A semantic assistant for mutation mentions in PubMed abstracts.

Jonas B Laurila[1], Alexandre Kouznetsov[1] and Christopher J O Baker*[1]

[1]Department of Computer Science & Applied Statistics, University of New Brunswick, Saint John, New Brunswick, E2L 4L5, Canada.

Email: Jonas B Laurila - j02h9@unb.ca; Alexandre Kouznetsov - alexk@unb.ca; Christopher J O Baker*- bakerc@unb.ca;

*Corresponding author

Abstract

Biomedical researchers consume and analyze PubMed abstracts on a daily basis seeking to update their existing knowledge with insights from newly published literature. Plain text descriptions fail to deliver contextual knowledge to users who require a comprehensive understanding of the content of a paper before deciding to access it. To achieve this biological named entities described in the abstracts must be linked to their related entries in biological databases and established controlled vocabularies such as SwissProt and Gene Ontology. Semantic Assistants support users in content retrieval, analysis, and development, by offering context-sensitive NLP services directly integrated in standard desktop clients, like a word processor. They are implemented through an open service-oriented architecture, using Semantic Web ontologies and W3C Web Services.
Here we present a deployment of the Semantic Assistants framework to provide links from mutation, protein, protein property, gene and organism mentions in abstracts to their related entry in standardized biological databases and controlled vocabularies. The underlying text mining pipeline used to identify named entities has previously shown high levels of precision and we make this functionality easily accessible through a Semantic Assistant, to end users when reviewing PubMed abstracts in through a Firefox client.

Introduction

The proliferation of annotation services for biologists reviewing scientific literature is based on the adoption of web services designed to provide links to contextual information provided in online databases. These annotation services, such as Reflect [1], tag gene, protein and small-molecule names and link them to external resources with sequence and structure information for given default organisms. Further deployments of annotation services, such as BioDEAL [2] facilitate a feedback-loop whereby scientists manually link biological concepts to published evidence through a web-browser frontend, making it possible for biologists to collectively build and share knowledge. BioDEAL also facilitates users who want to make use of natural language processing tools in their annotation work. Whereas BioDEAL leverages social networking between biologists a standardized framework, allowing rapid customization to new application scenarios by multiple stakeholders, is required.

Recent work on Semantic Assistants [3] has opened up the possibility of providing server side natural language processing of texts being reviewed or drafted on client side applications through web services. In so doing annotations can be pushed directly to users of dedicated desktop clients or browsing the web with browsers enabled with plug-in extensions. The Semantics Assistant framework was applied in the domain of Telecom [4] where annotations in the form of, of OWL-DL axioms from a telecom ontology, were linked to named entities via canonical names mapped to Semantic classes in the Telecom Ontology. In this contribution we illustrate the deployment of a Semantic Assistant for the relay of grounded mutations and mutation impact; moreover our implementation involves the use of GreaseMonkey, an extension to Mozilla Firefox, as the Semantic Assistant Client allowing the delivery of these mutation annotations when browsing PubMed on the Web.

Methods

Semantic assistants

In previous work [3] the notion of a semantic assistants and the semantic assistant architecture was established. Primarily this involves the integration of text analysis services and end-user clients. The architecture consists of four tiers, described here briefly: (i) Client tier; clients are typically word processors, web browsers or any other application that render text, (ii) Presentation and Interaction tier; a web server containing modules that translates results from NLP services into formats compliable with the clients, (iii) Analysis and Retrieval tier; the actual NLP systems, which for a given text produce output annotations, (iv) Resource tier, provide support for the NLP systems with external information from other documents or databases.

Semantic assistant clients

Reported semantic assistant deployments rely on clients that are plug-ins for text editors as e.g. OpenOffice. In current work we have implemented a Semantic assistant client with the use of Greasemonkey [5], an extension to Mozilla Firefox allowing users to install scripts supporting augmented browsing, i.e. making changes to webpage content. The script we have built pre-processes a PubMed entry page and sends the abstract, via a Java Servlet wrapping the Semantic Assistant Java API, to our mutation impact extraction service and annotates the abstract with the results as shown in Figure 1. An important difference between text in word processors and web page content is that the latter is not editable and contains both texts of interest and, depending on the web page selected, content that can be considered nonsense. For a given page the list of web services have to be extended according to the content they will process (which webpage and which part of that webpage). As an example, the NLP service ”mutation impact extractor” can be customized to a ”mutation impact extractor for pubmed abstracts”. Moreover text processing algorithms for some NLP services may be context dependent in that they may rely on co-occurrence of terms within a certain distance from named entities. Consequently multiple texts or other surrounding content might disrupt the underlying algorithms for entity recognition and disambiguation. This makes the construction of semantic assistant clients for web browsers a bit more tedious. We propose a client that will support browsing, augmented with semantic tags, by using Ajax technologies in combination with the existing semantic assistant architecture. The core of the client is a Java servlet making use of a Java API containing a precompiled client-side abstraction layer which performs the actual communication with the server. The exterior of the clients consist of a set of scripts on which they can make changes to HTML pages with annotations retrieved via asynchronous calls to the Java servlet.

