Eric E. Snyder
Associate Professor
Pennington Biomedical Research Center
Louisiana State University
6400 Perkins Road
Baton Rouge, LA 70808-4124
USA

Office Phone: (225) 763-3185
Mobile: (225) 235-6271
Office Email: eesnyder@pbrc.edu
N 030°23'18. 8" W 091°07'59. 4"
January 7, 2004 11:57:53

Professor Rob de Ruyter van Steveninck
Biocomplexity Faculty Search Committee
Biocomplexity Institute
Indiana University
Swain Hall West 117
Bloomington, IN 47405-7105
USA

Dear Professor de Ruyter van Steveninck,

I would like to apply for the position of Assistant or Associate Professor, at Indiana University, as described in your October 30, 2003, advertisement in *Nature*. I have included: a representative paper and my *curriculum vitae*. I have also arranged for three professional references to send letters on my behalf.

I hope that my interests and expertise may find a good fit within your institution.

Sincerely,

Eric E. Snyder

Research Interests and Plans

The accurate prediction of gene structure in eukaryotes is an important problem for computational molecular biology and has significant biomedical implications. It can also be a platform for testing our understanding of transcription, pre-mRNA processing and translation. In the last 15 years, great strides have been made towards the goal of predicting the complex intron/exon structure of mammalian genes. Yet, even the best programs claim to accurately predict the complete coding region of a typical human gene in less than half of the test cases. Clearly, there are significant gaps in our understanding of the fundamental principles governing pre-mRNA processing. These gaps are even larger than this evidence indicates because all gene prediction programs to date have been pragmatic efforts to maximize predictive performance, rather than systematic efforts to model the behavior of the biological system. The lessons learned from these programs are further compromised by the limited and often unrealistic training and test sets used.

While some view this as evidence for the inadequacy of current algorithms, others question whether it still makes sense to expect a single canonical structure for every gene. This latter view is engendered by the growing appreciation for the widespread role of alternative splicing in the generation of protein diversity. This process, in which a single transcript can be spliced into different forms, dramatically complicates the gene structure prediction problem. It also encourages a major rethinking of this problem and the assumptions that have gone into the current approaches to its solution. My current interests are therefore focused on analyzing the factors involved in alternative splicing with the goal of developing a *biological* model for transcription and pre-mRNA processing. This model will guide the construction of a *computational* model, which can predict the most likely mRNA products produced from a primary transcript under a particular set of conditions.

Solving this problem will require research on many fronts:

- splice site identification and modulation
- tissue-specific splicing factors
- splicing kinetics and co-transcriptional pre-mRNA processing
- exon assembly algorithm design and development
- content statistics for exons, introns, UTRs and intergenic regions
- integration of similarity-based measures into gene predictors

Splice Site Identification and Modulation

The separation of splice sites from non-sites is a classic problem in motif analysis. Although many complex methods such as neural networks, maximal dependence decomposition and discriminant analysis have been applied, none offer dramatically better performance than a simple weight matrix model. This suggests that a significant part of what makes a splice site functional must lie outside the site itself. The identification of proteins that enhance or suppress the activity of splice sites by binding to nearby *cis* elements has validated this notion. Although the binding sites for these factors have been characterized, no one has effectively used this information to improve splice site recognition.

Tissue-Specific Splicing Factors

Alternative splicing factors are expressed in a tissue-dependent manner. This has made it difficult to use information about their binding sites for splice site recognition because gene structure is not typically considered to be tissue dependent, in spite of the fact that alternative splicing is prevalent and well-documented phenomenon. To make progress in splice site identification, we must consider not only the adjacent sequence for splice enhancers (or suppressors) but do so in a manner that takes into account the presence of the factors which bind to them. This information is not readily available now but can easily be obtained. Several large-scale full-length cDNA sequencing projects including the RIKEN mouse project and the Mammalian Gene Collection effort are capable of providing this information. One needs to measure the expression levels of alternative splicing factors in each of the ~50 libraries that contribute the majority of full-length cDNAs (and therefore gene structures and splice site positions). This assumes that each mRNA expression level correlate with the activity of the corresponding alternative splicing factor. If so, this data will help to study splice sites (and gene structure generally) under more defined conditions that have been possible previously.

