Efforts towards accessible and reliable bioinformatics

Matúš Kalaš

Dissertation for the degree of Philosophiae Doctor (PhD)

Department of Informatics
University of Bergen

2015

Scientific environment

The work presented in this thesis has been carried out at the Computational Biology Unit (CBU) at the Department of Informatics (II), Faculty of Mathematics and Natural Sciences, University of Bergen. Until 2013, CBU was part of the Bergen Center for Computational Science (later renamed to Uni Computing and recently Uni Research Computing), a branch of Unifob (a research company majority-owned by the University of Bergen, later renamed to Uni Research Ltd.). For the whole duration of my PhD, I was affiliated with the Department of Informatics as my home institute. I was affiliated also with the Molecular and Computational Biology research school (MCB) at the University of Bergen. This thesis was supervised by Professor Inge Jonassen at II and CBU, and co-supervised by Dr. Kjell Petersen at CBU and II, and Dr. Pål Puntervoll at CBU (now at Uni Miljø/Uni Research Environment, Uni Research Ltd.).

Parts of this work were performed in collaboration with the System administration team led by Kristoffer Rapacki at the Center for Biological Sequence Analysis (CBS), Department of Systems Biology, Technical University of Denmark (DTU); the IT department and now the Center of Bioinformatics, Biostatistics and Integrative Biology (C3BI) at the Institut Pasteur in Paris; the Research Group for Biomedical Informatics at the Department of Informatics, Faculty of Mathematics and Natural Sciences, University of Oslo and the Department of Tumor Biology, The Norwegian Radium Hospital, Oslo University Hospital; Peter Rice's Group and the Web Production/External Services team led by Rodrigo Lopez at EMBL-EBI in Hinxton, UK; the bioinformatics infrastructure team led by Christophe Blanchet at ICBP, CNRS, Lyon (now at L'Institut Français de Bioinformatique, Gif-sur-Yvette), the Advanced Interfaces Group led by Steve Pettifer at the School of Computer Science, University of Manchester; and the Burkhard Rost's group at the Bioinformatics and Computational Biology Department, Technische Universität München (TUM).

My work was funded by the Norwegian Research Council: projects eSysbio, FUGE Bioinformatics Platform, and ELIXIR.NO. My research was partially connected with the European projects EMBRACE, AllBio, and ELIXIR. In addition to these, my travels were supported with occasional travel fellowships from the MCB (2010) and from the European Conference on Computational Biology (2010, 2015), with contribution from the International Society for Computational Biology and the Irish Government.

Acknowledgements

Aims of the thesis

The aim of the presented work was contributing to making scientific computing more accessible, reliable, and thus more efficient for researchers, primarily computational biologists and molecular biologists. Many approaches are possible and necessary towards these goals, and many layers need to be tackled, in collaborative community efforts with well-defined scope. As diverse components are necessary for the accessible and reliable bioinformatics scenario, our work focussed in particular on the following:

In the BioXSD project, we aimed at developing an XML-Schema-based data format compatible with Web services and programmatic libraries, that is expressive enough to be usable as a common, canonical data model that serves tools, libraries, and users with convenient data interoperability.

The EDAM ontology aimed at enumerating and organising concepts within bioinformatics, including operations and types of data. EDAM can be helpful in documenting and categorising bioinformatics resources using a standard "vocabulary", enabling users to find respective resources and choose the right tools.

The eSysbio project explored ways of developing a workbench for collaborative data analysis, accessible in various ways for users with various tasks and expertise. We aimed at utilising the World-Wide-Web and industrial standards, in order to increase compatibility and maintainability, and foster shared effort.

In addition to these three main contributions that I have been involved in, I present a comprehensive but non-exhaustive research into the various previous and contemporary efforts and approaches to the broad topic of integrative bioinformatics, in particular with respect to bioinformatics software and services. I also mention some closely related efforts that I have been involved in.

The thesis is organised as follows: In the *Background* chapter, the field is presented, with various approaches and existing efforts. *Summary of results* summarises the contributions of my enclosed projects – the BioXSD data format, the EDAM ontology, and the eSysbio workbench prototype – to the broad topics of the thesis. The *Discussion* chapter presents further considerations and current work, and concludes the discussed contributions with alternative and future perspectives.

The three articles that are part of this thesis, are attached after the *Discussion* and References.

Table of contents

Contributions included in the thesis

Article I

Matúš Kalaš, Pål Puntervoll, Alexandre Joseph, Edita Bartaševičiūtė (now Karosiene), Armin Töpfer, Prabakar Venkataraman, Steve Pettifer, Jan Christian Bryne, Jon Ison, Christophe Blanchet, Kristoffer Rapacki, and Inge Jonassen **(2010)**. BioXSD: the common data-exchange format for everyday bioinformatics web services. *Bioinformatics*, **26**(18): i540–i546.

I have developed the BioXSD data model from analysing a wide variety of bioinformatics tools, exchange formats, and collaborators' requirements, coded and maintained the XML Schema and build scripts, examples, web page, and programmed the test workflow. I wrote the manuscript with edits from the co-authors.

Article II

Jon Ison, Matúš Kalaš, Inge Jonassen, Dan Bolser, Mahmut Uludag, Hamish McWilliam, James Malone, Rodrigo Lopez, Steve Pettifer, and Peter Rice **(2013)**. EDAM: An ontology of bioinformatics operations, types of data and identifiers, topics, and formats. *Bioinformatics*, **29**(10): 1325–1332.

I have contributed to the conceptual design and the development and maintenance of EDAM, led by Jon Ison. I implemented the content negotiation at *edamontology.org*, EDAM usage in eSysbio, and the semantic annotation with EDAM in BioXSD; and administer the website. I led the work on the manuscript, written together with Jon, Steve, and Inge.

Article III

Kidane Tekle, Håkon Sagehaug, Prabakar Venkataraman, Armin Töpfer, Matúš Kalaš, Paweł Sztromwasser, Anne-Kristin Stavrum, Siv Midtun Hollup, Michael Dondrup, Sattanathan Subramanian, Francisco Roque, Inge Jonassen, Kjell Petersen, and Pål Puntervoll **(Unpublished)**. eSysbio: a workbench proposal for collaborative computational biology. *Manuscript in preparation*.

I contributed to the design of eSysbio conceptually – in particular from the usability and maintainability viewpoints – and by analysing use cases and other requirements from the potential community. I implemented the usage of a subset of EDAM, developed the Web Service Interaction Ontology (WSIO), comprehensively explored related systems and efforts, tested the eSysbio prototype workbench regularly, and contributed to the manuscript.

Other contributions

All articles are freely available (open access). A click on an article's title or DOI will open the underlying link.

Steve Pettifer, Jon Ison, Matúš Kalaš, Dave Thorne, Philip McDermott, Inge Jonassen, Ali Liaquat, José M Fernández, Jose M Rodriguez, David G Pisano, Christophe Blanchet, Mahmut Uludag, Peter Rice, Edita Bartaševičiūtė (now Karosiene), Kristoffer Rapacki, Maarten Hekkelman, Olivier Sand, Heinz Stockinger, Andrew B Clegg, Erik Bongcam-Rudloff, Jean Salzemann, Vincent Breton, Teresa K Attwood, Graham Cameron, and Gert Vriend **(2010)**. The EMBRACE web service collection. *Nucleic Acids Res.*, 38(suppl 2,W1): W683–W688. 10.1093/nar/gkq297

As an active participant in the EMBRACE project, I developed BioXSD, wrote the corresponding part of the manuscript, and contributed to the design and later development of EDAM.

Sveinung Gundersen, Matúš Kalaš, Osman Abul, Arnoldo Frigessi, Eivind Hovig, and Geir K Sandve **(2011)**. Identifying elemental genomic track types and representing them uniformly. *BMC Bioinformatics*, **12**(1): 494. 10.1186/1471-2105-12-494

I contributed to the design of the GTrack format, and improved BioXSD into version 1.1 based on similar optimisation tactics as in GTrack. I wrote parts of the manuscript.

Tomas Klingström, Larissa Soldatova, Robert Stevens, Erik T Roos, Morris A Swertz, Kristian M Müller, Matúš Kalaš, Patrick Lambrix, Michael J Taussig, Jan-Eric Litton, Ulf Landegren, and Erik Bongcam-Rudloff **(2013)**. Workshop on laboratory protocol standards for the molecular methods database. *N. Biotechnol.*, **30**(2): 109–113. 10.1016/j.nbt.2012.05.019

I contributed with ideas for the standardisation of description and provenance of sample processing protocols, and shared our experience from EMBRACE, BioXSD, and EDAM.

Geir K Sandve, Sveinung Gundersen, Morten Johansen, Ingrid K Glad, Krishanthi Gunathasan, Lars Holden, Marit Holden, Knut Liestøl, Ståle Nygård, Vegard Nygaard, Jonas Paulsen, Halfdan Rydbeck, Kai Trengereid, Trevor Clancy, Finn Drabløs, Egil Ferkingstad, Matúš Kalaš, Tonje Lien, Morten B Rye, Arnoldo Frigessi, and Eivind Hovig **(2013)**. The Genomic HyperBrowser: an analysis web server for genome-scale data. *Nucleic Acids Res.*, **41**(suppl 2,W1): W133–W141. 10.1093/nar/gkt342

I contributed to the design of the core data format used by HyperBrowser, the GTrack.

Steffen Möller, Enis Afgan, Michael Banck, Peter JA Cock, Matúš Kalaš, László Kaján, Pjotr Prins, Jacqueline Quinn, Olivier Sallou, Francesco Strozzi, Torsten Seemann, Andreas Tille, Roman Valls Guimera, Toshiaki Katayama, and Brad Chapman **(2013)**. Sprints, Hackathons and Codefests as community gluons in computational biology. *EMBnet.J.*, **19**(B): 40–42. 10.14806/ej.19.B.726

and

Steffen Möller, Enis Afgan, Michael Banck, Raoul JP Bonnal, Timothy Booth, John Chilton, Peter JA Cock, Markus Gumbel, Nomi Harris, Richard Holland, Matúš Kalaš, László Kaján, Eri Kibukawa, David R Powell, Pjotr Prins, Jacqueline Quinn, Olivier Sallou, Francesco Strozzi, Torsten Seemann, Clare Sloggett, Stian Soiland-Reyes, Sascha Steinbiss, Andreas Tille, Anthony J Travis, Roman Valls Guimera, Toshiaki Katayama, and Brad Chapman **(2014)**. Community-driven development for computational

biology at Sprints, Hackathons and Codefests. *BMC Bioinformatics*, **15**(Suppl 14): S7. 10.1186/1471-2105-15-S14-S7

As a regular participant in the OpenBio Codefests and the Debian Med Sprints, I channelled the community's requirements, ideas, and spirit into BioXSD and EDAM, in turn contributing with ideas and promotion to other related projects, and with edits to these two manuscripts.

Toshiaki Katayama, Mark D Wilkinson, Kiyoko F Aoki-Kinoshita, Shuichi Kawashima, Yasunori Yamamoto, Atsuko Yamaguchi, Shinobu Okamoto, Shin Kawano, Jin-Dong Kim, Yue Wang, Hongyan Wu, Yoshinobu Kano, Hiromasa Ono, Hidemasa Bono, Simon Kocbek, Jan Aerts, Yukie Akune, Erick Antezana, Kazuharu Arakawa, Bruno Aranda, Joachim Baran, Jerven Bolleman, Raoul JP Bonnal, Pier Luigi Buttigieg, Matthew P Campbell, Yi-an Chen, Hirokazu Chiba, Peter JA Cock, K Bretonnel Cohen, Alexandru Constantin, Geraint Duck, Michel Dumontier, Takatomo Fujisawa, Toyofumi Fujiwara, Naohisa Goto, Robert Hoehndorf, Yoshinobu Igarashi, Hidetoshi Itaya, Maori Ito, Wataru Iwasaki, <u>Matúš Kalaš</u>, Takeo Katoda, Taehong Kim, Anna Kokubu, Yusuke Komiyama, Masaaki Kotera, Camille Laibe, Hilmar Lapp, Thomas Lütteke, M Scott Marshall, Takaaki Mori, Hiroshi Mori, Mizuki Morita, Katsuhiko Murakami, Mitsuteru Nakao, Hisashi Narimatsu, Hiroyo Nishide, Yosuke Nishimura, Johan Nystrom-Persson, Soichi Ogishima, Yasunobu Okamura, Shujiro Okuda, Kazuki Oshita, Nicki H Packer, Pjotr Prins, Rene Ranzinger, Philippe Rocca-Serra, Susanna Sansone, Hiromichi Sawaki, Sung-Ho Shin, Andrea Splendiani, Francesco Strozzi, Shu Tadaka, Philip Toukach, Ikuo Uchiyama, Masahito Umezaki, Rutger Vos, Patricia L Whetzel, Issaku Yamada, Chisato Yamasaki, Riu Yamashita, William S York, Christian M Zmasek, Shoko Kawamoto, and Toshihisa Takagi **(2014)**. BioHackathon series in 2011 and 2012: penetration of ontology and linked data in life science domains. *J. Biomed. Sem.*, **5**(1): 5. 10.1186/2041-1480-5-5

As a participant in the 4th BioHackathon, in 2011, I improved the compatibility of EDAM and BioXSD with the Semantic Web, and contributed with my bits to the manuscript.

László Kaján, Thomas A Hopf, <u>Matúš Kalaš</u>, Debora S Marks, and Burkhard Rost **(2014)**. FreeContact: fast and free software for protein contact prediction from residue co-evolution. *BMC Bioinformatics*, **15**(1): 85. 10.1186/1471-2105-15-85

I helped with designing the interoperability of FreeContact, improved BioXSD according to the corresponding requirements, and provided ideas and edits to the manuscript.

Hervé Ménager, <u>Matúš Kalaš</u>, Kristoffer Rapacki, and Jon Ison **(2015)**. Using registries to integrate bioinformatics tools and services into workbench environments. *Int. J. Softw. Tools Technol. Transfer*. 10.1007/s10009-015-0392-z

I contributed with initial ideas to these efforts and contributed to the manuscript.

<u>Matúš Kalaš</u>. WSIO (Web Service Interaction Ontology). http://wsio.org

I developed WSIO in order to facilitate automated invocation of Web services that deal with large data or time-consuming computation, based on the requirements of the EMBRACE and eSysbio projects, and open to future requirements and developments.

Jon Ison, <u>Matúš Kalaš</u>, Peter Rice. DRCAT (Data Resource CATalogue). http://drcat.sourceforge.net

DRCAT (pronounced "Doctor Cat") is a semantically annotated catalogue of web-accessible bioinformatics databases developed by Jon based on previous work of Christopher Southan, with Peter's and my contribution.

1 Background

The *Background* chapter of this thesis first briefly introduces the field of bioinformatics to a non-bioinformatician reader, and then outlines the main sources of accessibility and reliability problems with bioinformatics tools and data. Example approaches and efforts towards more accessible and reliable bioinformatics are presented throughout the rest of the chapter. For an interested reader, I can recommend Attwood *et al.* (2011) as one of interesting historical overviews of bioinformatics from the point of view of bioinformatics databases, or Hogeweg (2011) for her story of bioinformatics since the beginning.

1.1. Bioinformatics is an integral component of life sciences

Life sciences is an umbrella term covering a whole range of research disciplines about living organisms. With biology as the central component, life sciences include also fields such as ecology, medical research, pharmacology, and biotechnology. The research in life sciences focuses on topics including evolution, health and disease, ecosystems, life's diversity, genotype, phenotype, and their variations, mechanisms of life, and their applications in technology. To enable answering questions about these topics, and to organise the life-scientific knowledge, detailed information is being recorded about species and their relations, anatomy and development, of genes, proteins, other molecules, their interactions and functions, of whole genomes of species, and metagenomes of ecosystems.

Successive innovations in measuring and imaging technologies are enabling a massive growth in volume, quality, and diversity of produced biological data on the molecular level, reaching from fully sequenced genomes of species or individuals, through structures and movements of proteins and other molecules, to details about interactions between various kinds of molecules and elements in genomes. Epigenetic and phenotypic properties of living organisms are being captured under certain conditions: for example the expression levels of genes, or concentrations of various kinds of molecules under a given condition.

Bioinformatics is the discipline dedicated to computational processing, analysis, storage, and representation of biological data, mostly on the molecular level. Bioinformatics has over the last decades become an integral component of research in the fields of molecular biology, medicine, pharmacology, ecology, and biotechnology, in particular in cases of research where the amount of analysed data demands high-throughput computational processing. The post-paradigmatic, interdisciplinary nature of today's life-scientific research demands diverse expertise and methods to be developed and applied. The involved disciplines include biology, chemistry, and medicine, but also physics, mathematics including statistics and dynamic systems, and informatics including *e.g.* data management, algorithmics, software engineering, high-performance computing, machine learning, or text mining. Occasionally, cross-disciplinary life-scientific research reaches out even to disciplines such as environmental, social, Earth, or space sciences, law, ethics, linguistics, or philosophy.

Bioinformatics itself focuses on developing and applying algorithms, mathematical, and statistical methods to process molecular-biological data obtained from lab, bench, or field studies, in order to find answers to challenging scientific or technological questions. Types of data being processed include for example sequences and 3D structures of macromolecules such as DNA, RNA, proteins, their parts or complexes, microscope images, or measured concentrations of certain types of molecules or sequences. In addition to analysing laboratory data, bioinformaticians have a central role in producing, publishing, and maintaining derived data of scientific interest, such as annotations of loci in genomes, genes and gene products with their features and relations, alignments of related sequences or structures, evolutionary trees, or networks of interacting genes and molecules, with their systemic properties.

Other inter-disciplinary fields overlap with bioinformatics to a notable extent. Without trying to fully define them, example relations include:

- **Computational biology**. The terms *computational biology* and *bioinformatics* are often used interchangeably as close synonyms. On the other hand, they are sometimes distinguished along the lines of *bioinformatics* being the discipline of developing computational tools for biology and storing biological data, while *computational biology* being the discipline of developing analytical methods, applying tools, and using data for concrete biological research. In practical terms, however, these directions are developed together and can hardly be separated. The blurred distinction between bioinformatics and computational biology can be illustrated with two of the main bioinformatics and computational biology conferences – the *Intelligent Systems in Molecular Biology* and the *European Conference on Computational Biology* – both publishing their proceedings in the journal *Bioinformatics* (Lengauer 1999, 2002, Devignes and Moreau 2014, Moreau and Beerenwinkel 2015).

- **Genomics** (or genome biology) is the study of whole genomes including the sequences, relations between genes, mechanisms of gene regulation, evolution, and variation. In line with genomics, other **omics** disciplines focus on complete repertoires of different kinds of biological molecules or mechanisms, as fields of study or as measurement and recording methods. For example **proteomics** measures the repertoire of proteins present in a sample, and **metabolomics** the small molecules, metabolites. Complementing genomics, **epigenomics** studies the information not included in the genomic sequence itself, but in histone modifications and DNA methylation.
- **Systems biology** studies networks of interacting molecules or other agents in a cell, a cell compartment, tissue, organism, or ecosystem. These networks are typically modelled as mathematical dynamic systems, and the dynamic properties of the involved molecules and other measures are analysed and simulated computationally. One may for example predict concentration of a certain chemical constituent in a given system under given circumstances.
- **Biostatistics** is the statistical component of designing experiments, analysing and interpreting data, and doing predictions within biological disciplines.
- **Cheminformatics** intersects with bioinformatics when it comes to information about chemical compounds present in living organisms, *e.g.* to cataloguing their properties, or inferring their structure.
- **Immunoinformatics** – or computational immunology – applies computational methods including bioinformatics and genomics in immunology.

1.2.   The community of creative chaos

With exception of a few bigger institutes, the bioinformatics community is spread over thousands of independent research groups around the world. These are based at various departments and institutions, most frequently academic, and may be co-located with diverse related research disciplines: typically biology, medicine, biochemistry, computer science, scientific computing, or mathematics, but possibly also with other fields such as geology, marine and water research, or biotechnology. Having the broad common goal of exploring biological mechanisms, researchers have recorded numerous petabytes of data and developed thousands of software tools.

Large amounts of data have been collated in freely accessible public **databases**, provided and maintained by different groups and institutes. The *Nucleic Acids Research* journal's Molecular Biology Database Collection lists in 2015 more than 1500 diverse bioinformatics databases that are available to all researchers and to the general public

(Burks 1999, Baxevanis 2000, Fernández-Suárez *et al.* 2014, Galperin *et al.* 2015). Moreover, in addition to the public databases, many research groups and companies maintain their own private databases dedicated to their research.

The researchers and enthusiasts within the bioinformatics community keep developing **software tools** which encapsulate diverse novel algorithms for processing different kinds of biological data. A majority of these tools is either free and open-source, or at least freely available to academic users or in fact to everyone. The SEQanswers web portal currently includes information about almost 700 software tools (Li *et al.* 2012a). It covers primarily tools for processing sequencing data, and this list is far from being exhaustive.

The story of bioinformatics, however, does not end at developing and using individual tools and databases, but that is rather where it all starts! A bioinformatics (or rather computational biology) analysis needs to combine various steps, using multiple tools and databases. The complete or partial work flow of analysing certain data, with a certain scientific goal in mind, is referred to as an **analysis workflow**. Some workflows or their parts can be fully automated in the form of a computer program or script, running without user interaction from the initial inputs to the final outputs. *Automated workflows* are sometimes called also *pipelines*, but such distinction is not universally established and switched meanings occur, therefore I will avoid the term in the rest of the text. Other parts of workflows that are not automated may include interactive use of software tools or "manual" processing.

Analysis of biological data demands both the **integration** of different types and sources of data, and the integration of diverse software tools. In a particular workflow, the different types of data that are integrated may originate from various *in vivo* and *in vitro* sources, measured or imaged by various technologies, and represented in different formats. In addition, data generated within a particular project are usually compared with data stored in various public or private databases. Diverse computational tools need to be combined while processing the data, often together with steps of manual inspection and handling of the data, trials and errors in designing the workflows themselves, and finding the most appropriate parameters of the involved tools.

Additional special-purpose **scripts** often need to be written for automating particular parts of the analyses. In contrast to multi-purpose software tools, scripts usually aim at fitting a very specific situation. Scripts are often used, for example, in statistical analyses, such as when comparing various data values and finding significant differences, in graphical plotting of intermediate or final results, in data parsing, filtering, and editing.

In many cases, the software tools used in a workflow may run on the user's personal computer. However, a steadily growing portion of life-scientific research demands high

throughput of data analysis. In high-throughput analyses, certain steps of the workflows require time- and resource-consuming computation on powerful supercomputers and with large databases. The high-performance computational resources, in similar fashion to the databases, are provided by certain institutes as **services** that are available to a limited group of local users or publicly, accessible via a local network or the World Wide Web. In summary, bioinformatics workflows require data integration, integration of software tools, scripts, computational resources, services, and databases.

The self-organising character of the heterogeneous bioinformatics community, and the fast responses to emerging technologies, have been resulting in high productivity of novel data and scientific knowledge, accompanied by massive productivity of tools which have been enabling tremendous progress in life sciences. Although there are thousands of bioinformatics tools, databases, and other resources freely available to the whole community, they are not necessarily easy to find, use, compare, evaluate, and integrate with each other in order to find the best and most appropriate and fit them into the researchers' workflows. Researchers analysing biological data spend a substantial portion of their time navigating through the existing "creative chaos" (as coined by Stein 2002) and adapting to it. The downside of the creative freedom has been that the tools from different researchers come in very different forms, flavours, and qualities.

Chasms exist between the quality of documentation, between the ways of distribution, and between the degrees of usability ranging from the few user-friendly tools to ones no one except the author can use. Importantly, computational tools can be available with various types of **interfaces**, for example graphical user interface, command-line interface, web application, plugin to another application, or a programming library. Different types of tool interfaces are useful in different scenarios, and are described in the next section, 1.3 Efforts in mitigating the chaos (p.15). Unfortunately, many tools are only available with one type of interface, and in order to use them in a different way, an additional effort must be made of wrapping them with another interface.

In addition, the input data that are consumed by tools and the output data that are produced, or that can be extracted from distinct databases, vary hugely in the format in which they are represented. Even when common formats are used, they can be used in different ways, due to the flexibility of the formats. Also, the nomenclature inside the data may be used differently and thus cause possibly different understandings. Last but not least, major differences are usual in the presence and detail of accompanying metadata, affecting the practical *reliability* of the data. Efforts in standardising the representation of information are described in a dedicated section, 1.4 Standardising information and data representation (p.41).

Together with integration of tools and data, there is another crucial area of integration challenges: the integration of people, who are the users of bioinformatics tools,

producers of data, or providers of tools. One side is the "human-tool integration", where qualities of the tools – such as *accessibility* and *usability* – turn into either efficiency or effortfulness of the research. This is even more important for those prospective users of bioinformatics tools who are not computer specialists, such as biologists or medical doctors. Also non-researchers, for example secondary-school students, should be able to access and use the most basic publicly available biological data and bioinformatics tools. Another side is the "integration" of people with each other, that is enabling efficient collaboration between scientists, and between specialists in diverse disciplines. Broad collaborations are exemplified in section 1.5 Sharing experience and effort, *p.*46.

1.3.   Efforts in mitigating the chaos

To enable researchers to utilise the abundance of diverse computational tools and data resources more efficiently, several tactics and projects have been developed that focus on improving the *accessibility* and *reliability* of the involved tools and data resources. With the umbrella terms of accessibility and reliability, let us encompass broad and overlapping ranges of *quality aspects* of tools and data, outlined in the following paragraphs. For computational tools, these are also called *non-functional requirements* or *quality attributes*.

Accessibility can in a broad sense cover a set of interconnected qualities such as:

- *Usability*. Tools with good usability are user-friendly, efficient to work with and ergonomic. They minimise mistakes, and have low barrier to learn how to use them. Usability design of a particular tool can focus on a particular type of user and usage scenario.
- *Availability*. Means that tools can be downloaded, installed, and used; or accessed on a server with good response time and sufficient computational power. The usage should be *affordable*, ideally for free, for all scientists and the general public. *Free* and *open-source software* can by definition be used, studied, modified, and re-distributed freely (Stallman 1986, Perens 1997, 1999).
- *Interoperability* and *compatibility* refer to the smoothness of setup and use together with other tools and systems (*integration*): software, hardware, operating systems, programming languages, web browsers, or different types of interfaces (*e.g.* interactive graphical, programmatic, or command shell). Worth emphasising is the ease of using different tools together in a "manual" or automated workflow, and of replacing a tool in a workflow with another.

- *Documentation* available in good quality, and all necessary information easily *findable* (the documentation, binaries, source code, web locations). A relevant tool or resource should be findable for potential users that have not heard about it before.
- *Flexibility* allowing unexpected usage scenarios. This is often referred to as *re-usability*. Flexible tools are efficiently usable by different types of users, smoothly in different scenarios. *Scalability*, *maintainability* (ease of keeping the tool's functionality, its installation, and dependencies up to date), and possibilities to *extend* and contribute to further development can be mentioned as separate qualities related to flexibility.

Reliability is desired with respect to scientific results, data and conclusions, and tools. A high level of reliability can be achieved by satisfying a number of related qualities including:

- *Transparency* of results, computations, algorithms, efficiency, assumptions, of the development and maintenance process, and of weak points. Good transparency can enable *reproducibility*, and can be facilitated by recording *provenance* (the history of data), by availability and good quality of source code, and by sharing information that is not *sensitive.*
- *Confidence* and evidence supported by extensive, well-targetted testing and statistical evaluation, and comparability with similar tools or results.
- Reliable tools and resources should be well *maintained*, stable but up to date and non-volatile in functionality and availability (durable), with good versioning, updating, bug-fixing, and user support; free of unwanted side effects or unexpected behaviour; and well *compared* with related tools, possibly using some benchmarks.

Reliability and accessibility are naturally closely related. Documentation, scalability, interoperability, flexibility, source code availability and quality, robustness (with respect to parameter settings, improper use, high load, or failure), or openness for community participation, can all contribute to both accessibility and reliability of a tool or data resource. For example documentation and evidences – which may include example applications or benchmarks – may advertise a resource in a transparent, reliable way, thus improving its visibility to potential users. Another example, *free* and *open-source* software is available for use, with a good chance to be flexible, well-maintained, and reliable thanks to openness to modification and re-distribution and transparent due to its available source code. In the best case, the whole development of a particular software can be *transparent and participatory*, improving reliability of the developed software, and fulfilling the community's requirements. As a fundamental principle, tactics for making bioinformatics more accessible and reliable do focus on the *user*. The rest of this section lists a number of main approaches to targeting these various quality aspects of bioinformatics tools, together with examples where they are applied. In this way, a non-exhaustive overview of existing efforts is presented.

Approaches related to mitigating the chaos within bioinformatics *data* are presented in the next section, 1.4 Standardising information and data representation (*p.*41), while a short section on *collaborations* (1.5 Sharing experience and effort, *p.*46) closes the *Background* chapter.

Installable applications

Application software may be **available** for users to download and install onto their personal computers or their institution's servers. As the ultimate examples, the all-time most popular bioinformatics tools, Clustal and BLAST, thank their enormous proliferation to being **free** and **open-source**, easy to compile and install in all main operating systems and hardware, well **documented** (both algorithms and implementations), having user support, and being continuously **maintained** and improved until today (Higgins and Sharp 1988, Higgins *et al.* 1992, Thompson *et al.* 1994, 1997, Larkin *et al.* 2007, Sievers *et al.* 2011 for Clustal; and Altschul *et al.* 1990, 1997, Camacho *et al.* 2009 for BLAST).

As an interesting remark, rumours say that the MULTAL algorithm and its implementation (Taylor 1988) was at least comparably fast and accurate as Clustal at the time, but did not gain users possibly due to the lack of accessibility and support. Although MULTAL was free to use and available with its source code, it could still be considered a great *academic prototype*, as opposed to Clustal being an extensively supported and maintained *production software*. Source code that is available and in good quality, well-documented, with build scripts, easy to install, update, or use in other applications and on all main operating systems, with continuous improvements, and a well-supported user community naturally increase the **transparency** and **reliability** of the given software, thus attracting more and more **confiding** users. As opposed to applications available only remotely, locally-installable software is usable also within isolated computational resources handling **sensitive data**, where all or most of remote access is blocked.

Toolkits

To make software more visible to the users, and easier to install, manage dependencies, and use, many tools are provided together as toolkits, called also software *suites*. Tools within a suite are usually developed together, or following shared guidelines, have similar interfaces, and are nicely **compatible** among themselves, covering a certain domain of research. That means that they are easily

usable together in analysis workflows. Developing tools together as a toolkit, if designed carefully, may also make it easier to develop them further, which is a feature of good **maintainability**.

