Bioinformation by Biomedical Informatics Publishing Group *open access*

www.bioinformation.net **Hypothesis**

A cascaded approach to normalising gene mentions in biomedical literature

Hui Yang[1], Goran Nenadic[1,] *, John A. Keane[1]

[1]School of Computer Science, University of Manchester, Manchester, UK;
Goran Nenadic* - E-mail: G.Nenadic@manchester.ac.uk; * Corresponding author

revised September 30, 2007; accepted October 21, 2007; published online December 30, 2007

Abstract:

Linking gene and protein names mentioned in the literature to unique identifiers in referent genomic databases is an essential step in accessing and integrating knowledge in the biomedical domain. However, it remains a challenging task due to lexical and terminological variation, and ambiguity of gene name mentions in documents. We present a generic and effective rule-based approach to link gene mentions in the literature to referent genomic databases, where pre-processing of both gene synonyms in the databases and gene mentions in text are first applied. The mapping method employs a cascaded approach, which combines exact, exact-like and token-based approximate matching by using flexible representations of a gene synonym dictionary and gene mentions generated during the pre-processing phase. We also consider multi-gene name mentions and permutation of components in gene names. A systematic evaluation of the suggested methods has identified steps that are beneficial for improving either precision or recall in gene name identification. The results of the experiments on the BioCreAtIvE2 data sets (identification of human gene names) demonstrated that our methods achieved highly encouraging results with F-measure of up to 81.20%.

Keywords: gene name normalisation; gene name mapping; lexical variability; text mining

Background:

Finding, integrating and exploiting information on genes and proteins they encode is an essential task in the biomedical domain. Automated identification of gene and protein names in biomedical text is therefore a fundamental step in biomedical text mining. **[1, 2]** For example, the identification of both protein names that act as transcription factors and corresponding target genes is the first step in a semi-automated construction of regulatory networks from the literature. Gene/protein name identification refers to the process of linking a mention of a name in text to a relevant entry in a genomic database (e.g. Entrez Gene **[3]**, or UniProt **[4]**). However, due to the high level of variations, irregularities and ambiguity of the employed gene name nomenclatures associated with individual organisms, gene and protein identification remains a challenging task. **[1, 5, 6]** The problem is also magnified by authors using additional, non-standard "free" forms to refer to genes in the literature, and through ambiguity of gene names across different species.

The task of gene identification is generally addressed through a two-step process. The first step deals with the recognition of gene or protein mentions in text (gene mention identification). The aim here is to recognise strings that correspond to gene names. The second step provides a mapping of the detected mentions to standardised gene identifiers (gene name normalisation or mapping). It aims at generating a list of unique identifiers (typically from a referent genomic database) for each of the gene and protein mentions. The normalisation aids in treating different mentions associated with the same entity as equivalent, which is essential for information access and integration.

In this paper we focus on gene name normalisation. A gene candidate mention list can be produced by a gene tagger (e.g. **[7]**), and then each identified mention is to be assigned with a normalised unique gene identifier. Several approaches have been undertaken over recent years to provide solutions for gene name normalisation. **[8]** Dictionary-based methods have used existing terminological resources and various string matching approaches to locate gene mentions in text, and thus perform both tasks simultaneously (linking textual strings to matching database entries). Due to variability and ambiguity of gene names, simple pattern matching typically results in low precision and moderate recall (e.g. Hirschman *et al* **[5]** have reported extremely low precision rate (2% for full articles and 7% for abstracts) with recall in the range 31% (for abstracts) to 84% (for full articles) when using FlyBase). These approaches are generally enhanced with additional rule- and token-class based techniques, while distinguishing between important and less important constituents. **[6, 9, 10]** For example, Prominer **[11]** uses a gene dictionary that includes various spelling variants to support gene name matching, including an approximate matching procedure in which it treats each (candidate) string as a sequence of tokens, which are assigned to corresponding classes (e.g. measurement, digit, modifier, etc.). The classes are then used to weight mismatches in the approximate matching (e.g. the mismatch weight for the modifier class (which includes tokens such as receptor, precursor) is high).

Similarly, Tamames **[6]** tags every token in a document with a set of gene-relevant categories (e.g. core, chemical, type, location, etc.) and then allows for partial matching based on a scoring and evaluation of context. Post-filtering has been applied to help recognise valid matches in order to improve precision by using statistical techniques such as the maximum entropy model **[6]** and term occurrence associated with candidate gene identifiers **[12]**, or by employing machine learning (ML) techniques. ML techniques, in particular, may be difficult to develop as they need to rely on significant training data that would need to be generated for individual entries. **[13]**

The problems with gene name identification are well recognised in the text mining community, and several state-of-the art evaluations have been organised so far (e.g. BioCreAtIvE 1 and 2 **[10, 14]**). The variability of gene name usage is still enormous and there are several pending problems and difficult cases in linking mentions with database entries as identified by the BioCreAtIvE outcomes. For example, mentions that refer to multiple genes (e.g. gene families, enumerations, conjoined or range expressions of gene names, etc.) need additional work to map them to the corresponding entries. Although addressed by several researchers **[11]**, recognition and mapping of gene mentions that contain definitions of acronyms or gene symbols remains a challenge. The last BioCreAtIvE exercise also shows that difficult cases include permutations of gene names found in the lexicon. Finally, despite using different dictionaries and various spelling variation rules, there has been little discussion on their effect on precision and recall on a larger-scale task. Most of these issues are addressed in this paper.