Splicing Kinetics and Co-Transcriptional Pre-mRNA Processing

There is increasing evidence that splicing and other pre-mRNA processing activities are tightly coupled to transcription. Co-transcriptional splicing has a number of implications that have yet to be exploited for gene prediction applications. For example, insertion of transcriptional "pause" signals has been shown to affect splice site selection *in vivo* and *in vitro*. This suggest that in some circumstances, diffusion-limited assembly of the splicing complex may be rate limiting and that slowing transcription allows splicing to proceed on more proximal sites. On the other hand, the carboxy-terminal domain (CTD) of the RNA polII large subunit has been shown to bind splicing factors, thereby tethering them to the nascent pre-mRNA as it emerges from the polymerase complex. This has the effect of increasing the local concentration of these factors in the vicinity of the splice sites, creating, in effect, zero-order kinetics. By understanding factors influencing transcriptional velocity and enabling preloading of spliceosome components on the CTD we will gain invaluable insights into pre-mRNA splicing prediction.

Development and Testing of Exon Assembly Algorithms

In 1993, Gary Stormo and I introduced the use of dynamic programming (in the program GeneParser) as means of finding the optimal assembly of scored introns and exons into a gene prediction. Over the next decade, the basic method has been extended in a variety of ways. The most important was the application of hidden Markov models by Burge and Karlin in 1997 with the program GenScan; in the same year, Reese, *et al.* introduced a program called Genie using a very similar methodology. A significant problem in comparing the accuracy of these and related methods is that different methods are used to assess motifs (such as splice sites) and content measures (such as codon usage), making it impossible to determine whether performance differences are due to these factors or improvements in the assembly algorithm. To fairly compare assembly algorithms, they need to be compared using the same content and site statistics. To do this, I plan to build a modular gene prediction program in which the various components can be readily substituted. A thorough analysis of the assembly algorithms used by popular gene

prediction programs "reconstituted" using standardized components should help dispel confusion over the factors responsible for the differing levels of performance between commonly used programs. It also sets the stage for building a computational model of transcription and pre-mRNA processing.

Development of Content Statistics

Content statistics measure properties over an entire sequence interval, as opposed to site statistics, which are measured at a particular position in a sequence. The most well known content measure is the codon usage statistic, which measures the unequal use of nucleotide triplets characteristic of coding exons. This class includes related measures such as codon bias (the unequal usage of synonymous codons) and higher order frequency tables based on hexamers or octamers, as well as measures which are less obviously tied to the coding function such as local compositional complexity or CpG suppression. The prediction of 5'- and 3'-UTRs will be an essential part of gene prediction and modeling efforts; the development of new content measures will be important in the discrimination of these areas from adjacent intergenic regions.

Application of Similarity-Based Measures to Gene Prediction

The GeneParser program was also the first to use sequence similarity as evidence for protein coding potential. Genomic sequence intervals that contained BLASTX alignments between the genomic sequence of interest and a protein database were considered more likely to be coding that those that did not. Conversely, regions that aligned to characteristically intronic repetitive sequences such as Alu and L1 repeats were given negative scores for exon likelihood and positive scores for intron likelihood. From a purely predictive standpoint, the use of sequence similarity in gene prediction is a powerful adjunct. It assumes the presence of homologous sequences in the database and will not help to identify truly novel genes. Fortunately, as more and more organisms are sequenced to completion, the probability of encountering a gene with no paralogues continues to fall. As a result, this technique will be important to include in the gene prediction system under development. However, since knowledge of homologous sequences is clearly not a part of spliceosome functionality, it is not appropriate to include this information in the modeling system.

Interaction with Laboratory Researchers

Expression Analysis

Since predicting the sequence of the processed mRNA is the ultimate goal of this work, it is essential to have good experimental data on which to base our models. The high-throughput sequencing of full-length mRNAs from mouse (RIKEN) and other mammals including humans (MGC) are our basic working data sets. They are preferable to datasets based on RefSeq because the sequences are derived from a relatively small number of cDNA libraries constructed from well-defined tissues. Although there are on average two distinct cDNA species for every known gene, this is a low level of redundancy, which will result in a very sparse matrix of genes, tissues and spliced isoforms. Collaboration with other laboratories studying tissue-specific alternative splicing will be essential to create a database of sufficient depth for accurate model building. It is quite

reasonable to expect that the first successful models of alternative splicing will involve only a small number of very well studied genes.