The University of Wisconsin Genetics Computer Group software suite, also known as GCG or the Wisconsin Package (Devereux *et al.* 1984, Womble 1999a), was a toolkit that included implementations of the classical optimal sequence alignment algorithms (Needleman and Wunsch 1970, Smith and Waterman 1981), together with many other tools for analysis of nucleotide and amino-acid sequences. Although initially with public funding from NIH, GCG was developed at the University of Wisconsin as a commercial software with 50% discount for non-profit users, and gained broad popularity. Owned by the Genetic Computer Group Inc. and later Accelrys, GCG became obsolete and no longer maintained or supported since 2008. As a free, open-source alternative to GCG, the development of the European Molecular Biology Open Software Suite (EMBOSS, Rice 1998, Rice *et al.* 2000) started in 1998 based on the work on previous GCG extensions (GCGEMBL and EGCG, Rice *et al.* 1995, 1996), backed by the EMBnet community (Doelz 1992, Harper 1996, D'Elia *et al.* 2009) and initially funded by the Wellcome Trust. Providing hundreds of tools mostly for molecular sequence analysis, EMBOSS was further developed until recently (http://emboss.sourceforge.net/developers/changelog.html), and is still widely used today.

Classical examples of bioinformatics toolkits include also the Staden Package for sequence analysis and assembly (Staden 1977, 1978, 1979, 1986, 1996, Staden *et al.* 1999), PHYLIP for phylogenetics (Felsenstein 1981, 1985, 1989), WHAT IF for molecular structure analysis and modelling (Vriend 1990), the Vienna RNA Package for RNA structure modelling and analysis (Hofacker *et al.* 1994, Gruber *et al.* 2008, Lorenz *et al.* 2011), or Gromacs for molecular dynamics (Berendsen *et al.* 1995, van der Spoel *et al.* 2005, Hess *et al.* 2008, Pronk *et al.* 2013). More recent examples include the highly popular SAMtools for handling and analysis of aligned sequencing reads (Li *et al.* 2009), or GenomeTools developed at the University of Hamburg, which comprise genome analysis tools published separately but available as a coherent toolkit (Gremme *et al.* 2013).

Notably, there is no clear distinction between single software tools and software toolkits. On one hand, each software toolkit can be considered a coherent tool. On the other hand, a particular tool often provides different algorithms for alternative options and for different kinds of input data or usage scenarios, such as in BLAST, especially since the introduction of the re-implemented BLAST+ suite (Camacho *et al.* 2009).

Interactive graphical user interfaces

Application software can be available as executables that read parameters and input data, run the computation, write the output, and close the execution. Also called *command-line tools* or *programs*, these can be executed in a command shell or within a script.

Some applications are on the other hand – or in addition – equipped with an interactive *graphical user interface* (*GUI*), enabling interactive graphical visualisation. Once the graphical user interface is executed, it awaits a succession of user interactions, based most typically on using a pointing device instead of typing commands. Interactive graphical user interfaces thus increase **usability** and **transparency** in scenarios where interactive visualisation is beneficial, and **accessibility** for users that prefer not to type commands or write scripts.

As graphic displays were becoming affordable during the 1980s, interactive graphical visualisation tools started proliferating into bioinformatics, such as within the Staden (Staden 1982, 1984, 1990, Gleeson and Staden 1991) and WHAT IF (Vriend 1990) toolkits. While at the time of the first publication GCG offered graphics only as output printed by plotters (Devereux *et al.* 1984), graphical output on displays became available soon after. The interactive GUI was, however, introduced into the GCG toolkit only in the 1990s in form of the Wisconsin Package Interface (WPI) for the X Window System, followed by SeqLab (Womble 1999a).

Despite of the algorithms for automated alignment of multiple sequences, it turned out early-on that they need to be complemented with visualisation and "manual" editing. Editing of multiple-sequence alignments and their textual visualisation using ASCII characters became available with HOMED (Stockwell and Petersen 1987, Stockwell 1988) and ESEE (Cabot and Beckenbach 1989) editors. Graphical visualisation and editing were enabled soon afterwards, for example in the historical MACAW (Schuler *et al.* 1991), a comprehensive application for constructing alignments, which integrated manual editing with automated methods. Clustal – the all-time favourite multiple-sequence aligner – has since the 1990s been equipped with a GUI named CLUSTAL_X, programmed in C and available for all major operating systems (Thompson *et al.* 1997, Larkin *et al.* 2007). Currently perhaps the most popular graphical editor and analysis tool for multiple-sequence alignments, especially for proteins and RNAs, is Jalview (Clamp *et al.* 1998, 2004, Waterhouse *et al.* 2009, Fig. 1). It is programmed in Java and can thus run on all common operating systems.
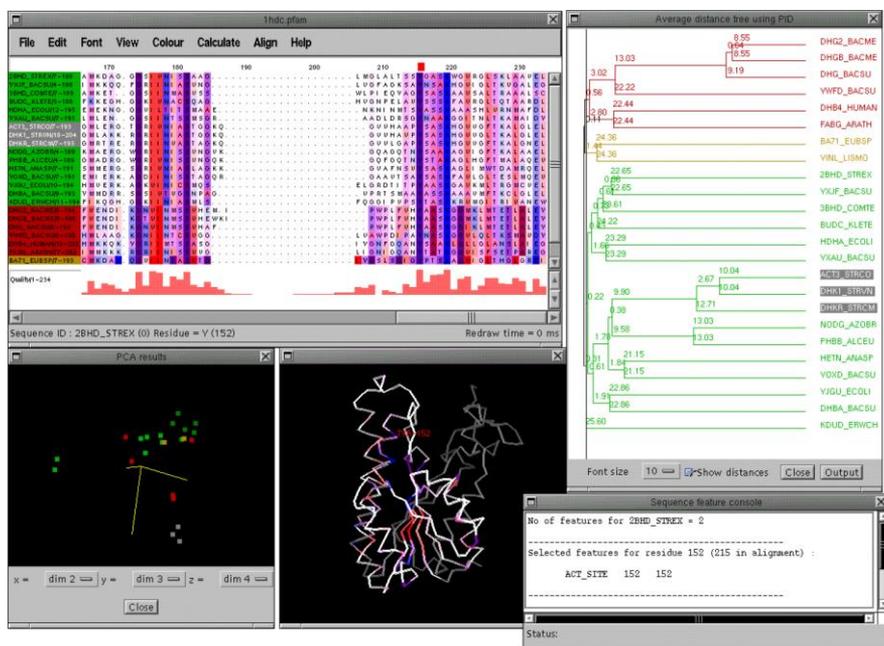
Fig. 1. A historical version of Jalview from Clamp *et al.* 1998**.**

Various GUI applications were developed in Java at the time of its increasing popularity, for example the genome browser Artemis for displaying and annotating whole-genome sequences (Rutherford *et al.* 2000), and J-Express for analysing data obtained from gene-expression microarrays and other high-throughput technologies (Dysvik and Jonassen 2001, Stavrum *et al.* 2008). At the time, J-Express enabled complete gene-expression analysis using statistical algorithms and data visualisations integrated in a relatively **accessible**, **transparent**, and comprehensive graphical application, as opposed to otherwise using a set of partially unpublished scripts such as in the foundational work of Eisen *et al.* (1998).

Interactive graphics are necessary for analysis of 3D structure of biomolecules, provided by multiple applications such as RasMol (Sayle and Milner-White 1995, Bernstein 2000), the popular VMD (Humphrey *et al.* 1996) and PyMOL (http://www.pymol.org), or the ambitious YASARA (http://yasara.org). A few other interesting examples of comprehensive interactive visual tools are Cytoscape (Shannon *et al.* 2003, Yeung *et al.* 2008) and ONDEX (Köhler *et al.* 2006) for exploring networks of interactions and relations such as between various molecules and genes; COPASI for analysing systems biology models (Hoops *et al.* 2006); the Integrative Genomics Viewer (IGV, Robinson *et al.* 2011, Thorvaldsdóttir *et al.* 2013), a genome browser with rich functionality; Utopia Documents (Attwood *et al.* 2010), a PDF reader for scientific articles, that interactively visualises mentioned molecules and active links to other data; and a contemporary tool Caleydo for exploring large heterogeneous data visually (Streit *et al.* 2009, Lex *et al.* 2012).

Web applications

In the previous subsection, I mentioned examples of interactive graphical user interfaces that are either developed as *native applications* compiled specifically for given combinations of operating system and hardware, or are developed for a particular *software framework*. Software frameworks – such as the X Window System, Java, .NET and Mono, or Qt – run on multiple operating systems and hardware architectures. Worth noting is that all these applications are sometimes disputably called "desktop" applications. Originating from the "desktop metaphor" of interactive GUIs, but indicating also specificity to desktop computers as opposed to mobile computers and devices, or computers in racks, such a term is a confusing misconception.

In addition to native applications and applications for multi-platform software frameworks, interactive graphical user interfaces can also be provided as *web applications*. Web applications are developed using a set of complementary languages defined for the *World Wide Web* (WWW, the inter-linked documents on the Internet, Berners-Lee *et al.* 1992). The *standard* languages, governed by the World Wide Web Consortium (W3C, http://www.w3.org, http://www.w3.org/standards), are primarily HTML, CSS, JavaScript, and more. Thanks to using web standards, a web application can run in any *web browser*: historically *e.g.* the break-through graphical Mosaic (Andreessen 1993, Vetter *et al.* 1994), Netscape, or the textual Lynx; nowadays *e.g.* Firefox, Konqueror, Opera, Safari, IE, or Chrome. Naturally, the web browser must comply with the latest versions of the web standards. In addition to **accessibility** and **transparency** fostered by interactive graphics, **compatibility** with standards ensures **interoperability** of web applications, enabling them not only to run on all applicable operating systems and hardware architectures, but also to work together one with another, via *e.g.* links or embedding.

Traditional web applications follow a *client-server* architecture. A rather simple *client* part (frontend) of the web app runs in a user's web browser. Behind the scenes, the client communicates – using HTTP, the communication protocol of the Web – with a *server* (backend) deployed on the side of the *provider* of the web application. The client page itself is located at a given URL of the web app, and automatically downloaded from the server to the user's computer via HTTP, too, increasing the **accessibility** by freeing the user from any installation, dependency management, updating, and usually also paying. The server most often gives **access** to some centralised computational or data resource, employing high-performance "parallel" computers and computer clusters, and making accessible the tools and data that would hardly be usable on local personal computers. A **reliable** server should be **scalable** for high demands and have ideally 100% online uptime (**availability**) with load balancing, a failover system, and enduring maintenance. While some client-server web applications ("web servers") are only provided as a piece of *software* which has to be installed on a server at a user's

institution, more commonly they are provided as a *service*: a deployed server instance with access to provider's computational and data resources – either exclusively or in addition to providing the server software.

The databases of biopolymer sequences were long ago distributed on paper (Fig. 2, *p.*23), followed by magnetic tapes and CD-ROMs. Due to massive growth in volume and increasingly frequent updates, the static media became insufficient. The databases had to start being accessible remotely on a public server, which was more practical due to being always up to date, and at the same time faster than navigating through the locally accessed media. Such servers were accessible consecutively via various network protocols, such as e-mail (Henikoff 1993), Telnet connections, FTP downloads, WAIS text searching and Gopher browsing (Parker 1993, Rice *et al.* 1993). However, to unleash the full power of links between data within and between the diverse bioinformatics databases, integrative portals were soon developed using the new technology of the World Wide Web. Just a couple of years after the Web was invented at CERN in Geneva, ExPASy was launched as the first web server within the life sciences in 1993, as well in Geneva (Appel *et al.* 1994). ExPASy has provided protein sequence data, their 3D structures and features, with mutation and disease information, and annotated images of proteomics gels, in an integrated **user-friendly** way that is still up-to-date today: via the standard web links. More examples of integrative, multi-database data-access web applications appeared shortly after: Entrez provided by at the National Center for Biotechnology Information in Bethesda (NCBI, Benson *et al.* 1990) was after CD-ROMs and a non-web client-server application launched together with the NCBI website in 1994 as a "dynamic" web application built from web forms and inter-linked "static" web pages, named WWW Entrez or WebEntrez (Schuler *et al.* 1996). In the same year, the Sequence Retrieval System (SRS, Etzold and Argos 1993) had its local command-line and its network interface amended with a "dynamic" client-server web application SRSWWW, available for install at users' institutions, and for public access at the European Molecular Biology Laboratory (EMBL) in Heidelberg (Etzold 1994). The European Bioinformatics Institute (EBI) in Hinxton was established during the transition period of 1992-95, as an outstation of EMBL responsible for maintenance and distribution of bioinformatics databases (summarised in Lopez *et al.* 2003). Among other media and protocols, these data were early-on provided via the Web (Emmert *et al.* 1994). Using WWW for client-server communication improved **accessibility** compared to other client-server protocols which could be disabled in certain networks for security reasons. Furthermore, web servers have typically not required users to register and log in.

Besides databases, client-server applications also gave access to computational tools running on shared computational resources, first via e-mail (Henikoff 1993) and later via web apps. WWW2GCG (Colet and Herzog 1996) was the first web GUI to the commercial GCG toolkit, followed by SeqWeb in 1997 with "dynamic" web pages implemented using JavaScript (Womble 1999b). These were client-server web
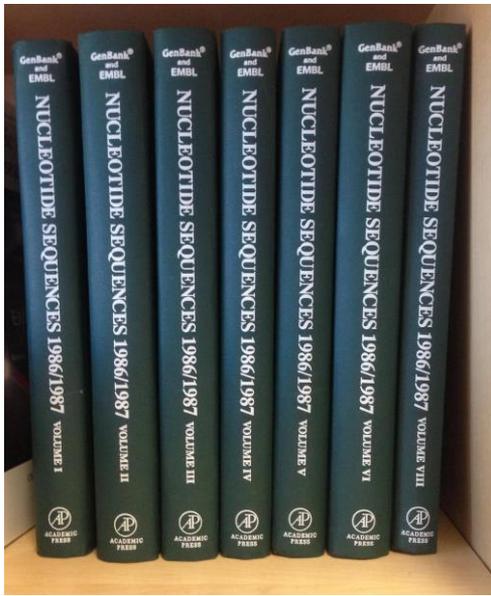
Fig. 2. GenBank and EMBL databases before the Web. Nucleotide sequences 1986/1987, volumes I to VII (David Landsman, Bethesda,). Various network access methods were provided afterwards, until settling down with the World Wide Web in mid 1990s.

applications running on local networks at research institutes, providing access to local GCG servers. In contrast, the WHAT IF toolkit has been provided as a publicly accessible web app (Rodriguez *et al.* 1998). Similarly, PredictProtein has for more than two decades been a user-friendly public server for integrative inference of a growing multitude of protein features: since 1992 as an e-mail server and later on the Web (Rost *et al.* 2004). Further examples of public websites giving access to integrated kits of tools are the Vienna RNA Websuite for the Vienna RNA Package (Hofacker 2003, Gruber *et al.* 2008); BiBiServ, the Bielefeld University Bioinformatics Server hosting tools developed in Bielefeld and elsewhere (http://bibiserv.techfak.uni-bielefeld.de, http://bibiserv2.cebitec.uni-bielefeld.de); and the Center for Biological Sequence analysis (CBS) at the Technical University of Denmark with a broad portfolio of their tools (http://cbs.dtu.dk/biotools, http://cbs.dtu.dk/services). Meanwhile – in the course of the last two decades – the websites of the major providers of bioinformatics databases grew into integrated portals that complement the access to data with numerous web-accessible tools enabling advanced searching and computations with the voluminous public data: *e.g.* NCBI (McGinnis and Madden 2004, Johnson *et al.* 2008, NCBI Resource Coordinators 2015), EBI (Lopez *et al.* 2003, Brooksbank *et al.* 2014, Li *et al.* 2015, Squizzato *et al.* 2015), the National Institute of Genetics in Mishima with the DNA Data Bank of Japan (NIG, DDBJ, Kodama *et al.* 2015), and ExPASy, now maintained within the Swiss Institute of Bioinformatics (Gasteiger *et al.* 2003, Artimo *et al.* 2012). To conclude this paragraph, let me emphasise again that the users of computational tools **available** as public web applications benefit from the **access** to high-performance computing facilities and the good **accessibility** without the need to install and administer necessary software or type commands. The efficiency is maximised when the computational tools are co-located with data resources: both with respect to computation and data transfer, and convenience for users thanks to integrated access.

After the dramatic triumph of *open science* and open-source bioinformatics when assembling the first draft of the human genome at UCSC in 2000 as a free public resource (Kent and Haussler 2001), the need arose to make the genome data **accessible** and efficiently **usable** for all researchers. The UCSC Genome Browser was developed

23

soon after (Kent *et al.* 2002) as a user-friendly web application giving access to numerous annotated genomes, and in addition enabling researchers to upload their own annotations for browsing them visually on a genome together with diverse public annotations. Ensembl, the infrastructure for automated genome annotation, provides another web-based genome browser for a multitude of species (Hubbard *et al.* 2002, Cunningham *et al.* 2015). On the other hand, Gbrowse is a popular web-based genome browser for relatively easy installations on servers dedicated to genomes of a particular species or group of species (Stein *et al.* 2002, Donlin 2007).

Web applications do not necessarily consist of a server and a client. Departure from the traditional client-server architecture is increasingly common among modern web applications that perform more computations themselves – within the user's web browser running on the increasingly more powerful personal computer or device – with less or no help from a remote server. Some web apps are even supposed to be installed and administered locally on a user's computer, but run in a web browser in order to achieve independence from hardware platforms and operating systems. Other apps are automatically downloaded from a web server when a user starts them, but do not communicate with the server while running. They can be updated automatically from the server when needed, thus freeing the user from installation and its maintenance. Other web applications are "server-agnostic", *i.e.* able to connect to multiple remote servers depending on configuration, user's choices, or automatically, offering great **flexibility** and **scalability** via good **interoperability** among the available servers and clients. Such applications often connect to so-called *Web services* which I will describe a couple of pages later (*p.*29). Going in an orthogonal direction, there are possibilities emerging of server-less web apps communicating directly with each other, in a *peer-to-peer* fashion (http://www.w3.org/TR/webrtc).

Some graphical bioinformatics tools are available as Java *applets* which are usually server-less and can be included (*embedded*) inside web applications: for example JalviewLite, a stripped-down version of Jalview (Clamp *et al.* 2004, Waterhouse *et al.* 2009); Jmol for viewing molecular structure (Herráez 2006); or Cytoscape Web and Ondex Web, the applet versions of respectively Cytoscape and ONDEX (Lopes *et al.* 2010, Taubert *et al.* 2013). To avoid the often troublesome need for additional, non-transparent plugins for web browsers, such as Java or Flash, rich embeddable web applications can nowadays be developed using pure web standards: HTML5 (http://www.w3.org/standards/webdesign, http://www.w3.org/TR/html5) supplemented with related web standards such as CSS and SVG, and with JavaScript (not related to Java!) – the programming language that can be run inside HTML pages within a user's web browser. Recent examples of interactive web apps for bioinformatics use JavaScript in way that hardly resembles the JavaScript of GCG's SeqWeb from 1997. JSmol is an HTML5/JavaScript version of Jmol (http://jsmol.sourceforge.net, http://chemapps.stolaf.edu/jmol/jsmol/jsmol.htm), while Jolecule is another HTML5 viewer of molecular structure (http://jolecule.appspot.com, reviewed in Porebski *et al.* 2013). From the abundance of

embeddable JavaScript genome browsers that have been developed, Anno-J (used in Lister *et al.* 2008) is 100% "server-agnostic", connecting to custom Web services. JBrowse is a JavaScript alternative to GBrowse (Skinner *et al.* 2009). It is a client-server genome browser with rich functionality, and can additionally be supplemented with a sequence-annotation editor Apollo (Lee *et al.* 2013). On the other hand, Dalliance is a lightweight genome browser (Down *et al.* 2011), and Genome Maps may in complexity fit somewhere between the two (Medina *et al.* 2013) All these apps can be embedded in other web applications – including user's own web pages – and run in all normal web browsers on all applicable platforms thanks to the **interoperability** achieved by compatibility with web standards. A special attention needs to be given to bioinformatics-specific JavaScript *libraries* of building blocks for developing custom web applications for visualising biological data. These include among others: JBio, an early comprehensive attempt by László Kaján (http://jbio.sourceforge.net); Scribl, a JavaScript library for drawing sequence features (Miller *et al.* 2013); and Cytoscape.js, a JavaScript-based successor of Cytoscape Web (http://js.cytoscape.org). Standing out is BioJS, an initiative and a growing collection of concise JavaScript building blocks for bioinformatics web applications, covering diverse types of bioinformatics data. BioJS components are easy to find, use, develop, contribute, and combine, due to following a set of common, well-designed guidelines, especially since version 2.0 (Gómez *et al.* 2013, Corpas *et al.* 2014, http://biojs.net). Various BioJS components are used together for example in PredictProtein (Yachdav *et al.* 2014). Standards-based components are inherently **transparent** with open source, and ought to be **flexible**, **reusable** in various applications, and **interoperable** with each other.

In this subsection we gave a deserved tribute to the World Wide Web – the "flagship" infrastructure for accessible reliable information and computation. For bioinformatics, WWW has been among the most crucial technologies soon after it was invented. In addition to web applications, *Web services* have been ubiquitous in bioinformatics, and are introduced a couple of pages further. In the end, I mentioned JavaScript libraries for bioinformatics web applications. Although using them for developing custom web apps may often require only minimum programming, they still belong – in addition to interactive visualisation – among programming libraries, which are the topic of the following subsection.

Programming libraries

In the previous two subsections, I wrote about interactive graphical user interfaces that foster accessibility and usability to users who do not feel confident with typing commands, and are usable in scenarios requiring visualisation. Data analysis workflows often require automation of some portions which need to be performed repeatedly, with different input data or parameters. Such portions of a workflow need to be implemented as some sort of a script that can be re-run many times, possibly even in a *high-throughput* fashion with large amounts of input data. As opposed to GUIs and "manual" workflows, it is essential for **usability** as a high-throughput workflow to run **without user interaction**. An automated workflow, however, in most cases needs to use one or more existing tools for analysing the data. The same is true for many tools themselves, that inside them use other underlying tools. For such purposes, the underlying tools have to be **accessible** and **usable** from within other tools and workflows. Tools with a *command-line interface* can be used inside batch scripts, and are accessible as external "native" tools from various programming languages, yet with possible limitations to efficiency, interoperability, and maintainability. For example, input and output data has to be typically sent and received via the file system, which may or may not be desired in a particular workflow, while portability to another system and management of dependencies and their versions can turn close to impossible.

An *Application Programming Interface* (*API*) is an interface to a certain tool, system, or other resource, that provides *programmatic access* from one or more programming languages (for example Python, R, Java, JavaScript, C, C++, Perl, Haskell, or Ruby to name a few). An API is often implemented as a *library*, a collection of operations, functions, data structures, and other objects in a particular programming language. A library can be available with or without its source code, and its interface can be used directly in users' programs or scripts in the given programming language, as opposed to calling external commands. Programming libraries – as APIs to computational tools or other resources – can either be provided separately from the tool or resource; or they can be part of the tool itself, often constituting the core of the tool's implementation, that other interfaces are built upon. A *language binding* for a library is some sort of a "wrapper library" in a different programming language than the "built-in" language of the original library, enabling the original library to be used from the other programming language.

Many bioinformatics tools and toolkits are implemented as an open-source core library, with other interfaces – such as command-line, GUI, or web app – built on top of it. While using such a straightforward architecture, these tools are inherently **accessible** via *multiple types of interfaces*, **usable** in various scenarios, **transparent** with their open source code, and more interfaces can be developed by anybody who wants to implement them, thanks to the public API of the core library. In addition, such

libraries are often proven **reliable** by usage in numerous tools. The core libraries are in many cases implemented in C or C++ for runtime speed, while language bindings may be provided for various other programming languages. This is the case in a great number of examples. To list some: the Vienna RNA Package has been built upon its core C library RNAlib (Hofacker *et al.* 1994), and later complemented with a Perl binding (Lorenz *et al.* 2011); SRS was implemented with a core C library suitable for APIs also for Perl, Tcl, and Python (Etzold *et al.* 1996); EMBOSS includes a layer of a C library called AJAX (Rice 1998; not the later "Asynchronous JavaScript + XML" Ajax) which has been used by numerous types of interfaces; SAMtools are constituted as a C library (Li *et al.* 2009), amended with command-line interface and numerous language bindings; and GenomeTools consist of multiple tools implemented around the libgenometools C library, distributed altogether as a package, with an additional API for scripting language Lua (Gremme *et al.* 2013).

In addition to such tool-specific libraries serving as APIs to given tools, various programming libraries aim to cover the broad field of bioinformatics or its parts, from a perspective of a software developer who implements new bioinformatics tools, or a computational biologist who writes scripts for their analyses. Numerous C++ libraries have been developed, that provide substantial portions of typical bioinformatics operations in a programmatic way: for example an early PDBlib for structural bioinformatics (Chang *et al.* 1994), and more sequence-oriented or generic ones such as BTL (Pitt *et al.* 2001), Libsequence (Thornton 2003), libcov (Butt *et al.* 2005), Bio++ (Dutheil *et al.* 2006), or the modern SeqAn optimised for speed (Döring *et al.* 2008). An extensive NCBI C++ Toolkit (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/CPP_DOC) comprises programmatic tools for sequence analysis and data retrieval, together with numerous data-handling and server utilities not specific to bioinformatics.

To avoid the need of always programming one's own new scripts from scratch for particular analysis workflows, and instead provide commonly shared reusable building blocks for both workflows and application development, BioPerl was initiated as a community effort in 1995, when Perl was the most popular language for scripting in bioinformatics (Chervitz *et al.* 1998, Stajich *et al.* 2002). Within the shared effort with a substantial level of self-organisation, BioPerl quickly evolved into a comprehensive toolkit library of well-integrated, **reusable** Perl modules for bioinformatics, that are smoothly **interoperable** with each other, easy to understand, developed in a similar style, and share common data representations. It offers functionality such as handling, parsing, transforming, and integrating data, or programmatic access to popular data resources and analysis tools – serving the typical needs of "glue code" in computational biology workflows, whether "manual" or high-throughput, and in bioinformatics applications. In the same spirit as BioPerl, community efforts followed soon with other popular programming languages, conceiving BioJava (Pocock *et al.* 2000, Holland *et al.* 2008, Prlić *et al.* 2012) and Biopython (Chapman and Chang 2000, Cock *et al.* 2009), later joined by BioRuby (Goto *et al.* 2010). These initiatives – together nicknamed Bio* or

Open-Bio – united under a common umbrella of the Open Bioinformatics Foundation (O|B|F or OBF, http://open-bio.org, reviewed in Mangalam 2002), together with other projects including EMBOSS, and attempts enabling certain scenarios of interoperability between the Bio* libraries, *e.g.* BioCORBA (http://www.bioperl.org/wiki/BioCORBA) and BioSQL (http://www.biosql.org). O|B|F supports and promotes free/open-source software within bioinformatics, and organises an annual Bioinformatics Open Source Conference (BOSC, Harris *et al.* 2015) and various "hackathons" gathering communities of collaborating software developers (*e.g.* Möller *et al.* 2013, 2014). Complementing the popular programming languages, enthusiasts develop integrated library toolkits for bioinformatics also in various "niche" languages, creating for example Biohaskell (http://biohaskell.org), BioClojure (Plieskatt *et al.* 2014), BioSmalltalk (Morales and Giovambattista 2013), or Biocaml (http://biocaml.org). The former Microsoft Biology Foundation (MBF) library for the .NET platform transformed into a free and open community effort .NET Bio (http://bio.codeplex.com).

Numerous biology-related libraries have been developed for R, the programming language that is particularly convenient for statistical analyses. One example is the comprehensive APE (Analysis of Phylogenetics and Evolution, Paradis *et al.* 2004). Using a slightly different approach than the integrated toolkit libraries of Bio*, Bioconductor was conceived as an even more open collection of R libraries for computational biology (Gentleman *et al.* 2004). Bioconductor libraries ("packages") are more independent from each other, while still following common guidelines and reusing common utilities in order to maintain a certain degree of interoperability and other qualities. With the richness of libraries available either on CRAN (http://cran.r-project.org) or Bioconductor, R grew into perhaps the today's most popular scripting language for data analysis and plotting in computational biology. Bioconductor-inspired Biogem and its dedicated repository (http://biogems.info, Bonnal *et al.* 2012) enable modular extensions to BioRuby, that are less tightly integrated and thus easier to develop in comparison to the integrated core of BioRuby. **Accessibility** for novice contributors is fostered by the automation provided by Biogem. While scripting and niche languages may be slower at runtime than "native" C and C++ due to their high-level constructs, and generic-purpose libraries may be less efficient at both runtime and "development-time" due to their complexity compared to narrowly specialised ones, they enable easy and quick development of user's workflows and applications, which may often be higher priorities than runtime efficiency or maintainability.

Let us now shortly get back to interactive graphical user interfaces. Towards the end of the previous subsection, I mentioned JavaScript libraries for programming web applications. BioJS – again especially since its version 2.0 (Gómez *et al.* 2013, Corpas *et al.* 2014, http://biojs.net) – is another example of an open collection of community-developed libraries, sharing the right minimal set of common guidelines for ensuring interoperability, so that the BioJS components can easily be combined together in users' custom applications. For programmatic integration, JavaScript APIs can also be

provided with Java applets, for example with JalviewLite (Clamp *et al.* 2004, Waterhouse *et al.* 2009) or Cytoscape Web (Lopes *et al.* 2010). Libraries for other programming languages also provide functionality for both static and interactive visualisations, including *e.g.* the Bio* and the NCBI C++ toolkits. These can as well be used for developing interactive GUIs, or client-server web apps with graphics generated on the server. A couple of example graphics libraries for drawing genomic data are GenomeDiagram integrated in Biopython (Pritchard *et al.* 2006); AnnotationSketch, a C library within GenomeTools, with Lua, Python, and Ruby bindings (Steinbiss *et al.* 2009); and Circleator using BioPerl, SVG, and CSS (Crabtree *et al.* 2014).

Web services

First to make the terminology clear: Any computational tool or data resource that is *not* provided in form of *software* that users would have to install on their side, but is instead *deployed* and running on a server of its *provider*, is a computational or data *service*. And if the access to the server is via the Web, it could in fact be broadly called a "web service". Thus a client-server web application – running at a provider's web server and accessible for users through web browsers – is after all in rather general terminology a "web service". In contrast, a *Web service* (often spelled with a capital, what we will follow) provides *programmatic access* – *i.e.* a programmatic API – to a computational or data server, over the Web. Occasionally, the term "Web service" was used to designate only Web services that used SOAP protocol (SOAP services), while the Web services using pure HTTP protocol would then be called web APIs, HTTP APIs, HTTP services, "REST" APIs, "REST" services, or "REST" resources. We will not follow such a confusing, unpractical distinction. Instead in line with the more common terminology, let us call all programmatic APIs over the Web synonymously *Web services* with a capital 'W' or *web APIs*.

