CRANFIELD UNIVERSITY

CRANFIELD HEALTH

MSc by Research

Academic Year 2005 - 2007

Nikol Simecek

DEVELOPMENT OF A DATABASE WITH WEB-BASED USER INTERFACE FOR TAQMAN ASSAY DESIGN

Supervisor: Dr Conrad Bessant

Date of Presentation: 26th January 2007

ABSTRACT

TaqMan RT-PCR (reverse transcription-polymerase chain reaction) is a technique used to measure the relative gene expression in a biological sample and is one of the core technologies used by the Molecular Pathology and Toxicology (MPT) Group at GlaxoSmithKline. Conducting TaqMan experiments is a complex process which involves the design of a TaqMan assay specific to a gene of interest. A wealth of data has been generated during assay design, but systems are not currently available to readily share this data within the MPT group.

There is a need for a central data storage repository so that data associated with assay design can be organised efficiently and rapidly accessed. Experiments are conducted within limited timeframes and resource is often limited so this would be of great benefit to the MPT group.

This thesis describes the development of a database to house data associated with TaqMan assay design, software to populate the database with minimal user interaction and a web based CGI application for members of the MPT group to query and submit data to the database. Finally, the output from testing the software is provided and discussed.

ACKNOWLEDGEMENTS

GlaxoSmithKline

I would like to thank the Investigative Preclinical Toxicology group at GlaxoSmithKline for sponsoring this project. In particular, I would like to thank Stephanie Gresham and Dr. Jon Lyon for their support throughout this project.

Cranfield University

I would like to thank Dr. Conrad Bessant for supervising this project, providing invaluable advice where needed and for reading this thesis. I would also like to thank the members of the Bioinformatics group for providing support when required.

TABLE OF CONTENTS

FIGURES

TABLES

Chapter 1. Introduction and Literature Review

The role of Safety Assessment at GlaxoSmithKline is to support the selection of targets and molecules with the lowest probability of toxicity, thus enabling safe clinical trials and successful compound registration. The Molecular Pathology and Toxicology (MPT) group within Safety Assessment use a number of Molecular Biology techniques to investigate toxic mechanisms of compounds in development.

One of the principal technologies used within the MPT group is TaqMan RT-PCR (reverse transcription-polymerase chain reaction). It is a targeted approach used to assess the expression of specific genes within a sample and it is one of the most sensitive techniques for mRNA (messenger RNA) detection and quantification. Its main applications within MPT are:

- Elucidating mechanisms of action of a particular toxicity
- Use as a screen for assessing particular types of toxicity
- Confirmation of microarray-derived gene expression data

1.1 Polymerase Chain Reaction

The Polymerase Chain Reaction (PCR) is considered to be one of the most important techniques in molecular biology and is used to amplify a specific target sequence of DNA within a sample. DNA is denatured into two separate strands by heating to 95°C. The temperature is lowered and two oligonucleotides (often referred to as forward and reverse primers) complementary to each end of the target sequence of interest bind to the template DNA (Creighton, 2005). A new DNA strand, complementary to the template is synthesised by the enzyme DNA polymerase (Dale, 2003). The process of denaturation, annealing and extension is then repeated using the newly formed DNA target as a template each time. During each round of PCR, the DNA target doubles in quantity resulting in an exponential increase of the target DNA sequence (Creighton, 2005).

The PCR reaction can be split into 3 phases; exponential, linear and plateau. During the exponential phase the reaction is 100% efficient and there is exact doubling of product. Towards the end of the reaction, during the linear phase, reagents are being consumed and the reaction begins to slow. Finally, during the plateau phase the reaction eventually ceases.

In addition to DNA, it is also possible to amplify mRNA using PCR. Reverse transcriptase is used to convert the mRNA sequence into either single stranded DNA or a double stranded DNA/RNA hybrid (Creighton, 2005). This forms the basis of many technologies used for monitoring gene expression levels in a sample.

1.2 TaqMan RT PCR

In traditional PCR methods, amplification is only detected in the final phase of the PCR reaction (plateau phase) usually by agarose gel electrophoresis. Quantification of starting material is therefore limited due to the poor sensitivity and resolution of this technique.

Real time PCR (RT-PCR) measures the accumulation of products during the PCR reaction. During the exponential phase of PCR, there is a quantitative relationship between starting material and PCR product. This feature of the PCR reaction enables the accurate quantification of starting material using real time PCR.

The 5’ nuclease assay is one of a number of real-time PCR methods. A TaqMan probe is included in the PCR reagent mix in addition to the primers used in a traditional PCR reaction. The probe, labelled with a fluorescent reporter at the 5’ end and quencher dye at 3’ end, anneals to the target between the forward and reverse primers. Fluorescence of the reporter is suppressed by the quencher by fluorescence resonance energy transfer (FRET), due to the close proximity of these dyes. As the DNA polymerase moves along the template, the probe is cleaved between the reporter and quencher dye by the 5’ nuclease activity of the polymerase. The reporter dye emits fluorescence as it is no longer suppressed by the quencher dye (Hawrami & Bruer, 1998). This is summarised in Figure 1. Cleavage will only occur if the probe is hybridised to its specific target. Reporter fluorescence increases during each PCR cycle and is proportional to the amount of PCR product. Fluorescence levels are measured at each PCR cycle by a detector and the starting amount of template can be calculated (Hawrami & Bruer, 1998).

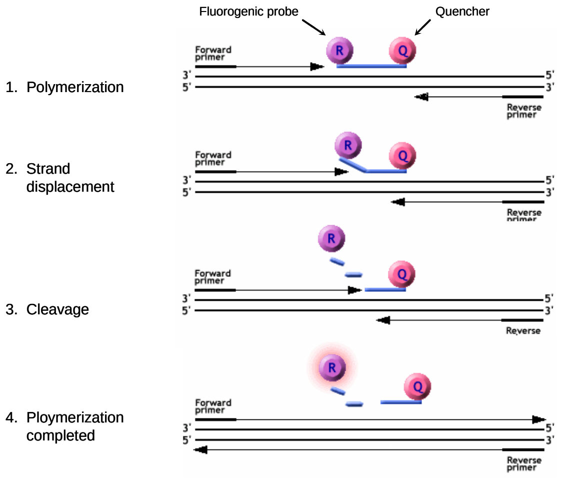


Figure 1 TaqMan Chemistry

1.3 TaqMan RT-PCR Experimental Protocol

The goal of a TaqMan experiment is to measure the expression levels of one or more genes in samples taken from an animal or cell culture experiment. There are three components that comprise a TaqMan experiment:

- Assay Design
- Sample Preparation
- Gene Expression Measurement

If an assay is not already available for the desired gene and species, a new assay consisting of a forward primer, probe and reverse primer should be designed. This process involves using software to design primer and probes specific to a genomic sequence. On receipt of the primer and probe sequence, the oligonucleotides are synthesised by vendors such as Proligo (www.proligo.com). They are then ready to use in a TaqMan experiment.

Before gene expression can be measured in a tissue or cell sample, a number of processes need to be carried out. The first step is the isolation of RNA. Once RNA has been isolated, the quantity and purity of RNA is determined using a spectrophotometer. The integrity of each RNA sample is checked by running each sample on a gel. Since RNA is unstable and prone to degradation, an aliquot of RNA is reverse transcribed or 'copied' into cDNA by following a standard protocol. Once the reaction is complete, levels of gene expression can be assessed in the cDNA sample. The expression levels of the selected genes are measured in the samples of interest by applying an aliquot of each assay (forward primer, reverse primer and probe) to an aliquot of sample (cDNA derived from an animal or cell culture experiment) in one well of a 96 or 384 well plate. The position of each sample and assay in the plate is recorded either in a laboratory notebook or by manually entering the details in an Excel spreadsheet. The plate is then placed inside a sequence detector which measures fluorescence emission in each well during each PCR cycle. This data, from which gene expression levels can be inferred, is collected by the Applied Biosystems sequence detection software (SDS). Data from each plate is contained within one SDS file and can be exported to Excel for further analysis. Copy numbers for each gene are calculated using a standard curve and statistical analysis of this data is performed using TaqMan Toolkit. TaqMan Toolkit is an add-on for Excel, developed in-house, for analysing data from genomic TaqMan experiments. Analysis methods available in this toolkit are Analysis of Variance, Analysis of CoVariance and Principal Components Analysis (PCA). Once statistically significant data has been generated it is often compared to other parameters such as clinical pathology data. Literature is reviewed and a number of pathway mapping tools are used to assist with interpreting the gene changes identified.

1.4 The Application of Bioinformatics: Databases

There is a wealth of data generated in any one TaqMan experiment and currently within the MPT group at GlaxoSmithKline this data is stored on users' PCs or in laboratory notebooks. It would benefit the group if data associated with the design of TaqMan assays was organised efficiently and stored in one central repository. Time would be saved since users would not have to search for assay availability. Also duplication of effort could be minimized since the user could quickly assess whether an assay had already been designed. As a result, the productivity of the group would increase which is particularly important in a business environment.

Databases provide an ideal solution as data is organised, easily accessed and updated.

1.4.1 Database types

There are several types of database model available for the storage of data. These include flat-file, hierarchical, network, relational and object-orientated database models (Stephens and Plew, 2001).

A flat-file database is the most basic type of database and is simply a collection of files stored in an orderly manner. The file, usually in text format, consists of data with a delimiter to separate one field from another (Buchanan, 2002). There are many disadvantages with this model for example the user is required to know the physical location of the data. It is also impractical to perform manual searches and so programs are required to access the data. With larger data-sets using a flat file system becomes inefficient (Gibas & Jambeck, 2001. Stephens and Plew, 2001).

Hierarchical databases consist of tables populated with data arranged in hierarchies similar in structure to family trees or organisational diagrams. The parent or root table at the top of the tree has child tables with related data below it (Stephens and Plew, 2001). A parent table can have many child tables but each child record must have only one parent record (King, 2002). The main advantage of these databases is that data can be quickly accessed, navigation around the database is relatively easy and groups of data can be related to each other (King, 2002. Stephens and Plew, 2001).

Network databases are an improved version of the hierarchical database model as they allow for each child table to have many parent tables. Child tables can be easily accessed without the need to access the parent table first. This ensures that the data is retrieved in a more efficient manner (Stephens and Plew, 2001). The main disadvantage with network databases is that modifications to their structure such as adding new tables or fields is difficult (King, 2002).

Relational databases, originally proposed by Dr E F Codd in the 1970s, are more flexible and provide a powerful way of organising and accessing data. A relational database consists of a group of related tables with the tables categorised in a logical manner. The tables consist of a number of rows and columns. Each row (record) contains a collection of data items (fields) with each item described by the column (attribute). The tables are related to each other through common column values called keys (Buchanan, 2002; Stephens and Plew, 2001).

Advantages of the relational database model include reduced data redundancy which allows for more economical data storage and ease of database modification and easy retrieval of data. As a result the relational database model is the most common type of database in use today (Buchanan, 2002).

Object orientated databases are databases that are defined, stored and accessed, using an object orientated programming language such as Java. Their development is still in its infancy and standards for this model have not been completely refined (Stephens and Plew, 2001).

The features of a relational database model described in this section make it an ideal choice for solving the data storage issues faced by the MPT group. For example, flexibility is important in an environment where procedures and processes are constantly evolving. The relational database model is well established and has many advantages over the other models including reliability, wealth of vendors, ease of data manipulation and definition and extensive programming support. These features are important in the context of this project for example the automation of data upload which will require the development of programs to interface with the database.

The following sections will focus on tools and methods associated with relational databases.

1.4.2 Database Integrity

It is extremely important to ensure that the integrity of a database is maintained so that the information produced is accurate and of the highest quality. Errors can arise as a result of incorrect data entry, incomplete data modification, unintentional data alteration and multiple users attempting to alter the same data. Data integrity is implemented by having column or table constraints (rules) so that inappropriate values are not entered into the database (Stein, 2003). In addition, default values can be used in order to minimise manual input of data. Referential integrity is enforced by the use of keys so that tables can be related to each other. A primary key is a column with unique entries so that records can be differentiated from each other (Stein, 2003). In addition to its primary keys, a table may have fields that correspond to keys in other tables. These are referred to as foreign keys. Primary and foreign keys can comprise of more than one column in some cases. It is important to maintain referential integrity by ensuring that the relationships between tables remain consistent. For example, a table’s foreign key value must match the value of the primary key in the table of which it is related to.

1.4.3 Relationships

As mentioned in the previous section, tables in a relational database are related to each other by primary and foreign keys. An example is illustrated in Figure 2.

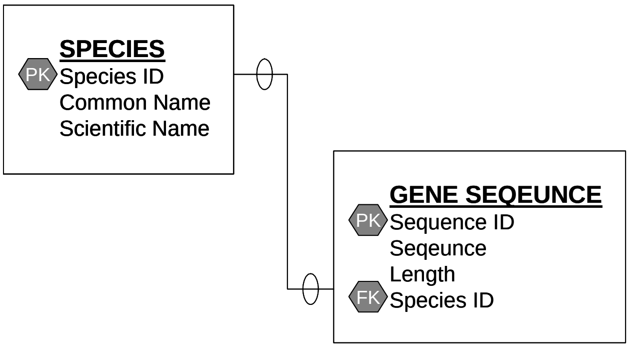


Figure 2 Example of two tables related by keys

Relationships can be classified as one-to-one, one-to-many or many-to-many (Rolland, 1998). In a one-to-one relationship a key value appears only once in the related table, whereas in a one-to-many relationship a key value can appear many times in the related table. In many-to-many relationships, a key value can appear many times in the related table and vice versa. As many-to-many relationships can cause problems when a database is implemented, they are usually broken down into a series of one-to-many relationships.

1.4.4 Database design

Planning the database carefully will ensure that the final product is efficient, flexible and easy to manage and maintain (Meloni, 2002). In terms of this project, flexibility is an important consideration. Although the database designed for this project will house data associated with TaqMan assay design, it should allow for expansion. For example, storing data generated from other areas of the TaqMan process such as sample preparation and gene expression measurement (section 1.3) would be invaluable. This would allow users to perform powerful biological queries in support of high priority projects. Also within a research environment, processes are constantly changing so a flexible database design is of great importance.

There are three phases of the design process; requirements analysis, data modeling and normalisation. Requirements analysis is the process of establishing database requirements. This is done by interviewing the end users and analysing current processes to capture the types of data generated.

The next phase is database modelling whereby the data is visually represented for example in the form of an entity relationship diagram. The final phase of the design process is normalisation which is carried out in order to reduce data redundancy (Stephens and Plew, 2001). This is described in more detail in section 1.4.5.

A well designed database should;

- be functional
- accurately represent the business's data
- be easy to use/maintain
- be secure
- have reduced data redundancy
- be easily backed up

1.4.5 Normalisation

Fundamental to the modelling and design of a relational database is the process of normalisation which aims to minimise data redundancy. Data redundancy, which refers to the duplication of data, should be kept to a minimum since it can result in unnecessary space being used, ambiguity, inconsistency and wasted effort (Stephens and Plew, 2001).

Normalisation consists of a set of rules that should be followed to reduce redundancy with each rule improving on the previous rule (Peitzsch, 2003). The rules are as follows:

- First Normal Form
  - The data must be divided up into logical groups i.e. entities
  - The entities should not contain repeating data
- Second Normal Form
  - The rules of the first normal form must be met
  - There should be no fields in a table that are not related to the primary key. These should be placed in a new table
- Third Normal Form
  - The rules of the first and second normal form must be met
  - No attributes depend on other non-key attributes i.e. there should be no fields in the table that can be broken down further

There are additional normal forms, however these are currently mainly theoretical and scarcely used. In addition to reducing data redundancy, the normalisation process aims to minimise null values. Null values are difficult to interpret since they can have one of a number of definitions, for example null can either mean the information does not exist, has not been entered or is not applicable. Normalisation also aims to prevent loss of information, known as deletion anomaly (Rolland, 1998), since it is possible to lose data unintentionally when a row in a table is deleted.

1.4.6 Relational Database Management Systems

Database Management Systems (DBMS) are software packages that allow for the access and storage of data. Examples of commercial Relational Database Management Systems (RDBMS) include Oracle, Microsoft Access, Microsoft SQL server and examples of open source products include PostgreSQL and MySQL (Stein, 2003).

DBMSs allow multiple users to access the data simultaneously using a query language (Stephens and Plew, 2001). Security can be enforced by limiting who can access and/or update the database and they also provide support for backup and recovery (Stein, 2003).

1.4.7 Structured Query Language

Structured Query Language (SQL) is a standardised query language used to communicate with relational databases (Buchanan, 2002). It is a declarative language i.e. it does not describe *how* data should be accessed but only *what* data to access.

SQL allows data to be modified, deleted and retrieved (Peterson, 2002) and consists of three sub languages (Stephens and Plew, 2001):

- Data Definition Language (DDL), used to define database structure
- Data Manipulation Language (DML), used to modify data
- Data Query Language (DQL), which allows for retrieval of data

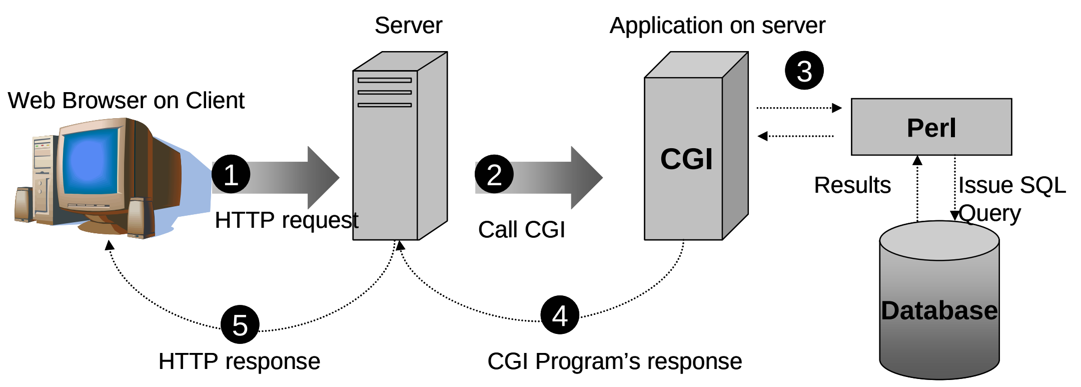
1.4.8 Database User Interface

As with most laboratory-based groups, knowledge of SQL within the MPT group is virtually non existent. It is therefore necessary to produce a well designed application whereby the users can easily query and load data into the database.

A popular approach is the design of networked database applications. Many biological databases are accessible via a Web interface so that data can be easily accessed and shared throughout the scientific community. The following examples are routinely used within the MPT group:

- Genetic Sequence Data Bank (GenBank). This database contains all publicly released genetic sequence data and can be accessed via the National Center for Biotechnology Information. (NCBI) Entrez retrieval system (www.ncbi.nlm.nih.gov/).
- PubMed. This provides access to citations from biomedical literature and is also available via the NCBI Entrez retrieval system.
- EMBL nucleotide sequence database. This is Europe's primary sequence resource and can be accessed via www.ebi.ac.uk/embl
- Gene Ontology Database. This can be searched using AmiGO (www.godatabase.org/cgi-bin/amigo/go.cgi). This interface provides access to genes, proteins and gene ontologies which are a description of how gene products behave in a cellular context.

The database resides on a server and is accessed by the client through a network using a familiar web browser such as Microsoft Internet Explorer or Mozilla Firefox. The Common Gateway Interface (CGI) is commonly used for Web servers to interact dynamically with users. It allows for external programs written using languages such as Perl, PHP and Java to run on a Web server and when a request is made, the server executes the CGI program. The request is transmitted to the database and the results are returned and displayed to the client as HTML output (Guelich *et al*, 2000). This is illustrated in Figure 3.



Abbreviations	
HTTP	Hypertext Transfer Protocol
CGI	Common Gateway Interface
PERL	Practical Extraction and Report Language

(Adapted from Guelich *et al*, 2000)

Figure 3 How a CGI application is executed

1.5 Project Aims

The aims of this project are:

- The development and implementation of a well designed database to house the data collected during the design of TaqMan assays. This will be achieved by:
    - Establishing the goals and objectives of the database

  - Analysing the assay design process so that data types may be identified
  - Generating an Entity Relationship schema
  - Implementing the schema using an appropriate Relational Database Management System
- The development of software to automate the upload of data into the database and an easy to use application for non SQL specialists to query the database. This will be achieved by:
  - Identifying and carefully planning the program requirements
  - Installing the appropriate software
  - Regular testing of the programs during development

Chapter 2. Database Design and Implementation

The first few sections of this chapter focus on the design phase of the database which consisted of defining database goals and objectives, requirements analysis and entity relationship modeling. The remainder of the chapter outlines database implementation from installing the relational database management system to creating the tables.

2.1 Rationale for developing a database

As outlined in section 1.4 there is currently no central storage facility within the MPT group for the data generated during a TaqMan RT-PCR experiment. On certain occasions this has resulted in duplication of effort or loss of data. At present scientists manually search for data which is extremely time consuming and can also result in biologically relevant data being overlooked. It is evident that there is a need to develop a database system to house the data generated since it will efficiently organise all the data and enable the end user to quickly search and access data of interest. The time saved and the increased accuracy would greatly benefit the MPT group particularly as time and resources are limited.

For the purposes of this project, a database was developed to house data derived from the TaqMan assay design process. Although there is a wealth of data generated during sample preparation and gene expression measurement, limiting the database to encompass only TaqMan assay data ensured that development and testing of the database and the accompanying software was a manageable task.

2.2 Database Goals and Objectives

The goals and objectives of the database are listed below:

- To consolidate all TaqMan assay data that exists within the MPT group into the database and to house any subsequent data generated during the design of TaqMan assays.
- The database should be flexible to allow for future expansion. For example it will be of great value to the end-user if data associated with sample preparation and gene expression measurement is included in future.
- The relational database model will be used. As outlined in section 1.4.1 relational databases have many advantages over other database models. The structure of relational databases can be easily modified, data can be accessed quickly and data integrity can be implemented. The database will therefore be implemented using an appropriate Relational Database Management System (RDBMS).
- The database should have integrity, i.e. the data should be accurate (data integrity) and data should be consistent between related tables (referential integrity)
- Data redundancy will be minimised through the process of normalisation.
- Where possible, data gathering and population of the database will be automated to ensure that user input is minimised thus reducing the potential for error

2.3 Requirements analysis

The first stage of designing the database was to thoroughly analyse the processes involved in designing a TaqMan assay. This was done with a view to identify the data generated or required during this process. Scientists, who would ultimately be the end users, were interviewed to ensure their needs were captured in the database design. A description of the process is outlined in the next section.

2.3.1 TaqMan assay design

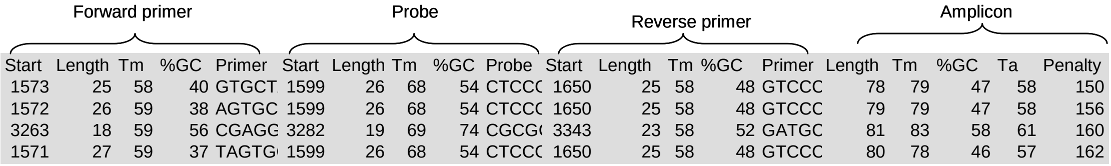
The initial step in designing a TaqMan assay is to select a gene sequence for the gene and species of interest. There are 3 main repositories for known genetic sequences. These are the U.S National Center for Biotechnology Information Genetic Sequence Data Bank (GenBank), European Molecular Biology Laboratory (EMBL) and the DNA databank of Japan. Each contains almost identical information due to international cooperative agreements. A gene sequence representing the gene of interest is retrieved as a text file from one of these nucleotide sequence databases along with its accession[*] number. Additional data is also retrieved from this source including the 'official'[†] gene name, gene description, a list of gene synonyms and the scientific and common name for species.

The gene sequence is imported into the Applied Biosystems primer design software,

[*] Accession numbers are unique identifiers of sequences within publicly available sequence databases

[†] Often there are many names used to describe a single gene within the public domain, however a gene has an 'official' name which is determined by recognised committees such as Human Gene Nomenclature Committee (HGNC), International Committee on Standardized Genetic Nomenclature for Mice and Rat Genome and Nomenclature Committee

Primer Express v.2.1 which automatically generates a list of 200 candidate assays. The list of assays can be exported by the user as a text file to any specified directory. The text file always consists of a header row and 200 records, an example of which can be found on the accompanying CD-ROM. The format of the exported file always remains consistent as there are no options within the software to modify the data export parameters. A description of the data contained within the text file is outlined below along with an example of the first 5 lines of a text file shown in Figure 4.



Forward primer					Probe					Reverse primer					Amplicon				
Start	Length	Tm	%GC	Primer	Start	Length	Tm	%GC	Probe	Start	Length	Tm	%GC	Primer	Length	Tm	%GC	Ta	Penalty
1573	25	58	40	GTGCT	1599	26	68	54	CTCCC	1650	25	58	48	GTCCC	78	79	47	58	150
1572	26	59	38	AGTGC	1599	26	68	54	CTCCC	1650	25	58	48	GTCCC	79	79	47	58	156
3263	18	59	56	CGAGG	3282	19	69	74	CGCG	3343	23	58	52	GATGC	81	83	58	61	160
1571	27	59	37	TAGTG	1599	26	68	54	CTCCC	1650	25	58	48	GTCCC	80	78	46	57	162

Figure 4 Example of the first 5 records in a text file exported from Primer Express

- **Start** refers to the starting position of the primer or probe relative to the entire gene sequence used for assay design
- **Length** refers to the length, in bases, of the primer, probe or amplicon
- **Tm** is the melting temperature of the primer or probe. This is the temperature at which 50% of the oligonucleotides are in double-stranded conformation and 50% are single stranded.
- **%GC** is the proportion of G's and C's within the primer, probe or amplicon
- **Ta** is the annealing temperature of a DNA fragment
- **Penalty** is a number calculated by the Primer Express software to reflect the number of criteria a TaqMan assay meets. The lower the number, the more criteria are met which indicates that the assay is more likely to succeed. A full

description of how the penalty score is calculated can be found in section B of the Primer Express V2.0 user manual. This section has been saved to the attached CD-ROM for reference purposes and is entitled Calculating Penalty Scores.pdf

Once the assays have been exported, the user then selects an assay that is able to meet a number of recommended design criteria outlined by Applied Biosystems. The sequences of the primers and probes belonging to the selected assay are submitted to a vendor so that they may be synthesised.

