Design and Analysis of Large Chemical Databases for Drug Discovery

by

Lap-Hing Raymond Lam

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2001

0-612-65249-1

Borrower's Page

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

The drug discovery paradigm has changed in two important ways. The human genome project is giving us many more new biological targets for drug discovery. Hundreds of unknown disease genes are expected to turn up in the next few years. Combinatorial chemistry and the availability of commercial compounds have made millions of compounds available for drug screening. It is no longer possible to test all available compounds for every new target of potential biological importance.

In this thesis novel statistical methods for design and analysis of large chemical databases are described. The design problem is to choose a representative set of thousands of chemical compounds from a library that may have hundreds of thousands to millions of compounds, for assay against a biological target (screening). The analysis problem is to find regions of a high dimensional space where active compounds reside. These methods improve the efficiency and effectiveness of the drug discovery process for reducing drug screening costs and time.

KEY WORDS: Space-filling design, Exchange algorithm, High dimensional space, Multiple mechanisms, Recursive partitioning, Cell-based analysis.

Acknowledgements

Needless to say, it is extremely challenging studying for a Ph.D. part-time and working full-time, especially when I have two young children (Adrian is now 2 and Cynthia is 7). Throughout the past three years of my PhD research, I have given over 20 talks on my research work in Canada, the US and the UK, submitted two papers to statistical journals, patented two statistical methods, won the American Statistical Association 2000 Statistics in Chemistry Award, recently entered into the Chambers statistical software competition, and still survived in the corporate world (my full-time job). Without the great support from the people below, I would not be able to accomplish so much.

My family: I can never thank Alicia, my wife, enough for her greatest understanding, support and encouragement. Cynthia has done a superb job keeping me away from the TV and baby-sitting her little brother. Adrian always smiles even when he turned off my PC in the middle of an intensive simulation research.

Will Welch: I would like to thank Will, my PhD co-supervisor, for his helpful advice, his meticulous review of my work and his spending one day every second week with me discussing the research findings. I always enjoy visiting him and feel more motivated after every biweekly meeting. He is an excellent supervisor. Although my two supervisors and I live far apart, we constantly communicate ideas through the convenient network established at work (e.g., email, telephone conferencing and video conferencing).

Stan Young: I would like to thank Stan, my other PhD co-supervisor, for introducing these interesting topics, for his encouragement, his patience, his valuable comments, and for his prompt responses to my hundreds of emails. I have learned so much from him, and not just about the interesting chemistry problems. For example, Stan has introduced me to the process of filing a patent application. He also has an extensive network of experts willing to help me in my research.

GlaxoSmithKline Canada: I greatly appreciate the good support from the Biomedical Data Sciences management who allow me to spend 20% of my work hours on the research work and the good cooperation from my fellow statisticians who allow me to use their computers when available. Without their support, it would have taken me weeks to months longer to carry out the same research work.

Last but not least, I would like to thank the GlaxoSmithKline chemists (Eugene Stewart, Chris Keefer, Jim Bentley, David Drewry, Eric Bigham, Brian Hudson), the professors (Hugh Chipman, Stefan Steiner, Doug Hawkins, Nick Cercone, Aijun An) and others (Mike Lutz, Christophe Lambert, Xin Chen and Pabak Mukhopadhyay) who have provided me much help in this research area.

Table of Contents

List of Figures

List of Tables

Chapter 1

Statistics and Drug R&D Process

1.1 Introduction

The process of drug research and development (R&D) is high risk and requires a long term engagement and investment. It costs more than US$500 million and takes approximately 12 years to develop one new drug and get it approved for sale (Levy, 2000). Many R&D projects are initiated but only a small number succeed. To discover a new drug, hundreds of thousands of compounds might be initially screened and thousands of modified compounds will typically be synthesized. Even after the discovery of active compounds, there is great attrition as the compounds are tested in animals and humans. So the odds of any specific compound becoming a new drug is quite small. And of those molecules that become drugs, only one-third lead to a positive return on investment. Drug patents normally run for 20 years, leaving only a few years to recoup the R&D cost. Once a drug patent expires in a particular country, other companies are free to manufacture generic copies of the drug (see 'An overview of the drug discovery and development process', 2001). For example, sale of GlaxoSmithKline's ulcer medication Zantac, the world best-selling drug in the early 90s, dropped from over US$5 million a day to thousands a day after the drug lost its patent protection (Ghangurde, 1997 and Appleby, 1999). Thus, the very survival of drug companies depends on improving those odds and reducing the R&D time.

This thesis studies modern statistical design and analysis methods that can enhance the efficiency and effectiveness of the drug discovery process, thus reducing drug screening costs and time. In particular, this thesis focuses on the drug selection stage of the drug R&D process. Current methods as well as novel methods developed during my Ph.D. research are described. Manuscripts for the novel design and analysis methods have been submitted for publication in statistical journals. In addition, the design method has won the American Statistical Association 2000 Statistics in Chemistry Award (AMSTAT News, December 2000). In this chapter, I will describe the current drug R&D process, recent changes in drug discovery, the thesis research problems, and give an outline of the thesis.

1.2 Motivation

Although the search for new drugs requires intellectual and technological contributions from many scientific disciplines (e.g., chemists, biologists, pharmacokinetists, engineers, etc.), it remains a highly empirical process. The low success rate is due to our imperfect knowledge of biological processes, to our ever-increasing medical objectives (e.g., drugs for oral contraceptives, impotency, and smoking cessation), and to the fact that new drugs have to be better than existing drugs.

The drug discovery paradigm has changed in two important ways. The human genome project is giving us many more new biological targets for drug discovery. There are currently only about 500 drug targets ('Discovering New Drugs', 1999 and 'The Promise of Biotechnology and Genetic

Research', 2000). The human genome project is expected to increase the number of targets to about 10,000. Combinatorial chemistry and the availability of commercial compounds have made millions of compounds available for drug screening. It is no longer possible to test all available compounds for every new target of potential biological importance. Applications of modern statistical methods to enhance the efficiency and effectiveness of the drug discovery process for reducing drug screening costs and time are needed. Recent advances in computer technology have enabled development of novel statistical methods for design and analysis of large chemical databases. Design of experiments is an efficient method for choosing a representative sample of compounds for screening; statistical analysis is useful to link chemical features to biological activities of the compounds. An ideal approach is to carefully select a relatively small subset of compounds for screening and to statistically model the molecular features important for biological activity. The statistical model can then be used to guide the selection of further compounds for screening. Design and analysis can increase the success rate in identifying lead compounds, thus leading to discovery of more innovative drugs in less time and with a much smaller number of compounds tested.

The relationship between chemical descriptors and biological activity is extremely complex for high throughput drug screening data. The challenges in design and statistical modeling of data of this sort will be discussed later in this chapter.

1.3 Statistics and Drug R& D

Consider a patient who has been suffering severe stomach pains in the past three months. After several visits to his doctor and after several tests, the doctor concludes that he has a duodenal ulcer. The doctor tells him to change his diet and take a week off work. The doctor then writes a prescription for an ulcer drug. This prescription is taken to a local drugstore and a 4-week supply of tablets is purchased from the pharmacist. After a few weeks, his ulcer is healed and he is back to his normal way of life. Those little tablets represent an enormous amount of effort and expense by the pharmaceutical company that discovered the drug. The tablets were developed over many years and began when a chemist first synthesized the chemical molecule that led to the medication that healed the ulcer.

Statistics plays an important role in drug R&D. Some of the statistical application areas are (1) screening of new chemical molecules, (2) development of pharmaceutical formulations, (3) evaluation of toxicology, absorption, distribution, metabolism, and excretion of drugs, (4) design and analysis of clinical studies, and (5) quality control of manufactured drug products. Statistical design and analysis are essential tools for the pharmaceutical industry to discover and properly develop drugs that will be judged approvable by regulatory agencies.

1.4 Current Drug R&D Process

Discovering and bringing a new drug to patients can take up to 15 years of R&D and cost hundreds of million dollars. The current R&D process can be divided into target selection, drug selection, pre-clinical research, clinical trials, and review and approval.

1.4.1 Target Selection

This stage involves choosing a disease to treat, understanding the biochemical pathways of the disease, and developing a model for the disease. This gives a biological target, usually a protein critical for a biological function. This stage takes approximately 1.5 years.

Discovery of new drugs can be full of surprises. Here is an interesting example illustrating how a drug aimed at a particular molecular target fortuitously improved a human health problem. In the early 1990s, researchers at Pfizer were searching for a drug that would reduce the chest pain due to angina. Since an increase in blood and oxygen supply should result in less angina, they looked for a drug that would dilate coronary arteries. They found a substance in the blood, cGMP, that caused arteries to dilate and identified an enzyme, PDE5, that broke it down. With PDE5 around, cGMP was destroyed and arteries shrank thin and became blood-poor; without PDE5 the arteries opened up. They discovered a molecule that PDE5 could securely lock onto, thus stopping binding with cGMP. It took approximately 4 years to find a molecule that allowed PDE5 to lock on but that the enzyme could not destroy. However, while conducting clinical tests on the drug effective against the chest pain of angina, doctors often reported pills missing. The drug is now known as Viagra and is prescribed for impotence (Discovering New Drugs, 1999).

1.4.2 Drug Selection

Thousands and thousands of compounds are screened in the hope of finding different classes of drugs that work within the model system. Once a biological target has been identified, the next step in the drug discovery process is to find new lead compounds (e.g., compounds that bind to the protein). New leads are chemical compounds that produce biological activities thought to represent therapeutic potential. The purpose of biological screening is to identify those compounds that possess the desired biological activity (e.g., good receptor binding). Those compounds identified are called 'hits' or 'active' compounds. The initial hits are unlikely to be the final drugs. Complex evaluations are necessary, and typically the initial hit is modified atom-by-atom to improve important characteristics of the molecule.

When a lead compound has been identified, its molecular structure is varied to increase the level of desirable biological activity, reduce the level of undesirable activity, or otherwise improve its pharmacologic profile. Lead optimization is the process of finding a compound that has some advantage over a related lead. This process can result in a better understanding of the physical-chemical determinants of the newly discovered activity, the reduction of undesirable side effects, experimental verification of the positional requirements of drug-receptor binding, modification of an absorption or metabolic rate, or an increase in the binding coefficient. Underlying the process is the common statistical purpose of characterizing and optimizing a response function. Suitable formulations of optimized compounds are then developed and tested in clinical trials. Because there is no guarantee that a potent compound will become a marketable drug, a large number of new leads are needed to feed into the drug development process. It is desirable to find lead compounds in structurally diverse chemical classes. If multiple chemical classes can be found, they provide optional starting points for lead optimization of activity, physical properties, tissue distribution, plasma half-life, toxicity, etc.

Typically, hundreds to several thousand compounds are synthesized in the lead optimization phase of drug discovery. The drug optimization stage takes one to three years.

1.4.3 Preclinical Research

Tens of molecules might show sufficient promise to proceed to detailed safety and effectiveness studies. This stage involves in vitro (in the laboratory) and in vivo (in animals and human volunteers) studies to show biological activity of the molecule against the targeted disease. This stage takes approximately 2 to 3 years.

1.4.4 Clinical Trials

No amount of pre-clinical testing can predict with absolute certainty how humans will handle and tolerate a new drug. Clinical trials are designed to determine scientifically the safety and efficacy of various treatment regimens. This stage takes approximately five to seven years. There are three phases to complete before a compound becomes an approved drug:

Phase 1: About five molecules move on to safety and dosage range studies. Each molecule is tested on small groups of healthy volunteers to determine dosage limits, the disposition of the molecule in the human body and its side effects or toxicity.

Phase 2: About four molecules move on to early safety and efficacy studies. Hundreds of patients with the targeted disease or condition are recruited, on a voluntary basis, to participate in clinical trials to assess the effectiveness of the drug (i.e., evaluate the drug's efficacy on patients suffering from the illness that the drug is intended to treat).

Phase 3: About two molecules move on to large scale comparative studies. Only one of the many molecules screened and synthesized may prove its worth as an innovative therapy. National and international studies are conducted in clinics, research establishments and hospitals, where physicians monitor patients with the disease closely to confirm the effectiveness of the product and identify possible adverse effects.

1.4.5 Review and Approval

All the information gathered by the company, including chemical structure and properties, production details, pre-clinical and clinical studies are evaluated by regulatory agencies. If the studies demonstrate that the new drug is safe and effective, the company receives a notice of compliance allowing this new drug to be marketed. The new drug is then made available to the public, generally by prescription. This stage takes approximately two years.

Post-marketing surveillance studies and further comparative drug studies may need to be conducted for several years after the drug is marketed.

1.5 Changes in Drug Discovery

Major changes are taking place in the drug R&D process, especially in drug discovery, as new technologies advance and become available. Medical genetics leads to many more disease targets. Combinatorial synthesis makes millions of compounds available. The use of robotics and miniaturization is now allowing researchers to quickly screen thousands of compounds per week. These all will make a major impact in the drug discovery process.

High Throughput Screening (HTS) technology, which is an automation of biological assays of compounds, can investigate thousands of compounds against biological targets per week. The availability of large numbers of cloned receptors and enzymes has provided a vast array of biological target systems for drug discovery screening (Cummins et al., 1996). Important drug therapies may result from inhibition of these receptor systems, and extensive effort is currently being directed toward development of large libraries of compounds for screening. HTS programs are routinely used today for the identification of lead molecules in pharmaceutical discovery programs. The molecules are often selected from large chemical inventories maintained by research pharmaceutical corporations.

Combinatorial chemistry provides the logistics of mass production of compounds and a wide range of molecular diversity for drug discovery. Using combinatorial chemistry techniques large numbers of compounds can be simultaneously synthesized. For example, there are over 10,000 carboxylic acids (i.e., organic acids) and over 4,000 amines (i.e. organic compounds containing nitrogen) in the Available Chemicals Directory, 1995. Using a simple chemistry reaction coupling carboxylic acids with amines (i.e. $A+B \rightarrow C$), there are over 40 million possible products.

With the advance in the human genome project, hundreds to thousands of new potential molecular targets for new medicines will be identified through the use of genetics in the upcoming years. Given the size of today's chemical libraries and the additional millions of new compounds made available by combinatorial chemistry, it is no longer possible to test all available compounds for every new target of potential biological importance. In this thesis applications of modern statistical methods to enhance the efficiency and effectiveness of the drug discovery process are proposed.

1.6 Chemical Data Sets and Descriptors

Methods to be described here can be applied to both continuous and discrete responses. For illustration, a data set with continuous activity outcome (Core98) and a data set with binary activity outcome (NCI) are included.

1.6.1 Chemical Descriptors

The first step in the process of determining features of compounds that are important for biological activity is describing the molecules in a manner that is both capable of being analyzed and relevant to the biological activity. A drug-like molecule is a small three dimensional object that is often represented by a two dimensional drawing. This two dimensional graph is subject to mathematical analysis and can give rise to numerical descriptors to characterize the molecule. Molecular weight is one such descriptor. There are many more. Ideally, the descriptors will contain relevant information and be few in number so that the subsequent analysis will not be too complex. To exemplify our methods we use a system of BCUT descriptors given by Pearlman and Smith (1998), which are derived from a method of Burden (1989). These descriptors are eigenvalues from connectivity matrices derived from the molecular graph. For each heavy (non-hydrogen) atom, a property is placed along the diagonal of a square matrix. The atomic property can be size, atomic number, charge, etc. Off-diagonal elements measure the degree of connectivity between two heavy atoms. Since eigenvalues are matrix invariants, they measure properties of the molecular graph. Being functions of all the heavy atoms in the molecule, the eigenvalues are thought to represent the properties of the molecule as a whole. There are 67 BCUT descriptors described by Pearlman and Smith (1998). These 67 BCUT numbers are highly correlated and computational chemists often use a subset of six BCUT numbers. A reason for the high correlations is that scientists often devise descriptors that measure the same general property of a compound. I will typically follow the lead of the computational chemists and use six BCUT numbers. For illustration, both sets of 6 and 67 BCUT descriptors are considered in the analysis described in Chapter 4.

1.6.2 Core98 Molecular Data (Continuous Response)

Biological activity scores were obtained on a chemical data set, Core98, comprising 23,056 compounds. Core98 is a chemical data set from the GlaxoSmithKline collection. Activity was measured as % Inhibition and theoretically should range from 0 to 100 with more potent compounds having higher scores. Biological and assay variations can give rise to observations outside the 0-100 range. Typically, only about 0.5% to 2% of screened compounds are rated as potent. The compounds are described by 67 BCUT numbers.

1.6.3 NCI Molecular Data (Binary Response)

An AIDS antiviral screen chemical database can be obtained from the National Cancer Institute (NCI) web site http://dtp.nci.nih.gov/docs/aids/aids_data.html. It provides screening results and chemical structural data on compounds. When we downloaded the database in May 1999, there were about 32,000 compounds. GlaxoSmithKline computational chemists generated BCUT numerical molecular descriptors for these compounds. However, due to poor structural representation and samples that contain unusual chemical substances that would normally not be considered drug candidates, some BCUT descriptors could not be computed for some compounds. These compounds were removed, leaving about 30,000 compounds with computed descriptors.

In theory, every compound does have a unique BCUT value. In reality, however, some compounds do fail the calculation for the following reasons. First, the available structural representation of a compound may fail to convert from its 2-D representation to a 3-D structure. The BCUT calculation is a two-step process that includes a 2-D to 3-D conversion via a software program called CONCORD. If CONCORD fails to generate a 3-D structure, then most of the BCUT values can not be calculated. Second, the compound may contain features that cannot be parameterized for the BCUT calculation. Some of the features (atoms, substructures, etc.) contained in a molecule may not have parameters assigned to them. For example, the charge of certain atoms such as selenium may not be available in the BCUT calculations. Thus, an error would occur and calculation would fail. But, from a practical aspect, this is ideal. Medicinal chemists have little interest in, for example, compounds with highly strained ring fusions (which would cause CONCORD to fail) or compounds containing selenium (which would cause the BCUT calculations to fail). Because CONCORD and the BCUT calculations are parameterized for medicinally relevant compounds only, these steps become filters to eliminate undesirable compounds.

Like the Core98 data, the same set of sixty-seven BCUT descriptors were computed for the NCI data. However, unlike the Core98 data where the response is continuous, the NCI compounds are classified as moderately active, confirmed active, or inactive. The first two classifications are combined as 'active', as there are very few active compounds (only 2% for both combined).

1.7 Design of Experiments of Screening Sets of Compounds

Various molecular descriptors (explanatory variables) can be readily computed to describe the chemical properties of every molecule in the database (e.g., the BCUT descriptors). The numerical descriptors form a high dimensional chemical space where both active and inactive compounds reside. The design problem is to use the chemical descriptors to choose a set of compounds for assay.

Chemists and biologists wish to explore the relationship between biological responses and the descriptors. They are usually unable to state the functional form of the relationship. However, from their experience the functions will be more spiky than smooth (McFarland and Gans, 1986). They require, therefore, a large set of design points to cover the descriptor space. The design points are to be selected from an existing collection of molecules. Some of the problems related to conventional design of experiments are as follows:

1. Biological activity may be present via several mechanisms. Activity due to a particular mechanism might be restricted to a small sub-region of the descriptor space, and different sets of descriptors might be critical for the different mechanisms. Compound screening, as stated earlier, aims to identify active compounds of several different structural classes. Therefore, a good design should select a sample of compounds that leads to identification of active compounds in multiple regions in a high dimensional chemical space. These regions are referred to as 'active' regions. Ideally, the sample should include a few compounds from each of the active regions. However, the locations of the active regions are not known before screening and often in practice, almost every available compound is screened (this is going to be changed though, as it becomes impossible to screen every compound for every new target).
2. The model is vague. The model relating the biological response to molecular properties is usually unclear beyond the assertion that similar objects are more likely to respond similarly. Two molecules must have fairly close values of all critical descriptors for similar biological activity (McFarland and Gans, 1986). Thus, the chosen subset, or experimental design, should "fill" or "cover" the numerical space in some sense. Ideally, selected molecules should be as dissimilar as possible and any candidate molecule not selected should be near a selected molecule.
3. There is more than one response. A number of biological screens will be in operation within the research division of a pharmaceutical company at any given time. Each screen may use different

receptors or enzymes, aiming to detect a different specific biological activity. It is hoped that the collective output of these screens will provide enough leads to contribute to the discovery process in a meaningful way. For the Core98 data set described earlier, 15 biological responses were generated.

4. There is a high-dimensional, discrete sampling space. The compounds to be chosen from are often a collection of restricted sampling points and not all combinations of the descriptor values are feasible. It is not possible to place a compound at certain positions in the space. You are restricted to the compounds you have or can make. Therefore, a standard experimental design procedure assuming a continuous or regular factorial space will not work. In addition, chemical compounds, including those in Core98, are generally made for a purpose; once a good compound is found, similar additional compounds are made. Therefore, candidate compounds are unevenly distributed in the space.
5. The number of candidate points (N_c) and the number of design points (n_d) are large. The number of possible combinations of the samples to be chosen from is so large that it becomes computationally impossible to consider every possible combination in the experimental design (Higgs et al., 1997) – it may take days or weeks or even months to compare every combination. In theory, to identify the optimal design, one needs to examine all possible subsets of size n_d from the N_c candidate points, thus performing N_c-chose-n_d subset evaluations. In practice, the magnitudes of N_c and n_d prohibit a full-scale optimization. For example, to choose a relatively small design of 500 points from a candidate set of 10,000 molecules, there are $2.53\text{x}10^{860}$ possible subsets. For moderate or large data sets, exhaustive search is not attempted, and heuristic algorithms are usually applied to find a very good design.
6. The number of descriptors can be very large. Tens to hundreds to thousands of molecule descriptors are possible (Cummins et al., 1996, Higgs et al., 1997, and Hawkins et al., 1997). This implies a high dimensional problem.

When there is no prior model relating biological response to molecular properties, the generally accepted procedure is to screen a diverse subset of the overall database, and then examine the compounds that are structurally similar to any promising leads that are found (Dixon and Villar, 1998). Measures of "diversity" and "similarity" are based on numerical molecular descriptors. Space filling designs (described in Chapter 2) are commonly used for selection of compounds for screening.

1.8 Analysis Problem

The set of possible compounds that could be made is usually huge and it is not practical (too expensive and time consuming) to screen every possible compound. Thus, statistical analysis of the data from the initial screen aims to uncover the relationship between the numerical descriptors and biological activity, to focus further screening on the most promising compounds. The relationship between descriptors and activity is extremely complex for high throughput screening (HTS) data and there are several challenges in statistical modeling of data of this sort.

1. The potent compounds of different chemical classes can be acting in different ways. Different sets of descriptors might be critical for the different mechanisms. Activity may be high for only very localized regions. A single mathematical model is unlikely to work well for all mechanisms.
2. The scarcity of active compounds makes identifying these small regions difficult. Even though a design or screen may include thousands or tens of thousands of compounds, it will usually have relatively few active compounds. For example, the National Cancer Institute data set described earlier has only about 2% active compounds.
3. There are many descriptors (i.e., curse of dimensionality). One can always find something interesting in a high-dimensional space. Whether it is real or not is another story.
4. Chemical descriptors are often highly correlated (e.g., the BCUT descriptors), as scientists often devise descriptors that measure the same general property of a compound.
5. HTS data can be subject to very large systematic and random measurement errors in assay results. However, there is no measurement error in the computer-generated numerical molecular descriptors.
6. Getting a good predictor of activity for unscreened compounds is difficult as biological activity scores are often nonlinear responses of compound properties involving thresholds and interactions.

Common statistical analysis methods such as linear regression models, generalized additive models, and neural nets are ineffective in handling these analysis problems (Young and Hawkins, 1998) and tend to give low accuracy in classifying molecules as active.

1.9 Outline of Thesis

This thesis discusses statistical methods for design and analysis of large chemical databases for high throughput drug screening. The design problem is to choose a representative set of thousands of chemical compounds from a large collection of compounds. Conventional experimental design methods are not developed for such large data sets. Here we introduce a more efficient and effective design method for these data sets. Our method can run hundreds to thousands of times faster and find a better (coverage) design than other design methods. The analysis problem is to find regions of a high dimensional space where active compounds reside. Here we introduce a new analysis method that gives better prediction of active compounds than existing methods. This research area is rather large and complex. To avoid computer memory and space issues due to a large volume of data, the methods are studied and developed using tens of thousands of compounds instead of millions but the algorithms should work on larger data sets.

Chapter 2 describes existing design and analysis methods for compound selection. Other design and analysis issues (e.g., dense coverage of space, outlying compounds, multiple testing, etc.) not yet covered will be discussed. Chapter 3 describes a novel design method developed to address the design issues. The new design method is a useful tool for selection of a subset representing a large set in a chemical descriptor space. Chapter 4 describes a novel analysis method developed to address the analysis issues. The new analysis method is capable of finding multiple, active regions and finding more hits up front (i.e., find a high proportion of hits from a small number of compounds screened). Chapter 5 gives a summary of the results and discusses some related areas for future research.

Throughout the thesis, the terms molecule, compound and structure are used interchangeably. Technically, a compound is a chemical substance with two or more elements. A molecule is the simplest structural unit that displays the characteristic physical and chemical properties of a compound. A structure is the graphical presentation of the bonds and atoms of a compound. In drug discovery, a compound is considered a chemical substance not yet defined while a molecule is a defined compound. Graphically, a molecule is referred to as a chemical structure.

Chapter 2

Design and Analysis Methods for Compound Selection

2.1 Introduction

Design of experiments and statistical analysis of designed experiments, when used properly, can increase the effectiveness in identifying active regions (where active compounds reside) and reduce the number of compounds screened. A sound approach is to carefully select, using design of experiments, a relatively small subset of compounds for screening and to determine, through statistical analysis, the molecular features important for biological activity. Rules developed from the analysis can then be used in further screening to focus attention on compounds most likely to be active. Such a sequential strategy is expected to be more efficient than screening all the compounds in a large collection (Jones-Hertzog et al. 2000).

Commonly used statistical methods for design and analysis of chemical databases and the problems related to these methods are described in this chapter.

