AN UPDATED OBJECT ORIENTED BOVINE QTL VIEWER AND GENOME-WIDE QTL META-ANALYSIS

A Dissertation

by

HANNI SALIH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2008

Major Subject: Genetics

AN UPDATED OBJECT ORIENTED BOVINE QTL VIEWER AND GENOME-WIDE QTL META-ANALYSIS

A Dissertation

by

HANNI SALIH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	David Adelson
	James Womack
Committee Members,	Christine Elsik
	Richard Furuta
Chair of Genetics Faculty,	James Wild

August 2008

Major Subject: Genetics

ABSTRACT

An Updated Object Oriented Bovine QTL Viewer and Genome-wide QTL Meta-analysis. (August 2008)

Hanni Salih, B.S., Baylor University

Co-Chairs of Advisory Committee: Dr. David Adelson
Dr. James Womack

Waves of bovine genomic data have been produced as a result of the bovine genome sequencing projects. In addition to the massive amounts of genomic sequence, significant annotation including single nucleotide polymorphisms, sequence tagged sites and haplotype blocks have been produced by the Bovine HapMap Project. Furthermore, many agriculturally significant traits in cattle such as milk yield and carcass weight are measured on a quantitative scale and have been genetically mapped as quantitative trait loci (QTL). QTL data can be used to generate another form of bovine annotation linking phenotype to genotype. However, it is impossible for humans to be able to analyze genomic scale data without computer based tools. Bioinformatic tools have been shown to greatly increase productivity and improve efficiency when dealing with large data sets.

My dissertation presents an integrated, extensible database that houses SNPs, STSs, haplotypes, and QTL. The database is presented to researchers through a restructured, object oriented Bovine QTL Viewer that displays multiple levels of bovine annotation synergistically. Evaluation of use of the viewer was performed using a survey based approach and measured quantitatively.

In addition, the QTL data from the database was used to analyze the frequency of gene ontology (GO) annotations within QTL regions. QTL regions were divided into 8 trait based groups. GO terms were counted within each category of QTL and in non-QTL regions of the genome. Top level GO term frequencies were generated from the counts and these frequencies were compared between QTL and non-QTL portions of the genome. Furthermore, specific sets of GO terms believed to be related to QTL categories were also used to determine if QTL regions were enriched for genes annotated with such GO terms. As a result, we determined that gene density varied significantly across QTL regions and that many QTL categories showed GO term frequency differences that could be related to the trait's biology. Furthermore, our selected GO term sets were shown to be significantly enriched in some QTL categories.

ACKNOWLEDGEMENTS

I would like to thank my primary advisor Dr. Dave Adelson. His dedication, wealth of knowledge and patience has been instrumental towards completion of this project and my development as a scientist. I am honored and very grateful to have participated in his lab.

I would like to extend thanks to my committee members Dr. Chris Elsik, Dr. Richard Furuta and Dr. James Womack. I am grateful for your support and efforts to help steer my graduate studies and this work.

I would also like to thank my fellow lab members Edward Ning Wei and Johnathan Rios. My studies were greatly enriched by being able to discuss ideas and problem resolutions with them. It was both a pleasure and an honor to be able to work in this lab.

Finally, I would unquestionably like to thank my family for their unwavering support. Through difficult times they have been both encouraging and helpful. Without them, surely this work would not have been possible.

NOMENCLATURE

QTL	Quantitative Trait Locus
DNA	Deoxyribonucleic Acid
SNP	Single Nucleotide Polymorphism
GO	Gene Ontology
STS	Sequence Tagged Site
GMOD	Generic Model Organism Database
SO	Sequence Ontology
LD	Linkage Disequilibrium
GOOF	Generic Object Oriented Framework
Btau4	Bovine Genome Fourth Assembly

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

1. Abundance of Novel Genomic Data

Since the domestication of cattle, man has tried to select for and promote useful traits in dairy and beef cattle. Today, researchers can use their knowledge of the composition and organization of the bovine genome to accelerate bovine genetic improvement. There have been three assemblies of the bovine genome to date and a fourth and final draft version has been recently released. The current wealth of sequence information available for the bovine genome has spurred the need for annotation and subsequent biological interpretation of the genome sequence. Bioinformatic tools are effective measures for bridging the gap between sequence annotation and biological interpretation. Furthermore, haplotypes have increasingly become useful genomic features for interpreting phenotype [1]. Understanding the relationships between haplotypes, SNPs and QTL can help researchers identify functional regions of the bovine genome and therefore, allow cattle researchers to more efficiently enhance economically important traits. Many humans are better able to understand and identify patterns/relationships when visually presented [2]. Visualizing the relationships between

This dissertation follows the style of *BMC Bioinformatics.*

types of bovine sequence annotation can help researchers to identify functionally significant sequences. Unfortunately, there was no integrated database housing multiple layers of sequence annotation including, but not limited to: haplotypes, SNPs, and QTL.

2. Tool Based Solution

A visual tool that can clearly display multiple levels of bovine genome annotation can be of great benefit for cattle researchers investigating the links between genotype and phenotype. In fact, there is strong demand for the development of tools to facilitate association studies [3-17]. Within the bovine research community a number of bioinformatics tools have been created to help analyze various aspects of the bovine genome [10, 14, 16]. ARKdb is a livestock genetic map viewer that handles animal genome mapping data and provides tools for data entry and display [11]. The database stores locus/marker data, references/papers, authors, genetic(linkage) map assignments, cytogenetic assignments, experiments, experimental techniques and results, PCR primer information, PCR conditions, hybridization conditions and enzyme information[11]. Databases are available that store QTL for non-bovine model organisms [18]. RatMap was an early model organism database and resource for the rat genome. This tool combined the integrated linkage map with gene sequence positions obtained from BLAST alignments of 4170 rat gene sequences [18]. This integrated linkage map made it possible to view polymorphic DNA-markers and QTLs in a single table with positions in base pairs and cM.

The Jackson labs have also contributed significant bioinformatics tools catered for mice genomics. A comprehensive database has been created to house multiple layers of annotation for the mouse genome [19, 20]. Nonetheless, there was no database that integrated bovine annotation QTL, SNPs and haplotype information into a single location. This need has been exacerbated by the release of the fourth bovine genome assembly.

In order to address this need, we have created a second generation Bovine QTL Viewer database that houses publicly available bovine SNPs, haplotype blocks, markers and QTL positioned on the fourth assembly of the bovine genome. Previous data from the first generation Bovine QTL Viewer database has been converted to new sequence coordinates in order to anchor QTL to the fourth genomic assembly. The data was then migrated to a new schema and additional bovine genomic annotation (e.g. SNPs and haplotype blocks) have been included.

Since the purpose of doing this was to make the data fully accessible to the cattle research community, a completely re-structured interface was created to search and display the integrated data. The new viewer presents data stored in the second generation database through the World Wide Web. The viewer was created using java object oriented programming and was implemented using the GOOF package. To our knowledge, this viewer is the first tool that integrates bovine QTL, SNPs and haplotypes with respect to the bovine genome fourth assembly and gene annotation.

Association studies are used to compare the prevalence of a particular genetic marker (i.e. SNP), or set of markers (i.e. haplotypes, QTL), in affected and unaffected individuals [21, 22]. The study of associations between traits and SNPs has in some cases revealed the genetic basis behind phenotype [3, 5, 17, 23]. The first step in case control association studies relies on TagSNPs to associate a haplotype block with phenotype [24]. Subsequent retrieval and analysis of candidate SNPs within the block can precisely isolate which marker(s) are associated with phenotype [25]. The greater the prevalence of a marker in affected individuals, the more likely it is evidence of an association between the disease phenotype and the marker(s) [22]. Early association studies were limited by the modest number of known polymorphisms [3]. However, recent technological advances have generated a large increase in the number of known polymorphisms and therefore, a renewed enthusiasm for association studies. Unfortunately, the large number of available candidate SNPs has made it difficult for the average biologist to identify and use the best candidates. Furthermore, in order to facilitate association studies, the SNPs must be validated, their linkage disequilibrium patterns described, and allele frequencies in various geographic and phenotypic groups must be determined. Such data must then be sorted and analyzed for functional annotation. The Bovine QTL Viewer and database can provide a resource to facilitate the genetic mapping of traits in cattle.

3. Genome Wide QTL Meta-analysis

Our QTL database is a unique resource and gives us the ability to analyze the distribution of QTL with respect to various genomic features and known functional sequences. As our initial meta-analysis we examined the association of functionally classified genes with QTL. To this end we measured gene density and gene ontology term frequency within QTL regions. QTL regions were pooled into 8 groups based on type. These eight QTL groups consisted of: Body Conformation, Carcass, Disease Resistance, Fat Metabolism, Growth, Milk Protein, Milk Yield and Reproduction. In addition, genomic regions where no QTL had been mapped were used as a control and are designated as non-QTL regions. Gene density and GO term frequencies were compared between QTL, non-QTL regions and the full genome to determine if there was evidence of functional clustering of genes within QTL regions. Our analysis revealed that QTL and non-QTL regions differ significantly in gene density and that evidence of functional clustering could be determined for some GO terms with some QTL classes.

CHAPTER II

BOVINE QTL VIEWER SECOND GENERATION DATABASE

1. Introduction

The Bovine QTL Viewer database has served the research community as an integrated database and repository for bovine quantitative trait loci (QTL) [14]. The QTL data and associated marker data were gathered from existing databases USDA-MARC and INRA. QTL were also gathered manually by manuscript analysis carried out by curators. The resulting relational database is designed to seamlessly integrate bovine QTL and marker data to allow users to obtain and view QTL data from disparate experiments across the bovine genome. While various types of data are stored in the database, the central data are QTL and other additional related information such as markers and references linked to the QTL.

Pertinent information regarding each individual QTL is stored within the QTL_INFO table of the Bovine QTL Viewer database. QTL are uniquely identified by the qtl_id attribute of the QTL_INFO table of the database. Qtl_id further serves as the primary key for the table. For each QTL, a lod score, f-statistic, the family from which the QTL was derived, and method of derivation are stored within the table. The literature which details identification of each QTL is stored into the bovine QTL Viewer database. A unique id is assigned for each reference thereby allowing multiple QTL to refer to single articles of literature. The title, citation, URL and year are readily stored. In addition, significance level and an edit history are also stored. QTL are positioned by flanking markers also known as sequence tagged sites. In order to map all QTL to a single genetic linkage map, all markers in the Bovine QTL Viewer database are part of the USDA-MARC linkage map. Details of each marker are stored in the marc_marker_info table of the database. Each QTL is associated with the particular trait and each trait is a member of a “category”.

The database has been modeled in order to allow each trait to fall within a few main biologically relevant categories (Figure 1). This allows researchers to search the database based on a general area or category of QTL they are interested in. Query results are limited to QTL that are within each broad selected category.

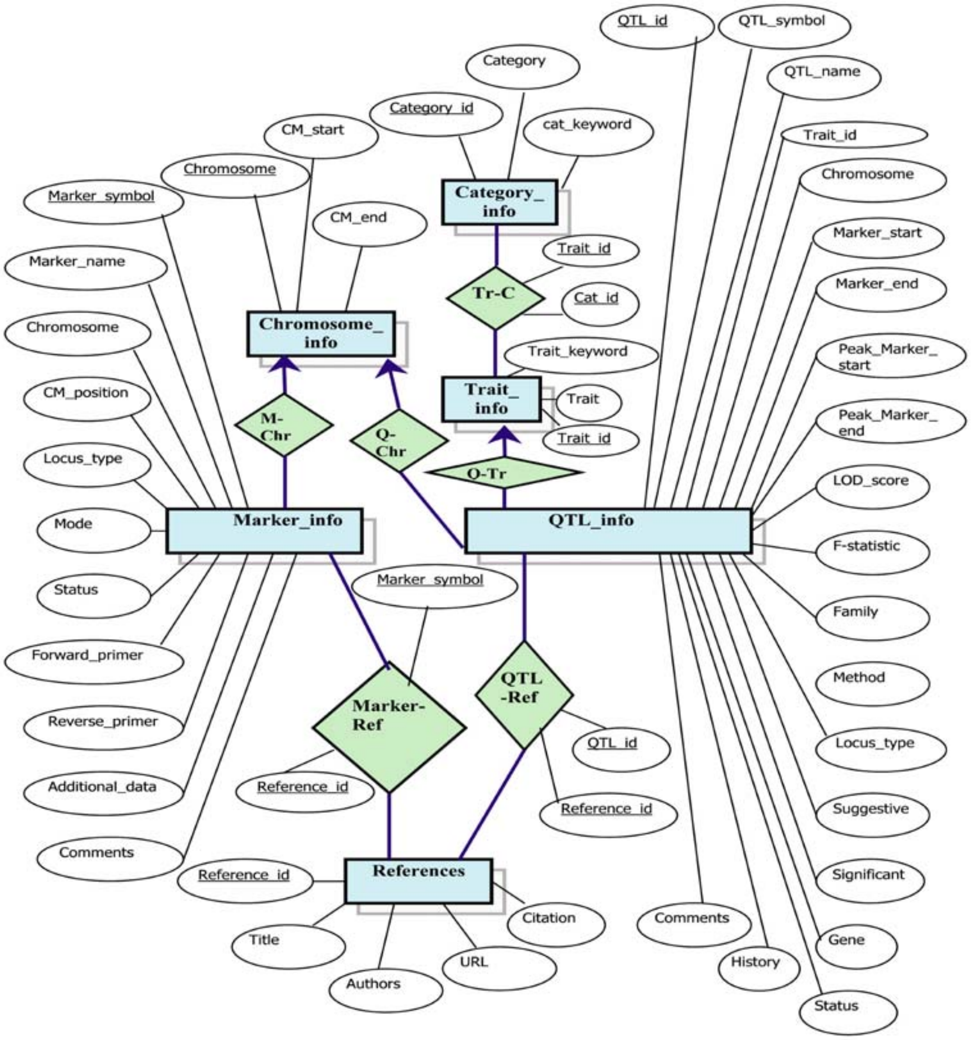


Figure 1 - Bovine QTL Viewer database E-R diagram.

While the previous Bovine QTL Viewer database is well suited for the viewer, its unique structure limits its integration with other databases. In order to overcome this, the backend database has been integrated into the Chado schema [9].

The Chado schema is a publicly available relational database schema that underlies many GMOD installations (Figure 2) [26]. GMOD is the result of a

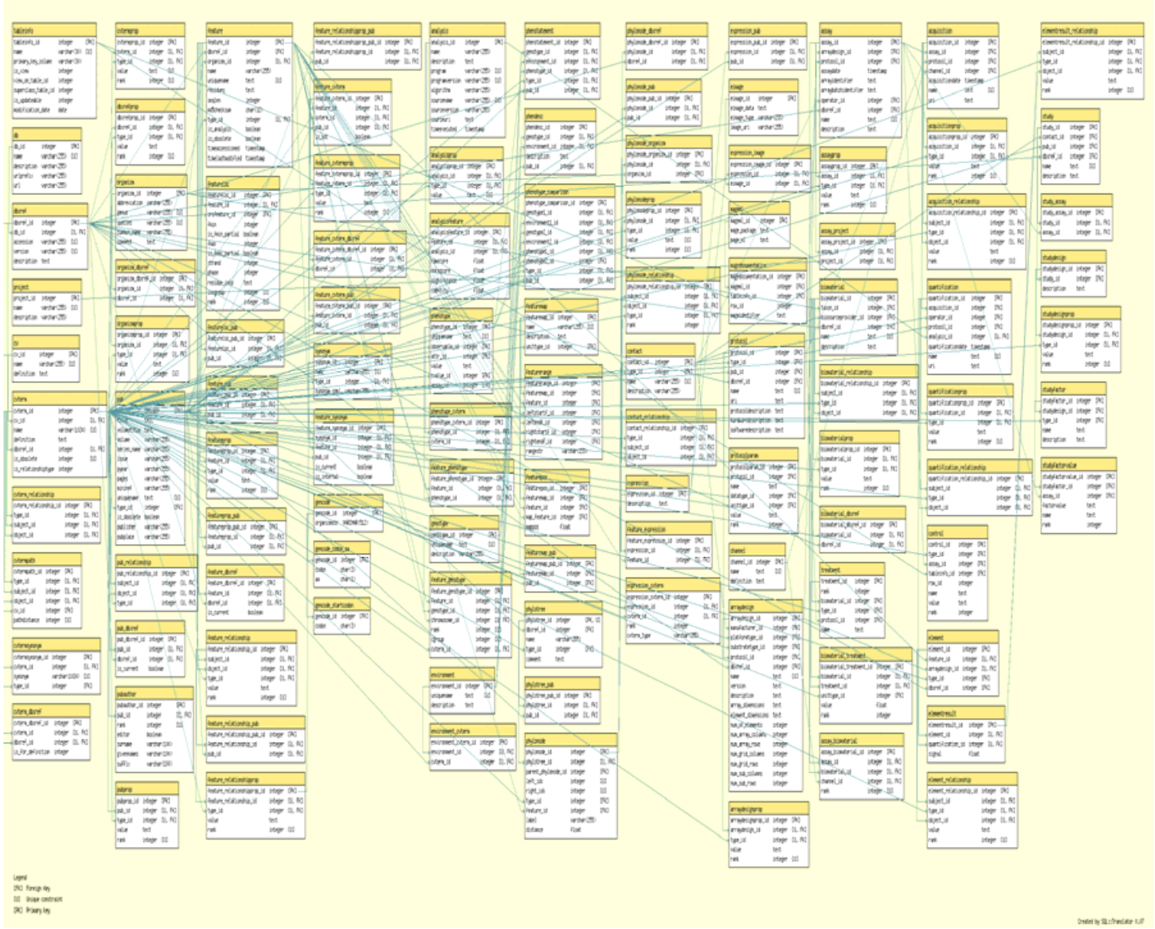


Figure 2 - CHADO Schema.

collaboration between several groups to develop a set of open-source software for managing model organism data [27].   Chado is able to represent many of the general classes of data frequently encountered in modern biology such as sequence, sequence comparisons, phenotypes, genotypes, ontologies, publications and phylogeny. With an aim towards widespread applicability, Chado has been designed to handle complex representations of biological knowledge and is a very sophisticated relational schema.

Chado is designed as a modular schema. Each module of the Chado database contains data independently stored within the database. Existing modules within the Chado database are:

- Audit – for database audits
- Companalysis – for data and computational analysis
- Contact – for people, groups and organizations
- Controlled Vocabulary (cv) – for controlled vocabularies and ontologies
- Expression – for summaries of RNA and protein expression
- General – for identifiers
- Genetic – for genetic data and genotypes
- Library – for descriptions of molecular libraries
- Mage – microarray data
- Map – maps without sequence
- Organism - taxonomic data
- Phenotype – phenotypic data
- Phylogeny – phylogenetic trees and organisms
- Publication (pub) – literature references and original manuscripts

- Sequence – sequences and sequence features
- Stock – specimens and biological collections

The updated Bovine QTL Viewer database has integrated all data from the first generation database schema into the Chado relational schema. A novel method of implementation of QTL into the Chado schema has been employed. Furthermore, the use of the Chado schema has allowed for additional bovine genome information to be stored synergistically with previous bovine QTL data. Btau4 has been incorporated into the database and covers ~95% (~7.1x depth) of the genome. In addition to genomic sequence data, the database stores marker data, SNPs and haplotype blocks. Storing multiple data types within the single database allows for multiple sophisticated queries by cattle researchers.

2. Materials and Methods

The current data entries in the second generation Bovine QTL Viewer database are summarized in table 1. The data has been collected from public databases/datasets and the cattle research community. Chromosome assemblies were obtained from the Human Genome Sequencing Center (HGSC) at Baylor College of Medicine. The genome has been sequenced to an approximately 7 fold depth using a hybrid tiling-path/whole-genome shotgun approach [28]. QTL data was migrated from our first generation Bovine QTL Viewer database [14]. The first generation database is continually updated with newly published QTL and most recently updated by Dr. Dave Adelson and this author. SNP data was provided by Baylor HGSC as part of a 2007 freeze release with the assembly. The SNP dataset was parsed and formatted into a gff3 [29] data file using a Perl script for loading into the database. Baylor HGSC and NCBI were the source of many of the STSs in the second generation database. However, some markers needed to anchor QTL were located using a BLAST-based approach. A Perl script was written to locate markers necessary for anchoring QTL, and assemble sequence data and/or primer information. If the marker could not be placed onto the assembly via BLAST, the primer information associated with the marker allowed for e-PCR searching and anchoring (full method detailed in chapter IV) [30, 31]. Literature/references describing the QTL were migrated from the first generation Bovine QTL Viewer database.

Table 1 - Data summary of second generation database.
Haplotype block number subject to change with coming additional data.

Data type	Count
Quantitative Trait Loci	843
Single Nucleotide Polymorphism	1634676
Sequence Tagged Sites	17342
Haplotype Blocks	943
References	81

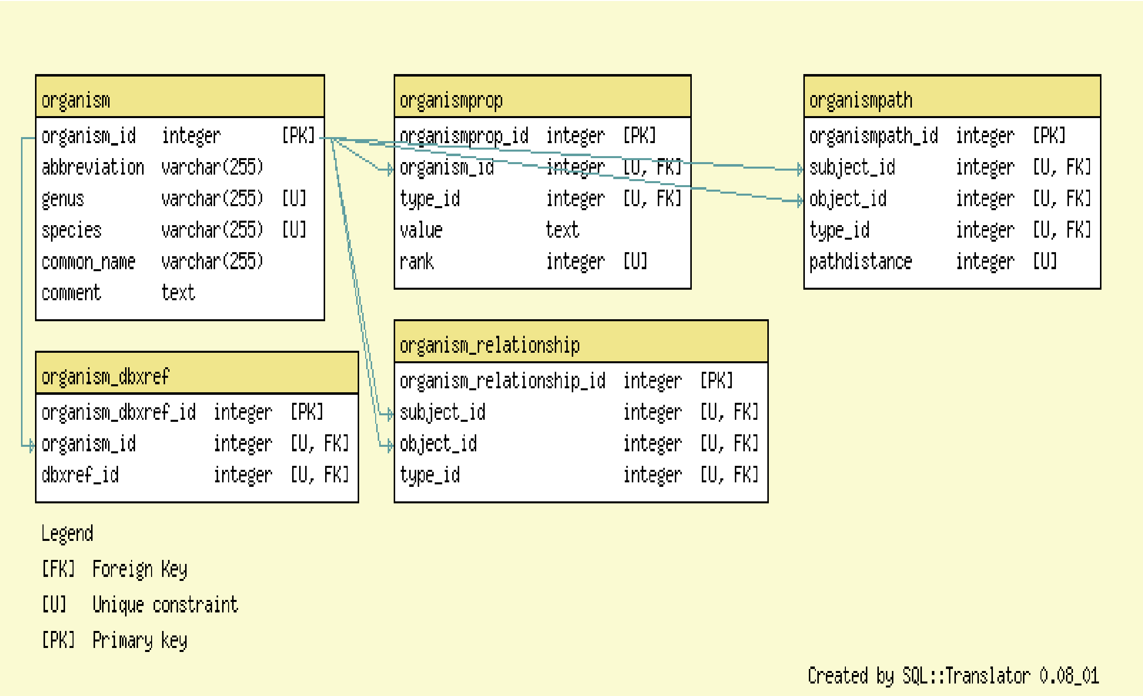
A Perl script was used to import references from the first generation database and formatted into an SQL load file. The SQL load file was loaded into the PostgreSQL Chado database through the PSQL dump utility. QTL, SNP and marker data were loaded into the second generation Bovine QTL Viewer database via the gmod_bulk_load_gff.pl bioPerl script. This Perl script has a tighter control of the syntax and requires the use of a controlled vocabulary (SO) [32].

The first generation database management system used was MySQL for ease of use and functionality reasons. However, the presented second generation database management system is based on the PostgreSQL, which provides increased security and more robust capabilities [33]. Development work was performed on the 'turing.sapac.edu' server. The production server for the database is the 'genomes.sapac.edu' server. Both servers are halves of an Altix XE310 machine with 8 cores and 16GB each.

3. Results

The Chado schema is capable of handling far more data types than are being stored in the first generation Bovine QTL Viewer database. However, this is of benefit because of the capability for future expansion and interoperability. In the migration of data from the old database to the new database a number of existing modules in Chado have been implemented.

The most basic module is the 'organism' module, which stores all taxonomic information for the cow (Figure 3). The central table of the organism module is the organism table. The relevant cattle organism taxonomic data was loaded into this table. A unique id was loaded into the attribute organism_id, this integer was an incremental increase from the default. Other attributes of the table were implemented as normal for the *Bos taurus* genus/species.



organism		
organism_id	integer	[PK]
abbreviation	varchar(255)	
genus	varchar(255)	[U]
species	varchar(255)	[U]
common_name	varchar(255)	
comment	text	

organismprop		
organismprop_id	integer	[PK]
organism_id	integer	[U, FK]
type_id	integer	[U, FK]
value	text	
rank	integer	[U]

organismpath		
organismpath_id	integer	[PK]
subject_id	integer	[U, FK]
object_id	integer	[U, FK]
type_id	integer	[U, FK]
pathdistance	integer	[U]

organism_dbxref		
organism_dbxref_id	integer	[PK]
organism_id	integer	[U, FK]
dbxref_id	integer	[U, FK]

organism_relationship		
organism_relationship_id	integer	[PK]
subject_id	integer	[U, FK]
object_id	integer	[U, FK]
type_id	integer	[U, FK]

Legend

[FK]	Foreign Key
[U]	Unique constraint
[PK]	Primary key

Created by SQL::Translator 0.08_01

Figure 3- Organism module.

The most substantial and often used module of the Chado database is the ‘sequence’ module [34] (Figure 4). Each itemized data type (i.e. chromosome, SNP, QTL) is stored as a ‘feature’ within the feature table of the sequence module. Based on the theory that a feature is a part of a sequence and a piece of a sequence is a sequence, the Chado schema does not distinguish between a sequence and a sequence feature. Both are represented as a feature in the feature table. Each ‘feature’ of the database is associated with a feature type which has specific nomenclature derived from the Sequence Ontology (SO) [32]. The type of the feature must be termed within SO. All feature types within the Bovine QTL Viewer database are SO terms. The SO provides a structured controlled vocabulary for the parts of a genomic annotation. Each feature within the Bovine QTL Viewer database has a unique sequence ontology term and identification (Table 2). Each feature is positioned relative to other features in the table.

Table 2 - Sequence ontology in the database.
Presents sample 'features' in the Bovine QTL Viewer database and sequence ontology terms/ID.

Feature	SO ID
Chromosome	SL:0000340
SNP	SL: 0000072
QTL	SL:0000771

For example, using the featureloc table rows, each SNP feature can be positioned relative to each chromosome feature. Features can also be located on more than one feature.

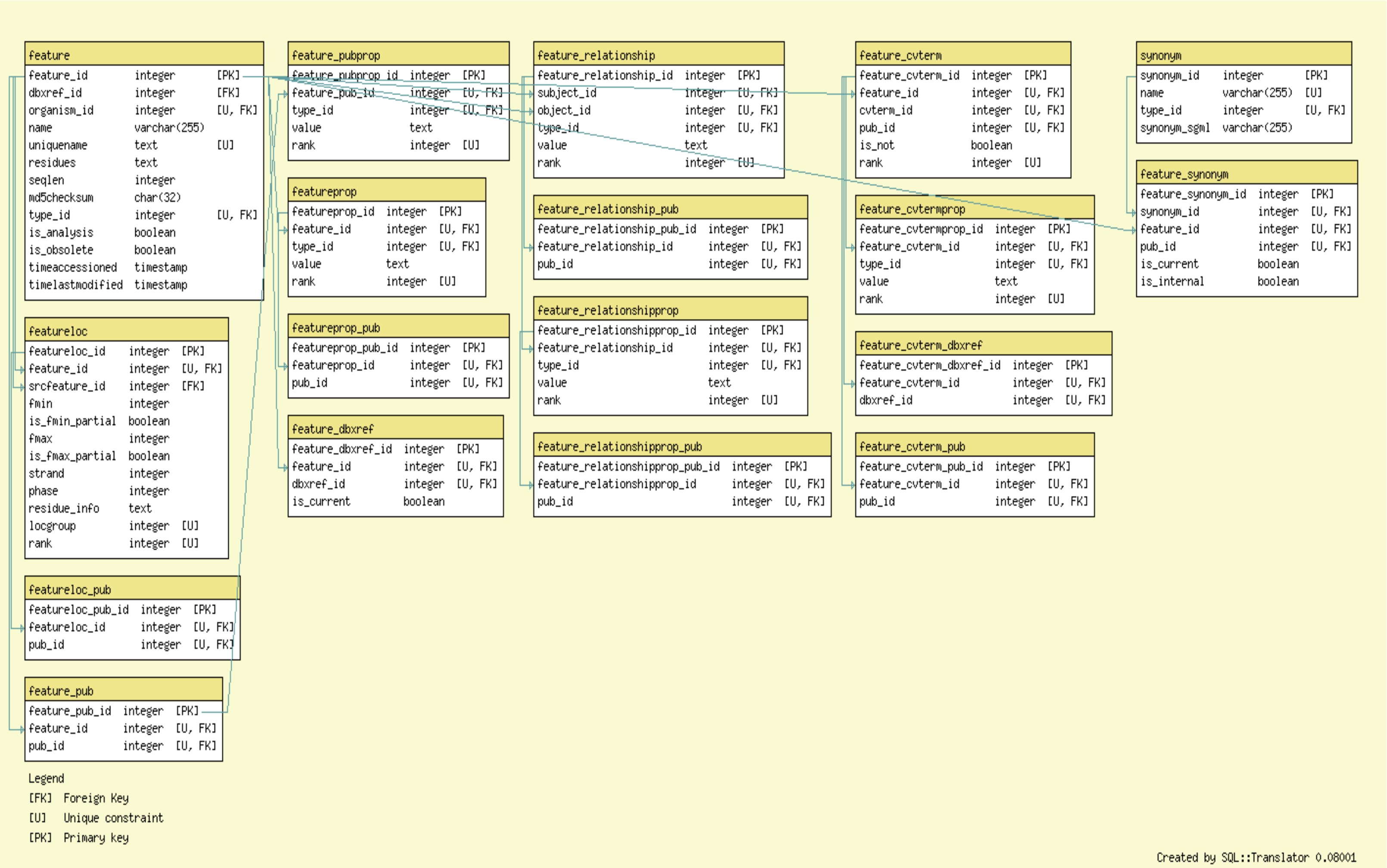

feature
feature_id integer [PK]
dbxref_id integer [FK]
organism_id integer [U, FK]
name varchar(255)
uniquename text [U]
residues text
seqlen integer
md5checksum char(32)
type_id integer [U, FK]
is_analysis boolean
is_obsolete boolean
timeaccessioned timestamp
timelastmodified timestamp
featureloc
featureloc_id integer [PK]
feature_id integer [U, FK]
srcfeature_id integer [FK]
fmin integer
is_fmin_partial boolean
fmax integer
is_fmax_partial boolean
strand integer
phase integer
residue_info text
locgroup integer [U]
rank integer [U]
featureloc_pub
featureloc_pub_id integer [PK]
featureloc_id integer [U, FK]
pub_id integer [U, FK]
feature_pub
feature_pub_id integer [PK]
feature_id integer [U, FK]
pub_id integer [U, FK]
Legend
[FK] Foreign Key
[U] Unique constraint
[PK] Primary key
feature_pubprop
feature_pubprop_id integer [PK]
feature_pub_id integer [U, FK]
type_id integer [U, FK]
value text
rank integer [U]
featureprop
featureprop_id integer [PK]
feature_id integer [U, FK]
type_id integer [U, FK]
value text
rank integer [U]
featureprop_pub
featureprop_pub_id integer [PK]
featureprop_id integer [U, FK]
pub_id integer [U, FK]
feature_dbxref
feature_dbxref_id integer [PK]
feature_id integer [U, FK]
dbxref_id integer [U, FK]
is_current boolean
feature_relationship
feature_relationship_id integer [PK]
subject_id integer [U, FK]
object_id integer [U, FK]
type_id integer [U, FK]
value text
rank integer [U]
feature_relationship_pub
feature_relationship_pub_id integer [PK]
feature_relationship_id integer [U, FK]
pub_id integer [U, FK]
feature_relationshipprop
feature_relationshipprop_id integer [PK]
feature_relationship_id integer [U, FK]
type_id integer [U, FK]
value text
rank integer [U]
feature_relationshipprop_pub
feature_relationshipprop_pub_id integer [PK]
feature_relationshipprop_id integer [U, FK]
pub_id integer [U, FK]
feature_cvterm
feature_cvterm_id integer [PK]
feature_id integer [U, FK]
cvterm_id integer [U, FK]
pub_id integer [U, FK]
is_not boolean
rank integer [U]
feature_cvtermprop
feature_cvtermprop_id integer [PK]
feature_cvterm_id integer [U, FK]
type_id integer [U, FK]
value text
rank integer [U]
feature_cvterm_dbxref
feature_cvterm_dbxref_id integer [PK]
feature_cvterm_id integer [U, FK]
dbxref_id integer [U, FK]
feature_cvterm_pub
feature_cvterm_pub_id integer [PK]
feature_cvterm_id integer [U, FK]
pub_id integer [U, FK]
synonym
synonym_id integer [PK]
name varchar(255) [U]
type_id integer [U, FK]
synonym_sgml varchar(255)
feature_synonym
feature_synonym_id integer [PK]
synonym_id integer [U, FK]
feature_id integer [U, FK]
pub_id integer [U, FK]
is_current boolean
is_internal boolean
Created by SQL::Translator 0.08001

Figure 4 - Sequence module.

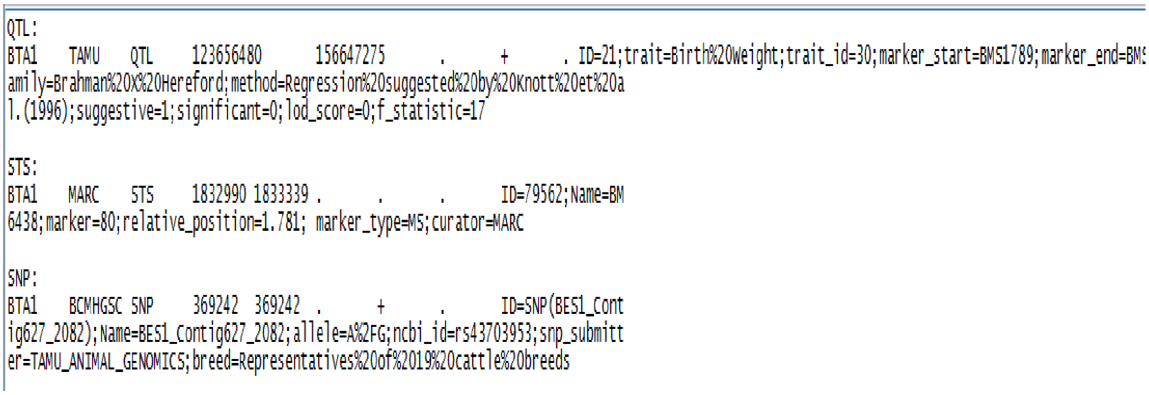
Bovine chromosomes, QTL, SNPs, and STSs are features stored in the feature table. Each has a unique id stored as the feature_id attribute. QTL id is an independent unique value created in the original Bovine QTL Viewer database. Chromosome features are identified by the string ‘BTA’ followed by the chromosome number. SNPs are identified by the Baylor Human Genome Sequencing Center unique id. STSs are identified by the marker name.

