

Benchmarking infrastructure for mutation text mining

Artjom Klein^{*1}, Alexandre Riazanov², Matthew M Hindle³ and Christopher JO Baker¹

¹Computational Science And Statistics Department, University of New Brunswick, Saint John, Canada

²IPSNP Computing Inc, Canada

³Synthetic and Systems Biology, Edinburgh University, Edinburgh, UK

Email: Artjom Klein* - aklein@unb.ca; Alexandre Riazanov - alexandre.riazanov@ipsnp.com; Matthew M Hindle - matthew.hindle@ed.ac.uk; Christopher JO Baker - bakerc@unb.ca;

*Corresponding author

Abstract

Background: Experimental research on the automatic extraction of information about mutations from texts is greatly hindered by the lack of consensus evaluation facilities and easy-to-use infrastructure for the testing and benchmarking of mutation text mining systems.

Results: We propose a community-oriented annotation and benchmarking infrastructure to support development, testing, benchmarking, and comparison of mutation text mining systems. The design is based on semantic standards, where RDF is used to represent the annotations, an OWL ontology provides an extensible schema for the data and SPARQL is used to compute various performance metrics, so that in many cases no programming is needed to analyze results from a text-mining system. While large benchmark corpora for biological entity and relation extraction are focused mostly on genes, proteins, diseases, and species, our benchmarking infrastructure fills the gap for mutation information. The core infrastructure comprises (1) an ontology for modelling annotations, (2) SPARQL queries for computing performance metrics, and (3) a sizeable collection of manually curated documents, that can support mutation grounding and mutation impact extraction experiments.

Conclusion: Our infrastructure is the first of its kind for mutation text mining. The use of RDF and OWL as the representation for corpora ensures extensibility. The infrastructure is suitable for out-of-the-box use in several important scenarios and is ready, in its current state, for initial community adoption.

Introduction

Mutation text mining. The use of knowledge derived from text mining for mentions of mutations and their consequences is increasingly important for systems biology, genomics and genotype-phenotype studies. Mutation text mining facilitates a wide range of activities in multiple scenarios including the expansion of disease-mutation database annotations [1], the development of tools predicting the impacts of mutations [2,3], the modelling of cell signalling pathways [4] and protein structure annotation [5,6]. The types of useful text mining tasks specific to mutations range from the relatively simple identification of mutation mentions [7], to very complex tasks such as linking ("*grounding*") identified mutations to the corresponding genes and proteins [8], or identifying mutation impacts [9,10] and related phenotypes [11]. Although the demand for mutation text mining software has lead to a significant growth of the experimental research in this area, the development of such systems and the publication of results is greatly hindered by the lack of adequate benchmarking facilities. For example, in developing a mutation grounding system [8] showing an encouraging level of performance accuracy, 0.73, on a homogeneous corpus of 76 documents, the authors achieved only 0.13 on a heterogeneous corpus of larger size. When the system was reimplemented (see [12]), the authors encountered another challenge – the evaluation of the new system by comparing it to the state-of-the-art was practically unaffordable, despite the existence of similar systems, due to the lack of consensus benchmarking infrastructure.

Such challenges and evaluation issues are not unique specific for mutation text mining and are also present in other domains of biomedical text mining. In the following subsection, we discuss benchmarking and evaluation difficulties in biological text mining in general, which are also relevant to mutation text mining.

Benchmarking and evaluation difficulties in biomedical text mining. Benchmarks, in the form of annotated corpora and related software utilities, are usually designed and created for specific text mining tasks and support fixed, usually hard-coded, evaluation metrics. Besides quantitative and qualitative characteristics – number of entities annotated, distribution of annotation types, etc. – a corpus is characterized by the format, annotation schema (semantics of annotations, annotation types), and evaluation metrics to calculate the performance of the text mining systems. For example, the benchmark for MutationFinder [7] (one of the most popular single point mutation extractors) is in a custom tabular format, stores annotations and raw text separately, has annotations of single point mutation mentions without references to their position in text and provides three different performance metrics.

Comparative evaluations between systems and evaluation of these systems on different *gold standard data* sets is an important aspect for performance verification and adoption. Developers need to be able to

convincingly evaluate their systems performance by comparing their results with extensive gold-standard data-sets and results of other systems. Since systems are often integrated as sub-programs in new text mining pipelines, other developers need to compare them when looking for a better candidate to be integrated in their new system. Biocurators use text mining tools to pre-annotate documents for manual annotation. They evaluate third-party tools in order to find a system which performs better on representative benchmarks.

There are several issues related specifically to comparative evaluation:

- **Availability** Often developers only publish their results and not their corpora and systems. The resources can not be re-used and tested, because they are just not available.
- **Reproducibility** It is often the case that developers do not provide instructions on how to reproduce their results, or the instructions require considerable effort from the user (download code, compile it, download corpus, train system on corpus, run test and evaluations). This might be a practical challenge for a person who wants to perform comparative analysis but does not have the specific skills or knowledge of the required tools.
- **Interoperability** Evaluation is hindered by the diversity and heterogeneity of formats and annotation schemas of corpora and systems. In order to compare systems, developers have to convert corpora and their systems to appropriate formats. Definitions and implementations of evaluation metrics are often format and schema dependent. Using a different schema or modifying a schema requires the re-implementation of metric calculation scripts. Re-usability is also hindered by the complexity of native corpora formats. Some of them are so idiosyncratic that special programs have to be developed to convert them to a unified format or the format used by the system to be tested.
- **Comparability** Corpora vary in qualitative and quantitative characteristics which might significantly affect the performance evaluation results. Ideally, text mining system must be tested on *large* corpora with *representative characteristics*.
- **Diversity of metrics** Text mining systems are usually evaluated by using such performance metrics as *precision* and *recall* and different flavours of these statistics are used by different system developers. For example, text mining results sometimes need to be evaluated with different granularities, e. g., the mutant protein property change may be evaluated by considering binary

outcomes (*has effect* vs *no effect*) or with higher granularity when the outcome may also identify the direction of the effect – e. g., *positive effect* or *negative effect*.

Our research goals. The lack of adequate benchmarking infrastructure for the community is a great hindrance to the objective evaluation, and publication, of mutation text-mining research. Therefore, we developed an extensible and multi-purpose infrastructure, based on a consensus corpora and utilities for the community, in order to make such benchmarking and evaluation easy.