2.1.1 Notation

In general, denote the k continuous descriptors by x_1, x_2,..., x_k. Within the full k-dimensional descriptor space, a p-dimensional (p-D) subspace is defined by p of the k descriptors ($1 \leq p \leq k$). For convenience, Xi will denote the 1-D subspace involving only x_i. Similarly, Xij, $Xijl$, etc. will represent 2-D, 3-D and higher-dimensional subspaces. For example, $X1$ is a 1-D subspace defined by x_1, and $X12$ is a 2-D subspace formed by x_1 and x_2. A subspace, then, is simply a subset of the descriptor variables, ignoring the remaining descriptors.

2.2 Design

The design objective is to choose a representative subset of compounds from a large collection of compounds for screening. Our problem differs somewhat from conventional design of experiments. First, the candidate set of possible explanatory variable combinations is discrete, and the set of discrete points can be large and highly irregular. Second, the model relating the biological response to molecular descriptors is not known, and the response function is likely to be highly non-linear. Third, the design set (the subset) and the candidate set (the collection) are usually large (e.g., choose hundreds to thousands of compounds from thousands to millions of compounds). In contrast, many existing design criteria and optimization algorithms are aimed at choosing a small sample from a continuous or regular (e.g., factorial) sampling space, with a specific model in mind.

In reviewing the many classes of designs available and the difficulties in using them for HTS data, we can distinguish between model-based and space-filling (model-free) designs. Model-based designs tend to be more popular for lead optimization, while space-filling designs are widely used to select subsets of molecules from large chemical databases for lead generation.

2.2.1 Model-Based Designs

Most work in classical design of experiments has a class of (regression) models in mind, at least implicitly. Given a model, design points can be chosen so that parameters in the model are efficiently estimated or a predictor has low variance, etc.

In the context of HTS, once a lead compound is identified, compounds near the lead are examined to find better leads. Compounds within a small neighbourhood of the lead compound usually belong to the same chemical class, so a simple model (e.g., linear regression) is more likely to work adequately. The predictor variables are the chemical descriptors. The fitted model can be used to choose compounds (synthesized or not yet synthesized) in the neighbourhood with the highest predicted activity.

There are many model-based design criteria; most are concerned with efficient estimation of the parameters in the model or efficient prediction of the response (see, for example, Cox and Reid, 2000, Chapter 7). For instance, the D-optimality criterion chooses a design to minimize the determinant of the variance-covariance matrix of the parameter estimators, hence giving efficient parameter estimation, whereas a G-optimal design minimizes the maximum variance of prediction over the experimental domain. D-optimality is the most common criterion for computer-generated optimal designs, as it is mathematically and computationally convenient and is invariant to linear reparameterization of the model.

To search for an optimal model-based design, an algorithm is applied to optimize the chosen criterion. A sequential search algorithm (Dykstra, 1971) starts with an empty design and adds successive design points once at a time so that the chosen criterion is optimized at each step. Such a sequential search is the fastest but least reliable method. A simple exchange method (Wynn, 1972 and Mitchell and Miller, 1970), starting with an initial random design, improves it by adding a candidate point and then deleting one of the design points until the design criterion cannot be improved further. This simple exchange method is the next fastest algorithm and is more reliable than sequential search. The most reliable but computer-intensive algorithm is the Fedorov exchange algorithm (Fedorov 1972). This method searches over all possible pairs of candidate and design points for each exchange and thus runs much slower. There are many other algorithms for optimizing a design criterion and most are variants on the basic idea of an exchange; see Cook and Nachtsheim (1980) and Tobias (1995, pp. 657-728) for reviews.

If the model is linear in the descriptors, compounds on the edge of the space are selected. Even if quadratic or higher-order terms are present, these designs tend not to represent the diversity of a chemical collection. Model-based designs are seldom used for lead generation, and hence will not be discussed any further.

2.2.2 Space-Filling Designs

Space-filling designs are useful in situations where the experimenter cannot specify the functional form of the response function. The most common space-filling designs for selecting a representative set of molecules are random designs, distance-based designs, and cell-based binning designs.

Random designs

The simplest designs are based on random sampling. In fact, most new leads have been discovered through random screening, in which large numbers of compounds are tested for a specific biological activity; and the active compounds are then selected for optimization. Young et al. (1996) used a constant radius hypersphere around each randomly selected compound to measure the coverage of the descriptor space. Because two compounds must have very similar values of all critical descriptors to have similar properties, each hypersphere extends only a small distance in each dimension and covers only a tiny region of a high dimensional space. If the sample size is relatively small (e.g., hundreds of compounds), these hyperspheres will not give a sufficient coverage of the high dimensional space, regardless of the type of designs used for the selection. Young et al. (1996) concluded that, unless a very large number of compounds are used to fill space, randomly selected compounds will cover as much space as carefully selected compounds. On the other hand, if the important dimensions for a particular problem are identified, and if a focused set of compounds is desired, then rational selection should be more effective than random designs.

Random designs are popular for the following reasons. First, it is very convenient to generate a large random design. No computation of distances among compounds and no optimization of a design criterion are required. Second, a large random sample tends to have a distribution in the space that is similar to that of the candidate compounds. Third, based on my experience with several data sets, random designs tend to give better coverage in low dimensional projections than conventional (distance-based and cell-based) designs focusing on coverage of a high dimensional space.

There are some problems with random designs, however. Random selection generally does not give a good representation of all compounds in the database. It tends to over-select compounds in the dense regions and to under-select compounds in the scarce regions. As the volume of the space increases exponentially with the number of descriptors, getting a good coverage of a high dimensional space is almost impossible. A random design does not guarantee a good coverage in any dimensions of interest.

The coverage of designs generated from both simple random sampling and stratified random sampling will be shown in Table 3-1.

Distance Designs

There are two main types of distance-based designs for selecting molecules from chemical databases: "Spread" and "Coverage" designs, also known as maximin and minimax distance designs. These methods first define a descriptor distance metric (e.g., Euclidean or Manhattan distance) to measure the similarities or dissimilarities of the molecules, and then find those molecules that 'fill' the space based on some distance criterion. Spread designs (Kennard and Stone, 1969) identify a subset of molecules that are maximally dissimilar with respect to each other. The spread criterion seeks to maximize the distances between design points. Coverage designs select a subset of molecules that are similar to the candidate set of molecules. Zemroch (1986) achieved this by clustering the candidate points and choosing a representative member of each cluster, whereas Johnson et al. (1990) sought a design to minimize the maximum distance from any candidate point to a point in the design. Finding a coverage design is usually much more computer-intensive than finding a spread design. The spread criterion depends only on the distances between design points, whereas the coverage criterion depends on the distances between all pairs of candidate points. The coverage criterion usually leads to a design with better representation of all compounds in the database, however. Higgs et al. (1997), Johnson et al. (1990), and Tobias (1995) give more detailed descriptions of these designs.

Because of the large numbers of candidate and design points, it is not possible to find the 'best' design by evaluating every possible subsets of compounds. Optimization algorithms such as sequential search and the exchange algorithms described earlier can be adapted to search for an optimal spread or coverage design (Higgs et al. 1997 and Marengo and Todeschini 1992). However, the existing exchange algorithms were not intended (i.e., are too computationally intensive) for design problems of the magnitude considered here. Alternatively, Higgs et al. (1997) and Zemroch (1986) apply clustering algorithms to approximate a coverage design. A price for using the clustering approach is that it can generate many small clusters with only one compound and few large clusters with hundreds to thousands of compounds. In addition, the number of clusters can be much larger or smaller than the number of design points required.

Even if the computational issues of finding an optimal design can be resolved, there are several problems with distance-based design criteria. First, two molecules with fairly close values of all critical descriptors are likely to have similar biological activity, but beyond some (unknown) threshold, there may be little relationship between distance and similarity of activity. Second, descriptors that are unrelated to target activity can have a significant impact on the distances between molecules in the space, and can make the "optimality" of a design irrelevant. Irrelevant variables can create a 'large' distance between two similar molecules. Without proper selection of descriptors, these optimal designs are not expected to improve the quality of rational sampling over that of random sampling. In general, these distance methods try to find a subset with optimal coverage of the entire descriptor space but pay little attention to the coverage in lower dimensional subspaces. The low dimensional coverage (i.e., 1-D, 2-D and 3-D) can be quite poor which can cause problems in estimating local, low-dimensional effects. Morris et al. (1993) and Morris and Mitchell (1995) addressed this issue by incorporating 1-D coverage into their spread designs. If the effects of the descriptors can be adequately modeled by main effects and lower order interactions, then good coverage in lower dimensions is more important than good higher dimensional coverage properties. Third, the presence of relatively few outlying observations leads to large, dominating inter-point distances. Very often this requires removal of many molecules to generate a sensible design.

Cell-based Designs

To measure the "coverage" of a descriptor space, the space is divided into cells. A good experimental design will ideally have at least one molecule in every cell. If so, we say the space is covered.

In the conventional cell-based method, the range for each of the k numerical descriptors is subdivided into m bins of equal size, yielding m^k cells or hypercubes, and the experimental design chooses at least one molecule from every cell. This method is attractive because it is easy to divide the descriptor space into cells and allocating even a very large dataset to these cells is straightforward. Missing diversity (i.e., empty cells) can easily be identified. Cummins et al. (1996) and Menard et al.

(1998) used cell-based binning methods to compare the relative diversity of molecular databases and to select diverse subsets of molecules.

A problem with many existing cell based binning methods is that they may generate too many cells. For a high-dimensional space, the number of cells can exceed the number of candidate molecules and thus the number of design points. For example, if k=6 and m=10, then this leads to a million cells. Most will be empty, even with respect to the candidates, and it is not possible to cover the cells with any design. Even if only 10% of the cells are nonempty, we would need 100,000 design points to cover these cells.

To reduce the number of cells, a common approach is to generate fewer, wider bins, but these bins may include rather dissimilar compounds. For example, Cummins et al. (1996) and Menard et al. (1998) limited the number of descriptors and the number of bins per descriptor. They also excluded hundreds to thousands of outlying candidate points (as outliers lead to an artificially large space). Cummins et al. used factor analysis to reduce the dimensionality of the descriptor space to four; in addition, they removed molecules in low density cells containing seven or fewer molecules (as a way of removing outlying molecules), excluding a total of 6,986 molecules. Menard et al. restricted the number of descriptors to 3-6 and the number of bins per descriptor to 4-7 and excluded a large number of candidate points by treating them as outlying observations. Even with these restrictions, there can be a large number of empty cells. For example, Menard et al. treated 10% of the 628,000 compounds as outliers and still reported over 80% of the 6^6 cells empty. Even with all these compromises, Cummins et al. and Menard et al. reported a large proportion of empty cells, many compounds densely clustered in a few cells, and many cells being singleton. Obviously, some of the excluded molecules can be potential leads. This indicates that the existing cell-based binning approaches are not adequate. Indeed, a very low cell occupancy is expected by Menard et al. – they recommended a targeted occupancy of 12-15%. With such a large proportion of cells being empty, it is not meaningful to measure coverage of the space.

In a high dimensional space, it is practically impossible or very difficult to densely fill the entire space with hundreds to thousands of design points. Since two molecules must have fairly close values of all critical descriptors for similar biological activity (McFarland and Gans, 1986), m should be relatively large. But this will generate too many cells even with a smaller number of descriptors, making it impossible to find design points that give good coverage with so many cells. On the other hand, if one knew, in advance, the few critical descriptors responsible for the particular biological activity, then one could have selected design points that gave good coverage over those relevant subspaces. In Chapter 3 uniform coverage designs aimed at addressing this issue are introduced. This design method keeps the number of cells low, allows the inclusion of all molecules and generates a high percentage of occupied cells.

Other Space-filling Designs

Two popular space-filling designs, currently only applied to more regular sampling spaces, are Latin hypercube designs (McKay et al., 1979) and uniform shell designs (Doehlert, 1970). Latin hypercubes have excellent 1-D coverage and are very popular in experiments with computer models. The main problem in applying these methods to compound selection is that for chemical compounds only certain combinations of descriptor values exist. You cannot place a point anywhere you want to. The candidate points come from a collection of chemical compounds. Even if all 1-D cells are not empty, the Latin Hypercube design can still pick compounds that do not exist (e.g., in an empty 2-D cell). The same problem applies to uniform shell designs. On the other hand, our proposed design always selects from the existing candidate points.

2.3 Statistical Analysis

In drug discovery, the search for lead compounds for a biological target usually involves screening a large number of compounds. The screened compounds form large structure-activity data sets containing information about the chemical structure or features of the each compound (quantified by descriptor variables) and the corresponding biological activity. Analysis of the structure-activity relationship enables development of prediction rules that guide the selection of promising compounds for screening, thus reducing the overall screening time and the total number of compounds screened. For HTS data sets, the analysis objective is to find active regions in a high dimensional space where active compounds reside and to develop prediction rules based on these active regions.

There are several difficulties with analysis of HTS data that make many conventional statistical methods ineffective or inadequate (Hawkins et al., 1997 and Young and Hawkins, 1998). The relationship between descriptors and activity is extremely complex for HTS data. Compounds might act in any of several different ways to elicit a biological response. Some compounds might bind at one site and others at another (alosteric) site. The chemical features important for one mechanism are unlikely to be important for another. Activity may be high for only very localized regions and can be highly nonlinear. Threshold effects may be present, where some chemical feature must be present at some threshold level for activity to occur but activity is constant above this level. Interaction effects, requiring the simultaneous presence of two chemical features, are also plausible. Therefore, statistical modeling needs to be able to accommodate multiple mechanisms, thresholds, interactions between descriptors, and nonlinearities.

In addition, generally only a small proportion (about 1%) of compounds screened are active. Most methods, however, are driven by criteria aimed at good overall prediction accuracy, criteria that are dominated by the overwhelming majority of inactive compounds. The imbalance of active and inactive compounds makes identifying active regions difficult.

There is also the general issue of curse of dimensionality (Hastie and Tibshirani 1990 and Scott and Wand 1991). For instance, the number of possible parameters in a polynomial regression model of degree 3 including interaction terms of 2 and 3 descriptors increases quickly. For k descriptors, there are $\binom{k+3}{3}$ parameters in total. There are, for example, 84 parameters for 6 descriptors and 54,740 parameters for 67 descriptors. In high dimensional space, nearly all data sets are sparse and show multicollinearity, making the fitted model highly unstable. Matching compounds with similar chemical features (descriptor values) becomes impractical in high dimensions, since virtually every compound is distinct in some dimensions. Classical statistical methods such as regression analysis were designed to work for low dimensional data and can quickly become extremely unreliable in high dimensions due to the curse of dimensionality.

Illustrations of some existing statistical analysis methods for HTS data and their related problems are given next.

2.3.1 Linear Regression Models

Most of the modeling issues described above apply to linear regression analysis. For illustration, stepwise regression is applied to the Core98 data (continuous response) with six descriptors. To allow for interactions and nonlinearities, polynomial regression models of degree 3 including interaction terms of 2 and 3 descriptors were fitted to the Core98 data using the stepwise-selection method. The 'best' model had $R^2 = 0.01$ and poor prediction accuracy in identifying compounds as active. This example illustrates that linear regression models are not reliable for HTS data. Linear regression models assume that activity varies linearly with descriptor values, which is an inappropriate assumption for HTS data. These models aim to minimize the mean sum of squares, a criterion that is dominated by the overwhelming majority of inactive compounds. Linear regression models cannot handle multiple mechanisms and the other analysis issues mentioned earlier.

For the NCI data (binary response), logistic regression models were also investigated. Overall, low prediction accuracy in classifying compounds as active and high prediction accuracy in classifying compounds as inactive were found. As only about 2% of compounds are active, any methods claiming all compounds as inactive will give an overall accuracy of 98%. The real challenge is to find a high proportion of active compounds. Logistic regression is not effective in handling the modeling issues (e.g., multiple mechanisms, thresholds, interactions, etc.) described above.

2.3.2 Cluster Significance Analysis

Cluster significance analysis (CSA) (McFarland and Gans 1986) aims to find embedded regions of activity in a high dimensional chemical space. Suppose that active compounds have a molecular weight between 400 and 500 and a melting point between 160 and 205 degrees C. If compounds that range in molecular weight from 250 to 750 and melting point from 120 to 270 degrees C are tested, then simple statistical analysis methods, linear regression, can miss finding the relationship. A simple plot of the data shows the cluster of active compounds (squares in Figure 4-1a). CSA computes the average Euclidean distance between active compounds in a subspace of the high dimensional space and compares that distance to the average distance of an equal number of randomly selected (active or inactive) compounds. If the actives are clustered more tightly, then that is evidence that the dimensions where the actives are clustered are the descriptors that are important for activity. Suppose that the descriptors are compared, two at a time and the active compounds are clustered closely together only in the subspace of molecular weight and melting point. That would imply that these two descriptors are important.

CSA tacitly assumes, however, that there is only one class of active compounds. If there are two widely separated clusters of active compounds in a low-dimensional projection, possibly from two mechanisms, the distances between compounds in different clusters will be large. The CSA criterion of average distance between active compounds might not be significant for this important projection. Even worse, activity from two or more mechanisms may be due to different subsets of descriptors. One might see no clusters when looking at the active compounds in the molecular weight and melting point projection example discussed above. Active compounds from other mechanisms due to different sets of descriptors might be spread throughout the molecular weight and melting point projection. These problems are discussed in more detail in Chapter 4.

2.3.3 Recursive Partitioning Analysis

The analysis of multi-mechanism data is difficult, and many statistical methods are not expected to be successful. Recursive partitioning (RP) is one exception where good results have been obtained (Hawkins et al. 1997, Young and Hawkins 1998, and Rusinko et al. 1999).

RP encompasses tree-based models, which date back at least to Morgan and Sonquist (1963). Well-known implementations include Formal Inference-based Recursive Modeling (Hawkins, 1999) and Classification And Regression Trees (Breiman et al. 1984). Venables and Ripley (1999, Chapter 10) give a good account of how the CART methods may be executed in S-Plus.

RP recursively splits a data set into progressively smaller and more homogeneous, disjoint subsets. The disjoint subsets are called nodes. The first node containing the entire data set is called the root node. Thus, each node is potentially the parent of two or more daughter nodes (but most commercially available tree software allows only binary splits). To choose an optimal partition for a node, all possible cut-points (ordered variables) or divisions of categories (unordered categorical variables) are examined. Each daughter node is split in turn until the nodes are judged homogeneous or some minimum sample size is reached. This separation of the data into smaller data sets can separate the components of a mixture into separate groups where only a single mechanism operates. An example of a RP analysis on the NCI data set is shown in Figure 2-1. The terminal nodes are used for prediction. By following the partitioning rules, a new, untested compound is assigned to one of the terminal nodes; it is given a score based on the activities of the tested compounds in that node.

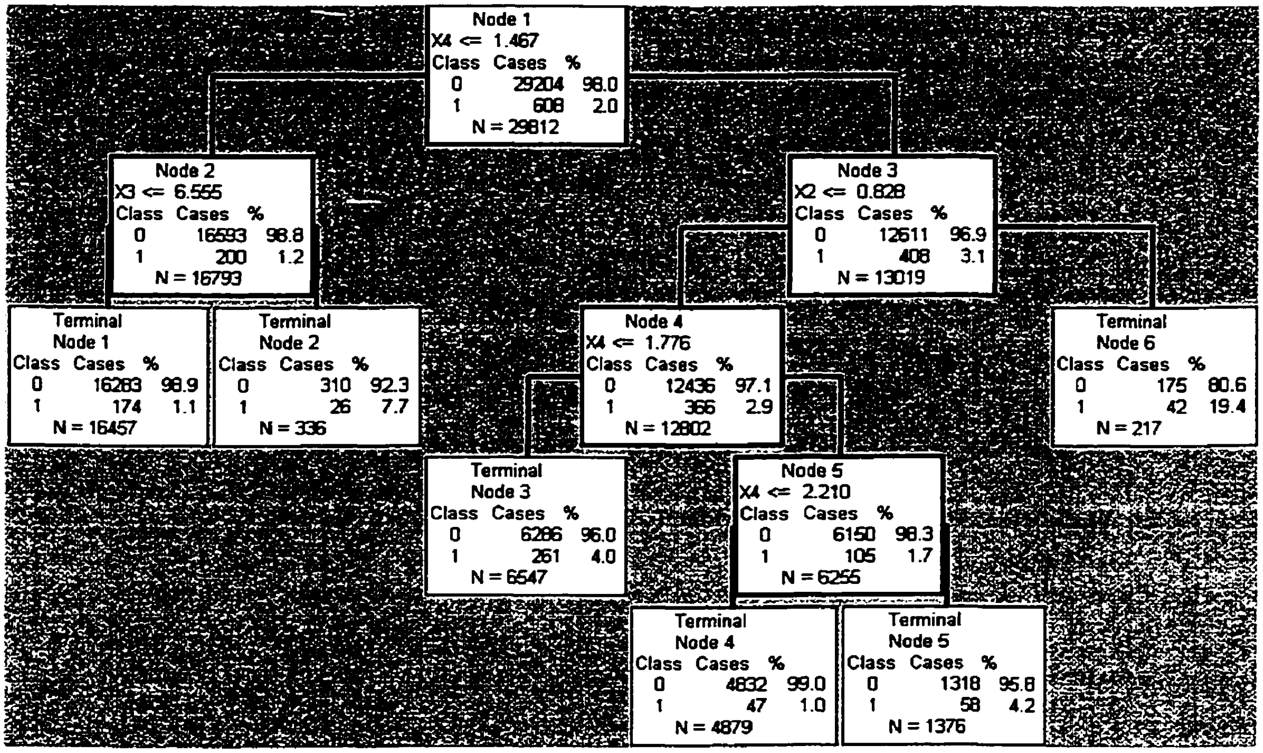


Figure 2-1. A Tree for the NCI Data.

Rules are used to split the NCI data set into progressively smaller subsets. All the data is present at the top of the splitting diagram. Classes 0 and 1 represent inactive and active compounds, respectively. The feature that best separates more active from less active compounds is used to split the data set. For example, at Node 1 x_4 with a cut-point of 1.467 is used to split the data into Node 2 and Node 3. Each compound ends up in one terminal node and the rules that lead to the node define the features important for that class of compounds. To get a better view of the splits, a small tree (of size 6) is chosen for illustration; this may not be optimal.

Depending on the type of response variable, classification trees are used for modeling a categorical response whereas regression trees are for modeling a continuous response. In both cases, however, the mathematical concepts behind building a tree are very similar. In general, there are two types of construction algorithm. One approach is to use a node splitting criterion (e.g., misclassification rate or deviance) to grow a large tree and then a cost-complexity criterion to prune back the tree to a smaller size. The other approach is, at each potential split, to perform a significance test (e.g., a t-test), adjusted for multiplicity, to determine whether to make a split or not. The latter usually runs much faster and can handle larger data sets but could miss some potential splits further down the tree.

As successful as RP has been for the analysis of HTS data sets (Jones-Hertzog et al. 2000), there are a number of possible problems. First, this approach selects one descriptor at a time to split the data set. But a single descriptor may not provide adequate information for the splitting process. In addition, when the descriptors are highly correlated, selecting one descriptor will likely lead to not selecting several others. The second problem relates to multiple mechanisms when some of the active regions are near or overlapping each other. This will be illustrated in Chapter 4. The third problem relates to the number of splits. Binary splits are often used in recursive partitioning. Problems can result if the activity pattern is inactive-active-inactive as, however the single cut point is chosen, actives will be combined with inactives. It is important to keep the following observation in mind: two compounds must have fairly close values of all critical descriptors for similar biological activity (McFarland and Gans, 1986) when there is a single mechanism. This means that partitions have to be narrow, and in several dimensions simultaneously, if all molecules from a partition are to have similar activity.

Performance comparisons between tree-based methods and a novel cell-based analysis method using the Core98 and NCI data sets are described in Chapter 4.

2.3.4 Other Analysis Methods

Methods such as generalized additive models and neural nets can handle nonlinear responses but are not effective in dealing with interactions and multiple mechanisms (Young and Hawkins, 1998 and Hawkins et al., 1997).

2.4 Summary

Our design problem is somewhat special. The candidate set of possible explanatory variable combinations is discrete and the set of discrete points can be large and highly irregular. Figure 3-1 shows the univariate and pairwise plots of the six descriptors for the 29,812 NCI molecules. It is clear that much of the space is empty. Either the collection is missing chemicals or it is not possible to make compounds with certain combinations of descriptors. In more than two dimensions this problem will be even worse. It is believed that two compounds must have very similar values of all critical descriptors to have similar properties. Thus, the design needs to cover the space densely. It is clearly impossible to achieve dense coverage in high dimensional space without an extraordinarily large design. The most common designs for selecting a representative set of molecules are random designs, distance designs, and cell-based binning designs. Unfortunately, these designs try to select a set of diverse molecules that give a good coverage of all candidate molecules in a high dimensional space. A special type of space filling design aimed for uniform coverage in all 1-D, 2-D, and 3-D projections is introduced in Chapter 3.

Of existing data-mining methods, classification and regression trees (recursive partitioning) have had the most success for HTS data (e.g., Hawkins et al., 1997 and Jones-Hertzog et al., 2000). Although these methods are generally well suited to modeling of local behaviour, they otherwise pay little attention to the complexities of HTS data. For HTS data, the analysis goal is to identify the most promising compounds for screening and thus the prediction accuracy is focused on classifying compounds as active. Most methods are driven by criteria aimed at good overall prediction accuracy, criteria that are dominated by the overwhelming majority of inactive compounds. Adjusting these methods to aim for high hit rates for the relatively few compounds chosen for further screening would bring them closer to the real goal. A new statistical analysis method called cell-based analysis is introduced in Chapter 4. This method can handle multiple mechanisms, thresholds, interactions, nonlinearities, and imbalance of active/inactive compounds. Some of the criteria developed for this new method can be transferred to other methods such as classification and regression trees.

