

Analysis and standardization of marker genotype data for DNA fingerprinting applications

by

CORNELIS ARNOLD SCHRIEK

Submitted in partial fulfilment of the requirements for the degree

Magister Scientiae

In the Faculty of Natural and Agricultural Sciences

Bioinformatics and Computational Biology Unit

Department of Biochemistry

University of Pretoria

Pretoria

South Africa

December 2010

Supervisors: Prof. F. Joubert and Prof. A.A. Myburg

UNIVERSITY OF PRETORIA

FACULTY OF NATURAL AND AGRICULTURAL SCIENCES

DEPARTMENT OF BIOCHEMISTRY

Full name: Cornelis Arnold Schriek

Student number: 21019712

Title of the work: Analysis and standardization of marker genotype data for DNA fingerprinting applications.

Declaration

1. I understand what plagiarism entails and am aware of the University's policy in this regard.
2. I declare that this dissertation is my own, original work. Where someone else's work was used (whether from a printed source, the internet or any other source) due acknowledgement was given and reference was made according to departmental requirements.
3. I did not make use of another student's previous work and submit it as my own.
4. I did not allow and will not allow anyone to copy my work with the intention of presenting it as his or her own work.

Signature: _____

Date: _____

Acknowledgements

I thank my supervisors Prof. Fourie Joubert and Prof. Alexander Myburg for patiently guiding me through this study. I would also like to express my gratitude to Dr. Daleen van Dyk and Miss Melissa Reynolds for all their assistance and to the South African forestry company who provided access to the microsatellite marker data used in the study. I'm especially appreciative toward my mother for proofreading the entire thesis and for her, my father's and my brother's endless support. Thanks also to all my friends who supported me throughout the degree. It truly would not have been possible for me to complete it without you all.

I would further like to acknowledge the South African National Bioinformatics Network (NBN) for funding for the project, and the University of Pretoria for providing me with the opportunity to study for a degree in Bioinformatics.

Dissertation Summary

Analysis and standardization of marker genotype data for DNA fingerprinting applications

Cornelis A. Schriek

Supervised by Prof. Fourie Joubert

Co-supervised by Prof. A.A. Myburg

*Submitted in partial fulfilment of the requirements for the degree **Magister Scientiae***

Bioinformatics and Computational Biology Unit, Department of Biochemistry

University of Pretoria, December 2010

Genetic polymorphisms can be seen as the occurrence of more than one form of a DNA- or protein sequence at a single locus in a group of organisms, where these different forms occur more frequently than can be attributed to mutation alone. The combination of genetic polymorphisms present in the genome of a particular individual is referred to as its genotype. A wide range of genotyping techniques have been developed to detect and visualize genetic polymorphisms. One such technique examines highly polymorphic repetitive DNA regions called microsatellites, also called “short tandem repeats” (STRs) and sometimes “simple sequence repeats” (SSRs) or “simple-sequence length polymorphisms” (SSLPs). A microsatellite region consists of a DNA sequence of identical units of usually 2-6 base pairs strung together to produce highly variable numbers of tandem repeats among individuals of a population. Microsatellite genotyping is a popular choice for many types of studies including individual identification, paternity testing, germplasm evaluation, genome mapping and diversity studies and can be used in many commercial, academic, social, and agricultural applications.

There are, however, many obstacles in effectively managing and analysing microsatellite genotype data. Currently, researchers are struggling to effectively manage and analyse rapidly growing volumes of genotyping data. Management problems range from simply the lack of a secure, easily accessible central data repository to more complex issues like the merging and standardization of data from multiple sources into combined datasets. Due to these issues, genetic fingerprinting applications such as identity matching and relatedness studies can be challenging when data from different experiments or laboratories have to be combined into a central database.

The main aim of this M.Sc study in Bioinformatics was to develop a bioinformatics resource for the management and analysis of genetic fingerprinting data from microsatellite marker genotyping studies, and to apply the software to the analysis of microsatellite marker data from ramets of *Pinus patula* clones with the purpose of analysing clonal identity in pine breeding programmes. The software resource developed here is called GenoSonic. It is a web application that provides users with a secure, easily accessible space where genotyping project data can be managed and analysed as a team. Users can upload and download large amounts of marker genotype data. Once uploaded to the system, DNA fingerprint data needs to be standardised before it can be used in further analyses. To do this, a two-step approach was implemented in GenoSonic. The first step is to assign standardized allele sizes to all of the input allele sizes of the microsatellite fingerprints automatically using a novel automated binning algorithm called CSMerge-1, which was designed specifically to bin data from multiple experiments. The second step is to manually verify the results from the automated binning function and add the verified data to a standardized dataset. Once the genetic fingerprints have been standardized, allele- and genotype frequencies can be viewed for any given marker. GenoSonic also provides functionalities for identity matching. One or more DNA fingerprints from unknown samples can be matched against a standardized dataset to establish identities or infer relatedness. Finally, GenoSonic implements a genetic distance tree construction function, which can be used to visualize relatedness among samples in a selected dataset.

The bioinformatics resource developed in this study was applied to a microsatellite DNA fingerprinting project aimed at the re-establishment or confirmation of clonal identity of *Pinus patula* ramets from pine clonal seed orchards developed by a South African forestry company at one of their new agricultural estates in South Africa. The results from GenoSonic's automated binning function (CSMerge-1) and the results from the identity matching and tree construction exercise were compared to results obtained by human experts who have analysed the data manually. It was demonstrated that the results from GenoSonic equalled or surpassed the manual results in terms of accuracy and consistency, and far surpasses the manual effort in terms of the speed at which analyses could be completed.

GenoSonic was developed with specific focus on reusability, and the ability to be modified or extended to solve future genotyping-related problems. This study not only provides a solution to current genotype data management and analysis needs of researchers, but is aimed at serving as a basic framework, or component library for future software development projects that may be required to address specific needs of researchers dealing with high-throughput genotyping data.

Preface

Background

Microsatellites, also known as “short tandem repeats” (STRs), “simple sequence repeats” (SSRs) or “simple-sequence length polymorphisms” (SSLPs) are regions of DNA where short repetitive units are strung together in tandem. The number of repetitive units for any microsatellite is usually highly variable among individuals, making the assessment of these regions a popular technique in many fields of application including the identification of individuals, parentage and relatedness studies, population genetics and linkage mapping studies.

Technologies have been developed to enable high-throughput microsatellite marker studies. This introduces a new set of challenges with regards to the effective management and analysis of marker data. Challenges include effectively managing large volumes of genotyping data within and across research teams, merging genotype data from multiple experiments to produce standardized datasets and effectively querying and visualizing large datasets.

Although sophisticated proprietary software packages for genotype data management are available, they are mostly very expensive. Only a handful of open-source applications exist, each with a slightly different focus, feature set, technological dependencies and limitations. The development of a robust, easy-to-use, extensible genotype data management platform will help to advance research in a number of genotyping application fields. In South Africa, an open-source package will be of particular value as most institutions that require genotyping support cannot afford the proprietary software developed overseas. Furthermore, there is an opportunity to develop software that will address the specific needs of local users operating in research areas that are unique to South Africa. This study was carried out with the aim of improving the management and data analysis relevant to marker genotyping studies by creating a software solution with a particular focus on repeat length DNA marker genotyping data such as those of microsatellites in plant genomes.

The software solution

The software solution developed during this study is called GenoSonic. It has been designed to provide solutions for various technical challenges faced routinely in marker genotyping and other genetic variation studies. The foremost goal was to give researchers a central portal through which they can manage their genotyping projects. In the past, researchers typically stored all of their genotyping data in Excel spreadsheets on local computers, which made effective management,

sharing, standardization and integration of data very difficult. GenoSonic permits researchers to upload, manipulate and analyse their genotypic data *via* a remote server through an easy-to-use web interface, eliminating the need for them to maintain, share or analyse their data locally.

Further noteworthy aspects of GenoSonic include functionality for standardizing genotyping data *via* a novel automated standardization algorithm called CSMerge-1, as well as allowing users to manually standardize genotypes before committing them to standardized datasets to be used in further analyses. The effective standardization of genotyping data is paramount when doing any kind of comparative marker analysis. GenoSonic includes functions for identity matching and visualisation of measures of relatedness by way of genetic distance trees.

Another very important aspect of GenoSonic is its underlying architecture. The software has specifically been designed to maximise the opportunities for reuse, extensibility, and maintainability, by following modern design principles and patterns. It will be shown that by creating sound software architecture, the flexibility, lifetime, and opportunities of a software solution can be expanded considerably with respect to the biological science that it enables. One way in which GenoSonic aims to illustrate this is by compartmentalising its architecture into a set of lowly coupled, almost independent components, which can be exchanged for newer or different components, depending on future needs. Another good example of how GenoSonic aims to achieve extensibility is by exposing its core functionality *via* a set of web services. This will allow future developers or researchers to extend the current functionality by writing custom applications that communicate with these services. These custom applications can be anything from websites to console applications and can be written using any language on any software platform. The service-oriented architecture thus changes the software like GenoSonic from a static solution for a specific problem into one that can evolve and be extended to serve future needs.

Research Approach

Introduction

This study was performed in three main phases. The first phase consisted of a preliminary investigation which served as an introductory review of the problem domain. The study then entered into an iterative agile software engineering cycle which was repeated until all requirements for the study had been implemented successfully. The final phase of the study utilized the new software solution in a case study, the aim of which was to prove the abilities of the software by using it solve a relevant biological problem.

Model detail

An iterative software engineering process (Figure 0.1) was used to develop a marker genotype management and analysis system which was validated in a case study with DNA marker data from a pine tree fingerprinting study.

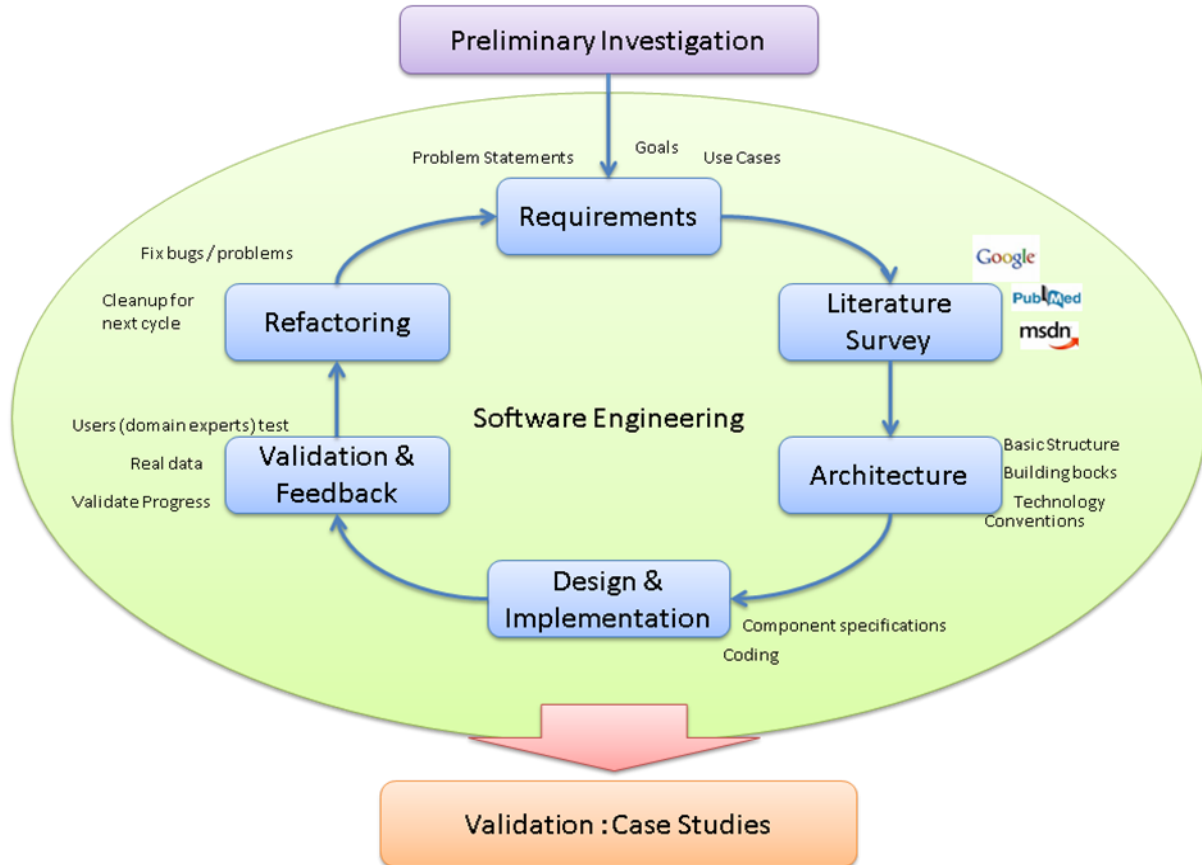


Figure 0.1: Model of the process followed for this study. *Main phases of this study beginning with the preliminary investigation, followed by an iterative software engineering process. The final software product is validated by way of a case study on real data.*

1. Preliminary investigation

The study was initiated by way of a broad investigation of the field of genotyping in general, which provided the basic frame of reference for the domain specific problems that were to be addressed.

2. Iterative software engineering process

Following the initial investigation, the study entered into an iterative development model, shown as the circular section in the model, which consisted of the following six phases:

a. Requirement gathering and functional specification

- The needs of scientists were determined with regard to the problem domain and converting these needs or requirements into a functional specification. The functional specification is basically a wish-list of all the functions the software should be able to perform.

b. Literature survey

- Once the user requirements had been determined, the next step was to investigate whether anything similar had already been done by anyone else. The following benefits could be gained by gathering this kind of intelligence:
 - “Re-inventing the wheel” could be avoided. If a viable solution already exists at a reasonable cost, there would be no need to create a new solution.
 - Parts of other solutions could be reused. In software development, this is a very useful way of fast-tracking projects.
 - Other’s successes and mistakes could be learned from. This is probably the most useful way in which science advances.
 - Ultimately, one could get a feel for the complexity and cost of fulfilling a specific user requirement and thus draw some conclusions about the feasibility of fulfilling the proposed requirement at an early stage of the project.

c. Architectural Design

- Once the requirements had been formalised and the necessary background research had been done, the following phase involved outlining the building blocks of the solution. Here decisions had to be made about the technologies to be used and specific architectural patterns and practices to be followed.

d. Software design and implementation

- After drawing up the plans for the structure, the solution was designed and implemented. The design phase basically specifies, in programming terms, exactly what a software component should be able to do and how it should do it, while the implementation phase is about getting the component to function according to its design specification, while abiding to the rules specified by the larger architecture of the system.

e. Feedback and validation

- After implementing the functionality relevant to a specific iteration, it had to be reviewed by all the stakeholders. This included letting end-users actually test out the completed component with real scientific data, which served as validation. The goal was to gather feedback about whether the solution fits the wishes of the user, what

changes or enhancements could be made, or what new requirements might have arisen. This was an important step as it enabled the solution to evolve along with user requirements.

f. Refactoring

- Following the feedback phase, the software usually underwent a level of refactoring and clean-up before the next iteration started. This mostly entailed small tweaks and clean-up of the code before commencing with the next iteration.
- Each consecutive iteration of the architectural specification-, design-, and implementation phases were subjected to a refactoring exercise of the previous iteration's implementation in order to fit the solution for the next set of requirements.

3. Application to the biological problem

The last phase of the study, after all development iterations were completed, was to validate the entire solution by testing it on actual genotyping data. The goal of the validation phase was to measure the accuracy, usability and speed of GenoSonic when utilizing it in a DNA fingerprinting study and comparing the results obtained to those obtained by human experts analysing the data manually.

Remarks on the software engineering process

The ability to adapt easily to changing requirements or architectures was the main benefit as well as the biggest pitfall to using this methodology. On the benefit side, this approach recognises the learning component of a development process. As the project develops, so do the stakeholders' knowledge about the domain and their ideas about what the solution should look like. Requirements need not be cast in stone at the beginning of a project, but rather serve as an initial guideline which can be revised as the project progresses. It is therefore also assumed that the implementation will be refactored continuously, which forces the software to be designed in a manner that is favourable to change. A disadvantage to this lack of formal structure is the relative difficulty in estimating the total magnitude of the work to be done. This could easily result in the project running longer because the 'end' had never been defined. This particular project managed this risk by continuously reviewing and revising estimates based on the set of milestones defined at any particular stage.

Dissertation overview

This dissertation is structured more or less in accordance with the phases of the development cycle described in the previous section. There are, however, a few differences in the arrangement of the content which should be taken note of in order to properly understand the context of the chapters in relation to the research model described in the previous section.

The introduction (**Chapter 1**) comprises the preliminary investigation as well as the full requirement specification as perceived at the end of the project. In other words, all the software development iterations described in the process model section are flattened to show only the final snapshot. A literature review is given of the field of DNA marker genotyping in general, the different genotyping technologies that exist and some of the typical research applications of DNA marker genotyping. The focus then narrows towards a certain area in genotyping, called microsatellite DNA fingerprinting, where the state of bioinformatics software that manage and analyse this type of data is reviewed. Having given a good introduction to the problem domain, the problem statement for this study is then formalised in the final section of the introduction. The problem statement is divided into a set of high-level objectives which, in software engineering terms, constitutes the outline of the user requirements for the software solution.

Chapter 2 is a technical discussion that deals with the underlying technological- and architectural- and design considerations. This chapter provides reasoning for specific software design decisions and serves to underline the importance of proper architectural design in improving the performance, flexibility, lifespan and overall value of a software solution.

Chapter 3 explains final implementation for each of the objectives in the problem statement from a user perspective. In other words, the chapter serves as a walkthrough of all the functionality included in the software solution from an end-user perspective, without elucidating any of the internal complexities of the system.

After the explanations of the functional and technical features of the software, the focus turns to the reason for the study: The management and analysis of microsatellite genotype data. **Chapter 4** describes the application of the resource to a genetic fingerprinting experiment aimed at the re-establishment or confirmation of clonal identity of *Pinus patula* ramets from pine clonal seed orchards owned by a South African forestry company at one of their new agricultural estates in South Africa. The data forms part of a larger collaborative molecular genetics research effort between the University of Pretoria and forestry companies in South Africa. As part of the study, the

software's ability to reproduce or improve on the process followed and results obtained by human experts who manually analysed the data was investigated.

Finally, **Chapter 5** reflects on all that has been learned and accomplished, and provides guidance toward possible future studies and projects that could be endeavoured by building on this one.

Table of Contents

ACKNOWLEDGEMENTS	III
DISSERTATION SUMMARY	IV
PREFACE	VI
Background	vi
The software solution	vi
Research Approach	vii
Introduction	vii
Model detail.....	viii
Remarks on the software engineering process	x
Dissertation overview	xi
TABLE OF CONTENTS	XIII
LIST OF TABLES	XVI
LIST OF FIGURES	XVII
LIST OF ABBREVIATIONS	XVIII
CHAPTER 1 INTRODUCTION	1
1.1 DNA marker analysis technologies: A review	2
1.1.1 Molecular marker genotyping technologies.....	2
1.1.2 Fields of application.....	12
1.2 Genotyping data management and analysis software	16
1.2.1 Existing information management software for microsatellite data	16
1.2.2 Concluding remarks	24
1.3 Conclusion	24
1.4 Problem Statement	26
CHAPTER 2 SYSTEM ARCHITECTURE AND DESIGN	28
2.1 General design principles	29
2.2 Overall architecture	29
2.3 Data Store	32
2.3.1 Database Design	32

2.4	Data Access Layer	38
2.4.1	SubSonic	39
2.4.2	T4 Templates	40
2.4.3	The Repository Pattern.....	41
2.5	Analysis Library.....	43
2.5.1	Model	43
2.5.2	Services.....	44
2.6	Services Layer	52
2.6.1	Data services.....	52
2.6.2	Analysis services	53
2.7	Web application Layer	54
2.7.1	ASP.net	54
2.7.2	Pages.....	55
2.7.3	Membership, Profiling and Security	56
2.7.4	Exception Logging.....	56
2.8	External Services Host	57
2.8.1	Windows Communication Foundation (WCF)	57
2.8.2	Core components	57
2.8.3	Security.....	59
2.9	Inversion of Control Container.....	59
2.10	Conclusion	63
CHAPTER 3 IMPLEMENTATION		64
3.1	Objective 1:A central repository	65
3.1.1	Step 1: Logging into GenoSonic via the website.....	65
3.1.2	Step 2: Maintaining projects and project groups	66
3.1.3	Step 3: Managing project data	67
3.2	Objective 2: Uploading and downloading data	69
3.2.1	Uploading data	69
3.2.2	Templates	70
3.2.3	Options	70
3.2.4	Downloading data	72
3.3	Objective 3: Binning and standardization	72
3.3.1	Step 1: Creating datasets from uploaded data for binning	72
3.3.2	Step 2: Automated binning.....	74
3.3.3	Step 3: Standardising binning results	75
3.4	Objective 4: Derive allele frequencies.....	79
3.5	Objective 5: Querying the datasets.....	80
3.5.1	Identity matching	80
3.6	Objective 6: Visualize measures of relatedness.....	82
3.7	Conclusion	83

CHAPTER 4 THE ANALYSIS OF CLONAL IDENTITIES OF <i>PINUS PATULA</i> RAMETS FROM CLONAL SEED ORCHARDS	84
4.1 Introduction	85
4.2 Materials and Methods	86
4.2.1 Fingerprint data gathering and manual analyses	86
4.2.2 GenoSonic project Setup	87
4.2.3 Import fingerprint data into GenoSonic	87
4.2.4 Binning and standardization	89
4.2.5 Derive allele frequencies	90
4.2.6 Identity matching	92
4.2.7 Visualize measures of relatedness.....	92
4.3 Results and discussion	94
4.3.1 GenoSonic binning and standardization functions compared to manual analyses.....	94
4.3.2 GenoSonic identity matching compared to findings of human experts	100
4.3.3 Confirmation and re-establishment of clonal identities.....	102
4.4 Conclusion	104
CHAPTER 5 CONCLUDING DISCUSSION	107
BIBLIOGRAPHY.....	111
APPENDIX A – ALLELE SCORES.....	118
APPENDIX B – IDENTITY MATCHING RESULT DATA.....	144
APPENDIX C – RELATEDNESS TREE.....	150

List of Tables

Table 1.1: Advantages and disadvantages of different molecular marker technologies	10
Table 1.2: Comparison of existing genotype data management and analysis software	23
Table 4.1: Microsatellite markers contained in Pine Panel A	87
Table 4.2: Output after CSMerge-1's progressive alignment step.....	90
Table 4.3: Output after CSMerge-1's QT Clustering step.....	90
Table 4.4: Allele frequencies for marker Marker1 in <i>Pinus patula</i> case study	91
Table 4.5: Genotype frequencies for marker Marker1 in <i>Pinus patula</i> case study	91
Table 4.6: First comparison between GenoSonic (automated) and expert (manual) results.....	94
Table 4.7: Second comparison between GenoSonic (automated) and expert (manual) results – after all heterozygous genotypes with one null allele have been converted to homozygous genotypes	95
Table 4.8: Renamed alleles in manual set.....	96
Table 4.9: Third comparison between GenoSonic (automated) and expert (manual) results – after manual sizes have been renamed.	96
Table 4.10: Fourth comparison between GenoSonic (automated) and expert (manual) results - After removing missing and erroneous data from comparison	97
Table 4.11: Means of clusters created by CSMerge-1	99
Table 4.12: Mismatches between GenoSonic's identity matching and human expert analysis	100
Table 4.13: Comparison of initial allele size scores to binned alleles in GenoSonic and manual analyses for uncorrelated identity matches.....	101
Table 4.14: Clonal mismatches assigned to other existing clonal groups	102
Table 4.15: Clonal mismatches that form new clonal groups.....	103
Table 4.16: Clonal mismatches with unique genotypes	104

List of Figures

Figure 0.1: Model of the process followed for this study	viii
Figure 1.1: Number of publications with the keyword “microsatellite” in PubMed over the last 15 years....	25
Figure 2.1: The GenoSonic System Architecture.....	30
Figure 2.2: GenoSonic Membership database schema.....	33
Figure 2.3: GenoSonic Project and Administration table schema	34
Figure 2.4: GenoSonic project data schema.	36
Figure 2.5: GenoSonic Upload templates schema	38
Figure 2.6: Interface hierarchy of data access repositories	42
Figure 2.7: Class hierarchy of data access repositories.....	43
Figure 2.8: Analysis library entity models	44
Figure 2.9: Illustrating the function of the ruler shift parameter in the binning algorithm	48
Figure 2.10: Illustrating the function of the size shift parameter in the binning algorithm	49
Figure 2.12: Exception Logging in GenoSonic using ELMAH	57
Figure 3.1: Login Page	65
Figure 3.2: Personal workspace	66
Figure 3.3: The Project Control	67
Figure 3.4: The expanded Project Control	67
Figure 3.5: Creating a new Run manually	68
Figure 3.6: Inline grid editing	69
Figure 3.7: Uploads page.....	70
Figure 3.8: Example of default csv file structure for allele uploads.....	71
Figure 3.9: Example of GeneMapper csv-file structure	72
Figure 3.10: Adding samples to a binning set	73
Figure 3.11: Binning set details in a matrix view	74
Figure 3.12: Matrix view of automated binning result details.....	75
Figure 3.13: Adding standard sizes to markers	76
Figure 3.14: Genotype Standardization Tool	77
Figure 3.15: Genotype Standardization Tool - Options when saving a standardized dataset.....	78
Figure 3.16: Matrix view of standardized dataset detail	79
Figure 3.17: Allele- and Genotype Frequencies page	79
Figure 3.18: Identity Matching input options	81
Figure 3.19: Identity Matching results	82
Figure 3.20: Example visualizations of GenoSonic tree structure output using PhyloWidget	83
Figure 4.1: Template mapping for allele upload	88
Figure 4.2: Visualization of GenoSonic tree structure output using PhyloWidget	93
Figure 4.3: Initial allele sizes for marker Marker2 within the size range 242 to 247.....	98
Figure 4.4: Initial allele sizes for marker Marker7 within size range 112 to 122	98

List of Abbreviations

AFLP	- Amplified Fragment Length Polymorphism
CODIS	- Combined DNA Indexing System
csv	- comma separated value
DNA	- Deoxyribonucleic acid
LINQ	- Language Integrated Query
PCR	- Polymerase Chain Reaction
RAPD	- Randomly Amplified Polymorphic DNA
RFLP	- Restriction Fragment Length Polymorphism
SNP	- Single Nucleotide Polymorphism
SQL	- Structured Query Language
SSR	- Simple Sequence Repeat
STR	- Short Tandem Repeat
T4	- Text Template Transformation Toolkit
VB	- Microsoft Visual Basic
XML	-Extensible Markup Language

Chapter 1

Introduction

1.1 DNA marker analysis technologies: A review

Fundamental knowledge of the field of DNA marker analysis is crucial to understand problems associated with managing and analysing marker genotype data. This section provides an overview of the different molecular marker technologies and discusses some of the most common applications of marker genotyping.

1.1.1 Molecular marker genotyping technologies

Genetic polymorphisms can be seen as the occurrence of more than one form of a DNA- or protein sequence at a single locus in a group of organisms, where these different alleles occur more frequently than can be attributed to mutation alone, i.e. the alleles are shared by two or more individuals of the population. DNA sequencing is the most direct and accurate way of assessing genetic polymorphisms, but, despite the tremendous increase in DNA sequencing throughput in recent years, at present marker genotyping by direct sequencing is still an expensive endeavour in terms of time, money, expertise and effort required. It is for these reasons that a wide range of other genotyping techniques have been developed to detect and visualize genetic polymorphisms. These techniques are collectively called molecular marker genotyping technologies.

The ideal molecular marker assay has a range of desirable properties. The marker should, for example, be highly polymorphic in nature, selectively neutral, codominant (i.e. it must be possible to determine the heterozygosity for a locus), and occur frequently across the genome. Assays should be high-throughput and as inexpensive as possible in terms of time, money, equipment, human labour and expertise required. The results from marker assays should also be highly reproducible among different laboratories (Weising *et al.* 1995). To date, however, no technology exists which sufficiently addresses all these requirements. Some marker systems are more suitable than others, depending on the specific type of study undertaken. A selection of molecular marker types and related genotyping technologies being used today is discussed along with a few general areas of application for the different types of molecular markers.

1.1.1.1 Protein-based markers (Allozymes)

The first molecular marker technology to be widely implemented was based on protein markers called allozymes (Prakash *et al.* 1969), which is short for the phrase “allelic variants of enzymes”. As the phrase suggests, this technology is based on the premise that allelic differences exist for many proteins among individuals of a population or among species. These allelic differences are basically variations in amino acid sequence due to changes in coding DNA sequences. Amino acids

differ in size and charge. Thus even if allozymes consist of the same number of amino acids, their total weight, charge and form would still differ if they are composed of slightly different amino acids.

Allozymes are visualized by protein electrophoresis (Prakash *et al.* 1969), a method that relies mainly on the fact that proteins with different sizes and shapes will travel through acrylamide or starch gel matrixes at different speeds when an electric current is applied to the gel. After electrophoresis the gel is treated with histochemical stains that target only specific proteins. The stained protein sites form a collection of bands called a zymogram pattern, and is used to visualize allelic variants of the protein (Shaw & Prasad 1970). All other proteins that migrated through the gel remain invisible, eliminating the need for protein purification before electrophoresis. The downside is that only around a hundred such unique stains have been developed, and they don't work equally well for each species or even for each tissue type within species. Typically a multi-locus starch gel electrophoresis will use 10 to 30 stains, which may each react with enzymes from multiple loci if the same enzymes are encoded by multiple genes. This small number of available assays makes allozymes unfit for mapping and association studies that require large numbers of markers to achieve genome-wide coverage. Allozymes are, however, still used in studies where only a few loci are genotyped and sample sizes are very large because of the cost effectiveness and simplicity of allozyme analysis (Krishnamurthy 2003).

1.1.1.2 DNA-based markers

Measuring the electrophoretic mobility of proteins is a rather indirect and inexact way of identifying DNA variation. Allozymes cannot detect the actual number of DNA changes between different alleles or where in the sequence the changes occur, only whether or not the resulting protein sequences differ in electrophoretic mobility. Mutations (in second and third codon positions) where different DNA codons code for the same amino acids, called silent mutations, also cannot be detected by way of protein analysis. Lastly, protein markers can only detect variation occurring in the coding regions of DNA, while much of the informative polymorphism in the genome is known to reside in non-coding and intergenic regions. For these reasons, along with the development of new DNA analysis techniques, there was a strong shift from protein- to DNA-based marker technologies.

1.1.1.3 Restriction Fragment Length Polymorphism markers

The discovery of restriction endonucleases (Linn & Arber 1968; Meselson & Yuan 1968) brought forth a new era in molecular marker technologies. Restriction enzymes cleave DNA at very specific

sequences of generally 4-6 base pairs in length (Roberts 1978). This gave rise to a marker technology called Restriction Fragment Length Polymorphism (RFLP) analysis (Botstein *et al.* 1980).

RFLP analysis is done by first digesting DNA with one or more endonucleases. These DNA fragments, thousands to millions depending on the restriction enzyme and the genome size, are then sorted according to size by way of gel electrophoresis. The next step is usually a visualization procedure called “Southern Hybridization” (Southern 1975). The DNA fragments in the gel are denatured and transferred to a nitrocellulose or nylon membrane using capillary action or electrophoresis while retaining the separation pattern acquired from gel electrophoresis. The membrane is then incubated with labelled ‘probes’, single-stranded DNA sequences with radioactive or enzymatic tags. DNA fragments will hybridize with these probes where complementary sequences occur and form labelled double-stranded DNA. The new double stranded fragments can then be visualized using autoradiography (X-rays), or in the case of enzymatic probes, treated with a substrate which the attached enzyme will convert into a coloured or fluorescent product.

There are different factors influencing the lengths of the DNA fragments that make up a digestion profile. Firstly, base substitutions and nucleotide insertions or deletions within a restriction site can render it inactive or can create new restriction sites. Secondly, base pair insertions and deletions within the resulting fragments will change its length. Lastly, sequence rearrangements will also change the digestion profile.

Over the years, hundreds of different restriction endonucleases have been discovered, enabling the cleavage of DNA at almost any short combination of 4 to 6 base-pairs. This allowed for the analysis of both coding and non-coding DNA sequences, as opposed to only protein sequences with allozymes. RFLP analysis is a very robust methodology, and results can easily be reproduced in different laboratories. It also has the feature of being highly discriminating, meaning that probes can be used to distinguish DNA at species or population level usually by using single-locus probes, as well at individual level using multi-locus probes. It can thus be used in phylogenetic analysis of related species, population genetics, biogeographical investigations and genetic fingerprinting related studies (Avisé 2004). RFLP analysis is also well suited for creating genetic linkage maps and association studies. In fact, some of the first such studies were done using RFLP markers (Karem 1989).

RFLPs, however, still require large amounts of high quality DNA to be used in a technique that is not readily automatable, time consuming, and quite costly compared to more modern techniques. Only a few loci can be detected for every assay (i.e. the multiplex ratio of RFLP analysis is low) and the availability of suitable probes and sufficient polymorphisms for certain species tend to be limiting factors for this type of analysis.

1.1.1.4 Minisatellites

Minisatellites, also called ‘variable number of tandem repeats’ (Nakamura *et al.* 1987), are repetitive units of 10 to about 70 base pairs found in tandem at different loci across a genome. Minisatellites were accidentally discovered by Wyman and White in 1980 (Wyman & White 1980), and subsequently Alec Jeffreys discovered that these regions were highly polymorphic (Jeffreys *et al.* 1985a) and could be used for individual identification. Gene conversion and high rates of unequal crossing over within these regions during meiosis result in changes in the number of repeated units (Jeffreys *et al.* 1999).

In 1985, Jeffreys *et al.* (1985b) introduced a new method called ‘DNA Fingerprinting’. This minisatellite marker assay is based on the same principle as RFLPs. A carefully chosen restriction enzyme is used to cut DNA on the outside of tandem repeat regions from multiple loci on the genome. The fragments are then sorted by size with electrophoresis and assayed using a Southern blotting and hybridization where repetitive DNA probes hybridize to the core units of repeat regions. The resulting barcode-like hybridization pattern is called the DNA fingerprint. Each homologous chromosome position reveals either one or two bands, depending on whether the individual is homozygous or heterozygous in the number of repeats for that locus. Usually there are many possible alleles (lengths) for each locus at the population level. The multi-locus, multi-allele nature of detected minisatellites makes the technology ideal for distinguishing individuals within a population. It can also be used in paternity testing, since each repeat array must be inherited from either the mother or father, except if mutation has occurred (Jeffreys *et al.* 1988).

The multi-locus characteristic of minisatellite analyses generally means that one cannot determine which repeat length in the blot belongs to which position in the genome. This means that one cannot ascertain whether an individual is heterozygous or homozygous at a specific location, and also that allele frequencies cannot be determined for a population. Regular population genetic studies could thus not be done by using the proportion of shared markers, which is the only quantifiable attribute for minisatellites (Kuhnlein *et al.* 1990; Lynch 1988). This shortcoming prompted the development of single-locus minisatellites (Armour *et al.* 1990). The method however remained difficult and

time-consuming to perform and was largely abandoned in favour of newer PCR-based methods like microsatellites (discussed next).

1.1.1.5 PCR-based markers

The invention of the polymerase chain reaction (Mullis & Faloona 1987; Saiki *et al.* 1988; Saiki *et al.* 1985) marked an important milestone in the advancement of molecular biology. PCR enables the rapid production of a vast number of copies of virtually any targeted piece of DNA in a test tube, thus eliminating the need for tedious cloning, isolation and purification of target DNA. PCR amplification is an indispensable method used to amplify the copy number of very small amounts of target DNA, such as can be found at a crime scene. Thus, PCR makes it possible for the sample to be analysed in a number of methods that require larger samples of DNA which will be described next.

***RAPD* markers**

Random Amplified Polymorphic DNA (RAPD) analysis is a technology that uses an arbitrary set of short PCR primers (usually 10 bases) to amplify many DNA fragments from a genomic DNA sample. The products from RAPD analysis are then separated by size with gel electrophoresis (Welsh *et al.* 1991; Williams *et al.* 1990). The resulting profile can be used to detect DNA polymorphisms, which are usually situated in the primer binding sites, or occur as length polymorphisms in the amplified fragments. Detailed methodology is described by Edwards (1998).

With RAPD analysis there is no specific target sequence. Primers will bind somewhere in the genome, but the exact locations are unknown. The origin and nature of the resulting amplified sequence fragments are thus unknown, until they are individually cloned and sequenced. This also means that RAPD analysis does not require any prior knowledge about the target sequence. This approach is useful when working with DNA from relatively uncharacterised organisms, or when DNA from only a few sources has to be compared. The RAPD method is very simple to perform, as the basic PCR method and gel electrophoresis do not require much expertise. Presynthesized primers are readily available and usually result in the amplification of many different fragments (10 to 20 per primer pair). Producing many fragments enables the very rapid detection of differences (polymorphisms) among individuals.

RAPD analysis does, however, suffer from some significant disadvantages. It has been shown to be poorly reproducible (Perez *et al.* 1998) as a result of inconsistencies in DNA quality and PCR components and conditions. This can, however, be avoided if care is taken to standardize the

procedure used (Munthali *et al.* 1992; Lowe *et al.* 1996). Another drawback is that RAPD markers are dominant, meaning that a location is either amplified or it is not. It is impossible to distinguish homozygotes from heterozygotes. Other issues include the ambiguity in the origin of identically sized fragments. There is no way of knowing whether two equal sized fragments come from the same location on the genome, except to clone and sequence the fragments, which is not feasible for large numbers of fragments and individuals.

SSRs, STRs and Microsatellites

PCR-based methods can be used to analyse highly polymorphic repetitive DNA regions called “short tandem repeats” (STRs) (Hamada *et al.* 1982), or microsatellites (Litt & Luty 1989) and sometimes “simple sequence repeats” (SSRs) (Tautz & Renz 1984). A microsatellite region consists of a repeated DNA sequence of identical units of usually 2-6 base pairs strung together in tandem.

The most arduous step in detecting microsatellite polymorphisms is to design the PCR primers. Usually this entails first constructing a genomic library for the target species. Clones from the library are then tested for the presence of microsatellite repeats, and those that do contain microsatellites are sequenced. Once the exact sequences of the regions flanking microsatellites are known, they are used to synthesize the PCR primers (this works well because flanking regions tend to be relatively conserved within species). Once these primers have been created, large numbers of samples can be easily assayed for Mendelian genotypes at specific microsatellite loci to display codominant alleles (Powell *et al.* 1996).

Microsatellites are similar to minisatellites with the exception that repeat units are much shorter. The total lengths of these regions vary enormously within a population, often with more than 20 alleles and a difference of up to 50 in repeat unit count. Microsatellites are abundantly dispersed throughout the euchromatic part of genomes, and are very common in most eukaryotes (Powell *et al.* 1996). The total length of the region is typically less than 100 base pairs, which is short enough to be easily amplified by PCR and measured accurately with the use of acrylamide gels (instead of agarose) which offer single nucleotide resolution (Tautz & Renz 1984). These features make microsatellites the method of choice for many types of studies including individual genotyping, paternity tests, germplasm evaluation, genome mapping, diversity studies, and phylogenetics.

Microsatellites have very high mutation rates, together with a complex pattern of addition and deletion of repeat units. This poses difficulties for population-genetic and biogeographical studies

(Balloux & Lugon-Moulin 2002), as alleles which are identical at present may have followed very different mutational paths in the past (Estoup *et al.* 1995). Technical problems such as PCR stutter-bands cause difficulties in automatically scoring alleles.

Microsatellites are the most popular class of marker technologies called “Sequence-tagged site” (STS) analyses. Other less frequently used techniques in this class include “Inter-simple sequence repeats” (ISSR), “Sequence characterized amplified regions” (SCARs), and “Cleaved amplified polymorphic sequences” (CAPS).

ISSRs are those sequences found between microsatellite repeat regions. ISSR analysis is based on the PCR amplification of those inter-microsatellite regions by using primers that recognise the repeat sequences plus a few bases of the flanking ISSR regions (Zietkiewicz *et al.* 1994).

SCARs are obtained by first doing a RAPD analysis, then dissecting bands of interest from the gel, cloning and sequencing them, and creating primers specific to those sequences. When these new primers are used, a clearer, normally single-locus, polymorphism pattern of the targeted site will arise (Paran & Michelmore 1993).

CAPS are similar to SCARs where only a certain region is amplified by designing specific primers, but have an extra step where the amplified fragments are digested with restriction enzymes to help identify additional polymorphisms within the sequences (Konieczny & Ausubel 1993).

AFLP (Amplified fragment length polymorphism) markers

This technique was originally invented by Zabeau and Vos in 1993 (Zabeau & Vos 1993), and then subsequently formalized in 1995 (Vos *et al.* 1995). It is based on a combination of restriction analysis and PCR amplification techniques, the goal being to amplify and score only a subset of the fragments generated by cleaving genomic DNA with restriction enzymes. AFLP markers can be used in a variety of studies including the construction of genetic maps, genotype-phenotype mappings (Vos *et al.* 1995), genetic fingerprinting, genetic diversity analysis, and a broad spectrum of population genetics and phylogenetics.

The basic procedure can be described as follows: First the DNA is digested by two restriction endonucleases, a frequent cutter (with a 4 bp recognition site) and a rare cutter (with a 6 bp recognition site). Synthetic double-stranded oligonucleotide adapters are then ligated to the ends of the fragments. A subset of these fragments is then PCR preamplified by using primers that will

only recognize specific sequences made up of the adapter oligonucleotides plus the one or two bases at the 3' end which will match the ends of the fragments. Usually the PCR step is actually split into two PCR procedures, involving a pre-selective and a selective step. In the pre-selective step, the primers used are complementary to the adapter sequences plus one additional base at the 3' end. In the subsequent "selective" step, the products from the previous step are used in a reaction where the primers have one or two further specifically selected additional bases. This allows for the visualisation of specific subsets of restriction fragments without the need for any prior knowledge about the sequence. Visualisation is mostly done by polyacrylamide gel-electrophoresis. A detailed laboratory procedure can be found in Matthes *et al.* (1998).

This method detects polymorphisms occurring in the restriction sites and adjacent bases. Different restriction enzymes plus different combinations of pre-selective and selective nucleotides in the PCR primers will affect the number of fragments visualized and the number of polymorphisms found. The more selective bases used, the more specific the search and the less the chances of finding polymorphisms. A fragment will either be amplified or not, thus polymorphisms are usually viewed as dominant, except if the change occurs inside the amplified fragment. Bands are then scored as present or absent.

SNPs (Single Nucleotide Polymorphism) markers

Single nucleotide polymorphisms are variations of single nucleotides in a genomic sequence. SNPs may occur anywhere in the genome and may have different phenotypic effects depending on their location. For instance, SNP variations in coding regions may affect the resulting protein sequences produced, except if the polymorphisms are silent mutations. SNPs in non-coding regions may affect other controls such as transcription factor binding ability or gene splicing. Of course, the SNPs could also have no effect whatsoever and merely be used as markers for nearby sequence alleles.

According to dbSNP, over 10 million SNPs have been found in human DNA (NCBI 2010), which accounts for 90% of all genetic variation in the species (Kendal 2003). Since the genome is more than 99% identical for all humans (Kruglyak & Nickerson 2001), one could argue that SNPs account for nearly all phenotypic variation observed in humans (barring environmental factors). SNPs are therefore ideal for use in studies aimed at linking genotypes to phenotypes (i.e. association genetics studies). SNPs also have very low mutation rates within species, meaning that alleles tend to be inherited from generation to generation without mutating. This makes SNPs easy to track in population genetic studies.

1.1.1.6 Summary: The advantages and shortcomings of each marker type

The following table summarizes the advantages and disadvantages of each of the marker types mentioned in previous sections.

Table 1.1: Advantages and disadvantages of different molecular marker technologies

Name	Advantages	Disadvantages
Allozymes	Cheap Robust, universal protocols, thus highly reproducible results Co-dominant markers, thus suitable for population genetics, phylogenetics, and genetic mapping studies	Only small number of available assays Indirect analysis based on phenotypes (resulting proteins) observed, it doesn't directly assay actual genetic variation Experiments require high quality fresh or frozen sample material
RFLP	Robust universal protocols, thus highly reproducible results Can be applied to any organism Co-dominant markers, able to measure heterozygosity Locus-specific, allows for synteny studies Based on sequence homology, thus suitable for phylogenetic studies between closely related species Has discriminatory power at individual as well as higher taxa levels Can be used for creating genetic linkage maps	Experiments cannot be readily automated, thus time consuming Technical expertise advised Very costly in time, human- and monetary resources Large amounts of sample DNA are required Only a few loci can be assayed at once Suitable probe libraries are required Some species exhibit very low levels of polymorphisms
Minisatellites	Highly polymorphic Easy to use	Multi-locus patterns cannot be used to determine heterozygosity Large amounts of sample DNA are required Slow (Southern blotting) Probes can be expensive High mutation rate (higher than microsatellites) can be problematic
RAPD	Cheap Large number of bands produced which can be further processed individually No prior sequence knowledge required Arbitrary primers can be bought easily Minute quantities of sample DNA required	Low reproducibility, thus cross-study comparisons very difficult Markers are mainly dominant Difficult to analyse, no prior knowledge on nature of amplified products Difficult to automate

Microsatellites:	<p>Minute quantities of sample DNA required.</p> <p>DNA need not be of excellent quality</p> <p>Highly informative because very polymorphic and thus large number of alleles</p> <p>Distributed throughout the genome</p> <p>Easy interpretation of results</p> <p>Easily automated</p> <p>High reproducibility</p> <p>Low ascertainment bias</p> <p>Easy to isolate</p> <p>Codominant markers</p>	<p>Mutation patterns are complex, can cause confusion</p> <p>Discovery procedure is complex</p> <p>Costly</p> <p>Cross-study comparisons can be problematic due to small differences in absolute allele sizes (binning problem)</p>
AFLP	<p>Can be applied to any DNA sample, thus has potential to be a universal DNA fingerprinting system</p> <p>Highly informative due to many bands generated</p> <p>No prior sequence knowledge required.</p> <p>No probes needed as primers can be fluorescently marked</p> <p>Useful for rapidly creating genetic maps or doing quick scans of a whole genome</p>	<p>Laboratory protocols are complicated</p> <p>Huge amounts of information generated, human expertise and computer assisted analyses required</p> <p>Mostly dominant markers</p> <p>AFLPs tend to cluster near centromeres & telomeres</p>
SNP	<p>Easy to genotype & automate</p> <p>Cross-study analysis is easy</p> <p>Large public data repositories available</p> <p>Markers are abundant & spread across genome</p> <p>Low mutation rate</p>	<p>Expensive assays due to requirement for sequence specific probes</p> <p>Ascertainment bias</p> <p>Low information content (a SNP can only be in a small number of allelic states)</p>
DNA Sequencing	<p>Highly reproducible results</p> <p>No ascertainment bias</p> <p>Highest possible resolution (maximum information content)</p> <p>Cross study analyses easy</p> <p>Many existing data repositories</p>	<p>Expensive</p> <p>Need adequate sequence depth to sample both alleles at each site</p> <p>Requires technical expertise</p>

1.1.1.7 Concluding remarks on genotyping technologies

There are many more marker techniques and applications which were not mentioned here, but are reviewed elsewhere. Man's relentless drive for knowledge and technological advancement ensures that the field of molecular marker analysis will keep advancing at a staggering pace. At present, scientists are finding more uses for molecular markers in nearly all biology-related study areas. Advancements in DNA sequencing technologies are rapidly reducing sequencing costs and increasing throughput. Soon whole-genome resequencing for marker genotyping could become easy

and affordable. Since whole-genome DNA resequencing poses the ultimate resolution in marker genotyping and does not suffer from any sampling bias, it may replace most current marker technologies altogether in the future.

1.1.2 Fields of application

1.1.2.1 Identification of individuals

Generally speaking, any organism can be uniquely identified by its own specific collection of alleles at different genetic loci which together represent a DNA fingerprint (Jeffreys *et al.* 1985b). DNA fingerprinting is used in many different applications. The most common application in humans is probably crime forensics, where tissue samples found at a crime scene can be used to identify suspects by matching their DNA, or to exonerate wrongly accused persons. In recent years most forensic assays switched to the use of short tandem repeat or microsatellite markers. Reasons include the high variability of microsatellites, the large number of available microsatellite loci, and the ability to amplify even minute samples with PCR.

An example approach to crime forensics is a system called CODIS (COmBined DNA Indexing System) which is employed by the US. The CODIS database is a system of pointers that enable crime laboratories across the country to compare DNA profiles. CODIS profiles consist of a specimen identifier, a DNA fingerprint, and the host laboratory of the profile. No personal identity information is saved within CODIS. In order to create a unique genetic profile of an individual, thirteen independent microsatellite loci from across the genome are assayed. CODIS includes different indices depending on the reason for the search, e.g. a convicted offenders index, missing persons index, etc. A matching algorithm is employed to produce a list of candidates matching a profile, which can then be confirmed or refuted by a qualified analyst.

Other uses of genetic fingerprinting include identification of mass disaster victims, paternity testing, authentication of expensive consumables like wine and caviar and detecting bacteria and other harmful micro-organisms that may pollute food, water, soil, and air. DNA fingerprinting is also applicable in the identification of plant breeding germplasm for variety protection and for the maintenance of clonal identity (Kirst *et al.* 2005).

1.1.2.2 Parentage and relatedness

Another application for DNA markers is that of determining the parentage of an individual (Helminen *et al.* 1988) and its relatedness to other individuals. In the case of diploid organisms, it is based on the premise that all individuals receive DNA from both parents. Specifically, for every

locus of chromosomal DNA (with two alleles) in the genome, one allele is inherited from the male parent, while the other is inherited from the female parent. Thus, one can compare the alleles of a child with those of its possible parents and generate a probability of relatedness.

With the use of genetic markers to ascertain parentage, we are able to study many evolutionary, behavioural and ecological subject fields. Studies on reproductive behaviour and gene flow within a population can be carried out with the help of molecular information even when mating is difficult to observe directly. In situations where mating can be monitored easily, the need for molecular information to determine parentage may still arise. In some species copulation also occur outside social mating pairs, which may lead to the introduction of illegitimate young. The impact of this behaviour on sexual selection and mating systems cannot be understood without applying molecular markers. The use of molecular markers help determine the genetic relatedness between individuals, which may, for example, explain special behaviour observed among presumed family members of a population. Irrefutable molecular knowledge of parent-child relationships also realise the ability to deduce the mechanisms involved in the transmission of phenotypic traits from parents to offspring.

Relatedness studies often gather molecular information on parentage from specific broods or clutches across many families of a population. This creates a profile of the genetic mating system, which may be quite different from the observed social mating system. This helps clarify questions on relationships between sexual selective pressures and observed traits resulting from these pressures, and may give insight into dispersal patterns of members of a population (Cruzan 1998; Estoup *et al.* 1994; Parker *et al.* 1998).

