WEB INTERFACES FOR BIOINFORMATIC DATA REPRESENTATION

by

WYATT SPEAR

A THESIS

Presented to the Department of Computer
and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

August 2004

"Web Interfaces for Bioinformatic Data Representation," a thesis prepared by Wyatt Spear in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science. This thesis has been approved and accepted by:

Dr. John Conery, Chair of the Examining Committee

8-24-04

Date

Accepted By

Dean of the Graduate School

An Abstract of the Thesis of

Wyatt J. Spear for the degree of Master of Science

in the Department of Computer and Information Science

to be taken August 2004

Title: WEB INTERFACES FOR BIOINFORMATIC DATA REPRESENTATION

Approved: ______________________________
Doctor John Conery

The field of genetics has advanced rapidly in recent years and will continue to do so. The progress of genetic research requires massive amounts of data to be collected, stored and processed. Because of this, much biological research is now reliant on computer science and the intersection of the two fields, known as bioinformatics. To some extent the advance of genetic research is limited by the computational resources available.

The computational requirements of specific genetic research operations as they relate to data summary, distribution and presentation can be met by a single software application. An intuitive, web based, database interface, known as a 'datamart' is one promising, unified means of addressing these issues. This thesis will describe the principles and applications of bioinformatic datamarts and illustrate an example of successful datamart development and deployment.

CURRICULUM VITAE

NAME OF AUTHOR: Wyatt Spear

PLACE OF BIRTH: Corvallis, Oregon

DATE OF BIRTH: October 23, 1982

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon

DEGREES AWARDED:

Master of Science in Computer and Information Science, 2004, University of Oregon

Bachelor of Science in Compuer and Information Science, 2003, University of Oregon

AREAS OF SPECIAL INTEREST:

Bioinformatics

Marine Biology

Parallel Computing

PROFESSIONAL EXPERIENCE:

Research Assistant, Neuroinformatics Laboratory, University of Oregon, Eugene, Summer 2003

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

CHAPTER ONE

INTRODUCTION

Modern science has come to rely on computers to an enormous extent. The ability to store and rapidly process potentially massive quantities of data is not merely relief from painstaking tedium for the scientific community. It also opens new areas of science to investigation, areas that, even with the finest observational instruments, would be quite arduous to pursue by relying solely on ink, paper and the sturdy but relatively slow human brain.

One of the scientific fields that has become most strongly coupled with and dependent upon computational aid is biology. In particular, the subfield of genetics is practically inseparable from the mass data processing capabilities only computers can provide. The genes possessed by all living things are basically strings of coded data. This make them uncannily similar to computational instructions themselves. Thus the processing and analysis of genetic code by computational rather than biological means is virtually a 'natural' progression of technological capacity.

Neither the relevant computational methods, the specifics of which are typically tangential to the goals of genetic research, nor the genetic code itself, which is still only starting to become understood in its entirety, are highly intuitive to most people. The former requires a significant level of professional expertise to contend with. The latter requires both a high level of knowledge of the subject and the conceptualization of potentially massive quantities of raw information, in many cases beyond the capabilities of human reasoning within any reasonable amount of time. Clearly computers must not only provide the raw data processing capabilities required by genetic research, but must also provide an interface between the researcher, the computational processes being invoked and the genomic data itself.

The nature of such an interface is implicitly unspecific. The requirements of research tasks in genetics, and of the individual researches, are diverse and rarely static. However, this does not preclude the development of a relatively generic paradigm for genomic computational interfaces. The task then becomes identification of an interface that, by its essential nature, provides the highest level of utility to the greatest number of individuals. An interface concept that promises to approach that level of optimality is the data mart.

Data mart-like interfaces are based off of successful implementations for user interfaces to commercial venues and have proved successful throughout their history of deployment. The same set of features that permit easy access to and evaluation of a given on-line merchant's full array of products and services can be applied to the full array of available options for accessing, processing and presentation of scientific data. Furthermore, data mart-like implementations are constantly being developed and improved upon both for commercial and academic purposes. This provides a constant influx of useful data for their deployment in new application-specific tasks such as genomics.

A BIOINFORMATIC WEB INTERFACE: THE TRNA MART

An example of the current trend toward computational resource development and deployment is the tRNA Mart in production for the Saks Research Group. The essential function of this system is the storage and recovery of specialized biodata.

The basic framework of the tRNA Mart was in place when I began to participate in the project. There were resource for accessing the system's database tables in basic formats and outlines of additional interface protocols for more advanced data recovery and processing procedures. I successfully implemented two core data access modes for the tRNA Mart. These were a textual data extraction mechanism with user definable output formats and a data-rich graphical output mechanism. In addition to increasing the requisite utility of the tRNA Mart during my participation in its development my work illustrated both the utility, bordering on necessity, of modular, sequential system development procedures and the high degree to which computational productivity is a prerequisite for biological, specifically genomic, research.

CHAPTER TWO

GENETICS: A BIOLOGICAL OVERVIEW

CELLULAR GENETICS

Naturally, before discussing the implementation and functionality of a data interface system it is necessary to carefully consider the genomic research applications that would benefit from such an interface.

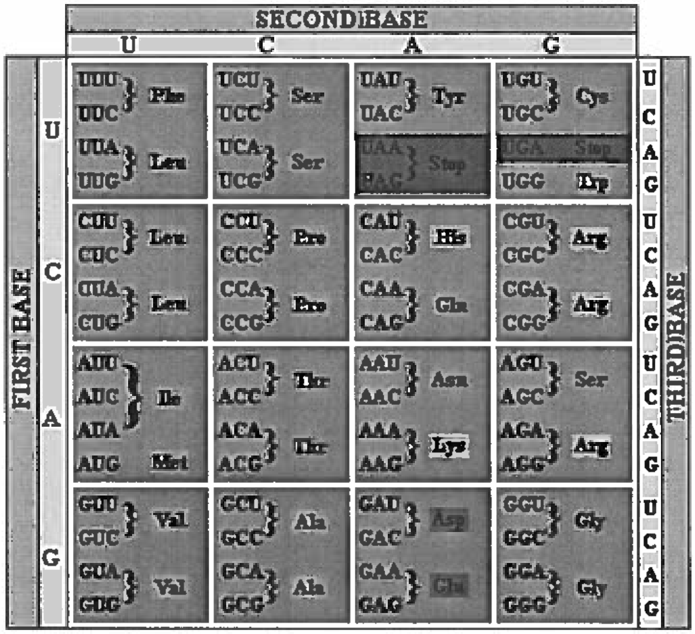
The field of genetics contains many subfields, each receiving varying amounts of attention from the scientific community. As the field advances, further specialization within its existing subfields and discovery of new ones has been the norm. This trend is unlikely to change in the near future.

The full diversity of modern genetic research is well outside the scope of this thesis. However there are some basic concepts that must be understood to even consider the matter in general terms.

Genes are stored on molecular strands of deoxyribonucleic acid (DNA). DNA is considered an informational polymer, that is a molecule comprised of discrete subcomponents whose order and position encode data. The genes contained within DNA are instructions that tell a cell how to construct the proteins it requires to operate. DNA is extremely well suited for this purpose and there is ongoing research into its artificial synthesis for use in storage of other forms of data.

The monomers, small molecular components, that comprise the DNA are known as nucleotides. The four nucleotides used in DNA are adenine, guanine, thymine and cytosine or A,G,T and C for short. The genetic code is comprised of sequences of nucleotides.

A triplet of nucleotides codes for a single amino acid. Amino acids are the building blocks of proteins so a sequence of amino acid coding nucleotide triplets is essentially biological notation for a sequence of amino acids. There are 20 amino acids used by the body and 64 possible combinations of nucleotide triplets. Three triplet combinations are also used as stop signals, indicating the termination of a coding sequence of DNA. This redundancy (see figure 1) helps avoid mutation by allowing slight changes in the genetic code without affecting the structure of the final protein.