Chapter 3
Uniform Coverage Designs

3.1 Introduction

The use of robotics and miniaturization is now allowing researchers to quickly screen thousands of chemical compounds (molecules) for biological activity. Combinatorial chemistry provides the logistics of mass production of compounds and a wide range of molecular diversity for drug discovery. The automation of biological assays, High Throughput Screening (HTS), allows for investigation of thousands of chemical compounds against biological targets per week. While this brute-force approach to lead generation certainly has its place in the field of drug discovery, it is not practical, given the size of today's chemical libraries (e.g., hundreds of thousands to millions of compounds), to test every available compound for every new target of potential importance.

Various molecular descriptors (explanatory variables) can be readily computed to describe the chemical properties of every molecule in the database. When there is no prior model relating biological response to these descriptors, the generally accepted procedure is to screen (test) a diverse subset of the overall database to find active compounds of several structurally different chemical classes, and then examine further compounds that are structurally similar to any promising leads. If multiple chemical classes can be found, they provide optional starting points for further optimization of activity, physical properties, tissue distribution, plasma half-life, toxicity, etc. Ideally, selected objects should be as dissimilar as possible and any candidate not selected should be near a molecule in the experimental design. Measures of "diversity" and "similarity" are based on the numerical descriptors. The assumption here is that similar chemical objects are more likely to have similar biological responses. Thus, if an initial subset is to be selected, the subset should "fill" or "cover" the numerical space. In high dimensional space, nearly all data sets are sparse, and it is not possible to densely cover a high-dimensional space with thousands of design points. Therefore, we focus on filling or covering low-dimensional projections of the space instead.

To measure the "coverage" of a descriptor space, we will be dividing the space into cells. In a conventional cell-based method, each of k numerical descriptors is subdivided into m bins of equal size, yielding m^k cells or hypercubes, and the experimental design chooses at least one molecule from every cell. A good experimental design will ideally have at least one molecule in every cell. If so, we say the space is covered. Cummins et al. (1996) and Menard et al. (1998) used cell-based methods to compare the relative diversity of molecular databases and to select diverse subsets of molecules.

Such cell-based methods are attractive for several reasons. It is easy to divide the descriptor space into cells, and allocating even a very large dataset to these cells is straightforward. Choosing a design by random sampling is also easy. Missing diversity (i.e. empty cells) can easily be identified.

The key problem with the conventional cell-based method is that a high-dimensional space will have too many cells to be covered by a modest number of compounds (design points). As two molecules must have fairly close values of all critical descriptors for similar biological activity (McFarland and Gans 1986), the number of bins, m, should be relatively large. If k=6 and m=10, say, we have one million cells, which cannot be covered by only thousands of design points. This is just the curse of dimensionality.

To reduce the number of cells, a common approach is to use fewer, wider bins in each dimension, even though these bins may include rather dissimilar compounds. For example, Cummins et al. (1996) and Menard et al. (1998) restricted the number of descriptors and the number of bins per descriptor. They also excluded hundreds to thousands of outlying candidate points (as outliers lead to an artificially large space). Even with these compromises, they reported a large proportion of empty cells, many compounds densely clustered in a few cells, and many cells being singleton. Indeed, a very low cell occupancy (i.e., at least one compound) rate is expected by Menard et al. (1998) — they recommended a target occupancy of 12-15%. If most cells are empty and hence most of the space is ignored, however, the utility of covering the remaining space is questionable, calling for new methods of binning and creating cells.

If only a few descriptors are responsible for the particular biological activity, however, it is possible to densely cover their low-dimensional subspace with just thousands of design points. A subspace is simply a subset of the descriptor variables, ignoring the remaining descriptors. Different sets of critical descriptors may be relevant to structurally different chemical classes, but hopefully only a few variables are involved at a time. If we knew, in advance, that certain subsets of descriptors were critical we could choose design points to give good coverage of the relevant subspaces. At the outset, we will probably not know which descriptors are critical, and we therefore aim for uniform coverage in every low-dimensional projection. With m=10 bins per descriptor, for example, it is theoretically feasible to cover all 10^3 cells in any three-dimensional subspace with about 1000 points. This is analogous to a fractional-factorial design projecting down to a full factorial in a few critical variables.

Thus, because of the practical difficulty of covering the numerical space of all descriptors, and the belief that probably relatively few descriptors are active for any given mechanism, we will concentrate on low-dimensional subspaces throughout this article, typically involving one, two, or three descriptors.

Designs with good coverage of low-dimensional subspaces have been suggested in many other contexts. For example, Dalal and Mallows (1998) proposed plans for testing software such that for any f input factors, all combinations of their levels occur at least once. Typically, f is 2, 3, or 4. Thus, these designs exhaustively cover the input-factor space when projected down onto f-dimensional subspaces. Although the objectives are similar, these plans cannot be directly applied to molecule selection. Suppose we grouped each descriptor's values into a moderate number of bins to generate "levels". For an experimental run, the Dalal and Mallows (1998) designs can choose any combination of levels (bins) over all factors (descriptors). Unfortunately, a set of candidate molecules will typically have some bin combinations that are empty. We start with a candidate set of molecules, and we cannot necessarily select an arbitrary combination of descriptor values and place a design point there. The haphazard combinations of descriptor values similarly rule out plans based on Latin hypercubes and orthogonal arrays with good projective properties (Owen 1992 and Tang 1993) that have been proposed for computer experiments. The same difficulty arises with many other designs aiming for uniform space-filling properties, for example, the uniform shell designs of Doehlert (1970) or number-theoretic methods for generating representative points motivated by discrepancy measures (e.g., Fang et al. 1994).

Algorithmic, rather than combinatorial, methods can generate space-filling designs from any given set of candidate points. They typically optimize some function of the inter-point distances. Johnson et al. (1990) proposed two classes of designs, based on either minimax or maximin distance criteria. Maximin designs maximize the minimum distance between design points. By making the design points maximally dissimilar they spread throughout the space; the algorithm of Kennard and Stone (1969) has this underlying objective. Alternatively, minimax distance designs minimize the maximum distance between candidate points and the design points. This criterion tries to find a design such that every candidate is close to a design point and hence the design covers the candidate space. Similarly, Zemroch (1986) clustered the candidate points and chose a member of each cluster to cover or represent the entire set. Thus, distance-based algorithms appear to be useful for molecule selection and have been applied in this context (Higgs et al. 1997) and are readily available in SAS (Tobias 1995, pp. 657-728).

There are several difficulties, however, with distance-based design criteria. All of the methods mentioned above are based on distance metrics calculated from all descriptors. As we have already noted, it is not possible to densely cover a high-dimensional space with only thousands of points. Low-dimensional coverage, which is more relevant if few descriptors are critical, is not directly considered and could be quite uneven (some results will be presented in Section 3.6.4). Moreover, the definition of an appropriate metric is problematic for molecular descriptors. Two molecules with fairly close values of all critical descriptors are likely to have similar biological activity (McFarland and Gans 1986), but beyond some (unknown) threshold, there may be little relationship between distance and similarity of activity. Finally, the presence of relatively few outlying observations leads to large, dominating inter-point distances. Very often this requires removal of many molecules to generate a sensible design.

The simplest designs are based on random sampling. In fact, most new leads have been discovered through random screening, in which large numbers of compounds are tested for a specific biological activity, and the active compounds are then selected for optimization. Young et al. (1996) used a constant radius hypersphere around each randomly selected compound to measure the coverage of the descriptor space. They concluded that, unless a very large number of compounds are used to fill space, randomly selected compounds will cover as much space as carefully selected compounds. Again, however, if relatively few descriptors are important, then a rational selection should be more effective than a random design. In Section 3.6.2 we examine the coverage of designs generated by simple random sampling and stratified random sampling.

The approach proposed in this chapter is to divide all low-dimensional subspaces into small cells and attempt to find a design that has one point in every cell of every subspace, so covering every low-dimensional subspace. In Section 3.2 we describe a National Cancer Institute (NCI) database that we will use to motivate and illustrate our methodology and notation for the general case. Section 3.3 discusses a data-adaptive descriptor binning method that leads to two- and three-dimensional cells such that only a small proportion are empty with respect to the candidates. To guide the choice of the design points from the candidates, we develop a uniform cell coverage (UCC) criterion in Section 3.4, and Section 3.5 describes a fast exchange algorithm to implement it. In Section 3.6 we apply the UCC criterion to the NCI data and compare computational time and quality of coverage relative to other methods. Finally, Section 3.7 provides some conclusions and discussion of further work.

3.2 Chemical Databases and Descriptors

3.2.1 The NCI Candidate Set

We illustrate our methods with the NCI AIDS antiviral screen database (Section 1.6.3), because it is a large database in the public domain and represents a problem of practical importance. There are 29,812 NCI compounds, of which 608 compounds (roughly 2%) are active.

We use six continuous BCUT variables as descriptors. They are based on the work by Burden (1989), who found that structurally similar compounds have similar BCUT values. They tend to characterize molecular bonding patterns and atomic properties such as surface area, charge, hydrogen-bond donor and acceptor ability.

Figure 3-1 shows the univariate distributions of the six descriptors for the NCI candidate molecules. The distributions exhibit multimodality and outlying values. The pairwise plots in Figure 3-2 show that the two-dimensional projections are complex, with much empty space. Either the collection is missing chemicals or it is not possible to make compounds with certain combinations of descriptors. In more than two dimensions this problem will be even worse.

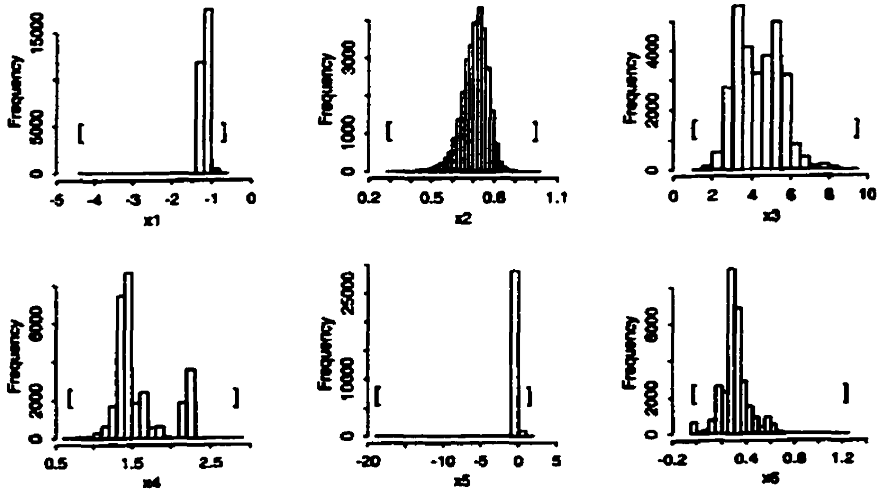


Figure 3-1. Univariate Distributions of the Descriptors in the NCI Data.

The "[" and "]" symbols denote a descriptor's range.

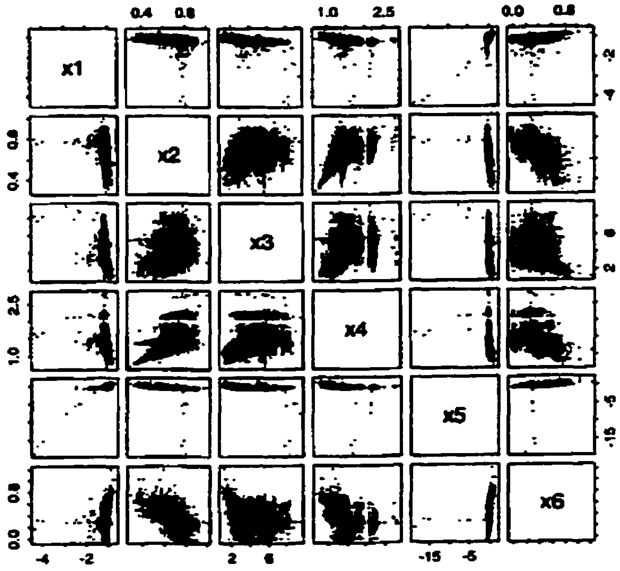


Figure 3-2. Pairwise Plots of the Descriptor Values in the NCI Data.

3.2.2 Notation

In general, denote the k continuous descriptors by x_1, x_2,..., x_k and let X_c be a candidate set of compounds with N points. The objective is to choose a representative set of n design points, X_d, to cover the descriptor space occupied by the candidate set.

Within the full k-dimensional descriptor space, a p-dimensional (p-D) subspace is defined by p of the k descriptors ($1 \leq p \leq k$). For convenience, Xi will denote the 1-D subspace involving only x_i. Similarly, Xij, $Xijl$, etc. will represent 2-D, 3-D and higher-dimensional subspaces. For example, $X1$ is a 1-D subspace defined by x_1, and $X12$ is a 2-D subspace formed by x_1 and x_2. A subspace, then, is simply a subset of the descriptor variables, ignoring the remaining descriptors.

3.3 Cell-Based Approach

We use a number of techniques to keep the cells small, yet limit their number, and to ensure that relatively few cells are empty in the candidate set. First, when we bin each descriptor, we adopt a data-driven hybrid binning method that makes bins larger towards the extremes. This avoids empty bins towards the limits of a descriptor's range, where molecules tend to be sparse. Second, we focus attention on low-dimensional subspaces, typically all 1-D, 2-D, and 3-D subspaces. By considering no more than three variables at a time, fewer cells are required to represent a subspace. Selecting a design with good coverage of all low-dimensional subspaces is analogous to a two-level fractional factorial design of Resolution IV. Such a design is a complete factorial for any subset of three or fewer variables (Box, Hunter, and Hunter 1978, p. 388) and can estimate all interaction effects if only three factors are found to be important. Third, every subspace considered has the same number of cells, avoiding the exponential increase with dimension.

3.3.1 Data-Driven Binning

For each descriptor, we first divide its range into mutually exclusive and exhaustive sub-ranges or bins (e.g., we use 729 bins in Section 3.6 for the NCI data). The bins for descriptor x_i immediately become the cells for the 1-D subspace *Xi*. For subspaces of higher dimension, cells will be formed from the bins of the descriptors forming the subspace (Section 3.3.2).

To construct bins, we use a hybrid of two simple-to-implement methods: equal width (EW) and equal frequency (EF). The EW method simply divides a descriptor's range into equal-width intervals. Alternatively, EF bins have their cut-points chosen to make the frequency of candidate molecules approximately equal in each bin.

In regions where there is a reasonable density of descriptor values, EW bins are compelling. When a molecule is chosen to represent a bin (and hence a cell), it is the size of the bin that determines the quality of coverage in the descriptor space, not the number of molecules in a bin. Another way of looking at this is that EF bins are very small where there is a high density of candidate molecules. Such regions will be over-represented in an experimental design, to the detriment of coverage in regions where candidates are sparse and bins are wide.

On the other hand, outlying or extreme descriptor values may inflate a descriptor's range, making many EW bins empty towards the extremes. This problem is compounded when we form cells in multiple dimensions (Section 3.3.2). To avoid empty bins, extreme candidates are sometimes removed from consideration (Cummins et al. 1996 and Menard et al. 1998). By definition, the EF method has candidate points in every bin and hence none are empty. Empty cells in 2-D or 3-D subspaces can still arise, but EF bins will tend to have fewer empty cells.

To combine the best features of EW and EF bins, we use a data-driven, hybrid method. EF bins are constructed for the extreme values. For example, the first percent of a descriptor's values can be placed in one bin, with a similar bin for the last one percent. EW bins are then used between these extreme bins. Thus, EW bins predominate, while the EF method for the extreme values avoids empty bins.

Figure 3-3 illustrates the advantage of this hybrid binning strategy, applying it to x_1 from the NCI data. Here, to keep the demonstration of binning and cell construction simple, we use 64 bins. (When we apply these methods to a realistic sized design in Section 3.6 we will use 729 bins.) With EW bins the frequencies shown in Figure 3-3(a) are very uneven: Of the 64 bins, 32 in the long tail to the left are empty. In contrast, all of the 64 hybrid bins shown in Figure 3-3(b) are occupied. The 62 equal-width bins in between the two 1% end bins are much narrower on the original x_1 scale. Compounds within these narrower bins are more likely to have similar activity.

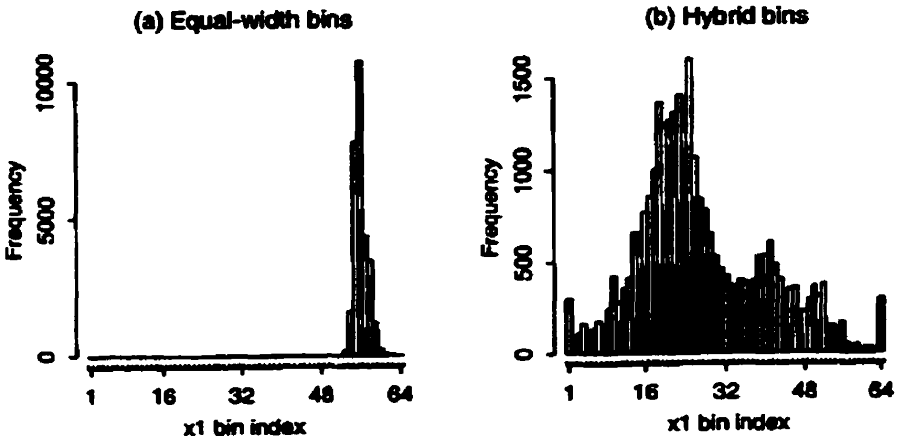


Figure 3-3. Candidate-Point Bin Frequencies for Descriptor x_1 in the NCI Data (64 Bins).

After binning, we make no further use of the raw x_i values; our uniform-coverage algorithm only uses the hybrid-bin index to characterize a molecule according to x_i. The data-adaptive binning procedure is repeated for each descriptor.

3.3.2 Forming Cells

To form cells for, say, a 2-D subspace, we combine the 1-D bins for each of its two descriptors. We keep the number of cells constant over subspaces, however, and hence avoid the curse of dimensionality. This is achieved by amalgamating 1-D bins when working in a higher-dimensional subspace. For example, if we have used 64 bins for each 1-D subspace, as in Figure 3-3(b), we divide the 2-D subspace X12 for descriptors x_1 and x_2 into 8 × 8 = 64 cells, as illustrated in Figure 3-4. The first cell, in the lower left corner, for instance, is formed from the first eight 1-D bins for X1 and the first eight 1-D bins for X2. This gives the 64 2-D cells shown. Similarly, we form 3-D subspaces of 4 x 4 x 4 = 64 cells by amalgamating 1-D bins 16 at a time for each descriptor.

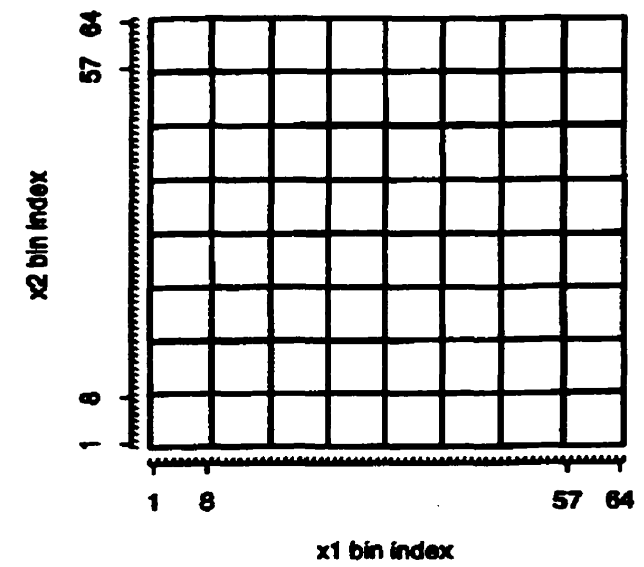


Figure 3-4. Construction of 64 2-D Cells from Descriptors with 64 Bins.

In general, suppose we consider 1-D, 2-D, and 3-D subspaces and want m cells per subspace. For 2-D subspaces, analogously to Figure 3-4, cells are formed in an $m^{1/2} \times m^{1/2}$ array, and for 3-D subspaces there is an $m^{1/3} \times m^{1/3} \times m^{1/3}$ array of cells. Thus, convenient values of m have integer square roots and cube roots: $2^{2x3}=64$, or $3^{2x3}=729$, or $4^{2x3}=4096$, etc. With k descriptors, there are

$$\binom{k}{1}+\binom{k}{2}+\binom{k}{3}=\frac{5}{6}k+\frac{1}{6}k^{3}$$

1-D, 2-D, and 3-D subspaces in total. When k=3, for example, there are 7 subspaces: $X1$, $X2$, $X3$, $X12$, $X13$, $X23$ and $X123$. When k=6, there are 41 subspaces and when k=10, there are 175 subspaces. For larger k, it might be necessary for computational reasons to reduce the number of subspaces by focusing on only 1-D and 2-D subspaces.

Including subspaces of 4-D and higher will usually not be practical. Chemists believe that two molecules must have fairly close values of all critical descriptors for similar biological activity (McFarland and Gans 1986). This means that bins have to be small if one molecule from a bin is to represent the rest. Yet, even with 10 bins per dimension, which is probably too few, there are 10,000 cells per 4-D subspace. Clearly, we would need to choose at least this many molecules if the experimental design is to cover every cell. Thus, it is not possible to give dense coverage of a 4-D subspace with a modest subset of molecules. For analysis, this implies that interaction effects are hopefully limited to no more than three factors.

How big should m be? Even with the data-driven binning method in Section 3.3.1, there will be some multi-dimensional cells with no molecules. The proportion of empty cells, which varies from subspace to subspace, will tend to increase with m. In addition, if n design points are to be selected, we would like n nonempty cells per subspace, so the space-filling design can cover distinct nonempty cells. These two considerations suggest that m should be approximately equal to n or a little larger.

3.4 Criteria for Evaluating Coverage

In a conventional cell-based design (Section 3.1), there is one set of cells based on all k descriptors. Simply picking a point from each occupied cell would guarantee a good coverage design. As already noted, however, this approach often generates many more cells than the number of design points, making good coverage impossible. In Section 3.3.2 we defined cells based on low-dimensional subspaces to overcome this problem. With more than one subspace, it is no longer straightforward to select a set of candidate points to give good coverage simultaneously in many subspaces. If, say, one point is chosen from each cell in a particular subspace, these points may be unevenly distributed in other subspaces. We now describe two measures of the quality of coverage; the second will be used in Section 3.5 as an optimization criterion to drive the numerical search for a good experimental design.

We first need some definitions and notation. Let X denote a set of points (molecules) in the descriptor space; X will typically be the entire set of candidate points, X_c, or a trial experimental design, X_d. The set X is said to *cover* cell i in subspace s if at least one of the points falls in that cell. Mathematically, we set up indicator variables $c_{si}(X)$ taking the value 1 if cell i in subspace s is covered and 0 otherwise.

3.4.1 Average Percentage of Cells Covered

The first experimental design criterion simply computes the percentage of cells that are covered by a design, averaged over all subspaces. Some cells are not covered by the candidate set, X_c and so cannot be covered by any choice of design; these cells are eliminated from consideration when computing the criterion.

In subspace s, the percentage of cells covered by a design X_d is defined to be

$$P_s = \frac{\sum_i c_{si}(X_d)}{\sum_i c_{si}(X_c)} \times 100\%,$$

where the summation is over all cells in the subspace (i.e., i=1, ... , m). We can then define the average percentage coverage over, say, all 1-D subspaces as

$$P_{1-D} = \frac{\sum_{s \in S_1} P_s}{|S_1|},$$

where S_1 is the set of all 1-D subspaces and $|S_1|$ is the number of such subspaces. For 2-D subspaces we define $P_{2\text{-D}}$ analogously, and so on.

We can then obtain the average percentage coverage, P. For example, if 1-D, 2-D, and 3-D subspaces are being considered, we have

$$P = \frac{P_{1-D} + P_{2-D} + P_{3-D}}{3}. \qquad (1)$$

The average could also be weighted, for example giving more weight to 1-D subspaces.

One deficiency of this criterion is that it ignores the distribution of design points in the covered cells. For instance, consider two very different designs: one has two points in each of 50 cells and the other has 1 point in each of 49 of these cells and 51 points in the remaining cell. With respect to these 50 cells, the coverage is 100% for both designs, yet we would prefer the first as the distribution of points is more uniform. Thus we report the criterion *P* in Section 3.6, but the selection of a design is a based on a modification that takes the uniformity of coverage into consideration.

3.4.2 Uniform Cell Coverage (UCC)

Suppose design X_d places $n_{si}(X_d)$ points in cell *i* of subspace *s*. If the candidate set X_c does not cover this cell, i.e., $c_{si}(X_c) = 0$, then $n_{si}(X_d)$ also has to be 0. For cells that are covered by X_c, i.e., $c_{si}(X_c) = 1$, we want the $n_{si}(X_d)$ counts to be approximately 1. Thus, ideally, $n_{si}(X_d) = c_{si}(X_c)$ for every cell. In subspace *s*, then, a measure of lack of uniformity is

$$U_s = \sum_i [n_{si}(X_d) - c_{si}(X_c)]^2 . \qquad (2)$$

Again, we can average these quantities over subspaces. The total lack of uniformity for 1-D subspaces, for example, is

$$U_{1-D} = \frac{\sum_{s \in S_1} U_s}{|S_1|},$$

and analogously for $U_{2\text{-D}}$, etc. Averaging with weights across, say, the 1-D, 2-D, and 3-D subspaces, we have the uniform cell coverage (UCC) criterion:

$$U = \frac{w_1 U_{1-D} + w_2 U_{2-D} + w_3 U_{3-D}}{w_1 + w_2 + w_3}, \qquad (3)$$

where w_1, w_2, and w_3 are user-supplied weights. A user might want to give more weight to 1-D coverage and least to 3-D coverage, for example. In all of the examples in Section 3.6 we use equal weights.

Minimizing *U* in (3) discourages uncovered cells in the design and tends to avoid having more than one design point per cell. This is the criterion used by the optimization algorithms of the next section.

The indicator variables $c_{si}(X_c)$ in (2) provide the target numbers of points per cell in the UCC criterion. With a simple modification to these targets, a generalized UCC is obtained. For example, suppose that the number of design points allows about two design points in each cell. We can set the target for a cell to 0, 1, or 2 if there are no candidate points, one point, or at least two points, respectively. In the examples of this chapter, we use (2) without modification.