1.1.2.3 Population genetics and gene flow

This division of molecular marker applications deals with genetic relatedness among different populations of a species. Typical assays aim to define the degree of genetic variation or average relatedness among the populations of a particular species for a specific set of molecular markers, and how this variation is dispersed among the populations. This helps to answer questions about population sizes, gene flow, mating patterns, selective pressures, genetic diversity, and bio-geographical histories (Gentile & Sbordoni 1998; González *et al.* 1998; Xu *et al.* 1998).

Populations of almost all species demonstrate significant genetic variation among different geographic locations (Ehrlich, Raven 1969). This may be due to the fact that parents choose their mates from other individuals in their close geographic proximities rather than randomly selecting mates from all locations and siblings usually begin their lives near one another (Turner 1982). Selander (1970) demonstrated that the population genetic structure of house mice is connected to

spatial clustering on microgeographical scale (Selander 1970). The author assayed genetic profiles of mice in different barns on a farm with the use of allozymes to reveal tribal family structures connected to geographical locations.

A somewhat more sophisticated area of population genetics that only recently began receiving attention is the inference of the current and past demographic processes of a population (Wall *et al.* 2002; Wilson & Balding 1998; Beaumont 1999; Chikhi *et al.* 2001). Demographic processes refer to forces influencing population structures with respect to rates of fertility and mortality as well as redistribution or migration (gene flow). These methods rely on the distribution of alleles across populations, making markers like microsatellites or SNPs potential technologies for such studies. Microsatellite could be suitable due to their high mutation rates and variability, but uncertainty of mutation patterns and rates complicate inference (Ellegren 2000; Schlotterer 2000). SNPs can also be applied if selected carefully to avoid ascertainment bias (Wakeley *et al.* 2001; Nielsen & Signorovitch 2003; Kuhner *et al.* 2000). The best approach however remains DNA sequencing from multiple genomic regions.

Earlier population genetic studies typically used RFLPs from mitochondrial DNA or allozymes as molecular markers, but recent studies increasingly used newer methods like microsatellites, SNPs, AFLPs, and DNA sequencing.

1.1.2.4 Species phylogeny and evolution

This area of application of molecular markers involves the estimation of evolutionary relationships among species or higher taxa. After species diverge from a common ancestor, their DNA evolves further in a crudely deducible time-dependent way. The more similar the genetic makeup of two species, the closer they are related evolutionarily. One of the main aims here is to reconstruct evolutionary trees to ascertain how different species on earth have developed and diverged over time. Molecular markers are also especially powerful in delineating clades where markers can identify species derived from a common ancestor.

A popular topic in phylogenetic studies is the mapping of phenotypic characteristics into evolutionary history. Although closely related species tend to share more phenotypic traits and behavioural features than distant taxa, circumstances exist where traits may evolve independently. A good example of how molecular markers help clarify phylogenetic questions is the evolution of powered flight in mammals. The ancestral species of all mammals were unquestionably terrestrial, yet bats are distinctively flight-specialized animals. It would therefore seem that their ability to fly evolved only once in history. Bats are divided into two groups, the small nocturnal microbats with

echolocation abilities, and the larger diurnal megabats. Since microbats demonstrate very different neuroanatomical traits (like the ability to echolocate) from megabats, whose neuroanatomical makeup more resembles that of primates, another hypothesis aroused suggesting that megabats are phylogenetically closer to primates than to microbats (Pettigrew 1986; Pettigrew 1994). If this is true then flight evolved separately in microbats and megabats. Subsequent molecular phylogenetic studies however refuted this hypothesis and generally showed that all bats are indeed monophyletic (Adkins & Honeycutt 1991; Bailey *et al.* 1992; Bennett *et al.* 1988; Mindell *et al.* 1991). In a final twist, it was shown that the ability to echolocate was thus either lost by megabats after species divergence, or else it was gained independently by different microbat lineages (Springer *et al.* 2001; Teeling *et al.* 2000; Teeling *et al.* 2002).

Another topic often studied in phylogenetics is biogeographic reconstruction. This is essentially the same as assaying the biogeographical past of populations, only at a higher taxonomical level and larger timeframe. One very interesting case study involved the evolution of *Drosophilidae* flies on the Hawaiian Islands. The Islands are home to a staggering 800 estimated species of *Drosophilidae*, which is astonishing since these islands only account for less than 0.01% of the total land area of the planet. It has been shown that all these Hawaiian flies are descendent from one group (Kwiatowski *et al.* 1994). It has also been determined geologically that all islands today have been above water for at most 5 million years. The question is therefore whether it was possible for this many species to evolve in such a short period of time. Molecular studies showed that divergence of these species date back to long before the islands could have emerged (DeSalle 1992a; DeSalle 1992b), and that a common ancestor must have come to the archipelago about 30 million years ago (Piano *et al.* 1997). This paradox was solved by evidence that other islands dating back 70 million years existed, neighbouring what is currently the Hawaiian Islands. According to molecular evidence, many speciations probably occurred on these islands which are no longer visible.

Many different molecular marker technologies have been employed to reconstruct phylogenies. DNA sequencing is however by far the most powerful tool in evolutionary genetics. With the use of PCR amplification, even minute traces of ancient DNA from fossils can be extracted and phylogenetically analysed (Paabo 1989).

1.1.2.5 Association studies

Estimating the probability that an individual may express some phenotypic trait by looking at its DNA has important applications in medicine and agriculture. This field is generally called association genetics. By looking at differences in the DNA of individuals in a species and associating these differences with certain phenotypes, e.g. susceptibility to a certain disease, one

can determine which alleles contribute toward the expression of that phenotypic trait. Thus, the knowledge gained by genotyping can be very useful in finding the molecular causes of disease and how to diagnose, prevent or treat them (The International HapMap Consortium 2003). The knowledge may also help to advance agriculture by enabling the breeding of organisms with superior genotypes for traits such as natural immunity to diseases, or larger and better fruits.

These types of studies require the use of a large number of markers dispersed across the genome. SNPs are usually good marker choices, although the ultimate choice will depend mainly on the availability of markers and the scale of the study.

1.2 Genotyping data management and analysis software

Although the review thus far has dealt with marker genotyping technologies in general, from here onwards the focus will be specifically on microsatellite genotyping, as this was the primary genotyping technology chosen for the study. Computers have enabled scientists to manage and analyse huge volumes of marker data. The following section will introduce a few of the software solutions currently available for the management and analysis of microsatellite genotype data. By understanding the aims of each solution, approaches followed, features implemented, and also the underlying architecture and technologies used, conclusions can be made about their usefulness and limitations with regards to the scientific questions that could be answered.

1.2.1 Existing information management software for microsatellite data

Five different software solutions that deal with management and analysis of microsatellite genotyping data, namely AGL-LIMS, STRlab, GenoDB, STRand, and PowerMarker are reviewed here. This is by no means an exhaustive collection of all the software of this type currently in existence. Indeed, only freeware and open-source solutions are mentioned here. These examples should, however, serve as a good representation of the options currently available to the average South African scientist.

Each review examines the software both in terms of its functionality and the technology it was built upon (including the underlying source code design, where available). The reason for evaluating both the functional and technical feature set of each package is to ascertain not only what the software can currently do, but also what the suitability for use in other scenarios may be. Various possibilities of reuse exist. The entire application can be used in its current form in a different study (possibly as part of a larger pipeline), or the existing functionality can be extended or modified with

new components, or some of the components of the application can be reused in the assembly of a new solution.

1.2.1.1 AGL-LIMS

The *Applied Genomics Laboratory's (AGL) Laboratory Information Management System (LIMS)* was developed by the International Crops Research Institute for the Semi-Arid Tropics (ICRISAT). It is designed to keep track of samples throughout each of the steps involved in genotyping, from experimental design to the final persistence of binned alleles. It is currently being used to manage microsatellite data captured by ICRISAT on cereal and legume crops (Jayashree *et al.* 2006).

AGL-LIMS logically consists of four main parts:

- **An experimental design section**, which allows for the specification of project parameters, upload of sample names and genotype identities, and for setup plate configurations
- **A sample tracking section** that guides the user through experimental steps like assigning samples to specific wells on plates, DNA quantification and normalization, as well as capturing information about PCR amplification for each marker. The system has built-in quality features such as alerting users to duplicate samples and allowing the management of samples that failed in one of the genotyping steps. Users can also upload raw output files from genotyping hardware and merge genotype data across experiments using a slight variation of the Allelobin (Idury RM 1997) automated allele binning algorithm (Jayashree *et al.* 2006).
- **A reporting module**, which provides functionality for creating reports (as tab delimited files) for PCR plate design, PCR marker detail, reagent calculations, and genotyping results.
- **A data storage section**, which allows users to further annotate data entered during the workflow process. It is essentially a collection of forms users can complete with information about protocols, markers, primers, experimental conditions and chemical compositions.

AGL-LIMS has been created as a web application that users can access through any browser. The underlying software was written using the Java 5 SE language, together with the Apache Struts web framework (see <http://struts.apache.org>). Data access code for multiple data stores also exist, as reported in the original article (Jayashree *et al.* 2006). These include Microsoft SQL Server and

PostgreSQL. The choice of technologies makes AGL-LIMS platform independent, both from the server and the user point of view.

The software has been released as open source, which allows anyone to use and change the source code, within the boundaries of the particular license. The application logic has been abstracted away from the web page (html / jsp) code. All application logic is encapsulated within Struts constructs called Actions classes. Even though the business logic is uncoupled from the actual webpage, it is still tightly coupled to the Struts framework, and thus would have to be refactored before it can be reused in the future.

The data access code is mixed with business logic within all the Action classes, and SQL code is written directly as strings in the code, which tightly couples the business logic to the specific data store. This means that if the type of data store were to change, say from PostgreSQL to Oracle, every SQL query would have to be updated manually in every instance that the database is used. The software contains unparameterized SQL statements from string inputs, which makes the database vulnerable to possible SQL injection attacks (Halfond *et al.*). This may be a relevant security threat depending on the scenarios in which this software is used.

To summarize the software architecture: There is a fairly clear separation between the user interface view and the controller logic, which enables the creation of a new web interface and reuse of the same Action classes. However, these classes are specific to the Struts framework, which means that some refactoring will be required in order to use the business logic in scenarios which does not implement the Java Struts Web framework. The data access code (SQL) is intertwined with the business logic, making the software tightly coupled to the specific type of SQL database server. This limits the choice of data store. Fortunately, PostgreSQL is widely used, platform independent and scales large datasets well. In conclusion, the AGL-LIMS system can be extended and customised, but only within the technological framework that it is currently implemented.

1.2.1.2 STRLab

STRlab (Stewart 2008) is designed specifically to manage DNA casework data used by forensic teams in criminal and civil investigations. The system is currently implemented by the South African Police Service. It is a comprehensive software package that consists of a number of integrated modules, each with a different focus, all contributing in some way to analysis and tracking used in DNA forensics, specifically by assessing short tandem repeat (STR) sequence data, also known as microsatellites. The modules included in STRLab are:

1. STRtrack

This module is responsible for tracking DNA samples during the sample extraction, and fragment analysis phases of genotyping. It provides a customizable framework to ensure that properly standardized procedures are followed and documented.

2. STRgazer

This module is used for analysing results from experiments or reference samples from the database. STRgazer was designed to be used either independently or as a secondary confirmation of results produced by Applied Biosystem's GenoTyper™. Basically it's a module that can be used to curate all sample electropherograms.

3. STRbase

This module is responsible for storing the completed DNA profiles. Once the DNA profile data have been assessed using STRgazer, it is stored in STRbase. This module can also be used independently from STRtrack and STRgazer by manually importing STR marker data or even RFLP data from external sources. It can also store any number and any type of polymorphic loci.

4. STRquest

This module is used to search the database for DNA profiles that match, or are related to input profiles. It also implements a manager that can schedule relatedness tests and save the results for future queries.

5. STRstat

This module provides statistical probability functions to assess the probability of identity or relatedness.

6. STRport

This is the reporting module, through which a user can easily draw reports about sample histories or overall system statistics.

STRlab is a desktop application, written in Microsoft Visual Basic, and runs only on Windows operating systems. It is dependent on a third party relational database called DataEase for Windows. Although STRbase 3.5 is freely available, it is not open-source. Because of its modular design, either the whole system can be utilized or only parts of the system can be utilized, for

example either the sample tracking module or STRgazer can be selected as a quality control measure to in-house projects. Except for being able to optionally use specific modules in conjunction with external import files, STRlab does not provide much opportunity for reuse or integration with additional software, databases, or scenarios. It is, however, still a very comprehensive and useful application for the task it was designed to do.

1.2.1.3 GenoDB

The Genotype Database (GenoDB) is a Microsoft Access database with accompanying Access forms created to handle large sample sets of dinucleotide microsatellite markers (Li *et al.* 2001). GenoDB allows a user to import raw genotype data from tabular files, for example Microsoft Office Excel files. Once the data has been imported, there are a number of automated scripts available to assist in the analysis and standardisation of the genotype profiles. These include:

- Automatic comparison of multiple reads from the same samples. This minimises errors that may be made during the peak (allele) scoring process.
- Automatic adjustment and binning of allele sizes across multiple experiments to help standardize all genotypes in the system.
- Detection of genotyping errors, and quality checks.

GenoDB also includes a number of forms through which electrophoresis gel files and samples sources for consequential genotypes can be tracked. The database is intended for use on a single desktop computer. Any authorisation and authentication security to be employed can be customized using the Microsoft Office Access database functionality itself.

1.2.1.4 STRand

STRand is a desktop application designed to easily visualize and analyse raw short tandem repeat (STR) sample data read from fluorescence based marker technologies (Hughes 2010). STRand was created by the Davis' Veterinary Genetics Lab (UC Davis School of Veterinary Medicine, Davis, California, USA), and has been used to analyse microsatellite data from several animal species. STRand can use the output genotype and gel collection files from a number of automated DNA sequencing platforms as input. Sample data can then be analysed visually with a responsive and adjustable electropherogram view linked to data grids showing all the scoring information. The software automatically calls alleles from the raw data. These calls can then be adjusted manually, and also be refined by setting up locus rules. Samples from different runs can be split into subsets,

or grouped together, superimposed and compared at the same time. The dye channels being shown can also be customized. This makes it easy to call the true alleles manually. STRand also includes a number of built-in automatic quality measures to alert the user of possible errors in the data, like an incorrect number of alleles per sample and poor quality size scores. Finally, results can be viewed in a visual table. Any remaining issues can be highlighted in this table (Locke *et al.* 2000). This table can then be exported as a Microsoft Office Excel or other tabular data file and used as an input for further analysis, such as the functions provided by PowerMarker. STRand's automated allele calling, quality checks, and powerful data visualisation capabilities with manual standardization make STRand an excellent tool to use as a first step in standardizing data before using it in further analyses.

STRand is an open-source application licensed under the GNU General Public License written for use on a Windows platform, and employs a Microsoft Access database. Users have to log in to use the software, and therefore multiple users can have their separate data stored on the same database. Strand was however not built for security and there are no passwords linked to user logins. The application and database is not intended to be shared across a network or among multiple parties that do not fully trust one another.

STRand was programmed in Microsoft Visual Basic 6 over the course of 10 years. As is the case with many software projects with such a long lifetime, the architecture has become somewhat inconsistent. In some places, a clear separation is made between user interface logic, business logic, analysis functions, and data access, but in other cases they are mixed together. This could, however, be rectified with a refactoring exercise.

Lastly, the data access (SQL) code is written directly in the VB code, which tightly couples the code to the use of a Microsoft Access database. Since STRand was developed to be a standalone desktop application this is not a problem, but it would make it very difficult to extend the software to connect to a centralised database server such as SQL Server, as one would have to change the syntax of SQL code at every location throughout the system where data access logic is employed.

Despite these deficiencies, it should be noted that STRand was for the most part written in a highly structured way given the technological constraints of a legacy programming language. This makes the source code fairly easy to understand, and thus easy to extend, if using Visual Basic 6. Unfortunately, the technology is too out-dated to offer any code reuse or interoperability with more modern languages. Functionally the software is still very useful, but the technology it is built upon is reaching its expiry date. A good option would be to rewrite the software in a newer language.

1.2.1.5 PowerMarker

PowerMarker was created by Jack Liu, with the purpose of managing, analysing, and visualizing genotype data from a user-friendly Windows desktop environment (Liu & Muse 2005). It can handle a number of different types of genetic marker data, including microsatellites, single nucleotide polymorphisms, and restriction fragment length polymorphisms. Input data include genotype datasets, allele frequencies and genetic distance, or phylogenetic tree data. Secondary data like allele frequencies and trees can also be inferred from the raw datasets. Once data has been entered into the system, a number of statistical analysis functions become available. These include summary statistics like allele frequencies, Hardy-Weinberg disequilibrium tests, linkage disequilibrium, structural and population analyses like F-statistics, co-ancestry calculations, population differentiation tests, phylogenetic analyses and association tests. Whereas the previously mentioned genotype information management systems focussed more on the generation, tracking, and standardization of genotype data, PowerMarker deals with the statistical analysis of the finalised genotype data.

PowerMarker was written in Visual Basic.net and requires the Microsoft .Net Framework 1.1 runtime. It is freely available. It is not open-source, but detailed explanations of all statistical functions are given in the user documentation. The software does not use a centralised database, but uses simple tab- or comma delimited files as data sources. Therefore it would be very easy to use this software as a statistical analysis component in conjunction with other laboratory information management applications that can output genotype data to simple text files.

1.2.1.6 Summarizing the differences

The following table is a summary of the various attributes of each software solution discussed in paragraphs above. The particular strengths, focus, technologies, and possibilities for reuse are noted for each solution in table 1.2 below:

Table 1.2: Comparison of existing genotype data management and analysis software

Software	Strengths / Focus	Technology	Extensibility / Reusability
AGL-LIMS	Experimental design Sample tracking Automated binning Distributed user environment (web)	Web application : Java Struts web framework, Java backing code PostgreSQL database Platform independent (Windows, Linux, Mac)	Open-source. Extendable by customising java code Code reuse limited to struts environment and PostgreSQL database New tools can be developed directly on PostgreSQL database
STRLab	Sample Tracking Quality Control Identity and relatedness searches and probabilities	Desktop application, for single user Visual Basic code DataEase for Windows database Microsoft Windows only	Not open-source Customisable experiment design, project parameters, quality control constraints Modules are loosely coupled; can be used independently in conjunction with external tools
GenoDB	Easy upload of genotypes from multiple experiments/ projects Automatic data standardisation Quality control procedures. Sample tracking	Microsoft Access Database Microsoft Access user input forms VB script procedures on the database	Open-source Can be ported to SQL Server to allow for larger datasets Custom applications can be written that connect to this database
STRand	Visual inspection and manipulation of raw data Genotype Standardisation, allele calling	Visual Basic 6 code Microsoft Access Database Windows operating system only Multiple user single desktop application, no security	Open-source , can customise VB code Reuse limited to VB 6 implementations Code can be ported to new technology after clean-up & refactoring New tools can be created that plugs directly onto MS Access database
PowerMarker	Summary statistics Structural and population analyses Phylogenetic analyses Association tests	Visual Basic. Net. 1.1 Windows operating system only Single desktop user Text files as data source	Not open-source Standard file input, therefore can be used as the analysis component of a project in conjunction with other management software

1.2.2 Concluding remarks

This section has reviewed five different software solutions that deal with the management and analysis of microsatellite genotyping data. Each solution has a slightly different focus, feature set, technological basis, and set of limitations. The main purpose of reviewing these applications was to understand the approaches taken to support scientists in their work with marker genotype data, and how these applications could be useful in future scenarios.

Overall it can be concluded that although each of the aforementioned applications could be conceived as viable solutions for specific or even general problems regarding the current needs for genotyping data management or statistical analysis, the opportunities for reuse and extensibility to suit future requirements are rather limited. It would, for instance, be very difficult to reuse any of the open-source code as modules in applications with different goals. This impairs the reach of an application in terms of usefulness to the broader scientific community, which may have diverse needs that are not covered by the base system. A complete re-write would probably be needed every few years to keep up with user needs and available technologies. It is therefore the intention of this project to not only provide a solution that satisfies the current needs of the pilot study, but also to show that with proper software architecture the maintainability, reusability and extensibility and therefore the lifetime and value to the community, can be greatly enhanced.

1.3 Conclusion

Over the last ten years a steady stream of research papers have been published that used microsatellite marker genotyping in some way. This is evident from a PubMed search for articles containing the keyword ‘microsatellite’. As shown in the Figure 1.1 below, over 2000 new publications have been registered on PubMed every year since 1999.

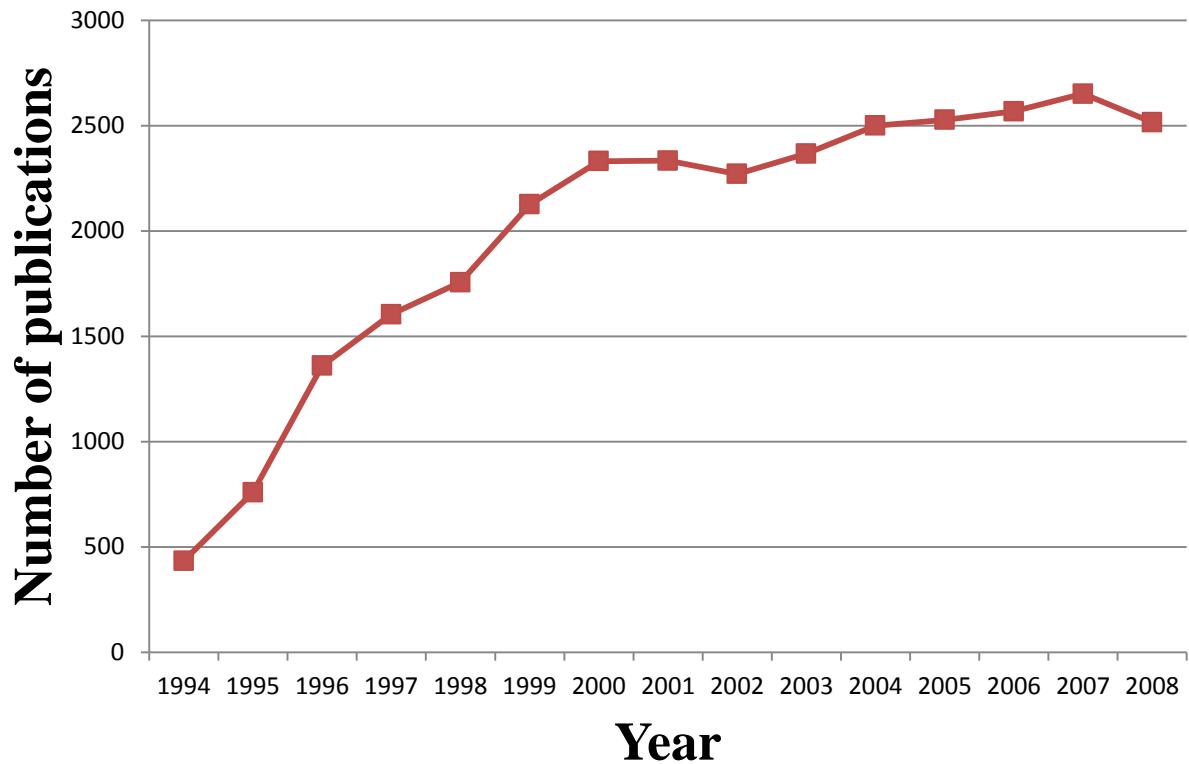


Figure 1.1: Number of publications with the keyword “microsatellite” in PubMed over the last 15 years.

One can reason from this that new microsatellite genotyping data is constantly being produced (most probably with increasing volumes as technology advances). As the accumulated datasets become larger, scientists are confronted with new challenges with regards to managing and analysing their data. Some of the most prominent problems faced by researchers include:

- The marker datasets have become too big to manage effectively in comma-separated-value (.csv) or Microsoft Excel files.
- It is very difficult to share working datasets across research teams and ensure that everyone has the same version of the project data, especially when teams are geographically distributed across the world.
- Deciding on the correct standard allele size for any given observed allele size is often problematic. The genotyping primary genotyping software may, for instance, score an allele of a dinucleotide repeat marker as having a size of 63.5, but the actual allele size could be 63 or 64. The reason that the observed sizes differ from the actual alleles is mostly due to slight variations in the experimental environment which cause the genotyping instrument to

measure the allele sizes a little larger or smaller than they actually are. A standardization step is therefore performed to categorize or assign (bin) each observed continuous size to the correct discrete allele. This either happens manually, or with the use of some automated software tool that has its own custom file input and output formats. After the automated binning algorithm has been applied to the dataset, the results still have to be inspected manually to ensure accuracy. This whole process is very unwieldy. It can be difficult to compare the initial observed sizes with results from automated binning and to compare the accuracy of different automated binning results obtained with different input parameters. It can also be very arduous to manually find and correct incorrect allele calls made by automated software.

- Merging genotype data from different experiments are even more difficult. These experiments could have been conducted at different times or in different laboratories using different instruments and protocols, and affected by environmental conditions. Often the resulting allele sizes differ by a few bases for two experiments done on the same individual. This lack of standardization makes it impossible to accurately compare or statistically analyse data from different experiments.
- The larger the datasets become, the more difficult they are to query. Simple tasks like searching for a matching fingerprint or finding all the genotypes for a specific sample could become very arduous.
- Other analyses functions like calculating genotype distributions for an entire dataset are also difficult.
- Analysis and management software are often not available to the entire research team as software can require licenses and can only be installed on certain operating systems.

1.4 Problem Statement

The aims of this study were to improve the management, standardization and analysis of marker genotype data through the development of a dedicated software application, and to apply the software to a genetic fingerprinting experiment aimed at the re-establishment or confirmation of clonal identity of *Pinus patula* ramets from a new pine clonal seed orchard against a reference set of DNA fingerprints from the original clone bank. Currently, researchers are struggling to effectively manage and analyse rapidly growing volumes of microsatellite genotyping data. Management problems range from the lack of a secure, easily accessible central data repository to more complex issues like the merging and standardization of data from multiple sources into combined datasets.

These larger datasets have to be utilized effectively in biological studies which often involve performing genetic fingerprinting analysis tasks such as identity matching and relatedness measures. For these reasons, the following functional goals were set for the software engineering project:

1. Users should have a secure, easily accessible database where they can manage their genotyping project data as a team.
2. Users should be able to easily upload and download project data.
3. Users should be able to standardize uploaded genotype data by way of both:
 - a. An automated binning tool
 - b. A manual standardization step
4. Users need a way to view allele frequencies for any given dataset and marker.
5. Users must be able to query standardized datasets. For the purpose of this study, only identity matching queries need to be implemented.
6. Users need a way to visualize the relatedness of samples in a dataset. For this study, a simple dendrogram visualisation has to be developed.

Along with the list of functional requirements, technical goals were defined to further enhance the value of the new software:

- The software should be founded on sound software design principles
- It should be relatively easy to extend its functionality in the future
- It should be easy to reuse components in different solutions. E.g. it should be possible to use the binning algorithms in isolation, or plugged onto different datasets.

Chapter 2 discusses the technical infrastructure and capabilities of GenoSonic, while the ways the functional requirements have been addressed in the implementation of GenoSonic are discussed in Chapter 3. Chapter 4 describes the application of the software in a genetic fingerprinting study and discusses results pertaining to the relative performance in terms of speed and accuracy in comparison to that of human experts.

Chapter 2

System architecture and design

This chapter explains the architectural-, technological-and design choices that were made during the creation of GenoSonic. The advantages of each pattern are discussed, elucidating the reasoning behind certain architectural-, technological-, or design decisions.

2.1 General design principles

Principles such as maintainability, scalability, extensibility, and interoperability were of great importance during the design of this application. Probably the most important principle to keep in mind was to create an architecture with low coupling. Coupling is the measure of how dependent one element is upon another, or how strongly elements are linked together. In this context, elements can be anything from classes to assemblies to entire modules. By having lower coupling, the impact of changing parts of the application can be reduced. This can be especially important in bioinformatics, as newer technologies and different research support requirements continually surface, creating the need for continual modification of the software. Another very important general principle was to keep a high cohesion when designing the architecture. This means keeping the functionality of each entity strongly focussed and to the point. High cohesion makes elements easier to understand, maintain, and reuse.

2.2 Overall architecture

This application was built using a layered and modularized system architecture. The layered part of the design meant slicing the system into layers that sit on top of each other, where the higher layers call functions from the lower ones. Each layer is responsible for a major role in the system, and has as little as possible direct influence on other layers in the system. The application logic was also arranged into different modules or libraries where each library provides a specific set of functionality.

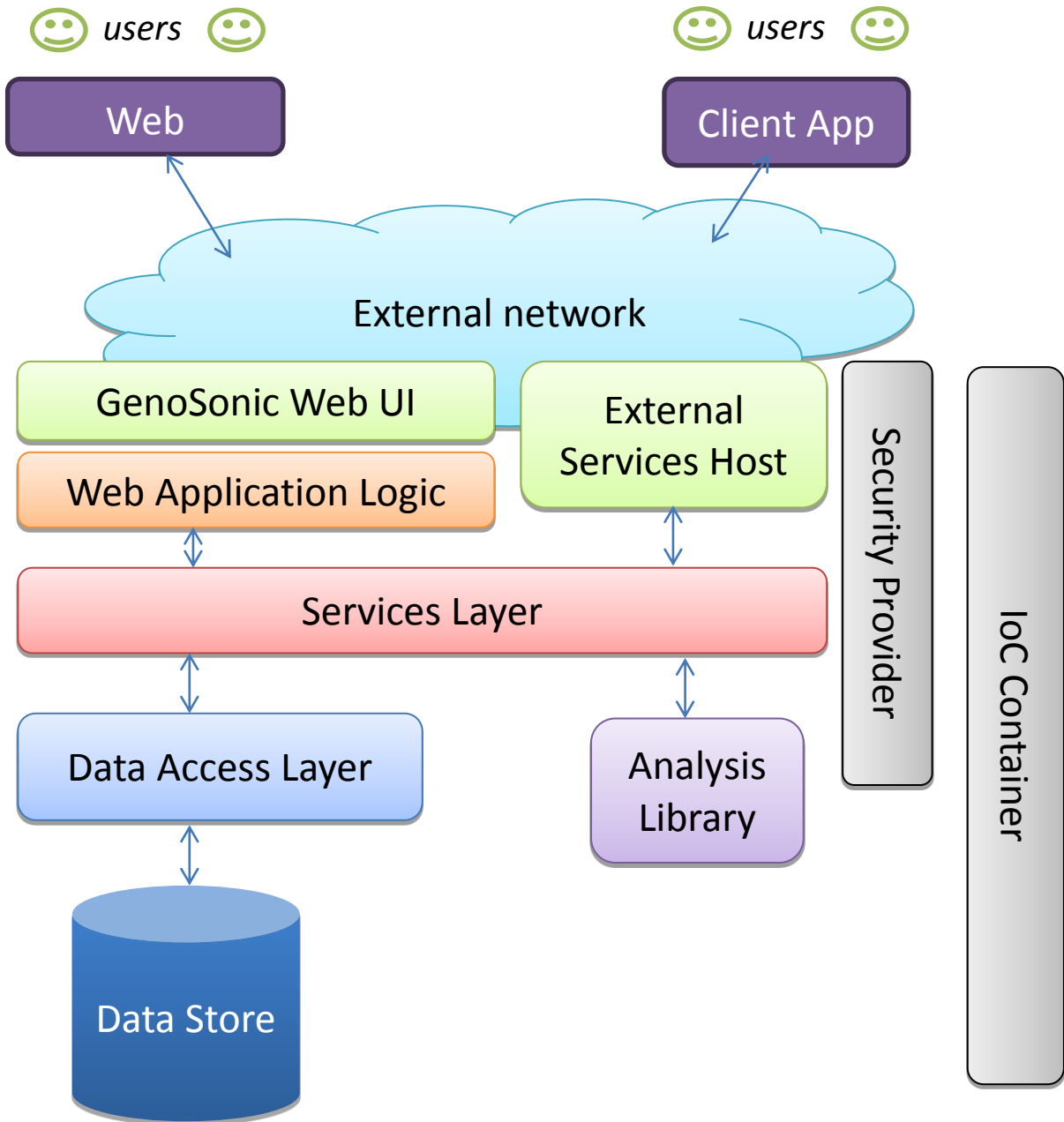


Figure 2.1: The GenoSonic System Architecture. *Main software components of GenoSonic together with points of interaction among the different components and external systems or users. GenoSonic consists of multiple layers, each with a focussed responsibility to form a coherent system.*

The main components of GenoSonic are shown in Figure 2.1. The diagram also shows the high-level interaction among different components, as well as the points of interface with users and other external clients. The main components of the system, as shown above, can be described as the following:

Data Store: This is where all the data is saved. This can be a relational database, XML file, text file, or any other data persisting mechanism.

Data Access Layer: This layer interacts directly with the data store and is responsible for all actions concerning the retrieval, persistence, manipulation, and deletion of data to and from the physical data store.

Analysis Library: This is an independent library that contains all the analysis functions written for GenoSonic, including binning, allele frequency calculations, identity matching, and tree construction.

Services Layer: This layer encapsulates nearly all the business logic of the system, and can be seen as the layer responsible for coordinating all server side activities. It consists of a set of services, where each service can be asked to perform a specific set of related tasks in GenoSonic. Together the services represent the complete set of operations any client application can invoke in GenoSonic.

External Services Host: This is an application interface layer that exposes a set of services to the outside world via secure web services for the most common data retrieval, data manipulation and analysis operations.

Web application layer: This layer consists of code that controls the look and feel of the website, as well as the code that interacts with the other layers and components in the application.

Security Provider: This is the component used to authenticate users in GenoSonic. In the default implementation GenoSonic uses ASP.NET's built-in SQL membership provider, but this can easily be changed to a different provider.

Inversion of Control Container: This component manages the lifecycles of all objects and their dependencies in GenoSonic. It is the mechanism that puts all the loose building-blocks together into a coherent system.

This logical N-layer approach creates a clear separation among user interface code, application controller code, domain logic and modelling, persistence and communication logic. This is called a multi-layered design. By decoupling the layers, one gains the ability to change one layer without having to change anything in the other layers, thus easing the impact of change. For example, this allows the data access logic to connect to interchangeable databases without having to modify anything in the service layer, application logic layer, or user interface layers. Another advantage is

that all related logic is organized together. The improved cohesion makes the application much easier to maintain, as changes need only be effected at one location.

The only real drawbacks to using a layered architecture are the effort and required expertise involved in creating the core components of the systems. A multi-layered infrastructure is much more complex than single layer applications, and requires a substantial amount of development before anything useful can be done. However, once it is up and running, the rewards become more and more substantial as the application grows in size and age.

2.3 Data Store

A number of requirements were taken into consideration in the choice of a suitable data storage mechanism. Firstly, very large amounts of data will potentially need to be saved. If this database is to handle all genotype data of multiple species, the data store must be able to efficiently handle large data sets. Secondly, because this is a distributed application, the data store must be able to handle concurrent connections. Thirdly, data within the database may change frequently, and many entries may need to be changed at once.

Possible technologies that were considered included XML files, a Microsoft Access database and higher-end Relational Database Management Systems (RDBMS), like Microsoft SQL Server, Oracle, MySQL, or PostgreSQL. XML files are more suitable for applications where the data is rather static, and very fast retrieval is not necessary. Another option would be an Access database, but although Access databases are very easy to set up and use, they do not scale well to large databases or concurrent users. The best option therefore would be to use a Relational Database Management System (RDBMS), like Microsoft SQL Server, Oracle, MySQL, or PostgreSQL.

For this project, Microsoft SQL Server 2008 was used, as it satisfies all the requirements of data store, and has the added advantages of being well integrated with Visual Studio (the development environment used), as well as being formally supported by the .net framework, the selected programming technology. As the data access layer is explained, it will however become clear that it would be easy to switch to a different data store if required.

2.3.1 Database Design

2.3.1.1 Overview

GenoSonic consists of two databases by default, namely GenoSonic and GenoSonicASPNET. The GenoSonic database stores the project and genotyping related data, while GenoSonicASPNET is

used to store all the membership related data, i.e. user profiles, login information and personalisation data.

2.3.1.2 Membership Database (GenoSonicASPNET)

This database is designed to work specifically with the ASP.net 2.0 membership provider. The schema can be generated automatically using the `aspnet_regsql.exe` command, which is part of the Microsoft .Net 2.0 framework. The reason this data is kept in a separate database is to decouple the security model from the scientific data. This will enable future developers to easily switch to a different security provider like LDAP (Lightweight Directory Access Protocol), Kerberos, or hook GenoSonic into any other in-house security system. The basic layout of the database is displayed in Figure 2.2::

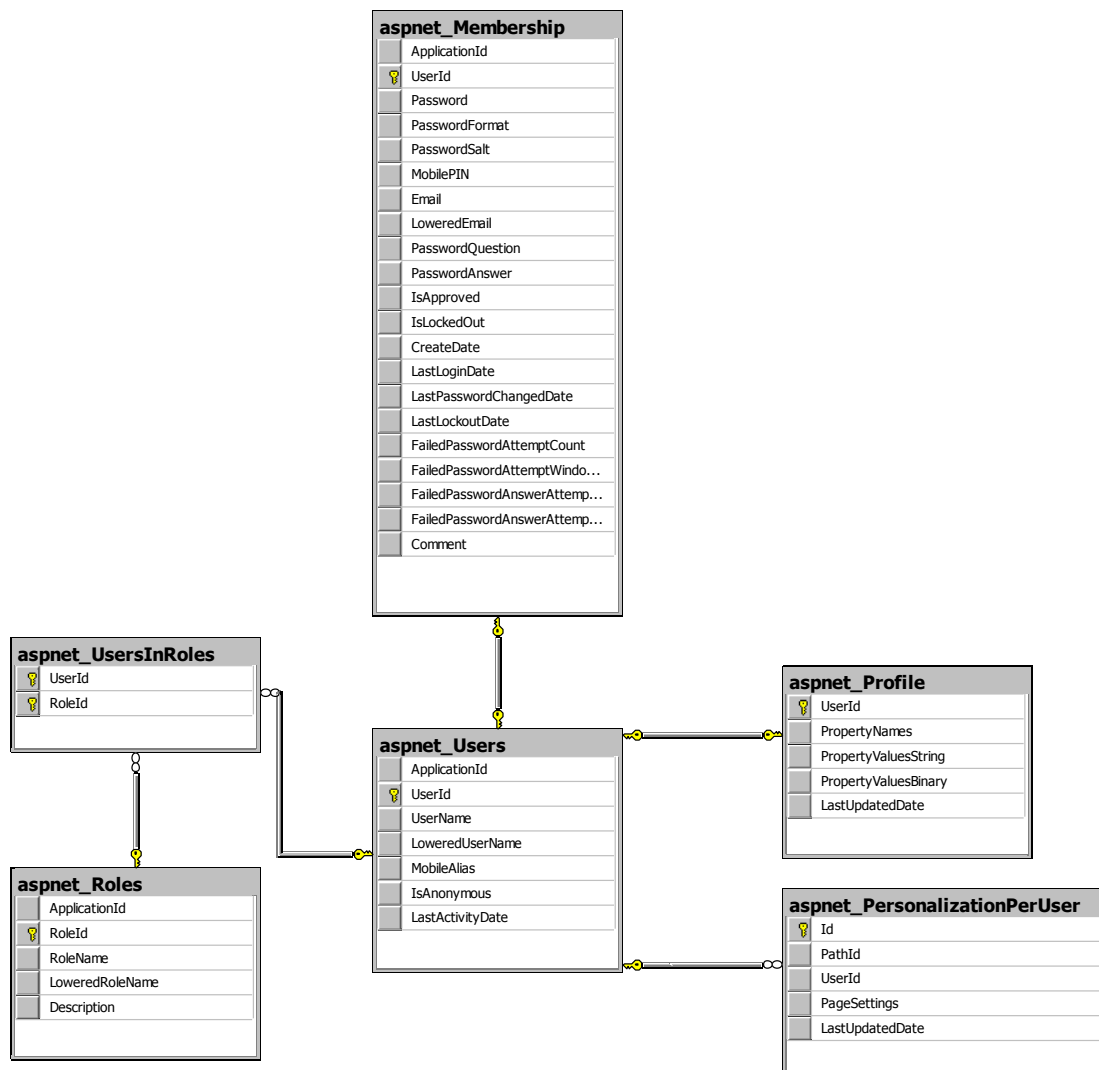


Figure 2.2: GenoSonic Membership database schema. *This database schema is used to store user credentials in GenoSonic’s ASPNet database.*

Aspnet_Membership: This table contains all security related data like password, last login, email address, authorization status, failed login attempts, etc.

Aspnet_Users: This table contains all the registered users.

Aspnet_Profile: This table contains all user profile information like title, institution, phone numbers, physical address, current working project, etc.

Aspnet_Roles: This table specifies all the roles that users can belong to in the system, for example user, power-user, curator, group administrator, or system administrator.

Aspnet_UsersInRoles: This table list every user-role combination, i.e. all the roles that a user belongs to.

2.3.1.3 GenoSonic Database

This database contains all the data relating to genotyping studies done in GenoSonic. The database design has been split into two diagrams for simplicity. The first diagram shows the table designs and relationships among project groups, projects and users. The second diagram depicts the structure used to store all the project data.

2.3.1.3.1 Project and Group Administration

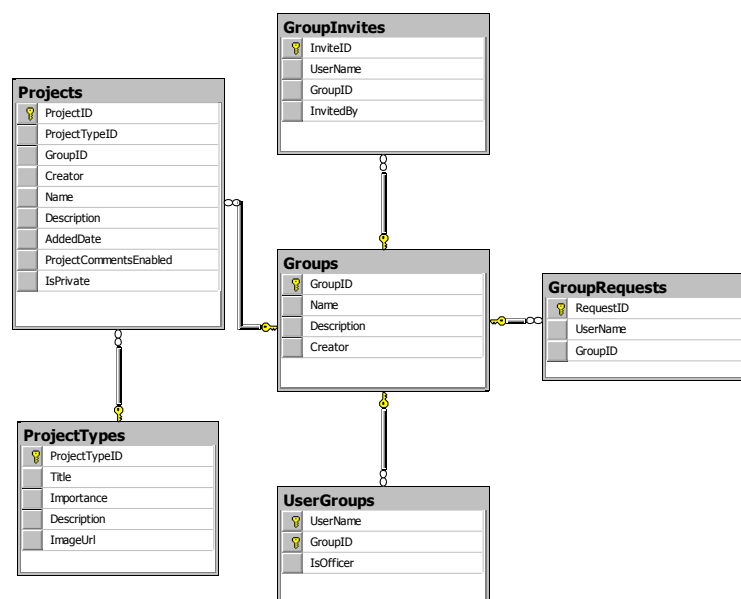


Figure 2.3: GenoSonic Project and Administration table schema. *This is a subsection of the GenoSonic database schema which deals specifically with project and group data. It shows how GenoSonic is able to link multiple users to multiple groups and projects and provide a way for users to join groups via invites and requests.*

Groups: Before a user can create any projects, he must first create a Group under which this project will be managed. This enables collaboration among researchers across different projects. For instance, a research unit may be working on a range of different biological studies at once and all the users in the unit should have access to all the projects being undertaken.

UserGroups: This table keeps track of the users belonging to groups.

GroupInvites: This table keeps track of all users that have been invited to join groups. Once the invite has been accepted or rejected, the entry is removed from the table.

GroupRequests: This table keeps track of all users that have requested to join groups. Once the user's request has been accepted or rejected by the group administrator, the entry is removed from the table.

Projects: This table keeps track of all the projects in GenoSonic. Every project must be assigned to a project group and a project type.

ProjectTypes: Every project is assigned a specific project type, e.g. microsatellite, SNP, or AFLP. This table will become useful once GenoSonic is extended to work with other marker types as well. Currently, 'microsatellite' is the only available option.

2.3.1.3.2 GenoSonic Project Data

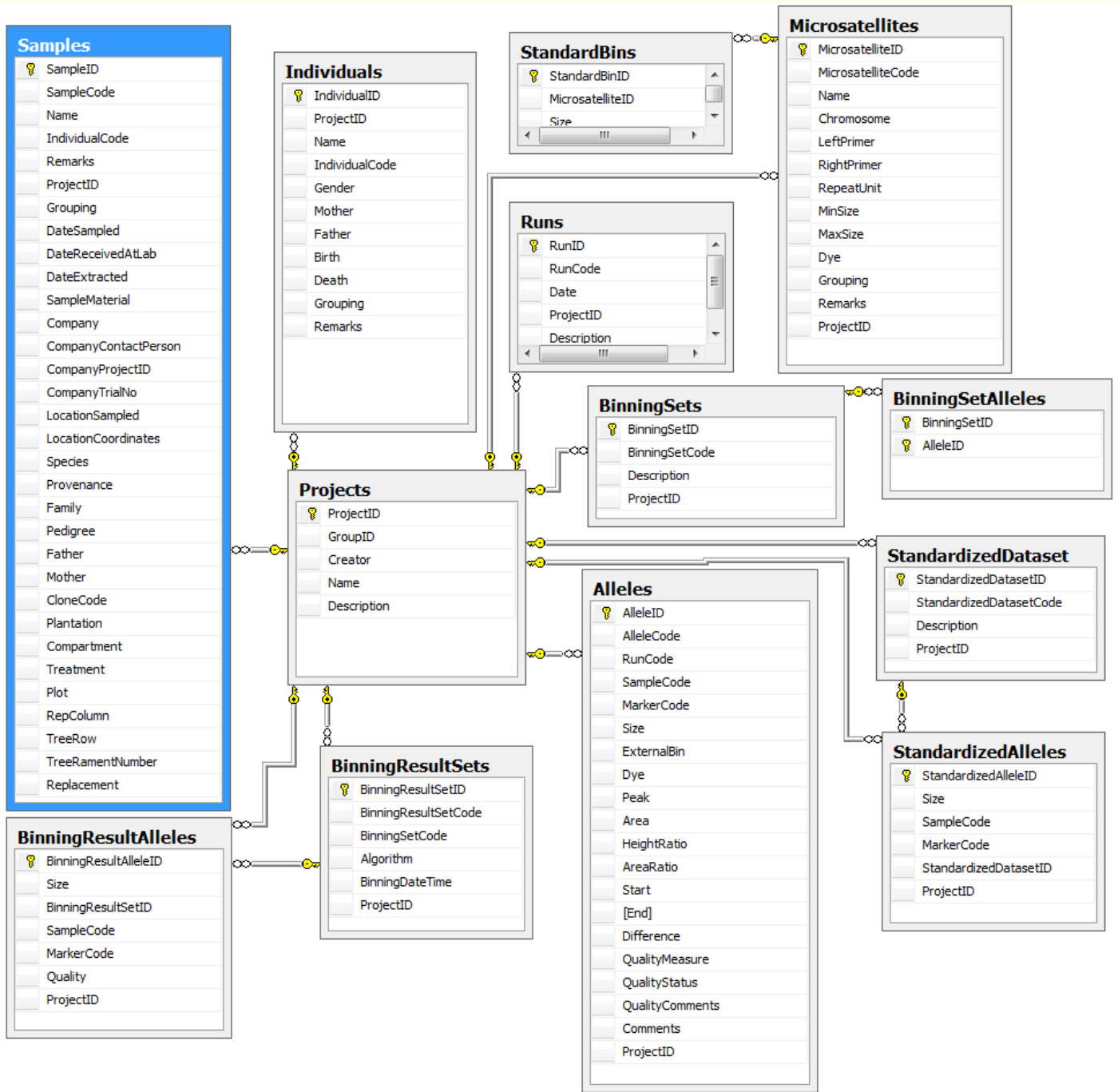


Figure 2.4: GenoSonic project data schema. *This database schema is used to store all project data in GenoSonic. It shows the table structures and relationships among tables which are used to store data like samples, alleles, marker data, and various analysis results.*

Microsatellites: This table keeps all the information about microsatellite markers. This includes the marker name, repeat unit, minimum and maximum size range, primers, and dye used.

StandardBins: Every microsatellite marker can have a set of predefined standard allele sizes. This information can be used to assess whether an assayed or binned genotype matches one of the predefined sizes before accepting it as a standardized genotype.

Individuals: This table contains information about individuals, like their names, date of birth and death, gender and parentage.

Samples: This table holds all the information about samples taken. This includes the individual code (if available), sampling date, location and information about who took the sample.

Runs: This table contains all the data about a specific experiments (or Runs) that were conducted on a set of samples. This table includes information like the date and description of the experiment.

Alleles: This table keeps all the result data from genotyping experiments. Every allele is linked to a sample-code, run-code, and marker-code to uniquely identify that specific result. More information like sample quality and predicted size calls that were created by the scoring machinery or software (like GeneMapper) are also recorded in this table.

BinningSets: Before genotypes can be used in analyses methods, they first have to be standardized. The data for the standardisation (Binning) process is grouped into BinningSets.

BinningSetAlleles: Every Binning set is linked to a group of alleles via this table. The set of alleles will then be the input to a binning process in attempt to automate the normalisation or standardisation of allele data.

BinningSetResults: This table contains information about the results of a binning process, like the date and method used to bin the BinningSet.

BinningSetResultAlleles: This table contains the new allele sizes of alleles as predicted by the binning operation.

StandardizedAlleles: This table contains all the called allele sizes by sample, marker and run that were deemed members of predefined standard sizes. The standard alleles will be the ones used in analysis functions.

StandardizedDataset: StandardizedAlleles can be grouped into StandardizedDatasets to use as input in analysis functions.

2.3.1.3.3 GenoSonic Upload Templates

The upload template tables store mappings between the GenoSonic table-column names and the column headers in .csv (comma separated value) files.

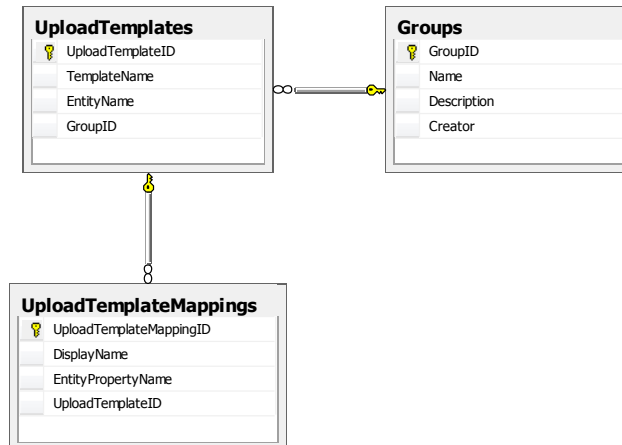


Figure 2.5: GenoSonic Upload templates schema. *A set of upload templates can be created for every study group which stores mappings between the GenoSonic table-column names and the column headers in .csv (comma separated value) files, thus enabling users to upload custom file formats*

UploadTemplates: Each template is specific to a certain type of entity in GenoSonic, i.e. a Sample, Individual, Run, Allele, or Microsatellite. Templates are saved at Group-level, which means the same templates can be used to upload data in multiple projects in a specific study group.

