Section 16. Applications

The Invited Speaker, Bonnie Berger, was not able to attend the Congress.

Recent Developments in Computational Gene Recognition

Serafim Batzoglou, Bonnie Berger, Daniel J. Kleitman, Eric S. Lander, and Lior Pachter

Abstract. We survey recent mathematical and computational work in the field of gene recognition, focusing on the techniques that have been developed to tackle the problem of identifying protein coding regions in genes. We also present a new approach to gene recognition which is based on a variety of tools we have developed.

1 Introduction

1.1 What do you do with 100KB of human genomic DNA?

Recent advances in DNA sequencing technology have led to rapid progress in the Human Genome Project. Within a few years, the entire human genome will be sequenced. The rapid accumulation of data has opened up new possibilities for biologists, while at the same time unprecedented computational challenges have emerged due to the mass of data. The questions of what to do with all the new information, how to store it, retrieve it, and analyze it, have only begun to be tackled by researchers [11]. These problems are distinguished from classical problems in biology, in that their solution requires an understanding not only of biology, but also of mathematics and computer science. Of the many problems, it is clear that the following tasks are of importance:

- Finding genes in large regions of DNA.
- Identifying protein coding regions within these genes.
- Understanding the function of the proteins encoded by the genes.

The important third problem, namely understanding the function of a newly sequenced gene, requires the solution of the second problem, identification of critical subregions which code for protein. Protein coding regions have different statistical characteristics from noncoding regions, and it is primarily this feature which

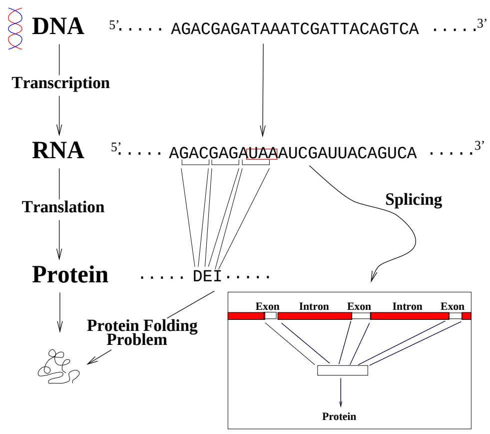

Figure 1: A schematic view of the transcription-translation process: During translation the T nucleotide becomes a U (Uracil). In this example, the boxed UAA triplet is not a codon and therefore does not end translation. Rather, the in-frame codons are "...GAC GAG AUA...". These are translated into "...D E I..." (D=Aspartic Acid, E=Glutamic acid, I=Isoleucine). Splicing occurs before translation. The translated amino acid sequence is folded into a protein.

enables us to distinguish them. An important aspect of work on the problem is the need to characterize these statistical differences and possibly explain their biological underpinnings. This paper surveys recent mathematical and computational approaches to developing algorithms for identifying protein coding regions within genes, and discusses some new methods we have recently developed.

1.2 Biological Background

For the purposes of our discussion, we will define a gene (see Figure 1) to be a single, contiguous region of genomic DNA that encodes for one protein (along with the $5'$ and $3'$ flanking regions that contain promoter signals, etc.) There are four different nucleotides that make up a sequence of DNA. These are Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). For our purposes, we will think of DNA as being a string on an alphabet of size 4 (A,C,G,T). When a gene is expressed, it is first copied in a process known as transcription. This forms a product known as RNA, which is a working template from which a protein is produced in a process known as translation. Before translation, the RNA undergoes a splicing operation [14] conducted by certain enzymes, which typically delete most of it, leaving certain blocks of the original strand of RNA intact. These blocks are called exons and the parts that are removed are called introns. The result of this pruning is the "mature" RNA, which is used during translation to make the protein. The protein consists of a sequence of amino acids linked together. During translation, each amino acid is produced by a triplet of consecutive nucleotides, known as a codon, according to a known map that is called the GENETIC CODE. This defines the CODING FRAME of the gene.

The gene actually has a "start" translation signal (ATG) and a stop translation sequence (TAA, TAG or TGA) both within exons; the sequence within exons between these forms the coding part of the sequence which contains all the information used to make the protein. The rest of the gene consists of introns, initial and final "non-coding exons" (these are exons that are glued together with the coding exons, but that are not used for making protein), as well as flanking regions containing biological signals of various sorts.