3.5 Fast Exchange Algorithm

3.5.1 Basic Exchange Algorithm

An optimization algorithm is needed to implement the minimization of the UCC criterion in (3). In other contexts, primarily efficient experiments for fitting regression models, there are many algorithms for optimizing a design criterion; see Cook and Nachtsheim (1980) and Tobias (1995, pp. 657-728) for reviews. Most of these algorithms are variants on the basic idea of an exchange. Starting with n points in a trial design, they exchange a point in the design for one in the candidate set to improve the design criterion and iterate until the criterion cannot be improved further. However, these methods were not intended for problems of the magnitude considered here (i.e., select thousands of points from hundreds of thousands) and would be far too slow.

To derive a computationally efficient algorithm for large designs, we could modify any one of several implementations of this idea. We choose to start with the Wynn (1972) algorithm, which we call the basic exchange algorithm below, because it is fast relative to other methods (Tobias 1995, pp. 657-728) and its simplicity facilitates adaptation. The modifications greatly reduce the computational effort, especially when dealing with very large candidate sets.

The basic exchange algorithm starts with a random subset of n points (an initial design) from the N candidates. The optimization criterion is then sequentially improved by a series of exchanges. (Wynn worked with the D optimality criterion, but we will use UCC.) In each exchange, a point in the candidate set replaces a point in the current design. An exchange is broken down into two steps. First, a point in the candidate list is found to add to the current design. The point added from the candidate list is the one with the best value of the design criterion for the modified design of $n + 1$ points. Second, a point in the new design of $n + 1$ points is removed; this point is chosen to give the best criterion value for the new design of n points amongst those that are subsets of the $n + 1$ points available. These exchanges continue until the criterion cannot be improved. We now describe the adaptations to this exchange concept.

3.5.2 Identifying Good Candidates for Exchange

The basic exchange concept is computationally inefficient for large candidate lists. In principle, we have to loop through the whole candidate list, X_c, to find only one candidate to add. Moreover, many of the initial n points will have to be replaced, requiring many loops if n is moderately large. The adaptations we first describe are aimed at obtaining many exchanges per X_c loop, thereby reducing the number of X_c loops required. Every time a candidate is visited, we note the improvement in the criterion if it were added to the design. Hence, an approximation to the distribution of improvements can also be maintained. As we pass through the candidates, whenever a candidate's change is in the upper tail of this distribution, it is deemed "good" and considered for an exchange. (A similar process will be described in Section 3.5.3 to search for a design point to delete and complete the exchange.) Thus, each X_c loop might identify many "good" candidates and carry out several exchanges.

Specifically, let δ_j denote the improvement (i.e., reduction) in the UCC criterion U in (3) if candidate j were added to the current n design points to give $n + 1$ points. The algorithm for identifying good candidates, with some explanation in parentheses, is as follows:

1. Initialize the δ distribution. Randomly select 100 candidate points. Compute their δ values, and denote the sorted values by $\delta_{(1)} \geq \ldots \geq \delta_{(100)}$. Set λ=n/N and $\delta^* = \delta_{(q)}$, where $q = \max(1, 100\lambda)$. (In Step 2, if candidate j has $\delta_j \geq \delta^*$, it will be considered for an exchange. This rule will try approximately n of the N candidates during the first X_c loop, because all n initial design points may have to be replaced.)

2. Loop through the candidates. For j=1,..., N do the following steps:
   - Compute δ_j and note the value for later use in updating δ^*.
   - If $\delta_j \geq \delta^*$, then:
     - Try exchanging candidate j with one of the current design points (see Section 3.5.3).
     - If candidate j was exchanged, then
     - Set δ_j = -100. (As candidate j is now in the design, introducing it again is undesirable.)

   else
     - Replace δ^* by $\delta^* + 10\lambda$. (A failed exchange suggests that δ^* is allowing poor candidates to be considered, i.e., δ^* is too small.)

3. If there was no improvement in the criterion in the last X_c loop, then stop.

4. Update δ^* for the next X_c loop. Sort the δ_j values from the last X_c loop and denote them by $\delta_{(1)} \geq \ldots \geq \delta_{(N)}$. Set λ to half the previous value and $\delta^* = \delta_{(q)}$, where $q = \max(10, N\lambda)$. Go to Step 2. (Decreasing λ reduces the number of exchanges considered, because fewer exchanges are likely to improve the criterion with successive passes through the list. We always want to consider at least 10 promising candidates in the next X_c loop, however, to be conservative about termination.)

Note that when a good candidate is found in Step 2, we do not re-start the X_c loop at the beginning. Rather we continue with the next candidate. These "floating" loops allow many exchanges in one X_c loop.

3.5.3 Identifying Design Points for Exchange

Whenever a "good" candidate for inclusion in the design is identified by the rules in Section 3.5.2, a design point must also be removed if an exchange is to take place. We evaluate the design points and identify a "bad" point, i.e., one that should be removed, using similar rules.

Specifically, for a fixed candidate j under consideration for inclusion, let Δ_i denote the overall improvement in the UCC criterion in Equation (3) if design point i of the n current design points is replaced by candidate j. Thus, Δ_i includes the δ_j contribution from adding candidate j. A distribution of Δ_i values is maintained, and we implement an exchange as soon as a "good" Δ_i value is found, rather than search all n design points. The details are as follows:

1. Initialization of the Δ distribution. If this is the first search of the design list, then:
   - Randomly select 100 design points, compute their Δ_i values, and denote the sorted values by $\Delta_{(1)} \geq \ldots \geq \Delta_{(100)}$.

- Set $\Delta^* = \max(0.01, \Delta_{(q)})$, where $q = \max(1, 100\lambda)$, using the λ value in effect for searching the candidate list. (Exchanges with $\Delta_i \geq \Delta^*$ will be implemented.)
- Set i=1. (Start at the top of the design-point list.)

2. Compute Δ_i and note the value for later use in updating Δ^*.
3. If $\Delta_i \geq \Delta^*$, then
   - Implement the exchange of design point i with candidate j.

   else if all design points have been tried, then
   - Let $\Delta_{\max}$ be the maximum Δ value over all the design points. If $\Delta_{\max} \geq 0$, then
     - Implement the exchange of the design point giving $\Delta_{\max}$ with candidate j.
4. If i=n, then
   - Update Δ^*. Sort the Δ_i values from the last X_d loop and denote them by $\Delta_{(1)} \geq \ldots \geq \Delta_{(n)}$. Set $\Delta^* = \max(0.01, \Delta_{(q)})$, where $q = \max(1, n\lambda)$, using the λ value in effect for searching the candidate list.
   - Set i=1;

   else
   - Set i to i+1.
5. If an exchange occurred in step 3 or all design points had been tried in step 3, then
   - Return to searching for the next "good" candidate to add. The next search for a "bad" design point to remove will start at Step 2 with the current value of i.

   else
   - Go to Step 2.

Note that in Step 3, an exchange can occur with $\Delta_{\max} = 0$, i.e., it does not change the criterion. Allowing "neutral" exchanges of this type may be useful to break away from a design that is only locally optimal.

3.5.4 Updating the UCC criterion

Finally, we describe how the criterion can be efficiently updated when only one point is changed, either when adding a candidate or when removing a design point.

When a point is added to or removed from the design, it will affect only one of the m cells in each subspace. Let z_s be the number of design points in the affected cell in subspace s. If we are adding a point, then z_s becomes $z_s + 1$, and the change to U_s in (2) is

$$[(z_s+1)-1]^2-(z_s-1)^2=2z_s-1.$$

Note that $c_{si}(X_c)$ in (2) must equal 1, as a cell must be covered by at least one candidate if a point is to be added (or removed). Similarly, when a design point is removed, the change to U_s in (2) is

$$[(z_s-1)-1]^2-(z_s-1)^2=3-2z_s.$$

3.6 Results

We now apply our data-driven binning method and our fast design algorithm to select 729 molecules from the 29,812 NCI molecules. One hundred uniform cell coverage (UCC) designs are derived, starting from 100 different initial designs based on simple random sampling.

3.6.1 Forming Cells

The distributions of the NCI molecules in 1-D and 2-D projections for all six descriptors are shown in Figure 3-1 and Figure 3-2. To apply the hybrid binning method described in Section 3.3.1, the first and last percentiles are assigned to EF bins, with EW bins between. There are six 1-D, 15 2-D, and 20 3-D subspaces, and each of these 41 subspaces is divided into 729 cells. Over the 41 subspaces, on average there are 81.4% nonempty cells in the candidate set of molecules; the worst subspace is X246 with 63.0% nonempty cells. Figure 3-5 shows the bin (1-D cell) counts for the 1-D subspaces. The plot for x_4 shows that adding a few extra EF bins in sparse regions could further increase the proportion of nonempty bins, but we do not pursue this as the proportion of nonempty bins is already high (588 bins out of 729 are occupied by at least one candidate). For 2-D subspaces, bins are amalgamated 27 at a time to generate 27 × 27 = 729 cells. Figure 3-6 depicts cells with at least one candidate point with a filled-in square. It is seen that the 2-D cells are fairly well covered by the candidates. Similarly, the 3-D subspaces (not shown) have 9 x 9 x 9 cells formed by amalgamating 81 bins at a time in each dimension and are fairly well covered by the candidates.

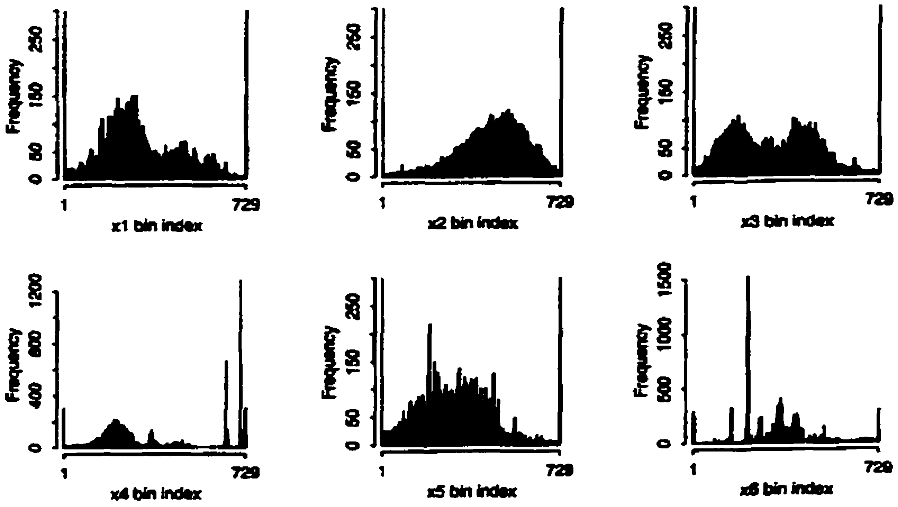


Figure 3-5. Candidate-Point Bin Frequencies for the NCI Data (729 Hybrid Bins).

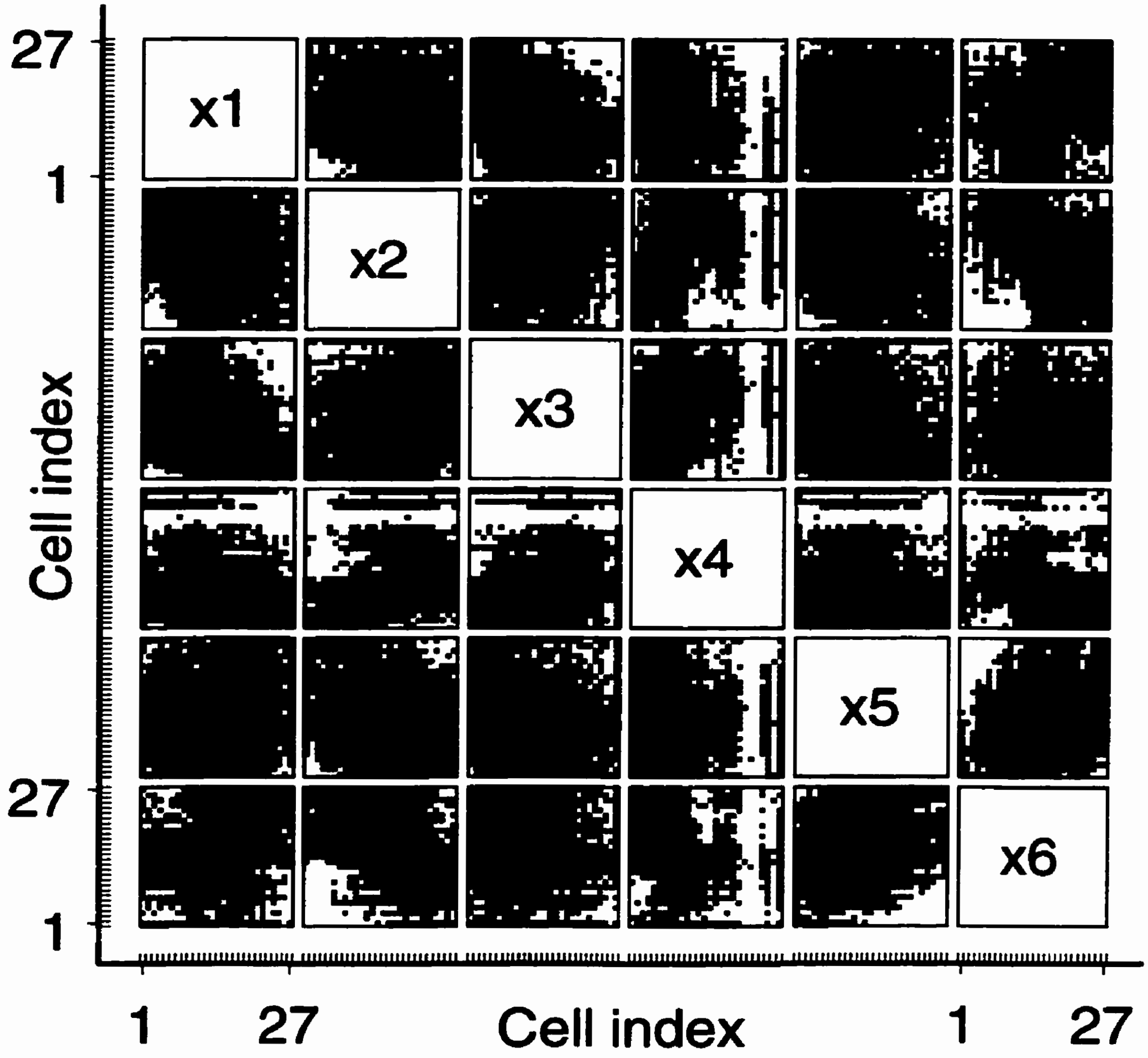


Figure 3-6. Candidate-Point 2-D Coverage for the NCI Data (729 Cells).

A filled-in square in a cell denotes at least one candidate point.

3.6.2 UCC Optimization Algorithm Versus Random Designs

The algorithm in Section 3.5 to minimize the UCC criterion in (3) gives, on average, a *U* value of 585. For comparison, we also generate 100 designs based on simple random sampling (SRS) of 729 points from the 29,812 candidates and compute their values of the UCC criterion. Figure 3-7 shows that the distribution of the *U* values given by the UCC optimization algorithm compares very favorably with the distribution of values under SRS. Because of the small range of the *U* values under UCC, the density looks like a straight line and is better displayed in Figure 3-8 where the *U* values are expanded around the small range. As a further comparison, we use stratified simple random sampling (StratRS) to choose another 100 designs. Following conventional cell-based approaches, the entire 6-D space is divided into $3^6 = 729$ cells, one design point is randomly selected from each nonempty cell, and then further points are randomly chosen to reach 729 points. The distribution of *U* values under StratRS also depicted in Figure 3-7 indicates that StratRS is preferable to SRS according to the UCC criterion, but the UCC optimization algorithm still performs considerably better. The SRS and StratRS distributions demonstrate that the simple strategy of randomly sampling many designs and choosing the best according to the UCC criterion is a poor substitute for the optimization algorithm. That is, even if we generate many random designs, spending the same computer time as required for generating a UCC design, the best random design is not expected to be competitive with respect to the UCC criterion.

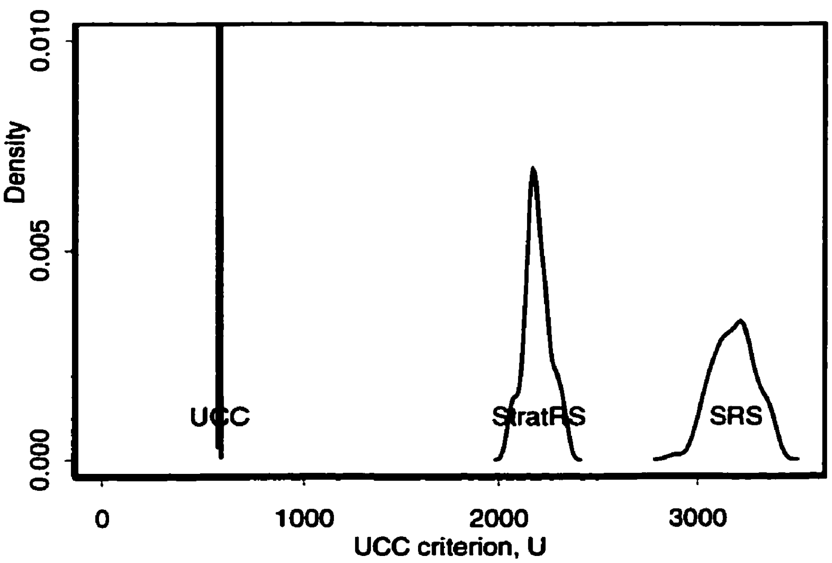


Figure 3-7. Distribution of UCC Values for 100 UCC Designs, 100 Simple Random Samples (SRS) and 100 Stratified Random Samples (StratRS).

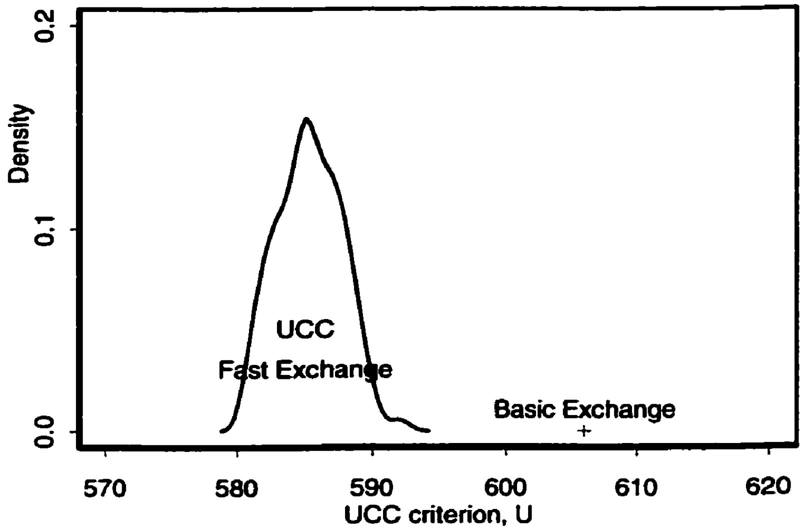


Figure 3-8. Distributions of UCC Values for 100 UCC Designs (Fast Exchange) and the Value Obtained by the Basic Exchange Algorithm (+).

Table 3-1 gives some numerical summaries of these comparisons. For UCC (fast exchange), SRS and StratRS, the numbers given are means across the 100 designs. We report the UCC criterion U in (3), the percent coverage criterion P in (2), and the 1-D, 2-D, and 3-D contributions to these two criteria. Note that U and its components are smaller the better measures, whereas P and its components are larger the better. In all cases, the designs produced by the UCC optimization algorithm perform best. For example, P is 75% on average for the designs from our algorithm versus 53% on average under StratRS. Note also that $U_{3\text{-D}}$ is the largest contributor to U, probably because there are slightly more empty cells in the 3-D subspaces.

Table 3-1. Coverage Criteria and Run Times for Various Designs

Design	UCC Criterion, U ($U_{1\text{-D}}$ / $U_{2\text{-D}}$ / $U_{3\text{-D}}$)	Average Percent Coverage, P ($P_{1\text{-D}}$ / $P_{2\text{-D}}$ / $P_{3\text{-D}}$)	Time (hh:mm)
UCC (Fast exchange algorithm) (mean over 100 designs)	585 (457/ 608/ 691)	75.1 (75.2/ 74.2/ 76.0)	0:34
Simple Random Sampling (mean over 100 designs)	3188 (2035/ 2937/ 4592)	45.5 (49.9/ 44.8/ 41.9)	0:01
Stratified Random Sampling (mean over 100 designs)	2193 (1979/ 2050/ 2551)	53.4 (53.4/ 52.5/ 54.4)	0:02
UCC (Basic exchange algorithm)	606 (489/ 615/ 714)	74.5 (73.8/ 74.2/ 75.5)	12:19
SAS PROC OPTEX Spread Design (Sequential)	3839 (7122/2365/2031)	59.4 (55.1/ 59.0/ 64.1)	2:32
SAS PROC OPTEX Uniform Coverage Design (Sequential)	3556 (2255/ 3107/ 5306)	45.4 (50.4/ 45.1/ 40.8)	133:29

For completeness, we also evaluate the high-dimensional (6-D) coverage for these designs. As these designs are associated with different design measures (e.g., "average distance from each candidate to the nearest design point" versus "average distance from each design point to the nearest other design point"), a particular measure may favor one design relative to the others. For simplicity and consistency, the evaluation is done by dividing the entire 6-D space into 729 cells (as for the stratified simple random sampling) and measuring the corresponding 6-D percent coverage and uniform coverage for the various designs. Following the uniformity measure of equation (2), a measure of lack of uniformity for the entire space is then

$$U_{6-D} = \sum_i [n_i(X_d) - c_i(X_c)]^2 ,$$

where $n_i(X_d)$ is the number of design points in cell i and $c_i(X_c) = 1$ or 0 if the candidate set X_c does or does not cover this cell. Table 3-2 gives some numerical summaries of these comparisons. By definition, the designs produced by StratRS have 100% coverage. The designs produced by UCC (fast exchange) have the next best percent coverage and the best uniform coverage among all designs. This suggests that although the UCC criterion focuses on low-dimensional coverage, it can generate designs with both good low-dimensional coverage and good high-dimensional coverage.

Table 3-2. High-Dimensional Coverage for Various Designs

Design	Uniform Coverage, $U_{6\text{-D}}$	Percent Coverage
UCC (Fast exchange algorithm) (mean over 100 designs)	1370	70.2
Simple Random Sampling (mean over 100 designs)	5416	40.8
Stratified Random Sampling (mean over 100 designs)	1687	100.0
UCC (Basic exchange algorithm)	1478	69.9
SAS PROC OPTEX Spread Design (Sequential)	2494	68.6
SAS PROC OPTEX Uniform Coverage Design (Sequential)	5303	37.9

Figure 3-9 compares the UCC design with the first StratRS design in terms of cell frequencies in 1-D projections. Descriptor x_1's candidates are fairly well behaved after hybrid binning; in contrast x_4's distribution is more difficult to handle. In both cases the UCC design is seen to have a much more uniform distribution of design points. For all descriptors, the UCC design has one or two design points in most cells. Some analogous 2-D projections of the design points are shown in Figure 3-10, where a filled-in square is plotted in a cell if there is at least one design point. It is clear that the UCC design has superior coverage of 2-D cells. In contrast, the cells that have at least one candidate but no design point are shown in Figure 3-11. Obviously, the UCC design has much less uncovered cells. The uncovered cells can be further reduced by adding more design points. Similar plots for the 3-D projections show the same pattern.

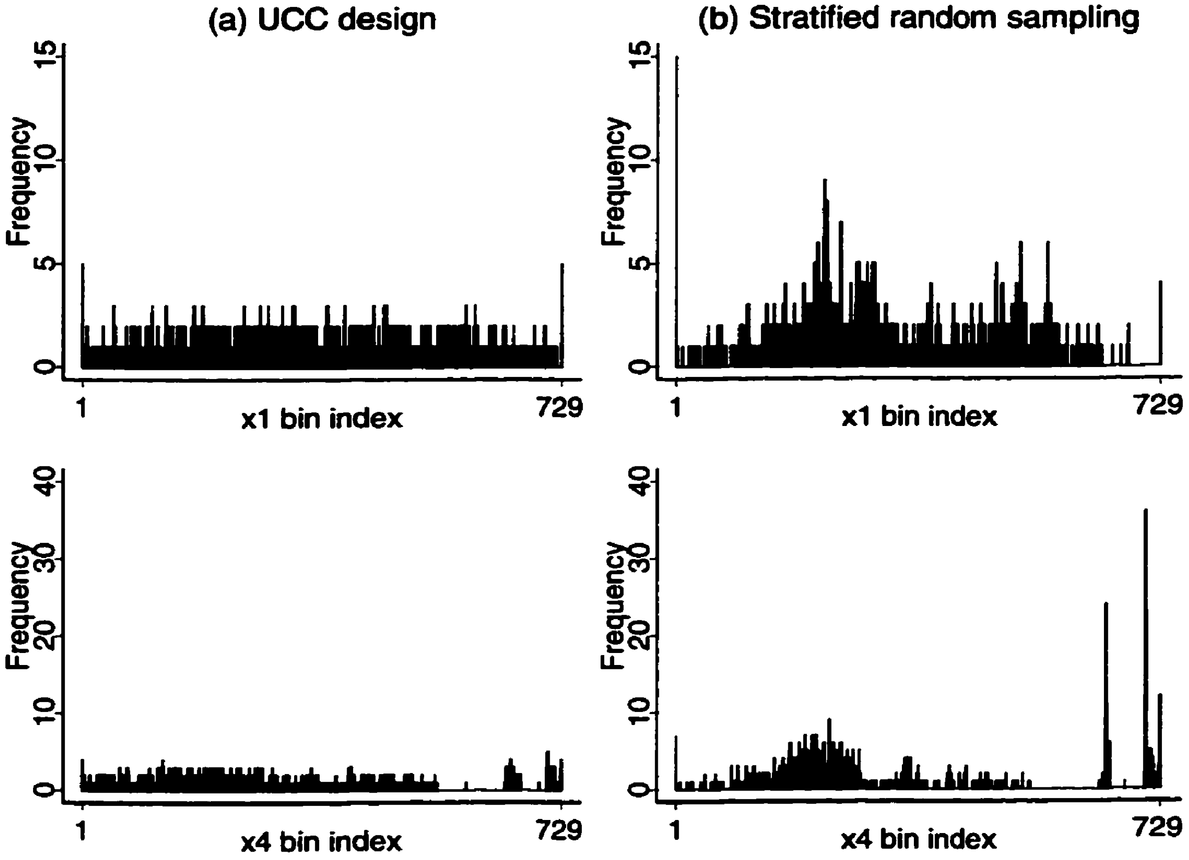


Figure 3-9. Design-Point Bin Frequencies for the NCI Data (729 Points in 729 Hybrid Bins).