UploadTemplateMappings: This is the column-to-column mapping. The DisplayName will be the header of the column in the .csv file, while the EntityPropertyName is the name of the property (or database column) to which it maps.

2.4 Data Access Layer

This layer is responsible for all actions concerning the retrieval, persistence, manipulation, and deletion of data to and from the physical data store. This is the only code that interacts directly with the database and must therefore be aware of all the details of the database, i.e. the connection settings, the tables, fields, and stored procedures available within the database. The reasons for keeping all data access logic separate are as follows:

- Many times the same database queries must be called from multiple locations in the application code. Having the data access logic in one place saves developers from having to duplicate the same access logic across multiple pages.
- Having hard-coded queries inside a user interface or even business logic will make it immensely complicated to switch to a different data store, as each of these queries will need to be tracked down and changed one by one.
- By having all the data access logic in one place, a developer can focus on the user interface or business logic without having to know anything more about the data access layer than its interface to the outside. Once the data access layer is completed, it can therefore be treated as a black box, with a known set of operations with which to interface.

2.4.1 SubSonic

The data access module is built on top of an open-source third party toolset called SubSonic 3.0 (Conery 2009a; Conery 2010) which was created by Rob Conery (Conery 2009b). SubSonic was specifically chosen because it is open-source, extremely flexible and easy to use. Other similar data-access toolsets and object-relational mapping frameworks that were considered include Microsoft's LinqToSql and NHibernate. SubSonic provides the functionality that simplifies communication with the data store and abstracts all database specific code away from the rest of the application. SubSonic exposes an easy to use IQueryable interface which can be used to write LINQ (Meijer *et al.* 2006) query expressions. The Microsoft Language Integrated Query (LINQ) is a set of extensions that allows one to query data using strongly-typed queries and return strongly-typed results within C#. Upon executing these LINQ queries, SubSonic translates the query expressions into database queries that are specific to the database being used. SubSonic then also parses the results back into the strongly-typed objects expected. The following simple example shows how to select all the records where the species is human from the Sample table in the current database, and then convert the records into a list of strongly typed Sample objects. All data access logic in GenoSonic is done via these types of LINQ or IQueryable<T> expressions:

```
var samplesQuery = from s in new SubSonic.Linq.Structure.Query<Sample>()
                   where s.Species == "Human"
                   select s;

var samples = samplesQuery.ToList<Sample>();
```

The advantages of this method are significant. Firstly, the application need never concern itself with any database-specific language syntax as it is translated by SubSonic's built-in functionality. Secondly, this would allow one to effortlessly switch the underlying data store without having to change any data access code except the connection string. Thirdly, because queries are strongly-typed, it provides compile-time assurance that the queries are well-formed.

2.4.2 T4 Templates

Microsoft Visual Studio's Text Template Transformation Toolkit (T4) was employed to generate all the entity classes such as the `Sample` class from the previous example, as well as the Repository classes which serve as the entry points to the data access layer from the rest of the application. Basically, T4 is a text-generation language built into Visual Studio. Code generation is achieved by creating a set of T4 template files within a project which Visual Studio then executes as scripts every time they are saved. GenoSonic employs a set of T4 templates modified from the ones created by Eric Kemp (Kemp 2009), which uses the open-source T4 Toolbox (Sych 2009) provided by Oleg Sych. These templates examine the GenoSonic database schema and then create a matching entity model, query surface and a set of repository classes.

By using code generation, the entity model can easily be synchronized with the database schema. Once the templates have been created, there is no need for tedious duplicate work like creating both the database table or field and the class or property in the code, nor does one constantly have to make sure that the models (or mappings between them) are properly synchronized. In order to add another type of data to handle one can simply create or update the table in the database, execute the template and compile the code.

One drawback of generating the domain model from the database schema relates to a common problem in the world of object-oriented programming called the **object-relational impedance mismatch** (Elmasri & Navathe 2006). Essentially it is a set of conceptual and technical difficulties that commonly occur when trying to map the relational database schema to the classes defined in the object-oriented programming space. The basic problem is that the data residing in a relational database must somehow be routed, or mapped, to the objects used in the business logic domain. If the database structure is tied too closely to the object space, eventually one would either have to sacrifice good database practices in order to work comfortably in object space, or conversely, lose the power of object oriented programming by bending it into a shape that fits the relational database. The other possibility is that the database schema is very different from the object space, in which case the mapping of data between the two layers can become very complicated.

GenoSonic has opted to generate the object space from the database schema, since the model is fairly simple and the time saved with code generation overshadows the clinical beauty of a true domain-model together with a complex mapping schema to the database.

As a side note: Specific smaller domain models that better fit its purpose in the code have indeed been created in certain cases, as in the case of the Analysis Module, whose model does not match the database perfectly. Here a set of mapping functions have been created to convert data between the Data Access Layer's model and the Analysis Module's model.

2.4.3 The Repository Pattern

The repository pattern (Fowler 2002) is used to provide another layer of abstraction between the data mapping layer (SubSonic) and the rest of the application. Repositories serve to further decouple the data access layer from other application logic by acting like an in-memory domain object collection. Application logic that needs access to the data store can query and manipulate the repositories in the same way it would interact with simple object collections. The code encapsulated by the repositories then carries out the actual mapping and query operations while the application logic need not have any knowledge about the specific implementation. This pattern provides a clean interface through which all data access operations are mediated and creates a true one-way dependency between application logic and the data access layers. Here is an example of the generic Repository interface provided by GenoSonic that is extended by the specific repository interfaces and implemented in all the repositories:

```
public interface IGenoSonicRepository<T>
{
    IQueryable<T> Query();

    IList<T> Search(string propertyName, string value);

    object Add(T item);
    void Add(IEnumerable<T> items);

    int Update(T item);
    int Update(IEnumerable<T> items);

    int Delete(T item);
    int Delete(IEnumerable<T> items);
    int Delete(object key);

    int DeleteMany(Expression<Func<T, bool>> expression);

    T GetByKey(object key);
}
```

The generic repository has been extended further in certain cases to fit the needs of the specific repositories:

Partial interface generated by T4:

```
public partial interface IAlleleRepository : IGenoSonicRepository<Allele>
{
    void Save(Allele item);
}
}
```

The other part of the interface created manually in the custom code:

```
public partial interface IAlleleRepository
{
    void Save(Allele item, Expression<Func<Allele, bool>>
expressionToFindExistingRecords);
}
}
```

In order to query the Alleles table in the database, application logic instantiates an instance of an object that implements the `IAlleleRepository` interface and interacts with it by calling any of the methods defined above. It needs no further knowledge about the inner workings of the query logic.

This diagram shows a subset of the repository interface hierarchy:

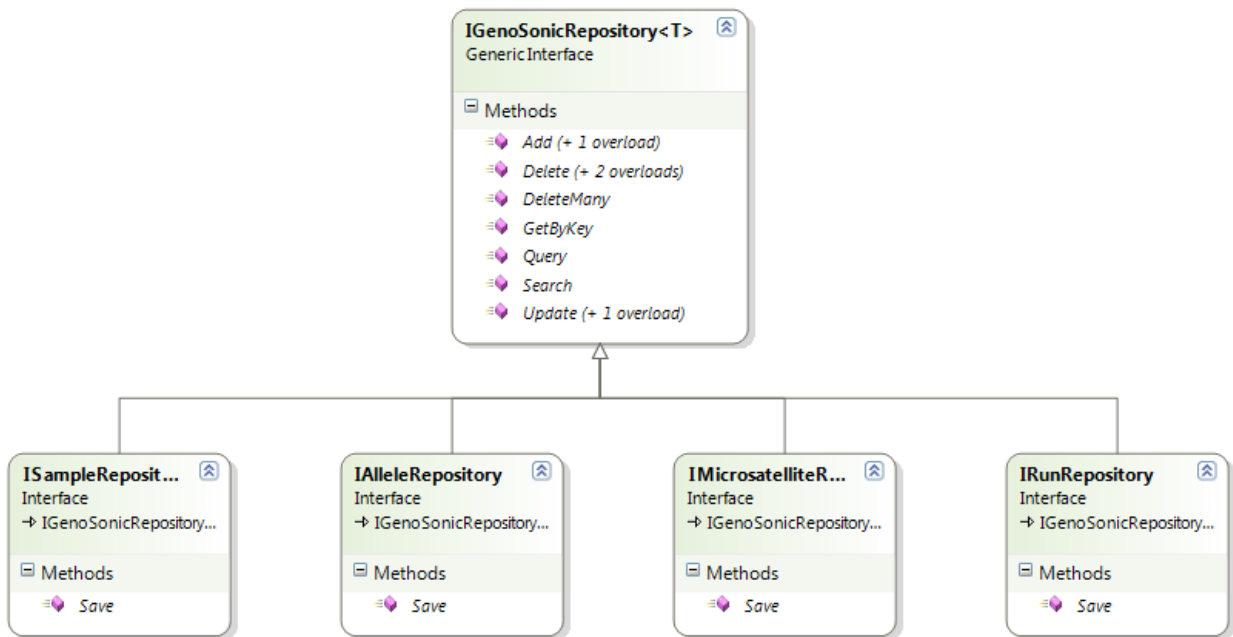


Figure 2.6: Interface hierarchy of data access repositories. *All repository interfaces extends a set of basic data-access members from the `IGenoSonicRepository` interface, while providing the opportunity to specify type-specific members on individual level.*

These interfaces are then implemented by the following class hierarchy:

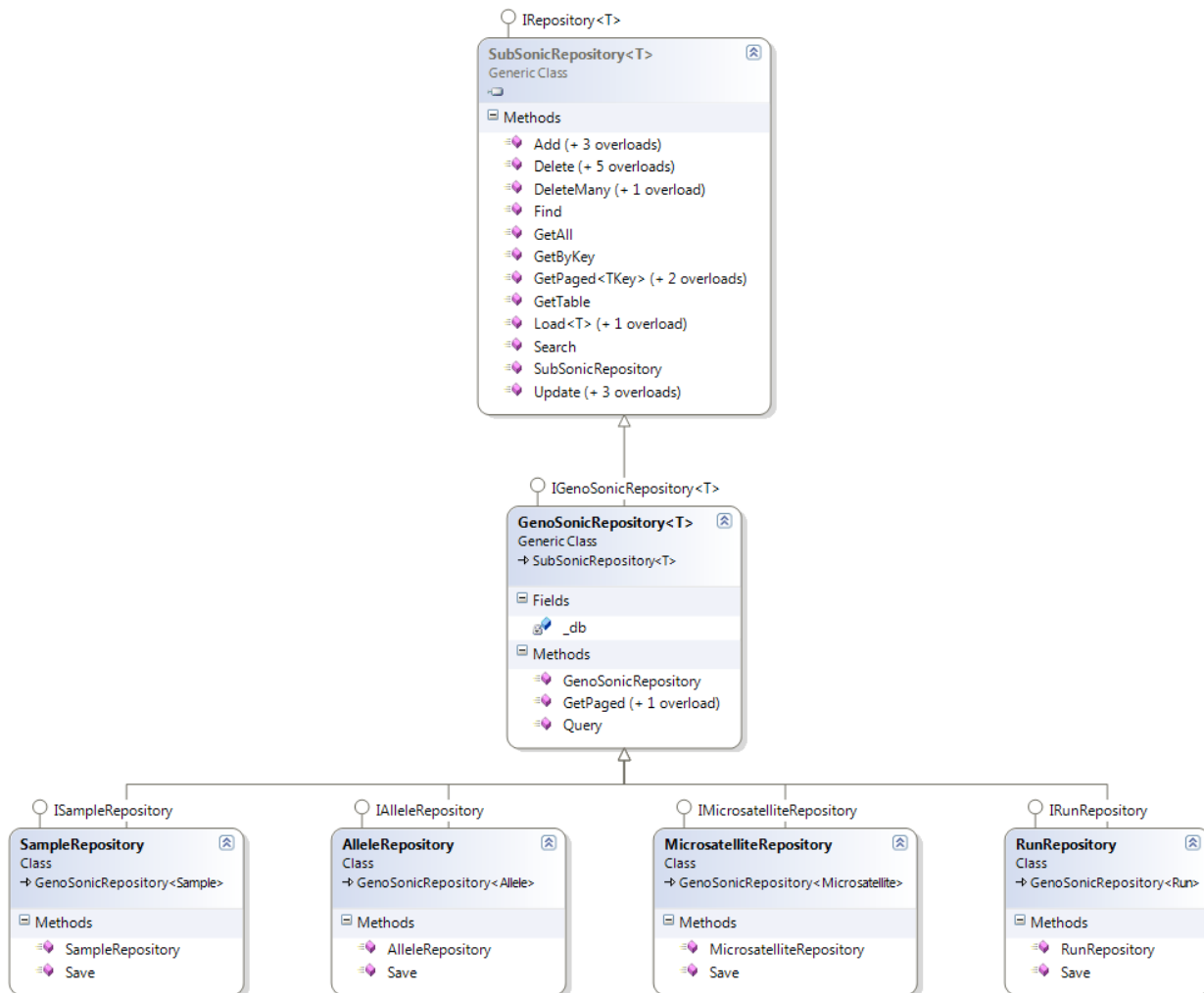


Figure 2.7: Class hierarchy of data access repositories. *The generic `GenoSonicRepository` inherits a set of basic data-access members from `SubSonic's SubSonicRepository` class, while the lowest level specific repository classes provide type-specific functionality on individual level.*

2.5 Analysis Library

This library contains all the analysis functions written for `GenoSonic`, including binning, allele frequency calculations, identity matching and tree construction. It was intentionally written to be independent from the other `GenoSonic` libraries so that it can be reused easily in other applications like command-line programs or desktop applications.

2.5.1 Model

In order to make the analysis library independent from the rest of `GenoSonic`, another simpler domain model had to be defined specifically for the analysis sub-domain. This is beneficial as the model can be shaped to support the analysis functions much more easily, without having to consider the impact on the rest of the system. Mapping functions were then built to convert the data between

the GenoSonic Data-Access domain model and Analysis sub-domain model. This is the basic model used by the analysis library:

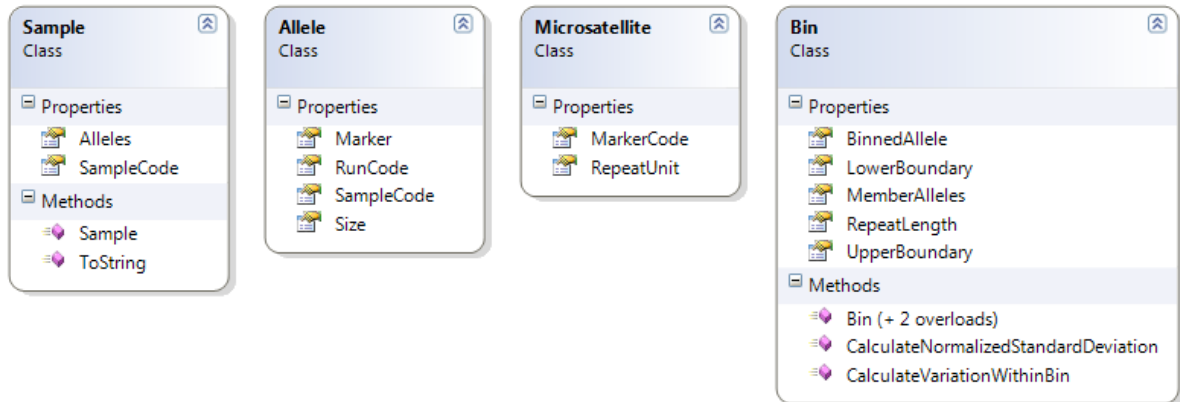


Figure 2.8: Analysis library entity models. *This is a simplified domain model, compared to the one used for data-access, designed specifically for the needs of the different binning and identity matching functions within the analysis library.*

In addition to the common entities shown above, each analysis section has a sub-domain model containing entities relevant only to the specific analysis functions. For instance, the Identity Matching section has entities for carrying identity-search-results, which are not shared with other sections.

2.5.2 Services

Each of the analysis functions were designed as self-contained services that implements one or more interfaces. The analysis library includes binning services, a relatedness service, identity matching service and a frequency service. Each service is accessible to external libraries via the specific functions available on the service interfaces. This makes the services easy to use and easy to replace with other implementations in the future without any impact on the rest of the application.

2.5.2.1 Binning

Binning can be defined as the assignment of standard names, or “bins”, to observed allele sizes for example the initial size may have been scored as 63.5 whereas the actual size (in terms of base pairs) of the repeat length is either 62 or 64. Automated binning algorithms like Allelobin (Idury RM 1997) have been developed to rapidly cluster real allele sizes into standard bins. Allelobin is essentially a curve-fitting algorithm that categorises continuous allele sizes into discrete bins by minimising the average standard deviation between bin midpoints and allele sizes. This algorithm was, however, not designed to merge data from different experiments. Because of differences in

experimental conditions, machinery, or protocols, there are always differences observed allele sizes which may lead to ambiguities with regard to the true nature of some alleles. These differences may result in alleles being binned incorrectly, which would adversely affect all further analysis relying on this data. With larger volumes of data, it is virtually impossible for a human to manually negate experimental noise and uniformly assign continuous allele scores to the most probable discrete bins, making the process very time-consuming and error-prone. It is for these reasons that automated binning algorithms have been developed.

GenoSonic has implemented two separate binning algorithms. The first one is an adaptation of the Allelobin algorithm described in (Idury RM 1997), which can be used for isolated experiments. The second is a novel algorithm all CSMerge-1, which has been developed to address the issues with merging data from different experiments.

2.5.2.1.1 CSMerge-1

The default binning algorithm implemented in GenoSonic is a completely novel algorithm called CSMerge-1. This algorithm has been designed specifically to merge genetic fingerprinting data from different runs, especially where certain samples are genotyped in more than one run, and then to accurately bin the merged datasets.

The CSMerge-1 algorithm consists of three major steps. The first is a progressive alignment step designed to align allele scores from different runs by comparing fingerprints from all shared samples between any two runs. This is done by minimising the total Euclidian distance between shared alleles by uniformly increasing or decreasing the allele sizes of either runs progressively. The second step is a clustering algorithm designed to group alleles with similar sizes together and to assign a standard size, or bin name, to each of the member alleles. The last is an optional step which will attempt to automatically choose the two best alleles (if the species ploidy is two) for samples that have more than two alleles assigned to them for any given marker, most probably because the sample has been assayed in multiple runs and has demonstrated more than two possible true alleles over the different runs.

The goal of the first step is to align the allele scores of different runs as closely as possible so they can be binned more accurately in the next step. This step begins by creating a list of shared alleles for each run that have similar sample-marker combinations with other runs. For each shared allele, lists are also constructed of all the possible other alleles in other runs that are in the same size vicinity, up to a user-specified offset. All of these possible matchups are called connections. The two runs with the most connections between them are aligned first. The distance that sizes may be

shifted is set to two base pairs by default, but can be changed by the user. This means that runs with alleles that have been misaligned by up to four base pairs can be aligned perfectly by increasing the all allele sizes of one run by two base pairs while decreasing the other run's sizes by two base pairs. The two runs are aligned by starting with a size shift of zero and then iteratively increasing the offset from zero into both the positive and negative directions for each combination of shifts and for either run. At each iteration, the fitness of the current alignment is assessed by calculating the average Euclidean distance between the allele sizes of each connection. The alignment where the average distance between the two runs is the lowest is then used. The sizes of all member alleles of the two aligned runs, not just the shared ones, are then adjusted according to the optimum shift that has been observed. If for instance, two runs with a given marker share sample and a single connected allele with a size of 40.2 in the first run and 39.9 in the second run, the algorithm could align these runs by decrementing all of the alleles in the first run by 0.2 while incrementing the second run by 0.1. This whole process is then repeated for the next two runs with the most connections between them. If one, or both, of these runs have been aligned before, the distance by which the sizes of each particular run can be adjusted are restricted so that the total size shift of a run and all of its aligned runs will always stay within the boundaries of the maximum shift (two base pairs) set by the user. Once the alignment between these two runs has been done, all the allele sizes for each run and all their previously aligned connected runs also have to be updated. This is achieved by creating a graph structure where each vertex is a run and each edge is an alignment between two runs. Every time a new run is to be aligned, it is added to the graph. Two runs can only be aligned if they are not yet connected via any path in the graph. Each run being aligned can only shift within the boundaries of all the previously aligned runs connected to the current run in the graph. After alignment, the allele sizes of all the runs in the graph connected to the current runs are adjusted according to the optimum shift. This process is repeated until there are no more runs that can be aligned.

The next step in CSMerge-1 is the clustering step. The goal of the clustering step is basically to recognise closely related measurements from genotyping machinery as being the same alleles. CSMerge-1 implements a version of a clustering technique called Quality Threshold (QT) Clustering (Heyer *et al.* 1999). This technique was originally invented for gene clustering, but could easily be adapted to cluster alleles as well. This method was chosen because the number of clusters need not be specified *a priori* and because the quality threshold of clusters could be user specified. The algorithm in CSMerge-1 works as follows: First a complete set of potential clusters is created. This is done by iteratively creating a cluster for each allele for a given marker and grouping all other allele sizes within a user-specified maximum threshold into another cluster. Next, the cluster with the most alleles is chosen as the first output cluster. If multiple clusters have

the same number of alleles, the cluster with the best standard deviation is chosen. The chosen cluster is then added to the output list and all member alleles of the chosen cluster are removed from all the other potential clusters. The same steps are repeated for the next best cluster, over and over until all alleles have been added to clusters in the output set. Alleles are then assigned the natural number closest to the mean of each cluster as their new binned sizes.

The final step in CSMerge-1 is an optional step which aims to reduce the number of unique alleles for every sample-marker combination to at most the given ploidy of the current species. For example, for a given sample from a diploid organism and a given marker, it will attempt to choose only the two allele sizes of the best quality from the set of possible alleles that may have been scored differently in different runs. This is done by simply choosing the allele sizes that occurs the most across all samples in the binning set.

2.5.2.1.2 Allelobin

Another optional binning algorithm implemented by GenoSonic is based on the Allelobin algorithm described in (Idury RM 1997). It is essentially a curve-fitting algorithm that categorises continuous allele sizes into discrete bins by minimising the average standard deviation between bin midpoints and allele sizes. The input parameters required to execute the binning algorithm on the service are the following:

- **Maximum ruler shift:** This parameter indicates the maximum distance the midpoint of the bin can be offset to either side of the true allele. For instance, if the maximum ruler shift is 0.5, the algorithm will move the midpoint of the bin from $x-0.5$ to $x+0.5$ in search of the bin location where the average distance between the bin midpoint and the member alleles are at a minimum.
- **Number of ruler shift intervals:** This parameter indicates the number of intervals the algorithm will use when iteratively moving the bin midpoint from $x-0.5$ to $x+0.5$ (given of maximum ruler shift of 0.5).

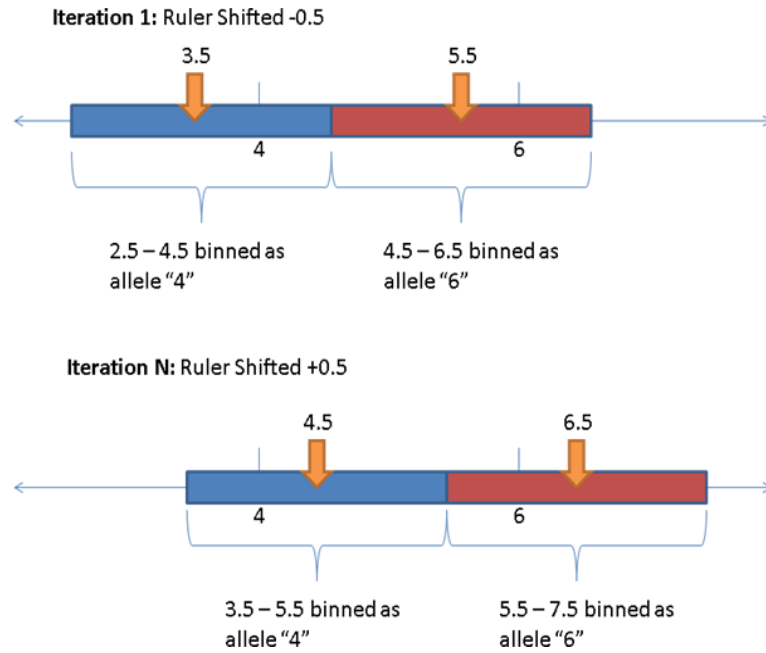


Figure 2.9: Illustrating the function of the ruler shift parameter in the binning algorithm. *The figure shows the first and final states of the binning algorithm as it moves the candidate binning configurations from an offset of -0.5 to +0.5 in an effort to find the offset where the mean distance between the midpoint of each bin and its member allele sizes are at a minimum.*

- **Maximum size shift:** This parameter indicates the maximum size difference a bin can have compared to the true repeat length of the marker. For instance, if the maximum size shift is 0.1, the bin size can vary between values 1.9 and 2.1 for a di-nucleotide repeat marker.
- **Number of size shift intervals:** This parameter indicates the number of intervals the algorithm will use when iteratively changing the bin size from $x-0.1$ to $x+0.1$ (given of maximum size shift of 0.1).

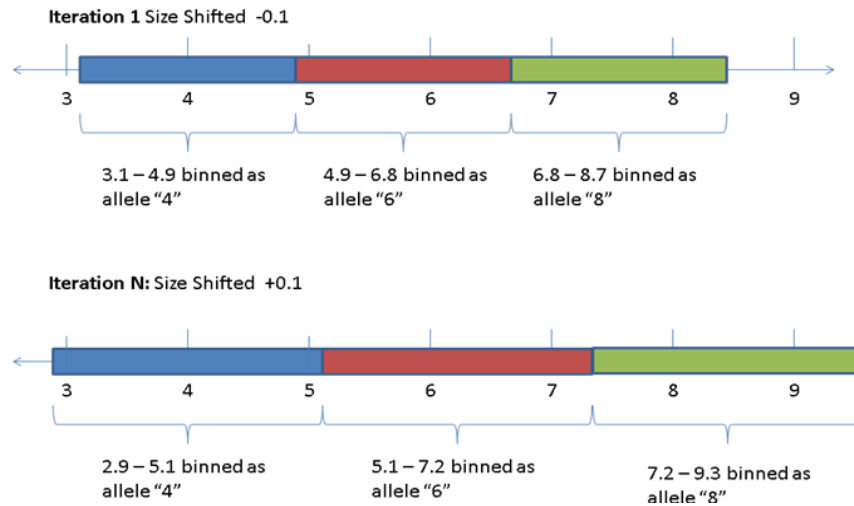


Figure 2.10: Illustrating the function of the size shift parameter in the binning algorithm. *The figure shows the first and final states of the binning algorithm as it changes the candidate binning configuration's bin sizes in an effort to find the bin size where the mean distance between the midpoint of each bin and its member allele sizes are at a minimum.*

- The last parameter required by the binning service is the list of input alleles. These input alleles can originate from many different samples, markers, and runs.

The objective of the algorithm is to assign every allele to a specific bin in such a way as to minimize the average distance between the midpoints of the bins and allele sizes. The first step in the binning function is to group the input alleles by run. Thus alleles originating from different experiments will be analysed separately. For each run, the binning algorithm will then analyse the fitness of every possible bin configuration by iterating over every ruler- and size-shift combination. A new bin configuration is constructed for every iteration. The first bin's lower boundary is always set to be smaller than the smallest input allele size and the upper boundary of the largest bin is set to be larger than the largest input allele size. The bins are constructed with upper- and lower boundaries adjacent to one another. In other words, bins do not overlap and there are no open spaces between bin boundaries. After the bins have been configured, the input alleles are assigned to the bins. An allele is deemed to be a member of a bin if its size falls between the lower- and upper boundaries of the bin. The fitness of the current bin configuration is then assessed by calculating the average standard deviation between bin medians and their respective member alleles over all the bins in the configuration. This process is repeated for every ruler- and size-shift combination. The bin configuration with the best fitness after all iterations have been completed is noted and the output allele sizes are then set to the label size of their respective bins using this optimal bin configuration.

If alleles from the same sample and marker combinations are present in multiple runs, the algorithm offers the option to automatically choose the alleles from the run with the best fitness as the true

alleles. This is done by comparing the fitness values from the algorithm described above and using the allele sizes with the lowest standard deviation. In addition, the algorithm offers the optional use of the quality measure fields generated by scoring equipment and uploaded with the allele sizes in order to determine which run's alleles to include in the result set.

2.5.2.2 Relatedness

The relatedness service contains a set of graph-drawing functions designed to visualize a measure of relatedness among samples. The service provides three functions, defined by this interface:

```
public interface IRelatednessService
{
    Graph ComputePairwiseRelatedTree(List<Sample> samples);
    string ComputePairwiseRelatedTreeAsNewick(List<Sample> samples);
}
```

These methods produce a graph structure from the given input using certain graph-building rules together with an implementation of an IDistanceCalculator which is responsible for providing a measure of the genetic difference between two given samples. The first method returns the actual graph structure in object-oriented format, while the second method converts the graph structure into a standard text notation for drawing tree structures called the Newick Tree Format.

The default tree drawing algorithm implemented in GenoSonic is a pairwise relatedness algorithm. The algorithm initializes by calculating the distances between all samples in the dataset. It also creates a graph containing an unconnected node for each sample in the dataset. The nodes of the two samples in the dataset with the shortest distance between them are then connected by creating a new parent node and drawing edges from this parent to each of the two child nodes. The parent node can be seen as a combination of both its children as it carries the alleles of both. Once the parent node has been created, the distance measures between it and all other unconnected nodes are calculated and the next two nearest samples are combined. This iterative combining of nodes continues until the last two unconnected nodes are combined to form the root of the tree.

By default, the tree drawing algorithm uses the following measure to calculate the distance between two samples: For each marker shared between two samples, the alleles in the second sample are found that are the nearest in size to the ones in the first sample. For each nearest correlation between member alleles of the two respective samples, the distance is calculated as the difference in size divided by the sum of the two sizes. These distances are summed together for each marker and

divided by the number of alleles of the sample with the least alleles. Lastly, the distances of all markers are summed together to produce a single distance value.

2.5.2.3 Identity Matching

The Identity Matching service provides a single method called `GetSimilarSamplesForDataset` that takes a list of unknown samples together with a list of reference samples and computes a data table with resulting probabilities of matches for each sample in the reference set.

The `GetSimilarSamplesForDataset` method also specifies two input parameters for setting the minimum required threshold for any pair of samples to be characterized as being similar. The first parameter is simply the minimum number of alleles that has to match exactly. The second parameter is the minimum probability of identity. The probability of an unknown sample matching any given sample in the reference set is calculated as the chained product of the probabilities of each of the matching alleles occurring in the test sample. Missing alleles are not treated as differences but are ignored in pairwise comparisons. Allele probabilities are calculated on the fly by assessing the allele distribution of the reference set. Thus the probability of an allele occurring is calculated as the number of occurrences of that specific allele in the reference set divided by the total number of alleles in the reference set. If both tests surpass the minimum threshold of similarity for the test sample in relation to a sample from the reference set, the match is added to the output set.

The output result-set is constructed as lists of samples from the reference set that are similar to the each of the unknown test samples respectively. Each result includes the name of the matching sample, a list of the matching alleles, the probability of identity for those matching alleles (as explained above), as well as the probability of non-identity. The probability of non-identity is a measure of dissimilarity between two candidate matches and is calculated as the chained product of the probabilities of each of the alleles in the reference sample that does not match any allele in the test sample. These lists of results that are returned by the service are passed to the presentation layer where they are formatted and displayed as tables of probable matches for each test sample which the user can easily interrogate.

2.5.2.4 Allele Frequencies

This is a simple service that returns a list of allele-or genotype frequencies, given a set of input alleles. Allele frequencies are the proportions of particular allele sizes occurring within a given set of input alleles.

2.6 Services Layer

This layer consists of a set of services which represents all the operations a user can invoke in GenoSonic. Each service can be asked to perform a specific set of related tasks in GenoSonic. It is important to realise that the web application layer, external services host, and all user interface-related code must communicate with this layer to perform any data retrieval or intelligent operation in GenoSonic. The services layer creates a clean and simple surface through which all application requests must pass and hides all the business logic and complex implementation details from the higher layers of the system.

This layer is a more a logical separation than a physical library, as the code that make up this layer is actually spread across different physical libraries. Logically, the service layer can be divided into two main groups, namely data services and analysis services.

2.6.1 Data services

The data services exposes operations with which to query, insert, update, and delete records in GenoSonic. Each service typically concerns itself with a single type of record in GenoSonic. Currently, the following services are available:

- AlleleService
- BinningResultService
- BinningInputSetService
- GroupService
- IndividualService
- MicrosatelliteService
- ProjectService
- RunService
- SampleService
- StandardBinServices
- StandardizedDatasetService
- UploadTemplateService

As an example of the typical operations available on a data service, consider the SampleService definition:

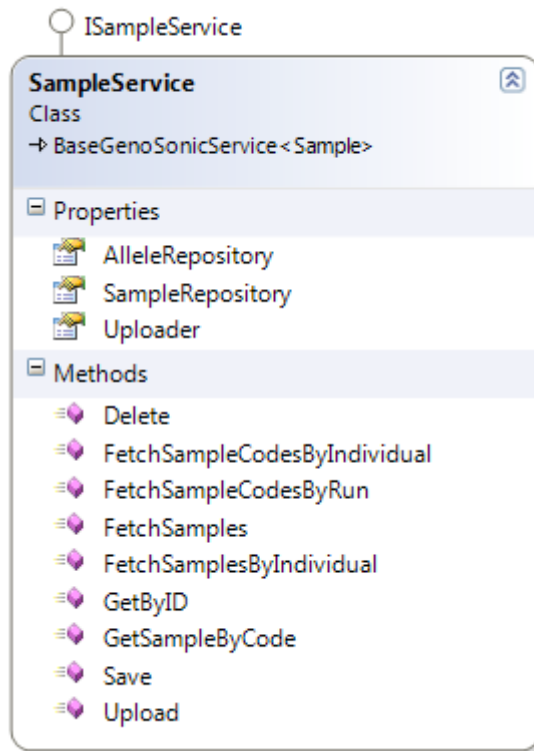


Figure 2.11: SampleService Definition. *This service contains operations to retrieve, save, upload, and delete sample records in GenoSonic.*

The service interfaces and accompanying entity models of the data services are located in a Contracts library, while the data service implementations are situated in the Core library. The reason for this physical separation has to do with client- and server-side requirements when communicating via web services. A client application needs knowledge about what a service looks like, i.e. its interface, but should not have access to its internal workings, i.e. its implementation. By separating interface and implementation, one is able to provide untrusted client applications with the Contracts library as part of a client software development kit so they can easily talk to the GenoSonic WCF services, but keep the Core library with the service implementations on the server side.

2.6.2 Analysis services

The analysis services are specified to each encapsulate a specific kind of analysis function in GenoSonic. These services are discussed in-depth in the Analysis Library section. Currently, the analysis services available in GenoSonic are the following:

- Binning services (CSMerge1BinningService and AllelobinningService) provide the automated binning functions
- FrequenciesService provides functions for calculating allele- and genotype frequencies.
- IdentityMatchingService provides search functions for matching unknown genetic fingerprint samples against datasets
- RelatednessService for drawing relatedness tree structures from genetic fingerprints.

All of the analysis services are situated in the Analysis library. The reason for keeping the analysis functions separate from the rest of the system is to further uncouple the application. This way, future developments can easily reuse the services in the analysis library without needing anything from the rest of GenoSonic.

2.7 Web application Layer

This layer consists of the user interface code and accompanying controlling application logic that makes up the GenoSonic dynamic web application.

2.7.1 ASP.net

GenoSonic utilizes Microsoft ASP.net 3.5 as its web application framework. Active Service Pages .net (ASP.net) provides the functionality to build dynamic web applications and web services. ASP.net is also built on the Common Language Runtime (CLR), which means that ASP.net code can be written in any .net language. The GenoSonic website was written in C#.net 3.5. The main components in the web application are pages called “web forms”. Web forms contain both static and dynamic content, and are usually made up of two files:

- a visual portion (the .aspx file) that contains all the static XHTML markup and dynamic server-side asp.net markup
- a logic portion (the .cs file) that contains all the controlling logic for events that occur on user interface elements, like handling button-clicks, page-loads, etc.

By separating the presentation bits from the application logic using this code-behind model, a higher level of decoupling and separation of concerns are achieved. This could allow a user-interface designer, for instance, to focus on the look and feel of the page in the .aspx markup file without being too concerned about the code that drives the UI in the .cs file, which would probably be written by a software engineer. This method provides a separation very similar to the divide between a view and a controller in model-view-controller frameworks (Fowler 2002).

2.7.2 Pages

As described in the previous section, pages are mainly composed of a markup file and an accompanying logic file. To reduce the complexity of each page, as well the amount of code duplication, some more layers of abstraction were implemented:

2.7.2.1 Master Page Template

Every page in GenoSonic implements a template page called a Master Page. The master page defines the basic layout of the pages, i.e. the header section, menu structure, main content and footer sections, together with content placeholders that have to be filled-in by the specific pages. Content pages then only need to specify the content to be placed in the various content placeholders, while the layout and common sections like the logo, menu and footer are rendered according to the master page layout.

2.7.2.2 Abstract base page classes

Every page in GenoSonic extends a base page class which contains functionality common to all pages in GenoSonic. This includes aspects like checking whether a valid user session exists for the current request and setting the current project context so the proper data will be loaded. The base page also carries some common dependencies like references to the project- and group repositories from the data access layer. The base page is also the place where dependency injection is requested from the inversion-of-control container (discussed in a different section). Lastly, the base page extends `System.Web.UI.Page`, which is the ASP.net Web Forms page class.

2.7.2.3 Web User Controls

In certain cases, there are specific recurring groups of UI elements that serve similar purposes across different pages. Instead of duplicating the code on every page, it was wrapped into reusable custom server-side controls called Web User Controls. The most common uses are grid controls and the project-choice control on most pages.

Grid controls were implemented using a third party library called Extjs Extender Controls (Diniz) developed by Rodrigo Diniz which is wrapper control for the Extjs Javascript library. This custom control provides a rich set of custom logic through a server-side grid control. Specifically, it enables interactive grid-editing, paging, sorting, column-reordering, multiple-row selection, and a host of other functions together with asynchronous post-back functions, giving an Excel-like experience inside the website.

2.7.3 Membership, Profiling and Security

GenoSonic uses ASP.net 2.0's built-in membership provider API for all its user security and profiling activities. ASP.net membership provides functionality to store user credentials and information, authenticate users, manage passwords, manage user sessions and even specify custom membership providers if required. As with the WCF services host, the default security provider for the website is set to the built-in ASP.net SqlMembership provider that uses the GenoSonicASPNET database, but this can be swapped with a different provider like Active Directory, LDAP, or Kerberos.

2.7.4 Exception Logging

Errors occur in every software solution. This is nearly unavoidable. It is, however, valuable to log as much information as possible about the exception when it occurs to enable the cause of the problem to be tracked down and rectified. GenoSonic uses an open-source third-party tool for error logging called Elmah (Aziz 2009). Elmah is short for Error Logging Modules and Handlers. It is a pluggable error logging component that can be latched onto any ASP.net web application. It can be used to log nearly all unhandled exceptions and stack trace details, send emails to site administrators, and even view the exception lists in a simple web page. In GenoSonic, Elmah is set up to log errors to a SQLite database file. Errors are then viewable by navigating to the GenoSonicWeb/Elmah subfolder using a browser. Here is an example of the Elmah page:

Error Log for /GenoSonicWeb on STYLE

[RSS FEED](#) | [RSS DIGEST](#) | [DOWNLOAD LOG](#) | [HELP](#) | [ABOUT](#)

Errors 1 to 15 of total 3 077 (page 1 of 206). Start with [10](#), [15](#), [20](#), [25](#), [30](#), [50](#) or [100](#) errors per page.

Host	Code	Type	Error	User	Date	Time
STYLE	404	Http	This is an invalid webresource request. Details...	Kudu	2009/11/29	03:57 AM
STYLE	404	Http	File does not exist. Details...	Kudu	2009/11/06	02:37 PM
STYLE	404	Http	This is an invalid webresource request. Details...	Kudu	2009/11/06	02:37 PM
STYLE	404	Http	File does not exist. Details...	Kudu	2009/11/06	02:36 PM
STYLE	404	Http	This is an invalid webresource request. Details...	Kudu	2009/11/06	02:36 PM
STYLE	404	Http	File does not exist. Details...	Kudu	2009/11/06	02:32 PM
STYLE	404	Http	This is an invalid webresource request. Details...	Kudu	2009/11/06	02:32 PM
STYLE	404	Http	File does not exist. Details...	Kudu	2009/11/06	02:32 PM
STYLE	404	Http	This is an invalid webresource request. Details...	Kudu	2009/11/06	02:32 PM
STYLE	404	Http	This is an invalid webresource request. Details...	Kudu	2009/11/06	02:32 PM
STYLE	500	NullReference	Object reference not set to an instance of an object. Details...	Kudu	2009/11/06	02:31 PM
STYLE	500	NullReference	Object reference not set to an instance of an object. Details...	Kudu	2009/11/06	02:27 PM
STYLE	404	Http	This is an invalid webresource request. Details...	Kudu	2009/11/06	02:27 PM
STYLE	404	Http	File does not exist. Details...	Kudu	2009/11/06	02:27 PM

[Next errors](#)

Powered by [ELMAH](#), version 1.1.11517.2009. Copyright (c) 2004-9, Atif Aziz. All rights reserved. Licensed under [Apache License, Version 2.0](#). Server date is Sunday, 29 November 2009. Server time is 03:57:46. All dates and times displayed are in the South Africa Standard Time zone. This log is provided by the SQLite Error Log.

Figure 2.12: Exception Logging in GenoSonic using ELMAH. *This third-party logging tool saves all exceptions that occur in GenoSonic to a SQLite database file and provides a web interface through which administrators can interrogate these logs. This example shows the basic error-log page provided by ELMAH.*

2.8 External Services Host

This layer can be seen as the gateway via which external applications can communicate with GenoSonic. It exposes most of the common data retrieval and manipulation operations, as well as some of the analysis functions to the external world via secure web services. This transforms GenoSonic from a closed application into an open and extensible services oriented architecture that allows external parties to communicate with GenoSonic programmatically.

2.8.1 Windows Communication Foundation (WCF)

The enabling technology used to create the services layer is called Windows Communication Foundation (WCF). It is part of Microsoft .net Framework 3.0 and contains a set of functionality that enables communication in distributed environments by providing a unified services-oriented programming model that combines Web Services, Message Queues, .Net Remoting and Distributed Transactions. WCF-based services uses SOAP messages for communicating between processes, thus making them interoperable with any other process (not necessarily .net-based) that can communicate via SOAP messages.

2.8.2 Core components

Each WCF service is made up of a set of key components. These include contract definitions, endpoint definitions, bindings, and a hosting environment. The contracts component consists mainly of a service contract and one or more data contracts for each service. The service contract

specifies the available service methods that can be called by a client, while the data contracts specify the structure of the data transfer objects. The external service contracts and the service interfaces from the previous section are, for the most part, exactly the same files. Here is an example of the service contract for the GroupService

```
[OperationContract]
public interface IGroupService
{
    [OperationContract]
    List<Group> GetAllGroups();

    [OperationContract]
    List<Group> GetGroupsByUserName(string userName);

    [OperationContract]
    int InsertGroup(string name, string description);

    [OperationContract]
    int UpdateGroup(int id, string name, string description);

    [OperationContract]
    int DeleteGroup(int id);
}
```

The next key component to every WCF service is the binding definition. It specifies the communication channels between service and client. This includes different protocols like HTTP, TCP and Message Queues (MSMQ). GenoSonic uses an HTTP binding with schema defined security standards. Here is an example of the binding declaration from the service host's web configuration file:

```
<bindings>
  <wsHttpBinding>
    <!-- Set up a binding that uses UserName as the client credential type -->
    <binding name="SqlMembershipBinding">
      <security mode="Message" >
        <message clientCredentialType="UserName" />
      </security>
    </binding>
  </wsHttpBinding>
</bindings>
```

The third key component to every WCF service is its endpoint. An endpoint is a URI that is exposed to the outside world which clients use to connect to the service. Each endpoint consists of an address, a binding and a contract, and is commonly referred to as the ABC of WCF services. The address specifies where the service is hosted, the binding specifies how to communicate with

the service, and the contract specifies what operations can be requested and how to format the data. Here is an example of the endpoint configuration for the GroupService:

```
<endpoint address="GenoSonicServices/Groups"  
  binding="wsHttpBinding"  
  bindingConfiguration="SqlMembershipBinding"  
  contract="GenoSonic.Services.IGroupService" />
```

The last key component is the hosting environment. The hosting environment is the place where the service lives. WCF services are specified to be completely agnostic of their hosting environments. As such, services can be hosted by a range of different applications including Windows applications, console applications, windows services, Microsoft Internet Information Service (IIS), or the Windows Activation Service (WAS). For simplicity and easy development, GenoSonic services were set up to be hosted in IIS together with the website.

2.8.3 Security

GenoSonic was built as a central repository of private genotyping project data. As such, only users with appropriate authorisation should be able to gain access to the data within any of the projects. WCF provides some powerful facilities for implementing security in services. GenoSonic uses WS-Security together with message level encryption provided by an X509 certificate. The X509 certificate is used to encrypt the SOAP message so as to protect the username, password, and other sensitive information which would otherwise be passed as clear text across the wire and could easily be intercepted. The username and password is then checked against the specified security provider. This should be the same provider as the one used by the GenoSonic website to register and authenticate users. In the default implementation GenoSonic uses ASP.NET's built-in SQL membership provider, but this can also easily be changed to a different provider.

Once the external application has been authenticated as a valid GenoSonic user, the requested method is executed. This involves yet another security mechanism in that the user should only be able to retrieve or manipulate data which he has rights to. For data retrieval operations, this is achieved by only returning data from projects that the user currently has access to. For data manipulation operations, a check is done first to see whether the data in the requested operation does in fact belong to a project that the user is allowed to edit before the operation is executed.

2.9 Inversion of Control Container

The previous sections of this chapter elucidated all the layers and components that form the building blocks of GenoSonic. These building blocks depend on one another (in a unidirectional

way) to perform certain tasks. This section explains the mechanism used in GenoSonic to configure and manage the dependencies among the different building blocks in order to produce a flexible working application.

As mentioned many times before, an important focus while developing GenoSonic was to build reusable components with high cohesion and low coupling. This meant developing blocks of code that are as loosely dependent upon each other as possible, yet can be made to work together in a safe and predictable way.

In classical applications, each object is mostly responsible for instantiating and maintaining the lifecycle of its own dependencies. For example, the Samples page will create an instance of the Samples Repository in order to request the relevant sample data. The Samples page thus depends on a Samples Repository object to provide it with sample data. Since the Samples page directly instantiates a Samples Repository, a tight coupling is introduced between the page and the repository. This example is very simple, but as the size and complexity of the application grows, so do the number of dependencies and the level of coupling. Here is some code to illustrate this example:

```
public partial class SamplesPage : BaseUserPage
{
    private List<Sample> LoadSamples(int projectID)
    {
        var sampleRepo = new SampleRepository();

        return sampleRepo.Find(x => x.ProjectID == projectID).ToList();
    }
}
```

Another problem could be that the steps involved in initializing a dependency may require multiple lines of code. If, for instance, a Samples Repository object is used in many different components across the website, all the initialisation steps need to be duplicated in each class that creates an internal instance of the Samples Repository. If the initialisation configuration then changes, for whatever reason, it has to be changed everywhere. For example, this code may be required to initialize a Samples Repository:

```
var sampleRepo = new SampleRepository();
sampleRepo.SetDatabaseConnectionString("someConnectionString");
sampleRepo.InitializePersistenceContext("default");

sampleRepo.AddLoggingEngine(logger);
```

If any of the initialization steps need to change, it would have to be changed everywhere that this class is instantiated.

It is thus for reasons of flexibility and maintainability that GenoSonic uses a Dependency Injection (Fowler 2002) pattern for mitigating the proliferation of dependencies. Dependency Injection, as the name suggests, is a way to inject instances of dependencies into a class, rather than having the class instantiate the objects by itself. Here the control is inverted by having an external party control the instantiation and lifecycle of the dependencies rather than the dependant deciding how its dependencies are created. Dependency Injection is achieved in GenoSonic by constructor injection and setter injection. With constructor injection, dependencies are passed into the object via an overloaded constructor. Setter injection is used where constructor injection is not possible, as in the case of web pages. Here the dependency provider injects dependencies by setting the public properties of the object after construction. Here is an example of the same code snippet from above, modified for setter injection:

```
public partial class SamplesPage : BaseUserPage
{
    public ISampleRepository sampleRepo { get; set;}

    private List<Sample> LoadSamples(int projectID)
    {
        return sampleRepo.Find(x => x.ProjectID == projectID).ToList();
    }
    .
    .
    .
}
```

And here it is modified to use constructor injection:

```
public partial class SamplesPage : BaseUserPage
{
    ISampleRepository sampleRepo;

    public SamplesPage(ISampleRepository repo)
    {
        sampleRepo = repo;
    }

    private List<Sample> LoadSamples(int projectID)
    {
        return sampleRepo.Find(x => x.ProjectID == projectID).ToList();
    }
    .
    .
    .
}
```

In both these cases, an external provider is now responsible for providing the `SamplesPage` with some object that implements the `ISampleRepository` interface. Notice that now the `SamplesPage` class neither cares what specific implementation of the sample repository it uses nor how or when it was created. All it knows is that it is getting an object that implements the rules specified by the `ISampleRepository` interface.

