BIOINFORMATICS
Vol. 1 no. 1 2003
Pages 1–8

Inducing Novel Gene-Drug Interactions from the Biomedical Literature

Sepandar D. Kamvar[1], Diane E. Oliver[2], Christopher D. Manning[1] and Russ B. Altman[2]

[1]Department of Computer Science, Stanford University, and [2]Department of Genetics, Stanford Medical Informatics, Stanford University,

ABSTRACT

Motivation: Knowledge about the interactions between genes and drugs is important in determining the efficacy and toxicity of medications. We present a supervised learning algorithm for inducing previously unknown gene-drug interactions by text-mining the biomedical literature. This algorithm takes as its input a set of known gene-drug relationships and a literature source. New gene-drug interactions are induced based on similarity to known gene-drug interactions, where similarity is determined according to the biomedical literature.

Results: Based on limited training data (258 gene-drug pairs), the algorithm induces correct unseen gene-drug relationships at precisions of over 60%, including gene-drug relationships that are not obvious from name similarity.

Availability: Please contact authors for availability.

Contact: sdkamvar@stanford.edu

INTRODUCTION

There is great variability in the way individuals respond to pharmaceutical drugs, both in terms of the toxicity of the drug and the efficacy of the treatment. This variability has several known causes, including environmental factors, clinical variables such as age and nutritional status, and inherited genetic differences.

A classic example of the effect of genetic variation on drug response is the variation of individual responses to the anti-leukemia drug 6-mercaptopurine. While most people metabolize the drug quickly, individuals with a genetic variation for thiopurine methlytransferase (TPMT) metabolize the drug slowly, leading to greater host toxicity in those individuals. The field of pharmacogenomics studies how genetic variation affects individual drug responses.

Recently, a large number of genetic discoveries have been reported in the biomedical literature, due largely to the sequencing of the human genome (Lander *et al.*, 2001) and novel technologies such as the microarray (Fodor *et al.*, 1991). It has been suggested that a principal challenge in the analysis of this data is the difficulty in linking information about the variation in human genes to the variation in drug response (Klein *et al.*, 2001).

We present here an algorithm that uses the biomedical literature to suggest gene-drug pairs that are likely to interact. (We define a gene-drug interaction to be an interaction between a drug and a gene-product that influences the metabolism of the drug or the action of the drug on a drug target. This is often also called a phamacogenetically significant gene-drug relationship.)

This algorithm is based on the idea that a gene-drug pair is likely to interact if the gene and drug are "similar" to the gene and drug in a pair that is known to interact. In order to use this idea to induce gene-drug interactions, it is necessary to define a notion of similarity between genes and between drugs. In this work, we say that the degree of similarity of two drugs is given by the degree of similarity of the literature about the two drugs. Likewise, we say that the degree of similarity of two genes is given by the degree of similarity of the literature about the two genes.

Discovering gene-drug interactions based on "guilt-by-association" analyses of the literature is reasonable because there are many ways in which these interactions can be similar. Drugs may be related by mechanism of action, indications for use, chemical structure, or side effects. Genes may be related by sequence, structure, expression, pathway membership, or cellular compartment. The literature is likely to discuss many of these areas, and thus may be a useful key for finding other gene-drug pairs that interact based on similar features. This work is predicated on the hypothesis that the signal from the literature will be sufficiently coherent and strong to suggest new interactions.

We present a novel supervised learning algorithm for inducing gene-drug interactions based on these ideas. This algorithm takes as its input known gene-drug interactions and a literature source, and suggests candidate gene-drug pairs that are likely to interact. Empirically, we show that this algorithm predicts novel and nontrivial gene-drug interactions. (Here, we use *nontrivial* to indicate those gene-drug interactions that could not reasonably be suspected by string similarity to gene names or drug

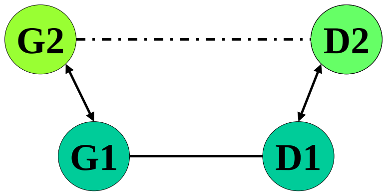

Fig. 1. If gene G_1 interacts with drug D_1, and gene G_2 has the property P that causes G_1 to interact with D_1, and drug D_2 has the property Q that causes D_1 to interact with G_1, then G_2 will interact with D_2. If we know the probabilities represented by the solid lines, we may induce the probability that G_2 will interact with D_2, represented by the dotted line.

names in the training set.) Furthermore, it predicts these interactions with remarkable accuracy; over 60% of the interactions it predicts in experiments have been shown to be correct.