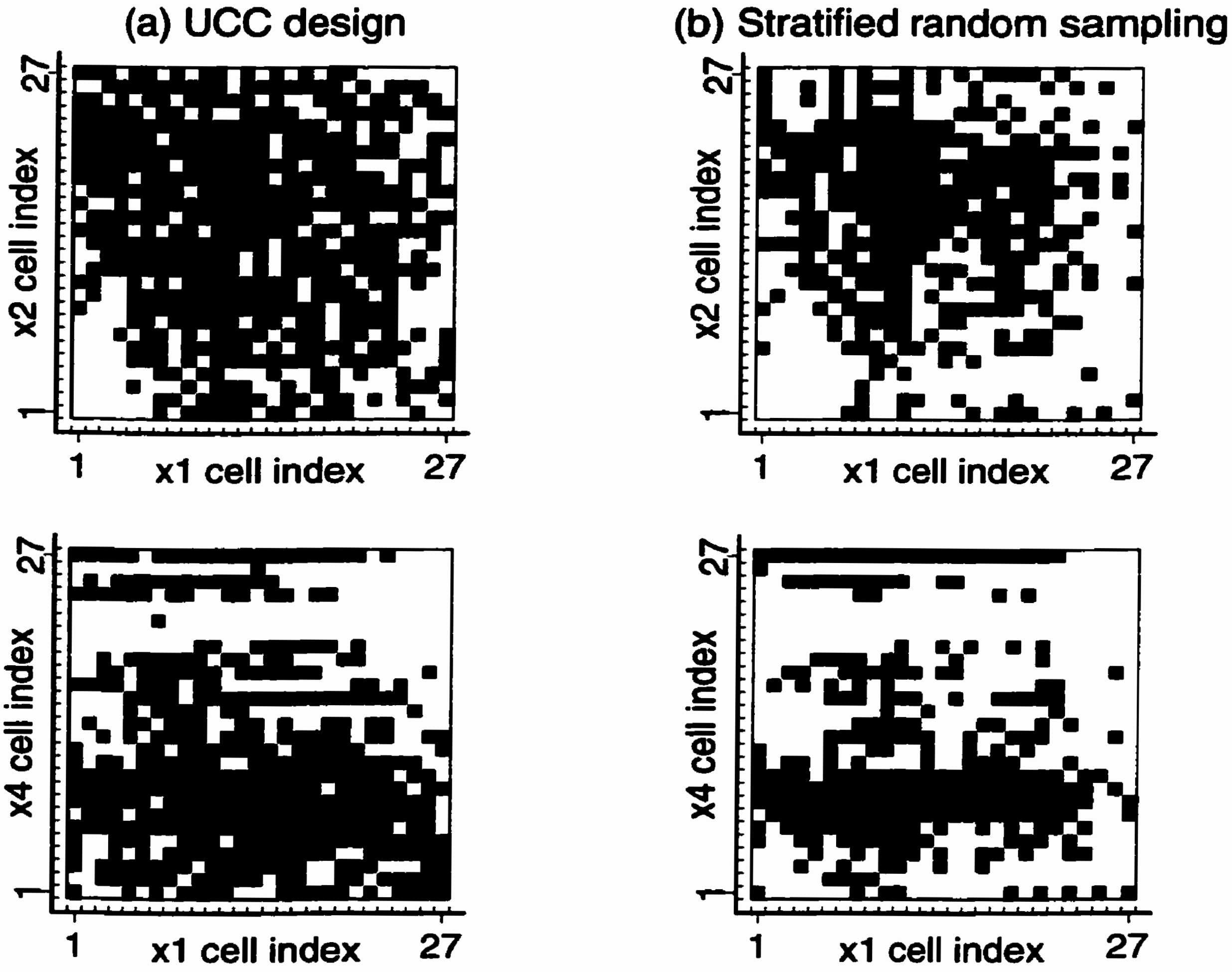


Figure 3-10. Design-Point 2-D Coverage for the NCI Data (729 Cells).

A filled-in square in a cell denotes at least one design point.

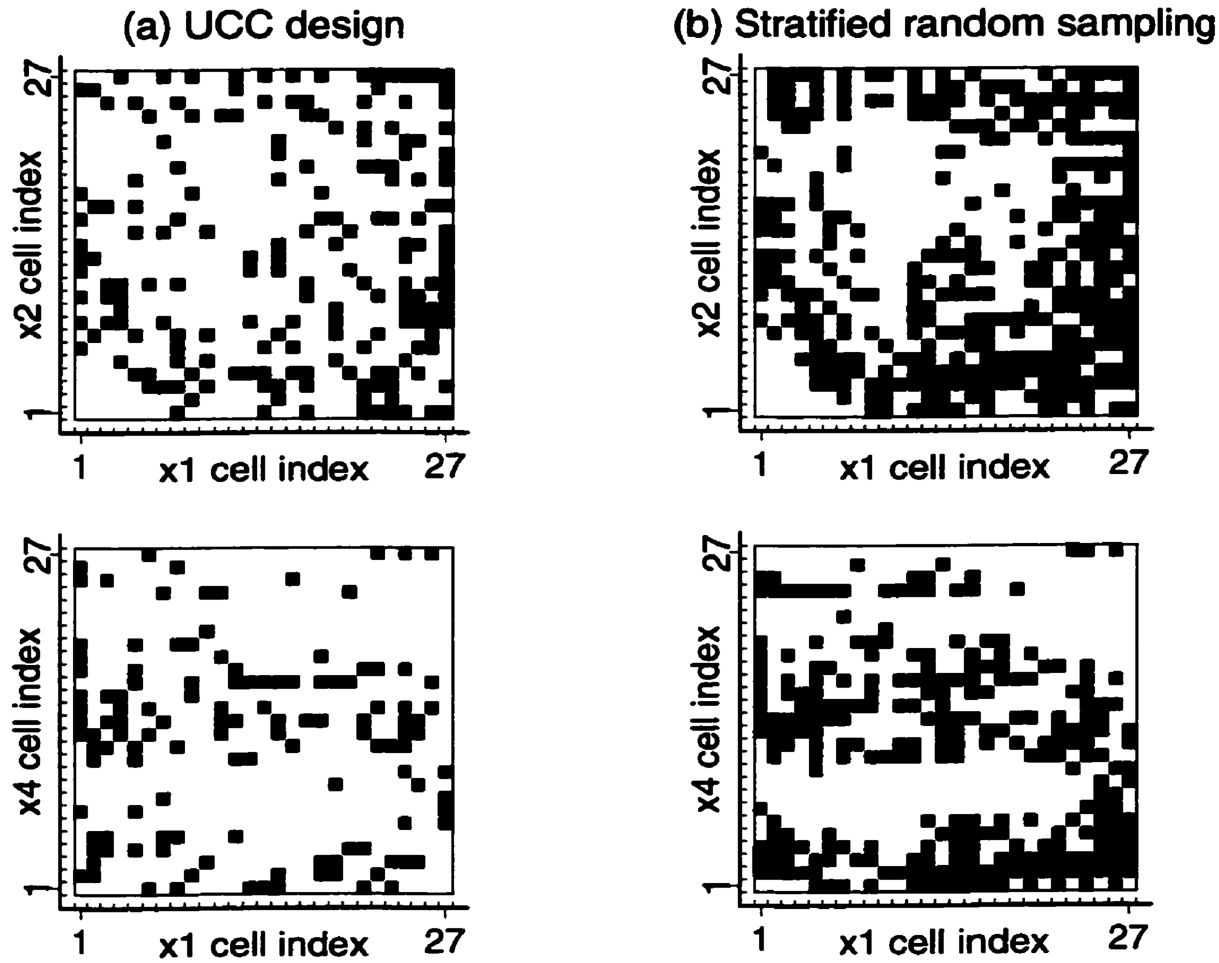


Figure 3-11. Cells Not Covered by a Design Point.

A filled-in square denotes cells with at least one candidate point but no point was chosen by the design.

In Table 3-3, the 100 designs are compared two at a time and the number of common compounds is reported. For 100 designs, there are $\binom{100}{2}$ possible pairwise comparisons. On average, there are 356 common compounds between two UCC designs whereas there are only 18 common compounds between two random designs based on simple random sampling. The many common compounds suggest that the UCC optimization algorithm led to similar uniform coverage designs even with very different starting designs.

Table 3-3. Number of Common Compounds Out of 729 When Two Designs Are Compared (4,950 Pairwise Comparisons Between 100 Designs)

100 random samples of 729 compounds from 29,812 candidates are chosen. These random samples serve as the initial starting points for the uniform cell coverage (UCC) designs. The number of identical compounds appear in two designs is summarized.

	Number of Common Compounds Out of 729	
	Random Selection	UCC Designs
Mean	18.1	355.7
Median	18	356
SD	4.1	16.0
Minimum	7	298
Maximum	34	411

3.6.3 Comparison With Basic Exchange

Figure 3-8 shows that the distribution of *U* values given by the fast exchange UCC algorithm compares very favorably with the *U* value under the basic exchange algorithm. Compared with the basic exchange algorithm, our algorithm makes about 2.5 times, on average, as many exchanges (1800 versus 754). It achieves this even though the number of passes through the candidate points is reduced by a factor of about 40 (19.8 passes versus 754), see Table 3-4. The reduction in iterations through the candidates leads to a run time of 34 minutes on average, whereas the basic exchange algorithm takes more than 12 hours (with the fast UCC update described in Section 3.5.4). These times relate to implementations in SAS PROC IML on a Pentium III 550MHz computer with 256MB RAM. Some preliminary runs with a C++ implementation indicate that the modified algorithm runs in less than a minute for problems of this magnitude.

Table 3-4. Number of Loops Required by UCC Optimization Algorithm (100 Tries From 100 Random Starting Designs)

Mean	Median	SD	Minimum	Maximum
19.8	19.1	3.9	13.8	32.4

In Table 3-1 we see that our fast exchange algorithm produces slightly better values of *P* and *U* here than does the basic exchange method. The increase in number of exchanges, including neutral exchanges, improves the ability to escape from local minima. Similar results were obtained with another database from GlaxoSmithKline.

Our fast exchange algorithm makes most of the improvement in the first few passes through the candidates. After only 10 passes through the candidate list, all 100 designs had already achieved better *U* values than that of the final design from the basic exchange algorithm. The percentage of total improvement in *U* value after 5, 10, 15, 20, 25 and 30 passes through the candidate list for the 100 UCC designs is summarized in Table 3-5.

Table 3-5. Percentage of Total Improvement Achieved After 5, 10, 15, 20, 25 and 30 Loops (100 Tries From 100 Random Starting Designs)

The fast exchange algorithm allows hundreds of exchanges for each pass through the candidate list (one loop). This enables large improvement in coverage within several loops.

	Percentage of Total Improvement				
Loop	Mean	Median	SD	Minimum	Maximum
5	98.81	99.28	1.688	88.43	99.74
10	99.87	99.90	0.104	99.24	99.98
15	99.97	99.99	0.035	99.87	100.00
20	99.99	100.00	0.015	99.92	100.00
25	100.00	100.00	0.003	99.98	100.00
30	100.00	100.00	0.000	100.00	100.00

3.6.4 Comparison With SAS PROC OPTEX

We also make comparisons with PROC OPTEX in SAS. There are two difficulties. First, our criteria focus on low-dimensional coverage whereas PROC OPTEX computes spread and coverage measures using all descriptors. Therefore, the *U* and *P* values reported in Table 3-1 for PROC OPTEX are unfavorable. Secondly, problems of this magnitude (729 points selected from 29,812) require substantial computing time. Even when design points are optimized one at a time in the sequential option, PROC OPTEX with the uniform coverage criterion requires over five days.

3.7 Conclusions and Discussion

Our design problem is somewhat special for the following reasons. Design points cannot be placed anywhere, because only certain compounds are available or can be made. Moreover, the set of discrete points can be large and highly irregular (see, for example, Figure 3-2). To have similar properties, it is believed that two compounds must have very similar values of all critical descriptors. Thus, the design needs to cover the space densely. It is clearly impossible to achieve dense coverage in more than three dimensions at a time without an extraordinarily large design. Hence, we have proposed designs that aim for uniform coverage in all 1-D, 2-D, and 3-D projections.

The aim of such experimental designs is not just to discover highly active compounds but to find several structurally different chemical classes. These provide options for further optimization of activity, physical properties, distribution, half-life, toxicity, etc. By covering the descriptor space uniformly, there is more chance of discovering multiple classes.

The design algorithm proposed here can efficiently deal with tens of thousands of compounds in the candidate set. Much larger sets of compounds will be of interest as technology advances. We are currently working to implement the algorithm with multiple processors, for example.

An open question is how to analyze the data resulting from very large designs. Current practice often simply ranks the compounds by potency and selects the few top-ranking compounds for further development. One challenge in statistical modeling is that the potent molecules are likely to be acting in several different ways: Different descriptors might be critical for the various mechanisms. A single mathematical model is unlikely to work well for all mechanisms. There has been some success using partitioning methods on these problems (e.g., Hawkins et al. 1997, King et al. 1992, and Klopman 1984). In a multi-stage design strategy, the initial design should cover the descriptor space as uniformly as possible. Analysis of the resulting data would be used to directing subsequent designs to subregions of high activity in critical descriptor projections.

Chapter 4
Cell-Based Analysis

4.1 Introduction

In screening for drug discovery, thousands to hundreds of thousands of chemical compounds are screened in the hope of discovering biologically active compounds. The evaluation of a single compound can cost from a few cents to several dollars depending upon the complexity of the assay. At the next stage of drug development, the active compounds or "hits" found by screening are typically modified atom-by-atom to improve activity and other important characteristics, such as tissue distribution, plasma half-life, toxicity, etc. The aim of the initial screen, then, is to find active compounds of several structurally different chemical classes, to provide a variety of starting points for subsequent optimization.

In addition to finding active compounds among those screened, it would be very useful to know how to find additional active compounds without having to screen each compound individually. We might initially screen part of a collection and use these data to predict which compounds in the remainder of the collection are likely to be active. Several cycles of screening are expected to be more efficient than screening all the compounds in a large collection (Jones-Hertzog et al. 2000). To do this we need to analyze the initial high throughput screening (HTS) data to find association rules linking biological activity (response variable) to specific values of the compound descriptors (explanatory variables).

The first step in the process of determining features of compounds that are important for biological activity is describing the molecules in a relevant, quantitative manner. A drug-like molecule is a small three-dimensional object that is often drawn as a two-dimensional structure. This two dimensional graph is subject to mathematical analysis and can give rise to numerical descriptors to characterize the molecule. Molecular weight is one such descriptor. There are many more. Ideally, the descriptors will contain relevant information and be few in number so that the subsequent analysis will not be too complex. To exemplify our methods we use a system of 67 BCUT descriptors (Section 1.6.1).

The relationship between descriptors and activity is extremely complex for HTS screening data, and there are several challenges in statistical modeling. First, the potent compounds of different chemical classes may be acting in different ways. Different mechanisms might require different sets of descriptors within particular regions (of the descriptor space) to operate, and a single mathematical model is unlikely to work well for all mechanisms. Also, activity may be high for only very localized regions. Second, even though a design or screen may include thousands of compounds, it will usually have relatively few active compounds. The scarcity of active compounds makes identifying these small regions difficult. Third, there are many descriptors (i.e., curse of dimensionality) and they are often highly correlated. This is the case for BCUT numbers. Fourth, many HTS data sets have substantial measurement error. Because of some or all of these complexities, common statistical analysis methods such as linear regression models, generalized additive models, and neural nets are ineffective in handling these analysis problems (Young and Hawkins, 1998) and tend to give low accuracy in classifying molecules as active.

The rest of the chapter is organized as follows. In Section 4.2 we describe two motivating data sets. Section 4.3 expands on the difficulties that current methods face with complex structure-activity relationships. In Section 4.4 we present a cell-based analysis method that overcomes these problems. It divides a high-dimensional (descriptor) space into many small, low-dimensional cells, scores cells according to the activities of their compounds, and uses the scores to prioritize further compounds for screening. This analysis method is highly related to the uniform cell coverage approach described by Lam et al. (2001) for selecting molecules for screening. Thus, the earlier work and the current article together provide an overall strategy for design and analysis of HTS data. In Section 4.5 we evaluate our analysis approach on the two data sets and show that it can improve prediction accuracy compared with recursive partitioning (trees), one of the few successful methods for HTS structure-activity data. Finally, Section 4.6 makes some conclusions and discusses further work.

4.2 Motivating Applications

The new method described here can be applied to both continuous and discrete responses. For illustration, a data set with continuous activity outcome (Core98) and a data set with binary activity outcome (NCI) are included.

4.2.1 Core98 Molecular Data (Continuous Response)

Core98 is a chemical data set from the GlaxoSmithKline collection (Section 1.6.2). Activity is available for 23,056 compounds. The response is % Inhibition for a given biological target and theoretically should range from 0 to 100%, with more potent compounds having higher scores. Biological and assay variations can give rise to observations outside the 0-100% range. Typically, only about 0.5% to 2% of screened compounds are rated as potent.

4.2.2 NCI Molecular Data (Binary Response)

NCI is a chemical data set from the National Cancer Institute AIDS antiviral screen database (Section 1.6.3). Unlike the Core98 data where the response is continuous, the NCI compounds are classified as moderately active, confirmed active, or inactive. We combine the first two categories into a single active class to give binary response data, as there are only 608 (roughly 2% of 29,812 compounds) active compounds.

4.2.3 Descriptor Variables

For both data sets we use BCUT descriptors based on the work of Burden (1989) to describe the compounds. The BCUT descriptors are eigenvalues from connectivity matrices derived from the molecular graph. The square connectivity matrix for a compound has a diagonal element for each heavy (non-hydrogen) atom. The diagonal values are atomic properties such as size, atomic number, charge, etc. Off diagonal elements measure the degree of connectivity between two heavy atoms. Since eigenvalues are matrix invariants, these numbers measure properties of the molecular graph and hence the molecule.

When we first started this research, only six BCUT descriptors were available to us. They were used in development of a uniform coverage design method (Lam et al., 2001). Subsequently, GlaxoSmithKline computational chemists also provided a larger set of 67 descriptors for the motivating applications. The larger set was suggested by Pearlman and Smith (1998). We found that the 67 BCUT descriptors are highly correlated in the two data sets. A reason for the high correlations is that scientists often devise descriptors that measure the same general property of a compound.

While our software for the cell-based analysis method can handle 67 descriptors, the computational time is much larger. For example, it takes roughly 100 hours versus 5 minutes for 67 versus 6 descriptors. Thus, we primarily use the smaller set in this chapter. The current software (written in SAS code) was aimed at testing the new methods and did not focus on efficiency in dealing with large data sets with many variables. We plan to implement the cell-based analysis algorithm using C++ code, which should run hundreds of times faster than the current software. Whether the larger set of descriptors has substantially more predictive power is a question of some interest to the computational chemists, however, and we make some comparisons in Section 4.5.

4.2.4 Dividing Data into Training and Validation Sets

For the purpose of demonstrating the validity of the new methods, we divide each of the original data sets into training and validation sets. We use the training data (treated as screened compounds) to build models (i.e., find active regions) and the validation data (treated as unscreened compounds) to evaluate prediction accuracy (i.e. verify if the activity in these regions remains high). The validation set gives a more unbiased evaluation of the statistical method than the training set. In real applications we would use all the assayed compounds to find active regions, as more data increases the prediction power.

There are 608 active compounds (roughly 2%) in the NCI data set. This population or random hit rate of 2% gives us a benchmark for the performance of our analysis method. If an analysis method gives hit rates (proportion of active compounds amongst those selected) in the validation set many times higher than the random hit rate, then it performs well

For the Core98 compounds, the activity response variable is on a continuous scale. The mean, standard deviation, and median of the measured activities are 7.8, 8.9, and 5.9%, respectively. We refer to the mean activity as the population or random activity value. As well as analyzing the data on this scale we can also classify the compounds with the top 1% of measured activities as active. This 1% random hit rate corresponds to 34.8% inhibition on the continuous scale. The population mean activity of 7.8% inhibition (continuous response) or the population active hit rate of 1% (binary response) again provide benchmarks for the analysis methodology.

We will use relatively small training sets, as one of our goals is to predict from a small screening design. The training molecules will be selected either using the Lam et al. (2001) uniform-coverage design algorithm or at random. With a 1-2% hit rate, a sample size of 4096 compounds gives roughly 40-80 active compounds, which should be sufficient to build a sound prediction model. (A sample size of 4096 is a convenient number for the design algorithm.) Table 4-1 shows the expected division of active compounds between the training and validation sets for the NCI data and for the Core98 data (binary response).

Table 4-1. Expected Distribution of Active Compounds Between a Training Set of 4096 Compounds and a Validation Set of the Remaining Compounds For Random Designs

Data set	All data # actives / # compounds	Training set # actives / # compounds	Validation set # actives / # compounds
NCI	608 / 29 812	84 / 4 096	524 / 25 716
Core98	231 / 23 056	41 / 4 096	190 / 18 960

4.3 Existing Methods

Here we describe two statistical analysis methods commonly used for analyzing chemical data sets.

4.3.1 Cluster Significance Analysis

Cluster significance analysis (CSA), introduced by McFarland and Gans (1986), aims to find embedded regions of activity in a high dimensional chemical space. CSA considers every subspace that can be formed by the predictors, from all one-dimensional subspaces up to the space of all predictors. A subspace is simply a subset of the descriptor variables, ignoring the rest. For each subspace, CSA computes the average distance between the active compounds and compares the average to the distribution of average distance for an equal number of compounds randomly selected from all compounds (active or inactive). If the actives are clustered tightly, as measured by a randomization significance test, this is evidence that the descriptors forming the subspace and the regions where the actives are clustered are important for activity.

A synthetic data set is instructive of the method and the potential problems. CSA tacitly assumes that there is only one class of active compounds forming one cluster in one or more subspaces. Suppose, however, that there are two mechanisms operating. (In practice, we would not necessarily know which mechanism is causing activity, nor even how many there are.) Mechanism M1 active compounds require that the descriptor molecular weight is between 400 and 500 and that the melting point is between 160 and 205 degrees C. These active compounds are denoted by squares in Figure 4-1(a). Mechanism M2 active compounds require that the descriptor LogP (the octanol and water partition coefficient) is in the range 3.0-4.0; they are shown by circles in Figure 4-1(a). Dots represent inactive compounds. Because molecular weight and melting point are unimportant for mechanism M2, the circles are spread throughout the subspace, making it difficult to detect clustering of the actives. Similarly, if we look at a subspace that includes LogP, as in Figure 4-1(b), the M1 actives are spread across the LogP dimension. Even in this somewhat simple situation, the CSA algorithm could have trouble. Similarly, if there are two or more active regions in a single subspace, in principle, a single measure of clustering might not detect them.

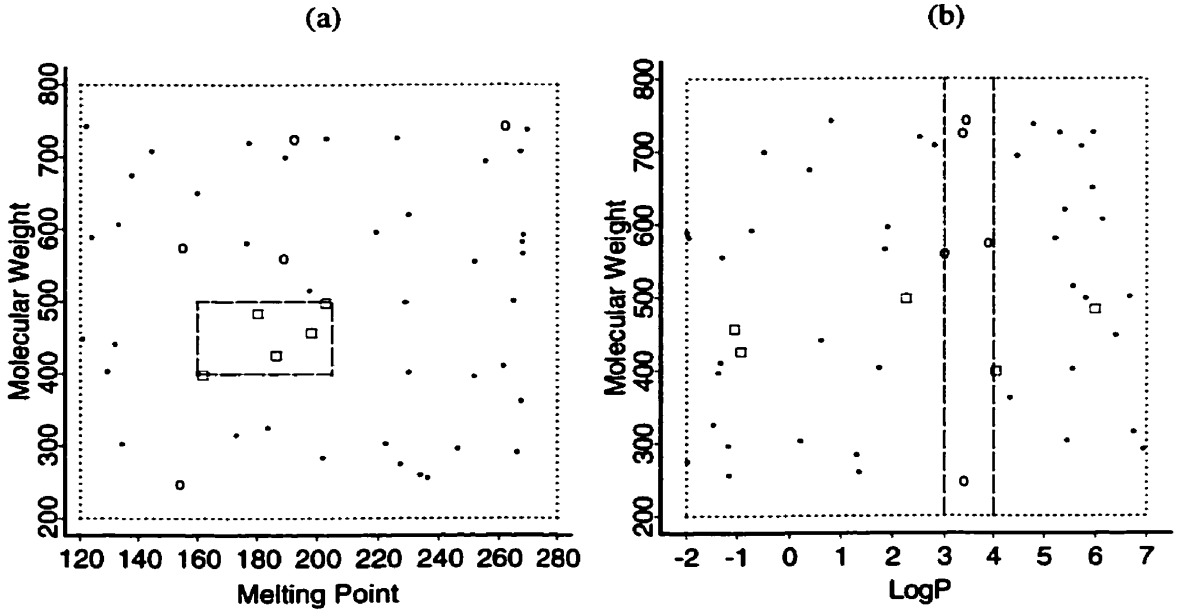


Figure 4-1. Distributions of Active Compounds from Two Mechanisms.

