Available online at www.sciencedirect.com
SCIENCE DIRECT
Information and Software Technology 47 (2005) 965–978
INFORMATION AND SOFTWARE TECHNOLOGY
www.elsevier.com/locate/infsof

Engineering a software tool for gene structure prediction in higher organisms

Gordon Gremme [a], Volker Brendel [b,c], Michael E. Sparks [c], Stefan Kurtz [a,*]

[a] Zentrum für Bioinformatik, Universität Hamburg, Bundesstrasse 43, 20146 Hamburg, Germany
[b] Department of Statistics, Iowa State University, Ames, IA 50011-3260, USA
[c] Department of Genetics, Development and Cell Biology, Iowa State University, Ames, IA 50011-3260, USA

Available online 8 November 2005

Abstract

The research area now commonly called 'bioinformatics' has brought together biologists, computer scientists, statisticians, and scientists of many other fields of expertise to work on computational solutions to biological problems. A large number of algorithms and software packages are freely available for many specific tasks, such as sequence alignment, molecular phylogeny reconstruction, or protein structure determination. Rapidly changing needs and demands on data handling capacity challenge the application providers to consistently keep pace. In practice, this has led to many incremental advances and re-writing of code that present the user community with confusing options and a large overhead from non-standardized implementations that need to be integrated into existing work flows. This situation gives much scope for contributions by software engineers. In this article, we describe an example of engineering a software tool for a specific bioinformatics task known as spliced alignment. The problem was motivated by disabling limitations in an original, ad hoc, and yet widely popular implementation by one of the authors. The present collaboration has led to a robust, highly versatile, and extensible tool (named GenomeThreader) that not only overcomes the limitations of the earlier implementation but greatly improves space and time requirements.

Keywords: Computational biology; Genome annotation; Similarity-based gene structure prediction; Intron cutout technique; Incremental updates

1. Introduction

Modern biology research is characterized by the ability to study questions from a genome-wide perspective. Whereas only a decade ago a research project would typically focus on a single gene or pathway, it is now possible to view and evaluate the same genes and pathways in the context of all the genes of an organism, mapped onto the chromosomes that constitute the species' entire genetic blueprint. Of course, these possibilities require prior correct identification and annotation of all the genes, a challenging problem that has not been entirely solved [7,8]. Whereas obtaining the genetic blueprint, or, more technically, genomic DNA sequencing and assembly, is a mostly hands on, experimental process, gene annotation is largely computational, involving both statistically based prediction methods and integration of various sources of experimental and knowledge-based evidence.

This paper illustrates the development of a versatile tool for gene structure prediction, named *GenomeThreader*. We describe the algorithms utilized by *GenomeThreader*. The main algorithmic contribution of this paper is the *intron cutout technique*, which allows prediction of gene structures stretching over large regions of a genome or chromosome. Such gene structures are often present in vertebrate genomes. The intron cutout technique consists of an efficient filtering step and a dynamic programming step, and we describe how to combine them.

Unlike most papers on similar topics written for the bioinformatics community, we do not stop with the algorithms, but continue with the description of implementation aspects. We consider these aspects very important, because only well engineered software tools can cope with the ever-changing requirements and fast growing data sizes in molecular biology.

We tried to keep the description of these implementation aspects generic to allow applications to problems other than gene structure prediction. Some details and ideas presented in the implementation sections may be straightforward or even be folk knowledge for an experienced computer scientist with focus on efficient implementation of algorithms. Nevertheless, we think that it is worthwhile to describe them here for the following reasons: First, it is interesting for a general computer scientist to see how the application of software engineering principles leads to robust and versatile software, solving an important problem in bioinformatics. Second, in the fast growing and interdisciplinary field of bioinformatics, software

* Corresponding author.
E-mail address: kurtz@zbh.uni-hamburg.de (S. Kurtz).

0950-5849/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2005.09.005

is often developed by researchers without formal education in computer science. These researchers are mostly not aware of certain implementation techniques and software engineering principles. This paper gives a source for otherwise undocumented techniques and software engineering principles applied to a particular problem in molecular biology.

