Computational Development of a Comprehensive Database of Drug-Drug Interactions – Amy Tai

PERSONAL SECTION

Computational biology has been "my" field for quite a while, albeit a nebulous term, but I honestly cannot pinpoint concrete reasons behind this long-term interest. Upon extensive reflection, I have developed a few milestones that cumulatively explain my fascination, but these few are only part of the complex system of cause and effect that have influenced my academic career.

It began freshman year, while I frustratingly browsed Google for hours after missing a question on a biology exam. What were exons and introns? I had never heard of them in class, and the textbook relinquished no information either. Surprisingly enough, that one night of rudimentary research sealed my fate for the next four years. I became interested in introns and exons and the still-open problem of differentiating between the two in an arbitrary segment of human DNA. Developing a computational method of binary classification was my first research project in bioinformatics. I worked on it for a good 18 months, after which I decided I needed to learn more theory; the method I developed used painfully simple arithmetic, and I was stuck unless I intended to derive an intelligent system from scratch.

In fact, for a time, I really was stuck. For the entire summer between sophomore and junior year, my research project rusted in a corner, because I had not yet discovered the true meaning of "computational" in "computational biology." To me, "computational" was still the four major operations: addition, subtraction, multiplication, and division; I was "computing" numbers, much like a cheap, gas-station calculator. Little did I know, however, that there is an

entire field of artificial intelligence focusing on classifiers, statistical learning methods, and intelligent systems. If you think about it, this is exactly the kind of tool that computational biologists need: some kind of artificial mind that will make decisions for you based on a certain input. You want to build some sort of object—computer program, robot, etc. (though robot is slightly fancier for our goals; nevertheless your program can be embedded in an interactive robot, if you truly wish)—that will make decisions for you on a large scale based on only numerical data that you give it. There are several advantages to this approach (and yes, now I will proceed to iterate through a list of the pros of computational methods). First, the quantitative nature eliminates the immediate need for laboratory methods. Of course, if we delve deeper into the concept of quantitative approaches, we discover a paradox, because it turns out that we need *experimental* data to train and develop accurate computational tools. Regardless, the ultimate goal of artificial systems is to make quicker decisions that can shepherd research in a particular direction, until the much slower laboratory results prove us wrong (or right—research is always a gamble in the unknown, of course). Second, we can tackle a large body of data in a very short amount of time. Imagine trying to annotate all 3 billion base pairs of the human genome by hand (meaning, in the laboratory). Scientists have been trying to do that even before the genome was decoded in 2001, yet we still know very little about the human genome. Under ideal conditions, a developed model can help scientists identify potential segments of human DNA with desired characteristics, after which scientists will have the locus of a specific DNA segment to study.

With that basic introduction to bioinformatics out of the way, let me tell you my personal experience with the field. As I mentioned previously, I was stuck for a summer because I had not yet discovered the wonders of applied mathematics. Then, on another fateful day, I stumbled

across a Bayesian network article on PLoS Bioinformatics. For some time, actually, a model had been developing in my head; with this model, I imagined an input layer, a magical hidden layer, and an output layer where the end result was reproduced. I did not know that this wondrous device was called a neural network, so when I read about Bayesian networks, I mistakenly thought that Bayesian networks were my culprit. As I read some more about them, though, I realized that what I wanted were neural networks, a handy mathematical tool that takes input and spits out an output for the researcher to decipher (more on neural nets in the paper). Things happened quickly from here. I got a hold of a statistical software package that can train neural networks and jumped right into the same project, now armed with neural nets.

But that was just the beginning that prepared me for my current research project, which started during the summer after junior year. I went to a summer program, the Research Science Institute, which is funded through the Center for Excellence in Education and held at mentorships sites around Boston for six weeks each summer. My mentors were conveniently computational biologists and involved in the computing departments of both MIT and Harvard Medical School. They had some graduate students working on a drug database, but because nothing really came out of the investigation, they handed the responsibility over to me. In short, they gave me the database and told me to get to work. I did.

