Preface

We are pleased to present the proceedings of the 11th Workshop on Algorithms in Bioinformatics (WABI 2011) which took place in Saarbrücken, Germany, September 5–7, 2011. The WABI 2011 workshop was part of the six ALGO 2011 conference meetings, which, in addition to WABI, included ESA, IPEC, WAOA, ALGOSENSORS, and ATMOS. WABI 2011 was hosted by the Max Planck Institute for Informatics, and sponsored by the European Association for Theoretical Computer Science (EATCS) and the International Society for Computational Biology (ISCB). See https://algo2011.mpi-inf.mpg.de/ for more details. The Workshop in Algorithms in Bioinformatics highlights research in algorithmic work for bioinformatics, computational biology, and systems biology. The emphasis is mainly on discrete algorithms and machine-learning methods that address important problems in molecular biology, that are founded on sound models, that are computationally efficient, and that have been implemented and tested in simulations and on real datasets. The goal is to present recent research results, including significant work-in-progress, and to identify and explore directions of future research.

Original research papers (including significant work-in-progress) or state-of-the-art surveys were solicited for WABI 2011 in all aspects of algorithms in bioinformatics, computational biology, and systems biology. In response to our call, we received 77 submissions for papers and 30 were accepted. In addition, WABI 2011 hosted a distinguished lecture by Vincent Moulton, of the University of East Anglia, UK. We would like to sincerely thank the authors of all submitted papers and the conference participants. We also thank the Program Committee and their sub-referees for their hard work in reviewing and selecting papers for the workshop.

We would especially like to thank Bernard Moret for all his advice and support in carrying out the role of being Co-chairs, as well as EasyChair for making the management of the submissions to WABI such an easy process. Thanks once again to all who participated in making WABI such a success in 2011. For us it has been an exciting and rewarding experience.

June 2011

Teresa Przytycka
Marie-France Sagot
Organization

Program Committee

Tatsuya Akutsu  
Jan Baumbach  
Tanya Berger-Wolf  
Paola Bonizzoni  
Dan Brown  
Michael Brudno  
Sebastian Böcker  
Robert Castelo  
Benny Chor  
Lenore Cowen  
Nadia El-Mabrouk  
Eleazar Eskin  
Liliana Florea  
Ana Teresa Freitas  
Anna Gambin  
Olivier Gascuel  
Raffaele Giancarlo  
Dan Gusfield  
Ivo Hofacker  
Barbara Holland  
Daniel Huson  
Igor Jurisica  
Carl Kingsford  
Gunnar Klau  
Mehmet Koyuturk  
Jens Lagergren  
Hans-Peter Lenhof  
Christine Leslie  
Stefano Lonardi  

Kyoto University, Japan  
Max Planck Institute for Informatics, Germany  
UIC, USA  
Università di Milano-Bicocca, Italy  
Cheriton School of Computer Science, University of Waterloo, Canada  
University of Toronto, Canada  
Friedrich Schiller University Jena, Germany  
Universitat Pompeu Fabra, Spain  
School of Computer Science, Tel Aviv University, Israel  
Tufts University, USA  
University of Montreal, Canada  
University of California, Los Angeles, USA  
University of Maryland, USA  
INESC-ID/IST, Technical University Lisbon, Portugal  
Institute of Informatics, Warsaw University, Poland  
LIRMM, CNRS - Université Montpellier 2, France  
Università di Palermo, Italy  
UC Davis, USA  
University of Vienna, Austria  
University of Tasmania, Australia  
University of Tübingen, Germany  
Ontario Cancer Institute, Canada  
University of Maryland, College Park, USA  
CWI, The Netherlands  
Case Western Reserve University, USA  
SBC and CSC, KTH, Sweden  
Center for Bioinformatics, Saarland University, Germany  
Memorial Sloan-Kettering Cancer Center, USA  
UC Riverside, USA
VIII Organization

