What's in a gene name?
Automated refinement of gene name dictionaries

Jörg Hakenberg
Bioinformatics Group, Biotechnological Centre
Technische Universität Dresden, 01307 Dresden, Germany
hakenberg@informatik.hu-berlin.de

Abstract

Many approaches for named entity recognition rely on dictionaries gathered from curated databases (such as Entrez Gene for gene names.) Strategies for matching entries in a dictionary against arbitrary text use either inexact string matching that allows for known deviations, dictionaries enriched according to some observed rules, or a combination of both. Such refined dictionaries cover potential structural, lexical, orthographical, or morphological variations. In this paper, we present an approach to automatically analyze dictionaries to discover how names are composed and which variations typically occur. This knowledge can be constructed by looking at single entries (names and synonyms for one gene), and then be transferred to entries that show similar patterns in one or more synonyms. For instance, knowledge about words that are frequently missing in (or added to) a name ("antigen", "protein", "human") could automatically be extracted from dictionaries. This paper should be seen as a vision paper, though we implemented most of the ideas presented and show results for the task of gene name recognition. The automatically extracted name composition rules can easily be included in existing approaches, and provide valuable insights into the biomedical sub-language.

1 Introduction

Recognition of named entities (NER), such as names referring to genes and proteins, forms a major building block for text mining systems. Especially in the life sciences, a large amount of different entity types and their instances exist. Two basic strategies for NER are classification- and dictionary-based approaches. Classifiers learn (or are given) models to decide whether a sequence of tokens refers to an entity or not. Such decisions are based on various forms of input, for instance, tokens and their sequence in a sentence, part-of-speech tags, characteristic suffixes, and trigger keywords[1] (Hakenberg et al., 2005). Models can be learned from a given training sample. Dictionary-based approaches rely on curated word lists containing (all known) representatives of an entity type. Manual or automated refinement of the dictionary and inexact matching strategies allow to cover a broad spectrum of name variations (Hanisch et al., 2005). Classification-based approaches have proven to be very robust towards unseen tokens and names, because they also incorporate knowledge on names of the given class in general[1] (Crim et al., 2005). Dictionaries, on the other hand, reflect the knowledge about an entity class at a given time, and such approaches cannot find instances unknown to them. However, the main advantage of dictionary-based NER is that they bring the explicit possibility to map recognized entities to the source of the entries (most times, a database.) This alleviates the task of named entity

[1] For example, a protein name often is/has a proper noun; many enzymes end with '–ase'; 'domain of' is often followed by a protein name.

BioNLP 2007: Biological, translational, and clinical language processing, pages 153–160,
Prague, June 2007.

identification (NEI) that is needed to annotate texts properly or link text-mined facts to database entries.

In this paper, we want to concentrate on dictionary-based approaches and present ideas of how these could be automatically refined and enriched. In such a setting, named entity recognition functions as a method of 'spotting' entities in a text, after which further identification (disambiguation) is needed. NER components thus should guarantee very high recall rates with a reasonable precision. NEI then refines the predictions of NER, eliminating false positive annotations and identifying names. That such a setup would perform quite well is reflected, for example, in a study presented by Xu et al. (2007). They showed that sophisticated disambiguation strategies currently yield up to 93.9% precision (for mouse genes; yeast: 89.5%, fly: 77.8%.) Participants in the BioCreAtIvE 2 challenge showed similar values for human genes (up to 84.1% precision, 87.5% recall, or 81.1% F1), see Morgan and Hirschman (2007) for a summary.

Hand-coded rules for creating spelling variations have been proposed before, see section on Related Work. Such rules are applied to synonyms to generate morphological and orthographical variations ("Fas ligand" → "Fas ligands" and "Ifn gamma" → "Ifn-γ", respectively). In the same manner, systems use known patterns for structural changes of names and mappings for lexical variations to enrich existing dictionaries ("CD95R" → "receptor of CD95" and "gastric alcohol dehydrogenase" → "stomach alcohol dehydrogenase"). Our research question in this paper is, how such rules can be learned automatically from dictionaries that contain entries of the same entity class with multiple, typical synonyms each. Learning about the composition of names comes down to an analysis of known names. A human, given the same task, would look through a lot of examples to derive term formation patterns. Questions to ask are:

- What are frequent orthographical and morphological variations?
- Which parts of a name get abbreviated?
- How are abbreviations formed?
- Which identical abbreviations can be observed in multiple names?
- In which way can a name structurally and lexically change?
- Which are the parts of a name that can be exchanged with other terms or skipped entirely?
- Which are the important parts of a name, which are additional descriptive elements?

