1 Introduction: Molecules, Machines and Men

The research described in this thesis focused on “interactive evolutionary algorithms and data mining for drug design”. That may sound impressive, but what does it mean? The purpose of this introduction is to ensure that people who do not know much about interactive evolutionary algorithms and data mining will have a pretty good idea what those are after reading the next few pages, and that people who are already familiar with the field will get a more intimate acquaintance with the problems we are trying to solve, and the perspective that we have. Interactive evolution and data mining are really just formal names and procedures for activities all of us already do in daily life: whenever you redecorate your room, you're performing a kind of interactive evolution, by asking yourself whether the room would look better if you painted it blue or soft yellow or added a large portrait of Barack Obama. Whenever you are browsing the newspaper, looking for interesting articles, you are doing a form of data mining. Science is often just common sense formalized. This thesis will discuss interactive evolutionary algorithms for drug design, as well as data mining, but these are merely sophisticated tools to achieve our goal: to find new or better drug molecules.

The research I have done was a collaboration between the department of Medicinal Chemistry, which focuses on developing biologically active molecules, and the Algorithms group of computer science, which investigates the use of computer methods to solve real-world problems. The subject of my research was therefore how to use computers (machines) to design drugs (molecules), which are discussed in the next two sections of this introduction. However, while creating the computer programs, we found out that by merely focusing on software and molecular structures we were neglecting something crucial: the scientists themselves. Creating a computer program that should be used by people required us to pay attention to how people think, and how a computer program can be made intuitive and easy to use. We also found that it was extremely useful to complement the molecule-generating capabilities of the computer with the experience and pattern-recognition ability of people. We therefore

also dedicated a section to the third factor in this research, the “interactive” in interactive evolution. After these sections on molecules, machines and man, there will be an introduction to some of the terms used in this thesis. Finally, we will discuss the aims of this thesis and give an outline of the chapters to follow.

Molecules, machines and man

Molecules

The human body viewed at normal scale already seems complex. However, when one zooms in to the microscopic level of cells and proteins, it becomes even more fascinating, for only on that scale the true complexity of our existence is revealed. The human body contains about one hundred trillion cells of over 200 distinct cell types, with 20,000 genes, which can produce at least as many proteins. It also contains a large variety of hormones, fatty acids, and other small organic compounds which help the cells and organs communicate and cooperate with each other in many ways, adjusting the activity of the body to whatever is needed in the circumstances in which we live.

Next to admiring the beauty and complexity of the workings of life, and satisfying our curiosity on how things work, there is also a very practical reason to strive to understand the human body: fighting disease. If we know how the human body works when it is healthy, and what happens when it falls ill, it should be easier to find a proper remedy for a disease. And in the end, it is not the understanding, but the action, the resulting medicine, that is important. However, even if one knows what is wrong in the body, the problem may still not be easy to correct.

Except from some cases in which the “diseased part” of the body can simply be removed (surgery), the most effective way to treat diseases is by administering drugs, which contain many billions of molecules of a specific compound. These drug molecules bind to biological molecules (usually proteins), either activating them or inhibiting them. This changes the behaviour of the protein, and thereby the behaviour of the cell, ultimately affecting the organ or even the whole body. For example, aspirin works by inhibiting cyclooxygenase, an enzyme which produces prostaglandins, compounds that cause pain. When someone takes aspirin, aspirin molecules diffuse through the gastro-intestinal wall and enter the bloodstream, where they block cyclooxygenase. With a reduced number of active cyclooxygenase enzymes which create pain-causing prostaglandins, less prostaglandins are produced, and so the pain is alleviated. By targeting the right step in biological processes, drugs can “reset” the

body to a healthier state, or at least alleviate the symptoms of a disease.

There are however still many diseases which cannot be treated well with current medicine, for example AIDS, many forms of cancer, and Alzheimer's disease. Finding drugs for these and other diseases is difficult for several reasons. First of all, the mechanism of a disease is often not clear, so it is not always known which protein to target. The second problem is that even if a good target protein is found, a molecule must be developed which binds to it effectively. Also, these molecules must be able to get to the right place in the body and not be metabolized or excreted before they can reach the diseased area. And finally, the molecules should not interact strongly with other biological molecules, which would cause harmful side effects. Finding a molecule that both interacts effectively with the target *and* has favourable "ADME-tox" properties (absorption, distribution, metabolism, elimination, toxicity) is a very difficult and time-consuming process: it costs on average over 800 million dollars and 12 years of development time to bring a drug to the market[1].