In `GenoSonic`, dependencies are configured, created and maintained by a construct called a `Container`. All dependency creation code is moved out of the classes and into the container configuration, effectively accomplishing the decoupling of dependant and dependency (via the use of interfaces) and enabling the reuse of complex object initialisation steps. `GenoSonic` uses `StructureMap` (Miller 2009) as its dependency injection container, an open-source project created by Jeremy Miller. Other possible dependency injection containers include Microsoft's `Unity` application block, the `Castle` project's `Windsor` and `Ninject`. All the containers can do almost the same thing, each with strengths and weaknesses in certain areas. `StructureMap` was chosen because it was very easy to set up with the use of its fluent interface (see examples below), it works well with generics, it is open-source and it satisfied all the needs of this project. Here is an example of the container setup for the `SampleRepository` alone:

```
ObjectFactory.Initialize(x =>
    {
        // First find all the types that can be created
        x.Scan(scanner =>
            {
                scanner.AssemblyContainingType<ISampleRepository>();
                scanner.WithDefaultConventions();
            });

        x.ForRequestedType<ISampleRepository>().
            TheDefaultIsConcreteType(typeof(SampleRepository))

        // Public Property Setters (Automatically sets the following
        // public properties for any object that uses the BuildUp method )
        x.SetAllProperties(y => y.OfType<ISampleRepository>());
    });
```

Here is a more extensive example of the configuration of the container:

```
ObjectFactory.Initialize(x =>
    {
        // Create find all the types
        x.Scan(scanner =>
            {
                scanner.AssemblyContainingType<ISampleRepository>();
                scanner.WithDefaultConventions();
            });
    });
```

```
x.ForRequestedType<IQuerySurface>().TheDefault.IsThis(querySurface);
    x.ForRequestedType<GenoSonicDB>().TheDefault.IsThis(querySurface);
// for ODS Controllers

x.ForRequestedType(typeof(IGenoSonicRepository<>)).TheDefaultIsConcreteType(typeof(GenoSonicRepository<>));

x.ForRequestedType(typeof(IRepository<>)).TheDefaultIsConcreteType(typeof(GenoSonicRepository<>));

// Public Property Setters (Automatically sets the following
// public properties for any object that uses the BuildUp method )
x.SetAllProperties(y => y.OfType<IQuerySurface>());
x.SetAllProperties(y => y.OfType<GenoSonicDB>());
x.SetAllProperties(y => y.TypeMatches(z => z.Name.Equals(typeof(IRepository<>).Name)));
// generic repo's

x.SetAllProperties(y =>
y.WithAnyTypeFromNamespace("GenoSonic.DataAccess.Repositories")); // specific repo's

});
```

This example looks fairly complicated, but it contains the entire configuration necessary to construct any object from the data access layer needed by the application code. If the user wants to swap a specific implementation in the future, he would only need to change a few lines of code in this container configuration. It would have zero impact on the rest of the code, as there are no concrete references to any of the repositories.

2.10 Conclusion

This chapter discussed the main technical components that make up the software solution. Prominent drivers behind technical decisions made were specifically gauged toward maximising the opportunities for reuse, extensibility, and maintainability, while following well known modern design principles and patterns. It was argued that by creating sound software architecture, the flexibility, lifetime, and opportunities of a software solution can be expanded considerably. This could allow future developers or researchers to easily extend or reuse the current functionality when creating custom solutions for their specific needs.

Chapter 3

Implementation

The software developed as part of the research project has been implemented in the form of a web application called GenoSonic. The goal of this chapter is to explain the functionality that exists in GenoSonic as seen from a user perspective, guided by the objectives set by the problem statement.

3.1 Objective 1:A central repository

-Users should have a secure, easily accessible place where they can manage their genotyping project data as a team

3.1.1 Step 1: Logging into GenoSonic via the website

The very first step for any new user of GenoSonic is to create a new user account by navigating to the registration page and entering a new user name and password. In order to use GenoSonic, users must then log in with the newly created username and password, as shown in the screenshot. Once the user credentials are authenticated with the security provider, the user's profile is loaded into the server session and the user is taken to his personal workspace.

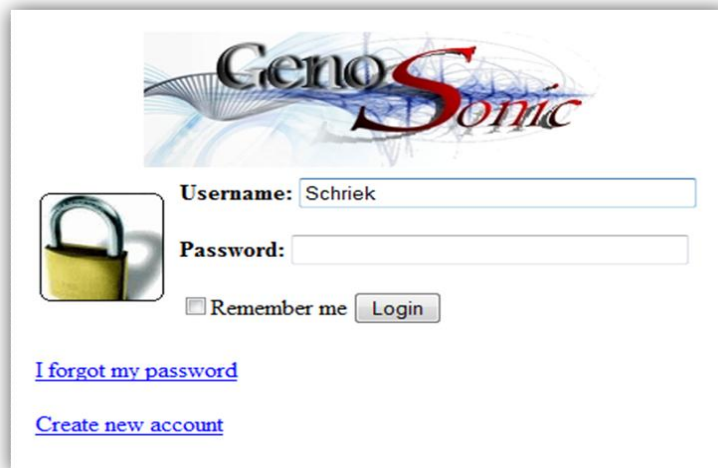


Figure 3.1: Login Page. *Users must enter valid user credentials in order to gain access to GenoSonic. New users must create a new account before continuing to their project workspace.*

3.1.1.1 The personal workspace

The personal workspace is the portal through which users can manage their genotyping projects. Here users can create projects, maintain project data and perform analysis functions. Below is a screenshot of the basic layout of the personal workspace. The sidebar on the left is a tree view that is used to navigate to different places in the workspace. The main sections of the workspace are Projects, Project Data, Binning, Standardized Data, and Analysis Functions. Each section will be discussed in detail further on in the chapter.

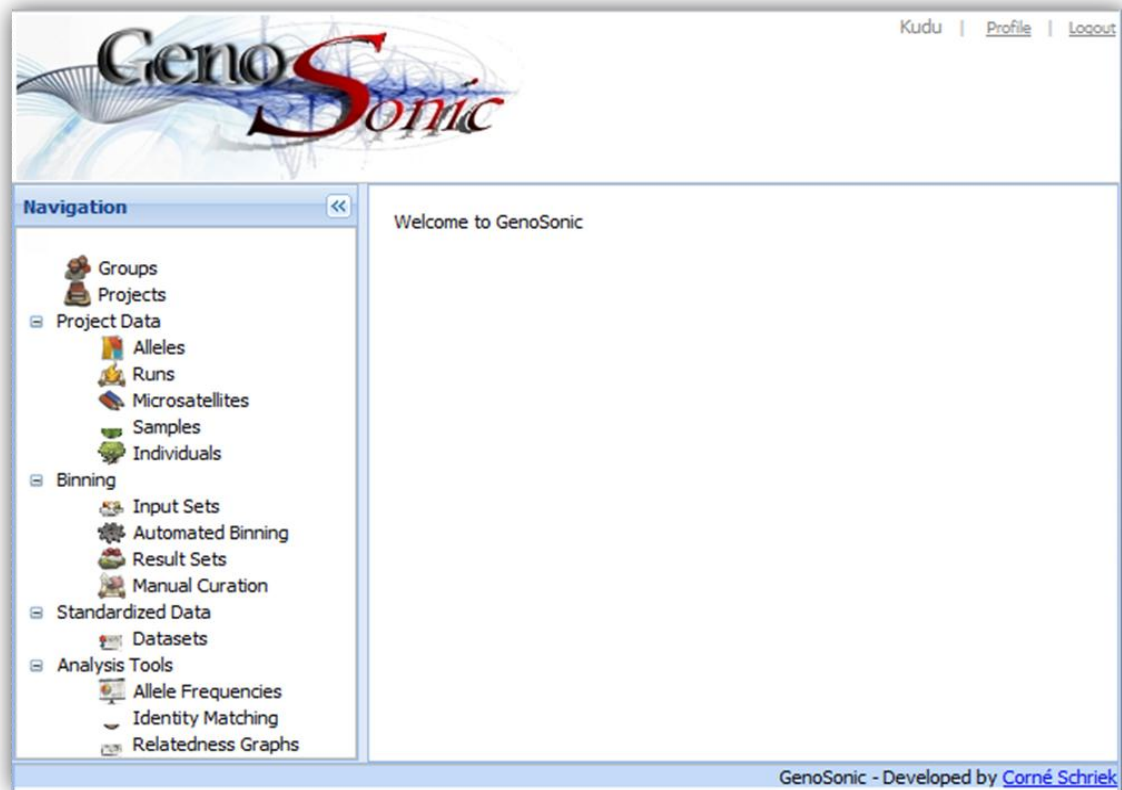


Figure 3.2: Personal workspace. *This is the basic layout of the workspace where users manage and analyse their project data.*

3.1.2 Step 2: Maintaining projects and project groups

In order to add data to GenoSonic, a user must first create a new project and register it under a project group, or alternatively he can request to join other project groups. GenoSonic also allows the creator of a project group to invite other users to join the group. Projects and groups can be manipulated from two main bases, namely the project control which appears at the top of every page, and the Projects and Groups maintenance section, which is the first navigation option on the sidebar.

3.1.2.1.1 The project control:

A custom dropdown control that can be used to set the current project and to create new projects and groups is displayed as the top of the main window of the workspace. This is the control in its collapsed state:

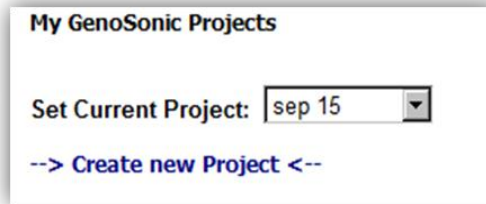


Figure 3.3: The Project Control. *This custom control appears at the top of most pages in GenoSonic. It provides a quick way to set the current project or create a new project.*

Upon clicking on → Create new Project ← , the following dropdown window slides open where the user can set project details:

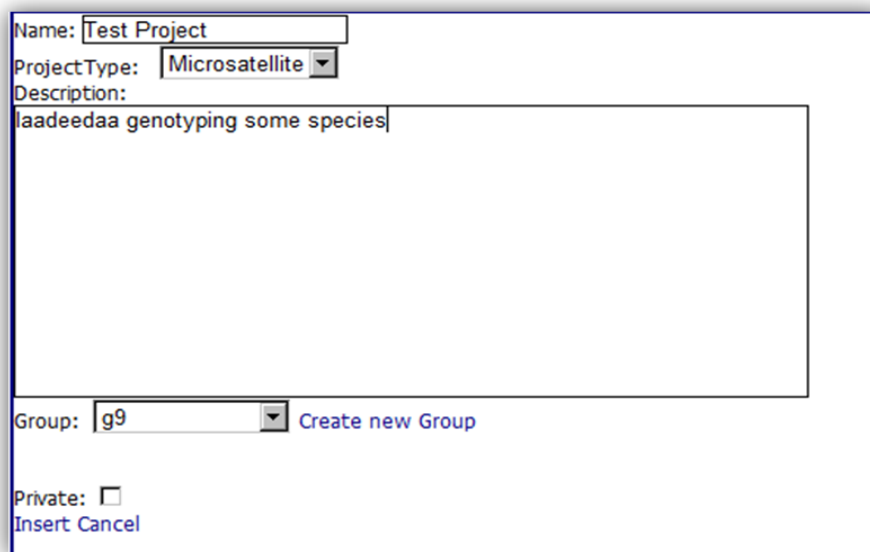


Figure 3.4: The expanded Project Control. *This accordion dropdown expands when the user elects to create a new project. The name, description, type, and study group of the project can be entered to initialize the new project.*

When creating a new project, a group can be selected from the dropdown box. Alternatively a new group can be created by clicking the ‘create new group’ link.

3.1.2.1.2 Projects and groups maintenance pages:

These pages can be used by users to maintain the projects and groups they own or belong to. Group owners can add or remove users to and from their groups. Project owners can also edit or delete their projects here.

3.1.3 Step 3: Managing project data

The basic types of data that can be added to a project are genotypes, samples, individuals, markers and runs. Biological samples taken from certain individuals are tested in experiments called runs to determine their genotypes (genetic makeup) at specific regions in their DNA targeted by microsatellite markers.

The user can manage each of these data types by navigating to the appropriate section through the sidebar. The main section of every data page has a grid listing all the records currently in the system. Functionality built into the grids include sorting, paging, filtering, reordering of rows and columns, customizing visible columns, row selection and even inline editing of cells. Every row in the grids also has a ‘more..’ link, which will take the user to a details view page of the particular record.

All of the data types can be captured in GenoSonic either typing it in manually on input forms, or by uploading it as comma separated value files. For adding new records one at a time, every master page for the various data types has a [New] button that will present the user with an empty input form to complete and save. Below is an example of adding a new Run to the project:

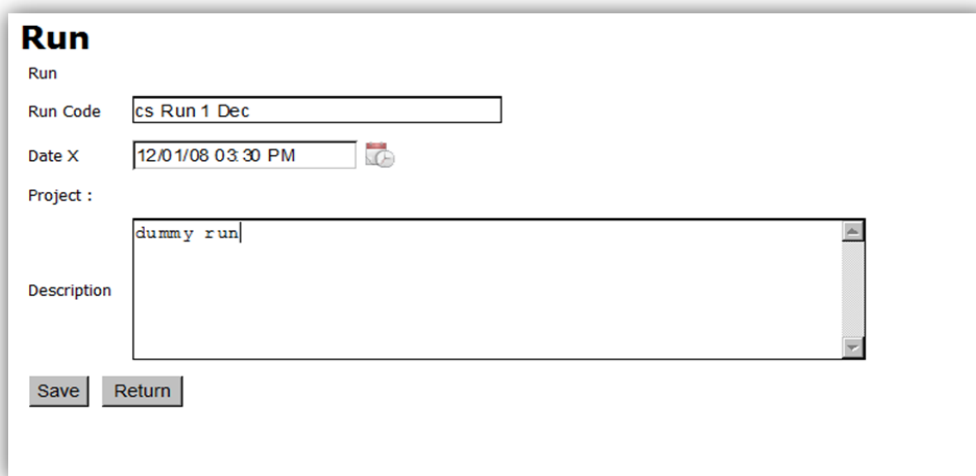


Figure 3.5: Creating a new Run manually. *This screenshot shows the Runs detail page, a typical detail page which can be used to insert or edit records in GenoSonic.*

Existing data can be edited in 3 ways:

- By uploading a new datasheet and setting the upload behaviour to ‘update’
- By clicking the “more..” link next to the relevant record on the data grid. This will take the user to a details page of the selected record that can then be edited
- By using the inline grid editing functionality. This functionality works in a similar to editing an Excel grid and is very handy for making small changes. The graphic below shows how to add IndividualCodes to Sample records.

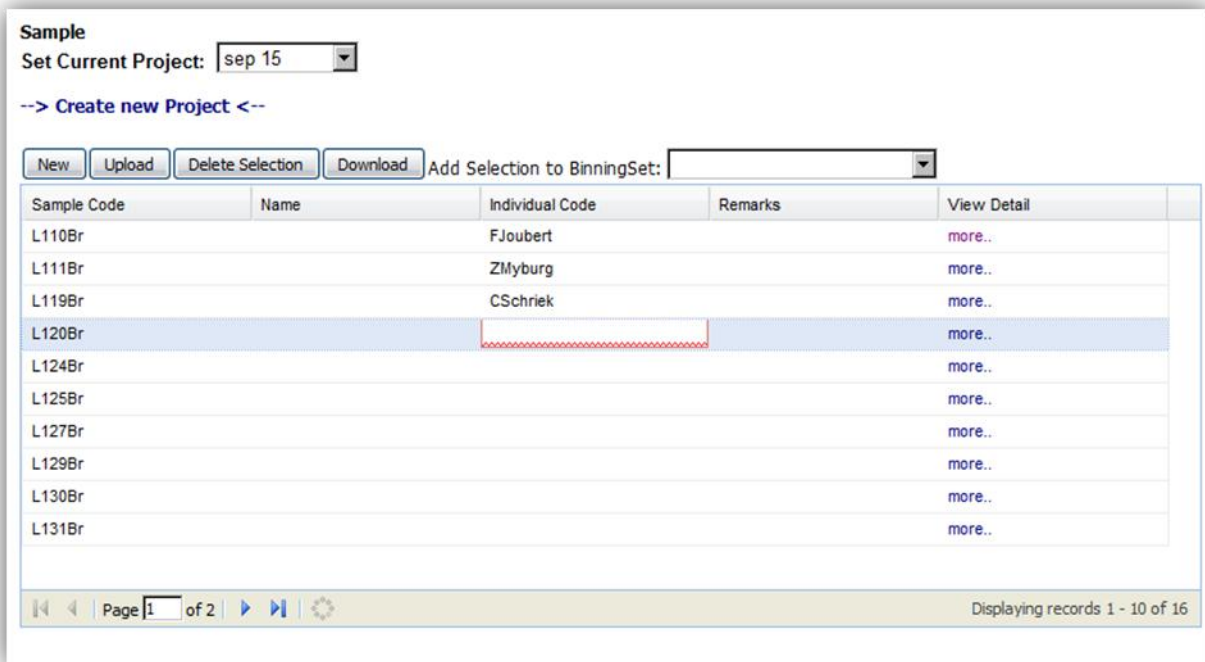


Figure 3.6: Inline grid editing. This functionality works in a similar to editing an Excel grid and is very handy for making small changes to records. To edit a record in the grid view, simply double-click the cell. This will change the cell into an editable textbox.

3.2 Objective 2: Uploading and downloading data

- Users should be able to easily upload and download project data

3.2.1 Uploading data

It would be impractical to feed large amounts of data into the system manually. The best way of importing mass amounts of data via a web interface, is to allow users to upload files containing the data, which can then be parsed and saved to the data store. Since most genotyping data is stored in tabular form and a common output file format from the physical machinery and GeneMapper is a comma-separated-value (csv) format which also works seamlessly with Microsoft Excel, it was the logical choice for GenoSonic to use this file format. The objective with the upload functionality is to make it as flexible, powerful and easy to use as possible.

Upload Data
 Current Project: TestProject ▾

--> Create new Project <--

Choose Data Type: AllelesTable ▾

Choose an Upload Template ▾

Use GeneMapper Style template

Ignore Character:

Upload behaviour:

Insert only - (Only records from the uploaded file that does not yet exist in the project will be inserted. Records that already exist will not be updated / overwritten)

Insert/Update - (All new records from the uploaded file will be inserted into the project, and all existing records will be updated)

Update Only - (Only records from the uploaded file that already exist in the project will be updated. Any records in the file that does not exist in the project will be ignored completely)

Delete all existing records of this type and insert new values - (Clear the project of all records of this type, and then add the records from the file to the project)

Automatically create linked data items if they do not yet exist

Figure 3.7: Uploads page. *Project data can be uploaded to GenoSonic from csv files via this web page. Users can choose the file they want to upload, the template that should be used to read the csv file and the upload behaviour i.e. how to add new records or override existing ones.*

3.2.2 Templates

The first step when uploading data to a project is to retrieve the template file for the specific data type. Depending on the database structure, the default upload template file is automatically generated and downloaded as a .csv file with all the column headers already filled in. Alternatively, users can specify custom templates. To do so, the user can choose a name for the template and then map the column headers of the input file to the property names of the record in GenoSonic. Once the user is satisfied that the selected template will match the column headers of the data in the input file he wants to be inserted, he can browse to the file and upload it. GenoSonic will then parse the uploaded file according to the selected template mapping and the specified upload options.

3.2.3 Options

The user can change the upload behaviour of the system by choosing between several available options.

3.2.3.1 Upload behaviours

The user must choose one of the following behaviours

- **Insert only:** Only entries in the uploaded file that does not exist in the database will be inserted. All existing entries will be ignored.

- **Insert/Update:** New entries will be inserted in the database and existing entries will be updated with whatever is specified in the spreadsheet, except for the fields containing the Ignore Character
- **Update Only:** Entries in the spreadsheet that does not exist in the database will be ignored completely, while existing entries will be updated with whatever is specified in the spreadsheet, except for the fields containing the Ignore Character
- **Delete all existing entries and insert new values:** All the data for this entity type in the current project will be deleted, and everything from the uploaded spreadsheet will be inserted into the database.

3.2.3.2 The 'Ignore' Character

This option is only valid for updates. If the item already exists in the database, the specific field with this value will not be overwritten (as opposed to entering an empty string).

3.2.3.3 Automatically create linked data items if they do not yet exist

This functionality is best described via an example. If, for instance, the user uploads genotype data first and has not yet created records in GenoSonic for the Runs, Markers, or Samples that the genotypes belong to, empty records of these related objects will automatically be created in the Runs, Samples, and Microsatellite tables, allowing the user to enter or upload the details of these entities at a later stage.

3.2.3.4 Use GeneMapper-style template

This functionality is only available when uploading allele data. This enables the uploading of csv files with multiple alleles specified in the same rows. For example, where the normal csv file would be structured like this:

RunCode	SampleCode	MarkerCode	Size	ExternalBin
Run 1	S001	Red 1	50.2	50
Run 1	S001	Red 1	126.2	126
Run 1	S002	Red 1	70.2	70
Run 1	S002	Red 1	90.2	90

Figure 3.8: Example of default csv file structure for allele uploads. *This schema shows alleles as a list of single records. It allows for multiple alleles per genotype, in fact, this schema does not group alleles by genotype at all.*

Output from GeneMapper puts all the alleles for each sample on a single line like this:

RunCode	SampleCode	MarkerCo	Size 1	Size 2
Run 1	S001	Red 1	50.2	126.2
Run 1	S002	Red 1	70.2	90.2

Figure 3.9: Example of GeneMapper csv-file structure. *This is the secondary schema option that can be selected for uploading alleles to GenoSonic. Each row in the table represents a genotype. Its constituent allele sizes are specified as Size1, Size2, etc. depending on the ploidy of the species and particular measurements of the relevant marker.*

When this option is enabled, GenoSonic will automatically figure out how many alleles are specified per sample, and insert the records into the database accordingly.

3.2.4 Downloading data

Any data in the project can be downloaded easily by simply selecting the relevant records from the grid or matrix, and clicking the download button. GenoSonic will then create a new comma-separated-value file and present it to the user as a download option through the browser. The user can then save this file locally. The created file is in the same format as the default upload template for the specific type of data, therefore the user can easily modify or add some records in Microsoft Excel and then upload the file again in order to update GenoSonic with the changes made locally (depending on the chosen upload options)

3.3 Objective 3: Binning and standardization

- *Users should be able to standardize uploaded genotype data by way of both:*
 - a. *An automated binning tool and*
 - b. *A manual standardization step*

After entering uploaded genotype data from experiments into GenoSonic, the user has the option of putting this data through a standardization process before using it in further analysis studies. The need for the standardization steps arises from the fact that there are always small variations in the uploaded allele sizes, due to different environmental conditions, laboratory procedures, or equipment. In order to effectively analyse the data in further studies, the data first has to be standardized.

3.3.1 Step 1: Creating datasets from uploaded data for binning

The first step towards achieving standardization is to add a set of assayed genotypes to datasets called binning-sets. These binning-sets can then be used as input to binning functions such as Allelobin. Binning-sets can be created in the same way that any of the other types of data records can be created by

using input forms, by navigating to the binning-sets page of the project workspace, which falls under the Binning category in the navigation sidebar. Uploaded genotypes can then be added to these binning-sets using one of the grid views to select the set of samples, runs, markers or genotypes and then clicking the “Add to binning-set” dropdown. If the binning-set already exists it can simply be selected and all the genotype data relating to the selected records will instantly be added to the binning-set. Alternatively the user can select the “Create New binning-set” option from the dropdown box to create the binning-set on the fly. A popup window will then appear that can be used to enter the name of the new binning-set.

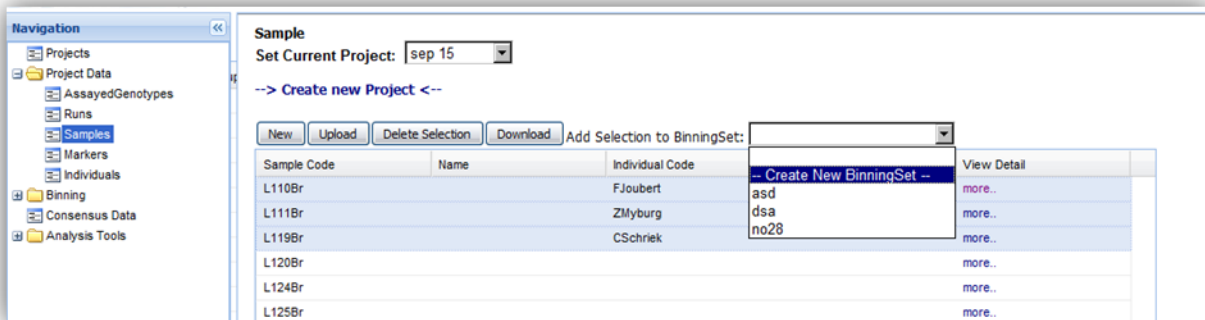


Figure 3.10: Adding samples to a binning set. *Multiple sample records can be selected from the grid. They can then be added to a binning set by selecting the desired dataset from the dropdown list as shown in the figure.*

The image above shows that specific samples can be added to a binning-set by selecting the records from the grid and then selecting the relevant binning-set from the dropdown list. This will add all genotype data connected to the selected samples to the binning-set. Assayed genotypes can be added to binning-sets in the same way by selecting particular individuals, runs, markers, or even single assayed genotypes. This allows the user to quickly and easily craft input binning-sets for further analysis.

The contents of each binning-set can be managed via the details view for binning-sets. Below is a screenshot of the details view of a binning-set. Here the information about each binning-set can be viewed or altered. This includes a Samples-by-Markers matrix view of all the uploaded allele sizes, or optionally the external bins sizes which might have been called by GeneMapper. These matrix views can be downloaded from this page for use in external applications.

BinningSet Matrix

Download the Matrix

Show:

Actual Sizes

External Bins (E.g. GeneMapper)

Delete Selected

Sample	Run	PITX3034...	PITX3034...	PITX3052...	PITX3107...	PITX3107...	PITX3118...	PITX3118...	PITX4056...
AB195	114 - 2009-08-24-01-A	200.4	202.7	226.2	156.8	162.9	210.6	213.9	420.6
AB291	114 - 2009-08-24-01-A	202.7	216	226	151.9		213.4		422.7
AB203	114 - 2009-08-24-01-A	211.6	218	226	156.8	162.8	210.6	214	422.6
AB299	114 - 2009-08-24-01-A	211.6	216	226	156.8		213.7		422.6
AB211	114 - 2009-08-24-01-A	200.3	202.4	226.1	151.9		210.5		422.6
AB307	114 - 2009-08-24-01-A	200.4	213.4	225.9	167.1		210.2	214	422.3
AB219	114 - 2009-08-24-01-A	202.6	216	226	156.8		213.7		422.4
AB315	114 - 2009-08-24-01-A	213.6	218	226	156.8	162.9	214.2		422.5
AB227	114 - 2009-08-24-01-A	213.5	219.8	226	167		210.5	213.8	422.6
AB323	114 - 2009-08-24-01-A	200.3		225.9	156.7	162.8	210.4	213.7	410.7
AB235	114 - 2009-08-24-01-A	213.8	216	226	157.6	162.9	210.5		422.6
AB331	114 - 2009-08-24-01-A	213.6	216	225.7	156.8	162.9	210.3	213.2	410.7
AB243	114 - 2009-08-24-01-A	202.4		225.9	156.8	162.8	210.1	213.4	424.5

Figure 3.11: Binning set details in a matrix view. Each row in the matrix represents a unique sample from a particular run, while each column represents a specific marker. The values are all the allele sizes the make up the binning set.

3.3.2 Step 2: Automated binning

Once the binning-set has been created, the next step is to run a binning analysis function on it. GenoSonic has implemented two binning algorithms. The first is a completely novel algorithm called CSMerge-1 which has been designed specifically to bin data from multiple runs, especially where certain samples are assayed in more than one run. The other binning option is a version of the Allelobin algorithm explained in Idury (1997), which is essentially a curve-fitting algorithm that categorises initial continuous allele sizes into discrete bins by minimising the average standard deviation between bin midpoints and initial allele sizes.

The automated binning function can be executed from the automated binning page. After binning has completed, the binning results will be available in the binning results window. Binning result sets contain the new binned allele sizes (or genotypes) of all the sample-marker combinations present in the input binning-set.

Result Details

Download Results Download Confidence Scores

Sample	PITX3034 (1)	PITX3034 (2)	PITX3052 (1)	PITX3107 (1)	PITX3107 (2)	PITX3118 (1)	PITX3118 (2)	PITX4056 (1)	PITX4056 (2)
AB195	200	203	226	157	163	210	214	421	425
AB291	203	216	226	152		214		423	
AB203	212	218	226	157	163	210	214	423	
AB299	212	216	226	157		214		423	
AB211	200	203	226	152		210		423	
AB307	200	214	226	167		210	214	423	
AB219	203	216	226	157		214		423	
AB315	214	218	226	157	163	214		423	
AB227	214	220	226	167		210	214	423	425
AB323	200		226	157	163	210	214	411	423
AB235	214	216	226	157	163	210		423	
AB331	214	216	226	157	163	210	214	411	425
AB243	203		226	157	163	210	214	425	

Figure 3.12: Matrix view of automated binning result details. *Each row in the matrix represents a sample, while each column represents a specific marker. The values are all the allele sizes the make up the binning result set.*

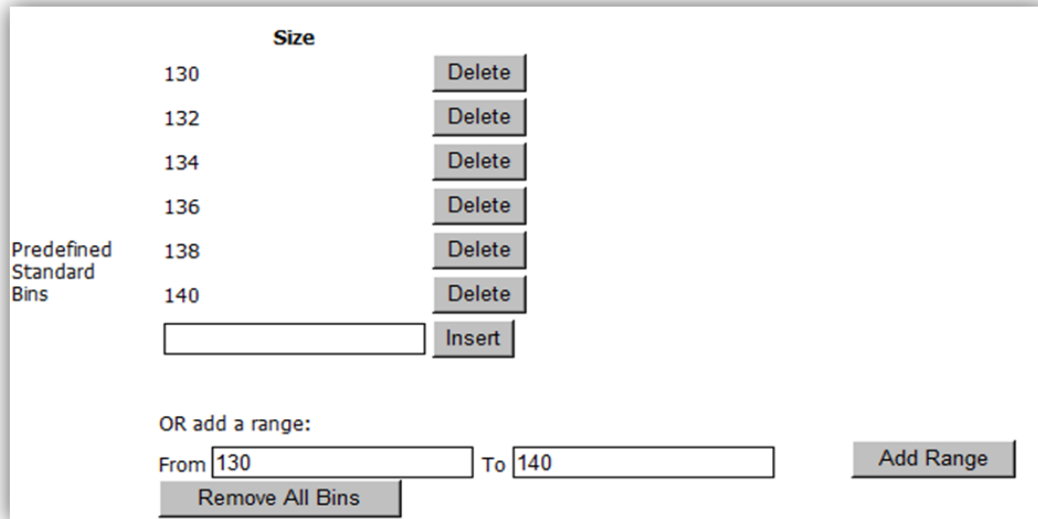
The details window of every binning result has a matrix view of the bins that every allele in the binning-set has been assigned to, as shown in the image above.

3.3.3 Step 3: Standardising binning results

The last step in the standardisation process is to manually examine the binning results in relation to predefined standard bins, external bins that might have been assigned by tools like GeneMapper or GeneMarker, quality scores, and the initial uploaded size scores. The user can then decide whether or not to accept the allele size proposed by the binning algorithm.

3.3.3.1 a. Specifying standard bin sizes

Before creating a standardized dataset, a set of standard bins should first be assigned to every marker used in the binning set. Standard bins can be seen as recognised possible allele sizes that could occur for a given microsatellite region.



Predefined Standard Bins		Size	Action
	130		Delete
	132		Delete
	134		Delete
	136		Delete
	138		Delete
	140		Delete
	<input type="text"/>		Insert

OR add a range:

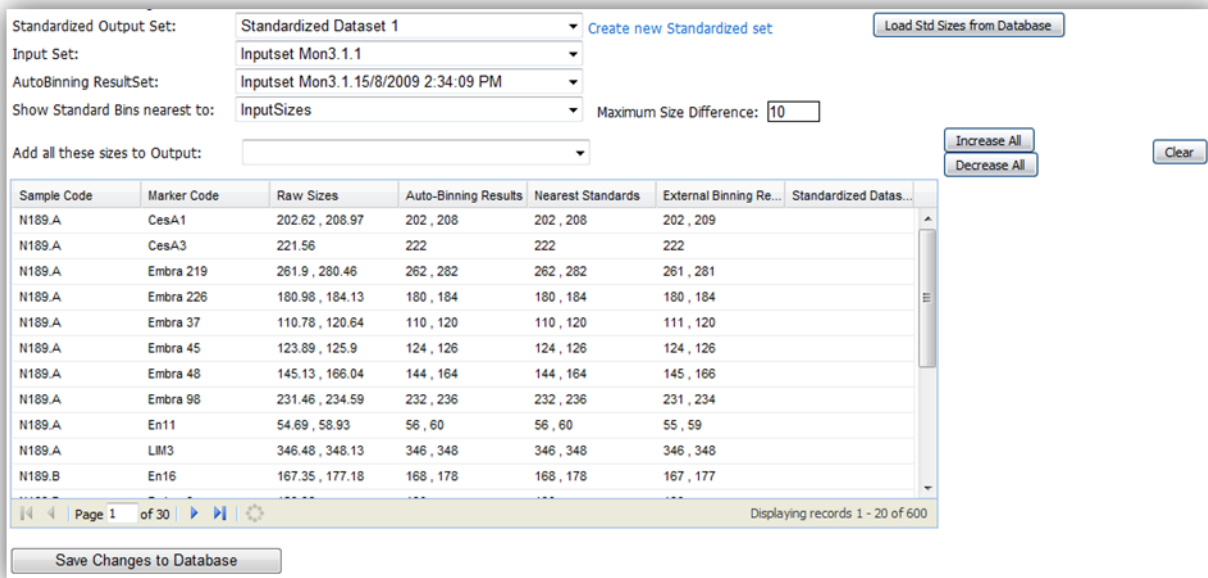
From To

Figure 3.13: Adding standard sizes to markers. *Standard sizes can be specified for each marker by navigating to the marker's detail page and entering the sizes either one by one or by specifying a size range as shown above.*

In the details view of the microsatellites page, the user has the option of adding standard bins. Standard bins can either be added one-by-one, or by entering a size range.

3.3.3.2 b. The standardization tool

Standardized datasets are defined as collections of genotypes that have been put through a binning analysis process and are seen as fit to be used for further analysis like identity matching, paternity testing, or association studies. In order to produce these standardized datasets, the manual standardization tool must be utilized. Essentially this is a webpage that is used to compare uploaded allele sizes measured by upstream genotyping software like GeneMapper or GeneMarker with the size calls predicted by automated binning software and the known standard bin sizes. The user can then choose which of the sizes to accept and create a new standardized dataset, or the user can modify an existing one.



Sample Code	Marker Code	Raw Sizes	Auto-Binning Results	Nearest Standards	External Binning Re...	Standardized Datas...
N189.A	CesA1	202.62, 208.97	202, 208	202, 208	202, 209	
N189.A	CesA3	221.56	222	222	222	
N189.A	Embra 219	261.9, 280.46	262, 282	262, 282	261, 281	
N189.A	Embra 226	180.98, 184.13	180, 184	180, 184	180, 184	
N189.A	Embra 37	110.78, 120.64	110, 120	110, 120	111, 120	
N189.A	Embra 45	123.89, 125.9	124, 126	124, 126	124, 126	
N189.A	Embra 48	145.13, 166.04	144, 164	144, 164	145, 166	
N189.A	Embra 98	231.46, 234.59	232, 236	232, 236	231, 234	
N189.A	En11	54.69, 58.93	56, 60	56, 60	55, 59	
N189.A	LIM3	346.48, 348.13	346, 348	346, 348	346, 348	
N189.B	En16	167.35, 177.18	168, 178	168, 178	167, 177	

Figure 3.14: Genotype Standardization Tool. This page is used to create standardized datasets for further analyses. Users choose the input set and binning result set they wish to use, and compare the sizes to the nearest standard marker sizes (if specified) and external size calls (if specified). Final sizes calls are entered into the last column of the grid, either manually or with the help of the automated options provided. Once the user is satisfied with the final size calls, the standardized set is saved to the database.

The screenshot above shows the layout of the genotype standardization tool. First the user has to choose or create the target standardized dataset. Next the input binning set is selected and optionally the results from an automated binning function executed on the chosen binning set can be selected. The grid view will then display all the genotypes in the binning-set with their uploaded sizes, automatic binning result sizes, nearest standard bins defined for the relevant microsatellite, external bin sizes (e.g. from GeneMapper or GeneMarker) and the current standardized allele sizes for the genotype. The user can then choose to move any of these columns to the standardized datasets, or manually type changes in the grid in the same way it can be done in Microsoft Excel.

Once the user is satisfied with the values in the standardized dataset, the save button is clicked. This will pop up a window with options for defining the persistence behaviour.

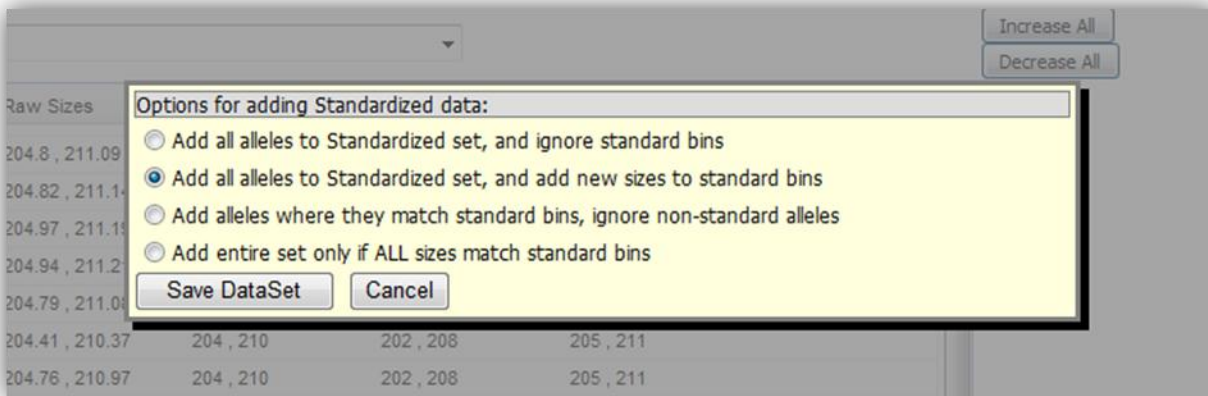


Figure 3.15: Genotype Standardization Tool - Options when saving a standardized dataset. *Before a standardized dataset is saved, the user must specify the behaviour of the save function with regard to standard bins. The default option is to add all novel allele sizes for a given marker to its set of standard bin sizes. However, a user may also want to enforce stricter rules like only adding allele sizes to a standardized set if those sizes match the predefined standard bin sizes exactly.*

The following options are available:

- Add all alleles to the Standardized set and ignore standard bins.
- Add all alleles to the Standardized set and add new sizes to standard bins: Any allele size specified within the standardized dataset that is not yet reflected in the list of standard bin sizes for that microsatellite marker will be added to the marker’s standard bin sizes.
- Add alleles where they match standard bins and ignore non-standard alleles: An allele will only be added to the standardized dataset if its particular size is also specified in the standard bins list for that marker.
- Add the entire set only if ALL sizes match standard bins: If even one sample has one allele that is not reflected as a standard bin size, nothing will be added to the selected standardized dataset.

3.3.3.3 c. Standardised sets

The output from the standardization tool is a standardized dataset. This standardized dataset can be used in further analyses like determining allele- and genotype frequencies, relatedness studies, identity matching, etc. Standardized datasets can be managed in the standardized data section. In addition there is a special section where all the data in the dataset can be viewed either as a list of genotypes, or as a samples-by-markers matrix with allele sizes filling in the contents of the grid. Both these views are also downloadable as .csv files. Below is a screenshot of this functionality:

Download Dataset

Standardized Dataset 1

List
 Matrix

SampleCode	CesA1 (1)	CesA1 (2)	CesA3 (1)	CesA3 (2)	Embra 219 (1)	Embra 219 (2)	Embra 226 (1)	Embra 226 (2)	Embra 37 (1)
N189.A	202	208	222		262	282	180	184	110
N46.A	202	208	222		262	282	180	184	110
N331.A	204	216	222	226	262	276	172	178	120
N332.A	204	216	222	226	260	276	172	178	126
N333.A	204	216	222	226	260	276	172	178	126
N334.A	204	216	222	226	260	276	172	178	126
N335.A	204	216	222	226	262	276	172	178	126
N336.A	204	216	222	226	262	276	174	180	
N337.A	204	216	222	226	262	276	172	178	126
N339.A	204	210	222	226	268	276	170	178	124
N340.A	204	210	222	226	268	276	170	178	124
N341.A	204	210	222	226	268	276	170	178	124

Figure 3.16: Matrix view of standardized dataset detail. Each row in the matrix represents a sample, while each column represents a specific marker. The values are all the allele sizes the make up the standardized dataset.

3.4 Objective 4: Derive allele frequencies

- Users should be able to view allele frequencies for any given dataset and marker.

GenoSonic has a simple page where the genotype and allele frequencies of a given standardized dataset can be viewed for any given marker. Frequencies are calculated dynamically and can be viewed as shown here:

Marker: PtTX 2146

Allele Frequencies:

Allele	Count	Frequency %
184	2	50
192	1	25
202	1	25

Genotype Frequencies:

View Detail

AssayedGenotype	Count	Frequency %
184/ null	1	33.33333333333333
192/ null	1	33.33333333333333
202/184	1	33.33333333333333

Figure 3.17: Allele- and Genotype Frequencies page. GenoSonic generates allele and genotype frequency grids for any marker chosen from a dropdown list of constituent marker codes of a given standardized dataset. The grids show all the different allele or genotype sizes, the number of times they occur in the dataset and their proportional frequency percentage.


3.5 Objective 5: Querying the datasets

- *Users must be able to query standardized datasets effectively. For the purpose of this study, only identity matching queries need to be implemented.*

3.5.1 Identity matching

Users can enter genetic fingerprint data from unknown samples or use known samples to match against standardized dataset data. Unknown sample data can either be typed in manually or be uploaded from a .csv file via the provided upload functionality, which works in exactly the same way as the general uploads described in an earlier section. Single known samples can also be chosen, or entire datasets can be chosen to do a bulk comparison.

The input samples are then screened against the reference standardized dataset and samples that match the input samples with either a user defined minimum probability of identity, or a minimum percentage of matches, are then shown in the resulting grid (or list of grids if more than one test sample were given as input).



Identity Matching

Project :

Reference Dataset :

Set test alleles:

Manually type in allele sizes
 Select from this dataset
 Upload from file
 Match to another dataset

Enter test sample details:

Marker Code	Allele 1	Allele 2
PtTX3034	<input type="text"/>	<input type="text"/>
PtTX3052	<input type="text"/>	<input type="text"/>
PtTX3107	<input type="text"/>	<input type="text"/>
PtTX3118	<input type="text"/>	<input type="text"/>
PtTX4056	<input type="text"/>	<input type="text"/>
PtTX4093	<input type="text"/>	<input type="text"/>
PtTX4114	<input type="text"/>	<input type="text"/>
PtTX4228	<input type="text"/>	<input type="text"/>
ript1072	<input type="text"/>	<input type="text"/>
sifg0166	<input type="text"/>	<input type="text"/>
sifg0167	<input type="text"/>	<input type="text"/>
sifg0737	<input type="text"/>	<input type="text"/>

Minimum Probability of Identity

Minimum Number of matching alleles

Figure 3.18: Identity Matching input options. A standardized dataset is chosen as the reference set. Test samples can be entered manually, selected from the reference dataset, uploaded from a file, or specified as another standardized dataset. Users can also specify the minimum number of matching alleles and minimum probability of identity that two samples must have in order to appear in the result set.

The probability of an individual having a certain allele is the number of times that allele occurs in the standardized set divided by the total number of alleles. All microsatellite locations in a test set are assumed to be independent. In other words the presence or absence of an allele for one microsatellite does not influence the probability of a certain allele occurring at a different microsatellite location being tested. The probability of an individual carrying the entire genetic fingerprint is then defined as the product of all the individual probabilities.

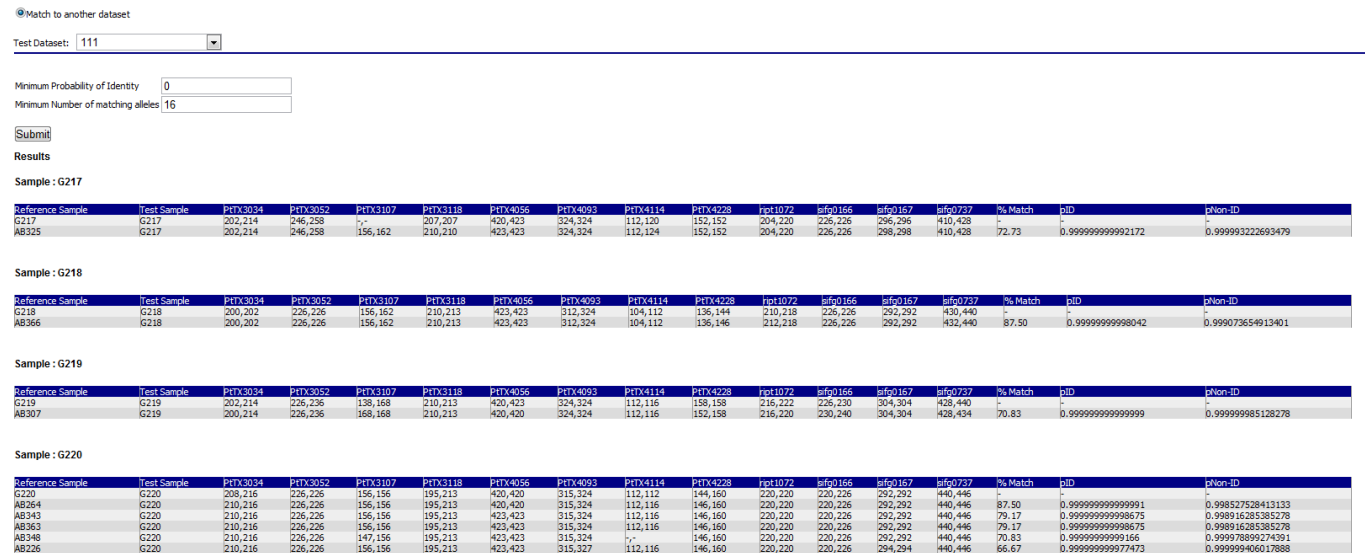


Figure 3.19: Identity Matching results. This screenshot shows the result of matching multiple samples from one dataset against another dataset. Each sample from the test set has, in this case, multiple probable matches in the reference set.

3.6 Objective 6: Visualize measures of relatedness

- Users need a way to visualize the relatedness of samples in a dataset. For this study, a simple relatedness dendrogram visualisation has to be developed

As a means of visualizing the relatedness of samples in a dataset, GenoSonic can create a pair-wise related tree structure using a neighbour-joining algorithm. It essentially builds a tree structure by assessing genetic distances among all samples and then pairing the closest ones iteratively until a root node is created.

The tree structure is then presented to the user in the Newick tree file format. It is a standard notation which can be used to represent tree structures in plain text. This text format can be parsed by almost every phylogenetic tree drawing software package. Here is an example of the Newick format:

```
((((AB320:0.266666666666667,G249:0.266666666666667):0.840524483147591,G234:0.840524483147591):0.476601362869988,(AB235:0.0132275132275132,AB384:0.0132275132275132):0.667507848176333,G242:0.667507848176333):0.476601362869988):0.454117176945001,((AB363:0,AB264:0):0.036550410711194,G220:0.036550410711194):0.297181545057036,AB330:0.297181545057036):0.454117176945001);
```


Here are some screenshots of how the text structure above can be visualised using a freely available phylogenetic tree drawing software package called PhyloWidget:

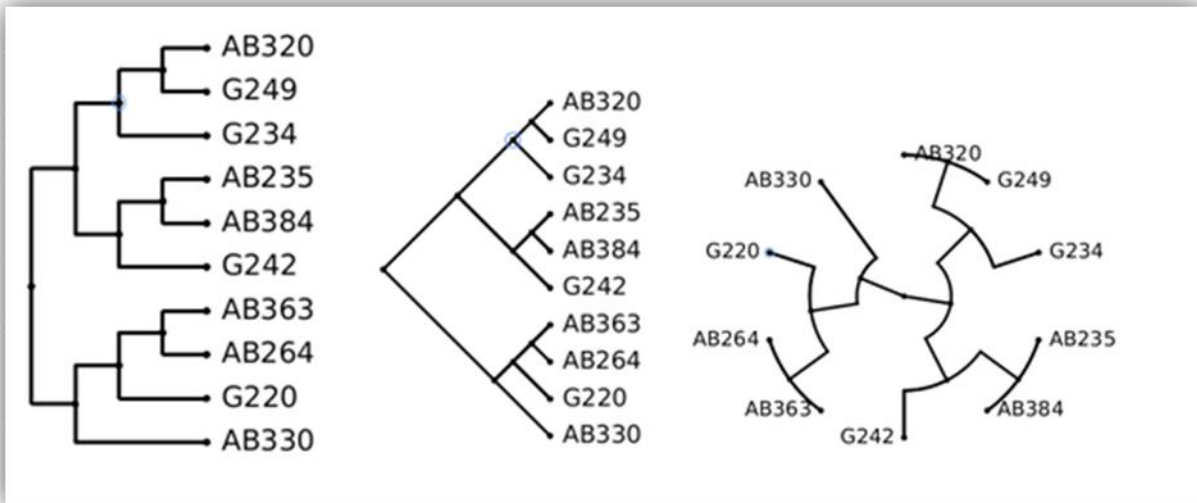


Figure 3.20: Example visualizations of GenoSonic tree structure output using PhyloWidget. *PhyloWidget* is given a tree structure as a standard input string in the Newick tree file format. It can then visualize this structure using a number of drawing algorithms, including the three shown here. Each leaf node represents a sample. The tree structure shows a measure of relatedness among samples

3.7 Conclusion

This chapter summarizes the functional features included in GenoSonic aimed at addressing the identified user requirements. It was shown how GenoSonic provides a secure, easily accessible space where users can manage their genotyping project data as a team. Users can easily upload and download large amounts of data to and from the system. Once uploaded to the system, genetic fingerprint data need to be standardised before it can be used in further analyses. To do this, GenoSonic has implemented a two-step approach. The first step is to assign discrete alleles sizes, or bins, to all the initial allele sizes of the fingerprints automatically by using an automated binning algorithm. The second step is to manually verify the results from the automated binning function and adding it to a standardized dataset. Once the genetic fingerprints have been standardized, allele- and genotyping frequencies can be viewed for any given marker. GenoSonic also provides functionalities for identity matching. One or many genetic fingerprints from unknown samples can be matched against a standardized dataset in order to establish identities and in some cases infer relatedness. Finally, GenoSonic implements a tree construction function which can be used to infer measures of relatedness among samples in a dataset.

Chapter 4

The analysis of clonal identities of *Pinus patula* ramets from clonal seed orchards

4.1 Introduction

This chapter addresses the application of the developed software solution to the analysis of clonal identities of *Pinus patula* ramets from seed orchards. The project from which data was obtained forms part of a larger collaborative molecular genetics research effort between the University of Pretoria and certain key commercial organisations in the forestry industry of South Africa. This particular project is a genetic fingerprinting assignment aimed specifically at the re-establishment or confirmation of clonal identities of *Pinus patula* ramets from privately owned seed orchards at a new agricultural estate in relation to ramets at the original clonal seed orchard located in another region of the country. Due to the sensitive nature of this data, the name of the forestry company had to be kept undisclosed together with the actual locations, molecular markers, and other sensitive information. The original clone bank will hence be referred to as Site G and the new orchard will be called Site AB.