A flow diagram of the assay design process is shown in Figure 5 along with the data collected at each stage of the process where applicable.

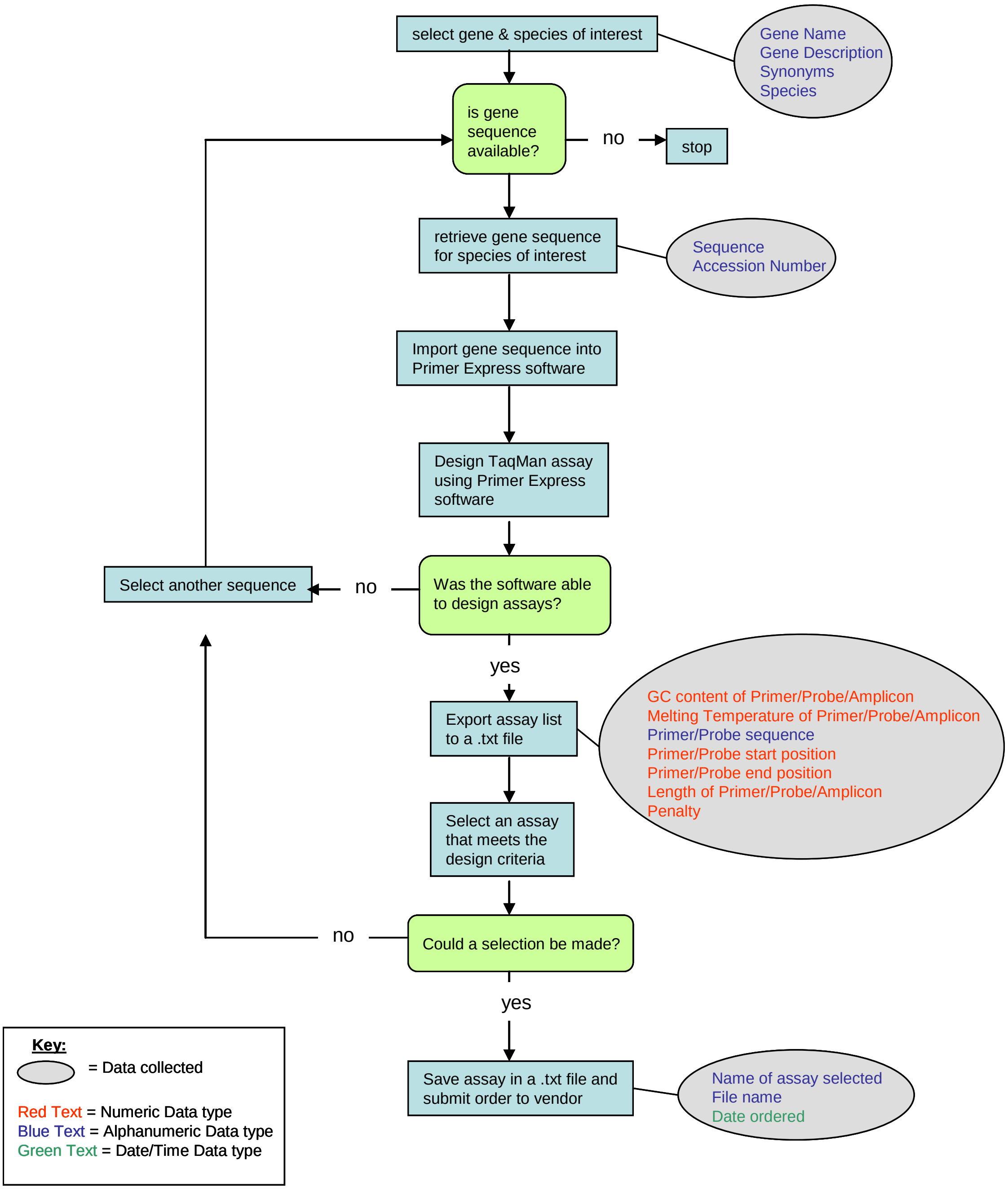

select gene & species of interest
Gene Name
Gene Description
Synonyms
Species
is gene sequence available?
no
stop
retrieve gene sequence for species of interest
Sequence
Accession Number
Import gene sequence into Primer Express software
Design TaqMan assay using Primer Express software
Select another sequence
no
Was the software able to design assays?
yes
Export assay list to a .txt file
GC content of Primer/Probe/Amplicon
Melting Temperature of Primer/Probe/Amplicon
Primer/Probe sequence
Primer/Probe start position
Primer/Probe end position
Length of Primer/Probe/Amplicon
Penalty
Select an assay that meets the design criteria
no
Could a selection be made?
yes
Save assay in a .txt file and submit order to vendor
Name of assay selected
File name
Date ordered
Key:
= Data collected
Red Text = Numeric Data type
Blue Text = Alphanumeric Data type
Green Text = Date/Time Data type

Figure 5 Flow Diagram of the Assay Design Process

2.4 Entity Relationship Modelling

Once all the data had been captured during the requirements analysis, the next step was to develop an Entity Relationship (E/R) model. The aim of the E/R model was to visually represent this data and the relationships that exist between it in preparation for database implementation.

2.4.1 Defining the entities, attributes and relationships

The first step in developing the E/R model was to categorise the data (attributes) captured in the requirements analysis into logical groups (entities). The data was categorised into groups shown in Table 1. Each entity was given an additional attribute in the form of a unique identifier to ensure that each record was unique.

Table 1 Entities and their attributes

Entity	Attributes
GENE	Gene ID Gene Symbol Gene Name
GENE SYNONYM	Synonym ID Synonym
GENE SEQUENCE	Gene Sequence ID Accession Gene Sequence
PRIMER PROBE	PR ID Start position Length Melting Temperature GC content Primer/Probe sequence Type(FP, RP or PRB)
SPECIES	Species ID Common Name Scientific Name
TAQMAN ASSAY	Assay ID Record Amplicon Length Penalty
TAQMAN FILE	File ID File Name Date/Time created

Once this had been carried out, the next step was to establish the relationships between the entities. There are three types of relationships that can exist between entities; one-to-one, one-to-many and many-to-many. For an explanation of these relationships refer to section 1.4.3 in the introduction. Each entity is related to another entity by their keys. The primary key is a column value within an entity that makes the row of data unique and the foreign key is a key that references the primary key in another entity.

The cardinality of each relationship is described below in Table 2.

Table 2 Description of the relationships between entities

ENTITY	ENTITY	DESCRIPTION OF RELATIONSHIP	CARDINALITY
GENE	GENE SYNONYM	One gene may have many synonyms however one gene synonym can be described by one 'official' gene name	One - to -Many
GENE	GENE SEQUENCE	Gene may have one or more sequences and a sequence must belong to one gene	One - to -Many
SPECIES	GENE SEQUENCE	A species may have many gene sequences and a gene sequence must belong to one species	One - to -Many
GENE SEQUENCE	TAQMAN FILE	A gene sequence may have many TaqMan files derived from it. A TaqMan file can only be derived from one sequence	One - to -Many
TAQMAN FILE	TAQMAN ASSAY	A TaqMan file may consist of many TaqMan assays. A TaqMan assay can only belong to one TaqMan file	One - to -Many

TAQMAN ASSAY	PRIMER/PROBE	A TaqMan assay consists of more than one primer/probe. A primer/probe can only belong to one TaqMan assay	One - to -Many

As previously described, many-to-many relationships should be resolved as they can cause confusion and are difficult to maintain. Initially there were instances where the relationship between entities was many-to-many. For example, the gene name attribute in the gene table originally included records for the 'official' gene name in addition to alternative gene names (synonyms). This meant that there was a many-to-many relationship between the gene entity and the gene sequence entity since a gene sequence could have more than one gene name and a gene name could be represented with more that one sequence. The process of normalisation eliminated such relationships and is outlined in the next section.

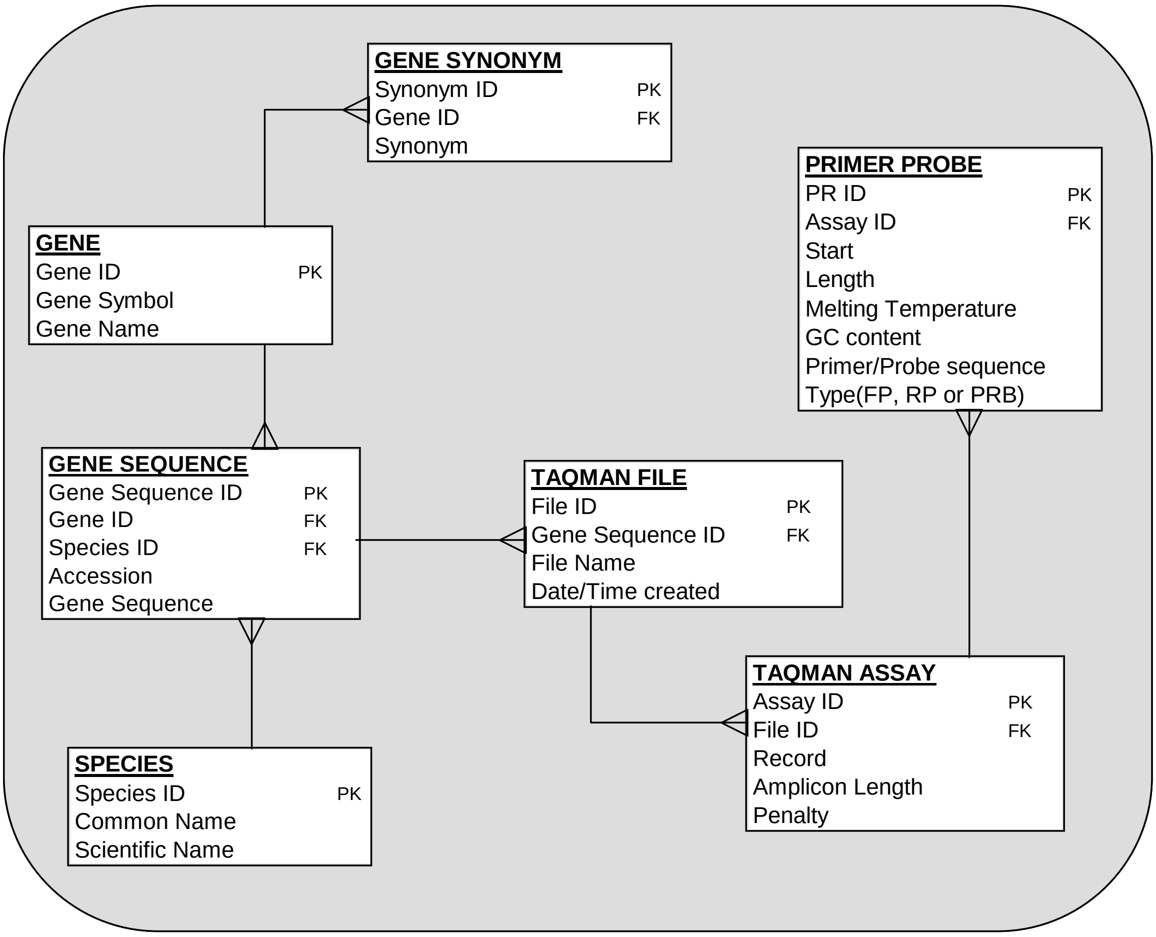
2.4.2 Normalisation

Before constructing the final E/R diagram, the entities and attributes were refined by the process of normalisation. This involved applying a set of rules to the attributes and entities in order to minimise data redundancy, increase flexibility of the database and to allow for data integrity to be easily maintained (Stephens and Plew, 2001). The final E/R model, as shown in Figure 6 meets the objectives of the 1st, 2nd and 3rd normal form.

- **1st Normal Form**. The data is divided into logical groups and there is no repeating information in the model. Each entity has a unique attribute, i.e. a primary key to ensure that each row in the table is unique. This means that the first normal form has been met.

- **2nd Normal Form**. There are no attributes within a table that are not directly related to the primary key.
- **3rd Normal Form.** No attributes depend on other non-key attributes, i.e. each attribute cannot be broken down further.

Although there are additional normal forms these are mainly theoretical and scarcely used. Ensuring that the database met the objectives of the third normal form was sufficient for the purposes of this project.



PK = Primary Key

FK = Foreign Key

Figure 6 Final Entity Relationship Diagram

2.5 Database implementation

With the database design in place, the next step was to implement the database using an appropriate relational database management system (RDBMS).

2.5.1 RDBMS installation

MySQL, distributed by MySQL AB at http://www.mysql.com, was chosen as the

RDBMS for this project because its use is widespread particularly for web applications, it is open source and it is able to interface with many programming languages (Meloni, 2002). This was important for the next phase of the project where software was developed to automatically upload data into the database. Software development will be discussed in Chapter 3. MySQL version 4.0.26 was installed on a local PC.

2.5.2 Determining the attribute data types

The entity relationship diagram shown in Figure 6 provided the framework for implementing the physical database. So that the tables could be created within MySQL, an applicable data type needed to be assigned to each of the columns. This was so that data integrity could be controlled using column constraints which safeguard against incorrect or inappropriate data being added to the database.

The data that would populate the database could be described by the following data types:

- **Unsigned Integer**. This is a normal sized integer within the range of 0 to 4294967295.
- **Text**. This is a field that can hold string data with a maximum length of 65535 characters.
- **TinyText .** This is a field that can hold a maximum of 255 characters
- **Datetime.** This is a date and time combination in the following format: YYYY-MM-DD HH:MM:SS
- **Enum.** This is an enumeration, i.e. list. A value must be selected from a list of values that has been created.

- **Timestamp.** This is date time in the following format: YYYYMMDDHHMMSS.

Each of the primary keys would be generated automatically by MySQL using the auto increment function which adds the next highest integer in a field.

Table **3** summarises the data types of each of the attributes.

Table 3 Description of the attributes for each entity

Table	*Columns*	*Data Type*	*Nullable*	*Auto inc*	*Constraint*	*Flags*
GENE	GENE_ID	INTEGER	NOT NULL	Y	PK	UNSIGNED
	GENE_SYMBOL	TINY TEXT	NULL			
	GENE_NAME	TEXT	NULL			
GENE_SEQUENCE	GENE_SEQ_ID	INTEGER	NOT NULL	Y	PK	UNSIGNED
	GENE_SEQ	TEXT	NOT NULL			
	GENBANK_ID	TINYTEXT	NULL			
	GENE_ID	INTEGER	NOT NULL		FK	UNSIGNED
	SPECIES_ID	INTEGER	NOT NULL		FK	UNSIGNED
GENE_SYNONYM	SYNONYM_ID	INTEGER	NOT NULL	Y	PK	UNSIGNED
	GENE_ID	INTEGER	NOT NULL		FK	UNSIGNED
	SYNONYM	TEXT	NOT NULL			
PRIMER_PROBE	PR_ID	INTEGER	NOT NULL	Y	PK	UNSIGNED
	PR_START	INTEGER	NULL			UNSIGNED
	PR_LENGTH	TINYINT	NULL			UNSIGNED
	PR_TM	TINYINT	NULL			UNSIGNED
	PR_GC	TINYINT	NULL			UNSIGNED
	PR_SEQ	TINYTEXT	NOT NULL			
	PR_TYPE	ENUM('FP','RP','PRB')	NOT NULL			
	ASSAY_ID	INTEGER	NOT NULL		FK	UNSIGNED
SPECIES	SPECIES_ID	INTEGER	NOT NULL	Y	PK	UNSIGNED
	COMMON_NAME	TEXT	NULL			
	SCIENTIFIC_NAME	TEXT	NOT NULL			
TAQMAN_ASSAY	ASSAY_ID	INTEGER	NOT NULL	Y	PK	UNSIGNED
	FILE_ID	INTEGER	NOT NULL			UNSIGNED
	RECORD	INTEGER	NOT NULL			UNSIGNED
	AMPLICON_LENGTH	INTEGER	NULL			UNSIGNED
	PENALTY	INTEGER	NULL			UNSIGNED
TAQMAN_FILE	FILE_ID	INTEGER	NOT NULL	Y	PK	UNSIGNED
	FILE_NAME	TINYTEXT	NOT NULL			
	DATE_TIME_CREATED	DATETIME	NOT NULL			
	DATE_TIME_TRANSFERRED	TIMESTAMP	NOT NULL			
	GENE_SEQ_ID	INTEGER	NOT NULL		FK	UNSIGNED

Key

PK = Primary Key

FK – Foreign Key

2.5.3 Controlled vocabularies

A 'controlled vocabulary' is a defined list of terms for a category of information. Using controlled vocabularies in a database should simplify queries issued to it. An example of where a controlled vocabulary was implemented for this database was for the primer type entity in the primer probe table. The enumeration data type was used to constrain this column so that an oligonucleotide (i.e. a primer or probe) could only be described by one of three pre-defined terms; FP, RP or PRB. If a control vocabulary was not implemented then a forward primer, for example, could be described by a multitude of terms such as Forward Primer, F Primer, FP, and Fwd Primer.

Retrieving forward primer data from the database would therefore be over-complicated. Controlled vocabularies were also implemented in the user interface and will be discussed in Chapter 3.

2.5.4 Creating the database and its tables

Creation of the database was done via the command line interface as follows:

1. Change the directory to mysql\bin

```
#prompt  > cd c:\mysql\bin
```

2. Issue a command to create a database entitled ‘taqbase’

```
#prompt  > mysqladmin –u user –p password create taqbase
```

There are two common methods for creating tables in MySQL. The first involves directly issuing commands using the MySQL monitor, for example:

```
#prompt > mysql –u user –p password
mysql > USE taqbase;
mysql > CREATE TABLE species (
            -> species_id INT UNSIGNED PRIMARY KEY NOT NULL DEFAULT
            -> ‘0’ AUTO_INCREMENT,
            -> common_name text,
            -> scientific_name text NOT NULL)
            -> TYPE=InnoDB;
```

The second method which was used for this project involves saving all table creation statements to a text file on the server. This file, entitled ‘table_create.sql’, can be found on the accompanying CD-ROM to this project. The following command was used to create all the tables using this file:

```
#prompt > mysql –u user –p password < /path/to/table_create.sql
```

2.5.5 Database security

When MySQL is installed, a database called mysql is automatically created. It stores data such as user privileges for specific fields and tables, command privileges for the specific user and hosts that can connect to the database.

Adding users and defining their privileges is important for maintaining a secure database since there are risks associated with allowing all users full access. Important data, for example, may be inadvertently deleted or corrupted.

Adding new users and defining their privileges was performed by connecting to MySQL as the root user and issuing the GRANT command. The command shown below would allow a specific user SELECT and INSERT privileges to all tables in taqbase database.

```
#prompt  > mysql –u root –p password
mysql > GRANT SELECT, INSERT
      -> ON taqbase.*
      -> TO user@hostname
```

```
        -> IDENTIFIED BY “password”;
```

For this project the database was implemented locally for development purposes, however additional security measures should be implemented if MySQL is installed on an external network. Data can be intercepted over a network and it is recommended a secure connection is used.

Chapter 3. Software Development

Although data could be added to the database by issuing INSERT commands at the MySQL command line interface, this method would be extremely time-consuming and prone to human error. To overcome this, the next phase of the project was to write programs to automate as much of the data upload as possible. The initial sections of this chapter describe the development of software to automate data upload into the database in order to minimise user interaction. The remainder of the chapter focuses on the development of a CGI application so that the database can be queried by members of the MPT group.

3.1 Perl

Perl (Practical Extraction Reporting Language) was chosen as the programming language for this project. The rationale for choosing Perl is outlined below:

- Modules can be downloaded for free from websites such as http://www.cpan.org/ and http://www.bioperl.org (Tisdall, 2001). This is helpful when time and programming expertise are restricted since existing code can be used or modified by the developer. BioPerl modules, for example, contain extremely useful bioinformatics functions such as sequence manipulation and access to various biological databases. This was particularly relevant for this project where data would be sourced from NCBI and uploaded directly into the database. CPAN (Comprehensive Perl Archive Network) provides access to Perl modules that can be used to manipulate relational databases which was another key component of this project.

- Perl is considered to be an ideal language for CGI (Common Gateway Interface) scripting. CGI programs run on a Web server to process a form or perform a search and return the results to the client. The CGI.pm module can be used in conjunction with Perl DBI to dynamically create web pages that display database query results. As described in the objectives of this project there was a requirement to develop a simple interface to query the database and return the results to the user.
- Perl is regarded as a relatively simple programming language and is considered to be an ideal choice for a biologist with limited or no prior computer programming experience. This was an important factor to consider due to the limited timeframes for the development of this software and the lack of programming knowledge within the MPT group.
- Perl is well suited to processing long strings such as DNA sequences (Gibas & Jambeck, 2001).

3.2 Interfacing with MySQL

So that Perl could interface with MySQL, the Perl modules DBI (Database Independence) and DBD (Database Driver) were downloaded from CPAN (Comprehensive Perl Archive Network) at http://www.perl.com/CPAN/.

To test that these modules had installed correctly a short Perl program was written to connect to the database. It simply calls the connect method from the DBI module to connect to the database. If the connection fails an error message is displayed to the user. This program was entitled test_database_connect.pl and is located on the attached CD-ROM.

3.3 Program design and Implementation

Data to populate the database could be derived from 2 main sources; a nucleotide sequence database (GenBank) and text files exported from the Primer Express software. This is illustrated in Figure 7 which shows the database tables colour coded according to data source. The two data sources are related via the gene sequence table and TaqMan file table where the gene sequence ID is present in both tables.

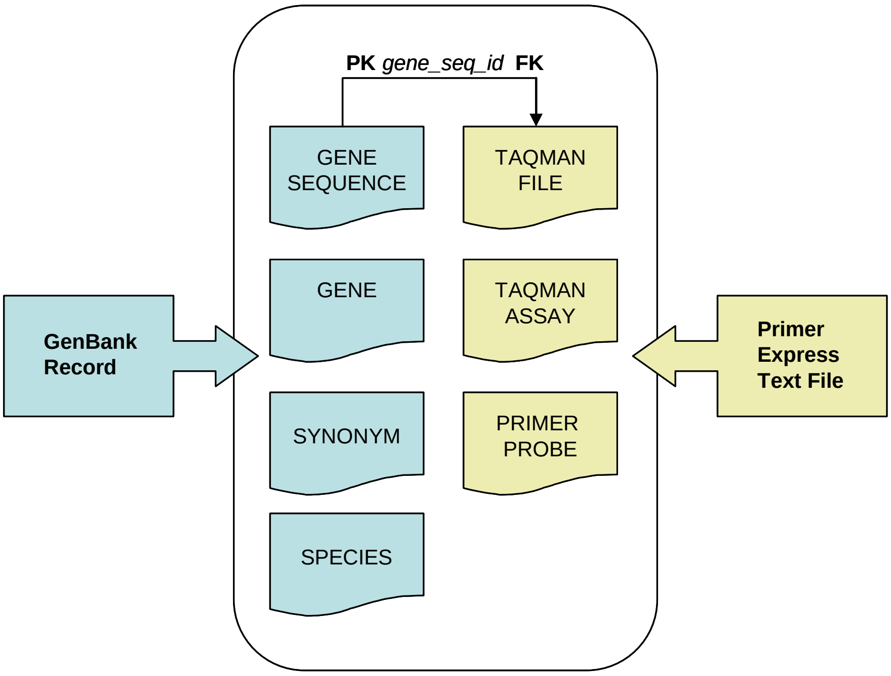
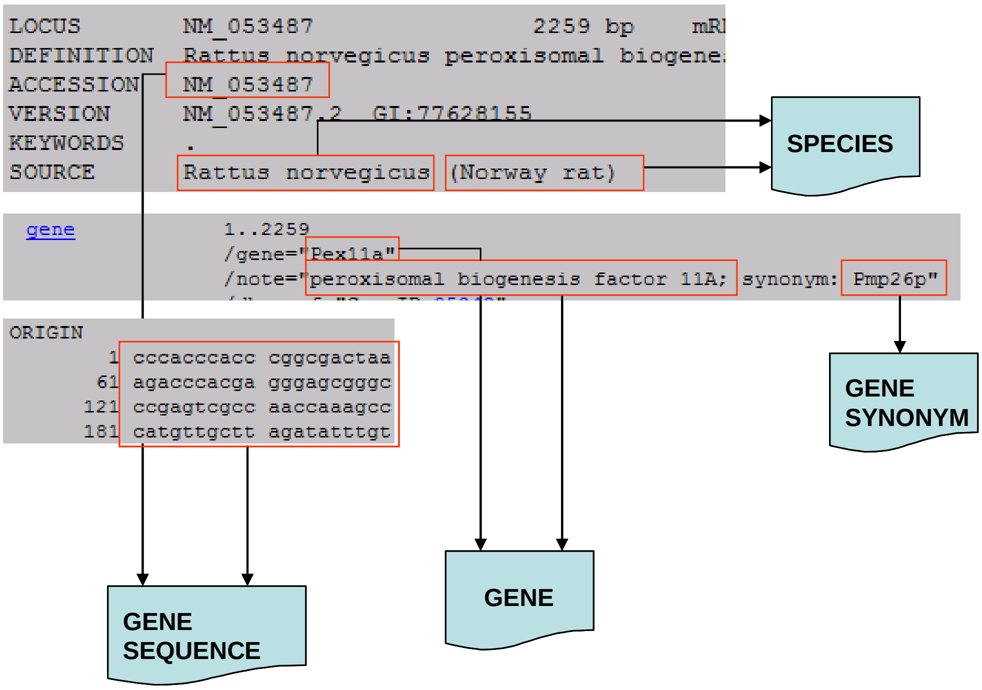


Figure 7 Summary of the data source for the tables in the database

3.3.1 Upload_Gene.pl

The objectives of upload_gene.pl are to populate the gene sequence, gene, gene synonym and species tables with data retrieved from a GenBank record. Figure 8 shows the relevant sections of a GenBank record that are used to populate the database.



```
LOCUS       NM_053487                  2259 bp    mRI
DEFINITION  Rattus norvegicus peroxisomal biogene:
ACCESSION   NM_053487
VERSION     NM_053487.2  GI:77628155
KEYWORDS    .
SOURCE      Rattus norvegicus (Norway rat)
```

```
  gene            1..2259
                  /gene="Pex11a"
                  /note="peroxisomal biogenesis factor 11A; synonym: Pmp26p"
```

```
ORIGIN
        1 cccacccacc cggcgactaa
       61 agacccacga gggagcgggc
      121 ccgagtcgcc aaccaaagcc
      181 catgttgctt agatatttgt
```

Figure 8 GenBank Record

A description of how the program performs this task is described below.

