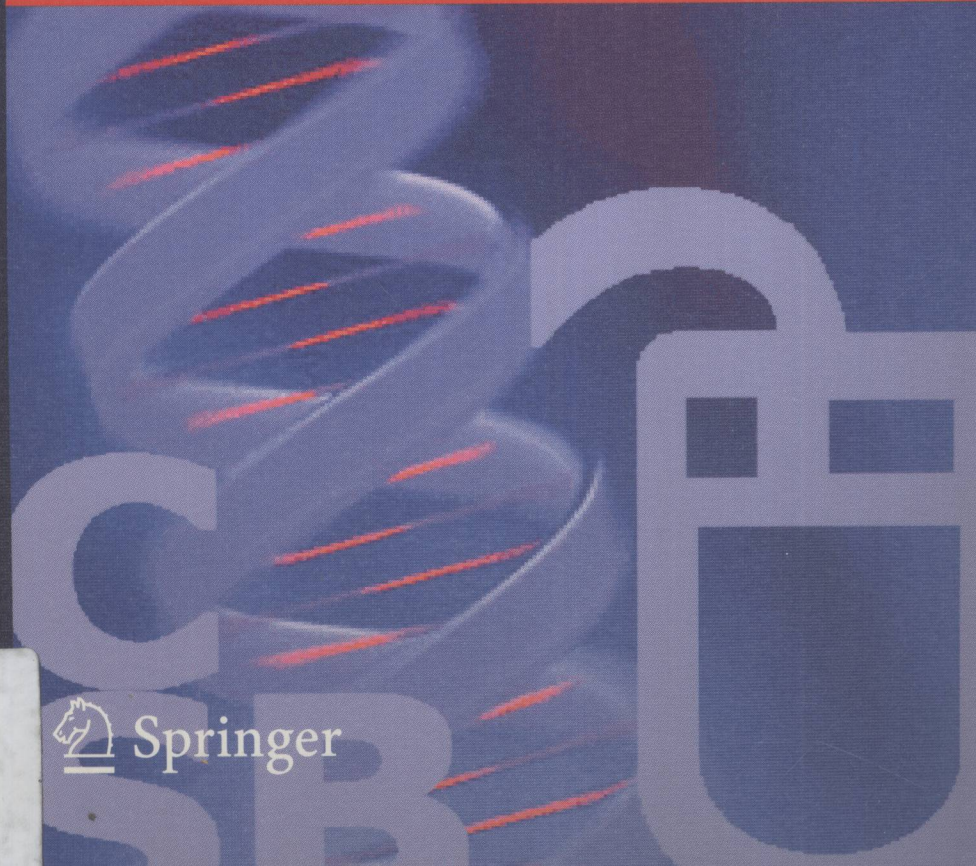
Journal Subline

LNBI 3737

Transactions on Computational Systems Biology III

Corrado Priami
Editor-in-Chief



Springer

Corrado Priami Emanuela Merelli
Pedro Pablo Gonzalez Andrea Omicini (Eds.)

Transactions on Computational Systems Biology III

Springer

Volume Editors

Emanuela Merelli
Università di Camerino
Via Madonna delle Carceri
62032 Carmerino, Italy
E-mail: emanuela.merelli@unicam.it

Pedro Pablo Gonzalez
Universidad Autonoma Metropolitana
Departamento de Matematicas Aplicadas y Sistemas
Unidad Cuajimalpa, Mexico
E-mail: ppgp@servidor.unam.mx

Andrea Omicini
DEIS, Alma Mater Studiorum, Università di Bologna
Via Venezia 52, 47023 Cesena, Italy
E-mail: andrea.omicini@unibo.it

Library of Congress Control Number: 2005937051

CR Subject Classification (1998): J.3, H.2.8, F.1

ISSN 0302-9743
ISBN-10 3-540-30883-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-30883-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11599128 06/3142 5 4 3 2 1 0