Requirements. To orient our work, we imposed the following requirements on the infrastructure:

- To maximize its utility for system testing and evaluation, the infrastructure must include as *large a gold standard corpus* (a collection of annotated texts) as possible. It must also contain results of the runs of different systems to facilitate comparison of their performance as well as comparative evaluation of new systems.
- To be useful to a larger community, the infrastructure should support *multiple mutation-related text mining tasks*, such as identifying mutations both on protein and DNA levels, mutation grounding to genes and proteins, identifying effects of mutations, etc.
- There should be support of annotations on different levels (to distinguish annotations on *sentence level* where positions of annotation in text are specified, and *document level* annotations assigned to the document itself, e. g. annotation “*in document PMID:7705350 mutation N30D is grounded to protein with UniProtID:P22643*”). Sentence level annotations are required by many important applications. For example, the curation of text-mined information about mutations intended for inclusion in databases is much more efficient if sentence level provenance is provided. However, some systems, especially early prototype, do not provide sufficiently precise references to text fragments.
- Query facilities are required to search the corpora and system results for performance analysis, data drill-down and computation of statistics, such as finding the number of annotated named entities, their types, distribution of annotation types within corpora, etc.
- The infrastructure must be easy to use and require only minimal effort from system developers. Ideally, many development tasks should be facilitated out-of-the-box, so that the developers do not need to create new data formats or write additional scripts in order to leverage the infrastructure.

Article overview. In this article we report on the design and implementation of an annotation and benchmarking infrastructure, to support the development, testing, benchmarking, and comparison of

systems for extracting information about mutations in the text-mining community. The article is outlined as follows. The *Methods* section describes the motivation for the choice of representation format, outlines the ontologies used for modelling annotations and briefly introduces our approach for calculating evaluation metrics. The *Results* section presents details of the seed corpora and methods for the calculation of performance metrics, and describes relevant utility programs. In the *Evaluation* section we describe two case studies, used to test the infrastructure. The *Future work* section announces the forthcoming extensions to the infrastructure. Finally, the *Conclusion* section summarizes the results and specifies how the infrastructure can be acquired or accessed.

Related work

The issues of availability (1) and reproducibility (2) mentioned in the introduction strictly depend on the motivation of researchers and developers to publish their corpora, text-mining systems and re-producible results. The issues of interoperability (3) and comparability (4) have been already addressed several times in the literature. In [13], six different corpora were analyzed with respect to their usage. The authors note the effect of design features and characteristics – especially the format of the corpus – on the usage rate. They empirically confirm that corpora in more common formats are more widely used than corpora in more *ad hoc* formats. The authors of [14] conclude that the format of a corpus is one major factor that hinders publishing it and re-use of the corpus outside of the lab that developed it, they propose to use special-purpose converters to rectify this and demonstrate the feasibility of the approach by writing a program to convert a protein-related corpus into two more popular formats.

The problems of interoperability, comparability and re-usability of text mining resources were specially addressed by the BioCreative group through the organization and promotion of the BioCreative Interoperability Initiative [15]. Its goals include promoting simplicity, interoperability, and the broad use and reuse of text mining resources by introducing and popularizing a new annotation standard – *BioC*, an interchange format for corpora and tools in BioNLP.

A practical attempt of standardizing and improving the interoperability of resources in the protein-protein interaction (PPI) domain was done by [16]. The authors compare five PPI corpora and two PPI extraction systems and point out that the transformation of the corpora - which are in XML format and have highly idiosyncratic native schema - into the unified format was a tedious process requiring significant effort. Nevertheless, complex transformation programs were developed for the corpora, although in several cases manual disambiguation could not be avoided. One of the main findings in [16] was that methods evaluated

on different corpora of different size, domains and annotations schemas vary significantly and the choice of corpus has a larger effect on the result than even the choice between different PPI extraction methods. They conclude that *“the BioNLP community faces a situation where it is difficult, if not impossible, to reliably identify the best published methods and techniques due to a lack of information on the comparability of their evaluated performance”*.

To predict and avoid the difficulties and issues identified, in particular, in the domain of PPI text mining, we propose to improve this situation by developing a centralised, publicly accessible multi-purpose benchmarking infrastructure for mutation text mining systems.

Methods

Representing gold standard annotations and system results in RDF

Typically, document annotations intended for the testing and evaluation of text-mining systems, are represented in various custom XML-based or tabular formats. XML is a standard and widely used generic format for corpora annotations and comes with a large number of tools. However, the processing of complex annotations in specific XML-based formats – parsing, storing, querying, evaluation – is usually impossible in practice with off-the-shelf XML tools [17]. Developers of text-mining systems need to create schema-specific parsers and processing scripts and change them every time the schema is changed or extended. This was the primary reason we chose RDF over custom XML-based formats, as the representation for our annotation, because the reusability and extensibility of data are among the main design goals of RDF. We also use OWL ontologies as a highly extensible form of data schemas. Our choice was additionally motivated by the fact that the RDF/OWL bundle is increasingly adopted as a medium for exchanging biomedical data. For example, it is the basis of the BIOPAX [18] format for representing biological pathway data.

The advantages of using the RDF/OWL bundle can be summarized as follows:

- **Extensibility and reusability.** Since the benchmarking infrastructure is intended for different mutation text mining tasks and all requirements can not be foreseen, the annotation representation must be extensible. Moreover, the same data may be used for different tasks (e. g., we have reused mutation impact corpora for a improving mutation grounding system [12]).

The use of RDF data with classes and properties defined in OWL ontologies makes it possible to support easy integration of new corpora with annotation schemas that need not be identical, as long as they are compatible. This simply amounts to *using compatible OWL ontologies and modelling*

patterns for RDF. Data defined modulo one ontology can be *simply merged* with data modulo another ontology. Moreover, additional alignments between the ontologies can be potentially provided by the annotation providers – corpus curators or text mining system developers to facilitate tighter integration of the data.

The reuse of data is in some cases also trivial because the RDF and OWL-based representation is *semantically explicit*: when a new text mining task has to be evaluated, it suffices to identify the relevant fragment of the OWL ontology.

- **Tool availability.** RDF and OWL are popular open formats and supported by a large number of open source and commercial tools. At least the following types of tools can be leveraged for the purpose of text mining annotation processing:

- The SPARQL query language can be directly used for calculating system performance metrics as well as for various drill-down searches in the gold standard corpora. There is no need to implement custom querying mechanisms.
- Multiple implementations of RDF databases (*triplestores*) are available that facilitate efficient storing and querying of large volumes of annotations.
- RDF and OWL APIs for multiple programming languages, including Java, C++, Perl and Python, facilitate easy programmatic generation and manipulation of corpus annotations or RDF data representing text mining results.
- OWL reasoners can be used for data integrity checking.

The available RDF/OWL tools facilitate out-of-the-box use of annotations and system results in the main use scenarios, such as system testing and evaluation.

Core Ontologies and Modelling

The schema of our benchmarking infrastructure comprises two ontologies: (1) an annotation ontology modelling annotations in corpora and system results, and (2) a domain ontology modelling entities and their relations extracted from text. We briefly discuss the ontologies here.

