Journal of Bioinformatics and Computational Biology
© Imperial College Press

TOWARDS A SYSTEMATIC EVALUATION OF PROTEIN MUTATION EXTRACTION SYSTEMS

RENÉ WITTE

Universität Karlsruhe (TH)
Institut für Programmstrukturen und Datenorganisation (IPD)
Am Fasanengarten 5, 76128 Karlsruhe, Germany
witte@ipd.uni-karlsruhe.de

CHRISTOPHER J. O. BAKER

Data Mining Department, Computing Division
Institute for Infocomm Research (I^2R)
21, Heng Mui Keng Terrace, Singapore 119613
cbaker@i2r.a-star.edu.sg

Received (25 July 2007)
Revised (20 September 2007)
Accepted (30 September 2007)

The development of text analysis systems targeting the extraction of information about mutations from research publications is an emergent topic in biomedical research. Current systems differ in both scope and approaches, which prevents a meaningful comparison of their performance and therefore possible synergies. To overcome this “evaluation bottleneck,” we developed a comprehensive framework for the systematic analysis of mutation extraction systems, precisely defining tasks and corresponding evaluation metrics that will allow a comparison of existing and future applications.

Keywords: mutation extraction systems; mutation evaluation tasks; mutation evaluation metrics

1. Introduction

Mutations of genes or proteins, either single nucleotide polymorphisms (SNPs) or point mutations, are described in an ever-increasing number of publications appearing on a daily basis. Their impacts can have far-ranging consequences in medical, agricultural, and industrial domains, making them an important target of knowledge acquisition processes. As manually curated databases, like the Protein Mutant Database (PMD),[1] cannot keep up with the high number of publications describing mutations, the development of tools and techniques for the automatic extraction of information about mutations is becoming an emergent research field. Such systems, typically based on text mining and natural language processing (NLP) techniques,[2–5] require an *evaluation* that both shows their suitability and allows a comparison with other approaches.

Large-scale evaluation efforts are well-known in the area of NLP in general and biomedical text mining in particular.[6] One of the major community-based efforts in recent years for the evaluation of NLP systems targeting the biology domain has been the *Critical Assessment of Information Extraction systems in Biology*[7] (BioCreAtIve)[a] challenge. While mutation analysis has not been a task in BioCreAtIve thus far, the general framework of how to address NLP evaluations in this domain remains applicable. An important point of these competitions is the development of a common, annotated corpus, also known as *gold standard*, that allows the comparative evaluation of different systems by running them on the same data.

Evaluation challenges, however, go beyond simply using a common dataset. Even a seemingly simple task like "mutation entity detection" can differ greatly in its scope as implemented by concrete systems. Comparing results from one system that includes, e.g., *normalization* or *grounding* with a system that does not will not result in an insightful analysis. Thus, a framework is needed that additionally includes the definition of *tasks* and *metrics*.

In this paper, we provide such a framework that includes the essential tasks underlying systems that have been designed to extract mutations and mutation annotations from the biomedical literature. In so doing we aim to bring transparency and maturity to an important and quickly evolving application domain. Primarily, we contribute two important aspects: *(i)* An analysis and precise definition of tasks performed by various systems in the mutation analysis domain and *(ii)* the definition of metrics for each task that allow for a systematic evaluation of a system.

Moreover, we take a holistic position whereby we discuss not only system tasks; but additionally pre-analysis information retrieval, post information extraction tasks such as mutation impact summarization and analysis, annotation export to third party clients, question-answering, and curation issues. By including references to these topics we also seek to highlight the many inter-dependencies that exist within systems designed to derive actionable knowledge from latent repositories of mutation information.

We see the development of the evaluation framework presented in this paper as a starting point for a future community-wide effort in this area. Developing annotation guidelines, selecting suitable data sets, developing a corpus, annotating texts, and setting up a competition will be the next logical steps. All of these require significant further investments. Clarifying the formal dependencies and practical challenges in this process, which go far beyond typical text mining evaluations, is the main motivation behind this critical commentary and forward-looking analysis.

2. Evaluation Tasks

We begin setting up our evaluation framework by providing precise definitions of a number of tasks performed in the mutation analysis domain. These tasks are complemented by corresponding metrics, defined in Section 3, which will allow for their fully automatic evaluation.

The tasks have been categorized taking a number of constraints into consideration.

[a] BioCreAtIve, `http://biocreative.sourceforge.net/`

Firstly, we do not expect that every system will perform each of these tasks. Rather, many systems might implement only a single one or a subset of these tasks, for example, only the recognition of mutation entities. Hence, the tasks have been modularized in such a way that more complex systems can be evaluated and assembled through a component-based approach of more basic tasks. At the other end of the spectrum, we define *system tasks* that aim to capture the final product delivered by a system, which is of particular interest to an end user. Secondly, we examine a number of existing systems and their analysis goals, in order to provide an evaluation framework that is meaningful to a large number of stakeholders. And thirdly, the tasks must allow for a meaningful and fully automated evaluation, given that a manually annotated gold standard is developed.

For each task, we provide a precise definition together with a brief motivation, as well as examples of current systems performing the task. The tasks are roughly ordered in increasing difficulty for current NLP systems. More advanced tasks generally require successful completion of the more basic tasks. For instance, a protein or gene can only be successfully grounded if it has been correctly identified. However, it is important to stress that the following tasks evaluate system *results*, independently of how a concrete system internally arrives at a solution. Hence, these tasks must not be confused with *approaches* implemented in a system: We describe the evaluation of end products, not the process of obtaining them.

2.1. *Entity Recognition Tasks*

The detection of *named entities* (NEs) is one of the basic steps performed by an NLP system. Hence, we first cover the detection of a number of entities fundamental for the mutation domain. This so-called *entity recognition* (ER) is a classical NLP task and its evaluation is well-understood. However, within the mutation domain, ER provides only the first level of analysis needed before applying additional analysis steps that are more biologically motivated, discussed in the following subsections.

The types of entities that are recognised in mutation extraction systems include genes, proteins, organisms, and mutations, which can take the form of experimentally introduced point mutations, DNA insertions or deletions, and naturally occurring single nucleotide polymorphisms.