Lecture Notes in Bioinformatics 3737

Edited by S. Istrail, P. Pevzner, and M. Waterman

Subseries of Lecture Notes in Computer Science

Preface

In the last few decades, advances in molecular biology and in the research infrastructure in this field has given rise to the "omics" revolution in molecular biology, along with the explosion of databases: from genomics to transcriptomics, proteomics, interactomics, and metabolomics. However, the huge amount of biological information available has left a bottleneck in data processing: information overflow has called for innovative techniques for their visualization, modelling, interpretation and analysis. The many results from the fields of computer science and engineering have then met with biology, leading to new, emerging disciplines such as bioinformatics and systems biology. So, for instance, as the result of application of techniques such as machine learning, self-organizing maps, statistical algorithms, clustering algorithms and multi-agent systems to modern biology, we can actually model and simulate some functions of the cell (e.g., protein interaction, gene expression and gene regulation), make inferences from the molecular biology database, make connections among biological data, and derive useful predictions.

Today, and more generally, two different scenarios characterize the post-genomic era. On the one hand, the huge amount of datasets made available by biological research all over the world mandates for suitable techniques, tools and methods meant at modelling biological processes and analyzing biological sequences. On the other hand, biological systems work as the sources of a wide range of new computational models and paradigms, which are now ready to be applied in the context of computer-based systems.

Since 2001, NETTAB (the International Workshop on Network Tools and Applications in Biology) is the annual event aimed at introducing and discussing the most innovative and promising network tools and applications in biology and bioinformatics. In September 2004, the 4th NETTAB event (NETTAB 2004) was held in the campus of the University of Camerino, in Camerino, Italy. NETTAB 2004 was dedicated to "Models and Metaphors from Biology to Bioinformatics Tools". It brought together a number of innovative contributions from both bioscientists and computer scientists, illustrating their original proposals for addressing many of the open issues in the field of computational biology. Along with an enlightening invited lecture by Luca Cardelli (from Microsoft Research), the presentations and the many lively discussions made the workshop a very stimulating and scientifically profound meeting, which provided the many participants with innovative results and achievements, and also with insights and visions on the future of bioinformatics and computational biology.

This special issue is the result of the workshop. It includes the reviewed and revised versions of a selection of the papers originally presented at the workshop, and also includes a contribution from Luca Cardelli, presenting and elaborating

on his invited lecture. In particular, the papers published in this volume cover issues such as:

- data visualization
- protein/RNA structure prediction
- motif finding
- modelling and simulation of protein interaction
- genetic linkage analysis
- notations and models for systems biology

Thanks to the excellent work of the many researchers who contributed to this volume, and also to the patient and competent cooperation of the reviewers, we are confident that this special issue of the LNCS Transactions on Computational Systems Biology will transmit to the reader at least part of the sense of achievement, the dazzling perspectives, and even the enthusiasm that we all felt during NETTAB 2004. A special thanks is then due to the members of the Program Committee of NETTAB 2004, who allowed us, as the Workshop Organizers, to prepare such an exciting scientific program: Russ Altman, Jeffrey Bradshaw, Luca Cardelli, Pierpaolo Degano, Marco Dorigo, David Gilbert, Carole Goble, Anna Ingolfsdottir, Michael Luck, Andrew Martin, Peter McBurney, Corrado Priami, Aviv Regev, Giorgio Valle, and Franco Zambonelli.

Finally, the Guest Editors are very grateful to the Editor-in-Chief, Corrado Priami, for giving them the chance to work on this special issue, and also to the people at Springer, for their patient and careful assistance during all the phases of the editing process.

June 2005

Emanuela Merelli
Pablo Gonzalez
Andrea Omicini

LNCS Transactions on Computational Systems Biology – Editorial Board

Lecture Notes in Bioinformatics

Vol. 3745: J.L. Oliveira, V. Maojo, F. Martín-Sánchez, A.S. Pereira (Eds.), Biological and Medical Data Analysis. XII, 422 pages. 2005.

Vol. 3737: C. Priami, E. Merelli, P. P. Gonzalez, A. Omicini (Eds.), Transactions on Computational Systems Biology III. VII, 169 pages. 2005.

Vol. 3695: M.R. Berthold, R.C. Glen, K. Diederichs, O. Kohlbacher, I. Fischer (Eds.), Computational Life Sciences. XI, 277 pages. 2005.

