

A HYBRID INTELLIGENT SYSTEM FOR FORMULATION OF BCS CLASS II DRUGS IN HARD GELATIN CAPSULES

Gunjan Kalra, Yun Peng
Laboratory of Advanced Information Technology
Dept. of Computer Science And Electrical Engineering
University of Maryland Baltimore County
{gkalra1, ypeng}@cs.umbc.edu

Mintong Guo¹, Larry L. Augsburger²
School of Pharmacy
University of Maryland at Baltimore
1. mguo001@umaryland.edu
2. laugsbur@rx.umaryland.edu

ABSTRACT

In this paper, we describe a hybrid intelligent system for formulation of BCS Class II drugs in hard gelatin capsules. Several significant challenges are involved in drug-formulation: the active ingredients and the fillers in the capsule must be chemically compatible according to biopharmaceutical principles; the formulation must be manufacturable; and it must meet the prescribed drug delivery requirement. Traditional trial and error approach to drug-formulation design is too costly and time consuming to meet the increasing demand for new drugs. To answer these challenges, we have developed a prototype hybrid intelligent system for automatic drug formulation. This system consists of a rule-based Expert System (ES) that conducts formulation design according to biopharmaceutical principles, and a neural network (NN) that predicts the quality of the formulation recommended by ES. Through interaction between the two modules, the hybrid system forms a (re) formulation-prediction cycle, and the quality of the formulation is improved with each iteration. The hybrid system is tested with sample drugs and is shown to be able to produce formulations with desirable performance measures.

1. INTRODUCTION

Most of the traditional expert systems have been based on symbolic logic-based knowledge representations such as rules. Such systems are good at capturing well-structured domain knowledge and experts' decision process but often perform poorly when used to model complex processes or interrelations. Neural networks (NN), on the other hand, are strong in discerning complex relationships between patterns through learning even when such relationships are ill structured or poorly understood. It has been widely accepted that a hybrid system consisting of both ES and NN may work much better than using each alone [3,6]. Many such hybrid systems have been reported in the literature, in which ES and NN are collaborated in different ways to achieve a common goal. One example of such hybrid systems is an auto-insurance claim processing system where the ES determines the responsibility and com-

pensation rates and the NN handles the uncertainty involved and tunes the compensation rates [9]. Other approaches include using neural networks to learn the rules for expert systems [7], and using expert systems to predict the weights for neural networks [8].

The pharmaceutical industry has been using expert systems and neural networks as individual stand-alone systems for automating their basic operations. One of the basic manufacturing processes in pharmacy is the development of drug formulation designs. Because of its varied physical and chemical properties, the actual therapeutic or medicinal substance (called active ingredients) may or may not be directly administrable to patients. Drug formulation is the process of converting this drug substance into medicines that can be administered to the patients. The final form of the medicine, either capsules or tablets must meet the prescribed drug delivery requirements and should also conform to the physical limits of the manufacturing process.

Despite the perception of capsules being simpler in terms of accommodating large doses of drug in capsule shell (direct filling of powders, granules and pump-able liquids into the shell), the design of drug formulations for hard gelatin capsules can present significant challenges [4]. Some of these challenges are:

- Selecting fillers (wetting agents, lubricants, disintegrants, etc.) needed to fill the capsule and maintaining compatibility amongst these ingredients;
- Measuring accurate quantities of ingredients;
- Maintaining uniformity (so that each capsule contains the prescribed amount of the active ingredient) in different manufacturing environments with varying dosage principles and operating characteristics;
- Meeting specific drug delivery performance design criteria (such as dissolution rate in human body).

Pfizer's Capsugel division, a leader in capsule manufacture and formulation has made pioneering efforts towards developing intelligent systems for automatic drug formulation in hard gelatin capsules. One such system, CAPsugel EXpert, is a decision tree-based ES that is able to make good formulation recommendations for certain class of drugs based on formulation knowledge collected

from experts in a network of over a dozen of international leading pharmaceutical companies [1,5].

