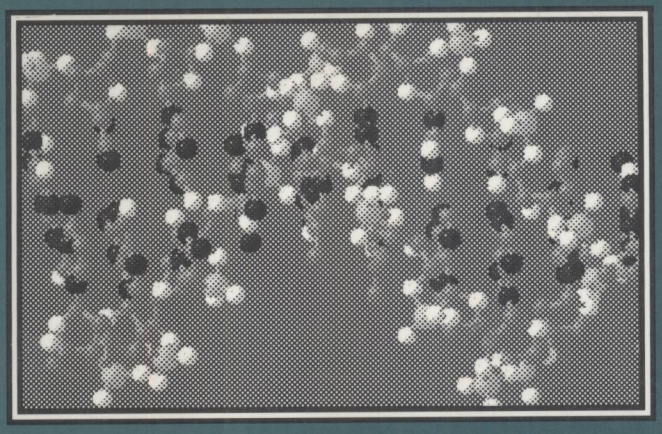

Proceedings

First International Conference on Intelligent Systems for Molecular Biology

Edited by

Lawrence Hunter
David Searls
& Jude Shavlik

July 6 – 9, 1993

National Library of Medicine,
Bethesda, MD

9860025

ISMB-93

Proceedings
First International Conference on Intelligent Systems for Molecular Biology

Edited by

Lawrence Hunter
David Searls
& Jude Shavlik

AAAI Press

Menlo Park • California

Copyright © 1993,
American Association for Artificial Intelligence

AAAI Press
445 Burgess Drive
Menlo Park, California 94025

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

ISBN 0-929280-47-4

Manufactured in the United States of America

QA7-53
I61
1993

ISMB-93

Proceedings

First International Conference on Intelligent Systems for Molecular Biology

Sponsoring Organizations

The National Institutes of Health,
National Library of Medicine

The Department of Energy,
Office of Health and Environmental Research

The American Association for Artificial Intelligence

The Biomatrix Society

The National Institutes of Health,
Division of Computer Research and Technology

此为试读,需要完整PDF请访问: www.ertongbook.com

Organizing & Program Committee

ORGANIZING COMMITTEE

Lawrence Hunter	*National Library of Medicine*
David Searls	*University of Pennsylvania*
Jude Shavlik	*University of Wisconsin*

PROGRAM COMMITTEE

Douglas Brutlag	*Stanford University*
Bruce Buchanan	*University of Pittsburgh*
Christian Burks	*Los Alamos National Laboratory*
Fred Cohen	*University of California, San Francisco*
Chris Fields	*Institute for Genomic Research*
Michael Gribskov	*University of California, San Diego*
Peter Karp	*SRI International*
Toni Kazic	*Washington University*
Alan Lapedes	*Los Alamos National Laboratory*
Richard Lathrop	*MIT and Arris Corporation*
Charles Lawrence	*Baylor University*
Michael Mavrovouniotis	*University of Maryland*
George Michaels	*National Institutes of Health*
Harold Morowitz	*George Mason University*
Katsumi Nitta	*Institute for New Generation Computer Technology*
Michiel Noordewier	*Rutgers University*
Ross Overbeek	*Argonne National Laboratory*
Christopher Rawlings	*Imperial Cancer Research Fund*
Derek Sleeman	*University of Aberdeen*
David States	*Washington University*
Gary Stormo	*University of Colorado*
Edward Uberbacher	*Oak Ridge National Laboratory*
David Waltz	*Thinking Machines Corporation*

Preface

The name of the gathering archived in these proceedings, the "First International Conference on Intelligent Systems for Molecular Biology," was carefully worded and bears some exegesis. To begin with, there is an obvious element of optimism in the use of the word "First." The organizers were confident that this would indeed be the inauguration of a continuing series of such meetings, based upon the growing level of participation in a number of predecessor colloquia of various types (including AAAI Symposia and Workshops). This optimism was fully justified by the response. Nearly 70 papers were received from around the world, as well as hundreds of inquiries. In the judgment of the editors, the submissions were of high quality for a new conference in a field not yet well established. Funding agencies were also enthusiastic, in part because of groundwork laid in a preliminary meeting to promote the development of infrastructure in this new sub-field of computational biology (jointly sponsored by the National Science Foundation, and the National Library of Medicine and attended by many on the program committee). The success of this particular aspect of that effort is evidenced by the planning already underway for the second conference in the series.