METHODS

A Probabilistic Framework

The algorithm is based on the idea that gene-drug pairs that are "similar" to gene-drug pairs that are known to interact are likely to interact. Let us formalize this notion. Imagine that we know that gene G_1 interacts with drug D_1, which we will write as $int(G_1, D_1)$. Suppose that gene G_1 interacts with drug D_1 because gene G_1 has a certain property P, and drug D_1 has a certain complementary property Q. We write this as $sp(G_1, G_2, P)$, $sp(D_1, D_2, Q)$. We can then say that if another gene G_2 has property P and another drug D_2 has property Q, then drug D_2 interacts with gene G_2. (Of course, this is a highly simplified abstraction, but it is useful in the formulation of the algorithm, and the empirical results of the algorithm suggest that it is not harmful to use such a simple abstraction.)

Let us write this proposition in the language of logic.

Proposition 1:

- IF gene G_1 interacts with drug D_1
  - AND gene G_2 has the property P that causes gene G_1 to interact with drug D_1
  - AND drug D_2 has the property Q that causes drug D_1 to interact with gene G_1
- THEN gene G_2 interacts with drug D_2.

Now, we may use Proposition 1 to identify the probability that drug D_2 and gene G_2 will interact:

$$p(int(G_2, D_2)) = p(int(G_1, D_1)p(sp(G_1, G_2, P) \times p(sp(D_1, D_2, Q)) \quad (1)$$

The probability of interaction for a gene-drug pair $p(int(G_1, D_1))$ represents our level of certainty that Gene G_1 and Drug D_1 interact. This can be given by supervisory information. For example, if the gene-drug pair (G_1, D_1) is known to interact, then $p(int(G_1, D_1)) = 1$. One can find information on known gene-drug interaction in databases such as the PharmGKB database (http://www.pharmgkb.org/). The other terms in equation 1 are a little more problematic. In most cases, the properties P and Q that cause drug D_1 and gene G_1 to interact are not known. However, we can use the following intuition to assign these probabilities. The greater the similarity between genes G_1 and G_2, the higher the likelihood that genes G_1 and G_2 share property P. Likewise, the greater the similarity between drugs D_1 and D_2, the higher the likelihood that drugs D_1 and D_2 share property Q. Therefore, we can define probabilities $p(sp(G1, G_2, P))$ and $p(sp(D_1, D_2, Q))$ *even if we don't explicitly know the properties P and Q that cause drug D_1 and gene G_1 to interact.*

Based on this intuition, let us estimate the probability that G_1 and G_2 share property P as:

$$p(sp(G_1, G_2, P)) \approx sim(G_1, G_2)$$

where $sim(G_1, G_2)$ is a measure of the textual similarity between descriptions of the two genees. We assume $sim(G_1, G_2) = 1$ if G_1 and G_2 are the same gene, and $sim(G_1, G_2) = 0$ if G_1 and G_2 have no properties in common.

Likewise, let us estimate the probability that D_1 and D_2 share property Q as:

$$p(sp(D_1, D_2, Q)) \approx sim(D_1, D_2)$$

Equation 1 now becomes:

$$p(int(G_2, D_2)) = p(int(G_1, D_1))sim(G_1, G_2) \quad (2)$$
$$\times sim(D_1, D_2) \quad (3)$$

Let us assume for the moment that we have a suitable similarity function for drugs and genes. We show how to derive these similarity functions in Section .

Evidence Combination

In the previous section, we described the case where information about only one gene-drug interaction is known (the interaction between G_1 and D_1. We now consider the case where information about more than one gene-drug interaction is known. Imagine now that we know that gene G_1 interacts with drug D_1 and also that gene G_3 interacts with drug D_3. We now have the following proposition.

Proposition 2:

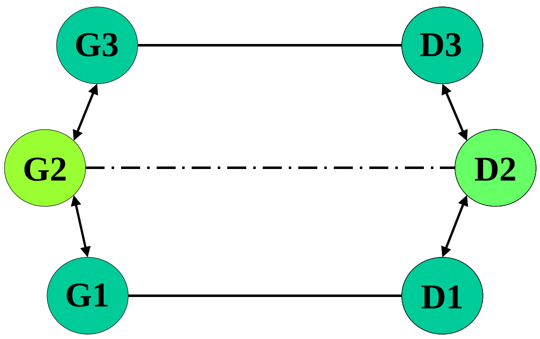

Fig. 2. Evidence Combination. If information is known about many gene-drug interactions, then this knowledge may be combined to determine the probability that gene G_2 and drug D_2 will interact.

