



Identifying Candidate Disease Genes with High-Performance Computing

TERRY A. BRAUN

tabraun@eng.uiowa.edu

Coordinated Laboratory for Computational Genomics, Department of Ophthalmology, Department of Biomedical Engineering, The University of Iowa, Iowa City, Iowa, USA

TODD E. SCHEETZ

Coordinated Laboratory for Computational Genomics, The University of Iowa, Iowa City, Iowa, USA

GREGG WEBSTER

Coordinated Laboratory for Computational Genomics, The University of Iowa, Iowa City, Iowa, USA

ABE CLARK

Alcon Laboratories, Fort Worth, Texas

EDWIN M. STONE

Department of Ophthalmology, The University of Iowa, Iowa City, Iowa, USA

VAL C. SHEFFIELD

Department of Pediatrics, The University of Iowa, Iowa City, Iowa, USA

THOMAS L. CASAVANT

Coordinated Laboratory for Computational Genomics, Department of Electrical and Computer Engineering, Department of Biomedical Engineering, The University of Iowa, Iowa City, Iowa, USA

Abstract. The publicly-funded effort to read the complete nucleotide sequence of the human genome, the human genome project (HGP), is nearing completion of the approximately three billion nucleotides of the human genome. In addition, several valuable sources of information have been developed as direct and indirect results of the HGP. These include the genome sequencing of model organisms (*Escherichia coli*, *Saccharomyces cerevisiae*, the fruit fly *Drosophila melanogaster*, the worm *Caenorhabditis elegans*, and the laboratory mouse), gene discovery projects (expressed sequence tags and full-length), and new high-throughput expression analyzes. These resources are invaluable in identifying the transcriptome and proteome—the set of transcribed and translated sequences. However, the bulk of the effort still remains—to identify the functional and structural elements contained within gene sequences. Addressing these challenges requires the use of high-performance computing. There are currently hundreds of databases containing biological information that may contain data relevant to the identification of disease-causing genes. Knowledge discovery using these databases holds enormous potential, if sufficient computing resources are utilized to process the overwhelming amounts of data. We are developing a system to acquire and mine data from a subset of these databases to aid our efforts to identify disease genes. A high performance cluster of Linux of workstations is used to perform distributed sequence alignments as part of our analysis and processing. This system has been used to mine the GeneMap99 database within specific genomic intervals to identify potential candidate disease genes associated with Bardet-Biedl syndrome (BBS).

Keywords: high-performance computing, disease gene discovery, bioinformatics, knowledge discovery, biological databases

1. Introduction

The primary objectives of the human genome project (HGP) [33] are to construct high resolution genetic and physical maps, to determine the complete nucleotide sequence of human DNA, and to identify and localize a comprehensive set of genes within the human genome. The HGP also included pilot studies with similar aims for the genomes of several strategic model organisms used extensively in research, as well as a component to analyze the ethical, legal, and social implications of genetic and genomic information [33].

A direct consequence of the HGP is the existence of a growing number of data collections that are used by investigators in numerous fields to aid in their efforts to collect information relevant to specific biological problems. These resources make it possible to identify disease genes, but the volume and complexity of these resources present new challenges. The amount of data available makes the utilization and integration of these resources impractical without the use of automated tools and software applications. This trend of increasing amounts of information will only continue as the completed human genome sequence makes possible the creation of high resolution maps, and similar sequencing and mapping projects in model organisms enables the construction of high resolution comparative maps between organisms. Other high-throughput genomic technologies, such as micro-array expression assays that are directed at solving biological problems on a genome scale continue to evolve. This paper proposes an architecture for developing high-performance computing solutions to discover and prioritize candidate disease genes using automation to identify and acquire associated data.

2. Disease gene identification

Gene discovery is defined as the process of finding novel genes from raw biological data. Expressed sequence tag (EST) sequencing projects and computational gene prediction from genomic sequence are common formats for gene discovery projects. The ultimate goal is to find mutations within genes that cause disease, or sequence variants that predispose one to disease. However, with the size and complexity of the human genome and the cost (time and money) of screening* genes, it is impractical to screen every gene in every population for a given disease. Instead, genes are chosen for screening based on available information. The idea is to prioritize genes such that those most likely to be involved with the disease are screened first. The information that determines the selection of genes may be based on data from local experiments, or from existing data resources. However, finding, obtaining, and integrating this information may be difficult due to the amounts and complexity of

Screening is the process of verifying that a gene contains deleterious mutations, such as through direct sequencing of affected individuals.

data. As methods to predict gene-disease association improve, fewer genes will need to be screened. It is the process of *candidate disease gene nomination* that recognizes genes with an increased likelihood of being associated with a disease, based on available data. From this nomination process, a list of candidate disease genes may be compiled. In the default case there is no data to differentiate likely genes from unlikely genes, and therefore, every discovered gene is a candidate.

A system that mines biological databases for disease gene discovery has enormous commercial application. The knowledge and understanding of disease genes and biology is valuable in identifying new drugs. In 1999, the top ten pharmaceutical companies spent \$25.4 billion in research and development to identify new drugs [1], and discovering genes associated with disease is an important approach for identifying new drug targets. A better understanding of how gene products function and how mutations affect gene function may also lead to more effective drugs. Although the development and testing of new drugs is a thorough, lengthy, and expensive process, the number of deaths in the United States from adverse drug reactions to prescription drugs was 106,000 (in 1994), the fourth leading cause of death [19].

Typically, the disease gene identification process begins with the recognition of a disease or observable phenotype(s) that may be used to find correlations and/or associations with presumably genetic loci. Various laboratory and computational techniques and methods are employed to generate experimental data that may be used to eventually localize an interval or gene; and this experimental data is used to identify existing relevant information and corroborate the experimental evidence. Disease gene identification is typically labor intensive involving laboratory experiments to corroborate or disprove the hypothesis of a nominated candidate disease gene as being associated with the disease.

