonen. SEQUAID: A DNA sequence assembly program based on a mathematical model. *Nucl. Acids Res.*, 12:307–321, 1984.

decompose this extremely hard problem into three combinatorial optimization problems: the overlap graph construction, the layout phase, and the alignment problem.

J. Kececioglu and E. Myers. Exact and approximate algorithms for the sequence reconstruction problem. *Algorithmica*, 13(1-2):7–51, 1995.

Here, for each of the subproblems either exact algorithms or approximate algorithms are suggested.

J. D. Kececioglu. *Exact and approximation algorithms for DNA sequence reconstruction.* PhD thesis, U. Arizona, 1991.

This thesis gives a detailed overview of all aspects concerning the sequence reconstruction problem such as the biological aspects, the related combinatorial optimization problems, the literature and the computer software available.

A pressing practical problem is the existence of repeats in biological sequences. While no definite solution to this problem has been found yet it is addressed, e.g., in

E. Myers. Toward simplifying and accurately formulating fragment assembly. *J. Comput. Biol.*, 2(2):275–290, 1995.

M. J. Miller and J. I. Powell. A quantitative comparison of DNA sequence assembly programs. *J. Comput. Biol.*, 1(4):257–269, 1994.

Eleven sequence assembly programs are compared for their accuracy and the reproducibility with which they assemble DNA fragments into a completed sequence.

A strongly idealized formalization of sequence assembly assumes that fragment orientation is known and no errors occur. The resulting problem is the *Shortest Common Superstring Problem.* Although this problem is NP-hard (see [4]), simple greedy algorithms seem to do quite well. There is a large number of papers analyzing greedy procedures and developing approximation algorithms for the shortest common superstring problem. We list only a few of them.

J. Tarhio and E. Ukkonen. A greedy approximation algorithm for constructing shortest common superstrings. *Theoretical Comput. Sci.*, 57:131–145, 1988.

J. Turner. Approximation algorithms for the shortest common superstring problem. *Inform. and Computation*, 83:1–20, 1989.

These two papers independently show that a greedily constructed superstring is close to optimal in the sense that it achieves a compression of at least 1/2 that of a shortest superstring. The compression of a superstring is given by the number of symbols "saved" compared to simply concatenating all the strings. However, their results do not imply a performance guarantee with respect to optimal length.

M. Li. Towards a DNA sequence theory. In *Proc. 31-th Annual IEEE Symp. Found. Comput. Sci.*, pages 125–134, 1990.

obtained the first nontrivial bound for the superstring problem. His algorithm produces a superstring of at most $n \log n$, where $n$ is the length of the shortest superstring containing the given fragments.

A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis. Linear approximation of shortest superstrings. *J. ACM*, 41(4):630–647, 1994.

give the first linear approximation factor with respect to length. The authors show that the greedy algorithm leads to a superstring with length at most $4n$, and give a variation of the algorithm leading to an improved approximation factor $3n$. Moreover, they show that the shortest common superstring problem is MAX SNP-hard.

C. Armin and C. Stein. Shortest superstrings and the structure of overlapping strings. *J. Comput. Biol.*, 2(2):307–332, 1995.

This is one of the most recent papers in a series of papers which subsequently improved the approximation factor for the shortest common superstring problem. Here, an approximation factor of 2.75 is obtained by using the structure of strings with large amounts of overlap. The algorithm runs in time $O(|S| + k^3)$, where $k$ is the number of given strings and $S$ is the sum of the lengths of all the strings.

T. Jiang, Z. Jiang, and D. Breslauer. Rotation of periodic strings and short superstrings. Technical Report, Max-Planck-Institut f. Informatik, Saarbrücken, Germany, May 1996.

Recently, Jiang, Jiang and Breslauer suggested an approximation algorithm for the shortest common superstring with approximation factor 2.667 (2.596 in the near future). Their algorithm is simpler than the previous approximation algorithms that lead to a factor less than three.

Most of the above authors conjecture the existence of an approximation algorithm of factor two, and that the greedy algorithm itself is a candidate for it, because no example is known where the greedy algorithm does worse.