Vol. 3692: R. Casadio, G. Myers (Eds.), Algorithms in Bioinformatics. X, 436 pages. 2005.

Vol. 3680: C. Priami, A. Zelikovsky (Eds.), Transactions on Computational Systems Biology II. IX, 153 pages. 2005.

Vol. 3678: A. McLysaght, D.H. Huson (Eds.), Comparative Genomics. VIII, 167 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005.

Vol. 3500: S. Miyano, J. Mesirov, S. Kasif, S. Istrail, P.A. Pevzner, M. Waterman (Eds.), Research in Computational Molecular Biology. XVII, 632 pages. 2005.

Vol. 3388: J. Lagergren (Ed.), Comparative Genomics. VII, 133 pages. 2005.

Vol. 3380: C. Priami (Ed.), Transactions on Computational Systems Biology I. IX, 111 pages. 2005.

Vol. 3370: A. Konagaya, K. Satou (Eds.), Grid Computing in Life Science. X, 188 pages. 2005.

Vol. 3318: E. Eskin, C. Workman (Eds.), Regulatory Genomics. VII, 115 pages. 2005.

Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004.

Vol. 3082: V. Danos, V. Schachter (Eds.), Computational Methods in Systems Biology. IX, 280 pages. 2005.

Vol. 2994: E. Rahm (Ed.), Data Integration in the Life Sciences. X, 221 pages. 2004.

Vol. 2983: S. Istrail, M.S. Waterman, A. Clark (Eds.), Computational Methods for SNPs and Haplotype Inference. IX, 153 pages. 2004.

Vol. 2812: G. Benson, R.D. M. Page (Eds.), Algorithms in Bioinformatics. X, 528 pages. 2003.

Vol. 2666: C. Guerra, S. Istrail (Eds.), Mathematical Methods for Protein Structure Analysis and Design. XI, 157 pages. 2003.

Table of Contents

Computer-Aided DNA Base Calling from Forward and Reverse Electropherograms

Valerio Freschi and Alessandro Bogliolo

STI - University of Urbino, Urbino, IT-61029, Italy
{freschi, bogliolo}@sti.uniurb.it

Abstract. In order to improve the accuracy of DNA sequencing, forward and reverse experiments are usually performed on the same sample. Base calling is then performed to decode the chromatographic traces (electropherograms) produced by each experiment and the resulting sequences are aligned and compared to obtain a unique consensus sequence representative of the original sample. In case of mismatch, manual editing need to be performed by an experienced biologist looking back at the original traces. In this work we propose computer-aided approaches to base calling from forward and reverse electropherograms aimed at minimizing the need for human intervention during consensus generation. Comparative experimental results are provided to evaluate the effectiveness of the proposed approaches.

1 Introduction

DNA sequencing is an error-prone process composed of two main steps: generation of an *electropherogram* (or *trace*) representative of a DNA sample, and interpretation of the electropherogram in terms of base sequence. The first step entails chemical processing of the DNA sample, electrophoresis and data acquisition [9]; the second step, known as base calling, entails digital signal processing and decoding usually performed by software running on a PC [4,5,6,10]. In order to improve the accuracy and reliability of DNA sequencing, multiple experiments may be independently performed on the same DNA sample. In most cases, forward and reverse experiments are performed by sequencing a DNA segment from the two ends. Bases that appear at the beginning of the forward electropherogram appear (complemented) at the end of the reverse one. Since most of the noise sources are position-dependent (e.g., there is a sizable degradation of the signal-to-noise ratio during each experiment) starting from opposite sides provides valuable information for error compensation. The traditional approach to base calling from opposite traces consists of: i) performing independent base calling on each electropherogram, ii) aligning the corresponding base sequences, and iii) obtaining a consensus sequence by means of comparison and manual editing. The main issue in this process is error propagation: after base calling, wrong bases take part in sequence alignment as if they were correct, although annotated by a confidence level. In case of mismatch, the consensus is manually generated either by comparing the confidence levels of the mismatching bases or by looking back at the original traces.