The disease gene identification process may utilize multiple iterations of candidate disease gene nomination. For example, linkage information may be used to nominate candidate disease genes based on position within the genome. Further consideration of the functional roles of protein products to specific tissues may implicate expression information (such as looking for ESTs that are only expressed in specific tissues). The insight gained from the expression information might then be used to design experiments (biological or computational) to nominate other related candidate disease genes. Pathway information is an obvious example of data for nominating related genes.

The availability of human genomic sequence has allowed the discovery and localization of thousands of genes. Prior to the availability of this resource, localizing a gene within the genome was a difficult task, and determining the genomic structure (exon and intron boundaries) was very time consuming and expensive. However, with the availability of genomic resources, the challenge has become reversed. Now, there are potentially millions of nucleotides of sequence available for a disease interval; and within these multi-megabase regions are sometimes several hundreds of genes, that may all be candidate disease genes—any of which may contain rare mutations that cause a disease. Because of the vast volume and complexity of available biological data, candidate disease gene nomination and prioritization is feasible only with sophisticated computer hardware and software.

Perhaps the most difficult challenge, is the efficient integration of very disparate types of data and the ability to infer meaningful relationships.

2.1. *Simple and complex diseases*

The human gene mutation database (HGMD) [34] lists more than 1,000 genes for which disease-causing mutations have been identified. The vast majority of these genes are associated with simple disorders, in which only one gene is involved. Yet the number of genes involved in complex disorders, diseases that are caused by multiple genes, is likely to greatly exceed this number because of the complexity of biological systems and the observed interaction of gene products in pathways. Complex diseases are caused by the interaction of the products of multiple genes and the environment. Demonstrating that a gene is the cause of a simple disease is relatively straightforward. The presence of mutations in a gene is strongly correlated to disease status. The process is further complicated by properties of disease such as reduced penetrance (the disorder associated to a mutation is not expressed), where having mutations is necessary but not sufficient. The combinatorics of gene-disease interactions make disease gene discovery significantly more challenging for complex disease. At one end of the spectrum is the scenario where mutations in any of several genes are sufficient to cause the disease. If one of these genes is assessed for mutations across a population of affected individuals, many of the individuals will have the disease, but will lack an observable mutation in the gene being assessed. This reduces the analytic power of the assessment. Combinations where multiple genes must harbor mutations make the analysis even more challenging. In this case, the number of observed mutations that are present in unaffected individuals increases dramatically, further reducing the differences between affected and unaffected individuals for a given gene.

Successful strategies and techniques for identifying interacting genes in complex disorders remain unrealized. However, the ability to utilize and integrate the available biological information and data will unquestionably play as important of a role in complex disorders, as it already has in simple disorders.

3. Existing heterogeneous complex data resources

There are numerous heterogeneous complex data sets available that may be useful in disease gene discovery. The various combinations with which these resources may be applied to the disease gene discovery problem are too numerous to enumerate. In addition, new resources frequently become available making others obsolete. Regardless of how resources and data are utilized, it is useful to define generic categories to arrange the numerous data resources for discussion. The following list of data resources define primary categories useful for disease gene discovery, identification of related information and data, and candidate gene prioritization. These are: *maps*, *sequence*, *expression*, *pathway*, *function*, *disease gene localization*, and *scientific literature*. Although there are too many data resources to exhaustively

enumerate (or utilize) every one, specific instances within each category have proven useful in the efforts to locate disease genes, and are discussed in Section 6. This list of categories is used in Section 5 to develop an architecture for utilizing some of the available data resources to identify and prioritize candidate disease genes. One of the major challenges is identifying which types of information contain the most useful for implicating candidate genes and gene sequences.

Map resources provide systematic localization of genetic loci. These include radiation-hybrid maps (GeneMap99) [31], genetic maps (Marshfield) [38], and comparative maps (Mouse Genome Database) [39]. *Sequence* refers to the primary data of nucleotides in DNA and RNA, and amino acids in proteins. Examples of sequence data include nucleotide and amino-acid sequences from various organisms, as well as relationships identified through sequence comparisons (i.e., homology). *Expression* resources provide information pertaining to the products of genes that serve the functional role in biological systems—usually as proteins or mRNA transcripts. Some examples of expression data include ESTs, in-situ hybridizations, microarray hybridization experiments, and Northern blots. *Pathway information* represents the networks of interaction between genes and proteins that exist in biological systems. Information may be obtained about interacting products by their relationships and placement in known pathways (e.g., galactose metabolism). *Functional data* refers to the behavior of biological components (typically, but not limited to genes and proteins), and the information derived from these components under specific conditions. This includes protein-protein interactions as observed in yeast-two-hybrids and co-immunoprecipitation, as well as actual enzymatic functions in cellular metabolism. *Disease gene localization* techniques are used to correlate phenotypes with genetic elements, and localize the genetic elements within a genome. For example, genotyping is a widely used technique that uses genetic markers to measure inheritance patterns in kindreds. Statistical methods are then used to calculate the likelihood of specific genetic regions segregating with the disease given the phenotypic observations. Finally, there is the *scientific literature*. There is great interest in mining this resource with many different approaches [7, 17], but it presents unique challenges. Most notable are the parsing of natural language, acquisition of electronic text and figures, deciphering figures for information, and assigning significance and associations to ideas based on word distributions and context.

Clearly there is overlap between these primary resources. For example, there exists data for the *MKKS* gene [24] that may be characterized as map, sequence, expression, and functional; map data because the gene is accurately localized to chromosome 20, sequence because the coding sequences and gene structure are known, expression data because an expressed product is identified, and functional data from the known and identified protein domains. The purpose of these categories is not to enumerate and categorize all specific instances of data resources, but to recognize that there are diverse, interrelated sources of data that need to be integrated, from which useful and insightful relationships may be inferred. These definitions are used as generic terms that may apply to many related diverse examples. Because new data sources are constantly becoming available, often with new types of data, the capability of the system to deal with new data types will be critical.