## 5.2 Sequencing by Hybridization

A method for DNA sequencing that requires considerable computational support is *Sequencing by Hybridization* (SBH). Hybridization is the process whereby two single-stranded DNA molecules form a double helix because of complementary base sequences in them. In SBH the basic operation is hybridizing a short piece of DNA of length $l$, called a probe or $l$-tuple, to a single-stranded target DNA sequence. Thus, an SBH-probe will yield a hybridization signal if there is a substring of the target DNA that matches the probe. In SBH this is done for all probes of a given small length in parallel providing information regarding which $l$-tuples occur in the target sequence. The computational task is to reconstruct the target DNA sequence from this information.

P. A. Pevzner. $l$-tuple DNA sequencing : a computer analysis. *J. Biomolecular Struct. Dynamics*, 7:63–73, 1989.

reduces the SBH reconstruction problem to the Eulerian path problem in a subgraph of the de Bruijn graph. The author also considers the occurence of erroneous signals (false positives) and missing signals (false negatives), and the case when repeats of length $l$ occur in the target DNA.

R. J. Lipshutz. Maximum likelihood DNA sequencing by hybridization. *J. Biomolecular Struct. Dynamics*, 11:637–653, 1993.

The author suggests a maximum likelihood method for the SBH reconstruction problem in the presence of false positives and false negatives and reduces SBH reconstruction to the graph matching problem.

P. A. Pevzner and R. J. Lipshutz. Towards DNA sequencing chips. In *Proc. 19-th Int. Conf. Math. Found. Comp. Sci., Lecture Notes in Comp. Sci. 841*, pages 143–158, 1994.

This article gives a survey of the state of the art in sequencing by hybridization through 1994.

Recently, modifications of sequencing by hybridization have been proposed to reduce ambiguities in sequence reconstruction:

S. Hannenhalli, W. Feldman, H. F. Lewis, S. S. Skiena, and P. A. Pevzner. Positional sequencing by hybridization. *CABIOS*, 12(1):19–24, 1996.

The authors consider the case where additional information about the approximate position of each $l$-tuple in the unknown DNA fragment is given. No polynomial algorithms for the *Positional Sequence Reconstruction Problem* are known. A special case is the *Bounded Positional SBH Reconstruction Problem*, where the range of positions for each $l$-tuple is bounded by a small fixed number. For this case the authors present two polynomial time algorithms.

R. M. Idury and M. S. Waterman. A new algorithm for DNA sequence assembly. *J. Comput. Biol.*, 2(2):291–306, 1995.

This article suggests the application of the SBH-related computational techniques to fragment assembly.

## 5.3 Digest Mapping

Restriction enzymes cleave a DNA molecule at an instance of a short specific pattern. Although not very precisely, the lengths of the fragments resulting from digestion by a restriction enzyme can be measured. Using two enzymes, say $A$ and $B$, first individually and then using $A$ and $B$ together, the experimentalist obtains three sets of fragments. These constitute a source of information for determination of the layout of the fragments on the underlying DNA strand. Let the cut sites for enzyme $A$ and $B$ be $a_1 < a_2 < \ldots < a_k$ and $b_1 < b_2 < \ldots < b_l$, respectively. Applying enzymes $A$ and $B$ simultaneously will cut at all these sites, say $c_1 < c_2 < \ldots < c_{k+l}$. This is the so-called double digest. The biological experiment yields the sets of fragment lengths $\bar{A} = \{a_1, a_2 - a_1, \ldots, N - a_k\}$, $\bar{B} = \{b_1, b_2 - b_1, \ldots, N - b_l\}$, and $\bar{C} = \{c_1, c_2 - c_1, \ldots, N - c_{k+l}\}$, where $N$ is the length of the target DNA. The task is now to reconstruct the restriction sites from the given sets $\bar{A}$, $\bar{B}$ and $\bar{C}$. More precisely, the *Double Digest Problem* is the following: Find an ordering of the elements in $\bar{A}$ and an ordering of the elements in $\bar{B}$ such that the double-digest implied by these orderings is $\bar{C}$. The papers quoted below not only show that it is difficult to produce a solution to this problem but that, additionally, the number of solutions generally is huge.