In this paper, we demonstrate methods to analyze names in order to find the semantically important parts. We map these parts to potential syntactic variations thereof observed within a name and its synonyms. We assess the frequency of such mappings (exchange of tokens, different ordering of tokens, etc.) and transfer this knowledge to all other names in the same dictionary. In this setup, understanding a name results in a structured decomposition of the name. Such decompositions provide knowledge on how to find (and identify) the name in arbitrary text, as they give insights into its mandatory, unique, and ambiguous[2] parts.

This paper should be seen as a vision paper, though we implemented most of the ideas presented herein and show first results. We first explain the idea behind learning name composition rules, motivated by manual curation as described in Related Work. We then explain the basic techniques needed for our analysis. We show how single entries (a name and all its synonyms) can be analyzed to find composition rules, and how these can be transferred to other entries. Preliminary results using some of the ideas presented here are also given. We conclude this paper with a discussion of the experimental methodology and an outlook.

1.1 Related Work

Current survey articles cover the spectrum of recent methods and results for biomedical named entity recognition and identification (Cohen and Hersh, 2005; Leser and Hakenberg, 2005). A recent assessment of named entity recognition and identification was done during the BioCreAtIvE 2 evaluation[3]. Official results will be available in April 2007. Naturally, a number of systems proposed before are highly related to the method presented in this paper. Hanisch et al. (2005) proposed the ProMiner system to recognize and identify protein names in text. They observed that the ordering of tokens in a name occur quite frequently, but do not change the seman-

[2] The latter two as compared to the whole dictionary.

[3] See http://biocreative.sourceforge.net .

tics of the overall name. They presented a model for protein names, partitioning tokens into token classes according to their semantic significance: modifiers (“receptor”), specifiers (“alpha”), non-descriptive tokens (“fragment”), standard tokens (“TNF”), plus common English words and interpunctuation. To evaluate the significance of tokens, they count their respective frequencies in a dictionary. Hanisch et al. extract a dictionary using various knowledge source (HGNC etc.) and expand and prune it afterwards. Expansion and pruning are based on manually defined rules (separating numbers and words, expanding known unambiguous synonyms with known synonyms, applying curation lists maintained by biological experts, predefined regular expressions). The final matching procedure found names by comparing (expanded) tokens and their classes to arbitrary text, where some token classes were mandatory for the identification and others could be missing. ProMiner yielded results between 78 and 90% F1-measure on the BioCreAtIvE 1 (Task 1B), depending on the organism-specific sub-task. The highest recall was found to be 84.1% for fly, 81.4% for mouse, and 84.8% for yeast genes.

We used a similar method, relying entirely on manually defined rules for name variations, for the BioCreAtIvE 2 GN task (Hakenberg et al., 2007). We expanded the dictionary applying these rules to every synonym (treating abbreviations and spelled-out names slightly different). This yielded a recall of 92.7 and 87.5% on the training and test sets, respectively (F1: 81.1%). In the aftermath of BioCreAtIvE 2, we now try to improve this high recall values further, by automatically analyzing the whole dictionary of gene names instead of manually composing useful rules in a trial–and–error approach.

2 Methods

We first want to present the overall idea of learning name composition rules, guided by specific examples. We first show how comparison of synonyms known for one gene name yields insights into the ‘meaning’ of the gene, and produces rules for structural and lexical variations of its name(s). Afterwards, we explain how such rules can be exchanged between different genes and add to the understanding of each genes ‘meaning.’

2.1 Techniques

We apply several techniques to the analysis of names. To detect abbreviations by pairwise comparison of synonyms, we use the algorithm proposed by Schwartz and Hearst (2003) as the core component[4]. We changed some of the details so that, for instance, the first letter of the potential abbreviation has to match the first letter of the proposed long form. We perform the detection of abbreviations not only on whole synonyms, but also on parts of each name (like for “TNF-alpha stimulated ABC protein”), so that this property of Schwartz and Hearst’s algorithm (S&H) is recovered. A trivial adaptation also reveals which parts of an abbreviation (one or more characters) map to which parts of the long form (one token, one partial token.) As S&H allows for missing tokens in the long form, we can also add the possibility for (few) characters in the abbreviation not being reflected in the long form.