Before the program is executed, a text file containing a list of accession numbers should be placed in the designated directory. When the program is executed, it initially checks for the existence of a text file containing accession numbers in the designated directory. The database is then queried with the accession number to see if it already exists in the gene sequence table. This is to prevent duplication of data within the database thus ensuring database efficiency. If the accession exists in the database, the program proceeds onto the next accession number in the list. If the accession does not exist then a query is submitted to GenBank. Gene sequence, gene name, gene synonym, gene description and species are then extracted from the GenBank record. So that data in the gene sequence table can be related to the species and gene tables, gene_id and species_id are required to populate the gene sequence table along with the gene sequence. In order to retrieve the correct ids from these tables, the species and gene data retrieved from GenBank is used to query the species and gene tables respectively. If the species and/or gene do not exist in the database the program first inserts the gene and/or species record and then retrieves the appropriate id. This program was used to populate the database with gene data where assays had been designed by the MPT group. Accession numbers were collated, representing a comprehensive list of all assays that have been designed within the group. These were placed in a text file and the Upload_Gene.pl program was executed.

The flow chart in Figure 9 summarises the program flow. A more detailed flow diagram can be found in Appendix 1. The Perl code for upload_gene.pl is on the accompanying CD-ROM.

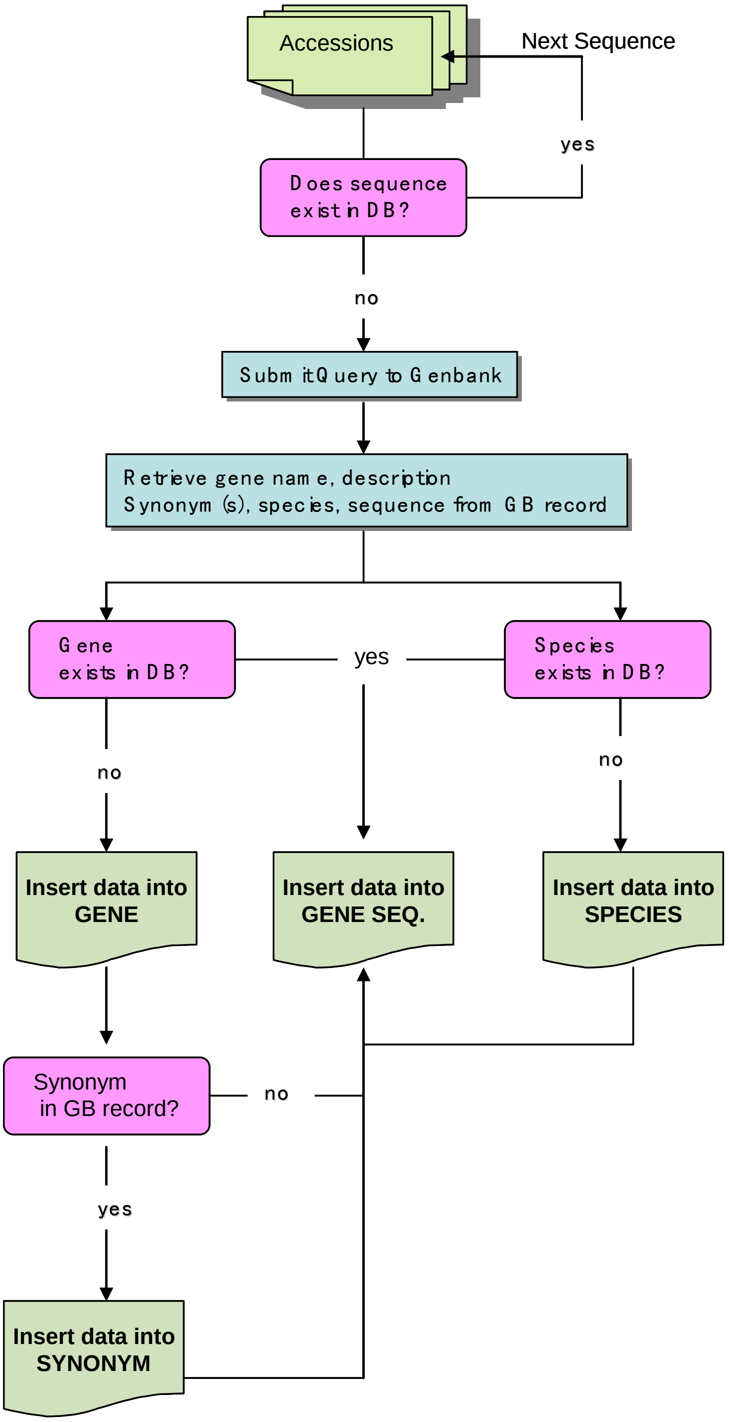


Figure 9 Flow diagram for Upload_Gene.pl

3.3.2 Upload_Assay.pl

Once the database had been populated with gene data, the next phase was to develop a program to upload data from Primer Express text files into the database. Initially a spider diagram was created outlining the objectives of the program. This is shown in Figure 10.

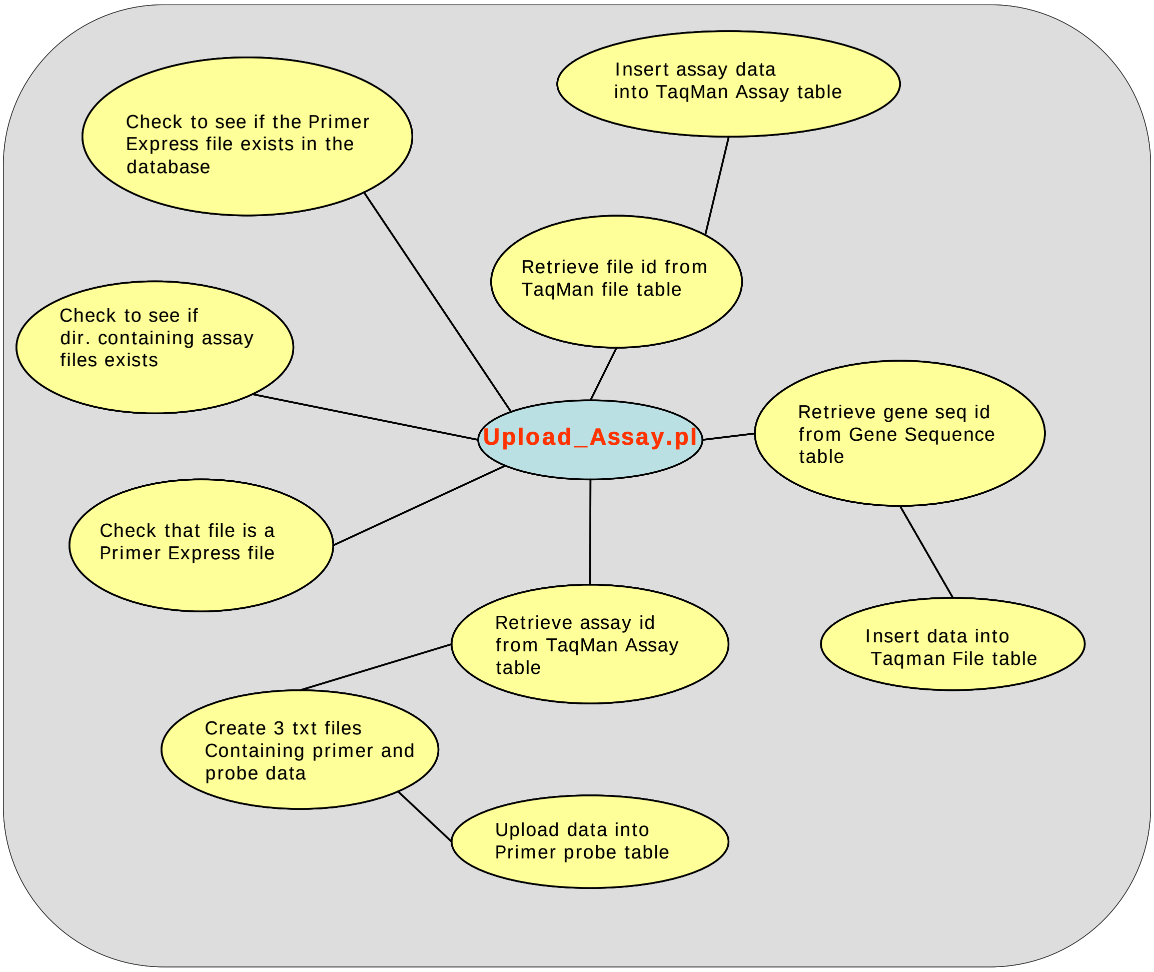
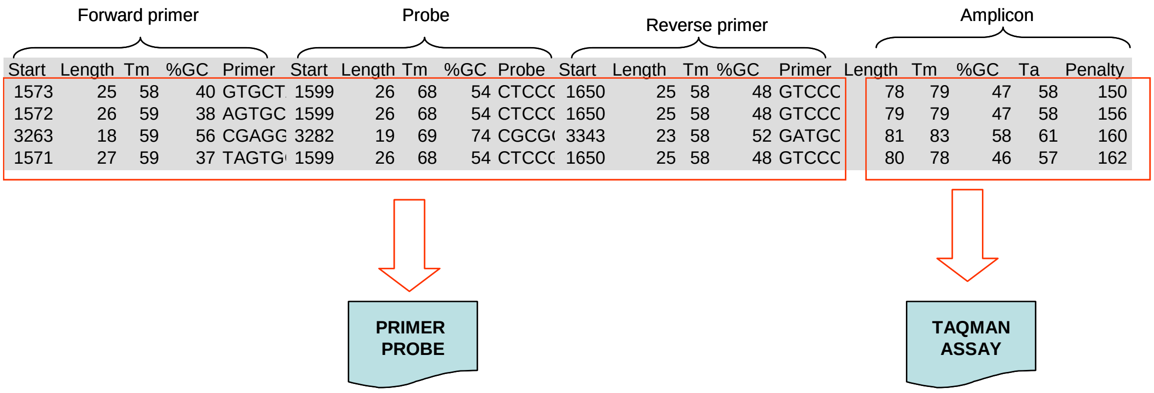

Upload_Assay.pl
Check to see if the Primer Express file exists in the database
Insert assay data into TaqMan Assay table
Retrieve file id from TaqMan file table
Check to see if dir. containing assay files exists
Retrieve gene seq id from Gene Sequence table
Check that file is a Primer Express file
Retrieve assay id from TaqMan Assay table
Insert data into Taqman File table
Create 3 txt files Containing primer and probe data
Upload data into Primer probe table

Figure 10 Spider Diagram outlining the objectives of upload assay.pl

Figure 11 shows the header row and the first 4 records of an exported Primer Express text file and which table the data would be uploaded into. A full description of the data is described in section 2.3.1.



Forward primer					Probe					Reverse primer					Amplicon				
Start	Length	Tm	%GC	Primer	Start	Length	Tm	%GC	Probe	Start	Length	Tm	%GC	Primer	Length	Tm	%GC	Ta	Penalty
1573	25	58	40	GTGCT	1599	26	68	54	CTCCC	1650	25	58	48	GTCCC	78	79	47	58	150
1572	26	59	38	AGTGC	1599	26	68	54	CTCCC	1650	25	58	48	GTCCC	79	79	47	58	156
3263	18	59	56	CGAGG	3282	19	69	74	CGCG	3343	23	58	52	GATGC	81	83	58	61	160
1571	27	59	37	TAGTG	1599	26	68	54	CTCCC	1650	25	58	48	GTCCC	80	78	46	57	162

Figure 11 Data source for the Primer Probe and TaqMan assay tables

There were several important factors to consider whilst planning the program:

- Assigning the correct gene id to the file. Since text files exported from Primer Express do not contain any reference to the sequence they were derived from, assigning the correct gene sequence id was particularly challenging. There is the option for the user to name their file with the accession number, however this is prone to errors (e.g. typographical or the assignment of an incorrect accession) and cannot be relied upon.
- Preventing duplication of data in the database so that database redundancy could be minimised.
- Ensuring that records in the TaqMan file, TaqMan assay and primer probe tables were related to each other with the correct keys.

Solutions to these issues are outlined in the brief description of the program below. A more comprehensive flow chart of the program can be found in Appendix 1.

Before the program could be executed, it was necessary to place the Primer Express files for upload in the designated directory. The program was then executed from the command line prompt. For subsequent upload of Primer Express assay data, a web-based user interface was developed for submission of data. This is described in section 3.6.

Initially the program checks for the existence of the designated directory and then the existence of Primer Express assay files using regular expressions*. If these conditions are met, the program loops through each file and queries the TaqMan file table in the database to see if it has already been uploaded. If data relating to the specified file does not exist in the database then the program proceeds with populating the database with assay file data.

The TaqMan file table is the first table to be populated with data. To ensure that this table is correctly related to the gene sequence table, the corresponding gene sequence id is required. Retrieving the correct id is performed by querying the gene sequence table with the following SQL command:

```
"SELECT gene_seq_id FROM taqman_test_3.gene_sequence
  where SUBSTRING(gene_seq, $probe_start, $probe_length) = '$probe_seq'"
```

The variables $probe_start and $probe_length refer to the position of the probe in the sequence it was derived from and the probe length. The variable $probe_seq contains the exact sequence of the probe. Data for these variables are obtained from the first record in the Primer Express file. The SQL statement retrieves the gene id where the substring of a gene sequence, determined by the probe start position and length, is an exact match of the probe sequence.

If a gene sequence id is retrieved, then the TaqMan file table and subsequent tables

* Regular expressions allow for pattern matching within strings.

are populated with data. So that the assay table could be related to the TaqMan file table, the correct file id is required. The database is queried to retrieve the most recently inserted auto increment value. This corresponds to the required file id since the file table is the most recently updated table. Each record along with the file id is then inserted into the TaqMan assay table. After each record is inserted, the most recently inserted auto increment value (i.e. assay id) is retrieved so that the assay table could be related to the primer probe table.

Rather than using the INSERT command to insert records one by one in the primer probe table (600 records for 1 assay file), data for the primer probe table including the assay_id is saved to 3 text files. Each text file contains data for forward primer, reverse primer and probe respectively. Data from the text file is directly uploaded into the database using the "LOAD DATA INFILE" command. This loads data in the database far more quickly (approximately 10 x) than using the INSERT command.

3.4 User Interface Objectives

With the database upload programs in place, the final phase of this project was to develop a web-based application so that the database could be populated with data and queried by members of the MPT group. This would be achieved by installing an appropriate Web Server and using the Perl module CGI.pm which provides an interface for common CGI tasks for example parsing input parameters and HTML code output (Guelich *et al*., 2000).

The first step in designing the user interface was to consider the interface requirements from the user's perspective. These requirements could be broken down into three main areas:

- Querying the database to see if the assay already exists and displaying the results to the Web Browser. Export assay data so assays can be ordered from various vendors.
- Retrieving a gene sequence and its associated data and displaying the results to the Browser. Export the gene sequence so a TaqMan assay can be designed using the Primer Express software.
- Uploading Primer Express assay files into the database.

In addition, the interface should allow for multi-user access to the database so that data can be shared within the MPT group. The interface should also be user-friendly to encourage users to adopt this system. Figure 12 provides a summary of user requirements.

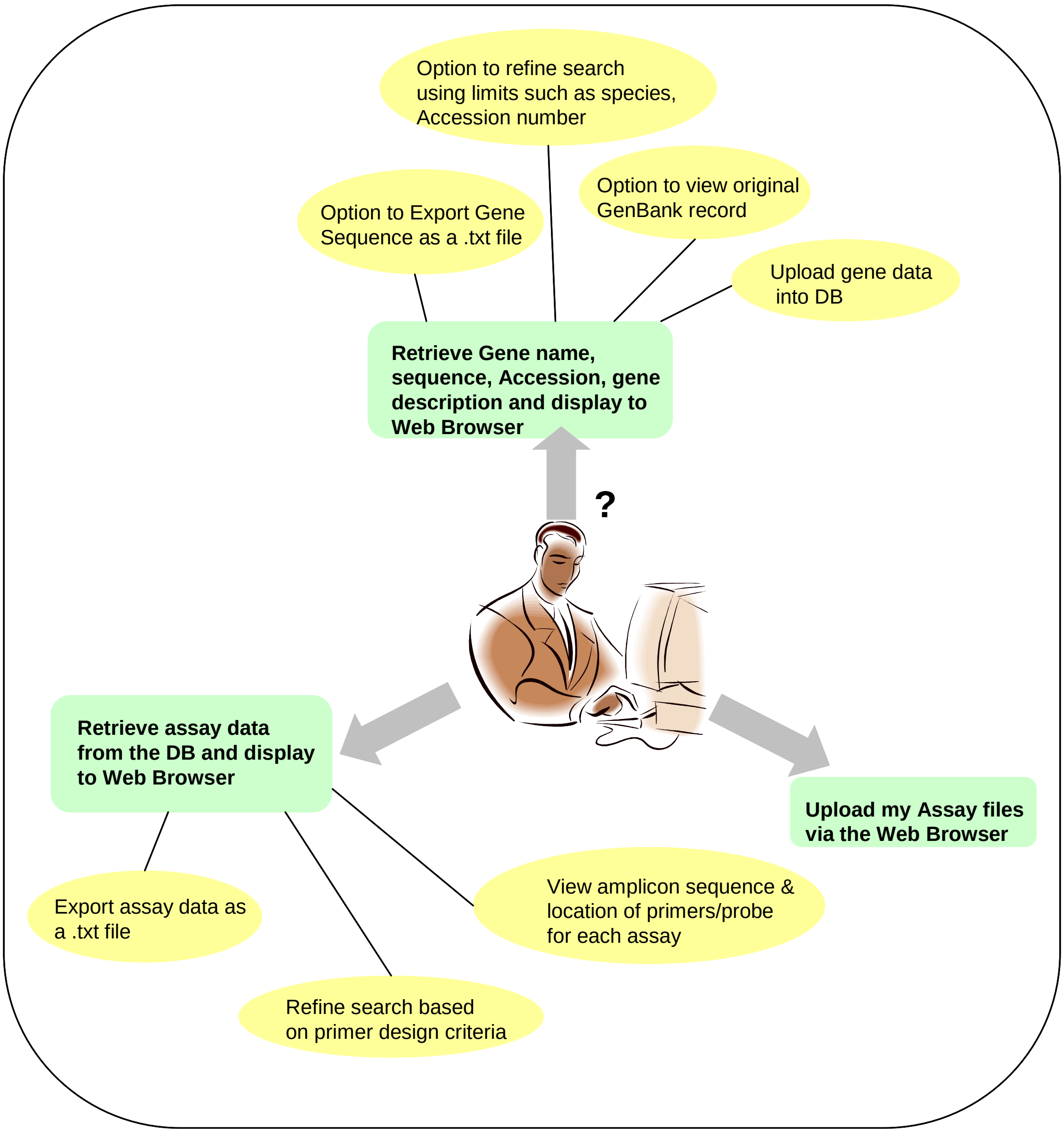

Option to refine search using limits such as species, Accession number
Option to Export Gene Sequence as a .txt file
Option to view original GenBank record
Upload gene data into DB
Retrieve Gene name, sequence, Accession, gene description and display to Web Browser
?
Retrieve assay data from the DB and display to Web Browser
Upload my Assay files via the Web Browser
Export assay data as a .txt file
View amplicon sequence & location of primers/probe for each assay
Refine search based on primer design criteria

Figure 12 Summary of user requirements

3.5 Installation and configuration of Apache

For this project, Apache was selected as the Web Server. It is open source, free and is one of the most popular servers available (Guelich *et al.*, 2000). Apache 2.0.59 was downloaded from http://httpd.apache.org/download.cgi and installed locally so that the interface could be developed without an internet connection. Before CGI programs could be executed on the server, it was necessary to modify certain parameters in the Apache configuration files. This was because Apache needs to know where the CGI programs are located and to enable the execution of CGI programs.

3.6 User Interface Development

It seemed logical to break down the development of the CGI programs into 3 distinct phases based on the user requirements outlined in section 3.4; retrieving a TaqMan assay, loading assay data and retrieving gene data. A welcome page was designed with links to the different functionalities of the application. The diagram in Figure 13 shows the CGI programs that were developed. All CGI programs can be found on the accompanying CD-ROM. A description of the programs that were developed is outlined in the next three sections.

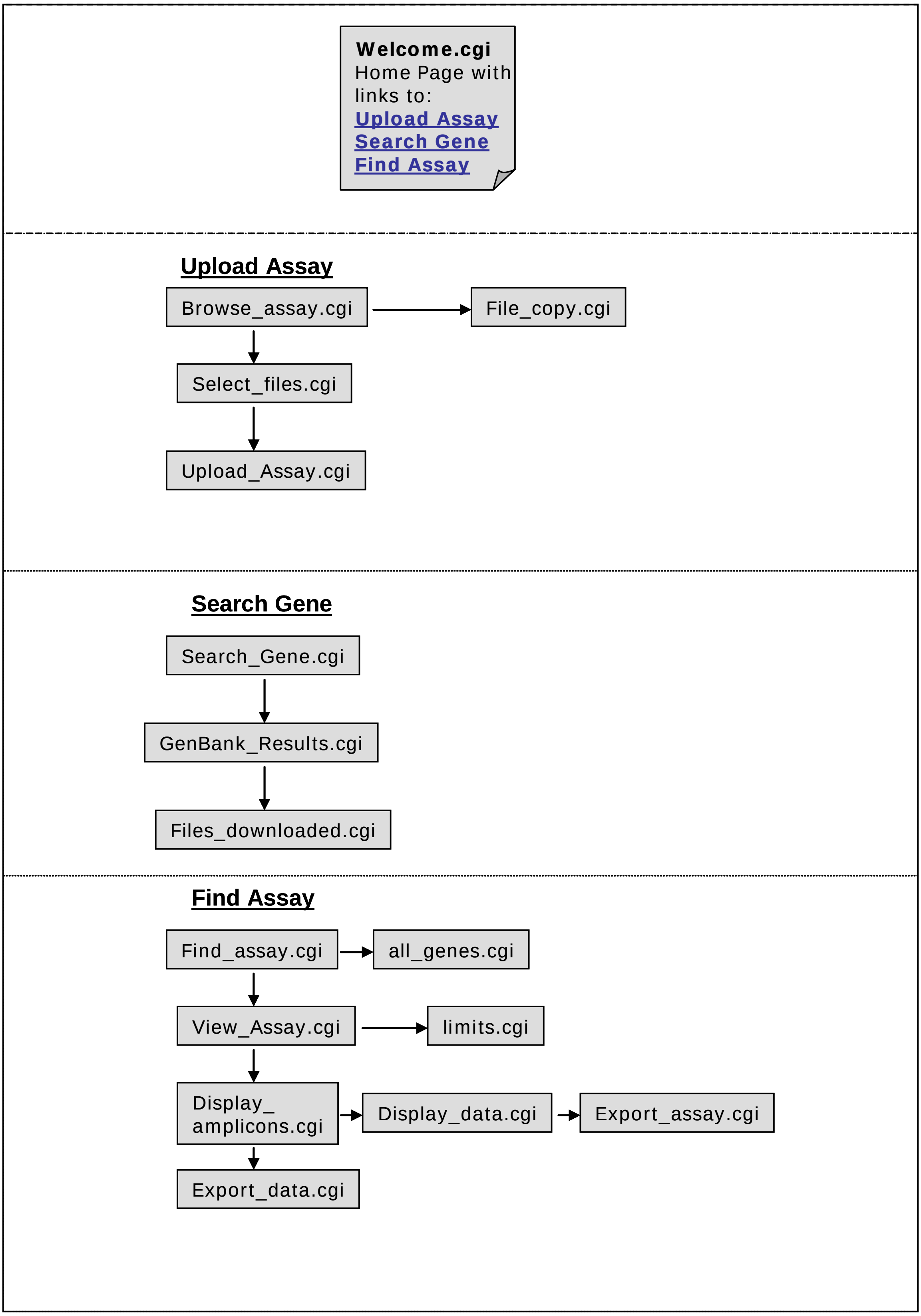


Figure 13 Summary of the CGI programs that were developed

3.6.1 Find TaqMan assay

This section describes the CGI programs that were developed to retrieve and display data associated with TaqMan assays.