The word "international" in the title reflects the observation that outstanding work in this field takes place in many countries around the world. Not only was the program committee drawn from Europe, North America, and Asia, but a gratifying fraction of the submissions were as well. It should also be noted that the conference is cross-cultural in a scientific sense as well. The organizers can attest that the rewards of such interdisciplinary work are balanced by difficulties that sometimes amount to outright culture clashes, not least of which are the differing attitudes toward conferences and conference proceedings. It is hoped that, as the conference series is established, these proceedings will be an attractive and respected venue for publication of original biological results as well as pragmatically-inclined applications of computational research. This inaugural volume would seem to bode well.

The words "Intelligent Systems" are the most problematic in the title. It was neither contrariness nor fear of an AI Winter that inspired this terminology; the organizers are all unabashed artificial intelligencers, and feel as well that this field represents a natural constituency for the technology push of AI to balance the applications pull of biology. Rather, the words "intelligent systems" were intended in part to promote inclusiveness, for example towards appropriate work in robotics, statistics, and databases — computational fields associated with AI, but not subsumed by it. In addition, the more general terminology was meant to let more emphasis fall on biological discovery. The final connective of the title, in fact, wavered for some time between "and" and "for," before the latter was chosen on the strength of its connotation of service. The choice of molecular biology as the domain (as opposed to biology generally) seemed a reasonable restriction, given the predominance of this arena in computational applications, and the need to provide some focus within the tremendous range of biological application areas.

The organizers accept any blame that may attach to these decisions, but they must confer praise on the program committee (listed elsewhere), which made timely and perspicacious comments on the papers submitted. Several other individuals deserve thanks for significant contributions to the organization of the conference: Laura Cuccia for secretarial support at the University of Wisconsin; Saundra Greenberg for secretarial support at the National Library of Medicine; Mark Craven and David Opitz for organizing the student volunteers; and Mike Hamilton at AAAI Press for ably publishing the proceedings.

To conclude, we would like to explicitly acknowledge and thank the funding agencies that made this conference possible: the National Library of Medicine, for grant R13-LM05518, as well as the use of its meeting facilities; the Department of Energy, Office of Health and Environmental Research foi grant DE-FG02-93ER61562; the American Association for Artificial Intelligence; The Biomatrix Society; and the National Institutes of Health, Division of Computer Research and Technology. We would also like to thank the sponsors of the infrastructure planning meeting that led to this conference: the National Library of Medicine for grant R13-LM20003 and the National Science Foundation for grant IRI-9123156.

–Lawrence Hunter, David Searls, & Jude Shavlik

Contents

此为试读,需要完整PDF请访问: www.ertongbook.com

ISMB-93

Proceedings

First International Conference on Intelligent Systems for Molecular Biology

Knowledge Discovery in GenBank

Jeffrey S. Aaronson, Juergen Haas & G. Christian Overton
Department of Genetics
University of Pennsylvania School of Medicine
Room 475, Clinical Research Building
422 Curie Boulevard
Philadelphia, PA 19104-6145
Internet: coverton@cbil.humgen.upenn.edu

Abstract

We describe various methods designed to discover knowledge in the GenBank nucleic acid sequence database. Using a grammatical model of gene structure, we create a parse tree of a gene using features listed in the FEATURE TABLE. The parse tree infers features that are not explicitly listed, but which follow from the listed features. This method discovers 30% more introns and 40% more exons when applied to a globin gene subset of GenBank. Parse tree construction also entails resolving ambiguity and inconsistency within a FEATURE TABLE. We transform the parse tree into an *augmented* FEATURE TABLE that represents inferred gene structure explicitly and unambiguously, thereby greatly improving the utility of the FEATURE TABLE to researchers. We then describe various analogical reasoning techniques designed to exploit the homologous nature of genes. We build a classification hierarchy that reflects the evolutionary relationship between genes. Descriptive grammars of gene classes are then induced from the instance grammars of genes. Case based reasoning techniques use these abstract gene class descriptions to predict the presence and location of regulatory features not listed in the FEATURE TABLE. A cross-validation test shows a success rate of 87% on a globin gene subset of GenBank.