ER-1: Protein Recognition. *Identify each occurrence of a protein name in a document, including abbreviations, as well as nominal and pronominal references.*

Protein recognition is one of the most widely performed NLP tasks in the biomedical domain, due to a number of evaluations that included protein recognition performance within more complex tasks, like the detection of protein-protein interactions. A number of protein taggers are publicly available, which makes it possible to provide baselines for evaluation.[b]

[b] See the list provided under `http://biocreative.sourceforge.net/bionlp_tools_links.html` for a number of examples.

Note that this task targets the detection of entities only, i.e., not mentions of higher-level concepts. Thus, if an occurrence of, e.g., "phosphatase" in a text refers to the family and not an individual protein, it must not be annotated as such in the gold standard. However, an extended named entity task could target those occurrences as well, e.g., by populating concepts in an ontology (”protein family”) with their mentions in a text.

While protein name recognition is challenging by itself, the inclusion of abbreviations adds another level of complexity to this task.[8] In the mutation literature in particular, authors often introduce their own abbreviations within a paper. Note that we aim to cover these in conjunction with entity recognition, i.e., we do not propose a separate "abbreviation recognition" task. Since the problem of abbreviation occurs for all kinds of named entities, our approach is to cover them within each task, but introduce additional markup within the gold standard to identify the type of reference (normal, abbreviated). This allows to perform evaluations of either lexical form separately, as well as combining both, based on the same dataset.

A similar problem occurs with nominal or pronominal references (e.g., "this protein," "it"). These become particularly important when impact information needs to be extracted from texts (see Section 2.5). Similar to abbreviations, we aim to include pronominal references, provided they are clearly labeled in order to distinguish them during evaluation. Marking these additional references at the same time as the named entities themselves makes it possible to perform coreference evaluations later on without additional re-annotation efforts, even when they are not included in a first competition.

ER-2: Organism Recognition. *Identify each occurrence of an organism in a text, including abbreviated forms and nominal and pronominal references.*

Similar to protein recognition, organism recognition has to mark every occurrence of an organism name in a document, including common (e.g., "arsenic fungus" for *Scopulariopsis brevicaulis*) and abbreviated forms (e.g., *C. fimi* for *Cellulomas fimi*). This includes genus, species, and strain information, if these parts are present. By including additional markup for these taxonomical units we provide for more precise entity recognition evaluations, as well as more advanced querying of the results (e.g., restricting the search space to a certain "genus"), while adding only minimal load to the developers of the gold standard.

Whereas organisms produce a range of proteins, each protein ID in the UniProtKB is linked to exactly one organism ID. Systems that perform the normalization and grounding tasks defined below thus can derive additional information for protein disambiguation using organism information detected in documents. Since such an "organism tagger" is a useful stand-alone component that can also be embedded in other bio-NLP tasks, it is defined here as a distinctive task. An example of a system performing organism detection is the online available EBIMed.[c]

ER-3: Mutation Recognition. *Detect each description of a mutational change described*

[c] EBIMed, `http://www.ebi.ac.uk/Rebholz-srv/ebimed/index.jsp`

in a document, including abbreviated and full-text forms, as well as nominal and pronominal references. Distinguish between single-point mutations and mutation series, as well as insertions or deletions.

Finding mutation expressions in texts is obviously one of the core tasks in the mutation domain. However, the definition of what precisely constitutes an individual mutation entity is even more difficult than for a protein or organism entity, due to the high number of variations possible for describing mutational changes performed on a protein. These include the more standardized single-letter (e.g., "A"), three-letter (e.g., "Ala"), and full (e.g., "Alanin") amino acid descriptions, but also graphical representations, tables, mutations described on the DNA level, as well as full-text narratives ("We changed the amino acid…"). See, e.g., Rebholz-Schuhmann et al.[9] for a number of examples. Especially with regard to full-text descriptions, it can be difficult to decide which part of a sentence to mark as the actual mutation in the gold standard. This can be alleviated by computing precision/recall values with partial overlaps as discussed in the evaluation section below; however, clear annotation guidelines still need to be developed for this case.

Additional variations are introduced by mutations that *insert* or *delete* part of a sequence. Here, the recognized mutation entity must be supplemented by a feature marking it as such a deletion or insertion, which will allow evaluations based both on the detection performance alone (not including the features) and additionally including the kind of the mutation (by including the feature when comparing the detected entity with the gold standard).

Another level of complexity is added when researchers report the combined impacts of an experimentally introduced mutation series, manifesting themselves as a single mutant phenotype: These must not be confused with a series of unrelated single-point mutations, since a characteristic of the mutated protein might only be reported or manifest itself when *all* mutations within such a series are present. An example of this kind of mutation is:[d]

> Introduction of arginines to the Ser/Thr surface (ST series)

ST1	S186R
ST2	S186R, N67R
ST3	S186R, N67R, T26R
ST4	S186R, N67R, T26R, Q34R
ST5	S186R, N67R, T26R, Q34R, S40R
ST6	S186R, N67R, T26R, Q34R, N69R

> …
>
> *"The hydrolysis experiments revealed that the five-arginine mutants (ST5 and ST6) had an increased thermostability in the presence of the substrate. When the hydrolysis of xylan was followed as a function of time, the mutants ST5 and ST6 retained their activity at 65°C and pH 5 remarkably better than the wild-type XYNII."*
>
> *"The mutants with three (ST3) or four arginines (ST4) did not exhibit improved stability in the presence of the substrate at 65°C compared to the wild-type XYNII."*

[d] From: Turunen O, Vuorio M, Fenel F, and Leisola M., "Engineering of multiple arginines into the Ser/Thr surface of Trichoderma reesei endo-1,4-β-xylanase II increases the thermotolerance and shifts the pH optimum towards alkaline pH." *Protein Engineering* vol.15 no.2 pp.141–145, 2002. PMID: 11917150

Thus, when introducing "mutation series" into this evaluation task, a system must then also distinguish the detected mutation type (single/series). This can be achieved by including an annotation feature labeling it as a mutation series, assigning a unique ID as a feature, which can then be referenced from other mutations within the series. Finally, each detected mutation must be annotated as belonging to either the nucleotide ("DNA") or amino acid level ("AMINOACID").