Annotation Ontology

The Annotation Ontology (AO) [19] is an open-source ontology for annotating scientific documents on the web. We use AO to model corpora annotations as well as the relevant parts of the text-mining systems

results. Our annotations are metadata anchored to whole documents or specific fragments of texts. They are characterized by type and optional features. In AO, annotations are resources and realized as instances of the class **Annotation**. Each annotation has a **hasTopic** property. The value of the property is an entity extracted from text, e. g. mutation, protein, etc. This entity represents the type of the annotation.

We distinguish between two kinds of annotations: (1) *Document level annotations* are not anchored to specific fragments of text. They annotate a document as a whole, e. g. “mutation A is contained in document B”, “protein A is the topic of document B”. The annotations are linked to documents via the **annotatesDocument** property. (2) *Text level annotations* are anchored to specific fragments of text, e. g., “Mutation A appears in document B at position P”. Text level annotations are linked to text via instances of the **TextSelector** class. Text selector identifies a text fragment by its positions in the text or by its context. The property **context** binds annotations with text selectors.

Domain Ontology

The Mutation Impact Extraction Ontology (MIEO) [20] is central to our infrastructure. It currently describes classes and properties necessary to represent core types of information about mutations at protein level, identified in texts, and identified impacts of mutations on the molecular functions of proteins. For example, **AminoAcidSequenceChange** is the class for mutations at protein level. Instances of **ProteinVariant** are most specific types of protein molecules that completely identify the corresponding amino acid sequences. Instances of **ProteinPropertyChange** represent identified changes of protein properties that can be linked to (1) the properties that change, (2) the corresponding documents and specific text fragments, and (3) the mutations they result from. To characterize a property change, e. g., as positive, which may correspond to increased activity, we can use the subclass **PositiveProteinPropertyChange**. Protein properties, such as molecular functions, are also modelled as individuals whose types are currently taken from the Gene Ontology [21].

Note that some of the mutation tasks we are interested in are related to the extraction of relations between entities rather than just identifying some entities of interest. We use custom reification for such relations, in particular to facilitate linking them to documents and more specific text fragments. For example, extracted statements of mutations impacting protein properties are represented as instances of the class **StatementOfMutationEffect** instead of just straightforwardly linking the involved entities with appropriate non-reified predicates.

For better interoperability, our MIEO uses the Semanticscience Integrated Ontology (SIO) [22] as an upper

ontology, and the LSRN ontology [23] to represent records and identifiers, as illustrated in the next section.

Modelling example.

We provide an RDF graph in a pseudo-N3 notation as an example of how the gold standard corpus data and mutation impact text mining system results are represented in our framework. Note that

non-mnemonic ontological identifiers are replaced with pseudo-identifiers using the corresponding labels:

e. g., `sio:SI0_000011` and `sio:SI0_000300` are replaced respectively with `sio:'has attribute'` and

`sio:'has value'`.

```
# Description of a singular amino acid substitution N30A:
:singular_mutation1 a mieo:AminoAcidSubstitution;
  mieo:mutationHasWildtypeResidue mieo:Asparagine;
  mieo:mutationHasMutantResidue mieo:Alanine;
  mieo:mutationHasPosition
    [ a sio:'position';
      sio:'has value' "30"^^xsd:integer ] .

# Description of a singular amino acid substitution N50A:
:singular_mutation2 a mieo:AminoAcidSubstitution;
  mieo:mutationHasWildtypeResidue mieo:Asparagine;
  mieo:mutationHasMutanResidue mieo:Alanine;
  mieo:mutationHasPosition
    [ a sio:'position';
      sio:'has value' "50"^^xsd:integer ] .

# Combined mutation ("mutation series") consisting of the two singular mutations:
:mutation a mieo:CombinedAminoAcidChange;
  sio:'has member' :singular_mutation1, :singular_mutation2;
  sio:'has attribute' :number_of_singular_mutations .
:number_of_singular_mutations a sio:'count';
  sio:'has value' "2"^^xsd:integer .

# Mutation application ("grounding") to a specific protein:
:mutation_application a mieo:ProteinMutationApplication;
  mieo:isApplicationOfMutation :mutation;
  mieo:isApplicationOfMutationToProtein :protein .

# Description of the protein:
:protein a mieo:ProteinVariant; # it's a specific variant (uniquely identifies the sequence)
  mieo:proteinHasSequence :protein_sequence;
  sio:'is subject of' :uniprot_record .

# Standard SIO way to link entities, DB records and IDs:
:uniprot_record a lsrn:UniProt_Record;
  sio:'has attribute'
    [ a lsrn:UniProt_Identifier;
      sio:'has value' "P22635" ] .

# Provenance is mostly done with ao:hasTopic :
:document a sio:'article' .
:ann_mutation a ao:Annotation;
  aof:annotatesDocument :document;
  ao:hasTopic :mutation .
:ann_protein a ao:Annotation;
  aof:annotatesDocument :document;
  ao:hasTopic :protein .

:document sio:'is subject of'
  [ a lsrn:PMID_Record;
    sio:'has attribute'
      [ a lsrn:PMID_Identifier;
        sio:'has value' "17526795"^^xsd:string ] . ] .
```

Note that, for simplicity, the RDF data in this example are in “flat” RDF. In practice this is not convenient because we need to somehow separate the gold standard data from system results. Moreover, it is necessary to separate results coming from different systems or different experiments. In practice, we use *named graphs* [24] for this purpose: results from different experiments and gold standard data from different corpora are placed in separate named graphs.

Performance metrics computation with SPARQL

An infrastructure intended for benchmarking and evaluation must support the computation of performance metrics, such as precision and recall. Different variants of these statistics vary in different systems, by their domains, level of specificity and granularity. For example, [25] proposes over 15 different metrics for evaluation of protein mutation extraction systems. Our infrastructure has to be sufficiently flexible to accommodate many different uses. This is achieved by using SPARQL to retrieve entities, such as different flavours of true and false positives, that need to be counted in order to calculate a particular metric. The current version of SPARQL (1.1) offers a sufficient degree of flexibility. In particular, the *negation-as-failure* related features – `FILTER NOT EXISTS` and `MINUS` – facilitate easy qualification of some system results as *false positives* by checking whether they are absent from the gold standard data, as will be illustrated in the *Evaluation* section.

Design of the seed corpora

To facilitate a preliminary evaluation of our infrastructure, we seeded it with several corpora supporting several mutation text mining tasks: (1) mutation grounding to proteins, (2) extraction of mutation impacts on molecular functions of proteins, (3) impact sentence-recognition, and (4) grounding impact-sentences to mutations.

