MolabIS: A Labs Backbone for Storing, Managing and Evaluating Molecular Genetics Data

Dissertation
for the award of the degree
"Doctor rerum naturalium" (Dr.rer.nat.)
of the Georg-August-Universität Göttingen

within the doctoral program for Environmental Informatics (PEI)
of the Georg-August University School of Science (GAUSS)

submitted by

TRUONG, VAN CHI CONG

from An Giang, Vietnam

Göttingen, 2013

Thesis Committee

- **Prof. Dr. Burkhard Morgenstern**
  Department of Bioinformatics, Institute of Microbiology & Genetics,
  University of Göttingen, Germany
- **Prof. Dr. Stephan Waack**
  Center for Computational Sciences, Institute of Computer Science,
  University of Göttingen, Germany
- **Dir. & Prof. Dr. Eildert Groeneveld**
  Department of Breeding & Genetic Resources, Institute of Farm Animal Genetics,
  Friedrich-Loeffler-Institut (FLI), Germany

Members of the Examination Board

- Referee: **Prof. Dr. Burkhard Morgenstern**
  Institute of Microbiology & Genetics, University of Göttingen, Germany
- Co-referee: **Dir & Prof. Dr. Eildert Groeneveld**
  Institute of Farm Animal Genetics, Friedrich-Loeffler-Institut, Germany

Further Members of the Examination Board

- **Prof. Dr. Stephan Waack**
  Institute of Computer Science, University of Göttingen, Germany
- **Prof. Dr. Wolfgang May**
  Institute for Informatics, University of Göttingen, Germany
- **Prof. Dr. Tim Beißbarth**
  Department of Medical Statistics, University of Göttingen, Germany
- **Prof. Dr. Carsten Damm**
  Institute for Informatics, University of Göttingen, Germany

Date of the oral examination: 13th February 2014

Acknowledgements

Certainly, this thesis would not have been possible without the help, encouragement and support of many individuals. I take this opportunity to extend my sincere gratitude and appreciation to all those who were involved in my PhD project. I would like to greatly acknowledge:

Prof. Dr. Burkhard Morgenstern for accepting this thesis, providing me useful guidance to proceed through the PhD program and supporting me during my study.

Dir. & Prof. Dr. Eildert Groeneveld for offering the opportunity to study this interesting topic, for sharing valuable knowledge, practical experience and essential advice necessary for me to develop the project and to complete my dissertation.

Prof. Dr. Stephan Waack for participating my PhD committee and giving me helpful suggestions and encouragement.

Dr. Zhivko Duchev, Dr. Linn Fenna Groeneveld, Mr. Detlef Schulze and Mr. Helmut Lichtenberg for their cooperation in the software development.

Dr. Steffen Weigend and Mrs. Annett Weigend for their cooperation in explaining the workflows and providing lab data to develop the software.

Dr. Martina Henning and Dr. Ulrich Baulain for sharing their constructive comments and ideas in my daily work and helping me to settle my family in Mariensee.

My colleagues at the Institute of Farm Animal Genetics in Mariensee for their friendly assistance and warm working environment.

The molecular labs surveyed in Germany and Vietnam for their data support and cooperation.

My friends for exchanging life experience and giving help whenever needed.

The German Federal Ministry of Education and Research (BMBF) for the financial support of this study.

My parents and my parents in law for their unconditional support and taking care of my daughter during my study.

Last but not least, my wife, Tran Nguyen, and my daughter, Thien Thu Truong, for their love and encouragement helped me to finish this long journey. They are the inspiration and motivation in my life.

Declaration

I hereby declare that I have written this PhD thesis myself independently, and that I have not submitted it at any other universitie worldwide.

TRUONG, VAN CHI CONG
December 2013
Göttingen, Germany

List of Publications

Papers in Peer Reviewed Journals

- **Truong CVC**, Duchev ZI and Groeneveld E: "Data Framework for Efficient Management of Sequence and Microsatellite Data in Biodiversity", *Archives Animal Breeding*, 2013, 56(6):50-64.

- **Truong CVC** and Groeneveld E: "An Efficient Approach to the Deployment of Complex Open Source Information Systems", *Bioinformation*, 2011, 7(4):152-153.

- **Truong CVC**, Groeneveld LF, Morgenstern B and Groeneveld E: "MolabIS - An Integrated Information System for Storing and Managing Molecular Genetics Data", *BMC Bioinformatics*, 2011, 12:425+.

- Duchev ZI, **Truong CVC** and Groeneveld E: "CryoWEB: Web software for the Documentation of the Cryo-preserved Material in Animal Gene Banks", *Bioinformation*, 2010, 5(5):219-220.

- Groeneveld E and **Truong CVC**: "A database for efficient storage and management of multi panel SNP data", *Archives Animal Breeding*, 2013, 56(103).

Papers and Posters in Proceedings of Conferences

- Krostitz S, **Truong CVC**, Müller U and Groeneveld E: "Development of Tools for Quality Assurance of Breeding Programs - QS@Breeding", Proceedings of the BLE Innovationstage in Bonn-Bad Godesberg, 29-30 October 2012.

- **Truong CVC** and Groeneveld E: "Deployment of Open-Source Bioinformatics Software Using Virtualization", Proceedings of the Annual Conference of German Society for Animal Production (DGfZ/GfT) in Halle, Germany, 12-13 September 2012.

- Krostitz S, **Truong CVC**, Müller U, Fischer R, Bergfeld U and Groeneveld E: "AroundBLUP – ein effektives Softwaretool zur Evaluierung von Zuchtwertschätzungen", Proceedings of DGfZ/GfT in Halle, Germany, 12-13 September 2012.

- **Truong CVC**, Krostitz S, Fischer R, Müller U and Groeneveld E: "A Software Pipeline for Animal Genetic Evaluation", Book of Abstracts of the 63rd Annual Meeting of the European Association for Animal Production (EAAP) in Bratislava, Slovakia, 27-31 August 2012.

- Groeneveld E and **Truong CVC**: "SNPpit - Efficient Data Management for High Density Genotyping", Book of Abstracts of the 63rd EAAP in Bratislava, Slovakia, 27-31 August 2012.

- Müller U, Fischer R, **Truong CVC**, Groeneveld E and Bergfeld U: "WebLOAD - A Web Frontend to Create a Consistent Dataset from Multiple Text Files", Book of Abstracts of the 63rd EAAP in Bratislava, Slovakia, 27-31 August 2012.

- **Truong CVC** and Groeneveld E: "A Perl Toolkit for Large-scale SNP Genotype Data Management", Proceedings of DGfZ/GfT in Freising-Weihenstephan, Germany, 6-7 September 2011, B20.

- Krostitz S, **Truong CVC**, Müller U, Bergfeld U and Groeneveld E: "Von PEST zu ZwISSS - Eine Software Pipeline", Proceedings of DGfZ/GfT in Freising-Weihenstephan, Germany, 6-7 September 2011.

- **Truong CVC** and Groeneveld E: "MolabIS – An Open Source Information System for Sequencing and Genotyping Workflows", Proceedings of the 9th World Congress on Genetics Applied to Livestock Production (WCGALP) in Leipzig, Germany, 1-6 August 2010.

- Krostitz S, **Truong CVC**, Müller U and Groeneveld E: "Development of Tools for Quality Assurance of Breeding Programs - QS@Breeding", Proceedings of the BLE Innovationstage in Berlin, Germany, 6-7 October 2010.

- **Truong CVC** and Groeneveld E: "Information Management System for Sequences and Microsatellites Data", Proceedings of DGfZ/GfT in Giessen, Germany, 16-17 September 2009.

- **Truong CVC**, Duchev ZI and Groeneveld E: "MolabIS - Effective Management of Genetic Data in Farm Animal Biodiversity Studies", Book of Abstracts of the 60th EAAP in Barcelona, Spain, 24-27 August 2009.

- **Truong CVC**, Duchev ZI and Groeneveld E: "A Software Package for Managing and Evaluating DNA Sequence and Microsatellite Data", Proceedings of the GIL Conference - Demands on IT in Agriculture, Forestry and Food Industry by Globalization and Climate Change in Rostock, Germany, 09-10 March 2009.

- **Truong CVC**, Duchev ZI and Groeneveld E: "CryoWEB - A Web Application for Managing National Genebanks", Proceedings of DGfZ/GfT in Bonn, Germany, 17-18 September 2008.

- **Truong CVC**, Duchev ZI and Groeneveld E: "Design and Implementation of an Information System for National Genebanks Management", Book of Abstracts of the 59th EAAP in Vilnius, Lithuania, 24-27 August 2008.

- **Truong CVC**, Duchev ZI and Groeneveld E: "A Formalized Workflow for Management of Molecular Genetics Data", Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies (RIVF) in Ho Chi Minh, Vietnam, 13-17 July 2008.

- **Truong CVC** and Groeneveld E: "Workflow for Storing, Managing, and Evaluating Molecular Genetics Data", Proceedings of DGfZ/GfT in Stuttgart, Germany, 26-27 September 2007.

Presentations at Workshops

- Groeneveld E, Müller U, Kostritz S, Fischer R and **Truong CVC**: "New development in breeding value estimation", Pig Information Day, Pretoria, South Africa, 16 May 2012.

- Groeneveld E and **Truong CVC**: "SNP data management in breeding programs", SA-Studbook, Bonsmara Beef Board, South Africa, 11 May 2012.

- Groeneveld E, Müller U, Kostritz S, Fischer R and **Truong CVC**: "Quality assurance in beef breeding programs", SA-Studbook, Bonsmara Beef Board, South Africa, 11 May 2012.

- Groeneveld E, Müller U, Kostritz S, Fischer R and **Truong CVC**: "Quality assurance in breeding programs", Beef breeders conference, Ladysmith, Kwa-Zulu-Natal, South Africa, 10 May 2012.

- Groeneveld E and **Truong CVC**: “SNPpit - SNP data management”, University of Pretoria, Department of Animal and Wildlife Science, South Africa, 7 May 2012.
- **Truong CVC** and Groeneveld E: “SNP data management”, SNP Workshop in Institute of Farm Animal Genetics, Mariensee, Germany, 8-9 February 2012.
- **Truong CVC** and Groeneveld E: “Towards integration of gene bank and genomic data”, EFABISnet International Conference in Palermo, Italy, 01-02 December 2010.
- **Truong CVC** and Groeneveld E: “CryoWeb Lectures”, International CryoWeb Workshop in Institute of Farm Animal Genetics, Mariensee, Germany, 9-13 February 2009.
- Groeneveld E, Duchev ZI, Henning M and **Truong CVC**: “National genebanks and CryoWeb software”, International CryoWeb Workshop in Institute of Farm Animal Genetics, Mariensee, Germany, 9-13 February 2009.
- **Truong CVC,** Duchev ZI and Groeneveld E: “CryoWeb Lectures”, CryoWeb Workshop in National Institute of Animal Husbandry, Hanoi, 30 November - 4 December, 2009.
- **Truong CVC,** and Groeneveld E: “MolabIS - Open source information system for managing data in molecular labs”, MolabIS Workshop in National Institute of Animal Husbandry, Hanoi, 30 November - 4 December, 2009.
- **Truong CVC** and Groeneveld E: “MolabIS: Integration into the Labs IT Infrastructure”, MolabIS Workshop in Cantho University, Vietnam, 10-11 December, 2009.
- **Truong CVC** and Groeneveld E: “National sample management information system”, Biotechnology Seminar in Leibniz University, in Hannover, Germany, 1-7 July 2007.

Contents

Abstract

Using paper lab books and spreadsheets to store and manage growing datasets in a file system is inefficient, time consuming and error-prone. Therefore, the overall purpose of this study is to develop an integrated information system for small laboratories conducting Sanger sequencing and microsatellite genotyping projects.

To address this, the thesis has investigated the following three issues. First, we proposed a uniform solution using the workflow approach to efficiently collect and store data items in different labs. The outcome is the design of the formalized data framework which is the basic to create a general data model for biodiversity studies. Second, we designed and implemented a web-based information system (MolabIS) allowing lab people to store all original data at each step of their workflow. MolabIS provides essential tools to import, store, organize, search, modify, report and export relevant data. Finally, we conducted a case study to evaluate the performance of MolabIS with typical operations in a production mode. Consequently, we can propose the use of virtual appliance as an efficient solution for the deployment of complex open-source information systems like MolabIS.

The major result of this study, along with the publications, is the MolabIS software which is freely released under GPL license at http://www.molabis.org. With its general data model, easy installation process and additional tools for data migration, MolabIS can be used in a wide range of molecular genetics labs.

Chapter 1

Introduction

1.1 Preamble

Along with the development of other scientific disciplines, informatics has attracted the attention of many scientists worldwide. The application of computer science to the effective exploitation of specialized information is an indispensable need in most areas, especially in biology [12]. The term "bioinformatics", therefore, has been a hot topic [43, 10]. With a tremendous progress in the last few years [3, 4, 5, 7], today bioinformatics has become a relatively stable discipline [48, 6].

Bioinformatics may be understood in many different ways [28, 33, 13, 29, 61, 39]. For instance, according to National Center for Biotechnology Information (NCBI), "bioinformatics is the field of science in which biology, computer science, and information technology merge to form a single discipline" [40]. The objective is "the collection, organization and analysis of large amounts of biological data, using networks of computers and databases" [2]. Generally, although there are no established definitions, this young interdisciplinary field is always considered a combination of computer science and biological science along with other disciplines [1].

Molecular biology and genetics has developed at an incredible rate. Scientific advances in molecular technologies such as Polymerase Chain Reaction (PCR) [31], genotyping technologies [47] have revolutionized methods with which scientists approach biological problems. Consequently, the processing time of experiments have been shortened significantly. Complicated experiments, which previously could only be carried out in months or even years, today give results in hours. It means that more and more experiments are conducted in molecular labs. This leads to the explosive growth in the amount of biological data. Therefore, the demand for the development of tools and methods to analyze, manipulate and manage biological data is also increasing.

1.2 Bioinformatics software

In the context of bioinformatics, the term "software" implies computer applications which involve database systems and computational programs. These applications can be divided into three basic groups: (i) data analysis and process, (ii) data report and visualization, and (iii) data storage and management.

The first group is the core part of bioinformatics that deals with a wide range of computational processes and analytical techniques [14]. It usually involves a degree of algorithmic complexity and principles driving advances in bioinformatics. The two basic areas of this group are the fundamental research on design of algorithms and the implementation of algorithms for various applications. For instance, dynamic programming algorithms to compare two biological sequences are important contributions [41, 51]. Over the past years, the research community has devoted special attention to developing many bioinformatics software packages for this group. Thus, many open-source projects (e.g. EMBOSS [49], BioPerl [52], Bio* Toolkits [34], BioConductor [21], BioJava [26], BioMart [25], BioPython [11], BioRuby [23]) have been developed to facilitate the development of bioinformatics software [53].

Because of the difficulties inherent in fully understanding large datasets increasing in size and complexity, editing tools and data visualization techniques have become an attractive field in bioinformatics research. Therefore, much effort has also been devoted to the second group. A large number of visualization tools and editors have been developed (e.g. DNA alignment [37, 50], protein modeling [46], microsatellite tools [44, 35], gene cluster visualization [45]). Software tools range from simple programs to complex integrated systems to serve different needs. Most of them inherit computational methodologies or algorithms developed in the first group. The major objective is to provide means which can help biologists visually view and efficiently operate their experimental results.

While the first two groups mainly focus on data exploration and knowledge discovery, the last group addresses efficient solutions for long-term data storage and management. The center of third group is the development of databases that is considered a hinterland of bioinformatics [60]. This group bridges the gap between data analysis software from two groups above and data inputs which should be available in custom formats. In other words, database applications help us to collect, store and organize structured data so that it may be quickly retrieved and exported in formats required by other software.

1.3 Data management

In the last ten years, we have witnessed a continuous rapid growth in volume and diversity of biological data. Consequently, data management has become essential to many molecular labs. This has been driving demands for software to efficiently manage all kinds of data generated from different experiments. In this context, information systems are excellent means to store structured data, manage experimental results and support lab work.

In the field of laboratory informatics, LIMS (Laboratory Information Management Systems) is a specialized class of software which implements functions addressing data management [32]. Since the needs of data management vary from lab to lab, the features of LIMS are very different. Therefore, a LIMS designed for a specific lab is difficult or even impossible to be used in other labs. For instance, a LIMS developed for chemical labs is not suitable for medical diagnostic labs. Even LIMS developed for molecular biology labs very diverse to meet various needs such as mutation screening [59], functional genomic analysis [15], or management of biologic information [8]. Thus, the term "LIMS", which is used in this thesis, implies information systems for molecular genetics labs. We focus on the management of samples and molecular data rather than other kinds of information (e.g. lab infrastructure, chemicals, financial).

In recent years, a large number of information systems have been developed for molecular genetics labs. Many data management systems have been successfully employed in large-scale biology projects [62]. Nevertheless, most of them focused on the storage and management of microarray data [19, 9, 30, 54, 36, 58, 20] and proteomics data [22, 38, 16]. Some early efforts were also directed towards developing information systems to keep track of sequencing [60, 18] or genotyping workflows [27].

1.4 Objectives

The overall purpose of this thesis was to contribute a general data management solution for small molecular genetics labs to efficiently store and manage data derived from their research workflows. The scope of our project is to develop an open-source information system to manage relatively larger datasets generated from Sanger sequencing and microsatellite genotyping workflows. It is focused on biodiversity studies with the following three areas contributing to the final system: (i) the design of a formalized data framework, (ii) the implementation of an integrated information system, and (iii) the development of a solution for software deployment.

1.5 Thesis layout

The thesis is organized as a manuscript-based document which consists of eight chapters. Chapter 1 is a general introduction to bioinformatics software. It gives an overview of LIMS development for molecular biology labs. The objectives and scope of the thesis are also supplied in this chapter. Chapter 2 lists out all relevant publications which form the body of the thesis.

The following three chapters are original papers which have been published or accepted for publication by peer-reviewed journals. Particularly, chapter 3 provides a detailed description of a method used for constructing a formalized data framework to manage data in biodiversity studies [79]. We present fundamental procedures of the workflow approach for collecting and representing data streams and data items. Besides, this chapter also indicates a uniform solution to efficiently store variable data items in different labs. Chapter 4 mainly describes the design and implementation of MolabIS [81]. We explain different aspects of database design with an emphasis on the general data model derived from the results in the previous chapter. In addition, we present the application architecture and technologies involved in the implementation of MolabIS as a web-based information system. Moreover, all functionality of MolabIS is also provided in this chapter. Chapter 5 deals with finding a proper solution to deploy complex open-source information systems [80]. We conduct a case study to evaluate the performance of MolabIS on a real system and four virtual systems running MolabIS appliances. Then, the benchmark results are reported to conclude that the virtual appliance is sufficiently fast for normal production mode.

Chapter 6 and Chapter 7 (peer-reviewed papers) are additional contributions to strengthen the thesis. The former presents the development of a web-based information system for the data management of a national animal gene bank [66]. The latter proposes a database design for efficient storage and management of SNP data [70].

Chapter 8 summaries the achieved results and gives a general discussion. We finally consider some possibilities for future work.

1.6 References

[1] Bioinformatics.org wiki, 2012.

[2] The state of the genome: glossary, 2012.

[3] ALTMAN, R. B. Editorial: Annual progress in bioinformatics. *Briefings in Bioinformatics 6*, 1 (2005), 4–5.

[4] ALTMAN, R. B. Annual progress in bioinformatics 2006. *Briefings in Bioinformatics 7*, 3 (2006), 209–210.

[5] ALTMAN, R. B. Editorial: Current progress in bioinformatics 2007. *Briefings in Bioinformatics 8*, 5 (2007), 277–278.

[6] ALTMAN, R. B. Editorial: Current progress in bioinformatics 2010. *Briefings in Bioinformatics 11*, 1 (2010), 1–2.

[7] ALTMAN, R. B. Editorial: Current progress in bioinformatics 2012. *Briefings in Bioinformatics 13*, 4 (2012), 393–394.

[8] BAUCH, A., ADAMCZYK, I., BUCZEK, P., ELMER, F.-J. J., ENIMANEV, K., GLYZEWSKI, P., KOHLER, M., PYLAK, T., QUANDT, A., RAMAKRISHNAN, C., BEISEL, C., MALMSTRÖM, L., AEBERSOLD, R., AND RINN, B. openBIS: a flexible framework for managing and analyzing complex data in biology research. *BMC bioinformatics 12*, 1 (2011), 468+.

[9] BRAZMA, A., PARKINSON, H. E., SARKANS, U., SHOJATALAB, M., VILO, J., ABEYGUNAWARDENA, N., HOLLOWAY, E., KAPUSHESKY, M., KEMMEREN, P., LARA, G. G., OEZCIMEN, A., ROCCA-SERRA, P., AND SANSONE, S.-A. Arrayexpress - a public repository for microarray gene expression data at the ebi. *Nucleic Acids Research 31*, 1 (2003), 68–71.

[10] CATTLEY, S. A review of bioinformatics degrees in australia. *Briefings in Bioinformatics 5*, 4 (2004), 350–354.

[11] 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. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics 25*, 11 (2009), 1422–1423.

[12] COHEN, J. Bioinformatics - an introduction for computer scientists. *ACM Computing Surveys 36*, 2 (2004), 122–158.

[13] COUNSELL, D. A review of bioinformatics education in the uk. *Briefings in Bioinformatics 4*, 1 (2003), 7–21.

[14] DELLA VEDOVA, G., AND DONDI, R. A library of efficient bioinformatics algorithms. *Applied bioinformatics 2*, 2 (2003), 117–121.

[15] DONOFRIO, N., RAJAGOPALON, R., BROWN, D., DIENER, S. E., WINDHAM, D., NOLIN, S., FLOYD, A., MITCHELL, T. K., GALADIMA, N., TUCKER, S., ORBACH, M. J., PATEL, G., FARMAN, M. L., PAMPANWAR, V., SODERLUND, C., LEE, Y.-H., AND DEAN, R. A. 'paclims': A component lim system for high-throughput functional genomic analysis. *BMC Bioinformatics 6* (2005), 94.

[16] DROIT, A., HUNTER, J., ROULEAU, M., ETHIER, C., PICARD-CLOUTIER, A., BOURGAIS, D., AND POIRIER, G. PARPs database: A LIMS systems for protein-protein interaction data mining or laboratory information management system. *BMC Bioinformatics 8*, 1 (2007), 483.

[17] DUCHEV, Z., TRUONG, C. V. C., AND GROENEVELD, E. Cryoweb: Web software for the documentation of the cryo-preserved material in animal gene banks. *Bioinformation 5*, 5 (2010), 219–220.

[18] DUNCAN, S., SIRKANUNGO, R., MILLER, L., AND PHILLIPS, G. J. Dragnet: Software for storing, managing and analyzing annotated draft genome sequence data. *BMC Bioinformatics 11* (2010), 100.

[19] EDGAR, R., DOMRACHEV, M., AND LASH, A. E. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research 30*, 1 (2002), 207–210.

[20] GATTIKER, A., HERMIDA, L., LIECHTI, R., XENARIOS, I., COLLIN, O., ROUGEMONT, J., AND PRIMIG, M. Mimas 3.0 is a multiomics information management and annotation system. *BMC Bioinformatics 10*, 1 (2009), 151.

[21] GENTLEMAN, R. C., CAREY, V. J., BATES, D. M., BOLSTAD, B., DETTLING, M., DUDOIT, S., ELLIS, B., GAUTIER, L., GE, Y., GENTRY, J., HORNIK, K., HOTHORN, T., HUBER, W., IACUS, S., IRIZARRY, R., LI, F. L. C., MAECHLER, M., ROSSINI, A. J., SAWITZKI, G., SMITH, C., SMYTH, G., TIERNEY, L., YANG, J. Y. H., AND ZHANG, J. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology 5* (2004), R80.

[22] GOH, C.-S., LAN, N., ECHOLS, N., DOUGLAS, S. M., MILBURN, D., BERTONE, P., XIAO, R., MA, L.-C., ZHENG, D., WUNDERLICH, Z., ACTON, T., MONTELIONE, G. T., AND GERSTEIN, M. SPINE 2: a system for collaborative structural proteomics within a federated database framework. *Nucleic Acids Res 31*, 11 (2003), 2833–2838.

[23] GOTO, N., PRINS, P., NAKAO, M., BONNAL, R. J. P., AERTS, J., AND KATAYAMA, T. Bioruby: bioinformatics software for the ruby programming language. *Bioinformatics 26*, 20 (2010), 2617–2619.

