Natural Language Processing for Bioinformatics: The Time is Ripe

Jeffrey T. Chang is a Ph.D. candidate in the Russ Altman lab in the Biomedical Informatics program at Stanford University. His work is focused on applying natural language processing techniques to biological problems ranging from pharmaco-genomics to sequence homology searches. Jeffrey has helped teach informatics classes at Stanford and has also taught a Python Programming Language tutorial at the Pacific Symposium on Biocomputing.

Soumya Raychaudhuri is also a M.D./Ph.D. candidate in Russ Altman's lab. His biological interests range from structural biology to interpretation of microarray data. His methodological approaches encompass machine learning methods and natural language processing. Soumya has helped teach informatics classes at Stanford.

Presentation: The tutorial will be a practical introduction on applying natural language processing (NLP) to biological research. It will focus on basic methodology used in current efforts published in the literature and in our lab. At the end of the tutorial, attendees will have been exposed to a broad range of work in the field and understand the technologies applied to solve those research questions.

Audience: This tutorial will be targeted towards a general audience that has a basic understanding of biology and probability. We will assume no prior knowledge about natural language processing or machine learning algorithms.

Length: The tutorial will require one half day.

Published literature is a fundamental medium for scientific communication. In recent times, much literature has become available electronically: most biomedical abstracts are available through PubMed while many full text articles are available from electronic publishers. The vast amount of literature produced in biomedicine presents challenges and opportunities for scientists. First, computational methods need to be developed to help scientists analyze and summarize bodies of literature. Second, the literature can now be viewed as a source of knowledge that may be incorporated into bioinformatics algorithms using natural language processing (NLP) techniques. NLP is becoming a tool that can help automate tasks.

In this tutorial we will introduce the natural language processing, drawing from current research in the biological domain:

1. Preparing the text for processing.
2. Summarizing a group of documents.
3. Identifying specific facts from individual documents or sentences.
4. Incorporating literature to refine biological algorithms.

I. Text Handling (Pre-processing)

The first part of the tutorial will focus on pre-processing methods that prepare text for further analysis. *Tokenization* is the task of segmenting a document into words. The simplest tokenization strategy is to partition text on white space and punctuation. More sophisticated approaches can treat synonyms or compound noun phrases as a single concept. For example, the application may benefit from treating "G protein-coupled receptor" as a single concept rather than four separate words. Such noun phrases are ubiquitous in biology. Next, a *stemming* algorithm may be used to find variations of the same word. Suffixes are removed, e.g. "kinases" may be changed to "kinase." Stemming transforms variations of the

same word into a single one, reducing vocabulary size. Lastly, s*top word removal* algorithms remove words that add little information.

After tokens are identified, *tagging* is done. Tagging algorithms annotate tokens by part of speech (syntactic) or class (semantic). The former identifies nouns, verbs, adjectives, etc. and the latter assigns pre-defined classes, such as gene names, protein names, or drug names [Baclawski, Fukuda, Proux 1998, Rindflesch, Yoshida].

Finally, we will discuss representations of text. Once a document has been appropriately tokenized and tagged, it is necessary to represent the text in a structure that facilitates computations. Because of its simplicity, we will introduce vector space models, which represent documents as vectors of word counts. This reduces our collection of documents into a matrix on which standard statistical techniques can be applied [Stephens].

II. Corpus Analysis

In the second part, we will introduce methods used to analyze bodies of text, with special emphasis on machine learning. For a set of text documents, it is often useful to summarize its contents. This is analogous to searching for words that best describe those documents. Another useful operation is clustering the documents, which categorizes them into sub-groups.

To perform these tasks, machine learning methods are applied. These can be categorized as either supervised, those that require a training set, and unsupervised, otherwise. One supervised machine learning method is *naïve bayes*; it is simple, effective, and easy to implement. Its performance rivals more sophisticated methods such as *maximum entropy methods* and *nearest neighbor methods*. These algorithms are applied towards categorizing documents according to pre-defined groups [Stapley, Tamames, Usuzaka].

Unsupervised methods encompass clustering and dimensional reduction strategies. We will introduce simple clustering strategies such as *self organizing maps* and *K-means clustering*. *Singular Value Decomposition (or Principal Component Analysis)* is a method commonly employed to reduce the high dimensionality of document data. Clustering can be used to discover latent patterns in the literature [Iliopoulos].

