Archiv Tierzucht 56 (2013) 6, 50-64

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.

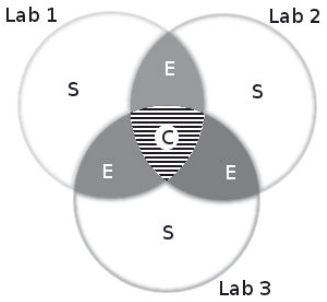

Figure 1
An example of data collection from three labs: the data items are classified into three data groups so called »Core« (C), »Extended« (E) and »Specific« (S).

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.

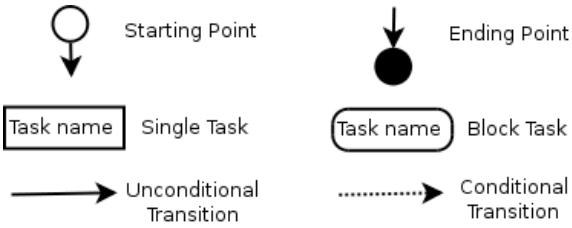

Figure 2
Graphical notations for presenting workflows

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.

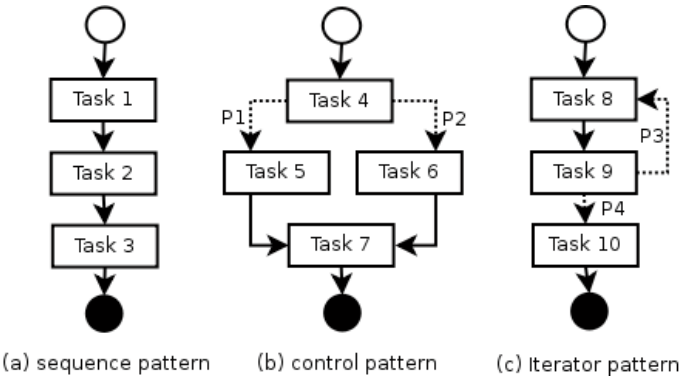

Figure 3
Workflow patterns are used to construct workflows

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.

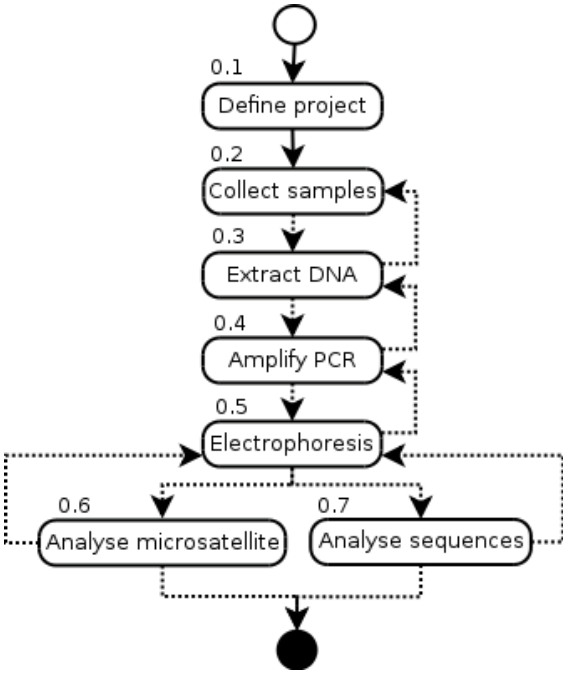

Figure 4
General workflow of biodiversity studies with seven block tasks

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

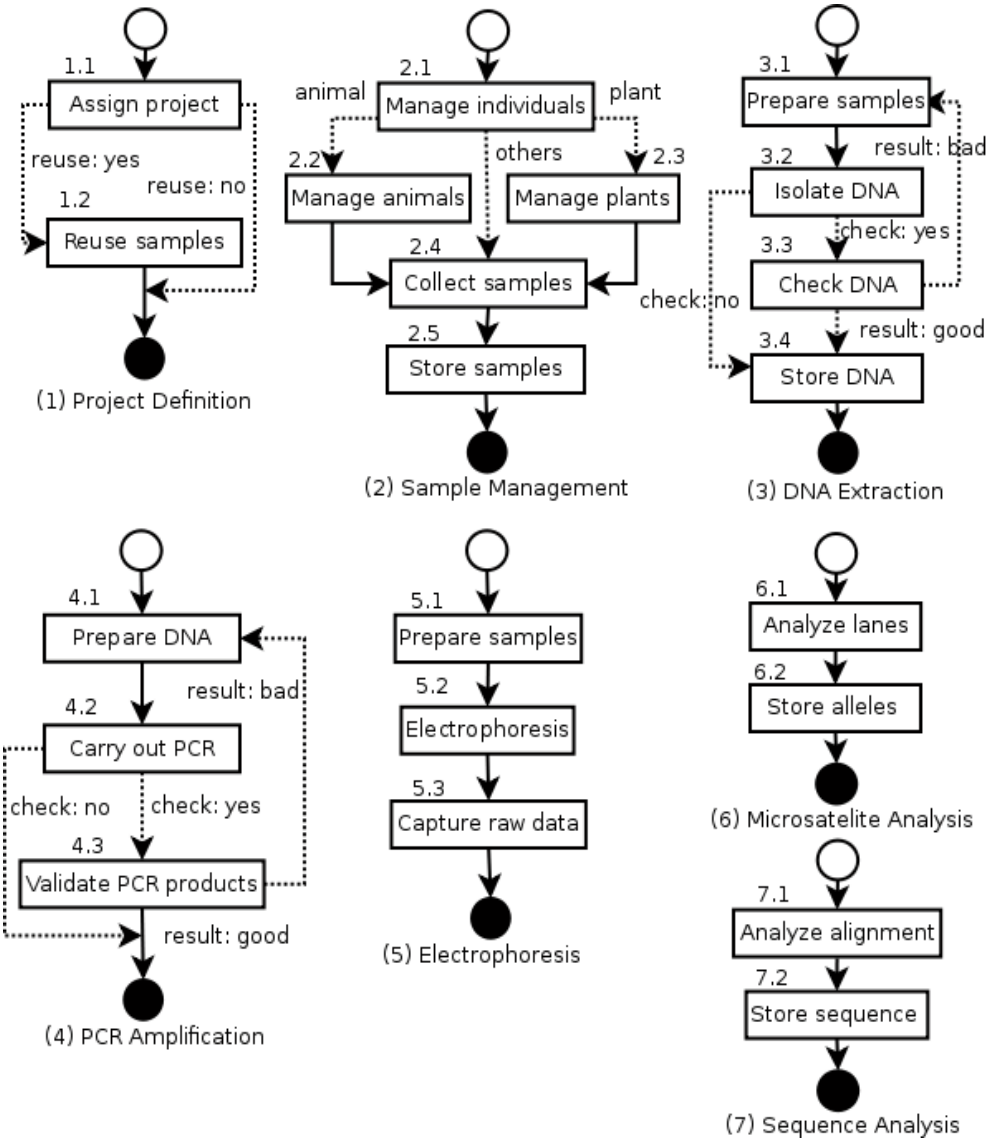

Figure 5
Sub workflows with single tasks

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.

Acknowledgements

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