Rule-based Gene Normalization with Statistical and Heuristic Confidence Measure

William Lau[1]
william.lau@nih.gov

Calvin Johnson[1]
johnson@mail.nih.gov

[1] High Performance Computing and Informatics Office, Division of Computational Biosciences, Center for Information Technology, National Institutes of Health, Bethesda, Maryland, United States

Abstract

In the gene normalization task, a rule-based approach has certain advantages including the fact that no gold standard is likely to contain all the genes that need to be considered. We have developed a rule-based algorithm that includes pattern matching for gene symbols and an approximate term searching technique for gene names. The algorithm performs confidence estimation by appropriately weighting measures of uniqueness, inverse distance, and coverage. An F-measure of 0.753 has been achieved, using nominal confidence-measure weights.

Keywords: gene normalization, rule-based, approximate term search, confidence measure

1 Introduction

The gene normalization algorithm we entered to Task 2 of the second BioCreAtIvE challenge is a prototype component of a text mining tool for genetic association studies [1]. The goal of this tool is to provide a means to systematically identify associations between sets of genes and diseases using information available in the MEDLINE literature. The assumption is that if the co-occurrence frequency between a gene and a disease is of statistical significance, they probably have an underlying biological relationship. Since simple string matching of the genes has yielded poor performance [3], we developed the gene normalization algorithm presented in this paper.

Several elements were taken into consideration when we designed the approach we employ herein. We chose a rule-based approach since many genes that we anticipated to encounter in the citations were not in the gold standard provided to us. We can tolerate a few random errors as they would not likely influence the association results. Given these factors, the algorithm we have developed is a balancing act between simplicity, accuracy, and computational efficiency.

2 Methods

2.1 Identification of Gene Mentions

The algorithm detects the occurrence of gene mentions by matching input text against an EntrezGene dictionary. The procedure effectively combines the tasks of gene detection and gene identifier lookup. Instead of using the lexicon provided to us, we created our own knowledge base with more comprehensive synonyms. Different approaches were used in the detection for gene symbols (including “Other Aliases” in the EntrezGene database) and the detection of gene names (including “Other Designations”). Gene symbol tagging is based on pattern matching. A set of regular expressions rules are applied to evaluate every string separated by space and punctuation symbols. The rules are commonly used in gene recognition tasks to account for syntactic variations, such as the interchange of Roman and Arabic numerals, placement of dashes

* This research was supported by the Intramural Research Program of the NIH, Center for Information Technology.

and spaces, case difference and plurality. For the official symbols, we also generate new symbols by expanding the associated Greek letters into their full names, e.g. "CHKB" to "CHK beta" and "beta CHK".

For gene names, additional synonyms are generated by replacing common chemical names with their abbreviations. An approximate term matching technique has been employed to detect gene names. After breaking a gene name into individual tokens, each token is searched against the text. Subsequently, the phrase containing the most tokens is identified. The ratio between the number of tokens in the mention candidate and the total number of tokens, r_m, needs to be higher than a threshold (0.7 in our submissions) for the phrase to be accepted. However, the candidate had to include specific tokens as measured by the number of citations containing those tokens. (If a token's frequency of occurrence is low, it is too important to be ignored.) The system also maintains a list of allowed and prohibited missing words. If a word in the prohibited list, e.g. "receptor", is missing from the phrase, the candidate is rejected. On the other hand, if a word in the allowed list, such as "type" and "subunit", is missed, the algorithm calculates the ratio as if the word were not in the gene name. In addition, the candidates are allowed to contain at most two additional words between any two tokens providing that the words are frequently found in the biomedical literature. This approximate matching technique, which is similar to that proposed by Hanisch *et* al [2], can accommodate typical variations of gene name mentions, such as word ordering, found in the literature.

2.2 Confidence Measure of Gene Mention Candidates

After a gene mention is detected, the algorithm calculates a confidence score using several statistical and heuristic measures. The three main factors used in our submissions were *uniqueness*, *inverse distance*, and *coverage*. Each of them contributed to 20% of the confidence score:

