Bioclipse: Integration of Data and Software in the Life Sciences

OLA SPJUTH
Dissertation presented at Uppsala University to be publicly examined in B42, Uppsala Biomedical Center (BMC), Husargatan 3, Uppsala, Friday, November 27, 2009 at 13:15 for the degree of Doctor of Philosophy in Pharmacy. The examination will be conducted in English.

Abstract

New high throughput experimental techniques have turned the life sciences into a data-intensive field. Scientists are faced with new types of problems, such as managing voluminous sources of information, integrating heterogeneous data, and applying the proper analysis algorithms; all to end up with reliable conclusions. These challenges call for an infrastructure of algorithms and technologies to supply researchers with the tools and methods necessary to maximize the usefulness of the data. eScience has emerged as a promising technology to take on these challenges, and denotes integrated science carried out in highly distributed network environments, or science that makes use of large data sets and requires high performance computing resources.

In this thesis I present standards, exchange formats, algorithms, and software implementations for empowering researchers in the life sciences with the tools of eScience. The work is centered around Bioclipse - an extensible workbench developed in the frame of this thesis - which provides users with instruments for carrying out integrated research and where technical details are hidden under simple graphical interfaces. Bioclipse is a Rich Client that takes full advantage of the many offerings of eScience, such as networked databases and online services. The benefits of mixing local and remote software in a unifying platform are demonstrated with an integrated approach for predicting metabolic sites in chemical structures. To overcome the limitations of the commonly used technologies for interacting with networked services, I also present a new technology using the XMPP protocol. This enables service discovery and asynchronous communication between the client and server, which is ideal for long-running analyses.

To maximize the usefulness of the available data there is a need for standards, ontologies, and exchange formats, in order to define what information should be captured and how it should be structured and exchanged. A novel format for exchanging QSAR data sets in a fully interoperable and reproducible form is presented, together with an implementation in Bioclipse that takes advantage of eScience components during the setup process.

Bioclipse has been well received by the scientific community, attracted a large group of international users and developers, and has been awarded three international prizes for its innovative character. With continued development, the project has a good chance of becoming an important component in a sustainable infrastructure for the life sciences.

Keywords: Bioclipse, integration, life sciences, bioinformatics, cheminformatics, chemoinformatics, eclipse, rich client, xmpp, qsar-ml, web service, standard, ontology

Ola Spjuth, Pharmaceutical Pharmacology, Box 591, Uppsala University, SE-75124 Uppsala, Sweden

© Ola Spjuth 2009

ISSN 1651-6192
urn:nbn:se:uu:diva-109305 (http://urn.kb.se/resolve?urn=nbn:se:uu:diva-109305)
List of Papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.


II Wagener, J., Spjuth, O., Willighagen, E.L., and Wikberg J.E.S. *XMPP for cloud computing in bioinformatics supporting discovery and invocation of asynchronous Web services*. BMC Bioinformatics 2009, 10:279

III Carlsson, L., Spjuth, O., Adams, S., Glen, R.C., and Boyer, S. *Use of historic metabolic biotransformation data as a means of anticipating metabolic sites using MetaPrint2D and Bioclipse*. Submitted.


Additional Papers


- Eklund, M., Spjuth, O., and Wikberg, J.E.S. *An eScience-Bayes strategy for analyzing omics data*. Submitted.

- Junaid, M., Lapins, M., Eklund, M., Spjuth, O., and Wikberg, J.E.S. *Proteochemometric modeling of the susceptibility of mutated variants of HIV-1 virus to nucleoside/nucleotide analog reverse transcriptase inhibitors*. Submitted.


Contents

1 Introduction .................................................. 7
   1.1 Bioinformatics ........................................ 8
   1.2 Cheminformatics ...................................... 9
   1.3 Drug discovery ....................................... 10
      1.3.1 Computer-aided drug discovery ................. 11
   1.4 Knowledge Management in the life sciences .......... 12
   1.5 Integrated science ................................... 12
   1.6 eScience ............................................ 14
      1.6.1 Service-orientation .............................. 15
      1.6.2 High Performance Computing .................... 16
      1.6.3 Open science .................................. 17
2 Aims of studies ........................................... 19
3 Methods .................................................. 21
   3.1 Software integration ................................ 21
      3.1.1 Integration of local software .................. 21
      3.1.2 Integration of services ........................ 23
      3.1.3 Rich Clients ................................... 25
   3.2 Data integration ..................................... 26
      3.2.1 Standards ...................................... 26
      3.2.2 Exchange formats ............................... 26
      3.2.3 Ontologies ..................................... 27
      3.2.4 Semantic technologies .......................... 27
      3.2.5 Model-driven development ...................... 28
   3.3 Drug Discovery ....................................... 29
      3.3.1 Structure-Activity Relationship ............... 29
      3.3.2 Circular fingerprints ........................... 30
4 Results and discussion ................................... 31
   4.1 A workbench for the life sciences (Paper I and V) ... 31
   4.2 Discoverable and asynchronous services (Paper II) .... 34
   4.3 Near-real time metabolic site predictions (Paper III) ... 36
   4.4 A new standard for QSAR (Paper IV) .................. 37
   4.5 A scripting language for the life sciences (Paper V) ... 38
5 Concluding remarks ....................................... 41
6 Acknowledgements ....................................... 43
Bibliography .............................................. 45
1. Introduction

The sequencing of the human genome [1, 2] was the beginning of a new era in the life sciences, an era characterized by an enormous increase in the amounts of data produced. New high throughput techniques have emerged, capable of generating data at a rate that was unthinkable only a few years ago. This summer the Wellcome Trust Sanger Institute surpassed 10 Terabases of cumulative genome sequence data, and has during the last year doubled the rate to now generate 400 Gigabases of sequence data per week (the equivalent of about seven human genomes). Massively parallel sequencing technologies [3] (often called "next-generation" sequencing) are the most voluminous data producers, contributing about a Terabyte a month to public archives [4].

Apart from sequencing, it is today possible to obtain data regarding gene expression, mutations, small molecules, and proteins on large scale using array techniques with tens of thousands of probes. Another example is the measuring of drug-target interactions in ultra high throughput screening laboratories, capable of producing over 100,000 data points per instrument and day.

The data explosion has forced companies and genome centers to be equipped with Petabytes of storage and thousands of computer cores to process the flow of incoming data. The speed of data production is constantly increasing, and companies like Complete Genomics envision sequencing 10,000 genomes in 2010 [5], when the international collaboration 1000 genomes project [6] is not even completed.

The new techniques open up many possibilities. One example is in diagnostics, where diseases can be identified for example by measuring the expression of a selected set of genes or proteins (biomarkers). Other examples include personalized treatments [7], identification of new drug targets [8], insights into the variation between individuals with regard to diseases [9], and how we in rational ways can design new drugs [10].

Life science has thus in the last decade become a data-intensive field, and large international organizations use the Internet as a collaborative platform and to make data publicly available. It is nowadays a common task for biologists to interact with several out of the hundreds publicly available repositories [11]. There is also an increasing diversity in the type of data stored, for example genomic sequences, protein structures, mutations, interactions, expression levels, and pathways. Researchers are required to handle and analyze the immense data amounts in order to convert them into accessible knowledge, which has made the life sciences highly dependent on fields such as knowledge management and information technology.
Figure 1.1: a) The central dogma of molecular biology. DNA is transcribed to RNA, which in turn is translated to Proteins that carry out functions. It is the goal of bioinformatics to handle and analyze information from these and related topics using information technology. b) Multiple sequence alignment is a technique to compare several sequences, such as DNA, RNA, or protein. The process aims at finding the result with the best overall similarity between the sequences. Image generated with ClustalX [12, 13]. c) Phylogenetic analysis can be used to study evolutionary relatedness by sequence similarity, here presented as a dendrogram. The two most similar sequences are joined in a cluster, and the process is repeated until all clusters are joined. Image from [14].

1.1 Bioinformatics

Bioinformatics is the application of information technology to the field of molecular biology, with the goal to handle, analyze, and visualize information and data from the central dogma; the transcription of DNA to mRNA and the translation of mRNA to proteins (see Figure 1.1(a)), and related fields. This encompasses such various tasks as management and provisioning of biological data, algorithms and tools to aid in problem solving, and analyses to answer biological questions. Over the past few decades, major advances in the field of molecular biology, coupled with advances in genomic technologies, have led to an explosive growth in the biological information generated by the scientific community. It is the quest of bioinformatics to create the infrastructure, technologies, and resources that enables analysis and reliable storage of these large data quantities.