[24] GROENEVELD, E., AND TRUONG, C. V. C. A database for efficient storage and management of multi panel snp data. *Archives Animal Breeding 56*, 103 (2013).

[25] HAIDER, S., BALLESTER, B., SMEDLEY, D., ZHANG, J., RICE, P. M., AND KASPRZYK, A. Biomart central portal - unified access to biological data. *Nucleic Acids Research 37*, Web-Server-Issue (2009), 23–27.

[26] HOLLAND, R. C. G., DOWN, T. A., POCOCK, M. R., PRLIC, A., HUEN, D., JAMES, K., FOISY, S., DRÄGER, A., YATES, A., HEUER, M. L., AND SCHREIBER, M. J. Biojava: an open-source framework for bioinformatics. *Bioinformatics 24*, 18 (2008), 2096–2097.

[27] JAYASHREE, B., REDDY, P. T., LEELADEVI, Y., CROUCH, J. H., MAHALAKSHMI, V., BUHARIWALLA, H. K., ESHWAR, K. E., MACE, E., FOLKSTERMA, R., SENTHILVEL, S., VARSHNEY, R. K., SEETHA, K., RAJALAKSHMI, R., PRASANTH, V. P., CHANDRA, S., SWARUPA, L., SRIKALYANI, P., AND HOISINGTON, D. A. Laboratory information management software for genotyping workflows: applications in high throughput crop genotyping. *BMC Bioinformatics 7* (2006), 383.

[28] KAMINSKI, N. Bioinformatics. a user's perspective. *Am J Respir Cell Mol Biol. 23* (2000), 705–711.

[29] KOCH, I., AND FUELLEN, G. A review of bioinformatics education in germany. *Briefings in Bioinformatics 9*, 3 (2008), 232–242.

[30] KOKOCINSKI, F., WROBEL, G., HAHN, M., AND LICHTER, P. QuickLIMS: facilitating the data management for DNA-microarray fabrication. *Bioinformatics 19*, 2 (2003), 283–284.

[31] LAUERMAN, L. H. Advances in pcr technology. *Anim Health Res Rev. 5*, 2 (2004), 247–248.

[32] LIMSWIKI. Glossary: Laboratory information management system, 2012.

[33] LUSCOMBE, N. M., GREENBAUM, D., AND GERSTEIN, M. What is bioinformatics? a proposed definition and overview of the field. *Methods of information in medicine 40*, 4 (2001), 346–358.

[34] MANGALAM, H. The Bio* toolkits - A brief overview. *Briefings in Bioinformatics 3*, 3 (2002), 296–302.

[35] MEGLÉCZ, E., COSTEDOAT, C., DUBUT, V., GILLES, A., MALAUSA, T., PECH, N., AND MARTIN, J.-F. Qdd: a user-friendly program to select microsatellite markers and design primers from large sequencing projects. *Bioinformatics 26*, 3 (2010), 403–404.

[36] MONNIER, S., COX, D. G., ALBION, T., AND CANZIAN, F. T.I.M.S: TaqMan Information Management System, tools to organize data flow in a genotyping laboratory. *BMC Bioinformatics 6* (2005), 246.

[37] MORGENSTERN, B. Dialign: multiple dna and protein sequence alignment at bibiserv. *Nucleic Acids Research 32* (2004), 33–36.

[38] MORISAWA, H., HIROTA, M., AND TODA, T. Development of an open source laboratory information management system for 2-D gel electrophoresis-based proteomics workflow. *BMC Bioinformatics 7*, 1 (2006), 430+.

[39] NATIONAL INSTITUTES OF HEALTH, WASHINGTON, U. Glossary: Bioinformatics, 2012.

[40] NCBI. What is bioinformatics?, 2012.

[41] NEEDLEMAN, S. B., AND WUNSCH, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology 48*, 3 (1970), 443 – 453.

[42] ORRO, A., GUFFANTI, G., SALVI, E., MACCIARDI, F., AND MILANESI, L. SNPLims: a data management system for genome wide association studies. *BMC Bioinformatics 9*, 2 (March 2008).

[43] OUZOUNIS, C. A., AND VALENCIA, A. Early bioinformatics: the birth of a discipline - a personal view. *Bioinformatics 19*, 17 (2003), 2176–2190.

[44] PARK, S. D. E. *Trypanotolerance in West African Cattle and the Population Genetic Effects of Selection*. PhD thesis, University of Dublin, 2001.

[45] PEJAVER, V. R., AN, J., RHEE, S., BHAN, A., CHOI, J.-H., LIU, B., LEE, H., BROWN, P. J., KYSELA, D., BRUN, Y. V., AND KIM, S. Geneclusterviz: a tool for conserved gene cluster visualization, exploration and analysis. *Bioinformatics 28*, 11 (2012), 1527–1529.

[46] PONS, J.-L., AND LABESSE, G. @tome-2: a new pipeline for comparative modeling of protein-ligand complexes. *Nucleic Acids Research 37*, Web-Server-Issue (2009), 485–491.

[47] RAGOUSSIS, J. Genotyping technologies for genetic research. *Annual review of genomics and human genetics 10*, 1 (2009), 117–133.

[48] RHEE, S. Y. Y., DICKERSON, J., AND XU, D. Bioinformatics and its applications in plant biology. *Annual review of plant biology 57*, 1 (2006), 335–360.

[49] RICE, P., LONGDEN, I., AND BLEASBY, A. Emboss: the european molecular biology open software suite. *Trends Genet 16*, 6 (2000), 276–7.

[50] SANCHEZ-VILLEDA, H., SCHROEDER, S. G., FLINT-GARCIA, S., GUILL, K. E., YAMASAKI, M., AND MCMULLEN, M. D. Dnaaligneditor: Dna alignment editor tool. *BMC Bioinformatics 9* (2008).

[51] SMITH, T., AND WATERMAN, M. Identification of common molecular subsequences. *Journal of Molecular Biology 147*, 1 (1981), 195–197.

[52] STAJICH, J. E., BLOCK, D., BOULEZ, K., BRENNER, S. E., CHERVITZ, S. A., DAGDIGIAN, C., FUELLEN, G., GILBERT, J. G., KORF, I., LAPP, H., LEHVASLAIHO, 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. The Bioperl Toolkit: Perl Modules for the Life Sciences. *Genome Res. 12*, 10 (2002), 1611–1618.

[53] STAJICH, J. E., AND LAPP, H. Open source tools and toolkits for bioinformatics: significance, and where are we? *Briefings in Bioinformatics 7*, 3 (2006), 287–296.

[54] SWERTZ, M. A., DE BROCK, E. O., VAN HIJUM, S. A. F. T., DE JONG, A., BUIST, G., BAERENDS, R. J. S., KOK, J., KUIPERS, O. P., AND JANSEN, R. C. Molecular genetics information system (molgenis): alternatives in developing local experimental genomics databases. *Bioinformatics 20*, 13 (2004), 2075–2083.

[55] TRUONG, C. V. C., DUCHEV, Z., AND GROENEVELD, E. Data framework for efficient management of sequence and microsatellite data in biodiversity. *Archives Animal Breeding 56*, 6 (2013), 50–64.

[56] TRUONG, C. V. C., AND GROENEVELD, E. An efficient approach to the deployment of complex open source information systems. *Bioinformation 7*, 4 (2011), 152–153.

[57] TRUONG, C. V. C., GROENEVELD, L. F., MORGENSTERN, B., AND GROENEVELD, E. Molabis - an integrated information system for storing and managing molecular genetics data. *BMC Bioinformatics 12* (2011), 425.

[58] VALLON-CHRISTERSSON, J., NORDBORG, N., SVENSSON, M., AND HÄKKINEN, J. Base - 2nd generation software for microarray data management and analysis. *BMC Bioinformatics 10* (2009), 330.

[59] VOEGELE, C., TAVTIGIAN, S. V., DE SILVA, D., CUBER, S., THOMAS, A., AND CALVEZ-KELM, F. L. A laboratory information management system (lims) for a high throughput genetic platform aimed at candidate gene mutation screening. *Bioinformatics 23*, 18 (2007), 2504–2506.

[60] WENDL, M. C., SMITH, S., POHL, C. S., DOOLING, D. J., CHINWALLA, A. T., CROUSE, K., HEPLER, T., LEONG, S., CARMICHAEL, L. K., NHAN, M., OBERKFELL, B. J., MARDIS, E. R., HILLIER, L. W., AND WILSON, R. K. Design and implementation of a generalized laboratory data model. *BMC Bioinformatics 8* (2007).

[61] WILLIAMS, J. M., MANGAN, M. E., PERREAULT-MICALE, C., LATHE, S., SIROHI, N., AND LATHE, W. C. Openhelix: bioinformatics education outside of a different box. *Briefings in Bioinformatics 11*, 6 (2010), 598–609.

[62] WRUCK, W., PEUKER, M., AND REGENBRECHT, C. R. A. Data management strategies for multinational large-scale systems biology projects. *Briefings in Bioinformatics* (2012).

Chapter 2

List of Publications

This thesis is based on the following original papers:

- **Chapter 3: Truong CVC**, Duchev Z and Groeneveld E: "Data Framework for Efficient Management of Sequence and Microsatellite Data in Biodiversity", *Archives Animal Breeding*, 2013, 56(6):50-64.
- **Chapter 4: Truong CVC**, Groeneveld LF, Morgenstern B and Groeneveld E: "MolabIS - An Integrated Information System for Storing and Managing Molecular Genetics Data", *BMC Bioinformatics*, 2011, 12:425+.
- **Chapter 5: Truong CVC** and Groeneveld E: "An Efficient Approach to the Deployment of Complex Open Source Information Systems", *Bioinformation*, 2011, 7(4):152-153.
- **Chapter 6:** Duchev ZI, **Truong CVC** and Groeneveld E: "CryoWEB: Web software for the Documentation of the Cryo-preserved Material in Animal Gene Banks", *Bioinformation*, 2010, 5(5):219-220.
- **Chapter 7:** Groeneveld E and **Truong CVC**: "A database for efficient storage and management of multi panel SNP data", *Archives Animal Breeding*, 2013, 56(103).

Chapter 3

Data Framework for Efficient Management of Sequence and Microsatellite Data in Biodiversity Studies

Citation:

Truong CVC[1], Duchev Z and Groeneveld E. "Data Framework for Efficient Management of Sequence and Microsatellite Data in Biodiversity Studies", *Archives Animal Breeding*, 2013, 56(6):50-64.

Original Contribution:

Truong CVC collected data, designed the data framework, and wrote the manuscript.

[1]Corresponding author

Archiv Tierzucht
Archives Animal Breeding

ISSN: 0003-9438

Original study

Data framework for efficient management of sequence and microsatellite data in biodiversity studies

Cong V. C. Truong, Zhivko Duchev and Eildert Groeneveld

Department of Breeding and Genetic Resources, Institute of Farm Animal Genetics (FLI), Neustadt, Germany

Archiv Tierzucht 56 (2013) 6
doi: 10.7482/0003-9438-56-006

Received: 8 December 2011
Accepted: 13 June 2012
Online: 8 February 2013

Corresponding author:
Cong Van Chi Truong; email: cong.chi@fli.bund.de
Department of Breeding and Genetic Resources, Institute of Farm Animal Genetics (FLI), Höltystr. 10, 31535 Neustadt, Germany

Archiv Tierzucht 56 (2013) 6, 50-64

Open Access

Original study

Data framework for efficient management of sequence and microsatellite data in biodiversity studies

Cong V. C. Truong, Zhivko Duchev and Eildert Groeneveld

Department of Breeding and Genetic Resources, Institute of Farm Animal Genetics (FLI), Neustadt, Germany

Abstract

In recent years, software packages for the management of biological data have rapidly been developing. However, currently, there is no general information system available for managing molecular data derived from both Sanger sequencing and microsatellite genotyping projects. A prerequisite to implementing such a system is to design a general data model which can be deployed to a wide range of labs without modification or customization. Thus, this paper aims to (1) suggest a uniform solution to efficiently store data items required in different labs, (2) describe procedures for representing data streams and data items (3) and construct a formalized data framework. As a result, the data framework has been used to develop an integrated information system for small labs conducting biodiversity studies.

Keywords: data modeling, biodiversity, molecular genetics, information system

Abbreviations: BLOB: binary large object, DIT: data integration table, DNA: deoxyribonucleic acid, GPS: global positioning system, PCR: polymerase chain reaction, UDI: unknown data items

Introduction

In biodiversity studies, modern genetic techniques using molecular markers are extensively applied in many labs. These markers, sometimes called DNA markers, are considered versatile tools for exploring genetic diversity (Vignal *et al*. 2002, Baumung *et al*. 2004, Rudd *et al*. 2005). For instance, microsatellite markers and mitochondrial DNA markers are commonly used for assessing genetic structure (Rosenberg *et al*. 2001, Granevitze *et al*. 2007, Granevitze *et al*. 2009) and tracking ancestry through maternal lineages (Liu *et al*. 2006, Oka *et al*. 2007), respectively. This has resulted in relatively large amounts of heterogeneous data collected

Archiv Tierzucht 56 (2013) 6, 50-64
doi: 10.7482/0003-9438-56-006

Received: 8 Dezember 2011
Accepted: 13 June 2012
Online: 8 February 2013

Corresponding author:
Cong Van Chi Truong; email: cong.chi@fli.bund.de
Department of Breeding and Genetic Resources, Institute of Farm Animal Genetics (FLI), Höltystr. 10, 31535 Neustadt, Germany

and stored in labs over the years. Consequently, data analysis, retrieval and reuse are difficult and time-consuming since most operations are handled manually.

In practice, labs still use traditional methods to manage their data: paper lab books and file systems are major types of data storage; and spreadsheets are used as a typical means for data handling. From information collected in many labs, we summarize four issues which should be analysed for data integration. First, data streams (determining when and which data elements are created, recorded and retrieved) vary project by project and lab by lab. Second, most of the data is pipelined from one step to another. Third, data collected from various sources is stored in a variety of formats. Finally, data items required at each step in labs are not identical.

To address the above mentioned difficulties, several information systems (Jayashree *et al*. 2006, Wendl *et al*. 2007, Schönherr *et al*. 2009, Weißensteiner *et al*. 2010) have been developed. However, none of them provides a general solution to meet the varying requirements of molecular genetics labs. Indeed, the data models of these systems have been designed to serve specific needs of a particular lab, and thus are difficult or even impossible to be used elsewhere. In this context, a data model should be designed at the general level so that it can meet basic needs of different labs while at the same time specific requirements are also considered.

Biodiversity studies are usually conducted through a series of basic steps as specified in textbooks or technical documents. At each step (e.g. DNA extraction, electrophoresis) a number of lab activities must be performed. Depending on the research objective, experimental method and lab infrastructure, labs use their own protocols or procedures to conduct the lab work. Therefore, data processing operations as well as data storage needs are different from lab to lab. Here, we aim to build a data framework for creating a general data model which can capture data derived from Sanger sequencing (Sanger *et al*. 1975, Sanger *et al*. 1977) and microsatellite genotyping experiments of biodiversity studies.

Therefore, the objectives of this paper are to (1) describe a method used to efficiently store data items in different labs, (2) present procedures for representing data items systematically and (3) create a formalized data framework for developing an integrated information system in the context of biodiversity studies.

Methods

Data storage architecture

Molecular genetics labs conducting biodiversity studies may require common data items to store and keep track of their samples and molecular data. However, with different technologies, machines and research objects, labs also need additional data items to meet their specific requirements. Even within a lab, the details of data storage vary among projects and researchers. The following is a simple example of data collection for storing information on individuals. Since all labs need minimum information such as *individual ID*, *species* and *genetic group* to carry out their biodiversity analysis, it is easy to make an initial list of those essential data items. The list may get updated by some labs which require extension like *sex*, *photo, date of birth*. Yet other labs may have even more specific data items such as *color of plant*, *weight of animal*, *number of piglets* or *number of eggs*. Therefore, the more labs are surveyed, the more data items will be suggested.

The abstraction of the above observation leads us to proposing a three group classification, namely »core« (C), »extended« (E) and »specific« (S). Considering three labs only to build a common data framework will result in Figure 1. The challenge is now how to translate this abstract view into a real life database structure applicable to any lab.

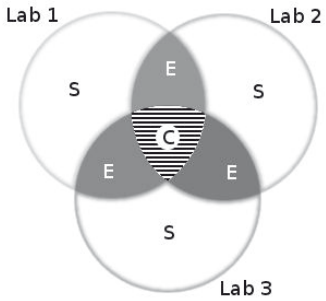

Figure 1
An example of data collection from three labs: the data items are classified into three data groups so called »Core« (C), »Extended« (E) and »Specific« (S).

There are a number of ways to choose data items for creating a common data framework. The first is to focus on data items required in all labs. The second is to store all data items suggested in any lab. The former helps to create a compact data framework, thus implementing software more easily and faster. However, common and specific needs of most labs are ignored. Obviously, this shortcoming can be resolved in the latter, but it suffers from another drawback. Because of storing a large number of data items from all groups, the data model becomes bulky and inefficient. It not only costs more effort in software implementation but also creates complex interfaces with dozens of unused inputs on the entry forms. A better way is only to store all data items of groups »C« and »E« in the database. For group »S«, labs would need to customize the data model to store their own data items. This modification of the data model requires a hand from a programmer, who is rarely available in molecular genetics labs. Clearly, none of these ways is a proper solution. In addition, all of the above suggestions may be applied only if we know exactly the labs wanting to use the software.

In this paper, we aim to construct a data framework with a minimum set of data items. The data framework is built so that it can meet requirements of labs without customization. The following is our solution to address this issue.

Based on the principles of carrying out lab work in biodiversity studies, we can define data items in group »C« easily. This group consists of essential information such as identifications (e.g. *sample ID*), experimental results (e.g. *gel image*) to keep track of samples and molecular data which is available in each lab. The extended data items in group »E« are specified from our experience. They are most commonly used data items supporting information about the time (e.g. *sampling date*) or the person involved (e.g. *action user*). The information in this group helps to efficiently search data or make meaningful reports. However, not all elements may be available in each lab. Hence, the remaining work is how to determine the data items in group »S« which may be very different among labs.

To facilitate this effort, we consider our data framework at an abstract level constructed by two parts. The first one comprises all data items in two groups »C« and »E« and the second one consists of specific data items in group »S«. Obviously, the former can be identified while the latter is unknown. In other words, the core and extended data items can be explicitly defined and named but the rest (specific data items) are unpredictable. In order to find a

proper mechanism, we determine the reasons why lab users want to keep specific data items in the database. Here, their major reason is to have more information on the stored samples. Almost all data items in group »S« such as budget of the project, details of lab work, chemicals, PCR program, etc. are not used for searching and tracking data. Hence, the major objective is to somehow store these data elements as referable components to the objects of interest. Thus, instead of decomposing unknown data items (UDI), we suggest to hold all in a uniform data storage block. In terms of database modeling, such storage of UDI can be implemented via either a text block with variable length or a binary large object (BLOB). The text block is suitable for keeping information which can be described as character strings. The BLOB is a data type which can hold a variable amount of data in a relational database. Thus, any operating system file such as graphics, audio, video or documents can be stored directly into the database as a BLOB in a binary format.

Representation of workflows

In order to capture data management requirements for the development of an information system, it is necessary to identify the business processes and the rules of data streams in a lab. In general, such processes can be described by various models such as Petri Net (Peterson 1981), Statecharts (Harel *et al*. 1997), TAMBIS (Baker *et al*. 1999), Regulatory Networks (Rzhetsky *et al*. 2000) and OPM (Dori 2002). However, Peleg (2002) stated that the workflow model of the Workflow Management Coalition (WfMC) (1999) is suitable for biological systems. Therefore, based on the workflow concept (Hollingsworth 1995), we define procedures for representing the workflows of biodiversity studies.

An information system is usually described in terms of business processes. Each reflects a specific subset of actions in the execution of scientific experiments. In biodiversity studies, for instance, DNA extraction and PCR amplification are considered two business processes which need to be described in form of workflows. The workflow approach in this case may be understood via four definitions as follows:

- **Definition 1:** A workflow describes the business process to be carried out in a lab, the order in which *tasks* are conducted, and the *data items* required in each task.
- **Definition 2:** A task is a data processing operation corresponding to a single unit of work performed within a workflow. A task might be a *single task* or a *block task*. A single task is a simple action, which has an atomic execution (i.e. one that cannot be divided into smaller executions). A block task is a complex action which is composed of a number of single tasks contributing to a given lab procedure. A block task is presented as a sub-workflow.
- **Definition 3:** A data item is a named data element in a given task. A data item may be an *input* or *output* element collected from any task in the workflow. An input might be descriptive information, a parameter, or an experimental protocol. An output might be an identification, an analytical result, or an output file generated from a machine or a software tool. A newly generated data item from a task should be considered an output if it is used as input in another task. But it is not required that all outputs of a task must be used elsewhere.
- **Definition 4:** The set of data items from all tasks in a workflow is termed *workflow data*. A collection of workflow data from all workflows makes up a common *data framework* which is the basis of a data model.

We model a workflow as a directed graph made up of nodes and arcs. Each node describes a task performed within a lab. Arcs connect nodes and define the movement of data from one node to the next. A transition is a directed arc in the graph between two nodes.

A workflow can be presented by using six graphical notations as shown in Figure 2. Two types of rectangles (normal and rounded) are used to depict two kinds of nodes, single task and block task, respectively. The task name is displayed in the rectangle, representing the node. Arcs are presented by arrows. Solid arrows indicate a transition between two tasks, which is executed unconditionally, whereas dashed arrows specify conditional routing, meaning that some conditions must be met before the transition is carried out. A workflow must begin from a starting point, denoted by a white circle and finish at an ending point shown as a black circle.

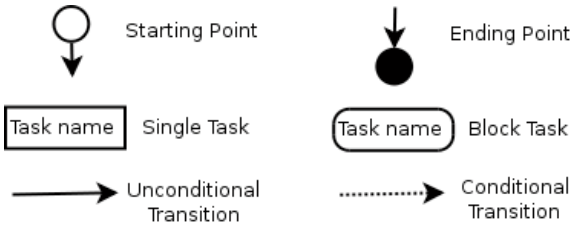

Figure 2
Graphical notations for presenting workflows

Figure 3 presents three patterns used to reflect different tasks in a lab. In the sequence pattern (Figure 3a), a task is performed after the completion of the preceding one, without any condition. The control pattern (Figure 3b) allows a transition from a task to split into multiple branches. Each is a conditional transition, which is carried out if the conditions of that branch are matched. The last pattern (Figure 3c) is used when one or more tasks in the workflow are repeated.

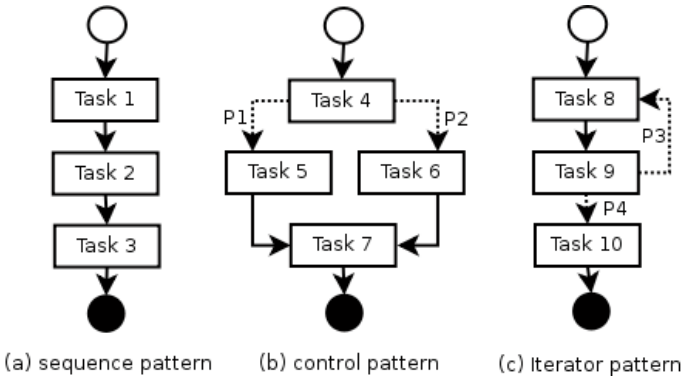

Figure 3
Workflow patterns are used to construct workflows

Each workflow consists of many data items which should be listed in a uniform way. Therefore, we use a term so-called Data Integration Table (DIT) to describe data items in a single workflow. Each DIT is created for a workflow. Table 1 is a template for creating DITs. In this template, two first columns (task, data item) show the task numbers and the names of data items. The third column (type) specifies the type of data item. It receives one of three values (C: core, E: extended, S: specific). If a data item in a task is taken from another, it will be identified with a task number in the fourth column (from).

Table 1
A template is used to produce DITs for workflows

Task	Data item	Type	From
1.1	data item 1	C	
1.1	data item 2	E	
1.1	data item 3	S	
1.2	data item 1	C	1.1
1.2	data item 4	E	
1.2	data item 5	S	

Results

In the context of biodiversity studies, workflows of DNA sequencing and microsatellite genotyping are represented in two levels. The first level is a general workflow with only block tasks. Each is described in details by a sub-workflow in the second level. All tasks in the workflows are labeled by an *x.y* pattern, where *x* stands for a workflow number and *y* is replaced by a task number within the workflow *x*.