Ion Mandoiu University of Connecticut, USA
Satoru Miyano Human Genome Center, Institute of Medical Science, University of Tokyo, Japan
Bernard Moret EPFL, Switzerland
Burkhard Morgenstern University of Göttingen, Germany
Vincent Moulton University of East Anglia, UK
Gene Myers HHMI Janelia Farm Research Campus, USA
Itsik Pe’Er Columbia University, USA
Nadia Pisanti Università di Pisa, Italy
Teresa Przytycka NCBI, NLM, NIH, USA
Knut Reinert FU Berlin, Germany
Juho Rousu University of Helsinki, Finland
Yvan Saeys Flanders Institute for Biotechnology (VIB), Ghent University, Belgium
Marie-France Sagot Université de Lyon, France
Cenk Sahinalp Simon Fraser University, Canada
David Sankoff University of Ottawa, Canada
Thomas Schiex INRA, France
Stefan Schuster LS Bioinformatik, Universität Jena, Germany
Russell Schwartz Carnegie Mellon University, USA
Charles Semple Canterbury University, New Zealand
Mona Singh Princeton University, USA
Saurabh Sinha University of Illinois, USA
Joerg Stelling ETH Zurich, Switzerland
Leen Stougie Centrum voor Wiskunde en Informatica (CWI) and Technische Universität Eindhoven (TU/e), The Netherlands
Jens Stoye Bielefeld University, Germany
Michael Stumpf Imperial College London, UK
Jerzy Tiuryn Warsaw University, Poland
Hélène Touzet LIFL - CNRS, France
Alfonso Valencia Spanish National Cancer Research Centre (CNIO), Spain
Lusheng Wang University of Hong Kong City
Chris Workman DTU, Denmark
Alex Zelikovsky GSU, USA
Louxin Zhang National University of Singapore
Michal Ziv-Ukelson Ben Gurion University of the Negev, Israel
Elena Zotenko Garvan Institute, Australia

Additional Reviewers

Agius, Phaedra
Al Seesi, Sahar
Andonov, Rumen
Canzar, Stefan
Cho, DongYeong
Dao, Phuong
Degnan, James
Dehof, Anna Katharina
Della Vedova, Gianluca
Dondi, Riccardo
Doyon, Jean-Philippe
Duggal, Geet
Duma, Denis
Dörr, Daniel
Ehrler, Carsten
El-Kebir, Mohammed
Erten, Sinan
Federico, Maria
Fonseca, Paulo
Francisco, Alexandre
Frid, Yelena
Gambin, Tomasz
Gaspin, Christine
Geraci, Filippo
Ghiurcuta, Cristina
Gorecki, Pawel
Harris, Elena
Hellmuth, Marc
Hoener Zu Siderdissen, Christian
Hormozdari, Farhad
Huang, Yang
Hufsky, Franziska
Ibragimov, Rashid
Jahn, Katharina
Kelk, Steven
Kennedy, Justin
Kim, Yoo-Ah
Lempel-Musa, Noa
Lin, Yu
Lindsay, James
Linz, Simone
Lorenz, Ronny
Lynce, Ines
Malin, Justin
Manzini, Giovanni
Marschall, Tobias
Menconi, Giulia
Misra, Navodit
Montangero, Manuela
Mozes, Shay
Mueller, Oliver
Nicolae, Marius
Pardi, Fabio
Parrish, Nathaniel
Patro, Rob
Pitkänen, Esa
Polishko, Anton
Roettger, Richard
Rueckert, Ulrich
Ruffalo, Matthew
Russo, Luís
Rybinski, Mikolaj
Salari, Raheleh
Scheubert, Kerstin
Schoenhuth, Alexander
Scornavacca, Celine
Snir, Sagi
Startek, Michal
Stckel, Daniel
Stevens, Kristian
Sun, Peng
Swanson, Lucas
Swenson, Kristo
Szczurek, Ewa
Tantipathananandh, Chayant
Taubert, Jan
Thapar, Vishal
Truss, Anke
Ullah, Ikram
Weile, Jochen
Willson, Stephen
Winnenburg, Rainer
Winter, Sascha
Wittkop, Tobias
Wittler, Roland
Wohlers, Inken
Wolter, Katinka
Wu, Taoyang
Wu, Yufeng
Wójtowicz, Damian
Yorukoglu, Deniz
Zaitlen, Noah
Zakov, Shay
Zheng, Yu
Automated Segmentation of DNA Sequences with Complex Evolutionary Histories ............................................ 1
   Broňa Brejová, Michal Burger, and Tomáš Vinař

Towards a Practical $O(n \log n)$ Phylogeny Algorithm ................. 14
   Daniel G. Brown and Jakub Truszkowski

A Mathematical Programming Approach to Marker-Assisted Gene Pyramiding .................................................... 26
   Stefan Canzar and Mohammed El-Kebir

Localized Genome Assembly from Reads to Scaffolds: Practical Traversal of the Paired String Graph ......................... 39
   Rayan Chikhi and Dominique Lavenier

Stochastic Errors vs. Modeling Errors in Distance Based Phylogenetic Reconstructions (Extended Abstract) ................... 49
   Daniel Doerr, Ilan Gronau, Shlomo Moran, and Irad Yavneh

Constructing Large Conservative Supertrees .......................... 61
   Jianrong Dong and David Fernández-Baca

PepCrawler: A Fast RRT–Like Algorithm for High–Resolution Refinement and Binding–Affinity Estimation of Peptide Inhibitors (Abstract) ...................................................... 73
   Elad Donsky and Haim J. Wolfson

Removing Noise from Gene Trees .................................... 76
   Andrea Doroftei and Nadia El-Mabrouk