Notably, *interactive graphical web applications* – serving human-computer interaction – are as a type of interface disjoint with *Web services* which serve interfaces for other applications and scripts (Table 1, *p.*30). For maximum simplicity, we can say that if a web server provides us (via HTTP because it is a *web* server) with something formatted in HTML, then it is a *web page* (static) or a *web application* (dynamic); and if it provides us with something in another format, one that is suitable for "machine" consumption, then it is a *Web service*. Naturally, one web server can serve both web application(s) and Web service(s). However, in case a web resource provides only HTML format, *i.e.* for "human" consumption, but we still need to retrieve some of its data automatedly in our script or application, we need to painfully "dig" it from the often-changing and unsuitable HTML page, in an unmaintainable procedure called also "screen scraping" and coined "mediaeval torture" by Stein (2002).

types of tool interfaces	user interface (human-computer interaction)	partially supporting both "humane" and programmatic access	API (programmatic access)
running locally (or on a local server)	interactive application installed locally	command-line program	programming library
accessed remotely via the Web	client-server web application	"HTTP GET" service	Web service

Table 1. A simplified distinction of Web services and their relations to other types of tool interfaces. Note, however, that there are no precise borders (symbolised by the grey dotted lines) between local and remote applications, because remote access involves something running locally, and a local app may communicate with remote resources or be deployed from a remote resource. Hybrid apps with extensive local and remote portions have been increasingly popular, including server-agnostic apps, peer-to-peer networks, "fat" clients, and ubiquitous "hidden" use of external Web services. In addition, we can access via the Web and HTTP also locally-deployed web applications and Web services, which can be useful not only for testing but also for interoperability in certain scenarios.

As opposed to web applications, Web services provide programmatic APIs **accessible** from a user's *high-throughput* workflow in any of the common programming or scripting languages, and from other applications, facilitating **flexibility**. For better **accessibility** compared to other remote APIs, the communication with Web services is over the Web (*i.e.* HTTP) instead of other protocols which may be blocked, and typically does not require user accounts. **Interoperability** with most of the applicable programming languages and command shells is enabled by available utility software and libraries **compliant** with the *Web-service standards* governed by the World Wide Web Consortium (http://www.w3.org/standards/webofservices). Web services deployed on an appropriate server provide **interoperable access** to high-availability high-performance **scalable** computing resources and big databases, without cumbering the users with need to obtain and administer such resources or install and maintain the tools. However, to allow maintainability of tools that use the Web services, reproducibility of workflows, and to deserve users' confidence, providers must support their users and carefully maintain their services up-to-date but stable and non-volatile – with strict versioning of the interface, preferably even keeping deprecated versions alive.

Historically, various predecessors of Web services were providing programmatic access to remote bioinformatics resources, using various communication protocols. Perhaps the most widely used and most accessible at the time were e-mail servers, providing both "human" users and software applications with access to remote data and computational tools (Henikoff 1993). Ahead of its time was the sophisticated HASSLE protocol (Hierarchical Access System for Sequence Libraries in Europe), developed specifically for bioinformatics needs by Reinhard Doelz at Biozentrum, University of Basel (Doelz 1994, Doelz *et al.* 1994). It included automated search for available services within the network of sequence-data servers around Europe, batch remote execution with automatic failover, and a client user interface hiding all the sophisticated technicalities. CORBA was developed as an industrial technology for distributed object-oriented software systems. In bioinformatics, CORBA was used for

access to databases with genome maps (Hu *et al.* 1998, Jungfer and Rodriguez-Tomé 1998, Barillot *et al.* 1998, 1999); and a system for wrapping bioinformatics tools as CORBA APIs was developed by Martin Senger (1999) at EBI, named AppLab and used inside the later Soaplab, until Soaplab2 in 2007 (Senger *et al.* 2003, 2008). The Bio* initiatives developed BioCORBA for compatible distributed capabilities among BioPerl, BioJava, and Biopython (http://www.bioperl.org/wiki/BioCORBA). Java RMI – a lighter-weight remote API framework for Java only – has also been tried for distributed bioinformatics applications (Möller *et al.* 1999, Saqi *et al.* 1999). All these technologies required special network protocols other than the HTTP of WWW, causing difficulties to software administration and usage, such as being blocked in certain networks.

Proper Web services over HTTP began to flourish soon after being introduced in bioinformatics in the beginning of this millennium. DAS, the Distributed Annotation System, is a system for accessing sequence annotations from a large number of online resources, via HTTP Web services providing data in a dedicated XML format (Dowell *et al.* 2001, Prlić *et al.* 2007, Jenkinson *et al.* 2008). BioMoby was developed as special protocol on top of SOAP, HTTP, and XML for any kind of bioinformatics Web services and types of data (Wilkinson and Links 2002).

Numerous SOAP services were soon launched at various institutions (*e.g.* Kawashima *et al.* 2003, Krishnamurthy *et al.* 2003, Wang and Mu 2003, Crass *et al.* 2004), including the major providers of bioinformatics databases and tools, where SOAP has usually later been complemented or sometimes replaced by pure HTTP services. Early examples are NIG in Mishima providing access to DDBJ, other databases, and computational tools (Sugawara and Miyazaki 2003, Kwon *et al.* 2009); and EBI, including the Soaplab framework (Senger *et al.* 2003, 2008) which provided Web-service access to the EMBOSS toolkit (Rice *et al.* 2000), other Web services for access to EBI's databases and related tools (Harte *et al.* 2004, Pillai *et al.* 2005, Labarga *et al.* 2007, McWilliam *et al.* 2009, Squizzato *et al.* 2015), and later the JDispatcher framework for computational and data-searching Web services (Goujon *et al.* 2010, McWilliam *et al.* 2013, Li *et al.* 2015). Entrez Programming Utilities include Web services for accessing data at NCBI (NCBI Resource Coordinators 2014, NCBI Resource Coordinators 2015). Integrative, easy-to-use TogoWS services for retrieving and converting data are provided by the Database Center for Life Science (DBCLS) at the University of Tokyo and NIG (Katayama *et al.* 2010), while the ExPASy portal of SIB includes among other Web-service-accessible resources – and EMBOSS via Soaplab2 – also an HTTP Web service for integrative querying over a big portion of the provided databases (Artimo *et al.* 2012). Examples of providers of web-accessible bioinformatics tools, offering programmatic access to numerous Web services, are: the WHAT IF toolkit at the Radboud University Nijmegen (Hekkelman *et al.* 2010); the G-language Genome Analysis Environment (GAE) framework at Keio University with Web-service APIs (Arakawa *et al.* 2010); CBS at the Technical University of Denmark (http://cbs.dtu.dk/services/ws.php, http://cbs.dtu.dk/ws/doc); and BiBiServ of the Bielefeld University (http://bibiserv.techfak.uni-bielefeld.de, http://bibiserv2.cebitec.uni-bielefeld.de).

Web services are convenient for remote access to distributed resources especially if they have similar interfaces – with the same operations and formats of input and output data – thus being **interoperable** with each other. Interoperable Web services are conveniently **usable** together in automated workflows, comparable, and replaceable with each other (although that is of course not limited to Web services but holds for all kinds of programmatically usable tools). Web services usually share interfaces within an institution providing them, but it is seldom the case between different institutions. Exceptions exist, such as the DAS resources; PSICQUIC (Aranda *et al.* 2011, del Toro *et al.* 2013), the common Web-service interface to numerous databases of molecular interactions, standardised by the Human Proteome Organization's Proteomics Standards Initiative (HUPO-PSI, Martens *et al.* 2007); or the Web services of BioMart, a framework for uniform access to distributed bioinformatics databases (Kasprzyk 2011, Smedley *et al.* 2015).

Other than being useful within analysis workflows encoded in a researcher's scripts, Web services are ubiquitously used behind-the-scenes inside bioinformatics software. Remote access from within one application to other tools and data resources was common already in the old-days e-mail servers (Henikoff 1993); and is enduringly popular with DAS, accessed among others from Dalliance, IGV, UCSC Genome Browser, Ensembl, Gbrowse, or Jalview. Interestingly IGV, together with many other genome browsers, can access data from custom HTTP or FTP services in addition to DAS. The interactive reader Utopia Documents retrieves information and data underlying a scientific publication naturally via Web services (Attwood *et al.* 2010). Jalview could access remote computational tools at EBI and data via SRS already since its early versions (Clamp *et al.* 1998), and nowadays is complemented with dedicated JABAWS framework (Java Bioinformatics Analysis Web Services, Troshin *et al.* 2011), enabling deployment of new JABAWS-compatible Web services at users' sites, another example of smooth interoperability.

Catalogues, registries, and repositories

One of reasons for the creative chaos in bioinformatics is that it may often feel more straightforward to develop one's own new solution for the current purpose, compared to looking for what is available, what it does, and how it does it – what may often be onerous. And what is onerous for a group of researchers carrying out a project, can well be even more onerous for the ones reviewing their publication. Even worse it can get in situations when a researcher has no clue whether there is anything at all available and helpful for the given task. Despite (or maybe due to) the literature tsunami in life sciences, such scenario can happen easily – irrespective of whether it is a junior researcher not yet up-to-date, a senior researcher not up-to-date anymore with the new creations, or an expert in other subdomains of the field. While developing

one's own do-it-yourself single-purpose solutions may have obvious benefits in the degree of control and in fitting the purpose exactly, these contribute to the abundance of developments that are not well reusable, not well documented, transparent, reliable, or reproducible, and hardly accessible, interoperable, or maintainable. Decreasing the substantial burden of **finding** relevant tools is one of the purposes of catalogues, registries, and repositories. Another purpose of such collections is listing and **advertising** achievements of a certain project or institution.

Although the terms are often used arbitrarily or interchangeably (together with *e.g.* list, directory, or archive), it may be useful if we distinguish for clearer understanding:

- *Catalogues*, created by a group of authors who provide the published content using some sources, and who may or may not continue updating – *curating* – the content
- *Registries*, where *users* contribute the content over time – for example *registering* information about a tool they developed – and curate parts of the content
- *Repositories*, where software or other resources are deposited and can be obtained from. Repositories can of course also *register* or *catalogue* information, and software can be deposited as source code or binaries.

While vendors' catalogues often list commercial products, public registries and repositories are usual for free open software. Some of them do among other application domains contain also bioinformatics tools. This is the case of SourceForge and now growingly GitHub repositories that host big portions of open-source projects in bioinformatics, while Download.com and Softpedia list only few downloadable bioinformatics tools but include some commercial ones. The bioinformatics section of the non-commercial DMOZ registry (http://www.dmoz.org/Science/Biology/Bioinformatics) lists a considerable number of bioinformatics resources of various kinds, including both free and commercial tools. The Free Software Directory (http://directory.fsf.org) of the Free Software Foundation (FSF) is a substantial registry with rich semantic annotation, but contains little bioinformatics. Some programming languages have the available libraries organised in convenient centralised repositories (*archives*), which include substantial amounts of bioinformatics libraries for the given language: CPAN for Perl (http://www.cpan.org), CRAN for R (http://cran.r-project.org), RubyGems for Ruby (http://rubygems.org), and Hackage for Haskell (http://hackage.haskell.org). Multiple application-domain-agnostic public registries were developed for Web services during the "SOAP rush" of the previous decade, with ambitious features (*e.g.* that time's registry from Seekda or http://www.membrane-soa.org/soa-registry), but to my knowledge none withstood the course of time without deterioration.

Within the domain of bioinformatics, bigger institutes usually publish catalogues advertising the tools and databases the institute provides (*e.g.* NCBI at http://ncbi.nlm.nih.gov/guide/all, EBI at http://www.ebi.ac.uk/services, SIB via ExPASy at http://expasy.org, or

the Weizmann Institute of Science in Rehovot at http://miw.weizmann.ac.il). Similarly, distributed infrastructures such as DAS maintain registries of **compatible** Web services (http://dasregistry.org); shared library efforts that follow common guidelines register the **compliant** libraries (*e.g.* Bioconductor at http://master.bioconductor.org/packages/release and BioJS at http://biojs.io); and initiatives such as O|B|F document their achievements and affiliated projects (http://www.open-bio.org/wiki/Projects).

More representative selections of bioinformatics tools – not specific to a project, network, or institution – have been created in various forms ranging from journal articles (*e.g.* Gilbert 1998, 1999, online at http://iubio.bio.indiana.edu/soft/molbio/Listings.html) to websites, from personal listings (such as the spreadsheets I made for myself in order to write this chapter) to global projects. A great number of catalogues, registries, and repositories is available within the field, with substantial differences in types of tools or other resources they collect, in the amount and type of **information** they provide about the listed items, and in functionality they enable: ways of searching, accessing, exporting, or other.

The IUBio Archive for Biology – conceived in 1989 and maintained by Don Gilbert (http://iubio.bio.indiana.edu) – is a historically valuable archive of downloadable software and other resources. Bio Catalog (also Bio-Catalog or BioCatalog, Rodriguez-Tomé 1998, archived at http://iubio.bio.indiana.edu/soft/biosoft-catalog) was a catalogue of software for molecular biology and genetics, developed since 1992 within Généthon, co-founded by CEPH (http://www.cephb.fr/en), and later maintained at the EBI. In a similar style, DBcat was constructed at Infobiogen with contribution from Centre National de Séquençage and Généthon (Discala *et al.* 1999, 2000). Around the same time, Christian Burks created the Molecular Biology Database List (Burks 1999) of databases published in the *Nucleic Acids Research* (NAR) journal's annual special issue dedicated to databases (Bateman 2005, Galperin *et al.* 2015). This list has since been updated annually with the NAR Database Issue, under changing names and by changing maintainers (*e.g.* Baxevanis 2000). Several database catalogues were developed until today, storing both overlapping and distinct types of information about the databases, for example: BioRegistry with annotation generated from other resources, including rich **attribution** data and terms from the MeSH vocabulary (http://bioregistry.loria.fr, Devignes *et al.* 2010); MIRIAM Registry with **monitoring** of online availability (Juty *et al.* 2012); BioDBCore catalogue at the BioSharing portal (Galperin and Fernández-Suárez 2012); or the Integbio Database Catalog merging information from other Japanese database catalogues (http://integbio.jp/dbcatalog/en).

In the last paragraph, let me mention a few influential collections of different types of tools or information. Bioinformatics Links Directory is a catalogue of web links to bioinformatics tools and databases (Fox *et al.* 2005, Brazas *et al.* 2012), including ones published in another NAR's annual special issue, the Web Server Issue (Benson 2007, 2015). The Bioinformatics Links Directory has only limited information and navigation

functionality, but catalogues links to thousands of tools. myExperiment (Goble *et al.* 2010) is a repository of automated workflows defined in specific workflow languages (mostly graphical) executable in particular workbenches. BioCatalogue (opposed to previously mentioned Bio Catalog) is a registry for bioinformatics Web services (Bhagat *et al.* 2010) with community annotation inspired by social websites. In some cases, an internally-maintained catalogue of tools for computational biologists at a sizeable research institute may – in addition to its main purpose of serving the internal users – present a useful representative list with rich institute-unspecific information about numerous bioinformatics tools: for example the Weizmann Institute's BioPortal (http://bioportal.weizmann.ac.il/toolbox/overview.html). Special cases are registries that are maintained openly by their users in form of *wikis*, with a combination of structured information and free text with further documentation and comments such as users' experiences. Within bioinformatics, the main such example is the Software Hub of the SEQanswers wiki (SEQwiki, http://seqanswers.com/wiki/Software, Li *et al.* 2012a) dedicated to software for analysing sequencing data. The last example catalogue is OMICtools (Henry *et al.* 2014), a publicly accessible portal with contents owned by a small company STATSARRAY LLC. It provides information about thousands of bioinformatics tools, categorised and searchable as steps in typical computational biology workflows for analysing several types of "omic" biological data. Although limited to a set of stereotypical workflows, it offers this way a visual aid for more **accessible** navigation.

Workbenches

The term *workbench* originates from an analogy with actual workbenches for manual work. A workbench provides a stable, heavy-duty platform on top of which the work can be done conveniently. Various tools such as hammers, wrenches, or vices can be used on a workbench, attached to it, or possibly stored in some integrated toolboxes (Fig. 3). It is a user's choice which tools they use on a workbench, as long as the tools fit.

Fig. 3. A workbench. With "integrated" tools and "workflow recipes" (top left). © 2015 John Crane Ltd. Fair use.

35

Workbenches for bioinformatics and computational biology follow the same principles as workbenches for manual work. A bioinformatics workbench provides an *integrated analysis platform* which aims at enabling **convenient** data analysis, minimising user's effort. Various computational tools and data resources can be used in a workbench. In the best case, a user can add the tools they need, as long as they are somehow compatible with the workbench. However, adding custom tools requires effort with most workbenches. In workbenches that are publicly accessible over the Web, selections of tools are provided, covering the domain of research a workbench targets (*e.g.* sequence analysis and evolution, structure bioinformatics, or genomics). On the other hand, the workbenches that are installable at a user's local facility come often bundled with a "start kit" of main tools for the given domain.

To enable a convenient data analysis, workbenches integrate other essential functionality, in addition to computational ***tools*** and data services. They may include ***data*** management, visualisation, storage, or occasionally editing; management and execution of automated ***workflows***, workflow design, or scripting; and access to ***high-performance computing*** facilities.

Workbenches usually provide an **accessible** interactive graphical user interface – typically in form of a web application – providing the integrated tools and analysis functionality with a unified look-and-feel, mutual interoperability, and **usability** without typing commands or scripting (Fig. 4, *p.*37, Fig. 5, *p.*39). Other forms of accessing the integrated functionality of a workbench may, however, be included in addition to GUIs, allowing **flexibility** and accessibility for various groups of users and usage scenarios.

Workbenches often include functionality that aims at enabling **transparency** and **reproducibility** of the performed analyses: for example recording analysis steps (the workflow), details of the particular steps, provenance metadata; or enabling users to add human-written documentation. Such documentation, together with the performed workflow and used and obtained data, can often be shared publicly, enabling convenient **publishing** of transparent and reproducible results. In addition, various resources such as data and workflows can be shared between individual users or user groups, a useful functionality for **collaborative** work. Tools compatible with a particular workbench can usually be published in dedicated repositories, enabling **sharing** of effort of making the tools compatible (*i.e.* typically *wrapping* them with a given kind of interface).

From historical examples other than the various toolkits popular through the history of bioinformatics (*p.*17), HASSLE (Doelz 1994, Doelz *et al.* 1994) was a highly sophisticated system integrating distributed resources around Europe, far ahead of its time. GDE (Genetic Data Environment, Fig. 4) was an interactive graphical workbench for multiple sequence alignment (Smith *et al.* 1994, Eisen 1997), while SeqPup was an interactive graphical sequence editor (Gilbert 1999), both with access to custom computational

tools. HUSAR (Heidelberg Unix Sequence Analysis Resources, Senger *et al.* 1995) is an institution-specific system at the German Cancer Research Center in Heidelberg, based on GCG (Devereux *et al.* 1984) and with restricted access, still functional today. Vector NTI was a complex and extendable commercial workbench covering a broad spectrum of bioinformatics (reviewed in Lu and Moriyama 2004).

Since the beginning of the 21st century, the development of integrated analysis systems thrived in bioinformatics, resulting in a plethora of workbenches with diverse specialisations and designs. These include expandable, multi-functional interactive GUIs (more on *p.19*) that are rather narrowly specialised for a certain type of data: *e.g.* ones for molecular structure analysis, Jalview (Clamp *et al.* 1998, 2004, Waterhouse *et al.* 2009, Troshin *et al.* 2011) with functionality comparable to GDE but state-of-art, Norwegian J-Express for gene expression and similar analyses (Dysvik and Jonassen 2001, Stavrum *et al.* 2008) and MotifLab for analysis of regulatory regions in genomes (Klepper and Drabløs 2013), or the popular Cytoscape for analysis and visualisation of networks (Shannon *et al.* 2003, Yeung *et al.* 2008, Lopes *et al.* 2010).

Workflow systems focus on functionality including the design of automated workflows, their administration and execution. These are for example the well-known Taverna (Oinn *et al.* 2004, Hull *et al.* 2006, Wolstencroft *et al.* 2013), or from the newer ones *e.g.* the easy-to-use Armadillo (Lord *et al.* 2012) with data management and visualisation, and a pretty graphical workflow editor.

Workbenches available for use on publicly accessible web servers reached a considerable level of popularity, especially the comprehensive GenePattern (Reich *et al.* 2006) and Galaxy (Giardine *et al.* 2005, Goecks *et al.* 2010), both with active communities of users and contributors. In addition to access at the public web servers, these workbenches can be installed locally on a user's computer or an institute's server.
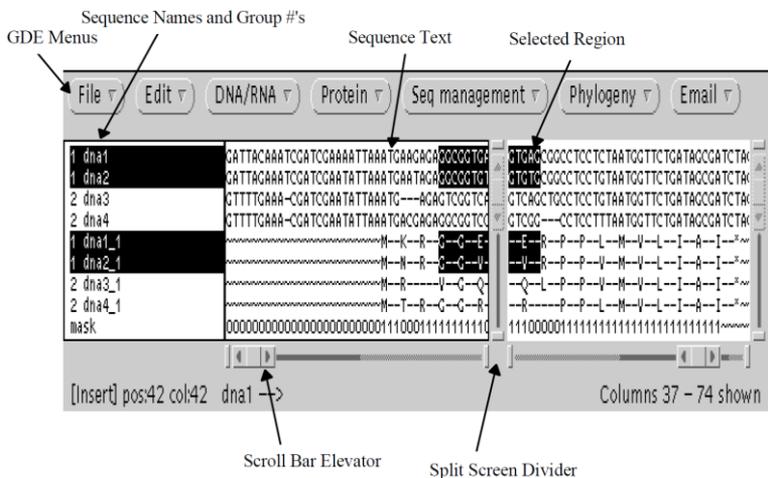
Fig. 4. Screenshot of the GDE workbench from Eisen 1997.

Thanks to a well-targetted community building and promotion, instances of Galaxy were deployed at various sites, with various sets of tools available. Such instances are often locally customised versions, *e.g.* the publicly accessible Genomic HyperBrowser (Sandve *et al.* 2010, 2013). Some institution-specific systems for access to local high-performance computing resources use tweaked versions of Galaxy, for example at the Institut Pasteur (slides http://wiki.sb-roscoff.fr/ifb/images/c/cc/Galaxy_Day_Institut_Pasteur.pdf), occasionally replacing single-site "home-made" solutions, such as at the University of Oslo where the new Galaxy-based LifePortal (http://lifeportal.uio.no, Kumar *et al.* 2015) replaced the previous, easy-to-use BioPortal (Kumar *et al.* 2009) with a simple web user interface.

Institut Pasteur and other sites provide also Mobyle, a popular workbench for sequence and structure analysis, with convenience features such automatic data retrieval and re-formatting, or suggesting tools and operations for the next step within a workflow (Néron *et al.* 2009). Chipster is a powerful workbench provided by the Finnish CSC - IT Center for Science, with extensive support for scripting and graphics (Kallio *et al.* 2011). Likewise the previous ones, Chipster is open-source and installable for free, with a restricted-access instance at CSC (http://chipster.csc.fi/access.shtml). UGENE (Okonechnikov *et al.* 2012) is another free and open-source, locally installable workbench that gained certain popularity, with optional commercial support. An interesting system is GenomeSpace, going one level up and integrating various workbenches and other tools, with convenient data management and sharing (http://genomespace.org, posters Reich *et al.* 2013, Garamszegi *et al.* 2015).

Non-free commercial systems are for example the CLC Bio workbenches (http://clcbio.com), or the user-friendly Geneious (http://geneious.com), with an old, slightly limited version available for free as Geneious Basic (Kearse *et al.* 2012, Fig. 5). BaseSpace is a comprehensive, accessible, and easy-to-use environment for computational biology (http://basespace.illumina.com). BaseSpace is free for use, with charging announced for data above 1TB, providing access to numerous free and non-free tools, mostly non-transparent.

Notable among recent developments for convenient deployment and execution of automated workflows – with Linux command-line tools – in high-performance computing facilities are *e.g.* Arvados and Nextflow. Arvados is a freely installable open-source system with functionality including data versioning and parallelisation, additionally provided as a commercial service (http://arvados.org). Nextflow is a free and accessible tooling for deploying and executing automated workflows on a growing number of supported cluster systems, with support for various scripting languages (poster Di Tommaso *et al.* 2014, update on slides http://speakerdeck.com/pditommaso/nextflow-a-tool-for-deploying-reproducible-computational-pipelines).
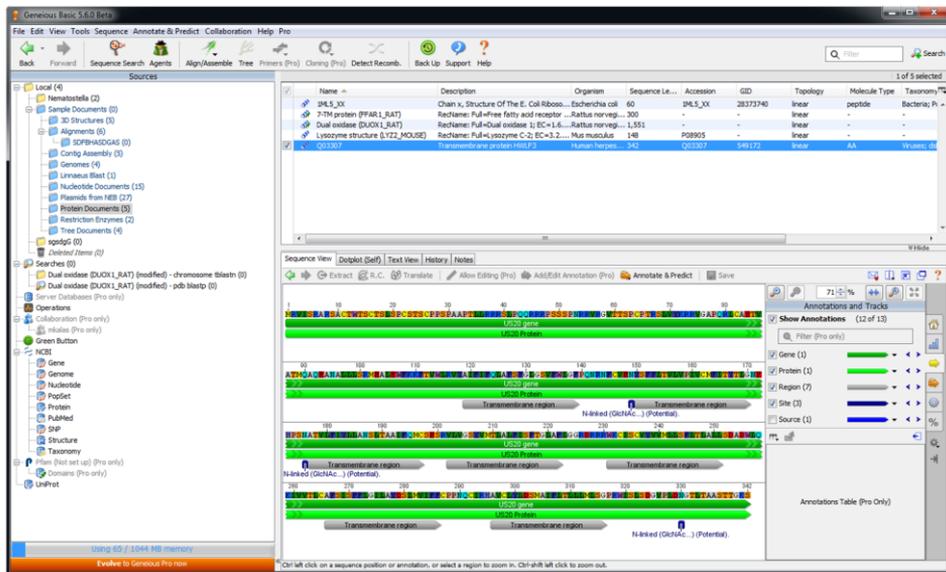
Fig. 5. A screenshot of the graphical user interface of the Geneious Basic workbench. Geneious Basic (Kearse *et al.* 2012) offered a good selection of data-retrieval, computational, and visualisation tools, with a playful user interface. Newer versions of Geneious are not available for free anymore, but this older Geneious Basic is still available in Bio-Linux, leading us to the next section.

System distributions

Operating systems – on personal computers nowadays most commonly Windows, Mac OS X, or some kind of Linux (properly GNU/Linux) – are normally distributed and installed together with a set of tools for basic tasks: GUIs, editors, APIs, a web browser, *etc.*. These *system distributions* (*not* "distributed systems" in sense of being de-centralised, but of being distributed as goods for the users) can be installable as a whole from *e.g.* DVDs or downloadable files. Some Linux distributions come already pre-equipped with a selection of well-tested bioinformatics tools. Such "Bio Linuces" make bioinformatics tools **accessible** and **available** for users' personal computers and their institutions' servers, without having to search, choose, and install the tools, manage their dependencies, or sometimes compile them. They are with few exceptions **free** and **open-source**. In addition, "Bio Linuces" can usually be booted up from a so-called *live* CD, DVD, or USB stick, so that users do not have to install them at all if they only need them temporarily, for example within a training workshop or an occasional analysis. "Live" examples include Bioknoppix (not maintained anymore, http://bioknoppix.hpcf.upr.edu), bioSLAX (http://bioslax.com), and especially the comprehensive and well-supported Bio-Linux (http://environmentalomics.org/bio-linux, Field *et al.* 2006) which is based on the **usability**-oriented Ubuntu distribution. A specialised

39

Linux distribution that partially overlaps with bioinformatics is *e.g.* OSDDlinux for chemo-informatics and drug discovery (http://www.osdd.net/news-updates/osddlinux).

Main Linux distributions are equipped with *package management* software which enables users to add new applications or libraries from dedicated *repositories*, without complications with installation, versions, compilation, and especially dependency management, making the system installations **maintainable** without complex administration. A couple of Linux distributions contain large numbers of bioinformatics tools available in their package repositories: Gentoo Linux (http://packages.gentoo.org/category/sci-biology?full_cat), and especially the foundational Debian which many Linux distributions are based on, including Ubuntu. Debian is the well-tested, **reliable**, well-supported, strictly **free** and **transparent** operating system maintained by an organisation of volunteers (Murdock 1994, Perens 1997). Debian users can, however, install non-free packages additionally. Debian contains a broad selection of free bioinformatics and life-scientific tools that are integrated into Debian by the Debian Med initiative (Möller *et al.* 2010, http://www.debian.org/devel/debian-med). Debian Med is, using the Debian terminology, a "Debian Pure Blend": a subset of Debian for a particular target-group of users, with an associated community that develops it and provides user support. Debian Med and Bio-Linux, the two main Linux initiatives for computational biology, evolved into a single integrated community, where the majority of Bio-Linux's "bio" packages is maintained under Debian Med, with a few additional ones that so far are Bio-Linux-only. It may be interesting to mention also Qlustar (http://qlustar.com), an example of a commercial distribution for high-performance computing in "supercomputer" centres. Qlustar is based on Debian and Ubuntu, so Debian Med and Bio-Linux can smoothly be used inside it, and it has an edition with somewhat limited functionality available for free to non-commercial use.

Virtual machines can be used to run one system installation inside another, for example Bio-Linux inside Mac OS X. Virtual machines can also be moved between different physical computers, and can be run in commercial "clouds" if users pay, or in specialised supercomputing centres if users have access to them, paid or free (*e.g.* http://research.csc.fi/computing-infrastructures). Using virtual machines running on remote computational services, one of the phenomena hidden behind the marketing buzzword of "cloud computing", makes high-performance computations **usable** and **accessible** to researchers **flexibly**, without the need for purchasing, installing, and maintaining the necessary hardware. Increasing number of bioinformatics tools are available as fully-installed virtual machines, that users can immediately deploy and start using locally, on a virtualisation-enabled server, or a "cloud" service. Examples include PredictProtein (Kaján *et al.* 2013, http://rostlab.org/services/ppmi), JBrowse (Skinner *et al.* 2009), and Galaxy (Afgan *et al.* 2010). Examples of virtual machines equipped with comprehensive sets of bioinformatics tools are DNALinux (http://dnalinux.com) and CloudBioLinux (http://cloudbiolinux.org, Afgan *et al.* 2012), the latter containing a substantial portion of contemporary bioinformatics tools via integration from various

repositories including Bio-Linux, Bio*, and Bioconductor. With CloudBioLinux, the whole bioinformatics "laboratory" is **available** in "a couple of clicks", on a user's local computer or in an eventual supercomputing facility. A light-weight alternative to virtual machines are *software containers* limited to one family of operating systems, such as the popular Docker for Linux systems (http://docker.com). In addition to installable tools and system distributions, virtual machines and software containers are the only other option for analysing **sensitive data** – provided that the virtual machine is verified safe – inside isolated computing environments (such as TSD at the University of Oslo, http://www.uio.no/tjenester/it/forskning/sensitiv/hjelp/brukermanual).