In practice, full-text papers converted from binary formats for processing, such as PDF, often exhibit corruptions in formatting or of special characters (e.g., "54G→T"). However, we will not address such problems within this evaluation. We see them as a belonging to a separate data preparation stage and presume a corpus with (semi-)automatically or manually cleaned markup, as was the case in the KDD Challenge Cup 2002.[e]

An example for a mutation tagging component is the MutationFinder system by Caporaso et al.[10] Since it is freely available under an open source license[f], it can be used as a baseline system for this task. In order to improve the performance, systems might want to include additional strategies, like a disambiguation of gene/amino acid mutations or sequence verification.

2.2. *Normalization Tasks*

While the named entity and relation recognition tasks are typical NLP tasks, the following tasks are becoming more specific to the biology domain in general and protein mutations in particular. *Normalization* assigns a canonical name to each entity, allowing an identification independent of its lexical form. This is particularly important for abbreviated entities. Normalization is often conflated with the *grounding* tasks (see the next subsection), as system implementations often combine both. However, as they are conceptually different tasks producing different results (lexical form for normalization vs. database reference for grounding), we also cover them separately here as well. Note that a system performing the grounding tasks described in the next subsection can trivially solve the corresponding normalization task as well. But mandating the grounding task would exclude systems from the evaluation that perform normalization on a purely lexical basis, without using external databases.

N-1: Protein Normalization. *Assign a canonical name to each of the protein entities detected in ER-1.*

This task assigns each protein entity detected before its full, canonical name. This requires the definition of a set of reference names, e.g., by using the ones provided by the UniProtKB[11] under "Protein name" (stored in record `DE`). Finding the correct canonical name will typically require some kind of disambiguation, as a protein name like "phosphatase" detected in a document can reference a multitude of distinct protein sub-families in

[e] Note that the preparation of a cleaned corpus from Web- and other documents is an additional task evaluated by the NLP community, e.g., within the Cleaneval effort for Web data (`http://cleaneval.sigwac.org.uk`).

[f] MutationFinder, `http://mutationfinder.sourceforge.net`

UniProtKB. Before further automated processing of NLP detected entities can take place, a precise identification of the protein entities is mandatory.

N-2: Organism Normalization. *Assign a canonical name to each of the organism entities detected in ER-2.*

Here, each organism mention detected by ER-2 is assigned the full, canonical name. We suggest that this should be the name recorded by the main reference for organisms, the NCBI taxonomy database,[12] under "scientific name." Note that this name might be again very different from the lexical form appearing in a document; e.g., the organism name *Vibrio subtilis* detected in a text would be assigned the canonical (scientific) name *Bacillus subtilis*. In fungal taxonomies, for example, there is a dual nomenclature, which derives from anamorph/teleomorph dichotomy between sexual and asexual states. Although the NCBI taxonomy database does include strain level information for some organisms, we do not aim to include them in this task as a systematic normalization is currently unfeasible.

N-3: Mutation Normalization. *Assign a canonical name to each of the mutation entities detected in ER-3.*

Each of the mutations detected by ER-3 is subsequently assigned a canonical name, by normalizing their descriptions to a single format. Here, we propose to use the format describing a mutation using the single-letter amino acid code of the wild type, followed by the residue position and its mutated amino acid form (e.g., "E75D"). Thus, in a given corpus, all other formats occurring in a document must be converted by a system and stored in the normalized format within a feature of the mutation annotation. A mutation phenotype comprised of a mutation series will have a list of normalized single-point mutations. Insertions are annotated as such *("Ins")* and normalized to the form "*residue–residue* POS–POE *inserted-residues-list*" (with POS as the starting position of the insertion and POE the end position). Similarly, deletions are annotated *("Del")* and normalized to either "*position residue*" (single deletion) or "POS–POE *residues-list*" (deletion range) indicating the deletion.

Note that the manually annotated gold standard should only contains mutations normalized to the amino acid level. Hence, a lexically ambiguous form like "A23C," which could also refer to a gene mutation, is not ambiguous in the gold standard. Mutations that have been marked as DNA level in ER-3 are not further normalised in this step.

2.3. *Relation Detection Tasks*

Relation detection links the recognised entities discussed above with each other. This is typically performed within an NLP system using a technique like deep syntactic analysis (full or partial parsing), heuristics (e.g, based on entity distance), or some statistical method.[13–15] Here, we only address binary relations that have been most frequently mined by existing systems. Additional mutations that affect gene regulation and their indirect consequences are not covered within the scope of this proposal, but could be added in future evaluations.

Our definition of relations is based on *normalized* entities. Hence, we can define the task more precisely than would be the case for lexical relations occurring in texts (for example, with pronominal references). Thus, the gold standard for relations is not based on their lexical expressions in a text, instead it is represented as a list of entity pairs, identified by their canonical name. For example, a particular protein↔mutation relation will be captured as a single tuple consisting of the normalized protein name and the normalized mutation name, no matter how often their relation is mentioned in a text, and in what form (e.g., using a pronoun). Such a gold standard can be developed much more rapidly and reliably than annotating each lexical relation in a text; the disadvantage is that systems not performing the normalization task yet still detecting relations cannot be evaluated within this task.

Note again that we do not prescribe a certain order in which a system has to perform the tasks of entity recognition, normalization, grounding, and relation detection, but only evaluate their end products, as discussed above. In fact, these tasks are interrelated and a system can make use of their semantic interdependencies, e.g., through restricting the space of possible solutions given external resources like an ontology or database.[16,17]

REL-1: Protein-Organism Relation Detection. *For each normalized protein, assign its source organism, if present in the document.*

This relation identifies the source organism for each protein. The result can either be recorded using a feature on each protein annotation, providing the normalized organism name, or by producing a list of $(P_{\text{name}}, O_{\text{name}})$ tuples with the normalized names.

REL-2: Protein-Mutation Relation Detection. *For each normalized mutation, identify the protein it is modifying.*

Here, a system needs to identify the protein that was changed by a mutation. The result will be represented analogously to the REL-1 task. Obviously, this task is essential if more than one protein is mentioned within a publication, which is almost always the case when analysing full-text papers.

2.4. *Grounding Tasks*

Grounding cross-links entities detected in documents with their real-world counterparts. This is an important prerequisite for further automated analyses, where bioinformatics algorithms are executed on the results of a text mining system. For these tasks, we rely on databases commonly used in the biological domain, in particular the UniProtKB for proteins, the NCBI Taxonomy database[12] for organisms, and the Protein Data Bank (PDB)[g] or Protein Mutant Database (PMD)[h] for mutations. Note that these tasks are much harder for a system than basic entity detection, since grounding typically requires more advanced disambiguation strategies.