The document annotations for mutation grounding identify extracted mutations and proteins, and relations between them. The annotations for mutation impact extraction additionally identify molecular functions of proteins and changes of these properties causally linked to some mutations, and provide references to supporting text-fragments. The annotated mutation impact sentences and associated with them mutations support tasks (3) and (4).

Results

Contents of the corpora

EnzyMiner-based corpus.

One of our seed corpora was based on an extract from the EnzyMiner [26] database of publication abstracts. It was annotated manually and comprised 38 semi-randomly selected full text documents with 176 different singular mutations linked to 48 different protein sequences. The selection was adjusted to ensure maximal diversity by having documents with proteins from all enzyme families and 24 different species. The corpus contained 440 statements (occurrences of impact information in text), 57 molecular functions and 20 combined mutations.

From herein we shall refer to it as “the EnzyMiner corpus”.

We annotated documents with mutation impact information which includes:

- Identified protein-level **mutations**, in the form of singular amino acid substitutions. They are represented as triples specifying the wild type and mutant residues, and the absolute positions of the mutations on the corresponding amino acid sequences. For situations when the effects of several simultaneous amino acid substitutions are considered in the document, we allowed them to be expressed as *combined mutations*, which are conceptually just sets of singular amino acid substitutions.
- **Proteins** to which the mutations are related, identified with UniProt IDs. The host organisms and sets of specific protein sequences can be identified via the UniProt IDs.
- **Protein properties** specified as Gene Ontology Molecular function classes.
- **Mutation impacts** qualified as *Positive*, *Negative* or *Neutral*.
- **Text fragments** the information was extracted from. Typical fragments contain mentions of protein properties, impact-directionality words, such as “increased” or “worse”, mutation mentions, protein and organism names, etc.
- **Documents** identified with PubMed IDs.

DHLA corpus.

This is a small corpus comprising 13 documents with 52 unique (per document) mutations on Haloalkane Dehalogenases, manually annotated similarly to the EnzyMiner documents (see [5]).

COSMIC-based corpus.

We have an extract from the COSMIC database [27] containing 63 documents for three target genes: FGFR3, MEN1 and PIK3CA. Unlike the EnzyMiner and DHLA corpora, this corpus does not identify mutation impacts, although it links mutations to proteins and, thus, is suitable for mutation grounding benchmarking.

KinMutBase-based corpus.

We retrieved 201 documents annotated with singular amino acid substitutions grounded to proteins, from the KinMutBase [28] database. We additionally curated the selection by running MutationFinder [7], which is a reliable tool for this purpose due to its very high recall, and comparing the results with the annotations in the database. Based on this comparison, we discarded about 70 documents that appear annotated with protein-level mutations that are not mentioned directly and are likely to have been inferred from SNPs by the curators. The final size of the corpus was 128 documents. In total, we had 271 mutations linked to 26 different UniProt identifiers.

Open Mutation Miner Impact corpus.

This corpus containing 40 documents was used in [9] to test the Open Mutation Miner system. It contains impact sentence annotations with the EC codes of enzymes, host organisms and mutations. An impact sentence describes a mutation impact on a protein property and does not necessarily contain a mutation mention. 48 of 2045 impact sentences were not grounded to mutations. If a sentence contained several impact mentions, it was annotated several times. Unlike the Enzyminer corpus, the OMM Impact corpus was not annotated with protein properties or mutation impact direction.

Corpora statistics.

The statistics for the corpora are summarized in Tables 1 and 2.

	Number of documents	UniProt IDs	Mutations*
EnzyMiner	38	49	176
KinMutBase	128	26	271
DHLA	13	4	52
PIK3CA	30	1	169
FGFR3	26	1	174
MEN1	7	1	22

Table 1: Corpus statistics for the mutation grounding task. (*) - unique per document.

	Number of documents	Impact mutations*	Impacts* (mutation, protein property, impact direction)	Impact sentences	Impact sentences grounded to mutations
OMM Impact	40	223	-	2045	1997
EnzyMiner	38	172	282	440	440
DHLA	13	52**	73	-	-

Table 2: Corpus statistics for mutation impact extraction tasks. (*) - Unique per document. (**) - The OMM Impact and Enzyminer corpora contain single point mutations as well as combined mutations. There are only single point mutations in the DHLA corpus.

RDF database

The RDF files representing our corpora are already relatively large, so for the purposes of efficient SPARQL querying we deployed the data to a Sesame triplestore. Users have the option of downloading the RDF data and using their own querying machinery, or accessing our DB via a public SPARQL endpoint. The details can be found on the project portal [29].

SPARQL queries for performance metrics

To implement and illustrate the idea of using SPARQL for performance metrics computation, we formulated several SPARQL queries sufficient for computing precision and recall for systems implementing four text mining tasks:

- **(T1) Mutation grounding to proteins.** The result is a pair of mutation and protein with corresponding UniProtID that identifies protein sequence. We adopted the definitions of precision and recall for this task from [10]: *precision* was defined as the number of correctly grounded mutations over all grounded mutations and *recall* was defined as the number of correctly grounded mutations over all uniquely mentioned mutations.
- **(T2) Extraction of mutation impacts on molecular functions of proteins – *mutation-impact relations*.** The metrics were also taken from [10]. For mutation-impact relations, *precision* was defined as the number of correct relations over all retrieved relations and *recall* was defined as the number of correct relations over all uniquely mentioned relations. In order for an extracted mutation-impact relation to be considered correct all the parts had to be correct i.e. the affected protein property, the direction of the impact and the cause mutation. If the protein property was a molecular function, it had to be normalised by grounding to Gene Ontology.

- **(T3) Impact sentence recognition.** This task was evaluated in [9]. *Precision* was defined as the number of correctly identified impact sentences over all recognized impact sentences. The sentence is correctly identified if it matched the gold standard.
- **(T4) Grounding impact sentences to mutations.** This task was also considered in [9]. *Accuracy* for this task was defined as the fraction of correctly identified impact sentences, grounded to correct mutations, over all correctly identified impact sentences.

For each task we wrote (1) a SPARQL query that selects relevant annotations in the gold standard data, representing correct cases, (2) a query that selected all relevant/retrieved results of the text mining system being evaluated, and (3) a query that selected only correct results. These selections were enough to calculate precision and recall.

Example queries.

We illustrate a specific example of a metric SPARQL query by presenting a slightly simplified version of the query used to select the correct results from mutation-impact relation extraction. According to the definitions in (T2), a result was defined as a tuple – a *document*, a *mutation*, a *protein property* changed by the mutation, and a *direction* of the property change. If the gold standard data RDF graph contained a corresponding subgraph, the result was considered *correct*. Technically we had to compare two named RDF graphs and obtain the corresponding intersection. The resulting query below assumes that the gold-standard data is kept in the named graph `http://example.com/gold-standard.rdf` and the system results came from another named graph `http://example.com/experiment.rdf`.