In this work we explore the feasibility of computer-aided consensus generation. We propose two different approaches. The first approach (called *consensus generation after base calling, CGaBC*) resembles the traditional consensus generation, except for

C. Priami et al. (Eds.): Trans. on Comput. Syst. Biol. III, LNBI 3737, pp. 1–13, 2005.

the fact that automated decisions are taken, in case of mismatch, on the basis of the quality scores assigned by the base caller to forward and reverse sequences. The second approach (called *base calling after trace merging, BCaTM*) performs base calling after error compensation: electropherograms obtained from forward and reverse sequencing experiments are merged in a single averaged electropherogram less sensitive to sequencing errors and noise. Base calling is then performed on the averaged trace directly providing a consensus sequence. The tool flows of the two approaches are shown in Figure 1. Two variants of the second approach are presented differing only for the way the original electropherograms are aligned for merging purposes.

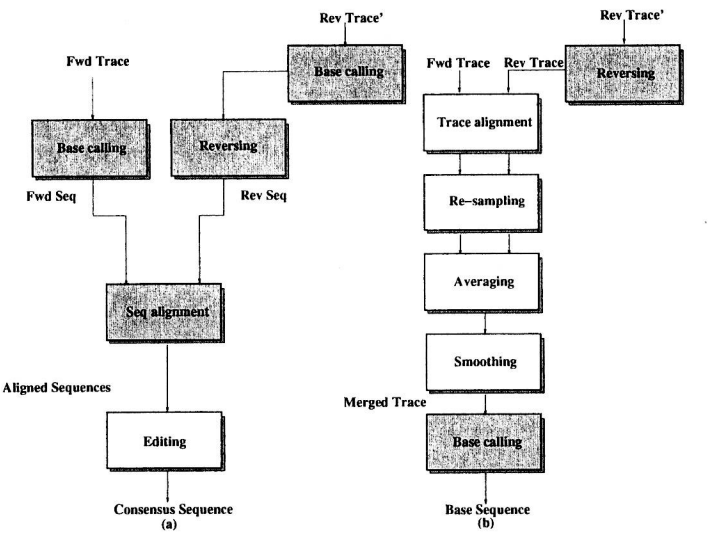


Fig. 1. Tool flow of computer aided base calling from forward and reverse electropherograms: (a) CGaBC. (b) BCaTM. (Rev Trace' denotes the original reverse trace to be reversed and complemented into Rev Trace)

The results presented in this paper show that reliable automated decisions can be taken in most cases of mismatch, significantly reducing the human effort required to generate a consensus sequence. The key issue, however, is how to distinguish between reliable and unreliable automated decisions. A quality score is assigned to this purpose to each base of the consensus sequence. If the quality is above a given threshold, automated decisions can be directly accepted, otherwise they need to be double checked by a human operator. The significance of the quality threshold is discussed in the result section.

1.1 Base Calling

Base calling is the process of determining the base sequence from the electropherogram provided by a DNA automated sequencer. In particular we refer to the DNA sequencing