Examples of major research areas in bioinformatics include comparative sequence analysis, gene finding, protein structure modeling and prediction, gene expression analyses, protein-protein interactions, genome-wide association studies, and the modeling of evolution (Figure 1.1(c)). To attack these problems it is becoming increasingly common to make use of computationally intensive techniques, including advanced
statistical analysis (such as machine learning, regression analysis, pattern recognition, and data mining) and high-end visualizations.

Bioinformatics is a comprehensive research field in the life sciences as information technology is more and more integrated into everyday research. One of the most commonly used methods is multiple sequence alignment (Figure 1.1(b)), which can be used to compare, search for similar, and compute the consensus or distance between two or more sequences. Another example is de novo prediction of protein structures from sequence, and while this is not yet feasible for other than small structures, homology modeling can be used to predict the structure of e.g. unknown proteins based on existing structures of another protein with a shared evolutionary history (protein sequences evolve more quickly than protein structures).

Data provisioning in Bioinformatics is commonly done over the Internet. The major genome databases Genbank [15], EMBL [16], and DDBJ [17] form the backbone of this infrastructure by proving the nucleotide sequences of sequenced species. Example of protein databases are Universal Protein Resource (UniProt) [18], the National Center for Biotechnology Information (NCBI) Protein database [19], and the Protein Data Bank (PDB) [20]. Many other databases provide specialized information, such as KEGG [21] and Reactome [22] which provides information about biological pathways, and ArrayExpress [23] which is a public repository for gene expression data.

1.2 Cheminformatics

Cheminformatics (also known as chemoinformatics or chemical informatics) focuses on the storage, and analysis of chemical structures. The field deals with both small molecules, such as metabolites and drugs, as well as larger structures, such as protein and crystal structures. Since biological processes at a low level are chemical interac-
tions, the field sometimes overlaps with bioinformatics, and the term biocheminformatics has been used to refer to work integrating the domains.

Common applications in cheminformatics include storage of chemical data in repositories (databases), the mining and screening of virtual molecular libraries (Figure 1.2(a)), and the modeling of chemical interactions in a target (Figure 1.2(b)). Most drug discovery applications also fall into the field of cheminformatics. Examples of public repositories in cheminformatics are: PubChem [26], ChemSpider [27], Drugbank [28], ChEMBL [29], and ChEBI [30].

1.3 Drug discovery

Drug discovery is the process by which drugs are discovered or designed. While most drugs historically have been identified as the active ingredient of traditional remedies or by serendipitous discoveries, new approaches to model and control disease at the molecular level deliver more rational approaches.

The drug discovery process generally comprises two major phases, the pre-clinical phase and the clinical phase (see Figure 1.3), with the former comprising testing in laboratory, and the latter testing on humans.

The pre-clinical phase starts with Target Identification, which is the process to identify a suitable target, e.g. a protein or a gene, which is associated with the disease and has functionality that can be affected by a drug in a positive fashion. Lead Identification (also known as Lead Discovery) is the process of finding chemicals that show a favorable interaction with the target, and which preferably do not interfere with other targets. A common technique for this is High Throughput Screening (HTS), where large libraries of synthetic or natural chemical substances are screened towards a target. The outcomes from the Lead Identification phase are called drug leads, and are compounds with such properties that they are deemed promising enough to take further in the drug discovery process.

Lead Optimization aims at optimizing the physiochemical properties of a drug lead to maximize its chances for success as a drug. This is generally done by varying the compound’s chemical structure to increase its activity against the target, reduce its activity against undesired targets, or to optimize its metabolic and pharmacokinetic properties. The Preclinical Development phase generally consists of toxicological studies, cell assays, and animal testing.

The second phase of drug discovery process (see Figure 1.3) is Clinical Development, which in itself consists of three phases. In phase 1 the drug is tested on healthy people.
volunteers, in phase 2 it is tested on people suffering from the disease (often in the order of hundreds) to produce a proof-of-concept, and in phase 3 on large patient populations (often thousands).

The process by which a single new drug is developed into a fully registered product is lengthy and extremely costly. The process can take as much as 15 years to complete, and is currently estimated to cost around a billion dollars per drug [31]. A major factor that contributes to these costs is that the vast majority of the drugs fail at some stage during the development process, and the few successful ones must pay for all these failures. In the last decades, the advent of molecular biology and genomic sciences has largely affected drug discovery. These technologies allows for a deeper understanding of the underlying molecular processes, and have increased the number of targets as well as improved the pre-clinical phases [32].

1.3.1 Computer-aided drug discovery

The recent advances in information technology have greatly affected drug discovery. Computer-aided drug design (CADD), where drugs are rationally designed using computational models, is to an increasing extent complementing traditional chemical research. Virtual Screening is one approach where computational filters are applied to screen virtual libraries of chemical structures, which can increase hit rates and substantially reduce the number of compounds needed to be assessed experimentally. Docking and structure-based methods are also used to design compounds de novo, and to improve lead properties when a target’s 3D structure is known.

Successful drug discovery requires enormous investments [33], and making the correct informed decisions as early as possible can avoid costly mistakes later in the development process. To increase productivity and reduce costs there is a desire to streamline the decision process by executing as much of the work as possible in silico [34, 35]. Computational models are becoming increasingly important for decision support, aiding with information such as predicted toxicity and solubility, and suggesting whether or not a certain compound should be discarded due to high probability of failure at later stages [36]. Such predictive models commonly requires integration of diverse and heterogeneous data types in order to build up a knowledge base to generate models from [34]. The increased complexity in the drug research needed to realize this has lately led to institutional changes promoting interdisciplinary research [32].

An important aspect of the drug optimization process is obtaining metabolic stability so that the candidate drug survives passage over the gut wall and through the liver, to reach the target tissue(s) unmetabolized in sufficient concentrations to achieve a pharmacological effect. Another aspect is to avoid and/or reduce formation of toxic metabolites from the drug candidate. Site-of-metabolism predictions aim at pinpointing the site in a chemical structure which is most likely to undergo metabolization, hence aiding with decision support in the drug optimization process.

Avoiding undesired drug-drug interactions is another important task where CADD plays an increasingly larger role. A drug can, for example, inhibit the metabolism of other drugs, which can have fatal consequences as the concentration of the other drug then might reach toxic levels. While drug metabolism occurs by the action of many metabolizing enzymes, the cytochrome P450 enzymes are extra important in this context. These enzymes exist in many isoforms in the body, and they are sub-
jected to genetic variations among the populations. Predictive genome-based models for enzyme inhibition by xenobiotics [37] is one method that probably can be entered into drug developmental pipelines in the not too distant future. Models for passage of drugs over biological barriers (e.g. drug passage over the blood-brain barrier and gastro-intestinal uptake) is another example.

1.4 Knowledge Management in the life sciences

It is important to not confuse data with knowledge - the former being merely a collection of facts that require interpretation in order to be converted into knowledge. The same set of data may afford alternative interpretations, and the concept of "provenance" denotes the history of how knowledge came to be. Information is somewhat loosely defined as 'useful data', with the potential to become knowledge. The three concepts together form a paradigm of gradual understanding: data - information - knowledge (Figure 1.4), and are key parts in the field of Knowledge Management [38, 39].

Knowledge Management is the process of systematically capturing, structuring, retaining, and reusing information to develop an understanding of how a particular system (e.g. a pathway, gene, or disease) works, and subsequently to convey this information meaningfully to other information systems (knowledge distribution) [40]. Traditionally, this interpretation of data was carried out manually by humans, but the data sets produced today require computers for their sheer size. This transition has led to a fundamental dependency on technological resources, such as networks and databases.

1.5 Integrated science

The emerge of high throughput technologies in the life sciences opens up many new exciting applications and holds promises to answer complex questions. However, to obtain new insights and knowledge, information needs to be transformed into executive summaries, which are brief enough for creative studies by a human researcher. Such summaries must for example include all the relevant elements in the study, the desired measurements, protocols, and preferably quality indicators [41]. Secondly, the information must be available in a format that can be fed into a computer program capable of using it. This has made it evident that standards must be established, defining what needs to be reported in a specific domain and how this data should be reported. Interoperable file formats for exchanging data are also needed so that information can be exchanged without data loss. Several standardization initiatives have emerged out of this need, proposing standards, ontologies (controlled vocabularies), and data exchange formats in various domains. Such initiatives includes the MGED consortium [42], FuGE [43], MIBBI [44], and HUPO PSI [45]. The process of standardization has advanced more in bioinformatics than cheminformatics, but the problems of incompatible file formats and standards have also been discussed in this field [46]. The importance of standardization is recognized by many actors of large scientific and economic impact [47, 48, 49, 50].
It is obvious that only by combining data will we be able to maximize its usefulness, and *Data Integration* is hence a central part of knowledge management. However, integrating data from different sources is not a trivial task, especially in the life sciences. This is partly because biological data and knowledge is very complex by nature, but also due to the fact that high-throughput instrumentation is a recent development for which the scientific community was not prepared [40]. This has left the life sciences with vast amounts of existing resources which are currently far from compatible on the data level.