3.1. *Potential value in utilizing diverse resources*

The availability of complete and working draft sequence in conjunction with ESTs, predictive gene methods, and full- or partial-length cDNAs, is a principle example of the potential value in combining and utilizing diverse data. Without the EST sequence, the location of a particular gene within genomic sequence is difficult to determine. Gene prediction applications suffer from the inability to accurately identify all exons of a gene—being either too sensitive and over predicting genes, or too discriminating and missing legitimate coding regions. The measured accuracy of several applications puts the accuracy rate in the range of 0.60 to 0.70 measured by the correlation coefficient [9]. Without a considerably larger segment of genomic sequence, the complete gene structure of a gene represented by an EST may be very difficult to determine. However, using sequence alignments between the EST and the genomic sequence may easily and quickly lead to knowledge of the location and genomic structure of the gene that could not be determined as efficiently without both pieces of information.

This simple example shows the potential value in combining data resources. However, two complicating distinctions are not illustrated in the previous example. and prioritization has complicating distinctions that are not illustrated by the previous example. These distinctions are that (1) determining if relevant data exists is often impractical without sophisticated software automation (i.e., what data sources are useful), and (2) acquiring and analyzing the relevant data without automated computational techniques is impractical. Often the determination of relevance cannot be examined until the data is acquired and analyzed which is a major investment in time and resources, further complicating the process of disease gene discovery.

4. **Motivation for computational methods**

The motivation for using computational methods for disease gene discovery, information identification, and candidate gene prioritization is based on two observations. First is the potential value of searching, filtering, and acquiring relevant portions of the overwhelming amount of diverse biological data through automation. Combining and integrating this data may lead to new insights and directions for disease gene discovery. Thousands of disease loci have been identified and mapped to distinct genomic intervals for which the causative gene remains unknown [41]. Due to limited time and resources, it is impractical to manually gather, assemble and integrate, format, process, analyze, and manage all of the related data that may provide the single key piece of information or relationship that leads to the discovery of a disease gene. The process of disease gene discovery, information identification, and acquisition often leads to further additional data (i.e., or feedback) that is integrated through multiple iterations of this process. The second observation is that the existing information is very dynamic with additions, deletions, modifications, and corrections. Due to this dynamic nature, the various steps and analyses of disease gene discovery, information identification, and

candidate gene prioritization may be replicated many times. Attempting to explore multiple loci across different intervals, accumulating diverse biological and genetic data for various relationships and temporally displaced data sets, is a problem that requires software automation.

Although vast biological databases exist, this is not the same as having access to all of the data contained in such a database. The very size that makes such databases rich with valuable information requires the existence of an interface. Often the interface itself limits access to the data set. With limited access to the data, certain desired analyses may not be available. Automated computational methods may be used to solve these types of challenges. There are two strategies to address this: (1) acquire the relevant data from the database(s) and perform the analysis locally, (2) develop computational analytical components to extract (or perform analysis on) a subset of the database that makes the analysis feasible. The first strategy of simply acquiring the data and performing analyzes locally is intuitively appealing because of its simplicity. The first issue to this strategy is whether it is feasible to access and acquire the data due to the structure and interface of the database. For example, GeneMap99 [31] is a radiation hybrid map. This may be used as a resource to identify a list of ESTs mapped to a specific genomic interval with some degree of confidence. This list of ESTs, and the associated sequence, may then be used to enumerate known genes via UniGene clusters [50] by BLASTing the EST sequences against EST databases. An application to perform this form of knowledge discovery is described in Section 6.1. The list of elements mapped on GeneMap99 may be acquired for local processing, but the complete collection of cross-indexed data resources through the browser interface are not contained in the data available for download.

Another example includes *genome browsers*, interfaces to genomic data, such as genomic sequence, gene sequences, mRNAs, ESTs, proteins, and other types of annotation. The genome browser at UCSC [48] provides a convenient interface to the human genome. Investigators routinely review genes and their associated sequences (mRNAs, ESTs, genomic context, and proteins) for a given map position, or interval. An interval for the BBS3 locus [28] was narrowed to a 6 cM (15 Mb) interval on chromosome 3. The browser is capable of quickly showing that there are 54 annotated (RefSeq) [44] genes in the interval. However, this particular browser cannot easily provide the coding sequence for all 54 genes, flanking sequences, and all ESTs and mRNAs associated with each gene (short of manually following hundreds of links for each gene). It is also relatively difficult to search the ESTs mapped to this interval for tissue library sources of cDNAs, which may be used to refine the list of 54 candidate genes. These functionalities are not provided within the browser, nor is it likely that the hosting site could provide enough computational resources to support the level of processing required for these types of analyses.

5. Meta tool architecture

This section describes a system architecture for integrating and utilizing contemporary and novel computational methods. The ultimate purpose is to identify

disease genes. Computational methods are used to extract biological information from multiple heterogeneous sources to nominate candidate disease genes and compile gene-related information. This information enables users to prioritize candidate disease genes.