1 Introduction

GenBank, the primary worldwide repository for nucleic acid sequence data, contains information on virtually all nucleic acid sequence that has been determined [Burks et al., 1991]. Each GenBank entry contains a FEATURE TABLE, a list of biologically significant features that, taken together, constitute GenBank's description of the structure of the sequence. Unfortunately, many GenBank entries suffer from incomplete, noisy (ambiguous or contradictory), and erroneous listings in the FEATURE TABLE. To a degree, these types of errors are inevitable in any large and complex database. The problem is exacerbated by the necessity of incorporating direct submissions from investigators into the database, without which GenBank would fall hopelessly behind in its effort to keep pace with the growing rate of sequence determination. However, investigators are often unfamiliar with the GenBank data description language, and as a result, fail to clearly represent their data. Typical obfuscations include missing features (e.g. introns not listed), mislabeled features (e.g. mRNA instead of prim_transcript or exon), incompatible boundary specifications among multiple features (e.g. between exon and CDS), and relegation of significant information (e.g. gene names in multigene entries) to free-text in the comment fields. Fortunately, much of this lost information can be recovered through analysis of the explicit information in the FEATURE TABLE to infer the implicit information. If we are to realize the full potential of GenBank, it is imperative that we develop tools that can discover this implicit data within GenBank.

We have developed several software tools in our laboratory designed to support this pursuit, chief among these is QGB [Overton et al., 1993], a system for performing complex queries on the information stored in flat-file and relational database versions of GenBank. Using a logic grammar as a model of gene structure, QGB corrects and disambiguates the listed features, discovers latent knowledge implicit in the FEATURE TABLE, and produces an idealized, augmented FEATURE TABLE as output. Queries in QGB, formulated in an SQL-like syntax, can be directed against the hierarchical sequence structures deduced by the logic grammar parser as well as other information in GenBank. A QGB query representing "return the locus ID, the definition line, and 10 bp 5' and 20 bp 3' to the 5'splice junction for all splice sites in all non-mammalian genes with complete coding sequences" would be constituted as

```
SELECT locus.id, definition,
  5'splice_site = JUNC(-10,pt:exon,pt:intron,20)
FROM '/databases/gbrel74/*.seq',
TO    myresults,
WHERE organism =\= mammalia AND
      definition AMONG
  ("complete cds" OR "complete coding sequence").
```

One of our major long-term goals is to apply computational approaches to the analysis and understanding of eukaryotic gene regulation. As part of this effort, we are taking various approaches towards discovering transcription elements and other regulatory signals in uncharacterized and partially characterized DNA sequences. Prediction of regulatory signals is of enormous practical value to researchers who can use this information to focus their costly and time-consuming experimental efforts on restricted regions of the DNA. We illustrate how our system can be used to automate the task of pattern recognition in the case where there are too few well-characterized examples of the regulatory sequences to apply statistical based machine learning methods towards inducing a pattern descriptor (see [Dietterich, 1990] for an overview of the requirements for statistical machine learning methods). As previously described [Overton & Pastor, 1991, Pastor et al., 1991], we have turned to a variant of **Case Based Reasoning** (CBR) [Kolodner, 1985, Kolodner et al., 1985], a form of reasoning by analogy, in this situation. One advantage of CBR is that it can succeed with only a few well-characterized examples if the uncharacterized test cases are sufficiently similar to some members of the example set. To do this, a CBR system makes use of domain knowledge, i.e., the similarity of the test case to some member of the case database, to replace the need for an accurate general gene structure model with an accurate local model. Methods of reasoning by analogy work well in this domain because similar biological systems are often **homologous**, that is derived from a common evolutionary ancestor, rather than being merely analogous. Furthermore, the CBR methodology matches the line of reasoning often used by biologists in practice: find a well-understood system similar to the new system, hypothesize the existence of features in the new system based on the features of the known system, then design and perform experiments to test for those features in the new system.

In CBR, well-characterized "cases" are organized and indexed in a case database. On the basis of the index, database cases are found which are similar to a test case and then these similar cases are used as templates to reason about the properties of the test case. The indexing scheme in our system is based on a static classification hierarchy constructed for attributes representing protein similarity and species similarity. The hierarchy is equivalent to a case database and the process of classifying a test case in the hierarchy amounts to the step of finding the most similar cases.

