Fine-Grained Protein Mutation Extraction from Biological Literature

Rui Wang*[1], Shirley W. I. Siu*[2], Rainer A. Böckmann[2]

[1]Computational Linguistics
Saarland University
Saarbrücken, Germany
rwang@coli.uni-sb.de

[2]Theoretical & Computational Membrane Biology
Saarland University
Saarbrücken, Germany
{siu,rainer}@bioinformatik.uni-saarland.de

Abstract—**Automatic extraction of experimental data on protein mutants from large volumes of biological texts can help building corresponding databases to facilitate research in relevant studies. Mutation extraction cannot be fully solved by the surface pattern matching but requires linguistic analysis of the plain text. Based on the existing regular expression method, we improved the mutation extraction by applying the dependency parsing technique from natural language processing (NLP). Furthermore, we extract valuable data about experimental measurements from the texts and relate them to the identified mutations. Our method was evaluated on MedLine abstracts. The results show great potential for future exploration.**

Keywords-natural language processing; mutation extraction; text mining; bioinformatics

I. INTRODUCTION

Experimental protein mutational analysis is used to test residues for their role in the function, stability, and folding of proteins and protein-protein complexes. Mutants created by substitution, deletion, or insertion of amino acid(s) in the wild-type protein using site-directed mutagenesis or directed evolution, are subsequently investigated using various experimental methods (thermodynamic analyses, x-ray crystallography, NMR, etc.) and the results are compared to outcomes from the wild-type.

Experimental results from mutagenesis studies are especially important for the tasks of protein engineering, for disease-related research, as well as for *in-silico* protein mutational scans. Mutations for the protein of interest are usually collected manually from the published literature by the individual laboratories. Mass collection of protein mutation data exists, but is rarely available mainly due to the time-consuming and laborious nature of such tasks. One exceptional effort of these databases is the Protein Mutant Database (PMD) [1]. To date, the PMD contains a collection of 218,873 mutants extracted from 45,239 publications over the past 30 years. Each record provides, besides the mutation, important secondary information such as the protein function, structure, sequence, stability, cross-reference, etc.

Hand-curated data are highly accurate but the progress is relatively slow, which is difficult to cope with the fast growing number of publications. With the popularity of the Internet, recent experimental results are rapidly accessible. Electronic availability of publications makes automatic processing and information extraction feasible.

The main challenges of applying natural language processing (NLP) techniques in the biological domain literatures (BioNLP) come from both the biological and the NLP perspectives. On the one hand, biological research would greatly benefit by a brief and precise summary of published experimental data; on the other hand, scientific publications use a very diverse terminology and many symbols, which is not a trivial task for the current NLP techniques. Examples of such applications include protein-protein interaction extraction [2], building of metabolic pathways [3], gene ontology construction [4], mutation extraction [5, 6], etc.

According to our close communication between researchers from both areas, we propose a fine-grained approach to address an important task in the BioNLP area, **protein mutation extraction (ProtME)**. This can be viewed as an example for relevant tasks, such as extraction of protein-protein interactions, membrane structural properties, etc.

The rest of the paper is organized as follows: In the next section, some related works are introduced; Section 3 describes the information needed by researchers in the biological field; Section 4 elaborates on our approach of automatic extracting of information via NLP techniques; the experimental results will be shown in Section 5, followed by a discussion; the last section concludes the paper and also points out some future directions to work on.

978-0-7695-3559-3/09 $25.00 © 2009 IEEE
DOI 10.1109/ICECT.2009.10

II. RELATED WORK

Tools for mutation data extraction such as Mutation Finder [6] and Mutation Miner [5] have recently been developed. The former focuses on the mutation extraction, while the latter also discovers relations between proteins and mutations. However, for mutational studies, not only the proteins and the mutations are the subjects of concern, but also the experimentally relevant information. In particular, the effects of a mutation e.g. on protein stability, are of upmost interest. Databases containing all this information will be of significant value to both experimental and theoretical research in biosciences. To our best knowledge, application of NLP for extracting such information has not been addressed so far.