To detect inexact matches (that is, slight variations in morphology or orthography), we use an adaptation of the biological sequence alignment algorithm (Needleman and Wunsch, 1970). Using the computed alignment score, this yields an immediate quantification of the similarity of two terms.

We compare the sequence of identified name parts (parts of a name where a mapping from this part to a part of the other synonym exists) in order to find parts that can be skipped or exchanged with each other. In addition, this yields insights into potential permuations of all parts of a name, and shows where certain parts typically do or do not occur.

2.2 Representation

Representation of information extracted by parsing *i)* a synonym or *ii)* all synonyms of a gene becomes a crucial basic part of our approach. *Concepts* have to be found in a name, for instance,

- *substance*: “serotonin”,
- *type*: “receptor”,
- *function*: “transcription factor”, or
- *family-member*: “family-member number 6”.

Also, for these concepts, rules have to be learned that match them against text (or vice versa): an ‘R’ hints on a receptor, a ’6’ at the end of a name (for instance, a noun phrase) hints on a family-member or

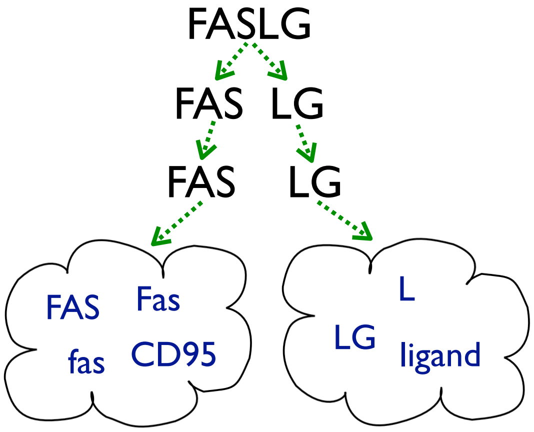
[4] The original algorithm decides whether a given short form can be explained by a given long form.

Type	Example token	Example name
Descriptor	antigen, ligand, inhibitor	P-30 antigen
Modifier	factor, family member, type	BRG1-associated factor
Specifier	alpha, IX, A	TNF alpha
Source	d, _HUMAN, p	dHNF-4

Table 1: Types of tokens that frequently occur in gene names. Also see Hanisch et al. (2005), though they introduce different conventions.

type. We rely on semantic types, which are defined using descriptions automatically identified from the syntax (lists of variations), rather than pure syntactical ones. This helps during classification of identified concepts: a syntactical concept would map "s" to "serotonin"; but additionally, we need to express that the given gene demands any arbitrary form of a reference to a substance, which is serotonin, in its name. Whether this occurs as the substance's name itself, an abbreviation, or synonym of the substance, and at which position in a text[5], then becomes less important concerning the matching strategy. Table 1 sums up some of the known types of tokens and examples we want to distinguish. Note that the proper type definition cannot automatically be assigned to a concept. Concepts can be identified as belonging to the same type only because they share certain properties (can be skipped, is a numerical entity, is a mandatory tokens that occurs at the end of a name.) In Table 1, the descriptors "antigen" and "ligand", for instance, appear to be of the same type, but analysis will reveal that while the mention of "antigen" in a name is skipped frequently, "ligand" represents a mandatory concept in many synonyms.

For the remainder of this paper, we subsequently break down a gene into the basic concepts described in one or more of its name. First, a gene is identified by a set of *names* (synonyms). Second, each name consists of multiple *parts*; proper separation and identification is a crucial step. Third, each part of a name then represents a certain *concept* that is typical for the gene. A gene is defined by all identified concepts. While a gene name part stores the information on where and if it occurs in the sequence of parts that ultimately form the (or rather a) name of the gene, concepts store information about variations. Knowledge about name parts and concepts is then transferred within each respective level only. Each such potential transfer we call a *composition rule*. An example, which we will also discuss in the next section, is the gene FASLG. Is has multiple synonyms, "FASLG" being one of those. This name can be separated into the parts "FAS" and "LG". The first part has the concept "FAS", which can appear in the variations "Fas", "fas", or "CD95", as we will see later; the second part has the concept "LG", a possible variation is "ligand":



(from top to bottom, levels depict the name, parts, concepts, and variations of each concept.)

