

A NEW APPROACH FOR GENE ANNOTATION USING UNAMBIGUOUS SEQUENCE JOINTS

Alexandre Tchourbanov*, Daniel Quest*, Hesham Ali*, Mark Pauley* and Robert B. Norgren[†]

* Department of Computer Science, College of Information Science and Technology, University of Nebraska at Omaha, Omaha, NE 68182-0116

[†] Department of Genetics, Cell Biology and Anatomy, University of Nebraska Medical Center, Omaha, NE 68198-6395

Abstract

We address the problem of accurate and automatic gene finding and annotation. This paper describes a new cDNA/DNA homology gene annotation algorithm that combines the results of both BLASTN search and spliced alignment. Compared to other programs currently in use, annotation quality is increased through the unambiguous joining of genomic DNA sequences and use of spliced alignment. We also address gene annotation with both non-canonical splice sites and short exons. Our approach has been tested on the **Genie** learning subset and the human RefSeqs. This approach exhibited sensitivity and specificity of 97% with the Genie learning subset.

Introduction

Gene annotation is an essential tool for the modern biologist; there is thus the need for fast and accurate annotation tools. The most widely used and precise methods of annotation are based on DNA/DNA, DNA/cDNA and DNA/Protein homology [MSSR02]. The **GIGOgene 1.0** program was created to address the problem of correct gene annotation. We combined several sources of information, such as alignment, sensors, unambiguous HSPs allocation and all-pairs-longest-path sequences connection, to improve gene annotation quality.

Our program complements existing software [MSSR02, FHZ⁺98] with additional features:

1. Parses BLASTN output to find the best match(es);
2. Finds exons smaller than eleven nucleotides;
3. Handles the non-canonical AT/AC splice rule;
4. Considers the optimal junction of HSPs plus the spliced alignment with additional precision and noise-tolerance;

GIGOgene is open source and can be customized by individual users.

Algorithmic steps

We provide a description of the detailed process for gene annotation. The process is divided into 8 steps as explained below:

1. Run BLASTN, with the low-complexity BLASTN filter turned off, querying curated RefSeq mRNA sequences against a DNA database extracted from NCBI HTG and Primate (PRI) flatfiles.
2. Parse the BLASTN output and identify sequences that score above a certain threshold. These sequences contain potential exons.
3. Exclude HSPs that are subsegments included in bigger matching segments; a certain noise tolerance is required.
4. Disambiguate the ordered sets of HSPs. Each sequence may result in several unambiguous ordered sets of HSPs.
5. Build an interval graph of overlapping unambiguous ordered sets of HSPs. Special attention is paid to the continuity of mRNA segments, following the biological rules mentioned below. Edges between HSP sets coming from the same sequence are not allowed.
6. Compact the interval graph. This way we identify the biggest composite sequence containing the maximum number of possible exons. For this step, we run the all-pairs-longest-path algorithm.
7. Use spliced alignment to identify possible intron boundaries in the DNA sequence. In order to save running time, we use *anchors* - short nucleotide sequences from mRNA and DNA that should contain exon/intron boundary fragments with donor/acceptor signals. A normal anchor does not have mismatches in *M* state. Once we have a mismatch, it may mean a short exon is present. If we need to, we can expand the anchor and rerun it with two full exons with an intron between to identify possible short exons.
8. Extract exonic structure from the ordered set of introns and write exon/intron structure into the database in FASTA format.

FCGBP gene annotation example

The problem of disambiguating the ordered set of HSPs arises during the assembly of the gene from several DNA sequences. Parts of a gene, or the entire gene itself, may have been replicated during evolution. This can result in ambiguity during the gene assembly process.