1.4.   Standardising information and data representation

Bioinformatics and computational biology have data in the centre of gravity: analysing biological data, comparing data, interpreting data, producing data that suggest new relations in nature. When researchers succeed in finding new insights, the excitement is naturally about the content of the data and some nice plots to present the results. Less effort may be put into "non-content" qualities of the data such as format, readability, terminology, consistency, reproducibility, or compatibility and comparability (interoperability) with other data. Similar holds when developing new computational tools or databases: the functionality and the content of the output or stored data are naturally the main focus, while flexibility of inputs and the "non-content" qualities of the output are secondary. However, when the results and tools are later used by other researchers in their analysis workflows, the **accessibility**, **usability** and **reusability**, **interoperability** with other data, and of course **provenance** and **reliability** of the data become of great importance. In order to mitigate the vast creative chaos in bioinformatics data, various types of efforts have been initiated and implemented.

Data formats

We can broadly say that a *data format* is a particular way of structuring information so that computer programs can read and "understand" it; of representing information as data items; and of encoding the data in computer memory or on a data medium. A particular *type of data* – for example a sequence of nucleotides of a gene with basic information about the gene – can be represented in many ways, in various formats. In

```
>sp|P43353|AL3B1_HUMAN Aldehyde dehydrogenase family 3 member B1 OS=Homo sapiens GN=ALDH3B1 PE=1 SV=1
MDPLGDTLRRLREAFHAGRTRPAEFRAAQLQGLGRFLQENKQLLHDALAQDLHKSAFESEVSEVAISQGEVTLALRNLRAWMKDERVPKNLAELGGKNPCYV...


>AL3B1_HUMAN P43353 ALDEHYDE DEHYDROGENASE 3B1 (EC 1.2.1.5). - Homo sapiens (Human).
MDPLGDTLRRLREAFHAGRTRPAEFRAAQLQGLGRFLQENKQLLHDALAQDLHKSAFESEVSEVAISQGEVTLALRNLRAWMKDERVPKNLAELGGKNPCYV...


>gi|4502043|ref|NP_000685.1| aldehyde dehydrogenase family 3 member B1 isoform a [Homo sapiens]
MDPLGDTLRRLREAFHAGRTRPAEFRAAQLQGLGRFLQENKQLLHDALAQDLHKSAFESEVSEVAISQGEVTLALRNLRAWMKDERVPKNLAELGGKNPCYV...


>sp_ac|P43353 \ID= AL3B1_HUMAN \DE="Aldehyde dehydrogenase family 3 member B1 (Aldehyde dehydrogenase 7)" \NCBITAXID=9606
MDPLGDTLRRLREAFHAGRTRPAEFRAAQLQGLGRFLQENKQLLHDALAQDLHKSAFESEVSEVAISQGEVTLALRNLRAWMKDERVPKNLAELGGKNPCYV...
```

Fig. 6. Examples of sequence records in FASTA format. 4 different records of the same sequence in the same format (FASTA), but with differently formatted accompanying information. Highlighted in blue is database, green identifier, red taxon, and violet version.

order to have a set of tools smoothly **interoperable** with each other, minimising the needs for converting formats when they are used together in a workflow, the tools should accept and output a particular type of data in a common format. There are numerous *de-facto* standard formats which are usable with broad spectra of bioinformatics tools, *e.g.* the tab-separated textual GFF (http://gmod.org/wiki/GFF3) and BED (Kent *et al.* 2002) for information about genomes, genes, biopolymers, their parts, and related measured or inferred values. These formats are to some extent readable also to humans, while similarly structured bigBed (Kent *et al.* 2010) and BAM (Li *et al.* 2009) are, however, compressed into binary files or blobs in order to save data volume for transfer and storage.

Specifications of data formats often allow certain freedom of representing some parts of the recorded information. An obvious example among bioinformatics data formats is the FASTA format (http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml) – a widely used textual format for genetic and biopolymer sequences – which leaves the structuring of accompanying information open (Fig. 6). Using the same format in different ways among various tools may hamper the interoperability, too.

In order to achieve better interoperability with tools, and in some way easier implementation or integration with other data, a machine-understandable specification of a data format can be provided in a schema language. A data schema can also be called a *data model*, and allows a degree of **automation** in processing data instances, such as in parsing, validating, printing, or compressing, by using available programmatic libraries that are not specific to a particular data format. XML formats are usually defined in a dedicated XML Schema (XSD, http://www.w3.org/XML/Schema, http://www.w3.org/2001/XMLSchema). XML formats in bioinformatics are for example MAGE-ML for microarray data (Spellman *et al.* 2002), SBML for models in systems biology (Hucka *et al.* 2003, 2004), CML and PDBML for molecular structure (Murray-Rust *et al.* 2001, Westbrook *et al.* 2005), phyloXML and NeXML for phylogenetic data (Han and Zmasek 2009, Vos *et al.* 2011, 2012), or recently BDML for spatiotemporal dynamics of biological objects (Kyoda *et al.* 2015). In addition to formats specialised on a particular

type of data in a specific sub-domain of bioinformatics, a couple of XML-Schema-based data models were developed for representing the common, basic types of bioinformatics data such as sequences, their annotations, or alignments: *e.g.* the HOBIT XML (Seibel *et al.* 2006) and the CBS Common Data Types (http://www.cbs.dtu.dk/ws/doc/datatypes.php).

Semantic-Web approaches represent data usually in RDF or a related format, in bioinformatics for example within the infrastructures of SADI (an evolution of BioMoby, Wilkinson *et al.* 2011), Open PHACTS (Williams *et al.* 2012), or TogoTable (Kawano *et al.* 2014). The UniProt database is available among other formats in RDF (http://www.uniprot.org/downloads), and RDF is used for the BioPAX format of pathway data (Demir *et al.* 2010), or initiatives supported by DBCLS and its BioHackathons since 2010 (Katayama *et al.* 2013, 2014).

Vocabularies and ontologies

Terminology used inside data formats may vary. Even if different data items are stored in the same format, the terminology used inside the data may be different. And even more problematic may be when the terminology is the same but the authors or tools that produced the data use it differently: with different meaning (semantics). *Controlled vocabularies* and *ontologies* aim at improving the **interoperability** and **comparability** between data, by defining common terminologies usable within and between data formats.

Common Semantic-Web vocabularies of data attributes and objects are used especially in RDF formats – *e.g.* the RDF vocabulary itself and RDFS (http://www.w3.org/TR/rdf-schema) or DOAP (http://github.com/edumbill/doap/wiki) – as data models in a similar fashion to other reusable data models for other formats, including XSDs mentioned above.

Adding a conceptualisation layer to pure terminologies, specialised "domain" ontologies aim at providing standardised concepts for the values of *enumerative* types of data attributes, within given
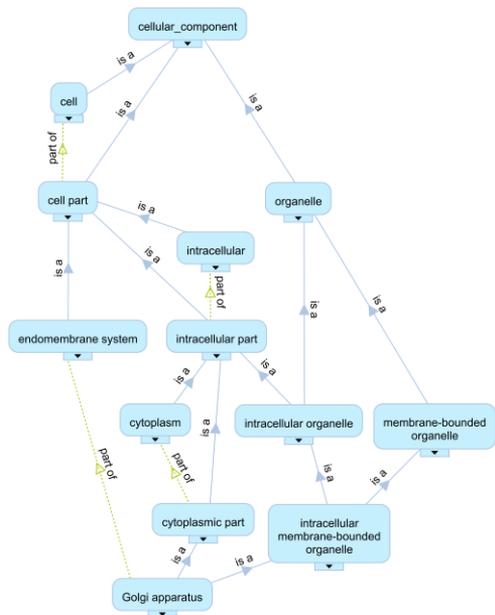

Fig. 7. Excerpt from GO. Shown is a hierarchy of concepts defining the concept of Golgi apparatus (graphics from NCBO BioPortal, Noy *et al.* 2009).

43

domains and aspects of their scope. For example Gene Ontology (GO, Ashburner *et al.* 2000) provides hierarchically organised enumerations of cellular components, molecular functions, and biological processes (Fig. 7, *p.*43). GO is used ubiquitously in biological data in various formats. Sequence Ontology (SO, Eilbeck *et al.* 2005) covers properties and features of nucleotide and amino-acid sequences. SO is mandatory in the current version of the GFF format (http://gmod.org/wiki/GFF3) but usable also in other data formats. Ontologies counteract misunderstanding of their enumerated concepts by rigorous definitions amended with generalisation-specialisation hierarchy (usually forming a directed acyclic graph) and additional relations between the concepts, supplementary data about the concepts, and external links to further information.

Metadata standards and provenance

Metadata are additional data that store some information about the primary data. *Provenance* is information about the origin and history of an artefact: in case of data, provenance metadata store information such as where the data comes from and how it was produced. *Metadata standards* define the required content and qualities of certain type of metadata for a given type of data. More concretely, in life sciences various so-called *minimum information standards* (also checklists or minimum information requirements) are defined for various types of biological data in order to allow **comparability** between data from various experiments, analyses, or conditions, and increase **transparency** and **reliability** of the data. These standards define required provenance including detailed information about the biological and technological conditions the data was produced in – from biological properties of the samples, via the sampling details and sample processing, to post-processing and handling of the measured data. Examples can be the Minimum Information About a Microarray Experiment (MIAME, Brazma *et al.* 2001) for gene-expression data, and the Minimum Information about a Genomic, Metagenomic, or MARKer-gene Sequence (MIGS/MIMS/MIMARKS, Field *et al.* 2008, Yilmaz *et al.* 2011) for sequencing, unified as MIxS, the Minimum Information about any (x) Sequence (see also Table 2, *p.*45). The numerous minimum information standards for biological data have been gathered into MIBBI (Minimum Information for Biological and Biomedical Investigations, Taylor *et al.* 2008), together with a couple of additional information standards such as BioDBCore defining required information about bioinformatics databases (Gaudet *et al.* 2011). Related initiatives have emerged, for example towards describing and documenting sample-processing protocols in molecular biology (Klingström *et al.* 2013).

Scope	Metadata standard	Data formats	Ontologies for enumerations	Supporting consortium
Gene expression measured with microarrays	MIAME (Brazma *et al.* 2001)	MAGE-ML (Spellman *et al.* 2002) MAGE-TAB (Rayner *et al.* 2006)	MGED Ontology (Whetzel *et al.* 2006)	MGED/FGED (http://www.mged.org http://fged.org)
Molecular interactions	MIMIx (Orchard *et al.* 2007)	PSI-MI XML (Hermjakob *et al.* 2004), MITAB (Kerrien *et al.* 2007)	MI (*ibid.*)	HUPO PSI (Martens *et al.* 2007)
Genome, metagenome, and marker-genes sequencing	MIGS, MIMS, MIMARKS (MIxS, Field *et al.* 2008, Yilmaz *et al.* 2011)	GCDML (Kottmann *et al.* 2008)	multiple	GSC (http://gensc.org)
Nucleotide and amino-acid sequence features		GFF3 (http://gmod.org/wiki/ GFF3	SO (Eilbeck *et al.* 2005)	GMOD (http://gmod.org)
Phylogenetics and comparative biology		NeXML (Vos *et al.* 2011, 2012)	CDAO (Prosdocimi *et al.* 2009)	EvoInfo, NESCent (http://zenodo.org/rec ord/19000)

Table 2. Example metadata standards, data formats, ontologies, and supporting consortia. In some cases, there are correspondences between metadata/information standards, data formats, and ontologies. Table shows examples of information standards, data formats that include the corresponding metadata, ontologies for enumerative values inside the (meta-)data, and organisations or consortia supporting the development, adoption, and maintenance of these corresponding standards.

Multiple efforts have been made towards reproducibility of computational analyses of scientific data. Commonly among these, some tooling records provenance metadata which can be used for re-running the analyses, such as in "executable papers" (*e.g.* Schwab *et al.* 2000), ISA tools (Rocca-Serra *et al.* 2010), and in multiple workbenches: *e.g.* GenePattern with a dedicated plugin to Microsoft Word for reproducibility of its workflows (Mesirov 2010), or Galaxy with its "histories" and webpages documenting results (Goecks *et al.* 2010). Some bioinformatics tools record **provenance** – information about the tool and the used parameters – conveniently as part of their output. Unfortunately, such habit is limited, not standardised, and usually not in an interoperable, machine-understandable format. As a new hope, a model for provenance metadata has recently been standardised by W3C as PROV (http://www.w3.org/TR/prov-overview).

1.5. Sharing experience and effort

In an over-simplified, "marketing" style, we could conclude about the types of efforts presented in the previous section, that formats serve **interoperable tools**, thus helping tools integration; ontologies serve **interoperable data**, helping data integration; and metadata standards serve **interoperable research**, helping integrate research results. The *Background* chapter of this thesis should not end without mentioning maybe the most important tier: the initiatives that help "integrating people", by sharing experiences and efforts between researchers.

In addition to bioinformatics *consulting* businesses, certain countries established national bioinformatics *help desk* networks, such as the former BioAssist of NBIC in the Netherlands (http://wiki.nbic.nl/index.php/BioAssist_Main_Page), or the former FUGE Bioinformatics Platform in Norway (http://www.forskningsradet.no/prognett-fuge/Bioinformatics/1234130619850), now continuing within the Norwegian Bioinformatics Platform (http://www.bioinfo.no/help-desk, Nygård and Jonassen 2014), assisting public and private research with computational biology, for free or paid.

Other than abundant mailing lists and group, online community websites are important sources of information about bioinformatics tools, resources, and methods. After the historical BIOSCI a.k.a. Bionet (listed in Gilbert 2004), BioMedNet, and others, current *fora* and *wikis* provide means of sharing documentation, hints, comparisons, reviews, discussions, or questions & answers about bioinformatics tools or methods: *e.g.* BioStar, serving as a crowd-sourced help desk (Parnell *et al.* 2011), and SEQanswers (Li *et al.* 2012b) together with its SEQwiki (Li *et al.* 2012a), specialised towards analysis of sequencing data. Among numerous bioinformatics wikis, some projects are connected directly to Wikipedia (how-to in Logan *et al.* 2010), *e.g.* RFAM (http://en.wikipedia.org/wiki/Wikipedia:WikiProject_RNA, Daub *et al.* (2008), experience in Gardner *et al.* 2011, Finn *et al.* 2012), or http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Computational_Biology with http://topicpages.ploscompbiol.org/wiki/Topic_Pages (Wodak *et al.* 2012), following the molecular- and cellular-biology efforts such as http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Molecular_and_Cellular_Biology and http://en.wikipedia.org/wiki/Portal:Molecular_and_cellular_biology.

Last but not least, global *networks* of researchers – organisations and consortia – help developing new ideas and projects, **long-term maintaining** the existing resources, sharing effort and reaching out to the community. In addition to the previously mentioned EMBnet (Doelz 1992, Harper 1996, D'Elia *et al.* 2009) and O|B|F (http://open-bio.org), another influential has been the International Society for Computational Biology (ISCB, http://www.iscb.org), the examples listed in Table 2 (*p*.45), or bioinformatics.org hosting collaborative community initiatives, with related fora and websites.

2 Summary of results

The *Summary of results* chapter summarises the achievements of the three projects from the enclosed Articles I – III, with additional retrospective background information, and their contributions to the bigger picture presented in the previous chapter *Background*. Present and future perspectives of these projects are discussed in the next chapter, *Discussion*.

2.1. BioXSD – a data model for basic bioinformatics data

In Article I (Kalaš *et al.* 2010) we present BioXSD, a proposed common exchange format for basic bioinformatics data. The BioXSD effort has been initiated within the EMBRACE project (2005-2010, http://www.embracegrid.info, Pettifer *et al.* 2010) and within the first DBCLS BioHackathon, in 2008 (Katayama *et al.* 2010), both of which identified the need for a common exchange format for bioinformatics Web services. The common data format had to be defined as a machine-understandable data model, using the standard XML Schema language (XSD, abbreviated from "XML Schema Definition", http://www.w3.org/XML/Schema). XSD-based formats are in particular useful with Web services and in object-oriented programming, and the BioXSD project took up the challenge of defining the common exchange format that is particularly suitable for, but not limited to these two usage scenarios. The BioXSD project started as a collaboration between institutes participating in EMBRACE, first CBS at DTU in Denmark with CBU at UiB in Norway, soon joined by IBCP in Lyon, France, and subsequent individual collaborators from the EBI in UK and other places.

In the BioXSD project, we defined XSD-based formats for those commonly used types of data within bioinformatics that do not have another widely-accepted XML format. The scope of BioXSD narrowed down to "sequence-centric" types of data – biomolecular sequences and sequence records, alignments, and feature records – accompanied by auxiliary types of data such as ones necessary for encoding external references (links to databases, tools, and ontologies). Dedicated, specialised XSD-based

formats existed for other types of data, for example systems biology models (SBML, Hucka *et al.* 2003, 2004), phylogenetic data (phyloXML, Han and Zmasek 2009, and NeXML, Vos *et al.* 2011, 2012), microarray data (MAGE-ML, Spellman *et al.* 2002), or genome sequence metadata (GCDML, Kottmann *et al.* 2008). Predecessors of BioXSD – with similar scope – were primarily the HOBIT XML for sequences, alignments, and RNA structure (Seibel *et al.* 2006), which did not achieve broader acceptance beyond the HOBIT network; and the *"common data types"* at CBS (for sequences and features, http://www.cbs.dtu.dk/ws/doc/datatypes.php) which served as a starting point for the work on BioXSD. DAS XML for feature records (Dowell *et al.* 2001, Prlić *et al.* 2007) mimicking the tabular GFF format, and the BioMoby XML formats (Wilkinson and Links 2002, Wilkinson *et al.* 2008) were not defined as XML Schemata and thus not usable in the standard way with Web services or object-oriented programming languages (with ordinary XML data-binding libraries). In addition, none of the four mentioned preceding formats were further developed and maintained at the time. For a reader who missed it, the landscape of data formats in bioinformatics is more broadly described in 1.4 Standardising information and data representation (*p.*41). BioXSD has been developed by analysing diverse requirements, tools, Web services, data formats, and ontologies.

In BioXSD version 1.0.0 – presented in Article I – the format of a ***sequence record*** includes the biomolecular sequence as a string, and the optional metadata that are supposed to identify the sequence (such as data resource and organism it originates from, accession, name, version, *etc.*), but does not allow to represent features of the molecule or its part. Only data needed for translation of the sequence between nucleotides and amino-acids can be included. ***Feature record*** (called *sequence annotation* in BioXSD 1.0), on the other hand, allows representations of any features and measured or inferred values related to biopolymers or genomes, for example transcription factor binding sites, gene expression data, secondary structure of RNA and proteins, active sites, variation, or "pairwise" alignments of other sequences to the "reference" sequence. Feature data may include references to any shared nomenclature, ontologies, data resources, and publications. It may also include **provenance** metadata (with details about processing performed with the data – computational or "manual" – but *not* about the underlying biological samples as in *minimum information standards*), and **attribution** metadata (*e.g.* links to publications that should be cited). The BioXSD feature record enables **integration** of diverse sequence- and genome-related data and metadata into an integrated representation, in a structure that can be automatically parsed by ordinary XML data-binding libraries (Fig. 8). A standalone ***alignment record*** in BioXSD represents a multiple sequence alignment, *i.e.* an alignment without target and reference sets of sequences, but all sequences treated equally. Likewise a feature record, a BioXSD alignment can include provenance and attribution metadata. Other types of data modelled in BioXSD are standalone **references** to data resources and entries, ontologies and nomenclature, taxonomies, or to computational tools. BioXSD also contains constrained simple types

48
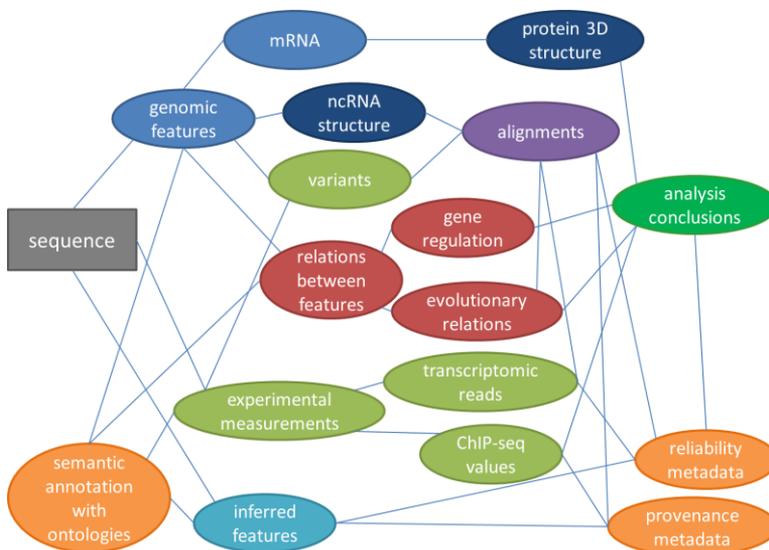
Fig. 8. BioXSD feature record. Example features illustrate the diversity of features, their relations. and metadata that can be recorded in an integrated BioXSD feature record. BioXSD defines also complementary, simple data structures for exchange of sequences (for figures see Article I) and multiple-sequence alignments.

(xs:simpleType), encouraging their use for compatibility and for input validation: the pure biomolecular sequence strings, accessions and other identifiers, resource names, subsets of real and integer numbers, safe global URIs, and more.

Definitions of BioXSD data objects and their parts are semantically annotated within the BioXSD XML Schema using SAWSDL *model references* (Kopecky *et al.* 2007, http://www.w3.org/2002/ws/sawsdl), pointing to concepts from the EDAM ontology (Article II). These annotations assign BioXSD objects a globally human- and machine-understandable semantics. BioXSD data objects can thus serve as ready-made, syntactically and semantically **interoperable** building blocks for tool interfaces. The BioXSD data model is rich enough (not in economic terms) to enable loss-less capture of diverse data that would otherwise require use of multiple different formats, and often even the introduction of new formats for untypical features, classifications, or measured values. In BioXSD, an innovatively broad range of experimental data, annotations, and alignments can be recorded in an **integrated** chunk of data, together with provenance metadata, documentation, and semantic annotation with concepts from ontologies of user's choice, improving both **interoperability** and **reliability** of the data and the tools that use it. Tools can produce and consume BioXSD directly, or BioXSD can be used as an intermediate canonical format, rich enough to enable conversions among diverse formats.

Within the early development of BioXSD, we successfully tested its compatibility with a selection of programming languages and XML data-binding libraries, while the

compatibility with the Web Service Interoperability *basic profiles* (WS-I, http://ws-i.org) has strictly been maintained throughout the project. We adapted a number of Web services to use BioXSD as their input and output format, and tested the convenience of programming an automated analytical workflow that uses different BioXSD-compatible Web services. The use of common format decreased the effort of workflow programming considerably. The adaptation to BioXSD was much less demanding in a case where the tool or Web service already used an XML format, compared to a case of changing from a plain-textual output to XML. The BioXSD project and its future directions are discussed in 3.1 Presence and future of BioXSD (*p.*57). The BioXSD web page (http://bioxsd.org) contains technical documentation, examples, news, and additional information.
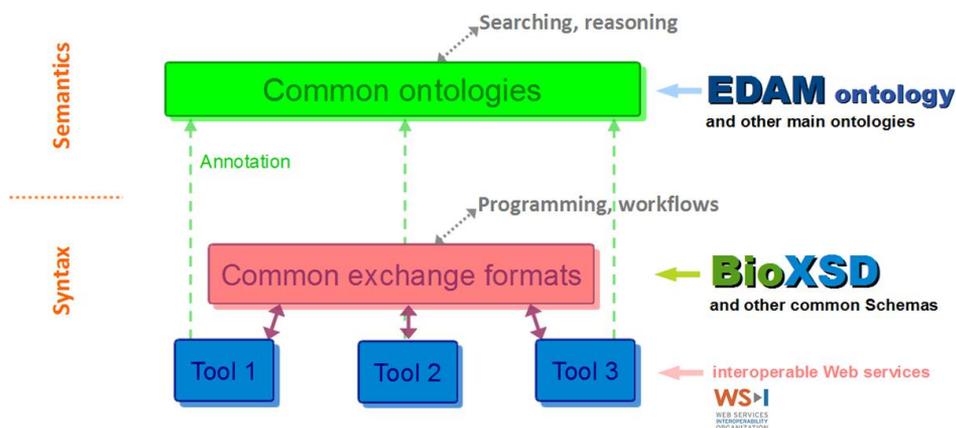

Fig. 9. The portfolio of interoperability standards proposed by the EMBRACE project. With interoperable Web services for tools, common formats (including BioXSD) for their inputs and outputs, and semantic annotation with common ontologies (including EDAM). Agreeing on common approaches along these lines enabled for example the development of integrated systems such as the Utopia Documents (Attwood *et al.* 2010) and eSysbio (Article III).

2.2. EDAM – the ontology of bioinformatics data and methods

Article II (Ison *et al.* 2013) describes the EDAM ontology, at the time in version 1.2. As with BioXSD, the work on EDAM was initiated by the EMBRACE project (2005-2010, http://www.embracegrid.info, Pettifer *et al.* 2010, Fig. 9), and the name *EDAM* was originally

an acronym standing for the *EMBRACE Data And Methods* ontology. The development of EDAM was started by the Peter Rice's group at the EBI with Jon Ison as the main developer, with substantial and regular advice from participants in EMBRACE, from which I soon became the second core developer. It should be emphasised that EDAM and BioXSD have been complementary projects, not based or fundamentally dependent on each other.

EDAM defines concepts, their hierarchy, and some simple relations between them. Defined concepts include operations and types of data used within bioinformatics, complemented with common topics related to bioinformatics, bioinformatics-specific data formats, and relations between the concepts. The sub-ontologies and types of relations comprised in EDAM are presented in Article II and sketched in Fig. 10. EDAM is comprehensive but does not aim at being exhaustive in every detail. Concepts in EDAM have not been as comprehensively covered by any previous related efforts. The most closely related effort was the myGrid ontology (Wolstencroft *et al.* 2007) which served as a starting point for the development of EDAM.
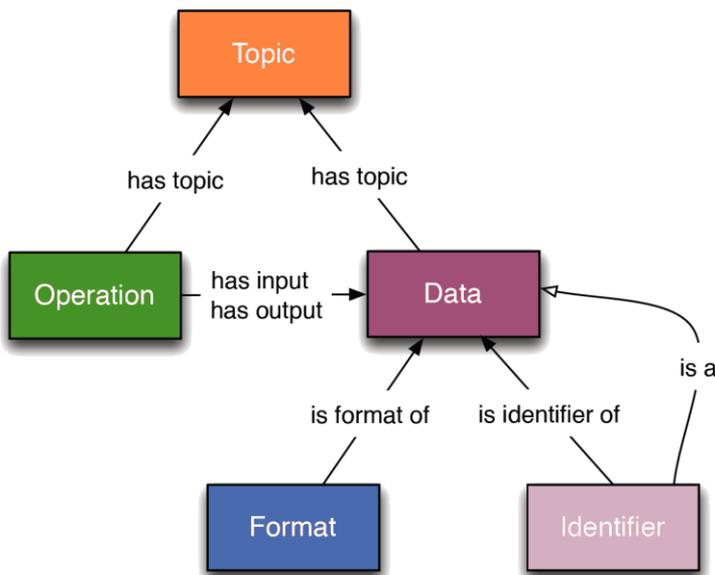

Fig. 10. The sub-ontologies and relations in EDAM.

EDAM has been developed with the primary goal of allowing the **categorisation** and **search** of bioinformatics tools and other resources using globally defined and commonly understood concepts. Such "semantic" navigation in the ocean of available tools and data resources has been strongly desired, applicable to improving information, searching, and filtering in *e.g.* registries, catalogues, toolkits, workbenches, or system distributions. At the time of publication, a number of tools

including EMBOSS (Rice 1998, Rice *et al.* 2000) and several Web services from different providers were annotated with EDAM, using the SAWSDL standard (Kopecky *et al.* 2007, http://www.w3.org/2002/ws/sawsdl) and having the annotations provided and maintained by the service providers, independently of catalogues and context. In DRCAT, our complementary development to EDAM, over 600 data resources were annotated with EDAM, using DRCAT's own format (http://drcat.sourceforge.net). Tools listed in the software catalogue at SEQwiki (Li *et al.* 2012a) were automatically linked to EDAM concepts, too. EDAM-enabled navigation and automated handling of annotated data, tools, and workflows can be useful in integrative workbenches, as was prototyped in eSysbio (Article III). EMBOSS was amended with an implementation of semantic search within its tools and within DRCAT.

Another application of EDAM has been helping the **interoperability** between different tools and formats, and eventually aiding automated format converters. BioXSD data-objects and data-parts definitions were assigned globally machine-understandable meanings via semantic annotations with EDAM concepts, using SAWSDL. EDAM enabled decision support and partial automation within workflow construction. Bio-jETI (Lamprecht *et al.* 2011) has been able to automatically generate workflows from EMBOSS and other tools, given a simple specification using EDAM concepts. In cases where it was possible to specify a desired task easily, a number of workflows was suggested by the built-in machine reasoner. The concept of automated construction of workflows demonstrated by Bio-jETI can be useful for generating small workflows or parts of bigger workflows which do rather mechanistic tasks, such as the mentioned format conversions. EDAM can serve as a **Semantic-Web vocabulary** for bioinformatics data, what was enabled and tested at the 4th DBCLS BioHackathon, in 2011 (Katayama *et al.* 2014). An example from prototyping EDAM-based conversion between BioXSD and RDF is at https://github.com/dbcls/bh11/wiki/BioXSD-sequence-record-in-RDF.

EDAM annotations of concrete pieces of data can contribute to **data provenance** by denoting how the data was processed. This is supported for example in BioXSD-formatted feature records and alignments. Last but not least, EDAM terms and synonyms can be used within **text mining** from documentation and literature on bioinformatics procedures, tools, or resources.

In addition to good coverage and relevance for applications, EDAM has been tuned for convenient **usability** by human annotators and tool users, for efficient **maintainability** by its developers, and for eventual **inter-operation** among diverse ontologies. Feedback with respect to all these goals was obtained from ongoing annotation efforts at the time, implementations, the development of EDAM itself, and a parallel development of a light-weight Web Service Interaction Ontology (WSIO, http://wsio.org). These experiences have been "feeding back" the iterative development of EDAM.

EDAM has *never* aimed at defining a data model, a data representation format, or an information standard. As a related example, the BioDBCore project (Gaudet *et al.* 2011) has one of its aims to define an information standard for information about life-scientific databases. We believe that such division of responsibilities, independence, and the clear and narrow focus, make the development of standards more efficient, the standards simpler, and more efficiently maintainable. EDAM is further discussed in 3.2 Presence and future of EDAM (*p.*59), including the developments after the publication of Article II. http://edamontology.org contains documentation, and serves as the base for EDAM's dereferenceable URIs.