Our goal in this project was therefore to investigate how we could help drug discovery become faster or better.

Machines

Finding new drugs for diseases is the 'why' of this project; let us now turn to the 'how': how can we improve the drug discovery process? In the past three decades, various methods have been developed to improve or speed up the drug design process: so-called "rational design", high-throughput screening, combinatorial chemistry, and, more recently, systems biology and bioinformatics. These methods, diverse as they are, have one striking common denominator: they all use computers.

Even while computers often only do "simple things fast", they can increase efficiency in scientific research tremendously. For example, when I was a MSc student, if I wanted to find information on a certain compound, I needed to manually search multiple annual editions of the chemical abstracts service (thick books), before I could jot down the numbers of the abstracts, which had to be looked up in *another* series of heavy books. Of course, if the abstract suggested that the article would be useful, I still had to locate the attic section and/or shelf where that specific edition of the journal was located, and then go to the copier to make a copy for myself. The process could take hours. Nowadays, using internet and search machines, one can find and print articles about a particular compound or topic in seconds or minutes. Next to doing fast calculations (allowing for example fast elucidation and visualisations of protein structure), and controlling complicated machinery (such as in high-throughput screening), information storage and distribution is probably the greatest benefit of IT.

For example, electronic lab journals allow companies to find out about already performed experiments much more easily than the "classical" method of finding a synthesis in a stack of paper lab journals.

However, what could we add to the already impressive array of computational techniques for aiding drug discovery? In this research we have focused on the possibilities of two fields of computer science: evolutionary algorithms and data mining.

Evolutionary algorithms address one of the traditional problems of computers: computers can be programmed to do anything that involves any sequence of fixed actions – but sometimes it is not known which actions are necessary to achieve the desired result. Finding a molecule that binds to a certain protein is a problem of this type: the goal is known, but there is no "procedure" that will systematically and unambiguously lead to the desired molecule. In practice, intelligent trial and error is needed. Evolution works this way too. First, it produces a large number of solutions (animals/plants) to certain problems (environments). Then, the best of these solutions procreate (are copied, changed/adapted and combined) to produce even better solutions in the next generation. Inventions and machines change over the generations just like organisms, and computer programs can simulate this by changing and combining the best designs of a collection of designs. In our case, those designs are molecules.

Data mining is another powerful technique, useful in cases where the programmer does not yet know what the "rules" of a system are, for example which factors in one's diet increase or decrease the risk of heart disease. By statistically analysing large amounts of data, data mining can unravel patterns in masses of data which may be hidden for the human eye. For example, software has been developed that correctly picked out the 10 known fraudsters (and about a dozen new suspects) in a database of the online auction site eBay – totalling one million transactions and 66000 users, far too much data for a human to analyze.[2,3] Likewise, data mining could give insights in hidden patterns in databases of molecules or drugs.

Looking for ways to help drug design, we therefore wondered how we could use data mining and evolutionary algorithms to our advantage.

Man

When one develops software that will also be used by others, a third factor needs to be taken into account, next to problem knowledge and computer knowledge: people. In my research this is more important than in day-to-day science, where for many scientists and programmers the existence of people almost seems an afterthought. Scientific papers are usually written in the passive voice, ranging from the standard "10

ml NaOH (1M) was added to the mixture" to the slightly deceptive "it was hypothesized that…", as if a hypothesis objectively and unambiguously follows from certain facts or experimental results. While perhaps scientists *should* behave objectively and perfectly rational, scientists are people, and people are not completely objective or rational, even though they may try. Therefore, if something needs to be used by humans, even if those humans are scientists, it is not sufficient that it is objectively and scientifically functional. And this is also true for software. Even a potentially useful computer program may not be used if people can't find the time or courage to read 500-page manuals to learn how to navigate through cumbersome, illogical menus. It was therefore important for us to pay attention to how people think, and how we could adapt the software to make it easier to use.