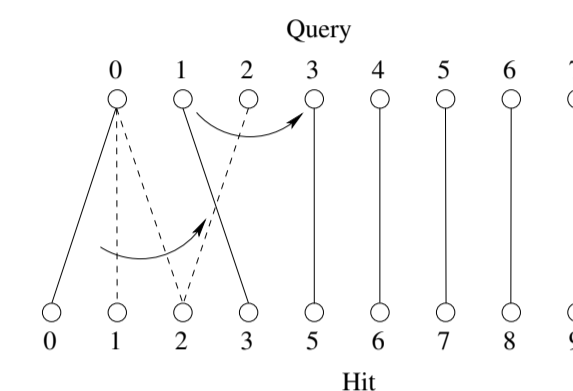


FIGURE 1: Idea behind disambiguating algorithm

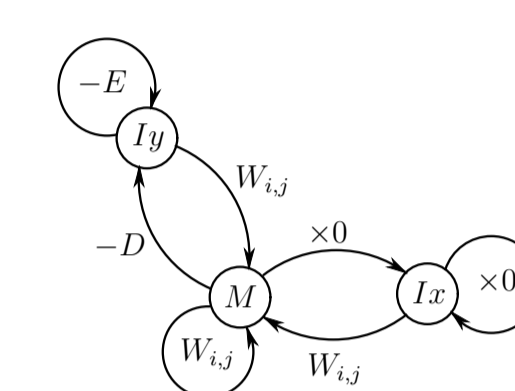


FIGURE 2: State diagram of the disambiguating algorithm

Below we consider our test case: Fc fragment of IgG binding protein (FCGBP) transcript NM_003890 aligned to genomic clones AC00784.1, AC011536.6 and AC006950.1.

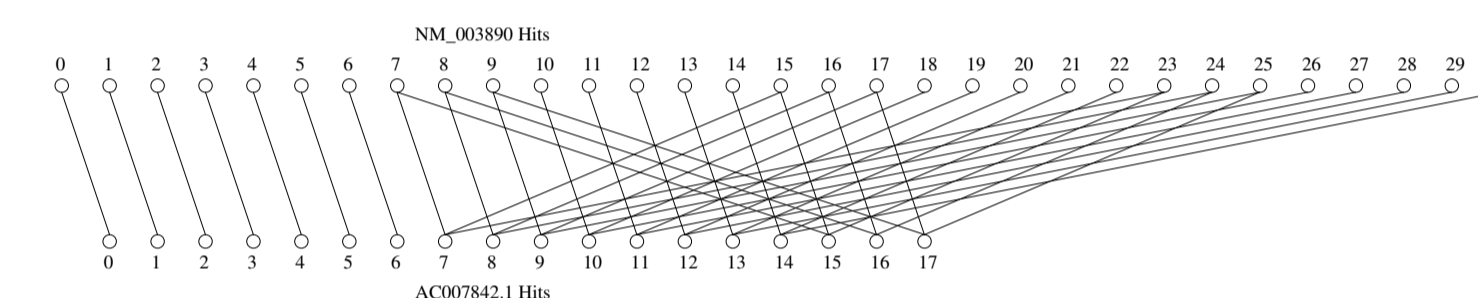


FIGURE 3.1: Ambiguity for the BLASTN alignment of NM_003890 with AC007842.1

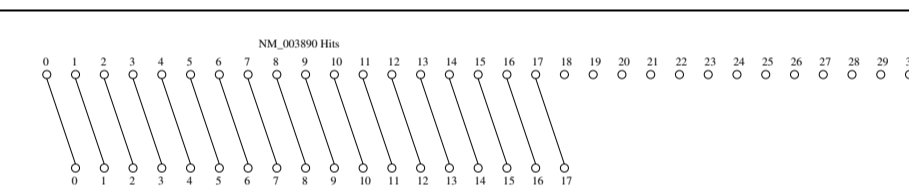


FIGURE 3.2: First non-ambiguous ordered set of HSPs from alignment of NM_003890 with AC007842.1

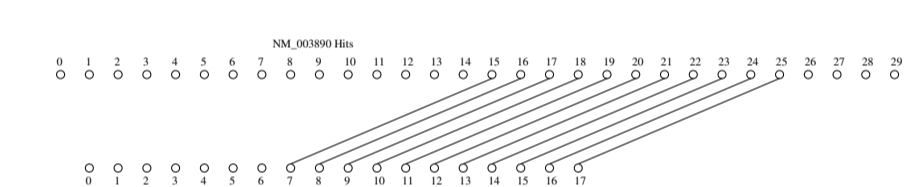


FIGURE 3.3: Second non-ambiguous ordered set of HSPs from alignment of NM_003890 with AC007842.1