GR-1: Protein Grounding. *Assign the correct UniProtKB ID to each detected protein*

[g] RCSB Protein Data Bank, `http://www.pdb.org/`

[h] Protein Mutant Database (PMD), `http://pmd.ddbj.nig.ac.jp/`

entity.

Protein grounding involves the correct identification of a database identifier for a (normalized) protein entity. This can be achieved by assigning a UniProtKB identifier to each detected protein. Grounding is important as any higher-level analysis tasks executed on text mining results requires additional information from these databases, like a protein's sequence. An example of a system performing this task is BioAR.[18]

GR-2: Organism Grounding. *Assign each organism entity its corresponding ID within the NCBI Taxonomy database.*

Similarly to protein grounding, organism grounding requires the correct assignment of a database ID to each detected organism entity. Here, we highlight the suitability of the NCBI taxonomy database,[12] which is also cross-referenced from UniProtKB entries, allowing an automated evaluation of the detected Protein↔Organism relations based on their detected ID information. This kind of grounding task is performed by, e.g, the Mutation Miner[17,19] system.

GR-3: Mutation Grounding. *Verify and if necessary positionally correct each mutation location to match its corresponding protein's sequence as obtained from UniProtKB.*

Mutation grounding requires the validation of the detected and normalized mutational entity on the protein sequence obtained from UniProtKB, based on the protein identified by Task REL-2 and its ID obtained through grounding (Task GR-1). Both for single-point mutations and mutation series, this may require the correct identification of the difference between the numbering used by the authors and the sequence as obtained from the UniProtKB. While for multiple mutations on a single protein and mutation series the legitimacy can be more easily verified using the offsets between mutations, this is more difficult for a single mutation, as a match of the amino acid of the mutation on the sequence might be purely coincidental. Examples of systems performing this task are MuteXt[20] for single mutations and Mutation Miner[21] for multiple mutations.

2.5. *Impact Analysis Tasks*

The tasks so far have only detected that a particular mutational change to a protein happened—but not what the consequences of that change were. However, an end user, like a protein engineer or biomedical scientist, requires more information about the *impact* of a change. Thus, systems for the analysis of mutation papers additionally need to extract impact information, requiring an evaluation strategy for these functions as well. Example sentences containing impact knowledge are:[i]

"The mutations at the C-terminus of the α-helix, Q162H, Q162Y, Q162L and

[i] All sentences from: Turunen O, Etuaho K, Fenel F, Vehmaanperä J, Wu X, Rouvinen J, and Leisola M., "A combination of weakly stabilizing mutations with a disulfide bridge in the α-helix region of *Trichoderma reesei* endo-1,4-β-xylanase II increases the thermal stability through synergism." *J. of Biotechnology*, 2001 June 1;88(1):37–46. PMID: 11377763.

> *Q162K, increased the half life of XYNII at 55°C (pH 5) to 36, 39, 26 and 11 min, respectively."*
>
> *"Q162H, Q162Y and Q162L did not show any stabilizing effect at 65°C (half-lives < 1 min)."*
>
> *"The mutations N11D and N38E did not have any significant effect: N11D increased the half life scarcely 1.5 times at 55°C, and N38E about 1.5 times at 57–60°C."*

IA-1: Unstructured Impact Analysis. *For each mutation obtained in GR-3, extract all sentences describing the impact of that mutation.*

For this task, we simply regard sentence-level information containing impact information as shown above. That is, for each mutation all sentences containing some kind of impact description will be marked up. These can then be presented to the user, e.g., in form of a table or full-text summary. That is, no further analysis or structuring of the impact takes place within this task.

IA-2: Structured Impact Analysis. *For each mutation obtained in GR-3, determine its impact according to a set of prescribed dimensions.*

While the unstructured impact task, delivering sentence extracts, is suitable for human end-users, it is not very useful for further automated processing of mutational information. For instance, when connected with a 3D-visualization, a user might want to filter out all mutations where the authors reported no impact on the protein's properties, or choose to display only those that change the thermostability of the wild-type protein. This requires that each mutation is annotated with respect to a number of dimensions, like *impacted property* and *measurement* information. We present a possible categorization within the system task SYS-3 below.

Note that a system performing this task can also trivially produce results for IA-1, by simply keeping track of the sentences where a certain impact was detected.

2.6. *System Tasks*

System tasks represent the actual end products delivered by a mutation analysis system. These tasks can be directly used to compare the performance of individual systems. However, due to their complex nature, system tasks give relatively little insight into specific challenges within the whole analysis process. For example, two systems might perform similar to each other within task SYS-1, whereas one system excels in GR-1 and GR-2 and another in GR-3. Without the evaluation of the more fine-grained tasks outlined above, it will not be possible to detect the opportunity of combining the complementary approaches of both systems to obtain an even better performance.

SYS-1: Extraction of Relevant Protein-Mutation Tuples. *Extract all distinct, relevant (protein, mutation) pairs from a document, where the protein is identified by its UniProtKB ID (Task GR-1) and the mutation is normalized and grounded according to Tasks N-3 and*

GR-3.

The main system task in the mutation domain is the detection of protein-mutation tuples. Each pair describes a relevant mutation (single or series) performed on a certain protein, produced by a specific organism. The keyword here is *relevant*, which puts further restrictions on the number of all possible protein-mutation tuples that can be extracted from a document. This is due to the variety of information contained in full-text research papers—as opposed to just processing the abstract. We see three major types to distinguish:

Primary Mutations (Type A): These are (P,M)-tuples concerning results of the mutational experiments described in a publication.

Secondary Mutations (Type B): Results from other mutational studies typically used for comparing the results of an experiment with previous results. These often appear in a "Discussion" section of a paper.

Spurious Mutations (Type C): These are tuples that do appear in a paper, but are neither part of the performed experiments nor their discussion. These can occur, for example, in the list of references, where the paper itself is cited for some other reason (e.g., the experimental setup).