Text mining pipeline

The backbone of a Semantic Assistant is the NLP service that process text and outputs annotations or deduced facts. We have previously developed such an NLP algorithms and semantic infrastructure which finds, annotates and disambiguates biological entities by using a combination of gazetteer-based approaches and methods for relation detection. The following entities are currently extracted: proteins, genes, organisms, point mutations, protein properties and mutational impacts on protein properties. In particular these algorithms facilitate the grounding (linking) of proteins and mutations to SwissProt entries and correct position on amino acid sequences [6] and protein properties to gene ontology concepts [7]. The results of grounding are made available to the end users through the semantic assistant. Here we outline briefly these algorithms that deliver grounded mutations and impacts end users of the Semantic Assistant Client.

Firstly the cross-linking entities found in text with their real-world counterparts, called grounding, requires the extraction and normalization of mutation mentions, for which we used the MutationFinder system [8] in combination with the GATE text mining framework as well as custom built gazetteer lists built from the text format version of Swiss-Prot. To facilitate grounding, a local store of mappings between names and primary accession numbers and amino acid sequences was created. To facilitate grounding of mutations to proteins and the correct amino acid residues on those proteins a set of candidate protein sequences must be established. Based on protein names found in target documents a pool of protein accessions and corresponding sequences is identified. Subsequently mutations extracted from the text are mapped onto the candidate sequences using regular expressions generated from the mutation mentions extracted from the text, where the regular expressions are constructed using multiple the wildtype residues and the distance between them. For regular expressions matching a candidate sequence, further matching of mutations from the target document is explored, taking into account the numbering displacement found when using the regular expression. Accession numbers of the correctly grounded proteins are considered to be the wildtype sequence of the protein in the document.

Secondly the extraction of mutation impacts from documents relies on both the identification of protein functions, found in noun phrases and the extraction of directionality terms also found in sentences adjacent to mutation mentions. This is achieved as the result of identifying Gene Ontology Molecular Function terms involving activity, binding, affinity or specificity as the head noun in noun phrases identified using the multi-lingual noun phrase extractor [9]. Identification of directionality of a mutational change is achieved using custom built gazetteer lists. Lastly relations must be established between directionality words and protein properties in order to identify impact statements and between mutants and these impacts statements. A rule based approach is used to identify relations and a scoring of significance is achieved using heuristics based on entity distance. Full details of these algorithms are outlined in [6, 7].

Discussion

Although we can show that our NLP services work for PubMed abstract, we know that the performance of the underlying grounding algorithms is decreased when switching from full-text to abstracts only. To enhance this performance and to retrieve more information, the semantic assistant client should send the full-text content to the service, this can be made via existing web services like EFetch [10], whenever the paper in question is publicly available as full-text.

For future work, we will host a web site with listings of available services. The users can, when choosing a service, also restrict it to only be run on certain web sites as e.g.:

```
http://www.ncbi.nlm.nih.gov/pubmed/*
```

To ensure that new customized services can be constructed and consumed by both service providers and end-users, template client scripts should be made available where parameters can be set for content restriction, i.e. only send content in specific document elements as e.g.:

```
<p id="abstract">content</p>
```

And also to set custom styles for output annotations. These customized client scripts can then be stored and listed as services, available to other users as well.

Conclusion

The adoption of new frameworks providing online annotations to already published content is an emerging theme trend in life science knowledge discovery. In this brief work we have shown that existing algorithms for grounding of mutation mentions and related content can be deployed to great effect albeit in prototype scale application. This serves both as a strong motivation for deploying further semantic assistants for other named entities and as a test bed to solicit further requirements from end users. In the process of deploying this prototype in an open source web browser we have identified constraints that will impact the design of next generation Semantic Assistants.