A few studies summarize large corpora of literature. The manual equivalent is to read all the literature and produce keywords that describe it. This is typically done using a statistical method that identify words that are most descriptive for some text. This has been applied towards describing genes from a microarray experiment [Shatkay], automatic annotation of sequence families [Andrade 1998], and summarizing the results of document clusters [Iliopoulos].

III. Information Extraction

Information Extraction methods are used to identify relations between types of objects. In biology, information extraction has been used to find gene-gene interactions, protein-protein interactions [Craven, Hishiki, Humphreys, Ng, Park, Proux 2000, Rindflesch, Sekimizu, Stephens, Thomas, Wong, Yakashuji]. The goal of these methods is to transfer knowledge from unstructured data, the literature, to a structured form that can be included in a database or knowledge base. The simplest methods include looking for co-occuring gene names within abstracts while more sophisticated methods identify the genes that frequently co-occur within documents and then examine sentences describing both of the genes to discover the relationship between them.

IV. Enhancing Bioinformatics Algorithms

This section of the tutorial will be less pedagogical than the previous ones; while the literature will be reviewed, the emphasis will be on discussion.

Two groups have used NLP techniques to refine homology search [Chang, MacCallum]. Both found that inclusion of document similarity along with sequence similarity when conducting a PSI-BLAST database query yields modest improvement in performance. Other potential applications for algorithm refinement using NLP approaches include validation of gene expression clusters using literature information and automated gene annotation [Andrade 1998].

Annotated References

Andrade MA, Bork P. (2000) Automated extraction of information in molecular biology. FEBS Lett. 476:12-7.

Review of the current work on information extraction in biology.

Andrade MA, Valencia A. (1998) Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. Bioinformatics. 14:600-7.

Uses z-score to find words most descriptive of FSSP families. Application toward automatic annotation.

Baclawski K, Cigna J, Kokar MM, Mager P, Indurkhya B. (2000) Knowledge representation and indexing using the unified medical language system. Pac Symp Biocomput. 493-504.

Tags documents based on UMLS semantic types for visualization.

Chang JT, Raychaudhuri S, Altman RB. (2001) Including biological literature improves homology search. Pac Symp Biocomput.

Includes text similarity score to help decide which sequences to include in each iteration of a PSI-BLAST search.

Craven M, Kumlien J. (1999) Constructing biological knowledge bases by extracting information from text sources. Proc Int Conf Intell Syst Mol Biol. 77-86.

Uses naïve bayes for extracting information on subcellular localization, cell localization, tissue localization, associated diseases, and drug interactions.

Fukuda K, Tamura A, Tsunoda T, Takagi T. (1998) Toward information extraction: identifying protein names from biological papers. Pac Symp Biocomput. 707-18.

Rule-based system for identifying protein names from text. Uses context, parts of speech, keywords, and clues from the strings.

Hishiki T, Collier N, Nobata C, Okazaki-Ohta T, Ogata N, Sekimizu T, Steiner R, Park HS, Tsujii J. (1998) Developing NLP Tools for Genome Informatics: An Information Extraction Perspective. Genome Inform Ser Workshop Genome Inform. 9:81-90.

Presents an information extraction system useful for biology.

Humphreys K, Demetriou G, Gaizauskas R. (2000) Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. Pac Symp Biocomput. 505-16.

Evaluates a general purpose information extraction in the biomedical domain. Uses rules for named entity recognition and template filling.

Iliopoulos I, Enright AJ, Ouzounis CA. (2001) Textquest: Document custering of medline abstracts for concept discovery in molecular biology. Pac Symp Biocomput.

Clustered documents of various organisms using k-means. Summarized each cluster using a log-odds score.

MacCallum RM, Kelley LA, Sternberg MJ. (2000) SAWTED: structure assignment with text description -- enhanced detection of remote homologues with automated SWISS-PROT annotation comparisons. Bioinformatics. 16:125-9.

Combined PSI-BLAST scores with vector-cosine text scores to improve the accuracy of homology search results. Applied to protein structure prediction.

Ng SK, Wong M. (1999) Toward Routine Automatic Pathway Discovery from On-line Scientific Text Abstracts. Genome Inform Ser Workshop Genome Inform. 10:104-112.