Ostensibly, the formulations recommended by CAPEX satisfy ingredient compatible constraints. However, due to the complex relationship between the formulation parameters (i.e., types and quantities of excipients, the size of capsules, etc.) and the performance measures (e.g., content uniformity and dissolution rate), the quality and feasibility of this formulation is not certain unless capsules with this formulation are actually manufactured and physically tested. If the lab test results are not satisfactory, the drug needs to be re-formulated. In other words, systems like CAPEX only automate half of the formulation process and leave the important half, the performance validation, to lab experiments which are very expensive and time consuming.

To answer these challenges, we have developed a prototype hybrid system that integrates a rule-based expert system and a backpropagation neural network for automatic formulation of Biopharmaceutical Classification System (BCS) class II drugs. BCS class II drugs are the drugs with high solubility and low dose-volume [4]. By utilizing the strengths of both paradigms, this system not only can effectively generate drug formulations that meet the biopharmaceutical constraints, but also has the potential to provide a facilitated way of validating the performance of the recommended formulation through analysis of previously developed formulations.

The rest of the paper explains various aspects of the hybrid expert. Section 2 describes the design and implementation of the system. Section 3 presents the test results and observations. Section 4 concludes with directions of future research.

2. SYSTEM DESIGN AND IMPLEMENTATION

Our hybrid expert system is designed to generate and validate drug-formulation designs for BCS class II. Piroxicam, a class II drug is used as a sample drug in this paper. For a recommended formulation, the current prototype only validates the *dissolution rate*, which is one of the most crucial drug delivery requirements for class II drugs. As depicted in Figure 1, the system consists of four modules:

- Formulation module (FM): a rule-base expert system which makes formulation recommendations based on logical rules that encodes formulation experts' knowledge and experience;
- Prediction module (PM): a backpropagation neural network which predicts dissolution rate of the recommended formulation using the mapping between formulation parameters and the dissolution rates learned from samples of lab test data;

- Parameter adjustment module (PAM): it allows human formulators to adjust formulation parameters when the predicted performance is not acceptable;
- Control module (CM): it integrates the other three modules via several information exchange channels and GUI windows.

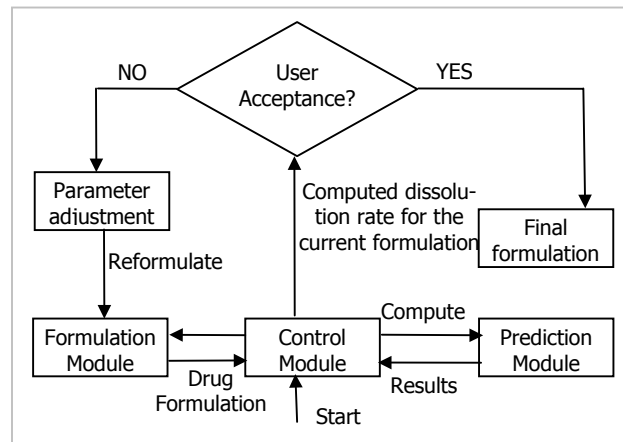


Figure 1. The Hybrid system architecture

In this system, the formulation process is a cycle of (re) formulation-prediction. It starts with input from the user of the basic information concerning the properties of the active ingredient (i.e., its dosage, particle size, desired dissolution rate) and excipients (e.g. wetting agents, lubricants, disintegrants). The formulation module then makes a formulation recommendation based on the rules that capture the biopharmaceutical principles and the physical and chemical properties of these substances from the expert formulators. The recommended formulation is sent to the prediction module for computing the predicted dissolution rate. The predicted dissolution rate is subjected to the user's inspection. If the user is not satisfied, he will be allowed to change the values of some parameters in the recommended formulation. The changed values are fed to the formulation module for re-formulation, and the cycle repeats. Re-formulation is necessary before going to NN again because the constraints between all substances (ingredient and excipients) that were previously satisfied may be invalidated by the selected new parameters.

The details of each module are given in the subsequent subsections.

2.1 Formulation Module

The formulation knowledge encoded in this module is given in the form of a decision tree from domain experts. As depicted in Figure 2 below, the ES consists of several decision modules:

- Content Uniformity (CU)
- Ordered Mixing (OM)
- Direct Fill (DF)

The ES also consists of conditions that are required for entering these modules. Based on the user input of basic information concerning the drug, the module first verifies the drug being formulated to be a class II drug. If the drug does not belong to class II, the system suggests the granulation process, which is beyond the scope of the prototype system, and the system exits.