2.3. eSysbio – a workbench prototype for accessible globally-distributed bioinformatics

The eSysbio research project aimed at developing a workbench that integrates tools, data, and people. The following high-level requirements have been targeted (unimplemented ones are marked in grey):

- **Flexibility** and scalability, enabling any sorts of low- or high-throughput analyses within bioinformatics and systems biology
- **Accessibility** and convenient **usability** also by non-programmers among life scientists and the general public. Allowing users to add the tools they need "on the fly", without programming or administration, and assisting them in choosing tools and parameters fitting the needs of their analysis
- Enabling efficient **collaboration** by sharing resources (mostly data, tools, scripts, workflows) between users around the globe. At the same time keeping **privacy** and security of sensitive, proprietary, or unpublished data, tools, or workflows
- Utilising and promoting **community participation** in providing and maintaining computational tools, scripts, workflows, and computational resources distributed around the world. Monitoring the availability of resources, their usage, quality, and evolution, and facilitating **attribution** to their authors and providers
- Interoperability and maintainability, utilising industrial standards governed by the World Wide Web Consortium and OASIS (http://oasis-open.org)
- Tracking detailed **data provenance** in order to enhance **transparency**, **reproducibility**, reliability, and accessibility of scientific results

The eSysbio project has been initiated and pursued at CBU in Bergen between 2007

and 2013, in collaboration with the Bergen Center for Computational Science (renamed to Uni Computing). Article III presents the design of a workbench system satisfying the above goals, and its publicly available prototype implementation.

While several integrated workbenches have been developed and successfully used in bioinformatics especially during the last decade, none of them provided a full combination of the desired features. From the most closely studied examples during the eSysbio project, Galaxy (Giardine *et al.* 2005, Goecks *et al.* 2010) and GenePattern (Reich *et al.* 2006) have been provided as publicly available web servers where all users could share data with each other, but could not add their custom tools. These workbenches could be installed in users' own facilities or in the Amazon Cloud, but then hampering the sharing of data with users of other instances of the particular workbench. In addition, supporting computational tools and resources distributed around the world has not been "natively" straightforward in either type of installation. Distributed computing has been more straightforward and "native" within Taverna (Oinn *et al.* 2004, Hull *et al.* 2006, Wolstencroft *et al.* 2013) or Mobyle (Néron *et al.* 2009), but similarly to Chipster (Kallio *et al.* 2011), these did not provide the functionality of sharing data. As the last but not least example, Geneious (http://www.geneious.com, Kearse *et al.* 2012) demands custom tools to be wrapped as Geneious-specific plugins programmed in Java, what cannot be done quickly or by non-programmers. Bioinformatics workbenches and other approaches to integrating bioinformatics tools are discussed in section 1.3 Efforts in mitigating the chaos (*p.*15).

The design of eSysbio consists of a set of publicly accessible web servers that host a number of relatively loosely coupled modules (unimplemented in grey):

- A user-friendly web application for accessing the system (with Java backend)
- A storage system for users' data, extendable with federated private storages
- A directory of computational tools available in eSysbio. These tools can be hosted anywhere around the globe (as Web services), and can be available publicly or only for chosen users
- Directories of public and private workflows (in BPEL language, http://oasis-open.org/committees/wsbpel) and scripts (in R language). Possibly other workflow and scripting languages later
- A directory of available interactive visualisation tools and editors for particular types of data
- A module that "understands" common ontologies, and enables decision support and semantic search among all included resources
- Engines for executing the workflows and scripts, and for invoking the Web services, extendable with private engines
- Web-service interfaces to all the backend modules of eSysbio, enabling programmatic access and integration into other frameworks

Although not all of these modules and features have been implemented in the

prototype system, the implementation of eSysbio provides a substantial part of the desired functionality. Available at http://esysbio.org, anybody can register as a user. Users can upload data (including large data sets, Fig. 11), add tools that are available as Web services, and add R scripts for *e.g.* data conversions and plotting (Bioconductor (Gentleman *et al.* 2004) is available). Users can organise all these resources into custom collections, share them with other users or user groups, and submit them for curation to enable public access. Analyses can be performed by executing the available – public or private – tools and scripts with uploaded or directly inserted inputs. Graphical user interfaces to tools are automatically generated from their WSDL files and XSDs (http://www.w3.org/TR/wsdl, http://www.w3.org/XML/Schema, relatively simple styles are supported), with help of the Web Service Interaction Ontology (WSIO, http://wsio.org). Results can be saved into the system, including their provenance details. Users can also execute a couple of predefined automated workflows. A subset of major EDAM concepts can be used for navigation among users' data items (Fig. 11).
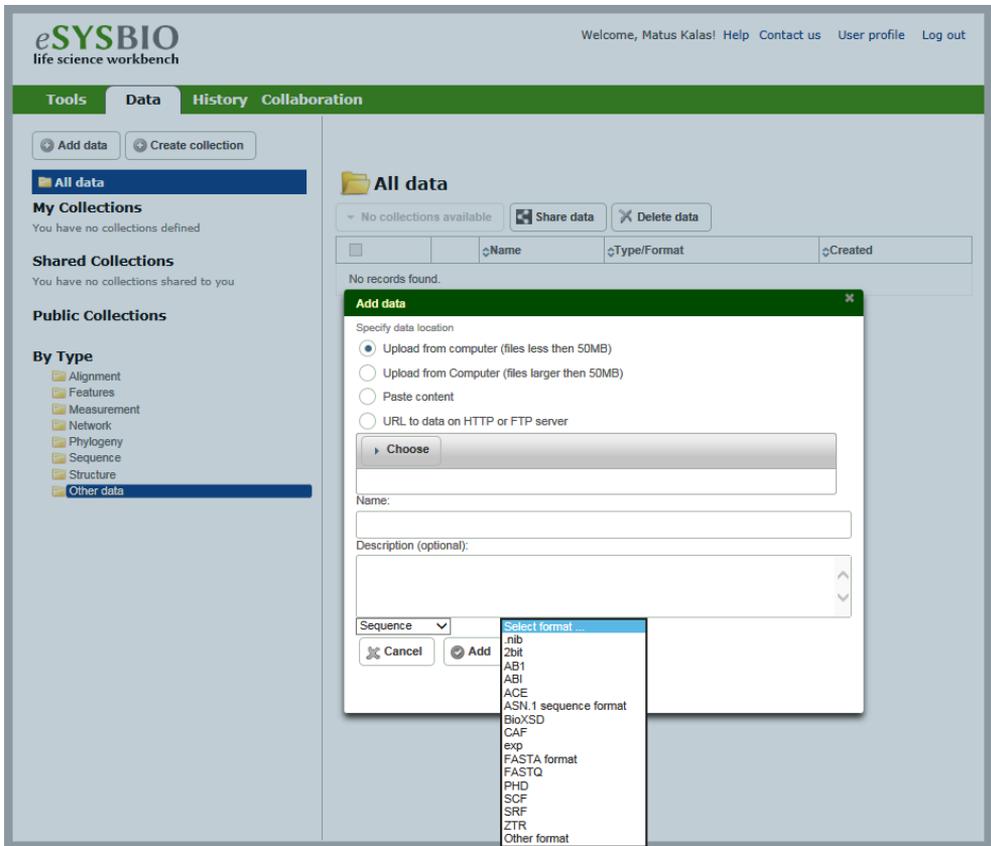

Fig. 11. A screenshot of the eSysbio prototype workbench. This screenshot gives a feeling about the provided data management functionality, and the look-and-feel of eSysbio. The types and formats of data are defined by a subset of EDAM concepts.

Article III presents also proof-of-concept examples of using the prototype workbench for high-throughput scientific analyses. Examples include finding genetic variation and individual variants from human exome sequencing reads, estimating gene expression from RNA sequencing reads, and analysing a metabolic network under various gene-expression conditions, using systems biology techniques. The examples integrate genomics and systems biology, what has also belonged to the goals of the eSysbio project.

The eSysbio design and implementation could be used also outside of life sciences, in form of separate installations with different dedicated directories of resources and different groups of curators and administrators. eSysbio is licensed as open-source, but the source code and binaries have not been published yet. The eSysbio system and its heritage within ongoing projects, are discussed in 3.3 Heritage of eSysbio (*p.*61).

3 Discussion

In the *Discussion* chapter, I discuss some weaknesses of the works presented in the enclosed Articles I - III, together with opportunities for improvements. The main new developments made after the publications are described. I conclude the presented topics with proposed future directions. These are proposed in relation to the state-of-art of the broad field of efforts towards more reliable and accessible bioinformatics (presented in the *Background*), and thus towards more efficient computational biological research.

3.1. Presence and future of BioXSD

In Article I we presented the work on BioXSD version 1.0, with its contributions summarised in 2.1 BioXSD – a data model for basic bioinformatics data (*p.*47). A natural question is why we need yet another data format for the common bioinformatics data such as sequences, alignments, and features. Indeed, perhaps too many formats exist already, but that may be a good reason both *against* and *for* another one. The combination of goals of BioXSD has been unique in its scope: to define a tree-based format (which at the time meant XML) for the common "sequence-centred" data, and **unify** the existing formats by being generic and rich enough in expressiveness to enable loss-less conversions. In particular, BioXSD has aimed at **minimising** the need for developing new tool- or resource-specific formats, for the given types of data, in cases where existing formats were not sufficient.

A number of developments has been planned since the early days of the project, that were not delivered at the time of the publication, many of which are still on the todo list today. Together with requirements identified soon after the publication, these include:

- Support for BioXSD among the O|B|F Bio* libraries and the EMBOSS toolkit
- A broad range of ready-made converters between BioXSD and other bioinformatics data formats, and RDF

- Websites supporting community participation and contribution
- Comprehensive and accessible online tutorials
- Binary BioXSD for large data
- Visualisation tools for BioXSD data
- Automated integration of information about databases and identifiers from EDAM and DRCAT
- Larger-scale adoption of BioXSD among bioinformatics Web services and other tools and data resources

Although BioXSD has been developed to be generic and convenient for representing a broad spectrum of data, updates enabling representation of emerging types of data, and optimisations of both the "expressive power" of the data model, and the size of the data, have been carried out continuously, and shall be continued. Major new version 1.1 of BioXSD was published together with **__GTrack__**, with optimisations described in detail in Gundersen *et al.* (2011). GTrack (http://gtrack.no) is a generic tab-separated format for sequence and genome features, smoothly **unifying** formats such as GFF, BED, or WIG, with certain *forward compatibility*. GTrack and BioXSD 1.1 include similar optimisations of data size, developed in a fruitful shared effort, and emphasising the usefulness of a reasonable **plurality** of well-developed generic formats – or standards in general – each beneficial for a different set of users and usage scenarios.

Further developments improved the semantic annotation of the BioXSD schema, complementing EDAM with foundational Semantic-Web vocabularies – RDFS (http://www.w3.org/TR/rdf-schema) and Dublin Core (http://dublincore.org) – in order to enable automated conversion between BioXSD and RDF. I worked on designing a generic converter between XSD-based data and RDF during the 4th DBCLS BioHackathon, in 2011 (Katayama *et al.* 2014, https://github.com/dbcls/bh11/wiki/BioXSD-sequence-record-in-RDF), but a proper implementation is still pending. We started exploring ways towards interoperability with PROV for provenance metadata (http://www.w3.org/TR/prov-overview), so far without a conclusive proposal.

At last, BioXSD became more **transparent** and more convenient for community **participation** after recently adopting a permissive CC BY-SA license (http://creativecommons.org/licenses/by-sa/4.0), moving guidelines for interoperability-preserving developments from license into a Code of Conduct (declared in http://bioxsd.org/BioXSD-1.1.xsd), and migrating the development of BioXSD to GitHub (http://github.com/bioxsd/bioxsd). GitHub adds the long-desired support for community contributions, complemented with additional channels for discussing BioXSD transparently: so far Google Groups and Twitter (http://bioxsd.org/#Contact, poster Kalaš *et al.* 2015).
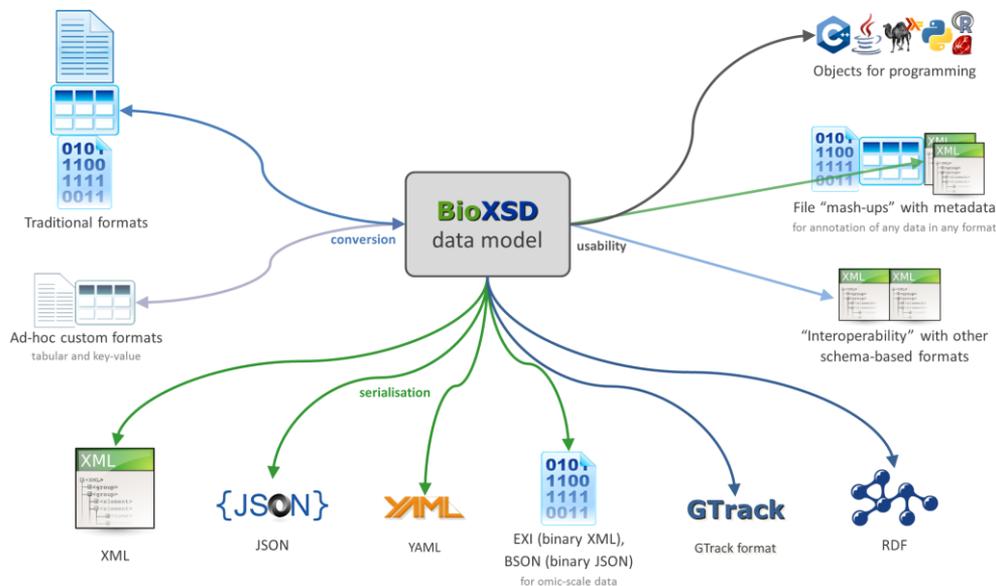
Fig. 12. The vision of BioXSD, with a broad portfolio of interfaces. The ongoing developments aim at enabling BioXSD as a single *data model* with various serialisations *i.e. exchange formats.*

More recent priorities for BioXSD reflect the gained popularity of newer tree-based formats, JSON and YAML (http://json.org, http://yaml.org), together with a need for convenient libraries dedicated for handling BioXSD data in multiple programming languages, including C++ and R (posters Kalaš *et al.* 2014, 2015, Fig. 12). BioXSD development and adoption were not very active during the couple of previous years, with limited personnel bandwidth due to some valuable contributors left, and other projects prioritised. However, the BioXSD project gained momentum again, with prospective new contributors, and the proposed developments on the priority list (more at http://bioxsd.org/#Ongoing). We are determined to keep BioXSD evolving, with needed participation of a broader community. Other than participation, the evolution towards a good standard format demands both effort and patience.

3.2. Presence and future of EDAM

Various topics can be disputed about the EDAM ontology which is presented in Article II, and its promised contributions are summarised in 2.2 EDAM – the ontology of bioinformatics data and methods (*p.*50). "Theoretical" questions may enquire the logical,

59

ontological, and linguistic appropriateness of EDAM's concept definitions, terms, synonyms, or relations. On the other **practical** side, in contrast, we may ask for example: What is a reasonable size of an ontology to be practically usable and maintainable, while being comprehensive and detailed enough? How do we manage maintenance of an ontology so that is kept up to date, when it models such a dynamic domain as bioinformatics? How should additional information within EDAM's concepts, and links to other resources be defined so that they enable various kinds of useful automations to be implemented? Just like all other ontologies, EDAM certainly has opportunities for improvements in all these topics. Being aware of these questions, the increasing experience from projects that adopt EDAM enables incremental improvements which are being incorporated into the further development of EDAM.

After the publication of Article II, the use of EDAM has been implemented – or is under implementation – in additional ongoing projects. The ELIXIR **Tools and Data Services Registry** (http://bio.tools) aims at registering all types of tools and data resources in bioinformatics, maintaining rich information about each registered tool, curated by the community. The Tools Registry uses EDAM for annotating all aspects of tools that EDAM covers: topics, operations, types of data, and data formats. Information recorded in the registry should be rich enough to enable for example automated integration of new tools into workbenches (Ménager *et al.* 2015c).

Several implementations that adopted EDAM were developed at the Pasteur Institute in Paris: from the earlier ones *e.g.* Bioweb (http://bioweb.pasteur.fr), and from the recent *e.g.* ReGaTE (http://regate.readthedocs.org). Although not yet publicly visible, implementations of EDAM are ongoing (slides Ménager *et al.* 2015b), for example in relation to Galaxy, within CCP4 and the INSTRUCT project (http://www.ccp4.ac.uk, http://structuralbiology.eu), at the EBI in UK, CBS in Denmark, and SIB in Switzerland, or within the work on the Common Workflow Language (CWL, http://common-workflow-language.github.io, slides Amstutz *et al.* 2015).

The development of EDAM was made more **transparent** and **participatory** after migrating to GitHub (http://github/edamontology/edamontology), with a number of new versions released since the publication of Article II. Improvements are for example a refactored *Topic* sub-ontology, and a well-defined organisation of responsibilities for the development of EDAM (poster Ménager *et al.* 2015a). Curation hackathons with experts in a particular sub-domain of computational biology are scheduled, which will enable maintaining EDAM up-to-date. In addition to implementations adopting EDAM, the high-priority pending developments include *e.g.* tooling for tailored validation and formatting of the EDAM source file, and for conversion between OWL and OBO format, which will improve the efficiency of maintaining EDAM and enable better participation from the community.

3.3. Heritage of eSysbio

The eSysbio project resulted in a prototype workbench system described in Article III, and further discussed in 2.3 eSysbio – a workbench prototype for accessible globally-distributed bioinformatics (p.53). One of the main novelties in eSysbio is allowing in one workbench both to add tools on-the-fly by all users, as for example in Taverna, and to share data with chosen users around the world, as for example in a public instance of Galaxy. The highlights of eSysbio include:

- Distinguishing public, private, and shared access to tools, scripts, workflows, and data
- Allowing any user to add and share any resource (implemented for data in any format, tools with a SOAP Web service interface, scripts in R, and workflows in the BPEL language)
- Sharing data without copying
- Providing access to resources for single users and for flexible user groups. Groups can correspond *e.g.* to collaborations, institutes, or global groups of users with common interests
- Providing an infrastructure for sharing *ad-hoc* R scripts

On the other hand, the eSysbio project identified additional desired functionality that has, however, not been implemented in the final prototype. For example:

- Integrating visualisation tools (tested only with JalviewLite), and allowing customisation of graphical user interfaces
- Monitoring the usage and evolution of tools, and benchmarking their performance
- Performing sensitivity analysis of single tools and workflows with respect to tool parameters, which could substantially aid transparent and reliable workflow design (has been performed for a single workflow in Sztromwasser *et al.* in preparation, Sztromwasser 2014)
- Automatically attributing all tools, resources, and other intellectual properties used for obtaining a given result, in form of citations or other references

Still publicly unavailable, in addition to the source code and binaries, are also the analysed requirements, design proposals and considerations, implementation proposals, and a published article. Sharing experience from the project is therefore hampered. In contrast to the listed highlights, I personally consider – from a time distance – some of the design decisions as unsuccessful, and some important features as missing (including ones that may apply to most workbenches). In particular:

- Although using standards is in general a good idea for maintainability and interoperability, sticking to only SOAP and BPEL was unlucky. A broader range of tool interfaces needs to be supported in order to be practically usable, not only SOAP Web services. As a workflow language, any popular programming or

scripting language would be both more **accessible** and more **interoperable** than BPEL. The best solution, however, could be the Common Workflow Language (CWL, http://common-workflow-language.github.io, slides Amstutz *et al.* 2015), if it conveniently supports Web services. CWL is being developed in a broad community collaboration, and should be usable as a "clean" workflow specification language that is easy to read, modify, implement, execute, and visualise.

- Missing version control of scripts, applicable types of data, and other resources. Useful with user's own resources, and even more useful with shared resources. Good support and enforcement of versioning and revision graphs would dramatically improve **transparency**, **reliability**, and **maintainability** of users' resources.

- Despite the presented motivation for developing a new workbench, and the practical aspects of working in a day-to-day co-located team, one could still challenge the feasibility of yet another workbench development, and in addition one carried out at a single institute. To help organise the creative chaos, a broader collaboration with a **durable**, patient **community**-based effort is preferable.

- Paradoxically, while workbenches are developed with **accessibility** and often also **transparency** among their aims, one may argue that learning to use the *command shell* is more accessible than developing and maintaining automated workflows and tool wrappers for a particular workbench, in addition to leaving more control in hands of a user and thus enabling more transparency (see *e.g.* Holly Bik at http://eukaryoticebullience.blogspot.com/2015/07/reflections-on-bosc2015-keynote-and.html, Kai Blin at http://phdops.kblin.org/2015-on-overengineering.html). Convenient (thus **accessible**), community-supported (thus **up to date** and **flexible**), free, open-source and widely-used (thus **reliable**) *de-facto* standard systems, suitable for life-scientific data analysis, are for example Debian with its derivatives (Möller *et al.* 2010) for tools, Git (http://git-scm.com) for versioning, and *e.g.* the IPython Notebook (http://ipython.org/notebook, Pérez and Granger 2007) for documenting workflows.

- With or without workbenches, substantially helpful for **transparency** and **reproducibility** is some tooling that *automatically records* workflows ("manual" or automated), with all intermediate scripts or eventual manual editing, and with all tool parameters and versions (for tools outside of Debian, a comprehensive tools registry would help). Such recorded workflows should be documented in both human-friendly and machine-reproducible way, independent of a particular workbench. Exemplary in this direction are workflows in GenePattern (Reich *et al.* 2006), although limited to tools and functionality available in GenePattern. While recording **provenance**, the tooling could at the same time record **attribution** information, ideally with support from the involved tools which would in an ideal world provide both provenance and attribution metadata in some standard format as part of their output.

As outcomes of eSysbio – in addition to the workbench design and development – the eSysbio project and team substantially contributed to a broad spectrum of other developments and research projects, related to eSysbio in various ways. These include the EMBRACE project and the EMBRACE Web service Registry (Pettifer *et al.* 2009a), together with BioXSD and EDAM presented in this thesis. Another development carried out within eSysbio is the lightweight Web Service Interaction Ontology (WSIO, http://wsio.org), which is applicable beyond the scope of eSysbio, and will hopefully have a chance to be developed further. Optimisations of globally-distributed computing with large data were explored and tested in the work of Paweł Sztromwasser and his collaborators within the eSysbio team, proposing the use and some standardisation of specially tuned Web services (Sztromwasser *et al.* 2011, Subramanian *et al.* 2010, 2012, 2013). Another notable technical development was the prototyped automated generation of web-based graphical user interfaces for invoking Web services, using WSDL, XSD, and WSIO, experience which may be revived in the future, despite of complex implementation challenges. Several systems biology and computational biology developments were carried out in collaboration between the eSysbio project and various other groups including "wet" biologists (*e.g.* Stavrum *et al.* 2013).

Experiences from eSysbio and from the FUGE Bioinformatics Platform (http://www.forskningsradet.no/prognett-fuge/Bioinformatics/1234130619850) formed essential contributions to the work within the Norwegian Bioinformatics Platform (http://www.bioinfo.no) which constitutes the Norwegian node of the European ELIXIR project (http://www.elixir-europe.org, developing computational, data, and learning infrastructures for bioinformatics and computational biology). A particular development that builds upon the experience from the eSysbio prototype, is **NELS** (Norwegian e-Infrastructure for Life Sciences, http://nels.bioinfo.no), developed in a national collaboration within the Norwegian Bioinformatics Platform, with Kidane Tekle from CBU as the lead developer. NELS provides a hardware and software infrastructure for storing and sharing research data. Example highlights of its implementation are:

- Based on standards where possible, but a broader and more up-to-date spectrum of them, compared to eSysbio
- Accessible via a diversity of interfaces: a web portal, Web services, standard SSH, and from other applications including Galaxy
- Federated identity management including authentication with FEIDE (http://feide.no/introducing-feide), and other identity providers hopefully in the near future

Again, one can ask why yet another system needs to be developed. A natural comparison of NELS would be for example with GenomeSpace (http://genomespace.org, posters Reich *et al.* 2013, Garamszegi *et al.* 2015). GenomeSpace stores all data on the Amazon cloud, and unfortunately cannot be easily adapted to function with the available storage solutions in Norway. In addition, NELS had to be developed with

some urgency, as it is needed in various research projects running in Norway. Hopefully, the work on NELS will become part of some broader international collaboration in the future. What I also hope will come into focus of the Norwegian Bioinformatics Platform soon, is good support for the **standard**, simple, and **transparent** computing with Debian and its derivatives; for *data versioning* – where applicable – using some standard versioning system such as Git; and for comprehensive human-accessible documentation of analysis workflows, again with some widely-used, *de-facto* standard(s).

The experiences from eSysbio – especially ones related to information about registered tools, their curation, and monitoring – have been channelled into the development of the Tools and Data Services Registry (http://bio.tools), since its initiation. It is great to see that the experiences and consideration from eSysbio are useful in multiple other projects. I hope they will be used more widely, preferably in loosely-coupled, open community collaborations, and that members of the eSysbio team (Fig. 13) including myself will have opportunities to contribute with experiences.

Fig. 13. The eSysbio team (in 2011). Missing on the picture is Kidane, who substantially contributed to finalising the prototype.

3.4.   Additional concluding remarks

As I expounded in this thesis, the creative chaos in bioinformatics is thriving. Highly productive and multi-directional is also the creativity within *organising* the chaos, including the work I contributed to the field. In BioXSD, we have been developing a somewhat universal data model and exchange format, soon to be complemented with software tooling, together potentially improving interoperability of tools and data integration. The EDAM ontology helps navigating in the vast jungle of various resources from tools and formats to training courses, with crowd annotation efforts picking up. In the eSysbio project, we explored and tested a number of novel designs towards accessible, reliable, and efficient computational biology, now using the experience in ongoing projects. I presented my three main projects, and mentioned a few complementary, on the broad and non-exhaustive background of efforts towards more accessible and more reliable bioinformatics. More accessible and reliable means more **efficient** – more efficient with respect to a researcher's effort.

There is, however, another and perhaps even more important kind of efficiency. After focussing on various *quality aspects* of tools for scientific computing throughout the whole thesis, mainly under the umbrella topics of accessibility and reliability, let us at last take a quick look right nearby: at the actual computational **efficiency**. With global computing being responsible for about 2-3% of GHG emissions, similar to civil aviation (Griffiths 2008), a lot may be at stake, especially taking into account the growing presence of life sciences among the biggest data "crunchers" (Marx 2013). All efforts towards improving the computational and data efficiency may thus be relevant: the individual, community, and organisational. I am not going to moralise here about which programming languages, kinds of tools, or precisions one should use, but instead I mention two example directions, mutually on different sides of the spectrum. The "small" direction, applicable to computation on anybody's laptop, is re-implementing popular tools in C++ with help of some highly optimised libraries, as for example in Kaján *et al.* 2014 (where I humbly contributed), speeding up casual tools by one or two *orders of magnitude*. Similarly in Saito and Rehmsmeier (poster 2015), in both of these cases by pure hacking, and both at the same time improving accessibility of the re-implemented tools. On the other side of the spectrum are investments into alternative, fast and energy-efficient specialised hardware, including GPU clusters, and most relevantly FPGAs (Fowers *et al.* 2012). Utilisation of special hardware gives yet more arguments for sharing computational *infrastructure*, for computational and data *services*, and for *community* efforts in developing algorithms, tools, and know-how.

In incremental efforts, an elaborate balance should be sought between the "user-efficiency" and the "energy-efficiency", together with all the other objectives related to ethics and fairness, reliability, maintainability, *etc.* After all, all these aspects matter also economically and competitively: getting your results faster and cheaper could also mean wasting less natural resources. I am eager to continue working in the field.

References

Afgan, E., Baker, D., Coraor, N., Chapman, B., Nekrutenko, A., and Taylor, J. **(2010)**. Galaxy CloudMan: delivering cloud compute clusters. *BMC Bioinformatics*, **11**(Suppl 12), S4. 10.1186/1471-2105-11-S12-S4

Afgan, E., Chapman, B., Jadan, M., Franke, V., and Taylor, J. **(2012)**. *Using Cloud Computing Infrastructure with CloudBioLinux, CloudMan, and Galaxy*. John Wiley & Sons, Inc. 10.1002/0471250953.bi1109s38

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. **(1990)**. Basic local alignment search tool. *J. Mol. Biol.*, **215**(3), 403 – 410. 10.1016/S0022-2836(05)80360-2

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. **(1997)**. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**(17), 3389–3402. 10.1093/nar/25.17.3389

Amstutz, P., Tijanić, N., Soiland-Reyes, S., Kern, J., Stojanovic, L., Pierce, T., Chilton, J., Mikheev, M., Lampa, S., Ménager, H., Frazer, S., Malladi, V. S., and Crusoe, M. R. **(2015)**. Portable workflow and tool descriptions with the CWL (Common Workflow Language) [v1; not peer reviewed]. *F1000Research*, **4**(ISCB Comm. J.), 278. Slides. 10.7490/f1000research.1110021.1

Andreessen, M. **(1993)**. NCSA Mosaic Technical Summary, *National Center for Supercomputer Applications*. Technical report.

Appel, R. D., Bairoch, A., and Hochstrasser, D. F. **(1994)**. A new generation of information retrieval tools for biologists: the example of the ExPASy WWW server. *Trends Biochem. Sci.*, **19**(6), 258–260. 10.1016/0968-0004(94)90153-8

Arakawa, K., Kido, N., Oshita, K., and Tomita, M. **(2010)**. G-language genome analysis environment with REST and SOAP web service interfaces. *Nucleic Acids Res.*, **38**(suppl 2,W1), W700–W705. 10.1093/nar/gkq315

Aranda, B., Blankenburg, H., Kerrien, S., Brinkman, F. S. L., Ceol, A., Chautard, E., Dana, J. M., De Las Rivas, J., Dumousseau, M., Galeota, E., Gaulton, A., Goll, J., Hancock, R. E. W., Isserlin, R., Jimenez, R. C., Kerssemakers, J., Khadake, J., Lynn, D. J., Michaut, M., O'Kelly, G., Ono, K., Orchard, S., Prieto, C., Razick, S., Rigina, O., Salwinski, L., Simonovic, M., Velankar, S., Winter, A., Wu, G., Bader, G. D., Cesareni, G., Donaldson, I. M., Eisenberg, D., Kleywegt, G. J., Overington, J., Ricard-Blum, S., Tyers, M., Albrecht, M., and Hermjakob, H. **(2011)**. PSICQUIC and PSISCORE: accessing and scoring molecular interactions. *Nat. Methods*, **8**(7), 528–529. 10.1038/nmeth.1637

Artimo, P., Jonnalagedda, M., Arnold, K., Baratin, D., Csardi, G., de Castro, E., Duvaud, S., Flegel, V., Fortier, A., Gasteiger, E., Grosdidier, A., Hernandez, C., Ioannidis, V., Kuznetsov, D., Liechti, R., Moretti, S., Mostaguir, K., Redaschi, N., Rossier, G., Xenarios, I., and Stockinger, H. **(2012)**. ExPASy: SIB bioinformatics resource portal. *Nucleic Acids Res.*, **40**(suppl 2,W1), W597–W603. 10.1093/nar/gks400

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. **(2000)**. Gene Ontology: tool for the unification of biology. *Nat. Genet.*, **25**(1), 25–29. 10.1038/75556

Attwood, T. K., Kell, D. B., McDermott, P., Marsh, J., Pettifer, S. R., and Thorne, D. **(2010)**. Utopia documents: linking scholarly literature with research data. *Bioinformatics*, **26**(18), i568–i574. 10.1093/bioinformatics/btq383

Attwood, T. K., Gisel, A., Eriksson, N. E., and Bongcam-Rudloff, E. **(2011)**. Concepts, Historical Milestones and the Central Place of Bioinformatics in Modern Biology: A European Perspective. Chapter 1 in Mahdavi, M. A. (*Ed.*). *Bioinformatics – Trends and Methodologies*, 3–38. InTech. 10.5772/23535

Barillot, E., Guyon, F., Cussat-Blanc, C., Viara, E., and Vaysseix, G. **(1998)**. HuGeMap: A Distributed and Integrated Human Genome Map Database. *Nucleic Acids Res.*, **26**(1), 106–107. 10.1093/nar/26.1.106

Barillot, E., Leser, U., Lijnzaad, P., Cussat-Blanc, C., Jungfer, K., Guyon, F., Vaysseix, G., Helgesen, C., and Rodriguez-Tomé, P. **(1999)**. A proposal for a standard CORBA interface for genome maps. *Bioinformatics*, **15**(2), 157–169. 10.1093/bioinformatics/15.2.157

Bateman, A. **(2005)**. Editorial. *Nucleic Acids Res.*, **33**(suppl 1,D1), suppl 1,D1. 10.1093/nar/gki133

Baxevanis, A. D. **(2000)**. The Molecular Biology Database Collection: an online compilation of relevant database resources. *Nucleic Acids Res.*, **28**(1), 1–7. 10.1093/nar/28.1.1

Benson, D., Boguski, M., Lipman, D. J., and Ostell, J. **(1990)**. The National Center for Biotechnology Information. *Genomics*, **6**(2), 389–391.