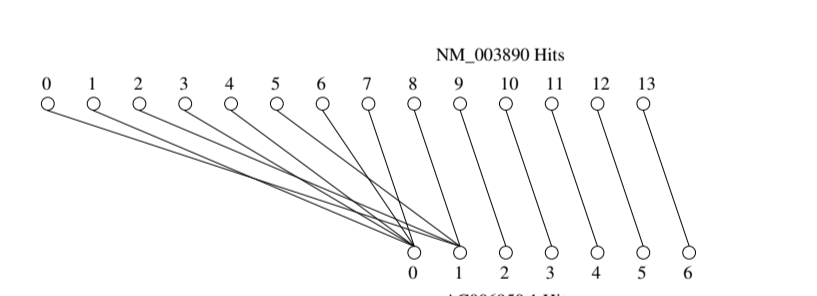


FIGURE 4.1: Ambiguous BLASTN alignment of NM_003890 with AC006950.1

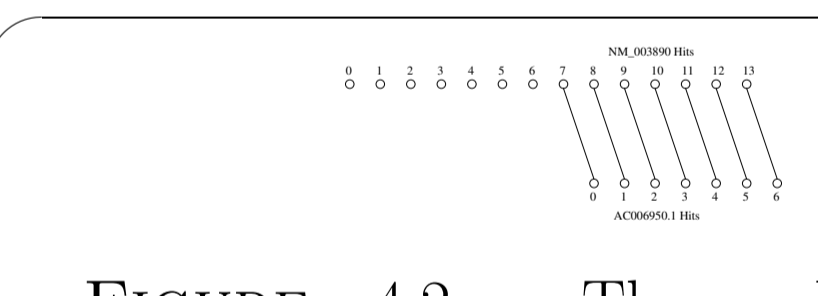


FIGURE 4.2: The only non-ambiguous ordered set of HSPs from alignment of NM_003890 with AC006950.1

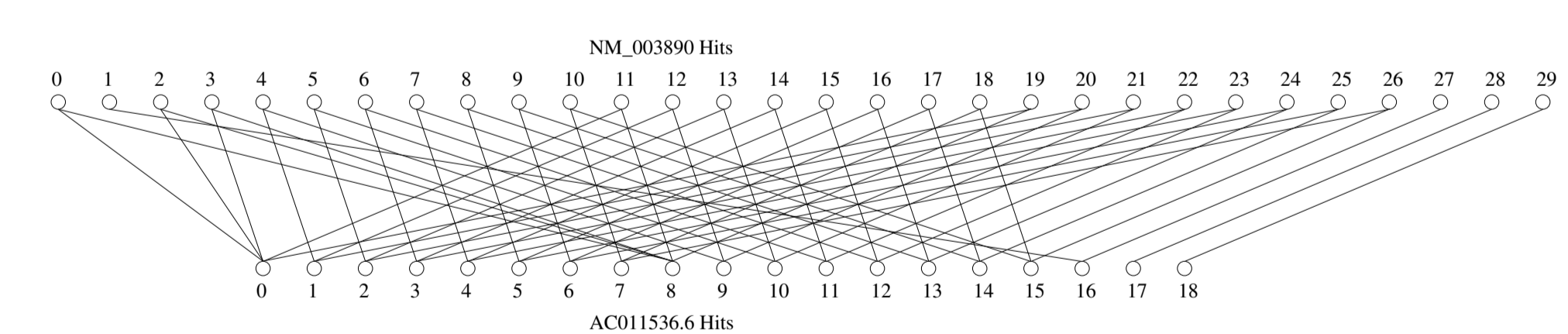


FIGURE 5.1: Ambiguity for the BLASTN alignment of NM_003890 matching AC011536.6

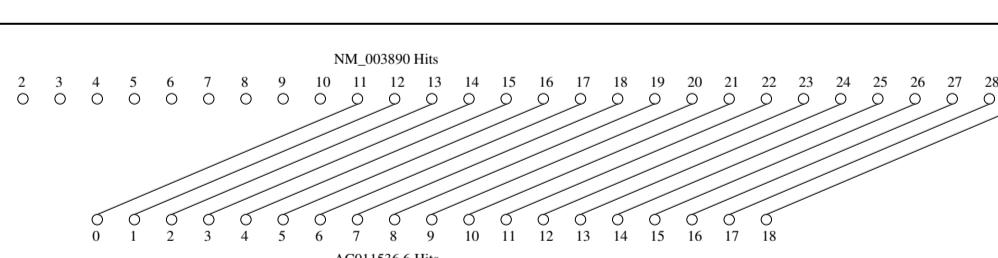


FIGURE 5.2: First non-ambiguous ordered set of HSPs from alignment of NM_003890 with AC011536.6