Consequently, our work in this context is to establish a framework for the automatic extraction process in order to construct a database of mutations. Built on the currently available tools, we focus on applying the dependency parsing to improve mutation recognition, protein-mutation relation recognition, and extracting important experimental conditions and observations from the literature. Furthermore, once the automatic extraction procedure is established, the task becomes very similar to the template-based information extraction (IE) in the conventional NLP field.

III. EXTRACTION TEMPLATE

Unlike in the traditional NLP, the **name-entities** (NE) of relevance in the biological domain consist of names of the *genes*, *proteins*, *organisms*, etc. These may be involved in certain **interaction events** (like *translation*, *binding*, and *mutation*) in which specific **relations** between them (such as *inhibit*, *activate*, and *increase stability*) are established.

Literature about mutational studies often provides information about (1) the proteins being studied; (2) the mutations performed; (3) under which conditions the experiment was performed; and (4) results or effects of the mutations. An example below extracted from a MedLine [13] abstract (PMID 10956001) describes a mutation experiment:

Ex1: *CcP (E290K) has a charge-reversal mutation in the tight-binding domain, which should weaken binding, and it weakens the 1:1 complex; K_I decreases 20-fold at 18 mM ionic strength.*

This example includes all four kinds of information: (1) *CcP* is an abbreviation for the protein *Cytochrome c peroxidase*, which forms a complex with another protein *Cc*; (2) The mutation *E290K* in *CcP* was performed (3) under the condition of *18mM ionic strength*; and (4) as a result, a decrease in K_I value was observed.

We detail the properties of the four kinds of information as follows:

Protein. Mutation experiments usually involve one or more proteins, in which mutations are selectively performed. And it is often mentioned in the title or in the beginning of the manuscript to set the focus of the study. Proteins without mutations are referred to as the wild-types (WTs), and mutated proteins are called the mutants.

Mutation. The mutation of a protein is usually expressed in the literature in a conventional way, e.g. "*mutating residue Arginine at position 23 to residue Alanine*" is typically written in the short form ARG23ALA or R23A. However, the identification of a mutation that is embedded in the context turns out to be a nontrivial task. For instance,

Ex2: *His-230* ***and*** *His-309 were mutated to phenylalanine.*

Ex3: *Asn-Gly pairs were changed into Leu (Asn244, Asn255, Asn437) or Ala (Asn276)*

In both cases, even a complicated pattern matching such as developed in [6] cannot recognize these mutations mentioned in the texts. The relation between the WT residue and the mutated residue can only be captured via the verbs *mutated* and *changed*. Therefore, linguistic analysis of the text is required for recognizing such mutations.

Condition and Result. Both of them are similar in the sense that they are often expressed in manuscripts as quantitative values. In addition, the resulting measurements may infer a qualitative relationship between the protein and the mutation, such as *weaken the binding complex* in Ex1.

To summarize, the extraction template can be described as follows,

```
MutationExperiment
<List<protein,List<mutation>>,List<condition>
 List<result>>
```

Thus, the task now is to recognize, extract, and reformulate the data from the source text and fill in the template with the obtained information. A mutation study might include a number of mutations, and each of these individual experiments is considered as one *MutationExperiment* object. Elements in the same object are therefore associated through retrieving the binary relations between them (Figure 1).

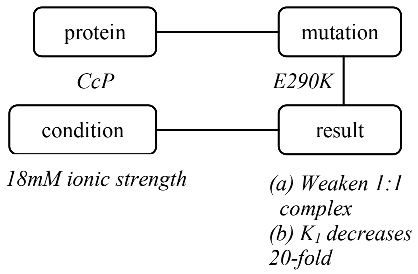

Figure 1. Binary relation for the MutationExperiment object created from Ex1.

IV. Extraction Approach

In order to fill in the template mentioned above, we need to use several linguistic preprocessing techniques and also extraction rules (or patterns) to obtain relevant information from the plain text.

A. Linguistic Preprocessing

We preprocess the raw texts using several linguistic modules, including the tokenizer, the Part-Of-Speech (POS) tagger, the Named-Entity (NE) recognizer, and the dependency parser.