The paper is organized as follows: Section 2 gives a brief introduction to the basic biological concepts needed to understand the paper. Section 3 introduces the computational problem addressed by the *GenomeThreader* software, namely the spliced alignment problem. Section 4 describes how to compute optimal spliced alignments. Section 5 introduces the intron cutout technique, which allows prediction of gene structures stretching over large regions of a genome of a higher organism. Section 6 explains how to compute a consensus spliced alignment from a set of spliced alignments. Section 7 is devoted to implementation and software engineering aspects. We describe the data structures implemented in *GenomeThreader*, sketch interfaces and test strategies and shortly describe the software development tools we employed. Some evaluation and performance results are given in Section 8. Section 9 closes with a discussion and an outline of future work.

2. Biological background

It suffices to review a few basic concepts of molecular biology for the reader not familiar with the subject. For a more thorough introduction, the reader is referred to textbooks of molecular biology [2,15].

Chemically, DNA is a polymer composed of four different types of nucleotides, denoted by A, C, G, and T. In the computational context of this work, we treat each DNA molecule as a string over the alphabet {A, C, G, T}. These strings can be as short as 100 symbols and as long as several million. The long strings represent the chromosomes of a species, and the entire set of all strings (chromosomes) comprise the genome of that species. Of note is that most DNA exists as an antiparallel helix of two complementary DNA molecules. Here, complementarity is defined by the consistent pairing of A's with T's and C's with G's on the opposing strands, and antiparallel refers to chemical directionality of the molecule. Thus, for example, ACCGTT pairs with AACGGT.

Genes are certain substrings of the chromosome strings. Here, we only consider protein-coding genes—parts of the genome that encode information for proteins, which are another type of polymer consisting of 20 different amino acids. The familiar genetic code describes the translation from the nucleotide alphabet into the amino acid alphabet. The underlying cellular processes are quite complicated, involving first a process of transcription, which generates a copy of a genic portion of genomic DNA as an RNA molecule (pre-mRNA, yet another polymer, but for our purposes we may consider it an exact copy of specified parts of the genomic DNA string). See Fig. 1 for a schematic explanation of the process. A curious feature of most genes in animals and plants is that the RNA molecule undergoes a process called splicing by which certain stretches (called introns) are cut out of the original molecule and only the remaining parts (exons), in their original linear order, provide the basis for translation into protein, the mRNA. The processed RNA can be sampled experimentally, either as full-length molecules (termed cDNA; the term results from the fact that, for experimental reasons, the RNA is reverse transcribed back into the complementary DNA string) or as fragments (termed ESTs—Expressed Sequence Tags).

The computational approach to gene finding discussed in this paper consists of aligning cDNAs and ESTs to genomic DNA (gDNA, for short) and thereby identifying the exons and

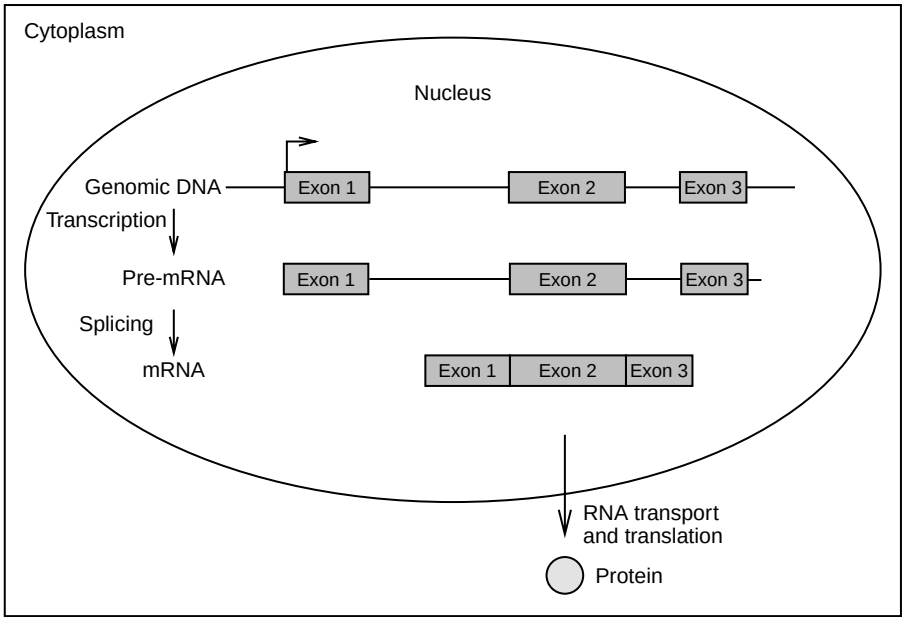

Fig. 1. Gene expression (simplified). More details are given in Refs. [2,15].