Distinguishing these tuples is important for a correct attribution of provenance information: Firstly, only the Type A mutations should be exported as the result for a specific publication, since all other types are not supported by the paper itself, but rather by a different publication. However, it is still helpful to identify these, e.g., for subsequent cross-validation with the original publications describing these experiments.

Note the similarity with the KDD Challenge Cup 2002, where genes had to be automatically curated from articles. Like with mutations, not all mentions of a gene occurring in a paper were curated, rather, the decision was based on experimental evidence as outlined in curation guidelines. An observation from the evaluation showed that simple bag-of-word approaches that did not perform more involved analyses performed rather poorly in this task.[6] Proper mutation type detection will generally require some kind of text tiling strategy, e.g., by assigning semantic labels to text segments like introduction, discussion, and list of references.

Formally, we can represent each tuple using the UniProtKB ID for a protein and the normalized and grounded mutation descriptor as described for GR-3 above.

SYS-2: Impact Summarization. *For each of the tuples detected in SYS-1, identify the impact of the mutational change with respect to the wild-type protein. The result can be of any form, but should contain all the knowledge stated in a source concerning the impact.*

The goal of this task is to identify impact-relevant information and present them to the user (cf. Task IA-1 in Section 2.5). This can take the form of sentence extracts, as is commonly done for automatic summarization. The motivation behind is that the end user can then quickly scan the extracted sentences for each mutation and assess its impact. A real-world system would cross-link each sentence with its original publication, allowing the user quick access to the primary source.

SYS-3: Structured Impact Description. *Generate a structured representation for each of the mutations detected by SYS-1 that includes information on the impact, its affected property, as well as any measurements given in the publication.*

This tasks further extends SYS-2: a system must detect impact information and extract them into a structured format. This could take the form of an ontology, where instances are created for a number of pre-defined classes, or simply a number of slots filled within a (XML-based) template. Similar to annotation guidelines for human users, a clear set of categories must be established before this task can be used for evaluating systems, e.g., in the form of an ontology[17] or XML Schema. Relevant information, typically found in mutation databases such as HGMD[j] or PMD,[k] includes:

Impact: either a *Measurable Impact* or a *General Impact*. For the first kind, the system must detect whether it is of the kind *Increase, Decrease,* or *Unchanged*.

ProteinProperty: what property has been changed in the protein? The system must identify a number of pre-defined options, such as *Catalysis* or *Activity*.

FeatureMeasurement: the precise unit of measurement and its value as described in a publication, e.g., in *DegreesCelcius, HalfLife (kCat), pH,* or *percentage*.

DiseasePhenotype: the traits or characteristics of a medical condition that are the consequence of mutational impact, which can originate from one or more level of a biological system, e.g., *gene, transcript, protein, metabolic, or signalling pathway*.

SYS-4: Impact Analysis *Provide an analysis of the mutations described in a publication, by comparing them with the properties of the wild-type version of a protein.*

For an end user, a mutational impact alone might not be useful, when the corresponding baseline information is not known. Often, these are only implicitly contained in full-text papers, when the authors only discuss the *changes* to the wild-type version, but not its original properties.

Ideally, a mutation analysis system will aid the user by displaying the corresponding properties of the wild-type protein alongside the extracted mutational information. Thus, a system will need to detect if this information is present and extract it alongside the impact information (Task SYS-3) or access external databases like BRENDA[22] or SABIO-RK,[23] using the grounded protein identifiers discussed above.

2.7. *Additional Tasks*

The tasks defined so far represent the core challenges in the mutation text analysis domain. However, this list is non-exhaustive: Complex mutation analysis systems often perform additional sub-tasks. An explicit evaluation of these tasks is interesting for system developers or even end users, in case they are directly affected by their performance. But each new

[j] Human Gene Mutation Database (HGMD), `http://www.hgmd.cf.ac.uk/ac/index.php`

[k] Protein Mutant Database (PMD), `http://pmd.ddbj.nig.ac.jp/`

task also adds to the burden of the developers of the gold standard and the organizers of an evaluation competition, which is is why we see the following tasks, while highly relevant, as being secondary tasks that can be addressed at a later point in time:

Information Retrieval. Before knowledge can be extracted from documents, they have to be found and retrieved first. While some systems already target the extraction of knowledge from the complete set of abstracts available through MEDLINE,[l] so far we know of no attempt to cover the full text of all mutation-related papers available through PubMed.[m] Thus, the integration of a mutation analysis system into a bioinformatics discovery pipeline (e.g., using Taverna[n,24]) still requires a prior information retrieval (IR) step, which can be evaluated using the standard IR measures (precision, recall, f-measure, precision@10, MRR).

Coreference Resolution. Coreference chains explicitly link the various textual descriptors of a single entity. These include pronominal references (e.g., “it” referencing a previously mentioned protein), nominal references (e.g., “this organism”), as well as abbreviations. Note that we already included these references with the entity detection tasks described in Section 2.1. Together with the additional information needed for grounding, an automatic evaluation of coreference chains will be possible without the need for developing a new gold standard. This evaluation could be used for either stand-alone coreference resolution systems or algorithms integrated in mutation analysis system. While a number of existing approaches already perform coreference resolution in the biology domain,[18,25–27] this task has so far not been evaluated within a competition.[o]

Export Tasks. Once systems have reached a documented level of maturity, integrating their results into productive applications is the next logical step. Unless a system’s output is meant for direct consumption by a human user, the generation of a structured result format is an additional step, which can introduce further error cases. Export tasks include, among others, the mapping of mutations to a structural homolog for 3D-visualization (as done by the Mutation Miner system[21]), the annotation of UniProtKB entries with new references (similar to a system like ProFAL[28]), the population of an OWL ontology,[17] or the automatic curation of new entries for the Protein Mutant Database (PMD).[1]

Question-Answering. An emerging application area of text mining systems is to provide answers to questions posed in natural languages—so-called question-answering (QA). Here, the results returned by a system have not only to be correct, but also pertinent to the question at hand. This requires as additional steps an analysis of the question(s), as well as the

[l] MEDLINE, `http://medline.cos.com/`
[m] PubMed, `http://www.ncbi.nlm.nih.gov/sites/entrez?db=pubmed`
[n] Taverna, `http://taverna.sourceforge.net/`
[o] However, coreference resolution has previously been evaluated in the general news domain, e.g., within the *Message Understanding Conference* (MUC) series of competitions.

generation of an answer, e.g., based on an ontology.[19] Note that the TREC Genomics Track 2007[p] also includes mutations as a question entity.