The paper is organized as follows: We provide background and motivate QGB's grammatical representation of gene structure, and describe its application to GenBank FEATURE TABLEs in Section 2. Section 3 details the construction of a classification hierarchy of genes that reflects the homologous relationship between similar biological systems. In Section 4, we discuss how gene class descriptions are recursively induced in the hierarchy from the instance grammars of genes. Section 5 explains how one CBR technique utilizes descriptive grammars and another utilizes sequence similarity in order to predict unknown regulatory regions in genes, and Section 6 suggests an application of CBR for correcting existing feature descriptions, rather than predicting new gene features. Finally, results of applying the system to the globin gene family subset of GenBank are given in Section 7, and a discussion can be found in Section 8.

Gene Primer

Our current work has focused on the globin gene family, whose proteins' function to transport oxygen, and include the myoglobins, hemoglobins and leghemoglobins. Figure 1 shows an abstract view of the structure of a canonical β-hemoglobin gene, but the essential features of this gene are typical of the genes of higher organisms. Substrings of a gene contain two types of information: information specifying the sequence of the gene's protein product, which is contained within the exon subsequences of the primary transcript, and information needed to **regulate** the process of gene expression.

Several hundred classes of elements are known that are part of the apparatus that controls gene expression, and more are discovered each year. These transcription elements typically range in length from 4 to 20 nucleotides, a size consistent with their presumed role as sequence specific recognition sites for binding of regulatory proteins. Upstream (located on the 5'flank of the primary transcript) **promoter** signal regions and downstream (located on the 3'flank of the primary transcript) **terminator** signal regions respectively act to define the start and stop points of the primary transcript. The class of promoters includes subsequences such as the **TATA, CAAT** and **CACA** boxes, which are found across a wide range of genes and species, as well as rarer subsequences that restrict gene expression to a specific tissue in an organism. Proper expression of a gene is critically dependent on the organization of the promoter sequences.

2 Gene Structure as a Grammar

QGB restructures the FEATURE TABLE of each GenBank entry into a collection of relational tuples, which are then processed by the **Sequence Structure Parser** (SSP) component of QGB. The SSP attempts to construct a parse tree expressing the structure of the gene described in the GenBank entry, or several parse trees in case the GenBank entry describes a gene cluster (see [Searls, 1993] for a full discussion of grammatical representation of gene structure).

In order to accommodate the peculiarities of the "language" of this domain we developed a generalization of the DCG formalism [Pereira & Warren, 1980] as an implementation technique of SSPs. Standard