L. Goldstein and M. S. Waterman. Mapping DNA by stochastic relaxation. *Adv. Appl. Math.*, 8:194–207, 1987.

show that it is NP-complete to decide if any solution of a given double digest instance exists. They also show that the number of solutions that produce the same single and double digests increases exponentially with the length of the DNA segment when the enzyme sites are modelled by a random process.

W. Schmitt and M. S. Waterman. Multiple solutions of DNA restriction mapping problem. *Adv. Appl. Math.*, 12:412–427, 1991.

The authors suggest partitioning the entire set of maps into equivalence classes. But still the number of equivalence classes grows very fast, as observed by the following author.

P. A. Pevzner. DNA physical mapping and alternating eulerian cycles in colored graphs. *Algorithmica*, 13(1-2):77–105, 1995.

The author studies the combinatorics of multiple solutions to the double digest problem and shows that the solutions are closely associated with alternating Eulerian cycles in colored graphs. Furthermore he gives a complete characterization of equivalent maps as introduced by Schmitt and Waterman.

S. Ho, L. Allison, and C. N. Yee. Restriction site mapping for three or more enzymes. *Comp. Appl. Biosci.*, 6:195–204, 1990.

The authors generalize the double digest problem to more than two enzymes.

G. Zehetner, A. Frischauf, and H. Lehrach. *Approaches to restriction map determination*, pages 147–164. IRL Press, Oxford, 1988. M. J. Bishop and C. J. Rawlings (eds.), Nucl. Acid and Protein Sequence Analysis, Practical Approaches.

This article gives a survey on commonly used methods for attacking this problem.

In the *Partial Digest Problem* only a single enzyme, say $A$, is used to cleave several clones of a segment. Let $a_1 < a_2 < \cdots < a_k$ be the set of the restriction sites. In contrast to a double digest experiment, the digestion process is stopped earlier, such that not all of the restriction sites are cut. Again, the task is to reconstruct the restriction sites from the given fragment length data. The *Probed Partial Digest Problem* is similar, except that a site $p$ on the DNA segment is labeled, and the sizes of only those restriction fragments are measured that contain this site. Both problems can be reduced to polynomial factorization [7], which is "unlikely to be NP-complete" ([Karp], see above).

Like in the double digest problem the number of solutions can grow fast for these problems. The following two papers deal with this:

S. S. Skiena, W. D. Smith, and P. Lemke. Reconstructing sets from interpoint distances. In *Proc. 6-th Ann. Symp. Computational Geometry*, pages 332–339, 1990.

L. A. Newberg and D. Naor. A lower bound on the number of solutions to the exact probed partial digest problem. *Adv. Appl. Math.*, 14:172–185, 1993.

The authors show how the degree of ambiguity can grow as a function of the number of restriction sites.

### 5.4 Mapping Using Hybridization Data

For large pieces of DNA digest mapping has mostly been superseded by mapping techniques that are based on experimental determination of overlap between fragments (also called *clones*). This overlap data can be represented as a graph in which vertices are clones and there is an edge between two vertices if the corresponding clones overlap. In the absence of error, this graph is an interval graph [6].

Several experimental methods to obtain the required overlap information are in use. One approach detects certain, very specific sequence segments in the fragments. This is done using probes hybridizing to specific bits of the sequence. The hope is that two clones share a probe if and only if they overlap. If a probe is absolutely unique it is called a *Sequence Tagged Site* (*STS*). Ideally, the matrix describing which clones contain such a probe will have the consecutive ones property [6]. Hence, the correct orderings can be found very easily using the PQ-tree data structure [2] to generate the set of all arrangements of probes consistent with the data. Alternatively, there is another experiment that allows determination of overlap between two clones directly.