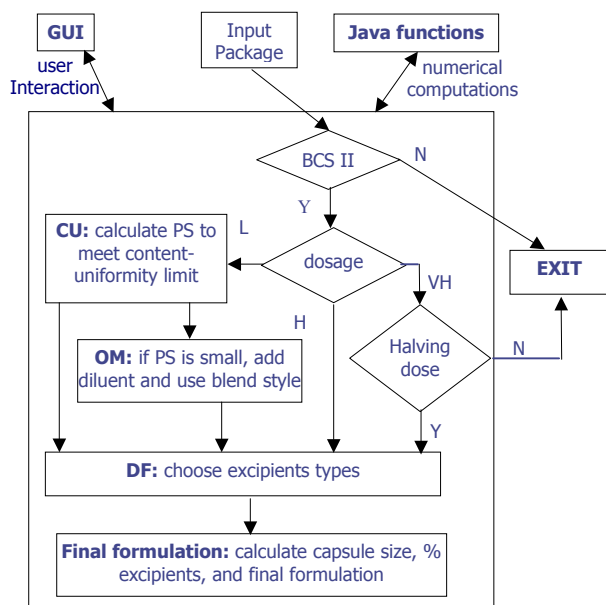


Figure 2. Decision tree for formulation

The decision tree may be a natural way to capture domain experts' formulation decision process. However, as a knowledge representation paradigm to support intelligent systems, this approach has its shortcomings. Among other things, a decision tree tightly couples the knowledge and the use of knowledge in individual decision paths; as a result all possible decision paths have to be explicitly enumerated in the tree. For this and other reasons, we have chosen the rule-based paradigm for the formulation module and implemented it in Prolog language.

It is straightforward to translate the decision tree into Prolog rules as each rule roughly corresponds to a decision (together with its conditions). The advantage of using logic rules over decision trees can be best seen in re-formulation process where user selects new parameter values to alter the dissolution rates of the current formulation. To handle this new information in a decision tree, the decision process may need to be started all over again along with the previously drawn conclusions discarded and user been repeatedly asked redundant questions. Alternatively, the tree may be extended to include additional decision nodes and links, resulting in a much larger tree. Whereas in rule-based systems the new information can be incorporated by asserting new facts in the Prolog knowledge base and retracting or removing the facts asserted for the old parameter values. The new assertions

will trigger the executions of only those rules that affect or utilize these parameters.

To get the system started the user specifies physical and chemical properties of the drug in an input package. The input package consists of the name of the drug, drug-dose, particle size of the drug, desired dissolution rate, specific surface area, content uniformity tolerance limits, permeability of the drug, bulk and tapped densities of the drug. The rule for CU module utilizes the java function responsible for computing a new particle size that conforms to the Content Uniformity limits. The user is prompted for the acceptance of this particle size. For drugs with small particle sizes, the control is passed on to the OM module. In order to increase the particle size, the OM module helps the user in selecting a blend-style to blend a proper excipient with the drug. The DF module prompts the user to enter various chemical and physical properties of the excipients and uses them to compute the exact percentages of the excipients to be added to the drug mixture. In the final stages of the formulation an appropriate capsule size is recommended to the user. User can choose to select a bigger capsule size as per the needs. Depending upon the total capsule volume and densities and percent volume of excipients, quantities (in milligram) of the excipient are computed. Final formulation is a tabulation of types and amounts of the excipient along with the intermediate results like Carr Index, capsule size etc. The formulation results are also written to the formulation file for later references.

2.2 Prediction Module

The prediction module is a backpropagation (BP) neural network with hidden nodes. There are a number of neural network models that can learn mappings between input and output patterns from the training data, including the competitive learning networks, adaptive resonance theory networks, probabilistic networks, and counter-propagation networks, etc. We choose the backpropagation network for the following reasons:

- It is the most powerful learning model in terms of the learning accuracy (it is known to be able to represent any L_2 function to any given precision) while many others can only learn mappings that are linearly separable.
- It does not require prior knowledge of the relationships between input and output variables as long as the training data contains all of the important factors that influence the value of the output variables.
- It is flexible in terms of accommodating mixed data types (numeric, ordinal, range, etc.)
- The learning algorithm has been well developed and it easy to use [2].