1. *Uniqueness* is an estimate of the probability that the candidate is referring to something other than the gene in question. If the mention has a very high frequency of occurrence in the literature, the score is reduced accordingly, because frequently occurring terms may have multiple meanings other than just being referred as genes.
2. For gene symbols, *inverse distance* is based on the edit distance of the candidate term to the formal reference in the database. It takes into consideration the variations in capitalization, ordering, and any omissions/additions of punctuations and spaces. The closer the mention matches the actual symbol, the higher the score. For gene names, since syntactic variations are common, the inverse distance is the harmonic mean of edit distance and r_m.
3. The calculation of the *coverage* score is quite different between gene names and gene symbols. For symbols, this score is calculated as follows:

$$\psi_S = \left(\frac{\tan^{-1}(2L-3)}{\pi} + 0.5 \right) \times s$$

where L is the symbol character length and, s is a scaling factor defined as:

$$s = \begin{cases} \left(e^{\frac{r_m - 1}{L}} \right)^2 & \text{if the candidate is enclosed} \\ 0.8 + 0.2e^{r_m - 1} & \text{otherwise} \end{cases}$$

The intuition is that the more characters the symbol has, the less likely it is that the term is used other than to represent the gene. If the term is enclosed by brackets, i.e. ([{}]), the gene name is probably mentioned in the text as well and score should be scaled accordingly.

For gene names, coverage is a weighted average of two ratios, r_l and r_m. r_l is the ratio between the character length of the candidate string and the corresponding name in the knowledge base. Thus,

$$\psi_N = \tfrac{1}{2} r_m^{\left(\frac{3 f_{th}}{f_m} + 1\right)} + \tfrac{1}{2} r_l$$

where f_{th} is the minimum occurrence frequency threshold for any missing words not in the allowed list (set to 20,000), and f_m is the occurrence frequency of the least common missing word. In addition to character length, the coverage metric for gene names also takes into account how many words are matched as well as the specificity of the words missing from the mention.

With 10% of the score reserved for future features, the remaining 30% of the confidence score was calculated using the factors listed in Table 1. Furthermore, we incorporated a boosting factor to reward or punish the candidate when there was other evidence in the text to suggest whether the mention actually referred to a gene. For example, if the text contained the chromosome location of the gene, its score would be boosted. If the mention was preceded or followed by supporting modifiers, such as "gene" and "encode", our level of confidence would increase. On the contrary, if counter-indicators, such as "test" and "cell line", appeared adjacent to the candidate, the score would drop. Whereas all the other factors were combined linearly to compute the final score, the boosting factor was added last as an exponent to the score.

Table 1: Factors used to calculate 30% of the confidence score when a gene mention is detected.

Factors	Contribution
Whether the mention is an official gene term?	18%
Whether more than one mention is detected for the gene?	10%
Whether the gene is approved by the HUGO Gene Nomenclature Committee?	2%

2.2 Disambiguation

When a string is associated to more than one gene identifier, the algorithm needs to determine which gene the authors actually intended. The disambiguation procedure is as follows. First, if a mention appears entirely within another longer mention, the algorithm removes the shorter mention. If some words of a mention overlap with another mention or if two mentions share the exact same term, the one with the lower score is removed. If the scores of two conflicting candidates are equal, their uniqueness scores are both reduced by half. If the candidate had more than one form of occurrence, e.g. both the symbol and the name were detected, the highest score was considered.

3 Analysis

Table 2 shows the performance of our three submissions to the competition. Only candidates with a confidence score higher than a threshold (t_a) were accepted. Since the time of the submissions, we fixed several minor bugs in the system, and an F-measure of 0.753 ($t_a = 0.65$) was achieved on the same set of data. We are currently investigating optimal weights for the various confidence measures, as opposed to the equal (0.2) weighting used here. Preliminary evidence suggests that a greater weight is appropriate for the uniqueness measure followed by the coverage score, and that the inverse distance weight should be reduced.

Table 2: Results of the submissions generated with different acceptance thresholds of the confidence scores.

Run	Threshold (t_a)	Precision	Recall	F-measure
1	0.6	0.655	0.796	0.719
2	0.625	0.690	0.782	0.733
3	0.65	0.726	0.749	0.737

References

[1] Becker, K.G., Hosack, D.A., Dennis, G.,Jr, Lempicki, R.A., Bright, T.J., Cheadle, C., et al., PubMatrix: a tool for multiplex literature mining, *BMC Bioinformatics*, 10(4):61-66, 2003.

[2] Hanisch, D., Fundel, K., Mevissen, H.T., Zimmer, R., Fluck, J., ProMiner: rule-based protein and gene entity recognition. BMC Bioinformatics, 6(Suppl 1):S14, 2005.

[3] Jenssen, T.K., Laegreid, A., Komorowski, J., Hovig, E., A literature network of human genes for high-throughput analysis of gene expression, *Nat.Genet.*, 28(1):21-28, 2001.