LU TP 00-32
June 25, 2002

A Monte Carlo Approach to Sequence Assembly

Erik Sandelin[1]

Complex Systems Division, Department of Theoretical Physics
Lund University, Sölvegatan 14A, S-223 62 Lund, Sweden
http://www.thep.lu.se/tf2/complex/

Submitted to *Bioinformatics*.

Abstract

Motivation:
Assembling shotgun sequencing data from repetitive DNA sequences is a non-trivial task. In existing sequence assembly methods repeats are resolved by either using statistical analyses to identify and separate fragments corresponding to repeats, or by using extra information, not contained in the fragments. In this paper we take a different approach. Using the simulated-tempering Monte Carlo method, we resolve repeats by performing an extensive search of the solution space.

Results:
The method is tested on two highly repetitive sequences with a two-copy and a three-copy repeat, respectively. We find that the method is able to correctly assemble these two sequences, except for a twofold degeneracy for the three-copy repeat sequence. The alternative solution obtained in this case is related by a simple symmetry to the correct one. The performance of the method is compared with that of simulated annealing. We find that simulated tempering is a competitive alternative to simulated

[1]To whom correspondence should be addressed

annealing.

Contact:
erik@thep.lu.se

Keywords:
Monte Carlo methods, simulated tempering, simulated annealing, shotgun sequencing, sequence assembly.

1 Introduction

Today it is not possible to accurately determine more than about 1000 consecutive base pairs of a DNA sequence (Waterman, 1995). This is a major limitation for large-scale sequencing projects. One approach to avoid this limitation is shotgun sequencing, which most notably has been used to successfully sequence the Drosophila genome (Myers *et al.*, 2000) and to produce a draft version of the human genome. In shotgun sequencing short fragments are randomly sampled from the target sequence. With sufficient oversampling, the original sequence can be inferred by assembling overlapping fragments. To find the correct assembly of the fragments is called the sequence assembly problem.

A large part of the DNA of complex organisms consists of repetitive sequences (Brown, 1999). These repeats present a challenge for assembly algorithms. Due to "false" overlaps, i.e. overlaps corresponding to fragments originating from different copies of a repeat, the repeats might produce incorrect solutions competing with the correct solution. To correctly assemble repetitive sequences, assembly algorithms must be able to discriminate these solutions.

Most sequence assembly algorithms are based on some variation of the greedy algorithm (Green, 1996; Huang, 1992; Huang, 1996; Kim and Segre, 1999; Myers, 1995; Sutton *et al.*, 1995). In the greedy approach, the fragments are assembled by repeatedly merging the pair of fragments with highest overlap. This is a very fast method. However, it typically explores only a tiny part of the solution space, and in the presence of repeats in the target sequence it can end up at an incorrect solution. Several greedy based methods which try to avoid this problem have been developed (Green, 1996; Huang, 1996; Kim and Segre, 1999; Myers, 1995; Sutton *et al.*, 1995). Some of these methods perform statistical analyses of the fragments in order to identify and separate fragments corresponding to false overlaps (Huang, 1996; Kim and Segre, 1999; Myers, 1995), while others use additional information, not contained in the fragments (Green, 1996; Sutton *et al.*, 1995).

A different approach to resolve repeats, without using additional information or separating fragments corresponding to false overlaps, is to use stochastic

methods to perform an extensive search of the solution space. In this study we apply the simulated-tempering Monte Carlo method (Lyubartsev *et al.*, 1992; Marinari and Parisi, 1992) to the sequence assembly problem. Simulated tempering is related to simulated annealing (Kirkpatrick *et al.*, 1983), which has been used previously for sequence assembly (Burks *et al.*, 1994a; Burks *et al.*, 1994b; Churchill *et al.*, 1993). Simulated tempering differs from simulated annealing in that it uses stochastic moves in temperature. It has been applied, for example, to protein (Irbäck and Potthast, 1995) and spin glass (Marinari and Parisi, 1992) models.