Note that, as in the modelling example, we replace non-mnemonic SIO identifiers with their labels, for better readability,

```
PREFIX sio:<http://semanticscience.org/resource/>
PREFIX lsrn:<http://purl.oclc.org/SADI/LSRN/>
SELECT DISTINCT ?pubmed_id ?mut_id ?protein_property_class ?property_change_class
WHERE {
  GRAPH <http://example.com/gold-standard.rdf> {
    ?document a sio:'article';
      sio:'is subject of'
        [ a lsrn:PMID_Record;
          sio:'has attribute'
            [ a lsrn:PMID_Identifier;
              sio:'has value' ?pubmed_id ] . ] .

    ?ann_mutation a ao:Annotation;
      aof:annotatesDocument ?document;
      ao:hasTopic ?mutation .
    ?mutation a mieo:CombinedAminoAcidSequenceChange;
      sio:'has unique identifier'
        [ a mieo:CombinedAminoAcidSequenceChange_Identifier;
          sio:'has value' ?mut_id ] .
```

```

?ann_mutation_application a ao:Annotation;
  aof:annotatesDocument ?document;
  ao:hasTopic
    [ a mieo:ProteinMutationApplication;
      mieo:isApplicationOfMutation ?mutation ] .

?ann_statement_of_mutation_effect a ao:Annotation;
  aof:annotatesDocument ?document;
  ao:hasTopic
    [ a mieo:StatementOfMutationEffect;
      mieo:arg1 ?mutation_application;
      mieo:arg2 ?property_change ] .

?ann_property_change a ao:Annotation;
  aof:annotatesDocument ?document;
  ao:hasTopic ?property_change .
?property_change a ?property_change_class;
  mieo:propertyChangeAppliesTo ?protein_property .

?property_change_class rdfs:subClassOf mieo:ProteinPropertyChange .

?ann_protein_property a ao:Annotation;
  aof:annotatesDocument ?document;
  ao:hasTopic ?protein_property .
?protein_property a ?protein_property_class .
?protein_property_class rdfs:subClassOf mieo:ProteinProperty .
}
GRAPH <http://example.com/experiment.rdf> {

  ?document2 a sio:'article';
    sio:'is subject of'
      [ a lsrn:PMID_Record;
        sio:'has attribute'
          [ a lsrn:PMID_Identifier;
            sio:'has value' ?pubmed_id ] . ] .

  ?ann_mutation2 a ao:Annotation;
    aof:annotatesDocument ?document2;
    ao:hasTopic ?mutation2 .
  ?mutation2 a mieo:CombinedAminoAcidSequenceChange;
    sio:'has unique identifier'
      [ a mieo:CombinedAminoAcidSequenceChange_Identifier;
        sio:'has value' ?mut_id ] .

  ?ann_mutation_application2 a ao:Annotation;
    aof:annotatesDocument ?document2;
    ao:hasTopic
      [ a mieo:ProteinMutationApplication;
        mieo:isApplicationOfMutation ?mutation2 ] .

  ?ann_statement_of_mutation_effect2 a ao:Annotation;
    aof:annotatesDocument ?document2;
    ao:hasTopic
      [ a mieo:StatementOfMutationEffect;
        mieo:arg1 ?mutation_application2;
        mieo:arg2 ?property_change2 ] .

  ?ann_property_change2 a ao:Annotation;
    aof:annotatesDocument ?document2;
    ao:hasTopic ?property_change2 .
  ?property_change2 a ?property_change_class;
    mieo:propertyChangeAppliesTo ?protein_property2 .

  ?property_change_class rdfs:subClassOf mieo:ProteinPropertyChange .

  ?ann_protein_property2 a ao:Annotation;
    aof:annotatesDocument ?document2;
    ao:hasTopic ?protein_property2 .
  ?protein_property2 a ?protein_property_class .
  ?protein_property_class rdfs:subClassOf mieo:ProteinProperty .
}
}

```

We comment briefly on the query composition. The two halves of the query (lines 5-35 and 36-66) correspond to the selection of relevant data from the gold standard corpora and from the experimental system results. Since our goal was to select only *correct* results, the two selections were joined on the instances of the variables `?pubmed_id` (identifying documents), `?wt_residue`, `?mut_residue` and `?position_value` (for the wild-type and mutant residues, and positions of the corresponding mutations), `?protein_property_class` (identifying studied properties) and `?property_change_class` (identifying the direction of the property change).

Note that the query could only be used to implement *micro averaging* that treats the whole corpus as one large document. If, for some reason, we were interested in *macro averaging* or needed to see performance results for separate documents, we could have additionally grouped the results by the PubMed ID values.

The following SPARQL query retrieved the correct results of *Impact Sentence Recognition* (T3).

`?text_selector` identifies the fragment of text referring to an impact modelled as an instance of `ProteinPropertyChange` class. Since impact sentences in the available corpora did not have exact start and end positions, we implemented an alignment procedure (see *Utilities* section for details) to match corresponding text fragments and connect corresponding `text selectors` via instance of `StringSimilarity`. Alignment of similar text fragments were applied before running the query.

```
SELECT DISTINCT ?pubmed_id ?property_change ?text_selector
WHERE {
  GRAPH <http://example.com/gold-standard.rdf>
  {
    ?document a sio:'article';
    sio:'is subject of'
    [ a lsrn:PMID_Record;
    sio:'has attribute'
    [ a lsrn:PMID_Identifier;
      sio:'has value' ?pubmed_id ] . ] .

    ?ann a ao:Annotation;
    aof:annotatesDocument ?document;
    ao:hasTopic ?property_change;

    ?property_change a ?property_change_class .
    ?property_change_class rdfs:subClassOf mieo:ProteinPropertyChange .

    ?ann ao:context ?text_selector .
    ?text_selector a aos:TextSelector .
  }
  GRAPH <http://example.com/experiment.rdf>
  {
    ?document2 a sio:'article';
    sio:'is subject of'
    [ a lsrn:PMID_Record;
    sio:'has attribute'
    [ a lsrn:PMID_Identifier;
      sio:'has value' ?pubmed_id ] . ] .

    ?ann2 a ao:Annotation;
    aof:annotatesDocument ?document2;
    ao:hasTopic ?property_change2 .
    ?property_change2 a ?property_change_class2 .
    ?property_change_class2 rdfs:subClassOf mieo:ProteinPropertyChange .
```



```

    ?ann2 ao:context ?text_selector2 .
    ?text_selector2 a aos:TextSelector .
}
?sim a mieo:StringSimilarity .
?text_selector sio:'has attribute' ?sim .
?text_selector2 sio:'has attribute' ?sim .
}

```

Since precision and recall formulas represent relatively simple arithmetic, they can be also calculated in a SPARQL query combining the SPARQL queries calculating correct, retrieved, and relevant result sets, if this is convenient.