Boosting Binding Sites Prediction Using Gene’s Positions .......... 92
   Mohamed Elati, Rim Fekih, Rémy Nicolle, Ivan Junier, Joan Hérison, and François Képès

Constructing Perfect Phylogenies and Proper Triangulations for Three-State Characters ........................................... 104
   Rob Gysel, Fumei Lam, and Dan Gusfield

On a Conjecture about Compatibility of Multi-states Characters ...... 116
   Michel Habib and Thu-Hien To

Learning Protein Functions from Bi-relational Graph of Proteins and Function Annotations ....................................... 128
   Jonathan Qiang Jiang
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph-Based Decomposition of Biochemical Reaction Networks into</td>
<td>139</td>
</tr>
<tr>
<td>Monotone Subsystems</td>
<td></td>
</tr>
<tr>
<td>Hans-Michael Kaltenbach, Simona Constantinescu, Justin Feigelman, and</td>
<td></td>
</tr>
<tr>
<td>Jörg Stelling</td>
<td></td>
</tr>
<tr>
<td>Seed-Set Construction by Equi-entropy Partitioning for Efficient and</td>
<td>151</td>
</tr>
<tr>
<td>Sensitive Short-Read Mapping</td>
<td></td>
</tr>
<tr>
<td>Kouichi Kimura, Asako Koike, and Kenta Nakai</td>
<td></td>
</tr>
<tr>
<td>A Practical Algorithm for Ancestral Rearrangement Reconstruction</td>
<td>163</td>
</tr>
<tr>
<td>Jakub Kováč, Broňa Brejová, and Tomáš Vinař</td>
<td></td>
</tr>
<tr>
<td>Bootstrapping Phylogenies Inferred from Rearrangement Data</td>
<td>175</td>
</tr>
<tr>
<td>Yu Lin, Vaibhav Rajan, and Bernard M.E. Moret</td>
<td></td>
</tr>
<tr>
<td>Speeding Up Bayesian HMM by the Four Russians Method</td>
<td>188</td>
</tr>
<tr>
<td>Md Pavel Mahmud and Alexander Schliep</td>
<td></td>
</tr>
<tr>
<td>Using Dominances for Solving the Protein Family Identification</td>
<td>201</td>
</tr>
<tr>
<td>Noel Malod-Dognin, Mathilde Le Boudic-Jamin, Pritish Kamath, and Rumen</td>
<td></td>
</tr>
<tr>
<td>Andonov</td>
<td></td>
</tr>
<tr>
<td>Maximum Likelihood Estimation of Incomplete Genomic Spectrum</td>
<td>213</td>
</tr>
<tr>
<td>from HTS Data</td>
<td></td>
</tr>
<tr>
<td>Serghei Mangul, Irina Astrovskaya, Marius Nicolae, Bassam Tork,</td>
<td></td>
</tr>
<tr>
<td>Ion Mandoiu, and Alex Zelikovsky</td>
<td></td>
</tr>
<tr>
<td>Algorithm for Identification of Piecewise Smooth Hybrid Systems:</td>
<td>225</td>
</tr>
<tr>
<td>Application to Eukaryotic Cell Cycle Regulation</td>
<td></td>
</tr>
<tr>
<td>Vincent Noel, Sergei Vakulenko, and Ovidiu Radulescu</td>
<td></td>
</tr>
<tr>
<td>Parsimonious Reconstruction of Network Evolution</td>
<td>237</td>
</tr>
<tr>
<td>Rob Patro, Emre Sefer, Justin Malin, Guillaume Marçais, Saket Navlakha,</td>
<td></td>
</tr>
<tr>
<td>and Carl Kingsford</td>
<td></td>
</tr>
<tr>
<td>A Combinatorial Framework for Designing (Pseudoknotted) RNA Algorithms</td>
<td>250</td>
</tr>
<tr>
<td>Yann Ponty and Cédric Saule</td>
<td></td>
</tr>
<tr>
<td>Indexing Finite Language Representation of Population Genotypes</td>
<td>270</td>
</tr>
<tr>
<td>Jouni Sirén, Niko Välimäki, and Veli Mäkinen</td>
<td></td>
</tr>
<tr>
<td>Efficiently Solvable Perfect Phylogeny Problems on Binary and k-State</td>
<td>282</td>
</tr>
<tr>
<td>Data with Missing Values</td>
<td></td>
</tr>
<tr>
<td>Kristian Stevens and Bonnie Kirkpatrick</td>
<td></td>
</tr>
<tr>
<td>Separating Metagenomic Short Reads into Genomes via Clustering</td>
<td>298</td>
</tr>
<tr>
<td>(Extended Abstract)</td>
<td></td>
</tr>
<tr>
<td>Olga Tanaseichuk, James Borneman, and Tao Jiang</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Finding Driver Pathways in Cancer: Models and Algorithms</td>
<td>314</td>
</tr>
<tr>
<td>Fabio Vandin, Eli Upfal, and Benjamin J. Raphael</td>
<td></td>
</tr>
<tr>
<td>Clustering with Overlap for Genetic Interaction Networks via Local Search Optimization</td>
<td>326</td>
</tr>
<tr>
<td>Joseph Whitney, Judice Koh, Michael Costanzo, Grant Brown, Charles Boone, and Michael Brudno</td>
<td></td>
</tr>
<tr>
<td>Dynamic Programming Algorithms for Efficiently Computing Cosegmentations between Biological Images</td>
<td>339</td>
</tr>
<tr>
<td>Hang Xiao, Melvin Zhang, Axel Mosig, and Hon Wai Leong</td>
<td></td>
</tr>
<tr>
<td>GASTS: Parsimony Scoring under Rearrangements</td>
<td>351</td>
</tr>
<tr>
<td>Andrew Wei Xu and Bernard M.E. Moret</td>
<td></td>
</tr>
<tr>
<td>OMG! Orthologs in Multiple Genomes - Competing Graph-Theoretical Formulations</td>
<td>364</td>
</tr>
<tr>
<td>Chunfang Zheng, Krister Swenson, Eric Lyons, and David Sankoff</td>
<td></td>
</tr>
<tr>
<td><strong>Author Index</strong></td>
<td>377</td>
</tr>
</tbody>
</table>
Automated Segmentation of DNA Sequences with Complex Evolutionary Histories