General workflow

Basically, biodiversity studies execute a fixed number of blocks. Specifically, data stream follows a sequence of seven steps. Each step is a block task depicted by the general workflow in Figure 4.

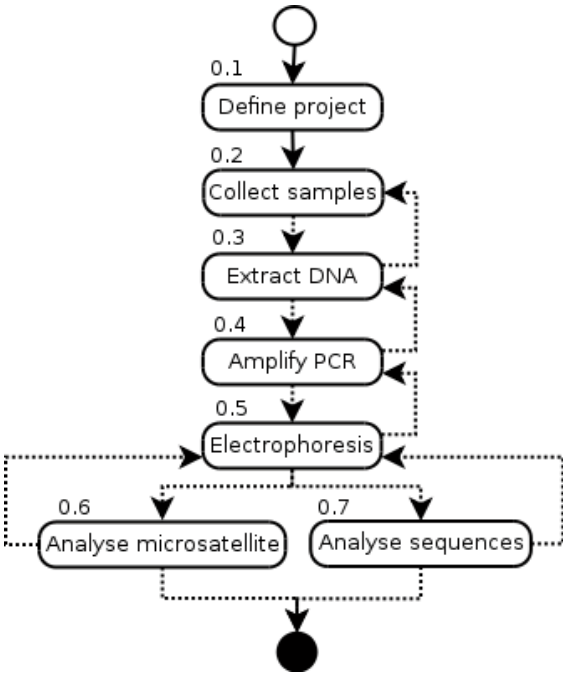

Figure 4
General workflow of biodiversity studies with seven block tasks

Each step has many data processing operations conducted in one time frame. The result of a step (output) is used as the input in the next step. Based on these features we can distinguish one step from the others to design the general workflow. In the following, each step is described and explained as a sub-workflow. Thus, there are seven workflows at the second level. Each workflow is mapped to a DIT (see Table 2 to Table 8). Our proposal for a common data framework has been submitted to three labs for evaluation. As can be seen from the last three columns in the DITs, the labs agreed with our definitions. The data items of a task are evaluated if the lab performs that task. For each data item, two symbols are used to indicate if the data item is needed (**x**: the lab requires such a data item; **-**: the data item is not needed).

Project definition

Biodiversity studies often deal with many samples collected from different genetic groups, or different localities of a certain species. A project is defined as research on a group of biological material, including original samples (e.g. blood, somatic cells) and DNA. The workflow in this step consists only of two single tasks (Figure 5.1). All data items of the workflow are given in Table 2. A project must be defined (task 1.1) before conducting other tasks. Each project has a *unique name*. Important information (e.g. objective of the project, expected results) is given in a *description*. Besides, a *keyword* used as a shortcut name and a *duration* for conducting the project are also suggested. Other details such as project manager, funding, resources, etc. may be stored in a *UDI* block. Once the project has been defined, it can start recording new samples in next step or reuse existing samples (task 1.2) from other projects. Therefore, for each sample in a project we need a data item *reused* to track if that sample is taken from another project.

Table 2
DIT for Workflow 1

Task	Data item	Type	From	1	2	3
1.1	project id	C		x	x	x
1.1	project name	C		x	x	x
1.1	description	E		x	x	-
1.1	keyword	E		x	-	x
1.1	begin date	E		-	-	x
1.1	end date	E		-	-	x
1.1	udi	S		x	x	x
1.2	project id	C	1.1	-	-	x
1.2	sample id	C		-	-	x
1.2	reused	E		-	-	x

Sample recording

Here, samples are understood as original biological material (e.g. blood, tissue), which will be used for the extraction of DNA in the next step. The workflow for recording samples has five single tasks, as shown in Figure 5.2. The DIT for this workflow is given in Table 3. The first task (task 2.1) records the origin of sample. Core data items such as *individual ID*, *species* and *genetic group* are essential information of individuals which are sampled. Instead of storing

many different data items to specify characteristic, color, shape and size of each individual, we suggest a color *photo* with a scale. In order to record a location where the individual is sampled, we propose to use global positioning system (GPS). This way, only two floating point values including GPS *latitude* and GPS *longitude* are collected. Depending on the type of individual, several extended data items such as a *description* of variety for plants (task 2.2) or *sire ID*, *dam ID*, *sex*, *date of birth* for animals (task 2.3) are needed. To ensure recording other information, we use a *UDI* block to keep all additional data items for each individual.

Table 3
DIT for Workflow 2

Task	Data item	Type	From	1	2	3
2.1	individual id	C		x	x	x
2.1	species	C		x	x	x
2.1	genetic group	C		x	x	x
2.1	photo	E		x	-	x
2.1	gps latitude	E		x	-	x
2.1	gps longitude	E		x	-	x
2.1	udi	S		x	x	x
2.2	individual id	C	2.1	-	-	x
2.2	sire id	E		-	-	x
2.2	dam id	E		-	-	x
2.2	sex	E		-	-	x
2.2	date of birth	E		-	-	x
2.3	individual id	C	2.1	x	-	-
2.3	description	E		x	-	-
2.4	project id	C	1.1	x	x	x
2.4	sample id	C		x	x	x
2.4	individual id	C	2.1	x	x	x
2.4	material type	C		x	x	x
2.4	unit amount	E		x	x	x
2.4	action user	E		x	x	-
2.4	production date	E		-	x	x
2.4	udi	S		x	-	x
2.5	sample id	C	2.4	x	x	x
2.5	storage location	C		x	x	x
2.5	vessel type	E		-	-	x
2.5	storage date	E		-	-	x
2.5	udi	S		-	-	x

Many biodiversity projects are conducted with several hundred samples. Each sample is collected from an individual of a certain breed or a genetic group. Therefore, we need the triplet of core data items *project ID*, *sample ID* and *individual ID* to manage samples within a breed or among breeds of a given project. Main data processing procedure (task 2.4) is to record the *type of material*, the *amount* or *unit* of sample, an *action user* who collected the sample and an *action date* when the individual was sampled. Details regarding the procedures of sample collection and usage may be given in a *UDI* block. The final task in this workflow

(task 2.5) is to capture information on a physical *storage location* of the samples after they are put in storage (e.g. tanks or freezers). This information is provided as hierarchical data, *possibly being different among labs. In addition to the storage location, we can store a type of vessel* (e.g. straw, tube, filter paper) which is used to contain the sample and a storage date. Other additional information such as a donor who gave the samples, costs per sample, temperatures of tanks, etc. are given in a *UDI* block.

DNA extraction

DNA extraction is typically the prerequisite for all subsequent steps in biodiversity studies. The workflow for the extraction of DNA is depicted in Figure 5.3 and the DIT for this workflow is shown in Table 4. The first task in the workflow is to prepare the samples (task 3.1). Only two data items (*project ID* and *sample ID*) are needed at this task to track which samples of a project are used in a DNA extraction. Then, DNA is extracted and purified to obtain a certain *volume* (task 3.2), which is available for polymerase chain reaction (PCR) or further studies. Each DNA should have a unique identification (*dna ID*) linked to a *sample ID*. We suggest using a *UDI* block to store other details related to procedures in this task.

Table 4
DIT for Workflow 3

Task	Data item	Type	From	1	2	3
3.1	project id	C	1.1	x	x	x
3.1	sample id	C	2.4	x	x	x
3.2	dna id	C		x	x	x
3.2	sample id	C	3.1	x	x	x
3.2	volume	C		x	x	x
3.2	udi	S		-	x	x
3.3	gel image	C		x	x	x
3.3	dna concentration	C		x	x	x
3.3	dna purity	E		x	-	x
3.3	dna id	C	3.2	x	x	x
3.3	sample id	C	3.1	x	x	x
3.3	lane	E		-	x	x
3.3	validation	E		-	x	x
3.3	action date	E		x	x	x
3.3	description	S		x	x	x
3.3	udi	S		x	x	x
3.4	dna id	C	3.3	x	x	x
3.4	storage location	C		x	x	x
3.4	storage date	E		x	-	x
3.4	action user	E		x	x	x
3.4	udi	S		-	-	x

The isolated DNA is usually checked to guarantee for both quantity and quality. This can be evaluated by using a spectrophotometer or determined by an agarose gel electrophoresis. The output of task 3.3 is *gel images* and *DNA concentrations* which may be stored along with

extended data items such as *dna purity*, *action date*. Besides, we also record information specifying samples shown up on the gel. Hence, each gel image is linked to a set of three data items (*sample ID*, *lane*, and *validation*). This information helps to retrieve the gel image which is useful to know whether the samples are valid or not. In addition, we suggest a *UDI* block for each gel image. Therefore, scientists can give additional text such as information of standards used in the gel or their ideas on the results obtained. The final task (task 3.4) is to capture information on the storage of DNA. Similar to the storage of samples, data items needed in this task are *dna ID, storage location, storage date, action user* and *UDI*.

PCR amplification

PCR amplification is a routine step in many molecular biology processes to produce many identical copies of a specific DNA fragment. The workflow, which is used to collect the data items in Table 5, is shown in Figure 5.4. There are three single tasks in this step. The first one is to prepare DNA samples (task 4.1). It relates to the retrieval and selection of DNA from the storage locations. In order to keep track of sample usage, the list of DNA samples (dna ID) amplified for a specific project (project ID) must be known. Depending on the research objective of each project, some lab work such as sample dilution, preparation of working solution, selection of PCR program, etc. are carried out. Since these lab procedures do not generate new data items, they are not considered tasks in this workflow. However, such information may be stored in a UDI block in the second task (task 4.2). An essential item in the second task is the information about markers used in the PCR. Because a multiplex PCR allows a simultaneous amplification of multiple targets on the same strand of DNA, more than one marker (or one pair of primers) should be recorded. For each electrophoresis, a unique amplification ID is required to group all related DNA samples using the same set of markers.

Table 5
DIT for Workflow 4

Task	Data item	Type	From	1	2	3
4.1	project id	C	1.1	x	x	x
4.1	dna id	C	3.3	x	x	x
4.2	amplification id	C		x	x	x
4.2	markers	C		x	x	x
4.2	dna id	C	4.1	x	x	x
4.2	udi	S		x	x	x
4.3	amplification id	C	4.2	x	x	x
4.3	gel image	C		x	x	x
4.3	dna id	C	4.2	x	x	x
4.3	lane	E		-	x	x
4.3	validation	E		-	x	x
4.3	udi	S		x	-	x

In principle, the results of PCR reactions are PCR products. However, labs do not keep these products for a long time and discard them once the final data is obtained. For that reason our data framework excludes the information on the storage of PCR products. But the details of

PCR validation are still needed (task 4.3). As the validation of DNA samples in the previous workflow, here the PCR products are also checked by an agarose gel electrophoresis. Consequently, the DIT of this workflow has similar data items as required in the previous one (Figure 5.3): *gel image*, *dna ID*, *lane*, *validation*, and *UDI*.

Electrophoresis

The PCR products obtained in the previous step (Figure 5.4) are prepared to perform the process of electrophoresis in this step (Figure 5.5). Firstly, we record the selection of DNA amplified by PCR to carry out the lab work (task 5.1). An *electrophoresis id* is also needed for each electrophoresis to group all analysed samples. For different purposes, labs may use same or different DNA sequencers (e.g. LI-COR Biosciences [Lincoln, NE, USA], ABI [Applied Biosystems, Foster City, CA, USA], Beckman Coulter [Pasadena, CA, USA]) to conduct the electrophoresis. This leads to the difference of the methods used among labs or projects. Therefore, the *purpose* and *method* of the electrophoresis (e.g. DNA sequencing by using polyacrylamide gel electrophoresis or microsatellite genotyping by using capillary electrophoresis) are extended data items in this task. There is some lab work related to the preparation of samples, for instance, creating working solutions. These lab procedures are not considered tasks because no useful data items are needed. However, if lab users require other information for such operations, we suggest using a *UDI* block here to store all additional information.

The result of the electrophoresis process is *electrophoresis product* consisting of data files, i.e. raw data. Therefore, the final task (task 5.3) of this workflow is to capture these files. Since different sequencers may generate different types of raw data (e.g. gel images, chromatogram files), a uniform storing method is needed. In this manner we also suggest using a *UDI* block to store all raw files in any format in the database. Extended data items of this task are *action user*, *electrophoresis date* and *software* which should be used to view and analyse the original raw data. Other specific information can be kept in the *UDI* block.

Table 6
DIT for Workflow 5

Task	Data item	Type	From	1	2	3
5.1	project id	C	1.1	x	x	x
5.1	amplification id	C	4.2	x	x	x
5.2	electrophoresis id	C		x	x	x
5.2	dna id	C	3.3	x	x	x
5.2	method	E		x	x	x
5.2	purpose	E		x	x	-
5.2	udi	S		-	x	x
5.3	electrophoresis id	C	5.2	x	x	x
5.3	product	C		x	x	x
5.3	action user	E		x	x	x
5.3	electrophoresis date	E		x	x	x
5.3	software	E		-	x	x
5.3	udi	S		x	-	x

Microsatellite analysis

This step deals with the handling of raw data to obtain microsatellite results. Microsatellites or simple sequence repeats (SSRs) are defined as loci where short sequences of DNA are repeated. Figure 5.6 and Table 7 describe the workflow and its DIT, respectively. First, the electrophoresis products generated from sequencers are visualized and analysed in lane analysis programs (e.g. RFLPscan [Scanalytics, Waltham, MA, USA], GeneMapper [Applied Biosystems, Foster City, CA, USA] – task 6.1). The output of these programs is scored alleles. Consequently, for each marker one *pair* of *alleles* (allele 1 and allele 2) is stored (task 6.2). Besides, a UDI block should be used to keep additional information.

Table 7
DIT for Workflow 6

Task	Data item	Type	From	1	2	3
6.1	project id	C	1.1	x	x	x
6.1	electrophoresis product	C	5.3	x	x	x
6.2	dna id	C	3.2	x	x	x
6.2	marker	C		x	x	x
6.2	allele 1	C		x	x	x
6.2	allele 2	C		x	x	x
6.2	udi	S		x	x	x

Sequence analysis

DNA sequencing is the process of determining the nucleotide order of a given DNA fragment. The workflow in Figure 5.7 depicts this analysis process to obtain final sequences. Raw sequences generated from sequencers are usually checked in alignment analysis programs (e.g. AlignIR [LI-COR Biosciences, Lincoln, NE, USA], CodonCode Aligner [CodonCode Corp., Centerville, MA, USA] – task 7.1). In some cases, these sequences need to be validated. The validated sequences are stored for subsequent analyses steps (task 7.2), whereas failed sequences are potentially redone. Thus, for each DNA sample (dna id) we store a marker name and a consensus sequence. Other information may be given in a UDI block. The data items of this workflow are shown in Table 8.

Table 8
DIT for Workflow 7

Task	Data item	Type	From	1	2	3
7.1	project id	C	1.1	x	x	x
7.1	electrophoresis product	C	5.3	x	x	x
7.2	dna id	C	3.2	x	x	x
7.2	marker	C		x	x	x
7.2	sequence	C		x	x	x
7.2	udi	S		x	x	x

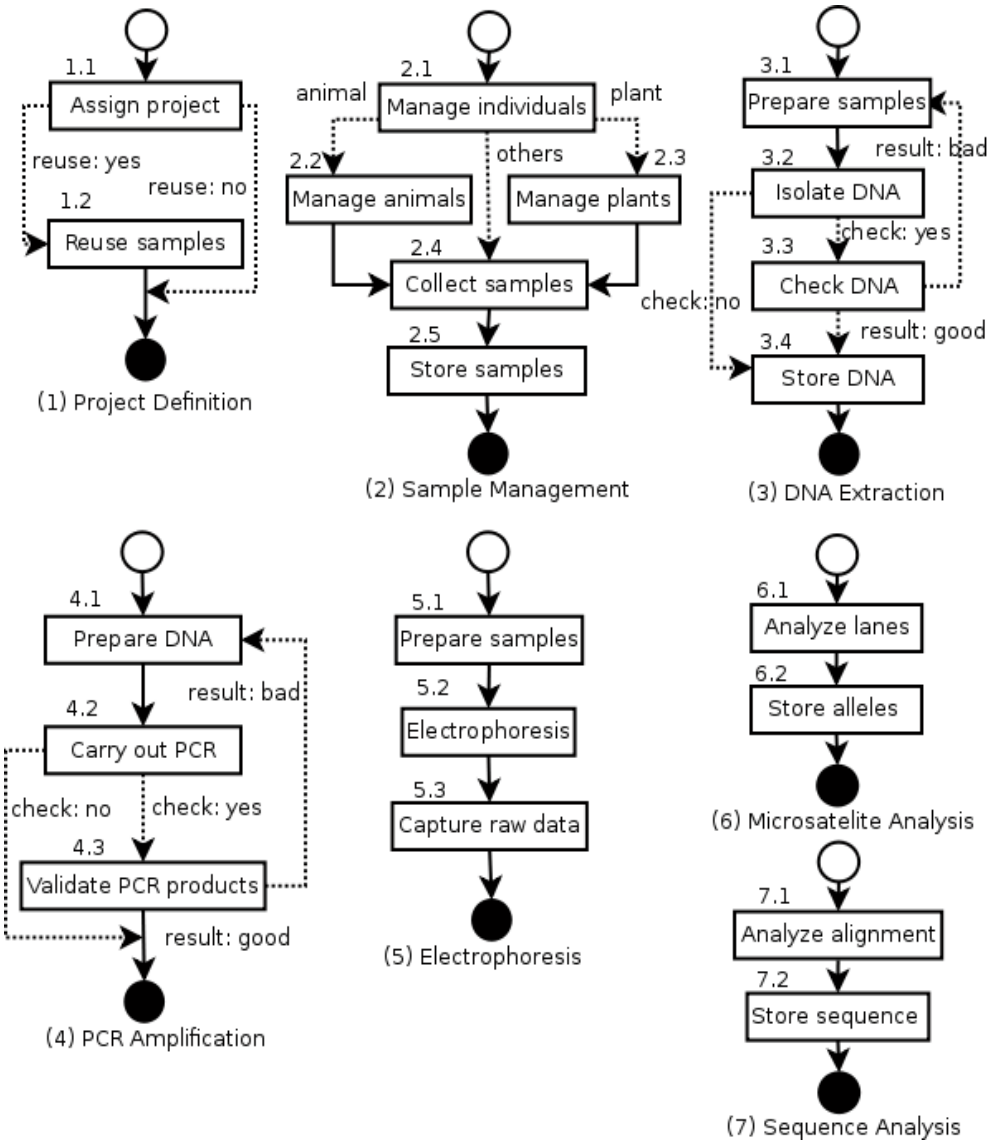

Figure 5
Sub workflows with single tasks

Discussion and conclusions

The purpose of this paper is twofold. First, it aims to promote the idea of classifying data into three groups and using UDI blocks to store all lab specific information, and second, to concretely present data streams and data items required in biodiversity studies via workflows.

Obviously, for database normalization, most data items in two groups »C« and »E« can be mapped to the properties of entities in a conceptual data model or the columns of tables in a logical data model via their names. The data framework with all items in both these

groups meets the basic needs of molecular genetics labs. Besides, the storage of additional data items as text blocks or BLOBs makes the data framework flexible to cover specific requirements in a wide range of different labs.

In addition, we also suggest to store the raw data files as BLOB in the database instead of decomposing them in different tables. The drawback is that it is difficult to search the data items inside the file. However, for archival purposes, this solution is superior since the original data files can be read by analysis software. Moreover, it does not require additional development effort to support specific future formats of data files, possibly created from new sequencer machines. Thus, the data framework can be used without modification.

The workflow approach is a useful method for describing data streams of repeatable work in which data is pipelined from a step to the other. Through the graphical representation of workflows, complex lab procedures have been simplified and modeled as understandable tasks. The workflows, which have been designed in this paper, focus on data streams of DNA sequencing and microsatellite genotyping projects. At each task, the details of data items are presented via DITs in a uniform way. In conclusion, the data framework created in this study is the basis to design a general data model in the context of data storage of biodiversity studies (Truong *et al*. 2011). The workflows and DITs have partly specified the use cases which contribute considerably to software implementation.

Acknowledgments

This work has been funded by Bundesministerium für Bildung und Forschung (BMBF, project number: VNB 03/B14). The authors gratefully thank surveyed labs for data support related to this study.

References

Baumung R, Simianer H, Hoffmann I (2004) Genetic diversity studies in farm animals – a survey. J Anim Breed Genet 121, 361-373

Baker PG, Goble CA, Bechhofer S, Paton NW, Stevens R, Brass A (1999) An ontology for bioinformatics applications, Bioinformatics 15, 510-520

Dori D (2002) Object-process methodology applied to modeling credit card transactions. Published in: Siau K (ed.) Advanced topics in database research vol. 1, IGI Global, Hershey, PA, USA, 87-105

Granevitze Z, Hillel J, Chen GH, Cuc NTK, Feldman M, Eding H, Weigend S (2007) Genetic diversity within chicken populations from different continents and management histories. Anim Genet 38, 576-583

Granevitze Z, Hillel J, Feldman M, Six A, Eding H, Weigend S (2009) Genetic structure of a wide-spectrum chicken gene pool. Anim Genet 40, 686-693

Hollingsworth D (1995) The workflow reference model. Workflow Management Coalition, Document Number TC00-1003, Draft 1.1

Harel D, Gery E (1997) Executable object modeling with statecharts. Computer 30, 31-42

Jayashree B, Reddy PT, Leeladevi Y, Crouch JH, Mahalakshmi V, Buhariwalla HK, Eshwar KE, Mace E, Folksterma R, Senthilvel S, Varshney RK, Seetha K, Rajalakshmi R, Prasanth VP, Chandra S, Swarupa L, SriKalyani P, Hoisington DA (2006) Laboratory Information Management Software for genotyping workflows: applications in high throughput crop genotyping. BMC Bioinformatics 7, 383

Liu YP, Wu GS, Yao YG, Miao YW, Luikart G, Baig M, Beja-Pereira A, Ding ZL, Palanichamy MG, Zhang YP (2006) Multiple maternal origins of chickens: Out of the asian jungles. Mol Phylogenet Evol 38, 12-19

Oka T, Ino Y, Nomura K, Kawashima S, Kuwayama T, Hanada H, Amano T, Takada M, Takahata N, Hayashi Y, Akishinonomiya F (2007) Analysis of mtDNA sequences shows Japanese native chickens have multiple origins. Anim Genet 38, 287-293

Peleg M, Yeh I, Altman RB (2002) Modelling biological processes using workflow and Petri Net models. Bioinformatics 18, 825-837

Peterson JL (1981) Petri net theory and the modeling of systems. Prentice Hall, Englewood Cliffs, NJ, USA

Rudd S, Schoof H, Mayer K (2005) PlantMarkers – a database of predicted molecular markers from plants. Nucleic Acids Res 33, 628-632

Rosenberg NA, Burke T, Elo K, Feldman MW, Freidlin PJ, Groenen MAM, Hillel J, Mäki-Tanila A, Tixier-Boichard M, Vignal A, Wimmers K, Weigend S (2001) Empirical evaluation of genetic clustering methods using multilocus genotypes from 20 chicken breeds. Genetics 159, 699-713

Rzhetsky A, Koike T, Kalachikov S, Gomez SM, Krauthammer M, Kaplan SH, Kra P, Russo JJ, Friedman C (2000) A knowledge model for analysis and simulation of regulatory networks, Bioinformatics 16, 1120-1128

Sanger F, Coulson AR (1975) A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. J Mol Biol 94, 414-448

Sanger F, Nicklen S, Coulson AR (1977) DNA sequencing with chain-terminating inhibitors. Proc. Natl Acad Sci 74, 5463-5467

Schönherr S, Weißensteiner H, Coassin S, Specht G, Kronenberg F, Brandstätter A (2009) eCOMPAGT – efficient combination and management of phenotypes and genotypes for genetic epidemiology. BMC Bioinformatics 10, 139

Truong VCC, Groeneveld LF, Morgenstern B, Groeneveld E (2011) MolabIS – An integrated information system for storing and managing molecular genetics data. BMC Bioinformatics 12, 425

Vignal A, Milan D, SanCristobal M, Eggen A (2002) A review on SNP and other types of molecular markers and their use in animal genetics. Genet Sel Evol 34, 275-305

Weißensteiner H, Schönherr S, Specht G, Kronenberg F, Brandstätter A (2010) eCOMPAGT integrates mtDNA: import, validation and export of mitochondrial DNA profiles for population genetics, tumour dynamics and genotype-phenotype association studies. BMC Bioinformatics 11, 122

Wendl MC, Smith S, Pohl CS, Dooling DJ, Chinwalla AT, Crouse K, Hepler T, Leong S, Carmichael L, Nhan M, Oberkfell BJ, Mardis ER, Hillier LW, Wilson RK (2007) Design and implementation of a generalized laboratory data model. BMC Bioinformatics 8, 362

WfMC (1999) Workflow Management Coalition Interface 1: Process Definition Interchange Process Model, Document Number WfMC TC-1016-P, Version 1.1

Chapter 4

MolabIS - An Integrated Information System for Storing and Managing Molecular Genetics Data

Citation:
Truong CVC[1], Groeneveld LF, Morgenstern B and Groeneveld E: "MolabIS - An Integrated Information System for Storing and Managing Molecular Genetics Data", *BMC Bioinformatics*, 2011, 12:425+.