	SECOND BASE				
FIRST BASE	U	C	A	G	THIRD BASE
U	UUU UUC } Phe	UCU UCC } Ser	UAU UAC } Tyr	UGU UGC } Cys	U C
	UUA UUG } Leu	UCA UCG } Ser	UAA UAG } Stop	UGA Stop UGG Trp	A G
C	CUU CUC } Leu	CCU CCC } Pro	CAU CAC } His	CGU CGC } Arg	U C
	CUA CUG } Leu	CCA CCG } Pro	CAA CAG } Gln	CGA CGG } Arg	A G
A	AUU AUC AUA } Ile	ACU ACC } Thr	AAU AAC } Asn	AGU AGC } Ser	U C A
	AUG Met	ACA ACG } Thr	AAA AAG } Lys	AGA AGG } Arg	G
G	GUU GUC } Val	GCU GCC } Ala	GAU GAC } Asp	GGU GGC } Gly	U C
	GUA GUG } Val	GCA GCG } Ala	GAA GAG } Glu	GGA GGG } Gly	A G

Figure 1. The 64 Nucleotide Triplets and Their Corresponding Products. (4)

The physical structure of DNA is fairly simple compared to other macromolecules, such as enzymes. Nucleotides bond to form base pairs. A bonds to T and C to G. A DNA strand is a double helix consisting of base pairs attached to each other like rungs on a ladder. When DNA is read or duplicated cellular machinery 'unzips' it along its center, dividing the base pairs into their component nucleotides. Then new complementary nucleotides can be attached to either side of the divided DNA strand, either forming two new identical strands, or allowing the construction of a copy of a localized genetic sequence.

Single strands of DNA are stored in tightly bundled packs known as chromosomes. Depending on the species in question a single chromosome can hold hundreds of thousands of individual genes. In all but the simplest of single celled organisms, chromosomes reside within the cellular nucleus.

DNA must be read by the cell to make use of the information it contains. Molecular machinery within the cell splits the DNA strand as described above, just in the area of the desired gene. The genetic code of the selected sequence is then transcribed it to another similar form of molecule known as ribonucleic acid (RNA). RNA has several functions within a cell, but those RNA molecules transcribed from DNA's protein coding sequences, known as mRNA or messenger RNA, contain the exact coding instructions for construction of a single protein. When the RNA comes in contact with another type of molecular machine known as a ribosome, the ribosome 'reads' the RNA and uses it as a template for the construction of the protein. Another form or RNA similarly transcribed from DNA does not hold proteomic code, but rather collects individual amino acids for transfer to the ribosome. This process, highly simplified here, is among the most fundamental components of cellular biology.

GENOMIC APPLICATIONS

An organism's genome defines its biological structure and functionality. The ability to understand and manipulate genetics is the ability to understand and manipulate life at its most fundamental level. There are a massive number of specific applications for such abilities.

Detection and treatment of diseases are among the most commonly considered applications of applied genomics and are likely to become the most popular when the relevant technology is perfected. However, the development of genetic technology is already allowing a wide array of activities that would have seemed impossible just a few decades ago. Virtually all such technologies are reliant, to one extent or another, upon computational aid for storage and analysis of genetic data.

GENETIC ENGINEERING

One of the largest of the broad categories genomic application is genetic engineering. With sufficient understanding of the genes within a given genome, it is possible to identify those that code for desirable or undesirable traits. Reproductive cells from the target organism can then either have undesirable genetic traits removed from or replaced within their chromosomes, or genes that code for desirable traits can be implanted. The result will be an organism that displays the desired traits. With today's technology this process is far from fool proof, but for it to succeed at all the content of the target creatures' genomes must be painstakingly cataloged and available for reference.

More advanced genetic engineering techniques, such as designing entirely new genes for custom proteins rather than relocating existing genes, are farther still from full realization. The field of genetic engineering remains sufficiently new and unexplored that the full implications of its broad application remain open to question.

PHYLOGENETIC ANALYSIS

While the biological processes involved in cellular replication and reproduction are highly fault tolerant, errors still occur. An alteration to a gene that results in behavior that differs from that of the original gene is rare, and beneficial results from such mutations are rarer still. However, evolution is dictated primarily by this genetic variability present in all organisms.

One important result of mutation's role in evolution is that, independent of the gross physical traits that were once the sole means of determining phylogeny, the genes themselves can act as accurate indicators of the degree of relatedness between species.

Phylogenetic analysis is generally accomplished by the selection of a specific set of genes or sequences of genetic code that have some degree of equivalence between the species being compared. The genetic code is then analyzed to determine the degree of divergence. This allows good approximations of species relatedness to be determined. Additionally, with accurate estimates of mutation rates, it is possible to estimate the approximate time at which two related species diverged from a common ancestor.

These techniques rely heavily on computational analysis of genetic code. The algorithms employed to efficiently produce a phylogenetic comparison may use code sequences thousands of nucleotides long, and must produce results with a high degree of statistical accuracy.

BIORESEARCH

While understanding of microbiology and molecular biology has increased drastically in recent years it is clear that knowledge of many of the functions of cellular genetics remains incomplete. Furthermore, understanding the contents of the genomes of humanity and other species is a scientific priority that is still far from being met.

The ability to merely look at a sequence of genetic code and differentiate exons (strings of genetic code that actually code for the creation of a protein) from introns (strings of genetic code that have no known functional purpose, but separate the coding sequences within DNA strands), let alone identify the function of any given exon at a glance is still far beyond current capabilities. Such capabilities will become increasingly necessary as knowledge of and reliance upon the field of genetics increases. Unquestionably, computational storage and analysis will be the means by which these goals are achieved.

GENETIC DATA REPRESENTATION

Clearly the close ties between genetics and computers necessitates an efficient means of translating from the molecular data strings to a format usable by computers. Fortunately, the essential simplicity of the genetic code is such that this translation is virtually trivial. The four nucleotides, A, C, T and G, are generally simply represented as standard text strings. String analysis and manipulation functions, already well developed for other applications, lend themselves readily to the computational requirements of bioinformatics. The similarity of bioinformatic data processing to preexisting computational applications is a great boon for genetic researchers.

While simple text files containing unmodified nucleotide sequences are sufficient for many purposes there are certain standards of genomic data presentation and elaborations upon the basic coding sequences that are employed in computational genetics.

IUPAC NUCLEIC ACID CODES

Ideally complete data on a given genetic sequence would always be available. However, this is not the case. Sometimes certain nucleotides in a genetic sequence are unknown, or can at best only be guessed at. In other situations, such as when considering multiple redundant sequences that code for the same protein, it simply may not matter what the specific nucleotides are so long as they code for the correct amino acids.

The International Union of Pure and Applied Chemistry (IUPAC) produced a set of nucleic acid codes to represent variability of nucleotides in a genetic sequence (see table X). The IUPAC codes allow for a concise representation of more general cases than the four standard nucleotides alone can provide. Although many genomic applications require only the four basic nucleotide symbols, A, C, T and G, the IUPAC codes are accepted as the general language of genetics, and are accepted and understood by the majority of genetic software applications.

Table 1. The IUPAC Nucleic Acid Codes (9)

Symbol	Substances Represented
A	adenine
C	cytosine
G	guanine
T	thymine
U	uracil
R	G A (purine)
Y	T C (pyrimidine)
K	G T (keto)
M	A C (amino)
S	G C
W	A T
B	G T C
D	G A T
H	A C T
V	G C A
N	A G C T (any)

STANDARDIZED GENETIC DATA FORMATS

As the field of genetics grows so does the number of computational applications it employs. Such applications require standardized input and output formats. There are several such formats in common use. The most basic of these is plain sequence format, which is simply a text file containing a single string of IUPAC code.