The name attribute of the feature table is nullable since not all features need names [27, 34]. The primary preferred short name is stored within this attribute. Because QTL are identified by trait, QTL have no current name associated with them. SNP name is the same as the unique id identical to the BHGSC identification tag. STSs name is identical to its unique id and is the name of the marker. Because the name attribute is nullable, names can be associated with multiple feature_ids, however this is avoided in the Bovine QTL Viewer database.

The feature_id value serves as a gateway term that relates tables within the sequence module. Each feature’s id allows connection to the featureloc table which identifies the location of the feature. Within featureloc, an individual id is assigned to every row within the table, therefore every featureloc_id is associated with a feature_id. Hence, feature_id’s can be located to various positions as stated prior. The srcfeature_id is the term that identifies the feature (by feature_id) from which the location is positioned. Attributes fmin and fmax store the integer positions of the feature within the row. QTL, SNPs and STSs are positioned onto the srcfeature_id through their base pair positions stored by the fmin and fmax attributes. The DNA strand from which the feature is located is stored within the strand attribute. The rank is associated with the number of srcfeatures to which the feature corresponds. A rank of zero is generally given when there is only one srcfeature of the feature. QTL, SNPs, and STSs are all given a value of zero for the srcfeature rank.

Feature_id also links features to their properties. Each feature may have a number of tags, or attributes, which may be unique to the Chado schema. These unique tags are inserted in the featureprop table and linked through the feature_id attribute to individual features. Each row within the featureprop table has a unique id, the featureprop_id to uniquely identify a feature_id relating to its feature property. QTL unique tags are stored within the featureprop table.

QTL have a number of attributes that are unique to the Chado schema and therefore have been stored within the featureprop table. Each QTL has an associated trait for which it is contributing. The previous database stored the trait as an id within the qtl_info table. The trait_id was related to an additional table that stored the trait name associated with the unique ids. In the new Bovine QTL Viewer database, a tag "trait" was created to handle this unique attribute of QTL and has been stored in the featureprop table (Figure 5). Spaces have been replaced with '%20' which allows spaces to be coded in html. This will allow the trait to be clearly seen with the javascript popup boxes in the Gbrowse view of the viewer. A file with all of the data from the first generation database was created and loaded into the second generation database via the GMOD Perl script.



```
QTL:
BTA1    TAMU    QTL     123656480       156647275       .       +       . ID=21;trait=Birth%20Weight;trait_id=30;marker_start=BMS1789;marker_end=BMS
amily=Brahman%20X%20Hereford;method=Regression%20suggested%20by%20Knott%20et%20a
l.(1996);suggestive=1;significant=0;lod_score=0;f_statistic=17

STS:
BTA1    MARC    STS     1832990 1833339 .       .       .       ID=79562;Name=BM
6438;marker=80;relative_position=1.781; marker_type=MS;curator=MARC

SNP:
BTA1    BCMHGSC SNP     369242  369242  .       +       .       ID=SNP(BES1_Cont
ig627_2082);Name=BES1_Contig627_2082;allele=A%2FG;ncbi_id=rs43703953;snp_submitt
er=TAMU_ANIMAL_GENOMICS;breed=Representatives%20of%2019%20cattle%20breeds
```

Figure 5- Gff3 data format.

Flanking marker data is maintained in the new database through the featureprop table. Start and stop markers in the previous Bovine QTL Viewer Database are of the varchar datatype limited to 20 characters and are stored within the qtl_info table. The markers are termed marker_start and marker_end. This data is maintained in the new database using a tag that goes by the same attribute name as the previous database, marker_start and marker_end. Also stored in the previous database' qtl_info table was breed information from the QTL. A breed tag has been created to maintain this information for every QTL. Additional data migrated from the qtl_info table of the previous database includes the family information, method by which the QTL was derived, logarithm of odds score and any f-statistic available. The significance or suggestive level is also maintained as a Boolean 1 (true) or 0 (false). Unique tags are associated with data types other than QTL as well. The new Bovine QTL Viewer database has incorporated new data types that were not in the previous database including: SNPs and STSs. Special tags for these data types have also been created.

Sequence Tagged Sites have attributes that required additional tags to fully cover necessary data. All the markers used to anchor the QTL within the bovine QTL Viewer database are curated by USDA-MARC. Therefore, a tag was created to store and maintain the integer id given to each marker by USDA. This value is stored under the tag “marker.” The previous database stored the linkage cM position in the marc_markers_info table. For the new database, a tag has been created for the linkage position and is stored within the featureprop table. The previous database maintained the type of marker presented by a two character nomenclature and was also stored in the marc_markers_info table. The nomenclature follows that created by USDA (i.e. MS = Microsattellite). STS from many different sources have been made available and henceforth, have been incorporated into the new Bovine QTL Viewer database. To identify the source of the STS, a curator tag was created. The curator of each STS is stored in the featureprop table of the database.

SNPs specific to the bovine genome further require additional tags to facilitate data flow. As SNPs can have multiple alleles, an allele tag has been created to store the possible alleles for each SNP. For future web linkage to NCBI, a tag has been created (ncbi_id) to maintain the id. The submitter of the SNP is stored in the featureprop table. In addition, a tag was created to store and maintain the breed of the SNP and has been stored in the featureprop table.

The sequence module is able to store the source of each feature. The source, or submitter, of each featured is stored as a dbxref. The relationship between the feature and the dbxref is maintained through the feature_dbxref table. The type of feature is maintained through the cvterm table. The cvterm_id is the sequence ontology id for the feature's type. The table feature_cvterm is within the sequence module and serves as a relational hub connecting feature and cvterm relational database tables.

Other modules in the Chado schema branch out from the fundamental sequence module by a consistent table nomenclature. Feature_X, where X=name of the fundamental table of the connecting module, is how the connection is made linking data within independent modules. The sequence module creates a connection to the genetics module via the feature_genotype table. The genetics module is of importance because this module stores and maintains bovine haplotype block data (Figure 6). Haplotypes

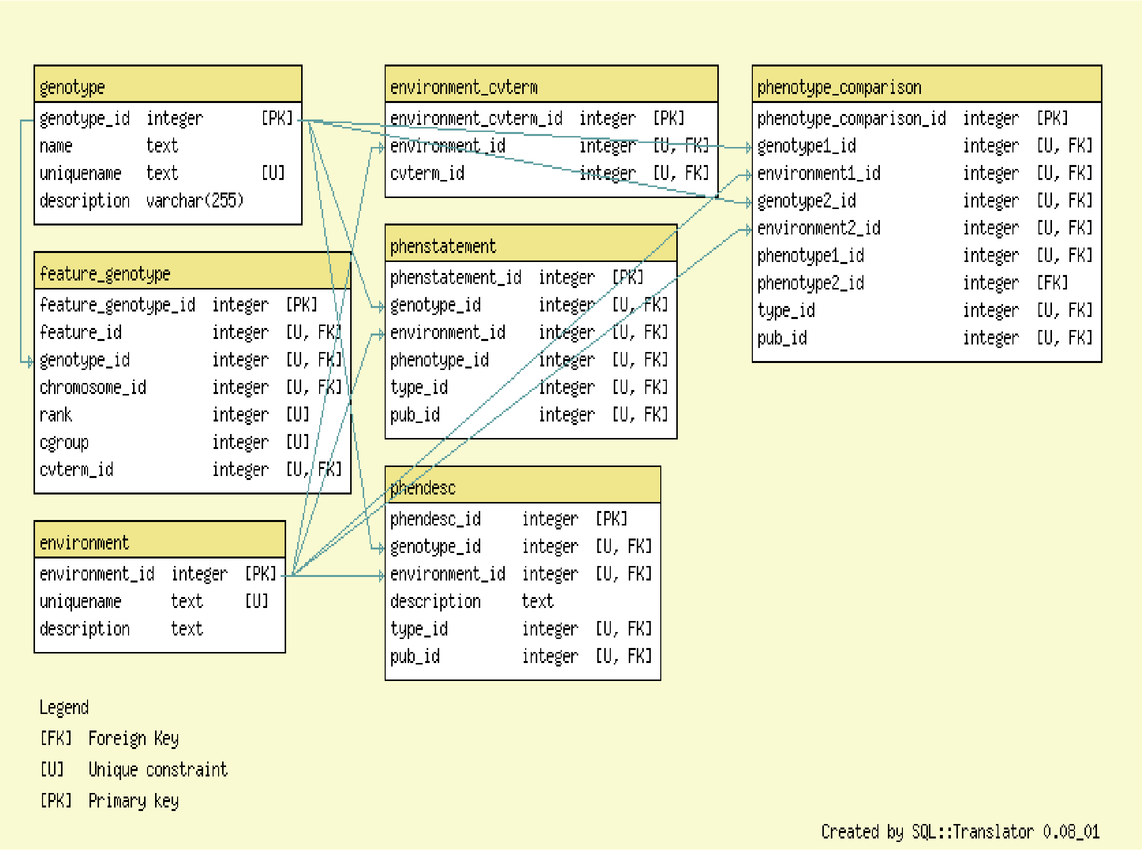

genotype
genotype_id integer [PK]
name text
uniquename text [U]
description varchar(255)
environment_cvterm
environment_cvterm_id integer [PK]
environment_id integer [U, FK]
cvterm_id integer [U, FK]
phenotype_comparison
phenotype_comparison_id integer [PK]
genotype1_id integer [U, FK]
environment1_id integer [U, FK]
genotype2_id integer [U, FK]
environment2_id integer [U, FK]
phenotype1_id integer [U, FK]
phenotype2_id integer [FK]
type_id integer [U, FK]
pub_id integer [U, FK]
feature_genotype
feature_genotype_id integer [PK]
feature_id integer [U, FK]
genotype_id integer [U, FK]
chromosome_id integer [U, FK]
rank integer [U]
cgroup integer [U]
cvterm_id integer [U, FK]
phenstatement
phenstatement_id integer [PK]
genotype_id integer [U, FK]
environment_id integer [U, FK]
phenotype_id integer [U, FK]
type_id integer [U, FK]
pub_id integer [U, FK]
environment
environment_id integer [PK]
uniquename text [U]
description text
phendesc
phendesc_id integer [PK]
genotype_id integer [U, FK]
environment_id integer [U, FK]
description text
type_id integer [U, FK]
pub_id integer [U, FK]
Legend
[FK] Foreign Key
[U] Unique constraint
[PK] Primary key
Created by SQL::Translator 0.08_01

Figure 6 - Genetics module.

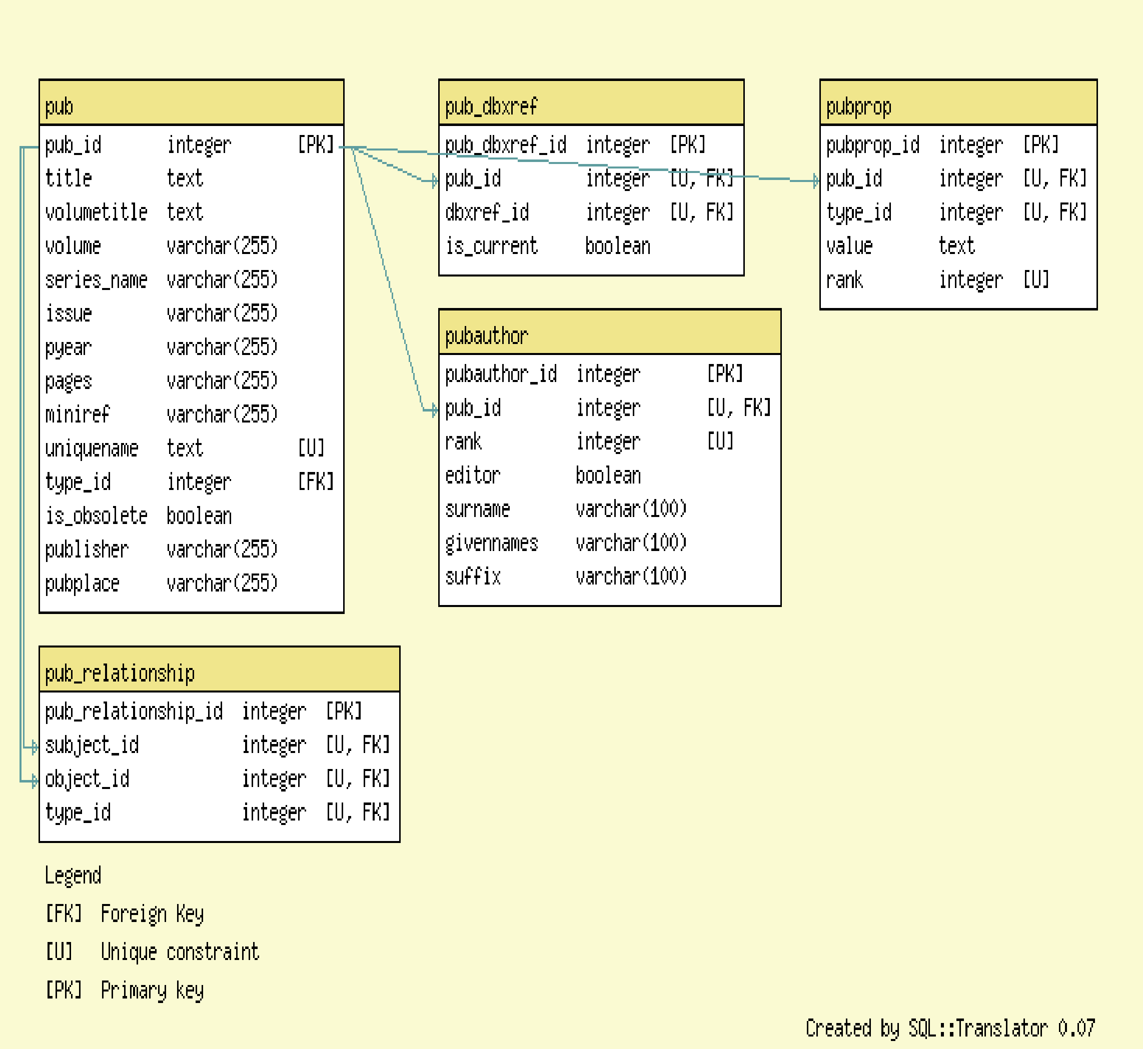
are sets of SNPs, on a single chromatid, that are observed to be in phase [35]. Groups of SNPs and associated data (i.e. breed) must be stored. The genetics module allows for such storage.

The feature_genotype table links SNP features of the second generation Bovine QTL Viewer database to the haplotype blocks they are associated with. The feature_genotype_id is an attribute that maintains a relationship between the SNP feature_id and the haplotype block genotype_id. This integer identification is increased incrementally as SNP to haplotype block connections are added. The SNP feature_id and haplotype_block genotype_id are stored within each row as well. A rank of SNPs is also required for each entry. Once the feature_genotype table was created in the database a connection between the SNP and haplotype block was created. A description of the haplotype block was then produced and stored within the genotype table of the module.

Each haplotype block is assigned a genotype identification within the genotype table of the database. The genotype_id is an integer whose value increments as more haplotype blocks are added into the table. Genotype_id is the primary key of the genotype table and the genotype table is fundamental to the module. Within the genotype table, a haplotype name, uniquename (human readable format), and description attribute are available for each block. The haplotype name follows the format of the bovine HapMap project, also suppliers of the Bovine QTL Viewer haplotype data (manuscript submitted).

The sequence module also creates a connection to publications via the feature_pub table. The feature_pub table works to unite each feature with an associated publication or manuscript. A unique id is assigned to this relationship through the feature_pub_id attribute. The feature_pub table also serves as a relational hub connecting the feature table within the sequence module to the pub table which is a main component of the pub module.

The pub module is another module that is utilized by the new Bovine QTL Viewer database and stores data that has been migrated from the old database. The pub module serves to store and maintain literary information that has spawned feature and feature attributes (Figure 7). Many features have been presented in manuscripts and those manuscripts may have other relevant information that may be of interest to researchers.



pub		
pub_id	integer	[PK]
title	text	
volumetitle	text	
volume	varchar(255)	
series_name	varchar(255)	
issue	varchar(255)	
pyear	varchar(255)	
pages	varchar(255)	
miniref	varchar(255)	
uniquename	text	[U]
type_id	integer	[FK]
is_obsolete	boolean	
publisher	varchar(255)	
pubplace	varchar(255)	

pub_dbxref		
pub_dbxref_id	integer	[PK]
pub_id	integer	[U, FK]
dbxref_id	integer	[U, FK]
is_current	boolean	

pubprop		
pubprop_id	integer	[PK]
pub_id	integer	[U, FK]
type_id	integer	[U, FK]
value	text	
rank	integer	[U]

pubauthor		
pubauthor_id	integer	[PK]
pub_id	integer	[U, FK]
rank	integer	[U]
editor	boolean	
surname	varchar(100)	
givennames	varchar(100)	
suffix	varchar(100)	

pub_relationship		
pub_relationship_id	integer	[PK]
subject_id	integer	[U, FK]
object_id	integer	[U, FK]
type_id	integer	[U, FK]

Legend

[FK]	Foreign Key
[U]	Unique constraint
[PK]	Primary key

Figure 7 - Publication module.

It is therefore necessary to store and maintain the information regarding each reference. The previous Bovine QTL Viewer database maintained literature information through the references table. The qtlreferences_info table maintains attributes for the reference identification (Integer), title of the manuscript presenting the QTL (varchar(255)), the citation (text), URL where the manuscript can be printed (text), and the year of presentation (Integer). Individual QTL are linked to the reference through the qtl_references table. The qtl_references table serves as a relational table that associates QTL with the manuscript in which it was described. This relational table allows for one QTL to be associated to multiple manuscripts in the event that more than one manuscript describes the same QTL.

Each manuscript may have many authors associated with its production and therefore listed as contributors to the work. An individual author may be involved with many publications. This many-to-many relationship for QTL to authors was handled in the first generation database by an additional table that stored author information, references_authors. This table had attributes for the reference identification (Integer) and the author name (varchar(50)).

In the second generation Bovine QTL Viewer database, a new approach is taken through the pub module to handle literary references. The pub table of the module is connected to the feature table through the feature_pub table. The pub table is the main table that stores the essential data contents of a contributing manuscript. The pub_id serves as a unique identifier for each publication. Because the pub_id is an integer, and the previous database stored references as integers, the new database was capable, and therefore configured to maintain the old databases' unique identifiers for each publication. The text datatype was the same in the first and second generation databases and therefore was migrated as is. New data entries that can now be implemented in the pub table includes: volume, volume title, series name, issue, pages and publisher. This new manuscript pertinent data was included for each reference, as available. Although the year of the publication was an integer in the previous generation database and a text in the new generation database, migration of an integer data type to a text data type is possible and has no side effects. The previous generation Bovine QTL Viewer database also stored marker reference information within the marc_references_info table. The new second generation database now handles this issue by storing references within the pub table in the same way that data is characterized for QTL. Control of the associated authors with a publication is handled through the publication id.

Authors in the publication module are handled in the pubauthor table. Each author is associated with a publication. The pubauthor_id attribute maintains a unique relationship between every publication and its author for that row of the table. The rank of that author, for that publication is maintained through the rank attribute. Whether or not the author of interest is the editor is stored as a Boolean value. Author first and last names are stored as given name and surname respectively. A suffix can also be stored if necessary.

Many of the tables in the Chado schema are not currently being implemented in the second generation Bovine QTL Viewer database. As more bovine annotation becomes available implementation of these tables will be seamless and contribute to the Bovine QTL Viewer as a whole.

4. Discussion

The goal of the second generation Bovine QTL Viewer database was to create a comprehensive, reliable and carefully curated database, to migrate data from the original first generation database and introduce new sequence annotation including SNPs and haplotype blocks. In order to achieve this goal, a scheme was carefully constructed which models the data from the first generation database and data from the most recent bovine genome annotation sets, into the widely applicable Chado schema. After carefully crafting a scheme, the approach was implemented and thus created a second generation database that has integrated publicly available data from disparate entities into a single database, with data relative to bovine QTL. The constructed database maintains flexibility to integrate new data types as they become available and is fully capable of adapting to changing research climates. New bovine sequence annotation, and the fourth and final bovine genome assembly require a tool that is capable of storing and querying this disparate data and being able to reach that data in a clear and cogent manner. The presented second generation Bovine QTL Viewer database has been developed as a response to this need and has been created in a fashion that is fully extensible for the future.

Benefits of the second generation database:

- Schema provides flexibility for future data additions.
- Modules allow for data independence within the database.
- New data type additions from the previous database including SNPs and haplotype blocks.

Future directions:

- New expression level data can be added to the database including genes, ESTs and gene transcripts.
- Additional tables can be added to maintain haplotype ancestry.

CHAPTER III

EVOLUTION OF A BOVINE QTL VIEWER

1. Introduction

The recent introduction of large amounts of high resolution genomic sequence data for the bovine genome has created a need for tools to help facilitate data access and retrieval. Research and studies have been conducted that demonstrate human's inclination towards learning through visual tools [2, 36]. A previous Bovine QTL Viewer was developed that displayed QTL with respect to their genomic position on a common linkage map [14]. This web accessible viewer has served as a facilitator for application of QTL and phenotypic data to agricultural enterprise research and as a result, has improved the efficiency of cattle research. However, high resolution bovine genomic maps have since been produced and increased bovine annotation is available that must be analyzed with respect to QTL data. With the increase in bovine annotation and sequence data, the bovine QTL viewer has evolved to meet the needs of the cattle research industry.

Bovine genomic maps have been created by the Baylor College of Medicine Human Genome Sequencing Center and have sequentially increased in resolution with amplified data and resources. To begin to accommodate the increase in genomic sequence data after the third assembly, the bovine QTL Viewer incorporated a Gbrowse view to display QTL with respect to the genome. Gbrowse, written with Perl CGI scripts, allowed for the creation of a map that displays the bovine genome sequence with respect to bovine QTL [37]. Further, supplementary tracks were created to incorporate additional bovine annotation including: SNPs, GC Content, STSs, and genes. This allowed for visual interpretation of sequential association between various levels of annotation that could provide insight into possible relationships that could be contributing to a phenotype of interest.

Including additional data types into the Gbrowse view allowed for additional tools to be added to the viewer specific to the new genomic data. A gene is a genomic region that produces protein [38]. Genes are of particular importance because of their effects on phenotype. In many instances, a researcher may be familiar with a gene by a derivative of its string (i.e. carboxy when looking for carboxylase). However, it may be difficult for a cattle researcher to find that gene of interest without otherwise knowing the exact term or id. In order to facilitate retrieval of gene data, a gene search tool was created to allow researchers to find genes first: by a string association of terms provided by the user and second: by its location within the genome. A string search is performed by the tool and a list of potential hits is presented to the user. The user then is able to select a gene of interest and is directed to that gene on the Gbrowse view illustrating the gene with respect to its location within the genome and other bovine annotation in the region. In addition to sequence annotation on the third assembly, Gbrowse also facilitated the incorporation of another bovine genomic map known as the composite map. The composite map was a map that incorporated data from various maps and used a newly created anchoring scheme to place data in its relative position. Gbrowse provided a new method of mapping QTL and other bovine annotation; however other aspects of the viewer evolved as well.

Updating the bovine QTL viewer's backend database has previously consisted of manually searching through animal science journals for articles that may potentially present novel QTL. This process has proven to be time consuming and very inefficient. In order to tackle this issue, a literature search tool was created. The literature search tool searches public databases for recent manuscripts that match specified terms consistently found in literature that present novel bovine QTL. This tool has saved time and increased efficiency for both the Bovine QTL Viewer administrator and the lab director.

The fourth and final bovine genome assembly has been released and additional bovine genomic annotation has been or is being produced. The bovine QTL Viewer must now again adapt to the new conversion in the landscape of data. This new wave of data now requires a fundamental change in the structure and view of the Bovine QTL Viewer. Here we present a new viewer that is capable of displaying multiple layers of bovine annotation at user request. The viewer is a modular, Java based, object oriented tool. This tool improves any future enhancements of the bovine QTL viewer because of the native schema and has become less dependent on the back end database and the outdated PHP programming language. I have constructed the new Bovine QTL Viewer by using the GOOF model [39]. This model allows developers to implement their data as objects. GOOF uses a model, view and control system. Each layer of model, view and control in GOOF uses three different Java frameworks, which maintain low-level API to communicate with the database and the web application server.

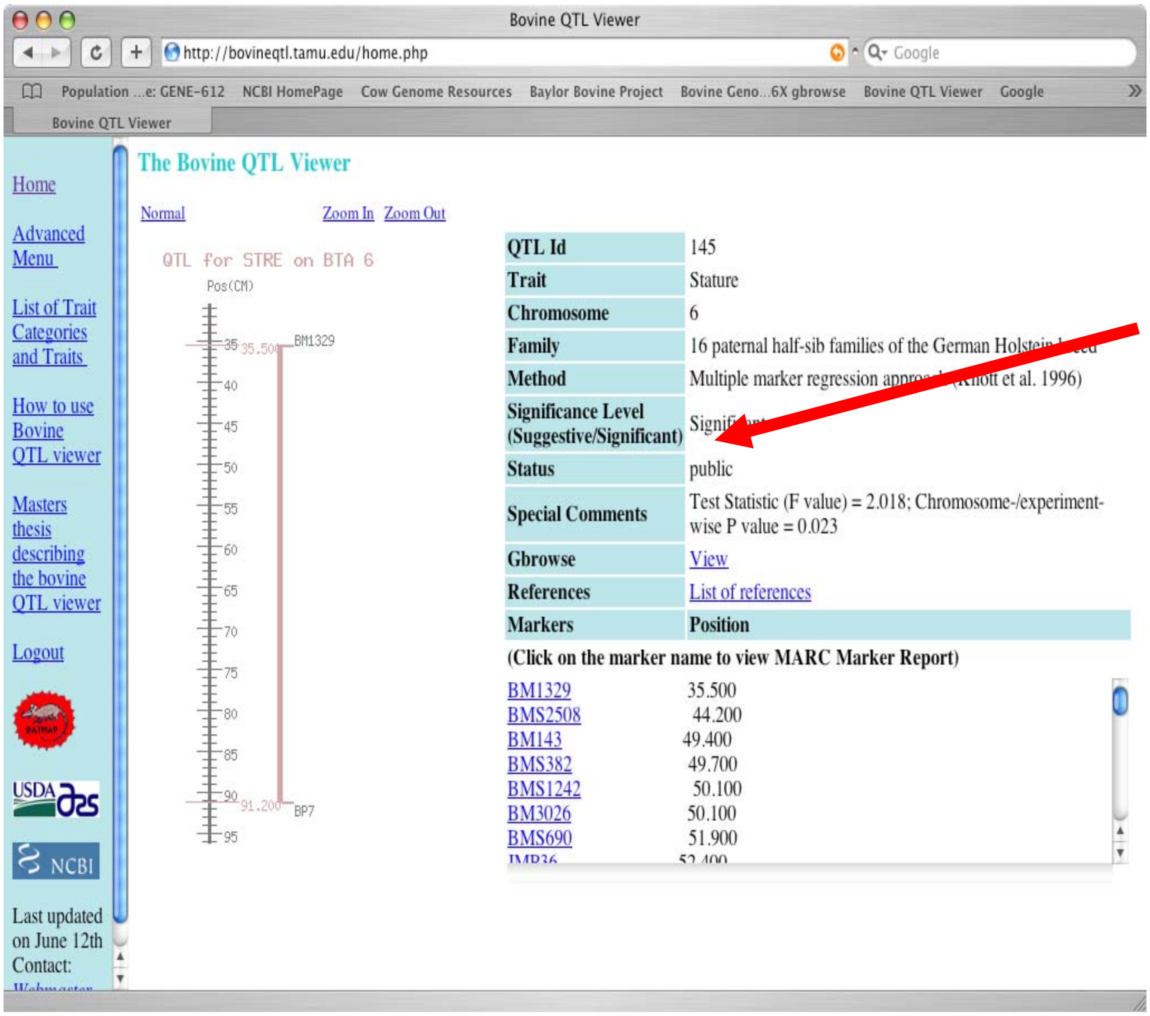
In addition to being able to build a viewer that collects and incorporates various data types re-structuring the viewer at this point in time allows for the quantitative implementation of user review to improve the viewer's design and work flow. I have used methods of pre- and post- survey analysis to evaluate the use and design of the tool. This provides for a quantitative method of demonstrating the use of the viewer. When creating community tools it is imperative to build your project based on the needs and preferences of your target community. I believe that the new Bovine QTL Viewer database will me meet the needs of the cattle research community and is extensible in order to meet future potential needs as well.

2. Materials and Methods

2.1 Introduction of Gbrowse

The first generation Bovine QTL Viewer provided a resource for cattle researchers to view QTL based on linkage maps. From the start page the user could query for QTL by two routes: by broad category or by trait. When querying by trait, the user can enter a chromosome (or all chromosomes), a lod score, an F statistic or significance level as desired. Based on the information entered, a list of QTL is presented through the viewer. The user can select QTL of interest and view a detailed page that describes the QTL in depth. If on the other hand the user searches by category, he/she will then select the chromosome(s) of interest. A figure is generated that displays all chromosomes in the genome and the location of all QTL, within the selected category, on a map. The user can select a chromosome of interest to get a further detailed view of the all the QTL represented on that chromosome. From the single chromosome page, the user can select a particular QTL and be directed to the aforementioned detailed QTL page.

The detailed QTL page provides an in depth description of the QTL presented. The user is presented the QTL's identification number (based on the unique id provided in the Bovine QTL Viewer database), trait, chromosome, F-statistic, lod score, family, and method of derivation. The significance level (whether the QTL is suggestive or significant) and status of availability (public/private) is presented as well. To accommodate the release of bovine sequence data by various groups including Baylor HGSC and NCBI, the detailed QTL page evolved to include a view of the QTL of interest with respect to the genome and other annotation (Figure 8).



The Bovine QTL Viewer

QTL Id	145
Trait	Stature
Chromosome	6
Family	16 paternal half-sib families of the German Holstein breed
Method	Multiple marker regression approach (Knott et al. 1996)
Significance Level (Suggestive/Significant)	Significant
Status	public
Special Comments	Test Statistic (F value) = 2.018; Chromosome-/experiment-wise P value = 0.023
Gbrowse	View
References	List of references
Markers	**Position**

(Click on the marker name to view MARC Marker Report)

BM1329	35.500
BMS2508	44.200
BM143	49.400
BMS382	49.700
BMS1242	50.100
BM3026	50.100
BMS690	51.900
IMP36	52.400

Figure 8 - Introduction of Gbrowse.
Expansion of the Bovine QTL viewer to include a Genome browser view of the QTL.

The Bovine QTL Viewer was expanded by incorporating a genome view of QTL using GBrowse [40]. GBrowse allowed our group to display QTL over bovine assembly sequence with respect to other sequence annotation including: genes, SNPs, GC content, and Sequence Tagged Sites (markers) (Figure 9). There has been

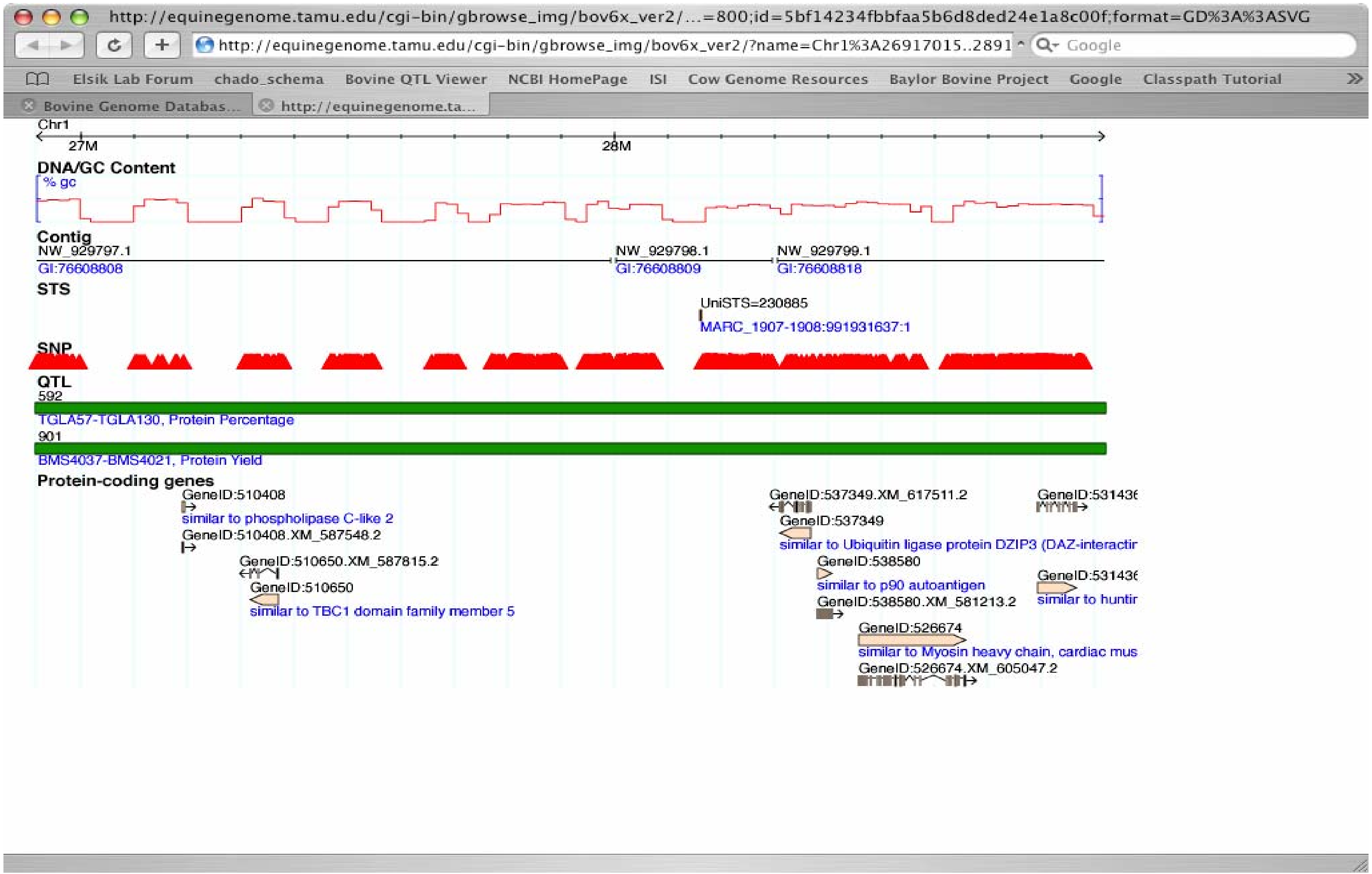

http://equinegenome.tamu.edu/cgi-bin/gbrowse_img/bov6x_ver2/...=800;id=5bf14234fbbfaa5b6d8ded24e1a8c00f;format=GD%3A%3ASVG
http://equinegenome.tamu.edu/cgi-bin/gbrowse_img/bov6x_ver2/?name=Chr1%3A26917015..2891
Google
Elsik Lab Forum
chado_schema
Bovine QTL Viewer
NCBI HomePage
ISI
Cow Genome Resources
Baylor Bovine Project
Google
Classpath Tutorial
Bovine Genome Databas...
http://equinegenome.ta...
Chr1
27M
28M
DNA/GC Content
% gc
Contig
NW_929797.1
GI:76608808
NW_929798.1
GI:76608809
NW_929799.1
GI:76608818
STS
UniSTS=230885
MARC_1907-1908:991931637:1
SNP
QTL
592
TGLA57-TGLA130, Protein Percentage
901
BMS4037-BMS4021, Protein Yield
Protein-coding genes
GeneID:510408
similar to phospholipase C-like 2
GeneID:510408.XM_587548.2
GeneID:510650.XM_587815.2
GeneID:510650
similar to TBC1 domain family member 5
GeneID:537349.XM_617511.2
GeneID:537349
similar to Ubiquitin ligase protein DZIP3 (DAZ-interactir
GeneID:538580
similar to p90 autoantigen
GeneID:538580.XM_581213.2
GeneID:526674
similar to Myosin heavy chain, cardiac mus
GeneID:526674.XM_605047.2
GeneID:53143¢
GeneID:53143¢
similar to huntir

Figure 9 - Gbrowse view.
Gbrowse view from first generation Bovine QTL Viewer

quantifiable success with this work based on direct user response.