3. Evaluation Metrics

In this section, we discuss the metrics necessary for an automatic evaluation of the tasks defined above. For each of these evaluations, we see the need for carrying out two distinctive runs: one for systems running on abstracts only, and one for full texts.

3.1. *Precision, Recall, and F-Measure*

The metrics *precision, recall* and *F-measure* are commonly used in the evaluation of NLP systems. They have been adapted from their Information Retrieval (IR)[29] counterparts. They are based on the notion of entities that have been correctly found by a system *(correct)*, entities that are *missing*, and entities wrongly detected by a system, i.e., *spurious* entities or false positives. To capture partially correct results, i.e., entities where gold standard and system response overlap without being coextensive (matching 100%), a third category *partial* is introduced in addition to correct and spurious results.

Following the definition implemented by the GATE annotation evaluation tool,[q] we can define *precision* as the ratio of correctly detected entities over all retrieved entities:

$$\text{Precision} = \frac{\text{Correct} + \alpha \cdot \text{Partial}}{\text{Correct} + \text{Spurious} + \alpha \cdot \text{Partial}} \tag{1}$$

The credit for partially correct entities can be adjusted through the parameter α. Note that $\alpha = 0$ results in the classical definition; the default configuration in the GATE evaluation tool is $\alpha = \frac{1}{2}$. An evaluation should be performed with various settings of this parameter to analyse boundary problems in NE detection. Sometimes *error rate* is used instead of precision, which is simply defined as $1/\text{Precision}$. Similarly, *recall* is the rate of correctly detected entities over all correct entities:

$$\text{Recall} = \frac{\text{Correct} + \alpha \cdot \text{Partial}}{\text{Correct} + \text{Missing} + \alpha \cdot \text{Partial}} \tag{2}$$

The F-measure combines both precision and recall using their harmonic mean:

$$\text{F-Measure} = \frac{(\beta^2 + 1)\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Recall}) + \text{Precision}} \tag{3}$$

Here, β is an adjustable weight to favour precision over recall (with $\beta = 1$, both are weighted equally). These measures can be applied directly to all entity recognition tasks (ER-1,2,3). A variation for measuring the partially correct results is to require a matching boundary on either side of an entity, which provides additional results for left-bounded/right-bounded evaluation.[6] This can be useful if the entity under consideration has known boundary issues, e.g., for organisms with/without trailing strain information. To evaluate additional

[p] TREC Genomics Track 2007 Protocol, http://ir.ohsu.edu/genomics/2007protocol.html
[q] GATE documentation, `http://gate.ac.uk/documentation.html`

annotation features, such as those stipulated for the entity recognition tasks in Section 2.1, the comparisons for correct/missing entities can be based on specific feature types, e.g., only non-pronominal references, only abbreviated entity names, and so on. This allows for a flexible evaluation from a single dataset.

Applying these measures to the relation detection tasks (REL-1,2) requires a somewhat different setup from the standard NLP evaluation: instead of looking at lexical entities, we consider tuples (pairs) of normalized names, which may or may not occur in the documents themselves. Thus, the evaluation is performed by matching a list of these pairs (from the gold standard) with a system's response. Here, we can again apply the measures for precision, recall, and F-measure over correct, missing, and spurious tuples. Note that in this case there is no overlap (partially correct results), as we are not comparing document offsets (lexical forms) but rather unique tuples.

Likewise, the system task evaluations also do not focus on lexical occurrences, but rather on the underlying concepts. That is, instead of examining *every* entity occurrence in a document, we only look at each *unique* mention of an entity. Thus, we measure directly how successful the system was at retrieving mutational information from a text. The reasoning is that these results are more interesting for end users who are thinking about adopting a certain tool or technique within a real-world knowledge acquisition process. For this evaluation, we need a gold standard containing the correct results of each type (A, B, C; as discussed in Section 2.6) for a document under analysis. Using the (unique) tuples represented in both the gold standard and system response, we can compute precision, recall, and F-value measures as for the relation tasks before.

3.2. *Accuracy and Error*

The measures *accuracy* and *error* report the percentage of correct and wrong results in a system's output. Among others, they are used in the evaluation of document classification or (part-of-speech) tagging tasks. By themselves, they are not very meaningful for system evaluation due to their insensitiveness to non-selected but relevant information (see Section 8.1 in Manning & Schütze[30] for a thorough discussion on this topic).

However, in combination with precision and recall they can provide additional insight for the normalization (N) and grounding (GR) tasks: Here, they can be applied to show the percentage of *correctly normalized entities* over all *correctly detected entities* (e.g., the ratio of all correctly normalized mutations over all correctly detected mutations). The reasoning behind this is that normalization for a wrongly detected entity is not meaningful, even if it has been computed correctly. Thus, the accuracy as defined above directly shows the percentage of correctly normalized entities. The same idea applies to the evaluation of grounding tasks.

3.3. *Metrics for Impact Analysis*

Impact detection and analysis poses further challenges to the evaluation. For their structured form (IA-2, SYS-3), we apply the standard precision/recall measures, provided both gold

Table 1. Overview: evaluation tasks and corresponding metrics

Metric Task	ER-1,2,3	N-1,2,3	REL-1,2	GR-1,2,3	IA-1	IA-2	SYS-1	SYS-2	SYS-3	SYS-4
Precision	✓	(✓)	✓	(✓)	✓	✓	✓	(✓)	✓	✓
Recall	✓	(✓)	✓	(✓)	✓	✓	✓	(✓)	✓	✓
F-Measure	✓	(✓)	✓	(✓)	✓	✓	✓	(✓)	✓	✓
Accuracy	(✓)	✓	(✓)	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
Error	(✓)	✓	(✓)	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
ROUGE					✓			✓		
BE					✓			✓		
Pyramids					✓			✓		

standard and system output follow a prescribed data format (e.g., filled-in templates or populated ontology[r] instances).