Tokenization. In fact, the first step is not as trivial as it looks like. For example, if we have a protein like *(5S,6E,8Z,11Z,14Z)-5-hydroperoxy-6,8,11,14-eicos atetraenoic acid (5S-HpETE)*, it is difficult to decide whether to take it as a whole or to tokenize it into several parts, and into how many parts. The current version of our system uses ABNER [7].

POS Tagging. This is almost the same as the conventional POS tagging in NLP, but the statistical model needs to be trained on the annotated in-domain data. We utilized LingPipe [8], which is a Hidden Markov Model (HMM) trained on the GENIA corpus[1].

Dependency Parsing. Dependency parsing has not been widely used in the BioNLP research, due to the efficiency problem when a large set of data needs to be processed. However, in our work, we use the keyword-based search as a filter to restrict the range of applying the parsing technique. In other words, we use the coarse-grained search to zoom into the “interesting” parts and then apply a fine-grained processing (i.e. dependency parsing) to obtain precise results. The graph below shows an example of a dependency parse tree of a sentence after applying the MST Parser [9].

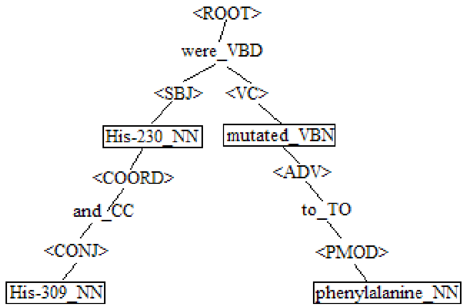

Figure 2. The dependency tree of Ex2.

B. Information Extraction

Useful information to be extracted can be roughly classified into two categories: object recognition (e.g. entities, measurements) and relation extraction. For the former case, according to the template introduced in Section 3, we need to extract proteins, mutation expressions, experimental conditions, and results; and for the latter case, relations between them should be identified.

Object Recognition. As we mentioned before, NE types in the biological domains are mainly gene names, protein names, etc. In our work, we focus on mutation expressions, experimental conditions, and experimental results.

For the mutation expression extraction, we use MutationFinder as the baseline system. After some error analysis of the preliminary results, we 1) further improved the regular expression used in MutationFinder and 2) incorporated linguistic information as well. In detail, the original MutationFinder searches for mutation expressions with all three fields, wild-type residue, mutation position, and mutated residue, while we additionally allowed some missing fields, like an isolated residue or a residue with a mutation position, to be extracted. The purpose is to use these residues as starting points together with the dependency parse tree to identify mutations expressed in the natural language sentences.

Ex2 is a good example to show possible further improvements of MutationFinder. In fact, the pattern, *WRESPPOS was mutated to MRES*[2], is included in the system, but the rigid surface string cannot capture various linguistic expressions with the same (or similar) meaning, like in Ex2. Therefore, we make use of the identified residues and mutation positions as entries to the dependency tree (as shown in Figure 2) and traverse up to discover the common ancestor node of the residue pair. This common ancestor node usually conveys the semantic relation between these two residues. In case of the verb *be*, we take the adjacent

[1] http://alias-i.com/lingpipe/demos/models/pos-en-bio-genia.HiddenMarkovModel

[2] WRESPPOS represents the wild-type residue with the mutation position; and MRES represents the mutated residue.

verb instead (i.e. *mutated* in Figure 2). The mutation position is either given with one of the residues (as in Figure 2) or as another descendant of the root node (e.g. *mutated … at the position 137*).

Both experimental conditions and experimental results are conveyed via numbers from the NLP perspective. However, there are still some different focused measurements to discover. For the experimental conditions, the *temperature* and *pH* value are the most important and straightforward information to extract. For experimental results, *DeltaG*, *DeltaDeltaG*, *K(cat)*, etc., are the most interesting results for researchers in the biological or biophysical area.

Relation Extraction. Most of the previous work focused on identifying biological entities, while some also concentrated on extracting relations between entities, e.g. the task of extracting protein-protein interactions.

