DESIGN, CONSTRUCTION AND IMPLEMENTATION OF A WEB-BASED DATABASE SYSTEM FOR TUMOR SUPPRESSOR GENES

By

YANMING YANG

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2003

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Li M. Fu, my major adviser, for his guidance in the establishment of the research project and advice on my study progress. My thanks also go to Dr. Mark Yang of the Department of Statistics and Dr. Donald McCarty of the Department of Horticultural Science for serving as my committee members and their suggestions for finalizing the thesis.

I would like to thank my wife, Xidan Zhou, and my daughters, JingRu and Kathleen, for their support to my personal life, their encouragement to my studies, and their sharing of frustration and happiness with me.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

DESIGN, CONSTRUCTION AND IMPLEMENTATION OF A WEB-BASED DATABASE SYSTEM FOR TUMOR SUPPRESSOR GENES

By

Yanming Yang

May, 2003

Chairman: Li M. Fu
Major Department: Computer and Information Science and Engineering

The rapid growth of Internet technology has revolutionized every area of the information world. The Web-based informatics system, as one of the most emphasized research and application fields, has experienced a booming development ever since the emergence of Internet technology. Information trafficked through the Web is involved in almost every aspect of modern society. One of the most remarkable benefits brought up by Internet information exchanges is the application of numerous databases. On the Web, many subject-specific databases have been developed, such as those in genome sequences and specific diseases. However, a database as a comprehensive information source for tumor suppressor genes has not come into being, although those genes are extremely important in controlling cancer development and are being intensively studied. Therefore, in the present project a Web-based database system for those specific genes is designed, constructed and implemented.

The database system contains data in 178 tumor suppressor genes, which represent the most up-to-date information in this area. A database home page was created to accommodate these genes in a pull-down window so that each gene can be viewed individually in a separated web page. Information displayed on each page includes gene name, aliases, source organism, chromosome location, expression tissues, gene structure, protein size, gene functions and major reference sources. The database was implemented in SQL*Plus and driven by Oracle DBMS system. Interactive operations between users and the database are coordinated by a middle-layered application program; i.e., a servlet. Queries to the database can be conducted through a user-friendly interface, and then processed through some utility programs so that the format of the queries becomes compatible to the underlying database. Query results are returned in HTML format in dynamically generated web pages. In conclusion, this database system is a user-friendly, comprehensive and up-to-date information source for clients in medical research and other related areas.

CHAPTER 1
INTRODUCTION

1.1 Internet and Database Development

The emergence of the Internet in the 20th century is one of the most influential events in the history of civilization. The development of Internet technology has already revolutionized the human society in many aspects, and undoubtedly will further change people's way of living, thinking, communicating, and so on. Information traffic through the Internet constitutes the ultimate target of Internet technology. Information trafficked through the Web (the World Wide Web) represents a large portion of human activities which include email communication, business promotion and execution, entertainment acquisition, scientific exploration and cooperation, online education, military deployment, etc. The development speed of information technologies can be claimed to be "explosive" and even becomes ever faster than before. In the last 30 years or so, Internet technology experienced a dramatic advance, which is usually beyond people's expectation. A survey conducted in September 2002 indicates that about 505.6 million people are online worldwide. Internet-generated revenue reached 8 million US dollars in 1994, and increased to 1234 billion dollars in 2001. Currently, 24.2 million Americans use online banking [1]. Without any doubt, applications of the Internet technology will reach every corner of the daily life of mankind. As a result, the development of the Internet itself and relevant technologies will also be enhanced dramatically.

As one of the most important application areas of the Internet technology, the development and application of Web-based databases have been significantly promoted

and remarkably emphasized. Through the Internet, those Web-based databases are generated, manipulated and used by people from all disciplines at various geographically-distributed locations. Currently, among the huge number of online databases accessible to the public, databases in biological studies represent one of the most important and well developed categories. The Human Genome Project, collectively conducted by scientists from the United States and other countries, stimulates the development and application of databases in those scientific fields of human being studies, such as database for a particular disease or for a metabolism category. As a branch of informatics, bioinformatics has been boomingly developed, and plays more and more important roles in revealing gene functionality, and hence in improving our understanding of the genetic basis for diseases and mechanisms, which could further enhance new drug inventions. Without the development of the Internet and the construction of related databases, it is impossible for bioinformatics to reach its high level in development and application.