We present a generic and effective cascaded method to match gene and protein mentions with unique identifiers by using a combination of exact and approximate matching between the mentions and dictionary entries. A gene name dictionary has been automatically re-engineered to support flexible matching, and a similar strategy (including morphological rules and linking orthographic variants) has been applied to gene mentions found in the literature. We differ from previous work by providing a canonical representation of gene synonyms and candidate gene mentions (from text), which are then compared in a cascade of different approaches, where more accurate steps are applied first. We also address multi-gene mentions (such as enumeration of gene names) and mentions that include text in parentheses. Finally, we systematically evaluate the effects of each step on precision and recall, and point to the main issues that still need to be solved.

Methodology:

Our gene name normalisation method works in several phases. The first phase is concerned with an automated creation and re-engineering of an extensive gene synonym dictionary. The second phase involves transformation and normalisation of gene mentions found in text. The following stage is related to exact matching, essentially based on a dictionary look-up which associates the gene mentions with the corresponding identifiers by using the gene synonyms dictionary generated in the first phase. Still, a simple lexicon look-up is often not sufficient to tackle highly variable and ambiguous gene mentions, especially for longer terms. Therefore, approximate matching uses a multi-stage procedure that includes, among others, permutation of components in gene names, and an approximate search in which a candidate synonym is allowed to contain one more component than a given mention. The major steps are explained below.

Automated generation of synonym dictionaries

Construction of lexical resources such as synonym dictionaries is a crucial step for a gene name normalisation system, as its quality and completeness affects the system performance. Dictionaries used in our system have been built fully automatically from two large, public, general genomic databases, Entrez Gene **[3]** and UniProt **[4]**. We have chosen the Entrez Gene database as a primary source, while UniProt has been used as a subsidiary one to enrich the content of the synonyms dictionary. Each entry in the synonym dictionary constructed consists of an Entrez Gene unique identifier and official symbol, along with a set of corresponding synonyms containing gene symbols and protein synonyms and aliases.

There are numerous ambiguous gene names, associated with multiple gene identifiers either from an inter-organism or from an intra-organism perspective. **[15]** Several systems have pruned their lexicons by moving ambiguous synonyms into a separate dictionary. **[11]** To avoid an extensive pruning of terms from the main dictionary, in cases where an official name was homonymous with another gene or protein synonym, we have kept only the official one in the corresponding entry, and moved the ambiguous aliases to a separate dictionary.

One of the main reasons why the recognition of gene or protein names in text is not trivial is that there are often variations in spelling (e.g. ‘IL-1’, ‘IL I’). Still, there is a fair amount of structure, regularity or common “patterns” in naming variations in genomic databases, which can give clues for reengineering these databases into a form that is more appropriate for text mining. In order to generate a suitable form of the dictionary, we have implemented a number of generic re-engineering rules that are geared towards representing gene names as regular expression patterns rather than strings. These rules are applied to normalise synonyms in the dictionary in order to resolve the problem of orthographic variants. The synonym normalisation procedure consists of the following main steps applied in the given sequence:

Organism prefix

Gene names appearing in the dictionary can begin with an organism prefix (e.g. ‘h’ or ‘hum’ for human; ‘p’ for yeast). For such gene names, this prefix has been made optional i.e. an additional gene name without organism indicator could be created, and added into the corresponding entry as a synonym.

Punctuation symbols
In general, strings contained within parentheses can be removed i.e. considered optional to form an alternative name (e.g. 'CD44 molecule' is considered as a variant of 'CD44 molecule (Indian blood group)'). There are some exceptions in which only the parentheses symbols are replaced by "white space" and strings within them are kept. These are: (1) if the string is a single character, and (2) if the parenthesized string consists of a combination of '+' and a digit, single letter, Greek letter, or chemical symbols. All other punctuation symbols are replaced with optional "white space".

Digits, Greek letters, Roman numbers and single letters
If a name contains digits, an additional synonym is made by separating the numerals from the rest of the name (e.g. 'RP13-16H11.4' generates 'RP 13 16 H 11.4'). Similarly, Greek letters appearing at the beginning or at the end are separated from names using a set of simple rules (e.g. 'Rev-ErbAalpha' gives 'Rev ErbA alpha'). A similar approach is undertaken for potential Roman numbers, taking into account the case of characters surrounding them (e.g. 'Rh VI' will be generated from 'RhVI', while 'ST3GALVI' will not produce an additional entry 'ST 3 GAL VI'). Finally, in some cases, single letters that start or end a gene name are also detached from it. More precisely, two cases are considered: if a word begins with a lower-case letter followed by a capital letter, and if the last letter differs in case from the preceding letter (e.g. 'bTrCP' will generate 'b TrCP'; 'CoA' will give 'Co A').

The application of the above rules has resulted in an alternative representation of the dictionary that contains spelling variations represented as a set of canonical representatives. For example, name 'alphaCP-4 protein' is represented as 'alpha_ε_CP_ε_4_protein' where '_ε_' denotes optional "white space" or separator. Obviously, canonical representatives may have resulted in some semantic distortion or ambiguity, which could impair precision. A simple way to limit this potential impact is to keep both the original and normalised forms of a synonym in the dictionary, but use the original forms first whenever possible. As a result, our dictionary contains two lists: the original synonyms list and the normalised canonical synonyms list.