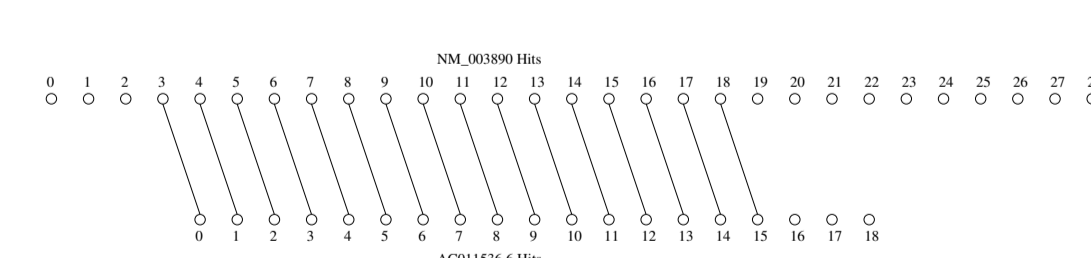


FIGURE 5.3: Second non-ambiguous ordered set of HSPs from alignment of NM_003890 with AC011536.6

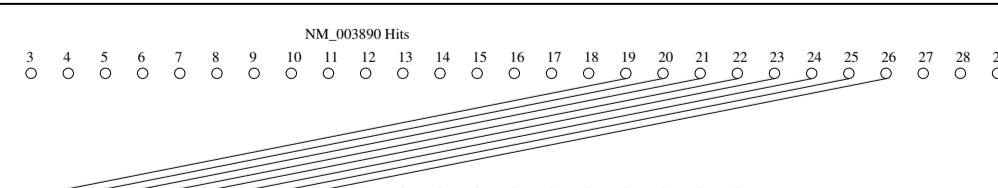


FIGURE 5.4: Third non-ambiguous ordered set of HSPs from alignment of NM_003890 with AC011536.6

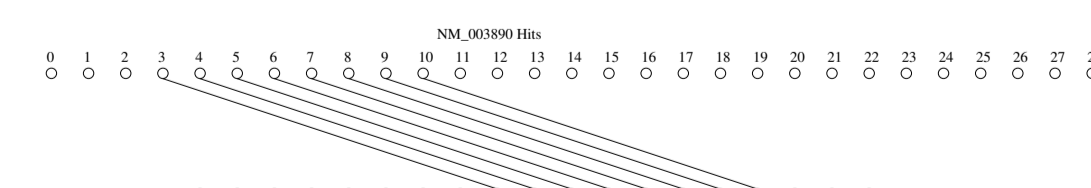


FIGURE 5.5: Fourth non-ambiguous ordered set of HSPs from alignment of NM_003890 with AC011536.6

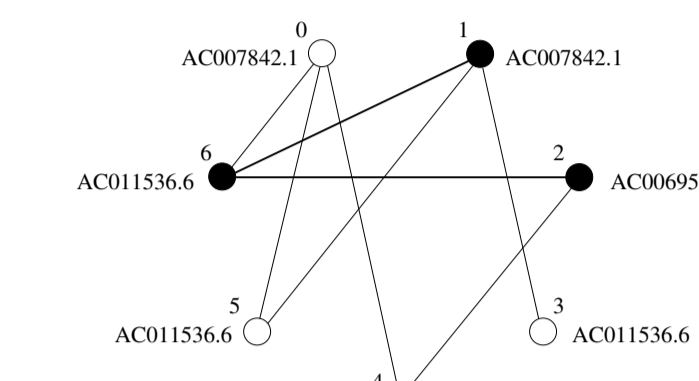


FIGURE 6: Interval graph to connect unambiguous FCGBP gene fragments

We used the affine gap penalty Needleman-Wunsch algorithm for spliced alignment. The spliced alignment algorithm runs in a linear memory following the recurrence for each step, as shown in the figures below.