- IF gene G_1 interacts with drug D_1
  - AND gene G_2 has the property P_1 that causes gene G_1 to interact with drug D_1
  - AND drug D_2 has the property Q_1 that causes drug D_1 to interact with gene G_1
- OR
- IF gene G_3 interacts with drug D_3
  - AND gene G_2 has the property P_3 that causes gene G_3 to interact with drug D_3
  - AND drug D_2 has the property Q_3 that causes drug D_3 to interact with gene G_3
- THEN gene G_2 interacts with drug D_2.

We may write Proposition 2 in terms of probabilities in the same way we wrote Equation 1 from Proposition 1. In Equation 1, we used the fact that, given two independent events A and B, $p(A \text{ AND } B) = p(A)p(B)$. This is often called a "noisy-AND" logic gate in probabilistic reasoning. Analogously, we may use the "noisy-OR" logic gate in expressing Proposition 2. (See (Russell & Norvig, 1995) for a review of probabilistic reasoning.)

$$p(A \text{ OR } B) = 1 - (1 - p(A))(1 - p(B)) \quad (4)$$

Let us define,

$$p(A) = p(int(G_1, D_1))sim(G_2, G_1)sim(D_2, D_1)$$

as in Equation 3. Likewise, let us define

$$p(B) = p(G_3, D_3 \text{ interact})sim(G_2, G_3)sim(D_2, D_3)$$

Therefore, from Proposition 2 and Equation 4:

$$P(G_2, D_2 \text{ interact}) = 1 - (1 - p(A))(1 - p(B)) \quad (5)$$

Data : A Set of Gene-Drug Pairs that are known to interact.
Result : The probability that Gene G will interact with drug D.

function *induceProbability(G,D)*
  $c = 1$;
  foreach *known interaction* (TG,TD) **do**
    $c = c(1 - sim(G,TG)sim(D,TD))$;
  endFor
  $prob = 1 - c$;
  return $prob$;

Algorithm 1: Algorithm to induce the probability that gene G and drug D will interact if several gene-drug interactions are known.

Algorithm 1 uses this idea to induce novel gene-drug interactions when supervisory information about multiple gene-drug pairs is known.

Similarity Function

It now remains to define a similarity function $sim(x, y)$ where x and y may be two genes or two drugs. Many similarity functions have been proposed for genes, including those based on sequence homology and expression data (Mount, 2001). Likewise, one may use similarity in chemical structure or mechanism of action to determine similarity in drugs.

We choose to use a similarity function based on the biomedical literature because the literature is likely to reflect information about many of these areas and others, and therefore is a more direct and relevant basis for observing similarity.

We determine the similarities as follows. We first collect a body of representative literature about each gene and drug of interest. For genes, we use relevant text in the LocusLink entry for the gene (http://www.ncbi.nlm.nih.gov/LocusLink/). This consists of the following sections: Overview, GeneRIFs (Gene References into Function) and associated Medline abstracts, and Gene Ontology terms. We get all the text we need using a hand-crafted web crawler on the LocusLink website. For drugs, we use the relevant text in the USPDI Drug Database (Micromedex, 2002). This consists of the following sections: Indications, Mechanism of Action/Effects, and Other Actions/Effects.

Once we have collected literature for each drug and gene of interest, we create a list of domain-specific stop words. We define stop words as those words that appear in only one description, or in over 75% of the descriptions. This method of identifying stop words has been used with success in similar applications (Chang *et al.*, 2001; Yang & Pedersen, 1997).