The splicing process is partially understood, and various SnRNP's (these are RNA-protein complexes involved in splicing) have been identified that are involved in the splicing mechanism. These SnRNP's (or spliceosomes) recognize various DNA sequences during splicing, and information about the consensus sites they recognize can be used to identify splice sites. Unfortunately, the biology is not understood to an extent that makes gene recognition possible on this basis alone [4]. Indeed, one of the main challenges for mathematicians and computer scientists working on these problems is to help biologists learn about splicing by detecting biologically significant signals in genomic databases.

1.3 THE COMPUTATIONAL TASK

The computational task we are concerned with is that of determining from an experimentally determined sequence of nucleotides, of length on the order of 100,000, where the genes are, and what proteins these genes produce. This endeavor has two parts, though in practice one handles them together: determining where each gene is, and determining which parts of its sequence are exons and which are introns. Here we focus on the latter of these two problems.

Nature uses a variety of biological signals, many of which remain to be identified. Fortunately (in view of our ignorance of the actual biological mechanism), we are not restricted to using only biological signals used by the cell. First, we know quite a bit about the constitution of intergenic and intronic sequences in humans. On the order of 30 percent of these sequences consist of certain REPEATS of various standard patterns or variations thereof [15]. Thus there are several hundred thousand copies of one or another variation of a sequence of length about 300 called Alu in the human genome, and many copies of other sequences as well. Due to the migratory nature of these repeats, and the mechanisms by which they occur, they are rare in exons. Secondly, the codons (and consequently amino acids) that code for protein, are not uniformly distributed, and their distribution differs from that of triplets in introns. This can help in distinguishing introns from exons. Other restrictions such as consistency in coding frame between exons greatly reduces the number of possible parses in a given gene. Indeed, even though in principle the number of parses is exponential in the number of potential splice sites identified, in practice many genes exhibit only a few possible parses after these numerous constraints are introduced.

The data available to us comes from a number of data bases, which contain examples of various kinds of biological sequences, as follows:

- The protein data base; it contains proteins whose amino acid sequences have been determined.

- The cDNA data base; it consists of what are essentially the DNA sequences of the exons of a gene only, and fragments thereof.
- Data bases of genes whose splicings into introns and exons are known.
- Data bases of genes of various species without such information.

There are numerous complications in this problem, perhaps the most significant of which is the unreliable nature of the annotated data. There are also examples of genes which have "alternate splicings" so that under different circumstances the same gene can produce different proteins by being spliced differently into introns and exons. Finally there are introns whose splice sites are very different from the common consensus, not to mention numerous other exceptions to "the rules."

2 Previous Work

Current methods can be broadly categorized as learning, or homology based. While we cannot attempt to discuss in detail the myriad of approaches available, we will briefly comment on two methods currently in use, namely the HMM (Hidden Markov Model) approach (used by `GENSCAN` [2] , `GENEMARK` [12] and `GENIE` [10]) and the homology based method (e.g. `PROCRUSTES` [6] and `AAT` [9]).

2.1 Learning based methods

Many of the most popular learning methods are based on a Hidden Markov Model approach [13] (although there are some notable exceptions to this, for example the language based system used in `GENLANG` [5]). It is assumed that the gene structure of a certain organism can be modeled probabilistically, with certain probabilities associated with being in certain "states," and transition probabilities associated with these states. The states usually model functional units of a gene, for example exons (in the three different reading frames), introns (sometimes also in three different flavors depending on the frame of the exon preceding them), as well as terminal and initial exons, etc. The exact true model to be used is "learned" from the data. Coding and non-coding exons are usually modeled using 3-periodic fifth-order Markov Models. The exact methods used to model the various other biological signals (splice sites, etc.) vary greatly between the different programs.

The main drawback of many of these approaches is that performance is very dependent on the learning sets used [7]. Generally, only a single data set, developed by Haussler, Kulp and Reese [8], has been used for training. Overtraining of the Markov Models leads to poor results when new genes are encountered. This is especially true in genomics because early sequencing efforts tended to focus on gene rich areas in the genome, leading to an overabundance of short, GC rich genes. Some programs such as `GENSCAN` have begun to deal with this issue by separately handling GC rich and GC poor gene candidates.

Another drawback to learning methods is that homology information is not used for the predictions (this is starting to change with the advent of homology integration in programs such as `GENIE`). The user of the program is responsible for performing his/her own homology searches using `BLAST` [1] or another program.