The neural network has two modes of operation, training and computation. The training component that

exhibits the BP learning is implemented in C whereas computation component that computes the dissolution rates for a given input pattern is implemented in Java so that it can be invoked by the system via an interactive GUI. Each input pattern consists of nine independent variables charactering a drug formulation. These are the surface area of the active ingredient, the filler particle size, and the types and percentages of various excipients. The output pattern includes the dissolution rate of the given formulation at three different time points (10, 30, and 45 minutes).

The quantity and quality of the training data affects the prediction accuracy of the neural network dramatically. Considering the high cost of experimental batches, the best way to deliver as much information as possible in a given series of experimental batches is to use statistical experimental design to develop and generate the training data. Three statistical experimental designs are applied, which are Central Composite Design (face-centered), L18c-Chakravarty and Box-Behnken design. A total of 63 samples were generated and used in NN training.

Various training parameters affect the training, including the learning rate, slope of the sigmoid function of the hidden and output nodes, number of epochs/iterations, learning mode (batch/sequential) and number of hidden layers and hidden nodes at each layer. Several experiments were conducted by varying the above parameters to select a best. The final combination chosen for training were: the network contains one hidden layer of 9 nodes, the training is done in sequential mode with learning rate of 0.2 and slope of 0.2; it stops at 2000 epochs.

To ensure that the trained mapping can be generalized to patterns unseen before, over-fitting should be avoided. To this end, we employed the commonly used cross-validation in the training. 8-10% of available samples are randomly selected and set aside as the cross validation (CV) set. The CV set thus does not participate in training; instead it serves as a set of test samples with which to verify the neural network's predictive power. Table 1 below gives a set of cross-validation results with five (~8%) CV samples. The predicted dissolution rates were very close to the target values in the cross-validation set.

In the table the data under the D** and predicted columns are the dissolution rates in 10, 30 and 45 minutes as predicted by the neural network and those under the columns D** and target are of dissolution rates of manufactured capsules generated by lab experiments. The results show that the predicted rates are very close to those from the actual lab tests (with the largest error of 6.8%). Considering the normal variance in real dissolution data and the size of the potential search space (~200,000), we can say that the trained NN with only 58 training samples demonstrated a remarkable performance in predicting the dissolution rate for a given drug formulation.

Table 1. Cross-Validation Results

D10	D10	Error	D30	D30	Error	D45	D45	Error
target	(pre-dicted)		target	(pre-dicted)		target	(pre-dicted)	
57.39	60.87	3.5	76.66	78.38	1.7	82.84	84.49	1.6
71.70	67.64	4.1	94.29	89.00	5.3	98.51	93.78	4.7
68.60	62.84	5.8	90.04	83.23	6.8	94.08	89.43	4.6
67.18	66.89	0.3	89.58	88.89	0.7	93.89	93.88	0.0
63.89	77.32	13.4	88.61	92.19	3.6	93.96	95.10	1.1

2.3 Parameter Adjustment Module

This module is a Java GUI interface that enables the user to change the formulation parameters in order to increase or decrease dissolution rates. The six parameters that can affect the dissolution rate have been identified; they are the lubricant amount, lactose percent in MCC/Lactose ratio, filler particle size, disintegrant amount, wetting-agent amount and SSF/MS ratio. The current implementation only allows the user to change one of the six parameters at a time. This can be easily extended to allow the user to change multiple parameters simultaneously. The new value for the selected parameter is then fed to the formulation module to trigger a re-formulation.

2.4 Control Module

The control module, a Java GUI application, controls the cycle of the formulation-prediction process. Control module provides an interface for the user to specify the input package to the system; it also renders various GUI that allows the user to interact with the system in terms of specifying the various physical and chemical properties of the drug and the excipients. Amongst the GUI classes, Prologfunctions class is the main file that starts the system and calls the ipPackage class that provides the user with the input package form. Gendialog class implements the functions that interface with the prolog predicates. Formulation and drate classes render the GUI for formulation and dissolution rates results respectively. Ann class calls the neural network to compute the predicted dissolution rate for the recommended formulation. Control module also provides the information flow between the formulation and prediction module. This is done by a java-prolog interface with the help of a prolog file that contains definitions of the Java functions and the corresponding prolog predicates. In Windows environment, this file is compiled into a dynamically linked library file.