2.3 Analysis of intra-gene variations

In this section we explain how we discover concepts and their appearances (exact tokens) within a set of synonyms under the assumption that they all belong to the same gene. Basically, this means that we can allow for more mismatches, lacking parts, or the like, as for comparing names of different genes.

Reconsider the example of the aforementioned FASLG gene (356)[6]. We show the synonyms known according to Entrez Gene in Table 2. Pairwise analysis of the synonyms provides insights as shown in Table 3.

Recombining the extracted concepts and using different variations for either, we can achieve some new potential names, for instance, FasL (capitalization) and CD95 ligand (replaced 'L' with identified

[5] Maybe within a somewhat confined neighborhood, for instance, in the current paragraph or in the abstract of the text.

[6] In the following, we will always show each gene's official symbol first and then known synonyms. Numbers in brackets refer to Entrez Gene IDs.

Apoptosis antigen ligand	APTL	apoptosis (APO-1) antigen ligand 1
Apoptosis (APO-1) ligand 1	APT1LG1	FAS antigen ligand
Apoptosis ligand	CD178	Fas ligand (TNF superfamily, member 6)
CD95L	FASL	TNFL6_HUMAN
fas ligand	FASLG	TNFSF6
FAS ligand	TNFL6	Tumor necrosis factor ligand superfamily member 6

Table 2: Synonyms of the FASLG gene that we use in our examples.

Synonyms	Composition rule learned	No.
FASL + FAS ligand	L ≡ ligand	1
FASLG + FAS ligand	LG ≡ ligand	2
FAS ligand + fas ligand	FAS ≡ fas	3
FASL + CD95L	FAS ≡ CD95	4
Tumor necrosis factor ligand superfamily member 6 + TNFSF6	T ≡ Tumor, N ≡ necrosis F ≡ factor, SF ≡ superfamily “member” before a number can be left out	5a,b 5c,d 5e
Apoptosis antigen ligand + Apoptosis ligand	“antigen” can be left out	6
FAS antigen ligand + FAS ligand	“antigen” can be left out	7
Apoptosis (APO-1) ligand 1 + Apoptosis ligand	“1” at end can be left out	8
TNFL6 + TNFL6_HUMAN	“_HUMAN” can be added to a name	9
Fas ligand (TNF superfamily, member 6) + FAS ligand	Fas ≡ FAS	10
Apoptosis ligand + APTL	Apoptosis ≡ APT	11
Apoptosis (APO-1) ligand 1 + APT1LG1	ligand 1 ≡ LG1	12

Table 3: Pairwise analysis of some synonyms for FASLG and some insights gained. Conclusions shown in the bottom part can be drawn using insights from the first part only. Rules like “X can be left out” imply that the opposite can also happen, “X can be added”, and vice versa. Multiple detections of the same rule (no. 6 & 7) increase its support, so the application of rules could be weighted accordingly.

long form) for the FASLG gene. In cases where neither part of a name can be mapped onto parts of another name, then no rule should be generated: comparing “CD178 antigen” to “CD95 ligand” should not result in the variation “CD178 ligand”. On the other hand, after removal of “antigen” (rules no. 6 & 7 in Table 3), “CD178” represents a variation of “CD95 ligand” (which in this case was already known from Entrez Gene.) In the following sections, we explain the detection of different kinds of variations in more detail and show examples.

Abbreviations

Detecting abbreviations is a crucial initial step in our analyses. Many variations are explained only across abbreviations and their long forms. More important, comparing abbreviations and long forms identifies the parts of either name, which can then be compared to parts of other names. Taking HIF1A (3091) as an example, we find the synonyms “HIF1 alpha”, “HIF-1 alpha”, “HIF-1alpha”, and “Hypoxia-inducible factor 1 alpha”. Schwartz and Hearst’s algorithm easily reveals that “1 alpha”, “1alpha”, and “1A” all map to each other; “H” can be mapped to “Hypoxia”, and so on. All in all, we learned that “Hypoxia-inducible factor 1A” could be a potential synonym for HIF1A.