There are several prominent projects in the life sciences that attack the problem of data integration. Sequence Retrieval System (SRS) [51] links sequence data among databases and provides a uniform search engine to query for sequences in a free text fashion. The MRS retrieval system [52] is a server that allows for very rapid cross-database searches of biological sequences. BioMart [53] is a website and Web service which allows for advanced querying of biological data sources which have been integrated in a data warehouse fashion, and has become a central point for accessing data from large data repositories such as Ensembl [54], UniProt [18], and Reactome [22]. The Distributed Annotation System (DAS) [55] is a decentralized system and protocol that allows multiple third-party annotators to provide annotations for sequences, which can be integrated by software clients on a need basis, and is used extensively in several major annotation projects.

The number of databases in the life sciences is much larger than in other disciplines [11] and the heterogeneity of data models and formats makes integration complex and demanding [56, 57]. Example of obstacles that must be tackled include different IDs for the same resources, incompatible database schemas, and free text annotations which severely hinders automatic data processing. Much information is currently only available in the scientific literature as part of published papers in scientific journals, and literature mining has emerged as an important research field [58] to bring this information into the open.

A key factor to achieve integrated sciences is that software components should be easy to integrate. The switching between applications is a tedious task in many analyses, and sometimes require more work than the actual analysis. In computer science, *Tight coupling* implies that the software components are connected in a way that makes it difficult to replace individual parts, reducing the possibility of reusing a component later in another context. Tightly coupled software is much easier and faster to develop, since modularity and generality need not be taken into account. *Loose coupling* implies that components are indeed separate, and can be reused in different contexts without much hassle. They interact when necessary, but remain
Figure 1.5: A conceptual view of eScience. Users interact over a network cloud with collaborators, software services, databases, and high performance computational resources. The transparent access to all these resources is a fundamental part of eScience. The cloud is usually the Internet, but could also be a private cloud.

uncoupled from each other. It is a much harder and more time-consuming process to develop loosely-coupled applications, but the benefits are large in the long run. Due to the fact that a scientific career largely depends on the number of published papers, much software is only developed up to the point of proof-of-concept, and then left in this state. Poor software design is unfortunately also common in the life sciences, mainly (and understandably) because scientists generally have greater expertise in the life sciences domain than in software design. Over the years this has left the field of bioinformatics with a plethora of tools that solve important scientific problems, but which are not easily integrated with other components.

1.6 eScience

eScience has emerged as a promising technology for tomorrow’s research. The term is used to describe the constantly increasing part of science that is computationally intensive and carried out in highly distributed network environments, or science that uses large data sets that require grid computing (see Figure 1.5). eScience is feasible due to the advancement of standardization processes, information technology, and the increasing awareness of interoperability with respect to data provisioning and software development. Many people believe that eScience will change the way science is conducted when information, tools, and computational resources are readily available to all researchers.
Life science is one of the fields where eScience will play an important role in the future. The field is distinguished by huge amounts of heterogeneous data accessible from an ever-growing number of data sources, and a wide range of software applications for analysis. However, much development is required in order for the life sciences to take full advantage of the many promises that eScience holds. New algorithms need to be designed that can make sense of the available data, and new statistics need to be developed as the field moves towards larger and larger data sets. Existing software tools must be made transparently accessible, and larger data sets also demand more computational power. A major problem with the current eScience technologies are that they are technically complex and require researchers with extensive computer experience. This is a major hindrance for making eScience available to the majority of researchers in the life sciences; a problem which will have to be remedied in order for eScience to reach its full potential.

1.6.1 Service-orientation

The increasing amount of available data, as well as the number and diversity of software applications, call for machine accessible interfaces in order for analysis to be effective. The term service-oriented architecture (SOA) denotes the utilization and interconnection of many, often geographically distributed, services to carry out a given task. SOA makes heavy use of Web services, which are software systems that provide machine-machine interoperable services over a network (commonly the Internet). Traditional web servers rely on web forms for invocation and only present results in HTML, where Web services can be invoked from any application. HTTP-based Web service technologies, like the W3C promoted Simple Object Access Protocol (SOAP) [59] (Figure 1.6) and REpresentational State Transfer (REST) [60], are today the most commonly used Web service technologies in bioinformatics.

SOA has a number of advantages over traditional architectures. It is cheap and standardized; there are no high costs associated with setting up a service, and there are standards to rely on that make them interoperable to a large extent. SOA is also a cheap means to obtain rights to use software without downloading a complete server setup, such as databases and application servers. The modularity of SOA enables services to be reused in different contexts without modification, and the sharing of hardware resources reduces the time these resources are idle. While SOA in larger institutions might mean sharing of internal capabilities, the major gain for academic research is the ability to reach out and deliver interoperable functionality as services to other researchers. Global service provisioning allows for integration of algorithms and methods that was previously extremely time-consuming, and in many cases practically impossible. It is common to make such services available over the Internet, practically forming a cloud of services and data. Generalizing services into other technologies than SOAP has made the term Cloud Service increasingly common, as well as the term Software as a Service (SaaS).

The Web service technology has been embraced in the life sciences [61, 62] and numerous service providers have established hundreds of services. The European Bioinformatics Institute (EBI) provides access to databases and common analysis tools in bioinformatics [63], including Soaplab which wraps the EMBOSS suite [64] as Web services. Entrez [65, 66, 67] is a federated search engine that allows for cross-
Figure 1.6: W3C Web services. Service Providers set up their services and publish metadata in separate documents, called WSDL files. The client reads the WSDL file and is then able to invoke the service.

database searches in bioscience with a Web service interface. Other large initiatives that promote Web services include the European initiative EMBRACE Network of Excellence [68], and the Cancer Biomedical Informatics Grid (caBIG) [69].

The advent of eScience has completely changed the face of software and data integration in the life sciences only in the last few years. Old implementations are converted into interoperable Web services, and most new developments are available via machine-accessible interfaces. This is changing how computational analysis and hypothesis generation in life science is conducted. The modularity of SOA implies that it is possible to select the needed components from the available services, and chain them together [70, 57].

1.6.2 High Performance Computing

High Performance Computing (HPC) denotes the infrastructure of hardware and software that provides scientists with the computational power, memory, and storage to carry out analyses that are not feasible on personal computers. The term includes parallel computing, distributed computation, GRID-computing, and other systems with large memory and storage. In the life sciences, HPC is used to analyze large quantities of data, perform high-end simulations, and enables studying of complex multi-domain problems. Examples of fields that make heavy use of HPC include molecular dynamics [71], large scale virtual screening [72], whole-genome analyses [73], and protein folding [74]. The handling of data from the new high throughput instruments also relies on HPC resources for processing and storage.

The term ‘Grid’ is commonly associated with high performance computing, but the term has grown to encompass more topics, for example data storage and access control. This effectively allows users of a Grid to establish a community to share data, software, and computers over a network, transforming it into a joint knowledge resource [75]. The European project Enabling Grids for E-sciencE (EGEE) [76] aims at simplifying and improving the access of Grid computing to scientists by establishing a multi-science Grid infrastructure, federating some 250 resource centres world-wide into providing over 40,000 CPUs and several Petabytes of storage. A major goal of the project is to simplify for demanding life science operations to make use of this infrastructure. The BioinfoGRID project [77] aims at developing and promoting this and other computational Grids for bioinformatics. MyGrid [78] is an eScience Grid project that aims to help biologists and bioinformaticians to perform workflow-based
in silico experiments. Cancer Biomedical Informatics Grid (caBIG) [69] is a project which is developing a service-oriented infrastructure of bioinformatics resources for cancer research.

1.6.3 Open science

The enormous quantities of data produced by the new instruments in the life sciences open up for holistic approaches to systems biology [79]. Global collaborations, the depositing of data in public repositories, interdisciplinary research, and community-developed software are democratizing research, empowering any scientist, anywhere, to participate in research even when local resources are scarce. Open science [80, 81] is a relatively new concept with four fundamental goals [82]:

- Transparency in experimental methodology, observation, and collection of data.
- Public availability and reusability of scientific data.
- Public accessibility and transparency of scientific communication.
- Using the Web to facilitate scientific collaboration.