Enlightened by the successful usage of dependency paths in many NLP tasks, e.g. relation extraction [10], question answering [11], recognizing textual entailment [12], we based our extraction approach on the dependency tree. In fact, the algorithm is quite similar to the one we used for improving the mutation extraction, but allows more flexibility. We constructed patterns using the following two basic functions manipulating the dependency tree,

1. Find the dependency path between A and B;
2. Find all the common ancestor verbs for A and B;

where A and B are objects extracted before (see Section 4.2.1).

Usually, the experimental conditions are expressed via numbers only, whereas the experimental results also contain qualitative expressions such as *increase* or *decrease*. Therefore, the mutation-condition relation extraction can be viewed as a binary classification of relations existing between the condition and the mutation; for the latter one, the mutation-result extraction also asks for the label of such relationships, that is, whether the mutation increases or decreases the measured value. For example,

Ex4: *MBP-H213A and H216A TfdA have* ***elevated*** *K(m) values for 2,4-D, and the former* ***showed*** *a* ***decreased*** *k(cat), suggesting these residues may affect substrate binding or catalysis.*

H213A and *H216A* are mutations, and *K(m)* and *k(cat)* are experimental results. We assume that they are all correctly extracted. Based on these objects, the relations between the mutations and the results are expressed via the words in boldface. Roughly speaking, the relations can be classified into two categories, labeled relations (e.g. *evaluated*, *decreased*) and unlabeled relations (e.g. *showed*). In experiments, we evaluated for both cases (see Section 5.1).

V. Experiments

In order to evaluate our approach, we setup two experiments for object recognition and relation extraction, respectively. For the mutation extraction, we used MutationFinder as our baseline system for comparison; and for other object extraction and relation extraction, two researchers from both biology and NLP areas manually checked the results.

The data used in our experiments contained 922 literature abstracts collected from the MedLine [13] bibliographical database which has been previously annotated in PMD with the keyword *mutagenesis*.

A. Experimental Results

We apply precision, relative recall[3], and f-measure as our evaluation metrics for the first experiment, the mutation extraction. Most of the results are verified with the gold standard automatically extracted from the PMD database; and the remainder is evaluated manually. The following table shows the results (MF stands for MutationFinder; MF+ME is our system),

Table 1. Results of mutation extraction

	MF (Baseline)	MF+ME
Precision	94.3	89.4
Relative Recall	88.3	100
F-Score	**91.2**	**94.4**

In total, we extracted 3,818 mutation instances, which outperformed the baseline system in terms of a large increase in recall and a drop in precision. We will present some examples in the next subsection, which also shows possible further improvements.

For the second experiment, since we do not have comparable systems (to our best knowledge), we manually read about 15% of the data to estimate the precision. The detailed metrics are Unlabeled Precision (whether there exist a relationship), Labeled Precision (whether the relationship is correct), and Labeled Accuracy (on top of the extracted relations, the correctly labeled ones). The following table shows the results:

Table 2. Results of relation extraction

	Exp. Conditions	Exp. Results
Unlabeled Precision	**69.6**	**88.5**
Labeled Precision	/	84.6
Labeled Accuracy	/	92.3

We notice that the performance of mutation-condition extraction is much lower than the one of mutation-result extraction. However, due to the relative scarceness of mutation conditions[4] in abstracts, the

[3] Since we did not manually read all the data, we assume the system that has the largest recall as 100 and other systems are relative values to it.

[4] The number of mutation-condition instances is only around 13.5% of the number of mutation-result instances.

performance is prone to errors and probably significantly underestimated.

B. Discussion

After taking a close look at both the gains and losses of our system, we have several interesting examples to show:

In the mutation extraction part, provided with dependency parsing, our system can deal with non-local dependency relations, e.g. in "We mutated Ala137 of T. brucei glycerol kinase into a serine", the mutation *Ala137Ser* can be resolved in our system but cannot be easily captured by regular-expression-based methods. However, when the entry (i.e. a residue or a residue with a mutation position) is mistakenly identified, the extracted mutation will be incorrect as well. In the example, "Mutants of tyrosine hydroxylase with alanine substituted for Phe300", although *tyrosine hydroxylase* is a protein, *tyrosine* itself is a residue, thus *Phe300Tyr* was wrongly reported.