We now look at the OR1G1 gene (8390). Consider two of its synonyms, “Olfactory receptor 1G1”, and “olfactory receptor, family 1, subfamily G, member 1”. Comparing the official symbol with the first synonym, it becomes clear that “OR” abbreviates “Olfactory receptor” using S&H. Comparing the synonyms, we find direct correspondences between both “1”s and “G”. AS we are still within one gene, is is safe to assume that all in all, “1G1” abbreviates “family 1, subfamily G, member 1”. This implies that concepts stating that we are within a gene family (subfamily, members) can be missing – whereas the respective values (“1”, “G”, “1”) are mandatory.

Another abbreviation that commonly occurs in gene names is the (abbreviated) mention of the organism (on the species level). For example, the gene GIMAP4 (55303) has “HIMAP4”, “IMAP4”, “IAN1”, “hIAN1”, and “human immune associated nucleotide 1” as known synonyms. From synonyms 1 and 2 we can infer that an “H” can be added to a name (just like “_HUMAN”, see Table 3.) The same is true for “h” (synonyms 3 and 4.) Comparing synonyms 1 or 4 to 5 leads to the conclusion that “H” and “h” both abbreviate “human.”

Lexical variations

In the set of synonyms for ADHFE1 (137872), we find “Fe-containing alcohol dehydrogenase 1” and “alcohol dehydrogenase, iron containing, 1”. Splitting these synonyms into their respective parts and then comparing both sets reveals that all but one part each can be exactly mapped to a corresponding part in the other synonym. From this almost exact match, we can conclude that the parts “Fe” and “iron” are synonyms of each other, potentially representing the same concept, and easy to confirm for a human.

In the same manner, we will find that “1B” can be sometimes expressed as “2”, and that “adaptor” and “Adapter” are orthographic variations of each other, by looking at some synonyms for AP1S2 (8905):

- Adapter-related protein complex 1 sigma 1B subunit
- adaptor-related protein complex 1 sigma 2 subunit
- adaptor-related protein complex 1, sigma 1B subunit

To detect these two changes, we first need to map parts to each other and then compare the names based on the sequence of the parts.

Structural variations

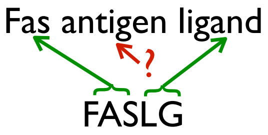
Changes in the structure of a name can be deduced when a safe mapping between most parts of a name exist. For the HMMR gene (3161), we find two evidences for such a variation, which also lead to the conclusion that “for” is an optional part. However, in our system, we would retain information concerning the positioning of “for” (at least, tendencies like “not the first” and “not the last” part.)

- Receptor for hyaluronan-mediated motility
- hyaluronan-mediated motility receptor
- Hyaluronan mediated motility receptor
- intracellular hyaluronic acid binding protein
- hyaluronan-mediated motility receptor (RHAMM)

Analysis of this example also finds that “hyaluronan” can start with an upper case letter (and that this occurs only when it is the first part of the name. “RHAMM” is the abbreviation for “Receptor for hyaluronan-mediated motility”, as revealed by S&H. This leads to the next conclusion, that abbreviations can immediately follow a gene name.

Descriptive elements

Comparing the sequence of identified name parts (parts of a name where a mapping from this part to a part of the other synonym exists) yields dissimilarities that result either from a dropped/added name part, or from a lexical variation. Consider the following example:



Inexact matching immediately identifies the mapping from “Fas” to “FAS”; abbreviation detection and/or alignment yields “ligand” as a long form/variation of “LG.” The sequence of name parts if the same in both synonyms, with an added “antigen” in the first synonym. An extracted composition rule could thus be that “antigen” is of additional, descriptive value only, and can be skipped. Knowing this, the first synonym should also match the strings “Fas ligand” and “FAS ligand” (in fact, both should.)

Another example is ZG24P (259291) with its synonym “uncharacterized gastric protein ZG24P”. As the official symbol clearly is an abbreviation (single word, upper case letters, numbers) and matches the last part of the synonym, we can assume that the first part is either another synonym or a mere descriptive element that explains the real gene name. Indeed, patterns like “uncharacterized ... protein” or “hypothetical protein” appear frequently as first parts of gene names.

2.4 Analysis of inter-gene variations

As we have so far analyzed synonyms of one and the same gene to extract knowledge on name composition, we can now apply this knowledge to the whole set of gene names. This means, that we add knowledge gained by analyzing one gene to other genes, wherever applicable. Essentially, this comes down to finding corresponding concepts in two or more genes’ names, and joining the information contained in each concept. If within one gene name it became clear that “L” and “ligand” represent the same concept, and for another gene “L” and “LG” are variations of the same concept, then a combined concept would have all three variations. The combined concept then replaces the old concepts. We apply the same idea to name parts, for which information about their ordering etc. was extracted.