process known as Sanger's method [9]. Four reactions of extension from initial primers of a given template generate an entire set of nested sub-fragments in which the last base of every fragment is marked with 4 different types of fluorescent markers (one for each type of base). Fragments are then sorted by length by means of capillary electrophoresis and detected by 4 optical sensors working at disjoint wavelengths in order to distinguish the emissions of the 4 markers. The result of a sequencing experiment is an electropherogram that is a 4-component time series made of the samples of the emissions measured by the 4 optical sensors. In principle, the DNA sequence can be obtained from the electropherogram by associating each dominant peak with the corresponding base type and by preserving the order of the peaks. However, electropherograms are affected by several non-idealities (random noise of the measuring equipment, cross-talk due to the spectral overlapping between fluorescent markers, sequence-dependent mobility, ...) that require a pre-processing step before decoding. Since the non-idealities depend on the sequencing strategy and on the sequencer, pre-processing (consisting of multi-component analysis, mobility shift compensation and noise filtering) is usually performed by software tools distributed with sequencing machines [1]. The original electropherogram is usually called *raw data*, while we call *filtered data* the result of pre-processing. In the following we will always refer to filtered data, representing the filtered electropherogram (hereafter simply called *electropherogram*, or *trace*, for the sake of simplicity) as a sequence of 4-tuples. The k-th 4-tuple (Ak, Ck, Gk, Tk) represents the emission of the 4 markers at the k-th sampling instant, uniquely associated with a position in the DNA sample. In this paper we address the problem of base calling implicitly referring to the actual decoding step, that takes in input the (filtered) electropherogram and provides a base sequence. Base calling is still a difficult and error-prone task, for which several algorithms have been proposed [4,6,10]. The result they provide can be affected by different types of errors and uncertainties: *mismatches* (wrong base types at given positions), *insertions* (bases artificially inserted by the base caller), *deletions* (missed bases), *unknowns* (unrecognized bases, denoted by N). The accuracy of a base caller can be measured both in terms of number of N in the sequence, and in terms of number of errors (mismatches, deletions and insertions) with respect to the actual consensus sequence. The accuracy obtained by different base callers starting from the same electropherograms provides a fair comparison between the algorithms. On the other hand, base callers usually provide estimates of the quality of the electropherograms they start from [3]. A quality value is associated with each called base, representing the correctness probability: the higher the quality the lower the error probability. Since our approach generates a synthetic electropherogram to be processed by a base caller, in the result section we also compare quality distributions to show the effectiveness of the proposed technique.

1.2 Sequence Alignment

Sequence comparison and alignment are critical tasks in many genomic and proteomic applications. The best alignment between two sequences F and R is the alignment that minimizes the effort required to transform F in R (or vice versa). In general, each edit operation (base substitution, base deletion, base insertion) is assigned with a cost, while each matching is assigned with a reward. Scores (costs and rewards) are empirically as-

signed depending on the application. The score of a given alignment between F and R is computed as the difference between the sum of the rewards associated with the pairwise matches involved in the alignment, and the sum of the edit operations required to map F onto R. The best alignment has the maximum similarity score, that is usually taken as similarity metric. The basic dynamic programming algorithm for computing the similarity between a sequence F of M characters and a sequence R of N characters was proposed by Needleman and Wunsch in 1970 [8], and will be hereafter denoted by NW-algorithm. It makes use of a score matrix D of $M+1$ rows and $N+1$ columns, numbered starting from 0. The value stored in position (i,j) is the similarity score between the first i characters of F and the first j characters of R, that can be incrementally obtained from $D(i-1,j)$, $D(i-1,j-1)$ and $D(i,j-1)$:

$$D(i,j) = max \begin{cases} D(i-1,j-1) + S_{sub}(F(i),R(j)) \\ D(i-1,j) + S_{del} \\ D(i,j-1) + S_{ins} \end{cases} \quad (1)$$

S_{ins} and S_{del} are the scores assigned with each insertion and deletion, while S_{sub} represents either the cost of a mismatch or the reward associated with a match, depending on the symbols associated with the row and column of the current element. In symbols, $S_{sub}(F(i),R(j)) = S_{mismatch}$ if $F(i) \neq R(j)$, $S_{sub}(F(i),R(j)) = S_{match}$ if $F(i) = R(j)$.

2 Consensus Generation after Base Calling (CGaBC)

When forward and reverse electropherograms are available, the traditional approach to determine the unknown DNA sequence consists of: i) independently performing base calling on the two traces in order to obtain forward and reverse sequences, ii) aligning the two sequences and iii) performing a minimum number of (manual) editing steps to obtain a consensus sequence. The flow is schematically shown in Figure 1.a, where the reverse trace is assumed to be reversed and complemented by the processing block labeled Reverse. Notice that complementation can be performed either at the trace level or at the sequence level (i.e., after base calling). In Fig. 1.a the reverse trace is reversed and complemented after base calling.

The results of the two experiments are combined only once they have been independently decoded, without taking advantage of the availability of two chromatographic traces to reduce decoding uncertainties. Once base-calling errors have been made on each sequence, wrong bases are hardly distinguishable from correct ones and they take part in alignment and consensus. On the other hand, most base callers assign with each base a quality (i.e., confidence) value (representing the correctness probability) computed on the basis of the readability of the trace it comes from.