With my unusual interest in predictive biology (predictions via computation, of course), the first thing I could think of was predicting unknown drug-drug interactions (DDIs). This database contains more than 4,000 drugs (see paper) and only a fraction of the possible interactions between the drugs are given. With the help of my tutor at RSI, I really got to get down and dirty with various concepts in statistics. Math became tangible, because I had to use probabilities to model the behavior of drug-drug interactions. Statistics was the key to

developing these probabilities, because it helped me find a pattern in known drug-drug interactions, in order to create a model of prediction for new DDIs. I fiddled around with various probability distributions using the statistical software and trying to integrate them into my Java programs. In fact, a large part of my research project was spent programming, because I had to find a way to transfer my raw data into numbers that I can put into the neural network. Programming is an excellent exercise in logic and thinking. In order to represent what I wanted to do in lines of code, I had to meticulously write out every step and calculation that I wanted the program to perform. This organization helped keep me on track and gave me a clearer picture of what I wanted to do with the database.

Although I was not done with my research at the end of the summer, I continued working on it throughout senior year and eventually submitted it as an Intel Science Talent Search project in November. To my great enthusiasm, I found out that my project made the semifinalist cut! When I did not make finalist standing, I was not too crestfallen, because the progress I had made with my research project was the true prize. I had figured out a way to continue my research, even beyond my submission to the STS. My research did not end with STS; in fact, as cheesy as it sounds, STS was really just the beginning. Now, I am still working on this project, with an even more resolved picture of what I want to do with this research. Looking back, I think one of the most exciting aspects of scientific research is that you can make it your own. All of my research projects have helped me develop a deeper understanding of bioinformatics and the individual fields that compose it—mathematics, computer science, biology, and chemistry. Moreover, because I have pursued these projects of my own volition, I have a personal and deep connection with the theory behind my research. With no preparation, I can field any question about my research, because I have worked on it for years, carefully identifying each of its errors,

successes, and possible extensions. This sort of experience seems very surreal for a high school student, but in reality, anyone can conduct research as long as they have the interest and ambition. I found problems in science and decided to pursue them. That is something that any individual—especially any curious high schooler—can do.

RESEARCH SECTION

Introduction

An understanding of drug-drug interactions (DDIs) impacts fields ranging from medicine to drug development to public health. In 2004, the average American took a combination of 12 prescription drugs per day [1]. This daily behavior seems trivial, but lack of proper DDI knowledge puts millions of individuals at risk, as a set of 12 drugs could cause more than 1000 lethal interactions. Also in 2004, more than 1% of all deaths were directly caused by DDIs, because these patients were oblivious to the life-threatening reactions that their drug repertoire would cause in the human body [2]. By developing a comprehensive database of DDIs, we can hopefully reduce the number of deaths associated with DDIs. The database would be used by doctors when initially prescribing drugs and be incorporated in software used by pharmacists to determine which drugs to dispense. There are existing programs that help pharmacists dispense better combinations of drugs, but these are sparse and often inaccurate. One of the goals of this investigation is to improve these programs in accuracy so that pharmacies can be more effective in preventing fatal DDIs.

The objective is to construct a comprehensive database of drug-drug interactions using a computational approach. There are too many DDIs to rely simply on laboratory experimentation; even if every person in the world identified one DDI per day for billions of years, there would still be more DDIs because of the limitless number of possible drug

combinations in varying environments.

There are, however, existing databases of DDIs, so it would be repetitive to create a new one from scratch. The structural goal of this project is to develop a way to make better DDI databases from an *existing* database.

Background

Let us establish the nature of the existing database. After understanding the database, we can capitalize on its structure to complete it. The Drug Advice eXpert (DAX) database includes more than 4,000 drugs and 50,000 interactions between any two drugs. Note that 4,000 drugs can potentially have more than 8 million interactions, hence this is a very sparse database. A small portion of the database is reproduced in Figure 1.