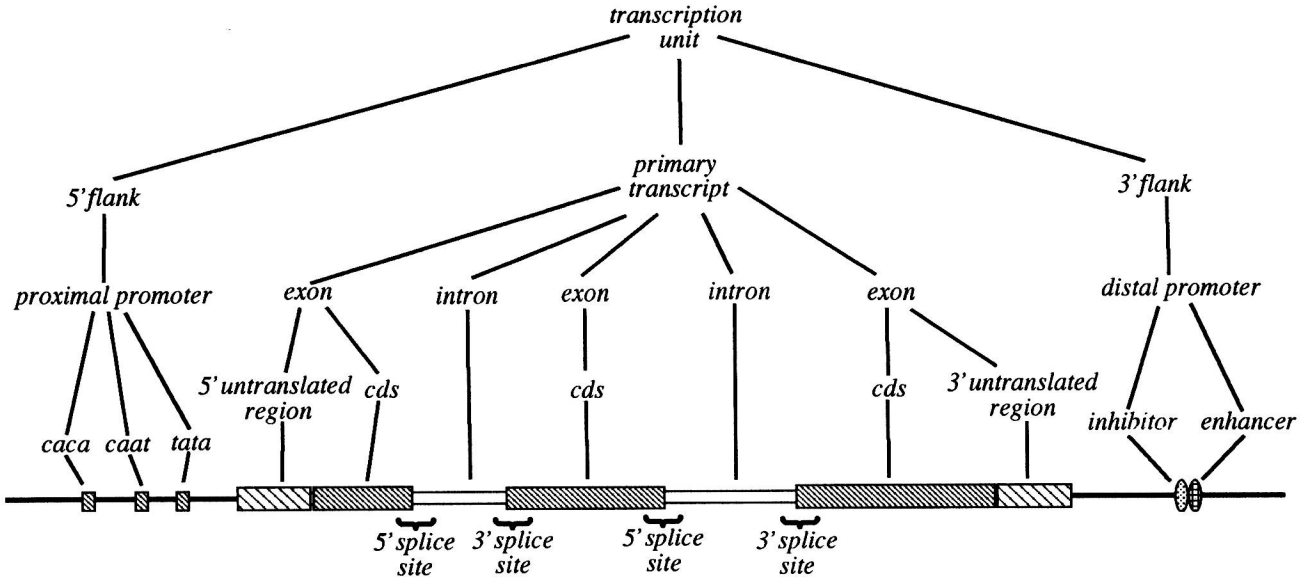

Figure 1: Idealized view of a parse tree for a typical eukaryotic protein coding gene.

DCG rules are of the form `LHS => RHS`, where the `LHS` (left-hand side) is a non-terminal and the `RHS` (right-hand side) any combination of terminals and non-terminals. **Terminals** correspond to words of the sentence being parsed (the leaf nodes of the parse tree), and **non-terminals** represent sets of phrases (subsequences of sentences) as defined by the grammar. Each interior node of a parse tree corresponds to a non-terminal as the sequence of terminals underneath such a node is one of the phrases of that non-terminal. The `LHS` non-terminal in the toplevel grammar rule is termed the start symbol.

In the context of nucleic acid sequences (NA) the distinction between terminals and non-terminals is less clear since genes can be described and investigated at various levels of abstraction. As shown in the parse tree of Figure 1, subsequences which may be considered as terminals in one context may become non-terminals in another (e.g., exons as subsequences of a primary transcript may be considered terminals, whereas exons would be non-terminals when parsed into coding sequences and untranslated regions).

Each feature of the FEATURE TABLE essentially describes one node (terminal or non-terminal) of the gene parse tree along with the DNA subsequence covered by it. These descriptions are often redundant, incomplete and even inconsistent, and the task of the SSP is to assemble complete parse trees expressing the same information in a structured non-redundant consistent fashion. Below is a simple example of an NA grammar rule expressing the fact that a transcription unit consists of a 5' flanking region, followed by a primary transcript and a 3' flanking region:

```
transcription_unit =>
   5'flank, primary_transcript, 3'flank.
```

Since grammatical elements (terminals and non-terminals) correspond to intervals of NA (sub)sequences [Overton et al., 1989], grammar rules can be naturally interpreted as interval relationships where '`=>`' means the interval on the `LHS` contains the intervals on the `RHS` ("part-whole" relationship), and a '`,`' between intervals means that the end of the first interval is the beginning of the second interval ("order of parts" relationship). Techniques have been developed for reasoning about temporal intervals [Allen, 1983], and these techniques can be extended to cover NA intervals. Incorporating these techniques into grammar rule formalisms makes it possible to model other interval relationships such as overlaps, starts, and ends [Pastor et al., 1991].

Contrary to standard parsers that take as input a list of terminals, the input to the SSP may contain non-terminals as well. To facilitate efficient processing the grammatical elements (features) on the input list are ordered by their start positions, lengths and ranks in the grammar hierarchy; for example, an exon occurs before a CDS fragment with the same boundaries. The square bracket notation, `[]`, is used to remove and add elements to the input list. When used on the `RHS` of a rule, they remove grammatical elements, and when used on the `LHS` they add elements. Therefore, an element can be removed, examined and replaced on the input list as in the following example which tests if the 5'flank boundary has been reached:

```
5'flank, [primary_transcript(S,E,I)] =>
    gap, [primary_transcript(S,E,I)].
```