Abbreviations used

NLP: Natural Language Processing; OWL-DL: A Web Ontology sub-Language; API: Application Programming Interface; HTML: HyperText Markup Language;

Competing interests

The authors declare that they have no competing interests.

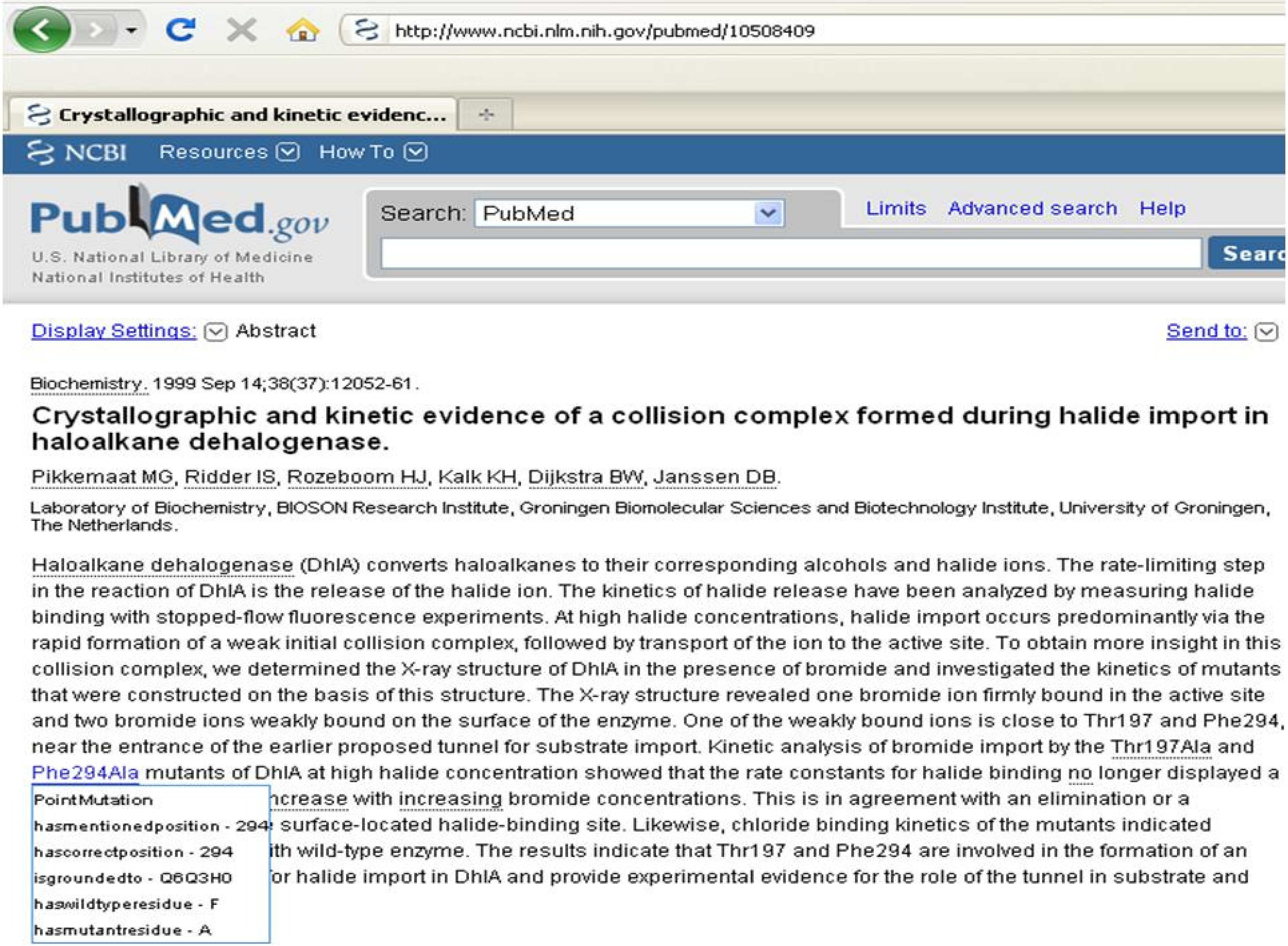
References

1. Pafilis E, O'Donoghue SI, Jensen LJ, Horn H, Kuhn M, Brown NP, Schneider R: **Reflect: augmented browsing for the life scientist**. *Nature Biotechnology* 2009, **27**:508–510.
2. Breimyer P, Green N, Kumar V, Samatova N: **BioDEAL: community generation of biological annotations**. *BMC Medical Informatics and Decision Making* 2009, **9**(Suppl 1):S5, [http://www.biomedcentral.com/1472-6947/9/S1/S5].
3. Witte R, Gitzinger T: **Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients**. In *3rd Asian Semantic Web Conference (ASWC 2008), Volume 5367 of* LNCS, Bangkok, Thailand: Springer 2009:360–374, [http://rene-witte.net/semantic-assistants-aswc08].
4. Kouznetsov A, Shoebottom B, Witte R, Christopher J OB: **Leverage of OWL-DL axioms in a Contact Centre for Technical Product Support**. In *OWL: Experiences and Directions (OWLED 2010)*, San Francisco, California, USA 2010.
5. **Greasemonkey.** [http://www.greasespot.net].
6. Laurila JB, Kanagasabai R, Baker CJO: **Algorithm for Grounding Mutation Mentions from Text to Protein Sequences.** *Seventh International Conference on Data Integration in the Life Sciences, Gothenburg, Sweden* 2010.
7. Laurila JB, Naderi N, Witte R, Riazanov A, Kouznetsov A, Baker CJO: **Algorithms and semantic infrastructure for mutation impact extraction and grounding**. *Ninth International Conference on Bioinformatics (InCoB2010). Tokyo, Japan* 2010.
8. Caporaso J, Jr WB, Randolph D, Cohen K, Hunter L: **MutationFinder: a high-performance system for extracting point mutation mentions from text**. *Bioinformatics* 2007, **23**:1862–1865.
9. **Multi-lingual Noun Phrase Extractor.** [http://www.semanticsoftware.info/munpex].
10. **EFetch.** [http://eutils.ncbi.nlm.nih.gov/corehtml/query/static/efetch_help.html].

Figures

Figure 1 : Tagged PubMed abstract.

The abstract is tagged according to the output of our mutation impact NLP service. Here we can see that the protein *Haloalkane Dehalogenase* is tagged together with two mutations on the same protein. Algorithms for protein and mutation grounding used by the back-end NLP system make sure the mutations refer to the correct position on the correct amino acid sequence as seen in the popup box, produced when the mouse pointer hover over a tag surrounding the *F294A* point mutation mention.