Pinus patula is a fast-growing species of pine trees, which serves as an important source of many wood-based products. It is used for various types of light construction, furniture, flooring, food containers, pallets, poles and fence posts, among other things. *Pinus patula* is also an important source of pulpwood from which paper is produced in South Africa. Lastly, it is a popular source of fuel for burning directly or via the production of charcoal.

As the demand for these products continues to rise, so too does the importance of continually researching ways to improve suitability of the trees for the various products. Homogeneity in size, shape, and quality of the wood are much sought-after traits to buyers as they allow the creation of various final products with minimal adjustments needed to equipment or processes. To growers, exact knowledge of the expected performance of their plantations in certain environmental conditions with regard to temperature, rainfall, elevation, etc. is also very important. The best way of realising such homogeneity and predictability is by breeding the best trees with all the desired traits and then creating cloned replicates or collecting seeds from a few specific superior individuals.

The forestry company in question owns multiple *Pinus patula* orchards in South Africa. Many of these orchards are made up of grafted clones and seed offspring from an elite set of 43 genetically unique pine trees. The orchard at Site G houses these trees which form the original clonal seed bank. These trees serve as the source of the genetic material (seed, pollen, or grafts) from which most other clonal orchards will be created. The orchard at Site G is used mainly in breeding

programs for creating new individuals or hybrids by way of controlled pollinations with related pine species.

Two additional clonal seed orchards have recently been grafted using scions from the original orchard. One orchard has been created at Site AB and the second in yet another different geographic location. The two new orchards are to be used for commercial seed collections and large scale creation of grafted clones.

The three orchards are intended to be copies of each other in that they should contain the same genetic material. There were numerous reasons for creating copies of the original orchard. One important reason was to be able to assess the performance of clones in different environmental conditions in the different locations. Another reason was to reduce the risk of losing their clone bank due to sickness, fire, or other natural disasters. It is important to be sure of the exact clonal identities of each of the trees in all three locations, which cannot be determined by phenotypic traits alone.

This study was aimed only at establishing or re-establishing the clonal identities of a sample of the young trees at Site AB with their clonal parents at the original orchard at Site G. The study consists of two sets of fingerprinting data. The first is a non-redundant reference set taken from the original *Pinus patula* orchard at Site G. The second set of samples comes from the trees that have been planted at Site AB. The objective of the study is to verify that all the new trees have indeed been labelled correctly as being clones of their respective originating ramets and, if not, to determine the correct clonal or unique identities.

4.2 Materials and Methods

4.2.1 Fingerprint data gathering and manual analyses

DNA was extracted from two sets of samples. The first is a non-redundant reference set from 43 unique individuals from Site G. The second set consists of 191 samples taken from the clones at Site AB. All DNA was extracted using the DNeasy extraction kit from Qiagen. Fingerprinting assays were then performed separately on the two sets during the months of August 2009 to November 2009, using a marker panel of twelve microsatellite markers, the details of which are shown in the table below:

Table 4.1: Microsatellite markers contained in Pine Panel A

Marker	Dye	Size range
Marker3	Fam (Blue)	145-170
Marker9		205-240
Marker6		302-345
Marker1	Vic (Green)	190-230
Marker11		290-325
Marker5		388-440
Marker8	Ned (Yellow)	145-168
Marker2		205-265
Marker12		410-455
Marker7	Pet (Red)	95-125
Marker4		195-220
Marker10		221-240

The reference set was fingerprinted over a series of six assays. Each subsequent assay was run only on samples for which scoring of alleles for certain markers failed on previous assays. This process was repeated six times until a set of 43 reference fingerprints had been obtained.

Similar to the reference set, the 191 samples from the Site AB clones were assayed repeatedly with the same marker panel until a complete set of fingerprints emerged. Only three samples had to be re-examined in a second assay due to scoring issues with these samples in the first assay. All of the other alleles were scored in a single experiment.

The raw scores from the fingerprinting assay were then analysed and binned manually by human experts using SoftGenetics GeneMarker 1.80. After manual binning, the Site AB clone fingerprints were compared manually to the reference set in order to verify or re-determine clonal identities.

4.2.2 GenoSonic project Setup

The output .csv files from GeneMarker after the checking and editing of size standards were then uploaded to GenoSonic to be analysed and results were compared with those produced manually by human experts. To achieve this, GenoSonic was installed on a central web server in the Bioinformatics Unit at the University of Pretoria, from where it was accessible to all stakeholders. Stakeholders (scientists, developers, assessors and clients) then registered as GenoSonic users, as described in Chapter 3. A new user group called the “GenoSonic Development Team” was created in GenoSonic and all the relevant users added to this user group. A new project was created and everyone belonging to the user group could access this new project. This step fulfilled the first user requirement set by the problem statement.

4.2.3 Import fingerprint data into GenoSonic

The next step, after projects and groups had been set up, was to feed data into GenoSonic. This was done by utilising GenoSonic’s Upload functionality with the help of custom upload templates. First,

an upload template was set up for alleles to map the columns in the .csv upload file to the fields in GenoSonic as follows:

Edit Upload Template :	
Property	Sheet Column Name
AlleleCode	
RunCode	Run
SampleCode	Sample
MarkerCode	Marker
Size	Size
ExternalBin	ExternalBin
Dye	Dye
Peak	Height
Area	Area
HeightRatio	Ht_Ratio
AreaRatio	Ar_Ratio
Start	Start
End	End
Difference	Difference
QualityMeasure	Score
QualityStatus	Quality
QualityComments	Quality Reasons
Comments	Comments

Figure 4.1: Template mapping for allele upload. *The first column shows the standard names of the properties in GenoSonic. The user can set the property names in the last column to match the column headers of the csv files to be uploaded.*

Once the allele upload template was ready to be used, a set of .csv files containing all of the output allele sizes from GeneMarker, after checking and editing of size standards, were uploaded to GenoSonic. During the upload process, GenoSonic automatically created the basic Sample-, Marker- and Run-records linked to each allele. The upload process created a total of 5471 allele records linked to 12 microsatellite marker records, 8 runs (6 from Site G and 2 from Site AB) and 234 samples, which matched the expected input exactly. The total upload time was less than three minutes, using a 10Mbit LAN connection to a Pentium 4 web server.

Following the allele upload, the details of each of the 12 microsatellite markers also had to be uploaded. Details like the repeat-unit size for each marker must be known to GenoSonic in order to perform the binning and relatedness analysis functions. This data could also have been entered manually using the marker detail views or the inline grid editing functionality provided by the web application. Further sample details could also be uploaded relating to the samples' origin etc., but was not needed for the purpose of this study.

Users who wanted local copies of the data could now easily download it by clicking the [Download] button present on each of the assorted project data pages. GenoSonic would then retrieve the data from the database and format it into comma separated value (.csv) files which the user could save to the local disk drive. This upload and download functionality allows users to easily manage and share project data, which satisfies the second requirement defined by the problem statement.

4.2.4 Binning and standardization

Genetic fingerprint data must be standardised before it can be used in analyses. To do this, GenoSonic has implemented a two-step approach. The first step is to automatically assign all of the uploaded continuous allele sizes, as determined by GeneMarker, to discrete bins using an automated binning function. The second step is to manually verify the results from the automated binning function and add it to a standardized dataset.

In this study, all of the uploaded alleles from both the reference set and the clone set were added to a single binning input set. All fingerprints were analysed simultaneously because the binning function (CSMerge-1) was specifically designed to progressively align all fingerprints from different runs that share samples. After alignment, the alleles were grouped into bins using a dynamic clustering technique called quality threshold clustering. The new binned size of each allele was then named after the natural number closest to the mean size of all the member alleles of each bin. The automated binning function was executed using the following parameters (See chapter 3 for definitions of each parameter):

- Maximum allowed shift during alignment : 1.5 base pairs
- Quality Threshold (maximum Euclidean distance) during clustering: 1.5 base pairs
- Species ploidy: 2

The first step of the CSMerge-1 binning function aligned the alleles of each run with the other runs by shifting them as shown in the table below. A summary of the numbers of alleles and runs with which any given run shares sample and markers is shown in Table 4.2 below. The value by which each allele size every run is increased or decreased is shown (“Shift”), together with the number of connected alleles and runs. See Chapter 2 for a complete definition of connected alleles.

Table 4.2: Output after CSMerge-1's progressive alignment step

Run	Shift	#Connected Runs	#Connected Alleles	#alleles scored
Site AB - 2009-08-24-01-A	0	1	17	3576
Site AB - 2009-08-26-01-A	-0.2	1	19	57
Site G - 2009-09-17-01-A	0	4	265	582
Site G - 2009-09-25-01-A	-0.1	4	315	336
Site G - 2009-09-29-01-A	0.2	4	307	307
Site G - 2009-10-02-01-A	0.1	4	198	198
Site G - 2009-11-04-02-A	-0.2	4	280	301

The next automated step that CSMerge-1 performed was to cluster all the alleles. Given a maximum diameter of 1.5 base pairs for each cluster, the algorithm clustered alleles into bins as summarized in the following table:

Table 4.3: Output after CSMerge-1's QT Clustering step

Marker	#Bins	Mean of Smallest Bin	Mean of Largest Bin	Average Std. Deviation	#Allele calls
Marker1	11	197.5	237.45	0.146	514
Marker2	9	210.5	259.62	0.187	485
Marker3	11	138.74	185.4	0.07	311
Marker4	11	195.06	219.6	0.134	434
Marker5	4	410.71	424.61	0.107	378
Marker6	12	303.28	341	0.114	484
Marker7	19	95.5	125.6	0.164	504
Marker8	15	137.45	182.4	0.1	517
Marker9	13	205.5	240.44	0.139	452
Marker10	12	220.69	239.26	0.099	465
Marker11	17	291.63	332.3	0.134	396
Marker12	14	410.51	462.9	0.128	531

The last step performed by the CSMerge-1 automated binning function was to include only the two best allele sizes (since the species ploidy is two) for each sample-marker combination in the output set. This was done by first selecting the binned allele sizes that occurred most frequently for a given sample-marker combination and then by allele quality (Euclidian distance from the bin mean) if two different sizes occurred the same number of times.

The final step in the standardization process comprises a manual evaluation of automated results. Here GenoSonic's manual standardization page was used to inspect results and make corrections as needed. The details and reasoning of this step are discussed in the results section.

4.2.5 Derive allele frequencies

A frequencies page was created in GenoSonic where the user can view allele- and genotype frequencies for any standardized dataset. Frequencies for the selected dataset are calculated on the

fly. The frequency tables for the Marker1 marker from the case study's reference sample set (Site G) are shown below as an example of the output GenoSonic can create. The allele frequencies are used by GenoSonic's identity matching algorithms to produce probabilities of identity between test fingerprints and reference datasets. In the case of this study, probabilities of identity that had been calculated from the Site G reference set's allele frequencies were used to determine the identities of the Site AB clones against the Site G reference set. The tables below are examples of allele- and genotype frequencies calculated for Marker1:

Table 4.4: Allele frequencies for marker Marker1 in *Pinus patula* case study

Allele	Count	Frequency %
200	42 / 464	9.05
203	174 / 464	37.5
205	4 / 464	0.86
209	6 / 464	1.29
212	30 / 464	6.47
214	119 / 464	25.65
216	62 / 464	13.36
218	17 / 464	3.66
220	8 / 464	1.72
237	2 / 464	0.43

Table 4.5: Genotype frequencies for marker Marker1 in *Pinus patula* case study

Genotype	Count	Frequency %
200 / 200	3 / 232	1.29
200 / 203	27 / 232	11.64
200 / 205	2 / 232	0.86
200 / 214	6 / 232	2.59
200 / 216	1 / 232	0.43
203 / 203	32 / 232	13.79
203 / 212	7 / 232	3.02
203 / 214	53 / 232	22.84
203 / 216	21 / 232	9.05
203 / 237	2 / 232	0.86
205 / 214	2 / 232	0.86
209 / 216	6 / 232	2.59
212 / 212	6 / 232	2.59
212 / 214	1 / 232	0.43
212 / 216	3 / 232	1.29
212 / 218	7 / 232	3.02
214 / 214	9 / 232	3.88
214 / 216	26 / 232	11.21
214 / 218	10 / 232	4.31
214 / 220	3 / 232	1.29
216 / 216	2 / 232	0.86
216 / 220	1 / 232	0.43
220 / 220	2 / 232	0.86

4.2.6 Identity matching

GenoSonic's identity matching functionality was utilized in the case study to determine clonal identities. All of the DNA fingerprints in the clone set from Site AB were compared to all of the DNA fingerprints in the reference set from Site G in order to determine or re-establish clonal identities. This was accomplished by using GenoSonic's bulk identification feature wherein the entire clonal set was compared to the entire reference set in a single step. GenoSonic produced lists of tables containing matches and near-matches for each clone to reference samples. Clones were also matched against each other in the same way to discover ramets within the clone set. Every match table in the identity search results displayed a list of matching and almost matching samples together with their respective probabilities of identity and non-identity and percentage of matching alleles. All of the results from the identity matching function in GenoSonic were compared to the results that human experts had obtained manually in order to verify the validity of GenoSonic's results.

4.2.7 Visualize measures of relatedness

As discussed in Chapters 2 and 3, GenoSonic uses a simple neighbour-joining algorithm to construct a pairwise related tree structure from a given standard dataset or file upload and presents this tree structure in the Newick file format which can then be copied or downloaded and visualised in most phylogenetic tree viewers.

An example subset of the tree structure generated by GenoSonic on the data from this study:

```
((((AB320:0.2666666666666667,G249:0.2666666666666667):0.840524483147591,G234:0.840524483147591):0.476601362869988,(AB235:0.0132275132275132,AB384:0.0132275132275132):0.667507848176333,G242:0.667507848176333):0.476601362869988):0.454117176945001,((AB363:0,AB264:0):0.036550410711194,G220:0.036550410711194):0.297181545057036,AB330:0.297181545057036):0.454117176945001);
```

Here are some screenshots of how the text structure above can be visualised using a freely available phylogenetic tree drawing software package called PhyloWidget:

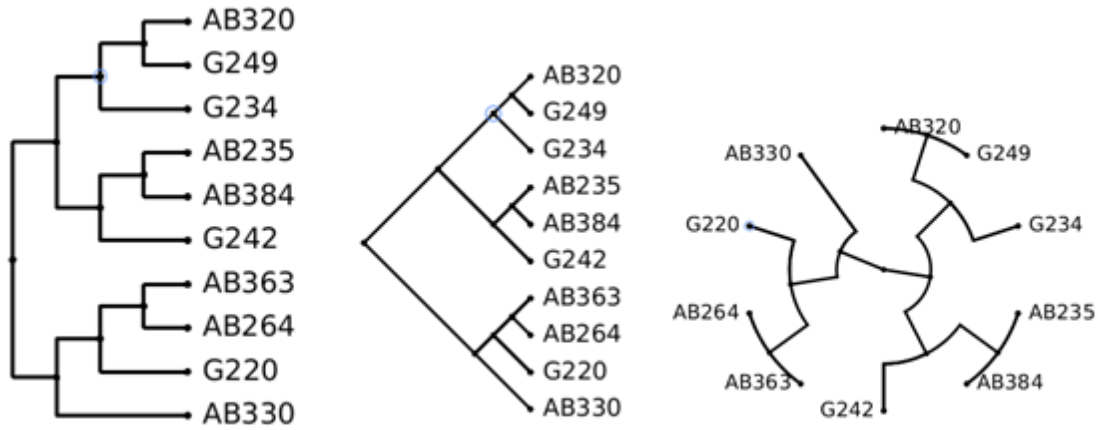


Figure 4.2: Visualization of GenoSonic tree structure output using PhyloWidget. *PhyloWidget* is given a tree structure as a standard input string in the Newick tree file format. It can then visualize this structure using a number of drawing algorithms, including the three shown here. Each leaf node represents a sample. The tree structure shows a measure of relatedness among samples.

These visual structures helped to examine and understand the similarity of fingerprints among samples from the reference- and clone sets. For instance, from the graphics above, it is very easy to recognise that Site AB clone AB320 probably originated from tree G249 or G234, while AB235 and AB384 came from tree G242 and AB363, AB264 and AB330 are probably all clones of tree G220. The Newick description for the complete relatedness tree is included in Appendix A. Example visual layouts are included on the supplementary DVD attached to this document.

4.3 Results and discussion

4.3.1 GenoSonic binning and standardization functions compared to manual analyses

The automated binning results from GenoSonic were compared with the standardized results experts obtained by manually analysing the fingerprints. At first glance, the two sets produced a rather abysmal correspondence, as shown below. There were, however, some substantial obstacles that had to be overcome before the results from the two methods could effectively be compared. The complete allele score tables of initial input data, the results from manual analyses and GenoSonic analyses are available in Appendix A. Comparisons, as shown in Table 4.6, include the number of matching alleles, number of mismatched alleles, the mismatch percentage, the total mismatch distance as the sum of the differences in number of base pairs, and the average mismatch distance as the average difference in number of base pairs per marker:

Table 4.6: First comparison between GenoSonic (automated) and expert (manual) results

Sample	# Matching alleles	#Mismatches	#Alleles	Mismatch%	Total mismatch distance	Average mismatch distance
Marker1	383	85	468	18.2%	12059	141.9
Marker2	40	426	466	91.4%	16205	38.0
Marker3	333	135	468	28.8%	19350	143.3
Marker4	171	296	467	63.4%	25895	87.5
Marker5	306	162	468	34.6%	68516	422.9
Marker6	275	192	467	41.1%	26764	139.4
Marker7	382	86	468	18.4%	6276	73.0
Marker8	316	151	467	32.3%	7443	49.3
Marker9	338	130	468	27.8%	19457	149.7
Marker10	340	128	468	27.4%	22317	174.4
Marker11	134	333	467	71.3%	38545	115.8
Marker12	354	114	468	24.4%	17965	157.6
Total	3372	2238	5610	39.9%	280792	141.1

The first problem had to do with homozygous genotypes and null alleles. By examining the manually analysed set, it seemed that there should have been no heterozygous genotypes where one of the alleles equalled null. The automated result set, however, had many such genotypes. This problem originated when the allele upload file was created by the upstream genotyping software. Homozygous genotypes (genotypes where both alleles have the same size) were scored as single alleles. The only way to determine whether the genotype actually contained two alleles with the same size would be through pedigree assessments of the individuals' ancestors or offspring. The necessary pedigree data were, however, unavailable at the time of the study. The "Replicate all single entries" option of manual standardization step in GenoSonic was used to change all

genotypes with one null allele into homozygous genotypes in the standardized dataset. Another way this could have been achieved would have been to change the way the upload file is created and upload both homozygous alleles to GenoSonic. This correction improved the comparison between results, although it should be pointed out that the presence of null alleles has probably been underestimated because of this blind modification of single-allele genotypes to homozygous genotypes. The new comparison is shown in Table 4.7:

Table 4.7: Second comparison between GenoSonic (automated) and expert (manual) results – after all heterozygous genotypes with one null allele have been converted to homozygous genotypes

Sample	# Matching alleles	# Mismatches	# Alleles	Mismatch %	Total mismatch distance	Average mismatch distance
Marker1	436	32	468	6.8%	893	27.9
Marker2	41	425	466	91.2%	1450	3.4
Marker3	409	59	468	12.6%	6881	116.6
Marker4	227	240	467	51.4%	1092	4.6
Marker5	462	6	468	1.3%	2538	423.0
Marker6	341	127	468	27.1%	1456	11.5
Marker7	426	42	468	9.0%	1006	24.0
Marker8	357	110	467	23.6%	755	6.9
Marker9	412	56	468	12.0%	979	17.5
Marker10	433	35	468	7.5%	527	15.1
Marker11	152	315	467	67.5%	4675	14.8
Marker12	387	81	468	17.3%	3595	44.4
Total	4083	1528	5611	27.3%	25847	59.1

The second problem that had to be addressed had to do with the different conventions used in the naming of bins. The clustering algorithm of CSMerge-1 names bins according to the natural number closest to the mean size of all observed member alleles of the particular cluster. The method in the manual process, however, did not necessarily keep to this convention. This means that, even though both the automatic and manual binning methods grouped the same alleles together into bins, there may be a one- or two-base pair difference in the ultimate size assigned to these bins. To overcome this hurdle in the comparison process, all of the differences of one- or two- base pairs were examined and, where applicable, were renamed uniformly in the comparison set to match the automated result set. This improved the mismatch percentage in the comparison considerably from 27.3% to just 5.9%. The tables below show which alleles had to be changed:

Table 4.8: Renamed alleles in manual set

Marker1		Marker2		Marker3		Marker4		Marker5		Marker6	
From	To	From	To	From	To	From	To	From	To	From	To
210	209	227	226	146	147	211	210			304	303
		258	257			208	207			314	313
		249	248			216	217			317	316
						201	200			335	336
										342	341
Marker7		Marker8		Marker9		Marker10		Marker11		Marker12	
From	To	From	To	From	To	From	To	From	To	From	To
122	123	138	137	227	228	223	222	323	322	452	451
		152	153	225	224			294	293	410	411
		140	139	207	208			319	318	462	463
				205	206			321	320		
								304	303		
								302	301		
								317	316		
								300	299		
								298	297		
								325	324		
								333	332		

The following table shows the comparison after sizes were renamed.

Table 4.9: Third comparison between GenoSonic (automated) and expert (manual) results – after manual sizes have been renamed.

Sample	# Matching alleles	# Mismatches	# Alleles	Mismatch %	Total mismatch distance	Average mismatch distance
Marker1	442	26	468	5.6%	887	34.1
Marker2	363	105	468	22.4%	1128	10.7
Marker3	411	57	468	12.2%	6878	120.7
Marker4	446	22	468	4.7%	869	39.5
Marker5	462	6	468	1.3%	2538	423.0
Marker6	461	7	468	1.5%	1337	191.0
Marker7	428	40	468	8.5%	1004	25.1
Marker8	453	15	468	3.2%	661	44.1
Marker9	454	14	468	3.0%	939	67.1
Marker10	452	16	468	3.4%	509	31.8
Marker11	453	15	468	3.2%	4368	291.2
Marker12	459	9	468	1.9%	3525	391.7
Total	5284	332	5616	5.9%	24643	139.2

The third problem was attributed to user error and missing data. It was determined that there were at least 72 alleles scored in the manually analysed set for which no supporting evidence could be found in any of the original output allele files from GeneMarker. This meant that either there must

have been another source of data which had not been included into the GenoSonic upload, or the alleles must have been incorrectly scored due to some user or other software error.

The fourth issue had to do with low quality data that had been included in the binning process. The upload file included 36 allele scores that were of very low quality, as determined by the peak heights and quality scores from GeneMarker. These alleles had been excluded from the results in the manual binning process by the human experts and had to be removed from GenoSonic as well. After the erroneous allele scores were removed, the comparison summary showed the following:

Table 4.10: Fourth comparison between GenoSonic (automated) and expert (manual) results - After removing missing and erroneous data from comparison

Sample	# Matching alleles	# Mismatches	# Alleles	Mismatch %	Total mismatch distance	Average mismatch distance
Marker1	446	22	468	4.7%	65	3.0
Marker2	367	101	468	21.6%	152	1.5
Marker3	457	11	468	2.4%	138	12.5
Marker4	450	18	468	3.8%	29	1.6
Marker5	468	0	468	0.0%	0	0.0
Marker6	465	3	468	0.6%	33	11.0
Marker7	436	32	468	6.8%	80	2.5
Marker8	457	11	468	2.4%	58	5.3
Marker9	458	10	468	2.1%	39	3.9
Marker10	454	14	468	3.0%	55	3.9
Marker11	467	1	468	0.2%	2	2.0
Marker12	467	1	468	0.2%	29	29.0
Total	5392	224	5616	4.0%	680	6.4

The next two problem areas were due to certain difficult decisions that the automated binning software had to make. The manual standardization step in the binning procedure became very important in the solution of these two problems.

In some cases, the automated binning algorithm chose to include the allele scores from different runs rather than the ones identified by human experts as being the correct scores. This could be seen clearly in the grid control of the manual standardization page. Twenty one samples had been assayed in more than one run (19 from the Site G reference set and 2 from the Site AB clones set). Of these, 21 alleles (from 7 different samples) were found to have been incorrectly chosen by the automated binning algorithm and had to be fixed in the manual standardization step.

The second problem with incorrect decisions made by the automated binning software, arose where CSMerge-1 grouped a set of closely distributed allele sizes into a single bin while the human experts created two bins and *vice versa*. As an example, there were still more than 100 mismatches

(21%) between the automated binning- and manual results for the marker Marker2 and almost all of these mismatches were due to the fact the CSMerge-1 had created a single bin, called 245, for all alleles within a 0.75 base pair radius, while human experts had created two bins, called 244 and 246, to group those same alleles. After re-assessing the initial data, shown in the chart below, it was found that, in this case, CSMerge-1 had probably binned the alleles correctly.

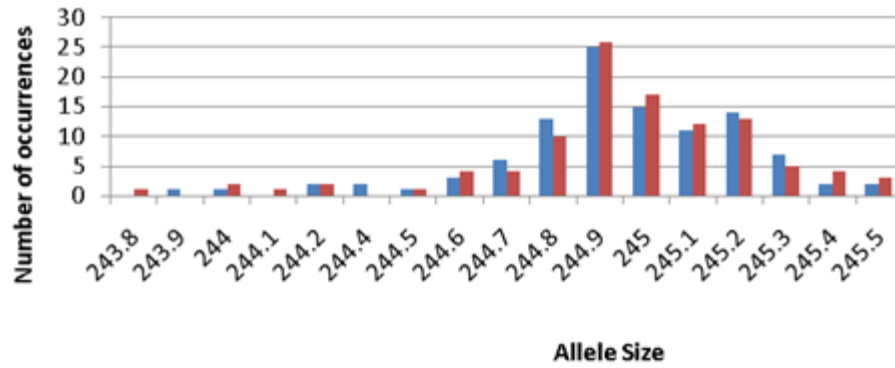


Figure 4.3: Initial allele sizes for marker Marker2 within the size range 242 to 247 before alignment (blue) and after alignment (red) using CSMerge-1. This example serves to show where CSMerge-1 can be very effective in categorising continuous allele sizes. Human experts created standard allele sizes 244 and 246, while CSMerge-1 created a single cluster called 245. This chart clearly shows the mean size from many different sizes across multiple experiments to be 245, which indicates that this is probably the actual size.

A more convoluted case was presented by dinucleotide repeat marker Marker7 for sizes ranging from 112 to 122. Experts had manually grouped alleles of this range into bins 112, 114, 116, 118, 120, and 122. CSMerge-1, however, created bins 112,114,115,116,117,118,119,120,122. By examining the initial data, it was found that CSMerge-1 had probably created a bin set closer to the true nature of the data observed.

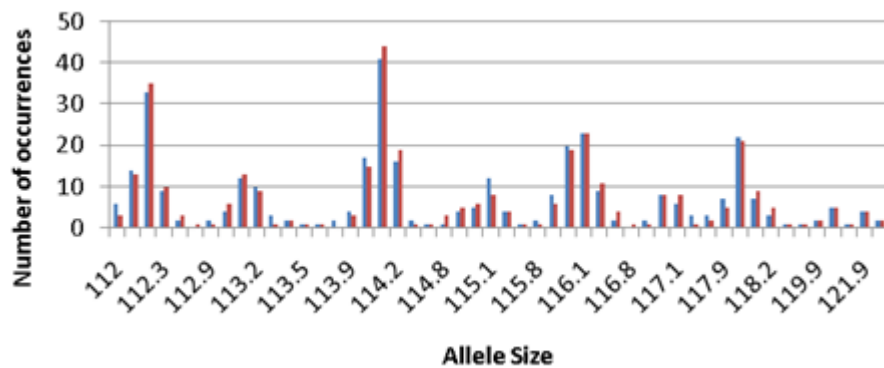


Figure 4.4: Initial allele sizes for marker Marker7 within size range 112 to 122 before alignment (blue) and after alignment (red) using CSMerge-1. This is an example of very noisy data. If allele sizes are separated by margins smaller than the standard repeat size, most likely due to the presence of single-base insertions or deletions, CSMerge-1 will be less accurate in its clustering step.

However, after examining the means of the clusters created by CSMerge-1 for this area, it became clear that in some cases the mean of the clusters did not correlate with the middle of the bell-shaped curves over each allele, as shown above.

Table 4.11: Means of clusters created by CSMerge-1

Mean size of cluster (base pairs)	Size Call
112.11	112
113.81	114
114.89	115
115.86	116
116.98	117
117.86	118
119.00	119
119.99	120
121.90	122
122.76	123

This was probably due to the fact that the operational parameters of the clustering algorithm had been set to group alleles together up to a distance of 1.5 base pairs between any two member alleles in a cluster, which is in this case a greater distance than the size of the differences between each allele. The algorithm creates bins from the clusters containing the most alleles first, and then from less populated clusters in descending order. This means that some of the allele sizes belonging to less frequently observed allele were probably binned into their neighbouring more frequent alleles. This explained why the mean of the cluster for the 112, 116 and 118 bins were centred exactly in the middle of their bell curves, while bin 113 was completely absent and bins 115 and 117's means were not in the expected positions. In this specific case, the problems may have been caused by high variability in the measurements produced by the genotyping equipment. If all these alleles did in fact exist in such a close (1 base pair) proximity to each other, there would probably continue to be a high error rate in binning no matter what method is used. It would actually be advisable to exclude this area or marker from identity matching studies as the probability of a false match or false mismatch occurring is too high.

Cases like the ones above should be investigated manually using the manual standardization step in GenoSonic. GenoSonic also provides the ability to specify a predefined set of standard bins for each allele. The bins created by CSMerge-1 could then be compared to the standard bins before being included in a standardized set. GenoSonic also has optional safeguards in place that would allow the inclusion of alleles into a standardized dataset only if their sizes exist in the list of standard bin sizes that have been predefined for a given marker.

4.3.2 GenoSonic identity matching compared to findings of human experts

The results from the identity matching function in GenoSonic were compared to the results obtained manually by human experts. In nearly all cases the findings by way of manual analyses were similar to the results generated by GenoSonic's bulk identity matching function. There were only three cases where the results did not match and these were caused by one of the same problems that have already been discussed in the section on binning and standardization. Samples AB327, AB233 and G259 were signified as having unique genotypes by GenoSonic, while human experts had assigned them to clonal groups as shown in the table below.

Table 4.12: Mismatches between GenoSonic's identity matching and human expert analysis

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID
AB233	Clone ID 23	Unique genotype	Clone ID 23
AB327	Clone ID 22	Unique genotype	Clone ID 10
G259	Clone ID 5	Unique genotype	Clone ID 5

These three differences can either be attributed to human error or to initial size scoring data having gone missing from the original output file set from GeneMarker that had been uploaded to GenoSonic.

Table 4.13: Comparison of initial allele size scores to binned alleles in GenoSonic and manual analyses for uncorrelated identity matches

AB327			
	Available initial sizes	GenoSonic binned Alleles	Manual binned Alleles
Marker 1	202.7	203, 203	203,203
Marker 2			246,260
Marker 3			
Marker 4	210.7, 213.9	210, 214	211,214
Marker 5	422.6	423, 423	423,423
Marker 6	325.7	326, 326	326,326
Marker 7			116,124
Marker 8			146,158
Marker 9			220,240
Marker 10	228.8	229, 229	229,239
Marker 11			308,314
Marker 12			410,440
AB233			
	Available initial sizes	GenoSonic binned Alleles	Manual binned Alleles
Marker 1	213.1,217.4	214,218	212,218
Marker 2	225.4,244.9	223,245	227,244
Marker 3			157,163
Marker 4	213.5,219.6	214,220	214,214
Marker 5	422.4	423,423	423,423
Marker 6	313.1,325.8	313,326	314,326
Marker 7	125.6	126,126	116,116
Marker 8	163.2,169.6	163,170	154,162
Marker 9	215.4,217.4	216,218	216,218
Marker 10	228	229,229	221,229
Marker 11	307.8,312	308,312	308,312
Marker 12	428,439.8	428,440	428,440
G259			
	Available initial sizes	GenoSonic binned Alleles	Manual binned Alleles
Marker 1			200,203
Marker 2	245.1	245,245	227,227
Marker 3			157,157
Marker 4			211,211
Marker 5			421,423
Marker 6	326	326,326	317,326
Marker 7	109.6	110,110	114,124
Marker 8	153	153,153	138,146
Marker 9	218.2	218,218	220,227
Marker 10	224.7	225,225	229,229
Marker 11			319,321
Marker 12			428,440

The results calculated by GenoSonic were exactly the same as those determined by human experts for all the other samples. A complete comparison of the clonal identities that were determined by the two methods respectively is available in Appendix A.

Given that in GenoSonic the identity matching process is as simplistic as selecting a few options and waiting a few seconds for the results in order to match two entire datasets against each other, whereas the manual process took in the order of days to properly perform clonal identification, it quite clearly demonstrates that GenoSonic can make a valuable contribution in genetic fingerprinting studies such as this one.

4.3.3 Confirmation and re-establishment of clonal identities

Overall, 233 samples were examined, consisting of 43 samples from reference Site G and 190 samples from clonal site AB. Clonal mismatches and other matching issues were found with 59 of the 233 samples. Twenty-eight clonal mismatches could be assigned to other existing clonal groups:

Table 4.14: Clonal mismatches assigned to other existing clonal groups

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID
AB348	Clone ID 11	Clone ID 9	Clone ID 9
AB302	Clone ID 13	Clone ID 28	Clone ID 28
AB259	Clone ID 16	Clone ID 24	Clone ID 24
AB283	Clone ID 16	Clone ID 36	Clone ID 36
AB374	Clone ID 16	Clone ID 36	Clone ID 36
AB226	Clone ID 16	Clone ID 9	Clone ID 9
AB343	Clone ID 16	Clone ID 9	Clone ID 9
AB241	Clone ID 18	Clone ID 4	Clone ID 4
AB265	Clone ID 18	Clone ID 4	Clone ID 4
AB333	Clone ID 26	Clone ID 23	Clone ID 23
AB212	Clone ID 27	Clone ID 31	Clone ID 31
AB246	Clone ID 27	Clone ID 31	Clone ID 31
AB381	Clone ID 28	Clone ID 13	Clone ID 13
AB339	Clone ID 28	Clone ID 23	Clone ID 23
AB230	Clone ID 3	Clone ID 25	Clone ID 25
AB285	Clone ID 3	Clone ID 25	Clone ID 25
AB306	Clone ID 3	Clone ID 25	Clone ID 25
AB314	Clone ID 31	Clone ID 25	Clone ID 25
AB207	Clone ID 33	Clone ID 2	Clone ID 2
AB245	Clone ID 33	Clone ID 2	Clone ID 2
AB369	Clone ID 33	Clone ID 2	Clone ID 2
AB240	Clone ID 4	Clone ID 13	Clone ID 13
AB370	Clone ID 4	Clone ID 13	Clone ID 13
AB273	Clone ID 5	Clone ID 44	Clone ID 44
AB252	Clone ID 7	Clone ID 1	Clone ID 1
AB320	Clone ID 9	Clone ID 26	Clone ID 26
AB235	Clone ID 9	Clone ID 40	Clone ID 40
AB384	Clone ID 9	Clone ID 40	Clone ID 40

Fourteen mismatched samples could be grouped together into seven respective clonal groups to create seven new clonal identities. Two of the new clonal groups, namely 26B and 27B, were

formed by splitting their original clonal groups (26 and 27) into two new groups respectively. In these cases, two reference samples had been received from Site G for each clonal group respectively. The DNA fingerprints of the reference samples within each clonal group were found to be very different, but matching clonal samples from Site AB were observed for each of the reference samples, thus forming new groups. These two cases indicate that there may have been labelling issues in the original orchard as samples from ramets of supposedly the same clonal group clearly have different genotypes. Lastly, five new clonal groups were created from mismatched clones at Site AB, which signifies the possibility of additional mislabelled clones existing at parent Site G.

Table 4.15: Clonal mismatches that form new clonal groups

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID
AB253	Clone ID 12	Clone ID 12-18m	Clone ID 12-18m
AB337	Clone ID 18	Clone ID 12-18m	Clone ID 12-18m
AB274	Clone ID 26	Clone ID 26B	Clone ID 26B
AB307	Clone ID 26	Clone ID 26B	Clone ID 26B
G249	Clone ID 26	Clone ID 26B	Clone ID 26B
AB311	Clone ID 38	Clone ID 27B	Clone ID 27B
G247	Clone ID 27	Clone ID 27B	Clone ID 27B
AB247	Clone ID 28	Clone ID 28-33m	Clone ID 28-33m
AB290	Clone ID 33	Clone ID 28-33m	Clone ID 28-33m
AB323	Clone ID 28	Clone ID 28-33m	Clone ID 28-33m
AB225	Clone ID 5	Clone ID 5B	Clone ID 5B
AB309	Clone ID 5	Clone ID 5B	Clone ID 5B
AB346	Clone ID 7	Clone ID 7-32m	Clone ID 7-32m
AB351	Clone ID 32	Clone ID 7-32m	Clone ID 7-32m

Unique genotypes were observed for eleven samples from Site AB together and six samples from Site G. The unique genotypes from Site AB could have come from many different locations. Since clones are often grafted onto a suitable rootstock, it is possible that the sample could have come from a tree that had developed from the root instead of from the grafted clone. A second possibility is that the sapling is in fact not a clone but a new individual that have grown from seed. There could be many other explanations of how these unique genotypes were introduced, but further speculation is not warranted here.

Table 4.16: Clonal mismatches with unique genotypes

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID
AB208	Clone ID 35	Unique genotype	Unique genotype
AB213	Clone ID 17	Unique genotype	Unique genotype
AB216	Clone ID 21	Unique genotype	Unique genotype
AB233	Clone ID 23	Unique genotype	Clone ID 23
AB255	Clone ID 29	Unique genotype	Unique genotype
AB319	Clone ID 12	Unique genotype	Unique genotype
AB324	Clone ID 22	Unique genotype	Unique genotype
AB327	Clone ID 22	Unique genotype	Clone ID 10
AB330	Clone ID 9	Unique genotype	Unique genotype
AB352	Clone ID 5	Unique genotype	Clone ID 5
AB383	Clone ID 26	Unique genotype	Unique genotype
G225	Clone ID 14	Unique genotype	Unique genotype
G226	Clone ID 15	Unique genotype	Unique genotype
G248	Clone ID 27	Unique genotype	Unique genotype
G251	Clone ID 15	Unique genotype	Unique genotype
G252	Clone ID 15	Unique genotype	Unique genotype
G259	Clone ID 5	Unique genotype	Clone ID 5

4.4 Conclusion

DNA fingerprints of 190 samples of *Pinus patula* clones from a new clonal seed orchard were ascertained and compared to 43 fingerprints from the original orchard in order to validate or re-establish clonal identities. Of the 233 samples assessed, 59 did not match the genotypes of their supposed clonal group. These results suggest that there may have been very high error rate present in the replication -, labelling-, or both processes. It would therefore be advisable to conduct a larger fingerprinting study on the remainder of the trees in both of the clonal orchards before doing any further work or research reliant on accurate knowledge of the trees' respective clonal identities. Indeed, the directive that had been given following the outcome of the manual analyses was to initiate a much larger DNA fingerprinting exercise on all, or at least most, of the trees in both orchards.

GenoSonic's capabilities have been tested with regard to its implementation of each of the functional requirements set for the project by utilising the software to analyse genetic fingerprint data from *Pinus patula* orchards with the goal of verifying or re-establishing clonal identities. The first goal with regard to assessing the suitability of the software to assisting in the study was to create a secure, easily accessible place where genotyping project data can be maintained. To achieve this goal, a website has been developed as an interface to a secure database. Depending on network access policies, the website could be made accessible to potentially the whole world. Project data are secure as users can only query data from projects of which they are members. At

this time one shortcoming is that there is no role-based security implemented for the site. This means that all users have the same permissions and no specific functionality has yet been created for administrators, power users, or project owners. This is definitely an improvement that can be made in the future.

The second goal was to provide users with ways to easily upload and download project data. Users can upload data to GenoSonic via .csv files. To ease the upload process, a simple templating system has been developed together with configurable upload behaviours. This makes uploading data to GenoSonic extremely flexible and easy to use. Downloading data from GenoSonic is also very simple in that one only needs to click the download button to receive a .csv file containing the entire project's dataset for the current data type. In the future the download functionality could be enhanced by allowing for the export of different data types, like XML, and enabling the download of subsets of data instead of the entire dataset- or project's records.

The third goal was to create functionality for the standardization of genotype data by way of an automated binning step combined with a manual assessment step. CSMerge-1 was employed in the automated step to merge and bin genetic fingerprint data from different experiments. It was concluded that CSMerge-1 is a much quicker and, in most cases, more accurate way of binning than doing it manually. The only areas where automated binning may lose accuracy are in cases when the data is very noisy and when the alleles are separated by less than the expected repeat size, possibly due to mutations in certain individuals. For these cases, however, GenoSonic's manual standardization step should be employed. This function allows users to manually inspect the results from automated binning and compare it with known standard sizes and external size calls before committing the results to standardized datasets. By combining the automated and manual binning steps, GenoSonic improves both the speed and accuracy of the binning process.

The fourth goal was to provide a way of determining allele- and genotype frequencies for a given dataset. GenoSonic supplies a simple web page where users can view such frequency data by choosing the desired dataset and marker. This allows users to easily draw conclusions about allele distributions and genetic diversity.

The fifth goal was to enable users to do identity matching enquiries on the datasets in GenoSonic. Users can enter allele sizes manually, upload files, choose single samples from datasets, or choose entire datasets to match against any given reference dataset. This very flexible querying method greatly increases the speed at which identity matching can be done, compared to manually

examining profiles. This is especially true when making use of the bulk-matching functionality which compares multiple test samples to multiple reference samples.

The sixth goal was to provide a way to visualize a measure of relatedness among samples. GenoSonic contains functionality for a relatedness tree to be constructed for any given dataset. The tree is given in a standard Newick tree text format, rather than rendering the image directly. Since there are many phylogenetic tree visualization tools in existence, it made much more sense to export the tree structure to a standard text format than to create yet another custom image rendering tool. This allows the user to import the tree structure into almost any specialized tree rendering software for performing further analysis or special rendering functions.

Overall, this case study proved that GenoSonic can be utilised in a genetic fingerprinting study since the results obtained by using GenoSonic correlated with and even surpassed the manual results in terms of accuracy and consistency, and far surpassed the manual effort in terms of the speed at which analyses could be completed. In terms of functional requirements, it was shown that the implementation of GenoSonic satisfied each of the functional goals specified by the problem statement. It can therefore be concluded that overall GenoSonic is a satisfactory solution to the requirements by this project.

Chapter 5

Concluding Discussion

Genotyping technologies like microsatellite marker assays can be used in many commercial, academic, social, and agricultural applications. There are, however, many obstacles in effectively managing and analysing microsatellite genotype data in larger volumes, over extended periods of time, or across different experiments that may have been assayed using different genotyping equipment and software. Some sophisticated proprietary genotyping data management software packages do exist to address these problems, but they are mostly very expensive. There are a few other open-source applications that could be conceived of as viable solutions for specific or even general problems regarding the current needs for genotyping data management or statistical analysis, but the opportunities for reuse and extensibility of these applications to suit unforeseen requirements are rather limited.

The main aim of this study was to provide a software solution that would address a specific set of user requirements and consequently enhance the capabilities and alleviate some of the complications associated with the management and analyses of microsatellite marker data in DNA fingerprinting studies. The bioinformatics resource that has been presented here is a software solution called GenoSonic. It is a web application that provides users with a secure, easily accessible space where they can manage their genotyping project data as a team. Users can easily upload and download large amounts of data to and from the system. Once uploaded to the system, genetic fingerprint data needs to be standardised before it can be used in further analyses. To do this, GenoSonic has implemented a two-step approach. The first step is to assign discrete alleles to all of the uploaded allele sizes of the fingerprints automatically using a novel automated binning algorithm called CSMerge-1, which has been designed specifically to bin data from multiple experiments, especially where certain samples are assayed in more than one experiment. The second step is to manually verify the results from the automated binning function and adding it to a standardized dataset. Once the genetic fingerprints have been standardized, allele- and genotyping frequencies can be viewed for any given marker. GenoSonic also provides functionalities for identity matching. One or many genetic fingerprints from unknown samples can be matched against a standardized dataset in order to establish identities and in some cases infer relatedness. Finally, GenoSonic implements a relatedness tree construction function which can be used to infer measures of relatedness among samples in a dataset.

All the abovementioned functionality in GenoSonic was applied to a genetic fingerprinting assignment aimed at the re-establishment or confirmation of clonal identities of *Pinus patula* ramets. The results from GenoSonic's automated binning function (CSMerge-1) and the results from the identity matching exercise were compared to results obtained by human experts who had analysed the data manually. It was shown that GenoSonic equalled or surpassed manual effort in

terms of accuracy and consistency, and far surpassed the manual effort in terms of the speed at which analyses could be completed.

The study concluded that there could be a disparity of over 25% between perceived clonal identities and actual identities of ramets in the newly created clonal seed orchard of this particular study. Mislabeled ramets could have been introduced by a number of issues in the cloning and labelling process, including mislabeled clonal parents, errors occurring while labelling trays of saplings in nurseries, unique individuals growing from seeds mixed with supposed clones and even rootstock of clones growing into trees where the grafts of clones failed. The only way of confirming or re-establishing genuine clonal identities for entire clonal seed orchards is by undertaking DNA fingerprinting studies such as this one, but on scale that would allow the testing of thousands of samples at a time. Manual analyses would require an immense effort in terms of complexity and man-hours to undertake such larger fingerprinting studies. GenoSonic has shown to be able to do exactly the kind of analyses needed at a much higher throughput than what could be achieved manually. The ability to analyse thousands of DNA fingerprints within minutes will become more valuable as the sample sizes of future studies continue to increase.

Researchers from many other fields could benefit from conducting similar kinds of higher-throughput DNA fingerprinting studies with GenoSonic. The software is, however, by no means limited to specific organisms or even specifically to DNA fingerprinting studies, but can be employed in many other kinds of studies that make use of microsatellite marker genotyping. For example, any study that uses microsatellite markers can benefit from the automated binning functionality (CSMerge-1) for standardizing microsatellite genotype data across multiple experiments, even though the results will be used for something other than DNA fingerprinting of individuals. The same can be said for the relatedness tree diagrams that can be created with GenoSonic.

GenoSonic has been developed by following an agile software engineering lifecycle. A formal software engineering approach was followed because it was an aim of this project to not only provide a solution that satisfies the current needs of the pilot study, but also to show that a software solution's maintainability, reusability and extensibility and therefore its lifetime and value to the community, can be greatly enhanced by following proper software design principles and patterns. One way in which this has been achieved in GenoSonic has been by compartmentalising its architecture into a set of lowly coupled, almost independent components, which can be exchanged for newer or different components, depending on future needs. Many of these components, like the binning and analysis functions, can also very easily be converted into independent applications,

while other components like the database or data access layer can be reused as core components of completely new applications. GenoSonic also exposes its core functionality via a set of web services, which will allow future developers or researchers to extend the current functionality by writing custom applications that can communicate with these services.

Future development may include creating a desktop client application for GenoSonic instead of using the current web browser interface. This might be considered since the most probable areas to which functionality in GenoSonic could be extended include performing Excel-grid-like functions, complicated graph-drawing functions, or chart-creation functions. Programming libraries and tools used to create these types of functionalities are currently much more advanced and better supported for desktop applications than browser-based web applications. This may, of course, also change dramatically in coming years as technology moves toward web-based, not necessarily web-browser based, computation.

Another consideration for future development is the extension of GenoSonic to facilitate the management of other types of genetic marker data, such as SNPs or AFLPs. This would free researchers from having to use only microsatellites in their genetic fingerprinting studies. It may also enhance inference capabilities, as single studies could utilise multiple marker technologies together to draw conclusions that would otherwise have been impossible.

Finally, since GenoSonic was written to be a malleable core solution consisting of reusable and interchangeable software components, it is hoped that the source code will be maintained online as an open-source project and that organisations from around the globe can use as their basic framework or as part of their analysis libraries when designing their own specific genotyping solutions.

Bibliography

- Adkins, R.M. & Honeycutt, R.L. 1991, "Molecular phylogeny of the superorder *Archonta* ", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 88, no. 22, pp. 10317-10321.
- Armour, J.A., Povey, S., Jeremiah, S. & Jeffreys, A.J. 1990, "Systematic cloning of human minisatellites from ordered array charomid libraries", *Genomics*, vol. 8, no. 3, pp. 501-512.
- Avise, J.C. 2004, "Restriction analyses" in *Molecular Markers, Natural History and Evolution*, Second Edition edn, Sinauer Associates, Inc., pp. 67-82.
- Aziz, A. 2009, *ELMAH : Error Logging Modules and Handlers for ASP.NET* [Homepage of Google Code], [Online]. Available: <http://code.google.com/p/elmah/> [2009, 11/29] .
- Bailey, W.J., Slightom, J.L. & Goodman, M. 1992, "Rejection of the "flying primate" hypothesis by phylogenetic evidence from the epsilon-globin gene", *Science (New York, N.Y.)*, vol. 256, no. 5053, pp. 86-89.
- Balloux, F. & Lugon-Moulin, N. 2002, "The estimation of population differentiation with microsatellite markers", *Molecular ecology*, vol. 11, no. 2, pp. 155-165.
- Beaumont, M.A. 1999, "Detecting population expansion and decline using microsatellites", *Genetics*, vol. 153, no. 4, pp. 2013-2029.
- Bennett, S., Alexander, L.J., Crozier, R.H. & Mackinlay, A.G. 1988, "Are megabats flying primates? Contrary evidence from a mitochondrial DNA sequence", *Australian Journal of Biological Sciences*, vol. 41, no. 3, pp. 327-332.
- Botstein, D., White, R.L., Skolnick, M. & Davis, R.W. 1980, "Construction of a genetic linkage map in man using restriction fragment length polymorphisms.", *American Journal of Human Genetics*, vol. 32, pp. 314-333.
- Chikhi, L., Bruford, M.W. & Beaumont, M.A. 2001, "Estimation of admixture proportions: a likelihood-based approach using Markov Chain Monte Carlo", *Genetics*, vol. 158, no. 3, pp. 1347-1362.
- Conery, R. 2010, *SubSonic Source code - Hosted on Github*. Available: <http://github.com/subsonic/SubSonic-3.0> [2009, 11/29] .
- Conery, R. 2009a, *SubSonic Project Homepage*. Available: www.subsonicproject.com [2009, 11/29] .
- Conery, R. 2009b, *Wekeroad: Rob Conery's Blog*. Available: www.wekeroad.com [2009, 11/29] .
- Cruzan, M.B. 1998, "Genetic markers in plant evolutionary ecology.", *Ecology*, vol. 79, pp. 400-412.
- DeSalle, R. 1992a, "The origin and possible time of divergence of the Hawaiian *Drosophilidae*: evidence from DNA sequences", *Molecular biology and evolution*, vol. 9, no. 5, pp. 905-916.
- DeSalle, R. 1992b, "The phylogenetic relationships of flies in the family *Drosophilidae* deduced from mtDNA sequences", *Molecular phylogenetics and evolution*, vol. 1, no. 1, pp. 31-40.