Despite these drawbacks, the utility of the programs mentioned cannot be overlooked. Indeed, the `GENSCAN` package is becoming increasingly popular amongst biologists, and other programs such as `GRAIL` [16] (based on neural networks) have been in use for years.

2.2 Homology Based Methods

The `PROCRUSTES` program [6] approaches gene recognition in a new, interesting way. The basic idea is that given a protein that is a homolog of the protein produced by the gene to be solved, one can determine the best way to parse the gene so that the resulting translated union of coding exons most closely resembles the target protein. This procedure can effectively be carried out using dynamic programming. Of course, the method is useless unless one can find a "good" match to the gene in question in a protein database (the `PROCRUSTES` program requires the user to find this input). Recently, cDNA databases are being used in analogous ways [9]. Exact estimates of how often these methods can be employed on new genes vary. Guesses range from 30-50 percent, with optimists arguing that these numbers will improve as the size of the databases increases.

2.3 Previous Results

The analysis and benchmarking of gene recognition tools has become a science in and of itself. Of the many articles addressing these issues, we mention the excellent surveys of Burset and Guigó [3, 7]. The non-homology based algorithms are not sufficiently accurate to be relied on. Accuracy claims range from 60-90 percent per nucleotide, and 30-80 percent per entire exon with exact numbers dependent on who is making the claim. In practice these numbers are probably very optimistic [7]. Indeed, on a new sequence set, the programs identified about 1 in 6 genes correctly and completely missed the exons in 25 percent of the sequences.

The alarming aspect of the current state of the field is that these programs perform much worse when tested on new data, namely genes that have been sequenced, whose intron/exon structure is known experimentally. This poor performance is probably due to a number of factors, the most significant of which is that current "learning" takes place on small data sets which are often filled with errors since they have been annotated by the very same programs that are learning from them!

In practice, those who find genes use a very different approach. They hope that the cDNA or protein (or a good part of these) that are produced by the gene lie in one of the corresponding data bases. They then submit their sequences to `BLAST` [1], a program that finds best matches to members of the data base. When it is possible to match parts of the gene with an entire protein, then one has the answer to our problem, either by examining the alignments by eye, or submitting the matches to a program such as `PROCRUSTES` [6]. As the databases grow, the likelihood of good matches to new genes increases. When this approach fails, they turn to the algorithms mentioned, and seek consensus results from them. The process is tedious, time consuming and does not necessarily produce correct results.

3 Innovations

We have developed a program (unpublished manuscript) based on the following ideas:

- Use of larger data bases such as the protein data base as a data source not only for homology, but for methods based on frequencies of k-tuples of nucleotides and amino acids. This greatly extends the amount of data available, and therefore allows consideration of k-tuples of much greater length than have been used heretofore.
- Use of a dictionary approach for finding matches as well as computing k-tuple frequencies from the databases. The idea of a dictionary has potential applications that go well beyond this particular problem.
- Attempt to use many separate indicators to distinguish exons, rather than integrating them immediately into one overall statistic.
- Use of not necessarily consecutive subsequences of nucleotides in our analysis.
- Distinguishing relatively long and not necessarily consecutive sequences of nucleotides and amino acids that occur unusually often in introns or exons, but not both, as markers for the same.
- Use of frame differentiation as an indicator for exons.
- Development of a visual program, which allows a user to see and evaluate predicted introns and exons, and experiment with alternative splicings, as well as predictions based on homology.
- Use of expected number of hits, rank statistics and other indicators in place of single maximal likelihood estimates.
- Use of gene data bases for homology-based identification of exons.
- Integration of repeat masking into the gene recognition process.
- Integration of homology-based and statistical approaches in the same program.
- Fast predictions using the above techniques, allowing for multiple homologs to be used in an automated fashion.

We briefly elaborate on two of these ideas below.

3.1 A FRAME TEST

Exons can be distinguished from introns in several ways. First, the nature of the translation code along with the nature of most proteins implies that the three possible reading frames (the first, second or third positions among the triplets that go to produce an amino acid) exhibit behavior that is usually quite different from one another. That is, if one examines a sequence of length 3 or more of nucleotides, one often finds that this sequence occurs much more often in one frame than another. This phenomenon becomes much more pronounced as the length of the sequence increases. Thus, most sequences of length 12 seem to have a pronounced bias toward a particular frame. The reason for this bias has to do with the genetic code. Mutations in the third position of a codon have much less effect on the resulting amino acid than, say, a mutation in the first position. Furthermore, an exon that is subject to an insertion or deletion of a single nucleotide will be translated into a completely different protein. Such changes are usually for the worse because natural selection has selected against them. There is much less of such strict conservation in introns. A single deletion or insertion appears to have little effect in an intron on anything, unless it occurs in a rare crucial place that will prevent the enzymes from splicing the intron. Perhaps for such reasons, introns tend not to show the frame bias seen in exons. In consequence, examining the presence or absence of consistent frame bias provides a good first reading of where the larger exons of the gene are. Furthermore some DNA subsequences look much more like exons than introns or vice versa, and detecting the presence of such can also help distinguish introns from exons.