- **find_assay.cgi.** This script generates an HTML page which consists of a text field so the user can enter a search term to query the database and retrieve a list of assays. The search term can be limited according to species (drop-down menu), gene symbol, accession and assay file name (radio-buttons). Using a drop-down menu for species is an example of the utilisation of controlled vocabularies (see section 2.5.3 for definition). This enables the database to be easily queried since the user is requested to select a species from a pre-defined list. This avoids typographic errors or the use of ambiguous search terms which may return no results to the user. The page also contains a hyperlink to all_genes.cgi.
- **all_genes.cgi.** This displays an HTML page consisting of a list of all assay files and their respective genes, gene description, accession and species that are present in the database.
- **view_assay.cgi.** This script queries the database using the search parameters received from the find_assay cgi script. A list of assay files along with their associated gene name and accession are retrieved from the database. The results are displayed on an HTML page along with a checkbox against each result. At the bottom of the page there is a scrolling box which contains primer probe design criteria and a link which navigates the user to limits.cgi.
- **limits.cgi.** This is a static HTML page which displays a full description of the

primer design criteria.

- **display_amplicons.cgi.** This script queries the database for assay_id, penalty score and primer/probe sequences using the file name and limits received from the view_assay.cgi script. A list of amplicons with their associated assay ids and penalty scores are displayed on an HTML page along with a checkbox for each amplicon. The amplicons are colour coded with the primer and probe sequences coloured red and green respectively. The remaining bases are coloured black. Each assay id provides a link to display_data.cgi.
- **export_data.cgi.** This program exports data associated with the selected amplicons as a text file to the directory specified in the Perl module TaqLibrary.pm. Once the data has been exported, the file name and location of the file is displayed on an HTML page. An example of an exported text file entitled amplicon_data.txt can be found on the attached CD-ROM.
- **display_data.cgi.** This script queries the database for additional assay data (e.g. primer/probe length, tm and %GC) associated with the assay_id received from display_amplicons.cgi and the results are displayed on an HTML page in a new pop-up window**.** In addition there is a scrolling box so that the user can select data for export.
- **export_assay.cgi.** This script queries the database with the parameters received from display_data.cgi and exports the data to a text file in the designated directory specified in Taqlibrary.pm. A message is displayed on an HTML page informing the user of the file name and its location.

3.6.2 Load TaqMan Assay

This section describes the cgi programs that were developed to upload Primer Express text files containing assay data into the database.

- **Browse Assay.cgi.** This script generates an HTML page consisting of a browse button so the user can search for a required assay file. In addition it gives the user instructions describing how to upload multiple files. At the bottom of the page there is a link to select_files.cgi.
- **File_copy.cgi.** This program copies the file that was selected by the user in browse_assay.cgi to the assay upload directory. An HTML page with a message is displayed in the browser depending on whether a file was selected or whether the selected file is already in the directory or not.
- **Select_files.cgi.** This program checks the designated assay folder to see whether it contains any files and whether these files are recognised assay files. An HTML page is generated to display a list of genuine assay files and another list of non-assay files if appropriate. Alongside the list of genuine assay files is a checkbox for the user to select files to upload into the database.
- **Upload_assay.cgi.** This program uploads assay data from files selected in select_files.cgi into the database. It is based on the upload_assay.pl program described in section 3.3.2. A message is displayed on an HTML page informing the user whether the files have been uploaded successfully.

3.6.3 Search Gene

This section describes the programs that were developed to retrieve gene data.

- **Search_gene.cgi.** This script generates an HTML page which consists of a text field so the user can enter a search term and submit a query to GenBank. The user can limit the search by accession, gene symbol, gene ontology and species

- **Genbank_results.cgi.** This program queries GenBank and displays an HTML page consisting of the accession and the gene description. Alongside each description is a checkbox so the user can select a record for download. The accession provides a hyperlink to the original GenBank record for the user to view.
- **Files_downloaded.cgi.** This program uploads gene data into the database and is based on the program Upload_Gene.pl outlined in section 3.3.1. Sequence data is downloaded as a text file to the designated directory specified in the Taqlibrary.pm and an HTML page is displayed to the user indicating that the sequence files have been downloaded to the designated directory.

3.7 Program design challenges and solutions

3.7.1 HTML Template

During the early stages of writing the CGI programs, HTML was embedded within the logic of the Perl code. As development progressed and the programs increased in size they became difficult to follow and the identification of errors and altering the programs became more challenging. A decision was made to separate the HTML code from the Perl code within the CGI program. Whilst this was appropriate for many of the CGI programs, this approach could not be applied if the Perl code contained loops and the output needed to be displayed as an HTML page. The Perl module

HTML::Template provided a solution to this problem. This module extends HTML with a few extra tags and enables the programmer to pass loop data from a CGI script into a template file containing HTML.

The CGI program points to the template file that will be used. It then creates an HTML::Template object, assigns a parameter and outputs the results as an HTML page.

The following template files were developed to display HTML pages and can be found on the CD-ROM; all_genes.tmpl, display_amplicons.tmpl, select_files.tmpl, upload_assay.tmpl and view_assay.tmpl. The file prefix corresponds to the CGI programs that use these template files, e.g. all_genes.cgi uses the template all_genes.tmpl.

3.7.2 Displaying colour coded amplicons

The display_amplicons.cgi program displays an HTML page with a list of colour-coded amplicons. This is an important feature since the user is able to quickly compare a list of amplicons and visualise the distance between the primers and probes. In order to achieve this there were two main problems to overcome:

1. No amplicon sequence in the database as this was not in the original Primer Express file.
2. Colour coding the amplicon in HTML so that the primer and probe sequences could be easily visualised.

Retrieving the primer and probe sequences alone were not enough to display the amplicon sequence due to the residual bases (labelled A and B) in-between the primers and probes as illustrated in Figure 14:

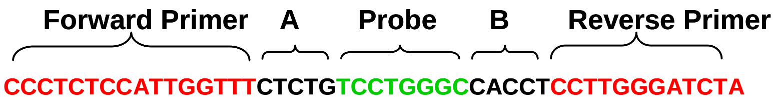


Figure 14 Amplicon

The approach used in this program was to query the database and retrieve a substring of the gene sequence that was used to design the assay. The forward primer start position and amplicon length were used in the SQL query to access the part of the gene sequence that represented the amplicon. The SQL substring function allows for part of a string to be accessed as follows:

"SUBSTRING(column name, position, length)", where column name would be the gene sequence, position would be the forward primer start position and length would be the amplicon length. The retrieved amplicon would then be saved in a variable.

So that the amplicon could be colour coded, the amplicon needed to be broken down into its individual components (i.e. primer sequence, probe sequence) outlined in Figure 14. The sequences for forward primer, reverse primer and probe were retrieved from the database and saved as individual variables. The sequences represented by A and B in the above figure were retrieved by comparing the amplicon sequence and primer/probe sequences. All matching bases and bases before or after the probe were removed, leaving the sequences represented by A and B. These were saved as variables, passed to the template file along with the probe and primer sequences, and colour coded accordingly.

3.7.3 Unique file names on export

Two of the cgi scripts, export_data.cgi and export_assay.cgi enabled the user to export selected data to a designated directory. To prevent files from being overwritten there was a requirement to assign each file a unique file name. The current date and time (localtime) was saved in a variable and used as a suffix for output file. This ensured that each filename was unique.

3.7.4 Displaying limited assay data

The view_assay.cgi program displays an HTML page which enables the user to select up to 7 primer design criteria in any combination. The selected values (parameters) are submitted to the view_assay.cgi script and SQL queries are issued to retrieve assay data that meet the selected criteria. This data is then displayed to the browser as an HTML page.

To achieve this it was necessary to address how the database would be queried since a specific SQL query must be issued for each selected criteria, for example:

- Retrieve assay ids where the probe GC content is between 30 and 80%

```
“SELECT taqman_assay.assay_id FROM primer_probe, taqman_file,
taqman_assay WHERE pr_type = 'PRB' and pr_gc between 30 and 80
AND taqman_file.file_id = taqman_assay.file_id
AND taqman_assay.assay_id = primer_probe.assay_id
AND file_name = ‘file name’”
```

- Retrieve assay ids where the amplicon length is between 50 and 150

```
"SELECT taqman_assay.assay_id FROM taqman_file, taqman_assay
 WHERE amplicon_length between 49 and 151
 AND taqman_file.file_id = taqman_assay.file_id
 AND file_name = ‘file name’”
```

If both criteria were selected an SQL query could be submitted consisting of both criteria as follows:

```
"SELECT taqman_assay.assay_id FROM taqman_file, taqman_assay
primer_probe, taqman_assay
 WHERE amplicon_length between 49 and 151
AND pr_type = 'PRB' and pr_gc between 30 and 80
AND taqman_file.file_id = taqman_assay.file_id
AND taqman_assay.assay_id = primer_probe.assay_id
AND file_name = ‘file name’”
```

Since there are 7 criteria that could be selected in any combination there are potentially 127 SQL statements that could be issued, with one statement specific to one combination of criteria. This figure was calculated by applying the equation below to the numbers 1 to 7 and then totalling the results.

$$\boxed{\text{Number of possible selections} = \frac{C!}{y! * (C - y)!}}$$

C = Number of criteria to select from
y = Number of criteria selected

It was not a viable option to create 127 SQL statements. Therefore to account for any possible combination of selections therefore an alternative approach was employed. SQL queries are submitted to the database for each selected individual criteria and the returned assay ids are saved in an array variable. Assay ids that are common to all arrays (i.e. the intersect) are those that meet all selected criteria. The results are then displayed to the user along with the amplicon data.

3.8 Testing the software

To ensure the programs developed for uploading the database were robust, database content was checked for accuracy. Upload_Gene.pl and Upload_Assay.pl contained code to prevent the duplication of data in the database. Once the data had been uploaded, this was confirmed by querying each table in the database for duplicate data by issuing the following SQL queries:

- Check the gene table for duplicates:

```
"SELECT gene_symbol, count(*) AS number FROM gene group BY gene_symbol HAVING count(*) > 1;"
```

- Check TaqMan file table for repeating files:

```
"SELECT file_name, count(*) AS number FROM TaqMan_file GROUP BY file_name HAVING count(*) > 1;"
```

- Check Species table for repeated records:

```
"SELECT scientific_name, count(*) AS number FROM species GROUP BY scientific_name HAVING count(*) > 1;"
```

- Check Gene Sequence table for repeated records:

```
"SELECT genbank_id, count(*) AS number FROM gene_sequence GROUP by genbank_id  HAVING count(*) > 1";
```

Similar queries were issued for the remaining tables in the database. Each SQL query is designed to retrieve records where there is more than one identical record for a specified field and to return the total of number of identical records.

When the SQL queries were submitted, no data was retrieved indicating that the programs did not allow any duplicate data to be uploaded into the database. To confirm this, the programs were further tested by running the programs to load data already known to be in the database. The SQL statements were then resubmitted. Since no results were retrieved for each of the queries, the programs are considered to be robust in terms of preventing duplication of data in the database.

Further checks were carried out to ensure the relationships between the tables were correct. This was done by comparing the original assay text files and GenBank records with data retrieved from the database. If any inconsistencies arose, then the error in the program was identified and amended.

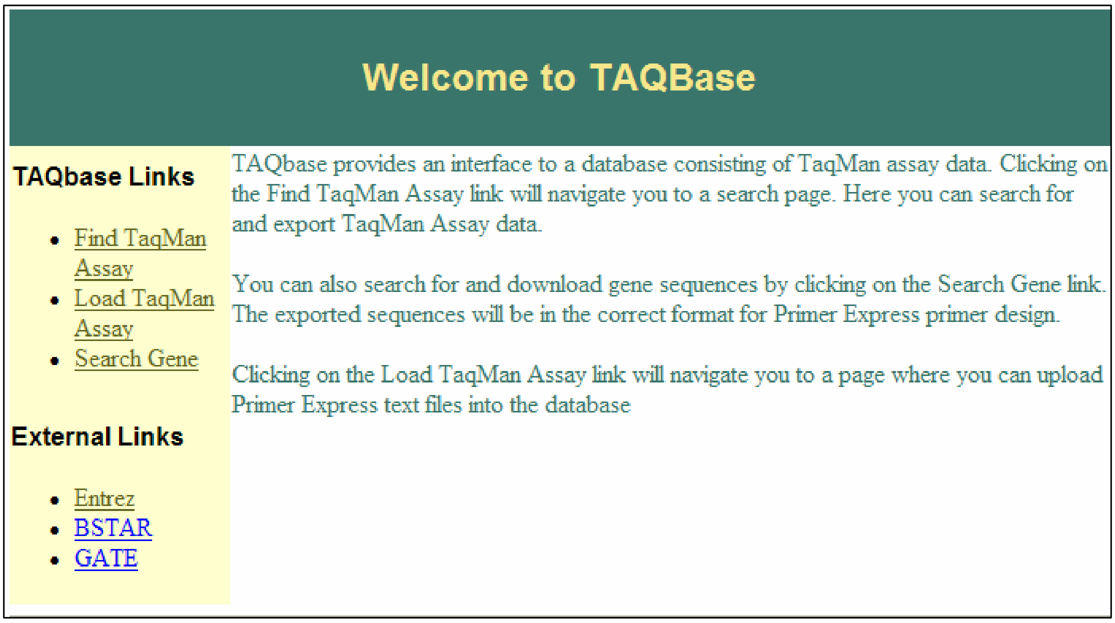
The user interface was tested by checking that all the links navigated to the correct cgi program. To ensure that the correct SQL results were displayed to the browser, the results were compared to results retrieved when the same SQL query was executed at the MySQL monitor. The interface was also tested by several members of the MPT group to see if their requirements were met and to assess how easy it was to navigate through the system.

Chapter 4. Results

This chapter shows the HTML output of the CGI programs that were developed for this project.

4.1 Welcome Page

The welcome page provides a brief summary of the application along with links for uploading an assay, searching for an assay and searching for a gene. The welcome page is shown below in Figure 15.



Welcome to TAQBase

TAQbase Links

- Find TaqMan Assay
- Load TaqMan Assay
- Search Gene

External Links

- Entrez
- BSTAR
- GATE

TAQbase provides an interface to a database consisting of TaqMan assay data. Clicking on the Find TaqMan Assay link will navigate you to a search page. Here you can search for and export TaqMan Assay data.

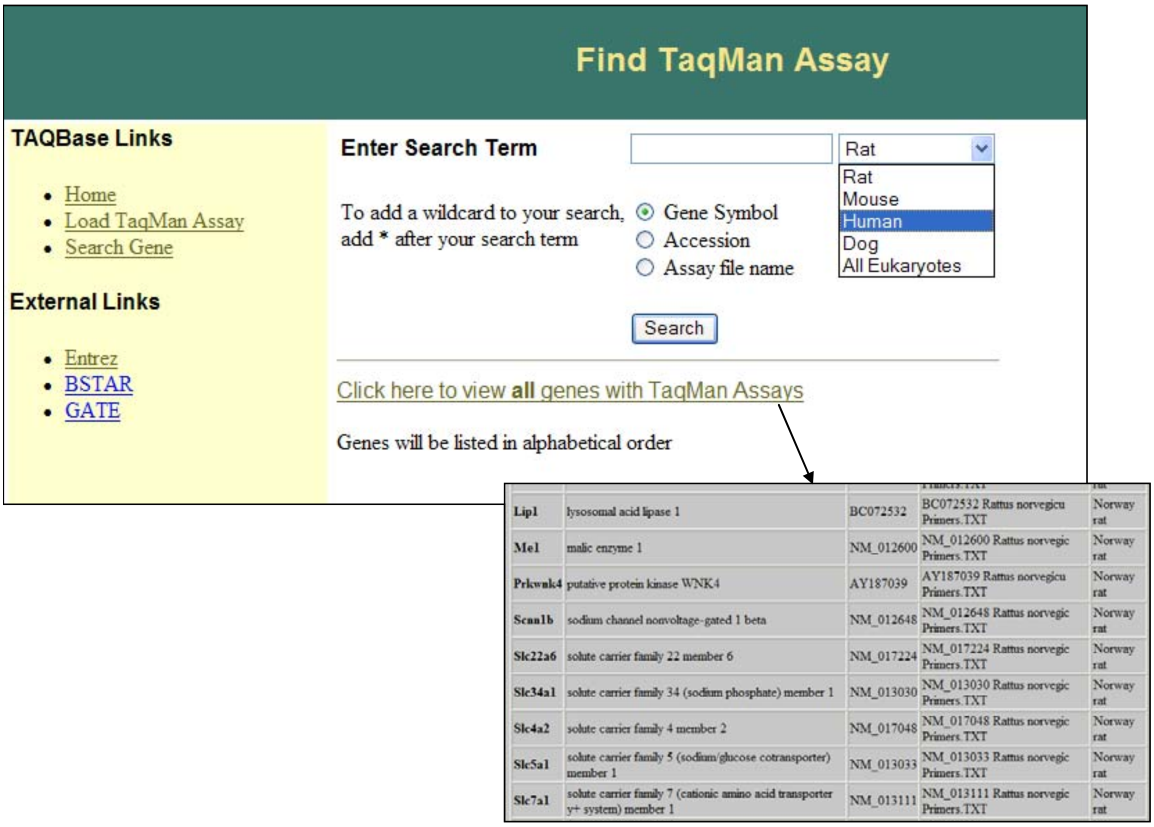
You can also search for and download gene sequences by clicking on the Search Gene link. The exported sequences will be in the correct format for Primer Express primer design.

Clicking on the Load TaqMan Assay link will navigate you to a page where you can upload Primer Express text files into the database

Figure 15 Welcome.cgi

4.2 Find TaqMan Assay

This section illustrates how the user can query the database to retrieve assay data, by clicking on the Find TaqMan Assay link on the welcome page. Initially the user is navigated to a search page as shown in Figure 16. This page provides a link where the user can view all the assays that exist in the database.



Find TaqMan Assay

TAQBase Links

- Home
- Load TaqMan Assay
- Search Gene

External Links

- Entrez
- BSTAR
- GATE

Enter Search Term

To add a wildcard to your search, add * after your search term

● Gene Symbol
○ Accession
○ Assay file name

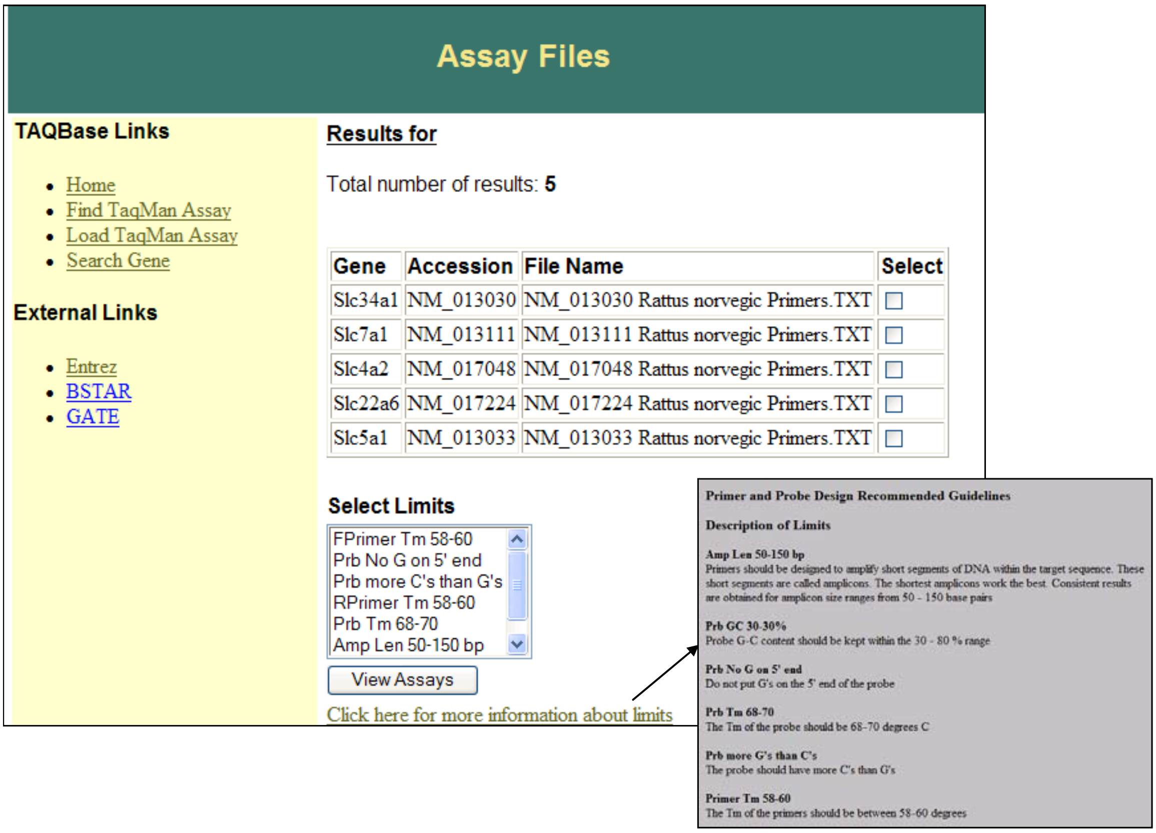
Click here to view **all** genes with TaqMan Assays

Genes will be listed in alphabetical order

Lip1	lysosomal acid lipase 1	BC072532	BC072532 Rattus norvegicu Primers.TXT	Norway rat
Me1	malic enzyme 1	NM_012600	NM_012600 Rattus norvegic Primers.TXT	Norway rat
Prkwnk4	putative protein kinase WNK4	AY187039	AY187039 Rattus norvegicu Primers.TXT	Norway rat
Scnn1b	sodium channel nonvoltage-gated 1 beta	NM_012648	NM_012648 Rattus norvegic Primers.TXT	Norway rat
Slc22a6	solute carrier family 22 member 6	NM_017224	NM_017224 Rattus norvegic Primers.TXT	Norway rat
Slc34a1	solute carrier family 34 (sodium phosphate) member 1	NM_013030	NM_013030 Rattus norvegic Primers.TXT	Norway rat
Slc4a2	solute carrier family 4 member 2	NM_017048	NM_017048 Rattus norvegic Primers.TXT	Norway rat
Slc5a1	solute carrier family 5 (sodium/glucose cotransporter) member 1	NM_013033	NM_013033 Rattus norvegic Primers.TXT	Norway rat
Slc7a1	solute carrier family 7 (cationic amino acid transporter y+ system) member 1	NM_013111	NM_013111 Rattus norvegic Primers.TXT	Norway rat

Figure 16 HTML pages for find_assay.cgi and all_genes.cgi

On submitting a search term, data is retrieved from the database and is displayed as shown in Figure 17. The user has the option to select assay files in order to view amplicon data. This search can be limited by selecting limits in the scrolling box.



Assay Files

TAQBase Links

- Home
- Find TaqMan Assay
- Load TaqMan Assay
- Search Gene

External Links

- Entrez
- BSTAR
- GATE

Results for

Total number of results: **5**

Gene	Accession	File Name	Select
Slc34a1	NM_013030	NM_013030 Rattus norvegic Primers.TXT	□
Slc7a1	NM_013111	NM_013111 Rattus norvegic Primers.TXT	□
Slc4a2	NM_017048	NM_017048 Rattus norvegic Primers.TXT	□
Slc22a6	NM_017224	NM_017224 Rattus norvegic Primers.TXT	□
Slc5a1	NM_013033	NM_013033 Rattus norvegic Primers.TXT	□

Select Limits

FPrimer Tm 58-60
Prb No G on 5' end
Prb more C's than G's
RPrimer Tm 58-60
Prb Tm 68-70
Amp Len 50-150 bp

Click here for more information about limits

Primer and Probe Design Recommended Guidelines

Description of Limits

Amp Len 50-150 bp
Primers should be designed to amplify short segments of DNA within the target sequence. These short segments are called amplicons. The shortest amplicons work the best. Consistent results are obtained for amplicon size ranges from 50 - 150 base pairs

Prb GC 30-30%
Probe G-C content should be kept within the 30 - 80 % range

Prb No G on 5' end
Do not put G's on the 5' end of the probe

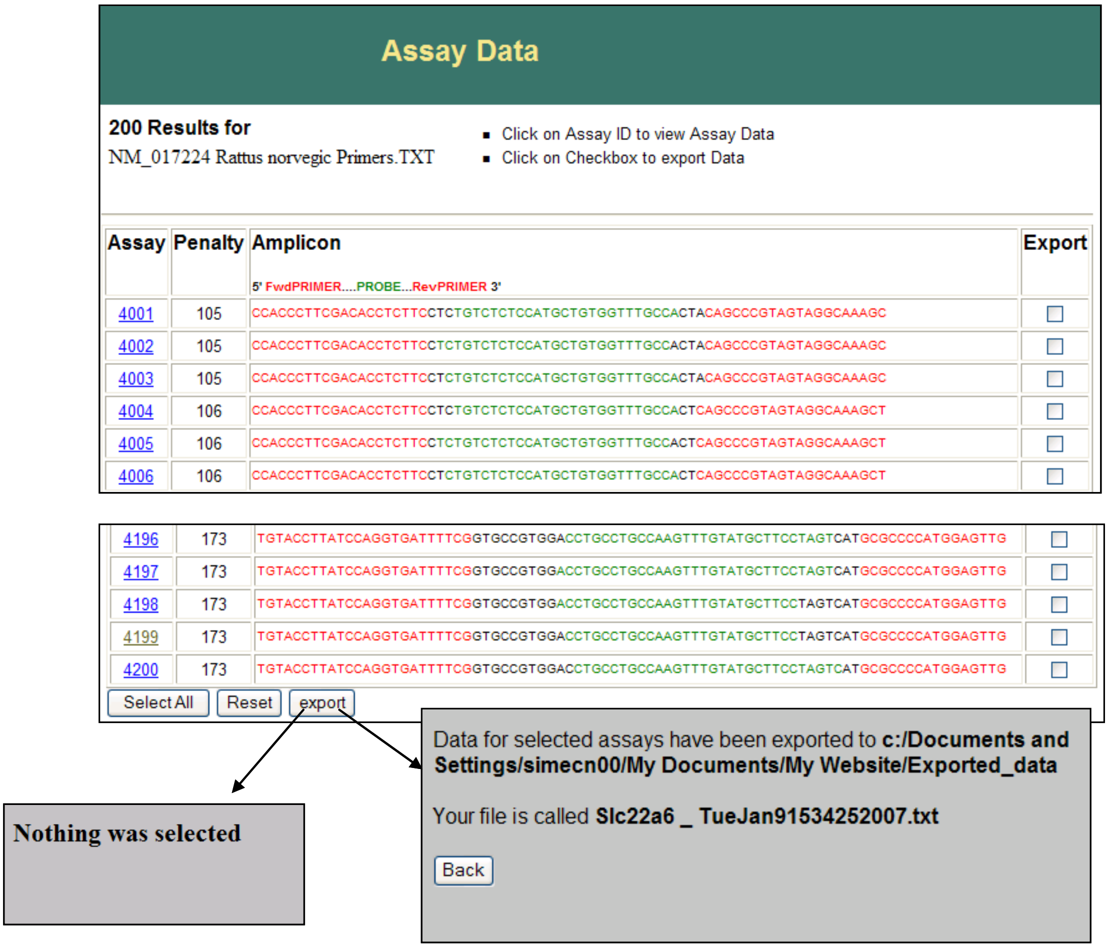
Prb Tm 68-70
The Tm of the probe should be 68-70 degrees C

Prb more G's than C's
The probe should have more C's than G's

Primer Tm 58-60
The Tm of the primers should be between 58-60 degrees

Figure 17 HTML pages for View assay.cgi and limits.cgi

Figure 18 shows how the amplicon data is displayed to the user and Figure 19 displays the HTML page that is generated when the user clicks on the assay ID hyperlink.