The objectives of Open science can be summarized in the concept "Open Data, Open Standards, and Open Source" (ODOSOS). The idea behind this is that if data is publicly available in an open standard, everyone can use it to reproduce and validate results. If the source code of algorithms is available, people can trust software implementations and easily make use of them. There are several reports on how this enables new research and increases productivity, for example in drug discovery [83]. An example organization which promotes ODOSOS in cheminformatics is the Blue Obelisk [84].

Open science also includes many social components. Wikis and blogs are now common terms for users of the World Wide Web, and scientists are embracing this form of communication as it allows for rapidly reaching collaborators and get responses, comments, suggestions, or educate others [85]. Web-based tools for social science are gaining momentum, with myExperiment [86], a portal for sharing Workflows in the life sciences, as one example.

1.6.3.1 Open source

The term Open Source refers to software development and distribution where the source code is made publicly available, and where anyone can build derivative works upon this as long as the new code is released under the same license. The concept is formally defined for computer science by the Open Source Initiative in the Open Source Definition [87] and the most prominent example is the development of the Linux operating system.

In the life sciences, open source has found many followers due to the iterative nature of science, and the benefits of allowing developers to reuse existing code and avoid 'reinventing the wheel'. One example is the parsing of file formats; a tedious task in the life sciences that developers are not keen on reproducing, but rather rely upon an existing framework for providing this functionality. The ease of use and straightforward availability has also made open source popular; simply download and use the software without complicated signing of contracts. That open source in most places do not impose licensing fees is important for academic research, since budgets for software are commonly limited.
There are numerous open source projects in the life sciences. Examples in bioinformatics are the Bio* toolkits [88] (including Bioperl [89] and BioJava [90]), Cytoscape [91], Strap [92], and JalView [93]. Examples in cheminformatics are the Chemistry Development Kit (CDK) [94], and JChemPaint [95]. The general statistics framework R and Bioconductor [96] deserves mentioning, as do many of the public repositories which are developed as open source software. There have been many papers published which are discussing the use and importance of open source in bioinformatics [97, 98], cheminformatics [84], and drug discovery [99, 100, 83].

Open source software in the life sciences has many advantages for software and data integration. Firstly, access to the source code makes it much easier to integrate a software component than if the source is not available. Secondly, open source software commonly use and promote open standards and interoperable technologies; and hence the goals of open source and software compatibility commonly align well. Further, the people developing open source are usually interested in the long term survival of the project and put more effort in producing sustainable applications that can be integrated with other components, adapted for new scientific areas, and improved by the wider scientific community. The Internet has also made open source development a form of scientific collaboration, where developers have the software implementation as a means of collaborating, often applying the resulting application on different problems and data.
2. Aims of studies

The work presented in this thesis is centered around the concepts of data and software integration, aiming to empower scientists with the emerging technologies of eScience in order to advance research. This includes research and development of:

- A sustainable infrastructure and extensible tools that enables the use of eScience in the life sciences, and which can be adapted for the constantly changing scientific landscape
- Methods and implementations to accommodate for the large data amounts generated by new high throughput technologies
- Standards, exchange formats, and ontologies for the life sciences, primarily targeted towards interdisciplinary research and drug discovery
- Protocols and best practices for interoperable services in the life sciences
3. Methods

3.1 Software integration

Software integration is a topic which comprises both integration of software that runs on the local computer (hereafter referred to as local software), and software that runs on remote computers over a network (hereafter referred to as services). Software integration and data integration are tightly coupled fields, and the easiest and most straightforward way of integrating existing applications is to feed the output of one application as input to another. If both applications can read the same file format the problem boils down to mainly technical implementation details, but if they do not share a common serialization then information needs to be translated from one format to another. This can be a difficult process since it might require advanced domain knowledge.

Running software on the local computer has some advantages. Firstly, users do not need to worry about security as sensitive data need not be transferred over the network. Many scientists are reluctant to submit novel chemical structures or sequences to remote services, in fear of compromising future publications or patents. Secondly, the standard desktop computers of today are quite powerful, capable of performing a substantial amount of work. Using a networked service does imply some extra processing and data transfer, and for smaller jobs this overhead may be larger than the speedup gained from a service running on a high performance computer. Thirdly, local applications have many advantages over software running in a Web browser when it comes to the user experience, as it allows for more responsive and richer user interfaces. Fourthly, transferring large amounts of data over networks, especially the Internet, can be error prone and time-consuming. If large data resources are available locally, a local analysis might be the preferred option.

SOA with interoperable Web/Cloud services has many advantageous properties, including modularity, reusability, ease of maintenance, standardized ways of interaction, and simple construction. Security can be increased by using authentication and encryption protocols together with certificates, or by restricting services to a secure, private network. These solutions can be quite complex to set up, but are crucial for scientists with sensitive data.

3.1.1 Integration of local software

Integration of local software aims at reducing the manual work required when repetitively switching between locally installed applications. An example use case is when running one piece of local software, saving results to a file, opening the file in another local application, loading the previously saved data, and continuing the analysis. This can be done by pipelining data between software, or turning applications into com-
ponents that are accessible from a unifying interface. There are several approaches to adapt an existing application for inclusion in an integrated system: 1) keep it as a standalone application and call it from command line in batch fashion, 2) wrap the component in another application that makes it integrable with the system, and 3) if the component is open source, change the implementation to be compatible with the integrating system. In all cases, the problem of delivering applications for different operating systems and the management of dependencies needs to be handled.

There exist numerous downloadable applications in the life sciences which provide functionality for highly specialized tasks, but very few provide a unified interface with a simple out-of-the-box solution. Existing open source projects for software integration are usually focused on integrating available software applications, commonly using command line invocation on UNIX and Linux based systems. Jemboss [101] wraps around the EMBOSS [64] collection of open source bioinformatics tools using loose coupling, and users can extend functionality by adding e.g. shell commands. ISYS [102] and Gaggle [103] integrates existing software tools and data sources by wrapping them in code to interchange data between them. This requires more work, but gives a tighter integration and more advanced communication among components.

3.1.1.1 Extensibility
An important feature for platforms that integrate local software is their extensibility. Most projects that fall into the integration category provide some sort of mechanism to add plugins to the application. Strap [92] is an application for protein alignment which has a plugin architecture based on the Java HotSwap [104] mechanism to replace parts of the application at runtime without recompilation. The Workflow software Taverna [105] has a plugin architecture that builds on Raven, which is a custom classloader that uses the Java build system Maven [106] at runtime to manage plugin dependencies. Cytoscape [91] is a workbench for visualizing biomolecular interaction networks which also has a custom-built plugin architecture for contributing functionality. An increasingly popular standard for component-based development is OSGi, which is a dynamic component model for the Java programming language defined by the OSGi Alliance [107]. Using OSGi, components can be added, removed, started, and stopped, and the system can detect and react upon these and other events.

3.1.1.2 Web applications
One solution to deliver integrated software to users is to provide a Web front-end to a system where the integration is handled. Users can then interact with the integration system via a browser connecting to the Web application (sometimes called a portal). When such applications are run on powerful servers, they perform well on computationally expensive tasks. Another major advantage with these systems is the ease of maintenance; there is only one location where software needs to be updated. Further, developers do not need to worry about cross-platform issues, as the software runs on a single server and is not distributed to the client computer. Examples of web portals in the life sciences include Entrez [65], which is the web portal of NCBI that allows for cross-database searches in various fields, CARGO (Cancer And Related Genes Online), which facilitates integration and visualization of information regarding cancer from various Internet resources [108], and AnaBench [109], which provides integrated access to tools for sequence analysis. Web-based applications have some drawbacks,
3.1.2 Integration of services

A service is a loose definition, and in this thesis it is defined as to encompass software with programmatic (machine-accessible) interfaces that can be run locally, or consumed over a network (Web services). When several services are available, an important question is how these are integrated to solve a specific problem. The obvious way is to write a small computer program (script) that calls the service, and many service providers offer libraries and snippets to simplify this.

3.1.2.1 Workflows

Writing scripts requires experience in programming, a skill that the vast majority of scientists in the life sciences do not possess. An alternative approach is to use Workflow software to coordinate services, which in theory allows for the graphical setup of complete scientific protocols (see Figure 3.1 for an example of a Workflow). This idea has received a lot of attention in the life sciences, and many initiatives provide packages for Workflow orchestration, including Taverna [105], Cyrille2 [110], and Knime [111].