In this paper we test simulated tempering on two highly repetitive sequences, with a two-copy and a three-copy repeat, respectively. It is found that simulated tempering successfully assembles the two-copy repeat sequence. For the three-copy repeat sequence it finds two solutions, the correct solution and a solution with the two internal non-repetitive segments interchanged. The efficiency of simulated tempering is compared to that of simulated annealing (Kirkpatrick *et al.*, 1983). For comparison, we also apply two greedy based methods to the same problems.

This paper is organized as follows. Section 2 describes the sequence assembly problem. In Sec. 3, brief descriptions of the greedy algorithm and the simulated-tempering method are given. Section 4 contains our results, and a summary is given in Sec. 5.

2 Sequence Assembly

As mentioned in the introduction, shotgun sequencing avoids the limitations of present-day sequencing technology by randomly sampling short fragments from the target sequence. The target sequence is subsequently inferred by assembling the fragments guided by their pairwise overlaps.

The problem studied in this paper is to reconstruct a target sequence with G base pairs, by assembling N fragments into a continuous sequence. The mean length of the fragments is denoted by $\overline{L}$, and an assembly is described by a particular ordering, or layout, of the fragments. A layout produces a consensus sequence (see Fig. 1), which is the proposed solution to the

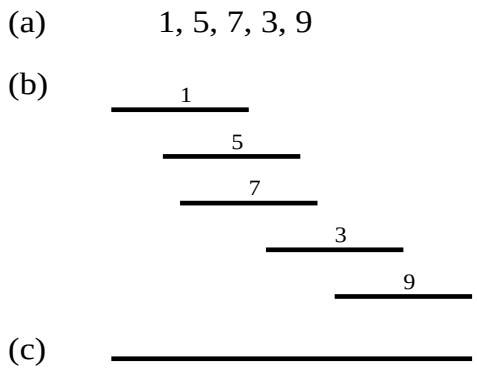

Figure 1: A schematic picture of a layout and the corresponding consensus sequence. The "projection" of a layout [(a) and (b)] produces a consensus sequence (c).

problem. The length of the consensus sequence is denoted by G_c. The success of sequence assembly depends on the amount of oversampling, and a useful quantity is the mean coverage $\overline{C}$, defined as $\overline{C} = N\overline{L}/G$.

The assembly procedure is hampered by

- Errors in the sequencing of the fragments.
- Unknown orientation of the fragments. We do not know from which of the two DNA strands a fragment originates.
- Incomplete coverage of the target sequence. Due to the stochastic nature of the sampling procedure, parts of the target sequence might not be covered at all.

In this study we consider a semi-idealized version of the sequence assembly problem, where the target sequence is completely covered, the orientation of the fragments is known, all fragments have the same length, 400bp, and the fragments are sequenced with an error rate of 2% and contain no insertions or deletions.

To calculate the overlaps of the fragments, we use a scoring of 3 for a match and -1 for a mismatch. Since we do not consider insertions or deletions, no dynamical programming is needed.

3 Methods

3.1 Greedy

Most sequence assembly methods rely on some variation of the greedy algorithm (Green, 1996; Huang, 1992; Huang, 1996; Kim and Segre, 1999; Myers, 1995; Sutton *et al.*, 1995). In the greedy algorithm, taking the pair of fragments with highest overlap as starting point, the layout is constructed by successively adding the fragment which has the highest overlap with the assembled fragments. This corresponds to a deterministic local optimization scheme and is a very fast algorithm. Moreover, if the target sequence contains no repeats, it will almost always lead to the correct layout. However, if the target sequence does contain repeats, the local optimization scheme of the greedy algorithm might fail due to "false" overlaps, i.e. overlaps corresponding to fragments originating from different copies of a repeat.

Since repeats are common in the DNA of complex organisms ($\sim 14\%$ of the human genome (Brown, 1999)), ways to deal with repeats must be invented. Indeed, several methods to handle repeats have been developed in the context of greedy algorithms (Green, 1996; Huang, 1996; Kim and Segre, 1999; Myers, 1995; Sutton *et al.*, 1995). In order to identify false overlaps, some methods (Huang, 1996; Kim and Segre, 1999; Myers, 1995) use statistical analyses of the given fragments, without any further information. Other methods (Green, 1996; Sutton *et al.*, 1995) require additional information, not contained in the fragments, to disentangle the repeats.