For systems creating free-form natural language summaries (IA-1, SYS-2), these measures can only be applied if both gold standard and system output work by sentence extraction. Then, relevant sentences can be marked and compared with the sentence set returned by a system, applying the precision&recall measures as before. However, more sophisticated systems might want to improve on these summaries, e.g., by rewriting sentences in order to remove duplicate information or dangling pronominal references or even abstracting from individual results. In that case, it becomes necessary to apply the evaluation metrics developed for automatic summarization, like ROUGE *(Recall-Oriented Understudy for Gisting Evaluation)*,[31] BE *(Basic Elements)*,[32] or Pyramids.[33] These metrics have been in use for a number of years in the NIST-sponsored DUC *(Document Understanding Conference)* competition in automatic summarization.[s] ROUGE, BE, and Pyramids all allow to compute precision/recall values for automatically generated summaries based on a set of manually developed example summaries, motivated by the fact that natural language allows a wide variety of ways to express equivalent knowledge in textual form.

3.4. *Summary and Outlook*

A summary of the metric/task relationships is contained in Table 1. Here, a "✓" shows that a metric is used to evaluate this task, while a "(✓)" indicates that the metric is applicable in general, but does not provide further insights into a system's performance and is hence not recommended for evaluating the task.

Some of the additional tasks outlined in Section 2.7 require further metrics for evaluation. For Information Retrieval (IR), the classical precision & recall measures obviously still apply, together with additional IR-specific measures such as *Precision@n*. Coreference resolution

[r] Here, we assume a pre-existing ontology structure, i.e., an *ontology population* task rather than an *ontology learning* task, as the latter would require more involved metrics from the areas of ontology matching & alignment, taking the created class hierarchy into account.

[s] Document Understanding Conference (DUC), `http://duc.nist.gov`

also employs precision and recall measures. However, their definition differs greatly from the ones presented above, instead relying on graph-based or cluster-based algorithms. See, e.g, the MUC[34] and CEAF[35] measures. The evaluation of Question-Answering (QA) again requires different considerations: for open-ended questions, such as those addressed in the DUC competitions 2005-2007, the aforementioned metrics ROUGE, Basic Elements, and Pyramids can be used. The evaluation of factual questions, on the other hand, will need to employ strategies as used in the TREC competitions—see, e.g., the 2007 genomics track.

4. Conclusions

The maturing of a research or technology domain is typically recognized by the emergence of standards, for example standards for data representation, exchange formats, processes or standards for evaluations, and in some cases legal standards also. In this paper, we mark the maturing of the mutation extraction domain by laying out the fundamental tasks, their context and evaluation metrics in a review format. In so doing we have identified that the evaluation of mutation extraction, a seemingly simple task, is surprisingly complex.

We have sought to provide a framework for a possible future community-wide evaluation effort of mutation analysis systems, and we also anticipate that our discussion will allow for a more precise positioning of the contributions by future publications. The current state, where "mutation detection" is used for completely different tasks, from simple entity detection to sequence verification, is not adequate for acceptance and advancement of this research field. Indeed there will be further opportunities to apply such metrics when automated mutation extraction is used for 'extrinsic' tasks, for example to rebuild existing mutation databases from legacy literature sources or assessment of the quality of existing manually deposited mutation database entries. Some valuable studies have already embarked on these tasks. These will eventually require accepted standards of assessment and possibly new evaluation metrics.

In presenting this overview our goal is to invigorate the debate in this valuable research domain so that renewed focus and the recently developed approaches and technologies will contribute to resolving the complexities of mutation extraction systems. The larger objective is to motivate for uninhibited access to mutation information and to the hard won observations describing their consequences, such that they are readily available for downstream experimental design and hypothesis generation by biomedical scientists.

Acknowledgments

We would like to thank Thomas Kappler and Vladislav Ryzhikov for their contributions within the evaluation of the Mutation Miner system.

References

1. Kawabata T., Ota M., and Nishikawa K. The protein mutant database. *Nucleic Acids Research*, vol. 27(1), 1999.
2. Ananiadou S. and McNaught J., editors. *Text Mining for Biology and Biomedicine*. Artech House, 2006.

3. Cohen A.M. and Hersh W.R. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, vol. 6:57–71, 2005.
4. Rebholz-Schuhmann D., Kirsch H., and Couto F. Facts from Text—Is Text Mining Ready to Deliver? *PLoS Biology*, vol. 3:188–191, 2005.
5. Spasic I., Ananiadou S., McNaught J., and Kumar A. Text mining and ontologies in biomedicine: making sense of raw text. *Briefings in Bioinformatics*, vol. 6, 2005.
6. Hirschman L. and Blaschke C. Evaluation of Text Mining in Biology. In Ananiadou and McNaught,[2] chapter 9.
7. Hirschman L., Yeh A., Blaschke C., and Valencia A. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, vol. 6(Suppl 1), 2005.
8. Chang J. and Schütze H. Abbreviations in Biomedical Text. In Ananiadou and McNaught,[2] chapter 5.
9. Rebholz-Schuhmann D., Marcel S., Albert S., Tolle R., Casari G., and Kirsch H. Automatic extraction of mutations from Medline and cross-validation with OMIM. *Nucleic Acids Research*, vol. 32(1):135–142, 2004.
10. Caporaso J.G., Jr. W.A.B., Randolph D.A., Cohen K.B., and Hunter L. MutationFinder: A high-performance system for extracting point mutation mentions from text. *Bioinformatics*, vol. 23(14):1862–1865, 2007.
11. Bairoch A., Apweiler R., Wu C.H., Barker W.C., Boeckmann B., Ferro S., Gasteiger E., Huang H., Lopez R., Magrane M., Martin M.J., Natale D.A., O'Donovan C., Redaschi N., and Yeh L.S.L. The Universal Protein Resource (UniProt). *Nucleic Acids Research*, 2005.
12. Federhen S. The Taxonomy Project. In J. McEntyre and J. Ostell, editors, *The NCBI Handbook*, chapter 4. National Library of Medicine (US), National Center for Biotechnology Information, 2003.
13. Hahn U. and Wermter J. Levels of Natural Language Processing for Text Mining. In Ananiadou and McNaught,[2] chapter 2, pages 13–41.
14. Leroy G., Chen H., and Martinez J.D. A shallow parser based on closed-class words to capture relations in biomedical text. *J. of Biomedical Informatics*, vol. 36:145–158, 2003.
15. Wattarujeekrit T., Shah P.K., and Collier N. PASBio: predicate-argument structures for event extraction in molecular biology. *BioMed Central Bioinformatics*, vol. 5(155), 2004.
16. Leroy G. and Chen H. Genescene: An Ontology-enhanced Integration of Linguistic and Co-occurrence based Relations in Biomedical Texts. *Journal of the American Society for Information Systems and Technology (JASIST)*, vol. 56(5):457–468, March 2005.
17. Witte R., Kappler T., and Baker C.J.O. Ontology Design for Biomedical Text Mining. In Baker and Cheung,[36] chapter 13, pages 281–313.
18. Kim J.J. and Park J.C. BioAR: Anaphora Resolution for Relating Protein Names to Proteome Database Entries. In S. Harabagiu and D. Farwell, editors, *ACL 2004: Workshop on Reference Resolution and its Applications*, pages 79–86. Association for Computational Linguistics, Barcelona, Spain, 2004.
19. Witte R., Kappler T., and Baker C.J.O. Enhanced semantic access to the protein engineering literature using ontologies populated by text mining. *Int. J. Bioinformatics Research and Applications (IJBRA)*, vol. 3(3):389–413, 2007.
20. Horn F., Lau A.L., and Cohen F.E. Automated extraction of mutation data from the literature: application of MuteXt to G protein-coupled receptors and nuclear hormone receptors. *Bioinformatics*, vol. 20(4):557–568, 2004.
21. Witte R. and Baker C.J.O. Combining Biological Databases and Text Mining to support New Bioinformatics Applications. In *10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, vol. 3513 of *LNCS*, pages 310–321. Springer, June 15–17 2005.
22. Schomburg I., Chang A., Ebeling C., Gremse M., Heldt C., Huhn G., and Schomburg D. BRENDA,