A backend database (MySQL) of Gbrowse was created to store all data presented in the browser. The database was built in the default GMOD database configuration (Figure 10) [40]. The database contains the default tables: fattribute, fattribute_to_feature, fdna, fdata, fmeta, and fgroup. It stores DNA sequence data as a long blob in the fdna table and stores all other sequence annotation data under the fdata table, based on keys from the fattribute table. The first generation viewer is running the bovine genome third assembly as the anchor DNA sequence.

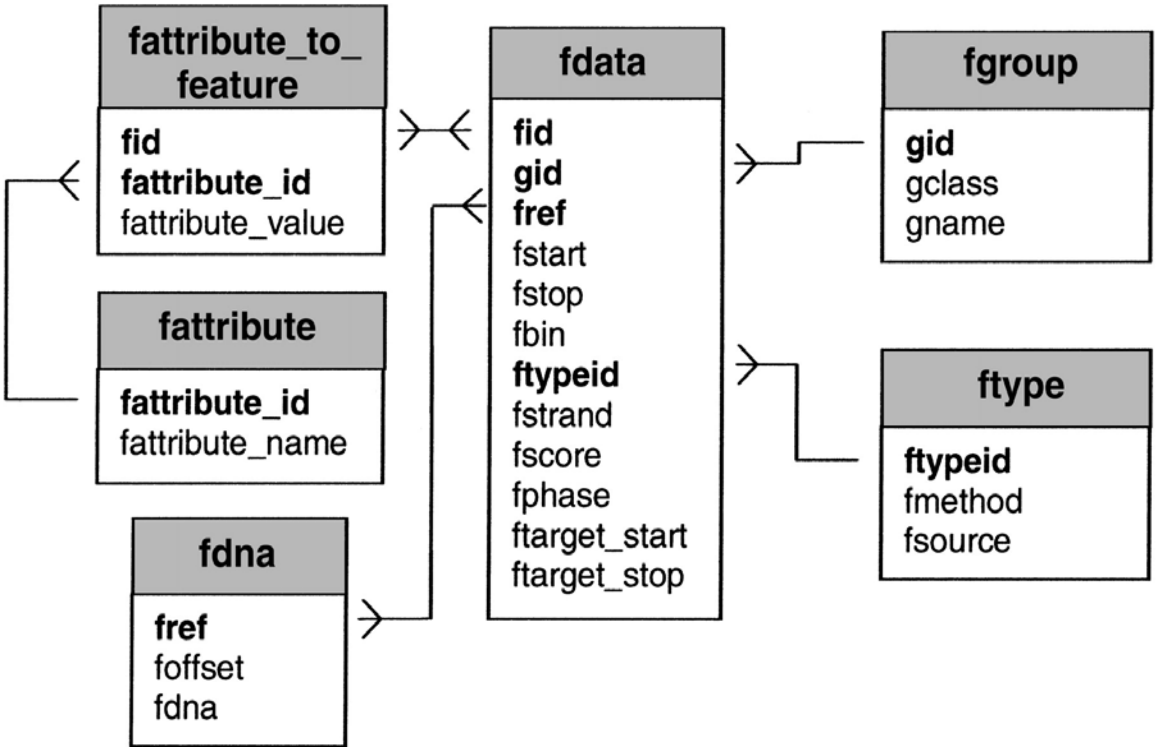


Figure 10 - Gbrowse database schema.
Backend database of the Gbrowse data for the first generation database.

The front web end of the genome browser facilitates user access to the data. Within Gbrowse the viewer sees the data within a “window.” The window can be scaled to the location within the chromosome depending on user preference. The Bovine QTL Viewer genome browser allows the user to scale the window up to a size of 150 Mb. The “window” can slide across the chromosome to view various regions of the chromosome in depth. Varying levels of annotation can be adjusted by turning on/off tracks. Tracks can be switched on/off by clicking checkboxes. The user can download data of interest within the window he/she is viewing by selecting from the “Reports & Analysis” drop down menu. A desired region can be viewed through the “window” by entering the location into the “Landmarks and Regions” text box. Further a feature of interest (i.e. a QTL, SNP, Chromosome, STS) can be viewed by entering its tag name (example: ‘STS’) followed by a colon followed by its name (example: BMS2790). The window will then be isolated over the region for that feature. Clicking on a particular QTL will re-route the user to the detailed QTL page on the viewer. This cyclical approach allows the user to drill deeper into the data in much same format as a reader identifying a reference of interest, reading that manuscript and finding another reference to find further information.

I have also provided links to the bovine composite map, which I have installed in Gbrowse [41]. The composite marker map is a comprehensive resource that supports investigations into relationships between genomic and phenotypic variation in cattle and aided in the assembly of the third version of the bovine genome [42]. This map provides increased marker detail, which allows more QTL to be viewed. The drawback is that it is not an exact sequence annotation, only estimation. The distances are arbitrary units, set up specifically for the map.

The establishment of a browser that allows the user to view QTL by its position along the genome provides many benefits. It allows the user to get a finer view of where the QTL is positioned in the genome. Secondly and most importantly, it allows for other bovine annotation to be placed and referenced with respect to QTL including SNPs, STSs and genes.

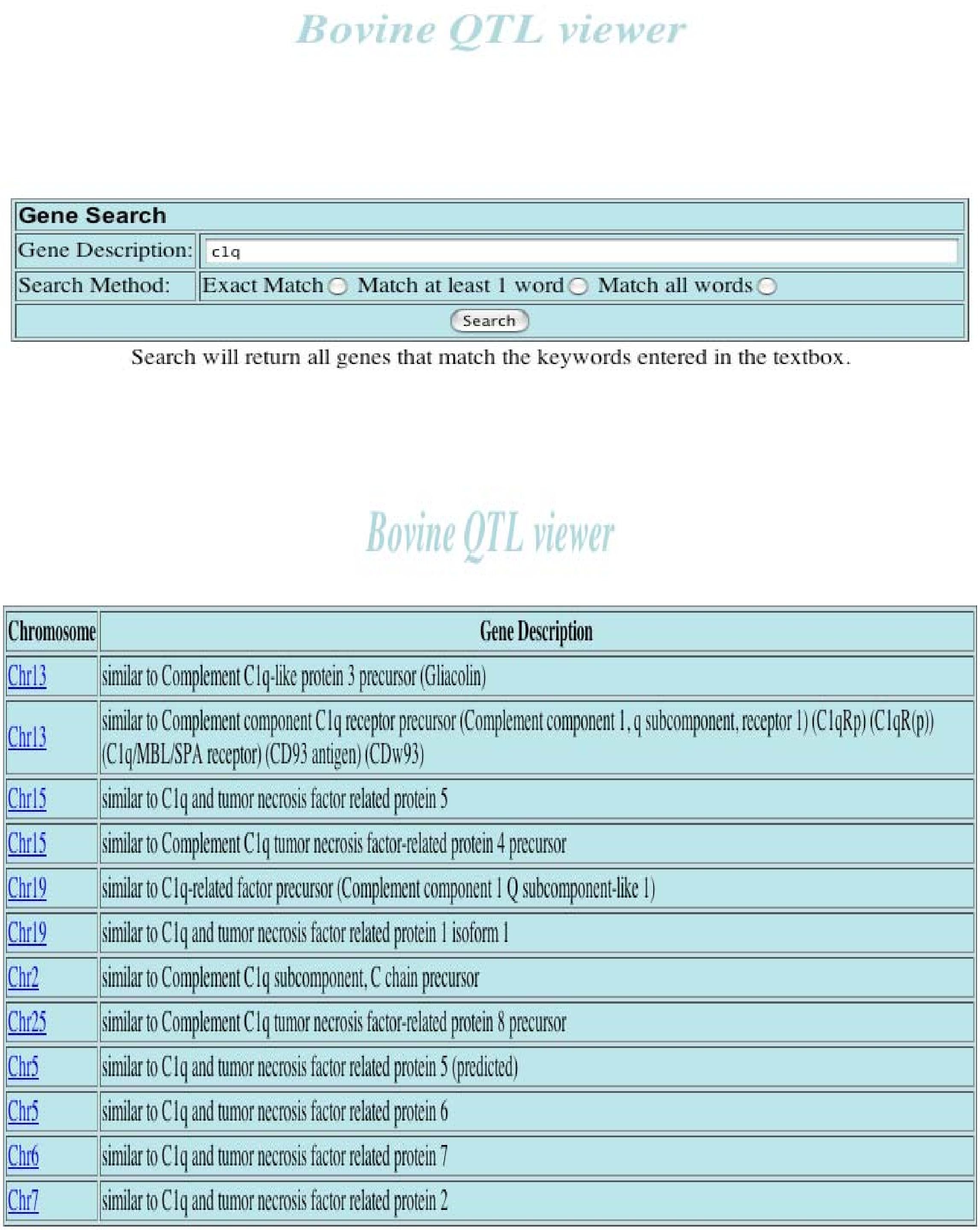
2.2 Development of a Gene Search Tool

In order to help facilitate user search for bovine annotation, a gene search tool was created and implemented. Many researchers may have an idea of their gene of interest but not know the exact term or gene id. A tool that could take the user's input and find string matches of the term can greatly increase efficiency and speed up the process for the user. With this idea in mind, a gene search tool was created that searches for matches of an input string based on its root. For example, if the user enters the term carboxylase, it will search genes definitions for the term 'carboxylase'. It excludes possible prefixes such as "de." Therefore, the search will not return false positives such as de-carboxylase. This tool is of great use because it provides insight into known functional sequence within QTL regions.

At the back end, a PHP page was written to receive the user's string through a text box. The user can search with the string based on three options: an exact match,

match at least 1 word, or match all words. This gives the user flexibility when entering multiple strings to search. When searching to match 1 word the script searches the Gbrowse database for terms that match through a query with OR statements. However, if searching for all words, it will query the database making use of AND statements.

The user can enter the gene search tool by a link on the options panel of the Bovine QTL Viewer (Figure 11). Upon entering the tool, the user is presented a blank text box with the three aforementioned options. The user will enter the term of interest and be presented with a list of genes that are matches to the inputted term. The list presents the chromosome and the description for each of the matching genes. The user can move further by selecting the hyperlinked chromosome attached to each gene in the table. Upon selection, the user is directed to the genome browser with the window covering the gene selected. At this point, the user is free to select tracks that can display multiple data types that may fall within this window including but not limited to: QTL, SNPs or STSs.



Bovine QTL viewer

Gene Search	
Gene Description:	c1q
Search Method:	Exact Match ○ Match at least 1 word ○ Match all words ○

Search will return all genes that match the keywords entered in the textbox.

Bovine QTL viewer

Chromosome	Gene Description
Chr13	similar to Complement C1q-like protein 3 precursor (Gliacolin)
Chr13	similar to Complement component C1q receptor precursor (Complement component 1, q subcomponent, receptor 1) (C1qRp) (C1qR(p)) (C1q/MBL/SPA receptor) (CD93 antigen) (CDw93)
Chr15	similar to C1q and tumor necrosis factor related protein 5
Chr15	similar to Complement C1q tumor necrosis factor-related protein 4 precursor
Chr19	similar to C1q-related factor precursor (Complement component 1 Q subcomponent-like 1)
Chr19	similar to C1q and tumor necrosis factor related protein 1 isoform 1
Chr2	similar to Complement C1q subcomponent, C chain precursor
Chr25	similar to Complement C1q tumor necrosis factor-related protein 8 precursor
Chr5	similar to C1q and tumor necrosis factor related protein 5 (predicted)
Chr5	similar to C1q and tumor necrosis factor related protein 6
Chr6	similar to C1q and tumor necrosis factor related protein 7
Chr7	similar to C1q and tumor necrosis factor related protein 2

Figure 11 - Gene search tool flow.

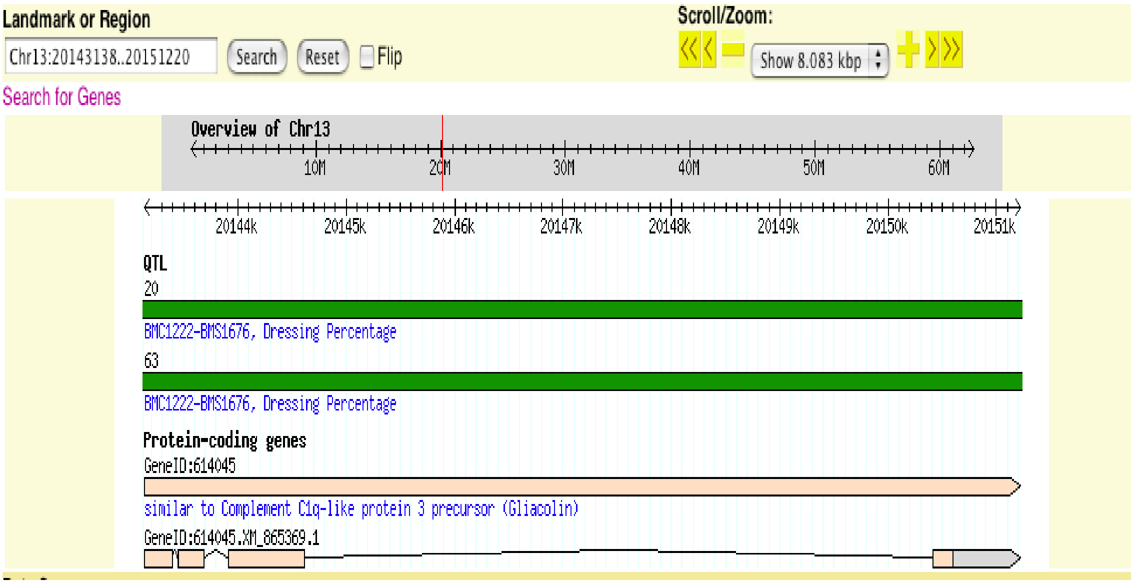


Figure 11 continued.

2.3 Literature Search Tool

Other search schemes have been implemented during the evolution of the Bovine QTL Viewer as well. Manual curation for literature and data specific to the Bovine QTL Viewer/database is a long and somewhat tedious process. Journals must be combed through for literature that may present novel QTL and once identified, those manuscripts must be analyzed and that data entered into the database. While it is not yet possible to use a computer to analyze the paper, it is possible to search and identify literature that may be presenting novel QTL. The literature search tool utilizes pubfetch [43], which searches the pubMed and agricola databases for literature that could be relevant to the Bovine QTL Viewer. The tool utilizes the pubfetch module provided by GMOD (Figure 12).

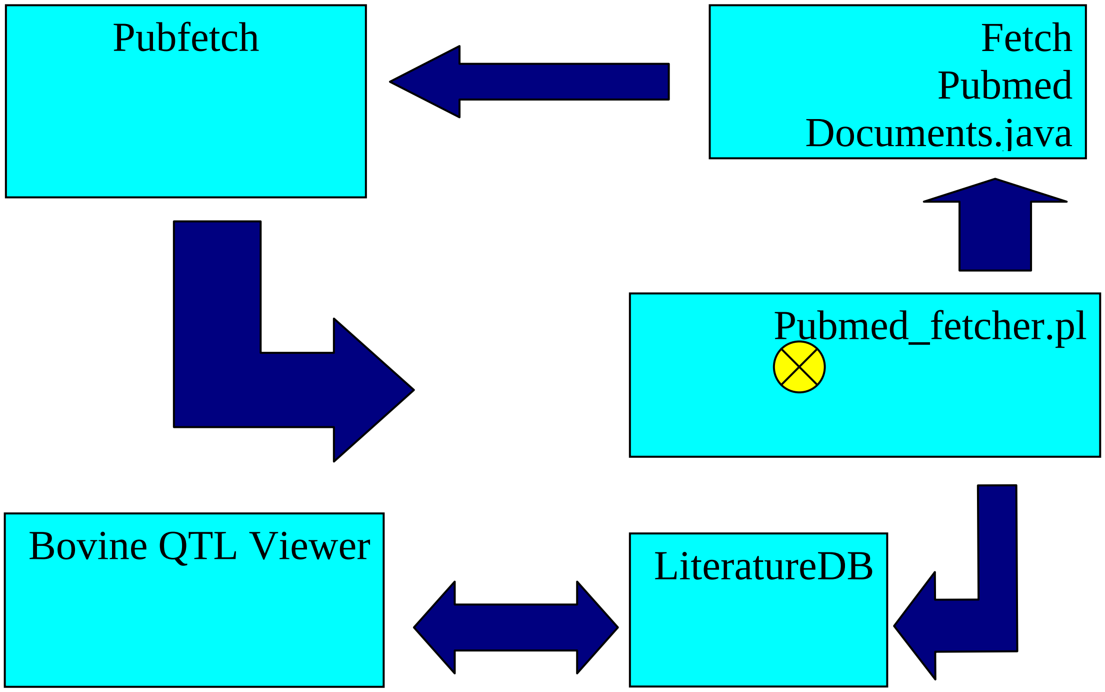


Figure 12 - Literature search data flow diagram.

This module searches the specified databases for key terms, in this instance it searches for the terms ‘bovine’ and/or ‘QTL’ in the paper’s title or abstract [43]. If this is true, features of the paper are stored in a local database (MySQL). A web page written with PHP, within the bovine QTL viewer’s administrator site, displays the data associated with the literature. An administrator can then check a summary page for papers that may need to be reviewed. The administrator can also delete papers that are not useful on the website, which are automatically deleted from the database. This tool is specifically designed to facilitate bovine QTL database administration in the QTL curation process.

One of the benefits of the literature search tool is that the administrator does not need to be privy to the activities of the backend. When the administrator logs into the viewer, there are additional options in the options panel from which the administrator can select. The literature search option has been added to the panel (Figure 13). Once

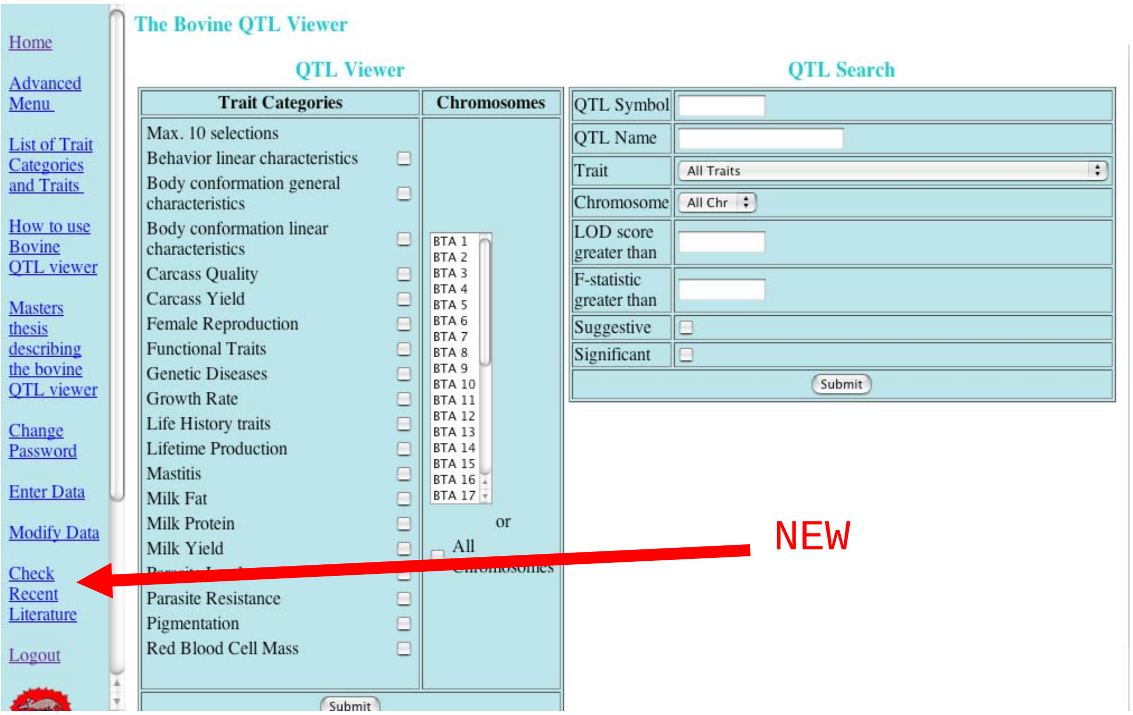
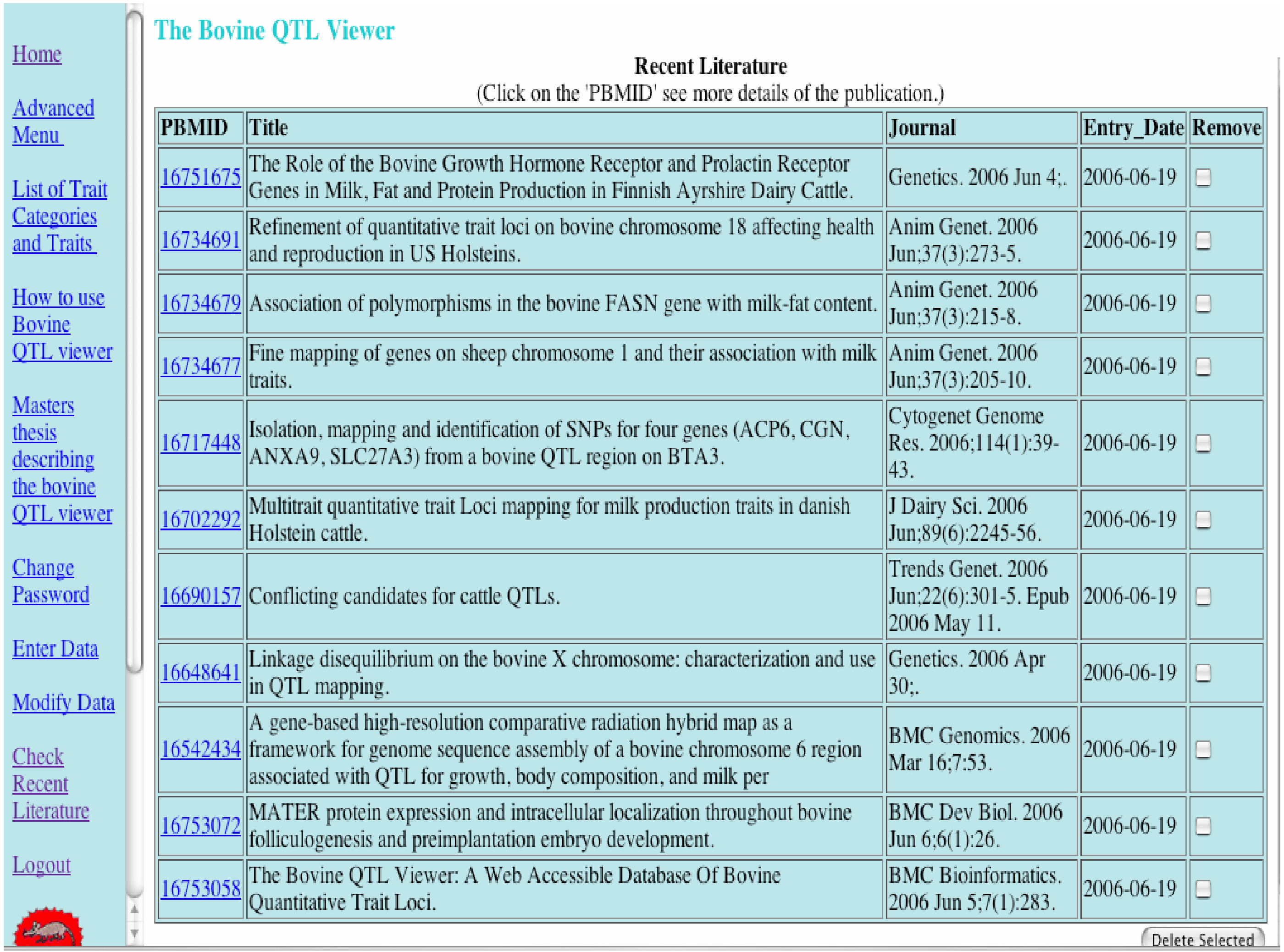


Figure 13 - Literature search option.

the administrator selects the link; he/she is taken to a page that presents all the QTL literature that is currently loaded into the literature search database (Figure 14). This page presents the unique identification for each paper in the database, which is also the pubmed id for that article of literature. The user is shown the manuscript title,


Home
Advanced Menu
List of Trait Categories and Traits
How to use Bovine QTL viewer
Masters thesis describing the bovine QTL viewer
Change Password
Enter Data
Modify Data
Check Recent Literature
Logout
Delete Selected

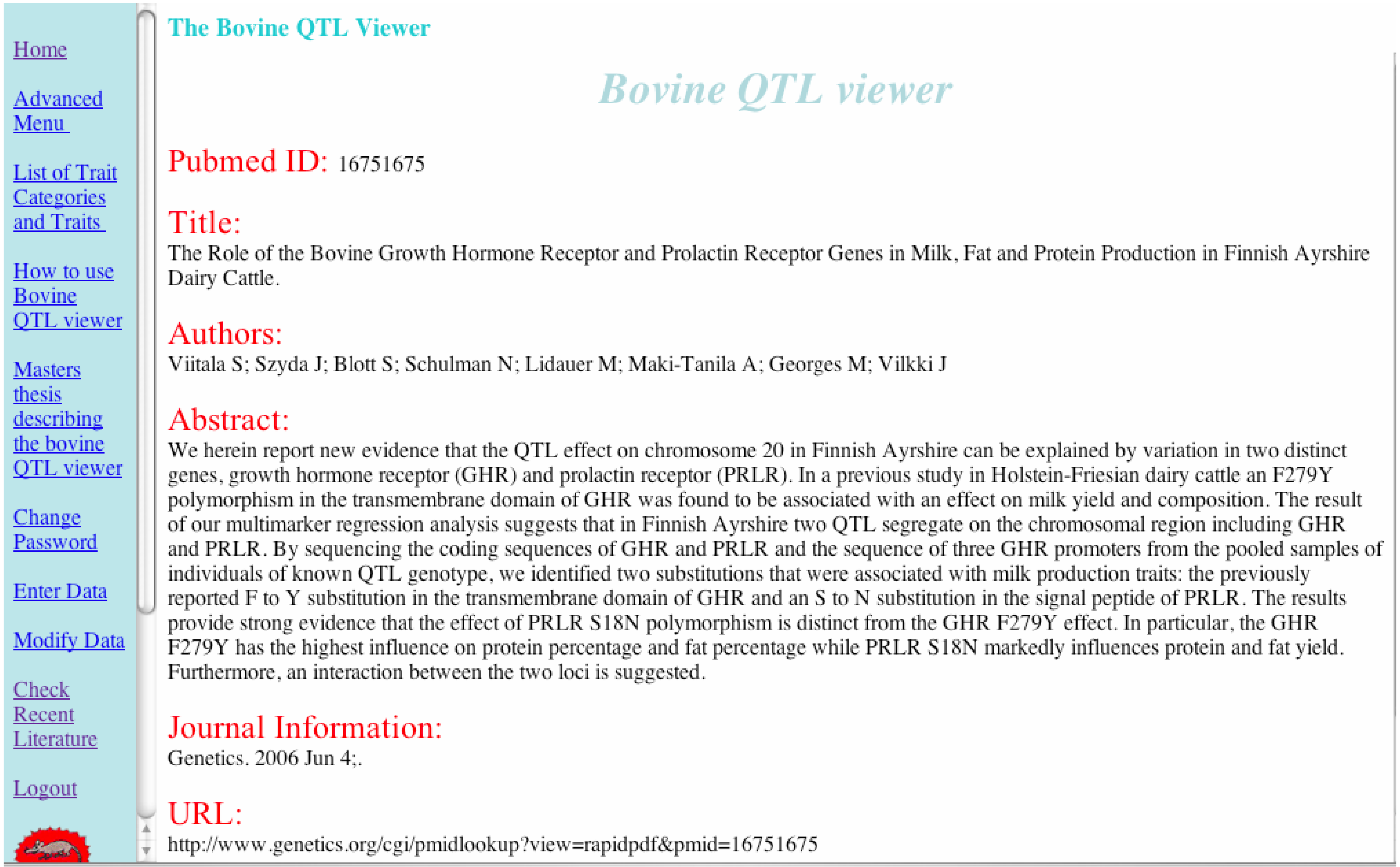
The Bovine QTL Viewer

Recent Literature

(Click on the 'PBMID' see more details of the publication.)

PBMID	Title	Journal	Entry_Date	Remove
16751675	The Role of the Bovine Growth Hormone Receptor and Prolactin Receptor Genes in Milk, Fat and Protein Production in Finnish Ayrshire Dairy Cattle.	Genetics. 2006 Jun 4;.	2006-06-19	□
16734691	Refinement of quantitative trait loci on bovine chromosome 18 affecting health and reproduction in US Holsteins.	Anim Genet. 2006 Jun;37(3):273-5.	2006-06-19	□
16734679	Association of polymorphisms in the bovine FASN gene with milk-fat content.	Anim Genet. 2006 Jun;37(3):215-8.	2006-06-19	□
16734677	Fine mapping of genes on sheep chromosome 1 and their association with milk traits.	Anim Genet. 2006 Jun;37(3):205-10.	2006-06-19	□
16717448	Isolation, mapping and identification of SNPs for four genes (ACP6, CGN, ANXA9, SLC27A3) from a bovine QTL region on BTA3.	Cytogenet Genome Res. 2006;114(1):39-43.	2006-06-19	□
16702292	Multitrait quantitative trait Loci mapping for milk production traits in danish Holstein cattle.	J Dairy Sci. 2006 Jun;89(6):2245-56.	2006-06-19	□
16690157	Conflicting candidates for cattle QTLs.	Trends Genet. 2006 Jun;22(6):301-5. Epub 2006 May 11.	2006-06-19	□
16648641	Linkage disequilibrium on the bovine X chromosome: characterization and use in QTL mapping.	Genetics. 2006 Apr 30;.	2006-06-19	□
16542434	A gene-based high-resolution comparative radiation hybrid map as a framework for genome sequence assembly of a bovine chromosome 6 region associated with QTL for growth, body composition, and milk per	BMC Genomics. 2006 Mar 16;7:53.	2006-06-19	□
16753072	MATER protein expression and intracellular localization throughout bovine folliculogenesis and preimplantation embryo development.	BMC Dev Biol. 2006 Jun 6;6(1):26.	2006-06-19	□
16753058	The Bovine QTL Viewer: A Web Accessible Database Of Bovine Quantitative Trait Loci.	BMC Bioinformatics. 2006 Jun 5;7(1):283.	2006-06-19	□

Figure 14 - Literature search results.


Home
Advanced Menu
List of Trait Categories and Traits
How to use Bovine QTL viewer
Masters thesis describing the bovine QTL viewer
Change Password
Enter Data
Modify Data
Check Recent Literature
Logout

The Bovine QTL Viewer

Bovine QTL viewer

Pubmed ID: 16751675

Title:
The Role of the Bovine Growth Hormone Receptor and Prolactin Receptor Genes in Milk, Fat and Protein Production in Finnish Ayrshire Dairy Cattle.

Authors:
Viitala S; Szyda J; Blott S; Schulman N; Lidauer M; Maki-Tanila A; Georges M; Vilkki J

Abstract:
We herein report new evidence that the QTL effect on chromosome 20 in Finnish Ayrshire can be explained by variation in two distinct genes, growth hormone receptor (GHR) and prolactin receptor (PRLR). In a previous study in Holstein-Friesian dairy cattle an F279Y polymorphism in the transmembrane domain of GHR was found to be associated with an effect on milk yield and composition. The result of our multimarker regression analysis suggests that in Finnish Ayrshire two QTL segregate on the chromosomal region including GHR and PRLR. By sequencing the coding sequences of GHR and PRLR and the sequence of three GHR promoters from the pooled samples of individuals of known QTL genotype, we identified two substitutions that were associated with milk production traits: the previously reported F to Y substitution in the transmembrane domain of GHR and an S to N substitution in the signal peptide of PRLR. The results provide strong evidence that the effect of PRLR S18N polymorphism is distinct from the GHR F279Y effect. In particular, the GHR F279Y has the highest influence on protein percentage and fat percentage while PRLR S18N markedly influences protein and fat yield. Furthermore, an interaction between the two loci is suggested.

Journal Information:
Genetics. 2006 Jun 4;.

URL:
http://www.genetics.org/cgi/pmidlookup?view=rapidpdf&pmid=16751675

Figure 15 - Detail page for literature search.

citation, the date it was entered into the literature database and a checkbox to delete. If it is clear that the article is not of interest based on the title or after analysis, the administrator can delete it directly from the viewer without having to manually access the database. If the administrator finds an article of interest, he/she can click on the pubmed id and be taken to a detail page for the manuscript. The detail page presents an overview of the article of interest (Figure 15). A pubmed id, title, list of authors, full abstract, journal information and URL are displayed. This provides all information needed for an administrator to access the article of interest. The literature search tool has proven to be a useful tool because it minimizes time spent and maximizes returned literature for the current administrator.

All tools presented thus far have been adaptations to new data and have been built on top of the viewer. However the fourth and final assembly, increasing amounts of SNP data and the use of SNPs to map QTL warrant a fundamental shift in the viewer. In order to address this need, a new viewer has been created to contend with the current and future state of bovine genomic data.

2.4 An Object Oriented Bovine QTL Viewer

An object oriented Bovine QTL Viewer was created to address needs of the cattle research community. A fundamental re-structuring of the first generation Bovine QTL Viewer was necessary due to new data and an outmoded model/language. The benefit and usefulness of the updated viewer is that: (a) It helps uncover potential relationships between disparate elements of data; (b) It conglomerates data from multiple sources, from multiple data types into a single database/viewer and c) It is built for new data released and is straightforwardly expandable for potential future data.

The second generation viewer is built using the GOOF model [39]. GOOF is a flexible and expandable informatics framework for organizing genomic data. Part of the beauty of GOOF is that it implements object oriented programming (OOP). OOP is a style of programming that uses "objects" and their interactions to design applications and computer programs [44]. An "object" is essentially a bundle of variables and related methods. The object can then be called and manipulated as the programmer desires throughout the project. GOOF implements OOP and is divided into three general layers: the data model layer, the data transaction layer and the web presentation layer. These layers are interconnected and data flows seamlessly between them.