The existing set of biological data resources and biological computation infrastructure are enormous and growing, with new, larger data resources are likely. As an example, the Molecular Biology Database Collection is a compilation of key on-line databases important to the biological community, and currently has 281 entries [5]. Consider just one entry of this compilation. UniGene [50] is a system for partitioning GenBank [32] sequences into clusters that are non-redundant and ideally each cluster represents one gene. Each cluster consists of sequences from transcripts of a gene, and additional information is cross-referenced when accessed through NCBI. This includes expression information such as tissue type and expression level from SAGE [46]; map information such as chromosome localization; and sequence information such as identified protein similarity to other model organisms. Millions of novel ESTs have been included making the collection useful for gene discovery [50]. As of February of 2003, the size of this database for human sequences is 3×10^9 bytes, representing 3,966,221 sequences and 128,826 clusters. Assuming UniGene is a representative data collection, then the list represents 8.43×10^{11} bytes of data (approximately 800 GB). Another catalog of databases [18] provides access to 350 distinct biological databases. This vast volume of information and numerous diverse data sources is what makes candidate disease gene nomination and prioritization challenging. First, each data source has potentially unique (and incompatible) data formats. Different data formats may require custom software “adapters” to access and manipulate data universally and independent of format, i.e., *data specificity*. Second, and perhaps even more difficult, is integration and correlation of disparate types of data, i.e., *data integration*. The scope of these two problems in light of the inherent data size and complexity problem motivates a modular solution to data specificity and data integration. A modular architecture enables the partitioning of specific problems and solutions into components that may be designed, implemented, and utilized independently of each other, and yet may contribute to the larger problems of data specificity and integration. New modules may be integrated as technology and formats change. An architecture of this nature provides a structure for developers and expert users to address some of the problems of specificity and integration by providing a model that is modular and adaptable to available and future data.

Figure 1 shows a graphical representation of the system architecture that defines a structure for integrating, developing, and applying high performance computational methods to mine existing biological data to (1) acquire data, (2) find interrelated and gene-related information, (3) filter data, (4) integrate information, (5) nominate candidate disease genes, and (6) prioritize disease genes.

The modules shown in Figure 1 correspond to the primary categories presented in Section 3. Each module is one or more self-contained software application(s) designed and implemented to solve specific problems of candidate disease gene nomination. The attraction of this scheme is the ability to incorporate and utilize existing applications and tools where significant time and effort has gone into

IDENTIFYING CANDIDATE DISEASE GENES

15

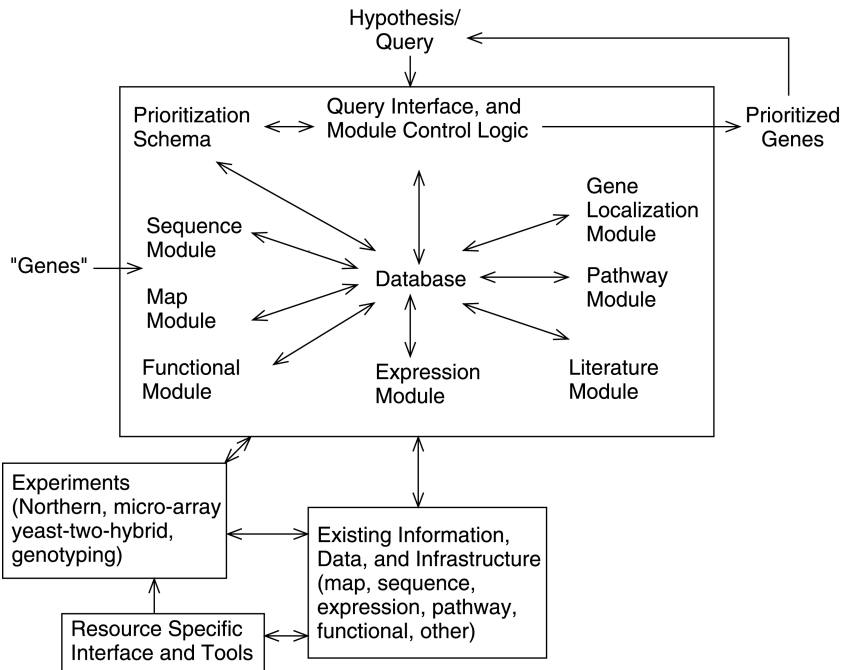


Figure 1. “Gene CaDi”—Candidate Disease Gene nomination and prioritization. Software tool architecture for developing, integrating, and utilizing high-performance computational methods for utilizing existing and future biological databases and computational methods to nominate candidate disease genes, locate interrelated information, filter extraneous data, and prioritize candidate disease genes.

development, and also to be able to develop applications and tools where existing computational methods are deficient. These custom applications may encompass novel analyzes, automation to tedious interfaces and applications, or simply adapters or “wrappers” to format data for different applications or tools. For example, the *Sequence Module* manipulates nucleotide and protein sequence. This includes sequence comparisons, the parsing and storage of sequence comparison results, sequence description and annotation extraction, and sequence acquisition. Perhaps the most popular and widely used sequence comparison application is BLAST [2]. There is no intent to duplicate existing and refined applications, such as BLAST. However, this architecture provides a structure for combining existing and novel applications to answer specific biological questions (including candidate disease gene nomination) that are difficult to address due to limitations of existing infrastructure and volume of information.

The applications of the *Map Module* have the capacity to extract mapped sequence elements from map resources (e.g., GeneMap99). The *Expression Module* incorporates diverse information from sources that quantify gene products and the intermediates between transcription and proteins. This includes levels of protein expression or mRNA transcripts, localization of proteins and transcripts, and

differential expression and differential transcription under various conditions and parameters. The applications of the *Functional Module* obtain information pertaining to the function of genes and proteins. This includes secondary structure predictions and comparisons, tertiary structure searches and comparisons, and information describing protein-protein interactions. This may also include other techniques such as detailed domain searches with hidden Markov models (HMMs) [35]. The *Disease Gene Localization Module* uses applications and techniques to find associations between disease phenotypes and genomic regions. Examples include the genotyping of microsatellite markers, linkage analysis, association studies, homozygosity mapping, and DNA pooling. Applications of the *Pathway Module* extract information from databases that describe networks of interactions between genes and gene products, such as the *Kyoto Encyclopedia of Genes and Genomes* (KEGG) [37]. The *Literature Module* includes applications from the efforts to extract information through association and context in the scientific literature.