the enzyme database: updates and major new developments. *Nucleic Acids Research*, vol. 32, 2004.

23. Wittig U., Golebiewski M., Kania R., Krebs O., Mir S., Weidemann A., Anstein S., Saric J., and Rojas I. SABIO-RK: Integration and Curation of Reaction Kinetics Data. In *Proceedings of the 3rd International workshop on Data Integration in the Life Sciences 2006 (DILS'06)*, number 4075 in Lecture Notes in Bioinformatics, pages 94–103. Springer, Hinxton, UK, 2006.
24. Goble C., Wolstencroft K., Goderis A., Hull D., Zhao J., Alper P., Lord P., Wroe C., Belhajjame K., Turi D., Stevens R., Oinn T., and Roure D.D. Knowledge Discovery for Biology with Taverna. In Baker and Cheung,[36] chapter 16, pages 355–395.
25. Castaño J., Zhang J., and Pustejovsky J. Anaphora Resolution in Biomedical Literature. In *International Symposium on Reference Resolution*. Alicante, Spain, 2002.
26. Gasperin C. Semi-supervised anaphora resolution in biomedical texts. In *Proceedings of the HLT-NAACL Workshop on Linking Natural Language Processing and Biology (BioNLP)*. New York City, NY, USA, 2006.
27. Vlachos A., Gasperin C., Lewin I., and Briscoe T. Bootstrapping the Recognition and Anaphoric Linking of Named Entities in Drosophila Articles. In *Pacific Symposium on Biocomputing*, pages 100–111. 2006.
28. Couto F.M., Silva M.J., and Coutinho P. ProFAL: PROtein Functional Annotation through Literature. In *VII Conference on Software Engineering and Databases (JISBD)*, pages 747–756. 2003.
29. Baeza-Yates R. and Ribeiro-Neto B. *Modern Information Retrieval*. Addison-Wesley Longman Limited, 1999.
30. Manning C.D. and Schütze H. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.
31. Lin C.Y. ROUGE: a Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*. Barcelona, Spain, July 25–26 2004. `http://www.isi.edu/~cyl/ROUGE/`.
32. Hovy E., Lin C., and Zhou L. Evaluating DUC 2005 using Basic Elements. In NIST, editor, *Proceedings of the HLT/EMNLP Workshop on Text Summarization DUC 2005*. Vancouver, BC, Canada, October 9–10 2005.
33. Nenkova A. and Passonneau R.J. Evaluating Content Selection in Summarization: The Pyramid Method. In *Proc. HLT/NAACL*, pages 145–152. 2004.
34. Vilain M., Burger J., Aberdeen J., Connolly D., and Hirschman L. A model-theoretic coreference scoring scheme. In *MUC6 '95: Proc. of the 6th conf. on Message understanding*, pages 45–52. ACL, 1995. ISBN 1-55860-402-2.
35. Luo X. On Coreference Resolution Performance Metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natur al Language Processing*, pages 25–32. ACL, Vancouver, British Columbia, Canada, October 2005.
36. Baker C.J.O. and Cheung K.H., editors. *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*. Springer Science+Business Media, New York, NY, USA, 2007.

René Witte is currently a research associate at the Institute for Program Structures and Data Organization (IPD), University of Karlsruhe, Germany, where he heads the Text Mining group. His work encompasses foundations in natural language processing, technical aspects of semantic systems including ontologies, semantic desktops, and the Semantic Web, as well as the application of text mining in diverse areas such as news analysis, building architecture, biomedical research and discovery, and software engineering. Previously, he worked as a post-doctoral fellow and research associate within the computational linguistics and software engineering groups, both at Concordia University in Montréal, Canada. He also has several years of industry experience as an information systems consultant. Dr. Witte holds a Doctorate of Engineering (Dr.-Ing.) and a Diploma in computer science from the Faculty of Informatics, University of Karlsruhe, Germany.

Christopher J. O. Baker is a Principal Investigator at the Institute for Infocomm Research, A∗STAR, in Singapore. His current focus is on the application of semantic web and text mining technologies in knowledge-based life science information systems. Prior to his current appointment he held the following scientific leadership roles; Bioinformatics Project Manager of the Génome Québec funded project, 'Ontologies, the semantic web and intelligent systems for genomics' and Group Leader in silico Discovery at Ecopia BioSciences Inc.

Dr Baker received postdoctoral training at Iogen Corporation and the University of Toronto after completing his Ph.D. studies in Environmental Microbiology and Enzymology at the University of Wales, Cardiff, UK. He is co-editor of the first book to illustrate deployment of the semantic web technologies in the life sciences.