Squares and circles represent compounds active via Mechanisms 1 and 2, respectively, while dots are inactive compounds. Active regions corresponding to these mechanisms are shown by dashed lines: (a) locations of compounds in the subspace formed by Molecular Weight and Melting Point, and (b) locations of compounds in the subspace formed by Molecular Weight and LogP.

4.3.2 Recursive Partitioning Approach

The analysis of multi-mechanism data is difficult, and many statistical methods are not expected to be successful. Recursive partitioning (RP), Hawkins and Kass (1982) and Breiman et al. (1984), is one method that has been successful with multiple mechanisms arising in drug-screening data (Hawkins et al. 1997, Young and Hawkins 1998, Rusinko et al. 1999, and Jones-Hertzog et al. 2000). RP selects a descriptor to partition the data into two or more groups or nodes that are more homogeneous. Each daughter node is partitioned in turn until the nodes are judged homogeneous or some minimum sample size is reached. This separation of the data into smaller groups can, at least in principle, isolate the active compounds due to a single mechanism.

As successful as RP has been for the analysis of HTS data sets, there are a number of possible problems. These problems are at least partially due to particular implementations of RP in existing software products, rather than the overall concept. First, RP selects one descriptor at a time to split the data set, but a single descriptor may not provide adequate information for the splitting process. In addition, when the descriptors are highly correlated, selecting one will likely lead to never selecting several others. It is important to keep the following observation in mind: two compounds must have fairly close values of all critical descriptors for similar biological activity (McFarland and Gans, 1986) when there is a single mechanism. This means that partitions have to be narrow, and in several dimensions simultaneously, if all molecules from a partition are to have similar activity. The second problem relates to multiple mechanisms when active compounds from these mechanisms cannot be easily separated. The two-mechanism data shown in Figure 4-1 illustrate the problem. Figure 4-2(a) gives a tree, generated by recursive partitioning. Because logP is never chosen as a partitioning variable, the logP subinterval containing the Mechanism 2 active compounds is not identified. The tree partitions are displayed in Figure 4-2(b); RP incorrectly splits the subspace formed by Molecular Weight and Melting Point into six regions. Here, partitioning one variable at a time is ineffective in dealing with multiple mechanisms. The third problem relates to the use of binary splits in many implementations. Problems can result if the activity pattern is inactive-active-inactive for a descriptor variable. With a single cut point, actives will be combined with inactives, possibly leading to the variable not being selected.

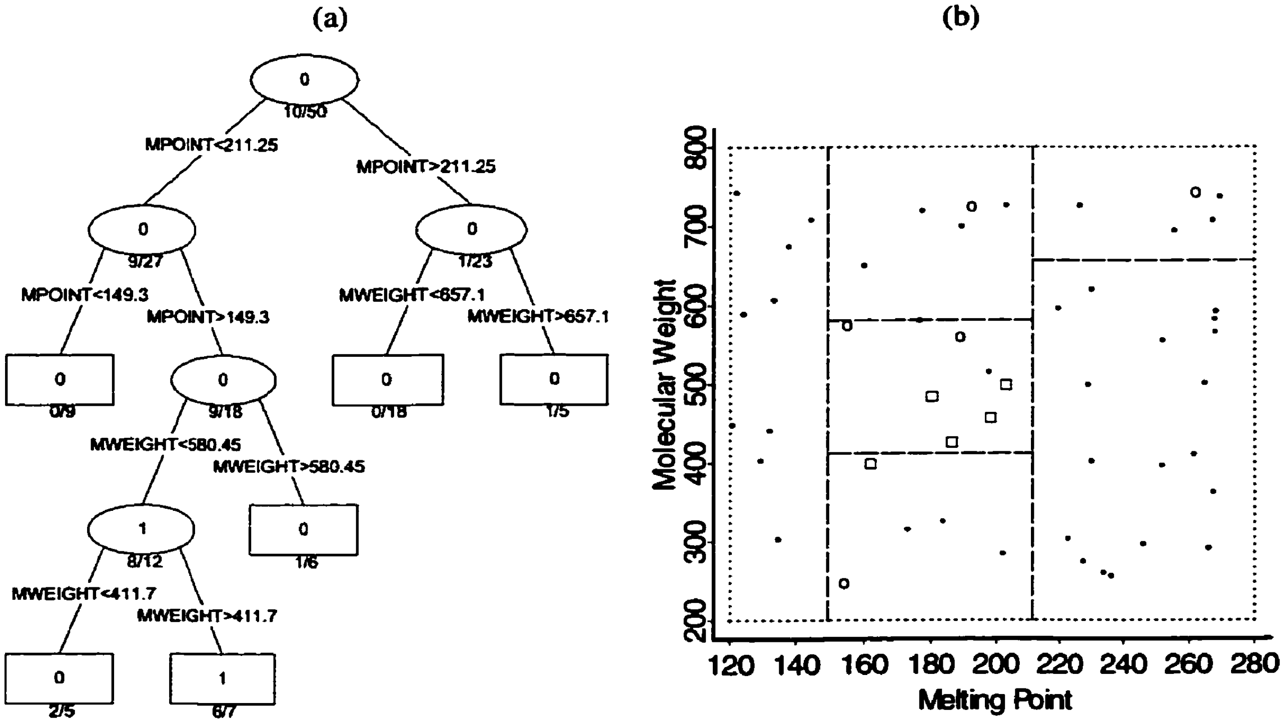


Figure 4-2. Recursive Partitioning of Two-Mechanism Data.

Recursive partitioning (S-Plus tree with default settings, e.g., minimum node size of 5) is used to split the data illustrated in Figure 4-1. (a) Nodes in the tree are classified as active and inactive and are labeled by 1 and 0, respectively. Terminal nodes are represented by rectangles. Under each node, the hit rate is printed. (b) The corresponding partitions are displayed.

4.4 Cell-Based Analysis

For convenience, we refer to a small region of a *d*-dimensional (sub)space as a *d*-dimensional cell. For example, a 2-D cell is a region of a 2-D space.

We introduce a cell-based analysis method that first identifies small regions (cells) with several active compounds in low-dimensional subspaces (projections) of a high-dimensional descriptor space and then uses the information on these cells to score new compounds and prioritize them for testing. The cell-based analysis algorithm involves five stages.

1. Divide the high-dimensional space into many tiny cells (Section 4.4.1).
2. Make a preliminary identification of good cells: those cells with several active compounds (Section 4.4.2). Cells with too few active compounds are removed as there is not enough evidence to achieve statistical significance.
3. Derive ranking scores for the good cells (Section 4.4.3).

4. Determine which of these cells have activity that is statistically significant (Section 4.4.4). Note that a cell might have some active compounds by chance and, because there are very many cells, multiplicity issue arises. We propose a permutation test to overcome this issue.
5. Score and prioritize untested compounds based on the good cells identified (Section 4.4.5). New compounds appearing frequently amongst the good cells are promising candidates for testing.

4.4.1 Forming Subspaces and Cells

We use the data-driven binning method described by Lam et al. (2001) to divide a space into cells. Then we shift these cells in the various dimensions to allow for forming active regions of different shapes.

Binning the Descriptor Space into 1-D, 2-D, and 3-D cells

The advantage of dividing a space into cells is that a number of methods can be developed to identify good cells, i.e., those with a high proportion of active compounds. It is also inherently local, allowing for the isolation of small active regions. We now review some methods for dividing a high-dimensional space into many small, low-dimensional cells.

In a conventional cell-based method, the range for each of the descriptors is subdivided into m bins of equal size. With the 67 BCUT descriptors, we would have m^{67} cells. Even with m=2, there are 2^{67} (or $1.5x10^{20}$) cells generated, most of which are empty even for the largest ever-existing chemical database. There would be more cells than data points. In addition, most compounds will be densely clustered in relatively few cells, making it difficult or impossible to separate active and inactive regions.

Following Lam et al. (2001), we focus our attention on low-dimensional subspaces, typically all 1-D, 2-D, and 3-D subspaces. This strategy is motivated by Pearlman and Smith's (1999) "receptor-relevant subspace" concept. They argued that often only two or three BCUT descriptor variables are important for activity against a particular biological receptor and that activity is highly localized within the relevant subspace. Secondly, we keep the number of cells constant over each subspace, avoiding the exponential increase in the number of cells with dimension. Consequently, the (average) number of compounds per cell does not decrease with dimension, maintaining statistical power for separating active and inactive regions (cells). Furthermore, if only a few descriptors are relevant for a particular mechanism, some low-dimensional cells containing only important variables are likely to be identified, facilitating understanding. In contrast, higher-dimensional cells would include unimportant variables. To keep the number of cells constant, higher-dimensional cells would also have to be larger in the subspace of important variables, possibly too large to isolate a localized active region. Lastly, to avoid empty cells caused by the scarcity of molecules towards the limits of a descriptor's range, we adopt a data-driven hybrid binning method that makes bins larger towards the extremes.

Briefly, cells are created as follows. Initially, we divide each descriptor into m bins. For each descriptor, these bins are immediately the cells for its 1-D subspace. To form the cells for a given 2-

D subspace, amalgamate the m 1-D bins into $m^{1/2}$ larger bins for each of its dimensions. There are $m^{1/2}$ x $m^{1/2}$ = m 2-D cells from combining these larger bins. Similarly, to form 3-D cells, we amalgamate each dimension's 1-D bins into $m^{1/3}$ bins; these are combined to give $m^{1/3}$ x $m^{1/3}$ x $m^{1/3}$ = m 3-D cells. Thus, all subspaces, whether 1-D, 2-D, or 3-D, have the same number of cells. To generate integer numbers of bins, it is convenient if m is an integer raised to the power of 6, e.g., 2^6 =64 or 4^6 =4096. We give further guidance below on choosing m. For more details in binning a high dimensional space into low-dimensional cells see the sections 'Forming Cells' and 'Data-Driven Binning' in Lam et al. (2001).

With k descriptors, there are $$\binom{k}{1}+\binom{k}{2}+\binom{k}{3}=\frac{5}{6}k+\frac{1}{6}k^3$$

1-D, 2-D, and 3-D subspaces in total. For every subspace, a molecule is in one and only one cell. The goal is to find a set of cells in which there are many active compounds and a high proportion of active compounds.

How large should the bin size be? Cells formed from large bins may contain more than one class of compounds. Moreover, if only part of the cell is good, active compounds will be diluted by inactive compounds and the cell may be deemed inactive. (Two compounds must have fairly close values of all critical descriptors for similar biological activity.) On the other hand, a cell formed by very fine bins may not contain all the compounds in the same class. Furthermore, very small cells will tend to have very few compounds and there will be little information to assess the quality of the cell. We make the bins fine, but not too fine, given N, the number of assayed compounds. For reliable assessment of each cell's hit rate, we would like at least 10 compounds per cell. This suggests that the number of cells per subspace should be no more than $N/10$.

Intra-subspace cells (not including the shifted cells described below) within a subspace are mutually exclusive and cover different sets of compounds. On the other hand, inter-subspace cells, cells from different subspaces, can cover the same set of compounds. The compound-selection method described in Section 4.4.5 takes advantage of the collective strength of inter-subspace cells and makes use of the small amount of extra information available when further highly correlated descriptors are added.

Shifted Cells

The data-driven binning method generates non-overlapping cells within a subspace. We call these the original, unshifted cells. Because the location and the shape of an active region are unknown, it is not possible to define the exact boundaries of a cell that perfectly fit an entire active region. The cell boundaries are fixed prior to analysis. For example, an active 2-D region with four active compounds can be sliced, by chance, into four 2-D cells with one active compound in each cell. In this case, none of the four 2-D cells will be identified as good cells and thus the active region will not be found.

To allow for the fact that the original binning may not be optimal, we also shift the original cells in the various dimensions to create overlapping cells (shifted cells). For example, Figure 4-3 shows the locations of 10 active compounds in the subspace formed by two descriptors, x_1 and x_2. To form 2-D cells, the range of each descriptor is divided into five bins here. We generate four sets of cells: one set of original, unshifted cells, two sets of cells with only one dimension shifted by half a bin, and one set of cells with both dimensions shifted half a bin. These four sets of cells are shown in Figure 4-3 (a)-(d), respectively. The good cells identified in analysis are then used to form active regions. If a good cell has to have at least three active compounds (as in Section 4.4.2), there is one active cell in each of Figure 4-3(a) and Figure 4-3(b) and there are two active cells in each of Figure 4-3(c) and Figure 4-3(d). The region formed by these overlapping active cells is shown in Figure 4-4. The counts are the number of times each active compound falls in an active cell. The dashed lines show how the active region could be adjusted to exclude sub-regions with no actives. Note that parts of the active region missed by the original binning are found.

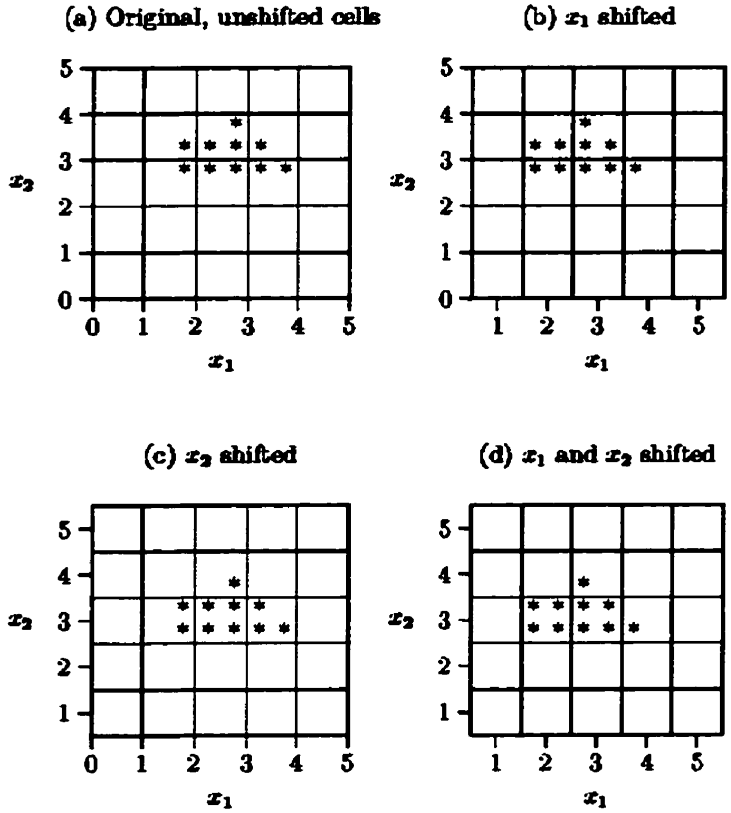


Figure 4-3. Shifted Bins (Five per Descriptor) and Overlapping Cells for 10 Active Compounds in a 2-D Subspace Formed by x1 and x2.

(a) original, unshifted cells; (b) only the x_1 bins are shifted by half a bin; (c) only the x_2 bins are shifted by half a bin; and (d) both the x_1 and the x_2 bins are shifted by half a bin.

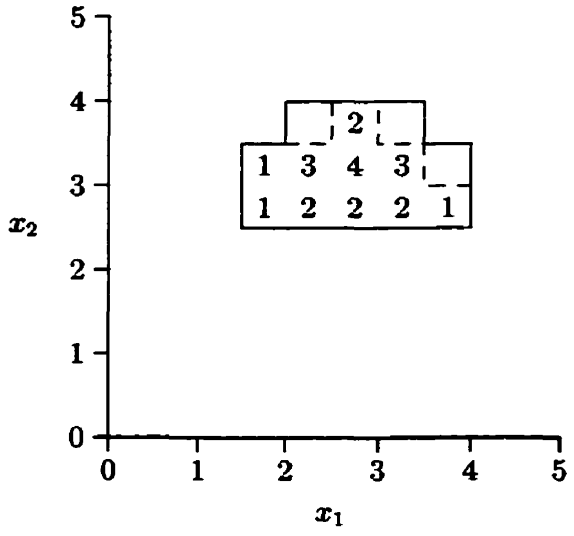


Figure 4-4. Overlapping Shifted Cells to Form an Active Region.

The box denotes the active region. The counts are the number of times each active compound is selected by active cells (those with at least three active compounds). The dashed lines show how the active region could be adjusted to exclude sub-regions with no actives.

The shifted cells provide an effective means of handling different shapes of active regions, at the price of looking at more cells. The number of cells created for a *d*-dimensional subspace is increased by a factor of 2^d and the number of bin cut-off points for each dimension is doubled. For example, if a 3-D subspace is divided into 4×4×4 = 64 cells, shifting will lead to a total of 8×64 cells, which is as many as an 8×8×8 arrangement. Therefore, this method allows us to use larger bins for the analysis, with more compounds per cell, and hence higher power for detecting activity.

We also investigated several methods for determining the shape and the size of an active region. However, we found that growing and shrinking a cell around an original, active cell to cover adjacent active cells was more complex and not as effective and efficient as shifting cells.

4.4.2 Preliminary Identificatio n of Good Cells

We make a preliminary reduction of the huge number of cells that can be generated, particularly when there are many descriptors. We search every cell and note the ones with several (say three) active compounds. These are preliminary good cells. Then we adjust the boundaries of the preliminary good cells to exclude sub-regions with no active compounds. In later stages of the analysis, the hit rate and other related statistics will be computed for each of the re-sized cells. Those cells with a low proportion of active compounds will be removed. Active regions will be created by combining the remaining good cells.

Preliminary Good Cells

After the (original and shifted) cells are constructed, the next step is to search for preliminary good cells: those with at least a certain number of active compounds. The required number of active compounds will depend on the total number of active compounds found in the data set. If only a few active compounds are available (e.g., less than 20), then all cells with two or more active compounds might be of interest. On the other hand, if there are hundreds of active compounds, then it is more efficient to pay attention to only those cells with, say, at least five active compounds. Of course one can examine every single cell with one active compound but this will generate many preliminary good cells by chance. For the examples described in Section 4.5, there are about 80 active compounds in the NCI training data set and about 40 active compounds in the Core98 training data set. In these examples, requiring two active compounds gives similar results to requiring three, but the latter generates fewer preliminary good cells.

The search for the preliminary good cells is straightforward. In principle, we just need to count the number of active compounds in every cell in every subspace. Because active compounds are usually rare in the data set, the search can be made computationally efficient by tracking them to the relatively few cells that they occupy. Subsequent stages of analysis are made much faster by working with the much-reduced list of preliminary good cells.

Re-sizing Cells

As the cell boundaries are fixed prior to analysis, a cell may cover both active and inactive regions and hence the observed hit rate of a cell can be misleading (active compounds may be diluted by inactive compounds, yielding a very low hit rate). To get a more focused region, we re-size each cell by trimming off the borders with no active compounds. Then, in each trimmed cell, we use the compounds remaining to determine the hit rate and other related statistics. These trimmed cells will be used later on to form active regions and to score and prioritize untested compounds for screening.

4.4.3 Ranking Cells

The next stage is to rank the re-sized cells (original and shifted). These rankings will be used in the later stage to score new compounds. All the ranking criteria are based on measures for individual cells.

With active/inactive binary-response data, a natural first choice for the identification of active cells is to compute the proportion of all the compounds in the cell that are active (the observed hit rate) and then rank the cells by these proportions. The main problem with this method is that it favors cells that happen to have a small number of compounds. Consider two cells with 2/2 and 19/20 active compounds, respectively. The first has a hit rate of 100%, but this is based on two compounds, a very small sample. The 95% hit rate for the second cell is based on 20 compounds and is much more reliable. Thus, in addition to the raw hit rate (H), we describe below two further criteria that take into account the statistical variability from sampling: p-value (P) and the binomial hit rate lower confidence limit (H_{L95}).

With a numerical assay value Y (e.g., percentage inhibition) for activity, we will similarly describe the raw mean activity score ($\overline{Y}$) and two criteria penalizing a small sample size: the lower confidence interval for the mean Y ($\overline{Y}_{L95}$) and the hit rate lower confidence limit based on a normal distribution for Y (H^{Y}_{L95}). Quantitative data of this type may also be converted to active/inactive classes by defining "Active" as $Y > c$ for some cut-off c, allowing all criteria to be used.

P-value (*P*)

Let N be the total number of compounds in a data set (e.g., 4096 compounds in the Core98 training set), and let N_a be the number of active compounds in the data set (e.g., 41 active compounds). Consider a given cell in a given subspace, which has n compounds, of which a are active.

Suppose the N_a active compounds are distributed such that they fall in or outside the given cell at random. Under this statistical null hypothesis, the probability of observing a actives out of n compounds is given by the hypergeometric distribution:

$$\text{Prob}(a;n,N_a,N)=\frac{\binom{N_a}{a}\binom{N-N_a}{n-a}}{\binom{N}{n}}$$

The p-value is the probability of having at least a active compounds out of n:

p-value = Prob($A \geq a \mid n$ compounds)

$$=\sum_{i=a}^{\min(N_a,n)}\frac{\binom{N_a}{i}\binom{N-N_a}{n-i}}{\binom{N}{n}} = 1-\sum_{i=0}^{a-1}\frac{\binom{N_a}{i}\binom{N-N_a}{n-i}}{\binom{N}{n}}.$$

If the p-value is small, there is little chance of seeing a or more active compounds out of n. Therefore, small P-values provide the most evidence against the null hypothesis of random allocation of actives in/outside the cell (and hence most evidence that the number of actives in the cell is better than chance). The P-value is computed for all cells and the cell with the smallest P-value is the top-ranked cell, etc.

The p-value approach tends to pick cells with large numbers of compounds even if they have fairly low hit rates. Suppose there are 40 active compounds in a data set of 4,000 compounds. Then 8 actives out of 80 (hit rate=0.10) gives $p=8.24\times10^{-7}$ but 3 out of 3 (hit rate=1.00) gives $p=9.3\times10^{-7}$.

The statistical evidence is stronger in the first case because of the larger sample size, even though the hit rate is much lower. This illustrates the major drawback of the *P* criterion: it tests whether the hit rate is significantly larger than random, not whether the hit rate is large.

Hit Rate (*H*)

In the above notation, the hit rate for a cell is *a/n*. It ignores the increased reliability from a larger sample size. For example, 1/1 gives a 100% hit rate but 9/10 gives a 90% hit rate, yet the cell with 9/10 seems more promising. Although commonly used, it is not a sensitive criterion for ranking active cells (regions). The next criterion introduced considers both the hit rate and its variability.

Binomial Hit Rate Lower Confidence Limit (H_{L95})

One can obtain an exact lower confidence limit on the hit rate for new compounds based on the binomial distribution. For the many possible compounds that would fall in a given cell, suppose that a proportion h are active, i.e., h is the hit rate. Assuming that the n compounds in the cell that have been assayed are a random sample of all the cell's possible compounds, the number of actives, A, is a random variable following a binomial distribution with n trials and probability h. The smallest value of h such that $\text{Prob}(A \geq a \mid h, n) = 0.05$ is the 95% binomial hit rate lower confidence limit (H_{L95}). It considers both the hit rate and its variability. Some examples of cell rankings using the H_{L95} method are given in Table 4-2.

Table 4-2. Illustrative Cell Rankings Using H_{L95}.

Cell	a/n (Hit Rate)	H_{L95}	Ranking
1	9/10 (0.9)	0.6058	1
2	3/3 (1.0)	0.3684	2
3	8/80 (0.1)	0.0507	3
4	1/1 (1.0)	0.0500	4

Mean Activity Score ($\bar{Y}$)

When a numerical assay value, Y, is available, the mean over all compounds in a cell gives the mean activity score ($\bar{Y}$). Because it is easier by chance to obtain a high mean from fewer compounds than from more compounds, $\bar{Y}$ tends to pick cells with few compounds and high activity values. Although commonly used, it is not a sensitive criterion for ranking active cells (regions). The next criterion introduced considers both the observed mean and its variability.

Lower Confidence Limit for Mean Y ($\overline{Y}_{L95}$)

Analogous to H_{L95}, with a numerical assay value, Y, one can use the lower confidence limit for the mean of the distribution giving the Y values, based on an assumption of sampling from a normal distribution. This criterion, $\overline{Y}_{L95}$, considers both the observed mean and the variability and is defined as

$$\overline{Y}_{L95} = \overline{Y} - \hat{\sigma} / \sqrt{n} \times t(n-1, 0.95),$$

where, based on n-1 degrees of freedom, $\hat{\sigma}$ is the sample standard deviation within the cell and $t(n\text{-}1, 0.95)$ denotes the 95% quantile of the t distribution.

Normal Hit Rate Lower Confidence Limit (H^{Y}_{L95})

With a numerical measure of activity, Y, and a cut-off for activity, c, one can derive a lower confidence limit for the hit rate, i.e., the probability Prob(Y>c), based on the assumption that the observed activities in a cell are randomly sampled from a normal distribution. This criterion is called H^{Y}_{L95}.

If the Y values are randomly sampled from a normal distribution with mean μ and variance σ^2, then by definition, H^{Y}_{L95} is

$$\Pr(Y > c) = 1 - \Pr(Y \le c) = 1 - \Pr\left(\frac{Y-\mu}{\sigma} \le \frac{c-\mu}{\sigma}\right) = 1 - \Phi\left(\frac{c-\mu}{\sigma}\right) = \Phi\left(\frac{\mu-c}{\sigma}\right),$$

where Φ is the standard normal cumulative distribution function.

Suppose σ is known or a good estimate is available (the pooled estimate described below will usually have many degrees of freedom). Then we can estimate Φ by

$\hat{\Phi} = \Phi\left(\frac{\overline{Y}-c}{\sigma}\right)$, where $\overline{Y}$ is the average Y value for the n compounds in the cell.

Let $Z = \frac{\mu - c}{\sigma}$, which we estimate by $\hat{Z} = \frac{\overline{Y}-c}{\sigma}$. We have $E(\hat{Z}) = \frac{\mu-c}{\sigma}$ and $Var(\hat{Z}) = \frac{\sigma^2}{n}\frac{1}{\sigma^2} = \frac{1}{n}$. Therefore,

$$\hat{Z} \sim N\left(\frac{\mu-c}{\sigma}, \frac{1}{n}\right) \text{ and } \Pr\left(\frac{\hat{Z} - \frac{\mu-c}{\sigma}}{1/\sqrt{n}} < Z_{.95}\right) = 0.95,$$

where $Z_{.95}$ is the 95% quantile of the standard normal distribution.