Utilities

As a part of our infrastructure, we created a small set of simple utilities, which facilitated access to the data:

- The *Sesame loader* and *query client* are simple command line applications that allow loading RDF graphs into a Sesame triplestore and executing queries from files.
- The *provenance enhancement* utility helps in situations when the sources of annotation data only provide fragments of texts as provenance, without specifying their positions in the text, such as in the Enzyminer and OMM Impact corpora. Note that the annotations in the OMM Impact corpus have position numbers, but, since the original text is not provided, the alignment of annotations with the original text still requires an additional simple program.

We implemented a procedure to align corresponding text fragments based on their similarity. To calculate a similarity score for two fragment candidates we use the implementation of Levenshtein distance algorithm from Lucene [30]. If fragments had differing lengths, we calculated the similarity score for their overlap. We avoided multiple alignments – each text fragment in a systems results can be aligned to the gold standard only once. Two fragments get aligned if the similarity score is above the threshold. We used a threshold equal to 0.83 to boost the precision on the Enzyminer corpus used for training to 100%. The price for this is a slight decrease in recall, which still reaches 0.99. The procedure achieved >0.99 for both precision and recall on the test corpus – the OMM Impact corpus.

Evaluation

Case study: Improving a mutation impact extraction system

In order to test the usability and validate the utility of our infrastructure, we have applied it to the testing and iterative performance evaluation of a project dedicated to the development of a robust mutation

impact extraction system [10], and to the evaluation of a mutation grounding subtask, intended for publication (see [12]). The purpose of the system was to identify protein-level mutations, ground them to the corresponding UniProt IDs and, most importantly, to extract information concerning which properties of the proteins are affected and how, if this is described in the processed document.

Since early versions of the system already produced output in RDF, modelled according to an ontology similar to MIEO, it was straightforward to adjust the system to produce output in a format compatible with our infrastructure. This was the major prerequisite to enable the evaluation of the system on our gold standard corpora and the subsequent comparison of results from different versions of the mutation grounding system.

Although the system previously showed reasonable performance on 76 documents, the performance on the larger and more representative data set comprising the Enzyminer and KinMutBase corpora, was very low. After an investigation in which we relied heavily on the analysis of system runs based on our annotations, including the provenance information, we have identified the mutation grounding module as a major performance bottleneck having only 0.32 precision and 0.08 recall. We focused our attention on the mutation grounding subtask. Our infrastructure was instrumental in this analysis because the task was also supported by the available gold standard annotations, and helped us to eventually improve the performance to 0.83 precision and 0.82 recall. More details on this effort can be found in [12].

The SPARQL negation-related features proved especially useful in this effort because they allowed us to identify *false negatives* – cases presented in gold standard and absent from system results, thus identifying potential targets for optimisation. The following query represents such a use case where the **FILTER NOT EXISTS** feature is applied to exclude correct system results for mutation grounding from the set of all results:

```
PREFIX sio:<http://semanticscience.org/resource/>
PREFIX mieo:<http://unbsj.biordf.net/ontologies/mutation-impact-extraction-ontology.owl#>
PREFIX lsrn:<http://purl.oclc.org/SADI/LSRN/>
PREFIX ao:<http://purl.org/ao/>
PREFIX aof:<http://purl.org/ao/foaf/>
SELECT DISTINCT ?pubmed_id ?wt_residue ?position_value ?mut_residue ?uniprot_record_id
WHERE {
    GRAPH <http://example.com/gold-standard.rdf> {

        ?document a sio:'article';
            sio:'is subject of'
            [ a lsrn:PMID_Record;
sio:'has attribute'
            [ a lsrn:PMID_Identifier;
              sio:'has value' ?pubmed_id ] . ] .

        ?ann_mutation a ao:Annotation;
            aof:annotatesDocument ?document;
            ao:hasTopic ?mutation .
        ?mutation a mieo:CombinedAminoAcidSequenceChange .
            sio:'has member' ?singular_mutation .
```

```

?singular_mutation a mieo:AminoAcidSubstitution;
    mieo:mutationHasWildtypeResidue ?wt_residue .
    mieo:mutationHasMutantResidue ?mut_residue .
    mieo:mutationHasPosition
[ a sio:'position';
  sio:'has value' ?position_value ] .

?ann_mutation_application a ao:Annotation;
    aof:annotatesDocument ?document;
    ao:hasTopic
[ a mieo:ProteinMutationApplication;
  mieo:isApplicationOfMutation ?mutation;
  mieo:isApplicationOfMutationToProtein ?protein ] .

?ann_protein a ao:Annotation;
    aof:annotatesDocument ?document;
    ao:hasTopic ?protein .
?protein a mieo:ProteinVariant;
    sio:'is subject of'
[ a lsrn:UniProt_Record;
  sio:'has attribute'
  [ a lsrn:UniProt_Identifier;
    sio:'has value' ?uniprot_record_id ] . ] .
FILTER (?uniprot_record_id != "")
}
FILTER NOT EXISTS {
  GRAPH <http://example.com/experiment.rdf> {

?document2 a sio:'article';
    sio:'is subject of'
[ a lsrn:PMID_Record;
  sio:'has attribute'
  [ a lsrn:PMID_Identifier;
    sio:'has value' ?pubmed_id ] . ] .

?ann_mutation2 a ao:Annotation;
    aof:annotatesDocument ?document2;
    ao:hasTopic ?mutation2 .
?mutation2 a mieo:CombinedAminoAcidSequenceChange .
    sio:'has member' ?singular_mutation2 .
?singular_mutation2 a mieo:AminoAcidSubstitution;
    mieo:mutationHasWildtypeResidue ?wt_residue .
    mieo:mutationHasMutantResidue ?mut_residue .
    mieo:mutationHasPosition
[ a sio:'position';
  sio:'has value' ?position_value ] .

?ann_mutation_application2 a ao:Annotation;
    aof:annotatesDocument ?document2;
    ao:hasTopic
[ a mieo:ProteinMutationApplication;
  mieo:isApplicationOfMutation ?mutation2;
  mieo:isApplicationOfMutationToProtein ?protein2 ] .

?ann_protein2 a ao:Annotation;
    aof:annotatesDocument ?document2;
    ao:hasTopic ?protein2 .
?protein2 a mieo:ProteinVariant;
    sio:'is subject of'
[ a lsrn:UniProt_Record;
  sio:'has attribute'
  [ a lsrn:UniProt_Identifier;
    sio:'has value' ?uniprot_record_id ] . ] .
FILTER (?uniprot_record_id != "")
}
}
}