It is important to note that we have a single description

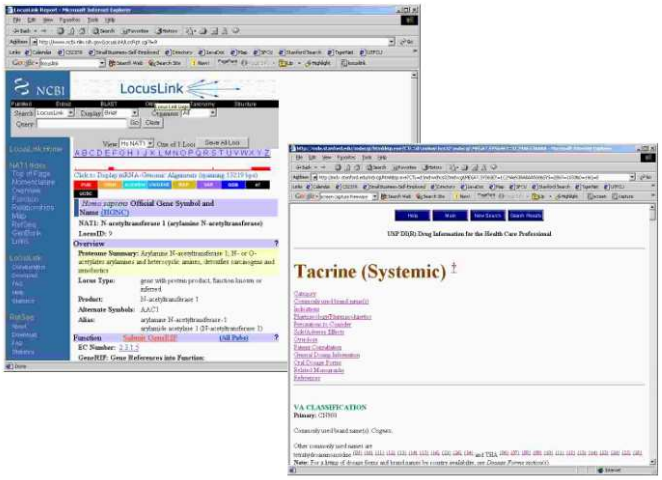

Fig. 3. Literature sources for determining similarity between genes and between drugs. The literature source for genes is LocusLink (http://www.ncbi.nlm.nih.gov/LocusLink/), and the literature source for drugs is the USP DI (United States Pharmacopeia Drug Information) drug database, which is obtained on the web through Micromedex (http://www.micromedex.com/).

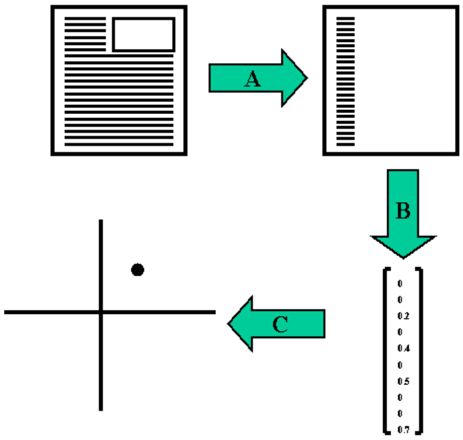

Fig. 4. Transforming text to vector representations. A. The document is first tokenized to words, and a list of stop words are taken out. B. The word counts for each document are stored in a vector, and the vector is normalized so that its length is 1. C. The resulting vector can be represented as a point in multidimensional "word" space. Points that are close to each other in word space represent documents that are similar to each other.

for each gene, and a single description for each drug. An alternative choice may be to use all the Medline abstracts that arise from a Medline search for each gene or drug. We chose not to use this approach because many genes return hundreds of Medline abstracts for a single query. In these cases, the data is often noisy and nonspecific, in that much of the content of the abstracts does not refer to the gene. LocusLink and USP DI give a concise and fully relevant description of each gene and drug, and are therefore more useful for our purposes.

We then create a vector space representation of each gene and drug based on its description. In the vector-space representation, each gene (and drug) is represented as a vector where each dimension represents the number of times the word appeared in the description of the gene or drug (Manning & Schütze, 1999). We then normalize each of these vectors so that its length equals 1.

The similarity between two genes (or drugs) is given by the dot-product of the literature vectors of the two genes (or drugs).

$$sim(G_1, G_2) = \vec{g_1} \cdot \vec{g_2} \tag{6}$$

For normalized vectors, this dot product gives the cosine of the angle between the vectors, and so this measure is known in information retrieval as the cosine similarity measure (Manning & Schütze, 1999). Descriptions with similar content yield scores close to 1, while those with dissimilar content yield scores close to 0. Therefore, it is plausible to use these scores to approximate the probabilities as discussed above.

The text processing steps are summarized in Figure 5.

Methods Summary

In summary, the algorithm proceeds as follows:

1. Identify a training set of genes-drug pairs that are known to interact.
2. Identify a list of genes and drugs of interest.
3. For each gene and drug of interest, create a vector representation as in Figure 5
4. For each gene-drug pair of interest, determine the probability of interaction using algorithm 1 and equation 6. Those gene-drug pairs that are assigned high probabilities are likely to interact, and those assigned low probabilities are unlikely to interact.

RESULTS AND DISCUSSION

In our experiments, we used a sample set of 137 drugs and 197 genes. These genes and drugs were chosen at random from the PharmGKB database (Hewett *et al.*, 2002), (http://www.pharmgkb.org). The vocabulary size was 1314 words for the genes, and 5399 words for the drugs. In this set, there were 516 gene-drug pairs that are known to interact. These known interactions were found in the PharmGKB database. We chose 258 of these pairs at random to use as a training set. The remaining 258 were used as a test set for validation purposes. We then used

Given	Predicts	With Probability	Known Interaction?
ACHE-neostigmine	ACHE-ambenonium	.99	Yes
ACTN3-tacrine	ADD1-tacrine	.46	Yes
ADRA1D-dapiprazole	ADORA3-dapiprazole	.96	Yes
GPX5-heparin	SMS-heparin	.19	No
CACNA1C-demecarium	CACNA1C-primaquine	.22	No

Table 1. Some examples of gene-drug pairs induced by the algorithm, given only one training example. Note that those pairs assigned a high probability are likely to interact, while those that are assigned low probabilities are likely not to interact. Also, notice that these inductions are not trivial.