Apart from making integration of services available for non-computer specialist, Workflows and parts of Workflows can be reused as parts of different analyses, or directly applied to similar problems. An example is the successful integration of microarray and QTL data, which was linked via pathways using a Taverna Workflow,

![Figure 3.1: An example Taverna Workflow to retrieve a number of sequences from the 3 species mouse, human, rat; align them, and return a plot of the alignment result. Workflow author: Stian Soiland-Reyes. Source: http://www.myexperiment.org/workflows/821](http://www.myexperiment.org/workflows/821)
and identified a candidate gene for Trypanosomiasis resistance [112]. The Workflow was then reused without change in Trichuris muris infection in mice, which led to the identification of new biological pathways.

The Workflow technology is still in its infancy. Constructing Workflows for the life sciences is not trivial, mainly because a lot of extra nodes are required to convert data between services. Further, long term sustainability has historically not been prioritized by service providers, as the scientists push on towards the next publication. Relying on external parties may hence be risky, because when services go down, or sometimes change without notice, this can break existing Workflows. However, the more Workflows that are produced, and the more standards and ontologies evolve and become adopted, the easier it will be to develop new Workflows.

3.1.2.2 BioMOBY

BioMOBY [113, 114] is a system for interoperability in the life sciences which consists of extensible ontologies for service and data types, and a central service registry (BioMOBY Central), where service providers can register their services (see Figure 3.2). The registry can then be queried for services by name, or for services that accept a certain data type as input or output (e.g. "give me all services that accept a DNA sequence as input").

3.1.2.3 Service discovery

A problem with the SOAP and REST technologies is that users must be aware of the existence and location of the services before they can be consumed. There is no built-in functionality to query for services, hence users are currently restricted to Web searches. The EMBRACE Network of Excellence provides a registry [115] where providers can register Web services, and users can search for existing services and monitor their status. BioCatalogue [116] is a new project aiming to provide a registration point, facilities for annotations, and a query interface for Web services. The previously mentioned BioMOBY project [113] also has an established registry where services can be discovered.

Figure 3.2: Architecture of the BioMoby system. Services are registered in the BioMOBY Central and clients can subsequently discover these via the Central, and invoke them directly. The BioMOBY Central has a registry and ontology of services and data types, which allows for looking up services based on input and output.
3.1.3 Rich Clients

Rich Clients are downloadable software applications that take advantage of the many offerings of eScience. In contrast to web-based systems, Rich Clients run on the local computer and hence take full advantage of today’s powerful laptops and workstations. They are equipped with a responsive GUI and allow for tight integration with the operating system (e.g. drag and drop, system tray), and usage of local file system and devices (e.g. printers, scanners), but still have the option to invoke remote services and resources (e.g. networked servers, clusters, and databases).

Rich Clients solve the many shortcomings of ordinary downloadable applications, such as delivering a cross-platform product, an integrated update system, and means to consume networked services. These are quite advanced features, and has lead to the emergence of frameworks called Rich Client Platforms (RCP), on which users can base developments in order to reuse common components. The two most well known platforms are Eclipse RCP [117] and NetBeans Platform [118].

3.1.3.1 Eclipse

Eclipse [119] is a universal tool platform that was originally built as an integrated development environment (IDE), but over the years it evolved into a general framework for application development and integration. In Eclipse, all code is split up into plugins, even the core modules (see Figure 3.3). A plugin in the Eclipse world is a collection of functionality that can be seamlessly integrated with other plugins, and can consist of algorithms, visualizations, data, menu options, and much more. This flexible architecture allows for components to be used as building blocks, and the minimal set of plugins needed to form a complete application is collectively known as the Eclipse RCP [117]. This technology enables software developers to focus on the actual application functionality without concern for the core plumbing, which is inherited from Eclipse.

The plugin-architecture of Eclipse is based on OSGi [107], offering the possibility to extend virtually any point in the framework; whether graphical or more lower level
such as the underlying domain object model. To define what can be extended, Eclipse utilizes the concept of *Extension points*, which exist for almost all basic functionality that developers would like to extend. If this is not enough, it is straightforward to create new extension points tailored to user needs, such as for functionality in a certain domain. Eclipse is also equipped with a powerful provisioning system, which means that it is easy to publish online updates to installed software and data. The advanced help and documentation functionality of Eclipse is also a major factor why it has become the most widely used RCP framework available today.

3.2 Data integration

3.2.1 Standards

Data standardization in the life sciences aims at harmonizing differences between how studies are reported to enable the utilization of information from different sources in order to analyze, compare, combine, validate, reproduce, and extend data and metadata. The *Minimum Information* standards are examples of standards which specify the minimum amount of metadata and data required to meet a specific aim in a field. The MGED consortium was a pioneer in bioinformatics with the Minimum Information About Microarray Experiments (MIAME) [120]. Another example is the Human Proteome Organization’s Proteomics Standards Initiative (HUPO-PSI), which is establishing standards and controlled vocabularies in proteomics [121], for example the Minimum Information About a Proteomics Experiment (MIAPE) [122].

3.2.2 Exchange formats

Exchange formats define how data or information is structured in a file to enable different software applications to accurately parse and serialize information. The most common language for this is the Extensible Markup Language (XML) [123], which is an extensible markup language that is widely used in bioinformatics as an easy to use and standardized way to store self-describing data [124]. XML formats for exchanging data in the life sciences are abundant, with examples including Chemical Markup Language (CML) for chemistry [125], Systems Biology Markup Language (SBML) [126] for systems biology models, Distributed Annotation System (DAS) [55] for exchanging information about annotations, Uniprot XML for exchanging protein data [18], and PubChem XML for exchanging chemical structures and assays [26]. A major obstacle in data integration is the widespread use of legacy text formats, which are not easily interpreted by software and hence prone to errors. The transition from legacy formats to well structured XML is an important step towards achieving data interoperability.

Minimum Information guidelines are normally expressed as a set of specifications and guidelines in a certain area, and usually formalized in data models, with accompanying XML-based data exchange formats and ontologies. Examples of such data exchange formats are MAGE-ML for MIAME [127], and HUPO-PSI Molecular Interaction format for the representation of protein interaction data [128]. It has now become a requirement that microarray data must be deposited in MIAME-compliant
public repositories using the MAGE-ML exchange format [127] in order to publish microarray experiments in most journals [129].

3.2.3 Ontologies
In the life sciences, ontologies (also known as controlled vocabularies) are formal representations that are used to define concepts and their relationships in a specific domain. For example, the Gene Ontology has produced a controlled vocabulary that unifies the representation of gene and gene product attributes across all species [130]. By selecting a concept from an ontology it is clear what the meaning is, eliminating uncertainty of spelling and allowing for joining data in a coherent fashion. The need for ontologies in the life sciences has been heavily advocated [131] and is an active research field.

Open Biomedical Ontologies (OBO) is an effort to create controlled vocabularies for shared use across different biological and medical domains [132]. The objective is to allow integration across ontologies by coordinating and reforming existing and new ontologies. The result is an expanding family of ontologies designed to be interoperable and logically well formed, and to incorporate accurate representations of biological reality.

3.2.4 Semantic technologies
A promising technology to deal with the numerous distributed resources in the life sciences is the use of the Semantic Web [133], which promises an infrastructure of machine-understandable content; a World Wide Web made of semantically linked data and not only HTML documents [40].

The Semantic Web toolbox comprises several components. The simplest language is the Resource Description Framework (RDF) [134], which can be used to represent information in the form of so-called triples: subject, predicate, and object (see Figure 3.4a). This simple language identifies everything with a Uniform Resource Identifier (URI), and is enough to express advanced relations. An even more powerful language is the Web Ontology Language (OWL) [135], which was designed as an extension of RDF and provides creation of, for instance, new class descriptions (collection of

![Figure 3.4: a) A triple consists of a subject (in this case Hemoglobin), a predicate (the relation 'is a'), and an object (Oxygen Transport Protein). b) Example of inferred knowledge. The relations A->B and B->C are known, which means that we can infer the relation A->C.](image)
possible resources) and logical combinations (e.g. unions) of other classes. SPARQL (SPARQL Protocol and RDF Query Language) [136] is a querying language for information stored in the form of triples, and allows for their retrieval from repositories (triple stores). Reasoning (also known as inference) in the Semantic Web context refers to the process of inferring logical consequences from a set of asserted facts or axioms, or in simple terms to derive new data from data that are already known (see Figure 3.4b). Querying is a form of inference since search results are inferred from a mass of data.

It is advocated that The Semantic Web for the Life Sciences (SWLS), when realized, will dramatically improve our ability to conduct bioinformatics analyses using the vast and growing stores of web-accessible resources [56]. Several interesting projects are currently demonstrating its use. One example is the Semantic Web for Health Care and Life Sciences (HCLS) Interest Group has the mission to "develop, advocate for, and support the use of Semantic Web technologies for biological science, translational medicine, and health care" [137]. Another example is Linking Open Drug Data (LODD) [138], a task force of the HCLS initiative that focuses on linking the various sources of drug data available on the Web together, to answer scientific questions and demonstrate how physicians and patients can take advantage of the connected data sets.