```

Case study: Comparative evaluation of mutation impact extraction systems

To investigate the potential of the infrastructure for comparative evaluation and analysis, we adapted the Open Mutation Miner (OMM) system [9] to produce outputs compatible with our infrastructure and compared the system with the mutation impact extraction system (MIES) [10] discussed in the previous subsection. Table 3 displays functional characteristics of both systems. Previously, MIES was evaluated on

	OMM	MIES
Mutation recognition	+	+
Mutation series recognition	+	-
Mutation-protein grounding	-	+
Impact sentence recognition	+	+
Impact sentence grounding to mutation	+	+
Protein property recognition and normalization	+	+
Impact direction recognition	+	+
Physical quantity recognition	+	-
Protein property-Physical quantity grounding	+	-

Table 3: Mutation impact extraction systems.

the DHLA corpus (details in [10]) using the metrics corresponding to the task (T2) *Extraction of mutation impacts on molecular functions of proteins*. The system achieved 0.86 precision and 0.34 recall. OMM was tested on the OMM Impact corpus (details in [9]) using the metrics for (T3) *Impact sentence recognition* and (T4) *Grounding impact sentences to mutations*. The performance of OMM was 0.71 precision and 0.714 recall on the former task and 0.77 accuracy on the latter task. We undertook a cross-evaluation of the systems – MIES on the OMM Impact corpus and OMM on the DHLA corpus. Moreover, both systems were evaluated on the new Enzyminer corpus. Technically, this was achieved by loading the corpora and system results into a Sesame triplestore and running the implemented SPARQL queries, to obtain metrics, using the Sesame Workbench web interface. The results of all the experiments are shown in Table 4. Here we summarize our findings from the comparative evaluation for both systems. On the mutation impact extraction task both systems had low performance for Enzyminer and significantly better performance on the DHLA corpus. This can be explained by presence of heterogeneity in the Enzyminer corpus which has 57 different protein molecular functions (versus only 1 in DHLA) and the low performance of the current versions of both systems on the grounding of protein molecular-functions. The grounding of molecular function – normalization of molecular functions by assigning Gene Ontology classes to them – remains a very challenging task because the rich hierarchy of classes makes determining exact specific GO classes nontrivial.

Task 1: Mutation Impact Extraction (P/R)		
	Enzyminer	DHLA
OMM	0.03/0.02	0.34/0.29
MIES	0.21/0.04	0.78/0.44

Task 2: Impact Sentence Recognition (P/R)		
	Enzyminer	OMM Impact
OMM	0.59/0.32	0.63/0.52
MIES	0.15/0.10	0.23/0.04

Task 3: Impact Sentence Grounding (A)		
	Enzyminer	OMM Impact
OMM	0.58	0.72
MIES	0.85	0.68

Table 4: Mutation impact extraction systems: evaluation results (micro averaging). P/R – precision/recall. A – accuracy.

MIES shows low results on the *Impact Sentence Recognition* task. MIES impact-extraction rules were trained on the DHLA corpus of 73 mutation-impact relations and consequently failed on a corpus several times larger. OMM was trained on a larger data set and, as a result, performed relatively well on both corpora. On the *Impact Sentence Grounding* task the systems performed similarly on the OMM Impact corpus (MIES - 0.68, OMM - 0.72) and MIES performed better on Enzyminer corpus (0.85 vs. 0.58).

Future work

Our current work is focused on defining the procedures for the submission of third-party human-curated annotations and system results.

In the future, we will further stress-test the infrastructure with text mining tasks other than mutation grounding and mutation-impact extraction, and more third-party mutation text mining systems. We are planning to extend the ontology, based on the new requirements identified through community involvement and our own research. In the near future, we also plan to extend the infrastructure to include protein properties other than molecular functions, such as enzyme kinetics, and DNA-level mutations (SNPs). New corpora for mutation mention recognition – the OMM Mutation [9] and MutationFinder [31] corpora – will be integrated.

Currently the infrastructure lacks graphical representation of annotations. RDF is not easy to read by a human, so we will implement an interface to load annotated data into one of the following graphical annotation toolkits – GATE [32], UIMA [33], or BART [34]. This will enable visualization, browsing, manual modification and analysis of annotations. We will also consider leveraging the DOME tool [35]

which is, to our best knowledge, the only graphical annotation toolkit supporting RDF. It represents annotations using the Annotation Ontology RDF model and is thus compatible with our benchmarking infrastructure.

Conclusions

We have reported preliminary results on the development of a community-oriented benchmarking infrastructure intended to relieve the developers of mutation text-mining software from the burden of developing *ad hoc* corpora and scripts for testing, benchmarking and evaluation of multiple mutation-related text mining tasks. While large benchmark corpora for biological entity and relation extraction (such as CALBC [36], BioCreative [37], GENIA [38], etc.) are focused mostly on genes, proteins, diseases and species, our benchmarking infrastructure fills the gap for mutation information. We have seeded the infrastructure with a sizeable gold standard corpus (282 documents). To maximize the reusability and extensibility of our infrastructure, we use RDF and OWL for the representation of annotation data and SPARQL queries as a means of flexible analysis of text mining results. The infrastructure was tested for benchmarking and comparative evaluation of mutation-impact extraction systems. We emphasize that for performance evaluation, corpora statistics calculation, and analysis of results we did not need to write any programming code and have only used SPARQL. We have undertaken this work with the goal of *initiating a community effort*; the future evolution of the benchmarking infrastructure will be based on feedback and contributions from the community.

Availability

The benchmark corpora, the ontologies, example outputs of our mutation text mining system, benchmarking SPARQL query templates and maintenance tools are available from the project Web page [29].

Acknowledgement

This research was funded in part by the New Brunswick Innovation Foundation, New Brunswick, Canada; and the Quebec-New Brunswick University Co-operation in Advanced Education - Research Program, Government of New Brunswick, Canada.