Pre-processing gene mentions
Given a gene or protein mention in a document, we first apply a multiple-step mention pre-processing, which initially involves reducing plural forms to singular and identifying organism prefixes if any (as described for the dictionaries). However, unlike the parentheses "removal" approach used for dictionaries, we treat parentheses as a potential source of an alternative gene mention name that could be used for matching against the dictionary. For example, from candidate gene mention 'interleukin (IL)-17E' we may consider two possible candidates for matching: 'interleukin -17E' and 'IL-17E'. A context-driven rule is used to generate alternative mention strings. More precisely, if a token that follows the right parenthesis is a digit, Roman number, Greek or single letter, or an "activity" descriptor (e.g. receptor, transporter, activator), then the token is concatenated to the text fragment extracted from the parentheses and used as a candidate mention referred to as the inside candidate, while a candidate obtained by deleting the text within parentheses is referred to as the outside candidate.

Note that pre-processing can generate more than one candidate gene name mention from an original mention. All generated candidates will be used for matching, with priority-based filtering used in conflicting cases (see below for details).

At this stage we also treat potential multi-gene name mentions, represented by enumerations and/or coordination of gene names (e.g. 'ZNF133, 136'; 'ORP 3 to 6'). For each multi-gene type, rules have been written to handle it based on different contextual conditions. Table 1 (supplementary material) presents examples of different context types. Note that the type of conjunctive coordinations or punctuation ('/' or '-') affects the generated candidate gene names ('ORP 3 to 6' would be transformed into four candidate terms, i.e. 'ORP 3', 'ORP 4', 'ORP 5' and 'ORP 6', whereas 'ORP 3/6' only into two candidates: 'ORP 3' and 'OPR 6').

Following pre-processing, the same name normalisation steps described for dictionaries are applied to each candidate mention. This consequently results in two lists of gene mentions, one with original mentions, and one with alternative representations, which will be used for exact matching against the corresponding canonical dictionary entries (see below).

We also generate an additional gene mentions list, which contains gene candidate mentions produced by using a set of token-based transformations. More precisely, each gene name is first lexically analysed, and split into a set of constituents. Each constituent is assigned with a token class (type). A set of token classes includes Digit, Single-Letter, Greek-Letter, Roman-Number, Chemical-Element, Chemical-Name, Stop-Word, Non-Descriptive, UpperCase-Single-Word, etc. Elements of some classes are pre-collected from external resources (e.g. Stop-Word from NCBI PubMed **[16]**, and two chemical lists from the UMLS lexicon **[17]**). Non-Descriptive tokens represent biological terms that do not have significant impact on gene name matching (the token list has been obtained from the ProMiner project **[18]**). Other words that do not fall in any of the previous categories are tagged as Common-Word tokens.

A number of token-based transformation rules are then applied to map semantically equivalent or related tokens in order to build a set of candidate forms. For example, similarly to previous work (e.g. **[11]**), Roman-Number tokens are replaced with equivalent Digit tokens, and Greek-Letter tokens with corresponding Single-Letter tokens, etc. Tokens that are potential well-known acronyms appearing within candidate gene names are expanded with their long forms. **[11]** However, at this stage, we rely only on a list of the most frequent acronym-long form pairs, which has been precompiled

from the original gene name dictionary. Finally, if two distinct Single-Letter tokens appear in a gene name mention and they are adjacent to each other, then these two tokens are merged (e.g., mention IL 2 R a also generates IL 2 Ra).

The token-based transformations rules are applied iteratively until no new forms are generated. As a result, each candidate mention is associated with a set of additional transformed forms that will be used for exact-like matching. Obviously, the gene name transformations extend the range of potential mentions, and thus could potentially improve recall.

After the gene mentions are pre-processed and potential alternative forms generated, we apply a two-stage approach to map them against the gene dictionary.

Stage 1: Exact Matching
This first stage combines exact matching between original and normalised dictionary lists and the pre-processed gene mentions lists. It is performed as a three-step cascaded matching procedure that involves different dictionaries, as any still unmatched entries from the mentions list are carried forward to the next step:

Step 1: Exact matching between original gene name mentions and the original synonym list; the resulting pairs are stored in E_Org list.

Step 2: Exact matching between normalised unmatched gene name mentions and the normalised synonym list; the resulting pairs are stored in E_Norm list.

Step 3: Using still unmatched gene mentions, perform exact matching between their corresponding token-based forms and the normalised synonym list; the resulting pairs are stored in E_Norm_T list.

After each step (apart from Step 1), we consider matches that originated from the same mention (e.g. matches with and without organism prefix removed from the same mention). In cases of several matches for the same mention, priority-based filtering is applied. More precisely, in cases of matches that originated from an organism prefix and/or parentheses removal, we assign different priorities that reflect their importance in gene identifier recognition. For example, for the mention 'interleukin (IL)-17E', the full-name i.e. the outside form 'interleukin-17E' would have a higher priority than the "parenthesized" (inside) form 'IL-17E'. Similarly, in case of a gene mention that includes an organism prefix, matching the original mention is put ahead of a potential match of the generated mention without the organism prefix. If there are no priorities to rely on, a majority matching is selected, and in cases where this is not possible, the candidate is passed to the subsequent step.