1.2 On-line Gene Databases

At present, there exists a large number of biology-related databases on the Web, which can be divided by category into resource database, sequence database, structure database, genome database, and other database, e.g., restriction enzyme database, protein kinase database, cell signal networks database, etc. Among the public-accessible databases, those manipulated by the National Center for Biotechnology Information (NCBI) accommodate the most diverse information on gene resource, transcript variants, mRNA and deduced protein sequences, structure information, and related references. Gene databases in a particular subject include Tumor Gene Database (Baylor College of Medicine) [2], Asthma Gene Database (Germany) [3], Stanford HIV RT and Protease Sequence Database [4], The Human Gene Mutation Database (University of Wales) [5],

Bacterial Polysaccharide Gene Database (The University of Sydney) [6], Breast Cancer Gene Database (Baylor College of Medicine) [7], Olfactory Receptor Gene Database (Fred Hutchinson Cancer Research Center) [8], Cardiovascular Gene Database (Bioinfromatics.org) [9], Obesity Gene Map Database (Pennington Biomedical Research Center) [10], Cardiac Gene Database (France) [11], etc. Mentioned here are just a few of them. The list of biology-related databases is extending rapidly thanks to the availability of Internet information exchange and rapid advances in biological and medical researches. However, to the author's knowledge, the currently existing database for tumor suppressor genes, factors whose inappropriate mutations lead to cellular changes in malignant transformation, is the one for gene p53 manipulated by World Health Organization, which is not freely accessible via the Web. In addition, there is a web page named "Tumor Suppressor Genes" created and maintained by Dr. Eugene Pergament of Northwest University and Dr. Morry Fiddler of DePaul University, Chicago, which contains about 30 tumor suppressor genes [12].

1.3 Tumor Suppressor Genes in Human Cancer

Cancer diseases are ancient problems existing in higher life. Cancer arises when cells grow and divide in an unregulated manner. While significant strides have been made in fighting cancer, people still have to face the fact that each year 1.4 million Americans are diagnosed with cancer and 560000 will die (1500 people each day), and within next 10-15 years, cancer could become the nation's number one cause of death [13]. The modern crusade against cancer can be traced back to the emergence of the concept--a War on Cancer, which is marked by the National Cancer Act of 1971 and subsequently invoked the phrase "Protecting our nation through research" [13]. Oncology studies and

cancer-related researches, including gene therapy and new drug development, attributed their progress largely to molecular revealing of genes and gene functions.

The concept of individual genes underlying the biology of malignant transformation can be traced back to the early 1900s when infectious viruses were discovered as causes to avian sarcomas. Many decades later, the identification of the oncogene *Src* brought up the notion of the dominant oncogene, a factor whose inappropriate activation confers cellular transformation to malignant status. In 1971, Alfred Knudsen predicted the existence of a second class of oncogenes whose contribution to cancer is recessively inherited [14]. His hypothesis was based on observations of cancer risk in familial cancer inheritance patterns and the notion that disease predisposition may represent a multi-hit phenomenon with loss of function mutations contributing to the malignant phenotype. Thus the concept of tumor suppressor gene (TSG) was introduced. The observation that bilateral retinoblastoma (RB) occurred with earlier onset than unilateral disease led Knudsen to formulate the two-hit theory for tumorigenesis [14]. This theory provided the underpinning for the search for tumor suppressor genes. In 1986-1987, the Rb gene was identified which fulfills the criteria of a tumor suppressor gene RB [15, 16]. The TSG concept has been abundantly validated by laboratory researches and clinical investigations. The most striking validation of TSG concept comes from the discovery of inactivating mutations or deletion of candidate genes in cancer-prone families, which include p53, the NF family genes, DNA mismatch repair genes, Wilms, von Hippel Lindau, and other genes. In all these cases, heterozygous germline disruption of a single allele is associated with cancer predisposition in affected family members.