Possibly the most ubiquitous format aside from plain sequence is the FASTA format. FASTA is a popular sequence analysis tool, now sufficiently ubiquitous to set the trend in sequence data formatting. The FASTA data format consists of a header line containing data relevant to the following sequence. This data may vary according to the specific requirements of the genomic application. Delineation by header lines allows multiple sequences to be stored in the same file, which increases processing efficiency in many cases.

CHAPTER THREE

INFORMATICS: DATA INTERFACES AND PROCESSING

GENOMIC SOFTWARE APPLICATIONS

Naturally, modern genetics makes significant use of existing standardized software applications and environments, such as MySQL databases. However, the number of applications used in computational genomics is constantly increasing. In addition to main-stream non-field-specific applications and field standard applications many research groups produce their own custom applications, or modify existing ones, to suit the exact needs of their projects. As a result, there are a staggering number of individualized variations of software applications with essentially the same functionality. Some codification and collection of the most popular and commonly employed functions can be found in the BioPerl software suite, a community maintained collection of Perl scripts for genetic and proteomic analysis.

A major standard applications in the field of computational genomics is BLAST(6). As described by its authors,"BLAST 2.0 (**B**asic **L**ocal **A**lignment **S**earch **T**ool), provides a method for rapid searching of nucleotide and protein databases."(1) Another similarly popular application, FASTA, a performs statistical comparison of genomic and proteomic sequences. Notably, both of these systems use web interfaces to access database systems.

A rapidly growing bioniformatic resource is the Ensembl database system. Ensembl is designed specifically to accommodate the requirements for mass storage of genetic data. Indeed, it might be considered a prototypical genetic data warehousing system. An independent data mart interface for the Ensembl database system, EnsMart, typifies the concept of a bioinformatic data mart. The basic functionality of the EnsMart system was described as follows:

> "The EnsMart system (www.ensembl.org/EnsMart) provides a generic data warehousing solution for fast and flexible querying of large biological data sets and integration with third-party data and tools. The system consists of a query-optimized database and interactive, user-friendly interfaces."(5)

Clearly, databases and their resultant elaborations, data warehouses and data marts, are a critical component of computational genomics. Indeed, it is arguable that once the genetic data is extracted from the cell and entered into a database, further analysis and manipulation of the data is the exclusive responsibility of the database, its interfaces and internal functions. There is no great advantage to separation of data storage and data manipulation, while advantages include decreased effort on the part of the user and potentially increased efficiency of computation as the requirements of data transfer between applications are reduced and the specific environments in which the data is stored become increasingly specialized and optimized for the task.

DATA MARTS

Data marts and data warehouses are concepts that have come into popular usage as computational data storage methods have become increasingly ubiquitous. The relationship between the two data storage and access mechanisms closely mirrors their classical tangible counterparts.

A data warehouse is a database system oriented toward the storage of a large quantity of data on a potentially broad array of topics. For example, a data warehouse for a small company might contain employee information, product data, supplier and customer information and similar entries relevant to the functionality of the business. Data warehouses are not typified by any particular interface restrictions. In many cases, they provide only whatever raw data access capabilities are granted by the database system they employ.

A data mart, by contrast, specializes in a more specific type of data. More importantly data marts typically possess specialized interfaces not only for data access, but for venue-specific processing and presentation capabilities as well.

For example, a small business might possess a data mart that references a database of the company's product information, with interfaces that permit product customization and visualization options in addition to raw data output. In many cases a data mart, rather than being a self contained data storage and access application, is merely the interface for a subset of the contents of a data warehouse.

FUNCTIONAL REQUIREMENTS AND CAPABILITIES

While the essential function of data marts is easy enough to define, the specifics of their implementation and functionality cover a vast array of possibilities. The data marts to be considered here have functional requirements that do not greatly set them apart from the generic concept. A concern that is first and foremost for many data marts is ease of use. A data mart that merely provides a new venue into which arcane database commands may be entered provides insignificant benefit to the end user. In many cases data marts act not only as an interface between the user and the data, but an interface between the field of computer science and whatever field the data mart itself serves. In addition to the obvious convenience and potential for performing critical data manipulations, availability of a simple data interface system relieves the burden on the researcher of gaining a significant new skill set just for the purpose of data acquisition and processing, which is in most cases tangential to the goals of the research.

Data marts are capable of reducing the requisite complexity and size of their associated databases in some cases. When using a raw database interface it may be necessary to include multiple variations on the same data content. When accessing the core data through a data mart it may be preferable to perform such manipulations on the fly. This increases data access overhead while decreasing storage overhead and is thus another component of data mart design entirely dependent on the specific circumstances of the environmental requirements of the system.

Some data marts are designed exclusively to serve internal research purposes. However, this practice fails to take advantage of one of the primary strengths of the data mart concept. That is the data marts natural applicability to data distribution. The data mart, like the commercial applications from which it was derived, is virtually native to Internet based implementations. Data marts produced in web environments and/or in other distributable, multi-platform systems such as Java provide a simple means of data access over an area as wide as the distribution of the web itself. Additionally, although the concept of the data mart implies a venue for data acquisition, the interface can just as easily be adapted to data input. Thus, the innate distributivity of a data mart allows for a distributed means of collaborative data accrual that benefits from the environment's implicit level of user friendliness.

IMPLEMENTATION

The proliferation of advanced web programming systems such as PHP and Javascript has helped promote the increased deployment of data marts. Additionally, as more people become at least partially proficient in basic web programming data marts and data mart like interfaces are certain to become still more prolific. The essential software requirements of a basic data mart are trivial. An HTML form that accepts some simple input and retrieves the relevant output from a database can be thought of as a proto-data mart. Any novel complexity to be found in a data mart will typically derive not from the basic database access mechanisms, but from the interfaces to those mechanisms and the algorithms that manipulate the data between its extraction and presentation to the user.

Generally the majority of functionality, and aesthetic value, that could be desired from a standard web based data mart can be provided by native web based tools. In some cases, however, augmentation by the inclusion of server side scripts, or even more advanced executable programs may be called for. In such cases it is important to consider the necessity of including such advanced levels of functionality in a data mart. While increasing web based interface functionality has advantages, including functions that require significant time and processing resources to execute should be avoided both to avoid strain on the server (especially when the data mart is intended to be publicly accessible) and to help the user avoid unexpected commitment of time to a data retrieval operation. It may be preferable to offer downloadable scripts or executables to perform more resource intensive data manipulations.

When designing the visual interface for a data mart there is a degree of subjectivity to the matter, however minimally the design should be intuitive and should not detract or distract from its function. In general, data marts intended for public access will be more graphically oriented, while those designed primarily for internal use focus more strictly on functionality and display less concern for visual appeal.

It is important to keep in mind that the end user of most data marts will be generally familiar with the data being dispensed. Thus excessive simplification and the associated redundancies of functionality and documentation can be obviated to the extent that will be the case. On the other hand, simplicity is preferable to unnecessary complexity. A data mart with an interface no more simple than the raw database interface is scarcely useful.

DEVELOPMENT

Exploration of the origins of life on the planet is one of the primary goals of the field of biology. Increasingly genomics has held the potential to advance this goal in ways never before encountered. The construction of data storage and access systems for such pursuits has become increasingly important. The data warehousing requirements of many genomic research projects can be met by standardized tools, such as EnsMart, 'out of the box'. However there are some areas of study that require more specific software utilities. Research specific software requirements may occur when a given pursuit requires specialized computational analysis, introduces data types or formats that are not supported by standard utilities, or the work would benefit from an interface that differs from those available in preexisting products.