The Bio2RDF project [139] aims to transform silos of bioinformatics data into RDF/OWL, and integrate and query across distributed knowledge bases to infer biological knowledge. The project has produced a semantic mashup of over 70 million triples built from 30 public bioinformatics data repositories, such as GO, NCBI, UniProt, KEGG, and PDB. This mashup was subsequently used to demonstrate the depiction of genes involved in dopamine neuron degeneration [140].

SADI (Semantic Automated Discovery and Integration) [141] is a framework where Web services and their output are exposed as Semantic Web resources, and where SADI can dynamically generate RDF graphs from these without the need to keep all data in a huge triple store. This exposes data that would not be found using conventional search approaches (referred to as the Deep Web [142]). The project’s first implementation is an application called CardioSHARE [143], which has exposed data for cardiovascular health research.

The previously mentioned BioMOBY does not make use of RDF/OWL, mainly because BioMOBY preceds the mainstream use of these technologies, but the project nevertheless exhibits similar behavior as Semantic Web applications. Several clients have been developed that take advantage of BioMOBY to discover services on the fly in exploratory biological analyses [144, 145] and it is also possible to include BioMOBY services in Taverna Workflows [146, 147].

3.2.5 Model-driven development

Software development is expensive and time-consuming, and specialists in a domain generally do not understand the technicalities of the process. Model-driven development (MDD) focuses on creating abstractions of a domain rather than of implementation details, with the aim to increase interoperability and compatibility between systems. It is common to make use of the Uniform Modeling Language (UML), which is a standardized language for creating data models [148]. The UML model can then be
Figure 3.5: The contents of a QSAR data set. Chemical structures are described mathematically using descriptors, which for example can be calculated properties (Descriptor 1) or an enumeration of the presence of structural fragments (Descriptor 2). The information about the measured effect is also added for each chemical structure (Response), and all values are concatenated into a data matrix (data set) which can be subjected to statistical analysis and build e.g. predictive models.

used to generate database schemas, implementations in most programming languages, and an XML Schema Definition [149] that defines an XML format for exchanging results according to the model. This top-down approach has several advantages: It is fast, allows modelers to focus on modeling and not on the implementations, and if the model is updated then implementations can easily be regenerated. However, it can be problematic to define the model detailed enough to generate enough implementation details. Some developers also argue that it makes people careless when too much code is generated, and that logical errors might slip through unnoticed.

3.3 Drug Discovery

3.3.1 Structure-Activity Relationship

An important task when modeling chemical and biological systems is to understand the effects of small molecules interacting with protein targets. In cases when the structures of the target is known, structure-based techniques, such as docking and molecular dynamics simulations, can be applied. However, reliable 3D structures are not available for many protein families. In such cases a ligand based approach, which aims to correlate structural and physico-chemical features of chemical entities to an observed biological activity, may be used.

Quantitative Structure-Activity Relationship (QSAR) is a ligand-based approach to quantitatively correlate chemical structure with a response, such as biological activity or chemical reactivity. The process is widely adopted and has for example been used to model carcinogenecity [150, 151], toxicity [152, 153], and solubility [154, 155], and the literature is replete with QSAR studies covering problems from lead optimization [156] to fragrance design, and detection of doping in sports [157]. In QSAR, chemical structures are expressed as descriptors, which are numerical representations such as calculated properties or enumerated structural fragments. Descriptors are concatenated into a data set (see Figure 3.5), which can be subjected to statistical analysis in order to build predictive models.
QSAR is a widely used technology for safety assessment of novel substances, and has been advocated by authorities such as OECD [158] and FDA [159]. The pharmaceutical industry make use of QSAR models in drug discovery, aiming to reduce the number of experiments that are performed in the wet lab [160].

Many QSAR data sets are based on a combination of different software tools, mixed with in-house developed solutions, and data sets are then manually set up in spreadsheets. Currently there exists no agreed-upon definition of descriptors, numerous descriptor implementations, and no standard for exchanging data sets in QSAR, making it a virtually impossible task to reproduce and validate analyses. This significantly hinders collaborations and re-use of data and models.

3.3.2 Circular fingerprints

Circular fingerprints (also known as spherical fingerprints) [161] can be used to represent the chemical environment of an atom in a form that can be processed by a computer. These fingerprints consist of atom occurrences in concentric layers radiating out from a central atom (see Figure 3.6). The first layer contains the atom currently centered upon, subsequent layers the bound atoms to the atoms in the previous layers, continued up to an arbitrary length. The atoms are commonly described by their atom type, which apart from the chemical name also includes hybridization and aromaticity.

Circular fingerprints have been shown to perform well compared to other descriptors in several cheminformatics applications, such as similarity searching [162] and the prediction of absorption, distribution, metabolism, excretion, and toxicity properties [161]. An important advantage of circular fingerprints is the interpretability of predictions, since the contribution of descriptors can be graphically visualized in the original chemical structure.

Figure 3.6: An illustration of a circular fingerprint with three level atom environment. The atom the environment is calculated for is at the centre of the environment, labelled black in this figure, and the atoms in the second and third layers are labelled light and dark grey, respectively.
4. Results and discussion

4.1 A workbench for the life sciences (Paper I and V)

Bioclipse (Paper I) originated from the proteochemometric technology [163] developed in Professor Jarl Wikberg’s research group, and the need to integrate cheminformatics with bioinformatics. Much time was spent on manual data conversions and switching between software applications, and this process did not scale. There was a need for a framework that integrated the necessary tools, and what started as a solution for proteochemometrics [164] soon grew to encompass other fields. A fruitful collaboration was established with Dr. Christoph Steinbeck’s research group, at that time based at Cologne University Bioinformatics Institute (CUBIC), and Bioclipse was officially announced in October 2005. The open source license Eclipse Public License (EPL) [165] was chosen as it places no constraints on external plugin licensing; it is totally open for both open source plugins as well as commercial ones. Many external collaborators joined the project, and Bioclipse expanded rapidly.

Architecture and features

Bioclipse is a workbench for the life sciences implemented as a Rich Client based on Eclipse [119]. The plugin-based architecture adheres to the OSGi standard and allows the workbench to be extended into virtually any direction, and also enables the reuse of plugins from other fields in the vibrant Eclipse ecosystem. Bioclipse was early equipped with features for importing, converting, editing, analyzing, and visualizing small molecules, proteins, sequences, and spectra (see Figure 4.1) and several mature libraries and frameworks were integrated to provide functionality, such as the Chemistry Development Kit (CDK) [94, 166] for cheminformatics and spectrum analysis, JChemPaint [95] for 2D editing of chemical structures, Jmol [167] for interactive 3D-visualization (see Figure 4.2(b)), the CML file format for chemistry [168], and BioJava [90] for sequence management. As a Rich Client, Bioclipse is also capable of interacting with remote Web services (see Figure 4.1), such as the WSDbfetch service at EBI [169, 63] to download bioinformatics data from public repositories. A major factor for the success of Bioclipse was however its focus on the user interface, where all technical details are hidden behind graphical tools in a user-friendly workbench (see Figure 4.2(a)).

Bioclipse 2 (Paper V) constituted a complete rewrite which provided the project with a strong foundation for integrating more components, and turned Bioclipse into a stable, scalable platform for the life sciences. The core was remodeled into a more flexible architecture, still based on Eclipse RCP but with Spring [170] added to provide dependency injection and aspect oriented programming (AOP). These architectural changes enabled Bioclipse with a new scripting language (see Section 4.5), and
made recording of GUI interaction possible. Several new graphical components were also developed, including a brand new version of the chemical 2D editor JChemPaint, a MoleculesTable for visualizing multiple molecules (see Figure 4.2(a)), an editor for working with biological sequences (see Figure 4.2(c)), and Balloon for 3D conformer generation [171]. A client for the PubChem eUtils [26] was added to enable the querying for and downloading of chemical structures, and several XMPP cloud services [172] were also integrated.

Scope
The scope of Bioclipse is manifold. For developers, Bioclipse provides a framework to integrate and develop new functionality, and the results can then be made available together with other developers’ contributions. This brings people from different fields together and promotes interdisciplinary collaborations, and an encouraging experience was when developers found points of interaction between plugins that were not thought of from the start. For ordinary users, Bioclipse is a workbench that provides much of the functionality which is required on daily basis. Advanced users can make use of the scripting functionality to automate tasks and create reproducible analyses that can be shared between parties.