However, both techniques are error prone. In practice, false positive overlaps will be reported and overlaps may not be recognized, i.e., there are false negatives. A further complication is that sometimes two clones are merged into one. This phenomenon is called chimerism.

M. S. Waterman and J. R. Griggs. Interval graphs and maps of DNA. *Bull. Math. Biol.*, 48(2):189–195, 1986.

study a model that combines digest mapping with information on overlap between the fragments. They introduced the representation of maps as interval graphs [6] and characterize the interval graphs arising in the specific experimental setup modeled in this paper. A simple linear time algorithm for recognizing and representing the data in interval representation is given.

Since in practice the overlap data is error prone the question arises if there exists an interval graph that is in some sense “close” to the given overlap data.

M. C. Golumbic, H. Kaplan, and R. Shamir. On the complexity of DNA physical mapping. *Adv. Appl. Math.*, 15:251–261, 1994.

Given some “noisy” and some correct part of the overlap data, the authors show that the following problem is NP-complete. Is there an interval graph induced by $E_1$ satisfying $E_0 \subseteq E_1 \subseteq E_2$ for given edge sets $E_0$ and $E_2$ ($E_0 \subseteq E_2$) (*Interval Sandwich Problem*).

H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping. In *Proc. 35-th IEEE Symp. Found. Comp. Sci.*, pages 780–791, 1994.

First, the authors consider the case of unidentified overlaps. Although the problem of building a map with fewest errors is NP-hard (*Proper Interval Graph Completion Problem*), the authors present a linear time algorithm which gives an augmenting set with no more than $k$ edges ($k$ fixed) if one exists. Observing that the arising interval graphs have small clique size, the authors use this fact to present a polynomial time algorithm for the proper interval graph completion problem with bounded clique size.

M. R. Fellows, M. T. Hallett, and W. T. Wareham. DNA physical mapping: Three ways difficult. In *Proc. European Symp. Algorithms, Lecture Notes Comp. Sci. 726*, pages 157–168, Berlin, 1993. Springer-Verlag.

consider the case of digest mapping when $k$ enzymes are used and noisy overlap data between the fragments are given. This is a generalization of the model used by Waterman and Griggs (1986). They study the following problem: Given a graph $G$ and a coloring $c$ of the vertices to $k$ colors, is there a supergraph of $G$ which is properly colored by $c$ and which is an interval graph? It is shown that this problem is NP-complete, and is not fixed-parameter tractable, i.e., it cannot be solved in time $f(k)n^{\alpha}$, where $\alpha$ is independent of $k$ (unless an apparently resistant problem can be solved).

The following articles give algorithms for constructing maps using STS probes.

E. D. Green and P. Greene. Sequence-tagged site (STS) content mapping of human chromosomes: Theoretical considerations and early experiences. *PCR Methods and Appl.*, pages 77–90, 1991.

In this article some background information on STS mapping is given. Furthermore, the authors discuss a strategy for developing clone-based STS maps of chromosomes.

D. S. Greenberg and S. Istrail. Physical mapping by STS hybridization: Algorithmic strategy and the challenge of software evaluation. *J. Comput. Biol.*, 2(2):219–273, 1995.

develop some algorithmic theory of the mapping process, and propose a performance evaluation procedure. Furthermore, they suggest various combinatorial optimization problems such as the hamming distance travelling salesman problem that could be useful for solving the practical mapping problem.

F. Alizadeh, R. M. Karp, D. K. Weisser, and G. Zweig. Physical mapping of chromosomes using unique probes. *J. Comput. Biol.*, 2(2):159–184, 1995.

The authors present several combinatorial methods for reconstructing a DNA fragment in the presence of errors. The methods include techniques for the hamming distance travelling salesman problem, and simulated annealing as well as screening methods for detecting errors in the given data.

Several experimenters have approached mapping using probes that do not satisfy the uniqueness condition.

F. Alizadeh, R. M. Karp, L. A. Newberg, and D. K. Weisser. Physical mapping of chromosomes: A combinatorial problem in molecular biology. *Algorithmica*, 13(1-2):52–76, 1995.

present the first result that effectively addresses the ordering problem for mapping with hybridization fingerprints and non-unique probes. The authors introduce approximations to a likelihood function quantifying overlap information. This leads to optimization problems that are reasonably tractable in practice, although they are NP-hard.