In such cases, the primary options are construction of a data mart system from the ground up, or modification of an existing system. Both options have advantages and disadvantages to be considered. When an existing standardized system can be converted to suit a project's specific needs the majority of the development work has already been done, and in many cases community support will be available. On the other hand if the available systems are so divergent from project requirements that extensive modifications are required it may be simpler to produce a new more specifically tailored system. Additionally, depending on the complexity of the tasks required by a given project, it may be simpler to construct a personalized system rather than deal with the presence of large quantities of unused functionality. Another important factor is the existing knowledge base of the individuals working on the system. For complex interfaces it may be ultimately more feasible to construct a new system with a familiar interface than deal with learning the to use available preexisting systems.

Ultimately, the preferable course must be determined on a case by case basis. Individual project requirements, availability of relevant preconstructed systems and difficulty of adjusting an existing system or developing ones own will all factor into the decision.

Weather constructing a new system or modifying or extending an old one, one of the most important development styles that can be applied to a data mart is modularity. By compartmentalizing the system's functionality it is made much easier to incorporate components of external data processing systems or interfaces with only minimal effort. Additionally, as the data mart's requirements change maintenance of old and incorporation of new functionality is strongly facilitated by a modular construction method.

BIOINFORMATIC DATAMARTS

The number of computational tasks and array of data forms required by the wide field of genomics is staggering. Specialization within the field is likely to result in continued development of task specific interfaces and increased complexity of general purpose interfaces. However, an archetypal implementation of a bioinformatic data mart is the EnsMart system.

The EnsMart database is designed to provide access to the data from the Ensembl genomic database system. The default Ensembl database includes genomic information from several species. There are two primary data types available as search focuses in EnsMart. Annotated gene data is essentially the default for most genomic databases and is naturally a primary data type in EnsMart. The other primary data type employed by EnsMart is known as Single Nucleotide Polymorphism or SNP. SNPs are individual nucleotides that display a degree of natural variability between the genomes of individuals within the same species. SNPs are worthy of their status as a primary database focus because of their potential importance in determining the variable traits of individuals, ranging from appearance to susceptibility to diseases.

The EnsMart interface is in fact comprised of three independent subsystems. MartView is a web based interface that grants high levels of system distribution and portability. MartExplorer is a stand alone application which, while duplicating the basic functionality of MartView, takes advantage of executable applications' data presentation and speed advantages over web based interfaces. MartShell is a purely text based interactive query system that interfaces with the EnsMart database system. While sacrificing the graphical interface capabilities of MartView and MartExplorer, MartShell provides important functionality by allowing command line based batch queries and pipelining queries to and query results from the system.

The graphical interfaces of the EnsMart system have a three step search routine derived from the structural composition of the database, which in turn are derived from the projected usage requirements of the system. The user first indicates which species in the system will be the subject of the search, along with the search focus. Focuses available by default are SNPs and three formats of genetic data.

With the species and focuses selected the system is effectively primed to return all of the genetic data in the selected focuses for the selected species. There are several mechanisms for filtering out unwanted genetic data, most of which are focus specific. Some of these mechanisms include indication of the physical location within the chromosomes of the desired data, indication of the relationship between the desired genomic data and its protein product, if any, and use of previously collected genetic datasets to filter by intersection. Throughout the filtration process the total number of individual items that will be produced by the filtration settings is displayed.

Finally, the filtered data must be output by the system. This step requires the user to define what specific data on each entry should be returned by the search. The available data depends on which focus has been selected, but include standard genetic header information and the same information that is employed as filtering criteria. Then an output format is selected and the query results are placed in the location indicated by the user.

The web based graphical EnsMart interface, MartView, is a Perl based system, while the stand alone MartExplorer is written in Java. Both systems are essentially platform independent, with the exception of some of the subordinate applications they employ in genomic processing. Fortunately those applications themselves are available for multiple platforms. In addition to the distributability afforded by its generic execution requirements, implementation in both Java and Perl allows EnsMart to be extended for more specific applications with relative ease. In particular, the ability to interface EnsMart with genomic data sets outside its standard Ensembl database content increases its potential usability significantly. Alterations to the database content necessitate complimentary alterations and additions to the focus and filtration systems, which are similarly designed for user extensibility.

CHAPTER FOUR

CASE STUDY: THE TRNA MART

THE FUNCTION OF TRNA MART

While more generic bioinformatic data marts such as EnsMart can indeed provide much of the core genomic data interface requirements, new system development remains a viable alternative to standardized system conversion. One genomic research operation that stood to benefit from development of a new bioinformatic data mart was the Saks Research Group. The group described its project goals as follows:

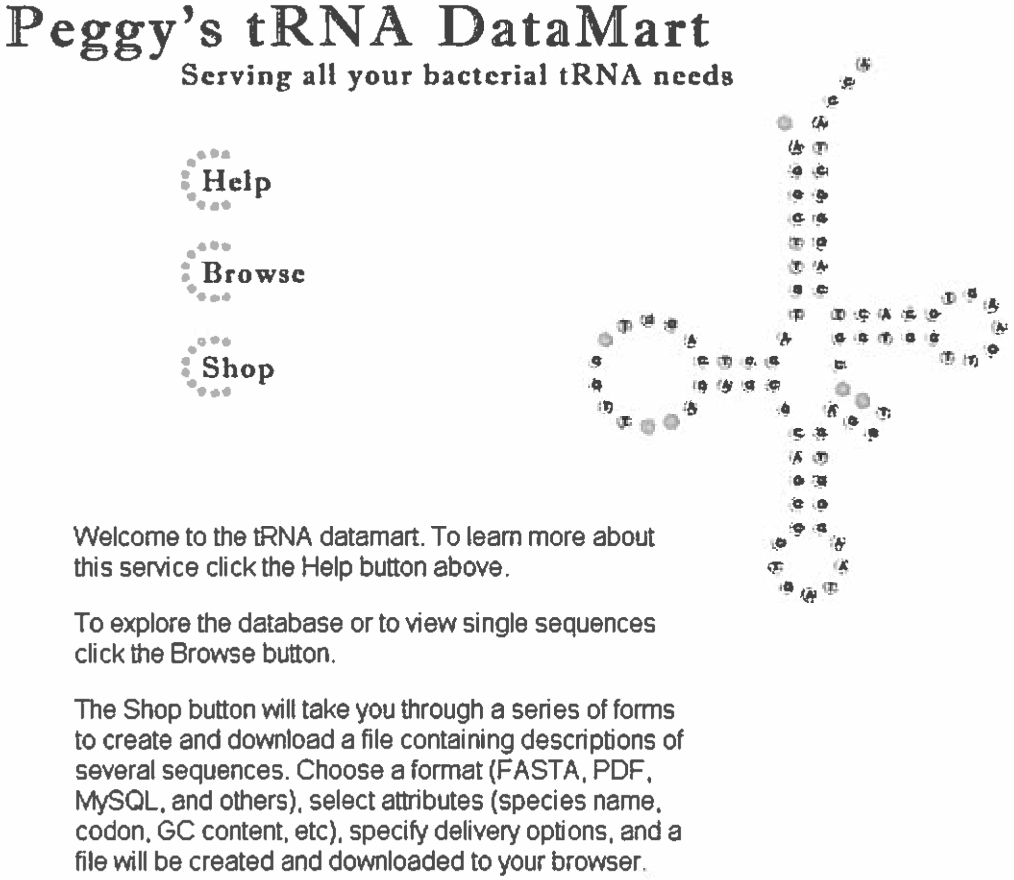
"Research in the Saks lab focuses on the essential cellular process of translation. We are primarily interested in two interrelated sets of research problems. The first set of problems involves characterizing, both in vivo and in vitro, the RNA-protein and RNA-RNA interactions that govern accurate and efficient translation. Besides providing basic information on the molecular mechanisms of translation, the results of these studies constitute the foundation for the second set of problems in the lab. These problems focus on elucidating the rules that govern the evolution of the translation machinery. For these studies, we draw on our understanding of translation to formulate hypotheses that we subsequently test, experimentally, using genetically tractable organisms as models. For both sets of problems, we combine biochemical, molecular, genetic, genomic, and evolutionary approaches. In addition, we are involved in collaborative projects to solve the structures of RNAs and proteins of the translation machinery that we are characterizing biochemically. " (7)

The group's focus on RNA and its structural dynamics are themselves somewhat outside the scope of standard data mart interfaces. Additionally the genetic data from the large number of microbial species the project uses in its studies are contained in a database system not specifically designed to interface with any preexisting data mart, making construction of an interface system in exact accordance with the project's database content a useful option.