Text Processing Summary

- For each gene
  - Look up LocusLink entry for gene, and extract text from the following sections of entry: Overview, GeneRIFs, and GO Codes.
  - Tokenize text to form list of words.
- For each drug
  - Look up USP DI entry for drug, and extract text from the following sections of entry: Indication, Mechanism of Action/Effects, Other Actions/Effects.
  - Tokenize text to form list of words.
- Form a stopword list that contains all words that appear in only one description, or in over 75% of the descriptions.
- For each gene G_i, create vector representation $\vec{g_i}$ as in figure 4
- For each drug D_i, create vector representation $\vec{d_i}$ as in figure 4

These vectors are used in the similarity function given by equation 6.

Fig. 5. Outline of Text Processing Method. The text processing was implemented by a web crawler and textual similarity tool written in Java.

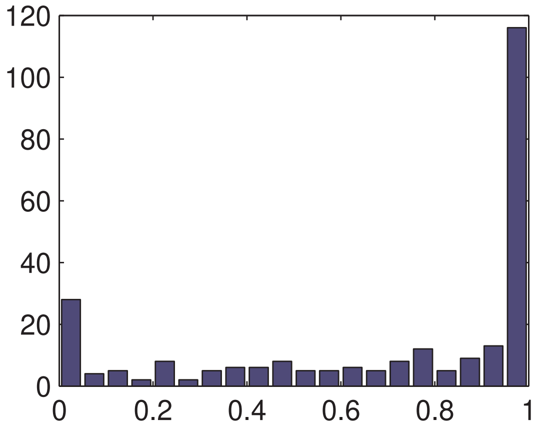

Fig. 6. Histogram of assigned probabilities for each known gene-drug interaction in the test set. The x-axis represents bins of assigned probabilities, and the y-axis represents the number of pairs that fall into each bin. As desired, most gene-drug interactions in the test set are assigned high probabilities. There are a also some that are assigned near-zero probabilities because the lack of training data. The size of the training set here is 258 interactions.

Algorithm 1 to assign probabilities to every possible gene-drug combination in the set of interest.

The desired behavior of the algorithm is to assign high probabilities to gene-drug pairs that actually interact, and low probabilities to gene-drug pairs that don't interact.

We first check to see if the known gene-drug interactions in the test set are assigned high probabilities. The probabilities of the gene-drug interactions in the test-set are given in figures 6 through 8. These figures show that most known gene-drug interactions in the test set are assigned high probabilities by the algorithm. This is exactly the desired behavior, because it suggests that pairs that are assigned high probabilities are likely to interact.

One will note that in figures 6 and 7 there are also a fair number of interactions in the test set that are given zero probabilities by the algorithm. This is because these interactions are not predictable by the training data. That is, neither the gene or drug in the interaction is similar to any gene-drug pair that is given in the training set. Therefore, there is no way to predict these by our algorithm. Figure 8 shows that this number gets smaller as we get more training data.

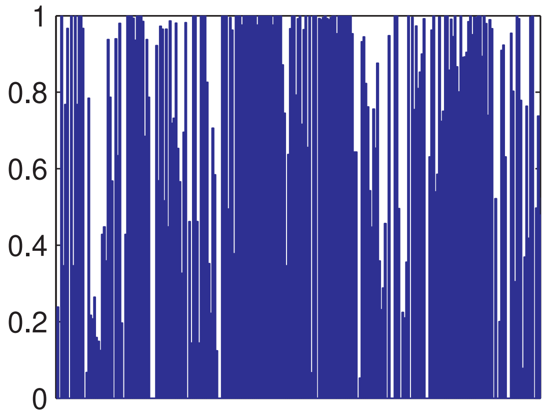

Fig. 7. Bar graph for assigned probabilities for interactions in the test set. On the x-axis is each individual interaction, and the y-axis is the assigned probability. Like figure 6 this shows that most interactions in the test set are assigned high probabilities.