Original Contribution:
Truong CVC designed the data model, implemented the MolabIS software, and wrote the manuscript.

[1]Corresponding author

Truong et al. BMC Bioinformatics 2011, 12:425
http://www.biomedcentral.com/1471-2105/12/425

BMC Bioinformatics

SOFTWARE Open Access

MolabIS - An integrated information system for storing and managing molecular genetics data

Cong VC Truong[1*], Linn F Groeneveld[1,2], Burkhard Morgenstern[3] and Eildert Groeneveld[1]

Abstract

Background: Long-term sample storage, tracing of data flow and data export for subsequent analyses are of great importance in genetics studies. Therefore, molecular labs do need a proper information system to handle an increasing amount of data from different projects.

Results: We have developed a molecular labs information management system (MolabIS). It was implemented as a web-based system allowing the users to capture original data at each step of their workflow. MolabIS provides essential functionality for managing information on individuals, tracking samples and storage locations, capturing raw files, importing final data from external files, searching results, accessing and modifying data. Further important features are options to generate ready-to-print reports and convert sequence and microsatellite data into various data formats, which can be used as input files in subsequent analyses. Moreover, MolabIS also provides a tool for data migration.

Conclusions: MolabIS is designed for small-to-medium sized labs conducting Sanger sequencing and microsatellite genotyping to store and efficiently handle a relative large amount of data. MolabIS not only helps to avoid time consuming tasks but also ensures the availability of data for further analyses. The software is packaged as a virtual appliance which can run on different platforms (e.g. Linux, Windows). MolabIS can be distributed to a wide range of molecular genetics labs since it was developed according to a general data model. Released under GPL, MolabIS is freely available at http://www.molabis.org.

Background

Recent advances in molecular genetics have led to a widespread use of molecular markers in genetic research for both animals and plants [1-3]. Particularly, microsatellite genotyping [4-6] and Sanger sequencing [7-9] are being widely used for different objectives in small-to-medium sized labs for biodiversity studies. DNA sequencing and microsatellite genotyping experiments often go through several major steps such as sample collection, DNA extraction, PCR amplification, electrophoresis and result analysis. Fundamental principles for conducting experiments are given in textbooks or technical documentation. Normally, lab users develop their own procedures, which they describe in lab protocols, to carry out lab work at each step. In other words, protocols provide essential information, such as how to prepare samples, what materials are needed, how to setup the machine, and what information to collect for workflow support, etc. for the completion of lab work. Although different labs may perform similar steps, the data processing operations at each step are not necessarily the same. Moreover, the demand for storage, use and management of data varies lab by lab. Therefore, identifying data items for data storage is essential. For the development of integrated information systems applicable to a wide range of labs, a general data model must be designed in the first phase. This data model must meet all requirements of different labs without additional programming or modification. In the second phase, the required functionality must be implemented resulting in a general software package.

We have previously developed a formalized workflow [10] and a data framework to concretely describe pipelined data processes and data items generated at each step which serves as the basis for the database design in the first phase. Accordingly, in these contributions, the

* Correspondence: cong.chi@fli.bund.de
[1]Department of Breeding and Genetic Resources, Institute of Farm Animal Genetics (FLI), Neustadt, Germany
Full list of author information is available at the end of the article

term "workflow" specifies the flow of operations (or tasks) relevant to data, not actual lab work steps. In other words, we only focus on the workflow for capturing and handling data. At each step of the workflow, we use a "data integration table" (DIT) to represent data items required in labs. Each DIT is a table with n rows and m columns where the values in the columns of each row specify names, data types, data sources and requirements of surveyed labs, respectively. The collection of these DITs forms a data framework which helps us to construct the general data model for developing MolabIS. The details, which focus on the construction of DITs as well as the methodology for building the formalized data framework, will be presented in another contribution.

Data handling in molecular biology labs

The challenges that small-to-medium sized labs face can be classified into five major issues. First, searching and keeping track of data is often inefficient, since heterogeneous data, possibly from different sequencers, is stored and managed in a non-standard way. Each scientist has her or his own way to handle data. Often, there is no naming convention among scientists for data objects such as individuals or samples. Second, it is difficult to share and merge data generated by different persons, because data is isolated among scientists and projects. In practice, data is often scattered and stored in inconvenient formats. Some information may be stored in paper lab books, whereas other data are kept in file systems. Third, due to the lack of a centralized database, making reports becomes difficult for project managers, because too much time has to be spent on combining data sets from various sources and locations. Fourth, sometimes data cannot be found and is thus lost. This problem is most prominent in labs with short term lab users like master or doctoral students. Typically, they come to the lab with their samples and leave the lab with their data. Fifth, scientists often spend much time on manually preparing and converting data. In order to start lab work such as PCR amplification or electrophoresis, a scientist has to know the availability and physical location of samples. This information is often found in a paper lab book, which may be difficult to retrieve. In addition, conversion and compilation of data for further analyses is carried out manually, which is, both time consuming and prone to error. Most of these challenges are often prominent in labs conducting biodiversity experiments, since sharing and synthesis of data among projects are regular incidents.

Requirements

To address the above challenges, we developed a proper information system for long-term data storage. It comprises essential tools to handle, retrieve, report and convert data effectively with a focus on biodiversity experiments. Such an information system must meet specific requirements as follows:

R1: The information system stores and manages sequence and microsatellite data of different projects in small-to-medium sized labs conducting Sanger sequencing and microsatellite genotyping experiments.

R2: It supports the management of individuals from which samples were derived, including their classification into species and breeds or varieties.

R3: Sample management is provided to keep track of all kinds of material (e.g. blood, tissue) from different projects collected by different users. The sample storage scheme is suitable for any physical storage location of samples in different labs.

R4: The information system provides functionality for managing the workflow and the traceability of samples in lab procedures. It allows tracing lab work such as DNA extraction, polymerase chain reaction (PCR), PCR validation, and electrophoresis to capture all original data from possibly different machines.

R5: The information system supports basic functionality (searching, viewing, retrieving and modifying) and the import of large amounts of samples, sequences and microsatellites from external files. Raw data received from different architectures of sequencers can be stored and retrieved in a uniform way.

R6: Ready-to-print reports can be generated easily to provide data and statistics of a certain project or an entire database.

R7: Sequences and microsatellites (final data) can be converted to various data formats for further analyses.

R8: The information system is a multi-user system which supports security and access control.

R9: The software package runs on different platforms (e.g. Linux, Windows) with a simple installation procedure which allows users with no experience in programming and database management to setup and use the system. The software is freely available to be used, distributed, and modified without restrictions. Therefore, open-source software, e.g. under the GPL license, is preferred.

R10: Migration of data from previous projects is supported by the software package.

Existing information systems

In recent years, biologists, bioinformaticians and computer scientists have spent much effort to confront the challenges of storing and managing heterogeneous data in a uniform way [11]. Therefore, a whole class of software systems has been developed to support lab work, appropriately called Lab Information Management Systems or LIMS. It has to be noted that there are many

types of labs with different requirements for data storage and management. Accordingly, LIMS developed for a chemistry lab will support very different work than a LIMS required in a molecular genetic lab. In the latter class, a number of LIMS developments have been reported. Most of them focused on the storage and management of processed data including microarray [12-14] and proteomics data [15-17]. Wendl et al. [18] developed an information system to keep track of sequencing workflows, but it does not support collecting information on individuals and microsatellite data. In 2006, a group of researchers developed AGL-LIMS [19], an open source information system for genotyping workflows which meets some of our requirements. As it focuses on microsatellite data in plants, sequencing is not supported. Further, the management of individuals, original samples along with the physical storage places are not considered. Recently, some database applications were devoted to the management of both Single Nucleotide Polymorphisms (SNP) genotype data and phenotype data [20,21]. Additionally, Weiβensteiner et al. extended their system developed in 2009 [21] to enable the import and storage of mtDNA and STR (Short Tandem Repeats) data [22]. In 2010, Ducan et al. also provided an open source web application to enable researchers to store, organize and retrieve their sequence data [23].

In general, the common objective of these information systems is to provide means for lab users to keep their data in-house and extract data for further analyses. However, they often aim to capture raw data from a specific platform [20], or import only final data, while ignoring raw data [22,23]. Most of them do not support the management of individuals and traceability of samples in lab procedures. Some systems [21-23] do not provide a solution for documenting lab data.

Since available information systems are designed in a specific context of a lab, installation and use in other labs is usually a challenge. To the best of our knowledge, there is no LIMS available, which meets all requirements stated above. We have therefore designed a general data model for labs conducting Sanger sequencing and microsatellite genotyping. In this paper, we present the design, implementation and features of MolabIS, an integrated information system for storing and managing sequence and microsatellite data in molecular genetics labs with a focus on biodiversity experiments.

Implementation

Database design

The first step in database design is the definition of a data model. In order to build such a data model, we need to know (1) differences in data streams in labs, (2) data types spawned from those data streams, (3) what data items should be stored at each step, and (4) how lab users use and retrieve their data. Figure 1 shows the conceptual database structure of MolabIS in form of the Entity-Relationship diagram (ERD) [24] using Crow's Foot notation. Specifically, the database structure could be divided into three groups of closely linked relations (tables). The first group consists of five tables (**codes**, **unit**, **contacts**, **blobs**, and **protocols**) which are used to store initial data, information on lab users and experimental protocols. The **codes** table keeps the references of foreign keys in the information system. Instead of using many tables to store foreign keys of different types, we grouped them together in one table. A column called "class" in the table **codes** stores classes of foreign keys such as SPECIES, or BREED. Table 1 lists 14 such classes used in MolabIS. Typically, each class is a drop-down list in the data entry forms of MolabIS. Each value in the class (a code) is a data item from the drop-down list. Therefore, whenever a user wants to choose a data item, which is not available in such drop-down lists (e.g. species), he or she should insert a new code for the corresponding class. Two tables **unit** and **contacts** manage all contacts stored in the database. By storing the content of files as binary large objects (BLOBS), all lab protocols are managed in the database via two tables **protocols** an **blobs**.

The second group with five tables (**organisms**, **transfer**, **storage**, **samples**, **storage-samples**) manages data on individuals, samples and DNA. The combination of two tables **organisms** and **transfer** allows us to store the detail of all individuals of any species and breed or variety. It also helps to accept any external identification system of animals or plants. Tracking of samples is conducted with the triplet of **samples**, **storage**, and **storage-samples**. Sample storage is managed by a five level hierarchy, creating a storage tree (see Figure 2, explained in detail later), in which each location has a single parent (upper location) and many children (lower locations). In relational databases, this data structure is organized in a single table with three columns "storage-id", "storage-name", and "parent-id" as in the **storage** table.

The last group consists of several tables which, deal with tracking the workflow. The collection of samples and the extraction of DNA are managed in tables **sample-collection** and **dna-extraction**, respectively. In addition to storing information on DNA, the **dna-extraction** also saves the traces of the original samples extracted. The details of PCR amplification and electrophoresis are recorded in the tables **pcr-amplification**, **pcr-markers**, **amplified-samples** and **electrophoresis**. Two tables **validation** and **gel-images** are used to store the information on the validation of DNA or PCR products and the content of gel images. Final data is stored in the two tables **sequences** and **microsatellites**.

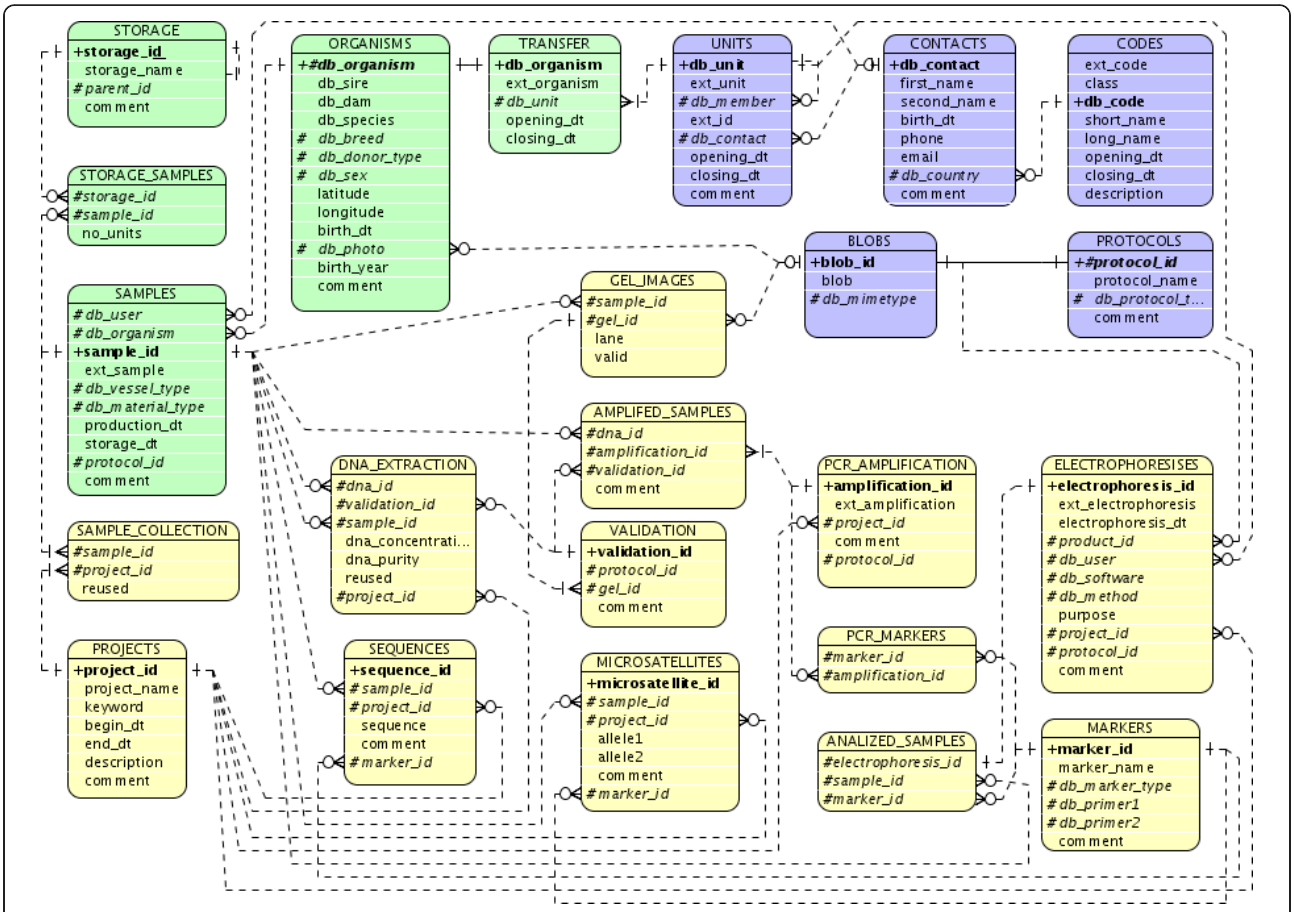

Figure 1 Entity-Relationship diagram of MolabIS. Entity-Relationship diagram using Crow's Foot notation presents the conceptual data structure used in MolabIS. Entities and relationships are represented as boxes and lines between the boxes, respectively. The database structure consists of 23 tables presented in three groups (three different colors). To simplify the complexity of the data model, foreign keys which are linked to Codes and Protocols are not shown.

Table 1 Classes in the codes table

#	Class	Description
1.	BREED	breeds of animals or varieties of plants
2.	COUNTRY	countries of users or contacts
3.	LANGUAGE	speaking languages of users or contacts
4.	MARKER_TYPE	types of molecular markers
5.	MATERIAL_TYPE	types of biological materials
6.	METHOD	electrophoresis methods for sequencing
7.	MIMETYPE	types of file extension
8.	PRIMER	names of PCR primers
9.	PROTOCOL_TYPE	types of experimental protocols
10.	PURPOSE	sequencing or genotyping
11.	SEX	genders of individuals
12.	SOFTWARE	software tools are used to analyze data
13.	SPECIES	species of individuals
14.	VESSEL_TYPE	types of vessels for storing samples

The codes table provides fourteen classes to keep the references of foreign keys. Each class has many different values. The values are used to make drop-down lists in the data entry forms.

In order to derive a general data model, two important points have been considered. First, the data model allows for storage of different data types of original data regardless of the hardware variations of sequencers. The database was designed on an abstract level to accept any type of raw files, for instance, gel images of a gel electrophoresis, or chromatogram files of capillary electrophoresis. Instead of using many different tables to serve different data types, all raw data files are stored as BLOBs in a single table. Second, the data model only comprises elements which are at least in principle available for every species, sample type, and lab. Other more specific elements can be stored in text blocks and BLOBs. As a result, the data model can be applied without customization to capture data of any species, breed (or variety), biological material type and hierarchical sample storage scheme.

Application architecture

MolabIS is an integrated information system which is developed on the basis of APIIS [25], a framework for

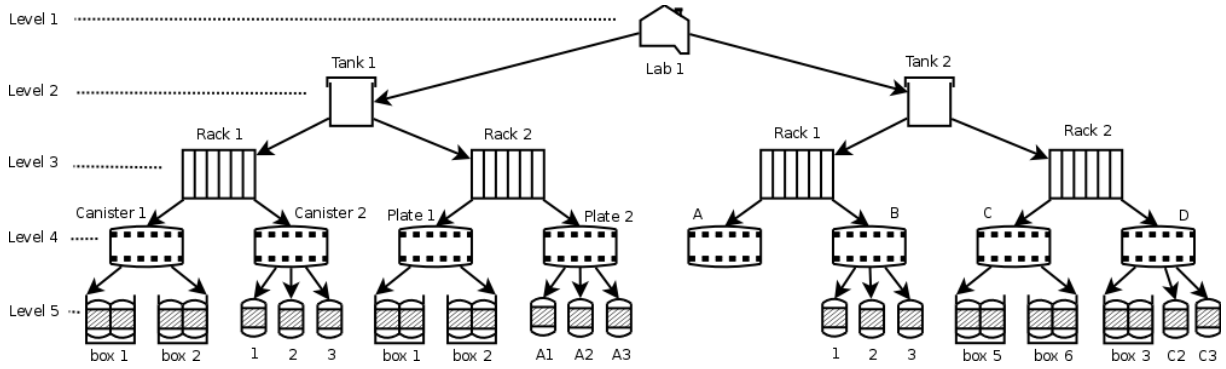

Figure 2 Management of sample storage. A five level storage schema is used to manage the location of samples. It is constructed as a tree model. The highest level (level 1) is the storage location. The lowest level (level 5) is the sample storage. The middle levels (level 2, level 3 and level 4) may be different among labs. The labels in all levels are defined by users.

developing adaptable platform independent information systems. It is a web application based on a three-tier client/server architecture (see Figure 3). On the client side, end-users from any computer in the local area network (LAN) can interact with the system to carry out all activities via a standard web browser (e.g. Firefox, Internet Explorer) on any operating system. No additional software packages or programs need to be installed on the client machine. The incorporation of Web 2.0 technologies such as Ajax [26] makes web interactions simpler and more effective. The menu bar helps end-users to easily navigate all web forms. Web layouts and dynamic interactions are controlled by Javascript, CSS (Cascading Style Sheets) and Prototype (an Open Source Javascript Framework) [27] to create an easy-to-use graphical user interface (GUI).

At the data tier, Postgres [28], an open source database management system (DBMS), is used to store application data and handle all data transactions. The application tier requires an Apache web server [29] running under the Linux operating system. On the top of APIIS [25], the MolabIS controller is central to the application tier to process user requests and to communicate with other components. The application source code is written mainly in the Perl programming language [30]. Many Perl modules, which are available on CPAN [31], are used to implement different functionalities in the system. The APIIS meta layer between the web server and the database server controls data transactions and error handling. Many open source software packages are integrated in MolabIS. Particularly, HTML::Templates [32] and CGI::Ajax [33] are two Perl modules used to produce and handle dynamic web forms. Since our objective is to have a uniform layout, form templates are all designed in the same manner. They are compiled by the MolabIS controller to create web pages, which are sent to the web browsers. The labels of form elements in each form template are variables translated from a text file in ASCII format, allowing easy changes of labels on the forms. The forms are designed so that a large number of data records (e.g. samples, DNA) can be entered, imported and processed. Because of its dynamic length, the form has to be broken down into smaller units called sub forms. A data

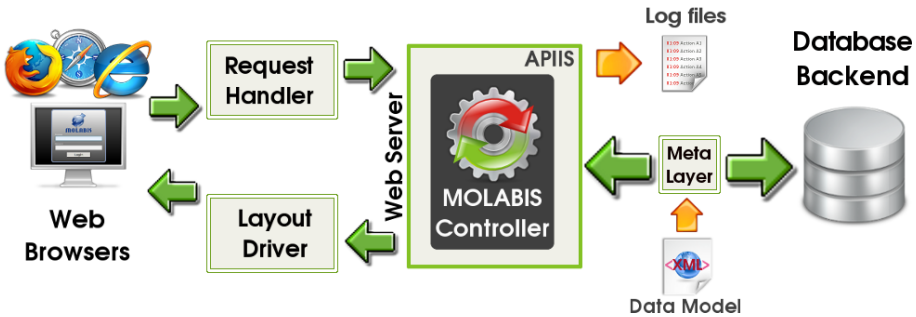

Figure 3 Application architecture. MolabIS is based on a three-tier client/server architechture including presentation tier, application tier and data tier.

buffer is implemented on the server to ensure the temporary storage of data of sub forms before they are submitted to the database.

As an APIIS application, the database of MolabIS is created from a XML (eXtensible Markup Language) [34] schema called "model file". The model file also defines a set of business rules for each table in the database. These business rules are checked at the meta layer in the APIIS framework to guarantee atomicity and consistency [35].

We selected an automatic report generation solution in JasperReports [36], an open source reporting library written in Java, to make ready-to-print reports in PDF format. It is integrated into the MolabIS controller with the assistance of the Inline::Java package [37]. JasperReports templates in XML were designed under iReport [38], an intuitive and visual report editor for JasperReports. These templates can be customized and checked independently without affecting the application code. Further, BioPerl [39] was used to support converting sequence data to a number of specific formats.

Security

The information system must provide mechanisms for user authentication to protect data from unauthorized accesses, according to the design requirements. Since users may play different roles in the system, they should accordingly be granted different rights for the utilization of the system and its data. The system controls the access of a user to functionality and data once he or she logged in successfully through "user roles". Each role is a definition of a group of access rights to determine which part of the program is hidden or shown. They also define which part of the database can be accessed and modified by the end-user. In our application, user roles are considered on both levels of system and database to assign proper tasks. Therefore, after a user account is created it has to be granted one "user role on the system tasks" (SR) and one "user role on the database tasks" (DR).

There are four SRs corresponding to four kinds of users. Each SR in this case is assigned a given number of system tasks depicted in Table 2. While the management of SRs handles access rights for different functions or modules of the application, the management of DRs is responsible for checking all activities related to the content of the database. Table 3 lists five DRs along with expected data access rights.

Sample tracking and management

Often sampling individuals (animals or plants) is the first phase of molecular genetics projects. Here we use the term "sample" to imply biological material, such as blood, semen, oocytes, embryos, somatic cells, or tissue from which DNA is extracted. Sample management allows recording three blocks of information: origin of sample, sample information, and the storage location of the sample.