The data model layer is where objects are created and defined. Many of the objects implemented in the Bovine QTL Viewer stem from the Chado database schema. A table in the Chado schema is a data object in the data model layer and each column of a table is a property of that object. Chado objects are built in within GOOF and did not require initialization. However, the data model layer has an interface to the data transaction layer. Chado objects can be manipulated to create relationships that will be used in the transaction layer. Through the addition of criteria on data objects, GOOF implementers can construct queries that are catered to their data. This prevents direct database queries. The Bovine QTL Viewer utilizes the "Feature" object created in GOOF. The main file components of the data model layer within the Bovine QTL

Viewer include the FeatureDao.java file and FeatureDaoHibernate.java file (Figure 16). FeatureDao.java is where the interface to the data transaction layer from the data object model layer actually takes place. FeatureDaoHibernate.java is where the criteria for an object is set, thereby manipulating the data from the Chado backend database in response to the needs of the viewer. Essentially, the query is made at this level using a language known as Hibernate SQL (HSQL). Being able to add criteria, rather than build long queries, simplifies the process. Therefore, object oriented data modeling provides the luxury that any change of the underlying database schema will be irrelevant because the data operations are independent of the underlying database.

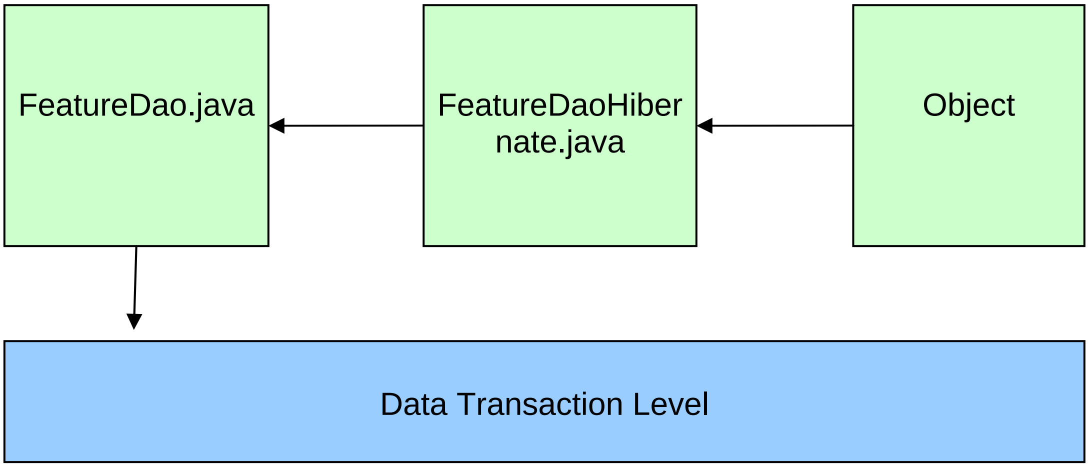


Figure 16 - Data model layer of the viewer.
The primary files of the data model layer for the Bovine QTL Viewer.

Once objects were set and an interface between the objects and the data transaction layer had been created, a transaction layer was established to link Chado objects to the Bovine QTL Viewer web interface (Figure 17). The data transaction layer delivers the data objects as requested by users on the web presentation layer. A data transaction manager was established for the target bovine QTL Chado objects. This manager is responsible for building the data transaction interfaces for the target data objects. One can think of the data transaction manager as a swing door. A swing door is a door that swings on a double hinge allowing it to swing both ways. The transaction manager is capable of letting data flow both ways; from the web layer to the data model layer or vice versa. The transaction manager for the Bovine QTL database is FeatureManager.java and currently holds calls to the hibernate query construction class.

Within the transaction layer the data transaction manager, FeatureManager.impl was set up to establish connections with data objects which the interface will communicate with. This file serves as the implementation of the data transaction manager. The name of the connection within the implementation was kept the same as that of the query class in the data model layer. Although not necessary, it was set up to maintain consistency.

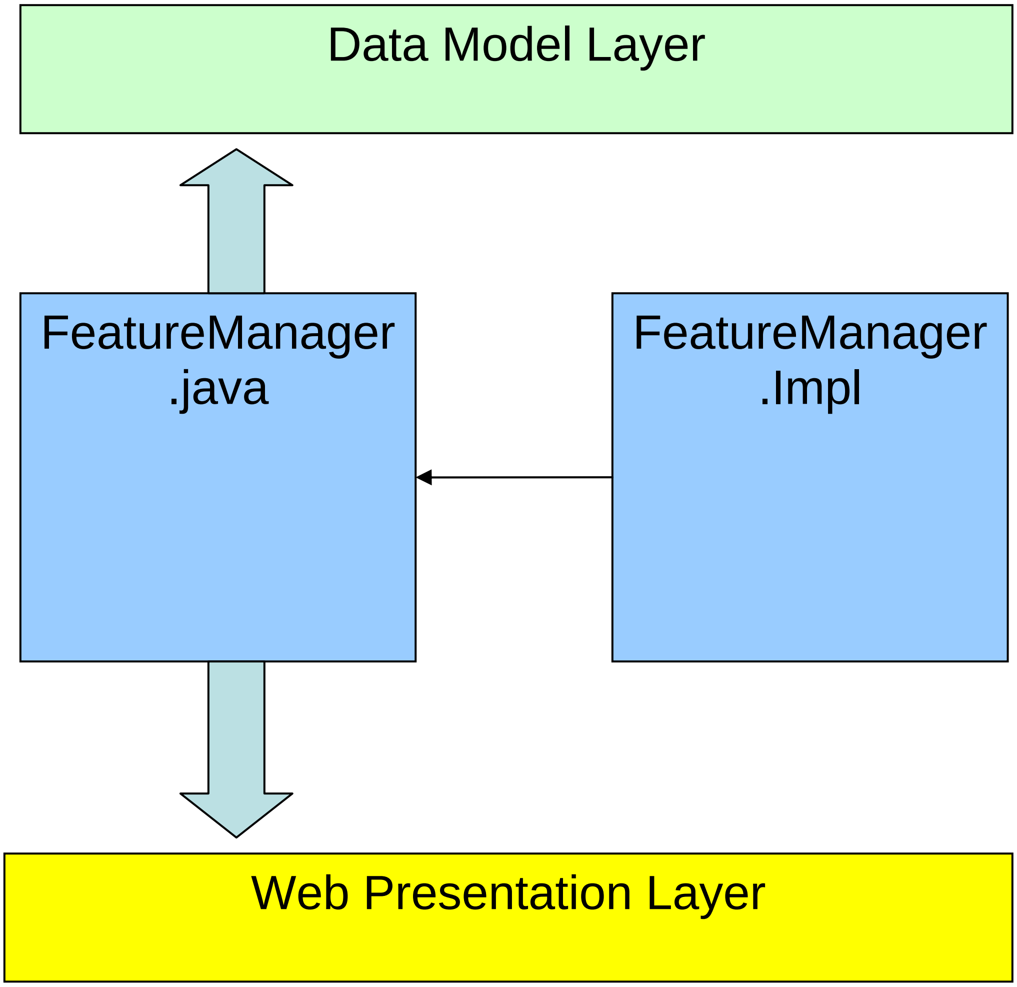


Figure 17 - Data transaction layer of the viewer.

The data transaction layer serves as the middleware in the goof design, a buffer between the data model layer and the web presentation layer. For the future of the Bovine QTL Viewer, new bovine annotation can be made as objects and reused with old actions allowing for greater versatility.

The web presentation layer is where the overall World Wide Web design is created for the user. In the web presentation layer "actions" are performed that when invoked by a user provides direction to the data flow (Figure 18). For example, in the Bovine QTL Viewer when a user selects a trait of interest and clicks the button to see the result, an action is performed and data flows to the data object layer and back to the web presentation layer. JSP, Java Server Pages, were developed to provide web content for the user to see. The JSP pages form what is known as the view module within the GOOF package. This module is composed of all the layouts of the user interface. Further, web text variables are defined within a layer known as the web interface resource definition layer. This layer facilitates management and maintenance of the interface.

In the Bovine QTL Viewer, actions are defined within the FeatureAction.java file. This file maintains all the methods in which actions are being performed. Each web action within FeatureAction.java is connected to the data transaction manager, FeatureAction.java, as requested by the user defined action and delivers the variable necessary to the transaction manager to pass to the data query. The xwork.xml file is a director of data flow responsible for direction web pages through actions. When the user performs an action on the web front end, xwork.xml directs the flow of web pages through a pre-directed action. With regards to the view model layer within the Bovine QTL Viewer, ApplicationResources.properties stores each web interface component variable. These variables are called as needed; however they are able to be called via any of the current JSP pages in the viewer.

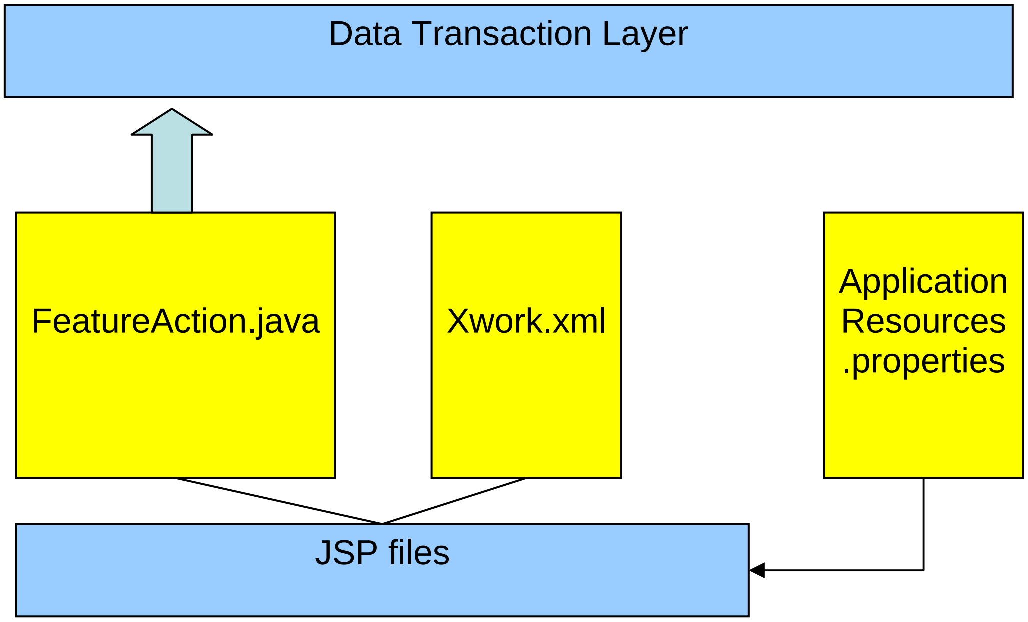


Figure 18 - Web presentation layer.

The second generation Bovine QTL Viewer allows users to approach data from a multitude of angles and further allows for more non-specific querying. Non-specific querying means that the user does not have to know exact terms or regions to be able to find their annotation of interest.

The title page presents the user with a brief description of the viewer and its mission statement. “Search the Database” allows the user to search the database for whatever may match his/her string. Similar to an entrez search, a list of matches will be reported. A list of matches and the number of matches are reported on the next web page. For example, if the user enters the term “Holstein” he/she will get a return of the number of QTL that come from the Holstein breed, the number of haplotypes available for the Holstein breed so on and so forth. The user can continue by clicking the category of interest. If the user selects QTL, he/she will be linked to a page that lists all the QTL that match the term selected. Each QTL can be clicked on for a detailed view of its information.

This type of search illustrates the benefit of moving to a second generation viewer. A new viewer that incorporates all this disparate data into a single database allows for more powerful queries and streamlines research efficiency.

Similar to the first generation QTL viewer, the user can query QTL by trait. When the user selects this option, he/she will be linked to a page that allows them to select a trait from a drop down menu. A chromosome of interest or all chromosomes can be selected. The query can be streamlined based on lod score, F-statistic and significance level. After completing the form, the user will see a page that illustrates the QTL that resulted from his/her query. An individual QTL can be selected and the user is sent to the aforementioned detail page.

The QTL detail page presents the bovine QTL identification number, trait from which it is derived, chromosome, F-statistic, lod score, family, and method of derivation. The significance level (whether the QTL is suggestive or significant) and status of availability (public/private) is presented as well. In addition, Gbrowse view of the QTL is embedded into the page. The user can click on the Gbrowse image in the viewer and utilize all of the features available through Gbrowse.

The user also has the added benefit of being able to search for haplotype blocks. When the haplotype search is clicked the user is directed to a page where additional information is requested. The user is prompted for a breed of interest, query region start, query region end and chromosome of interest. The user is then directed to a page that lists the haplotypes within the specified query. The user can then either view the SNPs that fall within the region, or click to a Gbrowse view of the haplotype block of interest.

Finally, the user has the option to just browse the genome. When the browse option is selected, the user is directed to a Gbrowse view where he/she is free to roam the genome for regions or features of interest.

The construction of the second generation Bovine QTL Viewer has been based on the needs and preferences of the cattle research community. In order to create a quantifiable method of determining the usefulness of the viewer, I have used survey methods to query my target audience.

3. Results

3.1 Evaluation

Before Construction of the Bovine QTL Viewer this author created a survey that would query members of the community about what they liked/disliked about the first generation Bovine QTL Viewer and where their data acquisition interests lie. The benefit of a pre-development survey is to eliminate any bias a user can acquire from a presented visual. Therefore, an illustration lacking survey is capable generating such data.

The survey created was geared towards answering three main premises from potential Bovine QTL Viewer users:

- What is the most desirable search style
- What are the most common data queries, and most interesting?
- How do most users prefer to view the data.

Questions such as: “What DO you like about the Bovine QTL Viewer” and “What DON’T you like about the Bovine QTL Viewer” are general questions aimed at determining whether there are any features of the viewer that are brightly positive and should be migrated to the second generation viewer, or what features may be very negative and unappealing and therefore not migrated. Other questions such as “Based on the data you normally deal with, what types of queries would you use on a genome browser”, are designed to determine what types of data the users are generally interesting in requesting. Further, for determining how most users prefer to view the data, questions such as “What search engine do you like to use and why?” and “Have you used entrez search before? What did or did you not like about it?” were asked.

A total of 10 people from the cattle research community were provided the survey and 5 people were able to complete it. Figure 19 illustrates responses to general questions regarding the first generation Bovine QTL Viewer. The frequency by which people expressed satisfaction and dissatisfaction with aspects of the first generation viewer is represented by a pie chart. A majority of users expressed satisfaction being able to search for QTL by trait or gene of interest. However, a number of users expressed dissatisfaction with issues pertaining to PHP forms. When a user enters one page, if he/she were to return to the previous page using the back button on the browser, he/she would lose data due to a refreshing of a form on the back end via PHP.

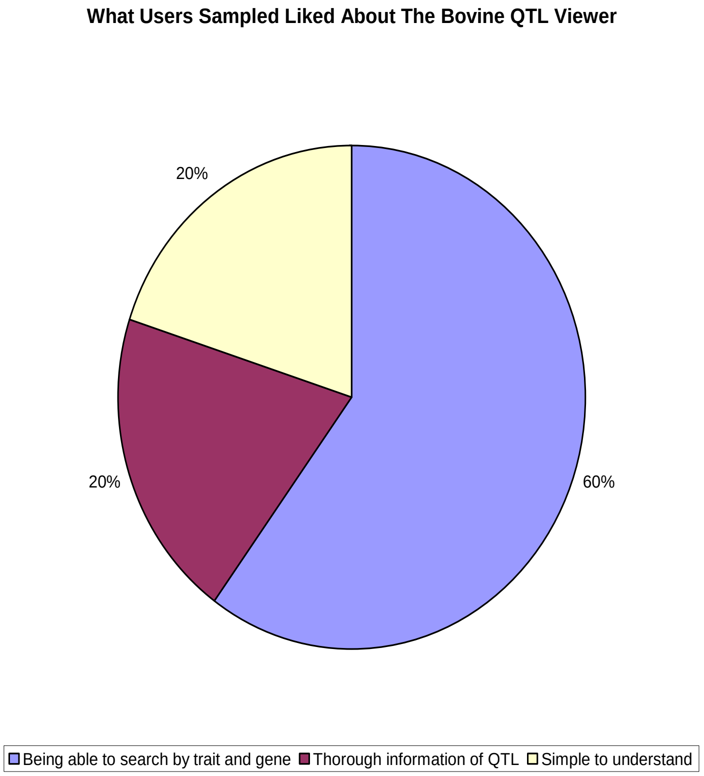


Figure 19 - User likes from previous viewer.

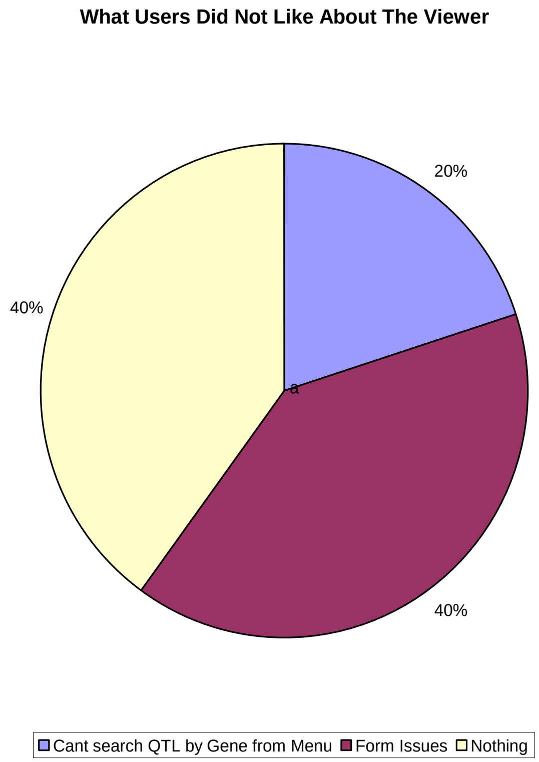


Figure 20 - User dislikes from the previous viewer.

After getting a sample of user likes/dislikes, users were prompted as to what is preferable search style. When asked what is their favorite search engine 100% of users surveyed responded as Google. When asked why Google is their favorite responses varied but all generally stated that the search results were very accurate.

A number of questions were created towards answering what types of data is preferable for the users research objectives. One question: “If applicable, do you normally look for QTL by:

Trait?

Manuscript?

Chromosome?

Location?

Overlaps?

All of the above? “

was created to paint a picture as to the data interests of most cattle research users. Figure 21 shows that most users are interested in searching QTL by trait and chromosome. Of those sampled, none selected all of the above and non searched QTL by overlaps.

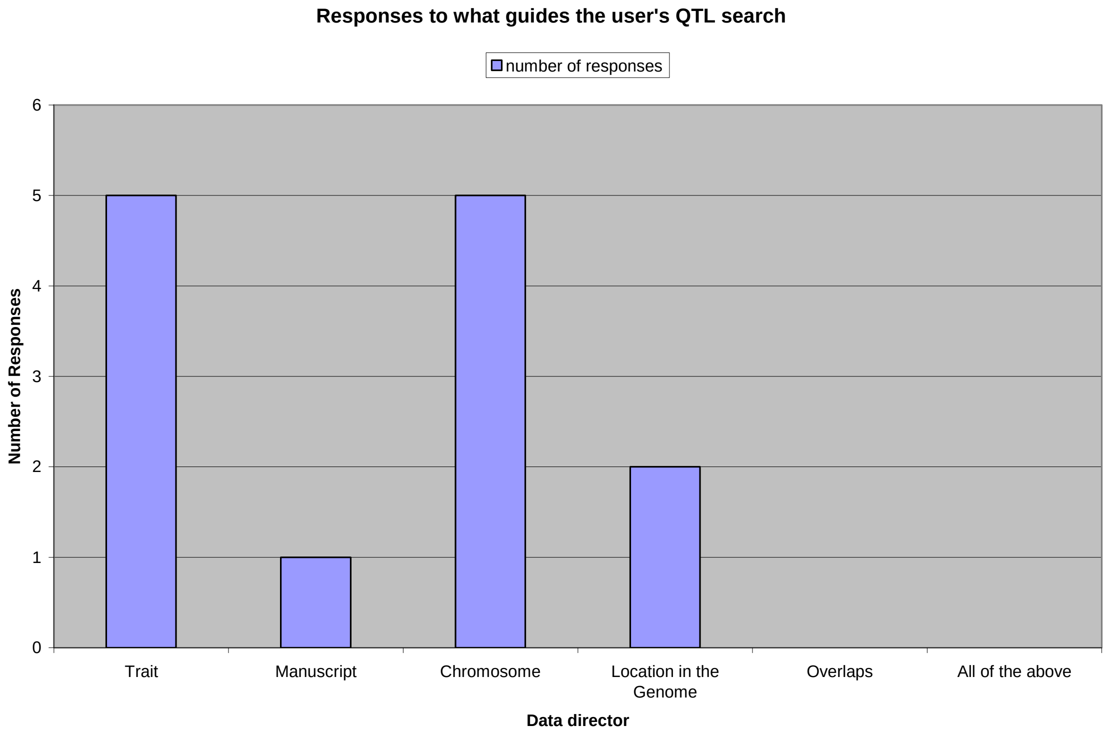


Figure 21 - User response to QTL search preference.
Displays the responses to how they normally search for QTL. (i.e. what data directs their queries.)

A post development survey was developed in order to quantifiably measure whether objectives of the second generation viewer have been met. The survey is primarily focused on querying users' satisfaction with:

- Ease of use
- Accessibility of novel data
- Overall experience

Preliminary post development survey results have shown a general satisfaction with the tool. The survey was completed by three members of bovine research community. When asked to rate their overall experience with the second generation Bovine QTL Viewer an average rating of 4.3 out of five was given.

Evaluation of the use of the Bovine QTL Viewer by pre- and post- migration survey provides a source of quantitative measurement. Given strong results from the pre- development survey (i.e. 100% of users said they preferred to search for QTL by trait) selected attributes of the first generation QTL viewer were migrated to the second generation viewer. In the future, an online survey can be attached to the Bovine QTL Viewer to be able to continually update the viewer based on user preferences.

4. Discussion

As the complexity of bovine genomic data has increased, the Bovine QTL Viewer has adapted to its bovine genomic surroundings (Figure 22). The Bovine QTL Viewer has served as a bioinformatic solution that has helped illuminate relationships between bovine genomic annotation and sequence. Bioinformatic solutions have been shown to be effective measures for bridging the gap between sequence annotation and biological interpretation [1, 18, 45-50]. Furthermore, haplotypes have increasingly become useful genomic features for interpreting phenotype. Understanding the relationships between haplotypes, SNPs and QTL can help researchers identify functional regions of the cow genome and therefore, allow cattle researchers to efficiently enhance economically important traits. In order to incorporate SNP and haplotype data and the release of the fourth and final assembly with genomic annotation, this author has created an object oriented Bovine QTL Viewer. In addition to all data presented in the first generation viewer, the viewer now stores and displays additional bovine genomic annotation including SNPs, STSs, and haplotype blocks. The second generation Bovine QTL Viewer is less dependent on the backend database and is free of the outdated PHP programming language of the first generation model. I have implemented the GOOF model for design and this has provided future expandability due to the three layered approach. One layer of the second generation viewer is dedicated to objects, which are bundled methods and values from the database. A second layer of the viewer is dedicated to transactions between the web front and the back end database.

The third and final layer, web presentation layer, is dedicated towards provided an interface between the user and the database through the World Wide Web. The backend database (PostgreSQL) implements the Chado schema for data storage. I have quantified the use and usability of the viewer through pre- and post- development survey of researchers in the cattle research community.

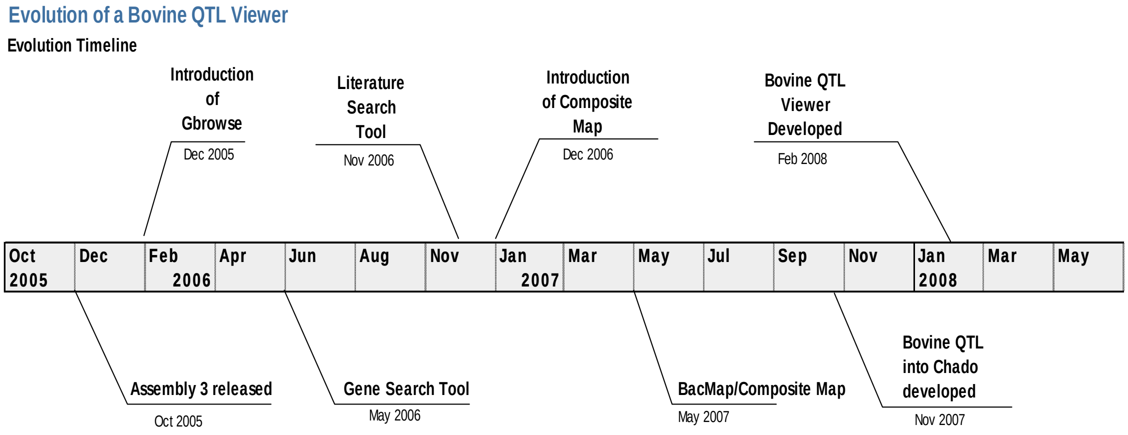


Figure 22 - Evolutionary timeline for the Bovine QTL Viewer.

For future work, the backend Chado database allows for expandability of data. In the event that new data or data types can be added to the viewer, the schema is designed to incorporate many different types of data particular to the *Bos taurus* species. The Bovine QTL Viewer implementation of the GOOF schema allows for adaptation of new additional data types to be presented and queried with little modification in the future.

Some future modifications to the second generation Bovine QTL Viewer can still be performed to further improve the tool. An online survey can be attached to the Bovine QTL Viewer to drive updates to the viewer based on user preferences. Based on user response on the online survey, if users would prefer to view QTL on linkage maps, linkage mapped chromosomes can be stored within the Chado schema. Storing linkage mapped chromosomes into the Chado database allows for illustration of the linkage maps.

CHAPTER IV

GENOME WIDE GENE ONTOLOGY META-ANALYSIS

1. Introduction

A key open question in the understanding of the function of the bovine genome is if genes and their known functions relate to their position within the genome itself. Some regions, such as chromosome ends and near centromeres, are expected to have little function [51, 52]. However, the relationship between genes, their known functions and quantitative trait loci is not well understood, especially within the bovine genome. By grouping QTL into non-redundant regions we have been able to analyze gene density and gene function (based on Gene Ontology) distribution across the genome with relation to their positions within, outside QTL regions and across the bovine genome.

A gene is defined to be a locatable region of genomic sequence, corresponding to the basic unit of inheritance, that codes for a protein [53]. Inheritance of some traits depends on the effects of multiple genes, each contributing fractionally to the phenotype. Some of these polygenic traits are also known as quantitative traits, which are encoded by Quantitative Trait Loci (QTL) [54]. Examples of quantitative traits include traits such as height and hair color which vary widely and are measured on a quantitative scale. QTL are located using linkage mapping [55]. *Bos taurus* has many traits which have been measured on a quantitative scale. Many of these quantitative traits are of economical and agricultural significance, so there is significant motivation to understand the nature of QTL and identify the gene or genes within a QTL responsible for the trait. The process of locating and understanding QTL in the genome is also important to the manipulation of genes in breeding programs and to the cloning and studying of genes in order to identify function [38].

Genes from many species have been intensely studied and their products have been characterized. However, the large and varied terminology for synonymous products has inhibited searching by both humans and machines [56]. The Gene Ontology Consortium has therefore provided a dynamic, controlled vocabulary for describing gene products in any organism [56, 57]. GO currently stores over 18,000 terms, each of which has an accession number, a name, a more detailed definition and other information relating a term to its parent terms. The GO project has developed three main structured, controlled vocabularies, or ontologies, that describe gene products in terms of their associated biological process, cellular components and molecular functions. Molecular function describes the biochemical activity of a gene product. Biological process describes the objective or biological goal to which a gene product contributes and cellular component describes the place in the cell to which the gene product is localized. The controlled vocabularies are structured so that they can be queried at multiple levels.

The ontology structure is composed of a directed acyclic graph, similar to a hierarchy (Figure 23). However they differ in that a child or deeper level term can have many parents, or further up the chain, less specialized parents [56]. Individual terms in an ontology are related by an ‘is-a’ or ‘part-of’ relationship. The described terms do not themselves describe specific genes or gene products, rather collaborating databases generate associations of GO terms to specific gene products [58]. Every GO term must follow what is known as the “true path” rule: if the child term describes the gene product, then all its parents must also apply to that product. Therefore, every go term every product on the path towards the final term of interest, must be in someway associated to the product in question [56].

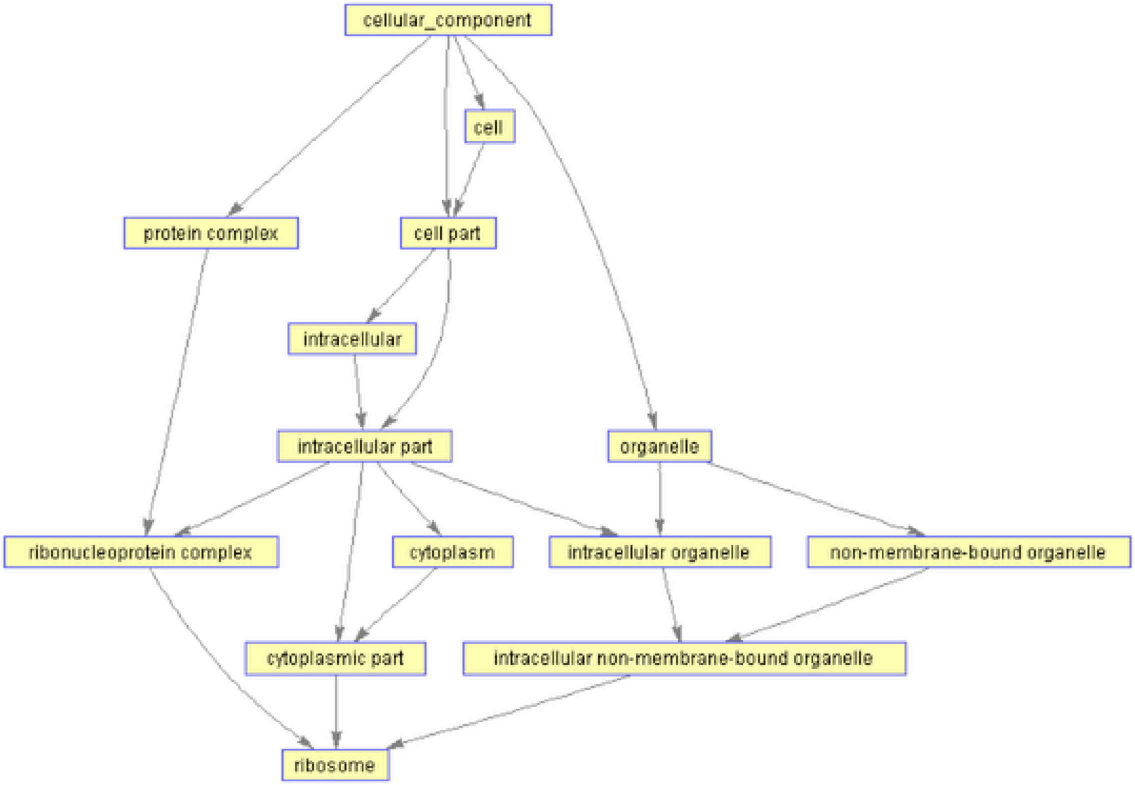


Figure 23 - Acyclyc form of GO [59].

It has been observed that on occasion genes are grouped into clusters in which several copies of genes with similar function are located near one another [60]. Within the human genome, growth hormone genes and genes encoding alpha and beta globins have been shown to be clustered, or grouped together, in the genome. Some genes located in clusters are actually gene families. A gene family is a set of biochemically similar genes with known homology [54]. However, clustered genes that are not part of gene families are not fully characterized. Particularly within the bovine genome, not much is known about genes with similar function clustered together regionally. Understanding functional gene clustering within QTL of similar function can provide added insight into how QTL regions are adding to agriculturally important quantitative traits. Therefore, it is of value to investigate whether genes are functionally clustered using the functional regions from QTL as a guide and the functional classes provided by GO terms as a trail. In this work, we developed a GO-based genome-wide *in silico* approach towards investigating the distribution of genes with respect to QTL regions and the distribution of functional aspects of the genes within the regions. QTL were grouped into 8 broad categories and analyzed to identify the relationship of the frequency of the GO in comparison to the QTL category of interest. Using data provided by Dr. David Lynn, NCBI, the Baylor Human Genome Sequencing Center, the Bovine QTL Viewer Database and GO, we showed a highly significant association between gene density and QTL regions [14, 56]. We further showed significant associations between GO terminology frequencies related to QTL category type within regions of the genome where the particular category of QTL spans non-redundantly. In addition, we analyzed the distribution of selected terms known to be associated with QTL category types inside and outside of QTL category regions and across the entire genome. Our analysis also provided the opportunity to develop a GO slim specific to bovine QTL and the bovine genome.

2. Materials and Methods

At the start of the analysis, the bovine QTL Viewer database housed 843 bovine QTL. For the development of a Gbrowse view for the bovine QTL viewer, flanking sequence tagged site markers were used anchor the QTL onto the third bovine genome assembly. NCBI has produced a set of sequence tagged sites placed onto the bovine genome assembly 3.1. Based on this publicly available bovine sequence tagged site data set, we were able to anchor 467 bovine QTL to the bovine genome. In order to increase the number of QTL for both view and analysis, flanking markers from unanchored bovine QTL were placed onto the assembly using the BLAST local alignment tool.

2.1 Locating Markers

STS sequence data was collected from the NCBI website using a Perl script. A local blast database for the bovine genome 3.1 assembly was created. All markers with unknown locations were blasted onto this assembly. Markers with 100% identity and

90% of full length were isolated. Top hits from this group were verified for chromosome by comparing with the marker's position on the linkage map. This process was repeated using MegaBLAST. However, after BLAST analysis, some markers were unable to be placed for two reasons: (1) BLAST analysis was not able to locate sequence matches with the specified criterion and (2) Some markers did not have NCBI accessions.

Markers that did not have NCBI accessions could not be anchored to genome via the BLAST/MegaBLAST tool due to a lack of sequence data [61]. All markers that did not have NCBI accession values and those that could not be placed with BLAST were isolated for e-PCR [30]. By searching for closely matching subsequences that have the correct order, orientation and spacing, e-PCR recovers sequence tagged sites. E-PCR allowed for placement of an additional 42 sequence tagged sites. E-PCR parameters were relaxed slightly by allowing as much as 1 gap and 1 mismatch. Placement of maximum possible STSs subsequently allowed anchoring of Bovine QTL.

In general, QTL are anchored to the genome by the flanking sequence tagged sites. Specifically, the starting base pair of the initial sequence tagged site serves as the starting position of the QTL. The ending base pair of the latter sequence tagged site serves as the end point of the QTL. Post BLAST and e-PCR, 603 bovine QTL were anchored onto the bovine genome assembly 3.1.

2.2 Identifying QTL Regions

In order to analyze regions based on binary in QTL or outside QTL, non-redundant QTL regions were generated. Overlapping segments of QTL were grouped and extended. Further, QTL were broken into 8 distinguishable categories based on characteristics. Categories included: Body Conformation, Carcass, Disease Resistance, Fat, Growth, Milk protein yield, Milk yield and Reproduction. QTL were grouped by their respective traits, if the trait was related to the broader category, it was grouped into that category. Figure 24 displays the size distributions of QTL by category.