A)

```
cluster(pos(<,[0,0]),pos(>,[1138,1138]),(
 t_u(pos(<,[0,0]),pos(>,[1138,1138]),(
  f_f(pos(<,[0,0]),pos(=,[97,97]),(
   gap(pos(<,[0,0]),pos(=,[26,26]),()
   promoter(pos(=,[26,26]),pos(=,[31,31]),()
   gap(pos(=,[31,31]),pos(=,[69,69]),()
   promoter(pos(=,[69,69]),pos(=,[72,72]),()
   gap(pos(=,[72,72]),pos(=,[97,97]),())
  p_t(pos(=,[97,97]),pos(=,[929,929]),(
   exon(pos(=,[97,97]),pos(=,[230,230]),(
    five_utr(pos(=,[97,97]),pos(=,[134,134]),(
     gap(pos(=,[97,97]),pos(=,[134,134]),())
    cds(pos(=,[134,134]),pos(=,[230,230]),())
   intron(pos(=,[230,230]),pos(=,[347,347]),(
    gap(pos(=,[230,230]),pos(=,[347,347]),())
   exon(pos(=,[347,347]),pos(=,[551,551]),(
    cds(pos(=,[347,347]),pos(=,[551,551]),())
   intron(pos(=,[551,551]),pos(=,[691,691]),(
    gap(pos(=,[551,551]),pos(=,[691,691]),())
   exon(pos(=,[691,691]),pos(=,[929,929]),(
    cds(pos(=,[691,691]),pos(=,[820,820]),()
    three_utr(pos(=,[820,820]),pos(=,[929,929]),(
     gap(pos(=,[820,820]),pos(=,[908,908]),()
     pA_sig(pos(=,[908,908]),pos(=,[914,914]),()
     gap(pos(=,[914,914]),pos(=,[929,929]),())))
  t_f(pos(=,[929,929]),pos(>,[1138,1138]),(
   gap(pos(=,[929,929]),pos(>,[1138,1138]),())))
```

B)

FEATURES	Location/Qualifiers
CDS	join(135..230,348..551,692..820)
	/codon_start=1
	/translation= NOT SHOWN
precursor_RNA	98..929
	/note="primary transcript"
mRNA	98..230
mRNA	348..551
exon	692..929
	/number=3

C)

FEATURES	Location/Qualifiers
trans_unit	<1..>1138
	/gene="alpha-globin"
5'flank	<1..97
promoter	27..31
	/sequence="ccaat"
promoter	70..72
	/sequence="ata"
prim_transcript	98..929
	/note="primary transcript"
exon	98..230
5'UTR	98..134
CDS	join(135..230,348..551,692..820)
	/codon_start=1
	/translation= NOT SHOWN
intron	231..347
exon	348..551
intron	552..691
exon	692..929
	/number=3
3'UTR	821..929
pA_sig	909..914
	/sequence="aataaa"
3'flank	930..>1138

Figure 2: The parse tree, the original FEATURE TABLE, and the augmented FEATURE TABLE for the HUMAGL1 α-hemoglobin gene. A) standard representation of the parse tree for HUMAGL1 generated by the SSP; B) the FEATURE TABLE as found in GenBank; C) a representation of the parse tree for HUMAGL1 as a corrected, augmented FEATURE TABLE. Note that the start position for each interval in the parse tree is one less than the start position in the corresponding FEATURE TABLE entry because the SSP indexes on the space between characters rather than the character itself.

where the logic variables `S` and `E` represent the start and end positions of the interval, and `I` provides context information about the features.

Alternative rule applications (disjunction) can be expressed as follows:

```
5'flank => promoter, 5'flank.
5'flank => gap, promoter, 5'flank.
```

and recursion is illustrated in this example:

```
primary_transcript =>
   exon, intron, primary_transcript.
primary_transcript => exon.
```

Practical grammars also need escapes to the underlying implementation language. Such escapes are also available in NA grammars to handle exceptional situations such as erroneous and missing input data.

We developed an NA grammar for the class of genes that code eukaryotic proteins. It has been successfully applied to a large number of eukaryotic globin genes such as the human α-hemoglobin gene entry (HUMAGL1) shown in Figure 2B. The parse tree generated from this table is shown in Figure 2A. Note that this parse tree includes two promoters and one polyA-signal that were inferred by the techniques discussed below. In addition, the SSP inferred that the label `precursor RNA` should be changed to `primary transcript`, two features listed as `mRNA` are actually exons, and two introns were missing from the original FEATURE TABLE.

This example illustrates only a few of the problems that complicate the development of grammars with broad coverage. Apart from mislabeling and omission of features, the SSP has to deal with FEATURE TABLEs describing multiple genes or alternative versions of genes (e.g., alternative splice sites, transcription start sites, and polyA additions sites). In such cases it is often necessary to refer to the qualifier fields

此为试读,需要完整PDF请访问: www.ertongbook.com