Assay Data

200 Results for
NM_017224 Rattus norvegic Primers.TXT

- Click on Assay ID to view Assay Data
- Click on Checkbox to export Data

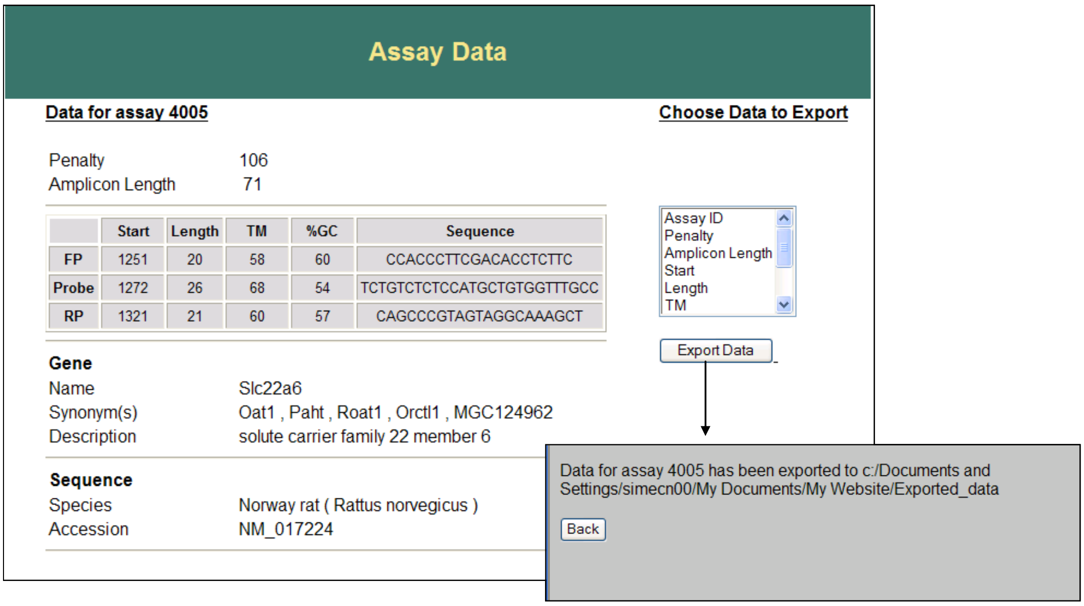
Assay	Penalty	Amplicon 5' FwdPRIMER....PROBE...RevPRIMER 3'	Export
4001	105	CCACCCTTCGACACCTCTTCCTCTGTCTCTCCATGCTGTGGTTTGCCACTACAGCCCGTAGTAGGCAAAGC	□
4002	105	CCACCCTTCGACACCTCTTCCTCTGTCTCTCCATGCTGTGGTTTGCCACTACAGCCCGTAGTAGGCAAAGC	□
4003	105	CCACCCTTCGACACCTCTTCCTCTGTCTCTCCATGCTGTGGTTTGCCACTACAGCCCGTAGTAGGCAAAGC	□
4004	106	CCACCCTTCGACACCTCTTCCTCTGTCTCTCCATGCTGTGGTTTGCCACTCAGCCCGTAGTAGGCAAAGCT	□
4005	106	CCACCCTTCGACACCTCTTCCTCTGTCTCTCCATGCTGTGGTTTGCCACTCAGCCCGTAGTAGGCAAAGCT	□
4006	106	CCACCCTTCGACACCTCTTCCTCTGTCTCTCCATGCTGTGGTTTGCCACTCAGCCCGTAGTAGGCAAAGCT	□
4196	173	TGTACCTTATCCAGGTGATTTTCGGTGCCGTGGACCTGCCTGCCAAGTTTGTATGCTTCCTAGTCATGCGCCCCATGGAGTTG	□
4197	173	TGTACCTTATCCAGGTGATTTTCGGTGCCGTGGACCTGCCTGCCAAGTTTGTATGCTTCCTAGTCATGCGCCCCATGGAGTTG	□
4198	173	TGTACCTTATCCAGGTGATTTTCGGTGCCGTGGACCTGCCTGCCAAGTTTGTATGCTTCCTAGTCATGCGCCCCATGGAGTTG	□
4199	173	TGTACCTTATCCAGGTGATTTTCGGTGCCGTGGACCTGCCTGCCAAGTTTGTATGCTTCCTAGTCATGCGCCCCATGGAGTTG	□
4200	173	TGTACCTTATCCAGGTGATTTTCGGTGCCGTGGACCTGCCTGCCAAGTTTGTATGCTTCCTAGTCATGCGCCCCATGGAGTTG	□

Nothing was selected

Data for selected assays have been exported to **c:/Documents and Settings/simecn00/My Documents/My Website/Exported_data**

Your file is called **Slc22a6 _ TueJan91534252007.txt**

Figure 18 HTML page for display_amplicons and export_data.cgi



Assay Data

Data for assay 4005

Penalty	106
Amplicon Length	71

	Start	Length	TM	%GC	Sequence
FP	1251	20	58	60	CCACCCTTCGACACCTCTTC
Probe	1272	26	68	54	TCTGTCTCTCCATGCTGTGGTTTGCC
RP	1321	21	60	57	CAGCCCGTAGTAGGCAAAGCT

Gene

Name	Slc22a6
Synonym(s)	Oat1 , Paht , Roat1 , Orctl1 , MGC124962
Description	solute carrier family 22 member 6

Sequence

Species	Norway rat (Rattus norvegicus)
Accession	NM_017224

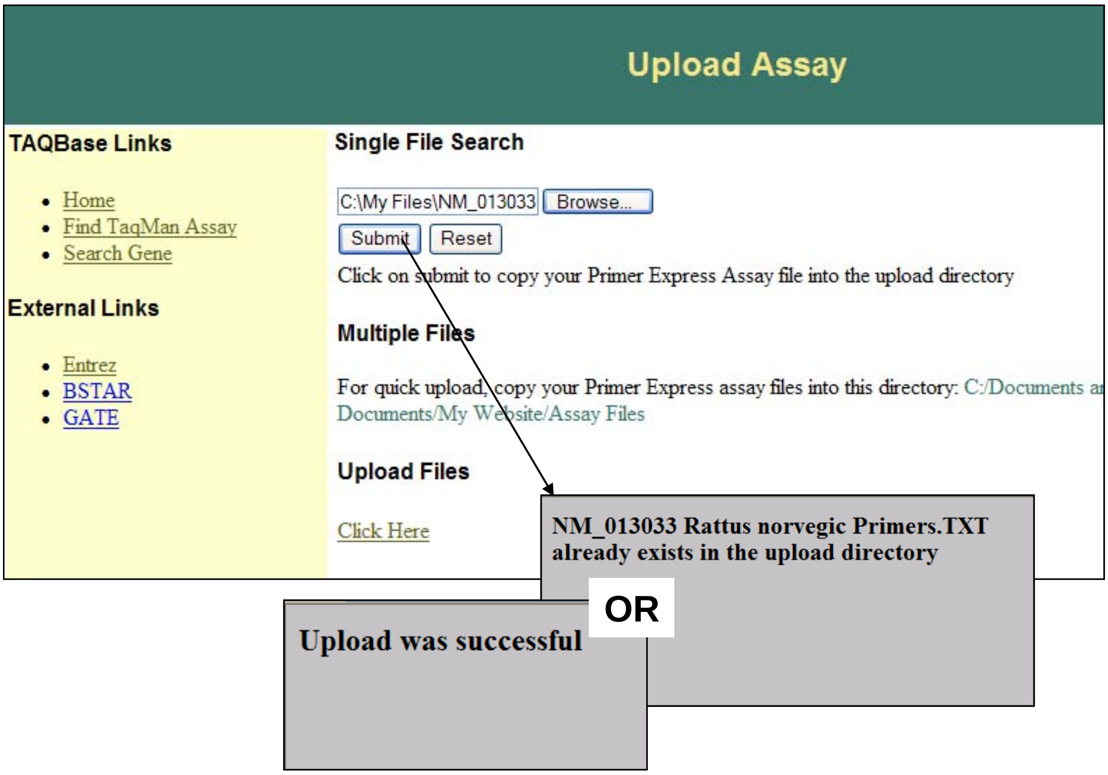
Choose Data to Export

Data for assay 4005 has been exported to c:/Documents and Settings/simecn00/My Documents/My Website/Exported_data

Figure 19 HTML page for display_data.cgi and export_assay.cgi

4.3 UploadTaqMan Assay

Figure 20 shows how the user can upload assay files into the database, by clicking on the Upload Assay link on the welcome page.



Upload Assay

TAQBase Links

- Home
- Find TaqMan Assay
- Search Gene

External Links

- Entrez
- BSTAR
- GATE

Single File Search

C:\My Files\NM_013033 Browse...

Submit Reset

Click on submit to copy your Primer Express Assay file into the upload directory

Multiple Files

For quick upload, copy your Primer Express assay files into this directory: C:/Documents an Documents/My Website/Assay Files

Upload Files

Click Here

NM_013033 Rattus norvegic Primers.TXT already exists in the upload directory

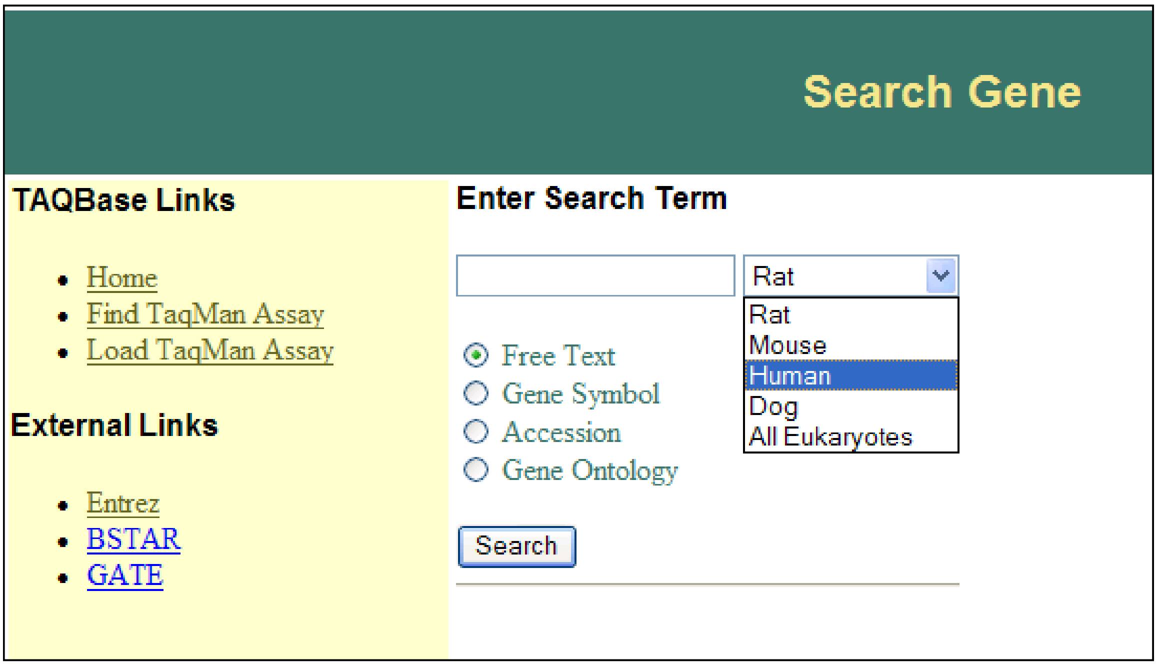
OR

Upload was successful

Figure 20 HTML for Select_Files.cgi and upload_assay.cgi

4.4 Search Gene

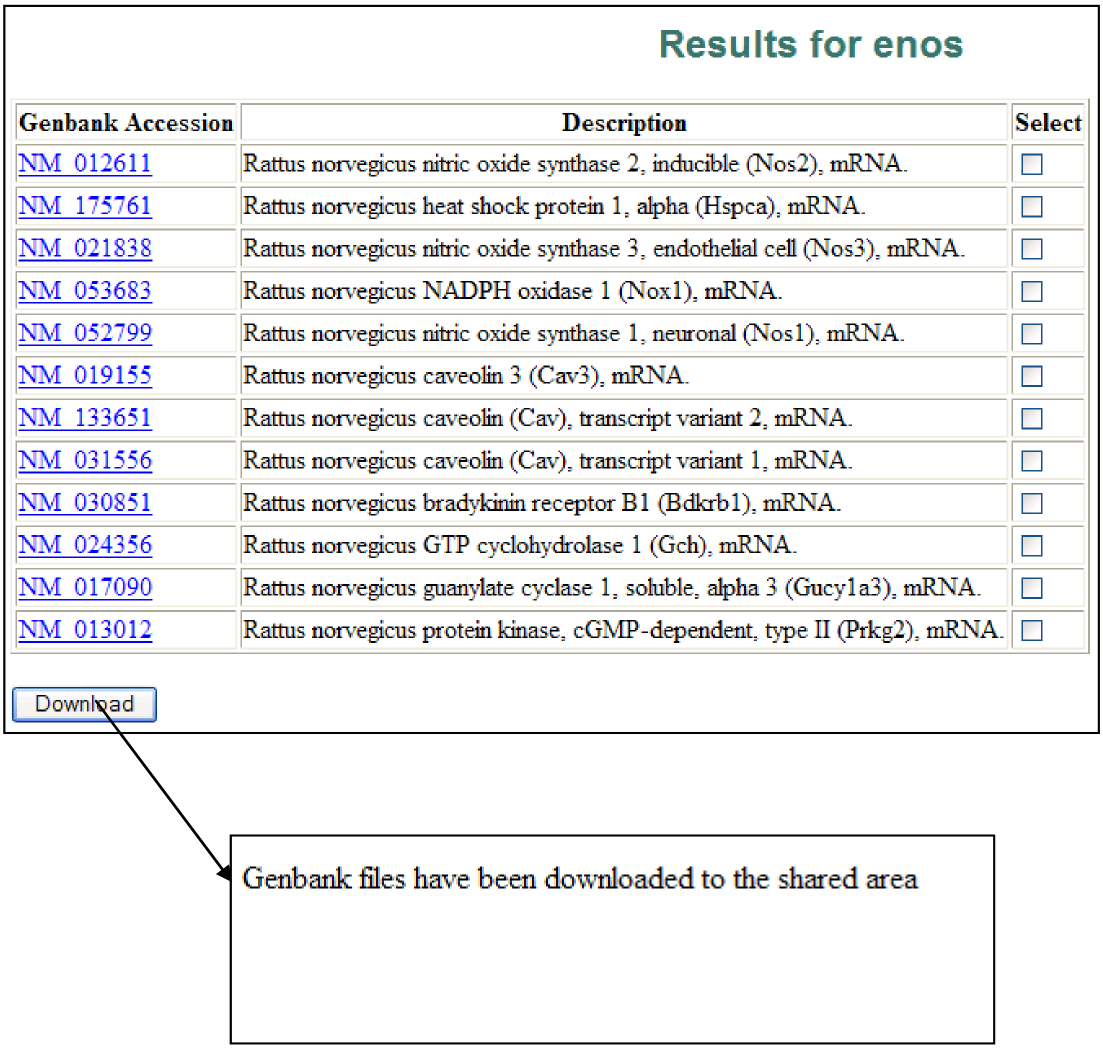
This section illustrates how the user can query the database to retrieve gene data and export gene sequences, by clicking on the Search Gene link on the welcome page. After clicking on this link the user is navigated to a search page as shown in Figure 21.



● Free Text
○ Gene Symbol
○ Accession
○ Gene Ontology

Figure 21 Search_gene.cgi

Once the user has submitted a search query the results are displayed in a table as illustrated in Figure 22 and the user can click on the GenBank accession to view the original record or download a GenBank file containing the gene sequence of interest.



Results for enos

Genbank Accession	Description	Select
NM_012611	Rattus norvegicus nitric oxide synthase 2, inducible (Nos2), mRNA.	□
NM_175761	Rattus norvegicus heat shock protein 1, alpha (Hspca), mRNA.	□
NM_021838	Rattus norvegicus nitric oxide synthase 3, endothelial cell (Nos3), mRNA.	□
NM_053683	Rattus norvegicus NADPH oxidase 1 (Nox1), mRNA.	□
NM_052799	Rattus norvegicus nitric oxide synthase 1, neuronal (Nos1), mRNA.	□
NM_019155	Rattus norvegicus caveolin 3 (Cav3), mRNA.	□
NM_133651	Rattus norvegicus caveolin (Cav), transcript variant 2, mRNA.	□
NM_031556	Rattus norvegicus caveolin (Cav), transcript variant 1, mRNA.	□
NM_030851	Rattus norvegicus bradykinin receptor B1 (Bdkrb1), mRNA.	□
NM_024356	Rattus norvegicus GTP cyclohydrolase 1 (Gch), mRNA.	□
NM_017090	Rattus norvegicus guanylate cyclase 1, soluble, alpha 3 (Gucy1a3), mRNA.	□
NM_013012	Rattus norvegicus protein kinase, cGMP-dependent, type II (Prkg2), mRNA.	□

Genbank files have been downloaded to the shared area

Figure 22 Genbank_results.cgi and files_downloaded.cgi

Chapter 5. Discussion and Conclusions

The objectives of this project were to develop a database to house TaqMan assay data, create programs to automate the population of the database and to develop a simple CGI application so the database could be queried. Each of these objectives will be discussed in the following sections.

5.1 Database design and implementation

The first objective of implementing a database to house TaqMan assay design data was met. Before commencing this project, the MPT group did not have any organised way of storing TaqMan assay data so this was a considerable improvement. The implementation of a database will increase efficiency within the group since accurate assay data can now be quickly and easily accessed. This is far preferable to searching through various directories or users laboratory notebooks where data may be overlooked or incorrect.

Various methodologies can be used when designing a relational database. The design phase is important because it helps to ensure the database is efficient, flexible and easy to manage and maintain. For this project the traditional method which includes requirements analysis, data modelling and normalisation, was used (Stephens and Plew, 2001).

Requirements analysis involved reviewing the TaqMan assay design process and identifying data to be included in the database. This was followed by data modelling. An entity relationship (E/R) diagram was created which was ultimately used as a framework to implement the physical tables in the Relational Database Management System, MySQL. During the creation of the E/R diagram, normalisation was carried out by following a set of rules to eliminate redundant data. These rules are outlined in section 1.4.5. The final implemented database was normalised to the 3rd Normal Form which means that the first 3 sets of rules have been followed. This is considered to be adequate for removing redundancy and to allow for flexibility and expansion (Meloni, 2002).

MySQL was the relational database management system selected for this project. Although there are many advantages in using this RDBMS (see section 2.5.1) it does have limitations, however, and these are described below.

During the development and testing of programs to populate the database, there were many occasions where data needed to be deleted from the database. This was due to the presence of errors in the programs that allowed large amounts of inaccurate data to be uploaded. The version of MySQL used for this project does not support the cascading deletion of foreign keys, for example when a record is deleted, records related to it are not automatically deleted. Therefore additional SQL statements must be executed to delete all related data in order to maintain database integrity or a simple Perl program can be written to perform this task.

5.2 Software Development

Software development for this project occurred in two phases and involved the development of programs to upload data into the database and a front-end CGI application for users to query the database. These will be discussed in the following sections along with a description of limitations.

5.2.1 Upload_Assay.pl

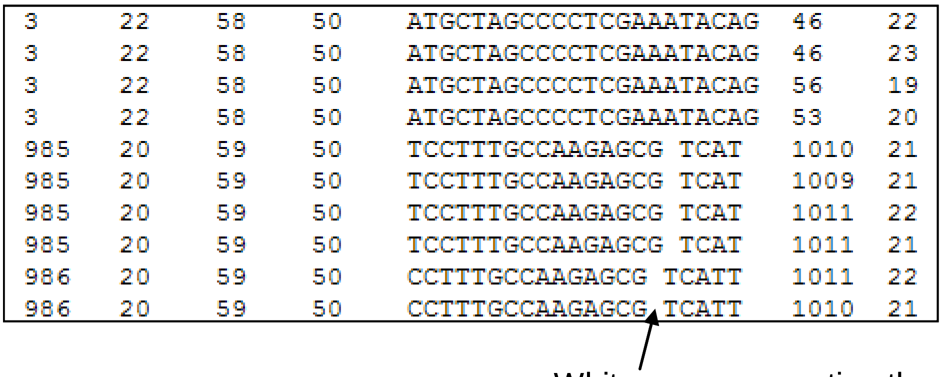
The aim of upload_assay.pl was to automate the upload of assay data from the Primer

Express text file into the database. Ideally the program should not output any errors during its execution, all data should be correctly loaded into the database so that referential integrity is enforced and it should prevent duplication of data in the database. Most of these goals were achieved by:

- Incorporating error checks in the code.
- Submitting queries to the database to check for the existence of records before inserting the record thus preventing duplication.
- Checking that the correct data had been uploaded into the database and revising the program design where necessary.

Although the program managed to achieve most of these goals it was found that on certain rare occasions incorrect data was loaded into the primer probe table. When the program was written it was based on the assumption that all Primer Express assay data was exported in the same format, i.e. each field within a record was separated by whitespace in the original Primer Express text file. The inconsistencies in the primer probe table were due to the fact that occasionally the forward primer sequence in the original text file contained whitespace. The sequence would then have been treated as two separate records rather than one which resulted in the errors.

This is illustrated in Figure 23 below.



```
3      22     58     50     ATGCTAGCCCCTCGAAATACAG   46     22
3      22     58     50     ATGCTAGCCCCTCGAAATACAG   46     23
3      22     58     50     ATGCTAGCCCCTCGAAATACAG   56     19
3      22     58     50     ATGCTAGCCCCTCGAAATACAG   53     20
985    20     59     50     TCCTTTGCCAAGAGCG TCAT    1010   21
985    20     59     50     TCCTTTGCCAAGAGCG TCAT    1009   21
985    20     59     50     TCCTTTGCCAAGAGCG TCAT    1011   22
985    20     59     50     TCCTTTGCCAAGAGCG TCAT    1011   21
986    20     59     50     CCTTTGCCAAGAGCG TCATT    1011   22
986    20     59     50     CCTTTGCCAAGAGCG TCATT    1010   21
```

Whitespace separating the sequence

Figure 23 Section of the Primer Express text file

The program was amended to account for this anomaly by adding an additional error checking step. The program checks each row in the Primer Express text file to see if each row of data contains the expected 20 fields. If the row contains 21 fields then the whitespace within the forward primer sequence is removed before the data is uploaded into the database. Although no other irregularities were found in the text files it should not be assumed that they will never occur. Additional error checking should be incorporated in the program to prevent incorrect upload of data.

During the initial stages of program development and testing, these errors were not identified since all the data from the text files had uploaded correctly. They were only discovered when data from a large number of assay files had been uploaded. This could have been avoided if more assay files were used to test the database and if the program contained error checking to ensure that every record was in the correct format before it was manipulated and uploaded.

There were a number of challenges associated with designing this program. The biggest challenge was ensuring that the correct gene sequence id was entered into the

TaqMan file table so that referential integrity could be maintained. As described in section 3.3.2, each assay file contained no reference to the sequence it was derived from other than the primer and probe sequences themselves and their relative position on the original sequence. This data was used to query the gene sequence table in order to retrieve the correct gene sequence id. Loading assay file data into the database was therefore dependent on the existence of the sequence data in the gene sequence table. If the gene sequence was not in the database the user would be informed and prompted to upload the appropriate gene sequence using the upload_gene program. This is not ideal because user interaction is required. The program therefore cannot be described as a fully automated system for uploading assay data into the database. A description of how this program could be improved will be discussed in section 5.4.

5.2.2 Upload_Gene.pl

The aim of this program was to query an external nucleotide database (GenBank) with a list of accession numbers and retrieve the appropriate data for upload into the gene sequence, gene, gene synonym and species tables. All data for a specific accession is retrieved from one GenBank record by calling methods from the BioPerl module and complementing these with regular expressions.

This program generally ensured referential integrity and data accuracy. The program could be improved, however, to increase its flexibility with regard to input data. A query could not be issued to the nucleotide database if the user could not provide the accession number. This was resolved through the development of the CGI program, genbank_results.cgi. This program incorporated the code for upload_gene.pl and additional code so that GenBank could be queried with gene name, free text and gene ontology in addition to the accession number.