Broňa Brejová, Michal Burger, and Tomáš Vinař

Faculty of Mathematics, Physics, and Informatics, Comenius University, Mlynská Dolina, 842 48 Bratislava, Slovakia

Abstract. Most algorithms for reconstruction of evolutionary histories involving large-scale events such as duplications, deletions or rearrangements, work on sequences of predetermined markers, for example protein coding genes or other functional elements. However, markers defined in this way ignore information included in non-coding sequences, are prone to errors in annotation, and may even introduce artifacts due to partial gene copies or chimeric genes.

We propose the problem of sequence segmentation where the goal is to automatically select suitable markers based on sequence homology alone. We design an algorithm for this problem which can tolerate certain amount of inaccuracies in the input alignments and still produce segmentation of the sequence to markers with high coverage and accuracy. We test our algorithm on several artificial and real data sets representing complex clusters of segmental duplications. Our software is available at http://compbio.fmph.uniba.sk/atomizer/

1 Introduction

Genome rearrangements and segmental duplications, acting on long stretches of DNA, pose a significant challenge to comparative genomics. Rearrangements change the order of segments in the genome, resulting in new gene orders and new chromosomal organization. In a typical rearrangement study, we aim at computing the shortest possible number of operations transforming one genome to another or reconstructing a phylogenetic tree and ancestral gene orders (Moret et al., 2001; Bourque and Pevzner, 2002; Adam and Sankoff, 2008).

Segmental duplications increase the length of the sequences by copying genetic material to new locations, creating complex gene clusters, hotspots of evolutionary innovation (Zhang, 2003). Reconstruction of duplication events within such regions is a key to understanding their organization, function, and evolution (Benson and Dong, 1999; Elemento et al., 2002; Zhang et al., 2009; Vinar et al., 2010; Lajoie et al., 2010).

Most algorithms for these tasks do not work directly on the original sequences, but rather on predetermined markers or synteny blocks. These are intervals of the sequence such that all events in the true evolutionary history introduce breakpoints only at or between the boundaries of these intervals, but not inside them. The notion of such intervals was first introduced by Nadeau and Taylor...
(1984) (conserved segments) and in this work we call them atomic segments or atoms. Splitting the sequence into atoms allows algorithm developers to abstract from modeling local sequence alignments, and instead to concentrate on larger-scale processes that reorder and duplicate blocks of segments.

Here, we introduce a new method for segmenting sequence into atoms, primarily targeted at a fine-scale analysis of relatively recent evolutionary events (for example events that happened in the last 85 My of mammalian evolution, which approximately translates to higher than 80% sequence similarity for neutrally evolving sequences). We test our method in the context of reconstruction of duplication histories; however, it is also applicable to other scenarios, including rearrangement studies. In the rest of this section, we give an overview of the methods used previously for this task, and we demonstrate examples of various problems in real data sets that we address by our work.

**Related work.** Many algorithms for rearrangement or duplication analysis use protein coding genes as atoms (Fitch, 1977; Benson and Dong, 1999; Moret et al., 2001; Elemento et al., 2002; Bourque and Pevzner, 2002; Bertrand and Gascuel, 2005; Lajoie et al., 2007; Adam and Sankoff, 2008; Lajoie et al., 2010). In order to do so, we need to annotate genes in the sequence and establish homology or even orthology among them. The order and strand orientation of the genes is then used as an input for further analysis.