The first block records data of individuals from which the samples are collected. Here, samples from any species and breed (or variety in plants) are accepted. The second block specifies the sample itself. A sample is collected from a certain type of biological material on a given date by a given person. Different types of biological material result in different types of vessels and different storage units (e.g. volumes of fresh blood in vial, units of dried blood on filter paper or weight of tissue sample in a tube). The final block describes when and where the samples are stored.

Sample storage is based on the storage facility and infrastructure of each lab. Therefore, our storage management system is designed to handle physical storage in a general way by providing a five level hierarchy. This flexible storage scheme is also used to manage the location of samples in national genebanks [40] and is also used for storing DNA in MolabIS. Normally, the highest level (level 1) is used for the storage location (e.g. labs, rooms). The lower levels could define various storage facilities (e.g. tanks, shelves, racks, canisters, etc.), while the lowest is the sample storage level in which the samples can be located by sequential search. Figure 2 is an example for defining the sample storage in a small lab, where all sample containers are kept in one place. It is a

Table 2 User rights on system functionality

	User role	(a)	(b)	(c)	(d)	(e)	(f)	(g)
1.	User administrator	•						•
2.	Lab manager			•	•	•	•	•
3.	Scientist		•	•	•	•		•
4.	Visitor			•	•	•		•

Each row defines access rights to seven functional blocks (a: manage users, b: use workflow, c: update data, d: generate reports, e: export data, f: administrate data, g: get help). *User administrators* can add users to or remove users from the system. They can update the data and grant new roles to existing users. *Lab managers* can use most of the functions in the system except data entry via the workflow. *Scientists* can deal with the workflow for data entry and use other functions of the system except the administration of common data in the lab. *Visitors* can use some functions such as viewing data, generating reports, converting data and reading helps. However, they are limited to work with the workflow and the administration.

Table 3 User rights on database manipulation

#	User role	Rights
1.	Read	access to application data
2.	Write	read and update application data
3.	Delete	remove application data
4.	Manage User	access and modify data related to users
5.	Full right	all of the above rights

Defining user roles on database tasks.

storage tree where each node at each level can have multiple sub-nodes in the lower level. Each leaf node is associated to either a box of vessels or a single vessel. In such labs, we may need only four storage levels (2 to 5) to keep track of samples since there is only one node as the root of the tree in the first level. This scenario can be extended easily for large labs where samples are physically stored in different places.

Since relational databases are not well suitable to store hierarchical data, we used a tree structure to model the storage of samples in a single table (see the **storage** table in Figure 1). Technically, this helps us to take advantages of tree search algorithms for easily implementing the functionality of sample retrieval such as searching a certain sample, listing samples in a level, printing a single path of storage places.

Data migration

One of the challenges for setting up a new information system lies in transferring large amounts of historical data collected and stored over the years to the database, prior to loading the new data into the database. Data migration is the process of transferring data from external data sources to a new database. This work can be done in either a visual loading mode or a batch loading mode. In the visual loading mode the user can employ a graphical interface to browse data from file systems, select proper data, enter related details and load everything to the database. This mode is provided in most of the information systems, and here MolabIS is not an exception, allowing this process to be carried out under the workflow. However, for large sets of data, this is time consuming, because data entry must be done manually step by step. In this case, the batch mode is more efficient. Instead of having many separated loads done manually in the visual loading mode, a big load can automatically be executed in the batch loading mode. This feature sets MolabIS apart from other information systems.

Results

MolabIS has been implemented as a web database application, written in approximately 40,000 lines of source code (Additional File 1). The main graphical user interface provides five different modules (Figure 4) which can be accessed from the navigation menu. All functionality has been developed to meet the requirements listed above. It provides essential tools for collecting data effiectively, searching and retrieving results easily, and making reports and extracting data quickly. The following list demonstrates the major features of MolabIS.

Data capture and storage

Data of very different formats (text, numerical data, images and archives as binary data) from the primary, final and descriptive data is stored in the central database, resulting in a transparent data handling

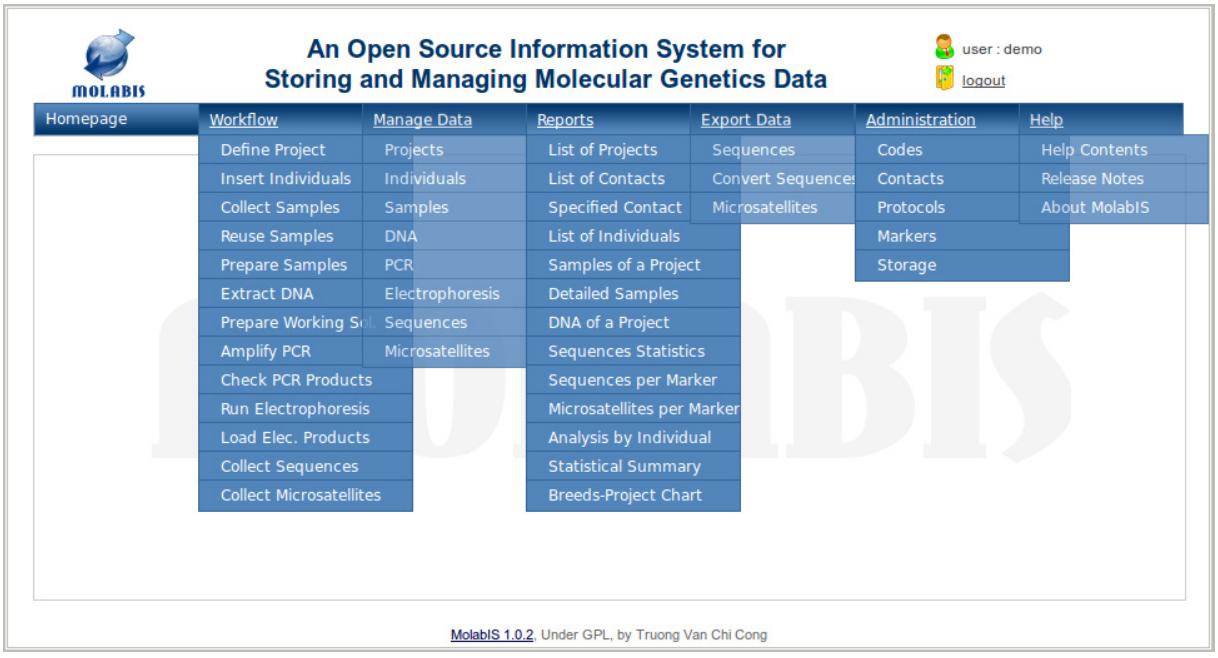

Figure 4 Main user interface. MolabIS provides five different modules which can be accessed from the menu bar in the main interface. From left to right, these modules are *Workflow*, *Manage Data*, *Reports*, *Export Data* and *Administration*. Each module consists of many sub-modules which allow users to communicate with the system via web forms.

independent of the data types. Instead of keeping files uploaded from the web browsers in the file system located in the web server all files are stored in a relational database as BLOBs. This approach avoids broken file links and storing many backup copies of data files. MolabIS allows lab users to store complete data sets of their projects: it captures both raw data and final data, as well as details of data operations and stores everything in a central database in a compact, coherent and uniform way. Figure 5 shows the data flows of the two methods supported in MolabIS for collecting data efficiently.

Workflow

Supporting the lab workflow is an important feature, as it allows users to easily keep track of their lab work and update their data in the database. Under the workflow (the left side of Figure 5), a scientist starts a project and then, step by step, interacts with the system to update the data until the project is finished. It is worth noting that data can be pipelined from one step to the next in the workflow. For example, samples exported in the step "Prepare Samples" in a spreadsheet format can be imported in the following step "Extract DNA" of the workflow. At each step, users are provided a web form or a sequence of sub-forms, for data entry. Forms are optimized for filling in data quickly (see Figure 6 and 7 for examples of two steps in the workflow).

Batch loading of historical data

In order to support data migration, "MLoader", an automation tool for bulk loading of historical data from previous projects has been developed. MLoader is a command line script written in Perl. It can be invoked at the back-end to import large datasets into a MolabIS database. All historical data must be available in electronic form to be accessed by the script (see the bottom right in Figure 5). In order to execute the script, a user

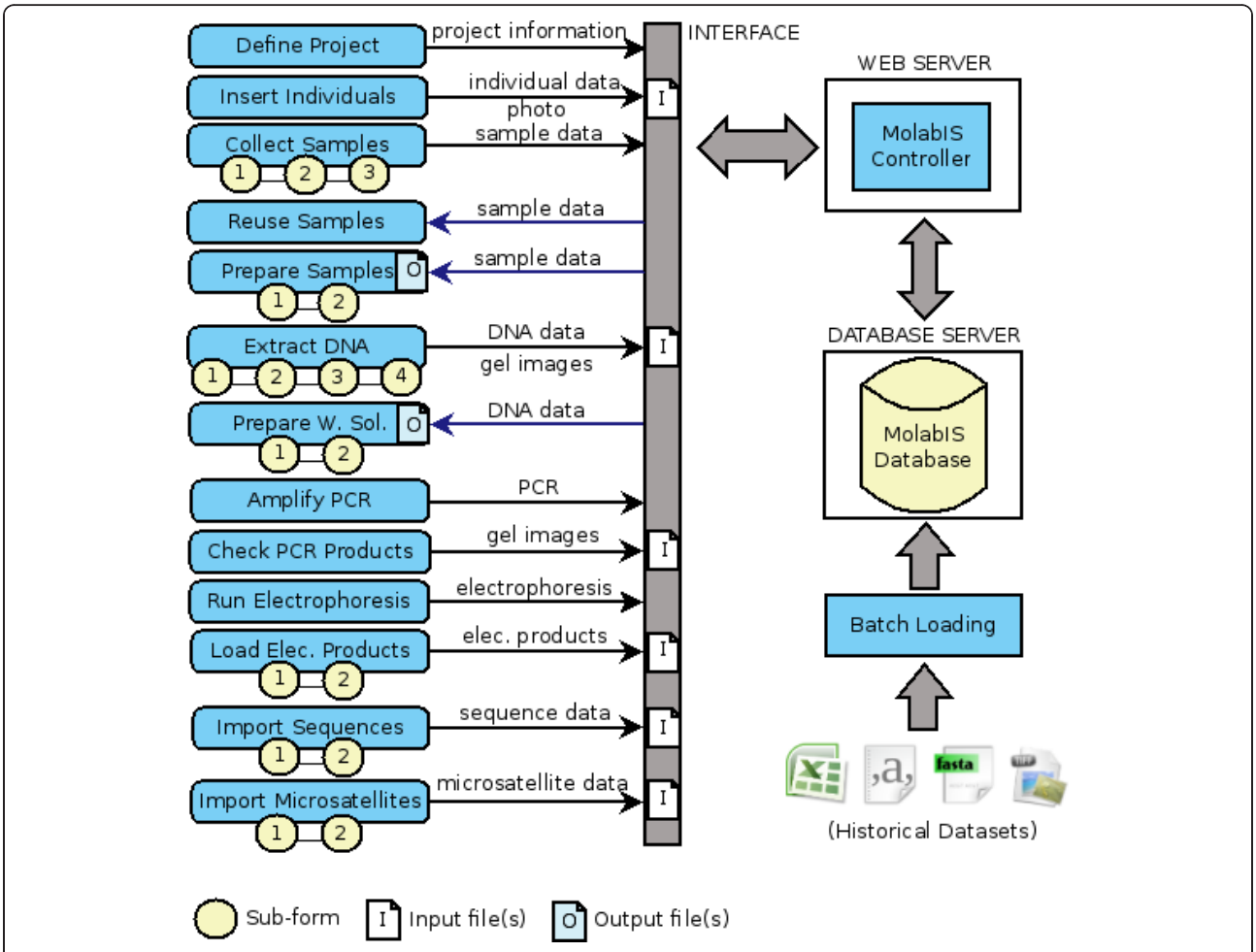

Figure 5 Data flows for capturing data in MolabIS. Data can be entered into the database through either a series of web forms under the workflow or a batch loading mode. The former is suited for inserting data of new projects while the latter is often used for data migration. The workflow feature of MolabIS enables users to insert or import a large number of data records via dynamic web forms. A form can consist of a sequence of sub-forms.

must supply parameters and data spreadsheets. All parameters are indicated in a configuration file which is made up of file records (each record is a name/value pair). It means that the user needs to declare what kind of data should be imported into the database. MLoader provides different options for loading part of or all data of a project (e.g. loading only information on individuals, importing samples and final data, importing samples with both raw and final data, importing only final data). To prepare data spreadsheets, a user may fill in empty templates, which are predefined in a given format. The spreadsheets can be supplied in XLS, CSV, or ODS format.

Data management

MolabIS not only keeps track of the workflow to capture and store different data types but also provides structured data handling capability i.e. it allows users to search for data across all projects, get back both raw and final data and modify any type of data stored in the database.

Search

Search functions are applied in the same manner for all web forms found under "Manage Data" and "Administration" in the main interface (Figure 4). A criteria based search mechanism is used, which allows the user to specify the criteria to be used in the search. Therefore, the search results can be extended or narrowed easily. Search results can be sorted according to any given field.

Data retrieval

In MolabIS, data objects are stored in a coherent manner, making data tracking much easier. The samples are considered the central entry point in the tracking model. Through the relationships among associated data objects as depicted in Figure 8 the system can locate related data. For instance, using the sample ID the user can retrieve information such as details on the sample itself, information on the individual from which the sample was collected, the storage location of the sample, details on the DNA extracted from the sample, raw data relevant to the sample, and final sequences obtained.

Data modification

MolabIS allows unrestricted data modification; lab managers can change any data field for codes, contacts, protocols, markers, storage places of samples in the lab. Scientists can update or delete all data objects stored in the database including individuals, samples, DNA, PCR amplifications, electrophoresis, sequence and microsatellite data of a project.

Generating reports

MolabIS creates ready-to-print reports in PDF format based on user specified parameters. With a few mouse clicks, users can download PDF files to their computers. Thirteen predefined types of reports have been developed in MolabIS (see the list under the menu "Reports" in Figure 4). The system can provide lists of projects, contacts and individuals. It can make reports about information on samples or DNA, along with storage locations for a given project. Besides, statistical reports for sequences and microsatellites can be done for a particular marker, a certain project, or the whole lab. MolabIS also allows users to generate a report to sum up the data volume in the entire lab or make a chart of sample distribution of a project. Since the reports are based on templates, developers can easily modify the predefined types of reports.

Exporting data

A further important feature of MolabIS is the export and conversion of final data to various formats required as input files for subsequent analyses, which is particularly useful for molecular labs working in the analysis of biodiversity.

Converting sequence data

Different analysis tools expect different data formats. This requires scientists to convert sequences from a given format to another. MolabIS offers a tool to extract sequences stored in the database and to automatically convert them to various formats. The current version of MolabIS supports conversion of sequences to seven data formats: FASTA, NEXUS, PHYLIP, MEGA, MSF, PSI-BLAS and PFAM (Figure 9). Furthermore, the system can export sequences collected from different projects into a single file, which is available for download as required for instance in phylogenetic analyses. As an additional service, users can have their uploaded sequences in FASTA format converted to other formats by MolabIS.

Converting microsatellite data

Microsatellite data is frequently stored as a matrix in which rows represent samples and columns markers. Many of bioinformatics tools such as Microsatellite Toolkit [41] (an add-in utility for Microsoft Excel) require diploid or haploid microsatellite data as input files. Preparation of these input files may be tedious. Here, MolabIS helps by extracting microsatellite data of samples from different projects and by exporting all to a single file. It allows the user to select one of three types of data formats (one-column diploid, two-column diploid, one-column haploid) for exporting. In addition, the user can choose Excel or CSV (Comma Separated Values) as the file format of the output. This process is depicted in Figure 10.

Performance and scalability

By using Postgres, MolabIS obviously meets the requirements regarding time and space complexity mentioned

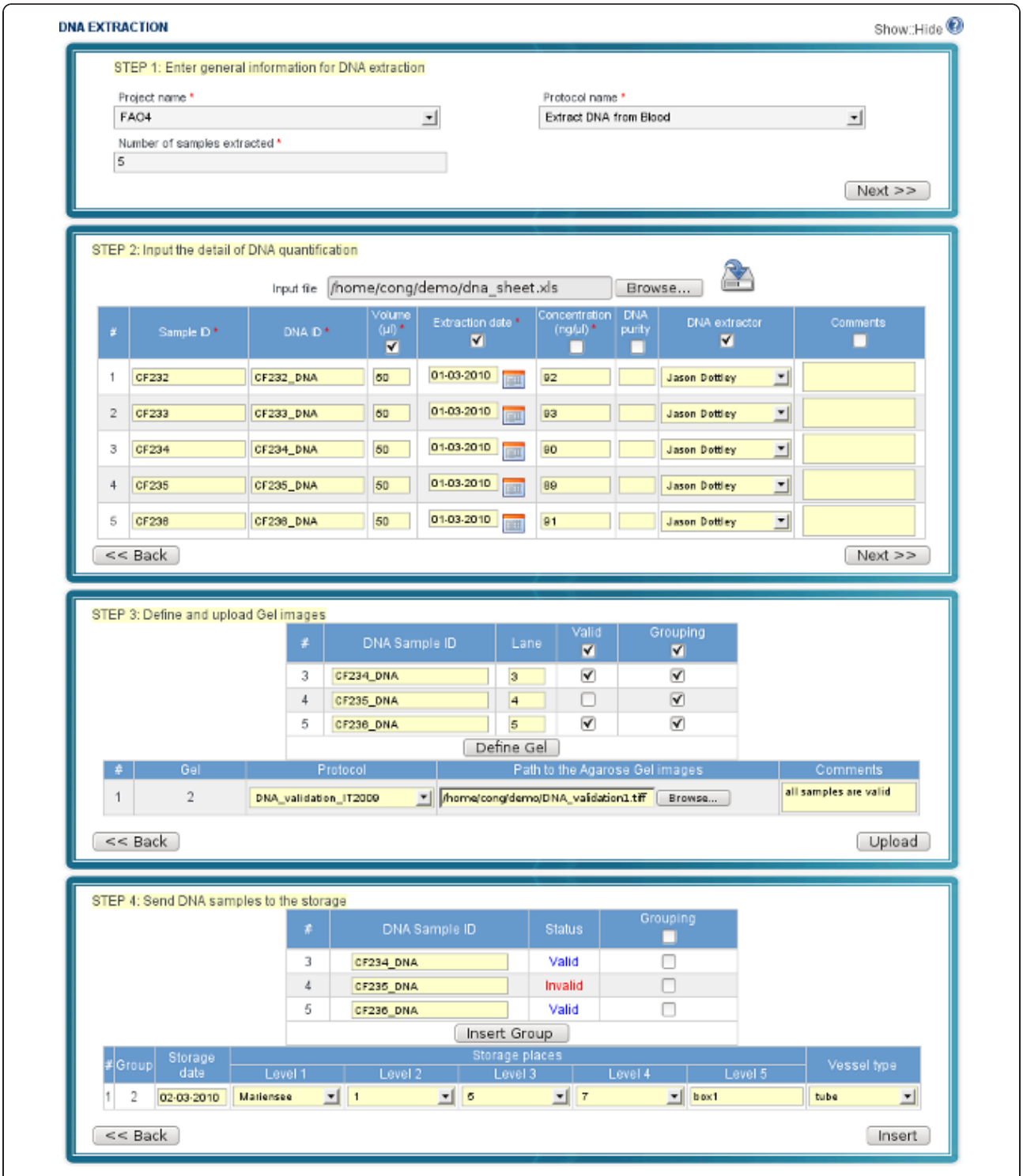

Figure 6 GUI for uploading DNA samples. The screenshot presents a sequence of web forms for inserting DNA samples into the database. Step 1: identify the number of samples which will be uploaded (e.g. 5); Step 2: enter data manually for all samples or upload data from the spreadsheet; Step 3: provide information on DNA validation and upload gel images (this step can be skipped if DNA samples are not checked); Step 4: select the storage locations of DNA samples.

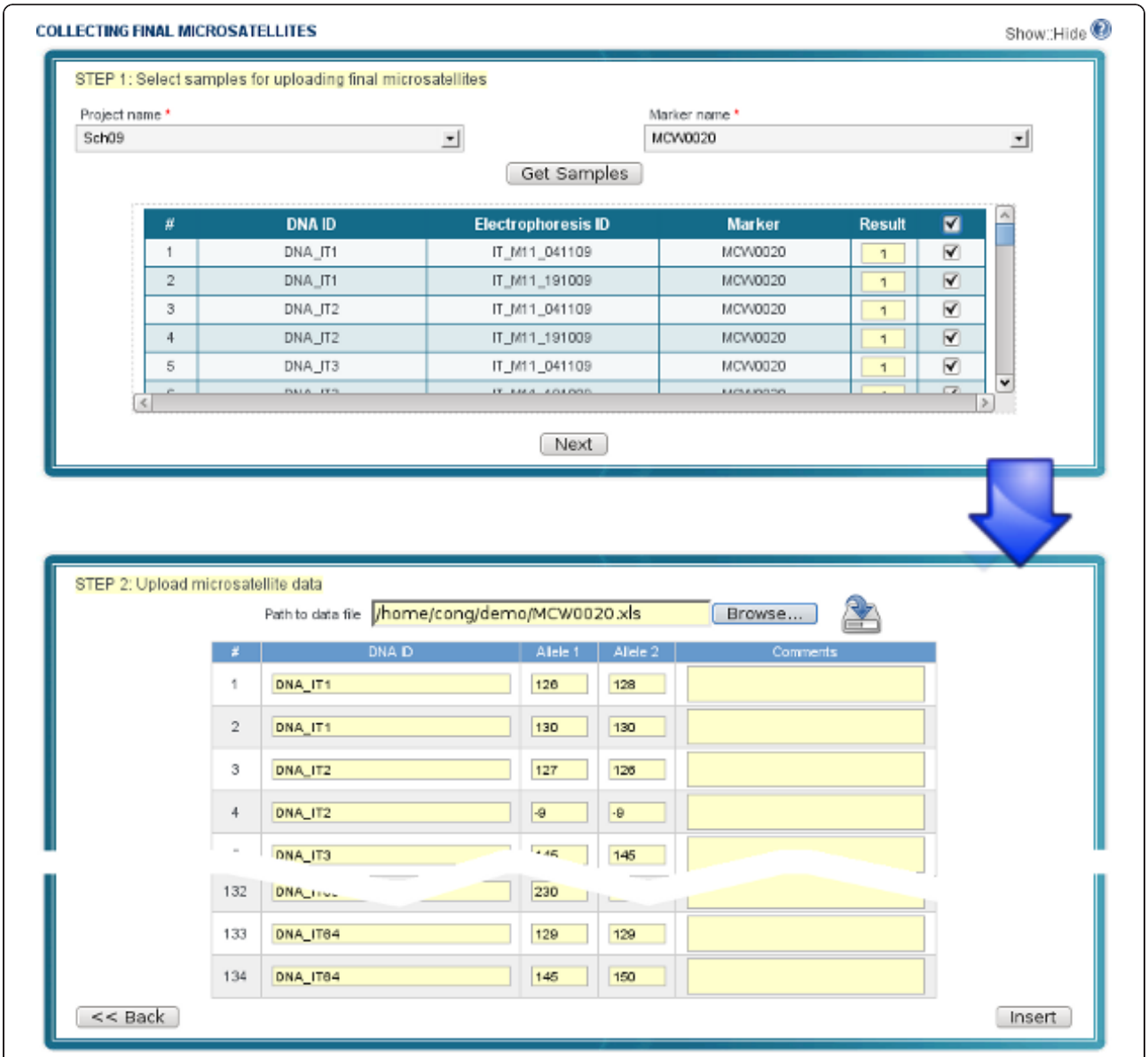

Figure 7 GUI for importing microsatellite data. The screenshot shows an example of improrting microsatellite data for 134 samples of a project. Step 1: identify samples of a project and select markers; Step 2: import microsatellite data from a spreadsheet and upload to the server.

in [21]. It can store large amounts of data and is only limited by the hardware configuration of the server. The software has been tested to ensure that it can be used by multiple users at the same time in a LAN, as well as the Internet. MolabIS runs without performance issues even when used by 10 simultaneous users.