On the positive side, it would be wrong to see humans merely as imperfect reasoning machines. Humans have evolved in nature, where there is usually lack of useful information combined with a huge amount of useless information that obscures the useful information there might be, where there are urgent problems with not enough time to calculate all odds and all possible ways out, and where an incredible amount of knowledge is required to achieve even the most modest results – even walking up stairs is something most programmers dread to program robots for. Humans are far superior to computers in detecting new patterns, making connections between pieces of information, and thinking "out of the box" to solve a problem. Humans can easily solve many problems which baffle the most advanced computers, for example, recognizing a face even if it is seen from the side, understanding words even if they're spoken in dialect, or walking through a house without bumping against walls or furniture.

It would therefore be ideal if we could not only use the capabilities of the computer, but let the talents of the human/scientist complement these. However, combining humans and computers is not easy to do right. The first main problem is that to be of any kind of use, software must be user-friendly – software that cannot be understood by the user will not be used, even if it has tremendous capacities. Second, what things can or should we delegate to the computer, and what things can we ask of a human user? And can we close the gap so that there can be useful collaboration?

The third issue we had to pay attention to in this research was therefore how to effectively make use of human-computer collaboration. Computers can make calculations of molecule properties quickly, while chemists have lots of experience and intuition on which molecules are drug-like and which molecules can and cannot be synthesized. Yet any cooperation between man and computer can only occur if the software is sufficiently intuitive and user-friendly. The first word processor I used,

Symphony, required the user to remember the key combination of <ALT>-<F1>-<B><A><B> to put anything in boldface, but such an interface would nowadays only discourage use. The last of the three questions is therefore how to design our chemical software in such a way as to ensure it will not only be useful, but also that it will be used.

Introduction to some of the terms and concepts used in our research

A number of computer science and cheminformatics terms will occur throughout this thesis, and while most of them will be explained in more formal terms in the following chapters, it may be useful for reader comprehension to clarify some of the most important concepts here.

Interactive Evolutionary Algorithm

One of the main aspects of evolution is selection, sometimes called "survival of the fittest". In evolutionary algorithms we also want the best solutions to survive and procreate, but to do that we have to determine what we mean by "best" or "fittest". Does "fittest" mean the strongest construction? The smallest molecule? The circuit board that gets the job done with least components? Or the circuit board that consumes least energy? Sometimes the fitness of a design can be calculated easily and objectively by a so-called "fitness function" which takes the organism/solution as input and returns a number that indicates its quality. Other times, though, the quality of a solution is difficult to calculate. For example, the ideal interior design of a room will depend on the taste of the human occupant.

Cases in which there is no objective way to calculate fitness are however not impossible to solve. Evolutionary algorithms can work if there exists *any* method to assign relative quality to solutions, and it is perfectly possible to have a human being as the "fitness function". That means that a human scores solutions or selects the ones he or she considers best. An evolutionary algorithm that uses a human to evaluate solutions is called an interactive evolutionary algorithm or interactive evolutionary computation (IEC). IECs have been used in many applications, varying from face image generation to help an eye-witness reconstruct the face of her attacker,[4] geophysics in which experts can distinguish realistic from unrealistic earth layer patterns, to helping people find better settings for their hearing aid.[5] Since interactive evolutionary algorithms can use both explicit and implicit/subconscious knowledge of

drug design present in human medicinal chemists, it also seemed a promising approach for our research.

Data mining

Governments, companies, universities and many other organizations nowadays have large databases which house enormous amounts of data. Such data is useful in its own right (for example, checking how much money your bank account contains), but these databases also bring the promise that one can discover patterns and laws in the data, much like Kepler discovered the laws of planetary motion from his astronomical data. However, most databases are so vast that it would be hard or impossible for a human to find laws and patterns. For that reason, many scientists are working on techniques collectively called "data mining", which means that they develop software that can automatically find relationships between data or parameters. Usually data mining is performed on database tables, for example, whether there is a correlation between the education and the income of a person, and if yes, what it is and how strong it is, but it can be applied to any collection of data. For example, data mining also can handle a "shopping basket" problem in which a supermarket wants to find out whether people usually buy product X with product Y (such as bread and peanut butter). In this thesis, our main investigation of data mining is described in chapter 3, while chapter 4 and especially chapter 5 discuss how we used data mining to improve our main evolutionary algorithm.