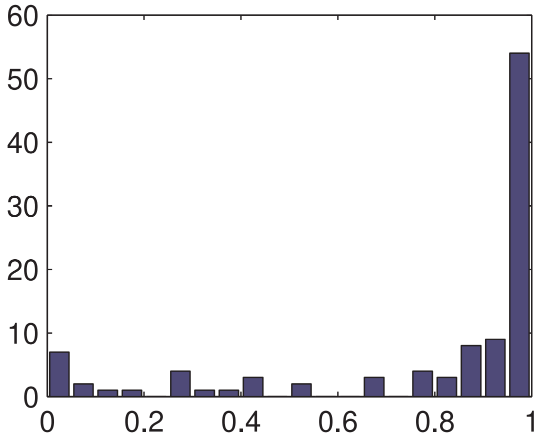

Fig. 8. Histogram of assigned probabilities for each known gene-drug interaction in the test set, where the training set is larger (412 interactions). Note that the number of interactions in the test set that are assigned near-zero probabilities is decreased.

We then check the converse: that those gene-drug pairs that are known not to interact are given low probabilities. Figure 9 shows that most of the drug-gene pairs that are not in the training or test set are assigned low probabilities. In fact, most are assigned probabilities of zero. It is interesting to note that the mean interaction probability assigned to the pairs in the test set (pairs that are known to interact) is 0.671, while the mean interaction probability assigned to the pairs that are not in the training or test set (pairs that are surmised not to interact) is 0.036. Again, this is exactly the desired behavior.

Table shows a few typical gene-drug pairs that are known to interact, and some gene-drug pairs that are known not to interact, and the probability of interaction assigned to these pairs by the algorithm.

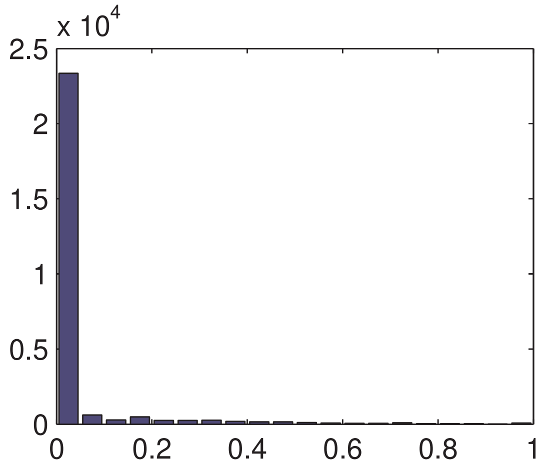

Fig. 9. Histogram for gene-drug pairs that are *not* in either the test or training set. These gene-drug pairs are surmised not to interact. As desired, the interaction probability assigned to most of these is near-zero.

We wish to quantify the correlation between the interaction probability assigned by the algorithm and the likelihood that the gene-drug pair will actually interact. We define the precision of the algorithm on a certain set of induced gene-drug pairs to be the number of pairs in that set that interact divided by the total number of pairs in that set. For example, we may have our algorithm return the top five gene-drug pairs with highest assigned probability. The precision of that set would be the number of those returned pairs that actually interact (that is, those that are in the test set), divided by five. This is called the precision of the algorithm at a cutoff of five. Figure 10 shows the precision at different levels of cutoff.

Alternatively, we may choose to cut off at a certain probability. For example, we may have our algorithm return all gene-drug pairs that have an assigned probability of over .9. Figure 11 shows the precision at different assigned probabilities.

In each of these figures, 258 gene-drug pairs are chosen at random from the PharmGKB database and used as training data. Then the accuracy is measured for subsets of top 150 induced gene-drug pairs. In figure 10 these subsets are the top n ranked gene-drug pairs, and in figure 11, the subsets are those gene-drug pairs that are assigned above a certain probability. In each of these experiments, the results are averaged over 10 runs where different training examples are chosen at random.