- Diniz, R. *Extjs Extender Controls* [Homepage of Codeplex], [Online]. Available: <http://www.codeplex.com/ExtJsExtenderControl> [2009, 11/29] .
- Edwards, K.J. 1998, "Randomly amplified polymorphic DNAs (RAPDs)" in *Molecular Tools for Screening Biodiversity*, eds. A. Karp, P.G. Isaac & D.S. Ingram, Chapman & Hall, pp. 171-205.
- Ellegren, H. 2000, "Microsatellite mutations in the germline: implications for evolutionary inference", *Trends in genetics : TIG*, vol. 16, no. 12, pp. 551-558.
- Elmasri, R. & Navathe, S. 2006, "Impedance mismatch" in *Fundamentals of database systems*, 5th edn, Addison Wesley, pp. 292.
- Estoup, A., Solignac, M. & Cornuet, J.M. 1994, "Precise assessment of the number of patriline and of genetic relatedness in honey bee colonies.", *Proceedings of the Royal Society of London: Biological Sciences*, vol. 258, pp. 1-7.
- Estoup, A., Garnery, L., Solignac, M. & Cornuet, J.M. 1995, "Microsatellite variation in honey bee (*Apis mellifera* L.) populations: hierarchical genetic structure and test of the infinite allele and stepwise mutation models", *Genetics*, vol. 140, no. 2, pp. 679-695.
- Fowler, M. 2002, *Patterns of enterprise application architecture*, Addison-Wesley Professional.
- Gentile, G. & Sbordoni, V. 1998, "Indirect methods to estimate gene flow in cave and surface populations of *Androniscus dentiger* (Isopoda: Oniscidea) ", *Evolution*, vol. 52, no. 2, pp. 432-442.
- González, S., Maldonado, J.E., Leonard, J.A., Vilà, C., Barbanti Duarte, J.M., Merino, M., Brum-Zorrilla, N. & Wayne, R.K. 1998, "Conservation genetics of the endangered Pampas deer (*Ozotoceros bezoarticus*)", *Molecular ecology*, vol. 7, no. 1, pp. 47-56.
- Halfond, W., Viegas, J. & Orso, A. "A classification of SQL-injection attacks and countermeasures", .
- Hamada, H., Petrino, M.G. & Kakunaga, T. 1982, "A novel repeated element with Z-DNA-forming potential is widely found in evolutionary diverse eukaryotic genomes", *Proceedings of the National Academy of Science, USA*, vol. 79, pp. 6465-6469.
- Helminen, P., Ehnholm, C., Lokki, M.L., Jeffreys, A. & Peltonen, L. 1988, "Application of DNA "fingerprints" to paternity determinations", *Lancet*, vol. 1, no. 8585, pp. 574-576.
- Heyer, L.J., Kruglyak, S. & Yooseph, S. 1999, "Exploring expression data: identification and analysis of coexpressed genes", *Genome Research*, vol. 9, no. 11, pp. 1106-15.
- Hughes, S. 2010, *STRand Nucleic Acid Analysis Software*. Available: <http://www.vgl.ucdavis.edu/informatics/strand.php> [2010, 10/14] .
- Idury RM, C.L. 1997, "A simple method for automated allele binning in microsatellite markers", *Genome research*, vol. 7, pp. 1104-1109.
- Jayashree, B., Reddy, P., Leeladevi, Y., Crouch, J., Mahalakshmi, V., Buhariwalla, H., Eshwar, K., Mace, E., Folksterma, R., Senthilvel, S., Varshney, R., Seetha, K., Rajalakshmi, R., Prasanth, V., Chandra, S., Swarupa, L., SriKalyani, P. & Hoisington, D. 2006, "Laboratory Information Management Software for genotyping workflows: applications in high throughput crop genotyping", *BMC Bioinformatics*, vol. 7, no. 383.

- Jeffreys, A.J., Barber, R., Bois, P., Buard, J., Dubrova, Y.E., Grant, G., Hollies, C.R., May, C.A., Neumann, R., Panayi, M., Ritchie, A.E., Shone, A.C., Signer, E., Stead, J.D. & Tamaki, K. 1999, "Human minisatellites, repeat DNA instability and meiotic recombination", *Electrophoresis*, vol. 20, no. 8, pp. 1665-1675.
- Jeffreys, A.J., Royle, N.J., Wilson, V. & Wong, Z. 1988, "Spontaneous mutation rates to new length alleles at tandem-repetitive hypervariable loci in human DNA", *Nature*, vol. 332, no. 6161, pp. 278-281.
- Jeffreys, A.J., Wilson, V. & Thein, S.L. 1985a, "Hypervariable 'minisatellite' regions in human DNA", *Nature*, vol. 314, no. 6006, pp. 67-73.
- Jeffreys, A.J., Wilson, V. & Thein, S.L. 1985b, "Individual-specific 'fingerprints' of human DNA", *Nature*, vol. 316, no. 6023, pp. 76-79.
- Karem, B.e.a. 1989, "Identification of the cystic fibrosis gene: genetic analysis.", *Science*, vol. 245, pp. 1073-1080.
- Kemp, E. 2009, *Subsonic's SubSonic-3.0-Templates at master - GitHub* [Homepage of GitHub], [Online]. Available: <http://github.com/subsonic/SubSonic-3.0-Templates> [2009, 11/29] .
- Kendal, W.S. 2003, "An exponential dispersion model for the distribution of human single nucleotide polymorphisms", *Molecular biology and evolution*, vol. 20, no. 4, pp. 579-590.
- Kirst, M., Cordeiro, C.M., Rezende, G.D. & Grattapaglia, D. 2005, "Power of microsatellite markers for fingerprinting and parentage analysis in *Eucalyptus grandis* breeding populations", *The Journal of heredity*, vol. 96, no. 2, pp. 161-166.
- Konieczny, A. & Ausubel, F.M. 1993, "A procedure for mapping *Arabidopsis* mutations using co-dominant ecotype-specific PCR-based markers", *The Plant Journal : for cell and molecular biology*, vol. 4, no. 2, pp. 403-410.
- Krishnamurthy, K. 2003, "Genetic Diversity" in *Textbook of biodiversity*, Illustrated edn, Science Publishers, pp. 14.
- Kruglyak, L. & Nickerson, D.A. 2001, "Variation is the spice of life", *Nature genetics*, vol. 27, no. 3, pp. 234-236.
- Kuhner, M.K., Beerli, P., Yamato, J. & Felsenstein, J. 2000, "Usefulness of single nucleotide polymorphism data for estimating population parameters", *Genetics*, vol. 156, no. 1, pp. 439-447.
- Kuhnlein, U., Zadworny, D., Dawe, Y., Fairfull, R.W. & Gavora, J.S. 1990, "Assessment of inbreeding by DNA fingerprinting: development of a calibration curve using defined strains of chickens", *Genetics*, vol. 125, no. 1, pp. 161-165.
- Kwiatowski, J., Skarecky, D., Bailey, K. & Ayala, F.J. 1994, "Phylogeny of *Drosophila* and related genera inferred from the nucleotide sequence of the Cu,Zn Sod gene", *Journal of Molecular Evolution*, vol. 38, no. 5, pp. 443-454.
- Li, J., Deng, H., Lai, D., Xu, F., Chen, J., Gao, G., Recker, R. & Deng, H. 2001, "Toward high-throughput genotyping: dynamic and automatic software for manipulating large-scale genotype data using fluorescently labeled dinucleotide markers", *Genome research*, vol. 11, pp. 1304-1314.

- Linn, S. & Arber, W. 1968, "Host specificity of DNA produced by *Escherichia coli*, X. In vitro restriction of phage fd replicative form", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 59, no. 4, pp. 1300-1306.
- Litt, M. & Luty, J.A. 1989, "A hypervariable microsatellite revealed by *in vitro* amplification of a dinucleotide repeat within cardiac muscle actin gene.", *American Journal of Human Genetics*, vol. 44, no. 3, pp. 397-401.
- Liu, K. & Muse, S. 2005, "PowerMarker: Integrated analysis environment for genetic marker data", *Bioinformatics*, vol. 21, no. 9, pp. 2128-2129.
- Locke, M., Baack, E. & Toonen, R. (2000), *The STRand Manual*.
- Lowe, A.J., Hanotte, O. & Guarino, L. 1996, "Standardization of molecular genetic techniques for the characterization of germplasm collections: The case of random amplified polymorphic DNA (RAPD).", *Plant Genetic Resource Newsletter*, vol. 1, no. 107, pp. 50-54.
- Lynch, M. 1988, "Estimation of relatedness by DNA fingerprinting", *Molecular biology and evolution*, vol. 5, no. 5, pp. 584-599.
- Matthes, M.C., Daly, A. & Edwards, K.J. 1998, "Amplified fragment length polymorphism (AFLP)" in *Molecular tools for screening biodiversity*, eds. A. Karp, P.G. Isaac & D.S. Ingram, Chapman & Hall, pp. 183-190.
- Meijer, E., Beckman, B. & Bierman, G. 2006, "LINQ: reconciling object, relations and XML in the .NET framework", *International Conference on Management of Data. Proceedings of the 2006 ACM SIGMOD international conference on Management of data* ACM, New York, NY, USA.
- Meselson, M. & Yuan, R. 1968, "DNA restriction enzyme from *E. coli* ", *Nature*, vol. 217, no. 5134, pp. 1110-1114.
- Miller, J. 2009, *StructureMap Home Page* [Homepage of SourceForge], [Online]. Available: <http://structuremap.sourceforge.net/Default.htm> [2009, 11/29] .
- Mindell, D.P., Dick, C.W. & Baker, R.J. 1991, "Phylogenetic relationships among megabats, microbats, and primates", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 88, no. 22, pp. 10322-10326.
- Mullis, K.B. & Faloona, F.A. 1987, "Specific synthesis of DNA in vitro via a polymerase-catalyzed chain reaction", *Methods in enzymology*, vol. 155, pp. 335-350.
- Munthali, M., Ford-Lloyd, B.V. & Newbury, H.J. 1992, "The random amplification of polymorphic DNA for fingerprinting plants", *PCR methods and applications*, vol. 1, no. 4, pp. 274-276.
- Nakamura, Y., Leppert, M., O'Connell, P., Wolff, R., Holm, T., Culver, M., Martin, C., Fujimoto, E., Hoff, M. & Kumlin, E. 1987, "Variable number of tandem repeat (VNTR) markers for human gene mapping", *Science (New York, N.Y.)*, vol. 235, no. 4796, pp. 1616-1622.
- NCBI 2010, *dbSNP Summary* [Homepage of NCBI], [Online]. Available: http://www.ncbi.nlm.nih.gov/SNP/snp_summary.cgi [2010, 10/14] .

- Nielsen, R. & Signorovitch, J. 2003, "Correcting for ascertainment biases when analyzing SNP data: applications to the estimation of linkage disequilibrium", *Theoretical population biology*, vol. 63, no. 3, pp. 245-255.
- Paabo, S. 1989, "Ancient DNA: extraction, characterization, molecular cloning, and enzymatic amplification", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 86, no. 6, pp. 1939-1943.
- Paran, I. & Michelmore, R.W. 1993, "Development of reliable PCR based markers linked to downy mildew resistance genes in lettuce", *Theor. Appl. Genet.*, vol. 85, pp. 985-993.
- Parker, P.G., Snow, A.A., Schug, M.D., Booton, G.C. & Fuerst, P.A. 1998, "What molecules can tell us about populations: choosing and using a molecular marker.", *Ecology*, vol. 79, pp. 361-382.
- Perez, T., Albornoz, J. & Dominguez, A. 1998, "An evaluation of RAPD fragment reproducibility and nature", *Molecular ecology*, vol. 7, no. 10, pp. 1347-1357.
- Pettigrew, J.D. 1994, "Genomic evolution. Flying DNA", *Current Biology : CB*, vol. 4, no. 3, pp. 277-280.
- Pettigrew, J.D. 1986, "Flying primates? Megabats have the advanced pathway from eye to midbrain", *Science (New York, N.Y.)*, vol. 231, no. 4743, pp. 1304-1306.
- Piano, F., Craddock, E.M. & Kambysellis, M.P. 1997, "Phylogeny of the island populations of the Hawaiian *Drosophila grimshawi* complex: evidence from combined data", *Molecular Phylogenetics and evolution*, vol. 7, no. 2, pp. 173-184.
- Powell, W., Machray, G.C. & Provan, J. 1996, "Polymorphism revealed by simple sequence repeats.", *Trends in Plant Science*, vol. 1, no. 7, pp. 215-222.
- Prakash, S., Lewontin, R.C. & Hubby, J.L. 1969, "A molecular approach to the study of genic heterozygosity in natural populations IV. Patterns of genic variation in central, marginal and isolated populations of *Drosophila pseudoobscura*.", *Genetics*, vol. 61, no. 4, pp. 841 - 858.
- Roberts, R.J. 1978, "Restriction and modification enzymes and their recognition sequences", *Gene*, vol. 4, no. 3, pp. 183-194.
- Saiki, R.K., Gelfand, D.H., Stoffel, S., Scharf, S.J., Higuchi, R., Horn, G.T., Mullis, K.B. & Erlich, H.A. 1988, "Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase", *Science (New York, N.Y.)*, vol. 239, no. 4839, pp. 487-491.
- Saiki, R.K., Scharf, S., Faloona, F., Mullis, K.B., Horn, G.T., Erlich, H.A. & Arnheim, N. 1985, "Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia", *Science (New York, N.Y.)*, vol. 230, no. 4732, pp. 1350-1354.
- Schlotterer, C. 2000, "Evolutionary dynamics of microsatellite DNA", *Chromosoma*, vol. 109, no. 6, pp. 365-371.
- Selander, R.K. 1970, "Behavior and genetic variation in natural populations", *American Zoologist*, vol. 10, no. 1, pp. 53-66.
- Shaw, C.R. & Prasad, R. 1970, "Starch gel electrophoresis of enzymes--a compilation of recipes", *Biochemical genetics*, vol. 4, no. 2, pp. 297-320.

- Southern, E.M. 1975, "Detection of specific sequences among DNA fragments separated by gel electrophoresis", *Journal of Molecular Biology*, vol. 98, no. 3, pp. 503-517.
- Springer, M.S., Teeling, E.C., Madsen, O., Stanhope, M.J. & de Jong, W.W. 2001, "Integrated fossil and molecular data reconstruct bat echolocation", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 11, pp. 6241-6246.
- Stewart, A. 2008, *STRlab - Forensic DNA samples and Profiles Database*. Available: <http://strlab.co.za/> [2010, 2010-12-03]
- Sych, O. 2009, *T4 Toolbox* [Homepage of Codeplex], [Online]. Available: <http://www.codeplex.com/t4toolbox> [2009, 11/29] .
- Tautz, D. & Renz, M. 1984, "Simple sequences are ubiquitous repetitive components of eukaryotic genomes", *Nucleic Acids Research*, vol. 12, no. 10, pp. 4127-4138.
- Teeling, E.C., Madsen, O., Van den Bussche, R.A., de Jong, W.W., Stanhope, M.J. & Springer, M.S. 2002, "Microbat paraphyly and the convergent evolution of a key innovation in Old World rhinolophoid microbats", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 3, pp. 1431-1436.
- Teeling, E.C., Scally, M., Kao, D.J., Romagnoli, M.L., Springer, M.S. & Stanhope, M.J. 2000, "Molecular evidence regarding the origin of echolocation and flight in bats", *Nature*, vol. 403, no. 6766, pp. 188-192.
- The International HapMap Consortium 2003, "The International HapMap Project", *Nature*, vol. 426, no. 6968, pp. 789-796.
- Turner, B.J. 1982, "The evolutionary genetics of a unisexual fish, *Poecilia formosa*", *Progress in clinical and biological research*, vol. 96, pp. 265-305.
- Vos, P., Hogers, R., Bleeker, M., Reijmans, M., van de Lee, T., Hornes, M., Frijters, A., Pot, J., Peleman, J. & Kuiper, M. 1995, "AFLP: a new technique for DNA fingerprinting", *Nucleic Acids Research*, vol. 23, no. 21, pp. 4407-4414.
- Wakeley, J., Nielsen, R., Liu-Cordero, S.N. & Ardlie, K. 2001, "The discovery of single-nucleotide polymorphisms--and inferences about human demographic history", *American Journal of Human Genetics*, vol. 69, no. 6, pp. 1332-1347.
- Wall, J.D., Andolfatto, P. & Przeworski, M. 2002, "Testing models of selection and demography in *Drosophila simulans*", *Genetics*, vol. 162, no. 1, pp. 203-216.
- Weising, K., Nybom, H., Wolff, K. & Meyer, W. 1995, "DNA fingerprinting in plants and fungi" in *DNA fingerprinting in plants and fungi*, ed. A. Arbor, CRC Press, pp. 1-3.
- Welsh, J., Petersen, C. & McClelland, M. 1991, "Polymorphisms generated by arbitrarily primed PCR in the mouse: application to strain identification and genetic mapping", *Nucleic Acids Research*, vol. 19, no. 2, pp. 303-306.
- Williams, J.G., Kubelik, A.R., Livak, K.J., Rafalski, J.A. & Tingey, S.V. 1990, "DNA polymorphisms amplified by arbitrary primers are useful as genetic markers", *Nucleic acids research*, vol. 18, no. 22, pp. 6531-6535.

- Wilson, I.J. & Balding, D.J. 1998, "Genealogical inference from microsatellite data", *Genetics*, vol. 150, no. 1, pp. 499-510.
- Wyman, A.R. & White, R. 1980, "A highly polymorphic locus in human DNA", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 77, no. 11, pp. 6754-6758.
- Xu, J., Kerrigan, R.W., Sonnenberg, A.S., Callac, P., Horgen, P.A. & Anderson, J.B. 1998, "Mitochondrial DNA variation in natural populations of the mushroom *Agaricus bisporus*", *Molecular Ecology*, vol. 7, no. 1, pp. 19-34.
- Zabeau, M. & Vos, P. (1993), *Selective restriction fragment amplification: a general method for DNA fingerprinting*, C12Q 1/68 (2006.01), C12Q 1/70 (2006.01), C07H 21/00 (2006.01), Netherlands.
- Zietkiewicz, E., Rafalski, A. & Labuda, D. 1994, "Genome fingerprinting by simple sequence repeat (SSR)-anchored polymerase chain reaction amplification", *Genomics*, vol. 20, no. 2, pp. 176-183.

Appendix A – Allele Scores

Table A.1: Allele score table from GeneMarker used as input for manual analysis vs. GenoSonic analysis in the *Pinus patula* study

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB195	114 - 2009-08-24-01-A	200.4	202.7	226.2	156.8	162.9	210.6	213.9	420.6	424.5	325.8	114.2	116.1	153.1	160.3	211.8	215.9	220.7	235.1	305.8	322.5	433.9	439.7		
AB196	114 - 2009-08-24-01-A	200.4	202.7	226.1	257	156.7	162.9	210.5	422.6		303.4	325.8	116.1	123.9	145.8		211.7	220.1	226.7		291.7	427.9	451.5		
AB197	114 - 2009-08-24-01-A	213.7	218.1		156.8	162.9	214.1	422.7			313.2	325.9	116.1		154.3	162.3	215.9	218	220.7	228.8	307.9	312.1	428.1	439.9	
AB198	114 - 2009-08-24-01-A	202.7	213.9	226.3	245.4	151.9		214.4	422.6	424.6	316.5	326	104.5	116.3	137.4	154.3	211.9	220.2	226.8		293.6	410.4	428		
AB199	114 - 2009-08-24-01-A	214	216.3	226.4		157.7	162.9	210.7	422.6		326	329.2	112.3	118.1	146	162.3	218.1		226.8	228.9	312.1	320.5	425	462.8	
AB200	114 - 2009-08-24-01-A	202.7	211.7	226.3	257.2	156.8	162.8	214	422.6		316.5	326.1	110.3	114.1	153.1	154.3	220.2		226.7	237.3	318.5		422.3	451.4	
AB201	114 - 2009-08-24-01-A	202.7	213.8	226.2		156.8	162.9	214.4	422.6		322.9	326.1	110.3	112.2	145.9	154.3	220.2	222.3	226.9	229.9	303.7	312.2	428	433.9	
AB202	114 - 2009-08-24-01-A	214.1	216.4	226.5		156.8	162.9	210.9	214.4	410.7	424.6		326.3		110.5	118.2		137.7	146	212.1	216.2	227.1	293.7	430.8	439.8
AB203	114 - 2009-08-24-01-A	211.6	218	226		156.8	162.8	210.6	214	422.6		325.9	329	110.3	114.1	150.1	160.3	219.9		226.8		293.6		422.2	
AB204	114 - 2009-08-24-01-A	202.7		226.2	257.1	156.8	162.9	210.7	213.9	424.6		325.9		112.3		145.9	153.2	222.2		220.6	228.9	301.5	318.5	439.8	
AB205	114 - 2009-08-24-01-A	202.6	215.8	225.9	245.2	151.9		210.6	422.5	424.6	325.4		114.2	122	145.8	153.1	220		227		293.5	312	439.7		
AB206	114 - 2009-08-24-01-A	200.3	202.6	245.2	248.3	156.7	162.8	210.5	422.6		310	326	110.2	112.2	153	154.2	218		220.6	226.7	301.4	318.4	442.8	445.7	
AB207	114 - 2009-08-24-01-A	200.5	202.5	226.2	245.3			210.8	422.6	424.5	316.5	325.9		118	154.2		218.1		233.1	237.3	305.7	314.2	422.1	430.8	
AB208	114 - 2009-08-24-01-A	215.9		226.3	245.4	151.9		213.5	422.6		326.1	335.5	114.3	116.2	146		218.1	220.2	227		293.7		410.5	445.7	
AB209	114 - 2009-08-24-01-A	213.9	216.3	226.4	235.9			210.8	214.4	422.6		326		112.3	118.1	146		220.2	222.3	226.8	235.2	307.9	320.7	433.8	462.8
AB210	114 - 2009-08-24-01-A	202.9	214	226.4	257.4	162.9		210.6	214.6	422.5	424.5	316.8	329.3	114.2	116.2	146.1	153.3	222.5	228.8	222.8	227.2	293.7		427.9	436.9
AB211	114 - 2009-08-24-01-A	200.3	202.4	226.1	259.8	151.9		210.5	422.6		316.1		112.2	114.1	145.9	160.2	211.7	215.8	233	237.2		307.7		434	
AB212	114 - 2009-08-24-01-A	202.6	213.7	226	256.9	162.9		207.6	210.4	422.6		315.8		112.2	116.1	145.9	166	220		226.8	228.8	293.5		436.9	439.8
AB213	114 - 2009-08-24-01-A	202.6	216	226	245	156.7		210.4	216.9	422.6		316.2	325.6	114	119.9	137.4	139.4	211.6	219.9	228.7	237.1	303.3		428	
AB214	114 - 2009-08-24-01-A	202.5	213.7	245.1	256.9	156.7	162.8	207.4	214.1	420.6	422.5	316.2	325.7	104.4	114.1	145.8	154.1	219.9		226.5		293.5		422.2	427.9
AB215	114 - 2009-08-24-01-A	211.5		245.2	257	162.9		210.5	422.7	424.6	325.9		114.2	120	145.8		220		226.5		299.4	305.8	439.8		
AB216	114 - 2009-08-24-01-A	202.6	213.6	245.1	257	156.7	162.8	207.4	214.2	424.6		322.7	325.9	112.2	118	156.2	158.2	215.9	220	226.5		314.1		430.9	448.7
AB217	114 - 2009-08-24-01-A	202.7	213.7	245.2	257	147	162.9	210.7	214.2	422.5		322.7	325.9	104.5	114.1	145.9	153.1	220.1	230.4	226.5		293.5		439.9	445.7
AB218	114 - 2009-08-24-01-A	202.8	213.8	226.3	257.2	162.9		207.6	210.9	422.6		316.5		112.3	116.1	146	166.2	220.2		226.8	228.9	293.6		436.9	439.8
AB219	114 - 2009-08-24-01-A	202.6	216	226	245.1	156.8		213.7	422.4		316.3	325.7	104.4	110.2	137.4	154.2	220	224.1	226.8		293.5		430.9	451.4	
AB220	114 - 2009-08-24-01-A	202.6	213.6	225.8		162.8		210.5	422.5		325.7	328.9	114.1		145.9	160.2	217.9	222	226.9		293.5		433.9	445.7	
AB221	114 - 2009-08-24-01-A	213.5	216	226	245	156.8			422.6		316.2	325.7	104.4	110.3	137.5	154.2	224.1		226.9		305.5	318.3	428	430.9	

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12														
AB222	114 - 2009-08-24-01-A	202.7	245.1	259.7	157.6	162.9	210.4	213.8	422.5	325.5	115.9	123.9	145.8	158.2	220	240.6	228.8	239.3	307.8	314.1	410.5	439.8					
AB223	114 - 2009-08-24-01-A	202.7	226.1	259.8			210.7		410.8	325.9	114.1	122	145.8	153	215.8	220	226.7		309.8	314.3	410.5	422.3					
AB224	114 - 2009-08-24-01-A	213.7	226.1		156.8	162.8	210.6		422.6	325.8	112.2	118	137.5	162.2	207.6	218	222.4	228.8	297.5	312.1	425	428					
AB225	114 - 2009-08-24-01-A	220	245.5	248.6	152		210.6	214	422.6	424.6	316.6	326.1	110.2	153.2	156.4	218.1	220.2	228.9	237.4	314.3		428	433.9				
AB226	114 - 2009-08-24-01-A	209.5	215.9	226.3		156.7		195.3	214.2	422.5	316.7	326.2	114.2	116.1	145.9	160.4	220.2		220.6	226.9	293.7	439.8	445.7				
AB227	114 - 2009-08-24-01-A	213.5	219.8	226	256.9	167		210.5	213.8	422.6	424.6		313	325.7	104.3	115.9		145.8	158.2	215.8	222	226.6	318.2	410.5	428		
AB228	114 - 2009-08-24-01-A	200.3	202.4	225.9	244.9	167		210.4		422.6	424.6		316.1	325.6	117.9			154.1		217.8	232.9	237.1	305.5	313.9	422.1	431	
AB229	114 - 2009-08-24-01-A	202.7			151.9			213.5					116.2					154.2		211.6		226.7					
AB229	114 - 2009-08-26-01-A	202.9	216.5	226.5	257.5			214.3		422.6		317	326.4	116.3				154.4		212.1	220.5	226.9		301.7	318.8	410.3	427.8
AB230	114 - 2009-08-24-01-A	202.6		226	256.9	156.8		210.4	213.7	410.7	422.7		322.6	325.8	110.2	112.1		145.8	156.2	217.8	219.9	226.6	228.7	291.6		410.4	439.9
AB231	114 - 2009-08-24-01-A	202.5	213.7					213.9		422.6			322.7	325.8	110.3	112.2		145.9	154.3	220	222.1	226.6	229.8	303.5	312.2		
AB232	114 - 2009-08-24-01-A	200.4	202.7	226.1		156.8	162.9	210.6	213.8	420.7	424.6		325.9		114.2	116.1		153.1	160.3	211.8	215.9	220.7	235.2	305.7	322.9	434	439.8
AB233	114 - 2009-08-24-01-A	213.1	217.4	225.4	244.9			213.5	219.6	422.4			313.1	325.8	125.6			163.2	169.6	215.4	217.4	228		307.8	312	428	439.8
AB234	114 - 2009-08-24-01-A	202.7	216.2	226.2	257.1	151.9		213.9		422.5			316.4	326	116.2			154.3		211.9	220.3	227		301.4	318.4	410.3	427.9
AB235	114 - 2009-08-24-01-A	213.8	216	226		157.6	162.9	210.5		422.6			325.7	328.9	112.2	118		145.9	162.2	217.9		226.7	228.8	312	320.4	425	462.9
AB236	114 - 2009-08-24-01-A	213.7	216	226		156.8	162.9	210.2	214.2	410.7	424.6		325.3		110.2	118		137.4	145.8	211.6	215.8	226.8		293.5		430.9	439.9
AB237	114 - 2009-08-24-01-A	200	202.5	244.9	248	156.8	162.8	210.4		422.5			309.6	325.6	110.1	112.1		153	154.1	217.8		220.6	226.5	301.2	318.1	442.8	445.7
AB238	114 - 2009-08-24-01-A	202.6	213.5	225.8		162.9		210.6		422.4			325.5	328.8	114			145.8	160.2	217.9	222	226.8		293.4		433.9	445.8
AB239	114 - 2009-08-24-01-A	202.7	213.8	226.6	245.2	151.9		213.8		422.6	424.6		316.4	325.8	104.5	116.2		137.5	154.3	211.8	220.1	226.8		293.6		410.4	428.1
AB240	114 - 2009-08-24-01-A	202.7	213.7	226.2	257.1	162.9		210.5	214.3	422.6	424.6		316.4	329	114.1	116.1		145.9	153.1	222.2	228.5	222.7	226.7	293.6	312.1	428	437
AB241	114 - 2009-08-24-01-A	200.2	202.4	226.1	259.9	151.8		210.6		422.5			316.4		112.2	114.1		145.9	160.4	211.7	215.9	232.9	237.2	307.8		433.9	
AB242	114 - 2009-08-24-01-A	202.8		245.5	260.1			210.8	214.1	422.6			326		116	124		146	158.4	220.3	240.9	228.9	239.5	308.1	314.3	410.4	439.8
AB243	114 - 2009-08-24-01-A	202.4		225.9	256.8	156.8	162.8	210.1	213.4	424.5			325.5		112.1			145.8	153	221.9		220.8	228.7	301.3	318.1	439.8	
AB244	114 - 2009-08-24-01-A	202.5	213.6	244.9	256.7	156.8	162.9	207.4	214.2	420.5	422.5		316	325.5	104.2	114.1		145.8	154.2	219.8		226.4		293.4		422.2	428
AB245	114 - 2009-08-24-01-A	200.3	202.4	225.9	244.9	157.5	167	210.2		422.5	424.5		316	325.5	117.9			154		217.8		232.9	237.1	305.4	313.9	422	430.9
AB246	114 - 2009-08-24-01-A	202.5	213.5	225.9	256.8	162.8		207.5	210.3	422.5			316.1		112.1	115.9		145.8	165.9	219.8		226.6	228.6	293.5		436.8	439.7
AB247	114 - 2009-08-24-01-A	200.4		226	248.3	156.8	162.9	210.6	213.8	410.7	422.6		322.6	325.8	114.1			145.8		218	220	226.6	239.3	316.2	318.3	439.8	
AB248	114 - 2009-08-24-01-A	213.7	216.1	226.2	235.6	156.7	162.9	210.5	214.4	422.3			325.3		112.4	118.3		145.7		220	222.1	226.9	235.1	307.6	320.3	433.9	462.9
AB249	114 - 2009-08-24-01-A	202.7	213.7	226.1	257	162.9		210.6	214.2	422.5	424.5		316.4	329	114.1	116		145.8	153	222.1	228.4	222.5	226.7	293.5		427.9	437
AB250	114 - 2009-08-24-01-A	211.7	216.1	226.3	245.3	156.8		213.9		422.5			316.4	326	104.4	110.3		137.5	154.2	224.2		226.8		305.6	318.4	428	430.9
AB251	114 - 2009-08-24-01-A	202.7	211.6	226	256.8	156.8	162.9	213.7		422.5			316.2	325.7	110.3	114.1		153	154.2	220		226.7	237.2	318.3		421.9	451.4

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB252	114 - 2009-08-24-01-A	202.5	213.7	225.9	156.9	163		422.5	322.4	325.6	110.3	112.1	145.8	154.2	219.9	222	226.9	229.8	303.5	311.9	428	433.9			
AB253	114 - 2009-08-24-01-A	213.5	216.1	225.7	151.8		210.2	213.6	420.5	422.5	325.1	113.9		145.7	165.9	215.7	219.8	226.7	239.2	311.7	318.1	410.4	436.8		
AB254	114 - 2009-08-24-01-A	211.5	217.9	226.4			210.4	214	422.4		325.7	328.8	110.3	114.2	150.1	160.3	220		226.8		293.5		422.4		
AB255	114 - 2009-08-24-01-A	216.1			156.8	162.9	213.8		422.6		316.4	325.8	114.2	116.3	153	154.2	220		226.7		293.5		428	439.8	
AB256	114 - 2009-08-24-01-A	213.9	216.1	226.1	157.7	162.9	210.7		422.6		325.9	329	112.2	118	145.9	162.2	218		226.7	228.8	312	320.4	425.1	463	
AB257	114 - 2009-08-24-01-A	202.6	213.7	226.1	257	162.8		207.3	210.6	422.5	316.1	112.2	116	145.9	166.1	220		226.6	228.8	293.6			436.9	439.8	
AB258	114 - 2009-08-24-01-A	200.6	202.8	226.3	257.2		210.8	214.1	422.6		325.8		114.2	118.2	146		222.3	228.5	226.8	228.9			434	439.9	
AB259	114 - 2009-08-24-01-A	202.4		225.9	256.7	156.7	162.8	210.4	213.6	424.4	325.1	112		145.8	152.9	221.9		220.9	228.6	301.2	318.1		439.8		
AB260	114 - 2009-08-24-01-A	202.5	211.5	225.5			213.6		422.7	424.6	315.9	325.5	110.2	118	145.8	154.2	219.9				293.5		428.1	439.9	
AB262	114 - 2009-08-24-01-A	211.3		245	256.8	162.9		209.9		422.5	424.5	325.2	114	119.9	145.6		219.9		226.5		299.2	305.6		439.8	
AB263	114 - 2009-08-24-01-A	213.7	216.1	226.1	157.7	162.9	210.5		422.1		325.5	328.8	112.3	118	145.8	162.2	217.8		226.8	228.7	311.9	320.3	424.9	462.8	
AB264	114 - 2009-08-24-01-A	209.3	215.7	226.1	156.9		195.1	214.1	422.2		316.4	325.8	114.1	116.1	145.9	160.3	220		220.6	226.7	293.5		439.9	445.8	
AB265	114 - 2009-08-24-01-A	199.9	202.1	225.9	259.8	151.8		210.3		422.3	315.8		112.1	114	145.8	160.2	211.5	215.7	232.7	237.1	307.6			433.8	
AB266	114 - 2009-08-24-01-A	202.6	216.1	226.1	245.2	156.8		213.8		422.5	316.5	325.9	104.4	110.3	137.5	154.2	220.1	224.2	226.9		293.4		430.9	451.4	
AB267	114 - 2009-08-24-01-A	202.6		226	256.8	156.9	162.9	210.4	213.7	424.6	325.5		112.2		145.9	153.1	222		220.8	228.7	301.3	318.2		439.8	
AB268	114 - 2009-08-24-01-A	213.6		225.9	156.8	162.9			422.4		325.1		112.2	118	137.5	162.2	207.5	217.8	222.2	228.7	297.4	311.9	425	428	
AB269	114 - 2009-08-24-01-A	211.5	217.9	225.8			210.5	213.6	422.2		325.5	328.7	110.3	114.1	150	160.2	219.9		226.9		293.5		422.4		
AB270	114 - 2009-08-24-01-A	202.7	216.1	226.1	256.8	151.9		213.7		422.3	316.2	325.6	116.2		154.2		211.7	220.1	226.9		301.3	318.2	410.5	428.1	
AB271	114 - 2009-08-24-01-A	213.7		226.1	256.9	156.8	163.1	213.7		424.5	322.5	325.7	110.3	118.1	145.9		220	222.1	220.7	226.7	318.2	324.5	410.5	428	
AB272	114 - 2009-08-24-01-A	211.5	218.1	226.1	245.2	156.9	163	213.7		422.4	313	325.8	116.1		154.3	162.4	215.9	218	220.8	228.8	307.8	312	428	439.9	
AB273	114 - 2009-08-24-01-A	213.8		226.2	156.7	162.8	210.7		422.4		325.7		112.2	118	137.5	162.2	207.7	218	222.5	228.8	297.4	312	425	428	
AB274	114 - 2009-08-26-01-A	200.4	213.7	226.2	235.8	167		210.8	214	422.5	326.1		112.2	116	153.1	158.3	216	220.1	230.9	239.4	303.6		427.8	433.7	
AB275	114 - 2009-08-24-01-A	211.5		244.9	256.8	162.9		210.4		422.6	424.6	325.5		114.2	120.1	145.6		219.9		226.5		299.2	305.6	439.8	
AB276	114 - 2009-08-24-01-A	202.6	213.5	225.9	256.7	163		210.4	214	422.7	424.7	316.1	328.7	114	116	145.8	153	221.9	228.1	222.4	226.6	293.4		428.1	437.2
AB277	114 - 2009-08-24-01-A	202.6	215.7	226	244.9	151.9		210.4		422.6	424.6	325.3		114.1	121.9	145.8	153	219.9		226.8		293.5	311.8	439.8	
AB278	114 - 2009-08-24-01-A	213.7	216	226	156.9	162.9	210.4	213.4	410.7	424.6	325.5		110.3	118.1	137.5	145.9	211.7	215.8	226.8		293.5		430.9	439.9	
AB279	114 - 2009-08-24-01-A	202.6		226.1	156.8	162.8	213.6		422.7	424.6	316.2	325.7			145.8	154.1	220		226.8		293.5		428.1	439.9	
AB280	114 - 2009-08-24-01-A	202.6	213.5	226	162.9		210.5		422.6		325.7	328.8	114.1		145.9	160.3	217.9	222	226.9		293.6		434	445.8	
AB281	114 - 2009-08-24-01-A	211.7	216.1	226.2	245.2	156.9		213.9		422.6	316.3	325.8	104.4	110.3	137.5	154.3	224.2		226.8		305.6	318.3	428	430.9	
AB282	114 - 2009-08-24-01-A	202.6	213.7	226.2	156.8	162.9	214.3		422.5		322.7	325.9	110.3	112.2	145.9	154.2	220.1	222.1	226.7	229.8	303.6	312	428	433.9	
AB283	114 - 2009-08-24-01-A	202.6	216	225.9	244.9	156.8		213.6		422.6	316.2	325.6	104.3	110.2	137.4	154.1	220	224	226.6		293.5		431	451.5	

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB284	114 - 2009-08-24-01-A	213.6	217.9	226	244.9	156.9	162.9	214.1	422.7		313	325.6	116	154.2	162.2	215.8	217.8	220.6	228.7	307.8	312	428.1	439.9		
AB285	114 - 2009-08-24-01-A	202.6		226	256.7	156.9		210.5	213.7	410.7	422.7	322.4	325.6	110.3	112.2	145.8	156.2	217.8	219.9	226.6	228.7	291.6	410.5	439.9	
AB286	114 - 2009-08-24-01-A	202.6		244.9	259.5			210.4	213.7	422.6		325.4		115.8	123.8	145.7	158.1	219.9	240.4	228.7	239.2	307.7	313.9	410.5	439.9
AB287	114 - 2009-08-24-01-A	202.6	211.5	226	256.8	156.7	163.1	213.7		422.6		316.2	325.7	110.2	114	153	154.1	220		226.6	237.1			422.2	451.4
AB288	114 - 2009-08-24-01-A	202.7	213.7					210.6	214.2	422.5		322.5	325.8	104.5	114.1	145.9	153.1	220	230.3	226.6		293.5		439.9	445.8
AB289	114 - 2009-08-24-01-A	213.8	216.2	226.2	235.8	156.8	162.9	210.5	214.4	422.5		325.8		112.3	118.1	145.9		220.1	222.2	226.8	235.2	307.7	320.5	433.9	462.9
AB290	114 - 2009-08-24-01-A	200.5		226.2	248.4			210.7		410.7	422.6	322.8	326	114.1		145.9		218.1	220.1	226.7	239.4	316.3	318.4	439.8	
AB291	114 - 2009-08-24-01-A	202.7	216	226	256.8	151.9		213.4		422.7		316.2	325.6	116.1		154.2		211.7	220	226.7		301.3	318.1	410.6	428.1
AB292	114 - 2009-08-24-01-A	202.6	213.6	225.9	256.7	162.9		207.3	210.5	422.7		316		112.1	116	145.8	165.9	219.9		226.6	228.7	293.4		437	439.9
AB293	114 - 2009-08-24-01-A	200.3	202.4	226	259.6	151.9		210.5		422.6		316		112.2	114	145.8	160.2	211.7	215.8	233	237.2	307.6		434	
AB294	114 - 2009-08-24-01-A	202.6	216	226	256.8	151.9		213.6		422.7		316.2	325.6	116.1		154.2		211.7	220	226.6		301.3	318.1	410.6	428.1
AB295	114 - 2009-08-24-01-A	200.5	202.5	226.1	245.1	167.1		210.5		422.6	424.6	316.3	325.7	118.1		154.2		217.9		233	237.2	305.5	314	422.1	431
AB296	114 - 2009-08-24-01-A	202.6	215.7	225.9	244.9	151.8		210.4		422.5	424.5	325.5		114.1	121.9	145.8	152.9	219.9		226.7		293.4	311.9	439.8	
AB297	114 - 2009-08-24-01-A	202.6	213.7	226		156.9	163	214.3		422.6		322.6	325.9	110.3	112.2	145.9	154.2	220	222.1	226.8	229.8	303.5	311.9	428	434
AB298	114 - 2009-08-24-01-A	202.7		226.2	257			210.7	213.9	424.5		325.8		112.3		145.9	153.1	222.2		220.7	228.8	301.4	318.3	439.7	
AB299	114 - 2009-08-24-01-A	211.6	216	226	244.9	156.8		213.7		422.6		316.1	325.6	104.4	110.3	137.4	154.1	224		226.6		305.5	318.2	428.1	431
AB300	114 - 2009-08-24-01-A	213.6	215.9	225.8		157.6	162.9	210.4		422.7		325.6	328.7	112.1	117.9	145.7	162.1	217.7		226.6	228.6	311.9	320.3	425.1	462.9
AB301	114 - 2009-08-24-01-A	200.3	202.4	225.9	259.6	151.9		210.5		422.7		316.1		112.1	114	145.8	160.1	211.6	215.8	232.9	237.1	307.4		434	
AB302	114 - 2009-08-24-01-A	202.7	213.6	245	256.8	162.9		210.5	214.1	422.5		322.4	325.6	104.4	114.1	145.8	153	220	230.2	226.7		293.5		439.9	445.8
AB303	114 - 2009-08-24-01-A	213.7		226		156.8	162.9	210.6		422.7		325.7		112.1	118	137.4	162.1	207.6	217.9	222.4	228.7	297.4	311.9	425.1	428.1
AB304	114 - 2009-08-24-01-A	202.7	213.7	226.1	257	162.9		210.5	214.1	422.6	424.6	316.3	329	114.1	116.1	145.9	153	222.1	228.4	222.5	226.7	293.5		428	437.1
AB305	114 - 2009-08-24-01-A	202.6	213.8	245.2	257	156.8	162.8	207.6	214.3	420.6	422.6	316.3	325.6	104.4	114.1	145.9	154.2	220		226.6		293.5		422.2	428
AB306	114 - 2009-08-24-01-A	202.7		226.4	257.2	156.8		210.6	213.9	410.6	422.6	322.8	326	110.4	112.3	146	156.3	218.1	220.2	226.9	228.9	291.7		410.1	439.9
AB307	114 - 2009-08-24-01-A	200.4	213.4	225.9	235.4	167.1		210.2	214	422.3		325.4		112.2	116	153	158.2	215.7	219.8	230.8	239.2	303.3		428	433.9
AB308	114 - 2009-08-24-01-A	200.3	204.7	225.9	244.9	156.8	162.9	210.3	213.9	422.7		322.3	328.6	114.1	118	153	158.1	217.8	221.9	226.7		318		431	439.9
AB309	114 - 2009-08-24-01-A	219.6		245	248	151.7		210.2	213.5	422.6	424.6	316	325.5	110.1		152.9	156.1	217.8	219.8	228.7	237.1	313.8		428	434
AB310	114 - 2009-08-24-01-A	202.6	216	226	245	156.8		213.7		422.7		316.3	325.7	104.4	110.3	137.5	154.2	220	224.1	226.6		293.5		431	451.5
AB311	114 - 2009-08-24-01-A	202.7		226.1		158.1	162.9	210.6		422.6		325.6		116.1	120	156.2	158.2	211.7	220	220.8	226.7	293.5		428.1	445.8
AB312	114 - 2009-08-24-01-A	211.5		245	256.9	162.9		210.5		422.6	424.6	325.7		114.1	120	145.8		219.9		226.6		299.3	305.6	439.8	
AB313	114 - 2009-08-24-01-A	202.6	213.5	226.4	245.1	156.8	162.9	214.2		410.7	424.6	325.8	329	112.2	114.1	145.8		222.1		226.8		293.5		422.1	428
AB314	114 - 2009-08-24-01-A	202.6		226.2	257.1	156.8		210.7	213.9	410.7	422.6	322.7	325.9	110.3	112.3	145.9	156.3	218	220.1	226.7	228.8	291.7		410.4	439.9

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB315	114 - 2009-08-24-01-A	213.6	218	226	244.9	156.8	162.9	214.2	422.5		312.8	325.6	115.9	154.1	162.2	215.8	217.8	220.8	228.7	307.7	311.9	428.1	439.9		
AB316	114 - 2009-08-24-01-A	202.6	213.7	245	256.8	156.8	162.9	207.4	214.1	420.7	422.6	316.1	325.6	104.4	114.1	145.8	154.1	219.9		226.6		293.5	422.3	428	
AB317	114 - 2009-08-24-01-A	202.6	211.4	225.9	256.7	156.8	162.9	213.5		422.6		316.1	325.5	110.2	114	152.9	154	219.9		226.6	237.1	318.1	422.4	451.5	
AB318	114 - 2009-08-24-01-A	202.6	216	226	245	156.8		213.6		422.6		316.2	325.7					220	224	226.6		293.4	431	451.5	
AB319	114 - 2009-08-24-01-A	202.6		245	256.9	162.9		210.5		422.6		313	325.7	114.1	116.1	145.9	154.2	219.9	228.2	226.6		293.5	422.4	448.8	
AB320	114 - 2009-08-24-01-A	200.4	213.5			142.3	167.1	210.2	214	422.5		325.6				153	158.2	215.8	220	230.8	239.3	303.4	428	433.9	
AB321	114 - 2009-08-24-01-A	211.6	218	226.2				210.7	213.9	422.6		325.9	329	110.3	114.1	150.1	160.3	220		226.7		293.6	422.2		
AB322	114 - 2009-08-24-01-A	202.7		245.3	259.9			210.5	214	422.6		326		116	123.9	145.9	158.3	220.1	240.8	228.8	239.4	307.9	314.2	410.4	439.8
AB323	114 - 2009-08-24-01-A	200.3		225.9	248.1	156.7	162.8	210.4	213.7	410.7	422.6	322.4	325.6	113.9		145.7		217.9	219.8	226.7	239.3	316	318.2	439.9	
AB324	114 - 2009-08-24-01-A	202.5		225.7	244.9	162.9		213.5		422.5		328.6	341	112.2	116.1	154.1	158.1	219.8	221.8	226.8		293.4	428	439.8	
AB325	114 - 2009-08-24-01-A	202.7	213.5	244.9	256.8	156.8	162.9	210.4		422.6	424.6	325.4		114.1	123.9	153		205.5	219.9	226.6		297.4	410.5	428	
AB326	114 - 2009-08-24-01-A	202.4		225.9				213.6		422.5		322.4	325.6	110.1	112	145.8	154.1	219.8	221.9	226.5	229.7		427.9	433.9	
AB327	114 - 2009-08-24-01-A	202.7						210.7	213.9	422.6		325.7									228.8				
AB328	114 - 2009-08-24-01-A	211.4	217.9	226.1	245.1	156.8	162.8	213.6		422.5		313	325.8	116		154.1	162.2	215.8	217.9	220.6	228.7	307.7	311.9	428	439.8
AB329	114 - 2009-08-24-01-A	202.6	216.1	226.2	256.9	151.9		213.7		422.2		316.3	325.7	116.1		154.2		211.7	220.1	226.8		301.3	318.2	410.4	428
AB330	114 - 2009-08-24-01-A	213.6	219.4	229		156.8		200	218	422.5		316.5	326	114.2	116.2	146	160.4	223.4		223.8	229.4	293.5	439.8	445.7	
AB331	114 - 2009-08-24-01-A	213.6	216	225.7		156.8	162.9	210.3	213.2	410.7	424.5	325.5		110.3	118	137.4	145.8	211.6	215.8	226.8		293.4	430.9	439.8	
AB332	114 - 2009-08-24-01-A	202.6	216	226	244.9	156.7		213.6		422.5		316.1	325.6	104.3	110.2	137.4	154.1	220	224	226.8		293.4	431	451.5	
AB333	114 - 2009-08-24-01-A	213.5	217.9	225.9	244.9	156.9	162.9	214		422.7		312.9	325.6	116		154.2	162.2	215.8	217.8	220.6	228.7	307.7	311.9	428.2	440
AB334	114 - 2009-08-24-01-A	202.6	213.5			162.9		210.2		422.6		325.6	328.8	114		145.8	160.2	217.9	221.9	226.6		293.4	433.9	445.8	
AB335	114 - 2009-08-24-01-A	202.6	213.7	226	245	151.8		213.7		422.6	424.6	316.3	325.7			137.4	154.2	211.7	220	226.6		293.5	410.5	428.1	
AB336	114 - 2009-08-24-01-A	200.3	202.5	226.1	259.8	151.9		210.5		422.6		316.3		112.2	114.1	145.9	160.3	211.8	215.9	233	237.2	307.7		434	
AB337	114 - 2009-08-24-01-A	213.8	216.1	226.2		151.8		210.7	214.2	420.6	422.6	325.9		114.1		145.9	166	215.9	220.1	226.7	239.4	312	318.4	410.4	436.9
AB338	114 - 2009-08-24-01-A	202.6		226.3	257.1			210.6	213.7	424.5		325.7		112.2		145.9	153.1	222.2		220.6	228.8	301.4	318.4	439.8	
AB339	114 - 2009-08-24-01-A	213.5	217.9	225.8	244.8	156.8	162.9			422.5		312.8	325.6	116		154.1	162.2	215.7	217.8	220.7	228.7	307.7	311.9	428	439.8
AB340	114 - 2009-08-24-01-A	213.6	216	225.9	235.4	156.9	162.9	210.2	214.1	422.5		325.4		112.2	118	145.8		219.9	221.9	226.6	235	307.5	320.2	434	462.9
AB341	114 - 2009-08-24-01-A	202.6		225.9	259.5	156.8	162.8	210.5		410.7		325.5		114.1	121.9	145.8	152.9	215.7	219.8	226.7		309.6	313.9	410.6	422.3
AB342	114 - 2009-08-24-01-A	211.5		245.1	256.9	162.9		210.4		422.6	424.6	325.5		113.4	120	145.8		220		226.7		299.3	305.6	439.8	
AB343	114 - 2009-08-24-01-A	209.3	215.7	226.1		156.7		195.2	214	422.6		316.3	325.8	114	116	145.8	160.2	220		220.8	226.6	293.5	439.9	445.8	
AB344	114 - 2009-08-24-01-A	211.6	218	226.1		156.9	162.9	210.6	213.9	422.7		325.7	328.9	110.2	114.2	150.1	160.3	220		226.7		293.5	422.3		
AB345	114 - 2009-08-24-01-A	213.5	218	226.2	245.2	156.9	162.9	214.3		422.5		313.1	325.9	116.1		154.3	162.3	215.9	218	220.8	228.8	307.8	312	428	439.8