Benson, G. **(2007)**. Editorial. *Nucleic Acids Res.*, **35**(suppl 2,W1), suppl 2,W1. 10.1093/nar/gkm484

Benson, G. **(2015)**. Editorial: Nucleic Acids Research annual Web Server Issue in 2015. *Nucleic Acids Res.*, **43**(suppl 2,W1), W1–W2. 10.1093/nar/gkv581

Berendsen, H., van der Spoel, D., and van Drunen, R. **(1995)**. GROMACS: A message-passing parallel molecular dynamics implementation. *Comput. Phys. Commun.*, **91**(13), 43–56. 10.1016/0010-4655(95)00042-E

Berners-Lee, T., Cailliau, R., and Groff, J. F. **(1992)**. The World-Wide Web. *Comput. Networks ISDN*, **25**(4-5), 454–459.

Bernstein, H. J. **(2000)**. Recent changes to RasMol, recombining the variants. *Trends Biochem. Sci.*, **25**(9), 453 –455. 10.1016/S0968-0004(00)01606-6

Bhagat, J., Tanoh, F., Nzuobontane, E., Laurent, T., Orlowski, J., Roos, M., Wolstencroft, K., Aleksejevs, S., Stevens, R., Pettifer, S., Lopez, R., and Goble, C. A. **(2010)**. BioCatalogue: a universal catalogue of web services for the life sciences. *Nucleic Acids Res.*, **38**(suppl 2,W1), W689–W694. 10.1093/nar/gkq394

Bonnal, R. J., Aerts, J., Githinji, G., Goto, N., MacLean, D., Miller, C. A., Mishima, H., Pagani, M., Ramirez-Gonzalez, R., Smant, G., Strozzi, F., Syme, R., Vos, R., Wennblom, T. J., Woodcroft, B. J., Katayama, T., and Prins, P. **(2012)**. Biogem: an effective tool-based approach for scaling up open source software development in bioinformatics. *Bioinformatics*, **28**(7), 1035–1037. 10.1093/bioinformatics/bts080

Brazas, M. D., Yim, D., Yeung, W., and Ouellette, B. F. F. **(2012)**. A decade of web server updates at the bioinformatics links directory: 2003–2012. *Nucleic Acids Res.*, **40**(suppl 2,W1), W3–W12. 10.1093/nar/gks632

Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., Gaasterland, T., Glenisson, P., Holstege, F. C., Kim, I. F., Markowitz, V., Matese, J. C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., and Vingron, M. **(2001)**. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat. Genet.*, **29**, 365–371. 10.1038/ng1201-365

Brooksbank, C., Bergman, M. T., Apweiler, R., Birney, E., and Thornton, J. **(2014)**. The European Bioinformatics Institutes data resources 2014. *Nucleic Acids Res.*, **42**(suppl 1,D1), D18–D25. 10.1093/nar/gkt1206

Burks, C. **(1999)**. Molecular Biology Database List. *Nucleic Acids Res.*, **27**(1), 1–9. 10.1093/nar/27.1.1

Butt, D., Roger, A., and Blouin, C. **(2005)**. libcov: A C++ bioinformatic library to manipulate protein structures, sequence alignments and phylogeny. *BMC Bioinformatics*, **6**(1), 138. 10.1186/1471-2105-6-138

Cabot, E. L. and Beckenbach, A. T. **(1989)**. Simultaneous editing of multiple nucleic acid and protein sequences with ESEE. *Comput. Appl. Biosci.*, **5**(3), 233–234. 10.1093/bioinformatics/5.3.233

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T. **(2009)**. BLAST+: architecture and applications. *BMC Bioinformatics*, **10**(1), 421. 10.1186/1471-2105-10-421

Chang, W., Shindyalov, I., Pu, C., and Bourne, P. **(1994)**. Design and application of PDBlib, a C++ macromolecular class library. *Comput. Appl. Biosci.*, **10**(6), 575–586. 10.1093/bioinformatics/10.6.575

Chapman, B. and Chang, J. **(2000)**. Biopython: Python Tools for Computational Biology. *ACM SIGBIO Newsl.*, **20**(2), 15–19. 10.1145/360262.360268

Chervitz, S., Fuellen, G., Dagdigian, C., Brenner, S., Birney, E., and Korf, I. **(1998)**. Bioperl: Standard perl modules for bioinformatics. *Bio Informatics Technology and Systems (BITS)*, **10**, 1611–1618.

Clamp, M., Cuff, J., and Barton, G. J. **(1998)**. JalView— analysis and manipulation of multiple sequence alignments. *EMBnet News*, **5**(4), 16–21.

Clamp, M., Cuff, J., Searle, S. M., and Barton, G. J. **(2004)**. The Jalview Java alignment editor. *Bioinformatics*, **20**(3), 426–427. 10.1093/bioinformatics/btg430

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and de Hoon, M. J. L. **(2009)**. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**(11), 1422–1423. 10.1093/bioinformatics/btp163

Colet, M. and Herzog, R. **(1996)**. WWW2GCG, a web interface to the fGCGg biological sequences analysis software. *Comput. Graph.*, **20**(3), 445–450. 10.1016/0097-8493(96)00014-3

Corpas, M., Jimenez, R., Carbon, S. J., García, A., Garcia, L., Goldberg, T., Gomez, J., Kalderimis, A., Lewis, S. E., Mulvany, I., Pawlik, A., Rowland, F., Salazar, G., Schreiber, F., Sillitoe, I., Spooner, W. H., Thanki, A., Villaveces, J. M., Yachdav, G., and Hermjakob, H. **(2014)**. BioJS: an open source standard for biological visualisation its status in 2014 [v1; ref status: indexed, http://f1000r.es/2yy]. *F1000Research*, **3**, 55. 10.12688/f1000research.3-55.v1

Crabtree, J., Agrawal, S., Mahurkar, A., Myers, G. S., Rasko, D. A., and White, O. **(2014)**. Circleator: flexible circular visualization of genome-associated data with BioPerl and SVG. *Bioinformatics*, **30**(21), 3125–3127. 10.1093/bioinformatics/btu505

Crass, T., Antes, I., Basekow, R., Bork, P., Buning, C., Christensen, M., Claußen, H., Ebeling, C., Ernst, P., Gailus-Durner, V., Glatting, K.-H., Gohla, R., Gößling, F., Grote, K., Heidtke, K., Herrmann, A., O'Keeffe, S., Kießlich, O., Kolibal, S., Korbel, J. O., Lengauer, T., Liebich, I., van der Linden, M., Luz, H., Meissner, K., von Mering, C., Mevissen, H.-T., Mewes, H.-W., Michael, H., Mokrejs, M., Müller, T., Pospisil, H., Rarey, M., Reich, J. G., Schneider, R., Schomburg, D., Schulze-Kremer, S., Schwarzer, K., Sommer, I., Springstubbe, S., Suhai, S., Thoppae, G., Vingron, M., Warfsmann, J., Werner, T., Wetzler, D., Wingender, E., and Zimmer, R. **(2004)**. The Helmholtz Network for Bioinformatics: an integrative web portal for bioinformatics resources. *Bioinformatics*, **20**(2), 268–270. 10.1093/bioinformatics/btg398

Cunningham, F., Amode, M. R., Barrell, D., Beal, K., Billis, K., Brent, S., Carvalho-Silva, D., Clapham, P., Coates, G., Fitzgerald, S., Gil, L., Girón, C. G., Gordon, L., Hourlier, T., Hunt, S. E., Janacek, S. H., Johnson, N., Juettemann, T., Kähäri, A. K., Keenan, S., Martin, F. J., Maurel, T., McLaren, W., Murphy, D. N., Nag, R., Overduin, B., Parker, A., Patricio, M., Perry, E., Pignatelli, M., Riat, H. S., Sheppard, D., Taylor, K., Thormann, A., Vullo, A., Wilder, S. P., Zadissa, A., Aken, B. L., Birney, E., Harrow, J., Kinsella, R., Muffato, M., Ruffier, M., Searle, S. M., Spudich, G., Trevanion, S. J., Yates, A., Zerbino, D. R., and Flicek, P. **(2015)**. Ensembl 2015. *Nucleic Acids Research*, **43**(suppl 1,D1), D662–D669. 10.1093/nar/gku1010

Daub, J., Gardner, P. P., Tate, J., Ramsköld, D., Manske, M., Scott, W. G.,Weinberg, Z., Griffiths-Jones, S., and Bateman, A. **(2008)**. The RNA WikiProject: Community annotation of RNA families. *RNA*, **14**(12), 2462–2464. 10.1261/rna.1200508

del Toro, N., Dumousseau, M., Orchard, S., Jimenez, R. C., Galeota, E., Launay, G., Goll, J., Breuer, K., Ono, K., Salwinski, L., and Hermjakob, H. **(2013)**. A new reference implementation of the PSICQUIC web service. *Nucleic Acids Res.*, **41**(suppl 2,W1), W601–W606. 10.1093/nar/gkt392

D'Elia, D., Gisel, A., Eriksson, N.-E., Kossida, S., Mattila, K., Kľučár, Ľ., and Bongcam-Rudloff, E. **(2009)**. The 20th anniversary of EMBnet: 20 years of bioinformatics for the Life Sciences community. *BMC Bioinformatics*, **10**(Suppl 6), S1. 10.1186/1471-2105-10-S6-S1

Demir, E., Cary, M. P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Wu, G., D'Eustachio, P., Schaefer, C., Luciano, J., Schacherer, F., Martinez-Flores, I., Hu, Z., Jimenez-Jacinto, V., Joshi-Tope, G., Kandasamy, K., Lopez-Fuentes, A. C., Mi, H., Pichler, E., Rodchenkov, I., Splendiani, A., Tkachev, S., Zucker, J., Gopinath, G., Rajasimha, H., Ramakrishnan, R., Shah, I., Syed, M., Anwar, N., Babur, O., Blinov, M., Brauner, E., Corwin, D., Donaldson, S., Gibbons, F., Goldberg, R., Hornbeck, P., Luna, a., Murray-Rust, P., Neumann, E., Reubenacker, O., Samwald, M., van Iersel, M., Wimalaratne, S., Allen, K., Braun, B., Whirl-Carrillo, M., Cheung, K. H., Dahlquist, K., Finney, A., Gillespie, M., Glass, E., Gong, L., Haw, R., Honig, M., Hubaut, O., Kane, D., Krupa, S., Kutmon, M., Leonard, J., Marks, D., Merberg, D., Petri, V., Pico, A., Ravenscroft, D., Ren, L., Shah, N., Sunshine, M., Tang, R., Whaley, R., Letovksy, S., Buetow, K. H., Rzhetsky, A., Schachter, V., Sobral, B. S., Dogrusoz, U., McWeeney, S., Aladjem, M., Birney, E., Collado-Vides, J., Goto, S., Hucka, M., Novère, N. L., Maltsev, N., Pandey, A., Thomas, P., Wingender, E., Karp, P. D., Sander, C., and Bader, G. D. **(2010)**. The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.*, **28**, 1308. 10.1038/nbt.1666

Devereux, J., Haeberli, P., and Smithies, O. **(1984)**. A comprehensive set of sequence analysis programs for the vax. *Nucleic Acids Res.*, **12**(1Part1), 387–395. 10.1093/nar/12.1Part1.387

Devignes, M.-D. and Moreau, Y. **(2014)**. ECCB 2014: The 13th European Conference on Computational Biology. *Bioinformatics*, **30**(17), i345–i348. 10.1093/bioinformatics/btu512

Devignes, M. D., Franiatte, P., Messai, N., Napoli, A., and Smaïl-Tabbone, M. **(2010)**. BioRegistry: Automatic extraction of metadata for biological database retrieval and discovery. *Int. J. of Metadata, Semantics and Ontologies*, **5**(3), 184–193. 10.1504/IJMSO.2010.034043

Di Tommaso, P., Chatzou, M., Baraja, P. P., and Notredame, C. **(2014)**. Nextflow: A novel tool for highly scalable computational pipelines. *figshare.* Poster. 10.6084/m9.figshare.1254958

Discala, C., Ninnin, M., Achard, F., Barillot, E., and Vaysseix, G. **(1999)**. DBcat: a catalog of biological databases. *Nucleic Acids Res.*, **27**(1), 10–11. 10.1093/nar/27.1.10

Discala, C., Benigni, X., Barillot, E., and Vaysseix, G. **(2000)**. DBcat: a catalog of 500 biological databases. *Nucleic Acids Res.*, **28**(1), 8–9. 10.1093/nar/28.1.8

Doelz, R. **(1992)**. The EMBnet Project—European molecular biology network. *Comput. Networks ISDN*, **25**(45), 464–468. 10.1016/0169-7552(92)90041-N

Doelz, R. **(1994)**. Hierarchical Access System for Sequence Libraries in Europe (HASSLE): a tool to access sequence databases remotely. *Comput. Appl. Biosci.*, **10**(1), 31–34. 10.1093/bioinformatics/10.1.31

Doelz, R., Eggenberger, F., and Wadley, C. **(1994)**. Biocomputing on a server network. *EMBnet News*, **1**(2), 6–8.

Donlin, M. J. **(2007)**. Using the Generic Genome Browser (GBrowse). In *Curr. Protoc. Bioinformatics*, (17), 9.9.1–9.9.24. John Wiley & Sons, Inc. 10.1002/0471250953.bi0909s17

Döring, A., Weese, D., Rausch, T., and Reinert, K. **(2008)**. SeqAn An efficient, generic C++ library for sequence analysis. *BMC Bioinformatics*, **9**(1), 11. 10.1186/1471-2105-9-11

Dowell, R., Jokerst, R., Day, A., Eddy, S., and Stein, L. **(2001)**. The Distributed Annotation System. *BMC Bioinformatics*, **2**(1), 7. 10.1186/1471-2105-2-7

Down, T. A., Piipari, M., and Hubbard, T. J. P. **(2011)**. Dalliance: interactive genome viewing on the web. *Bioinformatics*, **27**(6), 889–890. 10.1093/bioinformatics/btr020

Dutheil, J., Gaillard, S., Bazin, E., Glémin, S., Ranwez, V., Galtier, N., and Belkhir, K. **(2006)**. Bio++: a set of C++ libraries for sequence analysis, phylogenetics, molecular evolution and population genetics. *BMC Bioinformatics*, **7**(1), 188. 10.1186/1471-2105-7-188

Dysvik, B. and Jonassen, I. **(2001)**. J-Express: exploring gene expression data using Java. *Bioinformatics*, **17**(4), 369–370. 10.1093/bioinformatics/17.4.369

Eilbeck, K., Lewis, S. E., Mungall, C. J., Yandell, M., Stein, L., Durbin, R., and Ashburner, M. **(2005)**. The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biol.*, **6**, R44. 10.1186/gb-2005-6-5-r44

Eisen, J. A. **(1997)**. The Genetic Data Environment: A User Modifiable and Expandable Multiple Sequence Analysis Package (A GUIDE for the Graphical User Interface (GUI) GDE). In Swindell, S. (*Ed.*). *Sequence Data Analysis Guidebook*. *Methods Mol. Med.*, **70**, 13–38. Springer New York. 10.1385/0-89603-358-9:13

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. **(1998)**. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, **95**(25), 14863–14868.

Emmert, D. B., Stoehr, P. J., Stoesser, G., and Cameron, G. N. **(1994)**. The European Bioinformatics Institute (EBI) databases. *Nucleic Acids Res.*, **22**(17), 3445–3449. 10.1093/nar/22.17.3445

Etzold, T. **(1994)**. The Sequence Retrieval System (SRS) on the World Wide Web. *EMBnet News*, **1**(2), 5–6.

Etzold, T. and Argos, P. **(1993)**. SRS—an indexing and retrieval tool for flat file data libraries. *Comput. Appl. Biosci.*, **9**(1), 49–57. 10.1093/bioinformatics/9.1.49

Etzold, T., Ulyanov, A., and Argos, P. **(1996)**. SRS: Information retrieval system for molecular biology data banks. In *Computer Methods for Macromolecular Sequence Analysis*. *Methods Enzymol.*, **266**, 114–128. Academic Press. 10.1016/S0076-6879(96)66010-8

Felsenstein, J. **(1981)**. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, **17**(6), 368–376.

Felsenstein, J. **(1985)**. Confidence Limits on Phylogenies: An Approach Using the Bootstrap. *Evolution*, **39**(4), 783–791. 10.2307/2408678

Felsenstein, J. **(1989)**. PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics*, **5**, 164–166.

Fernández-Suárez, X. M., Rigden, D. J., and Galperin, M. Y. **(2014)**. The 2014 Nucleic Acids Research Database Issue and an updated NAR online Molecular Biology Database Collection. *Nucleic Acids Res.*, **42**(suppl 1,D1), D1–D6. 10.1093/nar/gkt1282

Field, D., Tiwari, B., Booth, T., Houten, S., Swan, D., Bertrand, N., and Thurston, M. **(2006)**. Open software for biologists: from famine to feast. *Nat. Biotechnol.*, **24**(7), 801–803. 10.1038/nbt0706-801

Field, D., Garrity, G., Gray, T., Morrison, N., Selengut, J., Sterk, P., Tatusova, T., Thomson, N., Allen, M. J., Angiuoli, S. V., Ashburner, M., Axelrod, N., Baldauf, S., Ballard, S., Boore, J., Cochrane, G., Cole, J., Dawyndt, P., De Vos, P., DePamphilis, C., Edwards, R., Faruque, N., Feldman, R., Gilbert, J., Gilna, P., Glockner, F. O., Goldstein, P., Guralnick, R., Haft, D., Hancock, D., Hermjakob, H., Hertz-Fowler, C., Hugenholtz, P., Joint, I., Kagan, L., Kane, M., Kennedy, J., Kowalchuk, G., Kottmann, R., Kolker, E., Kravitz, S., Kyrpides, N., Leebens-Mack, J., Lewis, S. E., Li, K., Lister, A. L., Lord, P., Maltsev, N., Markowitz, V., Martiny, J., Methe, B., Mizrachi, I., Moxon, R., Nelson, K., Parkhill, J., Proctor, L., White, O., Sansone, S. A., Spiers, A., Stevens, R., Swift, P., Taylor, C., Tateno, Y., Tett, A., Turner, S., Ussery, D., Vaughan, B., Ward, N., Whetzel, T., San Gil, I., Wilson, G., and Wipat, A. **(2008)**. The minimum information about a genome sequence (MIGS) specification. *Nat. Biotechnol.*, **26**, 541–547. 10.1038/nbt1360

Finn, R. D., Gardner, P. P., and Bateman, A. **(2012)**. Making your database available through Wikipedia: the pros and cons. *Nucleic Acids Res.*, **40**(suppl 1,D1), D9–D12. 10.1093/nar/gkr1195

Fowers, J., Brown, G., Cooke, P., and Stitt, G. **(2012)**. A Performance and Energy Comparison of FPGAs, GPUs, and Multicores for Sliding-window Applications. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '12)*, 47–56. ACM. 10.1145/2145694.2145704

Fox, J. A., Butland, S. L., McMillan, S., Campbell, G., and Ouellette, B. F. F. **(2005)**. The Bioinformatics Links Directory: a Compilation of Molecular Biology Web Servers. *Nucleic Acids Res.*, **33**(suppl 2,W1), W3–W24. 10.1093/nar/gki594

Galperin, M. Y. and Fernández-Suárez, X. M. **(2012)**. The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. *Nucleic Acids Res.*, **40**(suppl 1,D1), D1–D8. 10.1093/nar/gkr1196

Galperin, M. Y., Rigden, D. J., and Fernández-Suárez, X. M. **(2015)**. The 2015 Nucleic Acids Research Database Issue and Molecular Biology Database Collection. *Nucleic Acids Res.*, **43**(suppl 1,D1), D1–D5. 10.1093/nar/gku1241

Garamszegi, S., Mesirov, J. P., and The GenomeSpace Team **(2015)**. GenomeSpace: An environment for frictionless bioinformatics [v1; not peer reviewed]. *F1000Research*, **4**(ISCB Comm. J.), 349. Poster. 10.7490/f1000research.1110097.1

Gardner, P. P., Daub, J., Tate, J., Moore, B. L., Osuch, I. H., Griffiths-Jones, S., Finn, R. D., Nawrocki, E. P., Kolbe, D. L., Eddy, S. R., and Bateman, A. **(2011)**. Rfam: Wikipedia, clans and the decimal release. *Nucleic Acids Res.*, **39**(suppl 1,D1), D141–D145. 10.1093/nar/gkq1129

Gasteiger, E., Gattiker, A., Hoogland, C., Ivanyi, I., Appel, R. D., and Bairoch, A. **(2003)**. ExPASy: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res.*, **31**(13), 3784–3788. 10.1093/nar/gkg563

Gaudet, P., Bairoch, A., Field, D., Sansone, S.-A., Taylor, C., Attwood, T. K., Bateman, A., Blake, J. A., Bult, C. J., Cherry, J. M., Chisholm, R. L., Cochrane, G., Cook, C. E., Eppig, J. T., Galperin, M. Y., Gentleman, R., Goble, C. A., Gojobori, T., Hancock, J. M., Howe, D. G., Imanishi, T., Kelso, J., Landsman, D., Lewis, S. E., Mizrachi, I. K., Orchard, S., Ouellette, B. F. F., Ranganathan, S., Richardson, L., Rocca-Serra, P., Schofield, P. N., Smedley, D., Southan, C., Tan, T. W., Tatusova, T., Whetzel, P. L., White, O., and Yamasaki, C. **(2011)**. Towards BioDBCore: a community-defined information specification for biological databases. *Nucleic Acids Res.*, **39**(suppl 1,D1), D7–D10. 10.1093/nar/gkq1173

Gentleman, R., Carey, V., Bates, D., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J., and Zhang, J. **(2004)**. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.*, **5**(10), R80. 10.1186/gb-2004-5-10-r80

Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., Zhang, Y., Blankenberg, D., Albert, I., Taylor, J., Miller, W., Kent, W. J., and Nekrutenko, A. **(2005)**. Galaxy: A platform for interactive large-scale genome analysis. *Genome Res.*, **15**(10), 1451–1455. 10.1101/gr.4086505

Gilbert, D. **(1998)**. Free Software in Molecular Biology for Macintosh and MS Windows computers. *Access*, (June).

Gilbert, D. **(1999)**. Free Software in Molecular Biology for Macintosh and MS Windows Computers. In Misener, S., Krawetz, S. A. (*Eds.*). *Bioinformatics Methods and Protocols*. *Methods Mol. Biol.*, **132**, 149–184. Humana Press. 10.1385/1-59259-192-2:149

Gilbert, D. **(2004)**. Bioinformatics software resources. *Brief. Bioinform.*, **5**(3), 300–304. 10.1093/bib/5.3.300

Gleeson, T. J. and Staden, R. **(1991)**. An X windows and UNIX implementation of our sequence analysis package. *Comput. Appl. Biosci.*, **7**(3), 398. 10.1093/bioinformatics/7.3.398

Goble, C. A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D., Newman, D., Borkum, M., Bechhofer, S., Roos, M., Li, P., and De Roure, D. **(2010)**. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Res.*, **38**(suppl 2,W1), W677–W682. 10.1093/nar/gkq429

Goecks, J., Nekrutenko, A., Taylor, J., and Team, T. G. **(2010)**. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.*, **11**(8), R86. 10.1186/gb-2010-11-8-r86

Gómez, J., García, L. J., Salazar, G. A., Villaveces, J., Gore, S., García, A., Martín, M. J., Launay, G., Alcántara, R., del Toro, N., Dumousseau, M., Orchard, S., Velankar, S., Hermjakob, H., Zong, C., Ping, P., Corpas, M., and Jiménez, R. C. **(2013)**. BioJS: an open source JavaScript framework for biological data visualization. *Bioinformatics*, **29**(8), 1103–1104. 10.1093/bioinformatics/btt100

Goto, N., Prins, P., Nakao, M., Bonnal, R., Aerts, J., and Katayama, T. **(2010)**. BioRuby: bioinformatics software for the Ruby programming language. *Bioinformatics*, **26**(20), 2617–2619. 10.1093/bioinformatics/btq475

Goujon, M., McWilliam, H., Li, W., Valentin, F., Squizzato, S., Paern, J., and Lopez, R. **(2010)**. A new bioinformatics analysis tools framework at EMBL–EBI. *Nucleic Acids Res.*, **38**(suppl 2,W1), W695–W699. 10.1093/nar/gkq313

Gremme, G., Steinbiss, S., and Kurtz, S. **(2013)**. GenomeTools: A Comprehensive Software Library for Efficient Processing of Structured Genome Annotations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **10**(3), 645–656. 10.1109/TCBB.2013.68

Griffiths, M. **(2008)**. ICT and CO2 emissions. *POSTnote*, (319). The Parliamentary Office of Science and Technology (POST), UK.

Gruber, A. R., Lorenz, R., Bernhart, S. H., Neuböck, R., and Hofacker, I. L. **(2008)**. The Vienna RNA Websuite. *Nucleic Acids Res.*, **36**(suppl 2,W1), W70–W74. 10.1093/nar/gkn188

Gundersen, S., Kalaš, M., Abul, O., Frigessi, A., Hovig, E., and Sandve, G. K. **(2011)**. Identifying elemental genomic track types and representing them uniformly. *BMC Bioinformatics*, **12**(1), 494. 10.1186/1471-2105-12-494

Han, M. and Zmasek, C. **(2009)**. phyloXML: XML for evolutionary biology and comparative genomics. *BMC Bioinformatics*, **10**(1), 356. 10.1186/1471-2105-10-356

Harper, R. A. **(1996)**. EMBnet: an institute without walls. *Trends Biochem. Sci.*, **21**(4), 150–152. 10.1016/S0968-0004(96)80170-8

Harris, N. L., Cock, P. J. A., Chapman, B. A., Goecks, J., Hotz, H.-R., and Lapp, H. **(2015)**. The bioinformatics open source conference (bosc) 2013. *Bioinformatics*, **31**(2), 299–300. 10.1093/bioinformatics/btu413

Harte, N., Silventoinen, V., Quevillon, E., Robinson, S., Kallio, K., Fustero, X., Patel, P., Jokinen, P., and Lopez, R. **(2004)**. Public web-based services from the European Bioinformatics Institute. *Nucleic Acids Res.*, **32**(suppl 2,W1), W3–W9. 10.1093/nar/gkh405

Hekkelman, M. L., te Beek, T. A. H., Pettifer, S. R., Thorne, D., Attwood, T. K., and Vriend, G. **(2010)**. WIWS: a protein structure bioinformatics Web service collection. *Nucleic Acids Res.*, **38**(suppl 2,W1), W719–W723. 10.1093/nar/gkq453

Henikoff, S. **(1993)**. Sequence analysis by electronic mail server. *Trends Biochem. Sci.*, **18**(7), 267–268. 10.1016/0968-0004(93)90179-Q

Henry, V. J., Bandrowski, A. E., Pepin, A.-S., Gonzalez, B. J., and Desfeux, A. **(2014)**. OMICtools: an informative directory for multi-omic data analysis. *Database*, **2014**. 10.1093/database/bau069

Hermjakob, H., Montecchi-Palazzi, L., Bader, G., Wojcik, J., Salwinski, L., Ceol, A., Moore, S., Orchard, S., Sarkans, U., von Mering, C., Roechert, B., Poux, S., Jung, E., Mersch, H., Kersey, P., Lappe, M., Li, Y., Zeng, R., Rana, D., Nikolski, M., Husi, H., Brun, C., Shanker, K., Grant, S., Sander, C., Bork, P., Zhu, W., Pandey, A., Brazma, A., Jacq, B., Vidal, M., Sherman, D., Legrain, P., Cesareni, G., Xenarios, I., Eisenberg, D., Steipe, B., Hogue, C., and Apweiler, R. **(2004)**. The HUPO PSIs Molecular Interaction format—A community standard for the representation of protein interaction data. *Nat. Biotechnol.*, **22**(2), 177–183. 10.1038/nbt926

Herráez, A. **(2006)**. Biomolecules in the computer: Jmol to the rescue. *Biochem. Mol. Biol. Educ.*, **34**(4), 255–261. 10.1002/bmb.2006.494034042644

Hess, B., Kutzner, C., van der Spoel, D., and Lindahl, E. **(2008)**. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *J. Chem. Theory Comput.*, **4**(3), 435–447. 10.1021/ct700301q

Higgins, D. G. and Sharp, P. M. **(1988)**. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, **73**(1), 237 – 244. 10.1016/0378-1119(88)90330-7

Higgins, D. G., Bleasby, A. J., and Fuchs, R. **(1992)**. Clustal v: improved software for multiple sequence alignment. *Comput. Appl. Biosci.*, **8**(2), 189–191. 10.1093/bioinformatics/8.2.189

Hofacker, I., Fontana, W., Stadler, P., Bonhoeffer, L., Tacker, M., and Schuster, P. **(1994)**. Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, **125**, 167–188. 10.1007/BF00818163

Hofacker, I. L. **(2003)**. Vienna RNA secondary structure server. *Nucleic Acids Res.*, **31**(13), 3429–3431. 10.1093/nar/gkg599

Hogeweg, P. **(2011)**. The Roots of Bioinformatics in Theoretical Biology. *PLoS Comput. Biol.*, **7**(3), e1002021. 10.1371/journal.pcbi.1002021

Holland, R. C. G., Down, T. A., Pocock, M., Prlić, A., Huen, D., James, K., Foisy, S., Dräger, A., Yates, A., Heuer, M., and Schreiber, M. J. **(2008)**. BioJava: an open-source framework for bioinformatics. *Bioinformatics*, **24**(18), 2096–2097. 10.1093/bioinformatics/btn397