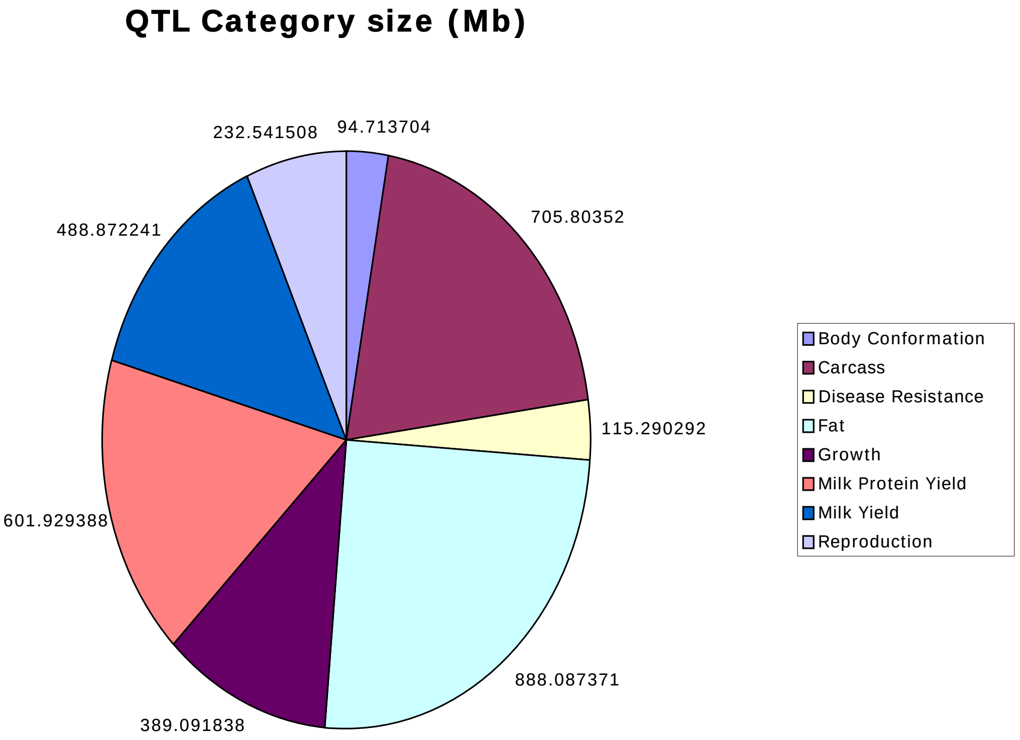


Figure 24 - QTL category sizes (Mb).

Each category of QTL was broken into non-redundant regions. Non redundant regions were created by combining overlapping regions of QTL per category into a single contiguous region. Figure 25 provides a sample of how the growth category of QTL are combined to create non-redundant regions. Non QTL regions are locations of the genome in which no QTL are known to be present. The total bovine genome assembly

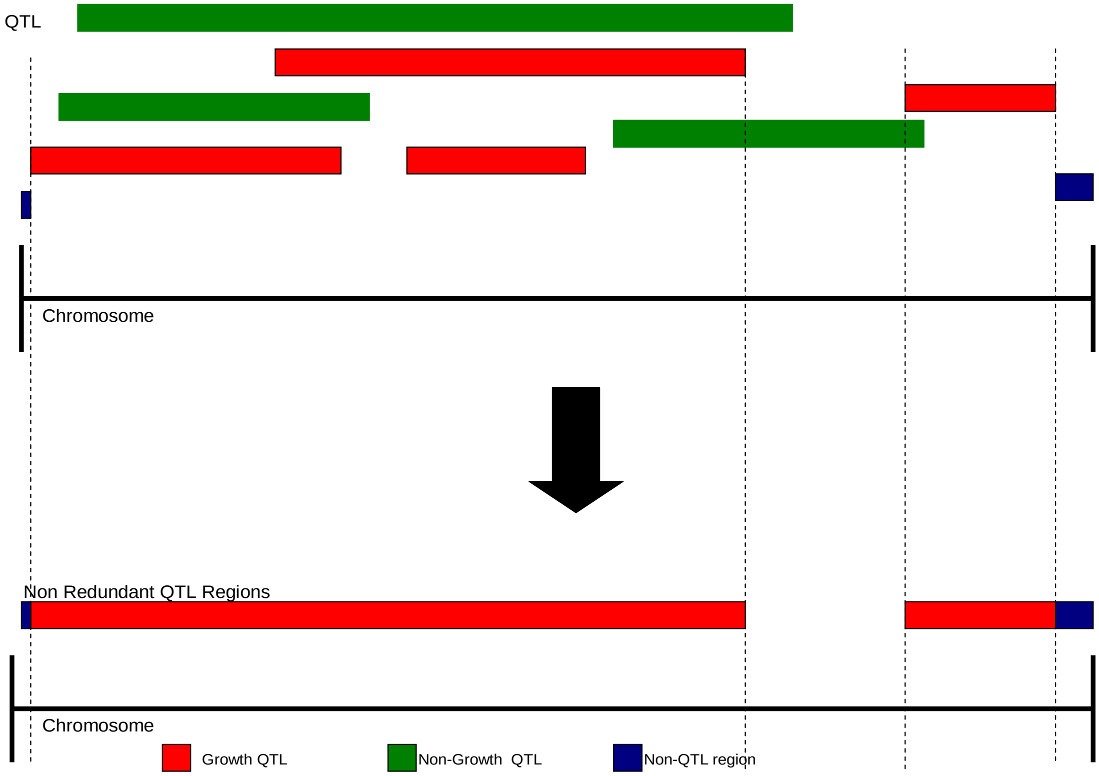


Figure 25 - Grouping QTL into non-redundant regions.

3.1 size is ~2686Mb. Gene Density was then calculated on a per category basis. Gene density is the number of genes located per category over the total size of the category. This result provided an estimation of the number genes found per Mb. QTL region gene density is variable by category and non-QTL region density serves as the control group because of the lack of QTL constraint. Given that non-QTL regions have no known QTL constraint, the gene density value can be used as the control expected density. The expected gene density value for each QTL category serves as a model to analyze variation. Figure 26 displays the variation of gene density between each QTL category gene density against the value to be expected without QTL constraint. Gene density across all QTL regions per category were calculated and compared for significance against non-QTL region gene density. A Chi-square value of 2767.99 with 7 degrees of freedom provides highly significant evidence for QTL regions harboring a bulk of the known genes within the gene assembly.

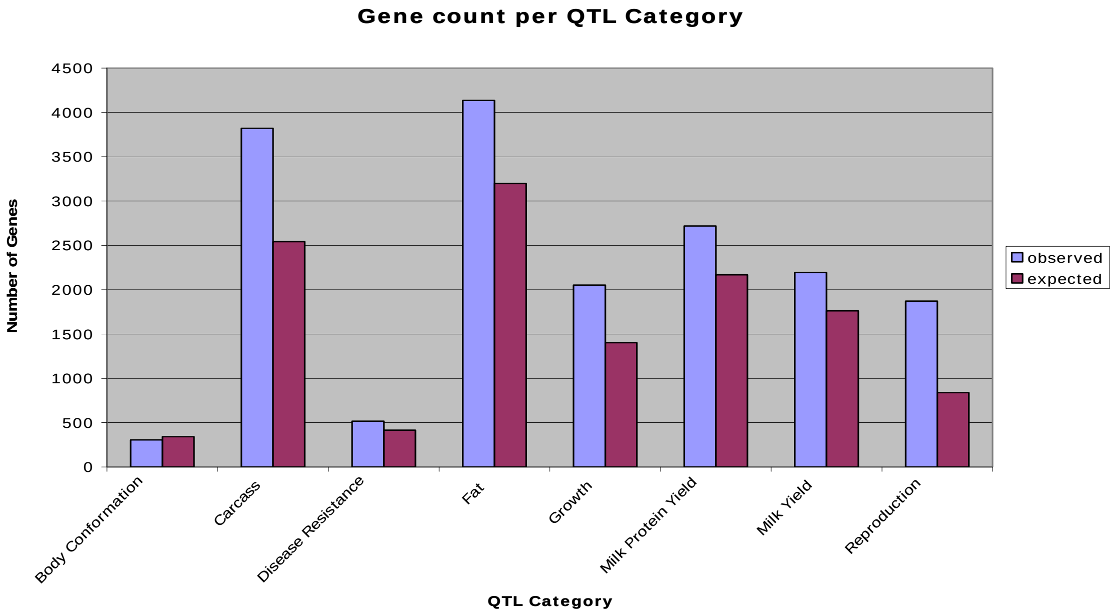


Figure 26 - Gene counts per QTL category.

A second method of determining statistical significance in gene density between QTL regions was used in addition. Each QTL category region was divided into 5Mb bins (Strategy used explained in detail below). Gene counts were measured in each bin across the regions. Plotting the distribution of bin gene counts and numbers of bins produced a normal distribution. This distribution provided confidence that a Welch's t-test could be used to identify statistical significance (Figure 27).

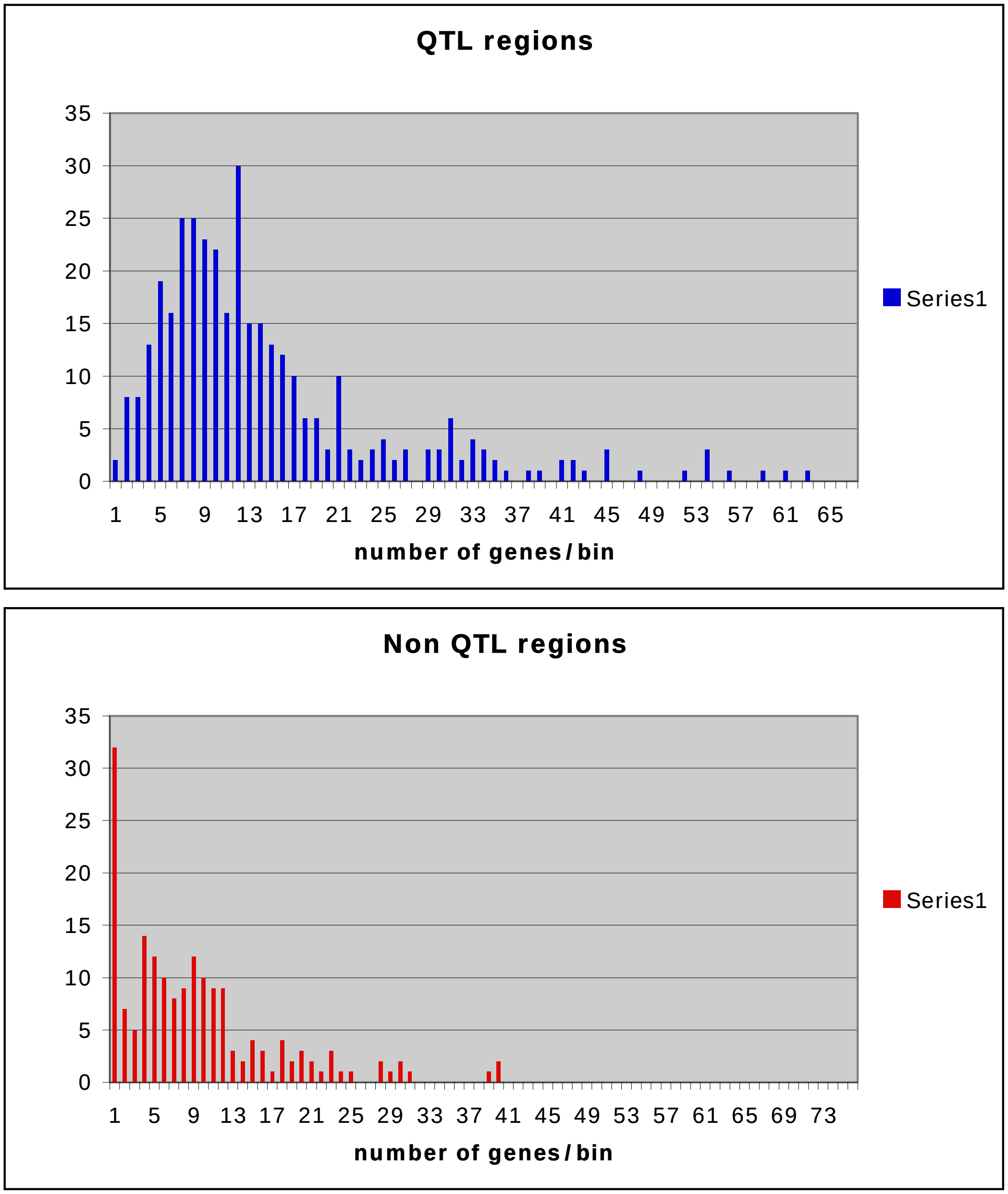


Figure 27– Gene distribution frequency table.
Illustrates the distribution of number of genes per bin within QTL and non-QTL regions. The distributions were analyzed for variance via Welch's t-test.

Table 3 – T-test analysis results.
Statistical results of the Welch’s t-test to identify significant gene density variance between QTL and non-QTL regions.

t-Test: Two-Sample Assuming Unequal Variances		
	QTL	Non-QTL
	Variable 1	*Variable 2*
Mean	5.328358209	2.315789474
Variance	55.34509272	23.73894737
Observations	67	76
Hypothesized Mean Difference	0	
df	111	
t Stat	2.823510335	
P(T<=t) one-tail	0.002815846	
t Critical one-tail	1.658697266	
P(T<=t) two-tail	0.005631692	
t Critical two-tail	1.981566695	

The Welch’s t-test displayed a p-value of 0.00563 which shows a high level of statistical significance when examining the difference between gene density in QTL regions and non-QTL regions (Table 3).

Each individual gene has a controlled structured vocabulary of its gene products in a species-independent manner known as a gene ontology [56]. Each GO has a unique numerical identifier and a term name (i.e. protein amino acid phosphorylation, receptor activity). Each term is further assigned to one of three main ontologies: molecular function, cellular component and biological process. Understanding the distribution of the top level ontologies can provide insight into the function and frequency of known functions of the genes within QTL and non QTL regions.

2.3 GO Analysis

Top level gene ontologies were measured across QTL, non QTL regions and the full genome. The Gene Ontology was provided and made publicly available by the Gene Ontology Consortium. A dump file of the Gene Ontology was downloaded and loaded into a MySQL database via the MySQLdump utility. Locations of the bovine genes were downloaded through ensembl and verified against a dataset provided by Lilian Lau of David Lynn’s lab. 14,354 bovine genes were loaded and stored into a MySQL database. Additional non annotated genes from the GLEAN 5 dataset were included for analysis. A total of 22,418 bovine genes were used for the analysis. For each non redundant QTL region, all bovine genes within the starting base pair of the region and the ending base pair of the region were identified. The ontology of each gene was located and measured iteratively.

In order to measure top level ontologies a GO acyclic graph traversal approach was taken [57]. As ontologies per gene were identified, terms were cycled up the GO tree to identify the top level ontology from which the specified ontology is derived. To account for multiple potential functions of a gene or synonyms of function, each GO term was counted and incorporated into the measurement.

Top level gene ontologies were measured in non-redundant QTL regions based on QTL categories. The methodology for determining the top level GO counts was also through a MySQL database. All genes within the boundaries of non redundant QTL regions per category were identified. GO for each gene was identified and terms counted. GO density was determined based on the number matches of the gene ontology term per total non redundant QTL category genomic size. Measuring top level gene ontologies provides understanding of general processes of genes in given regions, however moving further along the ontology hierarchy can provide a more descriptive understanding of function.

Second level gene ontologies were measured across all QTL regions and non QTL regions in order to obtain a finer image of gene functionality. By using the same strategy as performed for top level gene ontology, second level densities were calculated. After measuring second level ontology density across all non redundant QTL regions and non QTL regions, second level gene ontology densities were measured across individual QTL categories. Figure 28 shows a sample of second level gene ontology density based on QTL category. GO frequency values counted for each QTL category can be compared to expected values calculated from non QTL regions GO density. In order to further verify significance the full genome, non QTL regions and each QTL category regions of DNA sequence were divided into 5 Mb bins. The probability of observing at least as many gene ontologies as was actually observed

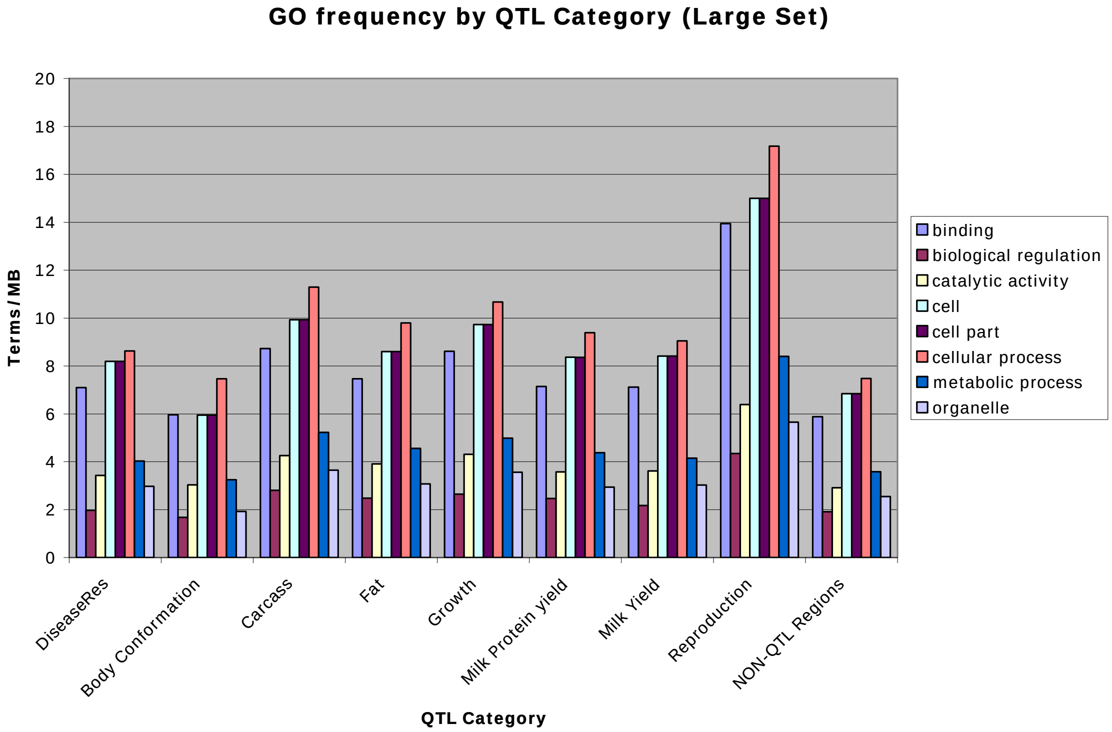


Figure 28 - Second level gene ontology frequency.

in each bin was computed under the assumption of a random (non-clustered) distribution (non QTL regions) [62]. Small probabilities will demonstrate that more gene ontology terms are observed than what would be expected.

2.4 Selected Terms GO Analysis

To take advantage of go syntactical ontology we also searched each genes' ontology for terms associated to each QTL category. Terms were selected based on known widely accepted terminology associated with the category of interest. Table 4 depicts terms searched for each QTL category.

Table 4 - Selected GO associated with QTL category.

QTL Category	Terms
Body Conformation	development, osteo, cartilage, muscle
Carcass	apoptosis, protease, collagenase, extracellular, tenderness
Disease Resistance	immune, defense
Fat	lipid, adipose, fat, trigliceride, insulin
Growth	growth, metabolic, nutrient, endocrine
Milk Protein Yield	protein, translation, amino acid, casein, ribosom
Milk Protein	metabolic, hormone activity, hormone, exocrine, metabolism, mammary, apoptosis
Reproduction	sexual characteristic, reproduction, sperm, oogen, ovulation, secondary sexual characteristic, steroid, hormone, endocrine, estrus

A Perl script was created that stores search terms into an array. All genes that fall within the boundaries of the non redundant QTL regions per category where queried from a MySQL database. The GO terms from each queried gene were mined for syntactical matches of the selected category associated term to the ontology itself. Total matches were calculated for each category. This allowed for the calculation of category associated term density.

Expected associated term gene ontology frequency was calculated based on the non QTL regions. Associated category terms were queried across the non QTL regions of the bovine genome and a density was calculated. The non QTL region associated term density was applied across each QTL category of interest to determine what would be the expected associated term frequency if random. Non QTL regions again serve as the control group. Figure 29 displays the density of associated terms match with GO for each QTL category compared with its expected. With the control group, significance of result can be calculated.

The binning strategy used for second level gene ontology frequency was applied to associated gene term frequency. The number of associated GO were calculated for each 5 Mb bin within each category of interest. The probability of observing at least as many associated gene ontology string matches in observed bins compared to expected was computed under the assumption of random distribution. A Welch's t-test determined the probability value. A small probability will indicate that more QTL category associated terms are matching GO terms than would be normally expected.

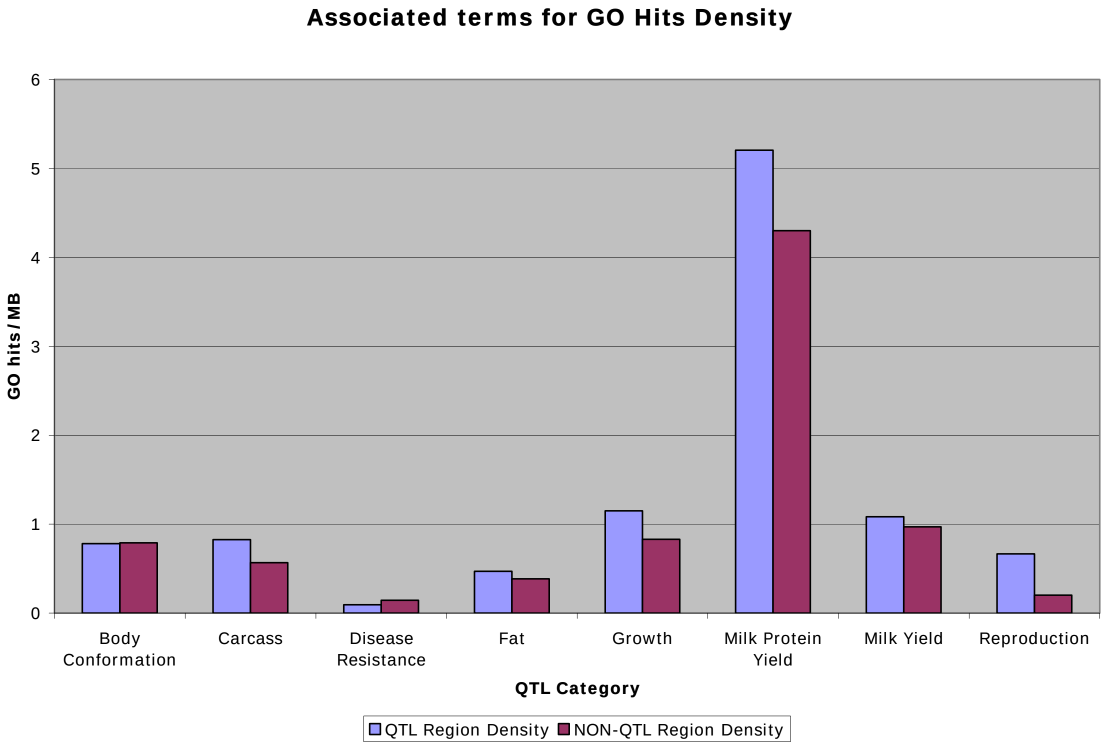


Figure 29 - Selected GO terms density compared with expected values.

3. Results

Although we can see that there are more GO than what would be normally expected based on our control group, a statistical measure of significance must be calculated in order to verify what is seen visually. In order to calculate significance, a t-test was performed on raw counts of gene ontology per 5Mb 'bin'. The t-test is a test of the null hypothesis that the means of two normally distributed populations are equal [63]. Specifically, a Welch's t-test was used which does not assume that the variances of the two populations are equal. The raw counts of gene ontology per bin were determined through a Perl script accessing a local GO database. GO per bin were calculated for all 8 QTL categories and were calculated across all second level GO (Table 5).

Table 5 - Raw counts sample for second level GO body conformation QTL category.

bin#	antioxidant activity_freq	auxiliary transport protein activity_freq	binding_freq	biological adhesion_freq
0	0	0	16	0
1	0	0	27	3
2	0	0	11	0
3	2	0	26	0
4	0	0	2	0
5	0	0	41	0
6	0	0	33	0
7	0	0	40	0
8	0	0	7	0
9	1	0	24	0
10	0	0	31	0

Associated terms within GO per QTL category were also validated through use of the Welch's t-test. Genes within each 5Mb bin for each QTL categories were isolated and ontologies were scanned for associated terms. Raw counts of associated terms were collected and stored for each bin. Non-QTL regions were analyzed in the same fashion and values were used as the control, expected group.

The R statistical package was used to generate t values across each term for each QTL category. After t values were generated, R was used to produce a p value, or probability value. The probability value provides a probability of obtaining a result at least as extreme as the given t value, under the null hypothesis [63]. This gives us a quantitative estimation of significance. In order to produce a normal distribution, raw count values were log transformed. However, a number of bins produce zero ontologies if: (a) there are no genes present or (b) the genes do not produce ontologies for specific second level GO. To overcome this obstacle, during transformation a value of 1 is added to avoid 0. As we are dealing with relative values (distribution) adding one does not change our eventual result.

In addition, the number of genes in a given bin can be a source of ascertainment bias. This is because bins with high numbers of genes, will more than likely produce high levels of GO. In order to negate this factor, the GO frequencies were 'normalized.' The number GO terms per bin was divided by the number of genes. This provided a ratio of raw count ontologies per gene within each bin. Taking these transformations into account, the formula became:

x = gene count, y = GO term count

$$\ln(y+1) / \ln(x+1)$$

After taking the transformation of the raw ontology counts per category per bin, values were plotted showing a distribution. Categories showed a normal distribution (Figure 30).

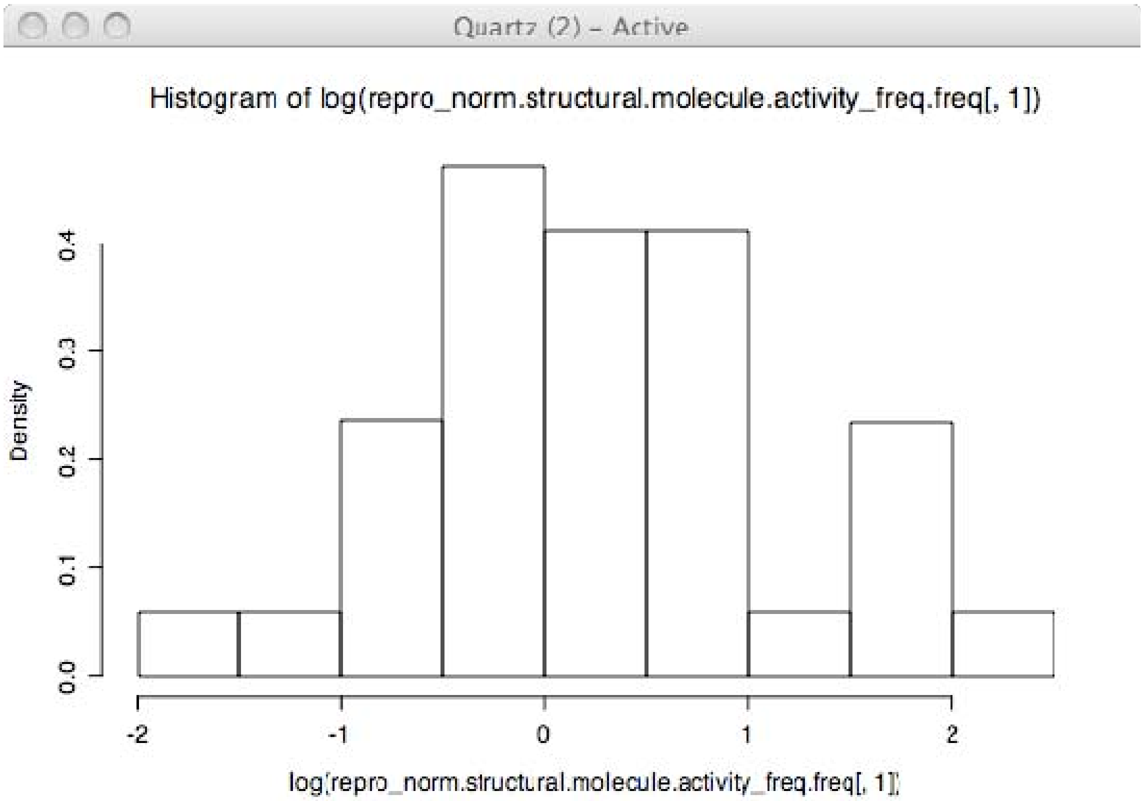


Figure 30 - Normal distribution of Reproduction QTL category for 'structural molecule activity' GO.

Three terms provided evidence of statistically significant over representation and were specific to the reproduction category of non-redundant QTL categories (Table 6).

Table 6 - T test analysis statistically significant second level GO terms.

QTL Category	GO Term	Description	T.value	p.value
Reproduction	GO:0005488	binding	4.145	8.77E-05
Reproduction	GO:0045202	synapse	-3.080	0.003208
Reproduction	GO:0044456	synapse part	-2.900	0.005332

Second level gene ontology distribution between QTL categories and the entire genome were further measured using another approach. Statistically significant over-representation of second level GO terms between terms in QTL categories compared to the term across the entire genome were identified using a hypergeometric distribution as the method of analysis [64, 65]. A hypergeometric distribution measures the number of successes in a sequence of n draws from a finite population without replacement. The sub population used was the genes found within each QTL category and the full population was all the genes together. In order to perform this analysis, the software GeneMerge was used to perform calculations [66]. GeneMerge generates rank scores for go term over representation through the hypergeometric distribution. Genes within each of the 8 QTL categories were grouped and GO term frequencies from those genes were compared against GO term frequencies found across the genome as a whole. By using the second level GO terms as the subpopulation, via GeneMerge we identified second level GO terms found to be statistically significantly over represented. Table 7 shows the terms found to be over represented.

Table 7 – Significantly over represented GO terms (second level).

QTL Category	GO Code	Description	Population Frequency	QTL / Population	p value
Body Conformation	GO:0044456	synapse part	0.004292225	4.001554261	0.000423999
Body Conformation	GO:0045202	synapse	0.008097703	3.063727944	0.000360381
Disease Resistance	GO:0022414	reproductive process	0.006637462	2.315301866	0.003037919
Fat	GO:0040007	growth	0.01039869	1.202900636	0.01968087
Growth	GO:0005623	cell	0.426877295	1.053967105	0.001435004
Growth	GO:0031974	membrane-enclosed lumen	0.025310854	1.24799951	0.006975395
Growth	GO:0044464	cell part	0.426833046	1.05407637	0.00140811
Growth	GO:0009987	cellular process	0.407761405	1.060107481	0.000706925
Growth	GO:0005215	transporter activity	0.051595203	1.185843492	0.003437039
Growth	GO:0030528	transcription regulator activity	0.049471216	1.15047095	0.016107834
Milk Protein	GO:0044421	extracellular region part	0.016328156	1.213751989	0.019582755
Milk Yield	GO:0005623	cell	0.426877295	1.045822768	0.004055757
Milk Yield	GO:0044464	cell part	0.426833046	1.045313026	0.004428211
Milk Yield	GO:0009987	cellular process	0.407761405	1.040496724	0.012333412
Milk Yield	GO:0030528	transcription regulator activity	0.049471216	1.162692048	0.007652378
Milk Yield	GO:0043234	protein complex	0.075799814	1.113890827	0.015965429
Reproduction	GO:0008152	metabolic process	0.272843931	1.056891152	0.017385493
Reproduction	GO:0005488	binding	0.38439754	1.066507532	0.000690466
Reproduction	GO:0009987	cellular process	0.407761405	1.042579697	0.015877113
Reproduction	GO:0030528	transcription regulator activity	0.049471216	1.244598127	0.000492496
Reproduction	GO:0043226	organelle	0.251028807	1.08464535	0.001499332
Reproduction	GO:0005198	structural molecule activity	0.025664852	1.265841051	0.005856023

Individual GO terms at their finest detail, were analyzed for over representation within QTL regions as well. Similar to the second level GO term analysis, the genes in the QTL category region formed the sub population, and the genes across the entire genome served as the complete population. Table 8 illustrates the lowest level GO terms that showed statistically significant over representation.

Table 8 – Lowest level GO terms that show significant over representation.

GO Term	Description	Population Frequen	QTL / Population	p-value
Body Conformation				
GO:0005254	chloride channel activity	0.001150493	8.2938225029	0.00028897629692
GO:0030594	neurotransmitter receptor activity	0.001592991	8.3859724417	0.00001612520678
GO:0048500	signal recognition particle	0.000398248	14.3759437328	0.00093809868877
GO:0004890	GABA-A receptor activity	0.001017744	11.2507483998	0.00001089633722
GO:0005230	extracellular ligand-gated ion channel activity	0.001592991	9.5839685049	0.00000135323341
GO:0007214	gamma-aminobutyric acid signaling pathway	0.001106244	8.6255697051	0.00023790868711
GO:0045202	synapse	0.007655206	2.9915280345	0.00074154901505
GO:0045211	postsynaptic membrane	0.003938227	4.3612449255	0.00022196426606
Carcass				
GO:0050806	positive regulation of synaptic transmission	0.000309748	3.3755064935	0.00019983448689
GO:0016600	flotillin complex	0.000309748	3.3755064935	0.00019983448689
GO:0051059	NF-kappaB binding	0.000884995	2.3628526384	0.00022384002856
GO:0006986	response to unfolded protein	0.002079738	1.8673001819	0.00020701089450
DiseaseResistance				
GO:0046785	microtubule polymerization	0.000353998	15.5042520132	0.00000532736753
GO:0031116	positive regulation of microtubule polymerization	0.000353998	15.5042520132	0.00000532736753
GO:0009925	basal plasma membrane	0.000486747	13.5309905249	0.00000163901602
GO:0001937	negative regulation of endothelial cell proliferation	0.000442497	12.4034156258	0.00002240295001
GO:0007154	cell communication	0.003893978	3.3827458938	0.00020993816802
GO:0042493	response to drug	0.001946989	4.5103278584	0.00032968160808
GO:0004620	phospholipase activity	0.000221249	19.8454201526	0.00001269879317
GO:0019900	kinase binding	0.000575247	9.5410740155	0.00010338646045
GO:0042470	melanosome	0.000486747	9.0206603499	0.00068970882494
GO:0043434	response to peptide hormone stimulus	0.000840745	6.5281080520	0.00076341224862
GO:0030659	cytoplasmic vesicle membrane	0.000353998	12.4034016106	0.00016124304105
GO:0016599	caveola	0.000530997	10.3361680088	0.00006580647426
GO:0019905	syntaxin binding	0.000575247	9.5410740155	0.00010338646045
GO:0009395	phospholipid catabolic process	0.000309748	14.1753275674	0.00008328520429
GO:0050998	nitric-oxide synthase binding	0.000353998	15.5042520132	0.00000532736753
GO:0030321	transepithelial chloride transport	0.000309748	14.1753275674	0.00008328520429
GO:0042311	vasodilation	0.000398248	11.0252389550	0.00028096772662
GO:0019861	flagellum	0.001106244	5.9536314276	0.00038686241813
GO:0007595	lactation	0.000840745	6.5281080520	0.00076341224862
GO:0030317	sperm motility	0.000840745	7.8337296624	0.00007291157047
Fat				
GO:0051059	NF-kappaB binding	0.000884995	2.3297935809	0.00010636504893
GO:0006986	response to unfolded protein	0.002079738	1.8506164002	0.00010062750212
Growth				
GO:0007586	digestion	0.001327492	2.7868249023	0.00026551223588
Milk Protein				
GO:0050785	advanced glycation end-product receptor activity	0.000265498	4.6653675256	0.00009674586385
GO:0051059	NF-kappaB binding	0.000884995	3.2657511177	0.00000435733577
GO:0006986	response to unfolded protein	0.002079738	2.2830481683	0.00002712605528
GO:0007584	response to nutrient	0.002168238	2.1898620130	0.00006250505759
Milk Yield				
GO:0005922	connexon complex	0.000796495	3.3126666551	0.00020523558057
GO:0030375	thyroid hormone receptor coactivator activity	0.000398248	4.6377274944	0.00009744009077
GO:0042809	vitamin D receptor binding	0.000707996	3.3540728828	0.00038242970444
GO:0042974	retinoic acid receptor binding	0.000309748	5.9628010486	0.00000371403303
GO:0004886	retinoid-X receptor activity	0.000442497	4.7702423789	0.00002028408385
Reproduction				
GO:0005882	intermediate filament	0.001858489	3.1631679667	0.00000137601439

Based on the results of the analysis a GO-slim was created. A GO-slim is a cut down version of the GO ontologies containing a subset of the terms in the whole GO [56]. They are of use for future research that includes giving summaries for microarray or cDNA collection when broad classification is required. Terms that were syntactically, and generally known to be associated with a particular QTL category and were found genomically in that region, were incorporated into the GO-slim.