3. TEST RESULTS AND OBSERVATIONS

The ultimate goal of the hybrid intelligent system is to help drug development scientists in designing formulations that meet certain design criteria. The cross-validation test only tests the predictive power of the NN module. Additional tests were conducted that went through the

entire hybrid system to see if it can generate drug formulations conforming to the design criteria and to have the desired dissolution rates. In these tests, three different formulations of piroxicam were generated by the system with different user inputs, and their predicted dissolution rates computed by the NN module. Three batches of capsules were manufactured and their dissolution rates measured at the lab. The comparison is given in Table 2.

In the table, each formulation row consists of two sets of dissolution rates, one predicted by the neural network and the other obtained from the laboratory experiments. The table also recorded different amounts of some the excipients that produced the three different formulations. The comparison reveals that, except one entry for formulation 3 at 30 minutes, the dissolution rates from the formulations produced by the hybrid system were very close to the lab test. The encouraging results demonstrate that the hybrid system, through its formulation-prediction cycles, is able to derive the recommended formulation to match the design criteria with good accuracy.

4. CONCLUSION

A prototype intelligent hybrid system is developed for automatic drug formulations in hard gelatin capsules. This system has the potential of supporting functions beyond the scope systems such as CAPEX that rely solely on a symbolic logic based paradigm. The formulation development is an iterative process in which a formulation can be incrementally modified to improve its manufacturability and its performance as drug delivery system. This may lead to a significant reduction of the number of lab tests needed to evaluate recommended formulation, and thus further cut the development cost and time.

The quality of the output can be further improved by taking advantage of the learning capability of the system which would allow the decision process be modified based on its past performance. Such a hybrid system framework also has the potential to be extended to incorporate other intelligent modules, such as optimization or the development of other dosage forms. Although the sys-

tem is tested only with a single drug, its and underlying principle can be generalized to formulations of other BCS class II drugs by exploring some other biopharmaceutical principles involved in drug classification and formulation.

5. ACKNOWLEDGEMENT

This work is supported in part by a gift from Pfizer's Capsugel Division.

6. REFERENCES

- [1] Capsugel Library, "Expert System for Formulation Support", Capsugel Inc., Greenwood, SC, 1996.
- [2] Hyakin, S., "Neural Networks: A Comprehensive Foundation", New York, NY: Macmillan, 1994.
- [3] Gallant, S.I., "Neural Network Learning and Expert Systems", 1993, *MIT Press*, Cambridge, MA.
- [4] Guo, M., "Development of a Prototype Hybrid System for Capsugel Formulation Support, Including a Consideration of New Excipients", PhD dissertation, University of Maryland School of Pharmacy, April 2002.
- [5] Lai, S., et al, "An expert system to aid the development of Capsule formulations", *Pharm. Tech. Eur.*, **8**, 60-68, 1996.
- [6] McGarry, K., Wertmer, S., and MacIntyre, J., "Hybrid neural systems: from simple coupling to fully integrated neural networks", *Neural Computing Surveys*, **2**, 62-93, 1999.
- [7] Sima J., "Neural expert systems", *Neural Networks*, **8**, (2), 261-271, 1995.
- [8] Towell, G. and Shavlik, J.W., "Knowledge-based artificial neural networks", *Artificial Intelligence*, **70**, 119--165, 1994.
- [9] Yoo, J., and Kang, H., "A hybrid approach to auto-insurance claim processing system", submitted manuscript, 2002.

Table 2: Predicted v/s Lab dissolution rates

filler size: 100 %lactose: 30 %lubricant: 4		Pre- dicted	lab- result	% Error	%wetting agent	% diluent
Formulation 1	d10	72.27	75.41	4.16	0.1	0.9
	d30	90.64	94.97	4.56		
	d45	95.99	98.73	2.78		
Formulation 2	d10	67.88	66.55	2.00	0.3	0.1
	d30	87.00	92.23	5.67		
	d45	96.60	96.69	0.09		
Formulation 3	d10	67.55	71.01	4.87	0.5	0.1
	d30	86.54	94.43	8.36		
	d45	97.11	99.07	1.98		