Uses the BioNLP toolkit, a rule-based system, to extract protein-protein interactions. Visualization of interaction pathways.

Park JC, Kim HS, Kim JJ. Bidirectional incremental parsing for automatic pathway identification with combinatory categorical grammar. Pac Symp Biocomput.

Uses a combinatory categorical grammar parser to extract protein-protein interactions.

Proux D, Rechenmann F, Julliard L. (2000) A pragmatic information extraction strategy for gathering data on genetic interactions ISMB. 8:279-85.

Combines various linguistics technologies (parts of speech tagging, shallow syntactic parsing) to identify gene interactions.

Proux D, Rechenmann F, Julliard L, Pillet V, Jacq B. (1998) Detecting Gene Symbols and Names in Biological Texts: A First Step toward Pertinent Information Extraction. Genome Inform Ser Workshop Genome Inform. 9:72-80.

Presents a method for detecting Drosophila gene names from text. Uses syntactic information as well as clues in the context for candidate gene names.

Rindflesch TC, Tanabe L, Weinstein JN, Hunter L. (2000) EDGAR: extraction of drugs, genes and relations from the biomedical literature. Pac Symp Biocomput. 517-28.

Uses the semantic information from the Unified Medical Language System to identify relations between drugs and genes.

Sekimizu T, Park HS, Tsujii J. (1998) Identifying the Interaction between Gene s and Gene Products Based on Frequently Seen Verbs in Medline Abstracts. Genome Inform Ser Workshop Genome Inform. 9:62-71.

Presents a system for detecting interactions between genes. First, looks for verbs that are indicative of interactions, and then looks for nouns around that verb based on a syntactic parse.

Shatkay H, Edwards S, Wilbur WJ, Boguski M. (2000) Genes, themes and microarrays: using information retrieval for large-scale gene analysis ISMB. 8:317-28.

Uses a vector cosine model to find similar genes in a microarray experiment. Then, finds terms that describe the groups of genes.

Stapley BJ, Benoit G. (2000) Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in Medline abstracts. Pac Symp Biocomput. 529-40.

Finds genes with similar function based on co-occurrences within MEDLINE abstracts. Calculates the similarity between pairs of genes from *Saccharomyces cerevisiae* and provides a Java application to visualize the results.

Stephens M, Palakal M, Mukhopadhyay S, Raje R. Mostafa J. (2001). Detecting gene relations from medline abstracts. Pac Symp Biocomput.

Uses a vector cosine scoring model to discover genes that are related.

Tamames J, Ouzounis C, Casari G, Sander C, Valencia A. (1998) EUCLID: automatic classification of proteins in functional classes by their database annotations. Bioinformatics. 14:542-3.

Classifies SwissProt sequences into broad functional classes Energy, Information, and Communication based on manually assigned keywords.

Thomas J, Milward D, Ouzounis C, Pulman S, Carroll M. (2000) Automatic extraction of protein interactions from scientific abstracts. Pac Symp Biocomput. 541-52.
Uses SRI's Highlight system for extracting protein interactions. Developed templates for finding interactions.

Usuzaka Si, Sim KL, Tanaka M, Matsuno H, Miyano S. (1998). A Machine Learning Approach to Reducing the Work of Experts in Article Selection from Database: A Case Study for Regulatory Relations of S. cerevisiae Genes in MEDLINE. Genome Inform Ser Workshop Genome Inform. 9:91-101.
Uses a supervised machine learning algorithm to find articles of interest to the user.

Wong L. (2001). PIES, a protein interaction extraction system. Pac Symp Biocomput.
Describes system that extractions protein interaction pathways from the literature and supports manipulations of the pathways.

Yakashuji A, Tateisi Y, Miyao Y, Tsujii J. (2001) Event extraction from biomedical papers using a full parser. Pac Symp Biocomput.
Evaluates the difficulties encountered when doing a full syntactic parse on biomedical literature.

Yoshida M, Fukuda K, Takagi T. (2000 Feb) PNAD-CSS: a workbench for constructing a protein name abbreviation dictionary. Bioinformatics. 16:169-75.
Defines a set of rules for recognizing the abbreviations of protein names from the literature.