http://www.ncbi.nlm.nih.gov/pubmed/10508409
Crystallographic and kinetic evidenc...
NCBI
Resources
How To
PubMed.gov
U.S. National Library of Medicine
National Institutes of Health
Search: PubMed
Limits
Advanced search
Help
Searc
Display Settings: Abstract
Send to:

Biochemistry. 1999 Sep 14;38(37):12052-61.

Crystallographic and kinetic evidence of a collision complex formed during halide import in haloalkane dehalogenase.

Pikkemaat MG, Ridder IS, Rozeboom HJ, Kalk KH, Dijkstra BW, Janssen DB.

Laboratory of Biochemistry, BIOSON Research Institute, Groningen Biomolecular Sciences and Biotechnology Institute, University of Groningen, The Netherlands.

Haloalkane dehalogenase (DhlA) converts haloalkanes to their corresponding alcohols and halide ions. The rate-limiting step in the reaction of DhlA is the release of the halide ion. The kinetics of halide release have been analyzed by measuring halide binding with stopped-flow fluorescence experiments. At high halide concentrations, halide import occurs predominantly via the rapid formation of a weak initial collision complex, followed by transport of the ion to the active site. To obtain more insight in this collision complex, we determined the X-ray structure of DhlA in the presence of bromide and investigated the kinetics of mutants that were constructed on the basis of this structure. The X-ray structure revealed one bromide ion firmly bound in the active site and two bromide ions weakly bound on the surface of the enzyme. One of the weakly bound ions is close to Thr197 and Phe294, near the entrance of the earlier proposed tunnel for substrate import. Kinetic analysis of bromide import by the Thr197Ala and Phe294Ala mutants of DhlA at high halide concentration showed that the rate constants for halide binding no longer displayed a
ncrease with increasing bromide concentrations. This is in agreement with an elimination or a
surface-located halide-binding site. Likewise, chloride binding kinetics of the mutants indicated
ith wild-type enzyme. The results indicate that Thr197 and Phe294 are involved in the formation of an
or halide import in DhlA and provide experimental evidence for the role of the tunnel in substrate and

PointMutation
hasmentionedposition - 294
hascorrectposition - 294
isgroundedto - Q6Q3H0
haswildtyperesidue - F
hasmutantresidue - A