4. Discussion

Studies in the human genome have shown that many genes with high levels of expression rates show apparent regional clustering [67]. But if genes are being clustered, a logical question becomes what are the functions of the clustered genes? Could clustered genes share a similar function? QTL are regions of the genome that have an associated function. We hypothesized that genes are functionally clustered and QTL regions can have an associated function. To test this hypothesis we used QTL as functional regions and compared frequencies of GO functional classes in those regions. We showed significant associations between GO terminology frequencies related to QTL category type within regions of the genome where the category of QTL spans non-redundantly.

Once we found statistical significance in GO frequencies within QTL regions, we wanted to have an understanding of the magnitude of effect for significance. We calculated the ratio of the GO density (per Mb) in QTL regions per GO term to the gene density in non-QTL regions per GO term and generated a heatmap to display the result (Figure 31). Categories that showed high significance in t-test analysis (i.e. metabolic_processXreproduction) show some strength in the ratio of GO density to non-QTL GO density. Figure 32 illustrates the ratio of third level GO term frequency in the

QTL category to the full genome [66]. Red indicates a higher frequency in the QTL regions, green indicates a higher frequency across the genome. White boxes represent GO terms that were not found in the QTL category regions. A few GO terms from specific categories have shown statistical significant over representation when compared to the full genome. The non-grayed out areas of the figure highlight these statistically significant GO terms.

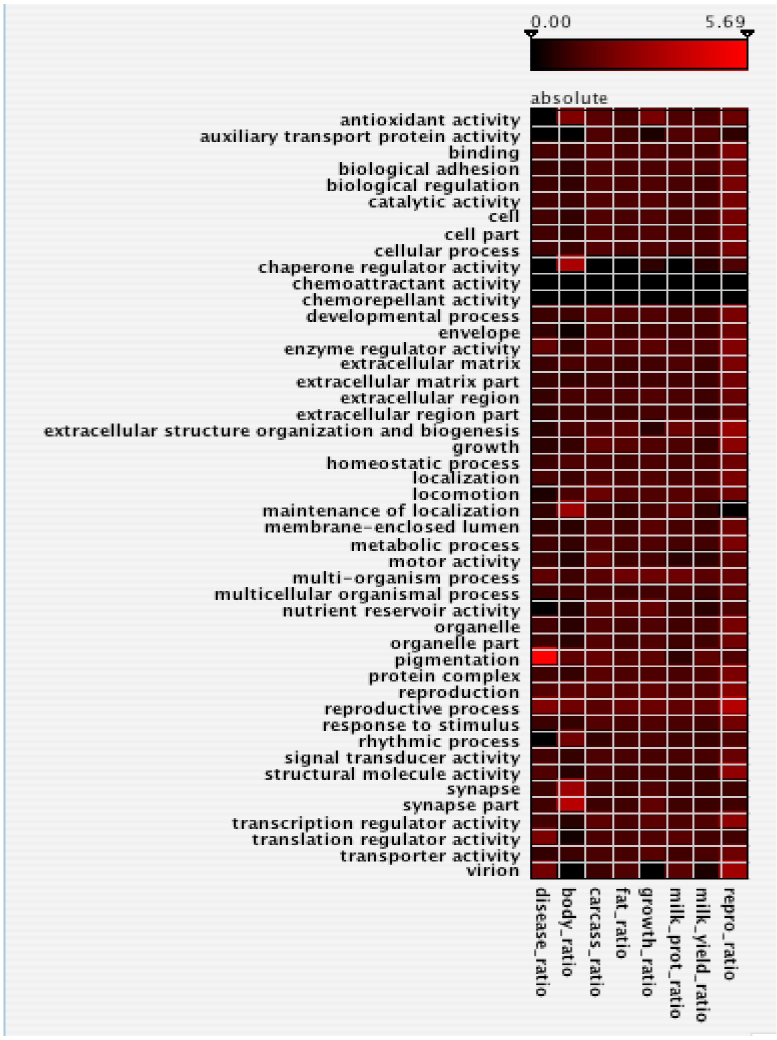


Figure 31 - GO per Mb ratio QTL to NON QTL heatmap.

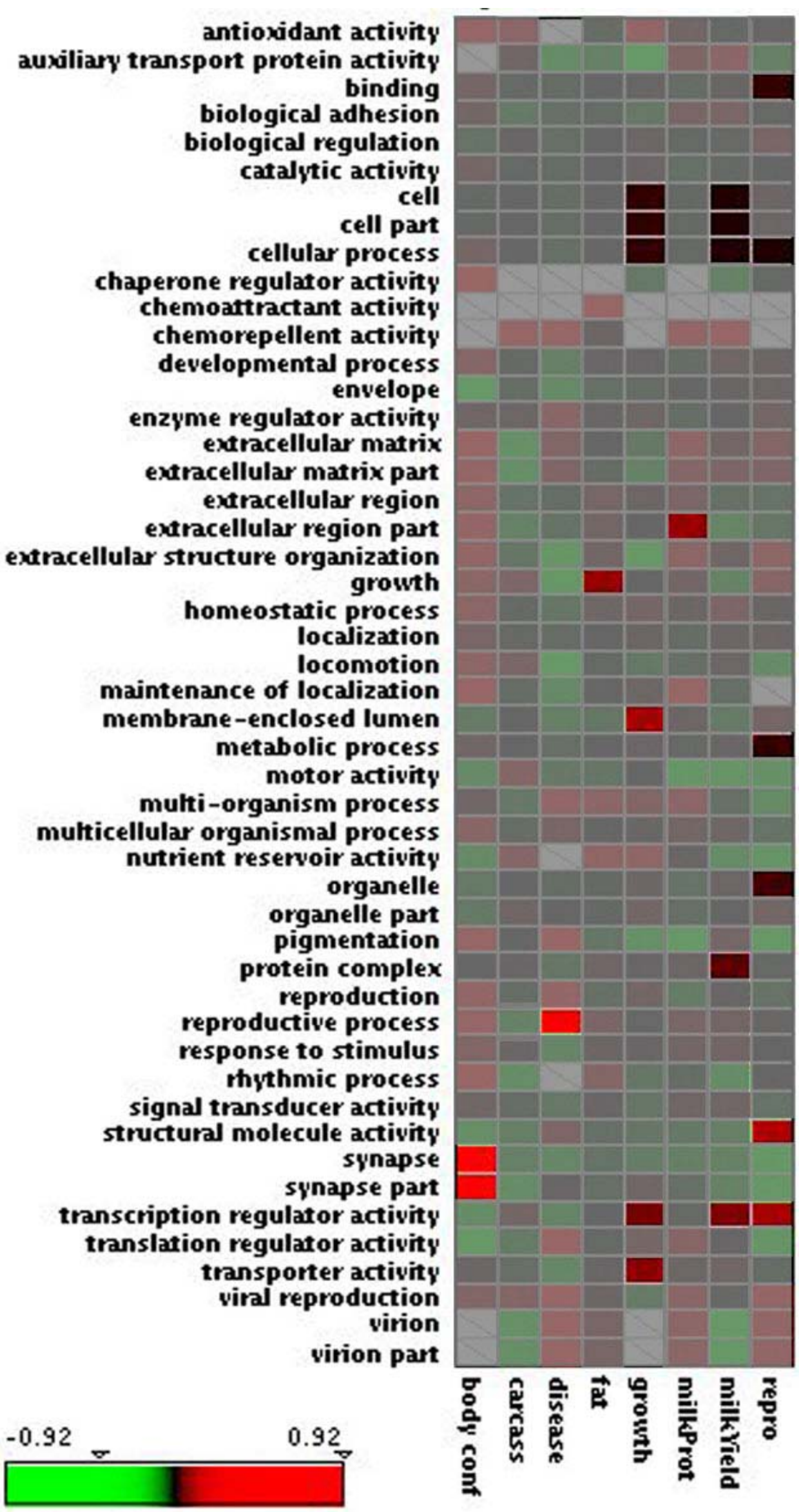


Figure 32 – Over represented GO term heatmap.
Ratio of GO term frequency in QTL region to entire genome with significant terms un-grayed.

Other aspects of functional gene clustering require further investigation. Gene rich regions have been shown to be associated with high GC DNA content [68, 69]. In the future, it would valuable to determine the GC content within the highly significant GO terms in associated QTL categories. Further, we can examine potential correlations between the GO terms and the frequency of CpG islands in genes in QTL regions and non-QTL regions.

In addition, a number of our selected GO terms associated with QTL categories showed significance in t-test analysis and hypergeometric analysis. It is possible that increasing the number of selected GO terms or re-selecting terms can identify more associations. However, this approach relies heavily on the selected terms.

Our analysis provides evidence of associations between GO terminology frequencies related to QTL category type. The future possibilities of using our GO based approach are numerous.

CHAPTER V

SUMMARY AND CONCLUSIONS

1. Summary

Many important agricultural traits such as weight gain, milk fat content and intramuscular fat (marbling) in cattle are quantitative traits. At the start of this project significant information regarding the mode of inheritance of quantitative traits were available however, most of this information was not integrated into a genomic context. The release of the bovine genome third assembly provided a large amount of genomic sequence and associated bovine annotation. The release of this data required that livestock genome researchers integrate sequence data not only with the existing gene maps, but more importantly with QTL and phenotypic data. As the genomic data became available, we adapted the Bovine QTL Viewer to begin to accommodate the community by storing and displaying this data. By adding the Gbrowse view we allowed users to view bovine QTL with respect to the underlying bovine genome third assembly.

The release of the bovine genome assembly also enhanced the value of locating and identifying SNPs across the genome. Adding the genome browser feature to the database provided researchers a visual tool to aid in the effort to understand SNP data in relation to their locations upon the chromosome as well as their locations among protein coding genes and QTL. In conjunction with SNPs, haplotypes have become extremely useful for displaying organization of variation between breeds of cattle. As such data has begun to accumulate and develop finer detail, a new database is necessary to seamlessly store multiple levels of bovine annotation into a single location. We have produced a database to meet these needs. By integrating the first generation Bovine QTL Viewer database and new bovine genotypic annotation into the Chado schema, we have created a flexible, extensible source of bovine genomic data with respect to bovine QTL.

The ability to visualize the data in the second generation Bovine QTL Viewer database can accelerate cattle genomics research and lead to improved enterprise productivity. The first generation Bovine QTL Viewer continually adapted to the changing research landscape. A gene search tool was implemented to facilitate identification of desired genes. Gene search streamlined the process of identifying genes associated with QTL regions, assisting research efficiency. In addition, a literature search tool was developed to assist the database administrator to update the database. Recent literature is scanned automatically for manuscripts presenting novel QTL. Identified potential papers are presented to the administrator via the web with full annotation. The administrator can chose to review or discard without any knowledge of action on the back end database. However, the recent tide of information for the bovine genome has necessitated a tool that can provide visualization assistance. Therefore, we have implemented a new approach towards creating and displaying data from the backend database. We have developed an object oriented viewer that displays bovine SNPs, STSs, haplotype blocks and QTL simultaneously. Finer genomic detail is achieved through the underlying fourth bovine genome assembly. The benefit of having a database with such bovine data is that analysis can be performed readily.

A global QTL analysis was performed to determine gene ontology frequency within QTL regions based on similar properties. QTL in the Bovine QTL Viewer were divided into 8 categories based on trait properties. Non redundant QTL regions were generated per category. Gene density was calculated across each QTL category and regions with no known or anchor-able QTL (non-QTL regions). Statistical analysis shows significant deviations in gene density across QTL regions compared to non-QTL regions. QTL regions were then binned into 5Mb chunks sequentially across QTL and non-QTL regions. Binning regions allowed data to be transformed and normalized. Subsequently a t-test was performed to analyze significance between second level ontology frequency between specific QTL categories and expected values generated from a control (non-QTL regions) group.

In addition to analyzing significance across second level ontology per QTL category, specific GO terms related to categories themselves were measured and analyzed for significance. The R statistical package was used to computationally analyze QTL regions with both approaches. T-test analysis showed significance across a number of terms. Both approaches: second level ontology scanning and selected related GO terms, produced significant values. Furthermore, numerous third level, and fine GO terms showed statistically significant over representation through hypergeometric analysis.

A full general summary of the presented research:

- Second Generation Bovine QTL Viewer Database
    - SNPs, haplotype blocks, QTL, STSs integrated into a single database.
    - Chado schema implemented
    - Model for QTL developed for Chado
- Evolution of a Bovine QTL Viewer
    - Added Gbrowse view for bovine QTL and other annotation
    - Gene search tool to facilitate functional sequence search
    - Literature search tool to facilitate QTL database update
    - Developed a completely restructured object oriented Bovine QTL Viewer
    - Evaluated use of viewer quantitatively
- Genome-wide QTL GO analysis
    - Analyzed gene density significance
    - Analyzed third level gene ontology term frequency significance based on grouped QTL regions through t-test analysis
    - Analyzed third level gene ontology and fine term GO frequency significance based on grouped QTL regions through hypergeometric analysis

- Analyzed selected GO terms related to QTL category frequency significance

REFERENCES

1. Barrett JC, Fry B, Maller J, Daly MJ: **Haploview: analysis and visualization of LD and haplotype maps**. *Bioinformatics (Oxford, England)* 2005, **21**(2):263-265.

2. Gabrieli JD, Brewer JB, Poldrack RA: **Images of medial temporal lobe functions in human learning and memory**. *Neurobiology of Learning and Memory* 1998, **70**(1-2):275-283.

3. Cervino AC, Li G, Edwards S, Zhu J, Laurie C, Tokiwa G, Lum PY, Wang S, Castellini LW, Lusis AJ *et al*: **Integrating QTL and high-density SNP analyses in mice to identify Insig2 as a susceptibility gene for plasma cholesterol levels**. *Genomics* 2005, **86**(5):505-517.

4. Blake JA, Eppig JT, Richardson JE, Davisson MT: **The Mouse Genome Database (MGD): expanding genetic and genomic resources for the laboratory mouse. The Mouse Genome Database Group**. *Nucleic Acids Research* 2000, **28**(1):108-111.

5. Cardon LR, Bell JI: **Association study designs for complex diseases**. *Nature Reviews Genetics* 2001, **2**(2):91-99.

6. Carlson CS, Eberle MA, Rieder MJ, Yi Q, Kruglyak L, Nickerson DA: **Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium**. *American Journal of Human Genetics* 2004, **74**(1):106-120.

7. Chakravarti A: **Single nucleotide polymorphisms:...to a future of genetic medicine.** *Nature* 2001, **409**:822-823.

8. Drysdale RA, Crosby MA: **FlyBase: genes and gene models**. *Nucleic Acids Research* 2005, **33**(Database issue):D390-395.

9. GMOD: **Chado - The GMOD Database Schema**. *Generic software components for model organism databases.* 2007.

10. Hawken RJ, Barris WC, McWilliam SM, Dalrymple BP: **An interactive bovine in silico SNP database (IBISS)**. *Mammalian Genome : Official Journal of the International Mammalian Genome Society* 2004, **15**(10):819-827.

11. Hu J, Mungall C, Law A, Papworth R, Nelson JP, Brown A, Simpson I, Leckie S, Burt DW, Hillyard AL *et al*: **The ARKdb: genome databases for farmed and other animals**. *Nucleic Acids Research* 2001, **29**(1):106-110.

12. Kelada Samir LGCA, Zahra a: **Dopamine transporter (SLC6A3Z) 5' region haplotypes significantly affect transcriptional activity in vitro but are not associated with Parkinson's Disease.** *Pharmacogenetics & Genomics* 2005, **15**(9):659-668.

13. Letovsky SI, Cottingham RW, Porter CJ, Li PW: **GDB: the Human Genome Database**. *Nucleic Acids Research* 1998, **26**(1):94-99.

14. Polineni P, Aragonda P, Xavier SR, Furuta R, Adelson DL: **The bovine QTL viewer: a web accessible database of bovine Quantitative Trait Loci**. *BMC Bioinformatics* 2006, **7**:283.

15. Twigger S, Lu J, Shimoyama M, Chen D, Pasko D, Long H, Ginster J, Chen CF, Nigam R, Kwitek A *et al*: **Rat Genome Database (RGD): mapping disease onto the genome**. *Nucleic Acids Research* 2002, **30**(1):125-128.

16. Weiss BP, S; Steinbach, D: **Bovmap Database**. *XXVth International Conference on Animal Genetics.* 21–25 July 1996, Tours, France. **S**uppl. 2, 59-59.

17. Zanotti M, Poli G, Ponti W, Polli M, Rocchi M, Bolzani E, Longeri M, Russo S, Lewin HA, van Eijk MJ: **Association of BoLA class II haplotypes with subclinical progression of bovine leukaemia virus infection in Holstein-Friesian cattle**. *Animal Genetics* 1996, **27**(5):337-341.

18. Petersen G, Johnson P, Andersson L, Klinga-Levan K, Gomez-Fabre PM, Stahl F: **RatMap--rat genome tools and data**. *Nucleic Acids Research* 2005, **33**(Database issue):D492-494.

19. Blake JA, Richardson JE, Davisson MT, Eppig JT: **The Mouse Genome Database (MGD). A comprehensive public resource of genetic, phenotypic and genomic data. The Mouse Genome Informatics Group**. *Nucleic Acids Research* 1997, **25**(1):85-91.

20. Blake JA, Eppig JT, Richardson JE, Davisson MT: **The Mouse Genome Database (MGD): expanding genetic and genomic resources for the laboratory mouse. The Mouse Genome Database Group**. *Nucleic Acids Research* 2000, **28**(1):108-111.

21. Ouzounis CA, Karp PD: **The past, present and future of genome-wide re-annotation**. *Genome Biology* 2002, **3**(2):COMMENT2001.

22. Bennett ST: **A Promoter Polymorphism in the Insulin Gene and Type I Diabetes**. *Nature Genetics* 1995, **9**:284.

23. Parra EJ, Kittles RA, Argyropoulos G, Pfaff CL, Hiester K, Bonilla C, Sylvester N, Parrish-Gause D, Garvey WT, Jin L *et al*: **Ancestral proportions and admixture dynamics in geographically defined African Americans living in South Carolina**. *American Journal of Physical Anthropology* 2001, **114**(1):18-29.

24. Gopalakrishnan S, Qin ZS: **TagSNP selection based on pairwise LD criteria and power analysis in association studies**. *Pacific Symposium on Biocomputing* 2006:511-522.

25. Cardon LR, Bell JI: **Association study designs for complex diseases**. *Nature Reviews* 2001, **2**(2):91-99.

26. **Chado - Getting Started** [http://www.gmod.org/wiki/index.php/Schema]. 01/08.

27. Mungall CJ, Emmert DB: **A Chado case study: an ontology-based modular schema for representing genome-associated biological information**. *Bioinformatics* 2007, **23**(13):i337-346.

28. Gibbs RA, Weinstock GM, Metzker ML, Muzny DM, Sodergren EJ, Scherer S, Scott G, Steffen D, Worley KC, Burch PE *et al*: **Genome sequence of the Brown Norway rat yields insights into mammalian evolution**. *Nature* 2004, **428**(6982):493-521.

29. **Generic Feature Format Version 3** [http://song.sourceforge.net/gff3.shtml]. 01/08.

30. Schuler GD: **Sequence mapping by electronic PCR**. *Genome Research* 1997, **7**(5):541-550.

31. Schuler GD: **Electronic PCR: bridging the gap between genome mapping and genome sequencing**. *Trends in Biotechnology* 1998, **16**(11):456-459.

32. Eilbeck K, Lewis SE, Mungall CJ, Yandell M, Stein L, Durbin R, Ashburner M: **The Sequence Ontology: a tool for the unification of genome annotations**. *Genome Biology* 2005, **6**(5):R44.

33. Worsley, JC: *Practical PostgreSQL*. O'Reilly Press, Sebastopol, California; 2002.

34. **Chado Schema Documentation** [http://www.gmod.org/wiki/index.php/Chado_Manual]; 2007.

35. Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, Higgins J, DeFelice M, Lochner A, Faggart M *et al*: **The structure of haplotype blocks in the human genome**. *Science* 2002, **296**(5576):2225-2229.

36. Kuniyoshi Y, Inaba M, Inoue H: **Learning by watching: extracting reusable task knowledge from visual observation of human performance**. *IEEE Transactions on Robotics and Automation* **10**:799-822; 1994

37. Stein LD, Mungall C, Shu S, Caudy M, Mangone M, Day A, Nickerson E, Stajich JE, Harris TW, Arva A *et al*: **The generic genome browser: a building block for a model organism system database**. *Genome research* 2002, **12**(10):1599-1610.

38. Hartl DLJ, Elizabeth W: *Genetics Analysis of Genes and Genomes, vol. 6.* Jones and Bartlett, Sudbury, U.K.; 2005.

39. Wei E: **GOOF - Generic Object Oriented Framework**. Texas A&M University, College Station; 2008.

40. **Gbrowse - Generic Genome Browser**. [http://gmod.cvs.sourceforge.net/*checkout*/gmod/Generic-Genome-Browser/docs/tutorial/tutorial.html?pathrev=stable.] 2007.

41. Snelling WM, Chiu R, Schein JE, Hobbs M, Abbey CA, Adelson DL, Aerts J, Bennett GL, Bosdet IE, Boussaha M *et al*: **A physical map of the bovine genome**. *Genome Biology* 2007, **8**(8):R165.

42. **Map Set Info "composite map IBMT March 2006"**. [http://cmap.medvet.angis.org.au/cgi-bin/cmap/map_set_info?map_set_aid=142] 2006.

43. **Generic Model Organism Database Construction Set - pubfetch**. [http://gmod.sourceforge.net/pubfetch.shtml]. 2007.

44. Cornell G, Horstmann, CS: *Core Java, Vol. 2.* Sun Microsystems Press, Palo Alto; 2001.

45. Chisholm RL, Gaudet P, Just EM, Pilcher KE, Fey P, Merchant SN, Kibbe WA: **dictyBase, the model organism database for Dictyostelium discoideum**. *Nucleic Acids Research* 2006, **34**(Database issue):D423-427.

46. Drysdale RA, Crosby MA: **FlyBase: genes and gene models**. *Nucleic Acids Research* 2005, **33**(Database issue):D390-395.

47. **Large Scale Bovine Snp Genotyping for Genomic Selection and Hapmap Development** [http://www.ars.usda.gov/research/projects/projects.htm?accn_no=410680]. 11/07.

48. Twigger S, Lu J, Shimoyama M, Chen D, Pasko D, Long H, Ginster J, Chen CF, Nigam R, Kwitek A *et al*: **Rat Genome Database (RGD): mapping disease onto the genome**. *Nucleic Acids Research* 2002, **30**(1):125-128.

49. Chen N, Harris TW, Antoshechkin I, Bastiani C, Bieri T, Blasiar D, Bradnam K, Canaran P, Chan J, Chen CK *et al*: **WormBase: a comprehensive data resource for Caenorhabditis biology and genomics**. *Nucleic Acids Research* 2005, **33**(Database issue):D383-389.

50. Harris TW, Chen N, Cunningham F, Tello-Ruiz M, Antoshechkin I, Bastiani C, Bieri T, Blasiar D, Bradnam K, Chan J *et al*: **WormBase: a multi-species resource for nematode biology and genomics**. *Nucleic Acids Research* 2004, **32**(Database issue):D411-417.

51. Zakian VA: **Structure and function of telomeres**. *Annual Review of Genetics* 1989, **23**:579-604.

52. Sullivan BA, Blower MD, Karpen GH: **Determining centromere identity: cyclical stories and forking paths**. *Nature Reviews* 2001, **2**(8):584-596.

53. Pennisi E: **Genomics. DNA study forces rethink of what it means to be a gene**. *Science* 2007, **316**(5831):1556-1557.

54. Hedrick PW: *Genetics of Populations, 3 edn.* Jones and Bartlett, Sudbury, U.K.; 2005.

55. Khatkar MST, P.C.; Tammen, I; Costa, F; Raadsma, H.W.: **Quantitative Trait Loci Mapping in Dairy Cattle: Review and Meta-Analysis**. *Genetics Selection Evolution* 2004, **36**(2):163-190.

56. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT *et al*: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium**. *Nature Genetics* 2000, **25**(1):25-29.

57. Robinson PN, Bohme U, Lopez R, Mundlos S, Nurnberg P: **Gene-Ontology analysis reveals association of tissue-specific 5' CpG-island genes with development and embryogenesis**. *Human Molecular Genetics* 2004, **13**(17):1969-1978.

58. Camon E, Magrane M, Barrell D, Lee V, Dimmer E, Maslen J, Binns D, Harte N, Lopez R, Apweiler R: **The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology**. *Nucleic Acids Research* 2004, **32**(Database issue):D262-266.

59. **Bioinformatics toolbox 3.1** [http://www.mathworks.com/products/bioinfo/demos.html?file=/products/demos/shipping/bioinfo/geneontologydemo.html]. 01/08.

60. Jameson JL: *Principles of Molecular Medicine, vol. 1*, Humana Press, Boston, Massachusetts; 1998.

61. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool**. *Journal of Molecular Biology* 1990, **215**(3):403-410.

62. Tao Y, Kupfer R, Stewart BJ, Williams-Skipp C, Crowell CK, Patel DD, Sain S, Scheinman RI: **AIRE recruits multiple transcriptional components to specific genomic regions through tethering to nuclear matrix**. *Molecular Immunology* 2006, **43**(4):335-345.

63. Walpole, RE, Myers, Sharon L., Ye, Keying: *Probability & Statistics, vol. 7,* Prentice Hall Press, Upper Saddle River, New Jersey; 2002.

64. Draghici S, Khatri P, Martins RP, Ostermeier GC, Krawetz SA: **Global functional profiling of gene expression**. *Genomics* 2003, **81**(2):98-104.

65. Boyle EI, Weng S, Gollub J, Jin H, Botstein D, Cherry JM, Sherlock G: **GO::TermFinder--open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes**. *Bioinformatics* 2004, **20**(18):3710-3715.

66. Castillo-Davis CI, Hartl DL: **GeneMerge--post-genomic analysis, data mining, and hypothesis testing**. *Bioinformatics* 2003, **19**(7):891-892.

67. Lercher MJ, Urrutia AO, Hurst LD: **Clustering of housekeeping genes provides a unified model of gene order in the human genome**. *Nature Genetics* 2002, **31**(2):180-183.

68. Zoubak S, Clay O, Bernardi G: **The gene distribution of the human genome**. *Gene* 1996, **174**(1):95-102.

69. Zerial M, Salinas J, Filipski J, Bernardi G: **Gene distribution and nucleotide sequence organization in the human genome**. *European Journal of Biochemistry / FEBS* 1986, **160**(3):479-485.

APPENDIX A

- Appendix A contains supplementary data for the second generation QTL Viewer database.

<u>Perl script to create data file for STS:</u>

```
#!/usr/bin/perl
use strict;
use DBI;
my $data_file = '../markers/Btau20070913.marker.map';
open(DAT, $data_file) || die("Could not open file!");
#CONNECTING TO THE DB for marker aliases
my $dbh = DBI->connect('DBI:mysql:usda_sts', '…', '…');
while(my $line=<DAT>)
{
   chomp $line;
   my @markers_array = split(/\t/, $line);

   #ASSIGNING VALUES TO QTL, MARKER START/END FOR QTL IN DB
   my $full_name = $markers_array[65];

   my @full_name_array = split(/\|/, $full_name);

   if($full_name_array[1] eq 'MARC')
   {
          #print "$line\n";
          #GETTING array of marker aliases for marker 1
          my $query = "select usda_name from marc where usda_id='$full_name_array[0]'";
          #print "$query\n";
          my $sth = $dbh->prepare($query);
          my $rv = $sth->execute;
          my @row_ary  = $sth->fetchrow_array;
          #my $name = $sth->fetchrow_array;
          #print "$row_ary[0]\n";
          if($row_ary[0] eq '')
          {
             #print "$line\n";
             print "NO: $full_name_array[0]  in DB\n";
          }
          #print "$markers_array[0] $markers_array[1]          $markers_array[2]
          $row_ary[0]|MARC|$full_name_array[2]|$full_name_array[3] $markers_array[4]\n";
   }
   else
   {
          #print "$line\n";
   }
}
#!/usr/bin/perl
use strict;
use DBI;

my $data_file="../qtl_indb_Jan4.txt";
open(DAT, $data_file) || die("Could not open file!");
#CONNECTING TO THE NCBI_STS DB
my $dbh = DBI->connect('DBI:mysql:bovineqtl_jan8_07', '....', '....');
while(my $line=<DAT>)
{
   chomp $line;
   my @qtl_indb_array = split(/\t/, $line);
   #print "$qtl_indb_array[0]-$qtl_indb_array[1]-$qtl_indb_array[2]-\n";
   #GETTING SEQUENCE DATA FOR MARKER1 OF QTL
   my $query ="select chromosome, trait_id, marker_start, marker_end, marker_peakstart,
marker_peakend, lod_score, f_statistic, family, method, locus_type, suggestive, significant, gene, status,
history from qtl_info where qtl_id=$qtl_indb_array[0]";
   my $sth = $dbh->prepare($query);
   my $rv = $sth->execute;
   my @row_ary  = $sth->fetchrow_array;
   print "$line      $row_ary[0]       $row_ary[1]       $row_ary[2]       $row_ary[3]       $row_ary[4]
          $row_ary[5]       $row_ary[6]       $row_ary[7]       $row_ary[8]       $row_ary[9]
          $row_ary[10]      $row_ary[11]      $row_ary[12]      $row_ary[13]      $row_ary[14]
          $row_ary[15]\n";
}
```

<u>Perl script to create QTL file:</u>

```
#!/usr/bin/perl
use strict;
use DBI;
my $data_file="../data_qtl_withBp.txt";
open(DAT, $data_file) || die("Could not open file!");
while(my $line=<DAT>)
{
   chomp $line;
   my @qtl_indb_array = split(/\t/, $line);
   #print "$qtl_indb_array[0]          $qtl_indb_array[1]          $qtl_indb_array[2]
          $qtl_indb_array[3]          $qtl_indb_array[4]\n";

   #sorting out the family issues (spaces) and storing into variable
   my $family=$qtl_indb_array[12];
   $family=~s/ /%20/g;
   $family=~s/,/%2C/g;

   #sorting out the method issues (spaces) and storing into variable
   my $method=$qtl_indb_array[13];
   $method=~s/ /%20/g;
   $method=~s/,/%2C/g;

   #sorting trait
   my $trait=$qtl_indb_array[3];
   $trait =~ s/ /%20/g;
   $trait=~s/,/%2C/g;
   my $m_start=$qtl_indb_array[1];
   my $m_end=$qtl_indb_array[2];
   my $lod_score=$qtl_indb_array[10];
   my $f_stat=$qtl_indb_array[11];
   my $trait_id=$qtl_indb_array[5];
   my $gene='';
   if($qtl_indb_array[17] ne '')
   {
          $gene=$qtl_indb_array[17];
          $gene=~s/ /%20/g;
          $gene=~s/,/%2C/g;
   }
   if($gene ne '' && $qtl_indb_array[4]==1){
          print "BTA$qtl_indb_array[4]      TAMU  QTL     $qtl_indb_array[20]
          $qtl_indb_array[21]        .         +         .
          ID=$qtl_indb_array[0];trait=$trait;trait_id=$trait_id;marker_start=$m_start;marker_end=$m_end
;family=$family;method=$method;suggestive=$qtl_indb_array[15];significant=$qtl_indb_array[16];lod_s
core=$lod_score;f_statistic=$f_stat;gene=$gene\n";
   }
   elsif($qtl_indb_array[4]==1){
          print "BTA$qtl_indb_array[4]      TAMU  QTL     $qtl_indb_array[20]
          $qtl_indb_array[21]        .         +         .
          ID=$qtl_indb_array[0];trait=$trait;trait_id=$trait_id;marker_start=$m_start;marker_end=$m_end
;family=$family;method=$method;suggestive=$qtl_indb_array[15];significant=$qtl_indb_array[16];lod_s
core=$lod_score;f_statistic=$f_stat\n";
   }
}
```

<u>More QTL code:</u>

```
#!/usr/bin/perl
use strict;
use DBI;

my $total_count=0;
my $count=0;

my $data_file="qtl_indb_Jan4.txt";
my $trait="";

open(DAT, $data_file) || die("Could not open file!");

while(my $line=<DAT>)
{
   chomp $line;
   my @qtl_indb_array = split(/\t/, $line);

   #ASSIGNING VALUES TO QTL, MARKER START/END FOR QTL IN DB
   my $qtl_id=$qtl_indb_array[0];
   my $marker_start =  $qtl_indb_array[1];
   my $marker_end =  $qtl_indb_array[2];
   my $trait = $qtl_indb_array[3];

##########################################################################################
#########
   #CONNECTING TO THE DB for marker aliases
   my $dbh = DBI->connect('DBI:mysql:marker_alias', '....', '....');

   #GETTING array of marker aliases for marker 1
   my $query = "select * from names where marker_name='$marker_start' or alias1='$marker_start' or
alias2='$marker_start' or alias3='$marker_start' or alias4='$marker_start' or alias5='$marker_start';";
   my $sth = $dbh->prepare($query);
   my $rv = $sth->execute;
   my @row_ary;
   my $i=0;
   my @alias_ary;
   while(@row_ary  = $sth->fetchrow_array)
   {
          for(my $z=0; $z<@row_ary; $z++)
          {
             $alias_ary[$i] = $row_ary[$z];
             $i++;
             #print "$alias_ary[$i-1]-";
          }

   }

   my $marker1_alias_string="name='$marker_start'";
   #CREATE A STRING TO SEARCH ALL ALIASES IN THE ALIAS ARRAY
   for(my $y=0; $y<@alias_ary; $y++)
   {
          if($alias_ary[$y] ne '')
          {
             $marker1_alias_string = "$marker1_alias_string or name='$alias_ary[$y]'";
          }
   }
   #print "$marker1_alias_string\n";
#####################################################################################
#########

          #GETTING array of marker aliases for marker 2
      my $query = "select * from names where marker_name='$marker_end' or alias1='$marker_end' or
alias2='$marker_end' or alias3='$marker_end' or alias4='$marker_end' or alias5='$marker_end';";
   my $sth = $dbh->prepare($query);
   my $rv = $sth->execute;
   my @row_ary2;
   my $a=0;
   my @alias_ary2;
   while(@row_ary2  = $sth->fetchrow_array)
   {
      for(my $z=0; $z<@row_ary2; $z++)
      {
         $alias_ary2[$a] = $row_ary2[$z];
         $a++;
         #print "$alias_ary2[$a-1]:$row_ary2[$z]-";
      }

   }
```

```
    my $marker2_alias_string="name='$marker_end'";
    #CREATE A STRING TO SEARCH ALL ALIASES IN THE ALIAS ARRAY
    for(my $y=0; $y<@alias_ary2; $y++)
    {
            #print "$alias_ary2[$y]-";
       if($alias_ary2[$y] ne '')
       {
          $marker2_alias_string = "$marker1_alias_string or name='$alias_ary2[$y]'";
            }
    }
    #print "$marker1_alias_string\n";

##############################################################################
#########



    #CONNECTING TO THE NCBI_STS DB
    my $dbh = DBI->connect('DBI:mysql:btau4_markers', '....', '....');

    #GETTING SEQUENCE DATA FOR MARKER1 OF QTL
    my $query = "select chromosome, start_bp from markers where $marker1_alias_string;";
    #print "$query\n";
    my $sth = $dbh->prepare($query);
    my $rv = $sth->execute;
    my @row_ary  = $sth->fetchrow_array;

    my $marker1_chr_start = $row_ary[1];
    my $marker1_chrom = $row_ary[0];

    #print "$marker_start----$marker1_chrom-----$marker1_chr_start\n";


    #GETTING SEQUENCE DATA FOR MARKER2 OF QTL
    #my $query2 = "select chromosome, stop_bp from markers where name='$marker_end';";
    my $query2 = "select chromosome, stop_bp from markers where $marker2_alias_string;";
    #print "$query2\n";
    my $sth2 = $dbh->prepare($query2);
    my $rv2 = $sth2->execute;
    my @row_ary2  = $sth2->fetchrow_array;

    my $marker2_chr_stop = $row_ary2[1];
    my $marker2_chrom = $row_ary2[0];
    #print "$marker_end-----$marker2_chrom---$marker2_chr_stop\n";

    if($marker1_chrom eq $marker2_chrom && $marker2_chr_stop > $marker1_chr_start)
    {
            #print "$marker_start----$marker1_chrom-----$marker1_chr_start\n";
            #print "$marker_end-----$marker2_chrom---$marker2_chr_stop\n";
            print "$marker1_chrom     BOSMAP          QTL       $marker1_chr_start
            $marker2_chr_stop           .           +           .           QTL $qtl_id ; Note
\"$marker_start:$marker_end, $trait\"\n";
   }

}
Use strict;
close DAT;
```