Inter-gene analysis also reveals the main distinctive features of single gene names or groups of names (for instance, families.) Some names differ only in Arabic/Roman numbers or in Greek letters. Potentially they belong to the same group, as different members or subtypes. Knowing how to find one family member implicitly means knowing how to find the others. Thus, it helps identify crucial parts (for the family name) and distinctive parts (for the exact member.) A matching strategy could thus try to find the family name and then look for any reference to a number. Knowledge about this kind of relationships has to be encoded in the dictionary, however. Spotting a gene family's name without any specific number could lead to the assignment of the first member to the match, see Table 3, rule no. 8 (or dismissing the name, depending on user-specific demands). Such information can also be used for disambiguating names. Analyzing the names "CD95 ligand" and "CD95 receptor" of two different genes, it can be concluded that "CD95" by itself contains not enough information to justify the identification of either gene directly. Finding other "receptor"s in the dictionary will also mark "receptor" as a concept crucial, but not sufficient, for identifying a gene's name in text. For "CD95", on the other hand, we have shown before that this token might be exchanged with others.

Knowledge about (partial) abbreviations, like in aforementioned "HIF" = "Hypoxia-inducible factor" and "OR" = "olfactory receptor", can be transferred to all synonyms from other entries in the dictionary that have the same long or short forms (but possibly do not mention the respective other in any synonym.) Similarly, presumed lexical variations ("gastric" versus "stomach") that have been found for one gene name (one concept) can be included in all corresponding concepts to spread the information that "gastric" can appear as "stomach" in text. This is necessary to detect the name "stomach alcohol dehydrogenase", where the corresponding Entrez Gene entry (ADH7, 131) does have the token "stomach" in any of its synonyms.

Also, synonyms mentioning the species (like "hIAN1" to depict human) are not contained for every entry. Learning that "h" can be added to a gene name helps recognizing such a variation in text for other names (the dictionary lacks the variation "hFasL" of FASLG, which is sometimes used.)

3 Evaluation and Conclusions

We evaluated some ideas presented in this paper on the BioCreAtIvE 2 (BC2) dataset for the gene normalization task. For the purpose of this study, we were interested in how our method would perform concerning the recall, as compared to methods based on hand-curated dictionary refinement. We conducted the following experiment: the BC2 GN gold standard consists of references to abstracts (PubMed IDs), genes identified in each abstract (Entrez Gene IDs) and text snippets that comprise each gene's name. For one abstract, there could be multiple, different snippets representing the same gene, ADH7 (131): "stomach alcohol dehydrogenase", "class IV alcohol dehydrogenase", or "sigma-ADH", all in the same abstract. For identification, it was sufficient in BC2 to report the ID, regardless of number of occurrences or name variations.

As the method presented in this paper lacks a matching strategy for spotting of names, we performed our initial evaluation on the text snippets only. Finding the right ID for *each* snippet thus ultimately yielded the recall performance. In the above example, we would try to identify ID 131 three times, counting every miss as a false negative. The methods presented above were able to yield a recall of 73.1%. With the original BC2 evaluation scheme, we achieve a recall of 84.2%. Compared to the highest result for our system with a manually refined dictionary, this figure is more than 8% lower. This shows that still, many name variations are not recognized. Some errors could be accounted to ranges or enumerations of gene names ("SMADs 1, 5 and 8"), others to not far enough reaching analyses: for detecting "SMAD8", we only had the synonyms "SMAD8A", "SMAD8B", and "SMAD9" for the correct gene in the dictionary (all are synonyms for the same gene, according to Entrez Gene). It should thus have been learned that the letter "A" can be left out (similar to "1", see rule no. 8 in Table 3.) Another undetected example is "G(olf) alpha" (GNAL, 2774). Rules to restrict either of the synonyms

- Guanine nucleotide-binding protein G(olf), alpha subunit
- guanine nucleotide binding protein (G protein), alpha stimulating activity polypeptide, olfactory type
- Adenylate cyclase-stimulating G alpha protein, olfactory type
- Guanine nucleotide-binding protein, alpha-subunit, olfactory type

to this mentioning in text could have been deduced as follows:

(1) Learn in another gene: description before "protein" can be left out ⇒ "G(olf), alpha subunit" could be a name of its own.