The goal of the tRNA Mart(2) is to provide easy distributed access to the tRNA specific genomic data required by the project. In addition to simple database access, the tRNA mart is or will be capable of performing data manipulation and synthesis tasks conducive to the group's needs.

THE IMPLEMENTATION OF TRNA MART

The tRNA Mart is based primarily on PHP, which both generates the web pages viewed by the user and accesses and processes the project's database content. PHP has the dual advantages of being a versatile language in itself, with text processing capabilities comparable to Perl and being a widely accepted multi-platform language. The remote user of the tRNA Mart must only possess standard HTML browsing capabilities to make full use of the system and anyone wishing to implement a tRNA Mart server on their own system need only possess the ability to run PHP.



Peggy's tRNA DataMart

Serving all your bacterial tRNA needs

Welcome to the tRNA datamart. To learn more about this service click the Help button above.

To explore the database or to view single sequences click the Browse button.

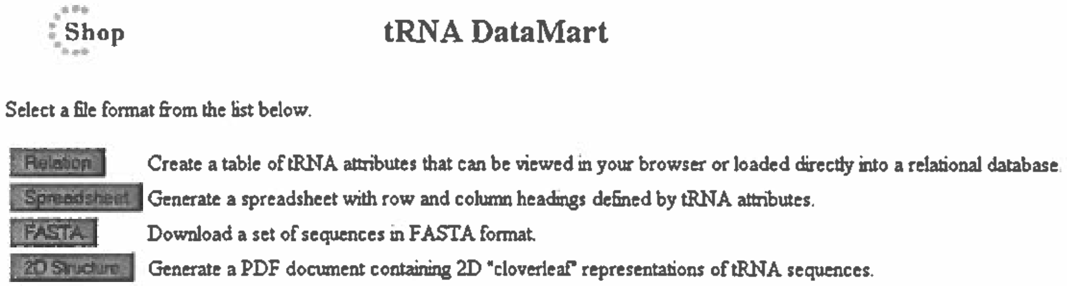
The Shop button will take you through a series of forms to create and download a file containing descriptions of several sequences. Choose a format (FASTA, PDF, MySQL, and others), select attributes (species name, codon, GC content, etc), specify delivery options, and a file will be created and downloaded to your browser.

Figure 2. The Opening Page of the tRNA Mart.

The current implementation of the tRNA Mart accesses genetic data stored in a MySQL database. While more specialized database systems do exist, the tRNA Mart was designed to access a preexisting database system, forgoing the necessity of data transfer and database system installation and familiarization. Additionally, the internal logic, where it is database-specific at all, is sufficiently modular to allow easy compensation for alterations to the current database structure, or transition to an entirely new database system should that become necessary.

The core content of the database accessed by the tRNA Mart consists of multiple aligned tRNA sequences of the microorganisms being studied. Descriptive data for each sequence is also provided, such as the scientific name of the organism from which the sequence was extracted and the protein for which the tRNA sequence codes. Additionally, the database contains sequence alignment information and graphical coordinate data used in the visualization of the sequences.

As of this writing there are four data output mechanisms provided by the tRNA Mart. Which of the four mechanisms will be employed is the first decision to be made by the user, in the current interface. This is in contrast to EnsMart's system, where the genetic sequences are selected first and the output mechanism determined at the end of the data mart session. However, the tRNA Mart's systemic modularity allows for adjustment to the order of operations, to conform with the needs of the research group.



tRNA DataMart

Select a file format from the list below.

Create a table of tRNA attributes that can be viewed in your browser or loaded directly into a relational database.

Generate a spreadsheet with row and column headings defined by tRNA attributes.

Download a set of sequences in FASTA format.

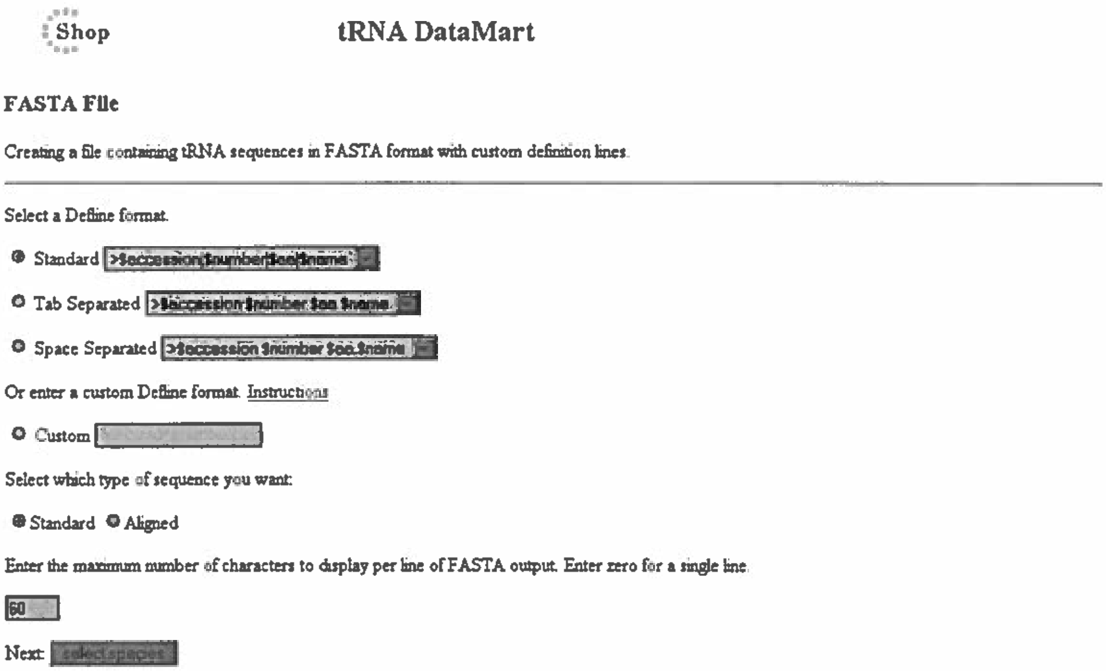
Generate a PDF document containing 2D "cloverleaf" representations of tRNA sequences.

Figure 3. The tRNA Mart Output Selection Screen.

The first and simplest output mechanism simply returns the raw database tables contained within the tRNA database. After selecting this option the user is able to select which of the database's tables are included in the output, what characters are used to separate the table columns (spaces or commas) and the presence of absence of table titles. The tables, once extracted, can be used for simple viewing, or uploaded to a new database.

A similar output mechanism provides the requested data in a simple spreadsheet format. The options available are comparable to those of the table viewer. The output options of both the table and spreadsheet mechanisms are made available with simple check boxes and radio buttons, for input of inclusive and exclusive options respectively.

The next output mechanism returns selected sequences in the FASTA format. The specific data included in the descriptive header of each sequence, as well as the order of that data and the delineating characters employed are user definable. FASTA output allows the database content to be processed by any genomic analysis tools that accept the format, of which there are many. The FASTA options selections differ from the others in that, in addition to static default formats to chose from, the user can define a custom sequence header style by inputting a sample header in the desired format. This is useful because, while the general FASTA format is standardized, the specific structure of the sequence headers remains somewhat variable thus the tRNA Mart can provide FASTA output consistent with whatever the project's requirements may be.



tRNA DataMart

FASTA File

Creating a file containing tRNA sequences in FASTA format with custom definition lines.