Quantitation of Alternative Splicing Factors

The ability to model the behavior of splice enhancers and suppressors is contingent on knowing the circumstances under which these motifs exert their effect. Since the activity of splice enhancers is mediated through specific alternative splicing factors, the expression of which can be estimated by traditional means (at the protein or nucleic acid level). I am currently looking for a collaborator who would be in a position to do quantitative PCR on the unnormalized RIKEN and/or MGC libraries. Even qualitative expression data on these factors would be useful when combined with the repertoire of spliced cDNAs from each library.

Summary

Gene prediction algorithms have evolved greatly over the last decade and have reached a performance ceiling. By exploiting these tools for new purposes, we can leverage this evolution to solve new problems. I believe understanding alternative splicing is one of the most important issues facing molecular biology today. Coupling existing algorithms with new sources of experimental data will allow us to ask much more precise questions of our genomic sequences. The naive question, "what is *the* sequence of the protein encoded by this gene?" can be replaced with, "how is my gene spliced under these circumstances, in this particular tissue?" The ability to answer this question will depend not only on the genomic sequence itself but also on the expression of genes that modulate pre-mRNA processing.

Teaching Interests

Background and Motivation

My career path was greatly influenced by a computational biology course I took as an undergraduate. Gary Stormo, who would later become my Ph.D. mentor, taught the course. At the time, I was working as a research assistant in a biochemistry lab doing detailed structure-function studies on calcium binding proteins. It seemed as thought scientists could spend his entire career laboriously working out the details of a class of proteins, perhaps even a single protein. Gary Stormo's class caught me at a time when I was wondering whether I wanted continue doing this sort of work. It opened my eyes to a whole world of problems for which data was already available—one needed only to analyze it in the appropriate way. This appealed to me for several reasons. Not only did the public sequence databases enable me to spend more time analyzing data, it allowed me to attack problems on a grander scale, problems that are more fundamental to biology as a whole. Consequently, I have wanted to develop a course of my own to share some of my enthusiasm for the subject and hopefully recruit new talent to the field.

Computational biology and bioinformatics mean many different things to different people. Sequence analysis is the sub-discipline at which I feel most at home, although expression analysis has become an important part of my own research.

The Course: Computational Analysis of Biological Sequences

Goals

I would like to create a computational biology course aimed at teaching the fundamental principles of nucleic acid and protein sequence analysis. Computational biology has become such an integral part of modern molecular biology that many practitioners of the latter do not understand the fundamental algorithms on which their tools are based. The goal of this course is to give students an understanding of the mathematical and algorithmic underpinnings of commonly used programs such as BLAST, Smith-Waterman, HMMER, GenScan, etc. By better understanding these programs, students can become more critical consumers of their output because they understand the assumptions and limitations inherent in the programs.

Requirements

The course will be one semester in length and aimed at third- or fourth-year undergraduates and graduate students. The course requires students to have had sufficient molecular biology and biochemistry to understand the central dogma in detail and have a working knowledge of enzymes of intermediary metabolism. Familiarity with a computer language such as C/C++ is not essential but highly recommended.

Structure

The class will be taught at two levels simultaneously. The undergraduate course will not require the use of programming skills. Problems sets would be limited to writing pseudocode and working our problems with pencil and paper. The graduate level course

will require the completion of additional exercises that requiring writing programs in C/C++ or another high-level language (at the instructors discretion). In some cases, student will be provided with a template containing code, which performs some basic functionality such as reading in sequences, allowing the students to focus on the biologically interesting aspects of the problem.

The function of the problem sets is to help students learn by doing. They will be graded but will only count for about 20% of the overall grade. Ideally, I would like to have two to three lectures a week, plus one or two study sessions with a TA where students can work through the problems in small groups. The TA would have an important responsibility to help the students teach themselves, rather than simply show them how to do the problems directly. Indeed, I would probably come to these sessions for the first few weeks to make sure the sessions were working as intended.