5.2.3 Transaction Processing

Transaction commands were used in the upload assay programs to limit the presence of non-related data in the database. They maintain database integrity by ensuring that groups of SQL queries are executed completely or not at all so no operations are aborted mid-processing. For example, a system failure may lead to the interruption of data upload so that a table is only partially populated with data or a series of related tables would not be populated. This would result in database integrity being compromised (Forta, 2004).

5.3 User Interface

The final phase of this project was to develop a user interface so that the database could be easily queried and results displayed to the user. A CGI application was developed to achieve this task. It consisted of a welcome page with links to HTML pages for uploading TaqMan assay data, searching for gene data and searching for assay data.

5.4 Future work

This section discusses future work that could be carried out to expand the database schema, improve the programs and increase the functionality of the user application.

5.4.1 Database expansion

Although the database designed and implemented for this project will be a valuable resource to the MPT group in its own right, expanding the database to incorporate additional data would certainly be of value.

For example, it would be useful to include an entity which consists of user data, i.e. user ID, first name and last name. The client may wish to retrieve assays from the database that only he or she has designed. With the current database schema, this query could not be executed.

The user entity would be related to the TaqMan file entity as shown in Figure 24.

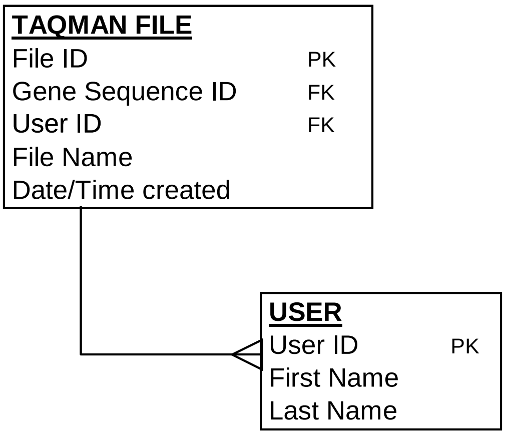


Figure 24 Relationship between user and TaqMan file table

From the client's perspective it would be useful to add additional attributes to the TaqMan assay table so that the following data could be recorded;

- has a specific TaqMan assay been used in an experiment?
- was the assay successful in the experiment?
- who has used a particular assay?

As outlined in section 2.3.1, the assay design process involves the creation of a text file containing a list of 200 candidate assays for a specific gene sequence. This data is then uploaded into the database by the programs that have been designed for this project. When a TaqMan experiment is performed, only one assay selected by the user is used. For future experiments where the same gene is to be investigated knowing which assay has been successful in previous experiments would be invaluable.

The next natural step in expanding the database would be to include experimental data and TaqMan results as described in section 1.3. Since the current database has been carefully designed, additional entities could easily be included. An example of an expanded schema which includes entities and attributes for experimental data and TaqMan results is shown in figure Figure 25. This schema has been normalised to the 3rd Normal Form using the methods described in section 2.4.2.

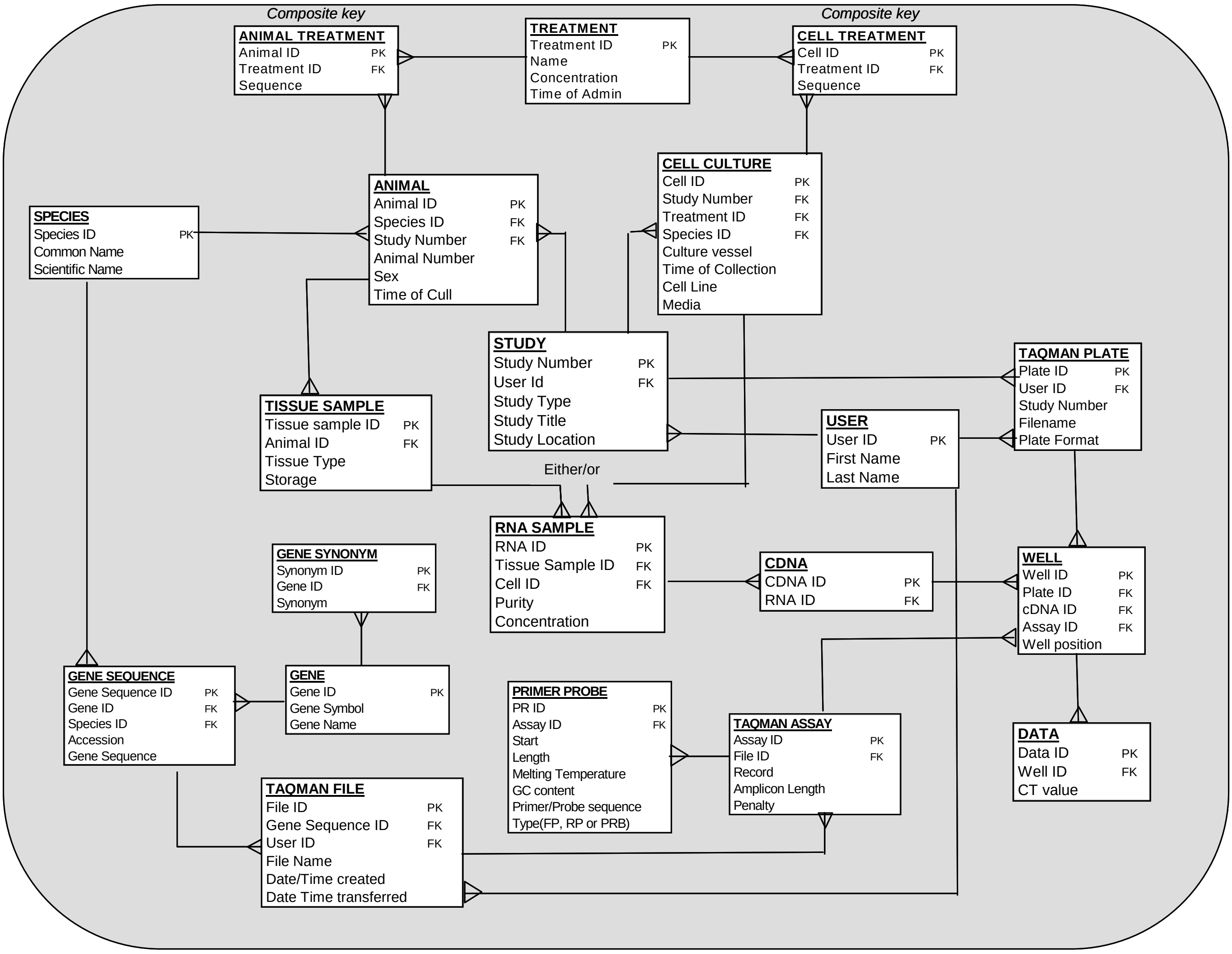


Figure 25 Example of Expanded Schema

If the database was expanded, there would be a need to address how the database would be populated with this additional data. Manual data entry would not be practical and would possibly lead to inaccuracies. One way of approaching this would be to perhaps implement a bar coding system for users to track and upload experimental data. In fact, there is already an initiative within GSK to use such a system within other departments so this could be a feasible option.

Another consideration is as the data expands, procedures should be put in place so the database is backed regularly to prevent loss of data. This can be performed using the mysqldump application. The output file created by this command contains the commands to create and populate the tables, (Meloni, 2002).

5.4.2 RDBMS upgrade

MySQL software is continually under development with additional features added for each subsequent version. As of writing this thesis, the most current version of MySQL is version 5.0. There are later versions of MySQL available (5.1 and 5.2) but these are pre-production releases. It is recommended to save the database tables and its contents using mysqldump, then install the latest version of MySQL and reload the tables and their contents. Alternatively, the database could be implemented in a commercial RDBMS where support is available within GlaxoSmithKline for example Microsoft SQL server or Oracle. Oracle is reliable, highly scaleable and has many tools to manage databases. Microsoft SQL is also a powerful DBMS with many advanced features such as data mining (Buchanan, 2002). If the database developed for this project is expanded then these RDBMS's should be considered.

5.4.3 Software improvements

As described in section 5.2.1, the upload of assay data into the database was not a fully automated process. If an assays corresponding gene sequence did not exist in the database, the user would be prompted to upload the gene sequence data before re-loading the assay data. Ideally code should be written so that the gene sequence is automatically retrieved from an external nucleotide sequence database and uploaded into in the local database. Potentially, this could be achieved by submitting the primer and probe sequences into a BLAST (Altschul *et al.*, 1990) program. A list of similar sequences would be returned. The program could then loop through this list until the primer and probe sequence matched at the specified position on the retrieved sequence. Sequence data could then be uploaded into the database and the correct gene sequence id retrieved. This approach would need to be investigated further to confirm that it is feasible.

As with many applications, there are many features which could be included to enhance the interface. Examples include:

- Enabling batch searches. For example it would be useful if the search gene HTML page included a batch search option where the user could submit a list of genes to query the database.
- Browsing for multiple assay files. Including an option where the user can browse and upload multiple assay files from the interface.
- Enabling the user to submit BLAST queries on the amplicons that are shown on the display_amplicons.cgi page. This would provide additional data relating to the specificity of an assay which is a consideration when using a TaqMan assay in an experiment.
- Allowing flexibility for uploading TaqMan assay data from sources other than Primer Express text files.

The CGI application could be considerably enhanced by the development of a fully integrated system. This would involve the incorporation of an algorithm for designing TaqMan assays within the system. There are a number of web based resources that could be used to achieve this goal. For example, there are a number of BioPerl modules such as Bio::PrimerDesigner which could be adapted to design TaqMan primers and probes. Alternatively an open source primer design application called PerlPrimer (Marshall, 2004) written in Perl is available for download at perlprimer.sourceforge.net.

Assay data could then be immediately uploaded into the database thus eliminating the need to separately launch the primer design software. Also, the difficulties associated with assigning the correct gene sequence ids to TaqMan assay files could be easily resolved.

5.5 Conclusions

A database to house the data associated with TaqMan assays has been successfully designed and implemented. Programs to automate the upload of data have been written and integrated into the final CGI application which has been developed so that the database can be easily queried and uploaded by users without any knowledge of SQL. Additional error checking, however, needs to be incorporated into the programs and further testing needs to be performed to ensure reliability and accuracy of the data within the database. Before the system can be used routinely within the group it is essential that the client trials the application and provides feedback so this can be incorporated into software revisions. This will be carried out within the MPT group to ensure the application meets their requirements. Development of software is an iterative process and user feedback is important for the development of a useful system. In addition, internet security measures should be implemented which will be done in collaboration with IT specialists at GSK. This project has taken the initial step in addressing data storage issues faced by the MPT group at GSK and has provided a means of rapidly accessing TaqMan assay data of interest.

References

ALTSCHUL, S.F., GISH, W., MILLER, W., MYERS, E.W. LIPMAN, D.J. (1990). Basic local alignment search tool. J Mol Biol 215(3):403-10

BUCHANAN, W. (2002) Mastering Computing. Palgrave Macmillan

BROWN, M.C. (2001) Perl : The Complete Reference. Osbourne.

CREIGHTON, T.E., (2005). Encyclopaedia of Molecular Biology. John Wiley & Sons

CHRISTIANSEN, T. & TORKINGTON, N. (1998) Perl Cookbook; Tips and Tricks for Perl Programmers. O'Reilly

DALE, J.W. & SCHANTZ, M. (2003). From Genes to Genomes; Concepts and Applications of DNA Technology. John Wiley & Sons

DAWSON, R. (2002). Relational Databases Design and Use. Group D Publications

DESCARTES, A. & BUNCE, T.(2000). Programming the Perl DBI. O'Reilly

FORTA, B. (2004). Teach Yourself SQL in 10 Minutes. SAMS.

GIBAS C. & JAMBECK P. (2001). Developing Bioinformatics Computer Skills. O'Reilly

GUELICH, S., GUNDAVARAM, S. BIRZNIEKS, G. (2000). CGI Programming with Perl. O'Reilly

HAWRAMI, K. & BRUER, J. (1999). Development of a flurogenic polymerase chain reaction assay (TaqMan®) for the detection and quantitation of varicella zoster virus. Journal of Virological Methods 79: 33 – 40

KING, K. (2002). SQL tips and Techniques. Premier Press

KOCHANOWSKI, B. & REISCHL, U. (1999). Quantitative PCR Protocols, Methods in Molecular Medicine, Vol. 26. Humana Press

MARSHALL OJ. (2004) PerlPrimer: cross-platform, graphical primer design for standard, bisulphite and real-time PCR. Bioinformatics 20(15):2471-2472

MELONI, J.C. (2002). Teach Yourself MySQL in 24 Hours. SAMS

PATWARDHAN N, SIEVER E., SPAINHOUR S. Perl in a Nutshell. O'Reilly

PEITZSCH, R.M. (2003). Modeling Biology Using Relational Databases. Current Protocols in Bioinformatics 9.3.1 – 9.3.28

PETERSEN, J.V. (2002). Absolute Beginner's Guide to Databases. Que Corporation

RICCARDI, G. (2003). Database Management with Web Site Development. Addison Wesley

ROLLAND, F.D. (1998). The Essence of Databases. Pearson Prentice Hall.

STEIN, L. (2003). Creating Databases for Biological Information: An Introduction. Current protocols in bioinformatics 9.1.1-9.1.9

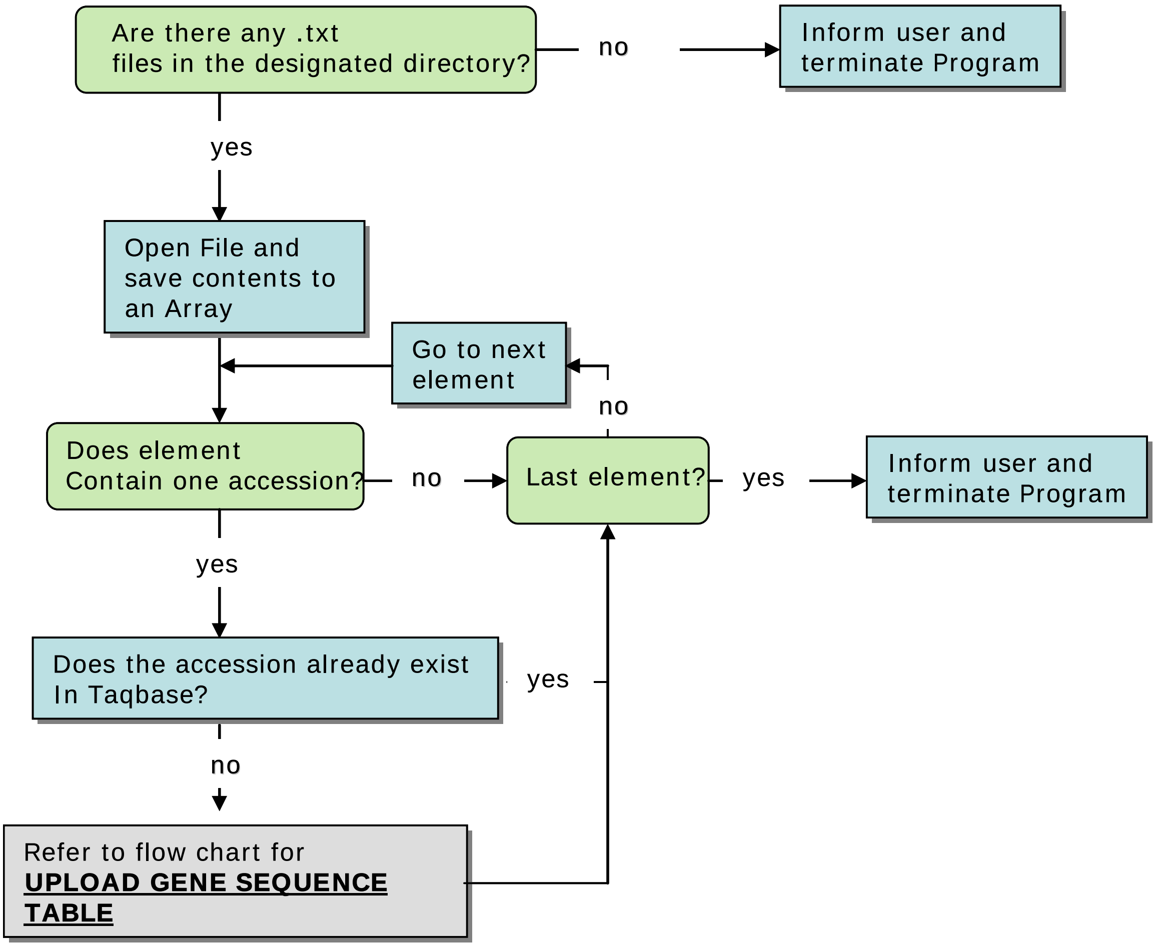
STEPHENS, R.K. & PLEW, R.R. (2001). Database Design. SAMS

TISDALL, J.D. (2001). Beginning Perl for Bioinformatics. O'Reilly

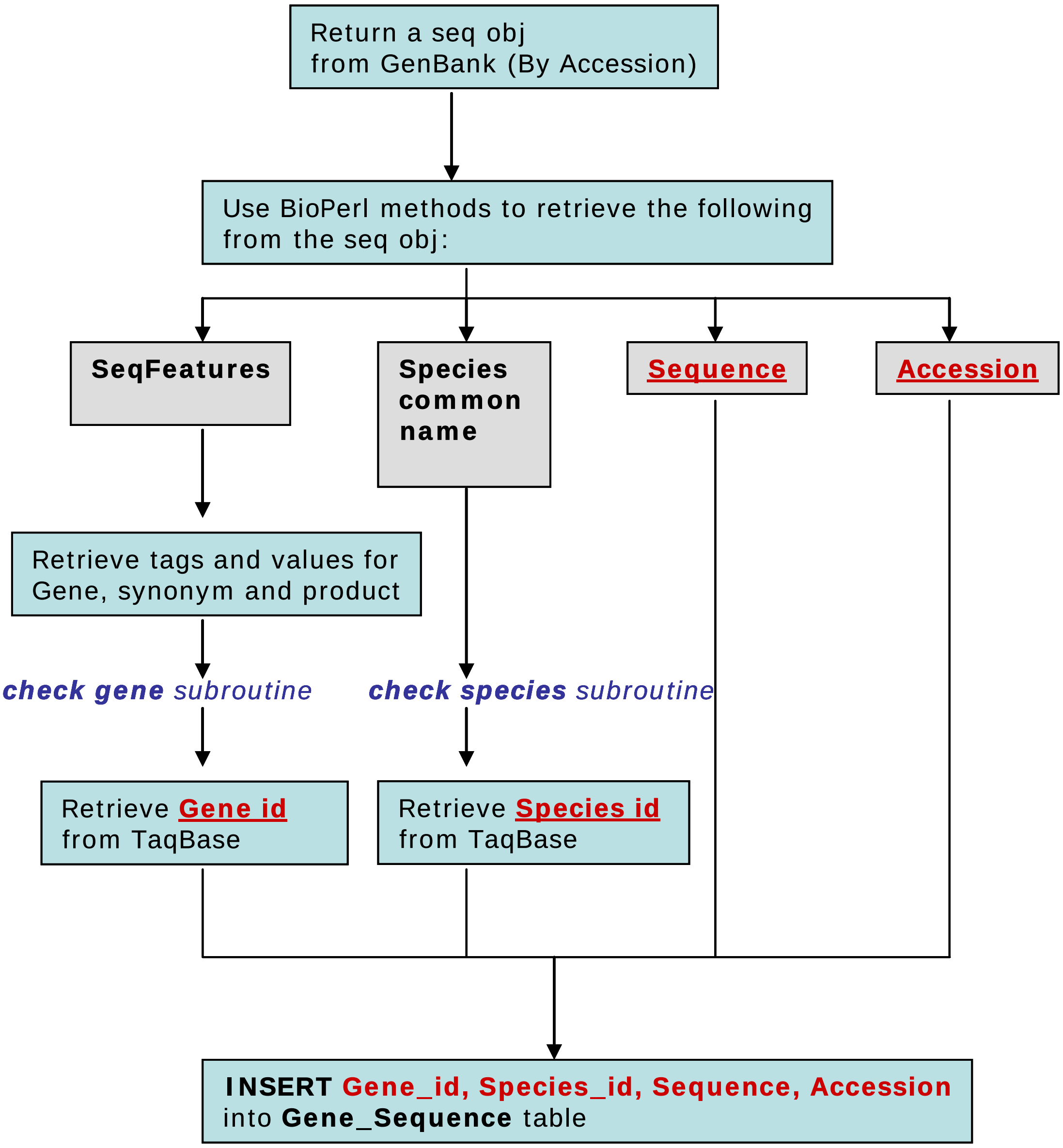
Appendices

Appendix 1

Flow chart for upload_gene.pl

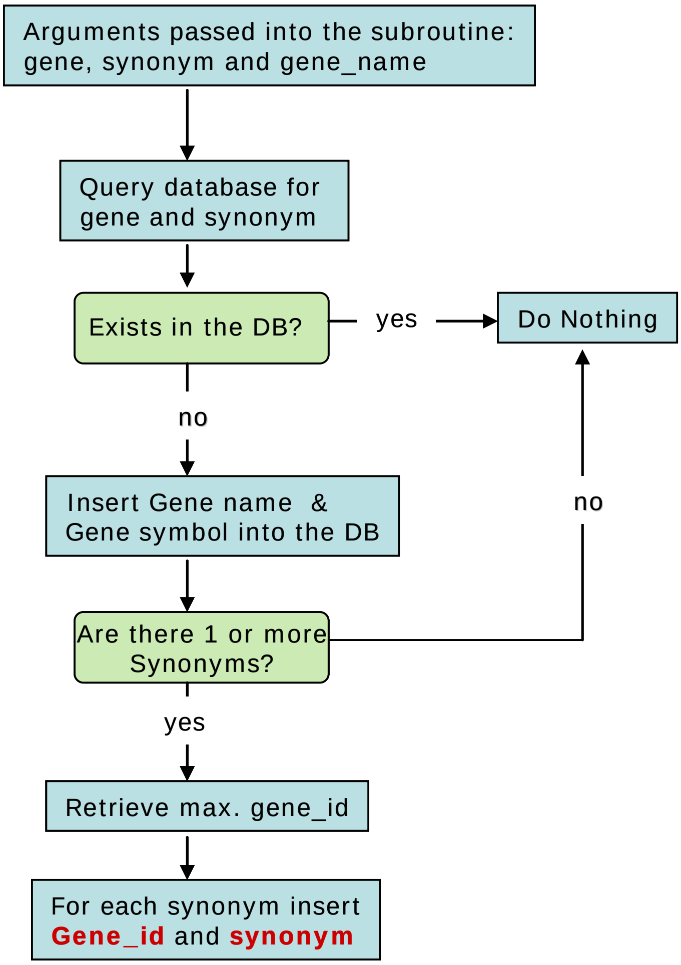

Are there any .txt files in the designated directory?
no
Inform user and terminate Program
yes
Open File and save contents to an Array
Go to next element
no
Does element Contain one accession?
no
Last element?
yes
Inform user and terminate Program
yes
Does the accession already exist In Taqbase?
yes
no
Refer to flow chart for UPLOAD GENE SEQUENCE TABLE

Flow Chart for Upload Gene Sequence Table

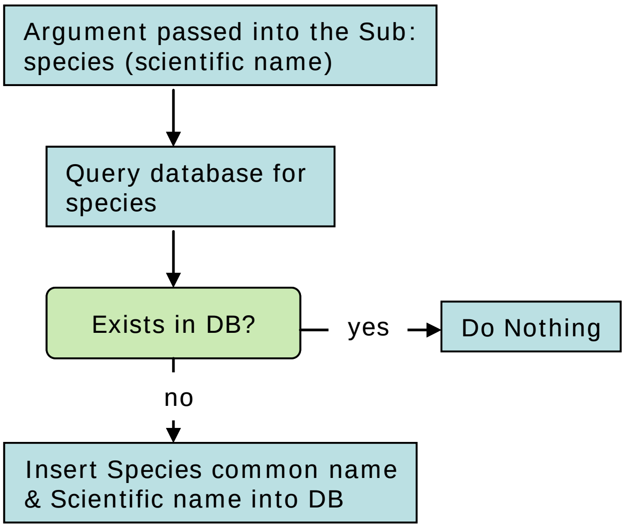

Return a seq obj
from GenBank (By Accession)
Use BioPerl methods to retrieve the following from the seq obj:
SeqFeatures
Species common name
Sequence
Accession
Retrieve tags and values for Gene, synonym and product
check gene subroutine
check species subroutine
Retrieve Gene id from TaqBase
Retrieve Species id from TaqBase
INSERT Gene_id, Species_id, Sequence, Accession into Gene_Sequence table