In a single sequencing experiment, base qualities are traditionally used to point out unreliable calls to be manually checked. When generating a consensus from forward and reverse sequences, quality values are compared and combined. Comparison is used to decide, in case of mismatch, for the base with higher value. Combination is used to assign quality values to the bases of the consensus sequence.

The proper usage of base qualities has a great impact on the accuracy (measured in terms of errors) and completeness (measured in terms of undecidable bases) of the consensus sequence. However, there are no standard methodologies for comparing and combining quality values.

The CGaBC approach presented in this paper produces an *aggressive consensus* by taking always automated decisions based on base qualities: in case of a mismatch, the base with higher quality is always assigned to the consensus. Since alignment may give rise to gaps, quality values need also to be assigned to gaps. This is done by averaging the qualities of the preceding and following calls.

Qualities are assigned to the bases of the consensus sequence by adding or subtracting the qualities of the aligned bases in case of match or mismatch, respectively [7]. Quality composition, although artificial, is consistent with the probabilistic nature of quality values, defined as $q = -10log_{10}(p)$, where p is the estimated error probability for the base call [5].

In some cases, however, wrong bases may have quality values greater than correct ones, making it hard to take automated correct decisions. The overlapping of the quality distributions of wrong and correct bases is the main problem of this approach.

A quality threshold can be applied to the consensus sequence to point out bases with a low confidence level. Such bases need to be validated by an experienced operator, possibly looking back at the original traces.

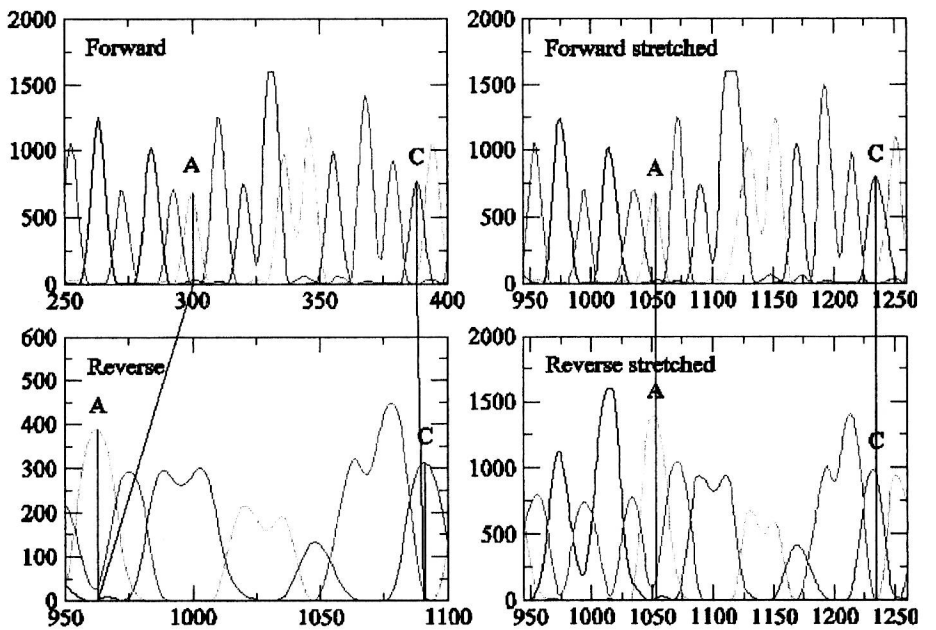


Fig. 2. Trace alignment issues (left) and sample re-positioning on a common x-axis (right)

3 Base Calling after Trace Merging (BCaTM)

The approach is illustrated in Figure 1b. We first obtain an average trace by combining the two experiments available for the given DNA, then we perform base calling on the averaged trace directly obtaining the consensus sequence. The rationale behind this approach is two-fold. First, electropherograms are much more informative than the corresponding base sequences, so that their comparison provides more opportunities for noise filtering and error correction. Second, each electropherogram is the result of a complex measurement experiment affected by random errors. Since the average of independent measurements has a lower standard error, the averaged electropherogram has improved quality with respect to the original ones.