3.2 Stochastic Methods

As discussed in the previous section, the greedy algorithm can be viewed as a local optimization method. Typically, it explores only a tiny fraction

of all possible solutions. For repetitive sequences, a more extensive search is required. This calls for stochastic methods, and an obvious candidate is simulated annealing (Kirkpatrick *et al.*, 1983), which indeed has been applied to this problem (Burks *et al.*, 1994a; Burks *et al.*, 1994b; Churchill *et al.*, 1993).

In the simulated-annealing approach to the sequence assembly problem, one defines an "energy" function E, or "objective" function, based on the overlaps of the fragments. A convenient choice is

$$E = -\sum_{i=1}^{N-1} O_{i,i+1}, \tag{1}$$

where $O_{i,i+1}$ is the overlap between the fragments at position i and $i+1$ in the layout. Simulated annealing tries to minimize this energy function using stochastic reshuffling of the fragments.

In the presence of repeats, there can be incorrect layouts with energies similar to that of the correct layout. In this situation it is unavoidable that simulated annealing sometimes generates incorrect solutions. However, the incorrect solutions can often be identified by observing regions in the consensus sequence with unusually high coverage, or by comparing the lengths of the consensus sequence and the target sequence. This means that we can use all fragments in the layout construction. There is no need to treat false overlaps separately.

Another appealing feature of stochastic methods is the possibility to include ancillary information in the layout construction (Burks *et al.*, 1994a). This is done by adding terms to the energy function which penalize solutions in conflict with this information.

It should be mentioned that simulated annealing is not the only other stochastic method that has been applied to sequence assembly. For example, Parsons and Johnson (Parsons and Johnson, 1995) have used a genetic algorithm.

3.3 Simulated Tempering

In simulated annealing, stochastic reshuffling moves, or Monte Carlo (MC) steps, are combined with a deterministic cooling scheme. The system is taken through a predefined set of temperatures, and a fixed number of MC steps is performed at each temperature. The hope is that the stochastic moves will help the system to avoid local minima, and that the cooling scheme will guide the system towards the global minimum.

Simulated tempering (Lyubartsev *et al.*, 1992; Marinari and Parisi, 1992) also uses a set of predefined temperatures. However, here not only the reshuffling moves but also the moves in temperature are stochastic, which in particular means that both upward and downward moves in temperature are allowed. The hope is that this will help the system to escape from local minima. Simulated tempering has earlier been applied to problems with rugged energy landscapes containing multiple minima, such as models of proteins (Irbäck and Potthast, 1995) and spin glasses (Marinari and Parisi, 1992).

To be more precise, in simulated tempering one simulates the joint probability

$$P(r,k) \propto \exp\left[-g_k - E(r)/T_k\right] , \tag{2}$$

where r denotes a particular layout and k is a temperature index. This distribution contains two sets of parameters; the temperatures that the system is allowed to visit, $\{T_k\}$, and a set of tunable parameters, $\{g_k\}$. In this study, the temperatures T_k were chosen according to (Hansmann and Okamoto, 1997)

$$T_k = T_{\min}\left(\frac{T_{\max}}{T_{\min}}\right)^{(k-1)/(K-1)} , \tag{3}$$

where $T_{\min} = T_1$ and $T_{\max} = T_K$ denote the lowest and highest allowed temperatures, respectively. The lowest temperature $T_{\min}$ should be chosen such that the system has a high probability of visiting the low-energy configurations, while $T_{\max}$ is chosen such that the system has good mobility in configuration space. The number of intervening temperatures should be chosen such that moves in temperature are accepted with a reasonable probability. In this study, a set of short trial runs were used to choose $T_{\min}$, $T_{\max}$ and K (see Table 1).

The g_k parameters govern the marginal distribution in k, $P(k)$. Hence, to have good mobility in k, it is essential to make a careful choice of $\{g_k\}$.