While this is perhaps the only solution applicable to distantly related sequences that need to be aligned at the protein level, it is not universal. First of all, necessary preprocessing, including gene finding and homology or orthology detection, is difficult and can introduce errors. Even in cases where reliable ortholog sets are available, this approach is not relevant for all evolutionary scenarios. Many incomplete pseudogenes present in human and other genomes clearly show that duplications and rearrangements do not respect gene boundaries. For example, the human PRAME gene cluster contains 38 copies of the PRAME locus (preferentially expressed antigens in melanoma), but more than a third of these copies are incomplete pseudogenes (Gibbs et al., 2007). Even more problematic are chimeric genes that contain a breakpoint inside an intron. The PRAME gene cluster contains a chimeric gene whose protein sequence consists of two parts with different phylogenetic ancestries. Its inclusion in a phylogenetic analysis may result in a completely incorrect phylogenetic tree.

Another example, where genes are not appropriate as atoms, is the UGT1A cluster. In the human genome, this cluster contains a single alternatively-spliced gene (UDP-glucuronosyltransferase) with at least 13 unique copies of the first exon that apparently arose by segmental duplication (Bellemare et al., 2010). To analyze this sequence, we would have to use exons as atoms instead of genes, but these exons are too short for a reliable gene tree reconstruction, which is a necessary step for many methods. Therefore it would be ideal to use also some of the surrounding non-coding sequences, which have been copied together with the exons, but that requires finding atomic segments unrelated to functional annotation.
Finally, a well-known KRAB zinc finger gene family in the human genome contains more than 400 genes in 25 gene clusters spread across several chromosomes (Schmidt and Durrett, 2004; Huntley et al., 2006). Each zinc finger gene is composed of the KRAB domain and between three and forty zinc fingers. In this family, we can see duplication of whole genes, as well as duplication of zinc finger domains within genes. For both UGT1A cluster and KRAB genes, the traditional selection of atoms (i.e. first exons or zinc finger domains) requires detailed functional annotation and a complex prior knowledge about the studied region. Even with such knowledge, we may create errors due to chimeric atoms or to loose valuable information due to insufficient sequence coverage.

Recently, a new approach to segmentation based on local sequence alignments has been introduced in the context of ancestral genome reconstruction (Ma et al., 2006). Briefly, using sequence alignment tools such as blastz (Schwartz et al., 2003) or UCSC chain/net pipeline (Kent et al., 2003), they identify significant local alignments of the sequence to itself with a desired level of homology and then use boundaries of these sequence alignments as atomic segment boundaries, and regions between the boundaries as atoms. Ma et al. (2006, 2008b) developed a heuristic pipeline that creates a map of such segments considering events of 50kb or more in length, later refining the boundaries to a finer precision. Their resolution is ideal for mammalian whole-genome analyses; however for smaller-scale events (such as analysis of gene clusters), finer resolution is needed.

In our previous work, we have used a simple greedy heuristic (SGH) for this type of analysis (Vinar et al., 2010). Analyzed sequences are first divided into non-overlapping segments of size 500, and for each segment we search for sequence homologies in the rest of the sequence. The segment with the highest number of homologs and its matching homologs are then designated as atoms. All segments overlapping new atoms are removed and the whole process is repeated.

There are several advantages to methods based on local alignments. The analysis is not dependent on possibly error-prone annotations. The atoms often span longer sections of the sequence which allows more accurate determination of phylogeny of individual atom instances. Finally, thanks to potentially finer resolution of atoms, we can use assumptions of infinite sites model (Ma et al., 2008a), such as low breakpoint reuse assumption that often helps to resolve symmetries in duplication event direction (Zhang et al., 2009).

Problem statement. Our goal is to investigate systematic approaches to segmentation of sequences into atoms. The input for our problem consists of several evolutionarily related sequences or one sequence with segmental duplications. We want to find non-overlapping segments called atoms (completely or partially covering the input sequences) and divide atoms into classes so that: (a) coverage of the sequences by atoms is high, to use as much sequence information as possible, (b) the number of atoms is low, to prevent unnecessary segmentation of long atoms, (c) two atoms of the same class share high sequence similarity across their entire length, (d) two atoms of different classes or two parts of the same atom do not appear to be homologous at a chosen sequence similarity threshold.
In practice, one has to find a trade-off these goals and to accommodate imperfect results of homology detection methods. In the rest of the paper we describe our new algorithm for the segmentation problem and evaluate its performance on both simulated and real sequences.

2 Algorithm

Alignments and breakpoint mapping. The input to our algorithm is a set of evolutionarily related input sequences. We use the LASTZ program (Harris, 2007) to align each sequence to itself and to every other sequence. One possible approach to sequence segmentation is to take the resulting set of local alignments, add a breakpoint at every boundary of a local pairwise alignment and to create an atom between every two adjacent breakpoints (Fig. 1). To classify atoms, we can simply create a graph with atoms as vertices and alignments between atoms as edges. Then we create one class for each connected component of this graph. This approach would work well on a perfect set of alignments, but on real data we can encounter various artifacts.