Averaging independent electropherograms is not a trivial task, since they usually have different starting points, different number of samples, different base spacing and different distortions, as shown in the left-most graphs of Fig. 2. In order to compute the point-wise average of the traces, we need first to re-align the traces so that samples belonging to the same peak (i.e., representing the same base) are in the same position, as shown in the right-most graphs of Fig. 2. By doing this, we are then able to process *homologous* samples, that is to say samples arranged according to the fact that in the same position on the x-axis we expect to find values representing the same point of the DNA sample. A similar approach was used by Bonfield et al. [2] to address a different problem: comparing two electropherograms to find point mutations. However, the authors didn't discuss the issues involved in trace alignment and point-wise manipulation.

We propose two different procedures for performing trace alignment. The first is based on the maximization of the correlation between the four time series, using a dynamic programming algorithm derived form the NW-algorithm. The second makes use of a preliminary base calling step to identify base positions within the trace to be used to drive trace alignment. The overall procedures (respectively denoted as *sample-driven alignment* and *base-driven alignment*) are described in the next sections, assuming that forward and reverse traces are available and that the reverse trace has already been reversed and complemented. All procedures are outlined in Fig. 3.

After alignment, forward and reverse traces are re-sampled using a common sampling step and their sample-wise average is computed to obtain the averaged electropherogram. Local smoothing is then performed to remove small artificial peaks possibly introduced by the above steps. Base calling is finally performed on the averaged electropherogram directly providing the consensus sequence. The entire process is outlined in Fig. 1.b.

3.1 Sample-Driven (SD) Trace Alignment

Sample-driven trace alignment aims at maximizing the correlation between the 4-component time series that constitute forward and reverse electropherograms. Aligned electropherograms are nothing but the original electropherograms with labels associated with samples to denote pairwise alignment. Homologous samples share the same label. The score associated with an alignment is the sample-wise correlation computed according to the alignment.

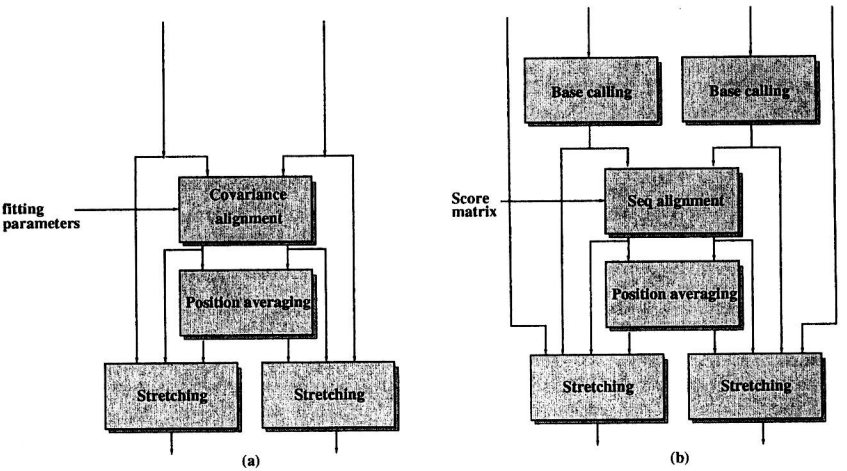


Fig. 3. a) Sample-driven alignment procedure. b) Base-driven alignment procedure.

The difference between pairwise correlation of electropherograms and pairwise correlation of standard time series is twofold. First, electropherograms are 4-component time series. The correlation between two electropherograms is the sum of the pairwise correlations between their homologous components. Second, due to the intrinsic nature of electrophoretic runs, electropherograms might need to be not only shifted, but also stretched with respect to each other in order to obtain a meaningful point-wise alignment. Stretching can be obtained by means of gap insertion at specific positions of one of the two electropherograms under alignment.

Despite the above-mentioned peculiarities, the correlation between electropherograms retains the additive property of the standard correlation. Hence, the alignment corresponding to the maximum correlation can be incrementally determined by means of dynamic programming techniques. In the next subsection we outline a modified version of the NW algorithm that handles electropherograms maximizing their correlation.