Indeed, the problem of determining the frame of an exon is essentially resolved using such frame differential methods. Using the above mentioned techniques, and examining rare subsequences, we can identify the frames of exons correctly 98 percent of the time.

Since the frame information is heavily dependent on the subsequence length used, the information becomes more definitive as the subsequence gets longer. It is valuable to use as large a data set as possible for determining which sequences look like what. The data sets usually used on this problem provide only enough data to consider 6-tuples of nucleotides, whose length is that of two amino acids in the resulting protein. The data has more intron information, and provides useful frequency data for sequences of length up to 9 in introns. Much larger data sets can be exploited by using protein and cDNA databases. We discuss this idea next.

3.2 DICTIONARY APPROACHES

The protein and cDNA data bases contain information derived exclusively from exons. However the latter is complicated because it contains both fragments of coding exons, and also fragments which include non-coding exons. The latter tend to look very different from the coding exons we wish to find, and in fact look much more like introns than like coding exons, in general. (As always there are exceptions.) The cDNA data base is also complicated by the fact that the gene can lie on either strand of the DNA, so that the cDNA can represent the reverse complement of the original DNA sequence (where complementation interchanges

C with G and T with A.) Our approach to utilizing these data bases is to compile dictionaries of fragments of protein script and of cDNA script. These differ from ordinary dictionaries since we really do not know how to distinguish words. This means we can define wordlets to be sequences of any kind we choose. Instead of giving the meaning of such wordlets, which we would dearly love to do, the dictionary provides for each wordlet in it, a list of all members of the corresponding data base that contain it. We have compiled such dictionaries, and it is not difficult to do so on not very expensive computing machines both for cDNA data bases and the protein data base. In the former we have done so for nucleotide sequences of length 11, and also for sequences having 11 significant places with every third place skipped (hence length 16.) In the protein data base we have constructed a dictionary of amino acid sequences of length 4. We have used these numbers because they are convenient; furthermore they permit conclusions to be drawn about longer sequences as well, so that we have not yet encountered a need for a dictionary with longer wordlets.

Such dictionaries have immediate application to finding homologies where they exist in these data bases, that is, to finding members of them which are the product of the gene in question or share one or more of its exons, or resemble the products of these exons. For by looking up each wordlet of appropriate kind that occurs in the gene under consideration, one can compute how many wordlets each entry in the data base shares with the gene in question. One can, furthermore, use the protein data base dictionary with wordlets of length 4 to find how many wordlets of length 5 or 6 or etc., each entry shares with the gene in question. In our case there is little noise for length 5, and by ordering the entries according to the number of wordlets of length 5 in common with our gene, and examining the top segment of the ordered list, we can see which proteins share exons with our gene, and can quickly identify any proteins homologous to any parts of it. The cDNA data base and appropriate dictionaries can be used for the same purpose. It is also possible to use the wordlet frequency information contained in the protein data base dictionary as an intron/exon indicator. We suggest ordering the wordlets according to frequency of occurrence in the data base, and summing the ranks in a moving window of 25 successive wordlets for each frame to indicate exons. Tests suggest that this method is an improvement over the use of raw frequency data. In particular, some false positive exon signals are removed. The use of ranks also limits the sensitivity of the prediction methods to the learning data.

4 Discussion

This problem we have discussed can be viewed, in part, as follows: we have a script that is written in interspersed parts in two “languages”; there are characteristic transitions between one and the other, and these are helpful for identification purposes, but only up to a point. (Subsequences that resemble transitions between intron and exon occur fairly often inside introns and sometimes inside exons). Our task is to distinguish the parts in each language. The natural hope is that one can identify introns by the mechanism through which they are spliced out in the process of protein making. These mechanisms involve enzymes which interact with both end segments of the intron that is cut out. If we could understand how the RNA strand arranges itself in the presence of these enzymes, understanding of the splicing process could allow us to predict what will be spliced. However, while we can extract some clues as to how good a potential splice site looks, such clues are not enough to solve the problem. The major clues which seem to help the most come from recognizing which parts of the gene appear to be written in "exon script". By "exon script" we mean subsequences that can be translated into sequences of amino acids which "make sense" as protein parts. As discussed in the previous section, various tests designed to extract distinguishing features can prove to be very useful. The dictionaries we have created also add a large range of sources from which we can obtain data for the various tests.