Furthermore, coordination is a big source of errors. Both the parsing error of "Asn-185 of CitS was mutated to Val and Glu-194 was mutated to Gln" and lexical knowledge of *respectively* in "Glu112, Ser113 and Ser115 that … replaced by Pro, Gly and Glu, respectively" lead to errors.

For the relation extraction part, more linguistic knowledge is needed. For instance, "*MBP-H213A and H216A TfdA have elevated K(m) values for 2,4-D, and the former showed a decreased k(cat) ...*", *the former* needs to be resolved and linked to *MBP-H213A* in order to capture its relation to the *k(cat)*. The negations and exceptions expressed by the context are also of great importance. For example, "*The C242S and C69A/C242S enzymes (but* ***not*** *the analogous C242A mutants* ***nor*** *the C69A or C69S mutants) exhibit approximately 10-fold increases in K(m)(HOB) and K(m)(AcAc) …*", *not* and *nor* negate the existence of some mutation-result relations.

VI. Conclusion and Future work

In this paper, we summarized our work on applying NLP techniques to analyze biological publications in order to help researchers in the area to quickly gather useful information. The task of ProtME is automatic extraction of information relevant to mutation experiments. In short, we 1) improved the mutation extraction through combining linguistic processing with a regular-expression-based system, and 2) explored the extraction of relations between the mutations and the experimental measurements.

It is worthwhile to mention that in biological or biochemical literature, much of the relevant information is mentioned in neighboring sentences, and some of them are even spanned over paragraphs. This makes the task even more challenging for the current NLP techniques. We note that our method may as well be applied to other similar data set. Also, extension from abstracts to full manuscripts is straightforward.

Acknowledgments

Rui Wang is supported by the PIRE PhD program. SWIS and RAB are supported by the Graduate School 1276/1 and by DFG BIZ 4/1.

References

[1] T. Kawabata, M. Ota, and K. Nishikawa, "The protein mutant database", Nucleic Acids Res. 27(1), 1999, pp. 355-7.

[2] N. Daraselia, A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin, and I. Mazo, "Extracting human protein interactions from MEDLINE using a full-sentence parser", Bioinformatics 20 (5), pp. 604-611, 2004.

[3] R. Hoffmann, M. Krallinger, E. Andres, J. Tamames, C. Blaschke, and A. Valencia, "Text mining for metabolic pathways, signaling cascades, and protein networks", sci. STKE 2005 (283), pe21, 2005.

[4] N. Daraselia, A. Yuryev, S. Egorov, I. Mazo, and I. Ispolatov, "Automatic extraction of gene ontology annotation and its correlation with clusters in protein networks", BMC Bioinformatics, 23 (243), 2007.

[5] R. Witte and C.J.O. Baker, "Combining biological databases and text mining to support new bioinformatics applications", NLDB 2005, LNCS 3513, pp. 310–321, 2005.

[6] J.G. Caporaso, W.A. Baumgartner Jr, D.A. Randolph, K.B. Cohen, and L. Hunter, "MutationFinder: a high-performance system for extracting point mutation mentions from text", Bioinformatics, 23(14), 2007, pp. 1862-1865.

[7] B. Settles, "ABNER: an open source tool for automatically tagging genes, proteins, and other entity names in text", Bioinformatics, 21(14), 2005, pp. 3191-3192.

[8] Alias-i, LingPipe 3.6.0., http://alias-i.com/lingpipe, 2008.

[9] McDonald, Ryan, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In Proceedings of HLT-EMNLP 2005, pages 523–530, Vancouver, Canada.

[10] Bunescu, R. and Mooney, R. 2006. Subsequence Kernels for Relation Extraction. In Proc. of the 19th Conference on Neural Information Processing Systems.

[11] Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. Natural Language Engineering, 7(4):343–360.

[12] R. Wang and G. Neumann. 2007a. Recognizing Textual Entailment Using a Subsequence Kernel Method. In Proceedings of AAAI-2007, Vancoucer.

[13] C. A. Bachrach and T. Charen, "Selection of MEDLINE contents, the development of its thesaurus, and the indexing Process", Med. Inform., London, Vol. 3, pp. 237–254.