	S1	S2
K	10	15
$T_{\min}$	145.8	112.0
$T_{\max}$	189.0	200.0

Table 1: Simulation parameters for the two data sets S1 and S2. Shown are the number of temperatures, K, and the highest, $T_{\max}$, and lowest, $T_{\min}$, temperatures.

This is typically achieved by means of trial runs (Irbäck and Potthast, 1995). The actual simulation of the distribution $P(r, k)$ is done by using ordinary separate updates of r and k. In this study, the update of r is done by selecting a pair of fragments at random and try to swap their positions in the layout. Temperature moves are to adjacent temperatures, $k \to k \pm 1$.

4 Results

In this study we test simulated tempering on two sequences, both of which contain repeats. One of the sequences has a two-copy repeat and the other a three-copy repeat. For the two-copy repeat sequence we find that simulated tempering is able to correctly assemble the data set without prior separation of fragments corresponding to false overlaps. For the three-copy repeat sequence, simulated tempering finds two solutions, the correct solution and a solution where the order of the two internal non-repetitive segments is interchanged (see Fig. 3 below). The efficiency of simulated tempering is compared to that of simulated annealing.

To monitor the progress of the simulations, we measure three quantities: the energy E of the layout, the length G_c of the consensus sequence, and the similarity M of the consensus sequence and the target sequence. To define M we first aligned the two sequences, using the same scoring scheme as for the overlaps. We then take M as the number of matches in the alignment divided by the length of the alignment.

As mentioned in section 3.2, the presence of repeats might produce incorrect

layouts with low energies, which make it impossible to identify correct solutions by just monitoring the energy of the layouts. However, often these incorrect low-energy layouts produce consensus sequences that are shorter than the correct solutions. The correct solutions can then be identified by monitoring G_c and compare it with the length of the target sequence, G.

4.1 Sequences

Starting with the 2126bp HUMATPK01 (accession code M55090) human brain DNA sequence (Sverdlov *et al.*, 1987) from GenBank (Release 116.0) (Benson *et al.*, 1999), we constructed two artificial sequences with a two-copy and a three-copy repeat, respectively, by duplicating a 500bp long region and inserting it at different places in the original sequence (see Table 2). By randomly choosing uniformly distributed starting points along the sequences, we generated a set of fragments for each target sequence. The number of fragments was chosen to give a mean coverage $\overline{C}$ of 7.3. Errors in the fragments were introduced by substituting base pairs with a probability of 2%.

Details of these two data sets can be found in Table 2. The fragments in these data sets completely cover the interior of the target sequences, but, as indicated by the differences between G and the length G_0 of the correct consensus sequence, a few bases at the ends are not covered.

4.2 Greedy

Although this study is focused on stochastic methods, it is interesting to see how greedy based methods perform on these two data sets. To this end, we consider two different methods. The first one amounts to taking an arbitrary pair of fragments as starting point, and then successively add the fragments with highest overlap. This is repeated for all possible starting points. The second algorithm is CAP2 (Huang, 1996; software downloaded by ftp from the author). CAP2 is meant specifically for repetitive sequences. However,

	S1	S2
G	2626	3126
1st copy	201–700	201–700
2nd copy	1301–1800	1301–1800
3rd copy	–	2301–2800
% repeats	38	47
N	48	57
L	400	400
$\overline{C}$	7.3	7.3
E_0	-47279	-56294
G_0	2619	3078
M_0	0.998	0.998

Table 2: Details of the two data sets S1 and S2. Shown are the length G of the target sequence, the positions of the repeats, the percentage of repeats in the target sequence, the number N of fragments, the fragment length L, the mean coverage $\overline{C}$, the energy E_0 of the correct layout, the length G_0 of the correct consensus sequence, and the similarity M_0 between the correct consensus sequence and the target sequence.

it resolves repeats using small differences between the copies of the same repetitive sequence. Such differences do not exist in our examples.