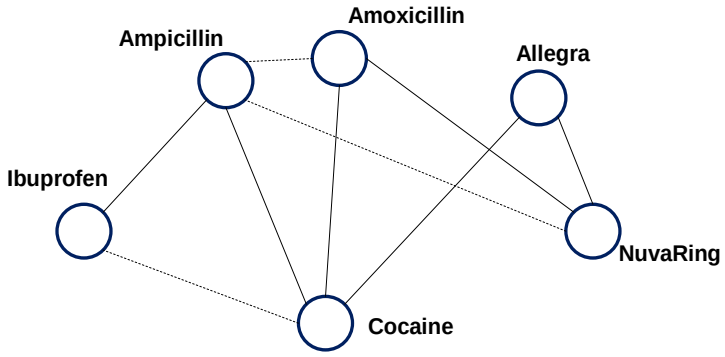

Figure 1: The solid lines indicate interactions already in the database. The dotted lines indicate examples of unknown interactions that we hope to predict with the neural network model.

Because the database was constructed through a literature search, each interaction also has a level of reliability, based on the reliability of the article and experiment from which the DDI was derived. The drugs are also organized into classes; drugs with similar chemical compositions are in the same class. Reliability and drug class will be used in the model (see Methods).

The immediate purpose of this project is to complete the DAX database by predicting all possible interactions between the more than 4,000 drugs in the database. To do this, we train our model with information from the 50,000 known interactions. We then use the model to predict the remaining interactions, which include the dotted lines in Figure 1.

Overview of Neural Networks

One of the most important aspects of this investigation is model development. We need a way to predict DDIs with sufficient accuracy. The model chosen was a neural network model. Neural nets combine many different algorithms to form efficient yet compact systems; a diagram of the model used is shown in Figure 3. During a "training" session, formulas are derived between nodes, which are the circular hubs in Figure 3. These formulas are established so that they minimize error within the training data set. Once the formulas are derived, they remain fixed for the "testing" session, where the model is tested on a new data set. The accuracy of the model is determined with respect to this test set. The specific qualities chosen for the input layer for our DDI model is explained in Methods.

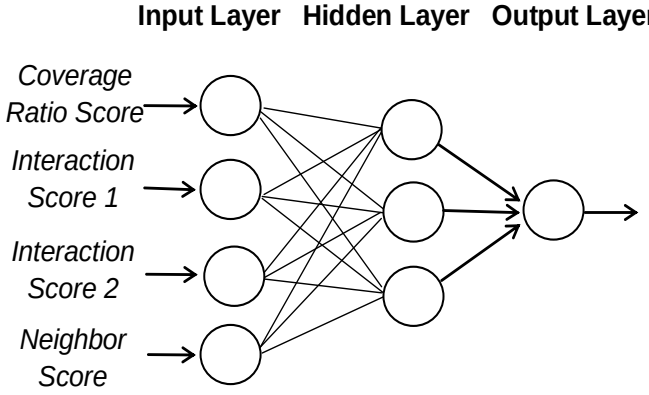

Figure 3: A schematic of the neural network model used to predict new drug-drug interactions. Formulas are inserted at the edges during training.

Methods

Each interaction is classified into one of six categories of effect. The descriptions of the categories are shown in Table 1. Along with the corresponding level of reliability (with Level 1 being the least reliable and Level 4 being the most reliable), each category-reliability combo was assigned to a number from 1-10. This final number represents the relative intensity of the interaction in question. Each interaction in the database is classified in a certain intensity class based on this table. This makes clinical data analysis easier, because only one number (the intensity) is associated with each interaction.

Category of Effect	Level of Reliability	Intensity of Effect
A (significance is unlikely)	All Levels	1
B	1, 2	2
(significance is questionable)	3, 4	3
Z	1,2	3
(same active ingredient/reconsider doses)	3,4	4
C	1	4
(plasma levels need to be monitored)	2, 3	5
	4	6
D	1	7
(severe clinical consequences)	2	8
	3	9
	4	10

Table 1: Conversion from effect and reliability to intensity.

The Input Layer

Four "scores" were chosen as qualities for the input layer. These four should help us predict the intensity of an interaction; for ease of discussion, "positive" interactions have intensities from 1-3 and "negative" interactions have intensities from 4-10. The scores are described as follows.