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB346	114 - 2009-08-24-01-A	200.4	213.8	226.3	156.8	162.9	210.7	217.1	410.7	422.6	322.8	326	110.4	124	160.3	164.2	218.1	220.1	222.5	226.7	318.4	433.9	445.7		
AB347	114 - 2009-08-24-01-A	213.5	216	226	245	156.9			422.4		316.1	325.6	104.4	110.3	137.5	154.2	224		226.8		305.5	318.2	428.1	430.9	
AB348	114 - 2009-08-24-01-A	209.3	215.6	225.9	146.8	156.8	194.9	214.1	422.6		316.1	325.6			145.8	160.2	219.9		220.8	226.8	293.5		439.9	445.8	
AB349	114 - 2009-08-24-01-A	202.7	213.6	226	256.7	163		210.6	214	422.7	424.7	316.1	328.7	114.1	116.1	145.8	153	222	228.2	222.4	226.6	293.4	428.1	437.1	
AB350	114 - 2009-08-24-01-A	213.8	216	226		157.6	162.9	210.6		422.6		325.7	328.8	112.1	117.9	145.8	162.1	217.9		226.6	228.7	311.9	320.4	425.1	462.9
AB351	114 - 2009-08-24-01-A	200.4	213.7	226.1				210.7	217.1	410.8	422.7	322.6	325.8			160.2	164.1	218	220	222.5	226.7	318.5		434	445.8
AB352	114 - 2009-08-24-01-A	200.4	202.7	210.6	226.1	156.8		210.6		420.6	422.5	316.4	325.8	114.1	123.9	137.5	145.9	220	226.2	228.8			428	439.8	
AB353	114 - 2009-08-24-01-A	202.7		245.2	259.8			210.6	213.9	422.6		325.9		115.9	123.9	145.9	158.3	220.1	240.7	228.8	239.3		410.5	439.9	
AB354	114 - 2009-08-24-01-A	213.9	216.2	226.2		156.7	162.8	210.7	214.3	410.7	424.6		326			137.5	145.9	211.9	216	226.8		293.6	430.9	439.8	
AB355	114 - 2009-08-24-01-A	200.1	202.5	225.8	256.8	156.9	162.9	210.4		422.6		303.1	325.6	116	123.9	145.7		211.6	220	226.9		291.6	428	451.6	
AB356	114 - 2009-08-24-01-A	200.2	202.3	225.9	244.8	167.1		210.3		422.6	424.6	316	325.4	118.1		154.2		217.7		232.8	237	305.5	314	422.1	431
AB357	114 - 2009-08-24-01-A	213.7		226		156.8	162.9	210.2		422.7		325.6		112.2	118	137.4	162.1	207.6	217.9	222.5	228.8	297.4	311.9	425.1	428.1
AB358	114 - 2009-08-24-01-A	202.6	213.5	226	256.8	162.9		210.2	214.1	422.6	424.6	316.1	328.8	114	116	145.8	153	221.9	228.2	222.4	226.6	293.4	428	437.1	
AB359	114 - 2009-08-24-01-A	202.8	213.6	226.2		162.9		210.6		422.6		325.8	328.9	114.1		145.9	160.3	218	222.1	226.9		293.4	434	445.8	
AB360	114 - 2009-08-24-01-A	202.7	216.1	226.1	256.9	151.9		213.7		422.5		316.2	325.7	116.2		154.2		211.8	220.2	226.8		301.3	318.2	410.5	428
AB361	114 - 2009-08-24-01-A	211.6		245.2	257	162.9		210.6		422.6	424.6	325.8		114.2	120	145.9		220.1		226.6		299.4	305.8	439.8	
AB362	114 - 2009-08-24-01-A	213.7	216.2	226.1	245.3	156.8		214.5		422.5		316.4	326	104.5	110.3	137.5	154.3	224.3		227		305.6	318.4	428	430.9
AB363	114 - 2009-08-24-01-A	209.3	215.6	225.6		156.8		195.4	214.1	422.4		316.2	325.6	114.1	116.1	145.8	160.2	219.9		220.6	226.8	293.5		439.9	445.8
AB364	114 - 2009-08-24-01-A	213.5	215.9	225.7	235.3	156.8	162.8	210.3	213.9	422.5		325.1		112.1	118	145.8		219.8	221.9	226.8	234.7	307.5	320.2	433.9	462.9
AB365	114 - 2009-08-24-01-A	202.6	213.7	226	244.9			213.7		422.7	424.7	316.2	325.6							226.6		293.4			
AB365	114 - 2009-08-26-01-A	202.7	213.9	226.3	245.5	151.8		214.1		422.6	424.6	316.8	326.2	104.5	116.3	137.5	154.2	211.9	220.3	226.7		293.7	410.3	427.8	
AB366	114 - 2009-08-24-01-A	200.3	202.6	225.7		156.9	162.9	210.5	213.5	422.6		312.8	325.5	104.4	114.1	137.5	145.9	211.6	217.8	226.9		293.5	431	439.9	
AB367	114 - 2009-08-24-01-A	202.7		226.1		156.8	162.9	213.8		422.6	424.6	316.2	325.7			145.9	154.2	220.1		226.7		293.5	428.1	439.9	
AB368	114 - 2009-08-24-01-A	202.6	211.6	226.1	256.9	156.8	162.9	213.8		422.6		316.3	325.8	110.3	114.1	153.1	154.3	220		226.6	237.2	318.3		422.3	451.4
AB369	114 - 2009-08-24-01-A	200.5	202.5	226.2	245.2			210.6		422.6	424.6	316.4	325.9	118		154.2		218		233	237.2			422.2	430.9
AB370	114 - 2009-08-24-01-A	202.8	213.8	226.3	257.1	162.8		210.7	214.5	422.6	424.6	316.5	329.1	114.1	116.2	146	153.2	222.3	228.6	222.6	227.1	293.6	312.1	427.9	437
AB371	114 - 2009-08-24-01-A	213.7	216	225.6				210.4		422.5		325.5	328.7	112.1	117.9	145.8	162.1	217.8		226.8	228.7	311.8	320.2	425	463
AB372	114 - 2009-08-24-01-A	213.5	217.9	225.9	244.9	156.8	162.9	214.2		422.7		312.8	325.5	116		154.1	162.2	215.8	217.8	220.9	228.7	307.7	311.9	428.1	439.9
AB373	114 - 2009-08-24-01-A	202.5		225.9	256.7	156.8	162.9	210.5	213.6	424.7		325.5		112.2		145.8	153	221.9		220.8	228.7	301.3	318.1	439.9	
AB374	114 - 2009-08-24-01-A	202.7	216.1	226	245	156.8		213.8		422.6		316.2	325.6	104.5	110.3	137.5	154.2	220.1	224.1	226.8		293.5		431	451.5
AB375	114 - 2009-08-24-01-A	213.4	218	226.1	245.1	156.8	162.8			422.6		313	325.7	113.5	115.9	154.1	162.2	215.8	217.9	220.6	228.7	307.7	311.9	428	439.8

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB376	114 - 2009-08-24-01-A	202.6	237.7	256.9	156.8	162.9	210.6	213.5	422.5	424.6	313.1	325.8	112.2	123.9	153.1	158.3	220	226.6	228.8	303.5	320.4	434	439.9		
AB377	114 - 2009-08-24-01-A	202.6	213.5	225.9	163.1		210.6		422.6		325.7	329	114.2		145.9	160.3	218	222.1	227	293.5		433.9	445.7		
AB378	114 - 2009-08-24-01-A	202.8	213.8	226.2	245.3	156.8	163	214.4	410.7	424.5	325.9	329.1	112.4	114.4	145.8		222.2		227	293.6		422.1	428		
AB379	114 - 2009-08-24-01-A	213.7		226	256.8	156.7	163.1	213.7		424.6	322.5	325.7	110.3	118.2	145.6		219.9	222	220.9	226.6	318.3	324.5	410.5	428.1	
AB380	114 - 2009-08-24-01-A	202.5	215.9	225.7	244.9	156.8		213.7	422.5		316	325.6	104.4	110.2	137.5	154.2	220	224	226.9		293.3		431	451.5	
AB381	114 - 2009-08-24-01-A	202.4	213.6	225.8	256.8	162.9		210.4	214.1	422.6	424.6	316.1	328.7	114.1	116	145.8	153	222	228.2	222.5	226.8	293.5	311.8	428	437.1
AB382	114 - 2009-08-24-01-A	202.6		226	259.6	156.8	162.9	210.2		410.6		325.6		114	121.9	145.8	153	215.7	219.9	226.7		309.7	314	410.7	422.3
AB383	114 - 2009-08-24-01-A	202.7	213.7	226.1	245.1	162.9		210.7	213.9	422.6		322.6	325.7	114.2	116.1	145.9	150.1	217.9		226.7	228.8	293.6		437	439.9
AB384	114 - 2009-08-24-01-A	213.7	216.2	226.2		157.7	162.9	210.6		422.4		325.9	329	112.2	118	145.9	162.2	217.9		226.9	228.8	312	320.4	425	463
AB385	114 - 2009-08-24-01-A	200.5	202.7	226.3	257	156.8	162.9	210.7		422.3	303.3	325.8	116	123.9	145.8		211.8	220.2	226.9		291.6		427.9	451.5	
G217	111 - 2009-09-17-01-A	202.6	213.4	244.7	256.6	138.7		209.4		422.7	424.7	325.2		113.2	122.9	153		205.5	219.8	226.5		297.3		410.7	428.2
G218	111 - 2009-09-17-01-A	200.4	202.6	225.9			210.1	212.7			312.9	325.5	103.6	113.2					226.6		293.4		431.1	440	
G218	111 - 2009-09-25-01-A	200.3	202.5	225.9			210.5	213.7	422.6		312.9	325.6	103.4	113.1	137.4	145.7	211.6	217.8	226.5		293.4		430.9	439.8	
G218	111 - 2009-09-29-01-A	200.3	202.5	225.8			210.4	213.6	422.7		312.7	325.3	104.4	114.1	137.5	145.8	211.6	217.7	226.6		293.3		431.2	440.1	
G218	111 - 2009-10-02-01-A	200.3	202.5	225.8			210.4	213.6	422.6		312.8	325.4	103.4	113.1	137.4	145.7	211.5	217.7	226.5		293.4		431	439.9	
G218	111 - 2009-11-04-02-A	200.5	202.6	225.8	156.8	162.9	210.6	213.9	422.6		313.1	325.8	104.5	114.2	137.5	145.9	211.7	217.9	226.9		293.5		430.9	439.8	
G219	111 - 2009-09-17-01-A	202.7	213.5	226	235.4	138.9	167.2	210.4	212.7	422.7	424.7	325.6		115.2	117.1	158.2		215.9	222	226.7	230.8	303.4		428.3	440
G220	111 - 2009-09-17-01-A	209.3	215.6	226.1			194.3	212.8	422.6		316.2	325.7	113.2	115.1	145.9	160.3	220		220.7	226.7	293.5		440	445.9	
G220	111 - 2009-09-25-01-A	209.3	215.6	226			195.2	213.8	422.6		316.2	325.6	113.2	115.1	145.8	160.2	219.9		220.8	226.7	293.5		439.9	445.8	
G220	111 - 2009-09-29-01-A	209	215.3	225.5			194.9	213.3	422.6		315.5	324.9	113	114.9	145.7	160	219.5		220.6	226.4	293.1		440	445.9	
G220	111 - 2009-10-02-01-A	209.1	215.4	225.8			195	213.5	422.6		316	325.4	113	114.9	145.7	160			220.6	226.5	293.3		439.9	445.8	
G220	111 - 2009-11-04-02-A	209.4	215.7	226.1	156.9		195.3	213.9	422.6		316.3	325.8	114.2	116.1	145.9	160.3	220.1		220.8	226.7	293.5		439.9	445.8	
G221	111 - 2009-09-17-01-A	202.5		244.5	259.3	138.7	210.2	213.5	422.6		325.1				145.8	158.1	219.7	240.1	228.6	239.1	307.4	313.6	410.6	440	
G221	111 - 2009-09-25-01-A	202.6	213.5	225.9	244.8	259.5	210.6	213.8	422.6		325.5		103.5	115	123	145.8	158.2	219.9	240.4	226.6		307.6	313.9	410.5	439.9
G221	111 - 2009-09-29-01-A	202.4		244.2	258.9		210.1	213.3	422.7		324.9		114.8	122.7	145.6	157.9	219.4	239.8	228.5	238.8	307.3	313.4	410.8	440.1	
G221	111 - 2009-10-02-01-A	202.5		244.7	259.4	138.7	146.9	210.4	213.6	422.4		325.3		115	122.9	145.7	158.1		228.7	239.1	307.6	313.8	410.4	439.9	
G222	111 - 2009-09-17-01-A	200.3	202.6	225.7	256.4		209.4	212.6	422.7		325.1		113.1	117	145.7		221.7	227.9	226.6	228.6	315.6	317.6	434.2	440.1	
G222	111 - 2009-09-25-01-A	200.4	202.6	226	256.8		210.6	213.8	422.6		325.7		113.3	117.2	145.9		222	228.1	226.6	228.7	316.1	318.3	434.1	440	
G222	111 - 2009-09-29-01-A	202.5	213.2	225.5		138.6	210.2	213.3	422.6		315.6	324.9	97.9	99.9					224.4						
G222	111 - 2009-10-02-01-A	200.4	202.6	225.9	256.6		210.5	213.7	422.6		325.4		113.2	117.1	145.8				226.6	228.7			434	439.9	
G222	111 - 2009-11-04-02-A	200.4	202.7	226.1	257	156.8	162.9	210.7	214	422.6		325.9		114.1	118	145.9		222.1	228.3	226.7	228.8	316.2	318.2	434	440

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
G223	111 - 2009-09-17-01-A	213.3	219.6	225.7	256.5	138.8	210.3	213.6	422.7	424.7	312.6	325.1	103.4	115.1	145.8	158.1	215.6	221.7	226.5	317.8	332.3	410.7	428.3		
G224	111 - 2009-09-17-01-A	202.6	213.5	225.8	256.4	163	209.9	212.6	422.7	424.7	315.8	328.4	113.1	115.1	145.8	152.9		222.4	226.5	293.3		428	436.8		
G224	111 - 2009-09-25-01-A	202.7	213.7	226.2	257		210.7	213.9	422.7	424.6	316.4	328.9	113.2	115.2	145.9	153	222.1	228.4	222.5	226.7	293.6	428.1	437		
G224	111 - 2009-09-29-01-A	202.5	213.3	225.5	256.2	138.8	210.2	213.4	422.8	424.8	315.7	328.2	113.1	115	145.7	152.8	221.6	227.8	222.3	226.5	293.3	428.3	437.3		
G225	111 - 2009-09-17-01-A	202.5	215.8	225.7		138.6	210.4	213.5	422.6	424.6	325.2		113.1	115	145.7	152.9	217.7	221.8	226.6		301.1	305.2	428.1	437	
G226	111 - 2009-09-17-01-A	200.4	215.8	225.9	244.8	138.7	209.5	212.6	422.7		325.1		95.5	113.1	152.9	158.1	215.8	217.9	222.4	226.6	311.6	320	422.4	431.2	
G227	111 - 2009-09-17-01-A	200.3	204.7	225.9	244.8		210.4	212.7	422.7		322.3	328.6	113.1	117	153	158.1	217.8	221.9	226.6		318		431.1	440	
G227	111 - 2009-09-25-01-A	200.3	204.8	226.1	245	138.7	210.7	213.8	422.6		322.5	328.8	113.2	117.1	153	158.2	217.9	222.1	226.7	239.3	318.2		430.9	439.8	
G228	111 - 2009-09-17-01-A	213.7		226	256.7		212.7		424.5		322.4	325.6	109.3	117.1	145.9		219.9	222	220.6	226.6			410.6	428.1	
G228	111 - 2009-09-25-01-A	213.7		226.1	257		213.9		424.6		322.7	325.9	109.5	117.1	145.8		220	222.1	220.6	226.7			410.4	428	
G228	111 - 2009-09-29-01-A	213.4		225.6	256.3		213.1		424.7		321.9	325.1	109.2	117	145.7		219.6	221.7	220.6	226.5			410.7	428.3	
G228	111 - 2009-10-02-01-A	213.6					213.9		424.6		322.3	325.5	109.3	117.1	145.7				220.7	226.6			410.5	428.1	
G228	111 - 2009-11-04-02-A	213.9		226.3	257.1		214.3		424.6		322.9	326.1	110.5	118.2	146		220.2	222.3	220.8	226.8			410.5	428.1	
G229	111 - 2009-09-17-01-A	211.3		244.6	256.5		210.3		422.7	424.7	325.2		111.1		145.7		219.7		226.5		299.1	305.4	440.1		
G229	111 - 2009-09-29-01-A	211.1	213.1				210	213.5	422.6	424.6	312.4	324.9	103.3	113	145.7		215.5	219.6	226.3		299	305.2	410.7	440.1	
G229	111 - 2009-10-02-01-A						210.4	213.8	422.7	424.7	312.9	325.6	103.5		145.8	158.2			226.6						
G229	111 - 2009-11-04-02-A	201.2		244.1	257.6	152.6	199.7		422.6	424.6	326.1		113.1	118.2	138.9		212		220.7		299.5	305.8	439.8		
G230	111 - 2009-09-17-01-A	202.5	213.5	244.6	256.4	138.7	207.2	213.5	420.7	422.7	315.7	325.1	103.3	113	145.7	154	219.7		226.5		293.3		422.4	428.2	
G231	111 - 2009-09-17-01-A	200.3	202.5	225.7		147	185.4	210.3	212.6	420.8	424.8	325.2		113.2	115.1	153	160.1	211.6	215.6	220.7		305.4	322	434.3	440.2
G232	111 - 2009-09-17-01-A	213.5	217.8	225.8	244.7	138.8	212.6		422.7		312.7	325.3	115.1		154.1	162.1	215.7	217.8	220.7	228.7	307.6	311.7	428.3	440.1	
G233	111 - 2009-09-17-01-A	202.6		225.9	256.6		209.5	212.6	424.7		325.4		111.3		145.8	153	220	221.9	220.7	228.7	301.2	318	440		
G233	111 - 2009-09-25-01-A	202.6		226	256.8		210.6	213.8	424.7		325.7		111.3		145.8	153	222		220.6	228.7	301.3	318.2	439.9		
G233	111 - 2009-09-29-01-A	202.6		225.7	256.4		210.4	213.4	424.7		325.2		111.2		145.8	152.9	221.8		220.6	228.6	301.2	318	440		
G233	111 - 2009-10-02-01-A	202.6		226.1	256.9		210.6	213.8	424.6		325.7		111.3		145.8	153			220.6	228.8			439.9		
G233	111 - 2009-11-04-02-A	202.7		226.1	256.9	156.9	163	210.7	213.9	424.7	325.8		112.3		145.9	153.1	220.1	222.1	220.7	228.8	301.4	318.3	439.9		
G234	111 - 2009-09-17-01-A	200.4	202.6	244.8	247.9		209.5		422.5		309.6	325.5			153.1	154.2	217.9		220.7	226.6			442.9	445.8	
G234	111 - 2009-09-25-01-A	200.3	202.6	244.8	248		210.3		422.7		309.7	325.6	109.2	111.2	152.9	154.1	217.8		220.6	226.6	301.3	318.2	442.9	445.8	
G234	111 - 2009-09-29-01-A	200.3	202.5	244.6	247.8		210.3		422.7		309.5	325.4	109.3	111.2	153	154.1	217.8		220.6	226.5	301.2	318	443.1	446	
G234	111 - 2009-10-02-01-A	200.2	202.5				210.9		422.5		309.6	325.6	109.2	111.2	152.9	154.1	217.9		220.6	226.6			442.8	445.7	
G234	111 - 2009-11-04-02-A	200.3	202.7	245	248.1	156.8	162.9	210.6		422.7	309.8	325.7	110.2	112.2	153	154.1	218		220.7	226.7	301.4	318.2	442.9	445.9	
G235	111 - 2009-09-17-01-A	202.7		226	256.6	138.8	212.8		422.6		325.6		113.4	119	145.9	158.1	206.6	220	226.7		313.7	318.1	428.1		

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12															
G236	111 - 2009-09-25-01-A	202.6	213.5	244.9	256.7		210.5	213.8	422.6		322.4	325.6		145.8	152.9	219.8	230.1	226.6		293.4		440	445.8					
G236	111 - 2009-09-29-01-A	202.3	213.1	244	255.9		210	213.2	422.7		321.6	324.7	103.1	112.7	145.6	152.7	219.3	229.5	226.3		293		440.2	446.1				
G236	111 - 2009-10-02-01-A	202.6	213.5	244.8	256.6		210.5	213.7	422.6		322.3	325.5	103.3	113	145.7	152.9			226.6		293.4		440	445.9				
G236	111 - 2009-11-04-02-A	202.6	213.6	244.9	256.8	162.9	210.6	213.9	422.6		322.5	325.7	104.4	114	145.8	153	220	230.2	226.6		293.5		440	445.8				
G237	111 - 2009-09-17-01-A	202.5		225.7			210.3	213.5	422.6		315.8	325.2					211.5	221.7	226.5		293.3							
G237	111 - 2009-09-25-01-A	202.6	213.5	226	245.1	256.9	210.6	214	410.8	422.7	424.7	313.1	325.7	328.9	103.5		145.9		215.8	222	226.7		293.5	428.1				
G237	111 - 2009-09-29-01-A	202.2	213	225.1	243.8		213		410.8	424.7		324.7	327.7		110.9	112.9		145.5		221.2		226.2	293	422.4	428.3			
G237	111 - 2009-11-04-02-A	202.6	213.6	226	245	156.8	162.8	214.1		410.7	424.5		325.8	329		112.2	114.1		145.8		222		226.8	293.5	422.1	428		
G238	111 - 2009-09-17-01-A	202.6	237.2	256.4		138.7		209.4	213.5	422.7	424.7		312.7	325.2		111.2	122.9		152.9	158.1	219.7		226.5	228.6	303.2	320	434.2	440.1
G239	111 - 2009-09-17-01-A	202.4	213.3	225.4				213				315.3	324.7					145.7	154.1	219.5		226.3	293	428.2	440.1			
G239	111 - 2009-09-25-01-A	202.6	213.6	226				214		422.6	424.6		316.3	325.7	109.3	117.1		145.8	154.2	220		226.7	293.5	428	439.8			
G239	111 - 2009-09-29-01-A	202.3	213.2	225.2				212.9		422.7	424.6		315.2	324.6	109	116.8		145.4	153.7	219.3		226.2	292.9	428.3	440.1			
G239	111 - 2009-10-02-01-A	202.6	213.5	225.8				213.5		422.7	424.7		316	325.5	109.3	117		145.7	154			226.6	293.4	428.1	439.9			
G239	111 - 2009-11-04-02-A	202.7	213.8	226.2		156.9	163	214.3		422.7	424.7		316.4	325.9	110.4	118.1		145.9	154.3	220.2		226.9	293.6	428.1	439.9			
G240	111 - 2009-09-17-01-A	202.5		225.7		138.8		213.5		422.7			325.1		103.4	113.1		145.7	152.9	215.7	219.7	226.5	228.6	293.2	440.1	446		
G241	111 - 2009-09-17-01-A	202.5	215.8			151.9		213.5				316	325.4									226.5						
G241	111 - 2009-09-25-01-A	202.6	216	226.1	256.9			213.8		422.6			316.5	325.9	115.2			154.1		211.7		226.8	301.4	318.3	410.5	428.1		
G241	111 - 2009-09-29-01-A	202.4	215.5	225.3	255.9	151.8				422.7			315.6	324.9	115			153.8		211.3	219.4	226.3	301	317.7	410.7	428.3		
G241	111 - 2009-11-04-02-A	202.7	216.1	226.2	257	152		213.8		422.5			316.5	325.9	116.2			154.3		211.9	220.2	226.8	301.4	318.3	410.4	428		
G242	111 - 2009-09-25-01-A	213.9	216.2	210.8	226.2	138.7		210.7		422.6			326	329.1	111.3	117.1		145.9	162.3	218		226.7	228.8	312.1	320.5	425	463	
G243	111 - 2009-09-17-01-A	213.6	215.9	225.9	244.8	138.7		213.7		422.7			316	325.5	103.4	109.3		137.4	154.1	224		226.6	305.3	318	428.3	431.1		
G244	111 - 2009-09-17-01-A	202.7	213.7	226	244.9	151.9		212.8		422.8	424.7		316.2	325.6	103.5	115.3		137.5	154.2	211.8	220.1	226.7		293.4	410.7	428.3		
G245	111 - 2009-09-17-01-A	213.5		225.8		138.7		210.4		422.8			325.4		111.2	117.8		137.4	162	207.5	217.8	222.4	228.7	297.3	311.7	425.4	428.4	
G246	111 - 2009-09-17-01-A	202.5		244.6	259.3			210.3	213.5	422.8			325.3		114.9	122.8		145.8	158.1	219.7	240.1	228.7	239.1	307.6	313.8	410.8	440.2	
G246	111 - 2009-09-25-01-A	202.7		245.4	260			210.9	214.1	422.7			326.1		113.9	123.1		146	158.4	220.2	240.9	228.9	239.5	308	314.4	410.5	439.9	
G246	111 - 2009-09-29-01-A	202.4		244	258.8			210.1	213.2	422.7			324.8		114.9	122.9		145.6	157.9	219.4	239.7	228.4	238.8	307.2	313.4	410.8	440.2	
G246	111 - 2009-10-02-01-A	202.5		244.7	259.4			210.4	213.6	422.7			325.4		114.8	122.8		145.7	158			228.6	239.2		410.5	439.9		
G246	111 - 2009-11-04-02-A	202.8		245.4	260	157.8	163	210.9	214.1	422.7			326.1		116.1	123		146	158.4	220.2	240.9	228.9	239.5	308	314.3	410.5	439.9	
G247	111 - 2009-09-17-01-A	202.4		225.6		138.6		210.2		422.7			325.3		113.6			156	158	211.5	219.6	220.5	226.5	293.3	428.3	446		
G248	111 - 2009-09-17-01-A	205	213.4	225.6		157	163	210.3		424.6			325.1		111.1	113		137.3	149.9	219.7		226.5		299	440	451.7		
G249	111 - 2009-09-17-01-A	200.3	213.4	225.9	235.4			210.4	213.5	422.7			325.5		112	115.1		152.9	158	215.7	219.8	230.8	239.2	303.4	428.3	434.2		

Sample	Run	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12													
G249	111 - 2009-09-25-01-A	200.4	213.5	226.1	235.6		210.6	213.8	422.6		325.7	111.3	115.2	153	158.2	215.8	220	230.9	239.4	303.4		428.2	434			
G249	111 - 2009-09-29-01-A	200.1	213	225.4	234.7		210	213.3	422.6		324.7	111	114.8	152.7	157.8	215.3	219.4	230.5	238.9	303		428.3	434.2			
G249	111 - 2009-10-02-01-A	200.4	213.4	225.9	235.4		210.4	213.8	422.6		325.5	111.3	115.1	152.9	158.1			230.8	239.2			428.1	434			
G249	111 - 2009-11-04-02-A	200.6	213.8	226.4	235.9	157.8	167.3	210.8	214.4	422.5		326.1	112.4	116.2	153.2	158.4	216.2	220.3	231.1	239.6	303.6		428	433.9		
G250	111 - 2009-09-17-01-A	200.2	202.5	225.8	256.6			209.4	422.8		303.3	325.6	114.9	122.7	145.7		211.6		226.5		291.5		428.3	451.8		
G250	111 - 2009-11-04-02-A			225.9		155.6	161.8	209.5					115	122.9	145.8		211.7									
G251	111 - 2009-09-17-01-A	197.5		210.2											182.4								413			
G251	111 - 2009-09-25-01-A	215.5	220	225.8	256.7		210.3	213.9	422.7	424.6		325.6	328.7	109.2	117	137.4	158.1	215.6	217.7	226.7	234.9	299.2	313.9	410.5	428.1	
G251	111 - 2009-09-29-01-A	215.2	219.6	225.4	256	157.1	163.1	210.1	213.5	422.6	424.6		324.8	327.9	110	117.8	137.2	157.9	215.3	217.4	226.4	234.7	298.9	313.4	410.6	428.3
G251	111 - 2009-11-04-02-A	215.5	220.2	226	256.8	156.9	163	210.6	214.1	422.7	424.7		325.7	328.9	110.3	118.1	137.5	158.2	215.8	217.9	226.8	235.1	299.3	314.1	410.6	428.2
G252	111 - 2009-09-17-01-A	204.8	213.7	245			210.6	213.8	422.7		325.8	113.2	117	153	158.2	220.1			226.7		293.6		434.2	442.9		
G252	111 - 2009-09-25-01-A	204.6	213.5	244.9			210.4	213.7	422.5		325.5	113.1	117	152.9	158.1	219.8			226.6	237.1	293.4		434	442.9		
G252	111 - 2009-09-29-01-A	204.6	213.3	244.2			210.2	213.4	422.8		324.9	113.1	116.9	152.9	158	219.5			220.5	226.5	293.1		434.3	443		
G252	111 - 2009-10-02-01-A	204.6	213.5	244.8			210.5	213.6	422.7		325.6	113.1	117	152.9	158.1				226.5	237			434	442.9		
G252	111 - 2009-11-04-02-A	204.8	213.6	244.9		158.1	163	210.6	214	422.7		325.6	114.2	118	153.1	158.3	220		226.7	237.2	293.5		434	442.9		
G253	111 - 2009-11-04-01-A	202.6	213.6	226	256.8	162.9		207.4	210.6	422.6		316.3	112.2	116	145.8	166	219.9		226.6	228.7	293.5		436.9	439.8		
G254	111 - 2009-11-04-01-A	202.4	213.5	225.9				213.7	422.6		322.4	325.6	110.3	112.2	145.8	154.1	219.9	221.9	226.6	229.7			428	433.9		
G255	111 - 2009-11-04-01-A	202.6	211.4	225.9	256.7	156.8	162.9	213.9	422.6		316.1	325.6	110.2	114	153	154.1	219.9		226.6	237.1	318		422	451.4		
G256	111 - 2009-11-04-01-A	213.7	216.1	226.1	235.6	156.8	162.9	210.7	213.9				112.2	118	145.9		220.1	222.1	226.7	235.2						
G257	111 - 2009-11-04-01-A	202.6	213.5	226		163		210.6	422.6		325.7	328.9	114.2		145.9	160.3	217.9	222	226.6		293.4		434	445.8		
G258	111 - 2009-11-04-01-A	202.7		226.1	256.9			210.7	213.9	410.8	422.7		322.7	325.8	110.3	112.2	145.9	156.3	218	220.1	226.7	228.8	291.7	410.6	440	
G259	111 - 2009-11-04-01-A			245.1										326		109.6			153		218.2		224.7			

Table A.2: Allele scores calculated manually by experts in *Pinus patula* study

Sample	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB195	200	203	227	227	157	163	211	214	421	425	326	326	114	116	152	160	212	216	221	235	306	323	434	440
AB196	200	203	227	258	157	163	211	211	423	423	304	326	116	124	146	146	212	220	227	227	292	292	428	452
AB197	212	218			157	163	214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB198	203	214	227	246	152	152	214	214	423	425	317	326	104	116	138	154	212	220	227	227	294	294	410	428
AB199	214	216	227	227	157	163	211	211	423	423	326	329	112	118	146	162	218	218	227	229	312	321	425	462
AB200	203	212	227	258	157	163	214	214	423	423	317	326	110	114	152	154	220	220	227	237	319	319	422	452
AB201	203	214	227	227	157	163	214	214	423	423	323	326	110	112	146	154	220	222	227	229	304	312	428	434
AB202	214	216	227	227	157	163	211	214	411	425	326	326	110	118	138	146	212	216	227	227	294	294	431	440
AB203	212	218	227	227	157	163	211	214	423	423	326	329	110	114	150	160	220	220	227	227	294	294	422	422
AB204	203	203	227	258	157	163	211	214	425	425	326	326	112	112	146	152	222	222	221	229	302	319	440	440
AB205	203	216	227	244	152	152	211	211	423	425	326	326	114	122	146	152	220	220	227	227	294	312	440	440
AB206	200	203	244	249	157	163	211	211	423	423	310	326	110	112	152	154	218	218	221	227	302	319	443	446
AB207	200	203	227	246			211	211	423	425	317	326	118	118	154	154	218	218	233	237	306	314	422	431
AB208	216	216	227	246	152	152	214	214	423	423	326	335	114	116	146	146	218	220	227	227	294	294	410	446
AB209	214	216	227	236			211	214	423	423	326	326	112	118	146	146	220	222	227	235	308	321	434	462
AB210	203	214	227	256	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	294	428	437
AB211	200	203	227	260	152	152	211	211	423	423	317	317	112	114	146	160	212	216	233	237	308	308	434	434
AB212	203	214	227	258	163	163	208	211	423	423	317	317	112	116	146	166	220	220	227	229	294	294	437	440
AB213	203	216	227	244	157	157	211	216	423	423	317	326	114	120	138	140	212	220	229	237	304	304	428	428
AB214	203	214	244	258	157	163	208	214	421	423	317	326	104	114	146	154	220	220	227	227	294	294	422	428
AB215	212	212	246	258	163	163	211	211	423	425	326	326	114	120	146	146	220	220	227	227	300	306	440	440
AB216	203	214	244	258	157	163	208	214	425	425	323	326	112	118	156	158	216	220	227	227	314	314	431	449
AB217	203	214	246	258	146	163	211	214	423	423	323	326	104	114	146	152	220	230	227	227	294	294	440	446
AB218	203	214	227	258	163	163	208	211	423	423	317	317	112	116	146	166	220	220	227	229	294	294	437	440
AB219	203	216	227	246	157	157	214	214	423	423	317	326	104	110	138	154	220	225	227	227	294	294	431	452
AB220	203	214	227	227	163	163	211	211	423	423	326	329	114	114	146	160	218	222	227	227	294	294	434	446
AB221	212	216	227	244	157	157			423	423	317	326	104	110	138	154	225	225	227	227	306	319	428	431
AB222	203	203	246	260	157	163	211	214	423	423	326	326	116	124	146	158	220	240	229	239	308	314	410	440
AB223	203	203	227	260			211	211	411	411	326	326	114	122	146	152	216	220	227	227	310	314	410	422
AB224	214	214	227	227	157	163	211	211	423	423	326	326	112	118	138	162	207	218	223	229	298	312	425	428

Sample	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB225	220	220	246	249	152	152	211	214	423	425	317	326	110	110	152	156	218	220	229	237	314	314	428	434
AB226	210	216	227	227	157	157	195	214	423	423	317	326	114	116	146	160	220	220	221	227	294	294	440	446
AB227	214	220	227	258	167	167	211	214	423	425	314	326	104	116	146	158	216	222	227	227	319	319	410	428
AB228	200	203	227	244	167	167	211	211	423	425	317	326	118	118	154	154	218	218	233	237	306	314	422	431
AB229	203	216	227	258			214	214	423	423	317	326	116	116	154	154	212	220	227	227	302	319	410	428
AB230	203	203	227	258	157	157	211	214	411	423	323	326	110	112	146	156	218	220	227	229	292	292	410	440
AB231	203	214					214	214	423	423	323	326	110	112	146	154	220	222	227	229	304	312		
AB232	200	203	227	227	157	163	211	214	421	425	326	326	114	116	152	160	212	216	221	235	306	323	434	440
AB233	212	218	227	246	157	163	214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB234	203	216	227	258	152	152	214	214	423	423	317	326	116	116	154	154	212	220	227	227	302	319	410	428
AB235	214	216	227	227	157	163	211	211	423	423	326	329	112	118	146	162	218	218	227	229	312	321	425	462
AB236	214	216	227	227	157	163	211	214	411	425	326	326	110	118	138	146	212	216	227	227	294	294	431	440
AB237	200	203	244	249	157	163	211	211	423	423	310	326	110	112	152	154	218	218	221	227	302	319	443	446
AB238	203	214	227	227	163	163	211	211	423	423	326	329	114	114	146	160	218	222	227	227	294	294	434	446
AB239	203	214	227	246	152	152	214	214	423	425	317	326	104	116	138	154	212	220	227	227	294	294	410	428
AB240	203	214	227	258	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	312	428	437
AB241	200	203	227	260	152	152	211	211	423	423	317	317	112	114	146	160	212	216	233	237	308	308	434	434
AB242	203	203	246	260			211	214	423	423	326	326	116	124	146	158	220	240	229	239	308	314	410	440
AB243	203	203	227	258	157	163	211	214	425	425	326	326	112	112	146	152	222	222	221	229	302	319	440	440
AB244	203	214	244	258	157	163	208	214	421	423	317	326	104	114	146	154	220	220	227	227	294	294	422	428
AB245	200	203	227	246	157	167	211	211	423	425	317	326	118	118	154	154	218	218	233	237	306	314	422	431
AB246	203	214	227	258	163	163	208	211	423	423	317	317	112	116	146	166	220	220	227	229	294	294	437	440
AB247	200	200	227	249	157	163	211	214	411	423	323	326	114	114	146	146	218	220	227	239	317	319	440	440
AB248	214	216	227	236	157	163	211	214	423	423	326	326	112	118	146	146	220	222	227	235	308	321	434	462
AB249	203	214	227	256	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	294	428	437
AB250	212	216	227	244	157	157	214	214	423	423	317	326	104	110	138	154	225	225	227	227	306	319	428	431
AB251	203	212	227	258	157	163	214	214	423	423	317	326	110	114	152	154	220	220	227	237	319	319	422	452
AB252	203	214	227	227	157	163			423	423	323	326	110	112	146	154	220	222	227	229	304	312	428	434
AB253	214	216	227	227	152	152	211	214	421	423	326	326	114	114	146	166	216	220	227	239	312	319	410	437
AB254	212	218	227	227			211	214	423	423	326	329	110	114	150	160	220	220	227	227	294	294	422	422
AB255	216	216			157	163	214	214	423	423	317	326	114	116	152	154	220	220	227	227	294	294	428	440

Sample	Marker1		Marker2		Marker3		Marker4		Marker5		Marker6		Marker7		Marker8		Marker9		Marker10		Marker11		Marker12	
AB256	214	216	227	227	157	163	211	211	423	423	326	329	112	118	146	162	218	218	227	229	312	321	425	462
AB257	203	214	227	258	163	163	208	211	423	423	317	317	112	116	146	166	220	220	227	229	294	294	437	440
AB258	200	203	227	258			211	214	423	423	326	326	114	118	146	146	222	227	227	229			434	440
AB259	203	203	227	256	157	163	211	214	425	425	326	326	112	112	146	152	222	222	221	229	302	319	440	440
AB260	203	212	227	227			214	214	423	425	317	326	110	118	146	154	220	220			294	294	428	440
AB261	203	216	227	244	152	152	211	211	423	425	326	326	114	122	146	152	220	220	227	227	294	312	440	440
AB262	212	212	246	258	163	163	211	211	423	425	326	326	114	120	146	146	220	220	227	227	300	306	440	440
AB263	214	216	227	227	157	163	211	211	423	423	326	329	112	118	146	162	218	218	227	229	312	321	425	462
AB264	210	216	227	227	157	157	195	214	423	423	317	326	114	116	146	160	220	220	221	227	294	294	440	446
AB265	200	203	227	260	152	152	211	211	423	423	317	317	112	114	146	160	212	216	233	237	308	308	434	434
AB266	203	216	227	246	157	157	214	214	423	423	317	326	104	110	138	154	220	225	227	227	294	294	431	452
AB267	203	203	227	258	157	163	211	214	425	425	326	326	112	112	146	152	222	222	221	229	302	319	440	440
AB268	214	214	227	227	157	163			423	423	326	326	112	118	138	162	207	218	223	229	298	312	425	428
AB269	212	218	227	227			211	214	423	423	326	329	110	114	150	160	220	220	227	227	294	294	422	422
AB270	203	216	227	258	152	152	214	214	423	423	317	326	116	116	154	154	212	220	227	227	302	319	410	428
AB271	214	214	227	258	157	163	214	214	425	425	323	326	110	118	146	146	220	222	221	227	319	325	410	428
AB272	212	218	227	244	157	163	214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB273	214	214	227	227	157	163	211	211	423	423	326	326	112	118	138	162	207	218	223	229	298	312	425	428
AB274	200	214	227	236	167	167	211	214	423	423	326	326	112	116	152	158	216	220	231	239	304	304	428	434
AB275	212	212	246	258	163	163	211	211	423	425	326	326	114	120	146	146	220	220	227	227	300	306	440	440
AB276	203	214	227	256	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	294	428	437
AB277	203	216	227	244	152	152	211	211	423	425	326	326	114	122	146	152	220	220	227	227	294	312	440	440
AB278	214	216	227	227	157	163	211	214	411	425	326	326	110	118	138	146	212	216	227	227	294	294	431	440
AB279	203	203	227	227	157	163	214	214	423	425	317	326			146	154	220	220	227	227	294	294	428	440
AB280	203	214	227	227	163	163	211	211	423	423	326	329	114	114	146	160	218	222	227	227	294	294	434	446
AB281	212	216	227	244	157	157	214	214	423	423	317	326	104	110	138	154	225	225	227	227	306	319	428	431
AB282	203	214	227	227	157	163	214	214	423	423	323	326	110	112	146	154	220	222	227	229	304	312	428	434
AB283	203	216	227	244	157	157	214	214	423	423	317	326	104	110	138	154	220	225	227	227	294	294	431	452
AB284	212	218	227	244	157	163	214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB285	203	203	227	256	157	157	211	214	411	423	323	326	110	112	146	156	218	220	227	229	292	292	410	440
AB286	203	203	246	260			211	214	423	423	326	326	116	124	146	158	220	240	229	239	308	314	410	440

Sample	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB287	203	212	227	258	157	163	214	214	423	423	317	326	110	114	152	154	220	220	227	237		422	452	
AB288	203	214					211	214	423	423	323	326	104	114	146	152	220	230	227	227	294	294	440	446
AB289	214	216	227	236	157	163	211	214	423	423	326	326	112	118	146	146	220	222	227	235	308	321	434	462
AB290	200	200	227	249			211	211	411	423	323	326	114	114	146	146	218	220	227	239	317	319	440	440
AB291	203	216	227	258	152	152	214	214	423	423	317	326	116	116	154	154	212	220	227	227	302	319	410	428
AB292	203	214	227	258	163	163	208	211	423	423	317	317	112	116	146	166	220	220	227	229	294	294	437	440
AB293	200	203	227	260	152	152	211	211	423	423	317	317	112	114	146	160	212	216	233	237	308	308	434	434
AB294	203	216	227	258	152	152	214	214	423	423	317	326	116	116	154	154	212	220	227	227	302	319	410	428
AB295	200	203	227	244	167	167	211	211	423	425	317	326	118	118	154	154	218	218	233	237	306	314	422	431
AB296	203	216	227	244	152	152	211	211	423	425	326	326	114	122	146	152	220	220	227	227	294	312	440	440
AB297	203	214	227	227	157	163	214	214	423	423	323	326	110	112	146	154	220	222	227	229	304	312	428	434
AB298	203	203	227	258			211	214	425	425	326	326	112	112	146	152	222	222	221	229	302	319	440	440
AB299	212	216	227	244	157	157	214	214	423	423	317	326	104	110	138	154	225	225	227	227	306	319	428	431
AB300	214	216	227	227	157	163	211	211	423	423	326	329	112	118	146	162	218	218	227	229	312	321	425	462
AB301	200	203	227	260	152	152	211	211	423	423	317	317	112	114	146	160	212	216	233	237	308	308	434	434
AB302	203	214	244	256	163	163	211	214	423	423	323	326	104	114	146	152	220	230	227	227	294	294	440	446
AB303	214	214	227	227	157	163	211	211	423	423	326	326	112	118	138	162	207	218	223	229	298	312	425	428
AB304	203	214	227	256	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	294	428	437
AB305	203	214	244	258	157	163	208	214	421	423	317	326	104	114	146	154	220	220	227	227	294	294	422	428
AB306	203	203	227	258	157	157	211	214	411	423	323	326	110	112	146	156	218	220	227	229	292	292	410	440
AB307	200	214	227	236	167	167	211	214	423	423	326	326	112	116	152	158	216	220	231	239	304	304	428	434
AB308	200	206	227	244	157	163	211	214	423	423	323	329	114	118	152	158	218	222	227	227	319	319	431	440
AB309	220	220	246	249	152	152	211	214	423	425	317	326	110	110	152	156	218	220	229	237	314	314	428	434
AB310	203	216	227	246	157	157	214	214	423	423	317	326	104	110	138	154	220	225	227	227	294	294	431	452
AB311	203	203	227	227	157	163	211	211	423	423	326	326	116	120	156	158	212	220	221	227	294	294	428	446
AB312	212	212	246	258	163	163	211	211	423	425	326	326	114	120	146	146	220	220	227	227	300	306	440	440
AB313	203	214	227	246	157	163	214	214	411	425	326	329	112	114	146	146	222	222	227	227	294	294	422	428
AB314	203	203	227	258	157	157	211	214	411	423	323	326	110	112	146	156	218	220	227	229	292	292	410	440
AB315	212	218	227	244	157	163	214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB316	203	214	244	258	157	163	208	214	421	423	317	326	104	114	146	154	220	220	227	227	294	294	422	428
AB317	203	212	227	258	157	163	214	214	423	423	317	326	110	114	152	154	220	220	227	237	319	319	422	452

Sample	Marker1		Marker2		Marker3		Marker4		Marker5		Marker6		Marker7		Marker8		Marker9		Marker10		Marker11		Marker12	
AB318	203	216	227	246	157	157	214	214	423	423	317	326	104	110			220	225	227	227	294	294	431	452
AB319	203	203	244	258	163	163	211	211	423	423	314	326	114	116	146	154	220	227	227	227	294	294	422	449
AB320	200	214			142	167	211	214	423	423	326	326			152	158	216	220	231	239	304	304	428	434
AB321	212	218	227	227			211	214	423	423	326	329	110	114	150	160	220	220	227	227	294	294	422	422
AB322	203	203	246	260			211	214	423	423	326	326	116	124	146	158	220	240	229	239	308	314	410	440
AB323	200	200	227	249	157	163	211	214	411	423	323	326	114	114	146	146	218	220	227	239	317	319	440	440
AB324	203	203	227	244	163	163	214	214	423	423	329	342	112	116	154	158	220	222	227	227	294	294	428	440
AB325	203	214	244	256	157	163	211	211	423	425	326	326	114	124	152	152	205	220	227	227	298	298	410	428
AB326	203	214	227	227	157	163	214	214	423	423	323	326			146	154	220	222	227	229	304	312	428	434
AB327	203	203	246	260			211	214	423	423	326	326	116	122	146	158	220	240	229	239	308	314	410	440
AB328	212	218	227	244	157	163	214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB329	203	216	227	258	152	152	214	214	423	423	317	326	116	116	154	154	212	220	227	227	302	319	410	428
AB330	214	220	229	229	157	157	201	218	423	423	317	326	114	116	146	160	222	222	223	229	294	294	440	446
AB331	214	216	227	227	157	163	211	214	411	425	326	326	110	118	138	146	212	216	227	227	294	294	431	440
AB332	203	216	227	246	157	157	214	214	423	423	317	326	104	110	138	154	220	225	227	227	294	294	431	452
AB333	214	218	227	244	157	163	214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB334	203	214			163	163	211	211	423	423	326	329	114	114	146	160	218	222	227	227	294	294	434	446
AB335	203	214	227	244	152	152	214	214	423	425	317	326			138	154	212	220	227	227	294	294	410	428
AB336	200	203	227	260	152	152	211	211	423	423	317	317	112	114	146	160	212	216	233	237	308	308	434	434
AB337	214	216	227	227	152	152	211	214	421	423	326	326	114	114	146	166	216	220	227	239	312	319	410	437
AB338	203	203	227	258			211	214	425	425	326	326	112	112	146	152	222	222	221	229	302	319	440	440
AB339	214	218	227	244	157	163			423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB340	214	216	227	236	157	163	211	214	423	423	326	326	112	118	146	146	220	222	227	235	308	321	434	462
AB341	203	203	227	260	157	163	211	211	411	411	326	326	114	122	146	152	216	220	227	227	310	314	410	422
AB342	212	212	246	258	163	163	211	211	423	425	326	326	114	120	146	146	220	220	227	227	300	306	440	440
AB343	210	216	227	227	157	157	195	214	423	423	317	326	114	116	146	160	220	220	221	227	294	294	440	446
AB344	212	218	227	227	157	163	211	214	423	423	326	329	110	114	150	160	220	220	227	227	294	294	422	422
AB345	212	218	227	244	157	163	214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB346	200	214	227	227	157	163	211	216	411	423	323	326	110	124	160	164	218	220	223	227	319	319	434	446
AB347	212	216	227	244	157	157			423	423	317	326	104	110	138	154	225	225	227	227	306	319	428	431
AB348	210	216	227	227	146	157	195	214	423	423	317	326			146	160	220	220	221	227	294	294	440	446