By repeating the first method for all possible starting points it turned out that correct consensus sequences could be obtained. The quality of these solutions was almost as high as for the stochastic methods (see below), and the method is faster than these. However, the energies of the solutions were significantly higher than those found with simulated tempering. A closer examination revealed that this is due to an uneven distribution of fragments originating from repeats, leaving some regions covered by only one fragment, and due to incorrect positioning of several repeat-flanking fragments. We take this as an indication that the method might run into severe difficulties for more general data sets. The solutions obtained by simulated tempering did not have these properties.

The CAP2 algorithm turned out to fail for both data sets. In both cases, it produced two disjoint sequences with low similarities to the target sequences. Most probably, the explanation for this is that the repeats are 100% identical in our examples.

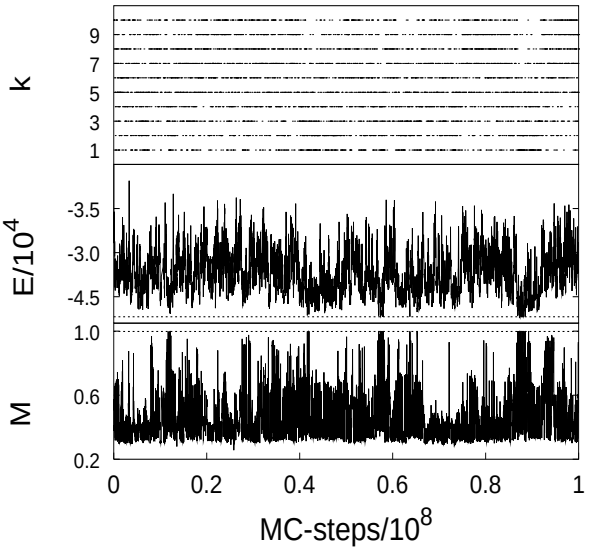

Figure 2: Runtime plots of the temperature index k, energy E, and similarity M to target sequence, for the S1 data set. The dashed lines are the corresponding values for the correct layout. Shown is only a third of the actual simulation.

4.3 Data Set S1

In simulated tempering, to properly sample the solution space, the g_k parameters have to be carefully tuned to ensure that the system visits all temperatures with approximately the same frequency. For the S1 data set, tuning of these parameters required about 2 CPU minutes (on a Pentium III 500 MHz processor). We then made a production run of 3×10^8 MC steps, corresponding to 30 CPU minutes. Runtime plots of E, M, and the temperature index k, for the first 10^8 MC steps is shown in Fig. 2.

From the top panel it can in particular be seen that the g_k's were indeed properly tuned. The middle panel shows the evolution of the energy, where the dashed line indicates the energy of the correct layout. The system makes several independent visits to the lowest energies, indicating that the solution space is properly sampled. A comparison with data for M in the bottom panel, shows that these low-energy solutions indeed produce correct consen-

E	M	G_c/G_0	G_c/G
-47287	0.998	1.000	0.997
-47283	0.998	1.000	0.997
-47279	0.998	1.000	0.997
-47275	0.998	1.000	0.997
-47271	0.998	1.000	0.997

Table 3: The five solutions found with lowest energies for the S1 data set. Shown are their energy E, their similarity M with the target sequence, their lengths G_c, in fractions of the target sequence length G and the correct consensus sequence length G_0.

sus sequences.

Table 3 shows the five solutions found with lowest energies. It can be seen that they all produce consensus sequences with the same similarity to the target sequence as the consensus sequence corresponding to the correct layout (see Table 2). The existence of several layouts producing the same correct consensus sequence reflects the fact that interchanging two fragments originating from the interior of repeats does not change the final result.

As a comparison, we also applied the simulated-annealing method to the S1 data set. Starting at the highest temperature used in the simulated-tempering run, and using the same set of temperatures, we performed a total of 10 runs, each starting in a different random layout. This number of runs corresponds to the number of independent visits to the lowest energies in the simulated-tempering run. By using 3×10^5 MC steps per temperature, the total CPU time for the 10 runs was approximately the same as for the simulated-tempering run. A layout producing a correct consensus sequence was found in only one of the ten runs, which shows that simulated tempering is a competitive alternative to simulated annealing.