To evaluate the performance and scalability of MolabIS, we have done three tests on three databases but with different sizes (1,000, 10,000, and 100,000 records of samples, respectively). The tests were conducted on a computer with an Intel(R) Core(TM) i5 2 × 2.30 GHz processor and 6 GB of RAM, running Kubuntu 11.04 and using Postgres 9.0. All tests used the same test cases, which are typical queries in a production mode. For each test, a test case was executed and benchmarked ten times at the front-end to calculate the mean response time. The results are reported in Table 4 showing response times in the order of seconds, thereby allowing the users to rapidly interact with the system. As expected, the response times are independent of the size of the database indicating that MolabIS scales well. Indeed, the differences in the response time among tests are insignificant (less than 0.30 seconds for each test case).

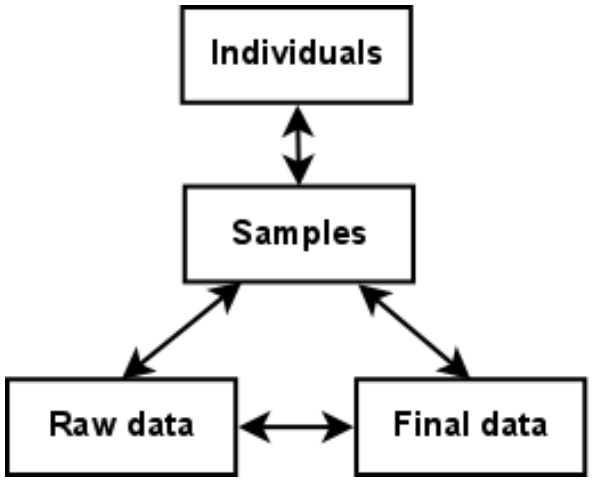

Figure 8 Data association. Relationship between associated data objects for keeping track of samples, and results.

Discussion

MolabIS was developed to overcome the challenges of molecular genetics labs in the context of data management as defined in the requirement section. In the following, we summarize how MolabIS addresses the requirements listed in the section "Background".

R1: While other information systems are often designed to collect data of either DNA sequencing projects or microsatellites genotyping projects, MolabIS is the only system to support both.

R2: MolabIS can manage information on individuals in plants and animals from any species and breed.

This feature is not supported in other information systems.

R3: The functionality of sample management in MolabIS is considered a complete software package for the storage and management of samples. MolabIS allows to track a large number of samples of different types. It provides a five-level hierarchical storage scheme ensuring the flexibility in the representation of physical storage locations of samples and DNA in different labs. The lab manager can define a new location, update and delete existing ones at any storage level.

R4: The workflow, one important feature in MolabIS, supports the experimental workflow in the wet lab efficiently and organizes the data entry accordingly. Data is pipelined from one step to the next in the workflow. At each step in the workflow, the details of lab work such as PCR amplification, PCR validation, and electrophoresis are recorded. This feature also highlights the difference between MolabIS and other systems, which only support importing final data.

R5: All data operations can be performed via a standard web browser including Internet Explorer 7+, Firefox 3.0+ and Safari 3+ running under a variety of operating systems. The Ajax technology used in MolabIS allows to create an interactive user interface, which has the quality of desktop applications. The users can search, view, update, and delete their data in a single

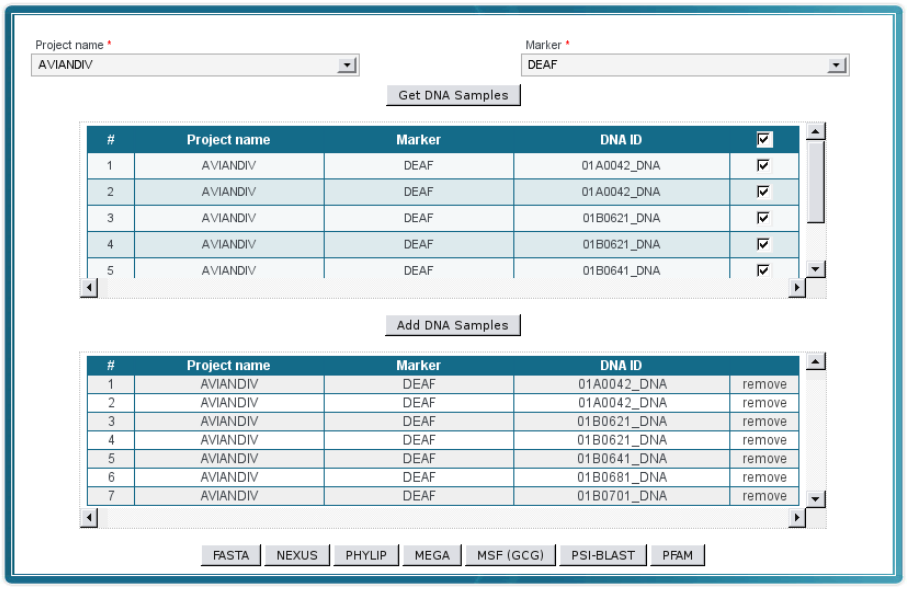

Figure 9 GUI for extracting sequence data to various formats. Sequences can be extracted and converted to different data formats. The GUI shows that users can select sequences from different projects for a specific marker to be merged into a multiple sequence file by clicking on a respective button at the bottom of the form. After pressing the button, the system will generate a file in a desired format which is available for download.

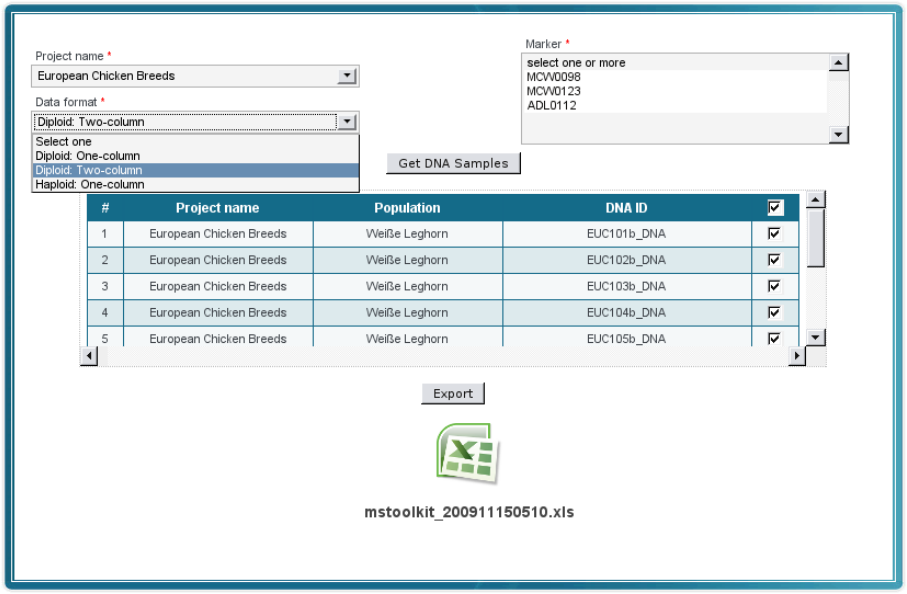

Figure 10 GUI for extracting microsatellite data to various formats. MolabIS can convert microsatellite data of a given project to various formats. The GUI shows that users can select a data format and a file format from drop-down lists for the output, identify markers and samples and click on the "Export" button to download the file.

form without switching screens. Raw data (e.g. gel pictures, chromatogram data) is stored independent of architectures of the sequencers. Therefore, MolabIS can manage all electrophoresis products, which can be obtained from different sequencers, in a uniform way. The import functionality of MolabIS has considerably enhanced the process of data entry. The details of samples and DNA can be imported in various file formats, such as .xls, .ods, or .csv. Moreover, sequence and microsatellite data can be imported into the database. Additionally, every data entry form can store additional information in a comment block thereby allowing MolabIS to function as a filing cabinet.

R6: JasperReports, an embeddable open source Java reporting library, is integrated in MolabIS to provide an effective reporting solution. The report templates are compiled with parameters specified by the user to extract data from the current database and generate the report. Although the system currently supports generating reports in PDF format, the report templates can easily be extended to other formats.

R7: MolabIS supports the retrieval of final data, as well as original files of raw data of any project. In addition, final sequences and microsatellites can be converted to various formats.

R8: Developed as a web application, MolabIS can be installed and used in a LAN or Internet, thus allowing many users to access the system simultaneously. Under the access rights control of MolabIS, data is used and shared in a secure manner. MolabIS is well-suited for localization. The text, labels, and context help in all web forms are read from an ASCII file (text file) which can be edited by any text editor.

R9: We used virtualization technology to package and deploy the application. Hence, the MolabIS appliance can be installed on different platforms (e.g. Linux, Windows). The installation process itself amounts to downloading the appliance file, installing the virtual player and running the appliance under the virtual player without any knowledge about its operating system or other software components. Under the GNU General Public License, MolabIS can be downloaded, installed and used

Table 4 Performance results of MolabIS

Test case	Number of samples in database		
	1,000	10,000	100,000
Insert 50 samples into database	6.55 ± 0.32	6.69 ± 0.27	6.47 ± 0.34
Retrieve 500 samples from database	1.62 ± 0.06	1.67 ± 0.06	1.91 ± 0.05
Export 7,000 microsatellites to CSV	2.16 ± 0.11	2.11 ± 0.10	2.10 ± 0.10

The response time is measured in seconds at the front-end of a client machine in a LAN. For each test case, the mean response time and standard deviation is calculated from the response times of ten runs. The databases consist of approximately 1,000, 10,000 and 100,000 samples for the three tests, respectively.

free of charge. This contrasts the traditional installation which starts with the installation and configuration of DBMS, web server, application framework and software components, thus requiring IT experts, who usually are not present in most labs.

R10: Loading data from previous projects can be carried out in a batch loading mode. The MLoader can be used to load large amounts of data collected and stored over the years. It executes a sophisticated system of foreign key loading and rollbacks. This facilitates the detection of similarly spelt keys and the restoration of origin data for wrong data loading.

The above list indicates that the requirements, as stated in the first section of this paper, have been met. Our software package was tested by third parties who are independent of the development of the application. Thorough testing has been carried out, in order to check for both technical bugs and missing functionality. Moreover, a user guide is available and released along with the software.

Conclusions

The development of MolabIS has solved the problems described in the first part of this paper. MolabIS is a web-based integrated information system which can be used to store, manage and handle data of DNA sequencing and microsatellite genotyping workflows. All operations can be done via a standard web browser running on any operating system. Developed as an open source software package, MolabIS takes advantage of other open source components. It brings benefits to both researchers and lab managers. For researchers, their data is stored safely with high reliability. In collaborative projects, the data can be shared in a secure manner. The system helps to reduce the workload and the time needed for searching and preparing data for subsequent lab work steps. The conversion of data formats is performed easily, thus saving time and avoiding human errors. For lab managers, MolabIS ensures long-term data storage and monitors the progress of different projects carried out by various lab members. In fact, MolabIS supports full documentation of genotyping and sequencing experiments, even with short term lab users (e.g. students or visiting scientists) and different genotyping platforms. With its general data model, MolabIS meets common requirements of various molecular genetics labs working in biodiversity. Released under the GNU General Public License, MolabIS can be downloaded, modified and used freely. MolabIS is distributed as an appliance in which all components and services are installed and pre-configured. Being a ready-to-use appliance, it can be run on different platforms by using a free player such as VMWare Player or VirtualBox with minimal installation effort.

Rapid advances in molecular genetic technology have led to a quick adaption of high throughput genotyping for SNP and NextGen Sequencing. Future releases of MolabIS will have to address this development, possibly also adding support for other molecular markers like AFLPs, which are still being used in many small labs, especially in developing countries. To accommodate these changes, the data model will have to be expanded, while preserving the core part of the sample management and all current functionality.

Availability and requirements

The source code, user guide and appliance of MolabIS are freely available at the project homepage http://www.molabis.org. We also provide a live demo for users who want to evaluate MolabIS without installation. Release notes and other information will be also updated on the project homepage.

Project name: MolabIS
Project homepage: http://www.molabis.org
Operating system: Platform independent
Programming language: Perl **Database:** Postgres
License: GNU GPL

Additional material

Additional file 1: Source code of MolabIS. The source code of MolabIS is provided as a Zip file.

Acknowledgements

This study was funded by the German Federal Ministry of Research and Education (BMBF) through the project MolabIS (VNB 03/B14). The authors are grateful to Zhivko Duchev for his helpful suggestions, Detlef Schulze for testing the software. We also thank the surveyed labs for data supports.

Author details

[1]Department of Breeding and Genetic Resources, Institute of Farm Animal Genetics (FLI), Neustadt, Germany. [2]Department of Primatology, Max Planck Institute for Evolutionary Anthropology, Leipzig, Germany. [3]Department of Bioinformatics, Institute of Microbiology and Genetics, University of Göttingen, Germany.

Authors' contributions

CT designed the data model, implemented the software, and wrote the manuscript. LG evaluated and enhanced the usability of the software and wrote the user's guide. EG initiated and supervised the project. BM co-supervised the project and revised the manuscript. All authors edited, read and approved the final manuscript.

Received: 19 May 2011 Accepted: 31 October 2011
Published: 31 October 2011

References

1. Vignala A, Milana D, SanCristobal M, Eggen A: **A review on SNP and other types of molecular markers and their use in animal genetics.** *Genet Sel Evol* 2002, **34(3)**:275-305.
2. Baumung R, Simianer H, Hoffmann I: **Genetic diversity studies in farm animals - a survey.** *J of Anim Breed and Genet* 2004, **121(6)**:361-373.
3. Rudd S, Schoof H, Mayer K: **PlantMarkers: a database of predicted molecular markers from plants.** *Nucleic Acids Res* 2005, **33**:628-632.

4. Rosenberg NA, Burke T, Elo K, Feldman MW, Freidlin PJ, Groenen MA, Hillel J, Maki-Tanila A, Tixier-Boichard M, Vignal A, Wimmers K, Weigend S: **Empirical Evaluation of Genetic Clustering Methods Using Multilocus Genotypes From 20 Chicken Breeds.** *Genetics* 2001, **159(2)**:699-713.
5. Granevitze Z, Hillel J, Chen GH, Cuc NTK, Feldman M, Eding H, Weigend S: **Genetic diversity within chicken populations from different continents and management histories.** *Animal Genetics* 2007, **36(6)**:576-583.
6. Granevitze Z, Hillel J, Feldman M, Six A, Eding H, Weigend S: **Genetic structure of a wide-spectrum chicken gene pool.** *Animal Genetics* 2009, **40(5)**:686-693.
7. Oka T, Ino Y, Nomura K, Kawashima S, Kuwayama T, Hanada H, Amano T, Takada M, Takahata N, Hayashi Y, Akishinonomiya F: **Analysis of mtDNA sequences shows Japanese native chickens have multiple origins.** *Animal Genetics* 2007, **38(3)**:287-293.
8. Liua YP, Wua GS, Yaoa YG, Miaob YW, Luikarte G, Baigf M, Beja-Pereirae A, Dingb ZL, Palanichamyb MG, Zhan YP: **Multiple maternal origins of chickens: Out of the Asian jungles.** *Molecular Phylogenetics and Evolution* 2006, **38**:12-19.
9. Johnson JA, Toepfer JE, Dunn PO: **Contrasting patterns of mitochondrial and microsatellite population structure in fragmented populations of greater prairie-chickens.** *Molecular Ecology* 2003, **12(12)**:3335-3347.
10. Cong TVC, Duchev ZI, Groeneveld E: **A Formalized Workflow for Management of Molecular Genetics Data.** *RIVF 2008 - International Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies, Ho Chi Minh City, Vietnam* 2008, 235-238.
11. Stocker G, Fischer M, Rieder D, Bindea G, Kainz S, Oberstolz M, McNally JG, Trajanoski Z: **iLAP: a workflow-driven software for experimental protocol development, data acquisition and analysis.** *BMC Bioinformatics* 2009, **10(390)**.
12. Kokocinski F, Wrobel G, Hahn M, Lichter P: **QuickLIMS: facilitating the data management for DNA-microarray fabrication.** *Bioinformatics* 2003, **19(2)**:283-284.
13. Swertz MA, de Brock EO, van Hijum SAFT, de Jong A, Buist G, Baerends RJS, Kok J, Kuipers OP, Jansen RC: **Molecular Genetics Information System (MOLGENIS): alternatives in developing local experimental genomics databases.** *Bioinformatics* 2004, **20(13)**:2075-2083.
14. Monnier S, Cox DG, Albion T, Canzian F: **T.I.M.S: TaqMan Information Management System, tools to organize data flow in a genotyping laboratory.** *BMC Bioinformatics* 2005, **6**:246.
15. Goh CS, Lan N, Echols N, Douglas SM, Milburn D, Bertone P, Xiao R, chung Ma L, Zheng D, Wunderlich Z, Acton T, Montelione GT, Gerstein M: **SPINE 2: a system for collaborative structural proteomics within a federated database framework.** *Nucleic Acids Res* 2003, **31(11)**:2833-2838.
16. Morisawa H, Hirota M, Toda T: **Development of an open source laboratory information management system for 2-D gel electrophoresis-based proteomics workflow.** *BMC Bioinformatics* 2006, **7**:430+.
17. Droit A, Hunter J, Rouleau M, Ethier C, Picard-Cloutier A, Bourgais D, Poirier G: **PARPs database: A LIMS systems for protein-protein interaction data mining or laboratory information management system.** *BMC Bioinformatics* 2007, **8**:483.
18. Wendl M, Smith S, Pohl C, Dooling D, Chinwalla A, Crouse K, Hepler T, Leong S, Carmichael L, Nhan M, Oberkfell B, Mardis E, Hillier L, Wilson R: **Design and implementation of a generalized laboratory data model.** *BMC Bioinformatics* 2007, **8**:362.
19. Jayashree B, Reddy PT, Leeladevi Y, Crouch JH, Mahalakshmi V, Buhariwalla HK, Eshwar KE, Mace E, Folkertsma R, Senthilvel S, Varshney RK, Seetha K, Rajalakshmi R, Prasanth VP, Chandra S, Swarupa L, Srikalyani P, Hoisington DA: **Laboratory Information Management Software for genotyping workflows: applications in high throughput crop genotyping.** *BMC Bioinformatics* 2006, **7**:383+.
20. Orro A, Guffanti G, Salvi E, Macciardi F, Milanesi L: **SNPLims: a data management system for genome wide association studies.** *BMC Bioinformatics* 2008, **9(2)**.
21. Schönherr S, Weiβensteiner H, Coassin S, Specht G, Kronenberg F, Brandstätter A: **eCOMPAGT - efficient Combination and Management of Phenotypes and Genotypes for Genetic Epidemiology.** *BMC Bioinformatics* 2009, **10(139)**.
22. Weiβensteiner H, Schönherr S, Specht G, Kronenberg F, Brandstätter A: **eCOMPAGT integrates mtDNA: import, validation and export of mitochondrial DNA profiles for population genetics, tumour dynamics and genotype-phenotype association studies.** *BMC Bioinformatics* 2010, **11(122)**.
23. Dunca S, Sirkanungo R, Miller L, Phillips GJ: **DraGnET: Software for storing, managing and analyzing annotated draft genome sequence data.** *BMC Bioinformatics* 2010, **11(100)**.
24. a Unified View of Data TERMT: **Peter Pin-Shan Chen.** *ACM Transactions on Database Systems* 1976, **1**:9-36.
25. Groeneveld E: **An Adaptable Platform Independent Information System in Animal Agriculture: Framework and Generic Database Structure.** *Livest Prod Sci* 2004, **87**:1-12.
26. Bozdag E, Mesbah A, Van Deursendag A: **A Comparison of Push and Pull Techniques for AJAX.** *Proceedings of the 2007 9th IEEE International Workshop on Web Site Evolution* 2007, 15-22.
27. **Prototype - a JavaScript Framework.** [http://www.prototypejs.org].
28. **PostgreSQL - an open-source object-relational DBMS.** [http://www.postgresql.org/].
29. **The Apache Software Foundation.** [http://www.apache.org].
30. Wall L, Schwartz RL: *Programming PERL* O'Reilly & Associates; 1991.
31. **Comprehensive Perl Archive Network.** [http://www.cpan.org].
32. Tregar S: **Perl module to use HTML Templates from CGI scripts.** *Online* 2002 [http://search.cpan.org/~samtregar/HTML-Template-2.6/Template.pm].
33. Mitchell D: *Using Ajax from Perl* 2006 [http://www.perl.com/lpt/a/977].
34. Harold ER, Means WS: *XML in a Nutshell* United States: O'Reilly Media; 2004.
35. Haerder T, Reuter A: **Principles of transaction-oriented database recovery.** *ACM Computing Surveys* 1983, **15(4)**:287-317.
36. Heffelfinger DR: *JasperReports for java developers: create, design, format and export reports with the world's most popular java reporting library* Packt Publishing; 2006.
37. LeBoutillier P: *Inline::Java - Write Perl classes in Java* 2005 [http://search.cpan.org/~patl/Inline-Java-0.52/Java.pod].
38. **iReport - Designer for JasperReports.** [http://sourceforge.net/projects/ireport].
39. Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, Dagdigian C, Fuellen G, Gilbert JG, Korf I, Lapp H, Lehvaslaiho H, Matsalla C, Mungall CJ, Osborne BI, Pocock MR, Schattner P, Senger M, Stein LD, Stupka E, Wilkinson MD, Birney E: **The Bioperl Toolkit: Perl Modules for the Life Sciences.** *Genome Res* 2002, **12(10)**:1611-1618.
40. Duchev Z, Cong TVC, Groeneveld E: **CryoWEB: a web software for the documentation of the cryo-preserved material in animal gene banks.** *Bioinformation* 2010, **5(5)**:219-220.
41. Park SDE: **Trypanotolerance in West African Cattle and the Population Genetic Effects of Selection.** *PhD thesis* University of Dublin; 2001.

doi:10.1186/1471-2105-12-425
Cite this article as: Truong *et al.*: MolabIS - An integrated information system for storing and managing molecular genetics data. *BMC Bioinformatics* 2011 **12**:425.

Chapter 5

An Efficient Approach to the Deployment of Complex Open Source Information Systems

Citation:
Truong CVC[1] and Groeneveld E: "An Efficient Approach to the Deployment of Complex Open Source Information Systems", *Bioinformation*, 2011, 7(4):152-153.

Original Contribution:
Truong CVC developed the MolabIS appliance, conducted the case study, and wrote the manuscript.

[1]Corresponding author

BIOINFORMATION
Discovery at the interface of physical and biological sciences

open access www.bioinformation.net Views & Challenges

Volume 7(4)

An efficient approach to the deployment of complex open source information systems

Truong Van Chi Cong* & Eildert Groeneveld

Institute of Farm Animal Genetics (FLI), Höltystrasse 10, 31535 Neustadt/Mariensee, Germany; Truong Van Chi Cong - Email: cong.chi@fli.bund.de; *Corresponding author

Received September 27, 2011; Accepted October 02, 2011; Published October 14, 2011

Abstract:
Complex open source information systems are usually implemented as component-based software to inherit the available functionality of existing software packages developed by third parties. Consequently, the deployment of these systems not only requires the installation of operating system, application framework and the configuration of services but also needs to resolve the dependencies among components. The problem becomes more challenging when the application must be installed and used on different platforms such as Linux and Windows. To address this, an efficient approach using the virtualization technology is suggested and discussed in this paper. The approach has been applied in our project to deploy a web-based integrated information system in molecular genetics labs. It is a low-cost solution to benefit both software developers and end-users.

Background:
Advances in open source have lead to the development of complex information systems which are usually implemented as component-based packages comprising third party and new software. In recent years, many web-based database applications in the field of bioinformatics (e.g. MOLGENIS **[1]**, AGL-LIMS **[2]**, DraGnET **[3]**) have been developed and released under GPL which can all be described as complex open source information systems (COSIS). Not surprisingly, most require complex installation procedures for operating system (OS), compiler, software framework, web server, database server, libraries, and other software components. Traditionally, the installation and configuration has to be done in the deployment environment, which is not under the control of the developer. As a result, the installation is complicated and often fails. It can therefore be concluded, that the widespread adoption of COSISs is severely limited by their difficult installation process.

Description:
Over the last years, hardware virtualization or "hypervisor" technology has been rapidly adopted in computing centers to make efficient use of existing hardware. The installation of software packages on the basis of a hypervisor has the same level complexity as a fresh install of an OS and the associated software. OS virtualization is a different approach which is sometimes used to facilitate software demonstration. However, it is often associated with sluggish performance. The installation of such a virtual machine (VM) on a standard host OS like Windows or Linux is very simple always resulting in a fully configured system. Here, we are proposing to use the OS virtualization approach not for testing COSISs where unresponsiveness may be acceptable, but rather for production mode operation.