References

1. Doughty E, Kertesz-Farkas A, Bodenreider O, Thompson G, Adadey A, Peterson T, Kann MG: **Toward an automatic method for extracting cancer- and other disease-related point mutations from the biomedical literature.** *Bioinformatics* 2011, **27**(3):408–15.
2. Bromberg Y, Overton J, Vaisse C, Leibel RL, Rost B: **In silico mutagenesis: a case study of the melanocortin 4 receptor.** *FASEB J* 2009, **23**(9):3059–69.
3. Winnenburger R, Plake C, Schroeder M: **Improved mutation tagging with gene identifiers applied to membrane protein stability prediction.** *BMC Bioinformatics* 2009, **10**(Suppl 8):S3.
4. Bauer-Mehren A, Furlong LI, Rautschka M, Sanz F: **From SNPs to pathways: integration of functional effect of sequence variations on models of cell signalling pathways.** *BMC Bioinformatics* 2009, **10**(Suppl 8):S6.
5. Baker CJO, Witte R: **Mutation Mining—A Prospector’s Tale.** *Information Systems Frontiers (ISF)* 2006, **8**:47–57.
6. Kanagasabai R, Choo KH, Ranganathan S, Baker CJO: **A workflow for mutation extraction and structure annotation.** *J Bioinform Comput Biol* 2007, **5**(6):1319–37.
7. Caporaso JG, Jr WAB, Randolph DA, Cohen KB, Hunter L: **MutationFinder: a high-performance system for extracting point mutation mentions from text.** *Bioinformatics* 2007, **23**(14):1862–1865.
8. Laurila JB, Kanagasabai R, Baker CJO: **Algorithm for grounding mutation mentions from text to protein sequences.** In *Proceedings of the 7th international conference on Data integration in the life sciences, DILS’10*, Berlin, Heidelberg: Springer-Verlag 2010:122–131.
9. Naderi N, Witte R: **Automated extraction and semantic analysis of mutation impacts from the biomedical literature.** *BMC Genomics* 2012, **13**(Suppl 4):S10.
10. Laurila JB, Naderi N, Witte R, Riazanov A, Kouznetsov A, Baker CJO: **Algorithms and semantic infrastructure for mutation impact extraction and grounding.** *BMC Genomics* 2010, **11**(Suppl 4):S24.
11. Rebholz-Schuhmann D, Marcel S, Albert S, Tolle R, Casari G, Kirsch H: **Automatic extraction of mutations from Medline and cross-validation with OMIM.** *Nucleic Acids Res* 2004, **32**:135–42.
12. Klein A, Riazanov A, Al-Rababah K, Baker CJ: **Towards a next generation protein mutation grounding system for full texts.** In *Proceedings of the 5th International Symposium on Semantic Mining in Biomedicine, SMBM’12* 2012:64–71.
13. Cohen KB, Ogren PV: **Corpus Design For Biomedical Natural Language Processing.** In *In Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases* 2005:38–45.
14. Johnson H, Baumgartner W, Krallinger M, Cohen KB, Hunter L: **Corpus Refactoring: a Feasibility Study.** *Journal of Biomedical Discovery and Collaboration* 2007, **2**:4.
15. **Track 1- BioC: The BioCreative Interoperability Initiative.**
<http://www.biocreative.org/tasks/biocreative-iv/track-1-interoperability/>.
16. Pyysalo S, Airola A, Heimonen J, Bjorne J, Ginter F, Salakoski T: **Comparative analysis of five protein-protein interaction corpora.** *BMC Bioinformatics* 2008, **9**(Suppl 3):S6.
17. Eckart R: **Choosing an XML database for linguistically annotated corpora.** *Sprache und Datenverarbeitung* 2008, **32**:7–22.
18. Demir E, et al: **The BioPAX community standard for pathway data sharing.** *Nature Biotechnology* 2010, **28**(9):935–942.
19. Ciccarese P, Ocana M, Garcia Castro LJ, Das S, Clark T: **An open annotation ontology for science on web 3.0.** *J Biomed Semantics* 2011, **2**:S4.
20. **The Mutation Impact Extraction Ontology (MIEO).**
<http://unbsj.biordf.net/ontologies/mutation-impact-extraction-ontology.owl>.
21. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nat Genet* 2000, **25**:25–9.

22. **The SemanticScience Integrated Ontology (SIO).** <http://semanticscience.org/ontology/sio.owl>.
23. **Life Science Record Name (LSRN).** <http://lsrn.org>.
24. Carroll JJ, Bizer C, Hayes P, Stickler P: **Named graphs, provenance and trust.** In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, New York, NY, USA: ACM 2005:613–622.
25. Witte R, Baker CJO: **Towards a systematic evaluation of protein mutation extraction systems.** *J Bioinform Comput Biol* 2007, **5**(6):1339–59.
26. Yeniterzi S, Sezerman U: **EnzyMiner: automatic identification of protein level mutations and their impact on target enzymes from PubMed abstracts.** *BMC Bioinformatics* 2009, **10**(Suppl 8):S2.
27. Forbes SA, Tang G, Bindal N, Bamford S, Dawson E, Cole C, Kok CY, Jia M, Ewing R, Menzies A, Teague JW, Stratton MR, Futreal PA: **COSMIC (the Catalogue of Somatic Mutations in Cancer): a resource to investigate acquired mutations in human cancer.** *Nucleic Acids Res* 2010, **38**(Database issue):D652–7.
28. Ortutay C, Väliäho J, Stenberg K, Vihinen M: **KinMutBase: a registry of disease-causing mutations in protein kinase domains.** *Hum Mutat* 2005, **25**(5):435–42.
29. **Mutation text mining benchmarking infrastructure.** <http://code.google.com/p/mutation-text-mining>.
30. McCandless M, Hatcher E, Gospodnetic O: *Lucene in Action, Second Edition: Covers Apache Lucene 3.0.* Greenwich, CT, USA: Manning Publications Co. 2010.
31. **The corpus developed for the evaluation of the MutationFinder tool.** <http://sourceforge.net/projects/bionlp-corpora/files/ProteinResidue/MutationFinder-1.1-Corpus.tar.gz>.
32. Cunningham H, Maynard D, Bontcheva K, Tablan V: **GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications.** In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA* 2002.
33. Ferrucci D, Lally A: **UIMA: an architectural approach to unstructured information processing in the corporate research environment.** *Nat. Lang. Eng.* 2004, **10**(3-4):327–348.
34. Stenetorp P, Pyysalo S, Topić G, Ohta T, Ananiadou S, Tsujii J: **brat: a Web-based Tool for NLP-Assisted Text Annotation.** In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics 2012:102–107.
35. Ciccarese P, Ocana M, Clark T: **Open semantic annotation of scientific publications using DOME0.** *J Biomed Semantics* 2012, **3** Suppl 1:S1.
36. Rebholz-Schuhmann D, Yepes AJJ, Van Mulligen EM, Kang N, Kors J, Milward D, Corbett P, Buyko E, Beisswanger E, Hahn U: **CALBC silver standard corpus.** *J Bioinform Comput Biol* 2010, **8**:163–79.
37. **The BioCreAtIvE challenge evaluation.** <http://biocreative.sourceforge.net>.
38. Kim JD, Ohta T, Tateisi Y, Tsujii J: **GENIA corpus—semantically annotated corpus for bio-textmining.** *Bioinformatics* 2003, **19**(Suppl 1):i180–2.