Each module has a set of capabilities displayed as options to a query interface. For example, the *Map Module* has the capability of acquiring mapped sequence elements based on map locations. A user specifies map locations depending on the requirements of the particular maps available to the system. For GeneMap99 [31] framework markers define intervals. Therefore a query interface informs the user about the *Map Module's* capability to access GeneMap99 by displaying this option. Also, the interface requires the user to provide the parameters needed by the module—in this case interval flanking markers. The *Map Module* then acquires all available sequence from the specified map for the given markers. Next, the *Sequence Module* may be employed to identify sequence and protein similarity. Again, the interface would list available sequence comparison and search capabilities such as BLAST [2] or HMMER [12].

5.1. Database

The database portrayed in the center of Figure 1 indicates a need for memory and the ability to store data and the state of an analysis for later recall. Depending on the requirements of each individual module, it is likely that modules will have local versions of data for processing and manipulations. However, more persistent system memory is required for various reasons. First, it is necessary to store results that are not easily reproducible or require manual and tedious curation. Also, for reasons discussed below, the storage of results may be needed for later analyses. For example, a hypothetical region of interest may contain 10 Mb of sequence. To identify the distinct UniGene [50] clusters within this interval, the entire interval could be BLASTed against a database of human ESTs [29]. Such a BLAST analysis is likely to generate thousands of regions of similarity, identifying hundreds of UniGene clusters. Rather than replicate this computationally intensive and time consuming analysis each time a gene (represented by a cluster) is chosen to be screened for mutations, storing the list of UniGene clusters is a more efficient solution. A second motivation for system memory is the temporal nature of analyses and data. The ability to identify new and relevant information appearing periodically

is desirable. This is related to the third reason to have system memory, and that is the dynamic nature of data. Not only does new information become available, but alterations and corrections to data are also common. Instead of reproducing the steps preceding a particular analysis, intermediate results may be stored and used to discover new and altered information. A fourth reason for system memory is the ability to store and share state-full information between investigators. Often the details of what data has been analyzed may be lost over time, with the removal and addition of different investigators, and in the vast amount of data and analyses involved. The storage of state and previous results is similar in concept to an *electronic laboratory notebook*. The final reason that system memory is useful is that certain analyses are conducive to automated or periodic processing, and these analysis require the storage of query information. For example, candidate genes with no known homologs across species may exist. It is useful to be able to periodically examine available sequence databases in attempts to quickly identify new homologs.

5.2. Existing information, data, and infrastructure

The motivation for this architecture is the data resources and infrastructure that already exist for vast collections of biological data. Each module represents independent applications with the capacity to repeatedly access specific instances of biological data and acquire or extract information as specifically designed for that module. In many cases the interface to available data already exists and is resource specific, such as the capability in GeneMap99 to identify and locate a single locus to which many ESTs may have been mapped. However a module may simply take advantage of existing interfaces to expand the amount of available information and data. For example, Section 6 describes the implementation of a module to repeatedly access GeneMap99 to identify multiple loci and ESTs in an interval.

5.3. Experiments

Experimental results (e.g., biological, biochemical, and genetic experiments—i.e., “benchwork”) are very important to the candidate disease gene nomination process. Experimental results may reflect very crucial and focused data not generally amenable to specific databases. Yet, this information may be key in identifying data and associations that lead to strong disease gene candidates. Such information may be vital to the filtering of extraneous data, implicate new protein functions, or supply data for a prioritization of candidate disease genes. Analyses performed by modules may revolve around the data and insight generated by a specific experiment and may be the prime motivation for queries.

5.4. Resource specific tools

Many of the existing databases have resource specific interfaces and tools. These are extremely useful to investigators who utilize these tools to obtain information and data from biological databases. Often these interfaces are designed to accommodate the maximum number of investigators with varied ranges of expertise. However, the users are constrained by the limits of the interface. For example, the interface to GeneMap99 [31] allows users to search for loci located on the map. Electronic copies of the list of sequence elements mapped are also available. The interface provides links to additional information and sequence throughout GenBank. However, there are no means for making hundreds of queries other than to manually enter each query. Although this is a useful mode of querying to obtain lists of mapped sequence elements for genomic intervals, it requires an impractical amount of network bandwidth and processor capacity from the data provider. The web accessible interface to a BLAST [2] server at NCBI [40] is another example of a useful and powerful tool. Users may perform sequence comparisons between a single sequence and databases of sequence to identify similar sequences. Again, the computational and network resources are not provided by the data provider to perform hundreds or thousands of sequence comparisons. One solution is to obtain the freely available BLAST software and databases to run locally, thus the only limit to the number of sequence comparisons is local computational capacity. The last example is the UniGene [50] database. The sequence comparison results from BLAST often reference UniGene clusters of unique genes. BLAST may be used to identify ESTs located in large (megabase) segments of genomic sequence. The sequence comparison results from BLAST analyses may be used to identify UniGene clusters by identifying an EST that has been placed in the cluster representation of a gene. Thus it is useful to obtain the list of UniGene clusters represented by thousands of BLAST sequence comparisons to identify genes in a genomic interval. However, no automated mechanism is provided to obtain this list of potentially hundreds of UniGene clusters from thousands of BLAST alignments to ESTs.

6. Results

6.1. Map module

Interval Search as an example of a map module. It mines specific maps for sequence elements with known map locations. The map used by this module is GeneMap99 [31] at NCBI [40], which is a radiation hybrid map of markers, ESTs, and sequence tagged sites (STSs). Prior to the availability of assembled and finished genomic sequence, this module provided the capability to identify and acquire sequence mapped within specific intervals.