There are two things that are interesting to note in these graphs. First, there is clearly a correlation between the probability of interaction assigned by the algorithm and the actual probability of interaction as measured by the percent of induced pairs that are known to interact. This validates our original assumptions, and gives a

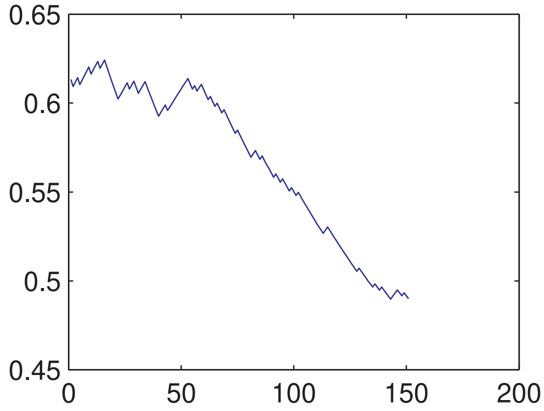

Fig. 10. Precision vs. Cutoff with 258 training examples. The x-axis represents the rank of the lowest-scoring induced gene-drug pair, and the y-axis represents the percentage of induced gene-drug pairs of that rank and higher that are known to be actual gene-drug relationships. Notice that this represents a lower-bound on the precision, since some induced gene-drug pairs that are not known to interact may be shown to interact. True precision numbers may only be achieved by testing each induced pair.

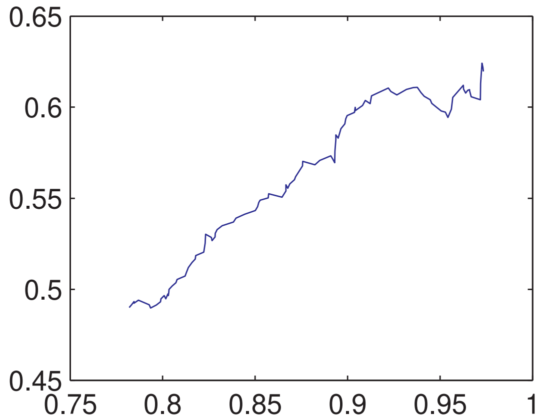

Fig. 11. Precision vs. Induced Probability with 258 training examples. The x-axis represents the probability that the algorithm assigns to a certain gene-drug interaction, and the y-axis represents the percent of gene-drug pairs assigned that probability and higher that are known to interact. Notice the correlation: the higher the probability assigned by the algorithm, the more likely it is that the pair will interact.

meaningful interpretation of the probabilities assigned by the algorithm.

The second interesting point is that this algorithm is surprisingly accurate for such a difficult task. A system that would choose gene-drug pairs at random would attain a precision of 1.9% for these experiments. Furthermore, it is important to note here that, unlike many tasks in information retrieval, modest precision numbers are acceptable because the finding of a novel

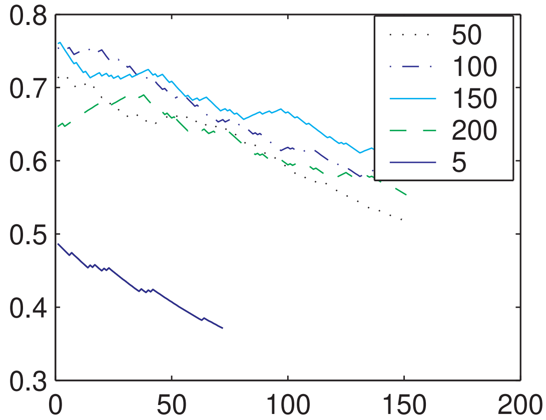

Fig. 12. Precision vs. Cutoff for different numbers of training examples. The legend shows the number of training examples that correspond to each curve. The x-axis represents the cutoff number, and the y-axis represents precision. The curves for 50-200 examples are similar. With only 5 examples, the algorithm clearly suffers.

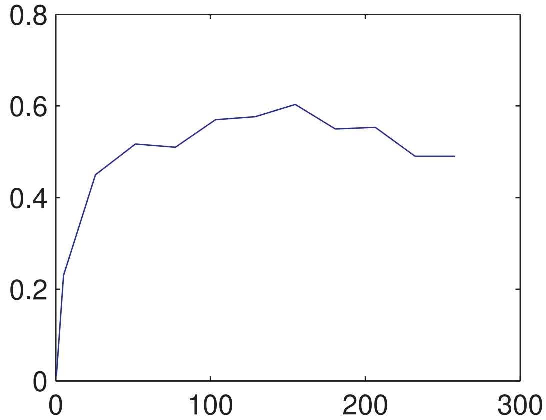

Fig. 13. Precision vs. Number of Training Examples. The x-axis represents the number of training examples, and the y-axis represents the precision at a fixed cutoff of 300 induced gene-drug interactions. This also shows that adding training examples is very helpful up to 50 training points, and starts become less helpful after that.

gene-drug interaction is valuable enough that it is worth investigating several candidate gene-drug pairs. In this light, the precisions shown in figures 11 and 10 are even more remarkable.