(2) Learn in this or another gene: "alpha subunit" can be expressed as "alpha" (or "subunit" skipped) ⇒ "G(olf) alpha" could be a name.

We see that most orthographical and morphological variations (Greek symbols/English words, singular/plural forms, capitalization) can be integrated quite easily in matching techniques. The general knowledge about such variations is far-reaching and can be applied to most domains. In contrast, structural and lexical variations are much harder to pinpoint and express in general ways; mostly, such possible variations are specific to a sub-domain and thus present the main challenge for our method.

The ideas discussed in this paper originated from work on the aforementioned BioCreAtIvE 2 task. In that work, we used manually designed rules to generate variations of gene names. Hanisch et al. (Hanisch et al., 2005) and other groups propose similar methods all based on human observation and experience leading to refined dictionaries. As many causes for name variations are easy to spot and express, we concluded it was entirely possible to gain such insights in an automated manner. Left undetermined is the potential impact of composition rules on machine-learning techniques that use dictionaries as input for features.

However, the methodology should work for other task using the same or similar initial observations (This remains to be proven.) We are currently applying the method to the analysis of Gene Ontology terms (Ashburner et al., 2000). There, many terms are mere descriptions of concepts than precise labels, and there are less additional synonyms (with structural and lexical variations.) A good starting point for assessing possible patterns in name composition could also be the MeSH controlled vocabulary. Entries in MeSH typically contain many structural and lexical variations, a deeper understanding of which bears more insights than of orthographical or morphological variations.

Readers of this manuscript should either gain more insights into name compositions of gene names –in order to help refining dictionaries based on manual rule sets–, or be convinced that the idea of learning composition rules can be tackled in automated ways, promising examples of and basic techniques for which we discussed herein.

Supplementary information

The extracted set of rules for name variations and an extended dictionary for human genes, originating from Entrez Gene, are available at http://www.informatik.hu-berlin.de/~hakenber/publ/suppl/ . The dictionary can directly be used for matching entries against text and covers 32,980 genes. The main Java classes are available on request from the authors.

References

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, et al. 2000. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25:25–29.

Aaron M. Cohen and William R. Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):57–71.

Jeremiah Crim, Ryan McDonald, and Fernando Pereira. Automatically annotating documents with normalized gene lists. 2005. *BMC Bioinformatics*, 6(Suppl 1):S13.

Jörg Hakenberg, Steffen Bickel, Conrad Plake, Ulf Brefeld, Hagen Zahn, Lukas Faulstich, Ulf Leser, and Tobias Scheffer. 2005. Systematic feature evaluation for gene name recognition. *BMC Bioinformatics*, 6(Suppl 1):S9.

Jörg Hakenberg, Loic Royer, Conrad Plake, Hendrik Strobelt. 2007. Me and my friends: gene mention normalization with background knowledge. *Proc 2nd BioCreative Challenge Evaluation Workshop*, April 23-25 2007, Madrid, Spain.

Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, Juliane Fluck ProMiner: rule-based protein and gene entity recognition. 2005. *BMC Bioinformatics*, 6(Suppl 1):S14.

Ulf Leser and Jörg Hakenberg. 2005. What Makes a Gene Name? Named Entity Recognition in the Biomedical Literature. *Briefings in Bioinformatics*, 6(4):357–369.

Donna Maglott, Jim Ostell, Kim D. Pruitt, and Tatiana Tatusova. 2005. Entrez Gene: gene–centered information at NCBI. *Nucleic Acids Research*, 33(Database Issue):D54–D58.

Alexander Morgan and Lynette Hirschman. 2007. Overview of BioCreative II Gene Normalization. In: *Proc 2nd BioCreative Challenge Evaluation Workshop*, April 23-25 2007, Madrid, Spain.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–53.

Ariel S. Schwartz and Marti A. Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. *Proc Pac Sym Bio*, 451–462.

Hua Xu, Jung-Wei Fan, George Hripcsak, Eneida A. Mendonça, Marianthi Markatou, and Carol Friedman. 2007. Gene symbol disambiguation using knowledge-based profiles. *Bioinformatics*, 23(8):1015–1022.