4.4 Data Set S2

For the S2 data set we used a different set of temperatures (see Table 1). The lengths of the tuning and production runs were the same as for the S1

No screening				*Screening*			
E	M	G_c/G_0	G_c/G	E	M	G_c/G_0	G_c/G
-56791	0.979	0.850	0.837	-56278	0.624	1.000	0.985
-56787	0.979	0.850	0.837	-56258	0.624	1.000	0.985
-56775	0.979	0.850	0.837	-56258	0.998	1.000	0.985
-56767	0.979	0.850	0.837	-56254	0.624	1.000	0.985
-56755	0.979	0.850	0.837	-56254	0.998	1.000	0.985

Table 4: The five solutions with lowest energies found for the S2 data set, before and after screening for sequence lengths between 95 and 100% of the target sequence length, G.

data set.

In contrast to the S1 data set, the lowest-energy solutions for the S2 data set correspond to incorrect consensus sequences. From Table 4, which shows the five lowest-energy solutions found, it can be seen that these solutions correspond to consensus sequences with lengths that are only 83% of the target sequence length. Furthermore, although the similarity with the target sequence is high, it is not as high as for the correct solution (see Table 2).

However, by screening for consensus sequence length, it turns out that it is possible to find correct solutions among the low-energy solutions. The right-hand side of Table 4 shows the five lowest-energy solutions with consensus sequence lengths between 95 and 100% of the target sequence length G. Table 4 shows that two of these solutions correspond to correct consensus sequences, while three of the solutions have the same length as the correct sequence, but a match of only 0.62. A closer examination of the latter solutions reveals that the repeats and the segments inbetween them are correctly assembled, but that the two segments which are flanked on both sides by repeats have been interchanged (see Fig. 3), a situation which our energy function cannot discriminate from the correct solution.

Choosing the set of temperatures, the number of MC steps, 1.5×10^5, and the number of runs, 11, in the same manner as for the S1 data set, we applied simulated annealing to the S2 data set too. Screening for consensus sequence length, as above, two of the runs found solutions producing incorrect consensus sequences with no simple relation to the correct solution. The other nine runs found solutions corresponding to the situation mentioned

above, with two internal segments interchanged.

5 Summary

In the presence of repeats in the target sequence, assembling shotgun data is a non-trivial task. Repeats produce incorrect assembly solutions competing with the correct solution, and it is a challenge for sequence assembly algorithms to discriminate these. The abundance of repeats in the DNA of complex organisms makes this a highly relevant problem.

Due to its speed and simplicity, the greedy algorithm is the most popular sequence assembly algorithm. In existing greedy based sequence assembly methods, repeats are resolved by either identifying and separating fragments corresponding to false overlaps, or by using extra information, not contained in the fragments.

In this paper we took a different approach. Using the stochastic method of simulated tempering, repeats were resolved by performing an extensive search of the solution space, without using extra information or separating fragments corresponding to false overlaps. The method was tested on two sequences with a two-copy and a three-copy repeat, respectively. It successfully solved the two-copy problem. For the three-copy problem, from the space of all possible solutions, it was able to single out two candidates, the correct solution and a solution with the two internal non-repetitive segments interchanged (see Fig. 3).

Our tests showed that simulated tempering is a competitive alternative to simulated annealing. Compared to simulated annealing, simulated tempering has the disadvantage that it contains an extra set of parameters that must be tuned. On the other hand, once this is done, it is sufficient to perform a single simulation; there is no need to restart the simulation from different initial layouts. Also, there is no cooling rate to be determined.

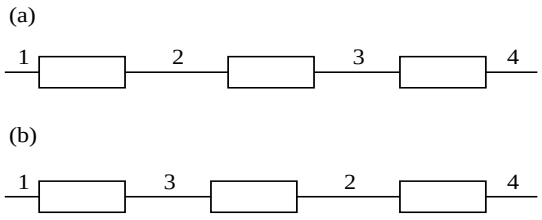

Figure 3: A schematic picture of the two solutions found for the S2 data set; (a) the correct solution and (b) the solution with the order of segment 2 and 3 interchanged. Boxes denote repeats and lines non-repetitive segments.