Neighbor Score

The neighbor score works by the principal of mutual friendship. If Alice and Bob and friends, and Bob and Carol are friends, then Alice and Carol have a good chance of being friends. Consider the possible interaction between Ampicillin and Amoxicillin in Figure 1. Both react with cocaine, theoretically increasing the probability that Ampicillin and Amoxcillin interact. The neighbor score also considers the reliability of interactions. For example, maybe we heard from an unreliable source that Alice and Bob are good friends. This would lessen the probability of a friendship between Alice and Carol. Similarly, if the logged interaction between Amoxicillin and cocaine was from an unreliable experiment, the probability of interaction between Ampicillin and Amoxicillin decreases.

Interaction Score

The interaction score calculated the fraction of all of a drug's known interactions that have an effect. For example, in Figure 1, if we consider the potential reaction between Ibuprofen and cocaine, Ibuprofen's interaction score would be 0.25, because out of its four possible interactions, only the one with Ampicillin is positive. Cocaine would yield an interaction score of 0.75, because three out of its four possible interactions are positive. Theoretically, the interaction score should reflect the reactivity of a drug, because more reactive drugs would have higher interaction scores. This score covers the next two nodes of the input layer in Figure 3.

Coverage Ratio Score

The coverage ratio capitalizes on the drug class organization of the database. Say we are considering the interaction between drugs 1 and 4 in Figure 4. Then the coverage ratio looks to the drug classes of 1 and 4, which are Class 1 and Class 2, respectively. There are 8 possible interactions between the two classes, and only five exist (the solid lines). Hence, the coverage

ratio is 5/8 = 0.625. Theoretically, the coverage ratio reflects the reactivity of two general drug types; if the relatives of drugs 1 and 4 react appreciably, then drugs 1 and 4 have a greater chance of interacting as well.

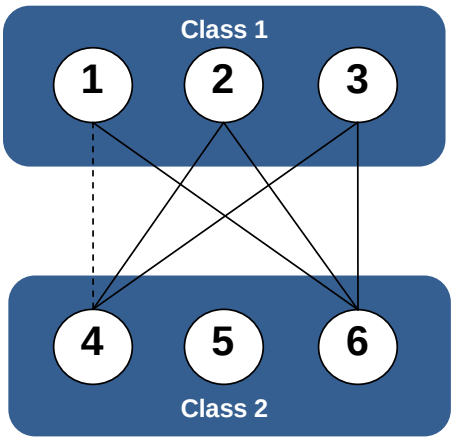

Figure 4: An example of drugs organized into classes.

Training and Testing Data Sets

The DAX database was stored in the form of an adjacency matrix. If we do not separate the training and test sets, then the model accuracy will be skewed. For example, say the neural net was training with interactions from drug A. If we test for interactions of drug A in the testing session, the model will already know some of the behavior of A, invalidating our results. Hence, each drug was only available in one set. We segregated the adjacency matrix by Figure 6, simulating two separate, mini-databases. There was approximately the same number of interactions available in each potential data set. Note that the potential training and testing sets are mutually exclusive, preserving the integrity of our model. Approximately 2,000 interactions were randomly chosen for training from the potential training set, and 2,600 were randomly chosen for testing from the potential test set.

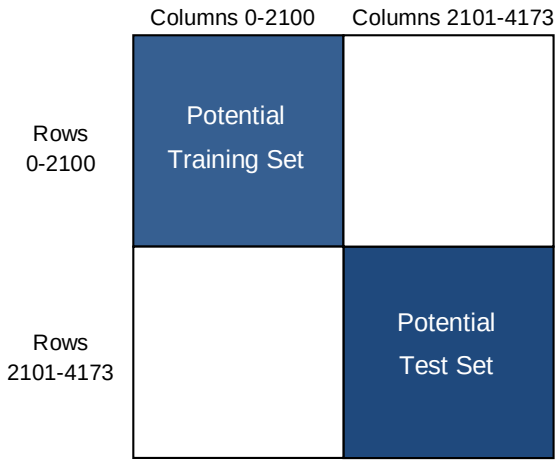

Figure 5: Segregation of the database into mutually exclusive training and test sets.