Virtualization technology integrates the OS and the fully-configured application into a unit called "Virtual Appliance" (VA). Since VAs are encoded in one large file, they can easily be distributed and executed in VM software such as VMware Player **[4]** or VirtualBox **[5]** which are freely available. Therefore, the installation becomes simple and amounts to downloading the VA file to the target machine which may be either Windows or Linux. CryoWeb **[6]**, which has been deployed using the OS virtualization, is a typical example in this context.

Obviously, a COSIS that is available as a VA would make it accessible to a much larger circle of users. However, use of VA for the deployment of COSISs will only be a good strategy if the

appliance is sufficiently fast for normal production mode operations. Investigation of this issue is the objective of the case study.

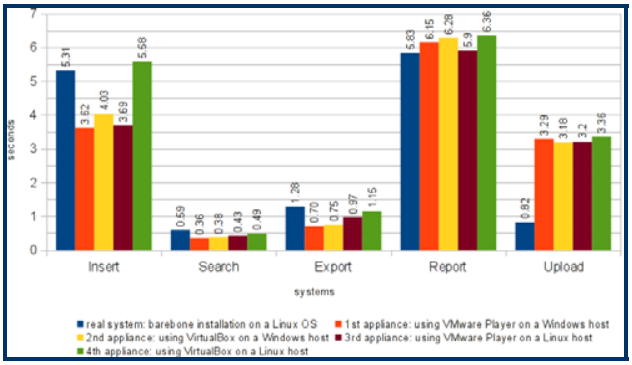

Figure 1: Benchmark results on five systems running MolabIS. The mean response time of ten runs are presented in seconds (vertical axis). Five test cases from left to right (horizontal axis) are: inserting 50 samples into the database (insert), searching 500 samples in the database (search), exporting 7000 microsatellites to a CSV file (export), reporting the details of 500 samples to a PDF file (report) and uploading a 3 MB electrophoresis file to the database (upload), respectively.

Case study:
MolabIS **[7]** is a web-based information system for storing and managing molecular data. All functionality of the system is operable through a web browser. In production mode, there may be multiple lab users accessing the system for data entry and retrieval. Typically, the use of MolabIS would require the lab users to install and configure Apache web server, Postgres database server, APIIS application framework **[8]**, Java libraries and Perl modules on a Linux platform. The installation will be painful and time consuming. In contrast, the installation of a ready-to-use appliance (available at the project homepage **[7]**) is done in a matter of minutes.

Our hypothesis is that the VAs' execution speed is fast enough for production mode operations of COSIS like MolabIS. To test this, we evaluated the performance of MolabIS in five environments. The first was a real system using barebone installation of the COSIS, the other four used Windows and Linux as host systems with VMware Player and VirtualBox as virtualization hypervisors on each **(Figure 1)**. Five test cases were chosen to represent typical production mode operations of MolabIS **(Figure 1)**. For each test case, the wall clock time between receiving the client request at the backend and its execution was measured in each environment. **Figure 1** presents the mean of ten such replicates. All tests were run on the same hardware (Intel Core i5 2x2.5 GHz processor and 6 GB of RAM).

The VA's execution speed is indeed fast enough for production mode as shown in **Figure 1**. In some cases the appliances-based COSIS even gives a better performance than the real system, a difference that is irrelevant in practical terms. Similarly, the differences among VM software found between Windows and Linux or between VMware Player and VirtualBox were negligible. As can be seen in the "Upload" case, the benchmarks indicate that VAs incur more overheads in data transactions increasing the size of virtual disks. However, for database applications like MolabIS, this virtualization overhead is acceptable.

Conclusion:
Virtualization is an efficient technique for the deployment of COSISs thereby expanding the target community considerably, opening up areas of use that would otherwise not be accessible. Apart from ease of installation, such VAs can run on different OSs, which would not be possible with a barebone installation. The benchmarks revealed no practically significant difference in response time between barebone installation and VAs, VM software and host OSs. Thus, VAs can be used effectively in the production mode for COSISs like MolabIS.

An added advantage of this approach is that VAs can be tested prior to distribution. Once packaged and tested misconfiguration during installation can be ruled out. VAs can be quickly deployed on different platforms regardless of the hardware variations of the physical servers.

References:
[1] Swertz MA *et al. Bioinformatics* 2004 **20**: 13 [PMID:15059831]
[2] Jayashree B *et al. BMC Bioinformatics* 2006 **7**: 383 [PMID:16914063]
[3] Dunca S *et al. BMC Bioinformatics* 2010 **11**: 100 [PMID:20175920]
[4] http://www.vmware.com
[5] http://www.virtualbox.org
[6] Duchev Z *et al. Bioinformation* 2010 **5**: 5 [PMID:21364801]
[7] http://www.molabis.org
[8] Groeneveld E. *Livest. Prod. Sci.* 2004 **89**: 297

Edited by P Kangueane
Citation: Truong & Groeneveld, Bioinformation 7(4): 152-153 (2011)

Chapter 6

CryoWEB: Web Software for the Documentation of the Cryo-preserved Material in Animal Gene Banks

Citation:

Duchev ZI[1], **Truong CVC** and Groeneveld E: "CryoWEB: Web software for the Documentation of the Cryo-preserved Material in Animal Gene Banks", *Bioinformation*, 2010, 5(5):219-220.

Original Contribution:

Truong CVC designed the database structure, implemented and tested CryoWEB software, and revised the manuscript.

[1]Corresponding author

Bioinformation Volume 5

open access

www.bioinformation.net Issue 5

Software

CryoWEB: Web software for the documentation of the cryo-preserved material in animal gene banks

Zhivko Duchev*, Truong Van Chi Cong, Eildert Groeneveld

Institute of Farm Animal Genetics, Friedrich-Loeffler-Institut, Hoeltystrasse 10, 31535 Neustadt, Germany; Zhivko Duchev - Email: zhivko.duchev@fli.bund.de; *Corresponding author

Received February 8, 2010; Accepted September 27, 2010; Published November 1, 2010

Abstract:
Many countries in the world have set up national domestic conservation programmes and collections of long-term storage of cryo-preserved animal genetic material. We have developed a web based software - CryoWEB for the documentation of such collections. The software is generally applicable to all livestock breeds, independent of the donor's species and the type of preserved material. The software can record basic herd-book information for the donor animal, the structure of the storage facilities, description of the stored samples and their distribution within the gene bank. It also traces the movements of the sample vessels within the storage facilities and the usage of sample units. The outputs implemented in CryoWEB address the informational needs of the gene bank manager in her everyday tasks.

Availability: CryoWEB is publicly available at http://cryoweb.tzv.fal.de

Keywords: CryoWEB, animal gene banks, documentation, Open Source

Background:
Conservation of the genetic diversity is one of the important concerns in the modern animal breeding. The 'Global Plan of Action' adopted at the United Nations Interlaken Conference on Animal Genetic Resources stipulates the setup on national gene banks worldwide. Here, CryoWEB can be used right away to serve as the electronic register.

In many countries a national conservation programme has been set-up, usually as a combination of the in situ, ex situ in vitro and ex situ in vivo methods. As part of these programmes a long-term collection of cryo-preserved animal genetic material from various breeds is established, e.g. in Brazil **[1],** USA **[2],** or France **[3].** Such collections are supposed to serve as a source of material for recovering breeds in distant future, or for supportive breeding. Therefore, the collection comprises various types of material, ranging from semen and embryos **[5]** to somatic cells **[6].** The frozen material is usually duplicated in more than one storage location for security reasons.

A consistent documentation system has been recognized (e.g. in **[6])** as an integral part of every gene bank. Such a system should collect and keep enough data for the successful identification and usage of the samples. This is a very important issue as the utilization of the stored material takes place long time after the collection when the access to primary data may be not possible. Therefore, the documentation system should contain at least a minimum data set to meet the following requirements:
For any sample chosen from the database, the user should be able to find the material in the storage facilities.
From the label on any vessel in the storage facilities the user should be able to find the information for the sample in the database.
For any sample identification there should be an exhaustive description of the protocol used for freezing the material and the procedures to be followed when thawing.

Sometimes the gene bank management organization prefers to develop a new documentation system, including specific traits relevant to the local testing and production procedures, e.g. a “days to market” field in the USA animal-GRIN database **[2].** Such systems require significant investments both in terms of funding and time for the initial development. On the other hand, in many gene banks the curators want to start recording data immediately, i.e. there is a demand for a uniform documentation system, which is easy to set up and use everywhere.
The here presented Open Source software CryoWEB was developed with the intention to be a generally applicable gene bank documentation system. It can be used uniformly across species, material types and storage facilities.

Methodology:
CryoWEB was developed in the Institute of Farm Animal Genetics (FLI) in Germany on the basis and concepts of the CryoIS software **[4]** used for the Dutch gene bank. Several of the main blocks were preserved, but major improvements were introduced. First of all CryoWEB is intended to be used “out of the box”, whereas CryoIS had to be customized. Secondly, the whole system was changed from desktop to a Web application. Finally, nationalization options and access control were introduced.

The main concept of CryoWEB is “less is more”. The system requests from the user a minimum set of data, which is available for all species and essential for the gene bank management. Nevertheless, the user can still store additionally all data she considers valuable. This is done in the form of archives of files, where heterogeneous information can be included, e.g. birth certificate of the donor, material transfer agreement, performance test results, heath certificate, etc.

ISSN 0973-2063 (online) 0973-8894 (print)

Bioinformation Volume 5

open access

www.bioinformation.net Issue 5

Software

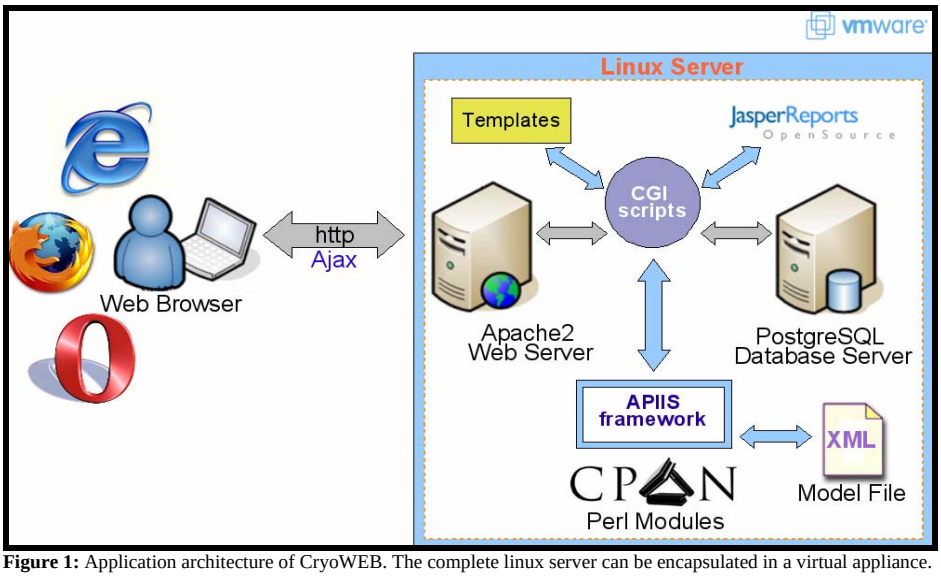

Figure 1: Application architecture of CryoWEB. The complete linux server can be encapsulated in a virtual appliance.

Input:
The data entry is logically grouped in four main blocks comprising several Web forms each. In the *Animals* block herdbook data about the donors and their pedigree may be recorded. In the *Storage Facilities* block the user specifies the structure of the tanks and freezers. In the *Samples* block the user enters data for the production and the freezing of the material, its distribution within the storage, moving and usage. The *Contacts* block provides a directory of people and organizations linked with the gene bank. The minimum data set fields are marked on each form.

Output:
CryoWEB has two types of outputs – screen outputs and reports for printing. In each data entry block the user can search and view stored data. A tree-view browsing of the hierarchy of storage locations is also developed.

The system can generate also a ready-to-print reports in PDF format. These include inter alia general gene bank statistics, information about a donor and all its samples, complete inventory of the storage facilities, history of movements in period, tracing a sample.There is also an option to export cumulated annual statistics per breed and material type for the EFABIS **[7]** network. However, this export is not based on the automated synchronization protocol used in the network **[8],** as the user must be able to correct manually the totals with the data from gene banks that use other software.

Software platform:
The software was written in Perl and runs under GNU Linux operating system, using PostgreSQL for database management system and Apache2 for web server. CryoWEB utilizes also several Perl modules from Comprehensive Perl Archive Network (CPAN) and the JasperReports framework for the outputs part. The application model of CryoWEB is given in **Figure 1**.

Distribution options:
CryoWEB is released under the GPL license and therefore the source code is freely available. Before installing this code and setting up a web page the user must have all required software (e.g. web server, additional Perl modules) in place. This assumes system administration knowledge which is not always available. To reduce the complexity of setting a complete Web system to a simple installation of a stand-alone application we offer second distribution option – virtual appliance. This is a virtual machine, where all the required software (including CryoWEB) is already installed and can be executed on Windows and Linux platforms using one of the free players from VMWare or VirtualBox.

Applications:
The CryoWEB software is installed as national gene bank information system in Netherlands, Slovakia, Slovenia, Austria, Switzerland, Iceland, Georgia, Estonia, Finland, Germany and Greece.

Acknowledgements:
The development of CryoWEB was partially supported by the European Commission, Directorate-General for Agriculture and Rural Development, under Council Regulation (EC) No 870/2004 (Action 20 - EFABISNET).

References:
[1] A da S Mariante *et al*. *Livest Sci* **120**:204 (2009)
[2] H Blackburn *Livest Sci* **120**:196 (2009)
[3] http://www.cryobanque.org/index.php?lang=en
[4] E Groeneveld *et al*. *Livest Prod Sci* **89**:297 (2004)
[5] G.Gandini *et al*. *Genet Sel Evol* **39**:465 (2007) [PMCID: PMC2682823]
[6] E Groeneveld. *AGRI* **36**:1 (2005)
[7] http://efabis.net
[8] Z Duchev & E Groeneveld *Bioinformation* **1**:146 (2006) [PMID: 17597877]

Edited by P. Kangueane
Citation: Duchev *et al.* Bioinformation 5(5): 219–220 (2010)

ISSN 0973-2063 (online) 0973-8894 (print)
Bioinformation 5(5): 219–220 (2010)

Chapter 7

A Database for Efficient Storage and Management of Multi Panel SNP Data

Citation:
Groeneveld E[1] and **Truong CVC**: "A database for efficient storage and management of multi panel SNP data", *Archives Animal Breeding*, 2013, 56(103).

Original Contribution:
Truong CVC designed the database structure, implemented Perl scripts for proof of concept, and revised the manuscript.

[1]Corresponding author

Archiv Tierzucht
Archives Animal Breeding

ISSN: 0003-9438

Short communication

A database for efficient storage and management of multi panel SNP data

Eildert Groeneveld and Cong VC Truong

Department of Breeding and Genetic Resources, Institute of Farm Animal Genetics (FLI), Neustadt, Germany

Archiv Tierzucht 56 (2013) 103
doi: 10.7482/0003-9438-56-103

Received: 17 July 2013
Accepted: 19 November 2013
Online: 20 November 2013

Corresponding author:
Eildert Groeneveld; email: eildert.groeneveld@fli.bund.de
Department of Breeding and Genetic Resources, Institute of Farm Animal Genetics (FLI), Neustadt, Germany

A database for efficient storage and management of multi panel SNP data

Eildert Groeneveld*, Cong VC Truong
Department of Breeding and Genetic Resources
Institute of Farm Animal Genetics (FLI)
D-31535 Neustadt, Germany
***E-mail: eildert.groeneveld@fli.bund.de**

Abstract

The fast development of high throughput genotyping has opened up new possibilities in genetics while at the same time producing immense data handling issues. A system design and proof of concept implementation are presented which provides efficient data storage and manipulation of single nucleotide polymorphism (SNP) genotypes in a relational database. A new strategy using SNP and individual selection vectors allows us to view SNP data as matrices or sets. These genotype sets provide an easy way to handle original and derived data, the latter at basically no storage costs. Due to its vector based database storage, data imports and exports are much faster than those of other SNP databases. In the proof of concept implementation, the compressed storage scheme reduces disk space requirements by a factor of around 300. Further, this design scales linearly with number of individuals and SNPs involved. The procedure supports panels of different sizes. This allows a straight forward management of different panel sizes in the same population as occur in animal breeding programs when higher density panels replace previous lower density versions.

Introduction

High throughput single nucleotide polymorphism (SNP) genotyping is evolving at a staggering rate, developing into a powerful tool in genetic analyses in all areas of biology. While its promises are immense, so are the data processing issues associated with it. Dropping genotyping costs and ever increasing marker densities result in a huge increase in data volume, which seems to develop faster than the already impressive rate at which data storage costs have come down in the past. Thus, increasing data storage requirements are an issue.

SNP data analysis workflows often result in filtering SNPs thereby creating genotype subsets for different purposes. This process can lead to a dramatic increase in disk space requirement: while each derived dataset will always be smaller than the original, their sizes will still be of the same magnitude, quickly using huge amounts of disk space.

High throughput genotyping is used across species with diverse sets of genotyping panels of different sizes ranging from a few thousand to millions. New panels of higher densities may be used to retype the same individuals as is done in regular animal breeding programs, resulting in ever growing datasets. Ac-

cordingly, information originating from panels of different densities will have to be managed and processed.

Space requirements of SNP data may be very large, resulting in specialized hardware having to be made available for storage. Therefore, it makes sense to centralize data management in a relational database, which can be accessed by multiple users. A number of database developments try to address some of the above issues [3, 1, 2].

However, the main shortcoming is the 'one genotype per row' (OGPR) storage scheme which leads to huge storage requirements and slow export times. Rios et al.(2010) improved on this scheme by storing one record per individual containing all genotypes together with each genotype's position. Our proposal goes well beyond this through a more efficient storage scheme and the development of the genotype set concept, which substantially enhances data manipulation.

Efficient management of SNP data has to address long term data storage, multiple genotyping panels of different sizes as are already in practical use in animal breeding and elswhere. Further, efficient selection of SNPs and fast exports to various formats for further processing are essential. This leads us to the following design.

The Design

Compressed storage

Typical files from genotyping labs easily have sizes of hundreds of Mbyte per individual. We propose to store the SNP genotypes of one individual in a compressed vector using the position as determined by the panel map, which leads to one genotype record per individual. When only the biallelic state of a SNP is of interest, two bit storage is sufficient, allowing 16 SNPs to be stored in one 32bit integer word. Accordingly, all genotypes from a 60,000 SNP panel can be stored in an integer vector of dimension 3,750 or 14.6 kb. Once a panel map is stored, any number of resulting genotypes can be loaded.

Often the call rates i.e. the GCscore need to be stored for each SNP, typically a floating point number in the range of 1 to 100. Here, a flexible scheme is proposed, allowing the user to determine the number of bits to be used for the score: 4 bits accommodate a resolution of 100/15. Using the same 4 bits on the range of 51% to 100% will double the resolution. As with the genotpyes, the GCscore is also designed as one vector per individual.

Set based manipulation

For easy data manipulation, we introduce the concept of effectively spaceless genotype sets. Each SNP panel comprises a specified set of SNPs. Using the SNP's position in the panel as its position in the genotype bit vector enables access to each SNP without explicitly having to state the SNP name. Once

genotypes are treated as vectors, a SNP dataset can be viewed as a matrix of genotypes with the SNPs constituting the columns while each genotyped individual leads to one row.

Often, only subsets of SNPs are used in analyses, perhaps only those from a particular chromosome, or SNPs with a minimum frequency. Using a bit vector *snp_sel_vec* of the dimension of the SNP panel provides a generalized approach: a .TRUE. or .FALSE. decides if a SNP is a member of a particular subset. A genotype set is then completely defined by adding a vector *indiv_sel_vec* which contains the individuals of the subset. Finally, a set name for the combination of a particular *snp_sel_vec* and *indiv_sel_vec*, uniquely specifies a genotype set. Thus, creating new derived genotype sets amounts to creating two new lists: one for the SNPs and the other for the individuals and giving this a genotype set name for easy access. The storage implications are clear: instead of having to store a matrix of dimension $nSNP * nIndividual$ only two vectors of dimension $nSNP$ and $nIndividual$ need to be stored for each derived matrix which are thus effectively spaceless.

It should be noted that the concept of genotype sets allows rapid implementation of general set operations like unions or intersections on the basis of any genotype set, original or derived.

Being a central repository for all data to be analysed, fast exports are critical. A data analysis step typically starts with an export using an appropriate format for downstream processing. Often exports are performed through costly SQL selects on the basis of SNP names [1, 2, 3].

However, export speed is a function of both the data storage and the retrieval scheme. Storage overhead of our approach is minimal: the *indiv_sel_vec* is an integer vector with as many 32bit words as there are individuals in the genotype sample, while *snp_sel_vec* has the dimension of the number of SNPs in the panel divided by 16 (for 2 bit storage). Thus, all genotypes from a 1 million panel will occupy 244kb in *snp_sel_vec*. Clearly, on this data volume basis our design will have a performance advantage over the OGPR paradigm which needs to process 1 million records per individual.

For data retrieval, the genotype set approach replaces SQL based SNP selection by much faster vector operation. An export of a genotype set amounts to these actions: firstly, *snp_sel_vec* and *indiv_sel_vec* are fetched for the chosen set. Secondly, for each individual the compressed genotype vector is retrieved through one SQL select and shrunk on the basis of the *snp_sel_vec* which can be implemented as fast shifts. Thus, the extraction speed is largely independent from the subset selection.

Performance of Proposed Design

The proof of concept implementation was done in Perl and PostgreSQL as the backend database server. Apart from conceptual simplicity, performance in terms of storage efficiency and speed of data import and export is of critical importance. This was investigated through a number of genotype datasets of

very different sizes (Table 1).

The timings refer to the import of SNPs after the panel map has been loaded, as this is done only once. Further, two bit storage for the biallelic state genotypes (A, B, AB, and no call) with no GCscore is assumed. The benchmarks were executed on an iCore 5 laptop (2.7GHz / 4GB RAM).

Importing data following our design, requires two steps. Firstly, the panel map is loaded which contains the panel size, the SNP names and chromosome location. A unique panel name is given, which needs to be specified whenever new genotype data is loaded in a second step. During the initial load, a *snp_sel_vec* is created with all bits set to 1. The individual IDs are stored in *indiv_sel_vec* as picked up from the data file. The combination of those two selection vectors then yields the first genotype set, uniquely identified by its symbolic name, which is the only information required for an export. Notice, that our design scales very well (column 5 and 6 in Table 1, with import and export time per 1 million SNP even going down as panel sizes increase.

Popular SNP database management systems store OGPR [1, 2, 3] resulting in huge storage requirements: using the HumapHap300 for 300 subjects, about 90 million of records will be generated [2]. In contrast, our design will create only 300 rows, which will clearly reduce both storage and processing requirements. Even the improved design presented by Rios et al.(2010) requires 5GB for 1.5 billion genotypes, while our design would require approximately .37GB.

A direct comparison to other implementations for computing time is difficult to make. Based on the timings from our design (1.55sec and .54/1 million genotypes, for import and exports as a weighted mean from Table 1), we expanded the software comparisons given in Table 1 in Mitha et al. (2011) for a 317K panel with 100 samples. The timings in minutes are n/a, 18, 4.2 and 0.82, for imports and 10, 93, 5, 0.29 for exports from SNPLims, GWASA, SNPpy single, and our implementation, respectively. Thus, our design seems to be an order of magnitude faster than the best of the contenders. This is not surprising, since the OGPR requires space for the sample ID, the SNP name and the chromosome location for each genotype. Using the database design given in Mitha et al. (2011) for the 317K dataset, we have a storage requirement of 2,721MB versus 8MB. Thus, our compressed bit vector storage scheme is around 300 times more efficient than the OGPR scheme of other packages.

As an example, on a 500GB disk 2 million individuals genotyped with a 317K panel can be stored using our scheme. In contrast, the same disk would store .146 million individuals with the design from Rios at al.(2010) , and only 7000 with the common OGPR storage scheme.