Select a Define format.

● Standard >$accession|$number|$aa|$name

○ Tab Separated >$accession $number $aa $name

○ Space Separated >$accession $number $aa $name

Or enter a custom Define format. Instructions

○ Custom

Select which type of sequence you want:

● Standard ○ Aligned

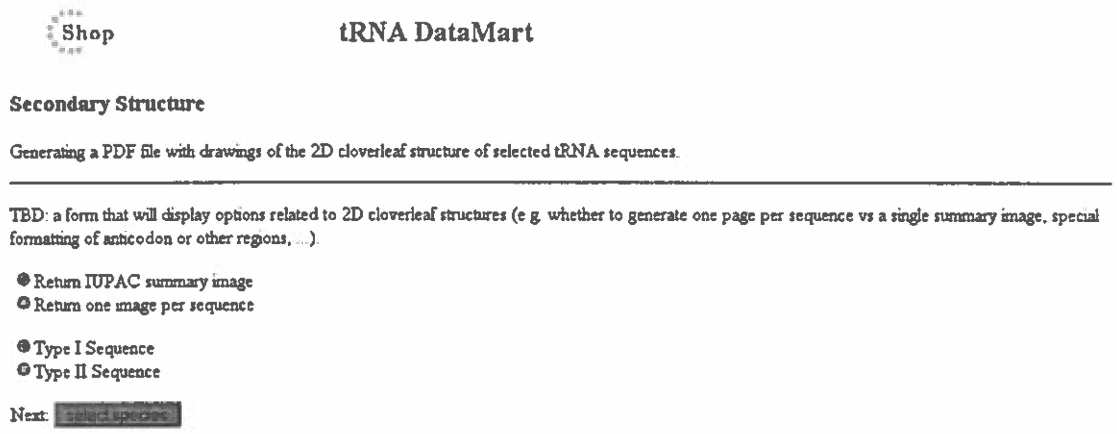
Enter the maximum number of characters to display per line of FASTA output. Enter zero for a single line.

60

Next: select species

Figure 4. The tRNA Mart FASTA Definition Screen.

The most specialized output function of the tRNA Mart to date is its graphical output mechanism. TRNA within a cell is typically shaped in a cloverleaf pattern. The graphical output function produces png images of the requested sequences that consist of the individual nucleotides laid out in the cloverleaf pattern, in a two dimensional approximation of the actual structure of the tRNA molecule. The individual nucleotide characters are superimposed on a dot whose color corresponds to a color arbitrarily assigned to that nucleotide. The most significant option specific to the graphical output mechanism is the IUPAC summary output. Rather than producing one image for each tRNA sequence selected, the graphical output can instead reduce all of the given sequences to a single string of IUPAC code. Where different nucleotides lie at the same position in the sequence, the algorithm simply inserts the IUPAC symbol that represents whichever combination of nucleotides are present. Additionally, the summary option superimposes each IUPAC symbol on a pie chart, rather than a simple dot. Each pie chart depicts the ratio of different nucleotides found at that position in the sequence. This provides the user with a means of rapidly determining the relative compositional structures of multiple tRNA sequences at once.



tRNA DataMart

Secondary Structure

Generating a PDF file with drawings of the 2D cloverleaf structure of selected tRNA sequences.

TBD: a form that will display options related to 2D cloverleaf structures (e.g. whether to generate one page per sequence vs a single summary image, special formatting of anticodon or other regions, ...).

● Return IUPAC summary image
○ Return one image per sequence

● Type I Sequence
○ Type II Sequence

Next: select species

Figure 5. The tRNA Mart tRNA Structure Options Screen

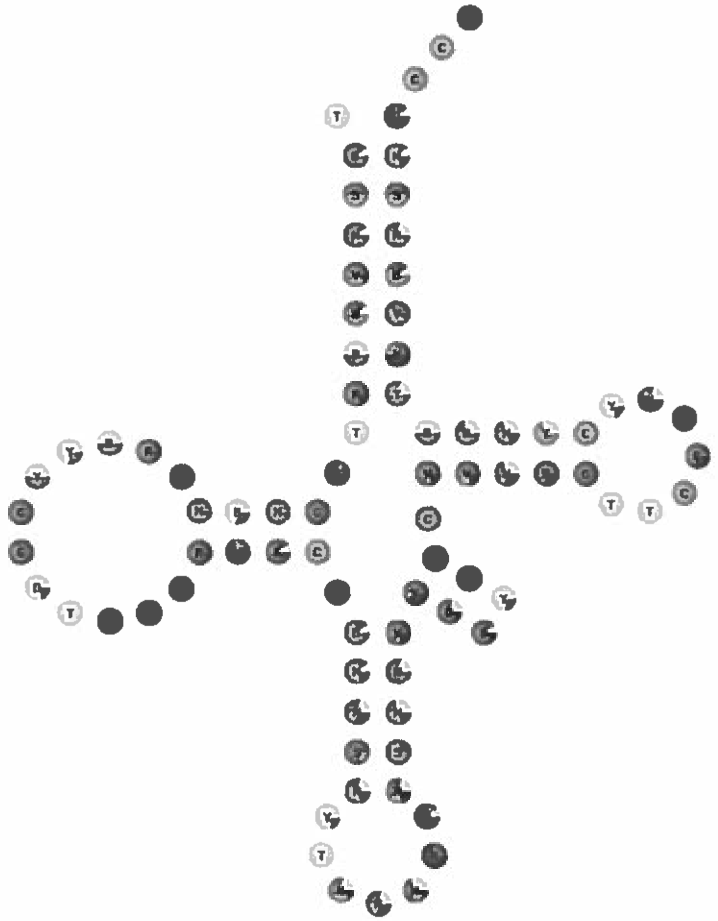
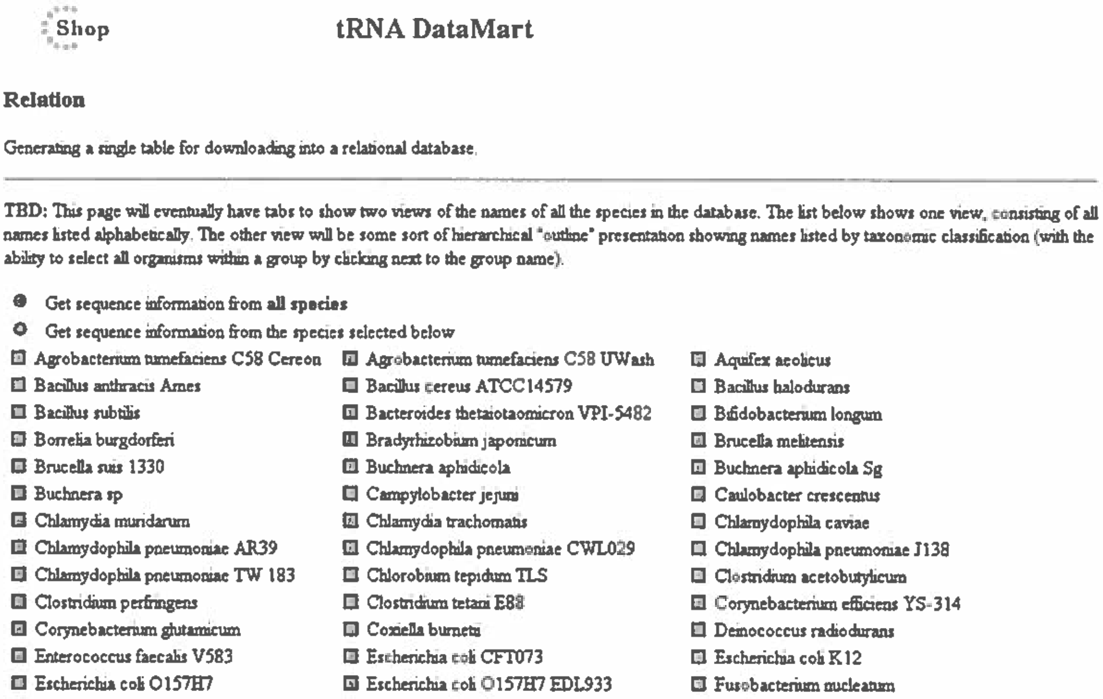


Figure 6. A Summary Image of Cloverleaf tRNA Molecules.