Since this is a supervised learning algorithm, it is useful to know the extent to which adding more training data helps the algorithm. Figures 12 and 13 shows that it is certainly useful to add training data when there is very little training data, but it doesn't help much after a certain point. For this experiment, adding more training data is useful up to 50 training points. After that, more training data doesn't make much difference.

Finally, it is important that the algorithm be able to induce gene-drug interactions that are not obvious from the training data. For example, it is not interesting if the algorithm induces that CA12 (Carbonic Anhydrase XII) interacts with tacrine, given that CA11 (Carbonic Anhydrase XI) interacts with tacrine. However, it is interesting if the algorithm induces that ACTN3 (alpha 3 actinin) interacts with tacrine, given the same information. Our algorithm does predict both of these. Table shows some examples of some nontrivial interactions induced by the algorithm.

It is somewhat surprising that a literature-based method for defining similarity, and its use in detecting new gene-drug interactions performs at the level we have achieved. We do not have an expectation that all predicted interactions will be real, and this method is primarily developed for generating and prioritizing possible interactions. The literature is a good source of information because the discussion of the gene-drug interactions will tend to use vocabulary that suggests which features of the gene, the drug, and the gene-drug interaction are important, and these can be detected with our statistical methodology. It is possible that with the addition of other data sources (for example, the patent literature or the full text journal articles from the pharmacogenetic literature), our performance will improve. Nonetheless, we have demonstrated relatively powerful analogical reasoning capabilities even with limited text sources.

CONCLUSION

The problem of determining genes and drugs that interact has been gaining a lot of interest, as scientists recognize that genetic polymorphisms in the targets of drug therapy can have great influence on the efficacy and toxicity of drug treatments. We address this problem by developing an algorithm that induces novel gene-drug interactions from the literature, based on the idea that gene-drug pairs that are similar to gene-drug pairs that are known to interact are also likely to interact. This algorithm has been shown empirically to induce correct and nontrivial gene-drug interactions.

ACKNOWLEDGMENTS

This research was supported in part by the Research Collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University research project on Concept Bases for Lexical Acquisition and Intelligently Reasoning with Meaning.

RBA is supported by NIH GM61374 and NIH GM64782.

REFERENCES

Chang, J., Raychaudhuri, S. & Altman, R. (2001). Using biological literature improves homology search. In *Pacific Symposium on Biocomputing 2001*. Mauna Lani, Hawaii, pp. 374–383.

Fodor, S., Read, L., Pirrung, M., Styyer, L., Lu, A. & Solas, D. (1991). Light-directed spatially addressable chemical synthesis. *Science*, **251**, 767–773.

Hewett, M., Oliver, D., Rubin, D., Easton, K., Stuart, J., Altman, R. & Klein, T. (2002). PharmGKB: the pharmacogenetics knowledge base. *Nucleic Acids Research*, **30**, 163–165.

Klein, T., Chang, J., Cho, M., Easton, K., Fergerson, R., Hewett, M., Lin, Z., Liu, Y., Liu, S., Oliver, D., Rubin, D., Shafa, F., Stuart, J. & Altman, R. (2001). Integrating genotype and phenotype information: an overview of the pharmGKB project. *The Pharmacogenomics Journal*, **1**, 167–170.

Lander, E., Linton, L., Birren, B., Nusbaum, C., Zody, M., Baldwin, J., Devon, K., Dewar, K., Baldwin, J., Dewar, K., Doyle, M. & FitzHugh, W. (2001). Initial sequencing and analysis of the human genome. *Nature*, **409**, 860–921.

Manning, C. & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA.

Micromedex (2002). USP DI. Drug Reference Guides. volume I. Drug Information for the Health Care Professional.

Mount, D. (2001). Bioinformatics: Sequence and Genome Analysis. Cold Springs Harbor Press.

Russell, S. & Norvig, P. (1995). Artificial Intelligence: A Modern Approach. Prentice Hall.

Yang, Y. & Pedersen, J. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*.