Rearrangement of the inequality gives

$$\Pr\left(Z_L < \frac{\mu - c}{\sigma}\right) = 0.95\text{, where } Z_L = \hat{Z} - \frac{Z_{.95}}{\sqrt{n}}.$$

A 95% lower confidence interval (CI) for $\frac{\mu - c}{\sigma}$ is (Z_L, ∞) and the corresponding 95% CI for $\Phi\left(\frac{\mu - c}{\sigma}\right)$ is ($\Phi(Z_L)$, 1) since Φ is a monotonic increasing function. Therefore, H_{L95}^{Y} can be estimated by $\Phi(Z_L) = \Phi\left(\frac{\bar{Y} - c}{\sigma} - \frac{Z_{.95}}{\sqrt{n}}\right)$.

We use a common estimate of σ for all cells within a subspace. For a given subspace, it is computed by pooling the sample variances over all cells:

$$\hat{\sigma}^2 = \frac{\sum (n_i - 1) s_i^2}{\sum (n_i - 1)},$$

where s_i^2 is the sample variance for cell i, and cell i has n_i compounds.

Relationships between the criteria

If a numerical measure of activity is available, all six criteria can be used. The cut-point c for activity (a hit) is used as follows. For P, H and H_{L95}, c is used to convert the data to "Active" / "Inactive" before they are computed. Both $\bar{Y}$ and $\bar{Y}_{L95}$ ignore c. For H_{L95}^{Y}, the Y distribution is modeled and c is used at the end to determine H_{L95}^{Y}.

4.4.4 Assessing the Impact of Multiple Testing

With 67 descriptors, there are a total of 50,183 1-D, 2-D, and 3-D subspaces. If each subspace is divided into 64 cells and the cells are shifted in the various dimensions (see Section 4.4.1), there are 25,101,952 (shifted and unshifted) cells. With so many cells, it is possible that by chance alone we will see cells with moderate activity.

Consider the p-value criterion. To adjust it for the total number of cells examined, C, we simply multiply each p-value by C. This is the Bonferroni correction (Miller 1981). In the training data, a cell is said to be a good cell by the p-value criterion if the Bonferroni adjusted P is small (say <0.05).

The Bonferroni correction tends to over-correct, but we can impose a minimum number of active compounds to define the cells relevant for correction. In the NCI example with 67 BCUTs and 25,101,952 cells, for example, only 5,587,591 cells have at least two active compounds, a smaller adjustment factor.

Probably the best way of addressing the multiple testing problem is to define the cut-off between active and inactive cells using a random permutation of the assay values. The Active/Inactive indicators or *Y* values in the training data are randomly reordered, i.e., randomly assigned to the descriptor combinations in the data set. If p-value is the criterion for ranking cells, one can set the cut-off as a small p-value in the lower tail of the distribution induced by randomization. Under random permutation of the data, no cells should be identified as good cells and the smallest p-value is just due to chance. For the actual data (without permutation) one can then use all cells with p-value smaller than this cut-off point.

Ideally, to estimate the p-value corresponding to a true significance level of say 5%, we would like to perform many random permutations. The sets of p-values from these randomizations would be combined into an empirical distribution, and the 5% point from this distribution is a multiplicity-adjusted critical value. This is too computationally expensive, however. Fortunately, for the cell-based analysis, one permutation provides many p-values (e.g., 25,101,952 cells and hence p-values). Thus, we take the 5% point from one permutation as the cut-off to determine whether there are any real active cells (versus false alarms). This procedure can be applied to any of the cell-ranking criteria in Section 4.4.3.

Cells with ranking scores in the actual data that beat the random-permutation cut-off are used to score and select new compounds. The subspaces and descriptor ranges associated with these cells indicate descriptors that are likely relevant to activity and subregions of activity, respectively. New compounds appearing in the most highly ranked cells or frequently amongst the good cells are promising candidates for testing, as described next.

4.4.5 Selection of New Compounds

We present three selection methods for choosing untested compounds for biological screening: 'Top Cells Selection', 'Frequency Selection' and 'Weighted Score Selection.' All the methods first rank cells according to one of the criteria in Section 4.4.3 and apply the random-permutation method of Section 4.4.4 to generate a list of good cells.

Top Cells Selection

In a database of new, unassayed compounds, top-cells selection chooses all the compounds falling in the best cell, then all those in the second best cell, and so on until the desired number of compounds to be tested is reached or until there are no good cells remaining. This approach does not combine strength from several good cells when scoring a compound. The next method takes advantages of the collective strength of the good cells, thus increasing the prediction power.

Frequency Selection

Frequency selection scores a new compound by the number of times it appears in the list of highly ranked cells. The first compound selected for screening is the one occurring with the maximum frequency, the second compound selected has the second largest frequency, and so on.

Frequency selection scores a compound based on many good cells and possibly many descriptors. A single cell belongs to a subspace involving only one, two or three variables, and cells are scored individually. In contrast, under frequency selection, if a new compound appears in several good cells in different subspaces, information is combined from the union of all the subspaces' descriptors. Thus, frequency selection can potentially make use of the small amount of extra information available when further highly correlated descriptors are added (see the comparison of 6 and 67 descriptors in Section 4.5.3).

Frequency selection provides a powerful way to rank new compounds for screening, often leading to a very high hit rate for the top ranked compounds. The next method introduced further improves the compound ranking by incorporating information on the order of the cells in the list.

Weighted Score Selection

Instead of just counting the frequency of occurrence in the list of good cells, we can give each cell in the list a weight and score a new compound based on the total weight over the cells in which it resides.

The cell-ranking criteria described earlier can be adapted as weight functions. We could use the H_{L95} value or –log(p-value) as weights, for example. The weight function should have several desirable properties: (1) If the list of good cells is extended, the relative weights of the cells in the original list should not change; (2) the weight function should be a smooth monotonic decreasing function of the cell's rank; and (3) the same weight should be assigned to cells rated equally by the cell ranking criterion.

For the numerical evaluations in Section 4.5, we use weighted score selection with H_{L95} (NCI binary-response data) or $\overline{Y}_{L95}$ (Core98 continuous-response data) values as weights. These are the criteria used for cell ranking to generate the list of good cells.

4.5 Performance Evaluatio n

We evaluate the performance of our cell-based analysis method using the 23056 Core98 compounds and–the 29812 NCI compounds. The objective of this evaluation is (1) to determine if the new methods lead to higher hit rates than random selection, (2) to assess the effect of the six cell selection criteria on hit rate, and (3) to determine whether our cell selection method can find real active cells or false alarms.

In addition, we compare the cell-based analysis method with recursive partitioning (the tree function in S-Plus, Clark and Pregibon 1992) in terms of identifying active compounds. Often, V-fold cross-validation is used to control tree size, but here this tends to result in a very small tree, sometimes with only a root node. This problem seems to arise because active compounds are rare in the training data, and the smaller hold-out samples have too few active compounds to compare different tree sizes. For simplicity, then, we use the default S-Plus tree (from default fitting options, e.g., minimum 5 observations per node) and do not attempt to prune this tree. (Some preliminary work by graduate student Marcia Wang also suggests that tree pruning is ineffective anyway.)

4.5.1 Evaluation Plan

To evaluate the cell-based analysis method for the two data sets, we carry out the following steps.

1. Divide the data into Training and Validation sets. Samples of 4096 compounds are selected to form the training data set; the rest of the compounds form the validation data set. Samples are chosen randomly or using uniform coverage designs (Lam et al. 2001).
2. Apply the data-driven hybrid binning method to bin all subspaces, and create 64 cells per subspace. This gives 64 compounds per cell on average. Create shifted cells from the original bins (Section 4.4.1).
3. Training set: Search for preliminary good cells with two or more active compounds (Section 4.4.2).
4. Training set: Compute summary statistics for the preliminary good cells: H_{L95} for the NCI binary-response data or $\overline{Y}_{L95}$ for the Core98 continuous-response data (Section 4.4.3). Perform a permutation test (Section 4.4.4) to find the cut-off point to separate good cells from false alarms. This generates a list of good cells (considered as 'real').
5. Validation set: Score and select new compounds from the validation set based on the good cells identified from the training set. Here we can rank the validation-set compounds using weighted score selection (Section 4.4.5).
6. Validation set: As compounds are successively selected, evaluate the hit rate (binary response) or mean activity value (continuous response) as performance measures.

We first look in detail at the multiplicity correction in Step 4, then present the final hit rate and mean activity performance results.

4.5.2 Good Cells Versus False Alarms

Bonferroni Correction

To test whether our cell-based method would give false-positive results, the activity values are randomly re-assigned to the compounds. The cell-based analysis is carried out on the permuted data. Using the p-value correction method described in Section 4.4.4, few cells are declared good. On the other hand, many good cells are found using the real activity values. This approach addresses the false positive problem, but is probably quite conservative.

Permutation Test

To illustrate how the permutation test in Step 4 works, we examine a random sample of 4096 compounds from the NCI data set with 67 descriptors. For this sample, we generate 25,101,952 cells (see Section 4.4.4) and analyze these cells twice, once with the original activity values and once with randomly re-arranged activity values. Under random permutation, the best cell had 7 out of 7 active compounds, with P=2.29×10^{-12} and H_{L95}=0.6518, as shown in Table 4-3. With the real data, Table

4-3 also shows there are 5,256 cells with a smaller p-value and 449 cells with a larger H_{L95} value, suggesting that these cells are indeed good active regions and that the descriptors are relevant to the activity.

Table 4-3. *P* and H_{L95} Values for Different Cut-Off Points and the Corresponding Number of Cells with a Better Value.

	Under Random Permutation Criterion value (#actives/#compounds)		Real Data #cells with better value	
	P	H_{L95}	P	H_{L95}
Best value	2.29E-12 (7/7)	0.6518 (7/7)	5,256	449
The 5% point	1.04E-4 (6/35)	0.2236 (2/2)	782,864	493,962

For defining the list of good cells and hence selecting new compounds for screening, we use a less conservative cut-off: the 5% point of the distribution under randomization. Using the 5% point, many more cells are found with better values. Collectively, these cells enhance the prediction power of the compound selection methods. Scoring new cells using weighted frequency of occurrence in the list of good cells (Section 4.4.5) is insensitive to adding some possibly spurious cells to the bottom of the list: these cells have low weights.

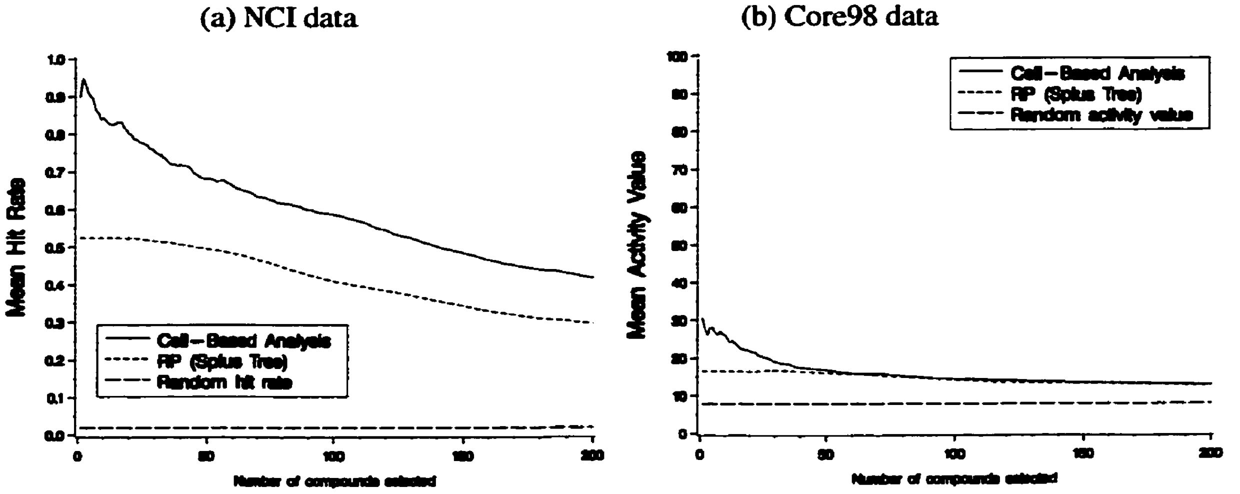
In this example, two practical issues are also revealed. As mentioned earlier, the raw hit rate H is not a sensitive cell-ranking criterion, and the permutation test based on H often leads to a hit rate cut-off at 100%, making identification of good cells difficult. Also, one has to be careful in the implementation of the P criterion as the p-values for good cells can be extremely small. In addition, P tends to pick cells with large numbers of compounds (Section 4.4.3), hence our use of H_{L95} as the primary ranking criterion for data with binary response. Similar comments apply to the Core98 continuous-response data and our preference for the $\overline{Y}_{L95}$ criterion.

4.5.3 Validation Hit Rates

A total of 80 training sets were generated, 40 from each of the NCI and Core98 data sets. Half of the training sets were generated using random selection and the other half were generated using uniform coverage designs (Lam et al., 2001). For comparisons between the cell-based (CB) analysis method and recursive partitioning (RP), the S-Plus classification and regression tree method with default settings is used (Venables and Ripley, 1999). Except where we specifically compare 6 and 67 descriptors, all analyses are performed using the original 6 descriptors to reduce the burden of computational effort.

Cell-Based Analysis Versus Recursive Partitioning

Twenty training sets of 4096 compounds were randomly generated from each of the NCI and Core98 compounds. These training sets were analyzed using both CB and RP analysis methods. The mean hit rates and mean activity results based on the validation sets are shown in Figure 4-5. For the NCI compounds, the CB analysis clearly dominates the RP analysis. Both methods generate hit rates many times higher than the random hit rate. For the Core98 compounds, the CB analysis outperforms the RP analysis for the first 50 compounds selected; thereafter the two methods are comparable. Again, both methods perform much better than the random-activity baseline. The Core98 activity values have much larger measurement error than the NCI activity; in addition, the Core98 compounds have fewer hits. Both of these facts make predicting active compounds difficult.



(a) NCI data

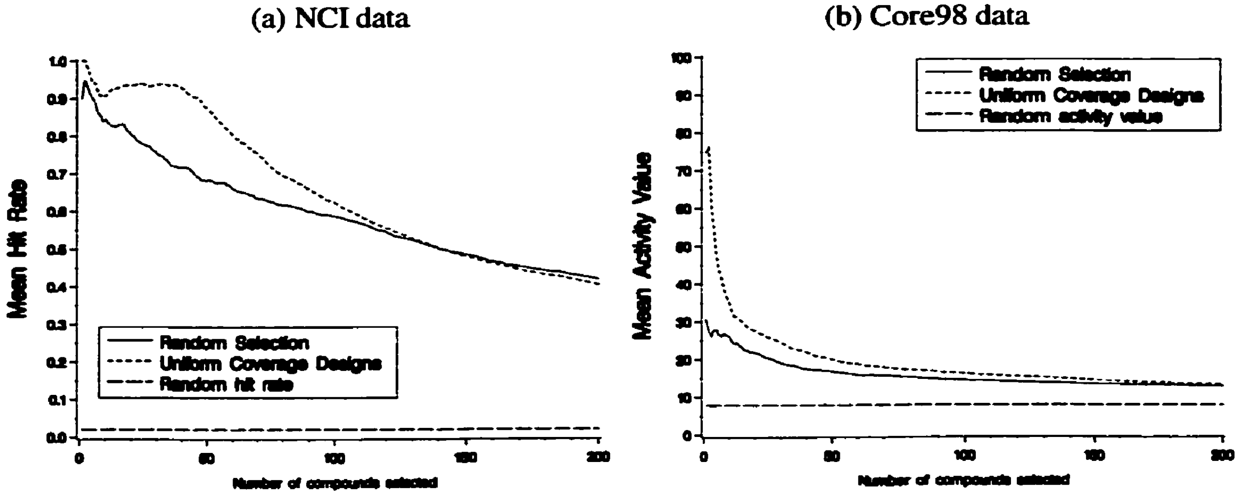
(b) Core98 data

Figure 4-5. Average Performance of Cell-Based Analysis (Solid Line) and Recursive Partitioning (Dashed Line) for 20 Random Samples When the 200 Validation-Set Compounds With the Highest Scores Are Selected.

(a) Mean Hit Rate for the NCI Binary Data and (b) Mean Activity for the Core98 Continuous-Response Data. The horizontal line near the bottom shows the expected performance under random selection of new compounds.

Impact of Design on Cell-Based Analysis: Uniform Coverage Designs Versus Random Selection

Here we investigate the impact of different designs for the training data on the performance of the CB analysis. Two types of designs are compared: simple random sampling (as in the CB versus RP comparison) and uniform coverage designs (Lam et al., 2001). By keeping the sample size within each cell fairly constant, the uniform coverage designs should provide good power across all cells. Twenty training sets of 4096 compounds are generated, using the two methods, from each of the NCI and Core98 data sets. These training sets are analyzed using the CB analysis method. The mean hit rate and activity results are shown in Figure 4-6. Using the uniform coverage designs, additional improvement in hit rate or mean activity is found for the first 100 compounds selected. The bumps in Figure 4-6(a) when only 1-25 compounds are selected are likely due to the discreteness of the binary response: a few extra hits will make a big impact on the hit rate.


(a) NCI data

(b) Core98 data

Figure 4-6. Average Performance of 20 Random Designs (Solid Line) and 20 Uniform Coverage Designs (Dashed Line) When the 200 Validation-Set Compounds With the Highest Scores are Selected By Cell-Based Analysis.

(a) Mean Hit Rate for the NCI Binary Data and (b) Mean Activity for the Core98 Continuous-Response Data. The horizontal line near the bottom shows the expected performance under random selection of new compounds.

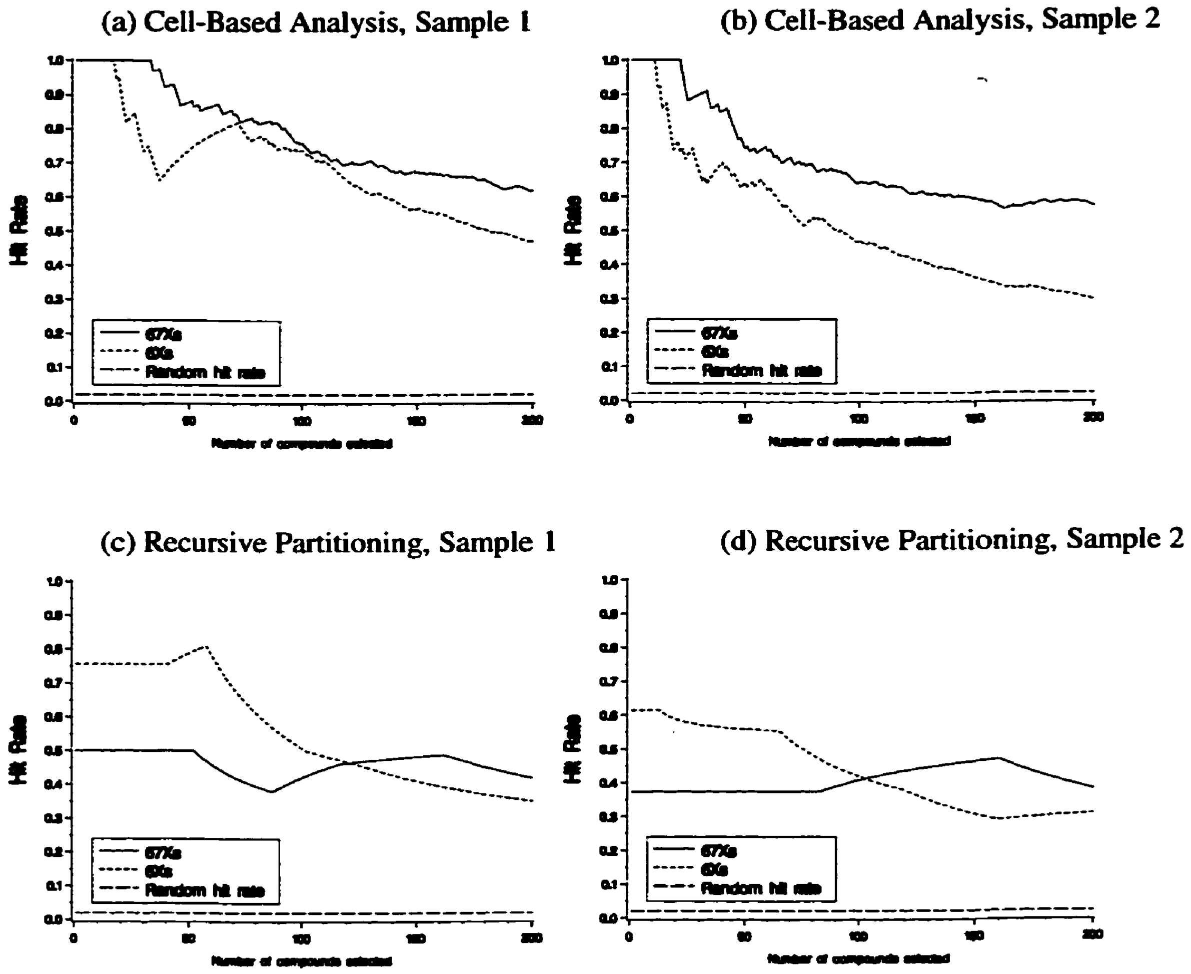
Six Versus 67 Descriptors

Because of high computational cost, only two samples from the 20 random training sets for the NCI compounds are chosen to evaluate the information gain from using more BCUT descriptors. The two samples have the highest and lowest validation hit rates at the 100th compound selected in the six-descriptor cell-based analysis: 74/100 hits and 47/100 hits, respectively. Re-analysis of the same two samples using the 67 descriptors gives the hit rate results shown in Figure 4-7. In both samples, the 67 descriptors lead to higher hit rates for the CB analysis. The CB analysis gains predictive power despite the strong correlations among the descriptors. This is not so for the RP analysis. The hit rate results at 100 compounds selected are summarized in Table 4-4.

Figure 4-7 also indicates that CB analysis is fairly robust to variability due to random sampling. Designs generated by different random samples will lead to training data with little overlap. Therefore, CB analysis will probably be working with rather different sets of preliminary good cells, cell scores, and compound scores. Nonetheless, as Figure 4-7 shows, the hit-rate performance is similar, especially if 67 descriptors are used. The differences between the hit-rate profiles for the two samples are small here relative to the differences between CB analysis and recursive partitioning. In general, CB analysis is not likely to be sensitive to small changes in the data (e.g., adding or removing a few compounds), because such changes will only affect a few cells and the method is inherently local.

Table 4-4. Hit rates, at the 100th Compound Selected, by Different Analysis Methods and by Different Sets of Descriptors.

Sample	Cell-Based Analysis		Recursive Partitioning	
	6 BCUTs	67 BCUTs	6 BCUTs	67 BCUTs
1	0.740	0.760	0.508	0.421
2	0.470	0.640	0.418	0.407



(a) Cell-Based Analysis, Sample 1

(b) Cell-Based Analysis, Sample 2

(c) Recursive Partitioning, Sample 1

(d) Recursive Partitioning, Sample 2

Figure 4-7. Hit Rates for Two Random Samples from the NCI Data Using Either 6 Descriptors (Solid Line) or 67 Descriptors (Dashed Line).

The figure also compares CB and RP analyses.

4.6 Conclusions and Disc ussion

These results confirm that (1) the cell-based analysis method is useful in identifying good cells, (2) many good cells are found, not false alarms, and (3) the BCUT descriptors are informative. Our cell-based analysis method leads to hit rates many times higher than the random hit rate. It consistently leads to very high hit rates for the top ranked compounds. To get a sense of the possible increases in efficiency, consider the following. Using random screening, one would expect to screen 1,000 NCI compounds to find 20 active compounds; however, using the CB prediction one can identify 20 active compounds by screening just 20 compounds: see the curves for 67 descriptors in Figure 4-7(a) and Figure 4-7(b). The CB prediction method compares favorably with RP here.

In principle, because it is inherently local, a cell-based analysis should be able to handle nonlinear, threshold, and interaction effects as well as multiple activity mechanisms. By combining scores from many cells (low-dimensional projections) it should also be able to extract further information from highly correlated descriptors.

On the other hand, linear regression models are not effective in handling these modeling issues. For illustration, polynomial regression models of degree 3 including interaction terms of 2 and 3 descriptors were fitted to the Core98 data using the stepwise-selection method. The 'best' model had $R^2 = 0.01$ and poor prediction accuracy in identifying compounds as active. For the NCI data, logistic regression models were also investigated. Overall, low prediction accuracy in classifying compounds as active and high prediction accuracy in classifying compounds as inactive were found. As only about 2% of compounds are active, any methods claiming all compounds as inactive will give an overall accuracy of 98%. The real challenge is to find a high proportion of active compounds.

Our goal is to find a set of regions (cells) in which there is a high proportion of active compounds. It is much easier to divide and cover low-dimensional subspaces and to identify low-dimensional active cells. Whereas RP evaluates one descriptor at a time, CB analysis evaluates 1-D, 2-D and 3-D cells (i.e., evaluates one, two and three descriptors at a time) and combines these cells when scoring to form high-dimensional active regions. Furthermore, the low-dimensional cells are formed from all combinations of different subsets of descriptors, so all descriptors can be effectively evaluated and the impact of irrelevant variables on analysis is reduced or eliminated. Therefore, focusing on low-dimensional subspaces is effective in finding active and inactive regions (cells).

Shifted cells provide an efficient and effective method for handling different shapes of active regions. In combination with re-sizing of cells, the boundaries of active regions can be better aligned. In compound selection, a compound appearing in more than one cell within the same subspace will be counted only once to avoid over-counting from the shifted cells. This is analogous to (1) forming an active region within a subspace, and (2) ranking the new compounds based on their (weighted) frequency in all active regions across all subspaces.