A *region of interest* is first implicated through experimental procedures such as genotyping and linkage analysis or homozygosity mapping. An existing framework marker specifies this region of interest on the map for the module. NCBI divides genome regions into intervals or “bins” with framework markers flanking the ends

IDENTIFYING CANDIDATE DISEASE GENES

19

of an interval. The module, an application written in Java [36], communicates with the NCBI webserver and obtains a list of all markers, ESTs, and genes within the specified interval. This is accomplished by accessing each sequence element and each adjacent element on the map until all elements are acquired for a specified interval. Each of the elements in the interval contains a “locus ID,” a number assigned by NCBI that serves as a unique identifier for a locus and places it on the map. Thus for each locus ID, there may be sequence available from mRNAs, ESTs, and genomic sequence (both working draft and finished). Each sequence is identified by an “accession number,” an identifier that NCBI assigns to each sequence submitted to the GenBank database [32]. There may be hundreds of sequences associated with each map location depending on the number of identified sequences. The module downloads each sequence from the NCBI database, concatenating sequence from the same map location into one file. A local directory structure is built to hold the markers, locus IDs, and sequences. A representation of this information is illustrated in Figure 2.

Multiple sequences mapped to the same location represent different reads and different sources of sequence. In the case of ESTs, sequencing of a gene may yield 300 bases, while a second sequence from the same gene may yield 500 bases, which may or may not overlap the initial sequence. The longest piece of available sequence can then be obtained from multiple ESTs mapped to the same location by assembling the different overlapping sequences. In addition, where there are several hundred sequences available, the process of assembling can reduce this large number of reads to a relatively few number of distinct representative sequences. Applications such as *phrap* (fragment assembly program) [43] are available that are designed to assemble overlapping sequences. However, genomic sequence requires that



Figure 2. Screen shot of Interval Search application and representation of data acquired from GeneMap99 and written to the local file system. Interval Search identifies and acquires each “locus identity” mapped to an interval in the genome designated by the proximal marker—in this example D15S160. The interval is defined as everything mapped between the proximal marker and the next framework (distal) marker. Each locus identity may contain several hundreds of mapped sequence elements (ESTs, STS, and markers) represented by “accession numbers.”

computational approaches to sequence assembly be efficient, since the complexity is a function of the number of nucleotides. The process of clustering can also produce non-redundant sequence sets, but typically does not compute an assembled sequence. This can result in a significant saving in memory. This system uses the *UIcluster* [49] to select the longest representative sequence for loci that contain multiple sequence reads.

The processing accomplished by this module results in a set of sequences mapped to a particular interval. This is particularly useful for unfinished and unassembled regions of the human genome. This set of sequence may then be used to discover and identify additional sequences that are not already mapped.

There are two primary deficiencies with the design of this module. First, it relies heavily on network and computational resources of the data provider—NCBI. In fact, NCBI monitors the number and frequency of network connections, and the amount of computational usage of users who try to consume excessive amounts of resources. Second, this module was designed, implemented, and applied before more than 10% of human draft and finished sequences were available. Without large genomic sequences to anchor markers, ESTs, and acquired intervening sequences, there was no practical method to determine if acquired sequences were actually within the interval. A key motivation for the design of this module was to identify new sequences as they were deposited into databases. Now that the majority of draft sequence available, this mode of query is not as valuable. A replacement module is being developed that solves these deficiencies by aligning UniGene EST sequences to genomic sequence to identify the genes represented by each cluster, and to mine expression information in the form of tissue library sources.

6.2. Sequence module

The sequence and map data generated by the Interval Search module of Section 6.1 is essential for identifying and acquiring short sequences mapped with high resolution. However, the majority of sequence spanning these mapped sequence elements remain unutilized. To remedy this, an existing sequence comparison application and databases were utilized as a sequence module. The BLAST (basic local alignment search tool) [2] application available from NCBI rapidly identifies similar sequences by comparing a query sequence against an existing database of sequences.

Public access to BLAST servers through a web interface is available at NCBI, and the BLAST computations are performed on computers at NCBI which has a fixed amount of available computational capacity. A common mode for using BLAST is for a user to BLAST a single, short sequence against one specific database using a web interface—sequence is “cut-and-pasted” into the interface. The BLAST application is also available for local installation and analyses. The number of BLASTing computations required to identify additional genomic sequence for an interval is a function of both the size of the interval and number of sequence elements mapped to the interval. Coupled with the need to BLAST potentially thousands of sequences against multiple databases, a local installation of BLAST is required.

Using the web interface and the computational resources of BLAST service providers is not feasible do to the scope of the problem.

Fortunately, BLAST computations are “embarrassingly parallel,” meaning that linear speedup of the computation may be achieved by increasing the number of processors working on the computation linearly. Therefore the time necessary for the BLAST computations is reduced by a factor of the number of processors applied to the computations. The large scale BLAST Module (LSB Module) performs BLASTs locally on 32 distributed BLAST nodes. These 32 nodes are 500 MHz Pentium III-class machines with 1–2 GB of memory. The BLAST databases are distributed across the nodes so that the databases are available on local disk. The interval sequence is BLASTed against the nt, nr, est, and htg databases [29]. The query sequence is placed on an NFS-mounted drive via giga-bit ethernet so that it is available to all machines. The portable batch system (PBS) [42] is used to distribute the BLASTing processes across all nodes based on processor load and availability. This system enables the BLASTing of thousands of sequences approximately 32 times faster than with a single node. A simple timing comparison shows different run times for BLASTing 32 megabases of genomic sequence flanked by markers D3S159 and D3S1271. The 32 Mb were partitioned into 10 kilobase-sized pieces and BLASTed against all sequences in UniGene. Using 32 nodes, the calculation took approximately 21 minutes, compared to approximately 13 hours on a single CPU. The BLAST analyses identify new sequence through sequence similarity, but the actual sequence implicated in a BLAST alignment description is only a reference to an accession number, thus the sequence beyond the region of local alignment is not conveniently available. Accession numbers in the BLAST alignment description identifies the sequence record within GenBank. Again, with the potential for 100 s of BLAST “hits,” an automated application is required. For this reason, a second sequence module was created to mine BLAST alignments for sequence records in GenBank. This module, titled *AcquiLink* (acquire link), is a simple Java application that parses through BLAST results, identifies GenBank sequence records, and acquires those sequence records automatically.