The dictionary idea has many potential applications in understanding protein secondary structure and function as well. Most approaches to understanding how proteins fold together and what they do have been based on global considerations. However, it is known that perhaps half of proteins on the average consist of certain specific structures, in particular alpha helices, beta sheets, loops, etc. We cannot expect to understand a paragraph merely by identifying the presence of words that we do not understand, and these may occur in paragraphs with entirely different meanings. Nevertheless linguists have obtained remarkable conclusions by examining word frequencies in texts (such as the claim that certain books of the bible were actually written by several different authors.) It may well be that protein wordlets that often occur in alpha helices, for example can provide clues as to the folding of other proteins that contain them, so that it may be possible after all to assign meaning to at least some of the wordlets in these dictionaries.

There is insufficient space left for us in this paper or on its margins, so that results will be reported elsewhere.

5 Acknowledgments

We thank Eric Banks, William Beebee, John Dunagan, Nick Feamster, Aram Harrow, Julia Lipman, Valentin Spitkovsky, Tina Tyan and Bill Wallis for helping in countless ways with the implementation of the ideas outlined in this paper. This project has been supported by Merck. Pachter has been partially supported by an NIH training grant and a Program in Mathematics and Molecular Biology graduate fellowship.

References

[1] S. F. Altschul, S. F. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[2] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268:78–94, 1997.

[3] M. Burset and R. Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–367, 1996.

[4] J. M. Claverie, O. Poirot, and F. Lopez. The difficulty of identifying genes in anonymous vertebrate sequences. *Computers and Chemistry*, 21(4):203–214, 1997.
[5] S. Dong and D. B. Searls. Gene structure prediction by linguistic methods. *Genomics*, 23:540–551, 1994.
[6] M. S. Gelfand, A. A. Mironov, and P. A. Pevzner. Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci. USA*, 93:9061–9066, 1996.
[7] R. Guigo. Computational gene identification: an open problem. *Computers and Chemistry*, 21(4):215–222, 1997.
[8] D. Haussler, D. Kulp, and M. Reese. A representative benchmark gene data set. *http://www-hgc.lbl.gov/inf/genesets.html*, 1996.
[9] Xiaoqiu Huang, Mark D. Adams, Hao Zhou, and Anthony Kerlavage. A tool for analyzing and annotating genomic sequences. *Genomics*, 46:37–45, 1997.
[10] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. A generalized Hidden Markov Model for the recognition of human genes in DNA. *Proceedings of the 4th Conference on Intelligent Systems in Molecular Biology*, 1996.
[11] E. S. Lander. The new genomics- global views of biology. *Science*, 274(5287):536–539, 1996.
[12] Alexander V. Lukashin and Mark Borodovsky. GENEMARK.HMM: new solutions for gene finding. *Nucleic Acids Research*, 26(4):1107–1115, 1998.
[13] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–285, 1989.
[14] P. A. Sharp. Split genes and RNA splicing. *Cell*, 77(6):805–815, 1994.
[15] A. F. A. Smit. Origin of interspersed repeats in the human genome. *Curr. Opin. Genet. Devel.*, 6(6):743–749, 1996.
[16] Y. Xu, R. J. Mural, M. Shah, and E. C. Uberbacher. Recognizing exons in genomic sequences using GRAIL II. In Jane Setlow, editor, *Genetic Engineering: Principles and Methods*, volume 16. Penum Press, 1994.

Serafim Batzoglou
Laboratory for Computer Science
Massachusetts Institute
of Technology
Cambridge, MA 02139
USA

Bonnie Berger and Lior Pachter
Department of Mathematics and
Laboratory for Computer Science
Massachusetts Institute
of Technology
Cambridge, MA 02139
USA

Daniel J. Kleitman
Department of Mathematics
Massachusetts Institute
of Technology
Cambridge, MA 02139
USA

Eric S. Lander
Whitehead Institute and
Department of Biology
Massachusetts Institute
of Technology
Cambridge, MA 02139
USA