Stage 2: Approximate Matching
Exact matching typically fails in handling two types of cases: components permutations (i.e. components contained in a gene mention do not appear in the same order as in its assumed matched synonym), and missing words (i.e. certain words are missing from either gene mentions or synonyms). Approximate matching is thus necessary to help identify potential synonyms that are likely to match mentions that have not been recognised in the first stage.

Similar to work described in **[6, 9, 11, 19]**, our approximate matching approach is a token-class, i.e. components-based, method. These approaches usually utilise some weighting schema, where weights have been estimated from a training data. As our approach aims to be easily adaptable for different species, we used a simpler method with "binary" weights: only classes specific for gene names (namely Digit, Single-Letter, Greek-Letter, Roman-Number, and Chemical tokens) are considered (equally) important for the matching process.

For the purpose of token-based approximate matching, tokenisation is first applied to both normalised mentions from text and normalised synonyms from the dictionary. We then carry out a two-step approximate matching approach.

In the first step we consider potential component permutations and components missing from a candidate dictionary entry. In case of permutation, two terms are treated as equivalent if their tokens match regardless of the order in which they appear (e.g. 'angiotensin II type 1 receptor' and 'angiotensin II receptor type 1'). The resulting set of matches is denoted A_Perm. In case of components missing from a candidate dictionary entry, non-specific tokens (Stop-Word and Non-Descriptive) are initially discarded from both the gene mention name and candidate dictionary entry. If the terms consequently have the same tokens, they are regarded as positive matches, which are denoted as A_NonSpec.

In the second step we consider cases where a potential dictionary entry contains one more token than a candidate mention. If the extra token belongs to a gene-specific class, the candidate synonym has to be discarded. More precisely, if a candidate dictionary entry contains all tokens of a gene mention (specific or non-specific) and an extra non-specific token, then this pair is accepted as matching (e.g. 'cytochrome c somatic' and 'cytochrome c'). The matching pairs resulting from this sub-step are denoted as A_Extra1. Further, if a candidate dictionary entry matches all the specific tokens of a mention, and the only surplus token does not belong to any specific token class, then the pair is added to a matching list called A_Extra2.

The overall two-stage matching approach is depicted in Figure 1.

Bioinformation by Biomedical Informatics Publishing Group *open access*

www.bioinformation.net **Hypothesis**

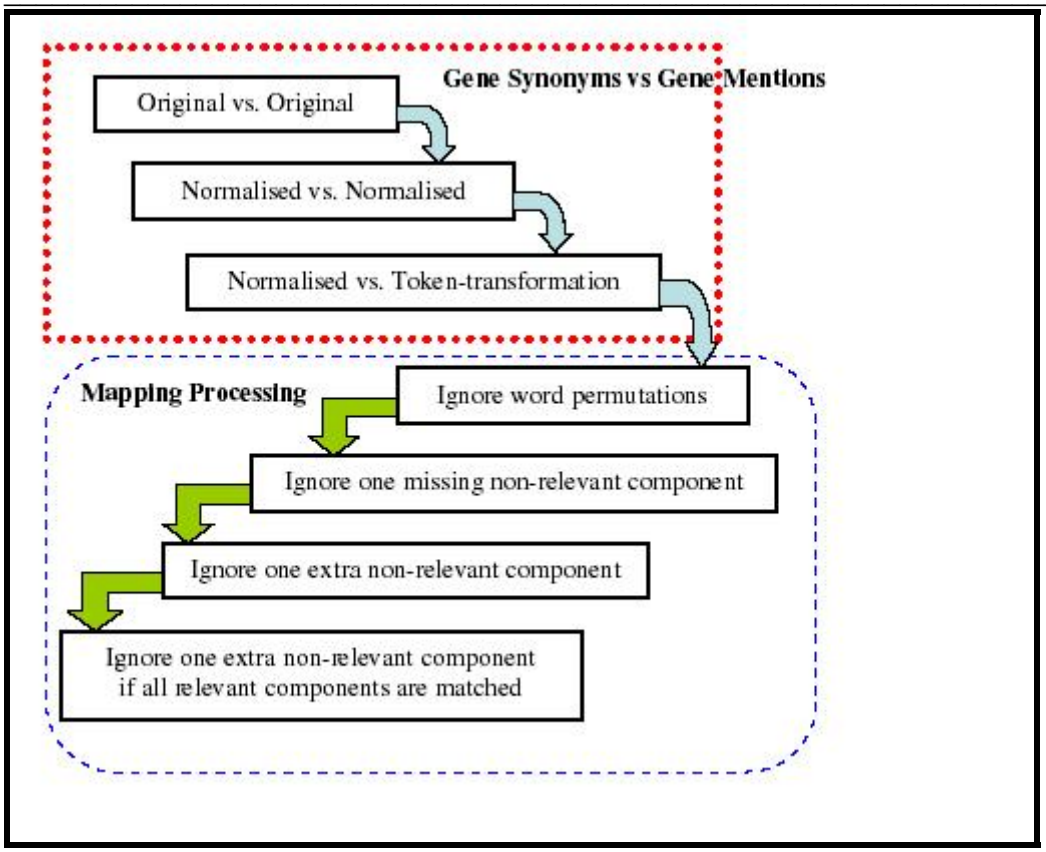

Figure 1: The main steps in the matching process

Experiments and results:

The experiments were conducted on the BioCreAtIve2 data. BioCreAtIvE (Critical Assessment of Information Extraction systems in Biology) is a community-wide exercise in evaluation of text mining and information extraction systems applied to solve specific tasks in the biological domain, including identification of gene name mentions and their normalisation, extraction of protein-protein interactions, protein function annotation etc. **[8, 14]**

For our experiments we used the BioCreAtIvE2 test and training data prepared for the human gene name normalisation (GN) task (see Table 2 in supplementary material for statistical information), in which gene name mentions have been previously marked. Our aim was to assess systematically the performance achieved by applying different matching approaches presented in this paper. We evaluated performance of the individual steps in the matching process on two different data sets which have been released for the GN task, as well as for their union. These two sets correspond to training (set-1) and testing (set-2) data for the BioCreAtIvE2 task. As our approach is rule-based, we have not used any of the sets for training, but for evaluation only. We report the results for each to allow for comparisons with other approaches and some discussions (since these two data collections may have slightly different distributions). We used standard evaluation metrics, namely recall, precision and F-measure shown in equation (1) (in supplementary material).

As described previously, to support the task we have re-engineered a synonym dictionary from the Entrez Gene database (see Table 3 in supplementary material for statistics). Adding additional entries from UniProt resulted in a 15.12% increase in the number of synonyms. Note also that the number of distinct normalised synonyms is smaller than the total number of distinct terms. The reason for this is that the normalisation process reduced some orthographic variation, so distinct gene names may have been mapped to the same normalised canonical form (e.g. both 'IL-1A' and 'IL1A' are normalised as 'IL_ε_1_ε_A').

In the following subsections we describe the results for each of the processing stages and steps (see Figure 2 for a reminder of the matching lists generated in each step).

Bioinformation by Biomedical Informatics Publishing Group *open access*

www.bioinformation.net **Hypothesis**

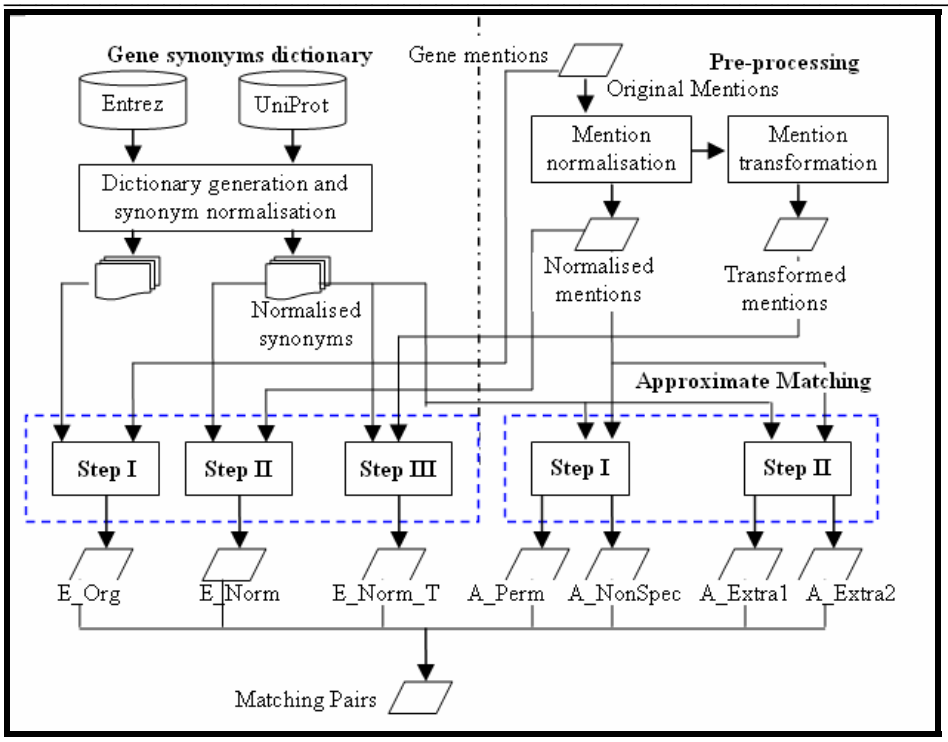

Figure 2: The overall approach and the matching lists in the gene name normalisation system

Evaluation of stage1: exact matching

Table 4 (see supplementary material) presents the results from the three exact matching steps. Around half of gene mentions in the two data sets can be easily and directly mapped into entries in the dictionary without any complex processing (E_Org list). Adding successfully matched normalised mentions (E_Norm list) has resulted in a significant improvement in recall (more than 10% of the total number of gene mentions). Among the specific approaches, organism prefixes were the most accurate, while canonical normalisation of gene mentions brought most additional matches with very high precision (98%) proving that it can be an effective method to reduce the effects of orthographic variants in gene name mapping.

As expected, processing multi-gene mentions did not perform as well, with precision ranging from 60% to 77% (overall 64%). One of the reasons might be the construction of the BioCreAtIvE2 gold standard that we have used for evaluation – for example, although we correctly parsed gene enumerated mention 'ZNF 133, 136 and 140' as referring to three candidates ('ZNF 133', 'ZNF 136' and 'ZNF 140'), the gold standard provided mapping for 'ZNF 140' only (so 'ZNF 133' and 'ZNF 136' were treated as false positives).