SUBROUTINES

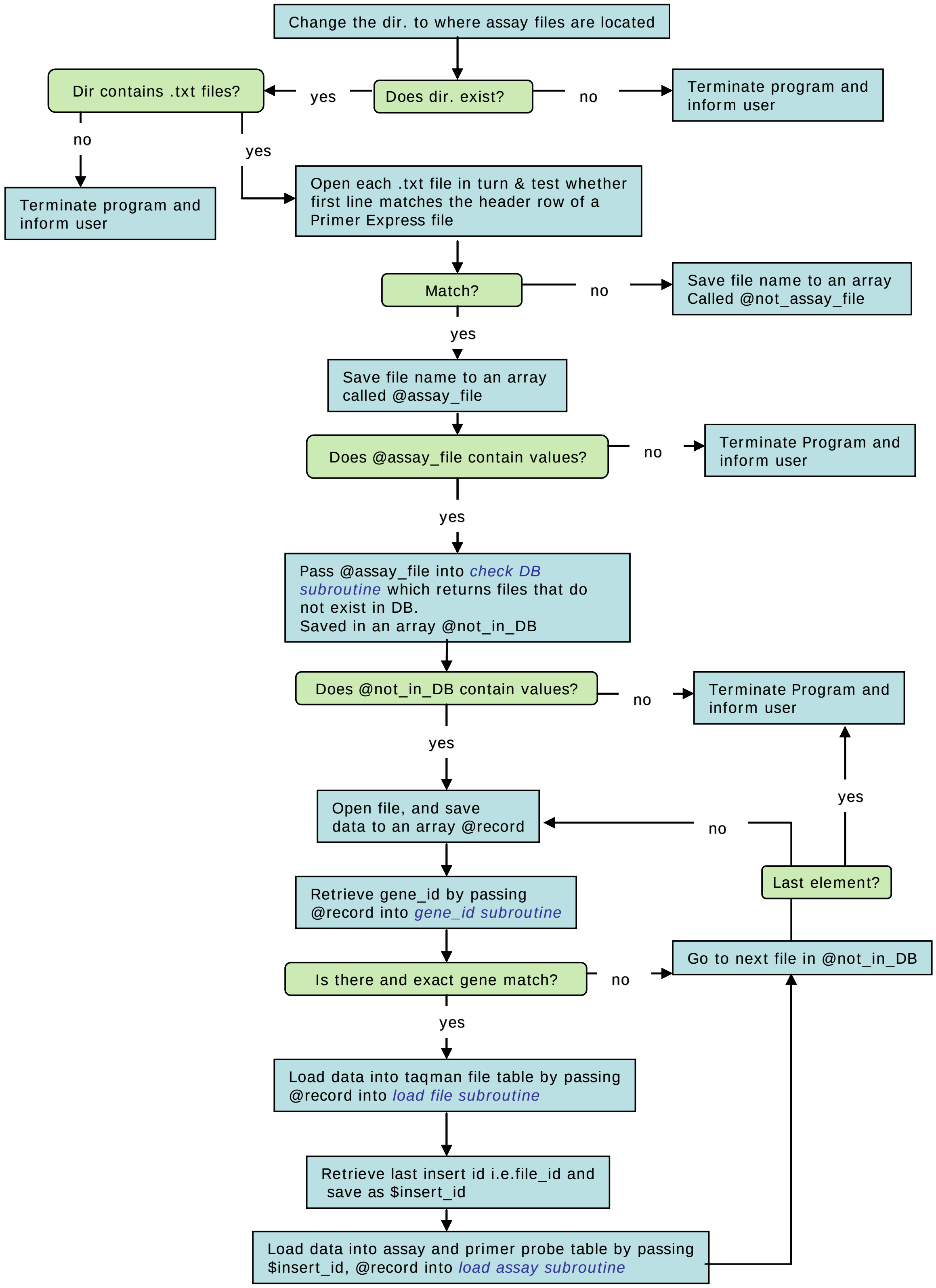
Sub check gene


Arguments passed into the subroutine: gene, synonym and gene_name
Query database for gene and synonym
Exists in the DB?
yes
Do Nothing
no
Insert Gene name & Gene symbol into the DB
no
Are there 1 or more Synonyms?
yes
Retrieve max. gene_id
For each synonym insert Gene_id and synonym

Sub check species

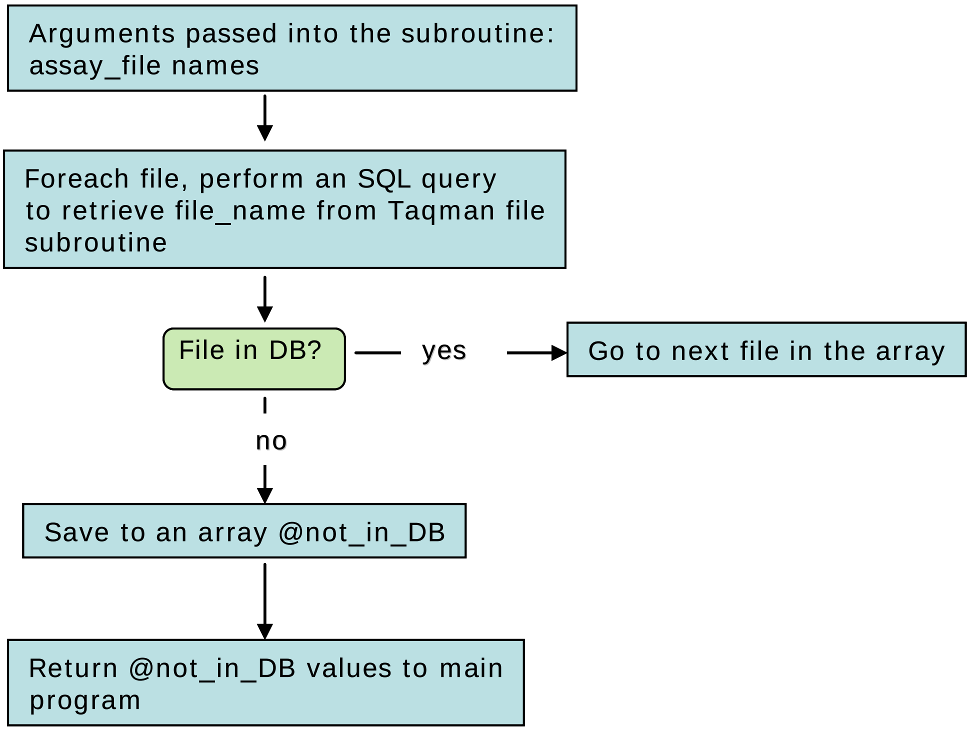

Argument passed into the Sub: species (scientific name)
Query database for species
Exists in DB?
yes
Do Nothing
no
Insert Species common name & Scientific name into DB

Flow chart for upload_assay.pl

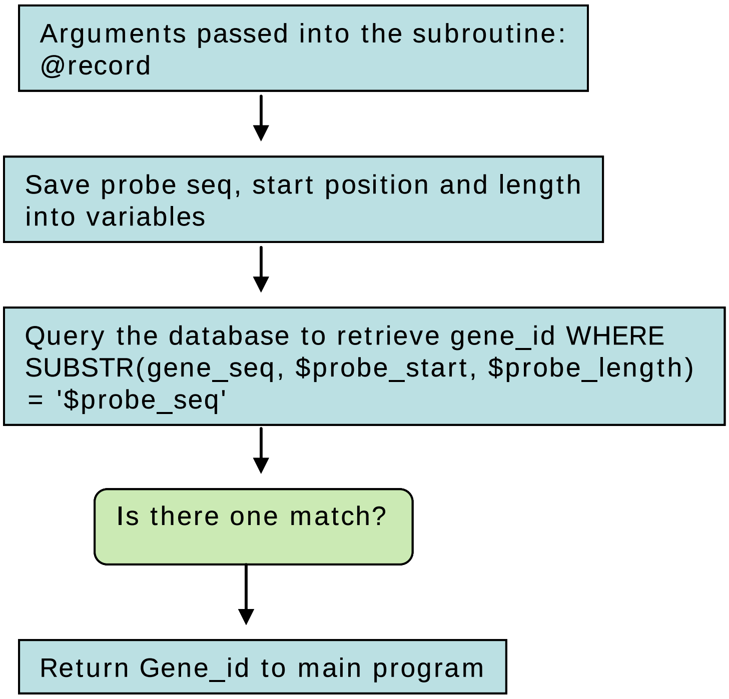

Change the dir. to where assay files are located
Does dir. exist?
yes
no
Terminate program and inform user
Dir contains .txt files?
no
yes
Terminate program and inform user
Open each .txt file in turn & test whether first line matches the header row of a Primer Express file
Match?
no
Save file name to an array Called @not_assay_file
yes
Save file name to an array called @assay_file
Does @assay_file contain values?
no
Terminate Program and inform user
yes
Pass @assay_file into check DB subroutine which returns files that do not exist in DB. Saved in an array @not_in_DB
Does @not_in_DB contain values?
no
Terminate Program and inform user
yes
Open file, and save data to an array @record
no
yes
Last element?
Retrieve gene_id by passing @record into gene_id subroutine
Is there and exact gene match?
no
Go to next file in @not_in_DB
yes
Load data into taqman file table by passing @record into load file subroutine
Retrieve last insert id i.e.file_id and save as $insert_id
Load data into assay and primer probe table by passing $insert_id, @record into load assay subroutine

SUBROUTINES

Sub check_DB


Arguments passed into the subroutine: assay_file names
Foreach file, perform an SQL query to retrieve file_name from Taqman file subroutine
File in DB?
yes
Go to next file in the array
no
Save to an array @not_in_DB
Return @not_in_DB values to main program

Sub gene_id


Arguments passed into the subroutine: @record
Save probe seq, start position and length into variables
Query the database to retrieve gene_id WHERE SUBSTR(gene_seq, $probe_start, $probe_length) = '$probe_seq'
Is there one match?
Return Gene_id to main program

Appendix 2

Contents of the accompanying CD-ROM

Below is a list of files that are included on the accompanying CD-ROM. They have been cross-referenced throughout this thesis.

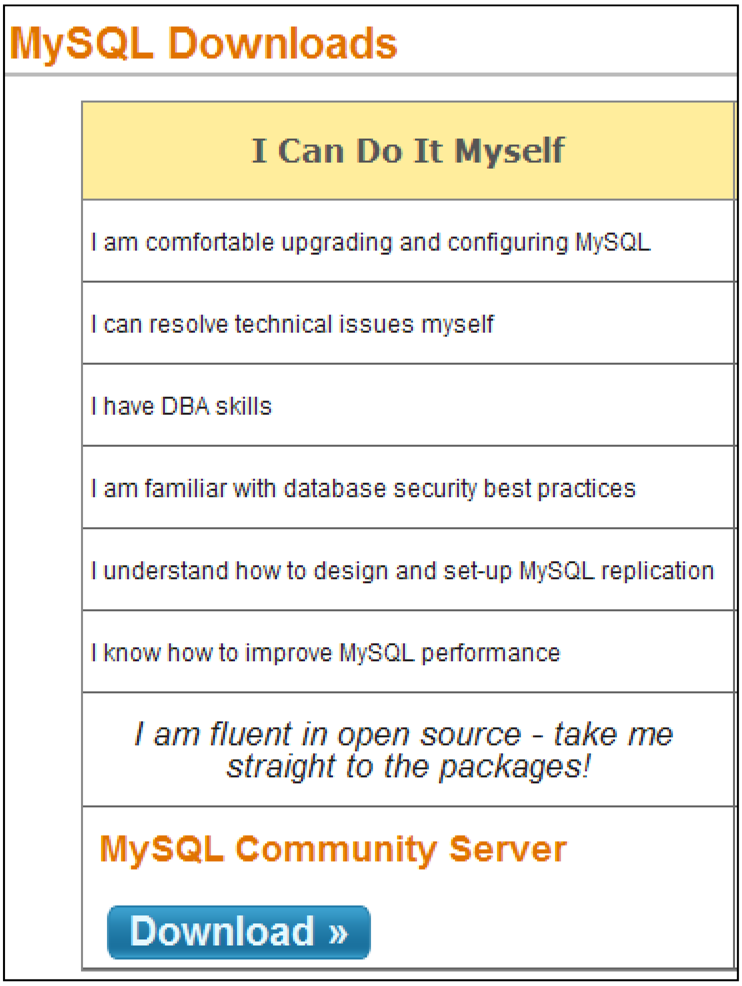
- **Zipped folder entitled Taqbase**. This contains all the necessary components to install TaqBase on a blank system. The following items are located within this folder:
    - **CGI Programs Folder.** This contains all the CGI programs that were written for user interface implementation. There are a total of 19 CGI programs.
    - **Templates Folder.** This contains 5 HTML template files.
    - **Perl Modules Folder.** This contains the Perl modules that are required for TaqBase to be fully functional
    - A mysqldump file called taqbase.sql.
- **Perl Programs Folder.** This contains all 3 Perl programs written for this project
- There are two text files; an example of a GenBank record and a Primer Express file.
- One pdf file containing an explanation of how penalty scores are calculated.
- One SQL file containing CREATE table statements for this database.

Appendix 3

User guide for installing the database and web-based user interface locally on Windows XP

1. Download and Install MySQL

- Click on the downloads link on the MySQL homepage http://www.mysql.com/ . Download MySQL community server by clicking on download as shown below:



MySQL Downloads

I Can Do It Myself

I am comfortable upgrading and configuring MySQL

I can resolve technical issues myself

I have DBA skills

I am familiar with database security best practices

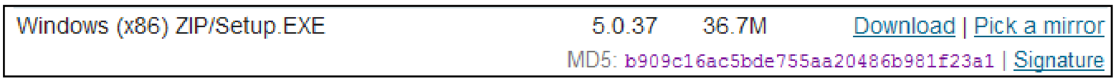
I understand how to design and set-up MySQL replication

I know how to improve MySQL performance

I am fluent in open source - take me straight to the packages!

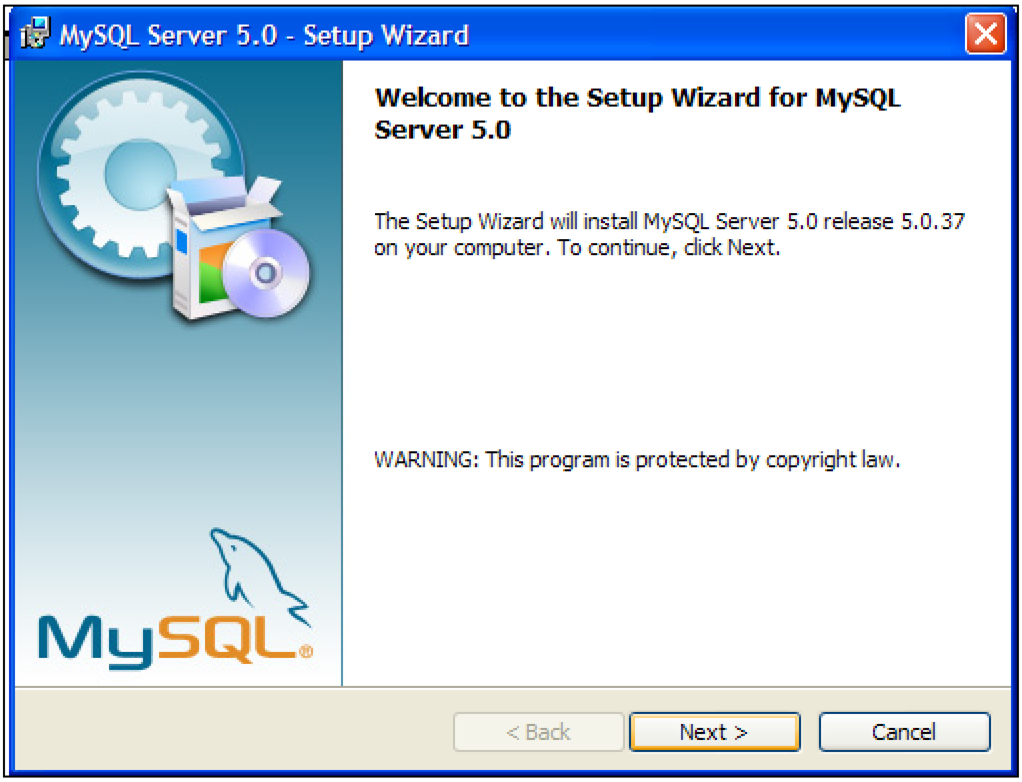
MySQL Community Server

- Scroll down the Web page until the following is reached:



Windows (x86) ZIP/Setup.EXE 5.0.37 36.7M Download | Pick a mirror
MD5: b909c16ac5bde755aa20486b981f23a1 | Signature

- Click on the Download link and when the file download window pops up, click on Save
- Save the zipped file to a temporary location such as your Desktop
- Close the dialog box when the download is complete
- To install MySQL, unzip the downloaded file to a temporary location
- Double click on the Setup icon and then click on run to launch the Setup wizard:



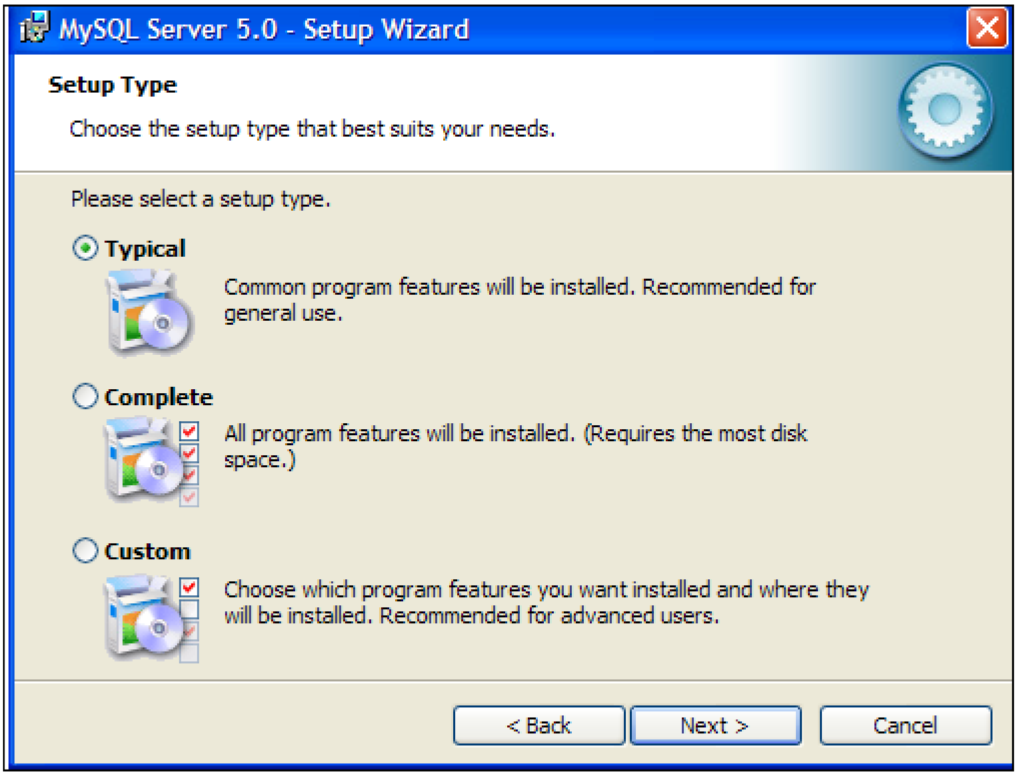
MySQL Server 5.0 - Setup Wizard

Welcome to the Setup Wizard for MySQL Server 5.0

The Setup Wizard will install MySQL Server 5.0 release 5.0.37 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law.

- Click on Next



MySQL Server 5.0 - Setup Wizard

Setup Type

Choose the setup type that best suits your needs.

Please select a setup type.

● **Typical**

Common program features will be installed. Recommended for general use.

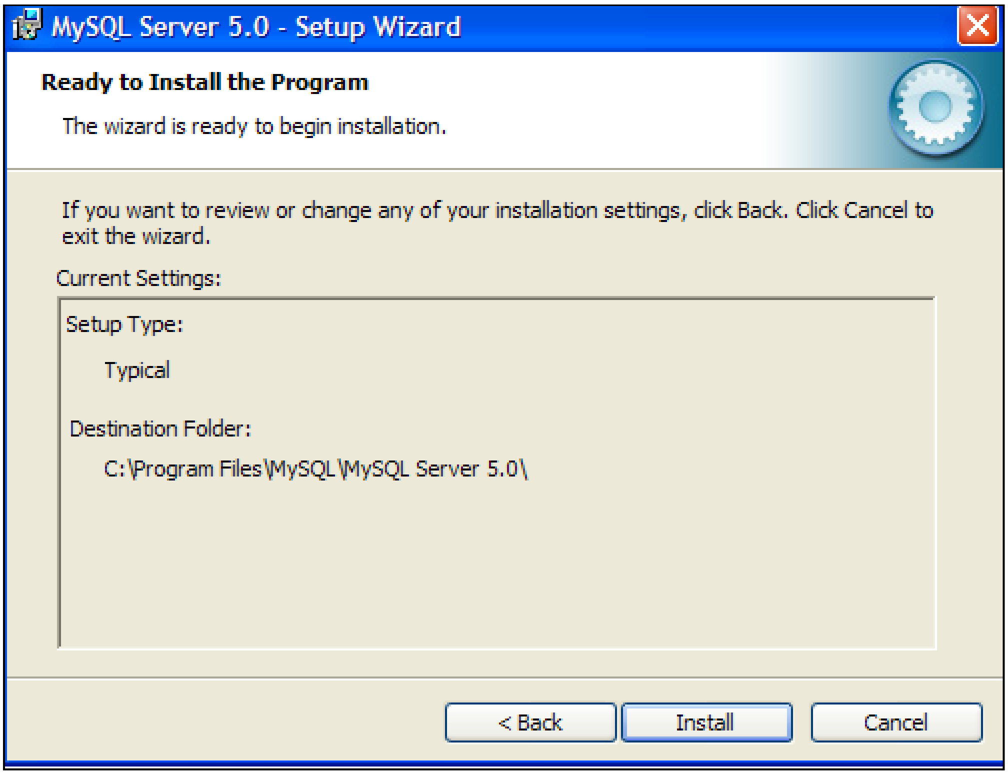
○ **Complete**

All program features will be installed. (Requires the most disk space.)

○ **Custom**

Choose which program features you want installed and where they will be installed. Recommended for advanced users.

- Click on Next again



MySQL Server 5.0 - Setup Wizard

Ready to Install the Program

The wizard is ready to begin installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Current Settings:

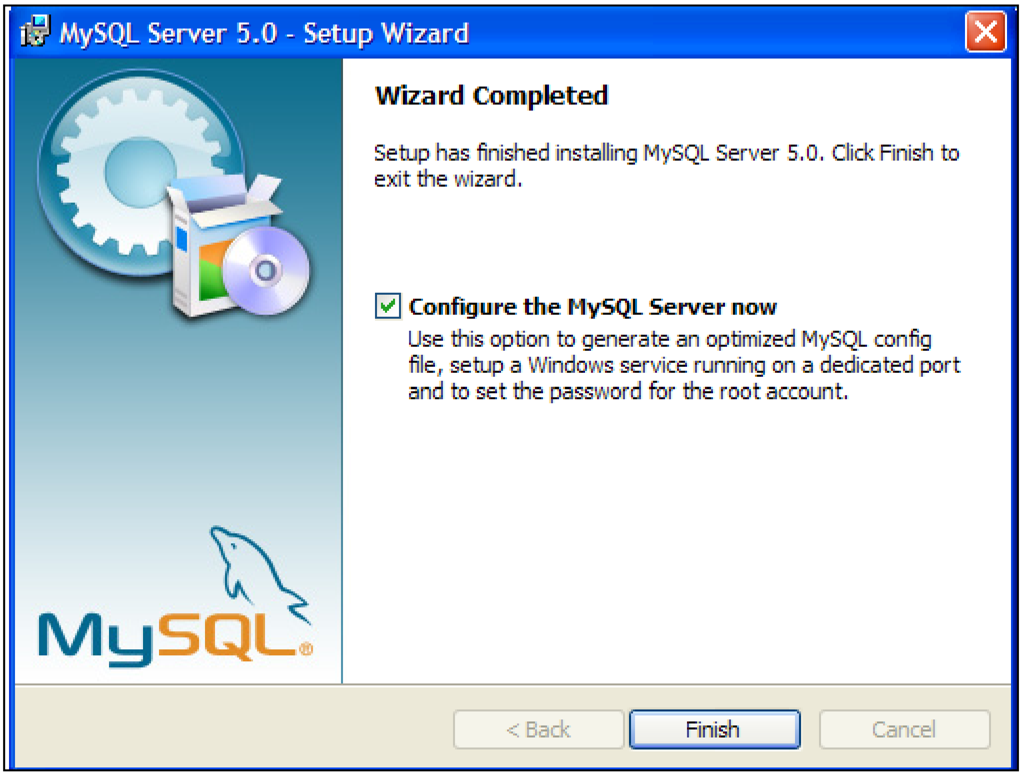
Setup Type:

Typical

Destination Folder:

C:\Program Files\MySQL\MySQL Server 5.0\

- Click on Install
- When installation is complete, a sign up window appears to create a MySQL.com account. If required, sign up to MySQL.com, otherwise click on the skip option and then click on Next



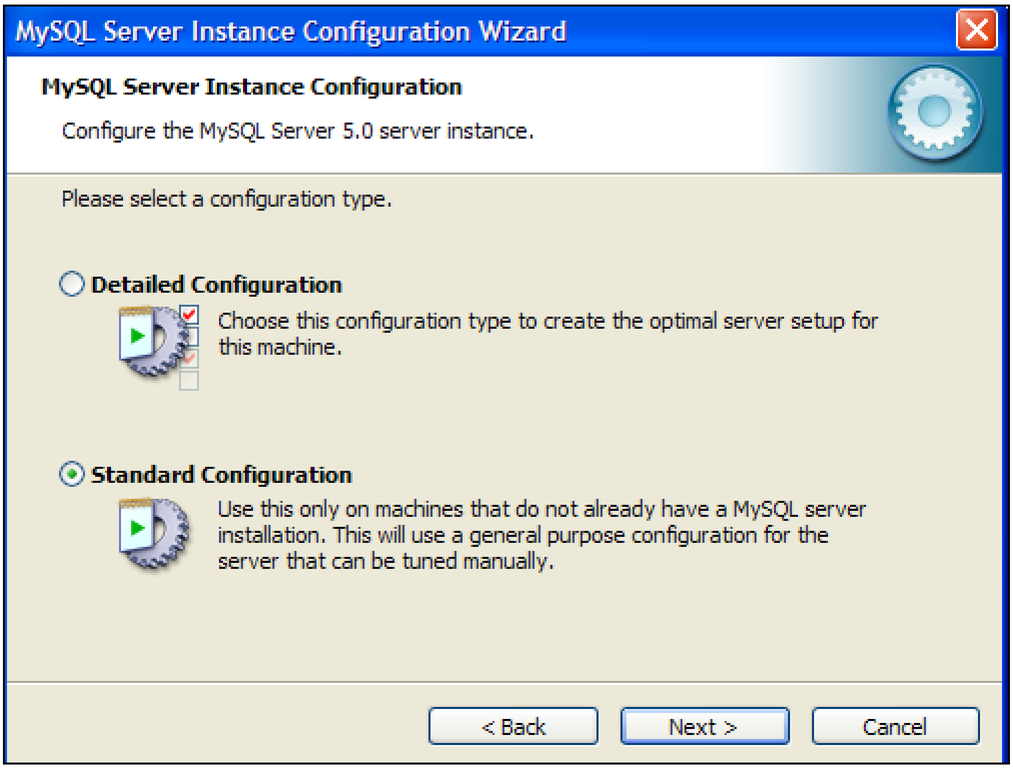
MySQL Server 5.0 - Setup Wizard

Wizard Completed

Setup has finished installing MySQL Server 5.0. Click Finish to exit the wizard.