In Fig. 2 we see an example where one homology was not found by the local alignment program, leading to a missing breakpoint and wrong class assignment in the resulting segmentation. To avoid this problem, we will map boundaries of every alignment through other overlapping alignments to create new breakpoints, as suggested by a dotted line in Fig. 2. Mapping a breakpoint through an alignment is easy if the breakpoint is located at an aligned nucleotide. If it is located at a nucleotide aligned with a gap, we find the nearest aligned nucleotide to the left or to the right (whichever is closer) and map it according to this nucleotide.

**Fig. 1.** Simple sequence segmentation. The figure shows a dotplot of alignments of a sequence to itself. We can consider each alignment boundary as a breakpoint; segments between neighbouring breakpoints will form atoms in classes 1, 2, 3, 4, 5.

**Fig. 2.** Example of an input with a missing alignment. Region A is aligned to B and end of B to C, but the alignment between end of A and C is missing. The missing breakpoint in A can be mapped from B through the alignment between A and B, as suggested by the dotted line. Without mapping we get incorrect segmentation.
**Iterated homology mapping.** In our algorithm we perform breakpoint mapping iteratively, as a newly mapped breakpoint may need to be mapped further to other regions. We call this general process *iterative homology mapping (IHM).*

Another common problem is that boundaries of overlapping local alignments often do not coincide exactly, but are spread around the true breakpoint. This is caused by uncertainty of sequence alignment near alignment boundaries. The problem is even more exacerbated by mapping breakpoints through alignments, as shown in Fig. 3. Breakpoints $x$ and $y$ in regions $B$ and $C$ will be mapped to region $A$ through pairwise alignments. Although ideally they should map to the same place, due to imprecision in alignment boundary between $B$ and $D$, or due to imprecision in pairwise alignment between $A$ and $B$, the new breakpoint $x'$ is at some distance from the new breakpoint $y'$. The new breakpoint $x'$ is then mapped to $C$ and $y'$ is mapped to $B$, again creating pairs of nearby breakpoints. In some cases, the iterative process may create long arrays of breakpoints originating from repeatedly mapping what should have been the same breakpoints through a cycle of imprecise alignments. It is clearly not desirable to create a very short atom between every pair of such nearby boundaries.

![Diagram](image)

**Fig. 3.** Example of imprecise breakpoint mapping. Breakpoints $x$ and $y$ are mapped to two different positions in $A$, and each of them is then further mapped to a new position in $B$ or $C$. Each of atoms $A$, $B$, and $C$ is thus split into three atoms instead of two.

To avoid this problem, we select a window size $W$ and allow at most one breakpoint within each window. This is achieved by clustering breakpoints and replacing each cluster of nearby breakpoints with a new breakpoint roughly at their center. The new breakpoints are chosen so that no two breakpoints are closer than $W$ and the sum of squared distances between the input breakpoints and their new representatives is minimized. This can be done in $O(NW^2)$ time by a dynamic programming algorithm.

Outline of our approach is shown in pseudocode of Algorithm 1. We start with a set of breakpoints created from alignment endpoints. In each iteration of our algorithm, we first cluster breakpoints as described above. Each new breakpoint is then checked against all alignments, and if it is inside an alignment,
Algorithm 1. Iterative homology mapping

**Data:** set of alignments $A$, window length $W$

```plaintext
1 $B \leftarrow \text{endpoints}(A)$; // current breakpoints
2 $B_M \leftarrow \emptyset$; // all mapped breakpoints
3 repeat
4 \hspace{1em} $B \leftarrow \text{cluster}(B, W)$;
5 \hspace{1em} $B' \leftarrow \text{select_to_map}(B, B_M, W)$; // get unmapped breakpoints from $B$
6 \hspace{1em} $B'' \leftarrow \text{map}(B', A)$; // map $B'$ through $A$
7 \hspace{1em} $B \leftarrow B \cup B''$; $B_M \leftarrow B_M \cup B'$;
8 until $B'' = \emptyset$
9 return $B$;
```

It is mapped to the other sequence in the alignment. However, we do not map breakpoints that were already mapped in one of the previous iterations. Since breakpoint position may change slightly in each iteration due to the clustering process, we only map breakpoints that are at a distance of more than $W$ from every previously mapped breakpoint. For this purpose we keep a list $B_M$ of all previously mapped breakpoints.

During the whole algorithm, we map at most $N/W$ breakpoints, and therefore the algorithm terminates in at most $N/W$ iterations. In practical instances, the number of iterations is usually quite low, ranging from two to six in the experiments reported in this paper.