Inactivating mechanisms for tumor suppressors are diverse, and are still being discovered. For example, in addition to traditional notion of loss of function mutations or deletions, the more recent appreciated inactivating mechanisms include transcriptional silence, targeted protein degradation, and functional disruption of tumor suppressor gene activities. These diverse mechanisms of inactivation highlight one of the most striking breakthroughs in cancer biology--the discovery of discrete pathways in which dominantly acting and tumor suppressing genes converge. The function convergence of dominant oncogenes and tumor suppressor genes in cellular growth or survival pathways indicates the complexity of carinogenesis. A gene's ability to fit into a certain regulatory pathway has become a virtual requirement for its eligibility as a cancer modifier. The pathways that tumor suppressor gene modulate in cancer are largely in the regulation of the cell cycle, cell death, growth factor signaling, DNA damage responses, and other stress responses. Nearly all tumor suppressors are believed to function through modulation of one (or several) of these pathways. In tradition, cell cycle regulation has been targeted in the etiology of cancer. However, more recent observations suggest that cell survival pathways exist as distinct, genetically selected entities. Either dys-regulated cell growth or inefficient cell death (or both) are strongly associated with tumorigenesis.

Normally, tumor suppressor genes inhibit growth and regulate cell behavior much like a car's brake. When oncogenes accelerate, the tumor suppressor genes encode proteins that stop continued division and growth. It's only when the tumor suppressor gene is damaged that the oncogene's activity goes unchecked. As an example, tumor suppressor gene p53 prevents replication of faulty DNA and prompts cell suicide. Mutations of this gene are implicated in over half of all human cancers.

1.4 Necessity and Significance of TSGDB

Because of the important position of tumor suppressor gene in both oncological studies and gene therapy practices, it will be necessary and significant to build a database to accommodate all of the discovered tumor suppressor genes, so that scientists and medical persons in the related community can have a convenient and reliable information source for their interests. In another aspect, the availability of Internet information and the popularity of the Web browsing make such kind of database more helpful and powerful in human being's fighting against cancer. Based on the necessity, possibility and significance to build a tumor suppressor gene database (TSGDB), the design, construction and implementation of the database was chosen as an MS thesis research project in the area of bioinformatics.

1.5 Thesis Organization

This thesis is composed of 6 chapters. Chapter 1 gives a brief description of the Internet and Web-based database development, TSGs, and the significance in constructing and implementing a Web-based database for TSGs. Chapter 2 describes the underlying technologies in Web-based database systems. Chapter 3 consists of information on data acquisition and Web page construction for TSGDB. Chapter 4 deals with the implementation of the database in the relational model. Chapter 5 describes the implementation of the Web-based database system. Chapter 6 presents a brief conclusion of the research project and proposes some future work for the database improvement. In addition, there is Appendix portion that contains some major parts of the utility programs for realizing some implementation-related functions and human TSG function summary.

CHAPTER 2
UNDERLYING TECHNOLOGIES FOR DBMS

A Web-based database system itself is a distributed system which can be accessed by users all over the world via the Internet. Over the tenure of the development, database systems were built with various data models, and each model has its own technology specificity and requirement. In the following sections of this chapter, the technology specificities in distributed systems and various data models are analyzed respectively.

2.1 Distributed Database Systems

A distributed database system enables users to access data residing anywhere in a related computer network with high level of transparencies in computer hardware, operating system, and data manipulation language and file structure. The data distributed across multiple remote computers will appear to the user as if it resides in the user's computer. Although this scenario is still suffering some functional limitations, including transaction management, standard protocols for remote connection and network topology, distributed database systems represent a way of overcoming the shortage limitations of traditional systems and a development trend in database utilization.

2.1.1 Distributed Environment Architecture

The design of a distributed database environment can be carried out either by incremental interconnection of the existing systems or by developing a completely new distributed DBMS environment.

Legacy system interconnection. Like other aspects of computer systems, legacy systems in database management are costly investments and are still useful though

performance is no longer satisfactory. Usually, the existing systems can not be replaced by the distributed systems at once. To preserve the existing system environments including hardware, software and database, some mechanisms for producing federated system must be provided. A federated system is a system that is composed of autonomous software components. Utilizing the federated approach is a practical solution toward a distributed environment, since it combines the existing systems with the newly extended nodes during the transition from the traditional centralized system to a distributed environment. Within such federated systems, pairs of nodes can be coupled in either a loose way (i.e., every node is autonomous) or a very tight way (i.e., nodes directly interact with each other). The node coupling approach directly affects system's design, execution and capability. For instance, coupling can affect the translation requirement between nodes. If both node components use the same representations, no translations are needed.