Bioclipse finds its place in several of the drug discovery phases due to its inherent capability to be tailored for different use cases. Scripting and database searches can for example be applied in the lead identification phase, and interactive graphical tools together with computational models for decision support are more suitable in the lead optimization phase.
Figure 4.2: Screenshots from Bioclipse in different applications. a) Multiple chemical structures visualized in the Molecules Table (top middle panel), with the selected structure rendered in a separate 2D view (bottom right panel) and computed properties (bottom left panel). Also shown is a 3D structure where monomers are selected in the outline (top rightmost panel) and highlighted in the interactive 3D visualization component Jmol (top right panel). The Javascript console (bottom middle panel) can be used to execute scripting commands. b) 3D visualization of the interaction between a protein and DNA rendered in Jmol. c) A multiple sequence alignment open in the Sequence Editor.

Community
The Bioclipse community is active and constantly growing. The Bioclipse Wiki [173] and the mailing lists are central places where development is discussed and documented, and the Bioclipse Blog [174] is primarily used for announcements. Bioclipse Planet [175] is a website that aggregates several blogs with the topic of using and developing Bioclipse. A Bugzilla portal is available where people can report bugs or request features [176]. During the Bioclipse development the versioning control system for source code was upgraded from the original Concurrent Versions Sys-
tem (CVS) [177] to Subversion (SVN) [178], and finally to the distributed versioning system Git [179]. The latter gives a high level of flexibility and simplifies keeping multiple versions of Bioclipse in the release pipe. It is a clearly stated goal of the Bioclipse project to further develop the infrastructure for its community, where use cases and solutions are openly discussed, and where end users and developers can share experiences and knowledge.

Bioclipse has from its initial release been well received by the scientific community, and in 2006 the project was awarded two innovation prizes; the JAX Innovation Award 3rd price and JAX Innovation Audience Award. Further, in November 2007 Bioclipse won the 'special prize of the jury’ in the international software contest Trophees du libre held in Soissons, France. The total number of downloads amounted in September 2009 to over 30,000, increasing with about 800 downloads each month.

Integrating management and analysis of small molecules (e.g. drugs) with biological entities (e.g. genomic and proteomic sequences) is an important task in many fields, and Bioclipse is the first open source project that provides this functionality in a graphical workbench. We envision the need for such integrated approaches to increase, as fields like proteochemometrics and genomic medicine are gaining in popularity [180, 181]. For the sake of this thesis, Bioclipse laid the foundation for the other studies and came to be the central part where all functionality was integrated.

4.2 Discoverable and asynchronous services (Paper II)

The two predominant technologies for Web services, SOAP [59] and REST [60], have several drawbacks, including lack of discoverability and the inability for services to send status notifications asynchronously. Much is attributed the fact that both technologies are based on the Hypertext Transfer Protocol (HTTP) which is intrinsically synchronous (a 'pull protocol'), meaning that clients must send a request to a server in order for it to reply. This is problematic for long-running jobs due to the fact that network connections can be terminated if there is no activity within a certain time. A common workaround for SOAP services is to implement a ticketing system (also known as 'polling’) where the client receives a ticket upon a request, and then repetitively asks if the job is finished using this ticket. This procedure not only leads to communication overhead, but the most important drawback is that such ticketing systems are not standardized, and hence it is not possible to build a general client for SOAP services with such implementations. As REST services by design are stateless, they are incapable of maintaining a state and are better suited for more rapidly responding services like data provisioning.

Extensible Messaging and Presence Protocol (XMPP) [182] is a decentralized XML routing technology that allows any entity to actively send XMPP messages to another entity. In Paper II we presented a novel approach for Cloud services based on XMPP, consisting of an XMPP Extension Protocol (XEP) to comprise discovery, asynchronous communication, and definition of data types in the service [183]. The XEP proposed by us, named IO Data, was subsequently accepted by the XMPP council, and implementations were developed for Bioclipse and Taverna. An XMPP
Figure 4.3: XMPP is formidable for providing the backbone in a ‘Service Cloud’, where clients like Bioclipse can invoke services on demand. Two advantages over traditional technologies is that XMPP Cloud services can be discovered, and communication between the server and the client can be asynchronous. This is ideal for long-running operations as it enables clients like Bioclipse to continue with other tasks.

server, ws1.bmc.uu.se\(^1\) was established, and services for calculation of chemical properties using CDK [166], and prediction of drug susceptibility for mutated HIV proteases [184] were established. Services based on XMPP + IO Data demonstrate several advantages over traditional W3C Web services:

1. Services can be discovered
2. Asynchronous communication eliminates the need for ad-hoc solutions like polling
3. Input and output types defined in the service allows for generation of clients on the fly, as well as simplifies inclusion in pipelines and Workflows

The use of the standardized protocol XMPP opens up for integration with other technologies, for example SMS services to send notices upon service completion, or protocols specifically designed for the transfer of large files. XMPP also comes with features which are not available in SOAP/REST, such as communication between servers to form federated networks. It is straightforward to constrain servers and form a 'private cloud', but equally easy to open them up to the global XMPP network and make services available for the public (see Figure 4.3). XMPP is an authenticated network, which means that users are required to log in with a username and password. This gives the opportunity to apply restrictions for services on user basis, as well

\(^1\)The server ws1.bmc.uu.se is hosted and administered by the computer department at Uppsala Biomedical Center (BMC).
as for monitoring and logging of service utilization. While the above use cases are technically possible to achieve with SOAP, this is standardized in XMPP and does not require additional extensions. The many advantages over existing technologies make XMPP + IO Data to an interesting candidate for the next generation of Web services in bioinformatics.

4.3 Near-real time metabolic site predictions (Paper III)

In Paper III we presented a fast and interactive implementation of site-of-metabolism (SOM) prediction based on an existing and validated algorithm to analyze ligands and visualize putative metabolic sites [185]. The method is called MetaPrint2D and is based on historic metabolic reactions which are preprocessed using circular fingerprints [161] and stored in a database. When a new compound is presented for the algorithm, the probability of metabolism occurring can be calculated for each atom as a ratio of the number of times an atom with this environment is present in the products and substrates part of the database. In this work we produced four databases, comprising the species Human, Dog, Rat, and All (the three species combined).

A plugin for Bioclipse was developed which allows for near-real time calculations of MetaPrint2D directly in a chemical editor. The results are visualized graphically in single molecules or in collection of molecules, with colors representing the probability of metabolism occurring at the atoms (see Figure 4.3). The time to predict new compounds is significantly lower than existing available software, making Bioclipse-MetaPrint2D the fastest available SOM tool, and the only one with near-real time updating when editing chemical structures. No access to networked computers is re-

Figure 4.4: Above: The structure of Valproic acid with MetaPrint2D results visualized. Atoms colored in red indicates a high probability of metabolism occurring at that site, yellow indicates medium, green low, and no coloring very low or no probability of metabolism occurring at that atom. Right: Screenshot from Bioclipse with the results from a MetaPrint2D calculation of a file comprising 5 well known drugs.
4.4 A new standard for QSAR (Paper IV)

In Paper IV we presented a step towards standardizing QSAR analyses by defining interoperable and reproducible QSAR data sets, comprising an open XML format (QSAR-ML) and a descriptor ontology (Blue Obelisk Descriptor Ontology). The ontology provides an extensible way of uniquely defining descriptors for use in QSAR experiments, and the exchange format supports multiple versioned implementations of these descriptors. Hence, a data set described by QSAR-ML makes its setup completely reproducible. We also provided an implementation for Bioclipse that allows for setting up QSAR data sets, and exporting results in QSAR-ML as well as traditional comma-separated formats. Model-driven development was used for the Bioclipse implementation, and a model was crafted and transferred into an XML Schema [149]. Implementation code was generated with the Eclipse Modeling Framework [187], and the generated model code was extended with OSGi functionality to facilitate addition of new descriptor implementations, both from locally installed software and via remote Web services. A GUI was also developed to hide all the technical details for end users, and allow for importing of chemical structures, selecting descriptors from the Blue Obelisk Descriptor Ontology, cherry-picking local and remote descriptor providers, adding responses and metadata, and finally performing all calculations and exporting the complete data set in QSAR-ML (see Figure 4.5).

Standardized QSAR opens up new ways to store, query, and exchange analysis, and makes it easy to join, extend, combine and also to work collectively with data. The Bioclipse-based implementation greatly simplifies integrating different descriptor calculation softwares, and makes it straightforward to produce QSAR-ML compliant data sets.
4.5 A scripting language for the life sciences (Paper V)

In Paper V we introduced a novel scripting language for the life sciences called Bioclipse Scripting Language (BSL). Technically it consists of an extensible set of functions that provides users with functionality which can be integrated into other programming languages, and Bioclipse version 2 included a reference implementation based on Javascript.