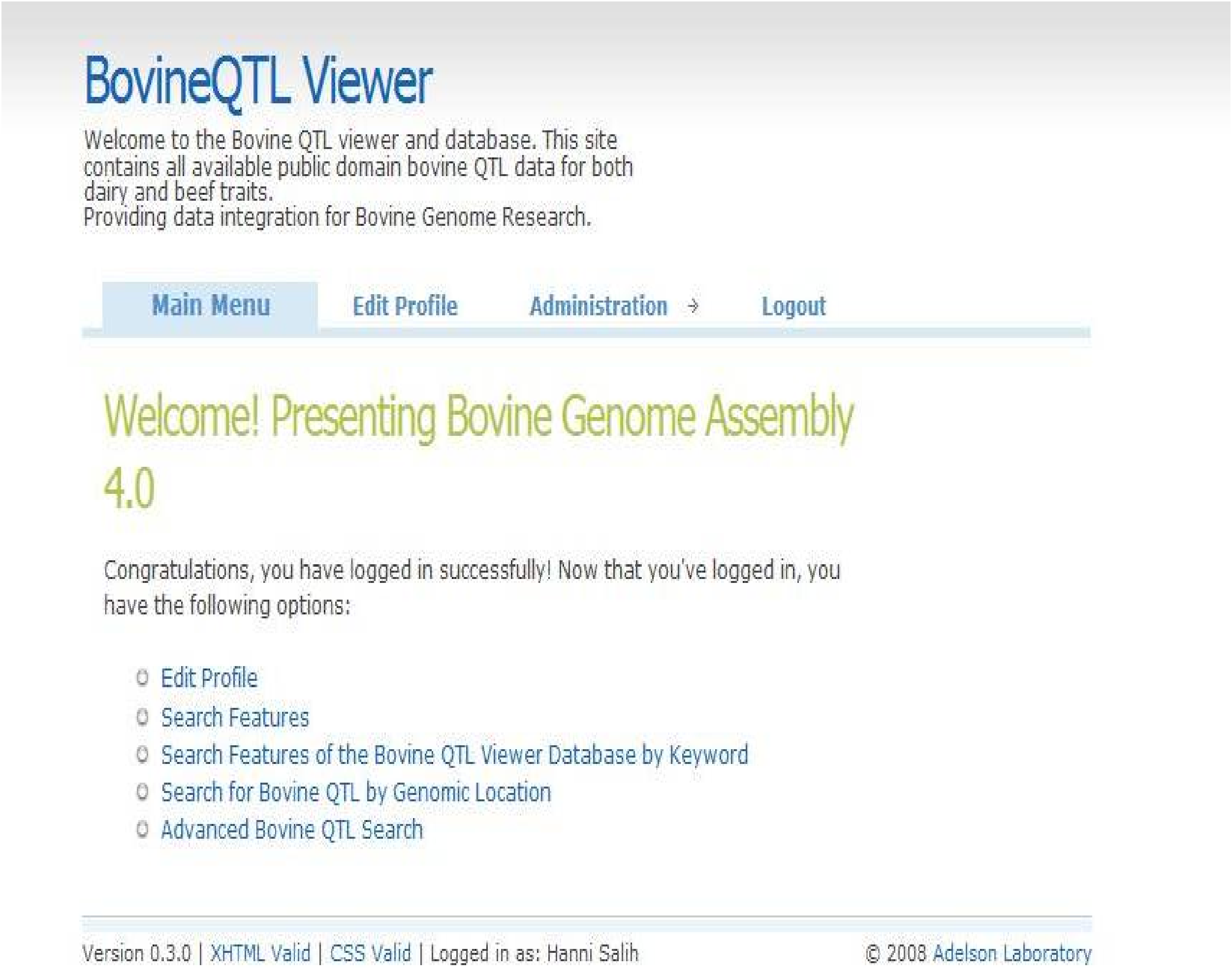
Perl script to generate SNP data:

```
#!/usr/bin/perl
use strict;
use DBI;

#File I am reading for SNPs
my $in_file = "Btau20070913.Assembly.snp.uniq";
open(DAT, $in_file) || die("Could not open file!");

#File I am writing to
open (MYFILE, '>>snps_assembly_uniq_post6.gff3');

while(my $line = <DAT>)
{
   chomp $line;

   my @tab_array = split(/\t/, $line);
            my $chrom;
            if($tab_array[0] =~ m/Chr(.+)/)
            {
                        $chrom=$1;
            }

   if( ($chrom>=7 && $chrom<10) || $chrom eq 'X'){
   my @allele = curlIT($tab_array[2]);
   print MYFILE "BTA$chrom       BCMHGSC         SNP       $tab_array[1]       $tab_array[1]       .
            $tab_array[3]       .
            ID=SNP($tab_array[2]);Name=$tab_array[2];allele=$allele[0];ncbi_id=$allele[1];source=Assem
bly;breed=hereford_assembly\n";}

}

##########################################################################
sub curlIT {
   my $subSnp = shift;
   my $cmd = "curl \"http://www.ncbi.nlm.nih.gov/sites/entrez?db=snp&cmd=search&term=$subSnp\" -o
\"./temp_curled_file.txt\" ";
   system $cmd;
   my $allele;
   my $ncbi_id;
   open (DAT2, "temp_curled_file.txt") || die("Couldnt do it");
   while (my $temp_line = <DAT2>)
   {
            if($temp_line =~ m/red">\[(\S)\/(\S)\]/)
```

```
          {
             #print "hello:$subSnp-$1\n";
             $allele = "$1%2F$2";
          }
          if($temp_line=~ m/">rs(\d+)<\/a><em> /)
          {
             #print "WHOO:$1\n";
             $ncbi_id="rs$1";
          }
   }
   my @allele_rs;
   $allele_rs[0] = $allele;
   $allele_rs[1] = $ncbi_id;
          return @allele_rs;
}
########################################################################

close DAT;
close DAT2;
close MYFILE;
```

APPENDIX B

- Appendix B contains supplementary data for the Bovine QTL Viewer.



BovineQTL Viewer

Welcome to the Bovine QTL viewer and database. This site contains all available public domain bovine QTL data for both dairy and beef traits.
Providing data integration for Bovine Genome Research.

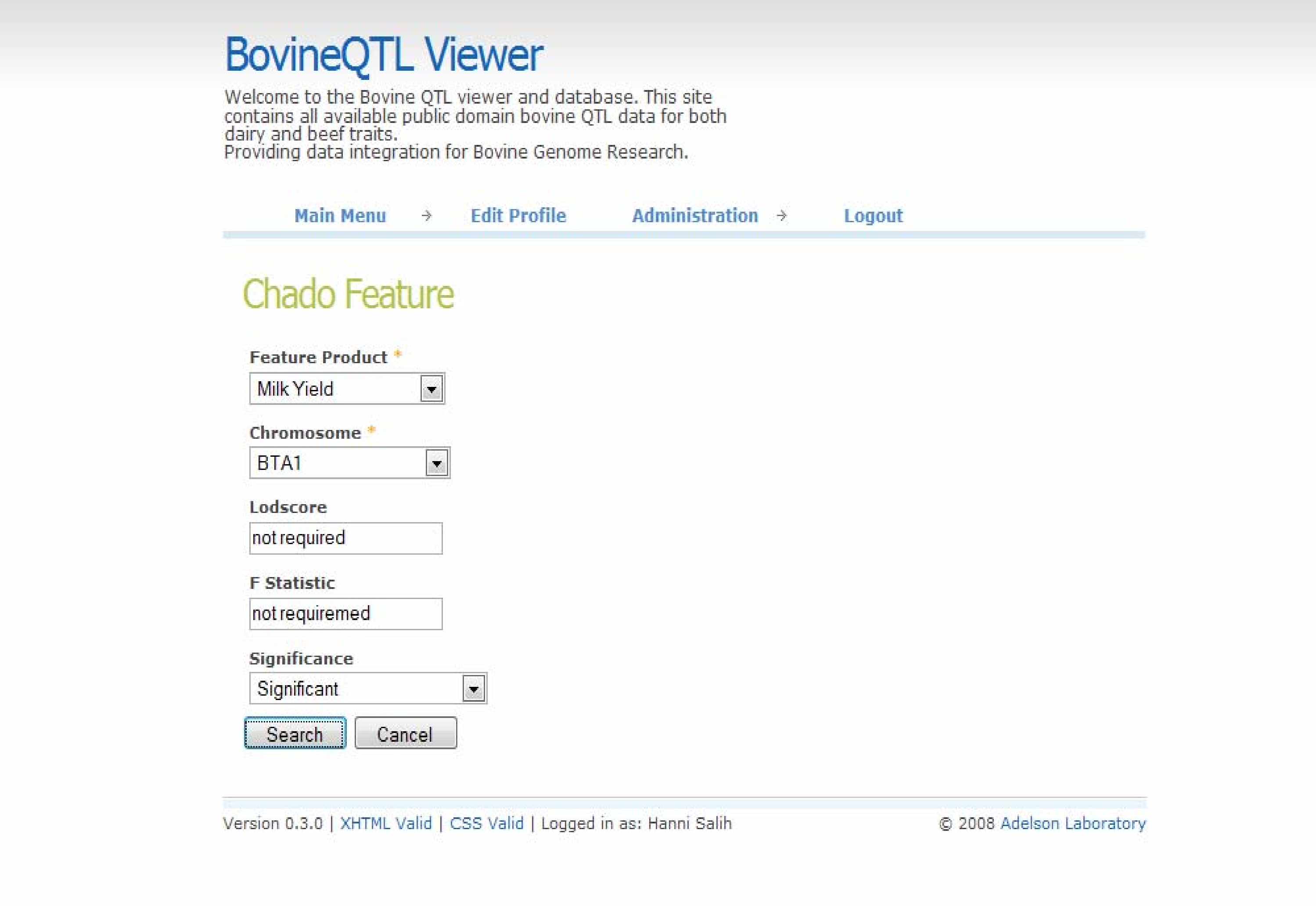
Welcome! Presenting Bovine Genome Assembly 4.0

Congratulations, you have logged in successfully! Now that you've logged in, you have the following options:

- Edit Profile
- Search Features
- Search Features of the Bovine QTL Viewer Database by Keyword
- Search for Bovine QTL by Genomic Location
- Advanced Bovine QTL Search

Version 0.3.0 | XHTML Valid | CSS Valid | Logged in as: Hanni Salih © 2008 Adelson Laboratory

Figure 33 – Main menu for the Bovine QTL Viewer.



BovineQTL Viewer

Welcome to the Bovine QTL viewer and database. This site contains all available public domain bovine QTL data for both dairy and beef traits.
Providing data integration for Bovine Genome Research.

Chado Feature

Feature Product *: Milk Yield

Chromosome *: BTA1

Lodscore: not required

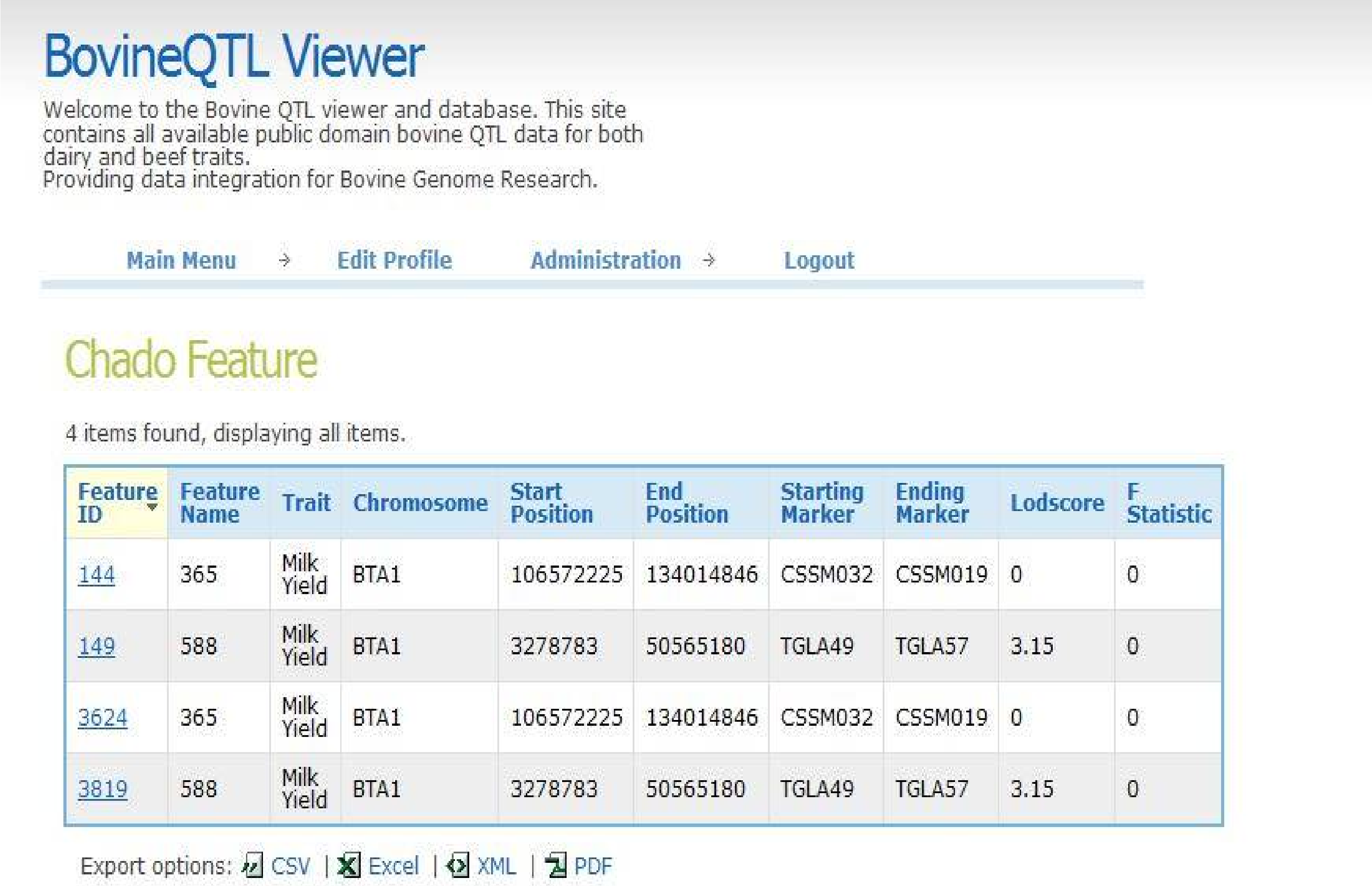
F Statistic: not requiremed

Significance: Significant

Version 0.3.0 | XHTML Valid | CSS Valid | Logged in as: Hanni Salih

© 2008 Adelson Laboratory

Figure 34 – Advance QTL search page.



BovineQTL Viewer

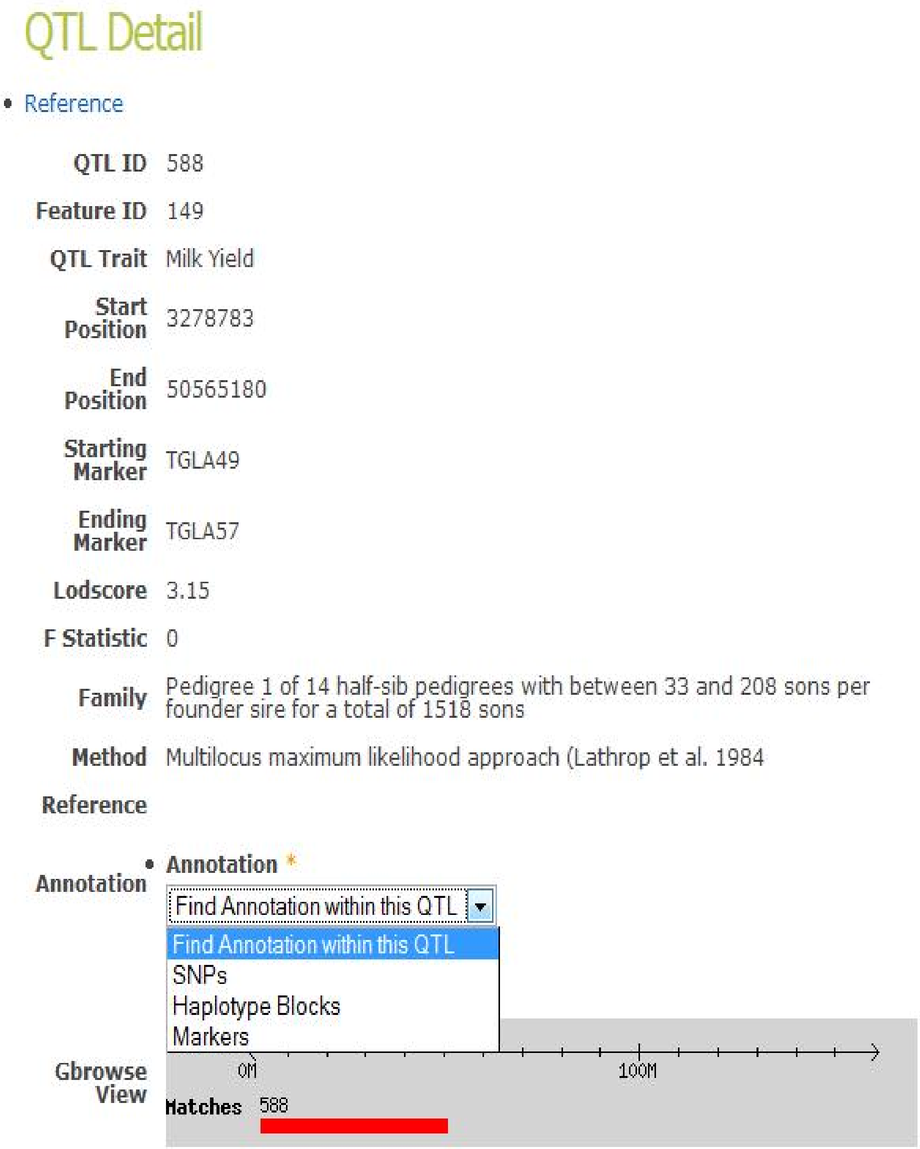
Welcome to the Bovine QTL viewer and database. This site contains all available public domain bovine QTL data for both dairy and beef traits.
Providing data integration for Bovine Genome Research.

Chado Feature

4 items found, displaying all items.

Feature ID	Feature Name	Trait	Chromosome	Start Position	End Position	Starting Marker	Ending Marker	Lodscore	F Statistic
144	365	Milk Yield	BTA1	106572225	134014846	CSSM032	CSSM019	0	0
149	588	Milk Yield	BTA1	3278783	50565180	TGLA49	TGLA57	3.15	0
3624	365	Milk Yield	BTA1	106572225	134014846	CSSM032	CSSM019	0	0
3819	588	Milk Yield	BTA1	3278783	50565180	TGLA49	TGLA57	3.15	0

Figure 35 - QTL list page.

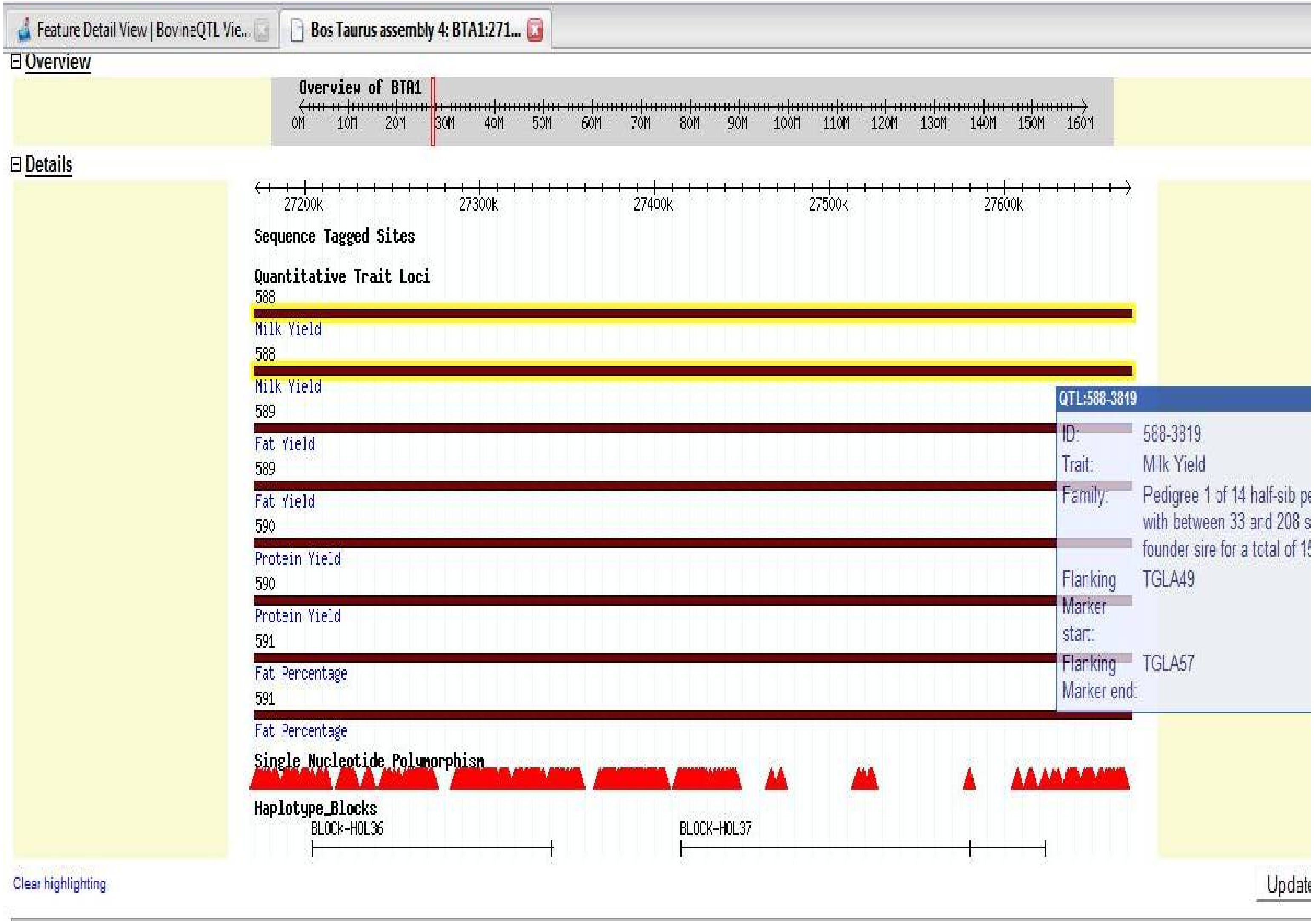


QTL Detail

- Reference

QTL ID	588
Feature ID	149
QTL Trait	Milk Yield
Start Position	3278783
End Position	50565180
Starting Marker	TGLA49
Ending Marker	TGLA57
Lodscore	3.15
F Statistic	0
Family	Pedigree 1 of 14 half-sib pedigrees with between 33 and 208 sons per founder sire for a total of 1518 sons
Method	Multilocus maximum likelihood approach (Lathrop et al. 1984
Reference	
Annotation	• Annotation * Find Annotation within this QTL
Gbrowse View	

Figure 36 – QTL detail view.



QTL:588-3819

ID: 588-3819

Trait: Milk Yield

Family: Pedigree 1 of 14 half-sib p with between 33 and 208 s founder sire for a total of 1

Flanking Marker start: TGLA49

Flanking Marker end: TGLA57

Figure 37 - Gbrowse view of second generation Bovine QTL Viewer.

Bovine QTL Viewer Pre-development Survey

Bovine QTL Viewer Survey

Site address: http://Bovineqtl.tamu.edu/

1. Have you used the Bovine QTL Viewer before?

    What **DO** you like about the Bovine QTL Viewer?
    What **Don't** you like about the Bovine QTL Viewer?

2. Based on the data you normally deal with, what types of queries would you use on a genome browser?

3. Do you prefer to see multiple data types (i.e. SNPs, QTL, haplotypes) at the same time?
    What data types would you like have available at the same time?

4. If applicable, do you normally look for QTL by:
    Trait?
    Manuscript?
    Chromosome?
    Location?
    Overlaps?
    All of the Above?

5. What other genome browsers have you used recently?

    Is there anything you liked about it?

    Is there anything you did not like about it?

6. What is your favorite search engine (i.e. google, yahoo, msn)? Why?

7. Have you ever used the entrez search box?

    Is there anything you liked about it?
    Is there anything you did not like about it?

8. Have you ever used Gbrowse (generic genome browser) before?

Is there anything you liked about it?
Is there anything you did not like about it?

Bovine QTL Viewer Post-development Survey

Bovine QTL Viewer Post Development Survey
http://Bovineqtl.tamu.edu/

1. How would you rate the overall functionality of the second generation viewer?

Low High
1………2..........3....…..4......…5

2. Do you feel that the novel data (i.e. SNPs, Haplotypes) of the new Bovine QTL Viewer is easily accessible?

Low High
1………2..........3....…..4......…5

3. Do you feel that the data from question 2 is useful for you?

Low High
1………2..........3....…..4......…5

4. Have your QTL searches improved in efficiency from the first generation to second generation viewer?

Low High
1………2..........3....…..4......…5

5. When using the “search the database” function, do you find your feature of interest?

Low High
1………2..........3....…..4......…5

6. How would you rate your overall experience with the second generation Bovine QTL Viewer?

Low High
1………2..........3....…..4......…5

7. Do you have any additional modifications you would like to see made to the viewer?

8. Do you have any additional comments?

```
[hanni@equinegenome bovineqtl]$ more gene_search.php
<html>
<head>
<title>Bovine Gene Search</title>
</head>
<? ?>
<body>
<br><br><br>
                <h1 align=center><font color=#A0D0D6><i><b><?php print "Bovine" ?> QTL
viewer</b></i></font></h1>
        <body bgcolor=#ffffff>
                <br><br><br><br><br>
                <form name=Gene_Search onsubmit="Search Bovine Genes" action=gene_search2.php
method=get>
                        <table align=center border=1 bgcolor=#B0E0E6>
                                <tr>
                                                <td colspan=2><font face=arial size=3><strong>Gene
Search</strong></font>
                                                </td>
                                </tr>
                                <tr>
                                         <td>Gene Description:</td>
                                         <td><input type=text maxlength=55 name=gene_descriptor size="70"></td>
                                </tr>
                                <tr>
                                                <td>Search Method:</td>
                                                <td>Exact Match<input type="radio" maxlength=20 value="Exact"
name="method">
                                                Match at least 1 word<input type="radio" maxlength=10 value="one"
name="method">
                                                Match all words<input type="radio" maxlength=20 value="all"
name="method"></td></th>
                                </tr>
                                <tr align=center>
                                         <td colspan=2><input type=submit value=Search></td>
                                </tr>
                         <table align=center ><tr><td align=center >
                         Search will return all genes that match the keywords entered in the textbox.
   <br>
  Searches Genes in BVGA3
                         </td></tr>
```

```
                    </table>
                </form>
        </body>

</html>
```

```
[hanni@equinegenome bovineqtl]$ more gene_search2.php
<html>
        <head>
                <META HTTP-EQUIV="Pragma" CONTENT="no-cache">
                <title>
                        <?php print "Bovine" ?> QTL Viewer @ <?php print $HOST_NAME?>
                </title>
        <body>
                <h1 align=center><font color=#A0D0D6><i><b><?php print "Bovine" ?> QTL
viewer</b></i></font></h1>
<?php
//**************************************************
//get the value from the user's input
$gene_descriptor=$_GET['gene_descriptor'];
$method=$_GET['method'];
//echo "$method";
        if($gene_descriptor == "")
        {
                echo "<td><font color=#FF0000 size=5>You must enter a description of a gene.</font></td>";
                exit;
        }
//**************************************************
?>
<Table align=center border=1 bgcolor=#B0E0E6>

        <TR>
                <TH COLSPAN=1>Chromosome</TH>
                <TH COLSPAN=1>Gene Description</TH>
        </TR>
<?php
//********************************************************************************
*****************************************************
//get the value from the user's input
$gene_descriptor=$_GET['gene_descriptor'];
$method=$_GET['method'];
if($method == "Exact")
{
        //variable holding the query to the bov6x_ver4 database
        //$str_sql="select fref, fattribute_value, fstart, fstop from fdata, fattribute_to_feature where
fattribute_to_feature.fid=fdata.fid AND fdata.ftypeid=6 AND
 fattribute_to_feature.fattribute_value LIKE '%$gene_descriptor%';";

       $str_sql="select d.fref, d.fstart, d.fstop, a.fattribute_value, g.gname from fdata as d, fgroup as g,
fattribute_to_feature as a where d.gid=g.gid AND d.fid=
a.fid AND a.fattribute_value LIKE '%$gene_descriptor%'";
}
elseif($method == "one")
{
        $gene_length=strlen($gene_descriptor);
        //echo $gene_length;
        $words=split(' ', $gene_descriptor);
        //echo $words[1];
        //echo count($words);
        $or_string = "'%$words[0]%'";
for($counter=1; $counter<count($words); $counter++)
{
  //$or_string = "$or_string OR fattribute_to_feature.fattribute_value LIKE '%$words[$counter]%'";
  $or_string = "$or_string OR a.fattribute_value LIKE '%$words[$counter]%'";
}
        //variable holding the query to the bov6x_ver4 database
        //$str_sql="select fref, fattribute_value, fstart, fstop from fdata, fattribute_to_feature where
fattribute_to_feature.fid=fdata.fid AND fdata.ftypeid=6 AND
 (fattribute_to_feature.fattribute_value LIKE $or_string);";
        $str_sql="select d.fref, d.fstart, d.fstop, a.fattribute_value, g.gname from fdata as d, fgroup as g,
fattribute_to_feature as a where d.gid=g.gid AND d.fid
=a.fid AND a.fattribute_value LIKE $or_string";
//echo $str_sql;
//echo $or_string;
//'%$gene_descriptor%';";
}
elseif($method == "all")
{
        $gene_length=strlen($gene_descriptor);
        $words=split(' ', $gene_descriptor);
        $and_string = "'%$words[0]%'";
        for($counter=1; $counter<count($words); $counter++)
        {
          //$and_string = "$and_string AND fattribute_to_feature.fattribute_value LIKE
'%$words[$counter]%'";
          $and_string = "$and_string AND a.fattribute_value LIKE '%$words[$counter]%'";
        }
        //variable holding ...
        //$str_sql="select fref, fattribute_value, fstart, fstop from fdata, fattribute_to_feature where
fattribute_to_feature.fid=fdata.fid AND fdata.ftypeid=6 AND
 (fattribute_to_feature.fattribute_value LIKE $and_string)";
        $str_sql="select d.fref, d.fstart, d.fstop, a.fattribute_value, g.gname from fdata as d, fgroup as g,
fattribute_to_feature as a where d.gid=g.gid AND d.fid
=a.fid AND a.fattribute_value LIKE $and_string";
//echo $str_sql;
}
elseif($method == "")
{
        echo "You must select an option.";
}
else
{
        echo "we couldnt get a match $";
}
```

```
//dbconnect values
$hostname="localhost";
$username="hanni";
$password="";
//$database="bov6x_gbw_v1";
//$database="bov7x_gbw_qtl";
$database="btau_3";
//connect to the database
$dbh=mysql_connect($hostname,$username,$password) or die("Failed to connect to mysql");
mysql_select_db($database, $dbh);
//stores the results
$result=mysql_query($str_sql,$dbh);
    if (!$result) {
            print "$str_sql  Could not run query: " . mysql_error();
            exit;
    }
    else{
        if(mysql_num_rows($result))
        {
            //loop through the array of results and print to the screen.
            while($row = mysql_fetch_row($result))
            {
                echo "<TR>";
                //echo "<td>$row[0]</td><td>$row[1]</td>";
                //echo "<td><a href=\"http://equinegenome.tamu.edu/cgi-
bin/gbrowse/bov6x_ver4?ref=$row[0];start=$row[2];stop=$row[3];nav4=1;plugin=\
">$row[0]</a><td>$row[3]</td>";
                //echo "<td><a href=\"http://genomes.tamu.edu/cgi-
bin/gbrowse/bovine31/?name=protein_id:$row[4]\">$row[0]</a><td>$row[3]</td>";
                echo "<td><a href=\"http://genomes.tamu.edu/cgi-
bin/gbrowse/bovine31/?name=id:$row[4]\">$row[0]</a><td>$row[3]</td>";
                echo "</TR>";
            }
        }
        else { echo 'No records exist.'; }
/*****************************************************************************************
*******/
    ?>
</Table>
    </head>
    <frameset cols="11%,*" border=0 >
    <body>
    </body>
    </html>
<?
}
?>
```