Docking

Docking is a term used for computer simulations of the interaction between small molecules and proteins. Small molecules such as drugs influence the behaviour of proteins by crawling into a "sensitive" place in the interior of the protein, much like a key enters a lock or a hand fits into a glove. Similar to the docking of ships in a harbour, a "docking program" will attempt to find the best fit of a small molecule into an enzyme or receptor. However, docking is a difficult problem, and many different docking programs have been developed, such as GOLD, FlexX, DOCK and Glide,[6] each having its own strengths and weaknesses. For drug design, the ideal is to predict how well a drug candidate would bind to a receptor, so one could select the most promising leads from a large library of compounds without having to perform expensive syntheses and biological testing. However, docking programs are yet far from reliable for finding such quantitative binding strengths, since the exact strengths of electrostatic interactions and hydrogen bonds between ligand atoms and the amino acids in a protein are unknown, and most docking programs cannot simulate how a

protein can mould itself around a ligand to improve binding. However, docking programs can often indicate how a molecule would fit into a protein, and despite their flaws they are currently the most reliable methods to theoretically compare binding affinities of a wide variety of small molecules. We used docking for the research in chapter 6, as despite its imperfections, docking is the best simulation of a `protein like`-system currently available.

Aims of this thesis

The aim of the research described in this thesis is to use evolutionary algorithms and data mining to help find new drugs.
For this purpose, we have:
-developed an internal representation of molecules and a set of mutations aimed to reach all possible molecules in chemical space.
-created a user interface that allows chemists to give input and feedback to the evolutionary algorithm efficiently and easily.
-mined large molecule databases to find frequent and infrequent substructures that can be used to design new molecules.

We also tested out the resulting interactive evolutionary algorithm in collaboration with the medicinal chemists at our laboratory. A set of compounds generated by the evolutionary algorithm was examined by the chemists, who selected the molecules they deemed most interesting and adjusted them for ease of synthesis. Subsequently, these compounds were synthesized to assess whether the methods we developed could indeed be used to find new biologically active molecules.

Outline of this thesis

This thesis will open with a review on the applications of evolutionary algorithms in drug design (chapter 2). Chapter 3 focuses on the question on how well current chemistry covers total chemical space –what is the real diversity of compounds? The answer is perhaps somewhat sobering (the term “chemical clichés” in the title of this chapter was coined for a reason), however we also indicate ways to use the data gathered to create more novel molecule scaffolds. Chapter 4 will discuss the Molecule Evoluator, a computer program we developed that uses an interactive evolutionary

algorithm to create novel chemical compounds by using both computing power and chemist's intuition. In chapter 5, we show that the results of the Molecule Evoluator can be improved by combining the evolutionary algorithm with the technique of data mining, and show how the parameters of the evolutionary algorithm can be set to reflect the results of our data mining – which is not as straightforward as it may seem! Chapter 6 tackles the question whether atom- or fragment-based approaches are preferable for evolutionary algorithms in molecule design, by using docking to approximate the fitness of the compounds generated by the Molecule Evoluator. The part dedicated to our investigations closes with chapter 7, which looks into some real-world results: creating novel biologically active compounds which have been discovered by collaboration between medicinal chemists and the Molecule Evoluator. Finally, chapter 8 closes this thesis with the conclusions and my perspectives on the future of computers in drug design.

> "We have so much time and so little to do. Strike that, reverse it."
> - Willy Wonka, *Charlie and the Chocolate Factory* (Roald Dahl)

I hope you're set for the journey. Let's get started.

References

[1] DiMasi, J.A., Hansen, R.W., and Grabowski, H.G. The price of innovation: new estimates of drug development costs. *Journal of Health Economics* **2003**, *22*, 151-185.

[2] Pandit, S., Chau, D.H., Wang, S., and Faloutsos, C. NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks. *WWW 2007* **2007**.

[3] Simonite, T. Network analysis spots online-auction fraudsters. *New Scientist* (online edition) **2006**, December 6th.

[4] Marks, P. How to recall the face that fits. *New Scientist* **2005**, March 19th, p24.

[5] Takagi, H. Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation. *Proceedings of the IEEE* **2001**, *89*, 1275 – 1296.

[6] Klebe, G. Virtual ligand screening: strategies, perspectives and limitations. *Drug Discovery Today* **2006**, *11*, 580-594.