Results and Analysis

We trained the neural network on the 2,000 training interactions. Then, we tested the network with the 2,600 test interactions. To ease decision-making, we said that any intensity less than 4 was a negative interaction and any intensity greater than 3 was a positive interaction. This cutoff makes sense from Table 1. With this cutoff, our "general" accuracy was **94.2 %**, with a false positive rate of **3.4%**. This type of decision-making is very general, because it only takes on a true/false value. *K*-accuracies provide a closer prediction.

Let us call *k*-accuracy the chance that a prediction will be within *k* intensity levels of the actual intensity. For example, say an interaction has intensity 8. Then any prediction from 7 to 9 is considered a correct prediction. The 1-accuracy was **64.8 %** and the 2-accuracy was **70.6 %**. *K*-accuracies are better for considering the definite, rather than relative, intensity of an interaction. However, for general predictions, we can safely say that the model can predict the existence of an interaction, based on the general accuracy above.

We also wanted to determine which input quality was most important. We did this by retraining networks with only three inputs. The accuracy of the resulting networks should reveal

the relative effect of each input on the output as predicted by our neural net. The accuracies are listed in Table 2.

Score Left Out	General Accuracy	1-Accuracy	2-Accuracy
None	94.2 %	64.8 %	65.7 %
Coverage Ratio Score	89.6 %	52.0 %	78.3 %
Interaction Score of 1st Node	68.0 %	63.2 %	72.3 %
Interaction Score of 2nd Node	93.2 %	53.7 %	70.6 %
Neighbor Score	84.8 %	59.6 %	66.3 %

Table 2: A summary of the accuracies of all the networks.

To analyze Table 2, it is more helpful to look at the general accuracies separately from the *k*-accuracies. In terms of general accuracy, the network without the 1st Interaction score performed most poorly, meaning this score has a high effect on the general predicting ability of the network. However, for 1-accuracy, which measures the accuracy of the network at greatest resolution, the network without the coverage ratio performed most poorly. Hence, the coverage ratio is better at determining the numerical intensity value for an interaction. The neighbor score also has a fairly strong effect, because the network without the neighbor score performed most poorly in terms of 2-Accuracy. In general, because each score contributes to a different measure of accuracy, we conclude that none of the scores has a more noticeable effect on prediction than the other scores.

Conclusion and Further Research

Our final model had an accuracy of 94.2 % at best, which means that the model we developed is a fairly good predictor of new DDIs. It shows promise for further modification so that we can complete the DDI database.

The input layer is the most important aspect of the model. Without proper predictive qualities, the model would be ineffective, analogous to trying to predict someone's weight from hair color, eye color, IQ, and favorite book. Generally, taking out a score increases the 2-accuracy of the model. This means, with less excess noise, the network is capable of making precise predictions. Conversely, when all four scores are used, the network has a better general accuracy, because it has a better idea of the "big picture", but has less resolution, i.e.- it has poorer k-accuracies. One possibility for increasing the resolution of our model is cleaning up the database. One of the reliability levels indicates that an interaction "is suggested by inconclusive case-reports, in-vitro studies or studies on related substances." Such a reliability is close to saying that an interaction is still unknown. Maybe the interactions that have been verified in this way should be removed and new, more reliable reports explored.

We could also try different models. Bayesian networks and support vector machines may be better models for our data, and comparing their predictive results with our neural network can reveal more about the distribution of DDIs. Even more conclusive validation is experimental evidence; the next step after a computational approach is a wet-lab approach, which is what our computational method replaces. Although wet-labs are time-consuming, their confirmation of the computational predictions can create a more robust model.

We can imagine many applications of this database down the line. For example, we can create software that tells pharmacists and doctors the best combination of drugs for certain symptoms. Say a patient needs to be treated with depression, insomnia, and allergies. Then the software will return, from a group of 500+ possible drug combinations, the optimal three drugs that will minimize interaction within the human body. The effectiveness of the software depends on the expanse and reliability of the database, which presently includes drug families ranging

from antibiotics to topical creams to hormone regulators. Further, once we perfect the model developed in this investigation, we can extend it to accepting parameters, such as pH or temperature in which two drugs interact. By incorporating more information about the environment of reaction, we can be more specific about when DDIs occur. All these applications stem from the basic notion behind this investigation: being able to predict when drug-drug interactions will occur.