R. Mott, A. Grigoriev, J. H. E. Maier, and H. Lehrach. Algorithms and software tools for ordering clone libraries: application to the mapping of the genome of schizosaccharomyces pombe. *Nucl. Acids Res.*, 21(8):1965–1974, 1993.

The authors describe a complete set of software tools for mapping of a genome that has been successfully applied to the genome of fission yeast.

# 6 Protein Threading and Lattice Models

The protein folding problem is the problem of predicting the three-dimensional fold that a given one-dimensional amino acid chain assumes. There is a vast amount of literature on the protein folding problem. For an overview see, e.g.:

G. D. Fasman. *Prediction of protein sturctures and the principles of protein conformation.* Plenum Press, 1989.

J. Kenneth M. Merz and S. M. L. Grand. *The protein folding problem and tertiary structure prediction.* Birkhäuser, 1994.

## 6.1 Threading

The *Threading Problem* is a generalization of the pairwise sequence alignment problem where for one of the two proteins a three-dimensional structure is given. An alignment thus implies spatial proximity between certain amino acids which is in turn weighted by a so-called pair-potential. The resulting optimization problem attempts to find an alignment that minimizes the sum over all implied pair-potentials.

R. H. Lathrop. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Engineering*, 7:1059–1068, 1994.

has shown that the Threading Problem is NP-complete (by reduction to One-in-Three 3SAT).

A branch-and-bound algorithm for a slightly modified version of threading is given in

R. H. Lathrop and T. F. Smith. Global optimum protein threading with gapped alignment and empirical pair score functions. *J. Mol. Biol.*, 255:641–665, 1996.

Other heuristic approaches to threading are summarized in

M. Sippl. Knowledge-based potentials for proteins. *Current Opinion in Structural Biology*, 5:229–235, 1995.

## 6.2 Lattice Models

Lattice Models are discrete, in most cases strongly simplified versions of the protein folding problem.

R. Unger and J. Moult. Finding the lowest free energy conformation of a protein is an NP-hard problem: Proof and implications. *Bull. Math. Biol.*, 55:1183 – 1198, 1993.

A. S. Fraenkel. Complexity of protein folding. *Bull. Math. Biol.*, 55:1199 – 1210, 1993.

give NP-completeness proofs for discretized versions of protein folding.

Algorithms for finding minimal energy structures for lattice models in proteins are summarized and tested in

K. Yue, M. Fiebig, P. D. Thomas, H. S. Chan, E. I. Shakhnovich, and K. A. Dill. A test of lattice protein folding algorithms. *Proc. Natl. Acad. Sci. USA*, 92:325 – 329, 1995.

W. E. Hart and S. Istrail. Fast protein folding in the hydrphobic-hydrophilic model within three–eights of optimal. In *Proc. 27-th Annual ACM Symp. Theory of Comput.*, pages 157–168, 1995.

give a factor $\frac{3}{8}$ approximation algorithm for lattice models.

# References


[1] Berman and Ramaiyer. Improved approximations for the steiner tree problem. *J. Algorithms*, 17:381–408, 1994.

[2] K. S. Booth and G. S. Lueker. Testing for consecutive ones property, interval graphs and planarity using PQ-tree algorithms. *J. Computer Syst. Sci.*, 13:335–379, 1976.

[3] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[4] J. Gallant, D. Maier, and J. Storer. On finding minimal length superstrings. *J. Computer Syst. Sci.*, 20:50–58, 1980.

[5] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[6] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, 1980.

[7] J. Rosenblatt and P. D. Seymour. The structure of homometric sets. *SIAM J. Algebraic Discr. Methods*, 3:343–350, 1982.

[8] A. Zelikovsky. The 11/6 approximation algorithm for the steiner problem on networks. *Algorithmica*, 9:463–470, 1993.