Generating candidates from mentions that contain parentheses proved to be useful for recall only to some extent. The precision of matches obtained from ignoring strings within parentheses ("outside" candidates) was high (cf. Table 4 (in supplementary material), parentheses removal), while it was unexpectedly low for candidates generated from the strings inside parentheses: we had 4 true positives and 8 false matches, giving the precision of 33% (data not shown). One possible explanation is that we have not used any sophisticated acronym resolution at this point to check if the string inside parentheses matches the context (e.g. 'ubiquitin-activating enzyme (E1)'). As generating new mentions from inside strings did not work well, it was not included in the final selection of normalisation steps.

The third exact matching step (E_Norm_T, using token-based transformations**)** resulted in encouraging results, with precision of 84% on average. The high precision performance proved that it can be a feasible way to improve opportunities of synonym matches by employing rules to produce more potential extended mentions with orthographic variants. It is interesting that these variants have not introduced many false positives as we have expected.

At the end of the exact matching stage, our system performed reasonably well (see Table 5 given under supplementary material), achieving overall F-measure of 76.36% (72.41% on set-1 and 79.71% on set-2), with precision of 95.68% and recall of 63.50%.

Evaluation of stage 2: approximate matching

The approximate matching stage resulted in four matching lists: A_Perm, A_NonSpec, A_Extra1, A_Extra2, and the corresponding results are presented in Table 6 (supplementary material). Overall, the token-based approximate matching resulted in recall improvement of 5.53% (7.34% on set-1 and 3.90% on set-2), while precision was in the range of 70%.

The permutation-based matching (A_Perm) achieved an outstanding performance in precision (comparable to exact matching), showing that constituent order had no significant influence on matching accuracy in gene name normalisation. The largest improvement in recall was made in matching that disregarded one extra non-specific token in a candidate dictionary entry (A_Extra1), in which a total of 65 new correct matches were returned. This result suggests that more than 3% of gene mentions miss one or more words when compared with the corresponding dictionary entries. Still, the precision of this step is the lowest of all the approximate matching steps.

Overall, the combined, two-staged cascaded approach (see Table 7 under supplementary material) achieved F-measure of 79.25% (precision 93.05% and 69.02% recall). Slightly better performance was achieved on set-2 (which is the BioCreAtIve2 test collection): F-measure of 81.20% (precision of 93.59% and recall of 71.72%), which is slightly better than the best performing system (F-measure of 81.1%) in the BioCreAtIve2 challenge.

Discussion:

We have analysed the effects of different matching approaches on performance. While exact and exact-like approaches (namely A_Norm, E_Norm, E_Norm_T, A_Perm) seem to be highly accurate and consistent in two data collections (set-1 and set-2), the performance of token-based approximate matching varied significantly. For example, A_Exact1 (ignore one extra non-specific token in a candidate dictionary entry) performed reasonably well on set-1 data, but the precision has decreased by almost 20% on set-2. One possible explanation for this is different distributions of the types of matching cases in the two data sets, which has been noted by the BioCreAtIvE participants.

Although generic, the gene mention normalisation approach proved to be useful and accurate, improving the overall recall by more than 7%, with precision of 97% (see Table 4 (supplementary material), RN row). In cases where multiple normalised candidate mentions have been generated for a single mention, in almost two thirds of cases all the generated candidates mapped to the same ID. This suggests that there are some common "patterns" or regularity in gene name variations that appear in the literature. In case of ambiguity (where several candidates for the same mention refer to different database entries), one of the approaches that we plan to employ is to do context-based post filtering by prioritising the mentions that occur elsewhere within an abstract over other candidates that have never occurred in the same document. This approach could be specifically useful for gene symbols, as Schuemie **[20]** reported that 30% of gene symbols in abstracts were accompanied by full names, and this document-wide distribution could help in gene identification.

When compared to other approaches that are based on exact matching using gene name dictionaries, our approach has better precision (93.59% compared to up to 84% of ProMiner **[11, 18]** on the same set-2 data) but lower recall (71.72% compared to 73-80%), suggesting that our rules for re-engineering the gene name dictionary were more restrictive and have not resulted in significant semantic distortion of normalised synonyms. This would also mean that the dictionary is geared for high precision applications.

Using normalised and transformed forms of gene mentions improved exact-match recall by almost 14%, while reducing the precision by only 2%. On the other hand, applying approximate matching improved recall by 5.53% (7.34% on set-1), with significant drop in precision (more than 10%). Our analysis of the false positive matches in approximate matching showed that there are three main types of errors that contribute to an increase in mismatching. As illustrated in Table 8 (supplementary material), the first category of errors corresponds to an extra word in a detected synonym that is an important component (such as receptor, protein or kinase). The second error type is related to surplus words contained in mismatched synonyms that belong to abbreviations of full-name gene mentions.

The third type of errors is related to ambiguous terms that have been removed from the synonym list. Obviously, the system would need to assign an approximate match in these cases as well. There are three possible solutions to improve precision in this case: (a) Identify important terms from the synonym list, which could not be missed in a gene mention when performing matching. (b) If an extra word in a detected synonym is composed of uppercase letters, filter out the match in the matching list. (c) The ambiguous synonym lists created during the dictionary re-engineering is used in gene name normalisation by a general disambiguation approach, e.g. based on machine learning.