QTL_IMAGE.php

```
/////****************************************************************************************
*************************************************************
                                        //Code to link this individual qtl to gbrowse.  Added by Hanni Salih on 12/16/05.
```

```
                        $query_chromo="select chromosome from qtl_info where qtl_id=$qtlid;";
                        $chromo=mysql_query($query_chromo,$db);
                        $chromo=mysql_fetch_row($chromo);
                        $temp_chromo=0+$chromo[0];
                        $chromo="Chr$temp_chromo";
                        $hostname="localhost";
                        $username="hanni";
                        $password="";
                        $dbh=mysql_connect($hostname,$username,$password) or die("Failed to
connect to mysql");
                        mysql_select_db("bov7x_gbw_qtl", $dbh);
                        $str_qtl="select * from fgroup where gclass='QTL' AND gname='$qtlid';";
                        $result_temp_qtl=mysql_query($str_qtl,$dbh);
                        $result_temp_qtl=mysql_fetch_row($result_temp_qtl);

                        print"<td bgcolor=#B0E0E6><b>Gbrowse</b></td>\n";

                        if($result_temp_qtl[2])
                        {
                          #print "<td><a href=\"http://$SERVER_NAME/cgi-
bin/gbrowse/bov6x_scaff_ver2/?name=QTL%3A$qtlid\">View</a></td>\n";
                          print "<td><a href=\"$GB_LINK?name=QTL%3A$qtlid\">Assembly 3.1
View</a></td>\n";
                        }
                        else
                        {
                          #print"<td><a href=\"warning.php?chrom=$chromo\">View</a></td>\n";
                          print"<td><font color=#C0C0C0><b>N/A</b></font></td>\n";
                        }
                        print"</tr><tr>";
                        mysql_close($dbh);
                        //end of the code for the link
/////****************************************************************************************
****************************************************************
/////****************************************************************************************
****************************************************************
                        //Code to link to the composite map   Edited by Hanni Salih: 12/17/06

                        //print"<td bgcolor=#B0E0E6><b>Composite</b></td>\n";
                        $dbh=mysql_connect($hostname,$username,$password) or die("Failed to
connect to mysql");
                        mysql_select_db("composite_map", $dbh);
                        $str_qtl="select * from fgroup where gclass='QTL' AND gname='$qtlid';";
                        $result_temp_qtl=mysql_query($str_qtl,$dbh);
                        $result_temp_qtl=mysql_fetch_row($result_temp_qtl);
                        print"<td bgcolor=#B0E0E6><b>Composite</b></td>\n";
                        if($result_temp_qtl[2])
                        {
                          //print "<td><a href=\"http://$SERVER_NAME/cgi-
bin/gbrowse/snelling/?name=QTL%3A$qtlid\">Composite View</a></t \
                          //d>\n";
                          //Code for the composite map, Edited by Hanni Salih:4/4/07
```

```
                             print "<td><a
href=\"$CMAP_LINK?name=$global_trait_id($qtlid)\">Composite Map View</a></td>\n";
                            }
                            else
                            {
                             print"<td><font color=#C0C0C0><b>N/A</b></font></td>\n";
                            }
                            print "<tr></tr>";
                            mysql_close($dbh);
/////***********************************************************************************
************************************************************
                                                  print"<td bgcolor=#B0E0E6><b>References</b></td>\n";
                                                  print"<td><a href=\"qtl_references.php?qtlid=$qtlid\">List of
references</a></td>\n";
                                                  print"</tr><tr>";
/*////////////////////////////////////////////////////////commented//////////
                                            print"<td bgcolor=#B0E0E6><b>Markers ------
Position</b></td>\n";
                                                                                        ?>
                                            </tr>
                                            <form name=markerdetails action=markerdetails.php
method="POST">
                                            <tr>
                                                                  <td ><select name=marker size=8>
<?
$result_marker_info=mysql_query($str_marker_info_sql,$db);
                                                                  for ($i=0;$i<$rows_marker_info;$i++)
                                                                  {
$fields_marker_info=mysql_fetch_row($result_marker_info);
                                                                     print "<option
value=$fields_marker_info[1]>$fields_marker_info[
2]------$fields_marker_info[0]\n";
                                                                  }
                                                                  ?>
                                                                  </select>
                                                                  </td>
                                                                  <td>
                                                                        <input type='submit' name='submit'
value="View the details o
f the selected marker">
                                                                  </td>
                                                                  <td>
                                            </form>
/////////////////////////////////////////////////////////commented/////////*/
```

Warning.php

```
[hanni@equinegenome bovineqtl]$ more warning.php
<?php

session_start();
//$userid=$_SESSION["s_userid"];
//$password=$_SESSION["s_password"]
```

```
//if($userid=="" && $password=="") $password=$_REQUEST["password"];
//if($userid=="") $userid=$_REQUEST["userid"];
//include("config.php");
?>
<html>
<head>
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<?php
//*************************************************************************************
******************************
//get the chromosome from the passed variable
$chrom=$_GET['chrom'];
//*************************************************************************************
*****************************
?>
</head>
<frameset cols="99%,*" border=0 >
<frame name="warning2frame" src="warning2.php?chrom=<?php print "$chrom"?>">
</frameset>
</html>
```

Warning2.php

```
<html>
<?php
//*************************************************************************************
******************************
//get the chromosome from the passed variable
$chrom=$_GET['chrom'];
//print "hey its:$chrom";
//*************************************************************************************
*****************************
?>
<font color="#810541">
***Unfortunately, due to data restrictions at this time, this qtl cannot be anchored to the bovine 6x
sequence.***
<br>Click "continue" to view the entire chromosome of which this qtl falls upon.</br>
</font>

<br>
<td><a href=http://equinegenome.tamu.edu/cgi-bin/gbrowse/bov6x_scaff_ver2/?name=<?php print
"$chrom"?> >Continue</a></td>
</br>
</html>
```

Gbrowse Conf File for Assembly 3.1

```
[GENERAL]
description   = Bovine Genome Chromosomes (Assembly 3.1) **TEST**

#----- for database connection----
db_adaptor    = Bio::DB::GFF
db_args       = -adaptor dbi::mysql
                -dsn bovine31
user          = nobody
```

```
pass        =
aggregators = match
        coding
        coding2{CDS,mRNA/mRNA}
        transcript2{exon,transcript/transcript}
      processed_transcript
      alignment
plugins = FastaDumper SequenceDumper GFFDumper
#RestrictionAnnotator BatchDumper
#OligoFinder Aligner
# list of tracks to turn on by default
#default features = BovineEstHits PigEstHits
#reference class  = Sequence
# examples to show in the introduction
examples = Chr28 ChrX
# "automatic" classes to try when an unqualified identifier is given
#automatic classes = Note sequence
automatic classes = Note sequence mRNA GenePrediction gene_id transcript_id protein_id ID
#instructions = "Click on list of all DNA IDs and then click on any ID to see the protein hits. Clicking on
any bar in image gives details of the BLAST hit."
### HTML TO INSERT AT VARIOUS STRATEGIC LOCATIONS ###
# inside the <head></head> section
head =
# at the top...
header = <TABLE border="0" cellpadding="0" cellspacing="0" height="100"
width="100%"><TR><TD><img border="0" src="http://genomes.tamu.edu/bovine/images/BovineBanner
Left100.png" width="250" height="100" align="left"></TD><TD><center><IMG
src="http://genomes.tamu.edu/bovine/images/BovineGenomeBannerTitleshorter.png" align="cente
r"></center></TD><TD><IMG
src="http://genomes.tamu.edu/bovine/images/BovineBannerRight100.png" align="right" height="100"
width="250"></TD></TR></TABLE>
# a footer
footer = <FONT face="Verdana" size="2"><P align="center">The Bovine Genome Database is supported
by USDA NRI, the Kleberg Foundation and the Texas Agricultural Expe
riment Station.</P></FONT><TABLE border="2" bordercolor="#800000" bordercolordark="#800000"
bordercolorlight="#800000" cellspacing="0" width="100%" cellpadding="5"
background="http://racerx00.tamu.edu/bovine/images/brown_grass.jpg"><TR><TD><FONT
face="Verdana" size="2">The Bovine Genome Database is hosted by the <A href="http:
//racerx00.tamu.edu/" target="_blank"><FONT face="Verdana">Animal Bioinformatics and
Computational Genomics Lab</A> at <A href="http://www.tamu.edu/" target="_blank
">Texas A&amp;M University</A>. If you have comments or if you wish to report a problem, please
contact the <A href="mailto:cmdickens@neo.tamu.edu"><FONT face="Verd
ana" size="2">Bovine Genome Database Administrator</A>. Photos courtesy of USDA NRSC and Mike
MacNeil, USDA/ARS Fort Keogh LARRL.</FONT></TD></TR></TABLE>
# Various places where you can insert your own HTML -- see configuration docs
#between the "examples" and "landmark or region" areas
html1 =
#between "landmark or region" and overview areas
html2 =
#between viewer and "data source" areas
html3 =
#between the "data source" and "tracks" areas
```

```
html4 = <p align="center">Click on a track option below to view methods and citations for track
data.</p>
#between the "tracks" and "image width" areas
html5 =
#between "image width" and upload your own annotations" area
html6 =
# what image widths to offer
image widths  = 450 640 800 1024
# default width of detailed view (pixels)
default width = 800
# Web site configuration info
stylesheet  = /gbrowse/gbrowse.css
buttons     = /gbrowse/images/buttons
tmpimages   = /gbrowse/tmp
# max and default segment sizes for detailed view
# mysql querry gives max dna_length as 2278051
max segment     = 150000000
default segment = 50000
# zoom levels
zoom levels = 1000 2000 5000 10000 20000 40000 50000 100000 500000 1000000 10000000 100000000
150000000
# colors of the overview, detailed map and key
overview bgcolor = lightgrey
detailed bgcolor = goldenrodyellow
key bgcolor      = lightorange
########################
# Plugin configuration
########################
#[Aligner:plugin]
#alignable_tracks   = EST other_insect_cox
#upcase_tracks      = CDS Motifs
#upcase_default     = CDS
########################
# Default glyph settings
########################
[TRACK DEFAULTS]
glyph         = generic
height        = 10
bgcolor       = lightgrey
fgcolor       = black
font2color    = blue
label density = 1000
bump density  = 1000
# where to link to when user clicks in detailed view
#link         =
link_target   = _blank
title         = $name ref:$ref - Start:$start - End:$end
################## TRACK CONFIGURATION ####################
# the remainder of the sections configure individual tracks
###########################################################
[SNP]
feature       = SNP
glyph         = triangle
```

```
fgcolor      = red
bgcolor       = red
point        = 1
orient       = N
key          = SNP, Baylor HGSC
description   = 1
category     = SNP, Marker, QTL
#link         = /db/gene/locus?name=$name;class=Locus
citation      = Single Nucleotide Polymorphisms provided by Baylor College of Medicine Human Genome
Sequencing Center.
[STS]
feature      = STS
glyph        = generic
stranded     = 1
bgcolor      = orange
height       = 10
description  = 1
key          = STS, NCBI UniSTS
category     = SNP, Marker, QTL
link         = http://www.ncbi.nlm.nih.gov/genome/sts/sts.cgi?uid=$name
citation = UniSTS for Bos taurus aligned to assembly by <a
href="http://www.ncbi.nlm.nih.gov/mapview/map_search.cgi?taxid=9913">NCBI</a>.
[QTL]
feature      = QTL
glyph        = generic
stranded     = 1
bgcolor      = green
height       = 10
description  = 1
link         = http://bovineqtlv2.tamu.edu/qtl_image.php?qtlid=$name
key          = QTL
category     = SNP, Marker, QTL
#Added by:Hanni Link for QTL track modified by Hanni on 2/2/07 as a temporary fix for php issues on
genomes.
[DNA]
glyph          = dna
global feature = 1
height         = 40
do_gc          = 1
fgcolor        = black
axis_color     = blue
strand         = both
key            = GC Content of DNA
link_target    = _blank
citation       = Percent GC content of scaffold DNA is displayed.
```

APPENDIX C

- Appendix C contains supplementary data for the genome-wide meta-analysis.

Table 9- Gene density table.

QTL Category	Number of Genes	QTL Category size	Density per MB
Body Conformation	307	94.713704	3.24134721
Carcass	3820	705.80352	5.41227111
Disease Resistance	517	115.290292	4.48433247
Fat	4133	888.087371	4.65382139
Growth	2051	389.091838	5.2712491
Milk Protein Yield	2719	601.929388	4.51714114
Milk Yield	2193	488.872241	4.48583457
Reproduction	1871	232.541508	8.04587541
NON-QTL Regions	3543	983.843941	3.60118089
ALL-QTL Regions	9166	1703.257559	5.38145271
Full Genome	12763	2687.101631	4.74972731

Script pipeline for meta-analysis:

```
#!/usr/bin/perl
use strict;
use DBI;

#Open the file from the input line
my $file = "$ARGV[0]";
chomp ($file);
open (DAT, $file) || die("Could not open file");

my %go_hash=();

#loop through each line of the file
while(my $line=<DAT>)
{
   #print "$line\n";
   chomp $line;
   my @line_ary=split(/\t/, $line);
   #print "$line_ary[1]\n";
   my @go_array=split(/--/, $line_ary[1]);
   #print "$go_array[0]\n";
   for(my $i=0; $i<@go_array; $i++)
   {
          if(exists ($go_hash{$go_array[$i]}) )
      {
         my $value = $go_hash{$go_array[$i]} +$line_ary[2];
         $go_hash{$go_array[$i]} = $value;
      }
      else
      {
         $go_hash{$go_array[$i]} = $line_ary[2];
      }
   }

}

#print out 2nd level go terms from the hash
my $k;
my $v;
while ( ($k,$v) = each %go_hash ) {
   $k =~ s/_/ /g;
   print "$k          $v\n";
}


close DAT;
#!/usr/bin/perl
use strict;
use DBI;

my $file = "$ARGV[0]";
chomp ($file);


open (DAT, $file) || die("Could not open file");
my @lines = <DAT>;

#CONNECTING TO THE NCBI_STS DB
my $dbh = DBI->connect('DBI:mysql:gene_ontology', 'hanni', 'hanni');

my $string="";
my $gene_counter=0;

for(my $z=0; $z<@lines; $z++)
{
   chomp $lines[$z];
   my @line_ary = split (/\t/, $lines[$z]);
   #print "$line_ary[0]-$line_ary[1]-$line_ary[2]\n";
   #For qtl regions
   #my $query = "select go_term, bovine_gene_id from go_withLocations where
bovine_chromosome=$line_ary[0] AND (bov_bp_start>=$line_ary[1] AND
bov_bp_end<=$line_ary[2])";
   #For non qtl regions
   #my $query = "select go_term, bovine_gene_id from go_withLocations where
bovine_chromosome=$line_ary[1] AND (bov_bp_start>=$line_ary[2] AND
bov_bp_end<=$line_ary[3])";
```

```
   #Fol all regions
   my $query;
   if($line_ary[0] =~ m/X/)
   {
          #print "hey: $line_ary[0]\n";
          #$query = "select go_term, bovine_gene_id from go_withLocations where bovine_chromosome
LIKE '%X%'";
          $query = "select go_term, bovine_gene_id from go_withLocations where bovine_chromosome
LIKE '%X%' AND (bov_bp_start>=$line_ary[1] AND bov_bp_end<=$line_ary[2])";
          #print "$query\n";
   }
   else{
          #$query = "select go_term, bovine_gene_id from go_withLocations where
bovine_chromosome=$line_ary[0]";
          $query = "select go_term, bovine_gene_id from go_withLocations where
bovine_chromosome=$line_ary[0] AND (bov_bp_start>=$line_ary[1] AND
bov_bp_end<=$line_ary[2])";
   #print "$query\n";}
   }

   my $sth = $dbh->prepare($query);
   my $rv = $sth->execute;
   while(my @row_ary  = $sth->fetchrow_array)
   {
          #$print "$row_ary[0]\n";
          $string = "$string ; $row_ary[0]";
          $gene_counter++;
   }
}



my %go_hash = ();

my @string_array = split(/ ; /, $string);



for(my $i=0; $i<@string_array; $i++)
{
   my $original=$string_array[$i];
   $string_array[$i] =~ s/"//g; #"
   $string_array[$i] =~ s/ /_/g;
   $string_array[$i] =~ s/_\[/ \[/g;
   chomp $string_array[$i];
   #print "$string_array[$i]\n";

   #if($string_array[$i] =~ m/(\S+) \[molecular_function\]/)
   #if($string_array[$i] =~ m/(\S+) \[cellular_component\]/)
   #if($string_array[$i] =~ m/(\S+) \[biological_process\]/)
   if( ($string_array[$i] =~ m/(\S+) \[biological_process\]/) || ($string_array[$i] =~ m/(\S+)
\[cellular_component\]/) || ($string_array[$i] =~ m/(\S+) \[molecular_function\]/) )
   {
          #print "  HEY:$1\n";
          #if($1 =~ m/biological_process/g){
          #print "Original:$original";          }

          if(exists ($go_hash{$1}) )
          {
             my $value = $go_hash{$1} +1;
             $go_hash{$1} = $value;
          }
          else
          {
             $go_hash{$1} = 1;
          }
          #print "  $go_hash{$1}\n";
   }
}

#print (%go_hash,"\n");

my $k;
my $v;

while ( ($k,$v) = each %go_hash ) {
   $k =~ s/_/ /g;
   print "$k        $v\n";
}

print "Number of genes: $gene_counter\n";

close DAT;
#!/usr/bin/perl
use strict;
use DBI;


my $file = "$ARGV[0]";
chomp ($file);


open (DAT, $file) || die("Could not open file");
my @lines = <DAT>;

#CONNECTING TO THE NCBI_STS DB
my $dbh = DBI->connect('DBI:mysql:gene_ontology', 'hanni', 'hanni');

my $string="";
my $gene_counter=0;

for(my $z=0; $z<@lines; $z++)
{
   chomp $lines[$z];
   my @line_ary = split (/\t/, $lines[$z]);
   #print "$line_ary[0]-$line_ary[1]-$line_ary[2]\n";
   my $chrom_string;
   if($line_ary[1] eq '')
   {
          $line_ary[1] =0;
          $line_ary[2] =0;
   }

   if($line_ary[0] =~ m/X/g)
   {
          $chrom_string = "bovine_chromosome='X'";
   }
   else
   {
          $chrom_string = "Bovine_chromosome=$line_ary[0]";
   }


   #For qtl regions
   my $query = "select go_term, bovine_gene_id, gene_name from go_withLocations where
$chrom_string AND (bov_bp_start>=$line_ary[1] AND bov_bp_end<=$line_ary[2])";
   #For non qtl regions
   #my $query = "select go_term, bovine_gene_id, gene_name from go_withLocations where
$chrom_string  AND (bov_bp_start>=$line_ary[2] AND bov_bp_end<=$line_ary[3])";

   print "$query\n";
   my $sth = $dbh->prepare($query);
   my $rv = $sth->execute;
   while(my @row_ary  = $sth->fetchrow_array)
   {
      #print "$row_ary[1]   $row_ary[2]\n";
      $string = "$string ; $row_ary[0]";
          $gene_counter++;
   }
}

print "Number of Genes: $gene_counter\n";
#!/usr/bin/perl
use strict;
use dbi;

#CONNECTING TO THE NCBI_STS DB
my $dbh = DBI->connect('DBI:mysql:GO_Official', 'hanni', 'hanni');

#my $file_name="./disease_resistance_terms.txt";
my $file_name = "$ARGV[0]";
chomp ($file_name);
open(DAT, $file_name) || die("Could not open file!");

my %go_hash = ();

while(my $line=<DAT>)
{
   chomp $line;
```

```
    my @line_ary = split(/\t/, $line);
    #print "$line_ary[0]--$line_ary[1]\n";
    my $terms = get_go2level($line_ary[0]);
    #print "$terms\n";
    my @term_array=split(/;/, $terms);
    my %hash = map { $_ => 1 } @term_array;
    my @term_array2 = sort keys %hash;

    print "$line_ary[0]         ";
    for(my $h=0; $h<@term_array2; $h++)
    {
            print "$term_array2[$h]--";
            #print "$line_ary[1]\n";

            if(exists ($go_hash{$term_array2[$h]}) )
            {
               my $value = $go_hash{$term_array2[$h]} +1;
               $go_hash{$term_array2[$h]} = $value;
            }
            else
            {
               $go_hash{$term_array2[$h]} = 1;
            }
    }
    print "$line_ary[1]\n";

}


# Loop that justs prints out what is in the hash
my $k;
my $v;
while ( ($k,$v) = each %go_hash ) {
    $k =~ s/_/ /g;
    #print "$k   $v\n";
}




sub get_go2level() {
    my $sub_go_term = shift;

    $sub_go_term =~ s/'/\\'/g;
    #print "$sub_go_term\n";

    my @AoA = ();
    my $size=0;

    #print "########## $sub_go_term #############\n";
    my $string=";
```

```
   my $query = "SELECT p.*, distance FROM  graph_path    INNER JOIN  term AS t ON (t.id =
graph_path.term2_id)   INNER JOIN  term AS p ON (p.id = graph_path.term1_id) WHERE t.name =
'$sub_go_term';";
   my $sth = $dbh->prepare($query);
   my $rv = $sth->execute;
   while(my @row_ary  = $sth->fetchrow_array)
   {
      #$print "$row_ary[0]\n";
      #$string = "$string;$row_ary[0]";
          $AoA[$size][0]=$row_ary[0];
          $AoA[$size][1]=$row_ary[1];
          $AoA[$size][2]=$row_ary[6];
          $size++;
   }

   for(my $i=0; $i<$size; $i++)
   {
          #print "$AoA[$i][0]-$AoA[$i][1]-$AoA[$i][2]\n";

          if( ($AoA[$i][1] eq 'biological_process') || ($AoA[$i][1] eq 'molecular_function') || ($AoA[$i][1]
eq 'cellular_component') )
          {
             my $nec_distance=($AoA[$i][2]-1);
             #print "we need to go backwards to the last distance-1: $nec_distance\n";
             #print "$AoA[$i][0]-$AoA[$i][1]-$AoA[$i][2]\n";
             #we need to loop back to get the last distance 1 less
             #for(my $z=$i; $z>=0; $z--)
             my $z=$i;
             while($AoA[$z][2]!=$nec_distance)
             {
                     $z--;
             }
             #print "The term:$AoA[$z][1] nec distance=$nec_distance actual_distance=$AoA[$z][2]\n";
             if($string eq ''){
                     $string="$AoA[$z][1]";}
             else{
                     $string = "$string;$AoA[$z][1]";}
          }
   }
   #print "$string\n";
   return $string;
}

close DAT;
#!/usr/bin/perl
use strict;
use DBI;

#my $file="./Lactation/lactation_regions_QTL_total.txt";

my $file=$ARGV[0];
open (DAT, $file) || die("Could not open file");
my @lines=<DAT>;
my @AoA;
my $array_size=0;

#loop through every chromosome
for(my $i=1; $i<30; $i++)
{
@AoA = ();
$array_size=0;
   for(my $z=0; $z<@lines; $z++)
   {
           chomp $lines[$z];
           my @line_ary = split(/\t/, $lines[$z]);
           if($line_ary[4] == $i)
           {
              generate_regions_list($line_ary[5], $line_ary[6]);
           }
   }


#print "Chromosome $y - Complete regions array: \n";
for(my $y=0; $y<$array_size; $y++)
{
   print "$i          $AoA[$y][0]       $AoA[$y][1]\n";
}

}

sub generate_regions_list(){
   my $sub_start = shift;
   my $sub_end = shift;

   my $in_array_flag=0;
   #check against master array then store into final array
   for(my $i=0; $i<$array_size; $i++)
   {
      if($sub_start >= $AoA[$i][0] && $sub_end <= $AoA[$i][1]){$in_array_flag=1;}
         #print "$sub_start, $sub_end is within the qtl region -- $AoA[$i][0]..$AoA[$i][1]\n";}
      elsif($sub_start >= $AoA[$i][0] && $sub_start <= $AoA[$i][1] && $sub_end > $AoA[$i][1]){
         #print "$sub_start, $sub_end is extending the qtl region--";
         $AoA[$i][1] = $sub_end;
         $in_array_flag=1;}
         #print "$AoA[$i][0]..$AoA[$i][1]\n";}
      elsif( ($sub_start < $AoA[$i][0] && $sub_end < $AoA[$i][0])  || ($sub_start > $AoA[$i][1] &&
$sub_end > $AoA[$i][1]) ){
         #print "$sub_start, $sub_end is outside of the QTL region -- $AoA[$i][0]..$AoA[$i][1]\n";
           }
      elsif($sub_start < $AoA[$i][0] && $sub_end >= $AoA[$i][0] && $sub_end <= $AoA[$i][1]){
         #print "$sub_start, $sub_end is extending the qtl region from the front--";
         $AoA[$i][0] = $sub_start;
         #print "$AoA[$i][0]..$AoA[$i][1]\n";
         $in_array_flag=1;
           }
```

```
  }

  if($in_array_flag ==0){
    $AoA[$array_size][0] = $sub_start;
    $AoA[$array_size][1] = $sub_end;
       #print "new element: $sub_start--$sub_end\n";
    $array_size++;}
}
#!/usr/bin/perl
use strict;

my $file=$ARGV[0];
open (DAT, $file) || die("Could not open the file");

my @lines = <DAT>;

my @go_array;

my $first=1;
my %go_hash=();

my @category_array = ('antioxidant activity',);

#while(my $line=<DAT>)
for(my $x=0; $x<@lines; $x++)
{
  my $line=$lines[$x];
  chomp $line;
  #print "$line\n";
  my @line_ary=split(/\t/, $line);

  if($line =~ m/(\d+)\t/)
  {

       for(my $c=0; $c<@category_array; $c++)
       {
         if(exists ($go_hash{$category_array[$c]}) ){
              print "$go_hash{$category_array[$c]}          ";}
         else{
              print "0  ";
         }
       }
       print "\n";
       #print "ENTERING A NEW BIN\n\n\n";
       if($line =~ m/span/){
         print "$line      ";}
       else{
         print "$line                              ";}
       %go_hash=();
  }
  else
  {
      if($line =~ m/GO->parents: (\w+)->(.+)/)
      {
             my %temp_hash = ();
             #print "$2--\n";
             my @term_array=split(/;/, $2);
             for(my $z=0; $z<@term_array; $z++)
             {
                    #print "-$term_array[$z]-\n";
                    if(exists ($go_hash{$term_array[$z]}) )
                    {
                       if(exists($temp_hash{$term_array[$z]}) ){}else{
                       my $value = $go_hash{$term_array[$z]}+1;
                       $go_hash{$term_array[$z]} = $value;}
                    }
                    else
                    {
                       $go_hash{$term_array[$z]} = 1;
                    }

                    $temp_hash{$term_array[$z]}=1;
             }
          }
   }

}

#print out 2nd level go terms from the hash
my $k;
my $v;
while ( ($k,$v) = each %go_hash ) {
   $k =~ s/_/ /g;
#    print "$k   $v\n";
}
for(my $c=0; $c<@category_array; $c++)
{
   if(exists ($go_hash{$category_array[$c]}) ){
          print "$go_hash{$category_array[$c]}          ";}
   else{
          print "0  ";
   }
}
print "\n";
close DAT;
#!/usr/bin/perl
use strict;
my $file=$ARGV[0];
open(DAT, $file) or die("Could not open the file");
my @lines = <DAT>;
my @temp=split(/\t/, $lines[0]);
my $bin_chrom=$temp[0];
my $bin_start=$temp[1];
my $bin_end=$temp[1]+5000000;
my $span_flag=0;
my $need_to_add_start;
my $fut_chrom;
my @new_bin;
my $go_to_span_flag=0;
for(my $i=0; $i<@lines; $i++)
{
   chomp $lines[$i];
   my @line_ary=split(/\t/, $lines[$i]);


   if($span_flag==1){
         $bin_end=$line_ary[1]+$need_to_add_start;
         print "$line_ary[0]       $line_ary[1]      $bin_end\n";
         $span_flag=0;
         #$bin_end=$bin_end+1;
         $bin_start=$bin_end+1;
         $bin_end=$bin_start+5000000;
   }

   #print "$line_ary[1]-$line_ary[2]\n";


   if($bin_end>$line_ary[2]){
         #my $end_chrom=check_chroms($line_ary[0]);
         my $range=$bin_end-$bin_start;
         if($range==4999999 || $range==5000000){
            print "$line_ary[0]     $bin_start        $bin_end          -$range-\n";}else{
            print "This is not a good bin\n";}
         my @temp2=split(/\t/, $lines[$i+1]);
         if($bin_chrom==$temp2[0] && $bin_end>$temp2[1]){
            #print "This is overlap: $bin_chrom, $temp2[0], $bin_end, $temp2[1]\n";
            $bin_start=$bin_end+1;
            $bin_end=$bin_start+5000000;
         }
         else{
            #print "This is not overlap: $bin_chrom, $temp2[0], $bin_end, $temp2[1], $temp2[2]\n";
            chomp $temp2[2];
            $bin_start=$temp2[1]+1;
            $bin_end=$temp2[1]+5000000;
            $line_ary[0]=$temp2[0];
            $line_ary[2]=$temp2[2];
            $i++;
         }
         #print "new binstart:$bin_start, new binend:$bin_end\n";
   }



   while($bin_end<$line_ary[2])
   {
         my $range=$bin_start-$bin_end;

         if($bin_end>$bin_start){
            print "$line_ary[0]        $bin_start           $bin_end            -$range-\n";}

         $bin_start=$bin_end+1;
         $bin_end=$bin_end+5000000;
         $span_flag=1;
   }

   if($span_flag==1){
         $bin_end=$bin_end-5000000;
         my $amt_left=$line_ary[2]-$bin_end;

         $need_to_add_start=5000000-$amt_left;
         my $temp_end = $bin_start+$need_to_add_start;

         print "span          $line_ary[0]        $bin_start           $line_ary[2]        ";
         $span_flag=1;
         $amt_left=0;
   }

}
sub check_chroms(){
   my $sub_input_chrom=shift;

   open(DAT3, "../Full_Genome/reference_sequences.gff") || die("Could not open references");
   while (my $line=<DAT3>)
   {
         if($line =~ m/Chr(\d+)/)
         {
            if($sub_input_chrom==$1){
                    my @temp=split(/\t/, $line);
                    return $temp[4];
            }
         }

   }

}


close DAT3;


#!/usr/bin/perl
use strict;
use DBI;
my $file=$ARGV[0];
open(DAT, $file) || die("Could not open file");
while(my $line=<DAT>)
{
   #CONNECTING TO THE NCBI_STS DB
   my $dbh = DBI->connect('DBI:mysql:gene_ontology', 'hanni', 'hanni');

   chomp $line;
    my @line_ary=split(/\t/, $line);
    #print "$line_ary[0]-$line_ary[1]-$line_ary[2]\n";
    my $string;
    if($line_ary[0]=~m/X/){
              $line_ary[0]="'X'";}

    my $query;
    if($line=~m/span/){
              #print "span";
              $query ="select go_term, bovine_gene_id from go_withLocations where
(bovine_chromosome=$line_ary[1] AND (bov_bp_start>=$line_ary[2] AND
bov_bp_end<=$line_ary[2])) OR (bovine_chromosome=$line_ary[4] AND (bov_bp_start>=$line_ary[5]
AND bov_bp_end<=$line_ary[6]))";
    }
    else{
              $query = "select go_term, bovine_gene_id from go_withLocations where
bovine_chromosome=$line_ary[0] AND (bov_bp_start>=$line_ary[1] AND
bov_bp_end<=$line_ary[2])";
}
    #print "$query\n";

    my $sth = $dbh->prepare($query);
    my $rv = $sth->execute;
    while(my @row_ary  = $sth->fetchrow_array)
    {
        #$print "$row_ary[0]\n";
        $string = "$string ; $row_ary[0]";
    }

    #print "$line\n$string\n\n";
    print "$line\n";
    #my $level3_ontology=get_3rd_level($string);
    my @level3_array=get_3rd_level($string);
    my $size2=@level3_array;
    print "Returned size: $size2\n";
    parse_level3_go(@level3_array);
}

close DAT;

sub get_3rd_level(){
    my $sub_string=shift;

    my @string_array = split(/ ; /, $sub_string);
    my @return_terms=();
    my $return_counter=0;

    for(my $i=0; $i<@string_array; $i++)
    {
              my $original=$string_array[$i];
              $string_array[$i] =~ s/"//g; #"
              $string_array[$i] =~ s/ /_/g;
              $string_array[$i] =~ s/_\[/ \[/g;
```

```
           chomp $string_array[$i];
           #print "$string_array[$i]\n";

           if( ($string_array[$i] =~ m/(\S+) \[biological_process\]/) || ($string_array[$i] =~ m/(\S+)
\[cellular_component\]/) || ($string_array[$i] =~ m/(\S+) \[molecular_function\]/) )
           {
              #print "$1,,";
              #print "We are about to go into the connect_3rd_level subroutine with:$1\n";
              my $term3=connect_3rd_level($1);
              print "GO->parents: $1->$term3\n";
              $return_terms[$return_counter]=$term3;
              $return_counter++;
              #print "$return_terms[$return_counter-1]+++++++$return_counter\n";
           }
   }
   my $size=@return_terms;
   print "Size of return: $size\n";
   return @return_terms;
}

sub parse_level3_go(){
   my @sub_ontology=shift;
   #print "-$sub_ontology-\n";
   my $size=@sub_ontology;
   print "Number of terms in subroutine array:$size\n";
   for(my $z=0; $z<@sub_ontology; $z++)
   {
           #print "$sub_ontology[$z]\n";
   }


}
sub connect_3rd_level(){
   my $sub_go_term = shift;
   #print "$sub_go_term--";
   $sub_go_term =~ s/_/ /g;
   $sub_go_term =~ s/'/\\'/g;

   my @AoA;
   my $size=0;

   #CONNECTING TO THE NCBI_STS DB
   my $dbh = DBI->connect('DBI:mysql:GO_Official', 'hanni', 'hanni');

   my $query = "SELECT p.*, distance FROM  graph_path    INNER JOIN  term AS t ON (t.id =
graph_path.term2_id)   INNER JOIN  term AS p ON (p.id = graph_path.term1_id) WHERE t.name =
'$sub_go_term';";
   my $sth = $dbh->prepare($query);
   my $rv = $sth->execute;
   while(my @row_ary  = $sth->fetchrow_array)
   {
      #$print "$row_ary[0]\n";
      #$string = "$string;$row_ary[0]";
```

```
        $AoA[$size][0]=$row_ary[0];
        $AoA[$size][1]=$row_ary[1];
        $AoA[$size][2]=$row_ary[6];
        $size++;
    }
    my $string;
    for(my $i=0; $i<$size; $i++)
    {
        #print "$AoA[$i][0]-$AoA[$i][1]-$AoA[$i][2]\n";

        if( ($AoA[$i][1] eq 'biological_process') || ($AoA[$i][1] eq 'molecular_function') || ($AoA[$i][1] eq
'cellular_component') )
        {
            my $nec_distance=($AoA[$i][2]-1);
            #print "we need to go backwards to the last distance-1: $nec_distance\n";
            #print "$AoA[$i][0]-$AoA[$i][1]-$AoA[$i][2]\n";
            #we need to loop back to get the last distance 1 less
            #for(my $z=$i; $z>=0; $z--)
            my $z=$i;
            while($AoA[$z][2]!=$nec_distance)
            {
                $z--;
            }

            #print "The term:$AoA[$z][1] nec distance=$nec_distance actual_distance=$AoA[$z][2]\n";
            if($string eq ''){
                $string="$AoA[$z][1]";}
            else{
                $string = "$string;$AoA[$z][1]";}
        }
    }
    #print "$string\n";
    return $string;



}
```

VITA

Name: Hanni Salih

Address: Department of Animal Science
2471 TAMU
Kleberg Center
College Station, TX 77840

Email Address: h_s1lih@neo.tamu.edu

Education: B.S., Bioinformatics, Baylor University, 2003
Ph.D., Genetics, Texas A&M University, 2008