Following the output mechanism specific option pages, the remainder of the options provided by the tRNA Mart are identical regardless of the output mechanism selected. The function of these steps is analogous to the filtration settings in the EnsMart. First there are options for selecting the individual species from which tRNA sequences will be examined. This is presently accomplished via selecting check boxes, one associated with each species. Eventually a tree structure will be used for this step, allowing the selection of multiple related species at once.



tRNA DataMart

Relation

Generating a single table for downloading into a relational database.

TBD: This page will eventually have tabs to show two views of the names of all the species in the database. The list below shows one view, consisting of all names listed alphabetically. The other view will be some sort of hierarchical "outline" presentation showing names listed by taxonomic classification (with the ability to select all organisms within a group by clicking next to the group name).

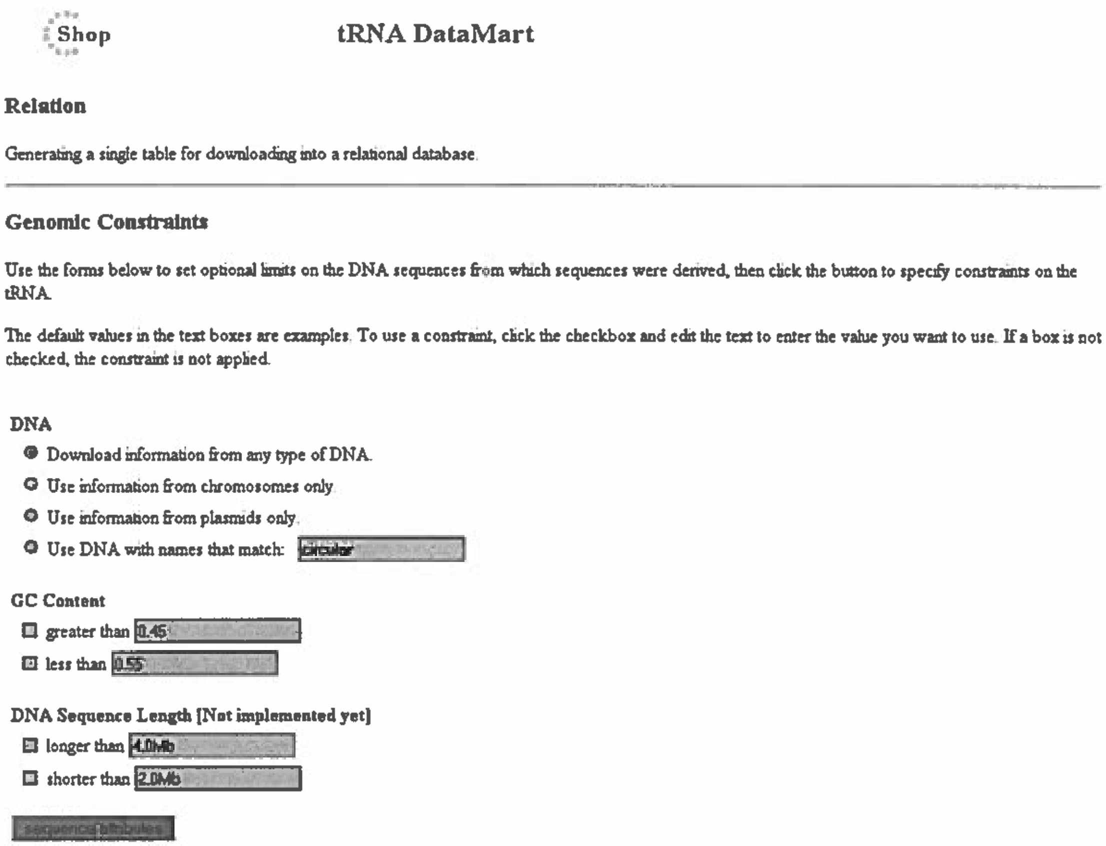
● Get sequence information from **all species**

○ Get sequence information from the species selected below

☐ Agrobacterium tumefaciens C58 Cereon ☐ Agrobacterium tumefaciens C58 UWash ☐ Aquifex aeolicus
☐ Bacillus anthracis Ames ☐ Bacillus cereus ATCC14579 ☐ Bacillus halodurans
☐ Bacillus subtilis ☐ Bacteroides thetaiotaomicron VPI-5482 ☐ Bifidobacterium longum
☐ Borrelia burgdorferi ☐ Bradyrhizobium japonicum ☐ Brucella melitensis
☐ Brucella suis 1330 ☐ Buchnera aphidicola ☐ Buchnera aphidicola Sg
☐ Buchnera sp ☐ Campylobacter jejuni ☐ Caulobacter crescentus
☐ Chlamydia muridarum ☐ Chlamydia trachomatis ☐ Chlamydophila caviae
☐ Chlamydophila pneumoniae AR39 ☐ Chlamydophila pneumoniae CWL029 ☐ Chlamydophila pneumoniae J138
☐ Chlamydophila pneumoniae TW 183 ☐ Chlorobium tepidum TLS ☐ Clostridium acetobutylicum
☐ Clostridium perfringens ☐ Clostridium tetani E88 ☐ Corynebacterium efficiens YS-314
☐ Corynebacterium glutamicum ☐ Coxiella burnetii ☐ Deinococcus radiodurans
☐ Enterococcus faecalis V583 ☐ Escherichia coli CFT073 ☐ Escherichia coli K12
☐ Escherichia coli O157H7 ☐ Escherichia coli O157H7 EDL933 ☐ Fusobacterium nucleatum

Figure 7. The tRNA Mart Species Selection Screen (Truncated for Space).

Next the user can set criteria for the DNA from which the sequences have been extracted. Still to be implemented is a system for restricting the tRNA specific attributes of each sequence returned by the system.



Shop tRNA DataMart

Relation

Generating a single table for downloading into a relational database.

Genomic Constraints

Use the forms below to set optional limits on the DNA sequences from which sequences were derived, then click the button to specify constraints on the tRNA.

The default values in the text boxes are examples. To use a constraint, click the checkbox and edit the text to enter the value you want to use. If a box is not checked, the constraint is not applied.

DNA

- ● Download information from any type of DNA.
- ○ Use information from chromosomes only
- ○ Use information from plasmids only.
- ○ Use DNA with names that match: circular