The processing accomplished by this module complements the results from the Interval Search module. Using the sequences identified by the Interval Search module, this module performs large scale distributed BLASTing to identify similar sequences across different databases. The LSB module, and the *AcquiLink* module, are also useful independently for large BLASTing tasks, for identifying sequence across different BLAST databases, and for identifying and acquiring additional sequence by parsing BLAST results.

6.3. Identification of the *BBS4* gene

Bardet-Biedl syndrome is a heterogeneous autosomal recessive disorder with the cardinal characteristics of obesity, pigmentary retinopathy, polydactyly, renal malformations, mental retardation, and hypogenitalism [4, 6, 15, 25] The *BBS4* locus was initially mapped to chromosome 15 using DNA pooling in a large inbred Bedouin kindred [10]. The interval was narrowed to an approximately 1 cM region

between markers D15S131 and D15S192 using haplotype analysis performed on this kindred and two smaller consanguineous families [8]. A BAC contig was assembled across the interval and used for random sample sequencing at 1X coverage. The sample sequence was subsequently used to identify approximately 3 Mb of additional sequence in the high throughput genome sequence (htgs) [29] database. This genomic sequence was BLASTed against the nr [29] and est [29] databases with the LSB Module.

The application, RepeatMasker [45], as part of the Sequence Module, was used to mask human repeats and regions of low sequence complexity. Large fragments of sequence were electronically cleaved into 612 distinct 5 kb pieces to facilitate a distributed BLAST [2] sequence analysis by localizing BLAST hits to smaller sequence fragments. The Large Scale BLAST Module was used to BLAST this sequence.

The Interval Search module was used to identify ESTs mapped to the interval in GeneMap99 flanked by markers D15S131 and D15S192. Using this sequence, additional sequence was identified using the Large Scale BLAST module by BLASTing against the htgs, nr, and nt databases, then additional sequence (presumably within the interval) was acquired using AcquiLink in the Sequence Module. This additional sequence was then re-BLASTed to further identify novel sequence in the interval. Using this process, approximately 93 kb of non-redundant sequence was acquired, and the following genes were identified from the sequence annotation: *PMII* (X76057.1), *POR* (M12078.1), *CYP1A2* (M31667.1), *iAPC* (M55053.1), *PML* (M79462.1), *CSK* (X74765.1), *CYP1A1* (K03191.1), *LOXLI* (NP_005567.1), fusion protein (AAA59972.1), *PML-RAR* (M73779), *UNC24*, (AF074953) and *SLP-1* (NP_004800).

With enough draft sequence now available to conclusively anchor the interval flanking markers and intervening sequence, the quality of the sequences acquired earlier by the Interval Search application may be evaluated. By creating a BLAST database out of the draft genomic sequence, and BLASTing the sequences acquired by the Interval Search application, the relevancy of sequence from Interval Search may be assessed. This analysis showed that 23 out of 133 sequence (17.3%) of the sequences acquired by Interval Search showed high similarity to the genomic interval sequence. Unfortunately, the Interval Search process failed to identify any of the ESTs from the cluster that contained the gene for *BBS4*. The cluster containing the *BBS4* gene [21] was identified using the Large Scale BLASTing Module. The results from BLASTing genomic sequence against ESTs from UniGene were manually examined and candidate clusters were selected based on similarity to previously identified genes [24]. Details of the *BBS4* mutations and molecular genetics are described in Myktyyn et al. [21].

Although the techniques implemented by these modules within the proposed software architecture were only indirectly used to identify the *BBS4* gene, the modules proved useful for automating the process of acquiring and analyzing of genomic sequence by a method not previously attempted. The sequence comparisons between genomic sequence and human ESTs were used to identify candidate genes that ultimately led to the identification of the *BBS4* gene. The software described here may be obtained from (<http://pdb.eng.uiowa.edu/~tabraun/is1.2>). Note that

the resources used by these applications having frequently changing interfaces requiring modifications to the applications for continued access to the resources.