Sample	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB349	203	214	227	256	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	294	428	437
AB350	214	216	227	227	157	163	211	211	423	423	326	329	112	118	146	162	218	218	227	229	312	321	425	462
AB351	200	214	227	227			211	216	411	423	323	326			160	164	218	220	223	227	319	319	434	446
AB352	200	203	210	227	157	157	211	211	421	423	317	326	114	124	138	146	220	227	229	229			428	440
AB353	203	203	246	260			211	214	423	423	326	326	116	124	146	158	220	240	229	239			410	440
AB354	214	216	227	227	157	163	211	214	411	425	326	326			138	146	212	216	227	227	294	294	431	440
AB355	200	203	227	258	157	163	211	211	423	423	304	326	116	124	146	146	212	220	227	227	292	292	428	452
AB356	200	203	227	244	167	167	211	211	423	425	317	326	118	118	154	154	218	218	233	237	306	314	422	431
AB357	214	214	227	227	157	163	211	211	423	423	326	326	112	118	138	162	207	218	223	229	298	312	425	428
AB358	203	214	227	256	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	294	428	437
AB359	203	214	227	227	163	163	211	211	423	423	326	329	114	114	146	160	218	222	227	227	294	294	434	446
AB360	203	216	227	258	152	152	214	214	423	423	317	326	116	116	154	154	212	220	227	227	302	319	410	428
AB361	212	212	246	258	163	163	211	211	423	425	326	326	114	120	146	146	220	220	227	227	300	306	440	440
AB362	212	216	227	244	157	157	214	214	423	423	317	326	104	110	138	154	225	225	227	227	306	319	428	431
AB363	210	216	227	227	157	157	195	214	423	423	317	326	114	116	146	160	220	220	221	227	294	294	440	446
AB364	214	216	227	236	157	163	211	214	423	423	326	326	112	118	146	146	220	222	227	235	308	321	434	462
AB365	203	214	227	246	152	152	214	214	423	425	317	326	104	116	138	154	212	220	227	227	294	294	410	428
AB366	200	203	227	227	157	163	211	214	423	423	314	326	104	114	138	146	212	218	227	227	294	294	431	440
AB367	203	203	227	227	157	163	214	214	423	425	317	326			146	154	220	220	227	227	294	294	428	440
AB368	203	212	227	258	157	163	214	214	423	423	317	326	110	114	152	154	220	220	227	237	319	319	422	452
AB369	200	203	227	246			211	211	423	425	317	326	118	118	154	154	218	218	233	237			422	431
AB370	203	214	227	258	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	312	428	437
AB371	214	216	227	227			211	211	423	423	326	329	112	118	146	162	218	218	227	229	312	321	425	462
AB372	212	218	227	244	157	163	214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
AB373	203	203	227	258	157	163	211	214	425	425	326	326	112	112	146	152	222	222	221	229	302	319	440	440
AB374	203	216	227	244	157	157	214	214	423	423	317	326	104	110	138	154	220	225	227	227	294	294	431	452
AB375	212	218	227	244	157	163			423	423	314	326	114	116	154	162	216	218	221	229	308	312	428	440
AB376	203	238	258	258	157	163	211	214	423	425	314	326	112	124	152	158	220	220	227	229	304	321	434	440
AB377	203	214	227	227	163	163	211	211	423	423	326	329	114	114	146	160	218	222	227	227	294	294	434	446
AB378	203	214	227	246	157	163	214	214	411	425	326	329	112	114	146	146	222	222	227	227	294	294	422	428
AB379	214	214	227	258	157	163	214	214	425	425	323	326	110	118	146	146	220	222	221	227	319	325	410	428

Sample	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB380	203	216	227	246	157	157	214	214	423	423	317	326	104	110	138	154	220	225	227	227	294	294	431	452
AB381	203	214	227	256	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	312	428	437
AB382	203	203	227	260	157	163	211	211	411	411	326	326	114	122	146	152	216	220	227	227	310	314	410	422
AB383	203	214	227	244	163	163	211	214	423	423	323	326	114	116	146	150	218	218	227	229	294	294	437	440
AB384	214	216	227	227	157	163	211	211	423	423	326	329	112	118	146	162	218	218	227	229	312	321	425	462
AB385	200	203	227	258	157	163	211	211	423	423	304	326	116	124	146	146	212	220	227	227	292	292	428	452
G217	203	214	244	256			211	211	423	425	326	326	114	124	152	152	205	220	227	227	298	298	410	428
G218	200	203	227	227	157	163	211	214	423	423	314	326	104	114	138	146	212	218	227	227	294	294	431	440
G219	203	214	227	236	167	167	211	214	423	425	326	326	116	118	158	158	216	222	227	231	304	304	428	440
G220	210	216	227	227	157	157	195	214	423	423	317	326	114	116	146	160	220	220	221	227	294	294	440	446
G221	203	203	246	260	157	163	211	214	423	423	326	326	116	124	146	158	220	240	229	239	308	314	410	440
G222	200	203	227	258	157	163	211	214	423	423	326	326	114	118	146	146	222	227	227	229	317	319	434	440
G223	214	220	227	258			211	214	423	425	314	326	104	116	146	158	216	222	227	227	317	333	410	428
G224	203	214	227	256	163	163	211	214	423	425	317	329	114	116	146	152	222	227	223	227	294	294	428	437
G225	203	216	227	227			211	214	423	425	326	326	114	116	146	152	218	222	227	227	302	306	428	437
G226	200	216	227	244			211	214	423	423	326	326	95	114	152	158	216	218	223	227	312	321	422	431
G227	200	206	227	244			211	214	423	423	323	329	114	118	152	158	218	222	227	239	319	319	431	440
G228	214	214	227	258	157	163	214	214	425	425	323	326	110	118	146	146	220	222	221	227	319	325	410	428
G229	212	212	246	258	163	163	211	211	423	425	326	326	114	120	146	146	220	220	227	227	300	306	440	440
G230	203	214	244	258			208	214	421	423	317	326	104	114	146	154	220	220	227	227	294	294	422	428
G231	200	203	227	227			211	214	421	425	326	326	114	116	152	160	212	216	221	235	306	323	434	440
G232	212	218	227	244			214	214	423	423	314	326	116	116	154	162	216	218	221	229	308	312	428	440
G233	203	203	227	258	157	163	211	214	425	425	326	326	112	112	146	152	222	222	221	229	302	319	440	440
G234	200	203	244	249	157	163	211	211	423	423	310	326	110	112	152	154	218	218	221	227	302	319	443	446
G235	203	203	227	256			214	214	423	423	326	326	114	120	146	158	207	220	227	227	314	319	428	428
G236	203	214	246	258	146	163	211	214	423	423	323	326	104	114	146	152	220	230	227	227	294	294	440	446
G237	203	214	227	246	157	163	214	214	411	425	326	329	112	114	146	146	222	222	227	227	294	294	422	428
G238	203	238	258	258			211	214	423	425	314	326	112	124	152	158	220	220	227	229	304	321	434	440
G239	203	203	227	227	157	163	214	214	423	425	317	326	110	118	146	154	220	220	227	227	294	294	428	440
G240	203	203	227	227			214	214	423	423	326	326	104	114	146	152	216	220	227	229	294	294	440	446
G241	203	216	227	258	152	152	214	214	423	423	317	326	116	116	154	154	212	220	227	227	302	319	410	428

Sample	Marker1		Marker2		Marker3		Marker4		Marker5		Marker6		Marker7		Marker8		Marker9		Marker10		Marker11		Marker12	
G242	214	216	227	227			211	211	423	423	326	329	112	118	146	162	218	218	227	229	312	321	425	462
G243	212	216	227	244			214	214	423	423	317	326	104	110	138	154	225	225	227	227	306	319	428	431
G244	203	214	227	246	152	152	214	214	423	425	317	326	104	116	138	154	212	220	227	227	294	294	410	428
G245	214	214	227	227			211	211	423	423	326	326	112	118	138	162	207	218	223	229	298	312	425	428
G246	203	203	246	260	157	163	211	214	423	423	326	326	116	124	146	158	220	240	229	239	308	314	410	440
G247	203	203	227	227			211	211	423	423	326	326			156	158	212	220	221	227	294	294	428	446
G248	206	214	227	227	157	163	211	211	425	425	326	326	112	114	138	150	220	220	227	227	300	300	440	452
G249	200	214	227	236	167	167	211	214	423	423	326	326	112	116	152	158	216	220	231	239	304	304	428	434
G250	200	203	227	258	157	163	211	211	423	423	304	326	116	124	146	146	212	220	227	227	292	292	428	452
G251	216	220	227	256	157	163	211	214	423	425	326	329	110	118	138	158	216	218	227	235	300	314	410	428
G252	206	214	244	244	157	163	211	214	423	423	326	326	114	118	152	158	220	220	227	237	294	294	434	443
G253	203	214	227	258	163	163	208	211	423	423	317	317	112	116	146	166	220	220	227	229	294	294	437	440
G254	203	214	227	227	157	163	214	214	423	423	323	326	110	112	146	154	220	222	227	229	304	312	428	434
G255	203	212	227	258	157	163	214	214	423	423	317	326	110	114	152	154	220	220	227	237	319	319	422	452
G256	214	216	227	236	157	163	211	214	423	423	326	326	112	118	146	146	220	222	227	235	308	321	434	462
G257	203	214	227	227	163	163	211	211	423	423	326	329	114	114	146	160	218	222	227	227	294	294	434	446
G258	203	203	227	258	157	157	211	214	411	423	323	326	110	112	146	156	218	220	227	229	292	292	410	440
G259	200	203	227	227	157	157	211	211	421	423	317	326	114	124	138	146	220	227	229	229	319	321	428	440

Table A.3: Allele scores calculated using GenoSonic in *Pinus patula* study

Sample	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB195	200	203	226		157	163	210	214	421	425	326	114	116	153	160	212	216	221	235	306	322	434	440	
AB196	200	203	226	257	157	163	210		423		303	326	116	124	146		212	220	227		292		428	451
AB197	214	218			157	163	214		423		313	326	116		154	162	216	218	221	229	308	312	428	440
AB198	203	214	226	245	152		214		423	425	316	326	104	116	137	154	212	220	227		293		411	428
AB199	214	216	226		157	163	210		423		326	329	112	118	146	162	218		227	229	312	320	425	463
AB200	203	212	226	257	157	163	214		423		316	326	110	114	153	154	220		227	237	318		422	451
AB201	203	214	226		157	163	214		423		323	326	110	112	146	154	220	222	227	230	303	312	428	434
AB202	214	216	226		157	163	211	214	411	425	326		111	118	137	146	212	216	227		293		431	440
AB203	212	218	226		157	163	210	214	423		326	329	110	114	150	160	220		227		293		422	
AB204	203		226	257	157	163	210	214	425		326		112		146	153	222		221	229	301	318	440	
AB205	203	216	226	245	152		210		423	425	326		114	123	146	153	220		227		293	312	440	
AB206	200	203	245	248	157	163	210		423		310	326	110	112	153	154	218		221	227	301	318	443	446
AB207	200	203	226	245			210		423	425	316	326	118		154		218		233	237	306	314	422	431
AB208	216		226	245	152		214		423		326	336	114	116	146		218	220	227		293		411	446
AB209	214	216	226	236			210	214	423		326		112	118	146		220	222	227	235	308	320	434	463
AB210	203	214	226	257	163		210	214	423	425	316	329	114	116	146	153	222	228	222	227	293		428	437
AB211	200	203	226	260	152		210		423		316		112	114	146	160	212	216	233	237	308		434	
AB212	203	214	226	257	163		207	210	423		316		112	116	146	166	220		227	229	293		437	440
AB213	203	216	226	245	157		210	217	423		316	326	114	120	137	139	212	220	229	237	303		428	
AB214	203	214	245	257	157	163	207	214	421	423	316	326	104	114	146	154	220		227		293		422	428
AB215	212		245	257	163		210		423	425	326		114	120	146		220		227		299	306	440	
AB216	203	214	245	257	157	163	207	214	425		323	326	112	118	156	158	216	220	227		314		431	449
AB217	203	214	245	257	147	163	210	214	423		323	326	104	114	146	153	220	230	227		293		440	446
AB218	203	214	226	257	163		207	211	423		316		112	116	146	166	220		227	229	293		437	440
AB219	203	216	226	245	157		214		423		316	326	104	110	137	154	220	224	227		293		431	451
AB220	203	214	226		163		210		423		326	329	114		146	160	218	222	227		293		434	446
AB221	214	216	226	245	157				423		316	326	104	110	137	154	224		227		306	318	428	431
AB222	203		245	260	157	163	210	214	423		326		116	124	146	158	220	240	229	239	308	314	411	440
AB223	203		226	260			210		411		326		114	123	146	153	216	220	227		310	314	411	422
AB224	214		226		157	163	210		423		326		112	118	137	162	208	218	222	229	297	312	425	428

Sample	Marker1	Marker2	Marker3	Marker4	Marker5	Marker6	Marker7	Marker8	Marker9	Marker10	Marker11	Marker12												
AB225	220	245	248	152	210	214	423	425	316	326	110	153	156	218	220	229	237	314	428	434				
AB226	209	216	226	157	195	214	423		316	326	114	116	146	160	220	221	227	293	440	446				
AB227	214	220	226	257	167	210	214	423	425	313	326	104	116	146	158	216	222	227	318	411	428			
AB228	200	203	226	245	167	210		423	425	316	326	118		154		218		233	237	306	314	422	431	
AB229	203	216	226	257	152	214		423		316	326	116		154		212	220	227		301	318	411	428	
AB230	203		226	257	157	210	214	411	423	323	326	110	112	146	156	218	220	227	229	292		411	440	
AB231	203	214				214		423		323	326	110	112	146	154	220	222	227	230	303	312			
AB232	200	203	226	157	163	210	214	421	425	326		114	116	153	160	212	216	221	235	306	322	434	440	
AB233	214	218	226	245		214	220	423		313	326	126		163	170	216	218	229		308	312	428	440	
AB234	203	216	226	257	152	214		423		316	326	116		154		212	220	227		301	318	411	428	
AB235	214	216	226	157	163	210		423		326	329	112	118	146	162	218		227	229	312	320	425	463	
AB236	214	216	226	157	163	210	214	411	425	326		110	118	137	146	212	216	227		293		431	440	
AB237	200	203	245	248	157	163	210		423	310	326	110	112	153	154	218		221	227	301	318	443	446	
AB238	203	214	226	163		210		423		326	329	114		146	160	218	222	227		293		434	446	
AB239	203	214	226	245	152	214		423	425	316	326	104	116	137	154	212	220	227		293		411	428	
AB240	203	214	226	257	163	210	214	423	425	316	329	114	116	146	153	222	228	222	227	293	312	428	437	
AB241	200	203	226	260	152	210		423		316		112	114	146	160	212	216	233	237	308		434		
AB242	203		245	260		210	214	423		326		116	124	146	158	220	240	229	239	308	314	411	440	
AB243	203		226	257	157	163	210	214	425	326		112		146	153	222		221	229	301	318	440		
AB244	203	214	245	257	157	163	207	214	421	423	316	326	104	114	146	154	220		227		293		422	428
AB245	200	203	226	245	157	167	210		423	425	316	326	118		153		218		233	237	306	314	422	431
AB246	203	214	226	257	163		207	210	423		316		112	116	146	166	220		227	229	293		437	440
AB247	200		226	248	157	163	210	214	411	423	323	326	114		146		218	220	227	239	316	318	440	
AB248	214	216	226	236	157	163	210	214	423		326		112	118	146		220	222	227	235	308	320	434	463
AB249	203	214	226	257	163	210	214	423	425	316	329	114	116	146	153	222	228	222	227	293		428	437	
AB250	212	216	226	245	157		214		423	316	326	104	110	137	154	224		227		306	318	428	431	
AB251	203	212	226	257	157	163	214		423	316	326	110	114	153	154	220		227	237	318		422	451	
AB252	203	214	226	157	163			423		323	326	110	112	146	154	220	222	227	230	303	312	428	434	
AB253	214	216	226	152		210	214	421	423	326		114		146	166	216	220	227	239	312	318	411	437	
AB254	212	218	226			210	214	423		326	329	110	114	150	160	220		227		293		422		
AB255	216			157	163	214		423		316	326	114	116	153	154	220		227		293		428	440	

Sample	Marker1		Marker2	Marker3		Marker4		Marker5		Marker6		Marker7		Marker8		Marker9		Marker10		Marker11		Marker12		
AB256	214	216	226		157	163	210		423	326	329	112	118	146	162	218		227	229	312	320	425	463	
AB257	203	214	226	257	163		207	210	423	316		112	116	146	166	220		227	229	293		437	440	
AB258	200	203	226	257			210	214	423	326		114	118	146		222	228	227	229			434	440	
AB259	203		226	257	157	163	210	214	425	326		112		146	153	222		221	229	301	318	440		
AB260	203	212	226				214		423	425	316	326	110	118	146	154	220			293		428	440	
AB261																								
AB262	212		245	257	163		210		423	425	326		114	120	146		220		227		299	306	440	
AB263	214	216	226		157	163	210		423	326	329	112	118	146	162	218		227	229	312	320	425	463	
AB264	209	216	226		157		195	214	423	316	326	114	116	146	160	220		221	227	293		440	446	
AB265	200	203	226	260	152		210		423	316		112	114	146	160	212	216	233	237	308		434		
AB266	203	216	226	245	157		214		423	316	326	104	110	137	154	220	224	227		293		431	451	
AB267	203		226	257	157	163	210	214	425	326		112		146	153	222		221	229	301	318	440		
AB268	214		226		157	163			423	326		112	118	137	162	208	218	222	229	297	312	425	428	
AB269	212	218	226				210	214	423	326	329	110	114	150	160	220		227		293		422		
AB270	203	216	226	257	152		214		423	316	326	116		154		212	220	227		301	318	411	428	
AB271	214		226	257	157	163	214		425	323	326	110	118	146		220	222	221	227	318	324	411	428	
AB272	212	218	226	245	157	163	214		423	313	326	116		154	162	216	218	221	229	308	312	428	440	
AB273	214		226		157	163	210		423	326		112	118	137	162	208	218	222	229	297	312	425	428	
AB274	200	214	226	236	167		210	214	423	326		112	116	153	158	216	220	231	239	303		428	434	
AB275	212		245	257	163		210		423	425	326		114	120	146		220		227		299	306	440	
AB276	203	214	226	257	163		210	214	423	425	316	329	114	116	146	153	222	228	222	227	293		428	437
AB277	203	216	226	245	152		210		423	425	326		114	122	146	153	220		227		293	312	440	
AB278	214	216	226		157	163	210	214	411	425	326		110	118	137	146	212	216	227		293		431	440
AB279	203		226		157	163	214		423	425	316	326			146	154	220		227		293		428	440
AB280	203	214	226		163		210		423	326	329	114		146	160	218	222	227		293		434	446	
AB281	212	216	226	245	157		214		423	316	326	104	110	137	154	224		227		306	318	428	431	
AB282	203	214	226		157	163	214		423	323	326	110	112	146	154	220	222	227	230	303	312	428	434	
AB283	203	216	226	245	157		214		423	316	326	104	110	137	154	220	224	227		293		431	451	
AB284	214	218	226	245	157	163	214		423	313	326	116		154	162	216	218	221	229	308	312	428	440	
AB285	203		226	257	157		210	214	411	423	323	326	110	112	146	156	218	220	227	229	292		411	440
AB286	203		245	260			210	214	423		326		116	124	146	158	220	240	229	239	308	314	411	440

Sample	Marker1		Marker2		Marker3		Marker4		Marker5		Marker6		Marker7		Marker8		Marker9		Marker10		Marker11		Marker12	
AB287	203	212	226	257	157	163	214		423	316	326	110	114	153	154	220	227	237				422	451	
AB288	203	214					210	214	423	323	326	104	114	146	153	220	230	227		293		440	446	
AB289	214	216	226	236	157	163	210	214	423	326		112	118	146		220	222	227	235	308	320	434	463	
AB290	200		226	248			210		411	423	323	326	114		146		218	220	227	239	316	318	440	
AB291	203	216	226	257	152		214		423	316	326	116		154		212	220	227		301	318	411	428	
AB292	203	214	226	257	163		207	210	423	316		112	116	146	166	220		227	229	293		437	440	
AB293	200	203	226	260	152		210		423	316		112	114	146	160	212	216	233	237	308			434	
AB294	203	216	226	257	152		214		423	316	326	116		154		212	220	227		301	318	411	428	
AB295	200	203	226	245	167		210		423	425	316	326	118		154		218		233	237	306	314	422	431
AB296	203	216	226	245	152		210		423	425	326		114	122	146	153	220		227		293	312	440	
AB297	203	214	226		157	163	214		423	323	326	110	112	146	154	220	222	227	230	303	312	428	434	
AB298	203		226	257			210	214	425	326		112		146	153	222		221	229	301	318	440		
AB299	212	216	226	245	157		214		423	316	326	104	110	137	154	224		227		306	318	428	431	
AB300	214	216	226		157	163	210		423	326	329	112	118	146	162	218		227	229	312	320	425	463	
AB301	200	203	226	260	152		210		423	316		112	114	146	160	212	216	233	237	308			434	
AB302	203	214	245	257	163		210	214	423	323	326	104	114	146	153	220	230	227		293		440	446	
AB303	214		226		157	163	210		423	326		112	118	137	162	208	218	222	229	297	312	425	428	
AB304	203	214	226	257	163		210	214	423	425	316	329	114	116	146	153	222	228	222	227	293		428	437
AB305	203	214	245	257	157	163	207	214	421	423	316	326	104	114	146	154	220		227		293		422	428
AB306	203		226	257	157		210	214	411	423	323	326	111	112	146	156	218	220	227	229	292		411	440
AB307	200	214	226	236	167		210	214	423	326		112	116	153	158	216	220	231	239	303		428	434	
AB308	200	205	226	245	157	163	210	214	423	323	329	114	118	153	158	218	222	227		318		431	440	
AB309	220		245	248	152		210	214	423	425	316	326	110		153	156	218	220	229	237	314		428	434
AB310	203	216	226	245	157		214		423	316	326	104	110	137	154	220	224	227		293		431	451	
AB311	203		226		158	163	210		423	326		116	120	156	158	212	220	221	227	293		428	446	
AB312	212		245	257	163		210		423	425	326		114	120	146		220		227		299	306	440	
AB313	203	214	226	245	157	163	214		411	425	326	329	112	114	146		222		227		293		422	428
AB314	203		226	257	157		210	214	411	423	323	326	110	112	146	156	218	220	227	229	292		411	440
AB315	214	218	226	245	157	163	214		423	313	326	116		154	162	216	218	221	229	308	312	428	440	
AB316	203	214	245	257	157	163	207	214	421	423	316	326	104	114	146	154	220		227		293		422	428
AB317	203	212	226	257	157	163	214		423	316	326	110	114	153	153	220		227	237	318		422	451	

Sample	Marker1		Marker2		Marker3		Marker4		Marker5		Marker6		Marker7		Marker8		Marker9		Marker10		Marker11		Marker12	
AB318	203	216	226	245	157		214		423	316	326	104	110			220	224	227		293		431	451	
AB319	203		245	257	163		210		423	313	326	114	116	146	154	220	228	227		293		422	449	
AB320	200	214				142	167	210	214	423		326			153	158	216	220	231	239	303		428	434
AB321	212	218	226					210	214	423		326	329	110	114	150	160	220		227		293		422
AB322	203		245	260			210	214	423		326		116	124	146	158	220	240	229	239	308	314	411	440
AB323	200		226	248	157	163	210	214	411	423	323	326	114		146		218	220	227	239	316	318	440	
AB324	203		226	245	163		214		423		329	341	112	116	154	158	220	222	227		293		428	440
AB325	203	214	245	257	157	163	210		423	425	326		114	124	153		206	220	227		297		411	428
AB326	203		226				214		423		323	326	110	112	146	154	220	222	227	230			428	434
AB327	203						210	214	423		326								229					
AB328	212	218	226	245	157	163	214		423		313	326	116		154	162	216	218	221	229	308	312	428	440
AB329	203	216	226	257	152		214		423		316	326	116		154		212	220	227		301	318	411	428
AB330	214	220	229		157		200	218	423		316	326	114	116	146	160	224		224	229	293		440	446
AB331	214	216	226		157	163	210	214	411	425	326		110	118	137	146	212	216	227		293		431	440
AB332	203	216	226	245	157		214		423		316	326	104	110	137	154	220	224	227		293		431	451
AB333	214	218	226	245	157	163	214		423		313	326	116		154	162	216	218	221	229	308	312	428	440
AB334	203	214			163		210		423		326	329	114		146	160	218	222	227		293		434	446
AB335	203	214	226	245	152		214		423	425	316	326			137	154	212	220	227		293		411	428
AB336	200	203	226	260	152		210		423		316		112	114	146	160	212	216	233	237	308			434
AB337	214	216	226		152		210	214	421	423	326		114		146	166	216	220	227	239	312	318	411	437
AB338	203		226	257			210	214	425		326		112		146	153	222		221	229	301	318	440	
AB339	214	218	226	245	157	163			423		313	326	116		154	162	216	218	221	229	308	312	428	440
AB340	214	216	226	236	157	163	210	214	423		326		112	118	146		220	222	227	235	308	320	434	463
AB341	203		226	260	157	163	210		411		326		114	122	146	153	216	220	227		310	314	411	422
AB342	212		245	257	163		210		423	425	326		114	120	146		220		227		299	306	440	
AB343	209	216	226		157		195	214	423		316	326	114	116	146	160	220		221	227	293		440	446
AB344	212	218	226		157	163	210	214	423		326	329	110	114	150	160	220		227		293			422
AB345	214	218	226	245	157	163	214		423		313	326	116		154	162	216	218	221	229	308	312	428	440
AB346	200	214	226		157	163	210	217	411	423	323	326	111	124	160	164	218	220	222	227	318		434	446
AB347	214	216	226	245	157				423		316	326	104	110	137	154	224		227		306	318	428	431
AB348	209	216	226		147	157	195	214	423		316	326			146	160	220		221	227	293		440	446

Sample	Marker1		Marker2		Marker3		Marker4		Marker5		Marker6		Marker7		Marker8		Marker9		Marker10		Marker11		Marker12	
AB349	203	214	226	257	163		210	214	423	425	316	329	114	116	146	153	222	228	222	227	293		428	437
AB350	214	216	226		157	163	210		423		326	329	112	118	146	162	218		227	229	312	320	425	463
AB351	200	214	226				210	217	411	423	323	326			160	164	218	220	222	227	318		434	446
AB352	200	203	210	226	157		210		421	423	316	326	114	124	137	146	220	226	229				428	440
AB353	203		245	260			210	214	423		326		116	124	146	158	220	240	229	239			411	440
AB354	214	216	226		157	163	210	214	411	425	326				137	146	212	216	227		293		431	440
AB355	200	203	226	257	157	163	210		423		303	326	116	124	146		212	220	227		292		428	451
AB356	200	203	226	245	167		210		423	425	316	326	118		154		218		233	237	306	314	422	431
AB357	214		226		157	163	210		423		326		112	118	137	162	208	218	222	229	297	312	425	428
AB358	203	214	226	257	163		210	214	423	425	316	329	114	116	146	153	222	228	222	227	293		428	437
AB359	203	214	226		163		210		423		326	329	114		146	160	218	222	227		293		434	446
AB360	203	216	226	257	152		214		423		316	326	116		154		212	220	227		301	318	411	428
AB361	212		245	257	163		210		423	425	326		114	120	146		220		227		299	306	440	
AB362	214	216	226	245	157		214		423		316	326	104	110	137	154	224		227		306	318	428	431
AB363	209	216	226		157		195	214	423		316	326	114	116	146	160	220		221	227	293		440	446
AB364	214	216	226	236	157	163	210	214	423		326		112	118	146		220	222	227	235	308	320	434	463
AB365	203	214	226	245	152		214		423	425	316	326	104	116	137	153	212	220	227		293		411	428
AB366	200	203	226		157	163	210	214	423		313	326	104	114	137	146	212	218	227		293		431	440
AB367	203		226		157	163	214		423	425	316	326			146	154	220		227		293		428	440
AB368	203	212	226	257	157	163	214		423		316	326	110	114	153	154	220		227	237	318		422	451
AB369	200	203	226	245			210		423	425	316	326	118		154		218		233	237			422	431
AB370	203	214	226	257	163		210	214	423	425	316	329	114	116	146	153	222	228	222	227	293	312	428	437
AB371	214	216	226				210		423		326	329	112	118	146	162	218		227	229	312	320	425	463
AB372	214	218	226	245	157	163	214		423		313	326	116		154	162	216	218	221	229	308	312	428	440
AB373	203		226	257	157	163	210	214	425		326		112		146	153	222		221	229	301	318	440	
AB374	203	216	226	245	157		214		423		316	326	104	110	137	154	220	224	227		293		431	451
AB375	214	218	226	245	157	163			423		313	326	114	116	154	162	216	218	221	229	308	312	428	440
AB376	203	237	257		157	163	210	214	423	425	313	326	112	124	153	158	220		227	229	303	320	434	440
AB377	203	214	226		163		210		423		326	329	114		146	160	218	222	227		293		434	446
AB378	203	214	226	245	157	163	214		411	425	326	329	112	115	146		222		227		293		422	428
AB379	214		226	257	157	163	214		425		323	326	110	118	146		220	222	221	227	318	324	411	428

Sample	Marker1		Marker2		Marker3		Marker4		Marker5		Marker6		Marker7		Marker8		Marker9		Marker10		Marker11		Marker12	
AB380	203	216	226	245	157		214		423		316	326	104	110	137	154	220	224	227		293		431	451
AB381	203	214	226	257	163		210	214	423	425	316	329	114	116	146	153	222	228	222	227	293	312	428	437
AB382	203		226	260	157	163	210		411		326		114	122	146	153	216	220	227		310	314	411	422
AB383	203	214	226	245	163		210	214	423		323	326	114	116	146	150	218		227	229	293		437	440
AB384	214	216	226		157	163	210		423		326	329	112	118	146	162	218		227	229	312	320	425	463
AB385	200	203	226	257	157	163	210		423		303	326	116	124	146		212	220	227		292		428	451
G217	203	214	245	257	139		210		423	425	326		114	123	153		206	220	227		297		411	428
G218	200	203	226		157	163	210	214	423		313	326	104	114	137	146	212	218	227		293		431	440
G219	203	214	226	236	139	167	210	213	423	425	326		116	118	158		216	222	227	231	303		428	440
G220	209	216	226		157		195	214	423		316	326	114	116	146	160	220		221	227	293		440	446
G221	203	214	245	260	139	147	210	214	423		326		115	123	146	158	220	240	229	239	308	314	411	440
G222	200	203	226	257	157	163	210	214	423		316	326	114	118	146		222	228	227	229	316	318	434	440
G223	214	220	226	257	139		210	214	423	425	313	326	103	116	146	158	216	222	227		318	332	411	428
G224	203	214	226	257	139	163	210	214	423	425	316	329	114	116	146	153	222	228	222	227	293		428	437
G225	203	216	226		139		210	214	423	425	326		114	115	146	153	218	222	227		301	306	428	437
G226	200	216	226	245	139		210	213	423		326		96	114	153	158	216	218	222	227	312	320	422	431
G227	200	205	226	245	139		210	214	423		323	329	114	117	153	158	218	222	227	239	318		431	440
G228	214		226	257			213	214	425		323	326	110	118	146		220	222	221	227			411	428
G229	212	214	245	257	152		210	214	423	425	313	326	104	114	146	158	216	220	221	227	299	306	411	440
G230	203	214	245	257	139		207	214	421	423	316	326	103	114	146	153	220		227		293		422	428
G231	200	203	226		147	185	210	213	421	425	326		114	116	153	160	212	216	221		306	322	434	440
G232	214	218	226	245	139		213		423		313	326	116		154	162	216	218	221	229	308	312	428	440
G233	203		226	257	157	163	210	214	425		326		111	112	146	153	220	222	221	229	301	318	440	
G234	200	203	245	248	157	163	210	211	423		310	326	110	112	153	154	218		221	227	301	318	443	446
G235	203		226	257	139		213		423		326		114	119	146	158	207	220	227		314	318	428	
G236	203	214	245	257	163		210	214	423		323	326	103	114	146	153	220	230	227		293		440	446
G237	203	214	226	245	157	163	210	214	411	425	326	329	112	114	146		216	222	227		293		422	428
G238	203	237	257		139		210	214	423	425	313	326	111	123	153	158	220		227	229	303	320	434	440
G239	203	214	226		157	163	213	214	423	425	316	326	110	118	146	154	220		227		293		428	440
G240	203		226		139		214		423		326		103	114	146	153	216	220	227	229	293		440	446
G241	203	216	226	257	152		214		423		316	326	116		153	154	212	220	227		301	318	411	428

Sample	Marker1		Marker2		Marker3		Marker4		Marker5		Marker6		Marker7		Marker8		Marker9		Marker10		Marker11		Marker12	
G242	214	216	210	226	139		210		423		326	329	111	117	146	162	218		227	229	312	320	425	463
G243	214	216	226	245	139		214		423		316	326	103	110	137	154	224		227		306	318	428	431
G244	203	214	226	245	152		213		423	425	316	326	104	116	137	154	212	220	227		293		411	428
G245	214		226		139		210		423		326		111	118	137	162	208	218	222	229	297	312	425	428
G246	203		245	260	157	163	210	214	423		326		116	123	146	158	220	240	229	239	308	314	411	440
G247	203		226		139		210		423		326		114		156	158	212	220	221	227	293		428	446
G248	205	214	226		157	163	210		425		326		111	114	137	150	220		227		299		440	451
G249	200	214	226	236	157	167	210	214	423		326		112	116	153	158	216	220	231	239	303		428	434
G250	200	203	226	257	155	162	209	210	423		303	326	115	123	146		212		227		292		428	451
G251	216	220	226	257	157	163	210	214	423	425	326	329	110	118	137	158	216	218	227	235	299	314	411	428
G252	205	214	245		157	163	210	214	423		326		114	118	153	158	220		227	237	293		434	443
G253	203	214	226	257	163		207	210	423		316		112	116	146	166	220		227	229	293		437	440
G254	203	214	226				214		423		323	326	110	112	146	154	220	222	227	230			428	434
G255	203	212	226	257	157	163	214		423		316	326	110	114	153	154	220		227	237	318		422	451
G256	214	216	226	236	157	163	210	214					112	118	146		220	222	227	235				
G257	203	214	226		163		210		423		326	329	114		146	160	218	222	227		293		434	446
G258	203		226	257			210	214	411	423	323	326	110	112	146	156	218	220	227	229	292		411	440
G259			245								326		110		153		218		225					

Appendix B – Identity matching result data

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID	Comments and reasons
AB201	Clone ID 1	Clone ID 1	Clone ID 1	
AB231	Clone ID 1	Clone ID 1	Clone ID 1	
AB282	Clone ID 1	Clone ID 1	Clone ID 1	
AB297	Clone ID 1	Clone ID 1	Clone ID 1	
AB326	Clone ID 1	Clone ID 1	Clone ID 1	
G254	Clone ID 1	Clone ID 1	Clone ID 1	
AB228	Clone ID 2	Clone ID 2	Clone ID 2	No reference fingerprint
AB295	Clone ID 2	Clone ID 2	Clone ID 2	No reference fingerprint
AB356	Clone ID 2	Clone ID 2	Clone ID 2	No reference fingerprint
AB230	Clone ID 3	Clone ID 25	Clone ID 25	
AB285	Clone ID 3	Clone ID 25	Clone ID 25	
AB306	Clone ID 3	Clone ID 25	Clone ID 25	
AB325	Clone ID 3	Clone ID 3	Clone ID 3	
G217	Clone ID 3	Clone ID 3	Clone ID 3	
AB211	Clone ID 4	Clone ID 4	Clone ID 4	No reference fingerprint
AB240	Clone ID 4	Clone ID 13	Clone ID 13	
AB293	Clone ID 4	Clone ID 4	Clone ID 4	No reference fingerprint
AB301	Clone ID 4	Clone ID 4	Clone ID 4	No reference fingerprint
AB336	Clone ID 4	Clone ID 4	Clone ID 4	No reference fingerprint
AB370	Clone ID 4	Clone ID 13	Clone ID 13	
AB225	Clone ID 5	Clone ID 5B	Clone ID 5B	
AB273	Clone ID 5	Clone ID 44	Clone ID 44	
AB309	Clone ID 5	Clone ID 5B	Clone ID 5B	
AB352	Clone ID 5	Unique genotype	Clone ID 5	see G259 Raw allele scores does not support manual findings, human error or missing input files?
G259	Clone ID 5	Unique genotype	Clone ID 5	
AB366	Clone ID 6	Clone ID 6	Clone ID 6	
G218	Clone ID 6	Clone ID 6	Clone ID 6	
AB252	Clone ID 7	Clone ID 1	Clone ID 1	
AB346	Clone ID 7	Clone ID 7-32m	Clone ID 7-32m	
G219	Clone ID 8	Clone ID 8	Clone ID 8	No clone samples provided
AB235	Clone ID 9	Clone ID 40	Clone ID 40	
AB264	Clone ID 9	Clone ID 9	Clone ID 9	
AB320	Clone ID 9	Clone ID 26	Clone ID 26	
AB330	Clone ID 9	Unique genotype	Unique genotype	
AB363	Clone ID 9	Clone ID 9	Clone ID 9	
AB384	Clone ID 9	Clone ID 40	Clone ID 40	
G220	Clone ID 9	Clone ID 9	Clone ID 9	
AB222	Clone ID 10	Clone ID 10	Clone ID 10	
AB242	Clone ID 10	Clone ID 10	Clone ID 10	
AB286	Clone ID 10	Clone ID 10	Clone ID 10	
AB322	Clone ID 10	Clone ID 10	Clone ID 10	
AB353	Clone ID 10	Clone ID 10	Clone ID 10	

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID	Comments and reasons
G221	Clone ID 10	Clone ID 10	Clone ID 10	
G246	Clone ID 10	Clone ID 10	Clone ID 10	
AB258	Clone ID 11	Clone ID 11	Clone ID 11	
AB348	Clone ID 11	Clone ID 9	Clone ID 9	
G222	Clone ID 11	Clone ID 11	Clone ID 11	
AB227	Clone ID 12	Clone ID 12	Clone ID 12	
AB253	Clone ID 12	Clone ID 12-18m	Clone ID 12-18m	
AB319	Clone ID 12	Unique genotype	Unique genotype	
G223	Clone ID 12	Clone ID 12	Clone ID 12	
AB210	Clone ID 13	Clone ID 13	Clone ID 13	
AB249	Clone ID 13	Clone ID 13	Clone ID 13	
AB276	Clone ID 13	Clone ID 13	Clone ID 13	
AB302	Clone ID 13	Clone ID 28	Clone ID 28	
AB304	Clone ID 13	Clone ID 13	Clone ID 13	
AB349	Clone ID 13	Clone ID 13	Clone ID 13	
AB358	Clone ID 13	Clone ID 13	Clone ID 13	
G224	Clone ID 13	Clone ID 13	Clone ID 13	
AB205	Clone ID 14	Clone ID 14	Clone ID 14	
AB277	Clone ID 14	Clone ID 14	Clone ID 14	
AB296	Clone ID 14	Clone ID 14	Clone ID 14	
G225	Clone ID 14	Unique genotype	Unique genotype	
AB203	Clone ID 15	Clone ID 15	Clone ID 15	
AB254	Clone ID 15	Clone ID 15	Clone ID 15	
AB269	Clone ID 15	Clone ID 15	Clone ID 15	
AB321	Clone ID 15	Clone ID 15	Clone ID 15	
AB344	Clone ID 15	Clone ID 15	Clone ID 15	
G226	Clone ID 15	Unique genotype	Unique genotype	
G251	Clone ID 15	Unique genotype	Unique genotype	
G252	Clone ID 15	Unique genotype	Unique genotype	
AB226	Clone ID 16	Clone ID 9	Clone ID 9	
AB259	Clone ID 16	Clone ID 24	Clone ID 24	
AB283	Clone ID 16	Clone ID 36	Clone ID 36	
AB308	Clone ID 16	Clone ID 16	Clone ID 16	
AB343	Clone ID 16	Clone ID 9	Clone ID 9	
AB374	Clone ID 16	Clone ID 36	Clone ID 36	
G227	Clone ID 16	Clone ID 16	Clone ID 16	
AB213	Clone ID 17	Unique genotype	Unique genotype	
AB271	Clone ID 17	Clone ID 17	Clone ID 17	
AB379	Clone ID 17	Clone ID 17	Clone ID 17	
G228	Clone ID 17	Clone ID 17	Clone ID 17	
AB196	Clone ID 18	Clone ID 18	Clone ID 18	
AB241	Clone ID 18	Clone ID 4	Clone ID 4	No reference fingerprint
AB265	Clone ID 18	Clone ID 4	Clone ID 4	No reference fingerprint
AB337	Clone ID 18	Clone ID 12-18m	Clone ID 12-18m	
AB355	Clone ID 18	Clone ID 18	Clone ID 18	

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID	Comments and reasons
AB385	Clone ID 18	Clone ID 18	Clone ID 18	
G250	Clone ID 18	Clone ID 18	Clone ID 18	
AB215	Clone ID 19	Clone ID 19	Clone ID 19	
AB262	Clone ID 19	Clone ID 19	Clone ID 19	
AB275	Clone ID 19	Clone ID 19	Clone ID 19	
AB312	Clone ID 19	Clone ID 19	Clone ID 19	
AB342	Clone ID 19	Clone ID 19	Clone ID 19	
AB361	Clone ID 19	Clone ID 19	Clone ID 19	
G229	Clone ID 19	Clone ID 19	Clone ID 19	
AB220	Clone ID 20	Clone ID 20	Clone ID 20	
AB238	Clone ID 20	Clone ID 20	Clone ID 20	
AB280	Clone ID 20	Clone ID 20	Clone ID 20	
AB334	Clone ID 20	Clone ID 20	Clone ID 20	
AB359	Clone ID 20	Clone ID 20	Clone ID 20	
AB377	Clone ID 20	Clone ID 20	Clone ID 20	
G257	Clone ID 20	Clone ID 20	Clone ID 20	
AB214	Clone ID 21	Clone ID 21	Clone ID 21	
AB216	Clone ID 21	Unique genotype	Unique genotype	
AB244	Clone ID 21	Clone ID 21	Clone ID 21	
AB305	Clone ID 21	Clone ID 21	Clone ID 21	
AB316	Clone ID 21	Clone ID 21	Clone ID 21	
G230	Clone ID 21	Clone ID 21	Clone ID 21	
AB195	Clone ID 22	Clone ID 22	Clone ID 22	
AB232	Clone ID 22	Clone ID 22	Clone ID 22	
AB324	Clone ID 22	Unique genotype	Unique genotype	
AB327	Clone ID 22	Unique genotype	Clone ID 10	Raw allele scores does not support manual findings, human error or missing input files?
G231	Clone ID 22	Clone ID 22	Clone ID 22	
AB197	Clone ID 23	Clone ID 23	Clone ID 23	
AB233	Clone ID 23	Unique genotype	Clone ID 23	Raw allele scores does not support manual findings, human error or missing input files?
AB272	Clone ID 23	Clone ID 23	Clone ID 23	
AB284	Clone ID 23	Clone ID 23	Clone ID 23	
AB315	Clone ID 23	Clone ID 23	Clone ID 23	
AB328	Clone ID 23	Clone ID 23	Clone ID 23	
AB345	Clone ID 23	Clone ID 23	Clone ID 23	
AB372	Clone ID 23	Clone ID 23	Clone ID 23	
AB375	Clone ID 23	Clone ID 23	Clone ID 23	
G232	Clone ID 23	Clone ID 23	Clone ID 23	
AB204	Clone ID 24	Clone ID 24	Clone ID 24	
AB243	Clone ID 24	Clone ID 24	Clone ID 24	
AB267	Clone ID 24	Clone ID 24	Clone ID 24	
AB298	Clone ID 24	Clone ID 24	Clone ID 24	

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID	Comments and reasons
AB338	Clone ID 24	Clone ID 24	Clone ID 24	
AB373	Clone ID 24	Clone ID 24	Clone ID 24	
G233	Clone ID 24	Clone ID 24	Clone ID 24	
G258	Clone ID 25	Clone ID 25	Clone ID 25	
AB206	Clone ID 26	Clone ID 26	Clone ID 26	
AB237	Clone ID 26	Clone ID 26	Clone ID 26	
AB274	Clone ID 26	Clone ID 26B	Clone ID 26B	
AB307	Clone ID 26	Clone ID 26B	Clone ID 26B	
AB333	Clone ID 26	Clone ID 23	Clone ID 23	
AB383	Clone ID 26	Unique genotype	Unique genotype	
G234	Clone ID 26	Clone ID 26	Clone ID 26	
G249	Clone ID 26	Clone ID 26B	Clone ID 26B	
AB212	Clone ID 27	Clone ID 31	Clone ID 31	
AB246	Clone ID 27	Clone ID 31	Clone ID 31	
G235	Clone ID 27	Clone ID 27	Clone ID 27	
G247	Clone ID 27	Clone ID 27B	Clone ID 27B	
G248	Clone ID 27	Unique genotype	Unique genotype	
AB217	Clone ID 28	Clone ID 28	Clone ID 28	
AB247	Clone ID 28	Clone ID 28-33m	Clone ID 28-33m	
AB288	Clone ID 28	Clone ID 28	Clone ID 28	
AB323	Clone ID 28	Clone ID 28-33m	Clone ID 28-33m	
AB339	Clone ID 28	Clone ID 23	Clone ID 23	
AB381	Clone ID 28	Clone ID 13	Clone ID 13	
G236	Clone ID 28	Clone ID 28	Clone ID 28	
AB255	Clone ID 29	Unique genotype	Unique genotype	
AB313	Clone ID 29	Clone ID 29	Clone ID 29	
AB378	Clone ID 29	Clone ID 29	Clone ID 29	
G237	Clone ID 29	Clone ID 29	Clone ID 29	
AB376	Clone ID 30	Clone ID 30	Clone ID 30	
G238	Clone ID 30	Clone ID 30	Clone ID 30	
AB218	Clone ID 31	Clone ID 31	Clone ID 31	
AB257	Clone ID 31	Clone ID 31	Clone ID 31	
AB292	Clone ID 31	Clone ID 31	Clone ID 31	
AB314	Clone ID 31	Clone ID 25	Clone ID 25	
G253	Clone ID 31	Clone ID 31	Clone ID 31	
AB260	Clone ID 32	Clone ID 32	Clone ID 32	
AB279	Clone ID 32	Clone ID 32	Clone ID 32	
AB351	Clone ID 32	Clone ID 7-32m	Clone ID 7-32m	
AB367	Clone ID 32	Clone ID 32	Clone ID 32	
G239	Clone ID 32	Clone ID 32	Clone ID 32	
AB207	Clone ID 33	Clone ID 2	Clone ID 2	No reference fingerprint
AB245	Clone ID 33	Clone ID 2	Clone ID 2	No reference fingerprint
AB290	Clone ID 33	Clone ID 28-33m	Clone ID 28-33m	
AB369	Clone ID 33	Clone ID 2	Clone ID 2	No reference fingerprint

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID	Comments and reasons
G240	Clone ID 33	Clone ID 33	Clone ID 33	
AB223	Clone ID 34	Clone ID 34	Clone ID 34	No reference fingerprint
AB341	Clone ID 34	Clone ID 34	Clone ID 34	No reference fingerprint
AB382	Clone ID 34	Clone ID 34	Clone ID 34	No reference fingerprint
AB208	Clone ID 35	Unique genotype	Unique genotype	
AB219	Clone ID 36	Clone ID 36	Clone ID 36	
AB266	Clone ID 36	Clone ID 36	Clone ID 36	
AB310	Clone ID 36	Clone ID 36	Clone ID 36	
AB318	Clone ID 36	Clone ID 36	Clone ID 36	
AB332	Clone ID 36	Clone ID 36	Clone ID 36	
AB380	Clone ID 36	Clone ID 36	Clone ID 36	
AB202	Clone ID 37	Clone ID 37	Clone ID 37	
AB236	Clone ID 37	Clone ID 37	Clone ID 37	
AB278	Clone ID 37	Clone ID 37	Clone ID 37	
AB331	Clone ID 37	Clone ID 37	Clone ID 37	
AB354	Clone ID 37	Clone ID 37	Clone ID 37	
AB229	Clone ID 38	Clone ID 38	Clone ID 38	
AB234	Clone ID 38	Clone ID 38	Clone ID 38	
AB270	Clone ID 38	Clone ID 38	Clone ID 38	
AB291	Clone ID 38	Clone ID 38	Clone ID 38	
AB294	Clone ID 38	Clone ID 38	Clone ID 38	
AB311	Clone ID 38	Clone ID 27B	Clone ID 27B	
AB329	Clone ID 38	Clone ID 38	Clone ID 38	
AB360	Clone ID 38	Clone ID 38	Clone ID 38	
G241	Clone ID 38	Clone ID 38	Clone ID 38	
AB209	Clone ID 39	Clone ID 39	Clone ID 39	
AB248	Clone ID 39	Clone ID 39	Clone ID 39	
AB289	Clone ID 39	Clone ID 39	Clone ID 39	
AB340	Clone ID 39	Clone ID 39	Clone ID 39	
AB364	Clone ID 39	Clone ID 39	Clone ID 39	
G256	Clone ID 39	Clone ID 39	Clone ID 39	
AB199	Clone ID 40	Clone ID 40	Clone ID 40	
AB256	Clone ID 40	Clone ID 40	Clone ID 40	
AB263	Clone ID 40	Clone ID 40	Clone ID 40	
AB300	Clone ID 40	Clone ID 40	Clone ID 40	
AB350	Clone ID 40	Clone ID 40	Clone ID 40	
AB371	Clone ID 40	Clone ID 40	Clone ID 40	
G242	Clone ID 40	Clone ID 40	Clone ID 40	
AB221	Clone ID 41	Clone ID 41	Clone ID 41	
AB250	Clone ID 41	Clone ID 41	Clone ID 41	
AB281	Clone ID 41	Clone ID 41	Clone ID 41	
AB299	Clone ID 41	Clone ID 41	Clone ID 41	
AB347	Clone ID 41	Clone ID 41	Clone ID 41	
AB362	Clone ID 41	Clone ID 41	Clone ID 41	

Sample ID	Assumed Clonal ID	GenoSonic assigned clonal ID	Manual assigned clonal ID	Comments and reasons
G243	Clone ID 41	Clone ID 41	Clone ID 41	
AB200	Clone ID 42	Clone ID 42	Clone ID 42	
AB251	Clone ID 42	Clone ID 42	Clone ID 42	
AB287	Clone ID 42	Clone ID 42	Clone ID 42	
AB317	Clone ID 42	Clone ID 42	Clone ID 42	
AB368	Clone ID 42	Clone ID 42	Clone ID 42	
G255	Clone ID 42	Clone ID 42	Clone ID 42	
AB198	Clone ID 43	Clone ID 43	Clone ID 43	
AB239	Clone ID 43	Clone ID 43	Clone ID 43	
AB335	Clone ID 43	Clone ID 43	Clone ID 43	
AB365	Clone ID 43	Clone ID 43	Clone ID 43	
G244	Clone ID 43	Clone ID 43	Clone ID 43	
AB224	Clone ID 44	Clone ID 44	Clone ID 44	
AB268	Clone ID 44	Clone ID 44	Clone ID 44	
AB303	Clone ID 44	Clone ID 44	Clone ID 44	
AB357	Clone ID 44	Clone ID 44	Clone ID 44	
G245	Clone ID 44	Clone ID 44	Clone ID 44	

Appendix C – Relatedness Tree