While component permutations seem not to be important in matching, ignoring one extra non-specific constituent provided quite a few false positives (precision of 63%) as well as quite a few true positives (improving recall by 3%). This means that our notion of specific components was too vague, and would need to be reconsidered if a better precision is needed for a given task.

We have experimented with "switching" on and off some steps in the cascade, and analysed their effect on performance in order to identify "optimal sequences" for improving precision or recall. The main conclusion of this exercise is that each step resulted in an overall improvement of F-measure (so, an optimal cascade with regard to F-measure would contain all steps), while some of the steps positively improved precision and some recall. For example, much higher precision (96.60% compared to 93.05%) could be achieved by switching off the steps that relate to treating enumerations and ignoring one extra non-specific component, with overall drop in F-measure of only less than 1.5%.

A possibility to switch particular steps on and off could be used to "control" performance trade-offs between precision and recall to suit specific applications. For example, if gene name identification is used to support a

curation task (e.g. description of gene regulatory networks) then the cascade can be geared for a higher precision by eliminating the steps that introduce a number of false positives. On the other hand, if gene identification is part of a hypothesis generation task that uses text-based features, then the user may put more emphasis on recall and select the steps that bring in more true positives. Similarly, if an application is involved in mining relations between genes, then multi-gene mentions would be an important step to incorporate.

Conclusion:

Mapping gene and protein mentions identified in biological text to referent genomic databases remains an essential and challenging task due to lexical and morphological variation, and ambiguities in the existing gene nomenclature. The goal of our work is to explore the potential in using an extended synonyms dictionary combined with a number of matching approaches to enhance the performance of gene name normalisation. The experiments have shown that only a half of gene mentions could be mapped using a direct dictionary lookup. Therefore, in this paper we have described an automated generation of a synonym dictionary that is suitable for accurate matching of gene and protein names. A multiple-step mention pre-processing method has been proposed to resolve morphological variation and provide alternative "readings" of gene mentions, along with two gene name matching approaches. These approaches have shown a positive impact on matching performance, in particular exact-like matching which improves recall by 14% with only 2% reduction in precision. Ignoring component ordering within a gene mention proved to be an accurate approach. The results presented in our experiments demonstrated that each step resulted in improvement in either precision or recall, or both. Together, the two stages achieved an F-measure of 81.20% at 93.59% precision and 71.72% recall on the BioCreAtIve2 test collection, improving the best performing results in the field.

The proposed approach is generic and the procedures and rules introduced are not specific to any species (the only module that would need "customisation" is treating organism prefixes). The experiments have shown that this simple and generic approach can be used to efficiently normalise gene name mentions, and can be customised to suit improvement in precision or recall depending on the task that follows gene name normalisation.

Still, we view this work as a first step, with a number of interesting problems remaining open for further research. First, we intend to investigate ways to develop more effective matching algorithms, and provide an efficient Web service to the community that would normalise gene mentions in Medline abstracts. In addition, as mentioned previously, an abbreviation dictionary in transformation of gene mentions could be used to improve precision. We already have an acronym detection module that will be integrated in the matching process. Further, it will be necessary to consider potential ambiguous synonyms when matching using a wider context (e.g. abstract).

Acknowledgement:

The authors would like to thank anonymous reviewers whose comments were extremely helpful. This work was in part supported by the bio-MITA project ("Mining Term Associations from Literature to Support Knowledge Discovery in Biology"), funded by the UK Biotechnology and Biological Science Research Council (BBSRC).

References:

[01] M. Krauthammer, *et al.*, *J Biomed Inform.*, 37: 6 (2005) [PMID: 15542023]
[02] G. Nenadic, *et al.*, *Bioinformatics*, 19: 8 (2003) [PMID: 12761055]
[03] http://www.ncbi.nlm.nih.gov/entrez/
[04] http://www.expasy.org/sprot/
[05] L. Hirschman, *et al.*, *J Biomed Inform.*, 35: 4 (2002) [PMID: 12755519]
[06] J. Tamames, *BMC Bioinformatics*, 6: S10 (2005) [PMID: 15960822]
[07] R. McDonald, *et al*, *BMC Bioinformatics*, 6: S6 (2005) [PMID: 15960840]
[08] A. Morgan, *et al*, *J Biomed Inform.*, 37: 6 (2004) [PMID: 15542014]
[09] K. Fundel, *et al*, *BMC Bioinformatics*, 6: S15 (2005) [PMID: 15960827]
[10] L. Hirschman, *et al*, *BMC Bioinformatics*, 6: S1 (2005) [PMID: 15960821]
[11] D. Hanisch, *et al*, *BMC Bioinformatics*, 6: S14 (2005) [PMID: 15960826]
[12] J. Crim, *et al*, *BMC Bioinformatics*, 6: S13 (2005) [PMID: 15960825]
[13] H. Liu, *et al*, *J Am Med Inform Assoc.*, 9: 6 (2002) [PMID: 12386113]
[14] A. Yeh, *et al*, *BMC Bioinformatics*, 6: S2 (2005) [PMID: 15960832]
[15] O. Tuason, *et al*, *Pac Symp Biocomput.*, 238 (2004) [PMID: 14992507]
[16] http://www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhelp.html#Stopwords
[17] http://www.nlm.nih.gov/research/umls/
[18] http://www.scai.fraunhofer.de/ProMiner/
[19] L. Tanabe, *et al.*, *Bioinformatics*, 18: 8 (2002) [PMID: 12176836]
[20] M. J. Schumemie, *et al.*, *Bioinformatics*, 20: 16 (2004) [PMID: 15130936]