References

1. B. Agnew. When pharma merges, R&D is the dowry. *Science*, 287(5460):1952–1953, 2000.
2. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.
3. J. B. L. Bard, R. A. Baldock, and D. R. Davidson. Elucidating the genetic networks of development: a bioinformatics approach. *Genome Research*, 8:859–863, 1998.
4. G. Bardet. Sur un syndrome d'obesite infantile avec polydactylie et retinite pigmentaire (contribution a l'etude des formes cliniques de l'obesite hypophysaire). Thesis: Paris Note: No. 479, 1920.
5. A. D. Baxevasis. The molecular biology database collection: an updated compilation of biological database resources. *Nucleic Acids Research*, 29(1):1–10, 2001.
6. A. Biedl. Ein Geschwisterpaar mit adiposo-genitaler Dystrophie. *Dtsch. Med. Wschr.*, 48:1630, 1922.
7. C. Blaschke and J. C. Oliveros. Mining functional information associated with expression arrays. *Functional Integrative Genomics*, 1:256–268, 2001.
8. E. A. Bruford, R. Riise, P. W. Teague, K. Porter, K. L. Thomson, A. T. Moore, M. Jay, M. Warburg, A. Schinzel, N. Tommerup, K. Tornqvist, T. Rosenberg, M. Patton, D. C. Mansfield, and A. F. Wright. Linkage mapping in 29 Bardet-Biedl syndrome families confirms loci in chromosomal regions 11q13, 15q22.3-q23, and 16q21. *Genomics*, 41:93–99, 1997.
9. M. Burset and R. Guigo. Evaluation of gene structure prediction programs. *Genomics*, 34:353–367, 1996.
10. R. Carmi, T. Rokhlina, A. E. Kwitek-Black, K. Elbedour, D. Nishimura, E. M. Stone, and V. C. Sheffield. Use of a DNA pooling strategy to identify a human obesity syndrome locus on chromosome 15. *Hum. Molec. Genet.*, 4:9–13, 1995.
11. J.-M. Claverie. From bioinformatics to computational biology. *Genome Research*, 10:1277–1279, 2000.
12. S. R. Eddy. A review of the profile HMM literature from 1996–1998. *Bioinformatics*, 14:755–763, 1998.
13. B. Ewing and P. Green. Analysis of expressed sequence tags indicates 35,000 human genes. *Nature Genetics*, 25:232–234, 2000.
14. V. Ganti and R. Ramakrishnan. Mining very large databases. *Computer*, 38–45, August, 1999.
15. J. S. Green, P. S. Parfrey, J. D. Harnett, N. R. Farid, B. C. Cramer, G. Johnson, O. Heath, P. J. McManamon, E. O'Leary, and W. Pryse-Phillips. The cardinal manifestations at Bardet-Biedl syndrome, a form of Laurence-Moon-Biedl syndrome. *New Eng. J. Med.*, 321:1002–1009, 1989.
16. J. L. Hennessy and D. A. Patterson. *Computer Architecture a Quantitative Approach*, Morgan Kaufman Publishers, Inc., San Francisco, CA., USA. p. 7, 1996.
17. T. Jenssen, A. Laegreid, J. Komorowski, and E. Hovig. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28:21–28, 2001.
18. D. P. Kreil and T. Etzold. DATABANKS—a catalog database of molecular biology databases. *Trends in Biochemical Sciences*, 24(4):155–157, 1999.
19. C. Kalb. When drugs do harm. *Newsweek*, p. 61, April 27, 1998.
20. E. S. Lander, et al. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, February 15, 2001.
21. K. Mykytyn, T. Braun, R. Carmi, N. B. Haider, C. C. Searby, M. Shastri, G. Beck, A. F. Wright, A. Iannaccone, K. Elbedour, R. Riise, A. Baldi, A. Raas-Rothschild, S. W. Gorman, D. M. Duhl, S. G. Jacobson, T. Casavant, E. M. Stone, V. C. Sheffield. Identification of the gene causing the human obesity syndrome, BBS4. June, 2001. To appear in *Nature Genetics*.
22. J. Ott. *Analysis of Human Genetic Linkage*, Johns Hopkins University Press, Baltimore and London, pp. 54–80, 1991.
23. S. L. Salzberg, D. B. Searles, and S. Kasif. *Computaitonal Methods in Molecular Biology*, Elsevier, Amsterdam, The Netherlands, pp. 228, 1999.

24. A. M. Slavotinek, E. M. Stone, K. Mykytyn, J. R. Heckenlively, J. S. Green, E. Heon, M. A. Musarella, P. S. Parfrey, V. C. Sheffield, and L. G. Biesecker. Mutations in MKKS cause Bardet-Biedl syndrome. *Nature Genet.*, 26:15–16, 2000.
25. S. Solis-cohen and E. Weiss. Dystrophia adiposogenitalis, with atypical retinitis pigmentosa and mental deficiency, possibly of cerebral origin: a report of four cases in one family. *Trans. Assoc. Am. Phys.*, 39:356–358, 1924.
26. R. H. Tamarin. *Principles of Genetics*, Wm. C. Brown Publishers. Dubuque, IA, 1996.
27. A. Watson. The universe shows its age. *Science*, 279(5353):981–983, 1998.
28. T.-L. Young, M. O. Woods, P. S. Parfrey, J. S. Green, E. O'Leary, D. Hefferton, and W. S. Davidson. Canadian Bardet-Biedl syndrome family reduces the critical region of BBS3 (3p) and presents with a variable phenotype. *Am. J. Med. Genet.*, 78:461–467, 1998.
29. <ftp://ftp.ncbi.nlm.nih.gov/blast/db/>
30. <http://www.ncbi.nlm.nih.gov/Entrez/>
31. <http://www.ncbi.nlm.nih.gov/genome/guide/human/>
32. <http://www.ncbi.nlm.nih.gov/Genbank/index.html>
33. <http://www.nhgri.nih.gov/HGP/>
34. <http://archive.uwcm.ac.uk/uwcm/mg/hgmd0.html>
35. <http://hmmer.wustl.edu/>
36. <http://www.java.sun.com/>
37. <http://www.genome.ad.jp/kegg/>
38. <http://research.marshfieldclinic.org/genetics/>
39. http://www.informatics.jax.org/reports/homology_map/mouse_human.shtml,
40. <http://www.ncbi.nlm.nih.gov>
41. <http://www.ncbi.nlm.nih.gov/entrez/Omim/mimstats.html>
42. <http://www.OpenPbs.org>
43. <http://www.genome.washington.edu/UWGC/analysistools/phrap.htm>
44. <http://www.ncbi.nih.gov/RefSeq/index.html>
45. <http://ftp.genome.washington.edu/RM/RepeatMasker.html>
46. <http://www.ncbi.nlm.nih.gov/SAGE/>
47. <http://searchlauncher.bcm.tmc.edu:9331/seq-search/struc-predict.html>
48. <http://genome.ucsc.edu/>
49. <http://eyeball.eng.uiowa.edu/clustering/>
50. <http://www.ncbi.nlm.nih.gov/UniGene/>