In Bioclipse 2, all functional code contributed by plugins is structured in Bioclipse Managers; e.g. code that provides access to 3D conformer generation using Balloon [171] is available in the BalloonManager. The Manager objects are published into the scripting environment, and hence the same objects that are called from the GUI are reachable from scripts (see Figure 4.6 and 4.7). Plugins can contribute new commands to BSL, and this is the recommended way of adding functionality to Bioclipse. An example is Jmol [167], which has its own scripting language exposed in BSL by a JmolManager. Another example is the BioWS plugin which makes the sequence alignment tool Kalign [188] available in BSL via a Web service provided by EBI [63].

The JavaScript Console is a canvas where users can enter commands in BSL and Javascript to execute functionality in Bioclipse (see Figure 4.2(a)). The GUI can also be used to interact with the Console, and it is possible to drag and drop resources into the JavaScript Console to facilitate the setup of commands. The JavaScript Editor can be used for writing scripts, containing one or more lines of code in BSL collected in an executable program, which are ideal to share between scientists (see Figure 4.7b). The use of open and collaborative services like Gist [189] and myExperiment [86] for sharing scripts is encouraged.

Cross-domain scripts are powerful tools for integrating functionality in the life sciences. They provide functionality similar to Workflows, but are much faster to create. The flexibility is also higher in scripts as they allow for conditional loops and straightforward conversion between different textual data types. When looking at Taverna Workflows today, many of them contain nodes at various places, comprising small

![Figure 4.6](#): Overview of the Bioclipse architecture describing the use of Managers to collect functional code. This architecture makes all functionality available from both the GUI and the Bioclipse Scripting Language.
Figure 4.7: a) Three different commands to query public repositories for sequences using BSL.
b) BSL script to query EMBL for two DNA sequences, align them using Kalgin, and write the alignment to a FASTA file. c) The first page of a graphical wizard in Bioclipse for executing the same queries as in (a).

scripts to handle such data conversion. The downside with scripts is that they require basic knowledge in programming, but they are very powerful for users who possess this skill. In the future we envision the possibility to convert BSL scripts to Workflows, and also to produce Workflow nodes that are able to execute BSL scripts.
5. Concluding remarks

The emergence of eScience holds many promises, but it also requires substantial development to reach its full potential. It is important to structure data and information according to well-defined standards so that analysis tools can make use of it, but due to the iterative nature of science it is also important to visually inspect, validate, and interact with the information during the process. Rich clients are ideal for these kinds of operations, and also for more exploratory ventures where the process is not known beforehand but discovered along the way. An illustrative example is the setup of QSAR data sets, which was demonstrated in Paper IV. Using the graphical tools of Bioclipse it is straightforward to experiment with different descriptors in a data set, and when satisfied the results can be exported in a standardized format (QSAR-ML). A second example is the integration of components in the Bioclipse Scripting Language, as demonstrated in Paper V. After successful data exploration, the protocol can easily be summarized in a reproducible script and exchanged between parties.

The adoption of Web services in the scientific community is an important step towards software interoperability, but is not the answer to everything. In fact, I believe that local implementations are superior in many cases, provided that the problems of cross-platform deployment and application updates are resolved. One example is MetaPrint2D (Paper III) where a local implementation was chosen over a Web service, delivering unprecedented speed in site-of-metabolism predictions, and hence opening up for new applications of the algorithm. A service-based implementation would in this case have reduced the performance due to communication overhead, and would also require constant network connection with the associated security concerns. However, Web services do have many advantages, and are sometime the only option in cases where the underlying data or implementations are not distributable. This is common when services are based on large or proprietary data sources, implemented in a platform-specific programming language, or requires HPC resources. A mix of local and remote functionality maximizes the usefulness in end user tools, and Rich Clients are excellent for dealing with these tasks. Paper II showed how the shortcomings of today’s Web services can be overcome with the XMPP protocol, and the implementation in Bioclipse demonstrated how well suited Rich Clients are for handling service discovery and concurrent, long-running jobs.

As the founder of the Bioclipse workbench (Paper I and V), I am proud of how the project has developed, and optimistic about the future. The decision to base the application on the Eclipse Rich Client Platform and hence the OSGi framework was very successful, and several other prominent software projects in the life sciences, such as Taverna [105] and Cytoscape [91], are now rewriting their applications to be based on this technology. The adoption of OSGi as a common architecture among software tools in the life sciences is promising, as it effectively allows for reuse and integration of software components between applications.
An important and neglected issue in eScience is long term sustainability. Relying on external parties for data and service provision is associated with a certain degree of risk. For example, data and functionality might be unavailable at the desired time of analyses due to e.g. network congestion or, more severely, changed or removed services. Versioning of data and services are very important in order to be able to audit and reproduce analyses at a later time, when data and services have been updated. Service mirroring, where services are duplicated over different systems, is another overlooked topic. Mirroring not only increases the speed of data transfers (as servers usually provide a faster connection if located geographically closer to the client), but also acts as a fail-safe system so that jobs can be sent to one of the mirrors if the main service goes down. Such an infrastructure is currently not commonplace in eScience, but is important for long term sustainability. When using XMPP services, several server components running on different platforms can offer the same service, effectively mirroring the services. Mirroring and load balancing with XMPP needs further research and development, but shows high potential for the future.
6. Acknowledgements

This work was carried out at the Department of Pharmaceutical Biosciences, Division of Pharmacology, Faculty of Pharmacy, Uppsala University.

I would like to thank the following people who have supported me during the years towards my thesis:

**Jarl Wikberg**, my supervisor and Professor. Thanks for all your help and support on my way towards becoming a scientist.

**Dr. Christoph Steinbeck**, European Bioinformatics Institute, Hinxton, UK.

**Dr. Lars Carlsson, Ernst Ahlberg Helgee, and Dr. Scott Boyer**, Global Safety Assessment, AstraZeneca R&D, Mölndal, Sweden.

**Professor Robert Glen** and **Samuel Adams**, Unilever Center for Molecular Informatics, Cambridge University, UK.

**Dr. Johannes Wagener**, Max von Pettenkofer-Institut, Ludwig-Maximilians-Universität, Munich, Germany.

**Dr. Rajarshi Guha**, NIH Chemical Genomics Center, MD, USA.

**Professor Peter Murray-Rust**, Cambridge University, UK.

**Dr. Erik Bongcam-Rudloff** and **Alvaro Martinez Barrio**, Swedish University of Agricultural Sciences (SLU), Sweden.


**Professor Johan Åqvist, Dr. Martin Nervall, and Göran Wallin**, Department of Cell and Molecular Biology, Uppsala University.

**Dr. Egon Willighagen** for all collaborations on Bioclipse, CDK, QSAR, and XMPP.

My fellow Bioclipse developers and group members **Jonathan Alvarsson, Carl Mäsk, Arvid Berg**, as well as **Sviatlana Yahorava, Aleh Yahorau, Maris Lapins, Peteris Prusis, and Muhammad Junaid**.

All other people working at the Department of Pharmaceutical Biosciences at BMC, especially **Erica Johansson, Magnus Jansson**, and **Agneta Hortlund**.
Many other people have made substantial contributions to the Bioclipse project: 

- Stefan Kuhn, Dr. Mark Rijnbeek, Dr. Gilleain Torrance, Dr. Tobias Helmus, Miguel Rojas, Dr. Thomas Kuhn, Eskil Andersson, Annsofi Andersson, Bjarni Júliusson, Sofia Burvall, Dr. Jerome Pansanel, and Samuel Lampa.

- Hanna, Kim, and Darta for lightening up Wednesdays.

- Martin "Puff" Eklund. It’s always christmas.

- My parents Ebba and Lennart and my brothers, Per and Kalle.

- All my friends.

- And finally, Nora, for helping me with the completion of this thesis and for bearing with me during this, quite busy, period. Looking forward to spending more quality time with you!

The research presented in this thesis was supported by grants from Swedish VR-M (04X-05957) and Uppsala University (KoF 07).
Bibliography


Acta Universitatis Upsaliensis

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Pharmacy 111

Editor: The Dean of the Faculty of Pharmacy

A doctoral dissertation from the Faculty of Pharmacy, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Pharmacy. (Prior to January, 2005, the series was published under the title “Comprehensive Summaries of Uppsala Dissertations from the Faculty of Pharmacy”.)