6 Acknowledgement

I would like to thank Anders Irbäck for valuable discussions and helpful comments on the manuscript. This work was supported by the Swedish Foundation for Strategic Research.

References

Benson, D.A., Boguski, M.S., Lipman, D.J., Ostell, J., Ouellette, B.F., Rapp, B.A. and Wheeler, D.L. (1999) GenBank. *Nucleic Acids Res.* **27**, 12–17.

Brown, T.A. (1999) *Genomes*. BIOS Scientific Publishers Ltd, Oxford.

Burks, C., Engle, M.L. and Parsons, R.J. (1994) Integration of Competing Ancillary Assertions in Genome Assembly. In Altman, R., Brutlag, D., Karp, P., Lathrop, R., and Searls, D. (eds). *International Conference on Intelligent System for Molecular Biology*, AAAI Press, Menlo Park, CA, pp. 62–69.

Burks, C., Engle, M.L. and Forrest, S. (1994) Stochastic Optimization Tools for Genomic Sequence Assembly. In Venter, J.C. (ed). *Automated DNA Sequencing and Analysis Techniques*, Academic Press, pp. 249–259.

Churchill, G., Burks, C., Eggert, M., Engle, M.L. and Waterman, M.S. (1993) Assembling DNA Sequence Fragments by Shuffling and Simulated Annealing. Technical Report LA-UR-93-2287, Los Alamos National Laboratory, Los Alamos, NM.

Green, P. (1996) PHRAP documentation.
http://bozeman.mbt.washington.edu/phrap.docs/phrap.html.
University of Washington, Seattle.

Hansmann, U.H.E. and Okamoto, Y. (1997) Numerical Comparisons of Three Recently Proposed Algorithms in the Protein Folding Problem. *J. Comput. Chem.* **18**, 920–933.

Huang, X. (1992) A Contig Assembly Program Based on Sensitive Detection of Fragment Overlaps. *Genomics*. **14**, 18–25.

Huang, X. (1996) An Improved Assembly Program. *Genomics*. **33**, 21–31.

Irbäck, A. and Potthast, F. (1995) Studies of an Off-Lattice Model for Protein Folding: Sequence Dependence and Improved Sampling at Finite Temperature. *J. Chem. Phys.* **103**, 10298–10305.

Kim, S. and Segre, A.M. (1999) AMASS: A Structures Pattern Matching Approach to Shotgun Sequence Assembly. *J. Comput. Biol.* **6**, 163–186.

Kirkpatrick, S., Gellat, C.D.Jr and Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science* **220**, 671–680.

Lyubartsev, A.P., Martsinovski, A.A., Shevkunov, S.V., and Vorontsov-Velyaminov, P.N. (1992) New Approach to Monte Carlo Calculation of the Free Energy: Method of Expanded Ensembles. *J. Chem. Phys.* **96**, 1776–1783.

Marinari, E. and Parisi, G. (1992) Simulated Tempering: A New Monte Carlo Scheme. *Europhys. Lett.* **19**, 451–458.

Myers, E.W. (1995) Toward Simplifying and Accurately Formulating Fragment Assembly. *J. Comput. Biol.* **2**, 275–290.

Myers, E.W. *et al.* (2000) A Whole-Genome Assembly of Drosophila. *Science* **287**, 2196–2204.

Parsons, R.J. and Johnson, M.E. (1995) DNA Fragment Assembly and Genetic Algorithms. New Results and Puzzling Insights. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T., and Wodakthe, S. (eds). *International Conference on Intelligent Systems in Molecular Biology*, AAAI Press, Menlo Park, CA, pp. 277–284.

Sutton, G.G., White, O., Adams, M.D. and Kerlavage, A.R. (1995) TIGR Assembler: A new tool for assembling large shotgun sequencing projects. *Genome Sci. Technol.* **1**, 9-19.

Sverdlov, E.D. *et al.* (1987) Family of human Na+, K+-ATPase genes. Structure of the gene of isoform alpha-III. *Dokl. Biochem.* **297**, 426–431.

Waterman, M. (1995) *An Introduction to Computational Biology.* Chapman & Hall, London.