**Atom classification.** Once the breakpoints are fixed, we want to group atoms to classes so that the atoms within a class form a cluster densely connected by alignments, and there are relatively few alignments between atoms from different classes. This can be formulated as an optimization problem, where we seek to minimize the weighted sum of the number of false positive alignments (alignments connecting atoms from two different classes) and false negatives (pairs of atoms in the same class not connected by an alignment). This is a weighted variant of the NP-complete Cluster Editing Problem [Shamir et al., 2004].

We solve this problem exactly by CPLEX software from IBM using integer linear programming (ILP) formulation. In order to efficiently process inputs with large numbers of atoms, we employ several simple heuristics. First of all, atoms that are in different components of the alignment connectivity graph will never be in the same class in the optimal solution of the ILP. Therefore we process each connected component separately. We also do not run ILP on components that form a clique, because the optimal solution has then cost 0. If the size of a component exceeds 200, for efficiency reasons we use a different graph clustering strategy implemented in the MCL program [Van Dongen, 2008].

In the process of classification we also assign a strand to each atom so that they are consistent with alignments (each alignment suggests either that the two atoms should be on the same strand or on the opposite strands).

**Segmentation postprocessing.** In the resulting segmentation, we sometimes see a pair of atom classes $a$ and $b$ such that each atom of class $a$ is always followed by an atom of class $b$ and an atom of class $b$ is always preceded by an atom of
Correct: 
Predicted: 

Fig. 4. An example of reciprocal best matches between two segmentations. BRMs are shown as arrows. Each atom is labeled by its class. BRM sensitivity is 4/6, specificity is 4/7. Classes 12 and 16 are correctly predicted.

class $a$. We replace each such pair by a single atom, because there is no evidence of a breakpoint between $a$ and $b$ at the segmentation level. We perform such postprocessing on predicted segmentations as well as on the true segmentation in the simulated data. If the segmentation does not cover the whole sequence, the two atoms do not need to be adjacent in the sequence, as long as there are no further atoms between them.

Since shorter atoms, with length close to window length $W$, are often less accurate than longer ones, in some tests we also filter out all classes that have all atoms shorter than some threshold $T > W$.

3 Experiments

Here, we evaluate our methods in the context of duplication history reconstruction of gene clusters from several related species. We have tested our new IHM algorithm on both simulated and real data, comparing it to the simpler algorithm from Vinar et al. (2010), which we will call SGH (Simple Greedy Homology).

Measuring segmentation accuracy. If we know the true segmentation of a sequence, which is the case for artificial sequences generated from an evolutionary model, we can measure the quality of the predicted segmentation directly. To compare two segmentations, we first compute reciprocal best matches (BRM) between their atoms (see Fig. 4). In particular, an atom from one segmentation is a BRM of an atom in another segmentation if they cover overlapping regions of the sequence and no other atom overlaps either of the two by a larger amount.

We use four quality measures comparing the predicted segmentation to the true segmentation. Let $p$ the number of atoms in the predicted segmentation, $t$ the number of atoms in the true segmentation, and $b$ be the number of BRM pairs between them. We define BRM sensitivity as $b/t$ and BRM specificity as $b/p$. If, for example, an algorithm splits a true atom into two predicted atoms, one of them will not have a BRM pair, and therefore the BRM specificity will decrease. Similarly, a predicted atom spanning two real atoms will lead to a decrease in the BRM sensitivity.

The other two measures focus on the correctness of atom classification. Class $C_1$ is correctly predicted if each of its atoms has a BRM atom in the same true class $C_2$, and each atom in class $C_2$ has a BRM atom in $C_1$. Class sensitivity is then $c/t$ and specificity is $c/p$, where $c$ is the number of correctly predicted classes, $t$ the number of true classes, and $p$ the number of predicted classes.
Data sets. We have tested our algorithms on 30 simulated data sets divided into three categories with different parameter settings (see the overview in Table 1). To produce them, we have simulated sequence evolution, allowing substitutions according to the HKY model [Hasegawa et al., 1985], short insertions and deletions, as well as large-scale deletions and duplications. The simulation started with a 100kb sequence and proceeded along the human, chimpanzee, and rhesus macaque phylogeny. The parameters of the substitution model, branch lengths of the phylogenetic tree, and rate and length distributions of short insertions and deletions were estimated from UCSC syntenic alignments [Fujita et al., 2011] of human, chimpanzee, and macaque on the human chromosome 22. Parameters of large-scale events (duplications and deletions) were taken from Vinar et al. (2010). The Slow and Fast data sets differ in the rate of large-scale events per site, with Fast data sets using 1.5 times the slower rate. The No indel data sets were taken from Vinar et al. (2010) (first 10 out of 20 sets labeled as 300 in that paper). These sequences were generated by a simpler simulator that did not allow short insertions and deletions and also assumed that the rate of large-scale duplication and deletion is constant per sequence and does not depend on the sequence length.