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. **(2006)**. COPASI—a COmplex PAthway SImulator. *Bioinformatics*, **22**(24), 3067–3074. 10.1093/bioinformatics/btl485

Hu, J., Mungall, C., Nicholson, D., and Archibald, A. L. **(1998)**. Design and implementation of a CORBA-based genome mapping system prototype. *Bioinformatics*, **14**(2), 112–120. 10.1093/bioinformatics/14.2.112

Hubbard, T., Barker, D., Birney, E., Cameron, G., Chen, Y., Clark, L., Cox, T., Cuff, J., Curwen, V., Down, T., Durbin, R., Eyras, E., Gilbert, J., Hammond, M., Huminiecki, L., Kasprzyk, A., Lehvaslaiho, H., Lijnzaad, P., Melsopp, C., Mongin, E., Pettett, R., Pocock, M., Potter, S., Rust, A., Schmidt, E., Searle, S., Slater, G., Smith, J., Spooner, W., Stabenau, A., Stalker, J., Stupka, E., Ureta-Vidal, A., Vastrik, I., and Clamp, M. **(2002)**. The Ensembl genome database project. *Nucleic Acids Res.*, **30**(1), 38–41. 10.1093/nar/30.1.38

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., and the rest of the SBML Forum: Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le Novère, N., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R., Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., and Wang, J. **(2003)**. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**(4), 524–531. 10.1093/bioinformatics/btg015

Hucka, M., Finney, A., Bornstein, B. J., Keating, S. M., Shapiro, B. E., Matthews, J., Kovitz, B. L., Schilstra, M. J., Funahashi, A., Doyle, J. C., and Kitano, H. **(2004)**. Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML) project. *Syst. Biol. (Stevenage)*, **1**(1), 41–53. 10.1049/sb:20045008

Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M. R., Li, P., and Oinn, T. **(2006)**. Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.*, **34**(suppl 2,W1), W729–W732. 10.1093/nar/gkl320

Humphrey, W., Dalke, A., and Schulten, K. **(1996)**. VMD – Visual Molecular Dynamics. *J. Mol. Graph.*, **14**(1), 33–38. 10.1016/0263-7855(96)00018-5

Ison, J., Kalaš, M., Jonassen, I., Bolser, D., Uludag, M., McWilliam, H., Malone, J., Lopez, R., Pettifer, S., and Rice, P. **(2013)**. EDAM: An ontology of bioinformatics operations, types of data and identifiers, topics, and formats. *Bioinformatics*, **29**(10), 1325–1332. 10.1093/bioinformatics/btt113

Jenkinson, A., Albrecht, M., Birney, E., Blankenburg, H., Down, T., Finn, R., Hermjakob, H., Hubbard, T., Jimenez, R., Jones, P., Kähäri, A., Kulesha, E., Macías, J., Reeves, G., and Prlić, A. **(2008)**. Integrating biological data - the Distributed Annotation System. *BMC Bioinformatics*, **9**(Suppl 8), S3. 10.1186/1471-2105-9-S8-S3

Johnson, M., Zaretskaya, I., Raytselis, Y., Merezhuk, Y., McGinnis, S., and Madden, T. L. **(2008)**. NCBI BLAST: a better web interface. *Nucleic Acids Res.*, **36**(suppl 2,W1), W5–W9. 10.1093/nar/gkn201

Jungfer, K. and Rodriguez-Tomé, P. **(1998)**. Mapplet: a CORBA-based genome map viewer. *Bioinformatics*, **14**(8), 734–738. 10.1093/bioinformatics/14.8.734

Juty, N., Le Novère, N., and Laibe, C. **(2012)**. Identifiers.org and MIRIAM Registry: community resources to provide persistent identification. *Nucleic Acids Res.*, **40**(suppl 1,D1), D580–D586. 10.1093/nar/gkr1097

Kaján, L., Yachdav, G., Vicedo, E., Steinegger, M., Mirdita, M., Angermüller, C., Böhm, A., Domke, S., Ertl, J., Mertes, C., Reisinger, E., Staniewski, C., and Rost, B. **(2013)**. Cloud Prediction of Protein Structure and Function with PredictProtein for Debian. *BioMed Res. Int.*, **2013**, 1–6. 10.1155/2013/398968

Kaján, L., Hopf, T., Kalaš, M., Marks, D., and Rost, B. **(2014)**. FreeContact: fast and free software for protein contact prediction from residue co-evolution. *BMC Bioinformatics,* **15**(1), 85. 10.1186/1471-2105-15-85

Kalaš, M., Puntervoll, P., Joseph, A., Bartaševičiūtė (now Karosiene), E., Töpfer, A., Venkataraman, P., Pettifer, S., Bryne, J., Ison, J., Blanchet, C., Rapacki, K., and Jonassen, I. **(2010)**. BioXSD: the common data-exchange format for everyday bioinformatics web services. *Bioinformatics*, **26**, i540–i546. 10.1093/bioinformatics/btq391

Kalaš, M., Gundersen, S., Kaján, L., Ison, J., Pettifer, S., Blanchet, C., Lopez, R., Rapacki, K., and Jonassen, I. **(2014)**. BioXSD: a data model for sequences, alignments, features and measurements. *F1000Posters*, **5**, 1503. Poster.

Kalaš, M., Gundersen, S., Kaján, L., Ison, J., Pettifer, S., Blanchet, C., Lopez, R., Rapacki, K., and Jonassen, I. **(2015)**. BioXSD — a data model for sequences, alignments, features, measured and inferred values [v1; not peer reviewed]. *F1000Research*, **4**(ISCB Comm. J.), 425. Poster. 10.7490/f1000research.1110178.1

Kallio, M. A., Tuimala, J., Hupponen, T., Klemela, P., Gentile, M., Scheinin, I., Koski, M., Kaki, J., and Korpelainen, E. **(2011)**. Chipster: user-friendly analysis software for microarray and other high-throughput data. *BMC Genomics*, **12**(1), 507. 10.1186/1471-2164-12-507

Kasprzyk, A. **(2011)**. BioMart: driving a paradigm change in biological data management. *Database*, **2011**. 10.1093/database/bar049

Katayama, T., Arakawa, K., Nakao, M., Ono, K., Aoki-Kinoshita, K. F., Yamamoto, Y., Yamaguchi, A., Kawashima, S., Chun, H.-W., Aerts, J., Aranda, B., Barboza, L. H., Bonnal, R. J., Bruskiewich, R., Bryne, J. C., Fernández, J. M., Funahashi, A., Gordon, P. M., Goto, N., Groscurth, A., Gutteridge, A., Holland, R., Kano, Y., Kawas, E. A., Kerhornou, A., Kibukawa, E., Kinjo, A. R., Kuhn, M., Lapp, H., Lehvaslaiho, H., Nakamura, H., Nakamura, Y., Nishizawa, T., Nobata, C., Noguchi, T., Oinn, T. M., Okamoto, S., Owen, S., Pafilis, E., Pocock, M., Prins, P., Ranzinger, R., Reisinger, F., Salwinski, L., Schreiber, M., Senger, M., Shigemoto, Y., Standley, D. M., Sugawara, H., Tashiro, T., Trelles, O., Vos, R. A., Wilkinson, M. D., York, W., Zmasek, C. M., Asai, K., and Takagi, T. **(2010)**. The DBCLS BioHackathon: standardization and interoperability for bioinformatics web services and workflows. *J. Biomed. Sem.*, **1**(1), 8. 10.1186/2041-1480-1-8

Katayama, T., Wilkinson, M., Micklem, G., Kawashima, S., Yamaguchi, A., Nakao, M., Yamamoto, Y., Okamoto, S., Oouchida, K., Chun, H.-W., Aerts, J., Afzal, H., Antezana, E., Arakawa, K., Aranda, B., Belleau, F., Bolleman, J., Bonnal, R. J., Chapman, B., Cock, P. J., Eriksson, T., Gordon, P. M., Goto, N., Hayashi, K., Horn, H., Ishiwata, R., Kaminuma, E., Kasprzyk, A., Kawaji, H., and Kido, N. **(2013)**. The 3rd DBCLS BioHackathon: improving life science data integration with Semantic Web technologies. *J. Biomed. Sem.*, **4**(1), 6. 10.1186/2041-1480-4-6

Katayama, T., Wilkinson, M. D., Aoki-Kinoshita, K., Kawashima, S., Yamamoto, Y., Yamaguchi, A., Okamoto, S., Kawano, S., Kim, J.-D., Wang, Y., Wu, H., Kano, Y., Ono, H., Bono, H., Kocbek, S., Aerts, J., Akune, Y., Antezana, E., Arakawa, K., Aranda, B., Baran, J., Bolleman, J., Bonnal, R. J. P., Buttigieg, P. L., Campbell, M. P., Chen, Y.-a., Chiba, H., Cock, P. J. A., Cohen, K. B., Constantin, A., Duck, G., Dumontier, M., Fujisawa, T., Fujiwara, T., Goto, N., Hoehndorf, R., Igarashi, Y., Itaya, H., Ito, M., Iwasaki, W., Kalaš, M., Katoda, T., Kim, T., Kokubu, A., Komiyama, Y., Kotera, M., Laibe, C., Lapp, H., Lütteke, T., Marshall, S., Mori, T., Mori, H., Morita, M., Murakami, K., Nakao, M., Narimatsu, H., Nishide, H., Nishimura, Y., Nystrom-Persson, J., Ogishima, S., Okamura, Y., Okuda, S., Oshita, K., Packer, N. H., Prins, P., Ranzinger, R., Rocca-Serra, P., Sansone, S., Sawaki, H., Shin, S.-H., Splendiani, A., Strozzi, F., Tadaka, S., Toukach, P., Uchiyama, Umezaki, M., Vos, R., Whetzel, P. L., Yamada, I., Yamada, I., Yamashita, C., Yamashita, R., York, W. S., Zmasek, C. M., Kawamoto, S., and Takagi, T. **(2014)**. BioHackathon series in 2011 and 2012: penetration of ontology and linked data in life science domains. *J. Biomed. Sem.*, **5**(1), 5. 10.1186/2041-1480-5-5

Kawano, S., Watanabe, T., Mizuguchi, S., Araki, N., Katayama, T., and Yamaguchi, A. **(2014)**. TogoTable: cross-database annotation system using the Resource Description Framework (RDF) data model. *Nucleic Acids Res.*, **42**(suppl 2,W1), W442–W448. 10.1093/nar/gku403

Kawashima, S., Katayama, T., Sato, Y., and Kanehisa, M. **(2003)**. KEGG API: A Web Service Using SOAP/WSDL to Access the KEGG System. *Genome Informatics*, **14**, 673–674.

Kearse, M., Moir, R., Wilson, A., Stones-Havas, S., Cheung, M., Sturrock, S., Buxton, S., Cooper, A., Markowitz, S., Duran, C., Thierer, T., Ashton, B., Meintjes, P., and Drummond, A. **(2012)**. Geneious Basic: An integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics*, **28**(12), 1647–1649. 10.1093/bioinformatics/bts199

Kent,W. J. and Haussler, D. **(2001)**. Assembly of theWorking Draft of the Human Genome with GigAssembler. *Genome Res.*, **11**(9), 1541–1548. 10.1101/gr.183201

Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., Haussler, and David **(2002)**. The Human Genome Browser at UCSC. *Genome Res.*, **12**(6), 996–1006. 10.1101/gr.229102

Kent, W. J., Zweig, A. S., Barber, G., Hinrichs, A. S., and Karolchik, D. **(2010)**. BigWig and BigBed: enabling browsing of large distributed datasets. *Bioinformatics*, **26**(17), 2204–2207. 10.1093/bioinformatics/btq351

Kerrien, S., Orchard, S., Montecchi-Palazzi, L., Aranda, B., Quinn, A., Vinod, N., Bader, G., Xenarios, I., Wojcik, J., Sherman, D., Tyers, M., Salama, J., Moore, S., Ceol, A., Chatr-aryamontri, A., Oesterheld, M., Stumpflen, V., Salwinski, L., Nerothin, J., Cerami, E., Cusick, M., Vidal, M., Gilson, M., Armstrong, J., Woollard, P., Hogue, C., Eisenberg, D., Cesareni, G., Apweiler, R., and Hermjakob, H. **(2007)**. Broadening the horizon - level 2.5 of the HUPO-PSI format for molecular interactions. *BMC Biology*, **5**(1), 44. 10.1186/1741-7007-5-44

Klepper, K. and Drabløs, F. **(2013)**. MotifLab: a tools and data integration workbench for motif discovery and regulatory sequence analysis. *BMC Bioinformatics*, **14**(1), 9. 10.1186/1471-2105-14-9

Klingström, T., Soldatova, L., Stevens, R., Roos, E. T., Swertz, M. A., Müller, K. M., Kalaš, M., Lambrix, P., Taussig, M. J., Litton, J.-E., Landegren, U., and Bongcam-Rudloff, E. **(2013)**. Workshop on laboratory protocol standards for the molecular methods database. *N. Biotechnol.*, **30**(2), 109–113. 10.1016/j.nbt.2012.05.019

Kodama, Y., Mashima, J., Kosuge, T., Katayama, T., Fujisawa, T., Kaminuma, E., Ogasawara, O., Okubo, K., Takagi, T., and Nakamura, Y. **(2015)**. The DDBJ Japanese Genotype-phenotype Archive for genetic and phenotypic human data. *Nucleic Acids Res.*, **43**(suppl 1,D1), D18–D22. 10.1093/nar/gku1120

Köhler, J., Baumbach, J., Taubert, J., Specht, M., Skusa, A., Rüegg, A., Rawlings, C., Verrier, P., and Philippi, S. **(2006)**. Graph-based analysis and visualization of experimental results with ONDEX. *Bioinformatics*, **22**(11), 1383–1390. 10.1093/bioinformatics/btl081

Kopecky, J., Vitvar, T., Bournez, C., and Farrell, J. **(2007)**. SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Internet Comput.*, **11**(6), 60–67. 10.1109/MIC.2007.134

Kottmann, R., Gray, T., Murphy, S., Kagan, L., Kravitz, S., Lombardot, T., Field, D., and Glöckner, F. O. **(2008)**. A standard MIGS/MIMS compliant XML Schema: Toward the development of the Genomic Contextual Data Markup Language (GCDML). *OMICS*, **12**, 115–121. 10.1089/omi.2008.0A10

Krishnamurthy, L., Nadeau, J., Ozsoyoglu, G., Ozsoyoglu, M., Schaeffer, G., Tasan, M., and Xu, W. **(2003)**. Pathways Database System: an integrated system for biological pathways. *Bioinformatics*, **19**(8), 930–937. 10.1093/bioinformatics/btg113

Kumar, S., Skjaeveland, Å ., Orr, R., Enger, P., Ruden, T., Mevik, B.-H., Burki, F., Botnen, A., and Shalchian-Tabrizi, K. **(2009)**. AIR: A batch-oriented web program package for construction of supermatrices ready for phylogenomic analyses. *BMC Bioinformatics*, **10**(1), 357. 10.1186/1471-2105-10-357

Kumar, S., Krabberød, A. K., Neumann, R. S., Michalickova, K., Zhao, S., Zhang, X., and Shalchian-Tabrizi, K. **(2015)**. BIR Pipeline for Preparation of Phylogenomic Data. *Evol. Bioinform. Online*, **11**, 79–83. 10.4137/EBO.S10189

Kwon, Y., Shigemoto, Y., Kuwana, Y., and Sugawara, H. **(2009)**. Web API for biology with a workflow navigation system. *Nucleic Acids Res.*, **37**(suppl 2,W1), W11–W16. 10.1093/nar/gkp300

Kyoda, K., Tohsato, Y., Ho, K. H. L., and Onami, S. **(2015)**. Biological Dynamics Markup Language (BDML): an open format for representing quantitative biological dynamics data. *Bioinformatics*, **31**(7), 1044–1052. 10.1093/bioinformatics/btu767

Labarga, A., Valentin, F., Anderson, M., and Lopez, R. **(2007)**. Web Services at the European Bioinformatics Institute. *Nucleic Acids Res.*, **35**(suppl 2,W1), W6–W11. 10.1093/nar/gkm291

Lamprecht, A.-L., Naujokat, S., Margaria, T., and Steffen, B. **(2011)**. Semantics-based composition of EMBOSS services. *J. Biomed. Sem.*, **2**(Suppl 1), S5. 10.1186/2041-1480-2-S1-S5

Larkin, M., Blackshields, G., Brown, N., Chenna, R., McGettigan, P., McWilliam, H., Valentin, F., Wallace, I., Wilm, A., Lopez, R., Thompson, J., Gibson, T., and Higgins, D. **(2007)**. Clustal W and Clustal X version 2.0. *Bioinformatics*, **23**(21), 2947–2948. 10.1093/bioinformatics/btm404

Lee, E., Helt, G., Reese, J., Munoz-Torres, M., Childers, C., Buels, R., Stein, L., Holmes, I., Elsik, C., and Lewis, S. **(2013)**. Web Apollo: a web-based genomic annotation editing platform. *Genome Biol.*, **14**(8), R93. 10.1186/gb-2013-14-8-r93

Lengauer, T. **(1999)**. The Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB'99). *Bioinformatics*, **15**(10), 775. 10.1093/bioinformatics/15.10.775

Lengauer, T. **(2002)**. Editorial. *Bioinformatics*, **18**(suppl 2), S1. 10.1093/bioinformatics/18.suppl_2.S1

Lex, A., Streit, M., Schulz, H., Partl, C., Schmalstieg, D., Park, P. J., and Gehlenborg, N. **(2012)**. StratomeX: visual analysis of Large-Scale heterogeneous genomics data for cancer subtype characterization. In *Computer Graphics Forum (EuroVis '12)*, **31**, 1175–1184. Wiley Online Library. 10.1111/j.1467-8659.2012.03110.x

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., and Subgroup, . G. P. D. P. **(2009)**. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**(16), 2078–2079. 10.1093/bioinformatics/btp352

Li, J.-W., Robison, K., Martin, M., Sjödin, A., Usadel, B., Young, M., Olivares, E. C., and Bolser, D. M. **(2012a)**. The SEQanswers wiki: a wiki database of tools for high-throughput sequencing analysis. *Nucleic Acids Res.*, **40**(suppl 1,D1), D1313–D1317. 10.1093/nar/gkr1058

Li, J.-W., Schmieder, R., Ward, R. M., Delenick, J., Olivares, E. C., and Mittelman, D. **(2012b)**. SEQanswers: an open access community for collaboratively decoding genomes. *Bioinformatics*, **28**(9), 1272–1273. 10.1093/bioinformatics/bts128

Li, W., Cowley, A., Uludag, M., Gur, T., McWilliam, H., Squizzato, S., Park, Y. M., Buso, N., and Lopez, R. **(2015)**. The EMBL-EBI bioinformatics web and programmatic tools framework. *Nucleic Acids Res.*, **43**(suppl 2,W1), W580–W584. 10.1093/nar/gkv279

Lister, R., O'Malley, R. C., Tonti-Filippini, J., Gregory, B. D., Berry, C. C., Millar, A. H., and Ecker, J. R. **(2008)**. Highly Integrated Single-Base Resolution Maps of the Epigenome in Arabidopsis. *Cell*, **133**(3), 523–536. 10.1016/j.cell.2008.03.029

Logan, D. W., Sandal, M., Gardner, P. P., Manske, M., and Bateman, A. **(2010)**. Ten Simple Rules for Editing Wikipedia. *PLoS Comput. Biol.*, **6**(9), e1000941. 10.1371/journal.pcbi.1000941

Lopes, C. T., Franz, M., Kazi, F., Donaldson, S. L., Morris, Q., and Bader, G. D. **(2010)**. Cytoscape Web: an interactive web-based network browser. *Bioinformatics*, **26**(18), 2347–2348. 10.1093/bioinformatics/btq430

Lopez, R., Duggan, K., Harte, N., and Kibria, A. **(2003)**. Public services from the European Bioinformatics Institute. *Brief. Bioinform.*, **4**(4), 332–340. 10.1093/bib/4.4.332

Lord, E., Leclercq, M., Boc, A., Diallo, A. B., and Makarenkov, V. **(2012)**. Armadillo 1.1: An Original Workflow Platform for Designing and Conducting Phylogenetic Analysis and Simulations. *PLoS ONE*, **7**(1), e29903. 10.1371/journal.pone.0029903

Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. **(2011)**. ViennaRNA Package 2.0. *Algorithms Mol. Biol.*, **6**(1). 10.1186/1748-7188-6-26

Lu, G. and Moriyama, E. N. **(2004)**. Vector NTI, a balanced all-in-one sequence analysis suite. *Brief. Bioinform.*, **5**(4), 378–388. 10.1093/bib/5.4.378

Mangalam, H. **(2002)**. The Bio* toolkits—a brief overview. *Brief. Bioinform.*, **3**(3), 296–302. 10.1093/bib/3.3.296

Martens, L., Orchard, S., Apweiler, R., and Hermjakob, H. **(2007)**. Human Proteome Organization Proteomics Standards Initiative: Data Standardization, a View on Developments and Policy. *Mol. Cell. Proteomics*, **6**(9), 1666–1667.

Marx, V. **(2013)**. Biology: The big challenges of big data. *Nature*, **498**(7453), 255–260. 10.1038/498255a

McGinnis, S. and Madden, T. L. **(2004)**. BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Res.*, **32**(suppl 2,W1), W20–W25. 10.1093/nar/gkh435

McWilliam, H., Valentin, F., Goujon, M., Li, W., Narayanasamy, M., Martin, J., Miyar, T., and Lopez, R. **(2009)**. Web services at the European Bioinformatics Institute-2009. *Nucleic Acids Res.*, **37**(suppl 2,W1), W6–W10. 10.1093/nar/gkp302

McWilliam, H., Li, W., Uludag, M., Squizzato, S., Park, Y. M., Buso, N., Cowley, A. P., and Lopez, R. **(2013)**. Analysis Tool Web Services from the EMBL-EBI. *Nucleic Acids Res.*, **41**(suppl 2,W1), W597–W600. 10.1093/nar/gkt376

Medina, I., Salavert, F., Sanchez, R., de Maria, A., Alonso, R., Escobar, P., Bleda, M., and Dopazo, J. **(2013)**. Genome Maps, a new generation genome browser. *Nucleic Acids Res.*, **41**(suppl 2,W1), W41–W46. 10.1093/nar/gkt530

Ménager, H., Kalaš, M., Grosjean, M., and Ison, J. **(2015a)**. The EDAM Ontology [v1; not peer reviewed]. *F1000Research*, **4**(ISCB Comm. J.), 227. Poster. 10.7490/f1000research.1000204.1

Ménager, H., Kalaš, M., Grosjean, M., and Ison, J. **(2015b)**. The EDAM Ontology [v1; not peer reviewed]. *F1000Research*, **4**(ISCB Comm. J.), 359. Slides. 10.7490/f1000research.1110110.1

Ménager, H., Kalaš, M., Rapacki, K., and Ison, J. **(2015c)**. Using registries to integrate bioinformatics tools and services into workbench environments. *Int. J. Softw. Tools Technol. Transfer*. 10.1007/s10009-015-0392-z

Mesirov, J. P. **(2010)**. Accessible Reproducible Research. *Science*, **327**(5964), 415–416. 10.1126/science.1179653

Miller, C. A., Anthony, J., Meyer, M. M., and Marth, G. **(2013)**. Scribl: an HTML5 Canvas-based graphics library for visualizing genomic data over the web. *Bioinformatics*, **29**(3), 381–383. 10.1093/bioinformatics/bts677

Möller, S., Leser, U., Fleischmann, W., and Apweiler, R. **(1999)**. EDITtoTrEMBL: a distributed approach to high-quality automated protein sequence annotation. *Bioinformatics*, **15**(3), 219–227. 10.1093/bioinformatics/15.3.219

Möller, S., Krabbenhöft, H., Tille, A., Paleino, D., Williams, A., Wolstencroft, K., Goble, C., Holland, R., Belhachemi, D., and Plessy, C. **(2010)**. Community-driven computational biology with Debian Linux. *BMC Bioinformatics*, **11**(Suppl 12), S5. 10.1186/1471-2105-11-S12-S5

Möller, S., Afgan, E., Banck, M., Cock, P., Kalaš, M., Kaján, L., Prins, P., Quinn, J., Sallou, O., Strozzi, F., Seemann, T., Tille, A., Guimera, R. V., Katayama, T., and Chapman, B. **(2013)**. Sprints, Hackathons and Codefests as community gluons in computational biology. *EMBnet.J.*, **19**(B), 40–42. 10.14806/ej.19.B.726

Möller, S., Afgan, E., Banck, M., Bonnal, R. J. P., Booth, T., Chilton, J., Cock, P. J. A., Gumbel, M., Harris, N., Holland, R., Kalaš, M., Kaján, L., Kibukawa, E., Powell, D. R., Prins, P., Quinn, J., Sallou, O., Strozzi, F., Seemann, T., Sloggett, C., Soiland-Reyes, S., Steinbiss, S., Tille, A., Travis, A. J., Guimera, R. V., Katayama, T., and Chapman, B. **(2014)**. Community-driven development for computational biology at Sprints, Hackathons and Codefests. *BMC Bioinformatics*, **15**(Suppl 14), S7. 10.1186/1471-2105-15-S14-S7

Morales, H. F. and Giovambattista, G. **(2013)**. BioSmalltalk: a pure object systemand library for bioinformatics. *Bioinformatics*, **29**(18), 2355–2356. 10.1093/bioinformatics/btt398

Moreau, Y. and Beerenwinkel, N. **(2015)**. ISMB/ECCB 2015. *Bioinformatics*, **31**(12), i1–i2. 10.1093/bioinformatics/btv303

Murdock, I. A. **(1994)**. The Debian Linux Manifesto. Included in release of Debian version 0.91.

Murray-Rust, P., Rzepa, H. S., and Wright, M. **(2001)**. Development of chemical markup language (CML) as a system for handling complex chemical content. *New J. Chem.*, **25**, 618–634. 10.1039/B008780G

NCBI Resource Coordinators **(2014)**. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **42**(suppl 1,D1), D7–D17. 10.1093/nar/gkt1146

NCBI Resource Coordinators **(2015)**. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **43**(suppl 1,D1), D6–D17. 10.1093/nar/gku1130

Needleman, S. B. and Wunsch, C. D. **(1970)**. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**(3), 443 – 453. 10.1016/0022-2836(70)90057-4

Néron, B., Ménager, H., Maufrais, C., Joly, N., Maupetit, J., Letort, S., Carrere, S., Tuffery, P., and Letondal, C. **(2009)**. Mobyle: a new full web bioinformatics framework. *Bioinformatics*, **25**(22), 3005–3011. 10.1093/bioinformatics/btp493

Noy, N. F., Shah, N. H., Whetzel, P. L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D. L., Storey, M.-A., Chute, C. G., and Musen, M. A. **(2009)**. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Res.*, **37**(suppl 2,W1), W170–W173. 10.1093/nar/gkp440

Nygård, S. and Jonassen, I. **(2014)**. Norwegian Bioinformatics Platform. *NBS Nytt*, **38**(2), 32–35.

Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., and Li, P. **(2004)**. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, **20**(17), 3045–3054. 10.1093/bioinformatics/bth361

Okonechnikov, K., Golosova, O., Fursov, M., and the UGENE team **(2012)**. Unipro UGENE: a unified bioinformatics toolkit. *Bioinformatics*, **28**(8), 1166–1167. 10.1093/bioinformatics/bts091

Orchard, S., Salwinski, L., Kerrien, S., Montecchi-Palazzi, L., Oesterheld, M., Stumpflen, V., Ceol, A., Chatr-aryamontri, A., Armstrong, J., Woollard, P., Salama, J. J., Moore, S., Wojcik, J., Bader, G. D., Vidal, M., Cusick, M. E., Gerstein, M., Gavin, A.-C., Superti-Furga, G., Greenblatt, J., Bader, J., Uetz, P., Tyers, M., Legrain, P., Fields, S., Mulder, N., Gilson, M., Niepmann, M., Burgoon, L., Rivas, J. D., Prieto, C., Perreau, V. M., Hogue, C., Mewes, H.-W., Apweiler, R., Xenarios, I., Eisenberg, D., Cesareni, G., and Hermjakob, H. **(2007)**. The minimum information required for reporting a molecular interaction experiment (MIMIx). *Nat. Biotechnol.*, **25**(8), 894–898. 10.1038/nbt1324

Paradis, E., Claude, J., and Strimmer, K. **(2004)**. APE: Analyses of Phylogenetics and Evolution in R language. *Bioinformatics*, **20**(2), 289–290. 10.1093/bioinformatics/btg412

Parker, M. **(1993)**. Biological data access through Gopher. *Trends Biochem. Sci.*, **18**(12), 485–486. 10.1016/S0968-0004(10)80001-5

Parnell, L. D., Lindenbaum, P., Shameer, K., Dall'Olio, G. M., Swan, D. C., Jensen, L. J., Cockell, S. J., Pedersen, B. S., Mangan, M. E., Miller, C. A., and Albert, I. **(2011)**. BioStar: An Online Question & Answer Resource for the Bioinformatics Community. *PLoS Comput. Biol.*, **7**(10), e1002216. 10.1371/journal.pcbi.1002216

Pérez, F. and Granger, B. **(2007)**. IPython: A System for Interactive Scientific Computing. *Comput. Sci. Eng.*, **9**(3), 21–29. 10.1109/MCSE.2007.53

Perens, B. **(1997)**. Debian's "Social Contract" with the Free Software Community. *debian-announce@lists.debian.org*, (msg00017). Re-published as Debian Social Contract, Version 1.0.

Perens, B. **(1999)**. The Open Source Definition. In *Open Sources: Voices from the Open Source Revolution*. O'Reilly.