Designed experiments (e.g., uniform coverage designs) can enhance the predictive power of cell-based analysis. The actual improvement in prediction can be much greater and can be better evaluated if a real test set (instead of a hold-out set for validation) is available, as compounds in the hold-out set are not always available in every cell identified from the training set. Uniform coverage designs tend to select roughly the same number of compounds from both crowded and sparse regions and might not leave compounds in the sparse regions for validation.

A good prediction method should obtain more hits for the highest ranked compounds. Because the total number of hits is a constant, the hit rate or the activity value typically decreases as the number of tested compounds increases, all the way down to the random rate when all compounds are tested. The CB analysis method is particularly effective in finding hits when few compounds are selected.

We primarily used H_{L95} for binary response data and $\overline{Y}_{L95}$ for continuous response data. These criteria take account of uncertainty from the sample size and have fewer assumptions.

CB analysis is a multi-stage automated analysis process, which requires extensive computing power. There are many opportunities to make the algorithm more efficient as well as to further enhance the prediction accuracy. We are currently investigating these opportunities. One can use a combination of different ranking criteria (e.g., the P and H_{L95} values) to define a 'common' cut-off or even to select multiple sets of good cells (different criteria may select different types of active cells). For a very large data set (e.g., millions of compounds with many descriptors), a fast algorithm to store and evaluate billions of cells is needed. Tuning parameters such as the minimum number of active compounds required for a preliminary good cell, choosing cut-offs for the good cells, and more sensitive weighting functions for scoring cells and hence compounds, will be studied. The current cell re-sizing method (Section 4.4.2) re-sizes each cell by trimming off the borders with no active compounds. This is done by setting the new boundaries of a cell to the descriptor ranges of the active compounds. The more active compounds available in the cell, the better the boundaries can be located. Using this simple re-sizing method alone, without the shifted cells method, may leave holes within an active region. The shifted and unshifted cells overlap each other, thus reducing or minimizing possible holes in an active region. Other cell re-sizing methods will be investigated.

Chapter 5
Conclusions and Discussion

5.1 Conclusions

The design and analysis problems for HTS data are quite different from those arising with conventional small data sets. Molecular databases can have thousands to millions of compounds, and there are many potential descriptors to characterize compounds. The complexities of the relationship between descriptors and activity for HTS data (e.g., multiple mechanisms, thresholds, interactions, nonlinearities, etc.) and the general issue of curse of dimensionality in high dimensions make many standard design and analysis methods inappropriate.

The novel design and analysis methods proposed here can overcome these difficulties. The uniform coverage design method first divides a high dimensional space into many low dimensional cells and then uses a fast exchange algorithm to optimize the uniform coverage of these cells. Uniform coverage designs are useful for finding diverse lead compounds for drug optimization. The cell-based analysis method first identifies cells with a high proportion of active compounds and then uses the information on these cells to score new compounds and prioritize them for screening. Cell-based analysis uses the collective strength of multiple cells to enhance the prediction power and is very effective in finding a high proportion of active compounds from a relatively small set of compounds selected.

Several cycles of screening are expected to be more efficient than screening all the compounds in a large collection (Jones-Hertzog et al. 2000). In a multi-stage design strategy, the initial design should cover the descriptor space as uniformly as possible. Analysis of the resulting data can then be used to direct subsequent designs to sub-regions of high activity in critical descriptor projections. An ideal approach is to use uniform coverage designs to select a small initial screening sample and then use cell-based analysis to develop prediction rules to guide the selection of further compounds for screening. This approach can find more leads in less time and with a much smaller number of compounds tested.

This thesis research explores the area of design and analysis of large data sets in pharmaceutical drug discovery. The proposed design and analysis algorithms can efficiently deal with hundreds of thousands of compounds. Further research is needed to enhance these algorithms to deal with even larger data sets and/or a larger number of descriptors. Computational and other potential areas for improvement and research are discussed next.

5.2 Further Research

One of the major difficulties of dealing with large data sets is speed or how fast a method can generate the intended outcomes, in terms of minutes, hours, days or even weeks. Many conventional design and analysis methods were originally developed for small data sets and not intended for problems of the magnitude considered here. For large data sets, many existing methods would be far too slow (e.g., could take weeks or even longer to run) or might not even work at all.

Our design and analysis algorithms can efficiently deal with hundreds of thousands of compounds with several descriptors. For example, our new design software took approximately 15 minutes to select 729 compounds from a data set with 100,000 compounds and 20 descriptors. It would have taken weeks to months to generate a uniform coverage design from the same data set using SAS Proc OPTEX.

However, as the number of compounds and the number of descriptors increase, even our methods can become too computationally intensive to run on any single computer. There are several ideas currently under investigation to reduce the burden of computational effort and to improve the performance of the design (better coverage) and the analysis (better prediction). Some of these developments are relevant to both design and analysis, while others relate to one of the areas, and it is convenient to describe them under such headings.

5.2.1 Design and Analysis

Number of Subspaces

It is much easier to divide and cover low dimensional subspaces and to identify low dimensional active regions. Other advantages of focusing on low dimensional projections of a high dimensional space, instead of the entire space, are discussed in Chapters 3 and 4. A price for considering all 1-D, 2-D, and 3-D subspaces is that the total number of subspaces increases quickly. With 6, 10, 20, and 67 descriptors there are 41, 175, 1,350, and 50,183 subspaces, respectively. (The thesis presented examples with 6 or 67 descriptors.) In terms of computer space and speed, the 67 descriptors will take roughly 1,200 times (50,183/41) more computer space than the six descriptors. Potentially, many more descriptors could be included. There are at least two ways to reduce the total number of subspaces considered.

1. Use dimension reduction methods such as principal component analysis (PCA). There are two ways to apply PCA to the 67 BCUT descriptors. One simple way is to perform PCA directly to all 67 BCUTs; the other is to first group the BCUTs using chemical knowledge and then perform subgroup PCA. Principal components within subgroups are probably preferable here, as they are more interpretable. Preliminary analysis results indicate that subgroup PCA can be useful in reducing dimension.

2. Focus on only 1-D and 2-D subspaces. The 3-D subspaces generate by far the majority of cells, and ignoring them reduces computation considerably for both the design and analysis methods.

   - Running the UCC optimization algorithm with only 1-D and 2-D coverage can also improve the 3-D coverage. For example, we started with a random selection that had U ($U_{1\text{-D}}$ / $U_{2\text{-D}}$ / $U_{3\text{-D}}$) and P ($P_{1\text{-D}}$ / $P_{2\text{-D}}$ / $P_{3\text{-D}}$) equal to 3170 (1929/2948/4634) and 45.9 (50.7/45.0/41.9), respectively; the optimization algorithm with only 1-D and 2-D coverage led to a design not only with better 1-D and 2-D coverage but also with better 3-D coverage. The corresponding values are 682 (353/602/1091) and 73.6 (79.5/74.0/67.2), respectively.

- Cell-based (CB) analysis with or without 3-D cells was performed on the NCI and Core98 data sets. CB analysis without 3-D cells seems to have better prediction results than the tree method (described in Chapter 4) but not as good as CB analysis with 3-D cells.

Multiple Processors

We are currently working to implement the algorithms on multiple processors. At GlaxoSmithKline, hundreds of PCs can be linked together to solve research problems.

One beauty of the CB analysis algorithm is that it can be easily implemented on multiple processors, and I have tried running eight computers at a time. The uniform design algorithm requires more work to run on multiple processors. Using multiple processors to search for multiple exchanges simultaneously and other ideas are under consideration.

5.2.2 Design

Program Algorithm in C

The original design software (written in SAS Proc IML) was aimed to test the new methods and did not focus on efficiency in dealing with large data sets with many descriptors. We have worked with a computational chemist and a computer science student to develop fast C code for the design algorithm. The new software runs much faster. For example, the original software took approximately 25 minutes to select 729 from 30,000 compounds while the new software took less than 30 seconds. For a large data set with many descriptors, an efficient computer algorithm to store millions of cells is needed. For example, selection of 4,096 compounds to densely cover all subspaces up to 3-D formed by 67 descriptors generates 50,183 subspaces and over 200 million cells, given that each subspace is divided into 4,096 tiny cells to get as dense coverage as possible.

Early Termination

Our design optimization algorithm makes most of the improvement in the first few loops through the candidates. From several data sets and many simulations (using different initial samples) we evaluated, over 99.5% improvement comes from the first 10 loops. Indeed, the majority of the exchanges occur in the first 10 loops. Stopping after 10 loops greatly reduces the run time for very large data sets, with only a small compromise in finding a very good coverage design.

Negative Exchange

The design optimization algorithm allows neutral exchanges (i.e., exchanges that do not change the design criterion) to break away from a design that is only locally optimal. An extension to this is to allow exchanges with a small negative improvement to the design criterion. Preliminary evaluation of this idea has been performed; this approach often finds a slightly better design (with less than 0.1% improvement) but requires many more loops.

Optimality of Design

There is no theorem to show that the UCC design optimization algorithm leads to the optimal design. However, extensive testing of the design optimization algorithm suggests that the designs generated should be in close proximity to the optimal design. Even the most reliable but computer-intensive Fedorov algorithm (Fedorov 1972) is not expected to find an additional improvement of 0.1% or more in U but is likely to take thousands of times longer to run. Alternatively (a more efficient approach), if time permitted, one can use our optimization algorithm to generate several or tens of designs (starting with different random samples) and then re-apply the algorithm to the new candidate set formed by the several designs. This should result in a design with the best coverage among all generated.

Additional Points From Large Bins

Recall that we bin each dimension using equal-frequency bins at the extremes and equal-width bins in between. Let us call these the outer and inner bins, respectively. The proposed design gives good coverage of the inner bins (where the majority of compounds reside) but pays less attention to the outer bins. The outer bins tend to be very wide and it is not possible to get good coverage. Many investigators (e.g., Cummins et al., 1996, Higgs et al., 1997 and Menard et al., 1998) would just ignore the outlying compounds. The proposed design selects a small number of design points from the outer bins.

One way to get a better coverage of the outer bins is simply to add more design points there but keep the number of design points in the inner bins unchanged. Under random selection, the number of compounds in an outer bin is roughly proportional to its number of candidate compounds. Alternatively, one can make the first and last bin percentages depend on the number of bins (or the number of compounds required for the design). Therefore, with n bins one could have first $1/n$ proportion of candidate compounds in the first bin, and similarly the last bin. If n is so large that $1/n$ does not get all the outliers, one needs several $1/n$ bins to catch them. This approach will increase the number of design points in the outer bins relative to the inner bins, giving the two types of bins a share of the design points proportional to the number of candidate compounds.

Higher D-Subspaces

Including subspaces of 4-D and higher will usually not be practical. Chemists believe that two molecules must have fairly close values of all critical descriptors for similar biological activity (McFarland and Gans 1986). This means that bins have to be small if one molecule from a bin is to represent the rest. Yet, even with 10 bins per dimension, which is probably too few, there are 10,000 cells per 4-D subspace. Clearly, we would need to choose at least this many molecules if the experimental design is to cover every cell. Another problem related to 4-D and higher subspaces is that there can be too many subspaces to consider. On the other hand, if the important dimensions are identified (from initial screening), and if focused regions are desired, then subsequent designs focusing on high dimensions can be effective.

Instead of using low-dimensional subspaces formed by descriptors, one can consider using low-dimensional subspaces formed by linear combinations of the descriptors (e.g., principal components).

To deal with a large number of subspaces, one can consider random selection of subspaces as a compromise. One should repeat the random selection several times and then choose the design that gives good coverage on all the random selections.

Weighting of Subspaces

In practice, the molecular features and their interactions (and hence the corresponding subspaces) important for biological activity are not known at initial screening. All subspaces are assigned equal weights in the measure of lack of uniformity (U) in Section 3.4.2. However, if some subspaces are known to be important, then U can be easily modified to incorporate different weights (i.e., weighted U_s).

Adjusting for Number of Empty Cells

The indicator variables $c_{si}(X_c)$ in Equation (2) in Section 3.4.2 provide the target numbers of points per cell in the UCC criterion. These targets can be modified to adjust for the number of empty cells in a subspace. For example, suppose that 0% and 50% of cells in Subspaces A and B, respectively, are empty. We can set the target for a cell in Subspace A to 1 (expecting one design point per cell) and in Subspace B to 0, 1 or 2 if there are no candidate points, one point, or at least two points, respectively.

5.2.3 Analysis

The CB analysis is a newly developed statistical method. It is a five-stage automated process that requires extensive computing power. There are many opportunities to make the CB analysis algorithm more efficient as well as to further enhance the prediction accuracy. We are currently investigating these opportunities.

Program Algorithm in C

The analysis software (written in SAS code) was aimed to test the new methods and did not focus on efficiency in dealing with large data sets with many descriptors. We plan to implement the CB analysis algorithm using C code which should run hundreds of times faster than the current software. For a very large data set (e.g., millions of compounds with many descriptors), a fast algorithm to store and evaluate billions of cells is needed.

Re-sizing Cells

The current cell re-sizing method re-sizes each cell by trimming off the borders with no active compounds. This is done by setting the new cell boundaries to the range of the descriptor values of the active compounds in the cell. The more active compounds available in the cell, the better the boundaries can be located. Using this simple re-sizing method alone, without the shifted cells method, may leave holes within an active region. The shifted and unshifted cells overlap each other, thus reducing or minimizing possible holes in an active region. Other cell re-sizing methods will be investigated. One simple way of avoiding holes would be to resize only if some inactive compounds can be cut away.

Multiple Ranking Criteria

One can use a combination of different ranking criteria (e.g., the P and H_{L95} values) to define a 'common' cut-off or even to select multiple sets of good cells (different criteria may select different types of active cells).

Tuning Parameters

Tuning parameters such as the number of cells per subspace, the minimum number of active compounds required for a preliminary good cell, choosing the cut-off for the good cells, and more sensitive weighting functions for scoring cells and hence compounds, will be studied. The number of cells per subspace and the minimum number of active compounds required for a preliminary good cell can be evaluated and self-adjusted during the preliminary identification of good cells using the training data.

Transferable Methods

Of existing data-mining methods, classification and regression trees (recursive partitioning) have had the most success (e.g., Hawkins et al., 1997, Jones-Hertzog et al. 2000). Although these methods are generally well suited to modeling of local behavior, they otherwise pay little attention to the complexities of drug discovery data. For example, most are driven by criteria aimed at good overall prediction accuracy, criteria that are dominated by the overwhelming majority of inactive compounds. Adjusting these methods to aim for high hit rates for the relatively few compounds chosen for further screening would bring them closer to the real goal. Some of the methods (e.g., the ranking criteria) developed in this thesis seem transferable to these existing methods.

Multiple Trees

Using multiple trees (e.g., bootstrap the data) and combining their predictions (bagging) can result in better prediction than using a single tree. On the other hand, the CB analysis is not sensitive to small changes in the data and hence multiple CB models are not expected to get significant improvement in prediction. The performance of multiple trees versus that of the CB analysis will be studied.

Diverse Compound Selection

The proposed compound selection methods focus on finding the most promising compounds that have high probability being active. They pay little attention to the diversity of the compounds selected. As the goal is aimed to find different classes of active compounds, new methods should incorporate diversity into the selection criterion (e.g., favor compounds appeared in different groups of cells).

Cluster Significance Analysis (CSA) and Clustering Analysis

CSA assumes that there is only one class of active compounds and is not designed for multiple mechanisms (Section 4.3.1). CSA could be improved by clustering the active compounds first and then computing distances within active clusters.

Size of Training set

Ideally, one would use all available data to derive the prediction model and collect new data to validate the results. Often additional data are not collected and different methods are evaluated by dividing the original data set into training and validation sets. The relative performances of different methods may vary with different training data sample sizes. Our experience tells us that RP needs 2,500 to 5,000 compounds to work and performs best for very large data sets. Impact of the size of training set on CB analysis and RP analysis will be studied.

Impact of Design on Recursive Partitioning

To evaluate the impact of different designs on analysis (or the performance of an analysis), the original data set is divided into training and validation sets (see Section 4.2.4). However, use of uniform coverage designs or random designs for the training data will result in different validation sets. Uniform coverage designs tend to select roughly the same number of compounds from both crowded and sparse regions and might not leave compounds in the sparse regions for validation. One strategy is to first split the data into training and validation sets and then take a sample (uniform coverage design or random design) from the training set for modeling. Therefore, the same validation set will be used for evaluation and some compounds from the training set will not be used. The main problem is that a bigger data set is needed to get a reasonable number of actives at all stages. We are currently searching for data sets with more than 200,000 compounds with both numerical descriptors and assay results.

The impact of uniform coverage designs and simple random designs on the performance of RP was investigated. The original data set was divided into training and validation sets using one of the two designs for the training data. Preliminary research results indicate that uniform coverage designs led to higher hit rate for the top ranked compounds but lower hit rate when the number of compounds selected was large. The uniform coverage designs might select too few active compounds for some active regions, making these regions difficult for RP to detect. Further investigation on the impact of using uniform coverage designs for the training data on the performance of the RP analysis will be carried out.

Bibliography

AMSTAT news (2000). The Membership Magazine of the American Statistical Association. December 2000. *AMSTAT News*, Issue 282, p41.

An Overview of the Drug Discovery and Development Process (2001). The Medicines Malaria Venture. http://www.malariamedicines.org/overview.htm

Appleby, J. (1999). Drugmakers Fight Back as Patents Near Expiration. *USA Today*, November 26, 1999. Pg. 4B. http://www.intrchg.com/News/drugmakers_fight_back_as_patents.htm

Bayley, M.J. and Willett, P. (1999). Binning schemes for partition-based compound selection. *Journal of Molecular Graphics and Modeling* 17, 10-18.

Box, G.E.P., Hunter, W.G., and Hunter, J.S. (1978). Statistics for Experimenters, New York: Wiley.

Breiman, L., Friedman, L., Stone, C.J. and Olshen, R.A. (1984). Classification and Regression Trees. Chapman and Hall.

Burden, F.R. (1989). Molecular Identification Number for Substructure Searches. *Journal of Chemical Information and Computer Sciences* 29, 225-227.

Clark, L.A. and Pregibon, D. (1992). Tree-Based Models, in Statistical Models in S. J.M. Chambers, and T.J.Hastie, eds. CRC Press, Boca Raton, Florida.

Cook, R.D. and Nachtsheim, C.J. (1980). A Comparison of Algorithms for Constructing Exact D-optimal Designs. *Technometrics*, 22, 315-324.

Cox, D.R. and Reid, N. (2000). The Theory of the Design of Experiments. Boca Raton: Chapman and Hall/CRC.

Cummins, D.J., Andrews, C.W., Bentley, J.A. and Cory M. (1996). Molecular Diversity in Chemical Databases: Comparison of Medicinal Chemistry Knowledge Bases and Databases of Commercially Available Compounds. *Journal of Chemical Information and Computer Sciences,* 36, 750-763.

Dalal, S.R., and Mallows, C.L. (1998). Factor-Covering Designs for Testing Software. *Technometrics*, 40, 234-243.

Discovery New Drugs (1999). How Things Work, Science and Technology. *Research/Penn State*, Volume 20, Number 2. http://www.research.psu.edu/rps/may99/newdrugs.html

Doehlert, D.H. (1970). Uniform Shell Designs. *Applied Statistics,* 19, 231-239.

Dykstra, O., Jr. (1971). The Augmentation of Experimental Data to Maximize $|X'X|$. *Technometrics*, 13, 682-688.

Fang, K.T., Wang, Y., and Bentler, P.M. (1994). Some Applications of Number-Theoretic Methods in Statistics. *Statistical Science*, 9, 416-428.

Fedorov, V.V. (1972). Theory of Optimal Experiments, translated and edited by W.J. Studden and E.M. Klimko. New York: Academic Press.

Ghangurde, A. (1997). Glaxo May Emerge As Sourcing Base As Zantac Goes Off Patent. *Indian Express Newspapers*, July 7, 1997. http://www.expressindia.com/fe/daily/19970707/18855573.html

Hawkins, D.M. (1999). Formal Inference-based Recursive Modeling. Release 2.2 University of Minnesota, St. Paul, MN.

Hawkins, D.M. and Kass, G.V. (1982). Automatic Interaction Detection. In *Topics in Applied Multivariate Analysis*; Hawkins, D.M., Ed., Cambridge University Press, UK. pp 269-302.

Hawkins, D.M., Young, S.S., and Rusinko, A. III, (1997). Analysis of a Large Structure-Activity Data Set Using Recursive Partitioning. *Quantitative Structure-Activity Relationship* 16, 296-302.

Hastie, T.J. and Tibshirani, R.J. (1990). Generalized Additive Models. Chapman and Hall, New York.

Higgs, R.E., Bemis, K.G., Watson, I.A. and Wike, J.H. (1997). Experimental Designs for Selecting Molecules from Large Chemical Databases. *Journal of Chemical Information and Computer Sciences* 37, 861-870.

Johnson, M.E., Moore, L.M. and Ylvisaker, D. (1990). Minimax and Maximin Distance Designs. *Journal of Statistical Planning and Inference* 26, 131-148.

Jones-Hertzog, D.K., Mukhopadhyay, P., Keefer, C., and Young, S.S. (2000). Use of Recursive Patitioning in the Sequential Screening of G-protein Coupled Receptors. *Journal of Pharmacological and Toxicological Methods*, 42, 207-215.

Kennard, R.W. and Stone, L.A. (1969). Computer Aided Design of Experiments. *Technometrics* 11, 137-148.

King, R. D., Muggleton, S., Lewis, R.A., and Sternberg, M. J., (1992). Drug Design by Machine Learning: The Use of Inductive Logic Programming to Model the Structure-Activity Relationships of Trimethoprim Analogues Binding to Dihydrofolate Reductase. *Proceeds of the National Academy of Sciences* 89, 11322-11326.

Klopman, G., (1984). Artificial Intelligence Approach to Structure-Activity Studies. Computer automated structure evaluation of biological activity of organic molecules. *Journal of the American Chemical Society* 106, 7315-7321.

Lam, R.L.H., Welch, W.J., and Young, S.S. (2001a). Uniform Coverage Designs for Molecule Selection. paper submitted and revised for *Technometrics. (This is essentially Chapter 3.)*

Lam, R.L.H., Welch, W.J., and Young, S.S. (2001b). Cell-Based Analysis for Large Chemical Databases. paper submitted to *Technometrics. (This is essentially Chapter 4.)*

Levy, M.D. (2000). The drug discovery and development process in the new millennium. *Journal of the Canadian Association of Gastroenterology*, Vol. 14, Issue 7. http://www.pulsus.com/GASTRO/14_07/levy_ed.htm

McFarland, J. W. and Gans, D.J. (1986). On the Significance of Clusters in the Graphical Display of Structure-Activity Data. *Journal of Medicinal Chemistry*, 29, 505-514.

McKay, M.D., Conover, W.J. and Beckman, R.J. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* 21, 239-245.

Menard, P.R., Mason, J.S., Morize, I. and Bauerschmidt, S. (1998). Chemistry Space Metrics in Diversity Analysis, Library Design, and Compound Selection. *Journal of Chemical Information and Computer Sciences*, 38, 1204-1213.

Miller, R.G. (1981). Simultaneous Statistical Inference. 2nd Ed. Springer-Verlag, New York

Mitchell, T.J. (1974). An algorithm for the Construction of D-optimal Experimental Designs. *Technometrics* 16, 203-210.

Morgan, J.A. and Sonquist, J.N. (1963). Problems in the Analysis of Survey Data and a Proposal. *Journal of the American Statistical Association*, 58, 415-434.

Morris, M.D. and Mitchell, T.J. (1995). Exploratory Designs for Computational experiments. *Journal of Statistical Planning and Inference*, 43, 381-402.

Morris, M.D., Mitchell, T.J. and Ylvisaker, D. (1993). Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction. *Technometrics* 35, 243-255.

Owen, A.B. (1992). Orthogonal Arrays for Computer Experiments, Integration, and Visualization. *Statistica Sinica*, 2, 439-452.

Pearlman, R.S. and Smith, K.M. (1998). Novel software tools for chemical diversity. *Perspect. Drug Discovery Design* 09/10/11 339-353.

Pearlman, R.S. and Smith, K.M. (1999) Metric Validation and the Receptor-Relevant Subspace Concept. *J. Chem. Inf. Comput. Sci.* 39, 28–35.

Rusinko, A, III, Farmen, M.W., Lambert, C.G., Brown, P.L., Young, S.S. (1999). Analysis of a large structure/biological activity data set using recursive partitioning. *Journal of Chemical Information and Computer Sciences*, 38, 1017-1026.

Scott, D. and Wand, M.P. (1991). Feasibility of Multivariate Density Estimates. *Biometrika*, 78, 197-205

Tang, B. (1993). Orthogonal Array-Based Latin Hypercubes. *Journal of the American Statistical Association*, 88, 1392-1397.

The Promise of Biotechnology and Genetic Research (2000). *The Pharmaceutical Research and Manufacturers of America (PhRMA)* http://www.phrma.org/publications/documents/backgrounders//2000-12-12.191.phtml

Tobias, R. (1995). SAS QC Software. *Volume 1: Usage and Reference*, SAS Institute Inc., Cary, N.C., 657-728.

Venables, W.N. and Ripley, B.D. (1999). Modern Applied Statistics with S-PLUS. 3rd Ed., New York, Springer.

Wynn, H.P. (1972). Results in the theory and construction of D-optimal experimental designs. *Journal of the Royal Statistical Society B*, 34, 133-147.

Young, S.S., Farmen, M. and Rusinko, A. (1996). Random Versus Rational - Which is Better for General Compound Screening? *Network Science*, www.netsci.org/Science/Screening/feature09.html

Young, S.S. and Hawkins, D.M. (1998). Using Recursive Partitioning to Analyze a Large SAR Data Set. *Structure-Activity Relationship and Quant. Structure-Activity Relationship*, 8, 183-193.

Zemroch, P.J. (1986). Cluster Analysis as an Experimental Design Generator, With Application to Gasoline Blending Experiments. *Technometrics* 28, 39-49.