Dynamic Programming Alignment Algorithm. For the sake of simplicity, we sketch the NW modified algorithm considering single-component traces. We will then generalize to the 4-component case. As previously introduced in section 1.2, the NW algorithm incrementally computes the optimal path through the dynamic-programming matrix (DP matrix) according to a specific optimality criterion. At each step a new entry (say, i,j) of the matrix is computed as the maximum score achieved by means of one of the three possible moves leading to that position: a diagonal move that adds a replacement score (that is a reward for the alignment of the i^{th} element of sequence F with the j^{th} element of sequence R) to the value stored in entry $(i-1, j-1)$; a vertical move that adds a deletion cost to the value stored in entry $(i-1, j)$; and a horizontal move that adds an insertion cost to the value stored in entry $(i, j-1)$. As far as alignment is concerned, insertions and deletions are symmetric moves: deleting an element from sequence F has the same effect of adding an element in sequence R. Although both insertions and deletions are needed to stretch the two sequences in order to achieve the

best alignment between them, the two operations are nothing but gap insertions in one of the two sequences.

According to the above observations, we outline the modified NW algorithm referring to two basic operations: alignment between existing samples of the two electropherograms (corresponding to a diagonal move in DP matrix) and insertion of a gap in one of the two electropherograms (corresponding to vertical or horizontal moves in the DP matrix). The score to be assigned to the alignment between existing samples of the two electropherograms (diagonal move) is computed as the correlation between the samples.

$$S_{diag}(i,j) = \frac{(F(i) - F_{avg})(R(j) - R_{avg})}{\sigma_F \sigma_R}$$

where F_{avg} and R_{avg} are the average values of the elements of F and R, while σ_F and σ_R are their standard deviations.

In order to assign a score to a gap we refer to the role the gap will play in the final alignment. After alignment, the two aligned electropherograms need to be processed in order to fill all the gaps possibly inserted by the NW algorithm. Synthetic samples need to be created to this purpose and added at proper positions. Such synthetic samples are introduced by interpolating the existing samples on both sides of a gap. In this perspective, the score to be assigned to a gap insertion in one of the two electropherograms (vertical or horizontal moves) is computed as the correlation between the synthetic sample (generated by interpolation) to be added to bridge the gap and the original sample of the other trace aligned with the gap. The score assigned with an horizontal move leading to entry (i,j), corresponding to a gap insertion in the forward trace F, will be:

$$S_{hor}(i,j) = \frac{(\frac{F(i)+F(i+1)}{2} - F_{avg})(R(j) - R_{avg})}{\sigma_F \sigma_R}$$

while the score assigned with a vertical move will be:

$$S_{ver}(i,j) = \frac{(F(i) - F_{avg})(\frac{R(j)+R(j+1)}{2} - R_{avg})}{\sigma_F \sigma_R}$$

If we deal with 4-component time series rather than with single-component traces, we can extend the algorithm by maximizing the sum of the four correlations:

$$\begin{aligned}
S_{diag}(i,j) &= \sum_{h\in\{A,C,G,T\}} \frac{(F^{(h)}(i) - F^{(h)}_{avg})(R^{(h)}(j) - R^{(h)}_{avg})}{\sigma^{(h)}_F \sigma^{(h)}_R} \\
S_{hor}(i,j) &= \sum_{h\in\{A,C,G,T\}} \frac{(\frac{F^{(h)}(i)+F^{(h)}(i+1)}{2} - F^{(h)}_{avg})(R^{(h)}(j) - R^{(h)}_{avg})}{\sigma^{(h)}_F \sigma^{(h)}_R} \qquad (2) \\
S_{ver}(i,j) &= \sum_{h\in\{A,C,G,T\}} \frac{(F^{(h)}(i) - F^{(h)}_{avg})(\frac{R^{(h)}(j)+R^{(h)}(j+1)}{2} - R^{(h)}_{avg})}{\sigma^{(h)}_F \sigma^{(h)}_R}
\end{aligned}$$

where index h spans the four components A, C, G and T.