Pettifer, S., Thorne, D., McDermott, P., Attwood, T., Baran, J., Bryne, J. C., Hupponen, T., Mowbray, D., and Vriend, G. **(2009)**. An active registry for bioinformatics web services. *Bioinformatics*, **25**, 2090–2091. 10.1093/bioinformatics/btp329

Pettifer, S., Ison, J., Kalaš, M., Thorne, D., McDermott, P., Jonassen, I., Liaquat, A., Fernández, J. M., Rodriguez, J. M., Partners, I., Pisano, D. G., Blanchet, C., Uludag, M., Rice, P., Bartaševičiūtė (now Karosiene), E., Rapacki, K., Hekkelman, M., Sand, O., Stockinger, H., Clegg, A. B., Bongcam-Rudloff, E., Salzemann, J., Breton, V., Attwood, T. K., Cameron, G., and Vriend, G. **(2010)**. The EMBRACE web service collection. *Nucleic Acids Res.*, **38**(suppl 2,W1), W683–W688. 10.1093/nar/gkq297

Pillai, S., Silventoinen, V., Kallio, K., Senger, M., Sobhany, S., Tate, J., Velankar, S., Golovin, A., Henrick, K., Rice, P., Stoehr, P., and Lopez, R. **(2005)**. SOAP-based services provided by the European Bioinformatics Institute. *Nucleic Acids Res.*, **33**(suppl 2,W1), W25–W28. 10.1093/nar/gki491

Pitt, W. R., Williams, M. A., Steven, M., Sweeney, B., Bleasby, A. J., and Moss, D. S. **(2001)**. The bioinformatics template library—generic components for biocomputing. *Bioinformatics*, **17**(8), 729–737. 10.1093/bioinformatics/17.8.729

Plieskatt, J., Rinaldi, G., Brindley, P. J., Jia, X., Potriquet, J., Bethony, J., and Mulvenna, J. **(2014)**. Bioclojure: a functional library for the manipulation of biological sequences. *Bioinformatics*, **30**(17), 2537–2539. 10.1093/bioinformatics/btu311

Pocock, M., Down, T., and Hubbard, T. **(2000)**. BioJava: Open Source Components for Bioinformatics. *ACM SIGBIO Newsl.*, **20**(2), 10–12. 10.1145/360262.360266

Porebski, B. T., Ho, B. K., and Buckle, A. M. **(2013)**. Interactive visualization tools for the structural biologist. *J. Appl. Crystallogr.*, **46**(Pt 5), 1518–1520. 10.1107/S0021889813017858

Pritchard, L., White, J. A., Birch, P. R., and Toth, I. K. **(2006)**. GenomeDiagram: a python package for the visualization of large-scale genomic data. *Bioinformatics*, **22**(5), 616–617. 10.1093/bioinformatics/btk021

Prlić, A., Down, T., Kulesha, E., Finn, R., Kähäri, A., and Hubbard, T. **(2007)**. Integrating sequence and structural biology with DAS. *BMC Bioinformatics*, **8**(1), 333. 10.1186/1471-2105-8-333

Prlić, A., Yates, A., Bliven, S. E., Rose, P.W., Jacobsen, J., Troshin, P. V., Chapman, M., Gao, J., Koh, C. H., Foisy, S., Holland, R., Rimša, G., Heuer, M. L., Brandsätter-Müller, H., Bourne, P. E., and Willis, S. **(2012)**. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, **28**(20), 2693–2695. 10.1093/bioinformatics/bts494

Pronk, S., Páll, S., Schulz, R., Larsson, P., Bjelkmar, P., Apostolov, R., Shirts, M. R., Smith, J. C., Kasson, P. M., van der Spoel, D., Hess, B., and Lindahl, E. **(2013)**. GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics*, **29**(7), 845–854. 10.1093/bioinformatics/btt055

Prosdocimi, F., Chisham, B., Pontelli, E., Thompson, J. D., and Stoltzfus, A. **(2009)**. Initial implementation of a Comparative Data Analysis Ontology. *Evolutionary Bioinformatics*, (5), 47–66. 10.4137/EBO.S2320

Rayner, T., Rocca-Serra, P., Spellman, P., Causton, H., Farne, A., Holloway, E., Irizarry, R., Liu, J., Maier, D., Miller, M., Petersen, K., Quackenbush, J., Sherlock, G., Stoeckert, C., White, J., Whetzel, P., Wymore, F., Parkinson, H., Sarkans, U., Ball, C., and Brazma, A. **(2006)**. A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB. *BMC Bioinformatics*, **7**(1), 489. 10.1186/1471-2105-7-489

Reich, M., Liefeld, T., Gould, J., Lerner, J., Tamayo, P., and Mesirov, J. P. **(2006)**. GenePattern 2.0. *Nat. Genet.*, **38**(5), 500–501. 10.1038/ng0506-500

Reich, M., Liefeld, T., Ocana, M., Jang, D., Bistline, J., Robinson, J., Carr, P., Hill, B., McLaughlin, J., Pochet, N., Borges-Rivera, D., Tabor, T., Thorvaldsdóttir, H., Regev, A., and Mesirov, J. P. **(2013)**. GenomeSpace: An environment for frictionless bioinformatics. *F1000Posters*, **4**, 804. Poster.

Rice, C. M., Fuchs, R., Higgins, D. G., Stoehr, P. J., and Cameron, G. N. **(1993)**. The EMBL data library. *Nucleic Acids Res.*, **21**(13), 2967–2971. 10.1093/nar/21.13.2967

Rice, P. **(1998)**. EMBOSS: A european software suite. *EMBnet News*, **5**(2), 6–7.

Rice, P., Lopez, R., Doelz, R., and Leunissen, J. **(1995)**. Program developement—EGCG 8.0. *EMBnet News*, **2**(2), 5–7.

Rice, P., Lopez, R., Doelz, R., and Leunissen, J. **(1996)**. EGCG 8.1 Released. *EMBnet News*, **3**(1), 2–4.

Rice, P., Longden, I., and Bleasby, A. **(2000)**. EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet.*, **16**(6), 276–277. 10.1016/S0168-9525(00)02024-2

Robinson, J. T., Thorvaldsdóttir, H., Winckler, W., Guttman, M., Lander, E. S., Getz, G., and Mesirov, J. P. **(2011)**. Integrative genomics viewer. *Nat. Biotechnol.*, **29**(1), 24–26. 10.1038/nbt.1754

Rocca-Serra, P., Brandizi, M., Maguire, E., Sklyar, N., Taylor, C., Begley, K., Field, D., Harris, S., Hide, W., Hofmann, O., Neumann, S., Sterk, P., Tong, W., and Sansone, S.-A. **(2010)**. ISA software suite: supporting standards-compliant experimental annotation and enabling curation at the community level. *Bioinformatics*, **26**(18), 2354–2356. 10.1093/bioinformatics/btq415

Rodriguez, R., Chinea, G., Lopez, N., Pons, T., and Vriend, G. **(1998)**. Homology modeling, model and software evaluation: three related resources. *Bioinformatics*, **14**(6), 523–528. 10.1093/bioinformatics/14.6.523

Rodriguez-Tomé **(1998)**. The BioCatalog. *Bioinformatics*, **14**(5), 469–470. 10.1093/bioinformatics/14.5.469

Rost, B., Yachdav, G., and Liu, J. **(2004)**. The PredictProtein server. *Nucleic Acids Res.*, **32**(suppl 2,W1), W321–W326. 10.1093/nar/gkh377

Rutherford, K., Parkhill, J., Crook, J., Horsnell, T., Rice, P., Rajandream, M.-A., and Barrell, B. **(2000)**. Artemis: sequence visualization and annotation. *Bioinformatics*, **16**(10), 944–945. 10.1093/bioinformatics/16.10.944

Saito, T. and Rehmsmeier, M. **(2015)**. A fast microRNA target prediction tool that provides a single-entry interface to multiple algorithms [v1; not peer reviewed]. *F1000Research*, **4**(ISCB Comm. J.), 529. Poster. 10.7490/f1000research.1110273.1

Sandve, G. K., Gundersen, S., Rydbeck, H., Glad, I. K., Holden, L., Holden, M., Liestøl, K., Clancy, T., Ferkingstad, E., Johansen, M., Nygaard, V., Tøstesen, E., Frigessi, A., and Hovig, E. **(2010)**. The Genomic HyperBrowser: inferential genomics at the sequence level. *Genome Biol.*, **11**(12), R121. 10.1186/gb-2010-11-12-r121

Sandve, G. K., Gundersen, S., Johansen, M., Glad, I. K., Gunathasan, K., Holden, L., Holden, M., Liestøl, K., Nygård, S., Nygaard, V., Paulsen, J., Rydbeck, H., Trengereid, K., Clancy, T., Drabløs, F., Ferkingstad, E., Kalaš, M., Lien, T., Rye, M. B., Frigessi, A., and Hovig, E. **(2013)**. The Genomic HyperBrowser: an analysis web server for genome-scale data. *Nucleic Acids Res.*, **41**(suppl 2,W1), W133–W141. 10.1093/nar/gkt342

Saqi, M. A., Wild, D. L., and Hartshorn, M. J. **(1999)**. Protein Analyst–a distributed object environment for protein sequence and structure analysis. *Bioinformatics*, **15**(6), 521–522. 10.1093/bioinformatics/15.6.521

Sayle, R. A. and Milner-White, E. **(1995)**. RASMOL: biomolecular graphics for all. *Trends Biochem. Sci.*, **20**(9), 374–376. 10.1016/S0968-0004(00)89080-5

Schuler, G., Altschul, S., and Lipman, D. **(1991)**. A workbench for multiple alignment construction and analysis. *Proteins*, **9**(3), 180–190. 10.1002/prot.340090304

Schuler, G. D., Epstein, J. A., Ohkawa, H., and Kans, J. A. **(1996)**. Entrez: Molecular biology database and retrieval system. In *Computer Methods for Macromolecular Sequence Analysis*. *Methods Enzymol.*, **266**, 141–162. Academic Press. 10.1016/S0076-6879(96)66012-1

Schwab, M., Karrenbach, N., and Claerbout, J. **(2000)**. Making scientific computations reproducible. *Comput. Sci. Eng.*, **2**(6), 61–67. 10.1109/5992.881708

Seibel, P. N., Krüger, J., Hartmeier, S., Schwarzer, K., Löwenthal, K., Mersch, H., Dandekar, T., and Giegerich, R. **(2006)**. XML schemas for common bioinformatic data types and their application in workflow systems. *BMC Bioinformatics*, **7**(1). 10.1186/1471-2105-7-490

Senger, M. **(1999)**. AppLab — CORBA-Java based Application Wrapper. *EMBL Outstation – European Bioinformatics Institute, UK*. Reference manual.

Senger, M., Glatting, K.-H., Ritter, O., and Suhai, S. **(1995)**. X-HUSAR, an X-based graphical interface for the analysis of genomic sequences . *Comput. Methods Programs Biomed.*, **46**(2), 131–141. 10.1016/0169-2607(94)01610-R

Senger, M., Rice, P., and Oinn, T. **(2003)**. Soaplab - a unified Sesame door to analysis tools. In Cox, S. J. (*Ed.*). *Proceedings of the UK e-Science All Hands Meeting 2003*, 509–513. EPSRC.

Senger, M., Rice, P., Bleasby, A., Oinn, T., and Uludag, M. **(2008)**. Soaplab2: more reliable Sesame door to bioinformatics programs. In *The 9th annual Bioinformatics Open Source Conference*.

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. **(2003)**. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res.*, **13**(11), 2498–2504. 10.1101/gr.1239303

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J. D., and Higgins, D. G. **(2011)**. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.*, **7**(1). 10.1038/msb2011.75

Skinner, M. E., Uzilov, A. V., Stein, L. D., Mungall, C. J., and Holmes, I. H. **(2009)**. JBrowse: a next-generation genome browser. *Genome Res.*, **19**(9), 1630–1638. 10.1101/gr.094607.109

Smedley, D., Haider, S., Durinck, S., Pandini, L., Provero, P., Allen, J., Arnaiz, O., Awedh, M. H., Baldock, R., Barbiera, G., Bardou, P., Beck, T., Blake, A., Bonierbale, M., Brookes, A. J., Bucci, G., Buetti, I., Burge, S., Cabau, C., Carlson, J. W., Chelala, C., Chrysostomou, C., Cittaro, D., Collin, O., Cordova, R., Cutts, R. J., Dassi, E., Genova, A. D., Djari, A., Esposito, A., Estrella, H., Eyras, E., Fernandez-Banet, J., Forbes, S., Free, R. C., Fujisawa, T., Gadaleta, E., Garcia-Manteiga, J. M., Goodstein, D., Gray, K., Guerra-Assunção, J. A., Haggarty, B., Han, D.-J., Han, B. W., Harris, T., Harshbarger, J., Hastings, R. K., Hayes, R. D., Hoede, C., Hu, S., Hu, Z.-L., Hutchins, L., Kan, Z., Kawaji, H., Keliet, A., Kerhornou, A., Kim, S., Kinsella, R., Klopp, C., Kong, L., Lawson, D., Lazarevic, D., Lee, J.-H., Letellier, T., Li, C.-Y., Lio, P., Liu, C.-J., Luo, J., Maass, A., Mariette, J., Maurel, T., Merella, S., Mohamed, A. M., Moreews, F., Nabihoudine, I., Ndegwa, N., Noirot, C., Perez-Llamas, C., Primig, M., Quattrone, A., Quesneville, H., Rambaldi, D., Reecy, J., Riba, M., Rosanoff, S., Saddiq, A. A., Salas, E., Sallou, O., Shepherd, R., Simon, R., Sperling, L., Spooner, W., Staines, D. M., Steinbach, D., Stone, K., Stupka, E., Teague, J. W., Dayem Ullah, A. Z., Wang, J., Ware, D., Wong-Erasmus, M., Youens-Clark, K., Zadissa, A., Zhang, S.-J., and Kasprzyk, A. **(2015)**. The BioMart community portal: an innovative alternative to large, centralized data repositories. *Nucleic Acids Res.*, **43**(suppl 2,W1), W589–W598. 10.1093/nar/gkv350

Smith, S.W., Overbeek, R., Woese, C. R., Gilbert, W., and Gillevet, P. **(1994)**. The genetic data environment an expandable GUI for multiple sequence analysis. *Comput. Appl. Biosci.*, **10**(6), 671–675. 10.1093/bioinformatics/10.6.671

Smith, T. and Waterman, M. **(1981)**. Identification of common molecular subsequences. *J. Mol. Biol.*, **147**(1), 195 – 197. 10.1016/0022-2836(81)90087-5

Spellman, P. T., Miller, M., Stewart, J., Troup, C., Sarkans, U., Chervitz, S., Bernhart, D., Sherlock, G., Ball, C., Lepage, M., Swiatek, M., Marks, W. L., Goncalves, J., Markel, S., Iordan, D., Shojatalab, M., Pizarro, A., White, J., Hubley, R., Deutsch, E., Senger, M., Aronow, B. J., Robinson, A., Bassett, D., Jr, and Brazma, A. **(2002)**. Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol.*, **3**(9). 10.1186/gb-2002-3-9-research0046

Squizzato, S., Park, Y. M., Buso, N., Gur, T., Cowley, A., Li, W., Uludag, M., Pundir, S., Cham, J. A., McWilliam, H., and Lopez, R. **(2015)**. The EBI Search engine: providing search and retrieval functionality for biological data from EMBL-EBI. *Nucleic Acids Res.*, **43**(suppl 2,W1), W585–W588. 10.1093/nar/gkv316

Staden, R. **(1977)**. Sequence data handling by computer. *Nucleic Acids Res.*, **4**(11), 4037–4052. 10.1093/nar/4.11.4037

Staden, R. **(1978)**. Further procedures for sequence analysis by computer. *Nucleic Acids Res.*, **5**(3), 1013–1016. 10.1093/nar/5.3.1013

Staden, R. **(1979)**. A strategy of DNA sequencing employing computer programs. *Nucleic Acids Res.*, **6**(7), 2601–2610. 10.1093/nar/6.7.2601

Staden, R. **(1982)**. An interactive graphics program for comparing and aligning nucleic acid and amino acid sequences. *Nucleic Acids Res.*, **10**(9), 2951–2961. 10.1093/nar/10.9.2951

Staden, R. **(1984)**. Graphic methods to determine the function of nucleic acid sequences. *Nucleic Acids Res.*, **12**(1Part2), 521–538. 10.1093/nar/12.1Part2.521

Staden, R. **(1986)**. The current status and portability of our sequence handling software. *Nucleic Acids Res.*, **14**(1), 217–231. 10.1093/nar/14.1.217

Staden, R. **(1990)**. An improved sequence handling package that runs on the Apple Macintosh. *Comput. Appl. Biosci.*, **6**(4), 387–393. 10.1093/bioinformatics/6.4.387

Staden, R. **(1996)**. The Staden Sequence Analysis Package. *Mol. Biotechnol.*, **5**(3), 233–241. 10.1007/BF02900361

Staden, R., Beal, K. F., and Bonfield, J. K. **(1999)**. The Staden Package, 1998. In Misener, S., Krawetz, S. A. (*Eds.*). *Bioinformatics Methods and Protocols. Methods Mol. Biol.*, **132**, 115–130. Humana Press. 10.1385/1-59259-192-2:115

Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., Fuellen, G., Gilbert, J. G., Korf, I., Lapp, H., Lehväslaiho, H., Matsalla, C., Mungall, C. J., Osborne, B. I., Pocock, M. R., Schattner, P., Senger, M., Stein, L. D., Stupka, E., Wilkinson, M. D., and Birney, E. **(2002)**. The Bioperl Toolkit: Perl Modules for the Life Sciences. *Genome Res.*, **12**(10), 1611–1618. 10.1101/gr.361602

Stallman, R. M. **(1986)**. What is the Free Software Foundation? *Gnu's Bulletin*, **1**(1), 8–9.

Stavrum, A. K., Petersen, K., Jonassen, I., and Dysvik, B. **(2008)**. Analysis of Gene-Expression Data Using J-Express. In *Curr. Protoc. Bioinformatics*, (21), 7.3.1–7.3.25. John Wiley & Sons, Inc. 10.1002/0471250953.bi0703s21

Stavrum, A.-K., Heiland, I., Schuster, S., Puntervoll, P., and Ziegler, M. **(2013)**. Model of Tryptophan Metabolism, Readily Scalable Using Tissue-specific Gene Expression Data. *J. Biol. Chem.*, **288**(48), 34555–34566. 10.1074/jbc.M113.474908

Stein, L. D. **(2002)**. Creating a bioinformatics nation. *Nature*, **417**(6885), 119–120. 10.1038/417119a

Stein, L. D., Mungall, C., Shu, S., Caudy, M., Mangone, M., Day, A., Nickerson, E., Stajich, J. E., Harris, T. W., Arva, A., and Lewis, S. **(2002)**. The generic genome browser: A building block for a model organism system database. *Genome Res.*, **12**(10), 1599–1610. 10.1101/gr.403602

Steinbiss, S., Gremme, G., Schrfer, C., Mader, M., and Kurtz, S. **(2009)**. AnnotationSketch: a genome annotation drawing library. *Bioinformatics*, **25**(4), 533–534. 10.1093/bioinformatics/btn657

Stockwell, P. A. **(1988)**. HOMED: a homologous sequence editor. *Trends Biochem. Sci.*, **13**(8), 322 – 324. 10.1016/0968-0004(88)90130-2

Stockwell, P. A. and Petersen, G. B. **(1987)**. HOMED: a homologous sequence editor. *Comput. Appl. Biosci.*, **3**(1), 37–43. 10.1093/bioinformatics/3.1.37

Streit, M., Lex, A., Kalkusch, M., Zatloukal, K., and Schmalstieg, D. **(2009)**. Caleydo: connecting pathways and gene expression. *Bioinformatics*, **25**(20), 2760–2761. 10.1093/bioinformatics/btp432

Subramanian, S., Puntervoll, P., and Sztromwasser, P. **(2010)**. Optimizing the Data-Traffic of Centrally Coordinated Scientific Workflow Systems. In *IEEE 19th International Conference on Web Services (ICWS)*, 685–688. 10.1109/ICWS.2010.71

Subramanian, S., Sztromwasser, P., Petersen, K., and Puntervoll, P. **(2012)**. Direct data transfer between SOAP web services in orchestration. In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services (IIWAS '12)*, 91–100, ACM. 10.1145/2428736.2428753

Subramanian, S., Sztromwasser, P., Puntervoll, P., and Petersen, K. **(2013)**. Pipelined Data-flow Delegated Orchestration for Data-Intensive eScience Workflows. *Int. J. Web Inform. Sys.*, **9**(3), 204–218. 10.1108/IJWIS-05-2013-0012

Sugawara, H. and Miyazaki, S. **(2003)**. Biological SOAP servers and web services provided by the public sequence data bank. *Nucleic Acids Res.*, **31**(13), 3836–3839. 10.1093/nar/gkg558

Sztromwasser, P. **(2014)**. Throughput and robustness of bioinformatics pipelines for genome-scale data analysis. *The University of Bergen, Norway*. Ph.D. dissertation. 1956/7906

Sztromwasser, P., Puntervoll, P., and Petersen, K. **(2011)**. Data partitioning enables the use of standard SOAP Web Services in genome-scale workflows. *J. Integr. Bioinform.*, **8**(2), 163. 10.2390/biecoll-jib-2011-163

Sztromwasser, P., Petersen, K., and Jonassen, I. **(In preparation)**. Sensitivity screening reveals influential parameters of a variant calling pipeline. *Manuscript in preparation*.

Taubert, J., Hassani-Pak, K., Castells-Brooke, N., and Rawlings, C. J. **(2013)**. Ondex Web: web-based visualization and exploration of heterogeneous biological networks. *Bioinformatics*. 10.1093/bioinformatics/btt740

Taylor, C. F., Field, D., Sansone, S. A., Aerts, J., Apweiler, R., Ashburner, M., Ball, C. A., Binz, P. A., Bogue, M., Booth, T., Brazma, A., Brinkman, R. R., Michael Clark, A., Deutsch, E. W., Fiehn, O., Fostel, J., Ghazal, P., Gibson, F., Gray, T., Grimes, G., Hancock, J. M., Hardy, N. W., Hermjakob, H., Julian, R. K., Kane, M., Kettner, C., Kinsinger, C., Kolker, E., Kuiper, M., Le Novère, N., Leebens-Mack, J., Lewis, S. E., Lord, P., Mallon, A. M., Marthandan, N., Masuya, H., McNally, R., Mehrle, A., Morrison, N., Orchard, S., Quackenbush, J., Reecy, J. M., Robertson, D. G., Rocca-Serra, P., Rodriguez, H., Rosenfelder, H., Santoyo-Lopez, J., Scheuermann, R. H., Schober, D., Smith, B., Snape, J., Stoeckert, C. J., Tipton, K., Sterk, P., Untergasser, A., Vandesompele, J., and Wiemann, S. **(2008)**. Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nat. Biotechnol.*, **26**, 889–896. 10.1038/nbt.1411

Taylor, W. R. **(1988)**. A flexible method to align large numbers of biological sequences. *J. Mol. Evol.*, **28**(1-2), 161–9. 10.1007/BF02143508

Thompson, J. D., Higgins, D. G., and Gibson, T. J. **(1994)**. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**(22), 4673–4680. 10.1093/nar/22.22.4673

Thompson, J. D., Gibson, T. J., Plewniak, F., Jeanmougin, F., and Higgins, D. G. **(1997)**. The CLUSTAL X Windows Interface: Flexible Strategies for Multiple Sequence Alignment Aided by Quality Analysis Tools. *Nucleic Acids Res.*, **25**(24), 4876–4882. 10.1093/nar/25.24.4876

Thornton, K. **(2003)**. libsequence: a C++ class library for evolutionary genetic analysis. *Bioinformatics*, **19**(17), 2325–2327. 10.1093/bioinformatics/btg316

Thorvaldsdóttir, H., Robinson, J. T., and Mesirov, J. P. **(2013)**. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Brief. Bioinform.*, **14**(2), 178–192. 10.1093/bib/bbs017

Troshin, P. V., Procter, J. B., and Barton, G. J. **(2011)**. Java bioinformatics analysis web services for multiple sequence alignment—JABAWS:MSA. *Bioinformatics*, **27**(14), 2001–2002. 10.1093/bioinformatics/btr304

van der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A. E., and Berendsen, H. J. C. **(2005)**. GROMACS: Fast, flexible, and free. *J. Comput. Chem.*, **26**(16), 1701–1718. 10.1002/jcc.20291

Vetter, R. J., Spell, C., and Ward, C. **(1994)**. Mosaic and the World Wide Web. *Computer*, **27**(10), 49–57. 10.1109/2.318591

Vos, R., Caravas, J., Hartmann, K., Jensen, M., and Miller, C. **(2011)**. BIO::Phylo-phyloinformatic analysis using perl. *BMC Bioinformatics*, **12**(1), 63. 10.1186/1471-2105-12-63

Vos, R. A., Balhoff, J. P., Caravas, J. A., Holder, M. T., Lapp, H., Maddison, W. P., Midford, P. E., Priyam, A., Sukumaran, J., Xia, X., and Stoltzfus, A. **(2012)**. NeXML: Rich, Extensible, and Verifiable Representation of Comparative Data and Metadata. *Syst. Biol.*, **61**(4), 675–689. 10.1093/sysbio/sys025

Vriend, G. **(1990)**. WHAT IF: A molecular modeling and drug design program. *J. Mol. Graph.*, **8**(1), 52–56. 10.1016/0263-7855(90)80070-V

Wang, J. and Mu, Q. **(2003)**. Soap-HT-BLAST: high throughput BLAST based on Web services. *Bioinformatics*, **19**(14), 1863–1864. 10.1093/bioinformatics/btg244

Waterhouse, A. M., Procter, J. B., Martin, D. M. A., Clamp, M., and Barton, G. J. **(2009)**. Jalview Version 2—a multiple sequence alignment editor and analysis workbench. *Bioinformatics*, **25**(9), 1189–1191. 10.1093/bioinformatics/btp033

Westbrook, J., Ito, N., Nakamura, H., Henrick, K., and Berman, H. M. **(2005)**. PDBML: the representation of archival macromolecular structure data in XML. *Bioinformatics*, **21**(7), 988–992. 10.1093/bioinformatics/bti082

Whetzel, P. L., Parkinson, H., Causton, H. C., Fan, L., Fostel, J., Fragoso, G., Game, L., Heiskanen, M., Morrison, N., Rocca-Serra, P., Sansone, S. A., Taylor, C., White, J., and Stoeckert, C. J. **(2006)**. The MGED Ontology: a resource for semantics-based description of microarray experiments. *Bioinformatics*, **22**(7), 866–873. 10.1093/bioinformatics/btl005

Wilkinson, M., Senger, M., Kawas, E., and The BioMoby Consortium **(2008)**. Interoperability with Moby 1.0—It's better than sharing your toothbrush! *Brief. Bioinform.*, **9**(3), 220–231. 10.1093/bib/bbn003

Wilkinson, M., Vandervalk, B., and McCarthy, L. **(2011)**. The Semantic Automated Discovery and Integration (SADI) Web service Design-Pattern, API and Reference Implementation. *J. Biomed. Sem.*, **2**(1), 8. 10.1186/2041-1480-2-8

Wilkinson, M. D. and Links, M. **(2002)**. BioMOBY: an open source biological web services proposal. *Brief. Bioinform.*, **3**(4), 331–341. 10.1093/bib/3.4.331

Williams, A. J., Harland, L., Groth, P., Pettifer, S., Chichester, C., Willighagen, E. L., Evelo, C. T., Blomberg, N., Ecker, G., Goble, C., and Mons, B. **(2012)**. Open PHACTS: semantic interoperability for drug discovery. *Drug Discov. Today*, **17**(21–22), 1188–1198. 10.1016/j.drudis.2012.05.016

Wodak, S. J., Mietchen, D., Collings, A. M., Russell, R. B., and Bourne, P. E. **(2012)**. Topic Pages: *PLoS Computational Biology Meets Wikipedia*. *PLoS Comput. Biol.*, **8**(3), e1002446. 10.1371/journal.pcbi.1002446

Wolstencroft, K., Alper, P., Hull, D., Wroe, C., Lord, P. W., Stevens, R. D., and Goble, C. A. **(2007)**. The myGrid ontology: bioinformatics service discovery. *Int. J. Bioinform. Res. Appl.*, **3**(3), 303–325. 10.1504/IJBRA.2007.015005

Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop, I., Nenadic, A., Fisher, P., Bhagat, J., Belhajjame, K., Bacall, F., Hardisty, A., Nieva de la Hidalga, A., Balcazar Vargas, M. P., Sufi, S., and Goble, C. **(2013)**. The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Res.*, **41**(suppl 2,W1), W557–W561. 10.1093/nar/gkt328

Womble, D. D. **(1999a)**. GCG: The Wisconsin Package of sequence analysis programs. In Misener, S., Krawetz, S. A. (*Eds.*). *Bioinformatics Methods and Protocols*. *Methods Mol. Biol.*, **132**, 3–22. Humana Press. 10.1385/1-59259-192-2:3

Womble, D. D. **(1999b)**. Web-Based Interfaces for the GCG Sequence Analysis Programs. In Misener, S., Krawetz, S. A. (*Eds.*). *Bioinformatics Methods and Protocols*. *Methods Mol. Biol.*, **132**, 23–30. Humana Press. 10.1385/1-59259-192-2:23

Yachdav, G., Kloppmann, E., Kaján, L., Hecht, M., Goldberg, T., Hamp, T., Hönigschmid, P., Schafferhans, A., Roos, M., Bernhofer, M., Richter, L., Ashkenazy, H., Punta, M., Schlessinger, A., Bromberg, Y., Schneider, R., Vriend, G., Sander, C., Ben-Tal, N., Rost, and Burkhard **(2014)**. PredictProtein—an open resource for online prediction of protein structural and functional features. *Nucleic Acids Res.*, **42**(suppl 2,W1), W337–W343. 10.1093/nar/gku366

Yeung, N., Cline, M. S., Kuchinsky, A., Smoot, M. E., and Bader, G. D. **(2008)**. Exploring Biological Networks with Cytoscape Software. In *Curr. Protoc. Bioinformatics*, (23), 8.13.1–8.13.20. John Wiley & Sons, Inc. 10.1002/0471250953.bi0813s23

Yilmaz, P., Kottmann, R., Field, D., Knight, R., Cole, J. R., Amaral-Zettler, L., Gilbert, J. A., Karsch-Mizrachi, I., Johnston, A., Cochrane, G., Vaughan, R., Hunter, C., Park, J., Morrison, N., Rocca-Serra, P., Sterk, P., Arumugam, M., Bailey, M., Baumgartner, L., Birren, B. W., Blaser, M. J., Bonazzi, V., Booth, T., Bork, P., Bushman, F. D., Buttigieg, P. L. L., Chain, P. S., Charlson, E., Costello, E. K., Huot-Creasy, H., Dawyndt, P., DeSantis, T., Fierer, N., Fuhrman, J. A., Gallery, R. E., Gevers, D., Gibbs, R. A., San Gil, I., Gonzalez, A., Gordon, J. I., Guralnick, R., Hankeln, W., Highlander, S., Hugenholtz, P., Jansson, J., Kau, A. L., Kelley, S. T., Kennedy, J., Knights, D., Koren, O., Kuczynski, J., Kyrpides, N., Larsen, R., Lauber, C. L., Legg, T., Ley, R. E., Lozupone, C. A., Ludwig, W., Lyons, D., Maguire, E., Methé, B. A., Meyer, F., Muegge, B., Nakielny, S., Nelson, K. E., Nemergut, D., Neufeld, J. D., Newbold, L. K., Oliver, A. E., Pace, N. R., Palanisamy, G., Peplies, J., Petrosino, J., Proctor, L., Pruesse, E., Quast, C., Raes, J., Ratnasingham, S., Ravel, J., Relman, D. A., Assunta-Sansone, S., Schloss, P. D., Schriml, L., Sinha, R., Smith, M. I., Sodergren, E., Spo, A., Stombaugh, J., Tiedje, J. M., Ward, D. V., Weinstock, G. M., Wendel, D., White, O., Whiteley, A., Wilke, A., Wortman, J. R., Yatsunenko, T., and Glöckner, F. O. O. **(2011)**. Minimum information about a marker gene sequence (MIMARKS) and minimum information about any (x) sequence (MIxS) specifications. *Nat. Biotechnol.*, **29**(5), 415–420. 10.1038/nbt.1823