Edited by A.T. Heiny, T. W. Tan & S. Ranganathan
Citation: Yang *et al.*, Bioinformation 2(5): 197-206 (2007)

Bioinformation by Biomedical Informatics Publishing Group *open access*

www.bioinformation.net **Hypothesis**

Supplementary material

$$R = \frac{TP}{TP + FN} \qquad P = \frac{TP}{TP + FP} \qquad F - measure = \frac{2PR}{P + R} \qquad \rightarrow \qquad (1)$$

TP (true positive) is the number of correctly matched gene names, FN (false negative) is the number of genes not mapped by the system, and FP (false positive) the number of genes that are incorrectly normalised.

Tables:

Multiple gene name mention	Involved gene names
creatine kinase M **and** B	creatine kinase M; creatine kinase B
M and B creatine kinase	M creatine kinase; B creatine kinase
ZNF133, 136 and 140	ZNF133; ZNF136; ZNF140
Cofactors A, D, E	Cofactors A; Cofactors D; Cofactors E
AKR1C1-AKR1C4	AKR1C1; AKR1C2; AKR1C3; AKR1C4
Hmad-3/4	Hmad-3; Hmad-4
ORP-3 to 6	ORP-3; ORP-4; ORP-5; OPR-6

Table 1: Multiple gene mentions examples

	# abstracts	# gene mentions	# matched gene identifiers
Set-1 (“training” data)	281	985	995
Set-2 (“test” data)	262	1092	1100
Set-0 (total)	543	2077	2095

Table 2: The BioCreAtIve2 GN data collections

	# Entrez terms	# combined terms	# distinct terms	#distinct normalised terms
NCBI Gene ID	146,022	146,022	146,022	146,022
Gene Symbol	206,514	212,404	195,028	193,211
Protein Name	182,625	235,577	191,711	188,468
Synonym per ID	2.66	3.06	2.65	2.61
Ambiguous synonym	–	–	61,242 (13.68%)	66,302 (14.81%)

Table 3: Synonym dictionary statistics

		Set-1				Set-2				Set-0			
		TP	FP	P	R	TP	FP	P	R	TP	FP	P	R
E_Org		441	11	0.97	0.443	603	13	0.98	0.548	1044	24	0.97	0.498
	OP	10	0	1.00	0.01	18	0	1.00	0.016	28	0	1.00	0.013
	PR	7	0	1.00	0.07	7	1	0.87	0.006	14	1	0.93	0.007
E_Norm	MG	24	16	0.60	0.024	17	7	0.77	0.015	41	23	0.64	0.019
	RN	74	3	0.96	0.074	82	1	0.99	0.075	156	4	0.97	0.074
	Total*	115	19	0.85	0.115	124	9	0.93	0.113	239	28	0.90	0.114
E_Norm_T		28	4	0.87	0.028	19	5	0.79	0.017	47	9	0.84	0.022
Total Exact Macth		584	34	0.94	0.587	746	27	0.97	0.679	1330	61	0.96	0.635

Table 4: Results of individual steps in exact matching. OP (organism prefix), PR (parenthesis removal), MG (multi-gene mentions), RN(Rule-based normalisation)

	Set-1	Set-2	Set-0
F-Measure	72.42%	79.71%	76.36%
Precision	94.50%	96.80%	95.68%
Recall	58.69%	67.90%	63.50%

Table 5: Matching accuracy for exact matching in the three data sets

	Set-1				Set-2				Set-0			
	TP	**FP**	**P**	**R**	**TP**	**FP**	**P**	**R**	**TP**	**FP**	**P**	**R**
A_Perm	14	1	0.93	0.014	9	0	1.00	0.008	23	1	0.96	0.01
A_NonSpec	12	4	0.80	0.012	4	2	0.68	0.003	16	6	0.73	0.007
A_Extra1	38	14	0.73	0.038	27	24	0.52	0.035	65	38	0.63	0.031
A_Extra2	9	1	0.90	0.009	3	1	0.75	0.003	12	2	0.87	0.006
Total approx. matching	73	20	0.78	0.073	43	27	0.61	0.039	116	47	0.71	0.055

Table 6: Results of the individual steps in approximate matching

	Set-1	**Set-2**	**Set-0**
Total TP	657	789	1446
Total FP	54	54	108
F-Measure	77.02%	81.20%	79.25%
Precision	92.40%	93.59%	93.05%
Recall	66.03%	71.72%	69.02%

Table 7: Overall matching accuracy for gene name normalisation in the three data sets

Type of error	**Normalised mention example**	**Mismatched synonym example**
Important biological word missed	NK 3	NK 3 *receptor*
	I kappa B alpha	I kappa B *kinase* alpha
	IL 1 ra	IL 1 ra *homolog*
Abbreviation of full-name protein	CRP 2	CRP 2 *BP*
	Trx	*MT* Trx
	Protein kinase 2	*SFRS* protein kinase 2
Ambiguity	TR 1	TR ALPHA 1
	DGI	DGI 1
	Mada	MADA

Table 8: Examples of false positive matches in token-based approximate matching (the mismatched words are marked in *italics*)