Segmentation accuracy on simulated data. We have segmented all simulated data sets with our new method IHM, setting $W = 250$ and discarding atoms shorter than 500. The SGH program from Vinar et al. (2010) also produces atoms of length at least 500. On the first two categories, IHM noticeably outperforms SGH, particularly in sensitivity at both the BRM and class levels (Table 2). Since both programs work at the resolution of 500bp, they cannot predict very short atoms, which is why their sensitivity is lower than their specificity. For comparison, we also show the accuracy of the true segmentation with the atoms shorter than 500bp filtered out. Our program is very close to this ideal sensitivity. Even without filtering, IHM maintains high specificity combined with much higher sensitivity than the filtered IHM segmentation.

The last category of data does not contain small-scale indels, which makes the segmentation problem much easier. Due to the higher number of short atoms, the sensitivity is relatively small at resolution of 500bp, but almost identical for both programs as well as for the true segmentation filtered at 500. Both programs have achieved perfect specificity on these data sets, that is, every predicted atom is covered by BRM and all predicted classes agree with the true segmentation. The marked difference is in the boundary placement accuracy. About 86% of

Table 1. Overview of simulated data sets. Each category contains ten data sets. The table lists mean values of various statistics over these data sets, or in the case of sequence length, over all sequences from all data sets in the category combined.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Sequence length (kb)</th>
<th>Segmentation No. events</th>
<th>No. atoms</th>
<th>No. classes</th>
<th>dupl.</th>
<th>del.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow</td>
<td>253</td>
<td>153</td>
<td>55</td>
<td>28</td>
<td>2.6</td>
<td></td>
</tr>
<tr>
<td>Fast</td>
<td>444</td>
<td>385</td>
<td>113</td>
<td>56</td>
<td>3.8</td>
<td></td>
</tr>
<tr>
<td>No indels</td>
<td>210</td>
<td>611</td>
<td>78</td>
<td>25</td>
<td>1.2</td>
<td></td>
</tr>
</tbody>
</table>
Table 2. Accuracy of segmentation on simulated sets. TRUE500 is the true segmentation with atoms shorter than 500 filtered out. SGH500 is the segmentation created by method from Vinar et al. (2010). IHM250 is our new algorithm with $W = 250$ and IHM250:500 is the same method, only with atoms shorter than 500 filtered out. Sensitivity and specificity at the BRM and class level are computed as explained in the text.

<table>
<thead>
<tr>
<th>Set</th>
<th>Program</th>
<th>BRM</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>sn</td>
<td>sp</td>
</tr>
<tr>
<td>Slow</td>
<td>TRUE500</td>
<td>86%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>SGH500</td>
<td>63%</td>
<td>97%</td>
</tr>
<tr>
<td></td>
<td>IHM250.500</td>
<td>86%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>IHM250</td>
<td>95%</td>
<td>100%</td>
</tr>
<tr>
<td>Fast</td>
<td>TRUE500</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>SGH500</td>
<td>65%</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td>IHM250.500</td>
<td>79%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>IHM250</td>
<td>92%</td>
<td>100%</td>
</tr>
<tr>
<td>No indels</td>
<td>TRUE500</td>
<td>55%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>IHM250.500</td>
<td>56%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>SGH500</td>
<td>56%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>IHM250</td>
<td>86%</td>
<td>100%</td>
</tr>
</tbody>
</table>

the BRM atoms produced by IHM have both their boundaries within 50nt of the correct boundary, but this fraction is only 19% for the SGH method. This is because IHM boundaries ultimately originate in alignment endpoints, whereas SGH starts with arbitrarily placed sequence windows.

*Influence of segmentation on evolutionary history reconstruction.* In Vinar et al. (2010), we have used the true segmentation of simulated sequences as a starting point for evolutionary history reconstruction under a probabilistic model encompassing substitutions and large-scale duplications and deletions. In this work, we compare the accuracy of the history reconstruction when run on different segmentations using a subset of the simulated data from Vinar et al. (2010) (Table 3). We observe that in this case, using the predicted segmentation instead of the true segmentation filtered at 500bp does not have a large impact on the accuracy of the history reconstruction, yielding in most cases the same or very similar number of events.

*Segmentation of primate gene clusters.* Finally, we have applied our algorithm to the study of three complex primate gene clusters (PRAME, AMY and UGT1A), considered in Vinar et al. (2010). Before running the segmentation algorithms, we have masked the sequences with RepeatMasker, and then excised all masked or unknown bases, obtaining a shorter sequence without repeats.

On all three sets, SGH has produced more atoms (Table 4). A large majority of IHM atoms (81-91% in different sets) have a BRM atom in the SGH segmentation, but only 37%-76% of classes agree completely with the SGH. We have used these new segmentations to infer evolutionary history under the model of large-scale duplications and deletions.