☑ **Configure the MySQL Server now**
Use this option to generate an optimized MySQL config file, setup a Windows service running on a dedicated port and to set the password for the root account.

- Ensure that Configure the MySQL Server now is selected and click on Finish
- Then click on Next to configure MySQL



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Please select a configuration type.

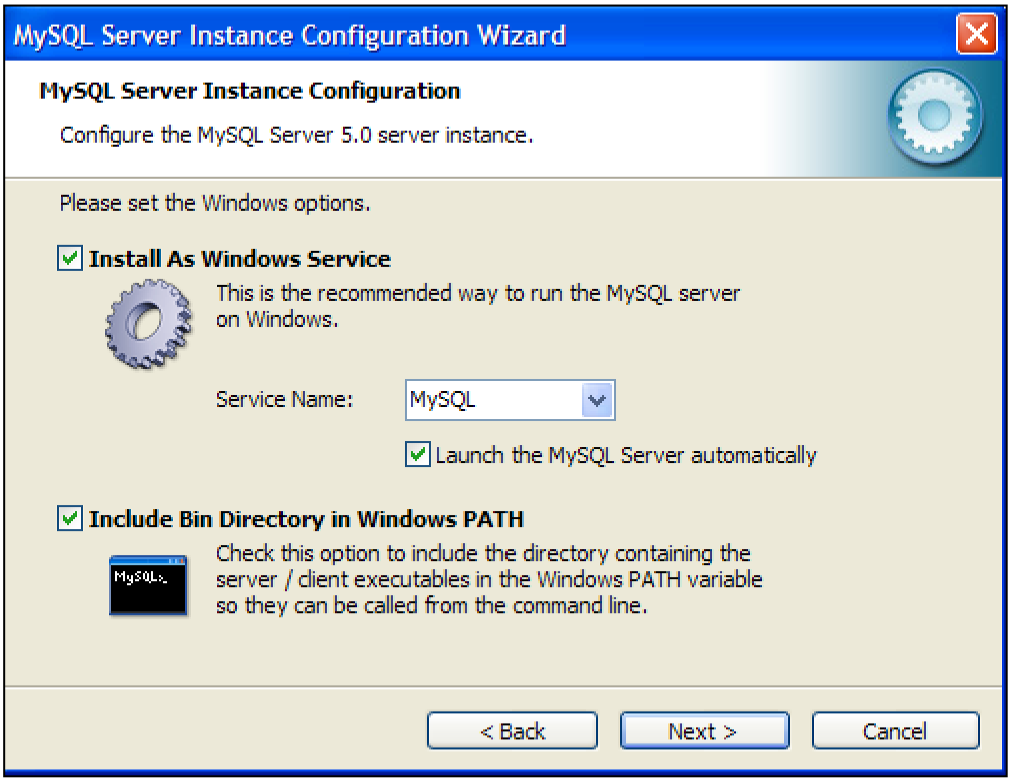
○ **Detailed Configuration**

Choose this configuration type to create the optimal server setup for this machine.

● **Standard Configuration**

Use this only on machines that do not already have a MySQL server installation. This will use a general purpose configuration for the server that can be tuned manually.

- If MySQL server is not on your machine, select Standard Configuration option and then click on Next



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Please set the Windows options.

☑ **Install As Windows Service**

This is the recommended way to run the MySQL server on Windows.

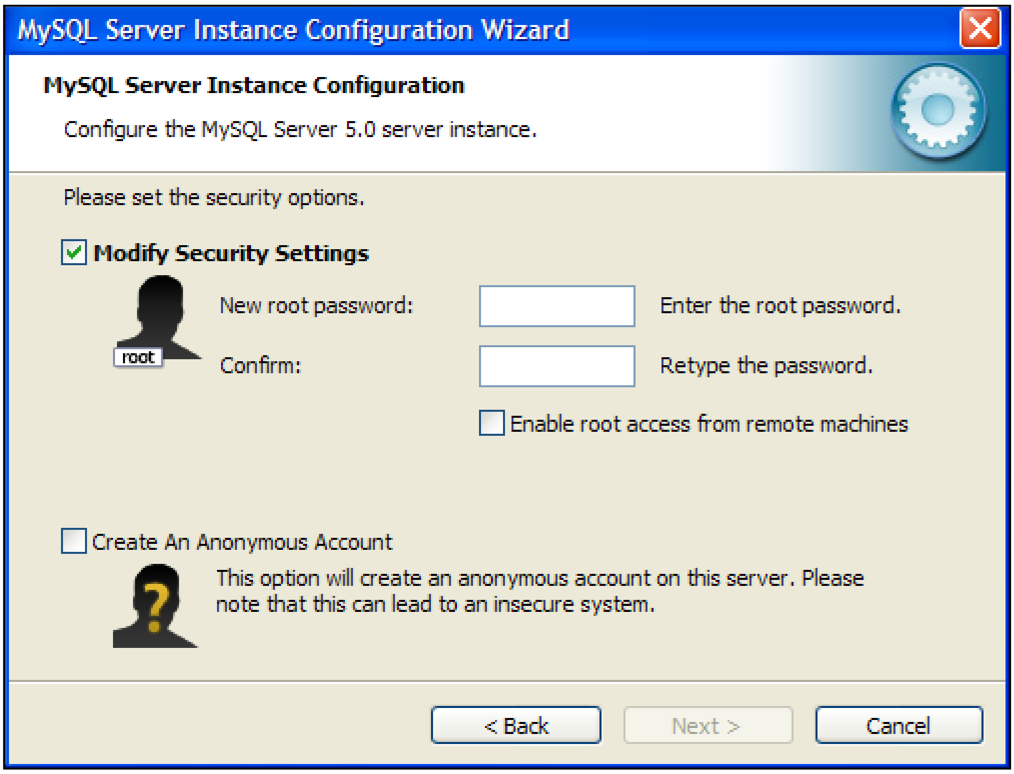
Service Name: MySQL

☑ Launch the MySQL Server automatically

☑ **Include Bin Directory in Windows PATH**

Check this option to include the directory containing the server / client executables in the Windows PATH variable so they can be called from the command line.

- Ensure that Include Bin Directory in Windows PATH is selected and then click on Next



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Please set the security options.

☑ **Modify Security Settings**

New root password: Enter the root password.

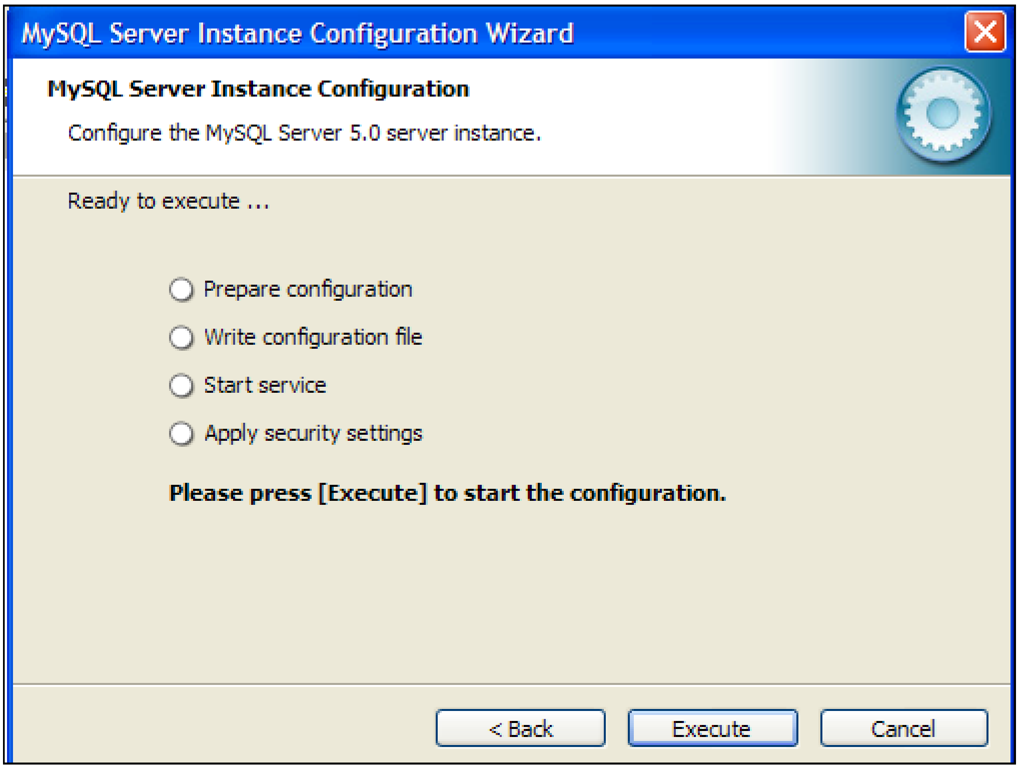
Confirm: Retype the password.

□ Enable root access from remote machines

□ Create An Anonymous Account

This option will create an anonymous account on this server. Please note that this can lead to an insecure system.

- Enter and confirm password of choice and then click on Next



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Ready to execute ...

○ Prepare configuration

○ Write configuration file

○ Start service

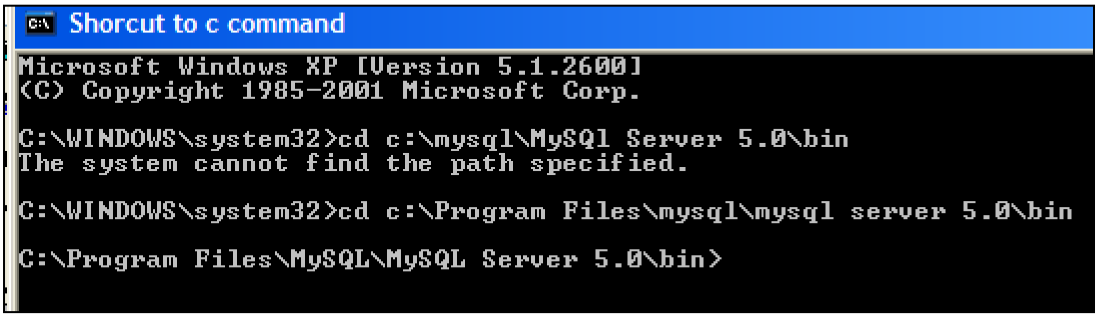
○ Apply security settings

Please press [Execute] to start the configuration.

- Click on Execute and when complete click on Finish to close the Wizard

2. Create TaqBase

- Unzip the TaqBase folder on the supplied CD-ROM to a temporary directory
- Transfer TaqBase.sql located within the TaqBase folder to the following directory: *your file path*/MySQL/MySQL Server 5.0/bin
- Go to the command prompt and change the directory by typing: cd c:\your file path path\mysql\mysql server 5.0\bin for example:



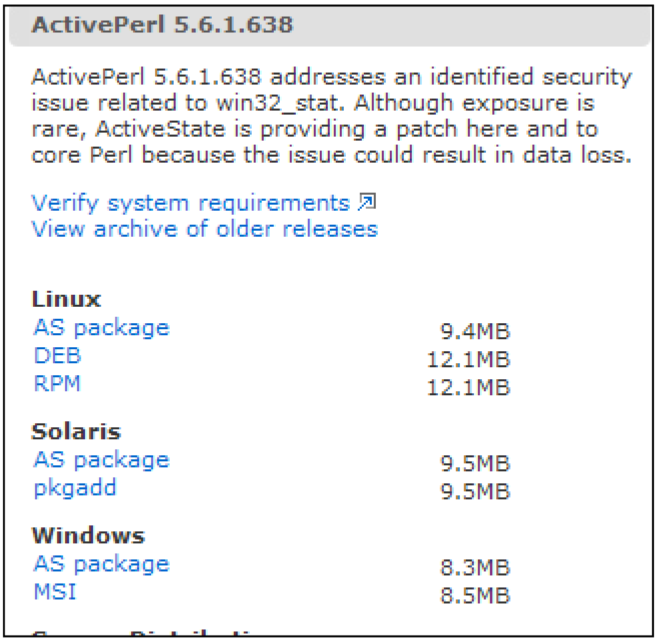
Shorcut to c command

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>cd c:\mysql\MySQl Server 5.0\bin
The system cannot find the path specified.

C:\WINDOWS\system32>cd c:\Program Files\mysql\mysql server 5.0\bin

C:\Program Files\MySQL\MySQL Server 5.0\bin>
```

- At the prompt, type Mysqladmin –uroot –p*password* create taqbase replacing ‘password’ with the password that was entered during the configuration of MySQL
- At the command prompt, change the directory as follows: cd c:\your file path\mysql\mysql server 5.0\data
- Then type mysql –hlocalhost –uroot -p*password* taqbase < taqbase.sql again replacing ‘password’ with the password that was entered during the configuration of MySQL
- Creation and population of the TaqBase should now be complete

3. Install Perl

- The Perl application can be found at http://www.activestate.com/downloads/
- Scroll down to the bottom of the webpage until ActivePerl 5.8.8.820, ActivePerl 5.6.1.638 is reached
- Click on get current release and then click on the Download button
- Enter contact details (optional) and then click on the Continue button
- Under the ActivePerl 5.6.1.638 heading as shown below, click on the MSI link under the Windows heading and then click on save



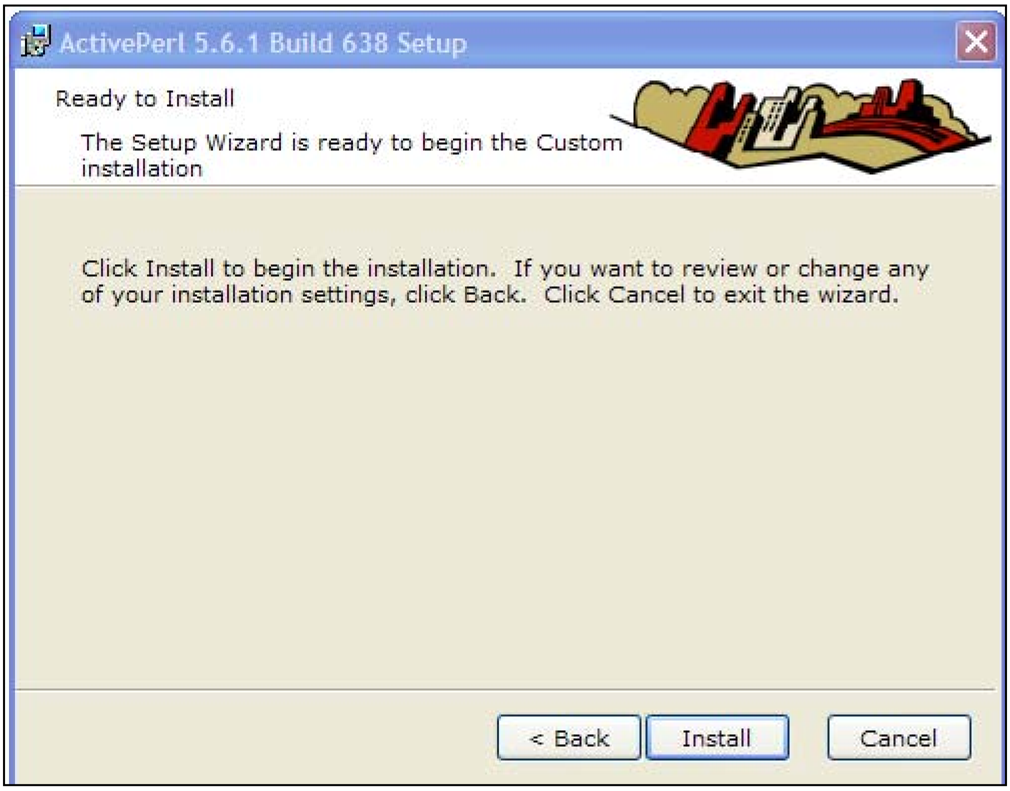
ActivePerl 5.6.1.638

ActivePerl 5.6.1.638 addresses an identified security issue related to win32_stat. Although exposure is rare, ActiveState is providing a patch here and to core Perl because the issue could result in data loss.

Verify system requirements ↗
View archive of older releases

Linux	
AS package	9.4MB
DEB	12.1MB
RPM	12.1MB
Solaris	
AS package	9.5MB
pkgadd	9.5MB
Windows	
AS package	8.3MB
MSI	8.5MB

- Download the file to a temporary directory and close the message box when download is complete
- Go to the folder where you have downloaded ActivePerl-5.6.1.638-MSWin32-x86 Windows Installer package and double click on it to launch the Wizard
- Once in the Wizard click on Next until the following is reached:



ActivePerl 5.6.1 Build 638 Setup

Ready to Install

The Setup Wizard is ready to begin the Custom installation

Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

- Click on Install and then click on Finish when installation is complete
- Perl should now be installed and ready to use

4. Install Perl Modules

The following Perl modules should be installed:

Time::Piece::MySQL

DBI

DBD::MySQL

Bio::Perl

HTML::Template

IO::String

- These modules can be downloaded from

http://ppm.activestate.com/PPMPackages/zips/ by clicking on this link: ActivePerl 6xx (e.g. ActivePerl 5.6.1.623)

- Locate the following zip files and click on the link to download the zip file to a temporary directory on you PC:

DBD-mysql-2.9004.zip

DBI.zip

Time-Piece-MySQL-0.03.zip

IO-String.zip

HTML-Template.zip

- Alternatively these zip files can be found in ‘Perl Modules for installing’ folder in the Taqbase folder that has already been unzipped
- Extract each of the zip files into a temporary directory
- To install each of the Perl modules, go to the command prompt and change the directory to where the ‘.ppd’ file is located within your unzipped file, for example type `cd C:\`*`Your file path`*`\CGI-3.00`
- Then at the command prompt type `ppm install `*`modulename`*`.ppd` replacing ‘*modulename*’ with the name of the ppd file.
- To download and install Bioperl go to here: http://bioperl.open-bio.org/wiki/Getting_BioPerl
- Scroll down to Bioperl 1.4.0, Stable Release heading
- Click on zip next to core modules and download zipped file into temporary directory on your PC
- Extract the files to a temporary location and then transfer the ‘Bio’ folder to

the \Perl\site\lib directory. The Bio folder is also supplied in the ‘Perl Modules for installing’ folder in the Taqbase folder you have already unzipped

5. Download and Install Apache Server

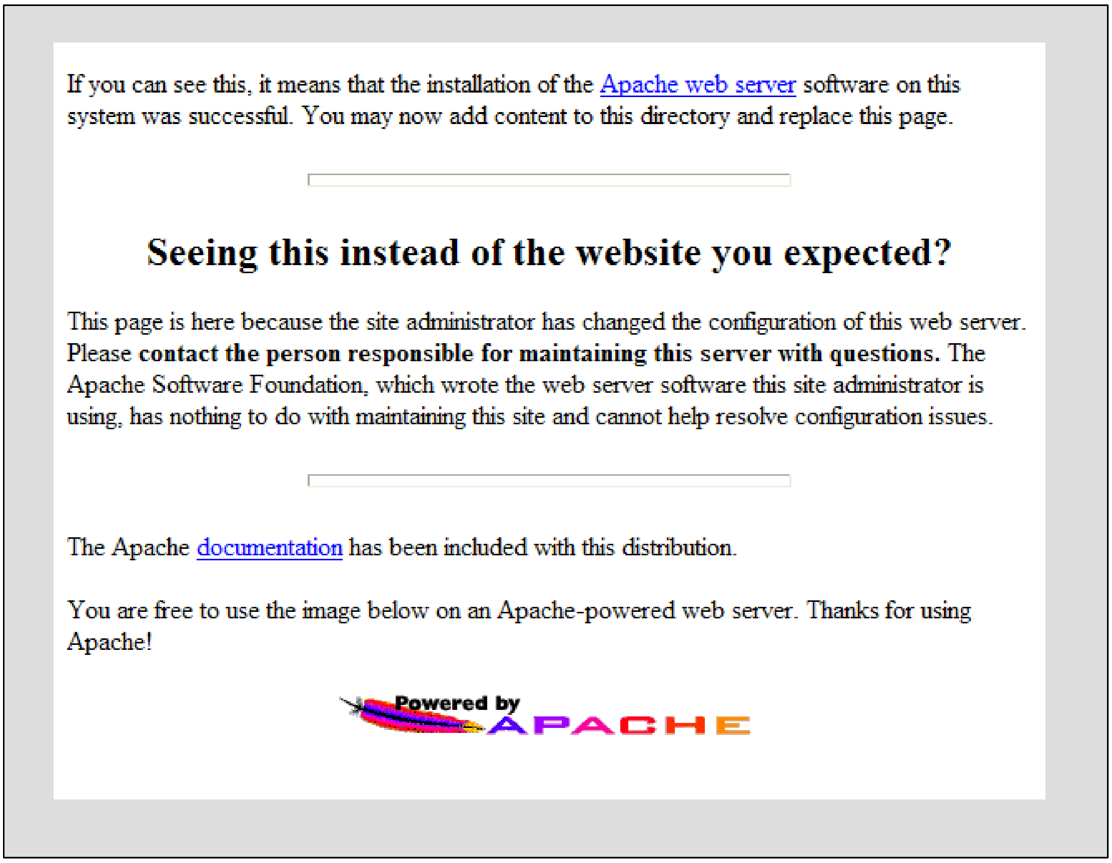
- Go to this website: http://httpd.apache.org/download.cgi and scroll down to here :

Apache HTTP Server 2.0.59 is also available

- Click on this link: Win32 Binary (MSI Installer): apache_2.0.59-win32-x86-no_ssl.msi
- Save the Windows installer package to a temporary directory
- Double click on the installer package and follow the instructions in the installation wizard.
- Keep clicking on Next until the server information window is reached
- For local installation of Apache type locahost for both network domain and the server name
- Keep clicking on Next, keeping all default options until the install window is reached and then click on Install
- Click on Finish when the software installation is complete
- To configure Apache, open the configuration file entitled httpd.txt. This file is located here: *Your File path*\Apache Group\Apache2\conf
- Search for this in the file:

```
#<Directory "C:/Documents and Settings/*/My Documents/My Website">
#    AllowOverride FileInfo AuthConfig Limit
#   Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
#   <Limit GET POST OPTIONS PROPFIND>
#       Order allow,deny
#       Allow from all
#   </Limit>
#   <LimitExcept GET POST OPTIONS PROPFIND>
#       Order deny,allow
#       Deny from all
#   </LimitExcept>
#</Directory>
```

- Remove # from each line and replace the file path in <Directory "C:/Documents and Settings/*/My Documents/My Website> with "*Your filepath*/Apache Group/Apache2cgi-bin"
- If the software has been installed and configured correctly you should see the following webpage at this url: http://localhost/ if installed locally (replace localhost with hostname if installed elsewhere)



If you can see this, it means that the installation of the Apache web server software on this system was successful. You may now add content to this directory and replace this page.

Seeing this instead of the website you expected?

This page is here because the site administrator has changed the configuration of this web server. Please **contact the person responsible for maintaining this server with questions.** The Apache Software Foundation, which wrote the web server software this site administrator is using, has nothing to do with maintaining this site and cannot help resolve configuration issues.

The Apache documentation has been included with this distribution.

You are free to use the image below on an Apache-powered web server. Thanks for using Apache!

5. Install CGI files and template files

- Locate the CGI programs folder, which also contain the TaqLibrary.pm perl module, within the Taqbase folder you have already unzipped.
- Transfer all programs within the folder to this location: *Your file path*\Apache Group\Apache2\cgi-bin
- Open the Taqlibrary.pm in a text editor such as notepad and make the changes requested in the module and then save.
- If you access websites via a proxy server open genbank_results.cgi in a text editor and remove the # from the following:

```
#configureProxy($query_obj);
```

```
#configureProxy($dbh);
```

- Re-save the file
- Locate the Templates folder within the Taqbase folder
- Transfer the folder and its contents to *Your file path* C:\Program Files\Apache Group\Apache2\htdocs

6. Launch TaqBase

If all the previous steps have been followed correctly, TaqBase should be ready to use as follows:

- Launch your Browser
- At the browser type: http://127.0.0.1/cgi-bin/welcome.cgi to launch TaqBase. This will only work for locally installed software. Replace 127.0.0.1 with alternative IP address if installed elsewhere

7. Uploading data into the database

All Perl programs for uploading data can be found on the attached CD-ROM in the Perl Programs directory. Only text files that have been exported from Primer Express software can be uploaded into the database. An example file is located on the attached CD-ROM. To run the upload_gene program a text file containing a list of accession numbers is required, with one accession on each row. The accession numbers should correspond to the assays that have been designed.

- Save the Perl programs and taqlibrary.pm to your working directory.
- At the command prompt change your directory to where your Perl files are

located using the **cd** command.

- First run the Upload_Gene.pl program at the command prompt as follows:

```
> upload_gene.pl.
```

- Then run the Upload_assay.pl program at the command prompt as follows:

```
> upload_assay.pl
```

- Your database should now be populated with data.