Supplementary Material

The proof of concept code is available under the GPL license at
ftp://ftp.tzv.fal.de/pub/snp/SNP-PoC.tar.gz

Table 1. Performance for five test datasets

data	n SNP [a]	n ind [b]	tot SNP [c]	imp [d]	exp [e]	DB stor [f]
set1	58	47	2.7	2.38	1.38	0.28
set2	36	403	14.7	4.37	1.02	0.27
set3	229	90	20.6	1.89	0.69	0.25
set4	4098	270	1106	1.60	0.53	0.25
set5	1458	1397	2036	1.50	0.54	0.25

[a]panel size: number of SNPs * 1,000
[b]number of individuals
[c]total number of SNPs in dataset * 1,000,000
[d]time to import 1 mio SNPs in seconds
[e]time to export 1 mio SNPs in seconds
[f]storage in MByte per 1 mio SNPs

References

1. C. Fong, D. C. Ko, M. Wasnick, M. Radey, S. I. Miller, and M. Brittnacher. GWAS analyzer: integrating genotype, phenotype and public annotation data for genome-wide association study analysis. *Bioinformatics*, 26(4):560–564, January 2010.

2. F. Mitha, H. Herodotou, N. Borisov, C. Jiang, J. Yoder, and K. Owzar. SNPpy-Database management for SNP data from GWAS studies. *Duke Biostatistics and Bioinformatics (B&B) Working Paper Series*, page 14, 2011.

3. A. Orro, G. Guffanti, E. Salvi, F. Macciardi, and L. Milanesi. SNPLims: a data management system for genome wide association studies. *BMC Bioinformatics*, 9(Suppl 2):S13, 2008.

4. Daniel Rios, William M. McLaren, Yuan Chen, Ewan Birney, Arne Stabenau, Paul Flicek, and Fiona Cunningham. A database and API for variation, dense genotyping and resequencing data. *BMC bioinformatics*, 11(1):238, 2010.

Chapter 8

Conclusions

8.1 Summary

With considerable advances in technologies and instruments in the field of molecular genetics, data management has become a major issue not only in key labs but also many small labs around the world. Spreadsheets and text files on a file system are only practical in labs where the datasets are not too large and heterogeneous. It is clear that the spreadsheets are often not very scalable for tracking and manipulating large amounts of data. Relational database are better suited for such tasks. Therefore, the application of database systems to the management of biological data has recently become one of important aspects in bioinformatics.

This thesis has investigated the management of data derived from Sanger sequencing and microsatellite genotyping workflows with an emphasis on biodiversity studies. More specifically, we successfully developed an open-source integrated information system which is easily installed and can be used in a wide range of molecular labs. As a result, our information system is a labs backbone for storing, managing and evaluating molecular genetics data. Indeed, the most important outcome of this study, along with the publications, is the MolabIS software (http://www.molabis.org) and its general data model.

To achieve the objectives mentioned in the first chapter, our project has passed through three research phases: data modeling, software implementation and deployment. The results of each phase were reported and discussed in our three peer-reviewed papers (Chapter 3, 4 and 5). In the following, we summarize key points and give a general discussion on the relevant results.

8.2 Results and discussion

8.2.1 Formalized data framework (paper 1)

The first paper aims to design a formalized data framework for managing sequence and microsatellite data in biodiversity studies [79]. This work has achieved three main findings as follows:

1. As mentioned in [79], the storage requirements of data items to describe the same entity (e.g. individuals, samples) differ among labs. The suggested data storage architecture can resolve this problem. The basic principle is to classify data items into three groups and use UDI blocks to store all specific information.

2. The proposed workflow approach is an effective tool to visually describe data streams in which data items are pipelined from one step to another. At each step, the data items are presented via DITs in a uniform way. Furthermore, the workflow along with DITs also indicate all relevant use cases in order to facilitate the software implementation in the following phase.

3. Based on the workflow approach, we have built the two-level workflow of biodiversity studies which focuses on the data streams of Sanger sequencing and microsatellite projects. Briefly, the data framework constructed in this study is the basis to design a general data model for developing the MolabIS software as described below.

8.2.2 Integrated information system (paper 2)

The major objective of the second paper is to implement the MolabIS software so that it can meet all requirements of the project [81]. Particularly, we have obtained the following results which make MolabIS different from other software.

1. Unlike other information systems, MolabIS keeps track of both Sanger sequencing or microsatellites genotyping workflows to capture all relevant data at the earliest possible stage. With its general design, MolabIS can store information on individuals from any species and population.

2. MolabIS provides essential tools to manage different types of genetic materials. Using a five-level hierarchical storage scheme, MolabIS can resolve different situations of sample storage systems which often use multiple levels to record physical locations of samples.

3. One of the important features which highlights the difference between MolabIS and other information systems, is the workflow support. This feature helps lab people to monitor their work to accordingly organize the data entry. Besides, the workflow feature also facilitates the sample preparation and other procedures that involve in data operations.

4. With the automatic reporting solution based on the redefined report templates, MolabIS can generate ready-to-print reports which are easily adjustable. All data elements stored in the database are quickly accessible. The raw data may be retrieved in original format. Moreover, the sequences and microsatellites can be exported to various formats for other data analysis tools.

5. Using new technologies of web 2, the operations for data manipulation are very flexible. The interface, which is well-suited for localization, is easy-to-use and compatible with different standard web browsers. The context help is given in all web forms. As a multi-user web application, MolabIS supports the access rights control to ensure that the data is always secured in the database.

6. MolabIS stores the raw data as BLOB in the database instead of decomposing them in different tables. Therefore, this way may store any type of raw data independent of hardware architecture.

7. Data migration is supported by an additional tool which can automatically import large sets of historical data from previous project in a batch mode.

8.2.3 Software deployment (paper 3)

Once the software has been implemented and tested extensively, it must be delivered to the end-users. However, it can be observed that many complex open-source information systems are very difficult to put into operation by the lab people because of their difficult installation and configuration procedures. Therefore, the objective of the third paper is to propose a solution using virtual appliances to efficiently deploy such information systems [80]. This solution involves several results as follows:

1. We indicated that the manual installation of complex software is a daunting task for users, especially non-IT persons. Unlike other software deployment solutions, the virtualization approach completely eliminates the installation and configuration complexity.

2. With this approach, the support costs as well as the time for the software installation and configuration are close to being eliminated. Since the virtual appliances

may run on different operation systems, users can continue to use the operating they are used to.

3. The results of a benchmarking experiment proved that the appliances are fast enough to run complex open-source information systems like MolabIS in a production mode.

8.3 Outlook

Towards the end of this thesis, we would like to give some perspectives for future possibilities or desirable further development. From our point of view, the research direction of the thesis may be extended to include the following issues.

Clearly, biology has entered the genomic era in which hundreds of genomes of different species have been sequenced. Rapid progress in next-generation sequencing (NGS) technologies [72] has facilitated large-scale discovery of SNPs. High throughput genotyping technologies are widely applied in genetic studies not only on humans but also on other populations. As the sequencing cost continues to decline, small labs which have previously applied the first-generation sequencing, are turning to invest in NGS. Consequently, these labs have to work with an ocean of genetic data.

Usually, scientific advances bring us many opportunities but at the same time also create new challenges. The advances in biology is not an exception. It seems that the concept of "big data" is having a big impact recently. On the one hand, high throughput technologies enables the discovery of new knowledge behind massive volumes of data and the analysis of complex traits in genetic studies to be done more easily than ever before. On the other hand, “big data” presents a number of challenges related to data analysis and management. It requires powerful hardware and complex algorithms. In this context, bioinformatics is a great candidate for “big data”.

SNP genotyping is the basis of SNP discovery (the most common application of NGS) to analyze genetic variations. Genome Wide Association Studies (GWAS) based upon SNPs is a powerful tool to discover common genetic variation across the whole genome of organisms. It is also a robust method to identify genes for complex diseases or traits. GWAS with hundreds of thousands of SNPs genotyped for thousands of individuals can yield billions of genotypes. It is a typical example of “big data” which should be stored, managed and analyzed effectively. Therefore, the development of new bioinformatics tools to manage such huge datasets is in high demand.

Typically, raw data generated from genotyping platforms (e.g. Affymetrix, Illumina) are analyzed by upstream sequencer attached software. Then, the final reports can be exported in various data formats [65] for downstream data analyses using dif-

ferent tools such as PLINK [76] or R libraries (e.g. GenABEL [63], SNPAssoc [69], GWAF [64]). This means that there are two separate analytical processes: one on raw data and and the other on final data (SNP genotypes). The former is often done by IT experts or bioinformaticians from companies conducting SNP genotyping while the latter is handled by biologists in molecular labs.

The current obstacles for lab users are the management of huge genotype datasets which range from few to several tens gigabytes. Clearly, such datasets require great disk space to store and substantial computing time to analyze. Some early efforts [76, 68] were directed towards data compression by storing genotype in binary formats. Besides, several software packages for SNP data analysis and management have been developed [76, 63, 71]. Based on their solutions or algorithms, in general, the disk space requirements may be considerably reduced for a dataset to be analyzed. Even the compression algorithm suggested recently in [77] can gain better performance. Although these contributions are useful tools for data analysis, they do not provide proper solutions for a long-term data storage for which relational databases are better suited. To address this, different database approaches have been implemented [75, 67, 73, 74]. Nevertheless, the major shortcoming is that the data storage scheme "one genotype per row" requires huge storage. This limitation has been improved in [78] by storing "one record per individual" containing all genotypes together. This solution has not achieved the compression ratio indicated in stand-alone software packages above [76, 63, 71]. Moreover, the existing applications do not provide an effective solution for storing and manipulating on derived datasets which are created from the original genotypes via the process of quality control of GWAS analysis.

To overcome above obstacles, we have proposed and designed a set-based database for efficient storage and management of SNP data [70]. Probably, additional efforts to the improvement of the existing algorithms and the development of a complete data management system are our future work.

8.4 References

[63] AULCHENKO, Y. S., RIPKE, S., ISAACS, A., AND VAN DUIJN, C. M. GenABEL: an r library for genome-wide association analysis. *Bioinformatics (Oxford, England) 23*, 10 (2007), 1294–1296.

[64] CHEN, M.-H. H., AND YANG, Q. GWAF: an r package for genome-wide association analyses with family data. *Bioinformatics (Oxford, England) 26*, 4 (2010), 580–581.

[65] DONG, C. SNPTransformer: A lightweight toolkit for Genome-Wide association studies. *Genomics, Proteomics & Bioinformatics 8*, 4 (2010), 268–273.

[66] DUCHEV, Z., TRUONG, C. V. C., AND GROENEVELD, E. Cryoweb: Web software for the documentation of the cryo-preserved material in animal gene banks. *Bioinformation 5*, 5 (2010), 219–220.

[67] FONG, C., KO, D. C., WASNICK, M., RADEY, M., MILLER, S. I., AND BRITTNACHER, M. GWAS analyzer: integrating genotype, phenotype and public annotation data for genome-wide association study analysis. *Bioinformatics (Oxford, England) 26*, 4 (2010), 560–564.

[68] FUKUSHIMA, K., WU, M.-H., BOCCHINI, S., RASYIDA, A., AND YANG, M.-C. PBAT based nanocomposites for medical and industrial applications. *Materials Science and Engineering: C 32*, 6 (2012), 1331–1351.

[69] GONZÁLEZ, J. R., ARMENGOL, L., SOLÉ, X., GUINÓ, E., MERCADER, J. M., ESTIVILL, X., AND MORENO, V. SNPassoc: an r package to perform whole genome association studies. *Bioinformatics (Oxford, England) 23*, 5 (2007), 644–645.

[70] GROENEVELD, E., AND TRUONG, C. V. C. A database for efficient storage and management of multi panel snp data. *Archives Animal Breeding 56*, 103 (2013).

[71] JOMBART, T., AND AHMED, I. adegenet 1.3-1: new tools for the analysis of genome-wide SNP data. *Bioinformatics (Oxford, England) 27*, 21 (2011), 3070–3071.

[72] METZKER, M. L. Sequencing technologies - the next generation. *Nature reviews. Genetics 11*, 1 (2010), 31–46.

[73] MITHA, F., HERODOTOU, H., BORISOV, N., JIANG, C., YODER, J., AND OWZAR, K. SNPpy - database management for SNP data from genome wide association studies. *PLoS ONE 6*, 10 (2011), e24982+.

[74] MUÑIZ FERNANDEZ, F., CARREÑO−TORRES, A., MORCILLO-SUAREZ, C., AND NAVARRO, A. Genome-wide association studies pipeline (GWASpi): a desktop application for genome-wide SNP analysis and management. *Bioinformatics 27*, 13 (2011), 1871–1872.

[75] ORRO, A., GUFFANTI, G., SALVI, E., MACCIARDI, F., AND MILANESI, L. SNPLims: a data management system for genome wide association studies. *BMC Bioinformatics 9*, Suppl 2 (2008), S13+.

[76] PURCELL, S., NEALE, B., TODD-BROWN, K., THOMAS, L., FERREIRA, M. A. R., BENDER, D., MALLER, J., SKLAR, P., DE BAKKER, P. I. W., DALY, M. J., AND SHAM, P. C. PLINK: A tool set for Whole-Genome association and Population-Based linkage analyses. *Am J Hum Genet 81*, 3 (2007), 559–575.

[77] QIAO, D., YIP, W. K., AND LANGE, C. Handling the data management needs of high-throughput sequencing data: SpeedGene, a compression algorithm for the efficient storage of genetic data. *BMC Bioinformatics 13*, 1 (2012), 100+.

[78] RIOS, D., MCLAREN, W., CHEN, Y., BIRNEY, E., STABENAU, A., FLICEK, P., AND CUNNINGHAM, F. A database and API for variation, dense genotyping and resequencing data. *BMC Bioinformatics 11*, 1 (2010), 238+.

[79] TRUONG, C. V. C., DUCHEV, Z., AND GROENEVELD, E. Data framework for efficient management of sequence and microsatellite data in biodiversity. *Archives Animal Breeding 56*, 6 (2013), 50–64.

[80] TRUONG, C. V. C., AND GROENEVELD, E. An efficient approach to the deployment of complex open source information systems. *Bioinformation 7*, 4 (2011), 152–153.

[81] TRUONG, C. V. C., GROENEVELD, L. F., MORGENSTERN, B., AND GROENEVELD, E. Molabis - an integrated information system for storing and managing molecular genetics data. *BMC Bioinformatics 12* (2011), 425.

Abbreviations

AFLP Amplified Fragment Length Polymorphism

AJAX Asynchronous JavaScript and XML

APIIS Adaptable Platform Independent Information System

ASCII American Standard Code for Information Interchange

BLOB Binary Large Objects

BMBF Bundesministerium für Bildung und Forschung

CE Capillary Electrophoresis

COSIS Complex Open Source Information Systems

CPAN Comprehensive PERL Archive Network

CPU Central Processing Unit

CSS Cascading Style Sheet

CSV Comma Separated Value(s)

DB Database

DBMS Database Management System

DIT Data Integration Table

DNA DeoxyriboNucleic Acid

DR user Roles on the Database tasks

ERD Entity-Relationship Diagram

FLI Friedrich-Loeffler-Institut

GB Gigabyte

GHz Gigahertz

GPL General Public License

GUI Graphical User Interface

GWAS Genome Wide Association Studies

HTML HyperText Markup Language

ID Identification

IS Information System

IT Information Technology

Lab Laboratory

LAN Local Area Network

LIMS Laboratory Information Management System

NCBI National Center for Biotechnology Information

NGS Next-Generation Sequencing

ODS Open Document Spreadsheet

OGPR One Genotype Per Row

OS Operating System

PAGE Polyacrylamide Gel Electrophoresis

PCR Polymerase Chain Reaction

PDF Portable Document Format

RAM Random-Access Memory

SNP Single Nucleotide Polymorphisms

SQL Structured Query Language

SR user Roles on the System tasks

SSR Simple Sequence Repeats

STR Short Tandem Repeats

UDI Unknown Data Items

VA Virtual Appliance

VM Virtual Machine

WfMC Workflow Management Coalition

XLS Microsoft Excel Spreadsheet

XML eXtensible Markup Language

Cong VC Truong

Zum Duvenwinkel 13
31535 Neustadt, Germany
+49(0)17632947712
+49(0)5034 879824
FAX +49(0)5034 871143
cong.chi@fli.bund.de

Education

since 2007 **PhD student in Bioinformatics**, *University of Göttingen*, Lower Saxony, Germany.
Thesis: "MolabIS: A labs backbone for storing, managing & evaluating molecular genetics data"

2002–2004 **MSc in Computer Science**, *Asian Institute of Technology*, Bangkok, Thailand.
Thesis: "An interactive web-based environment for teaching Java RMI using Learning Objects" (grade: excellent)

1994–1999 **BEng in Computer Science**, *College of Information Technology*, Cantho, Vietnam.
Thesis: "Design and implementation of an online examination system" (grade: excellent)

Project Experience

2010–2013 **Project Developer**, *Institute of Farm Animal Genetics (FLI)*, Neustadt, Germany.
"QS@Breeding: development of tools for genetic assurance of breeding programs" funded by the Federal Office for Agriculture and Food (BLE), Germany. (Reference: http://qs.tzv.fal.de)

2009–2010 **Project Developer**, *Institute of Farm Animal Genetics (FLI)*, Neustadt, Germany.
"FABISnet - An integrated network of decentralized country biodiversity and genebank databases" funded by the European Commission. (Reference: http://efabisnet.tzv.fal.de)

2007–2009 **Project Developer**, *Institute of Farm Animal Genetics (FLI)*, Neustadt, Germany.
"MolabIS - Information System for Molecular Genetics Labs" funded by BMBF, Germany. (Reference: http://www.molabis.org)

2006–2007 **Project Developer**, *Institute of Farm Animal Genetics (FLI)*, Neustadt, Germany.
"CryoWEB - Information System for National Genebanks Management".
(Reference: http://cryoweb.tzv.fal.de)

2001–2006 **Software Designer & Developer**, *Cantho University*, Cantho, Vietnam.
Different educational projects (E-learning System, Courseware, Ebooks, Schedule Management System) funded by Ministry of Education and Training, Francophone Institute, and World Bank.

Skills

OS	Linux, Windows, Mac OS X
Programming	JAVA, JavaScript, C/C++, SQL
Web design	HTML, CSS, AJAX, Prototype, JSON, XML, ExtJS, Firebug
Database	PostgreSQL, MSAccess, SQL Server, MySQL, MongoDB
Graphics	Photoshop, GIMP, Dia, Flash
IDE	JCreator, Quanta, Visual Studio, JBuilder, Ext Designer, Kdevelop, DreamweaverMX, iReport
Scripting	Perl, PHP, ASP, JSP, Shell
Development	CVS, Visual Paradigm, MS Project
Service	IIS, Apache, Virtual Appliance, JasperReport, Open Source
CMS	Wordpress, Joomla, osCommerce, phpBB, vBulletin, Moodle
Typography	LaTeX, Lyx, MS Office, Open Office
Others	phpMyAdmin, pgAdmin3, SEO, VirtualBox, cPanel, FB Page, Google Analytics, BlueJ

Teaching

BA/BSc Computer Graphics, Operating Systems, Databases, Web Programming, Distributed Programming, Object-oriented Programming.

Programmer HTML, DHTML, JavaScript, J2EE, C++, XML, Java Programming, Web Design and Internet Technologies.

Publications

2013

- Groeneveld E and **Truong CVC**: "A Database for Efficient Storage and Management of Multi Panel SNP Data", *Archives Animal Breeding*, 2013, 56(103).
- **Truong CVC**, Duchev Z and Groeneveld E: "Data Framework for Efficient Management of Sequence and Microsatellite Data in Biodiversity", *Archives Animal Breeding*, 2013, 56(6):50-64.

2012

- Krostitz S, **Truong CVC**, Müller U and Groeneveld E: "Development of Tools for Quality Assurance of Breeding Programs - QS@Breeding", Proceedings of the BLE Innovationstage in Bonn-Bad Godesberg, 29-30 October 2012.
- **Truong CVC** and Groeneveld E: "Deployment of Open-Source Bioinformatics Software Using Virtualization", Proceedings of the Annual Conference of German Society for Animal Production (DGfZ/GfT) in Halle, Germany, 12-13 September 2012.
- Krostitz S, **Truong CVC**, Müller U, Fischer R, Bergfeld U and Groeneveld E: "AroundBLUP – ein effektives Softwaretool zur Evaluierung von Zuchtwertschätzungen", Proceedings of DGfZ/GfT in Halle, Germany, 12-13 September 2012.
- **Truong CVC**, Krostitz S, Fischer R, Müller U and Groeneveld E: "A Software Pipeline for Animal Genetic Evaluation", Book of Abstracts of the 63rd Annual Meeting of the European Association for Animal Production (EAAP) in Bratislava, Slovakia, 27-31 August 2012.
- Groeneveld E and **Truong CVC**: "SNPpit - Efficient Data Management for High Density Genotyping", Book of Abstracts of the 63rd EAAP in Bratislava, Slovakia, 27-31 August 2012.
- Müller U, Fischer R, **Truong CVC**, Groeneveld E and Bergfeld U: "WebLOAD - A Web Frontend to Create a Consistent Dataset from Multiple Text Files", Book of Abstracts of the 63rd EAAP in Bratislava, Slovakia, 27-31 August 2012.

2011

- **Truong CVC**, Groeneveld LF, Morgenstern B and Groeneveld E: "MolabIS - An Integrated Information System for Storing and Managing Molecular Genetics Data", *BMC Bioinformatics*, 2011, 12:425+.
- **Truong CVC** and Groeneveld E: "An Efficient Approach to the Deployment of Complex Open Source Information Systems", *Bioinformation*, 2011, 7(4):152-153.
- **Truong CVC** and Groeneveld E: "A Perl Toolkit for Large-scale SNP Genotype Data Management", Proceedings of DGfZ/GfT in Freising-Weihenstephan, Germany, 6-7 September 2011, B20.
- Krostitz S, **Truong CVC**, Müller U, Bergfeld U and Groeneveld E: "Von PEST zu ZwISSS - Eine Software Pipeline", Proceedings of DGfZ/GfT in Freising-Weihenstephan, Germany, 6-7 September 2011.

2010

- **Truong CVC** and Groeneveld E: "MolabIS – An Open Source Information System for Sequencing and Genotyping Workflows", Proceedings of the 9th World Congress on Genetics Applied to Livestock Production (WCGALP) in Leipzig, Germany, 1-6 August 2010.
- Duchev ZI, **Truong CVC** and Groeneveld E: "CryoWEB: Web software for the Documentation of the Cryo-preserved Material in Animal Gene Banks", *Bioinformation*, 2010, 5(5):219-220.
- Krostitz S, **Truong CVC**, Müller U and Groeneveld E: "Development of Tools for Quality Assurance of Breeding Programs - QS@Breeding", Proceedings of the BLE Innovationstage in Berlin, 6-7 October 2010.

2009

- **Truong CVC** and Groeneveld E: "Information Management System for Sequences and Microsatellites Data", Proceedings of DGfZ/GfT in Giessen, Germany, 16-17 September 2009.
- **Truong CVC**, Duchev Z and Groeneveld E: "MolabIS - Effective Management of Genetic Data in Farm Animal Biodiversity Studies", Book of Abstracts of the 60th EAAP in Barcelona, Spain, 24-27 August 2009.
- **Truong CVC**, Duchev Z and Groeneveld E: "A Software Package for Managing and Evaluating DNA Sequence and Microsatellite Data", Proceedings of the GIL Conference - Demands on IT in Agriculture, Forestry and Food Industry by Globalization and Climate Change in Rostock, Germany, 09-10 March 2009.

2008

- **Truong CVC**, Duchev Z and Groeneveld E: "CryoWEB - A Web Application for Managing National Genebanks", Proceedings of DGfZ/GfT in Bonn, Germany, 17-18 September 2008.
- **Truong CVC**, Duchev Z and Groeneveld E: "Design and Implementation of an Information System for National Genebanks Management", Book of Abstracts of the 59th EAAP in Vilnius, Lithuania, 24-27 August 2008.
- **Truong CVC**, Duchev Z and Groeneveld E: "A Formalized Workflow for Management of Molecular Genetics Data", Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies (RIVF) in Ho Chi Minh, Vietnam, 13-17 July 2008.

2007

- **Truong CVC** and Groeneveld E: "Workflow for Storing, Managing, and Evaluating Molecular Genetics Data", Proceedings of DGfZ/GfT in Stuttgart, Germany, 26-27 September 2007.