GC Content

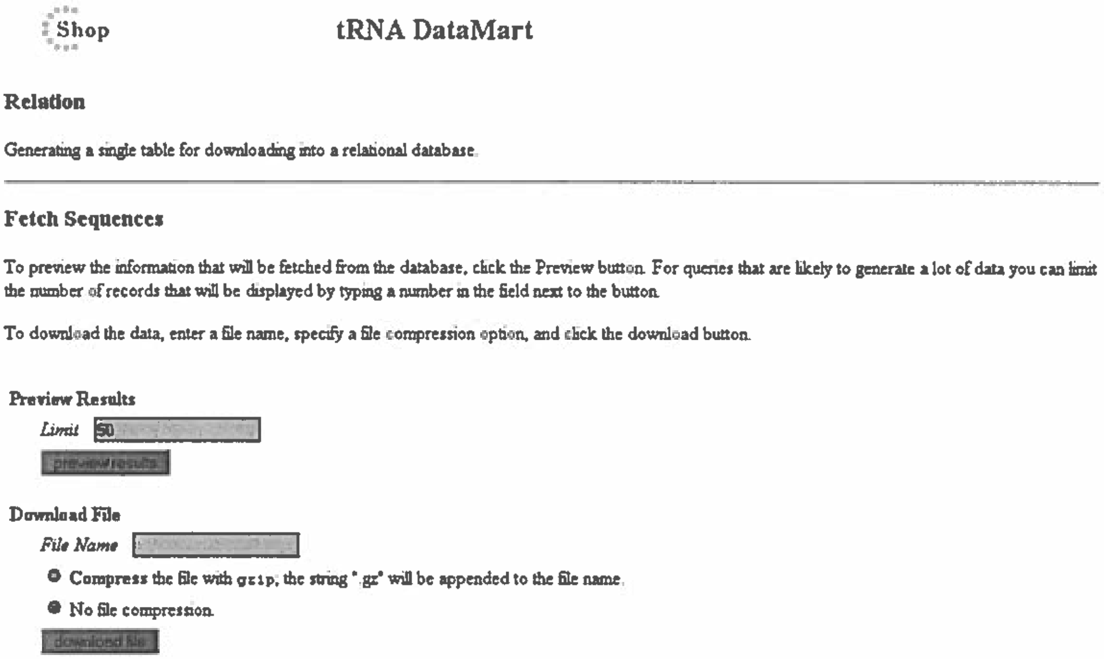
- □ greater than 0.45
- □ less than 0.55

DNA Sequence Length [Not implemented yet]

- □ longer than 4.0Mb
- □ shorter than 2.0Mb

Figure 8. The tRNA Mart Genomic Constraint Selection Screen.

The final output options provided by the tRNA mart include selecting between a full download of the selected sequences in the desired format, or a web based view of a smaller sample. If the former option is selected, the option of zip compression of the output is available.



tRNA DataMart

Relation

Generating a single table for downloading into a relational database.

Fetch Sequences

To preview the information that will be fetched from the database, click the Preview button. For queries that are likely to generate a lot of data you can limit the number of records that will be displayed by typing a number in the field next to the button.

To download the data, enter a file name, specify a file compression option, and click the download button.

Preview Results

Limit 50

Download File

File Name

○ Compress the file with gzip; the string ".gz" will be appended to the file name.

● No file compression.

Figure 9. The tRNA Mart Output Options Screen.

The modular structure of the tRNA Mart has allowed it to maintain a degree of usability, despite it incompleteness of its development cycle. Additional functionality to be implemented will be determined by the requirements of the Saks Group's ongoing research needs and potentially the needs of other research operations should the tRNA Mart display sufficient utility to spread beyond the single venue for which it was initiated.

CHAPTER FIVE

TRNA MART DEVELOPMENT

When I began work on the tRNA Mart the database system, interface structure and table and spreadsheet data access methods were already fully implemented. The FASTA output system provided core functionality, returning sequences with a single default sequence header format. A preexisting system for generating PDF images of non-summarized cloverleaf tRNA visualizations was available but not integrated into the tRNA Mart system itself.

The FASTA output mechanism's development was divided into two primary steps. First the default sequence headers were designed. Each header was a string consisting of a sequence of tokens representing the individual header buttons delineated by a predefined character. The FASTA PHP script would simply query the database for all of the information listed in the header and include each entry in the order provided in the header, separated by the indicated character, prior to the associated tRNA sequence. The custom generated sequences employed the same logic, merely allowing the user to list the header information and delineating characters as desired. Further modifications to FASTA output, such as placing an option for the maximum characters per line of

FASTA sequence were, and will remain, relatively simple to implement by virtue of the lack of systemic dependance on the FASTA module's output.

Development of the tRNA cloverleaf graphical output system started with the creation of an IUPAC code summarization system. The code used to extract the desired aligned sequences was identical to that for the FASTA output system. However, instead of outputting each sequence individually the IUPAC summarizer used a series of text parsing operations, made relatively simple by PHP's innate capabilities in that regard, to compare the the each character at the same position in each sequence and determine which IUPAC character represented all of the nucleotides present at that position. Keeping track of the total number of sequences and the total number of each type of nucleotide at each position made it possible to determine the percentage of each nucleotide present at each position.

Only after the IUPAC summarizer was complete did work start on integrating the graphical output system. The existing PDF based cloverleaf generation system was converted to PHP's native graphics output system. The coordinates of the cloverleaf pattern were stored on the tRNA Mart's associated database, thus the graphical manipulations consisted primarily of placing the relevant pie charts and IUPAC code letters at the indicated positions. The non summarized cloverleaf outputs were graphically more simple, requiring only solid circles to indicate the nucleotide positions, but the placement of multiple such images on the same output page required more advanced PHP based file and page control than the single image output by the IUPAC summarized cloverleaf images.

As with the FASTA output system, once the core functionality of the cloverleaf output module was complete, implementation of additional functionality such as including textual descriptions of the sequence or summarized sequences would be fairly simple. A more complicated potential future elaboration on the cloverleaf output generator is the use of image maps to give more specific information on the individual properties of each nucleotide in the sequence or sequences depicted.

CHAPTER SIX

CONCLUSION

The utility of data marts is not to be underestimated. The success of the tRNA Mart clearly illustrates many of the advantages of data mart deployment in bioinformatic applications. The utility of the application from its early stages of development, and its ease of extensibility demonstrate the viability of personalized data mart construction.

The technical requirements to make tRNA Mart entirely modular and generic from the end user's point of view have not yet been investigated. However the modular development style both makes this a simpler proposition for future development. It also grants those with relatively little previous exposure to the system the ability to adjust it to suit changing requirements without excessive effort or retreading of old ground.

The tRNA Mart's essential usefulness is unlikely to be compromised by future developments either in computational or biological capabilities and requirements. The ease with which its operational parameters can be adjusted makes the maintenance phase of the system's life cycle effectively perpetual, so long as its essential function of retrieval and coherent presentation of bioinformatic data is required.

The proliferation of data marts as a cornerstone of computational science is a trend that is likely to continue. While venue specific optimizations to such interfaces will be the norm, it remains important to consider the generic data mart as well. Improvements generally applicable to the concept, though arguably difficult to produce because of the essentially simple nature of a data mart, will benefit wider and wider ranges of users and fields as data marts become increasingly ubiquitous.

BIBLIOGRAPHY

1. *BLAST Information.* NCBI. 2004 July 18. <http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/information3.html>

2. Campbel, Reece, Mitchell. *Biology, Fifth Edition*. Menlo Park, CA: Benjamin/Cummings

3. Conery, John, et al. *tRNA Data Mart*. University of Oregon. 2004 Aug 22. <http://teleost.cs.uoregon.edu/tRNAmart/>

4. *Data Warehouse*. Wikipedia. 2004 Jul 19. <http://en.wikipedia.org/wiki/Data_warehouse>

5. Huskey, Robert J. *The Genetic Code*. University of Virginia. 2004 Aug 17. <http://www.people.virginia.edu/~rjh9u/code.html>

6. Kasprzyk A, Keefe D, Smedley D, London D, Spooner W, Melsopp C, Hammond M, Rocca-Serra P, Cox T, Birney E. *EnsMart: a generic system for fast and flexible access to biological data. Genome Research*. 2004 Jan 14. pp160-9.

7. McGinnis S, Madden TL. *BLAST: at the core of a powerful and diverse set of sequence analysis tools*. Nucleic Acids Research. 2004 Jul 1. pp20-5.

8. Moss, G. P. *Nomenclature for Incompletely Specified Bases in Nucleic Acid Sequences*. Queen Mary University of London. 2004 Jul 18. <http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html>

9. Saks, Margaret E. *Welcome to the Saks Research Group*. University of Oregon. 2004 Aug 15. <http://www.molbio.uoregon.edu/psaks/>