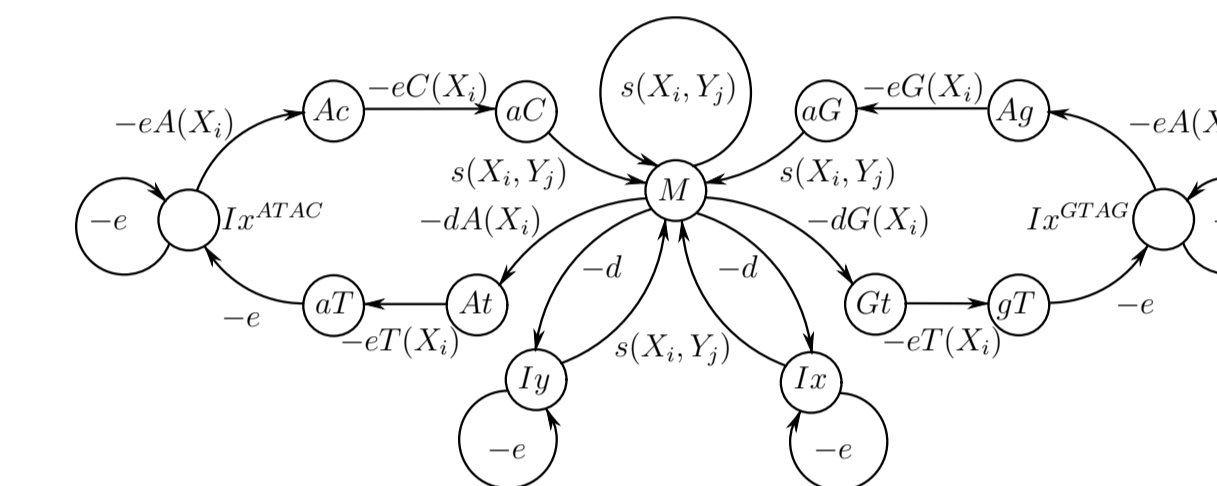


FIGURE 7: State diagram of the spliced alignment algorithm

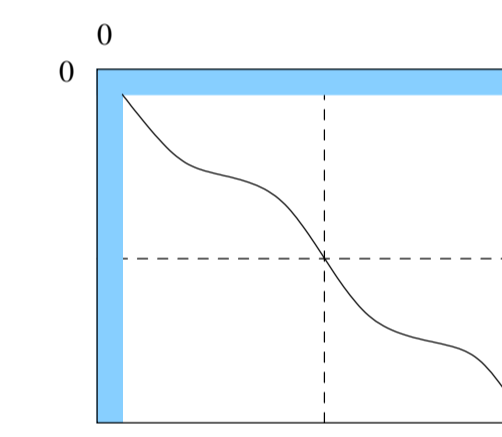


FIGURE 8.1: Find the split point

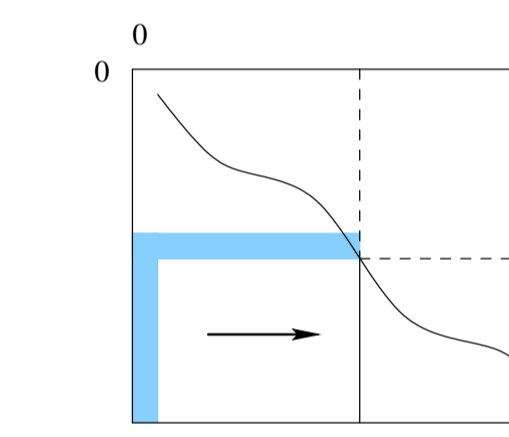


FIGURE 8.2: Restore the context for the recursive call

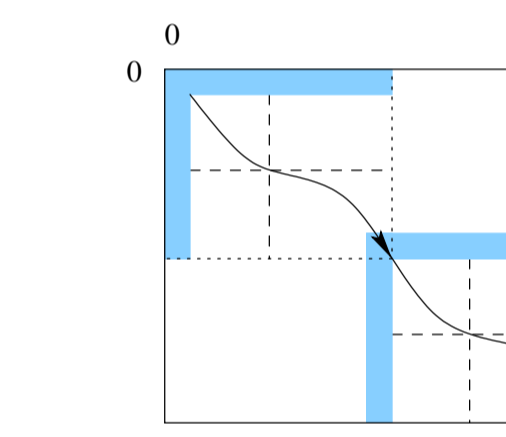


FIGURE 8.3: Make recursive call for the sub-sequences

Experimental results

The program was tested on the first 200 records in the **GENIE** gene finding data set available at <http://www.fruitfly.org/sequence/human-datasets.html/>; for this test, both specificity and sensitivity were 97%. In the test run, our program detected all the internal exon boundaries correctly.

Conclusion

GIGOgene 1.0:

1. incorporates biological rules;
2. uses three dynamic programming routines at different stages;
3. was tested on the **GENIE** data subset for annotation quality and run against the whole human genome to produce an exon/intron database;
4. has been used to compile a database of splice sites as a training set for other projects.
5. is available at <http://bioinformatics.ist.unomaha.edu/~achurban/>.

References

- [FHZ⁺98] Liliana Florea, George Hartzell, Zheng Zhang, Gerald M. Rubin, and Webb Miller. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Research*, 8:967–974, 1998.
- [MSSR02] Catherine Mathé, Marie-France Sagot, Thomas Schiex, and Pierre Rouzé. Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids research*, 30:4103–4117, 2002.