Class Project

If one wanted to eliminate the requirement for programming in the graduate level course, another possibility would be to add a class project in its place. Such a project would likely involve some sort of programming but in a less rigid sense. Students would be encouraged to select projects related to their own research using skills learned in the class. I would be happy to make suggestions to students unable to identify projects applicable to their own work.

Course Outline

1. Introduction (for biologists)
    a. Algorithms (DP, HMMs)
    b. Computational complexity
    c. Programming languages
2. Introduction (for computer scientists)
    a. The central dogma of molecular biology
    b. Biopolymers: DNA, RNA, protein
    c. Gene structure, protein functions, phylogenetics
3. Sequence alignment
    a. Pairwise alignment
        i. Longest common substring
        ii. Optimum global alignment (Needleman-Wunch)
        iii. Optimum local alignment (Smith-Waterman)
        iv. Protein alignments
            1. Substitution matrices
            2. Gap penalties
    b. Multiple alignment
        i. Dynamic programming approaches
        ii. Heuristics based on multiple pair-wise alignments
        iii. Other methods
    c. Fast database searching
        i. Hash-based methods
            1. BLAST

2. FASTA
ii. Interesting variations on BLAST program
1. PHI- and PSI-BLAST
iii. Hardware implementations
1. Peristaltic arrays, FPGAs and SIMD computers

4. Sequence Motifs: Representation and Identification
   a. Consensus sequences
   b. Weight Matrices
      i. WMM
      ii. WAM
      iii. Maximal Dependence Decomposition
   c. Artificial Neural Networks
      i. Introduction to Neural Networks
      ii. Training feed-forward back-propagation networks
      iii. Scoring sequences with neural networks
   d. Profile HMMs
      i. HMM Introduction
      ii. Training Profile HMMs
      iii. Scoring sequences with HMM profiles
   e. Finding motifs in unaligned sequences
      i. Iterative alignment procedures
      ii. Gibbs sampler
5. Sequence Assembly
   a. Sequencing strategies
      i. HGP: Mapping and sequencing
      ii. Celera: Whole genome shotgun sequencing
      iii. Pros and cons
   b. Algorithms
      i. Theoretical aspects
         1. Shortest common superstring is *NP*-complete
      ii. Greedy approach
      iii. Overlap-layout-consensus
      iv. Eulerian Paths
   c. The finer points
      i. Impact of repetitive sequences
      ii. Use of clone-end sequencing
6. Gene Prediction
   a. Review of gene structure
   b. Gene features
      i. Motifs
         1. Splice sites
         2. Promoters
         3. PolyA signals
      ii. Content measures
         1. ORFs and codon usage, reading frame consistency
         2. Local compositional complexity

3. CpG suppression
4. Repetitive sequences

c. Exon assembly algorithms
   i. Rule-based systems
   ii. Dynamic programming
   iii. HMMs

d. Gene Prediction by spliced alignment

7. Molecular evolution
   a. Genetic drift, neutral mutations and molecular clocks
   b. Phylogenetic trees
      i. Unrooted trees
      ii. Rooted trees
      iii. Maximum parsimony
      iv. Maximum likelihood
      v. Methods of invariants

Sample Homework Problems

1. Dynamic programming
   a. Using a spreadsheet calculator program (*e.g.* Microsoft Excel) to create the table, implement a local dynamic programming alignment algorithm using a gap penalty of $1n+4$, a mismatch penalty of -1 (match scores +1) and the following sequences as an example:
      i. ACACGCGGGGCAATGAGGTCATC
      ii. ACGCCCGGCAATGAGGCACTCATC
   b. Using the table created above, show the optimum traceback and the resulting alignment
   c. (extra credit) Can you find a way to automatically color the cells of the optimum traceback?
2. Motif analysis
   a. Given a list of 50,000 aligned sites (*e.g.* donor splice sites), write a program to create a Weight Matrix Model scoring function.
   b. Use this program to scan a given gene sequence for splice sites. Calculate the specificity and sensitivity of your scoring scheme.
   c. (Extra Credit) Using the same sites, create a Windowed Weight Array Model as described by Zhang and Marr (CABIOS 9(5): 499-509 (1993)).
   d. (Extra Credit) Use the WWAM program to score the same sequence used in b). Does your performance improve?