Loosely coupled systems represent the most modular and the easiest to maintain because changes in system implementation characteristics and its DBMS in one site usually can not affect those in the other sites. But loose coupling suffers some disadvantages which affect the data transaction quality and performance. With this coupling approach, users must have some knowledge of sites' characteristics to perform the requests. Usually, more translations are involved between nodes, thus the system performance can be undermined. In addition, lack of central authority to control consistency could lower data correctness rate.

The shortages of loose coupling can be overcome by the tight coupling approach. With tight coupling systems, users need not to know other sites' characteristics for their

requests. Because of the centralized control, consistency in using resources and in shared data management can be guaranteed. But, users with this kind of system can lose their freedom in choosing target format to central control mechanisms necessary to maintain the nature of tight coupling.

Node cooperation. In the truly distributed system, cooperation between nodes can be expressed by different methods, including defining transparency amount and defining node autonomy available to each other. Transparency refers to the degree to which a service is offered by the distributed DBMS so that the user does not need to know it. For example, in a distributed system with location transparency, if a client makes a request to a target object, the client should not need to know if the target object is local to his own machine or somewhere else. Server processes should be easily moved around from machine to machine without clients becoming aware of it. Node autonomy refers to the amount of independence that a node owns in making policy decisions. Examples of policy decisions include data ownership, data accessing policies, policies in using cooperation time and personnel, etc.

Interconnection of new systems. If an organization establishes its distributed database environment from the scratch, it has much more freedom to choose system products. Currently, vendors supply homogeneous distributed DBMSs with compatible family of software. They even offer a gateway mechanism from their distributed database software to other DBMSs, usually at additional development cost. This can free the organization from locking itself into a single vendor's proprietary distributed system products. There are also some other approaches in selecting distributed architecture choices [17]:

- Identical DBMS product at each node and a single proprietary communications network to connect all nodes, with possible different hardware environment.

- Standard confirming DBMS products at each node guided by standard communications protocols.

- With same data model and by using a single or a standard communications protocol, different DBMSs are connected.

- With different data models, by using a single or a standard communications protocol, different DBMSs are connected.

No matter which architecture choice is suitable for the organization, when designing a new distributed DBMS system, it is always advisable to consider a mixture of standard confirming DBMSs and communications protocols.

2.1.2 Distributed Database Implementing Approaches

Usually, three implementing approaches can be used in establishing a distributed database system, and adaptation of each implementing approach represents a unique data communications requirement [18].

Fragmentation approach. This approach can be further divided, based on how a table is broken, into horizontal, vertical and mixed fragmentations. Horizontal fragmentation breaks a table into rows and stores all fields (columns) of each row in a separate location within the table but only a subset of rows. Vertical fragmentation stores a subset of a table's columns in different locations. Mixed fragmentation combines both horizontal and vertical fragmentations. With fragmentation, local storage only needs to contain site-specific data. Site-nonspecific data, when needed, can be retrieved through the distributed system from other locations without user's awareness under the transparent characteristics.

Replication approach. With this approach, copies of total data around the network are stored locally. This approach achieves maximum data availability since all data are stored in each location, and can provide backup data copies in case a particular network node is corrupted. But, replication of all the network data at each site wastes storage resources and increases the workload of each node.

Allocation approach. This is a combination of fragmentation and replication. With allocation, only those data which are used at highest frequency are replicated and stored at the node so that maximum availability can be achieved without consuming too much storage resources.

For a particular distributed DBMS, the degree to which these approaches are combined to form a distributed database structure constitutes a major factor in determining the data communications requirement.

2.1.3 Pros and Cons of Distributed DBMSs

Compared to traditional DBMSs, distributed DBMSs possess some advantages.

- A distributed DBMS reflects the changing structure of an organization. The modular implementation of distributed databases allows data to change as the organization changes, e.g., an organization may spread geographically to different locations and the databases implementation can keep the original model.

- Data are located near the user and majority of the processes are executed locally, therefore communication overhead can be reduced and faster response time can be achieved.

- Greater data availability and system reliability can be achieved.

- Workload can be distributed to different sites, and better balance of system usage can be achieved.

- Faster data recovery can be supported from other sites of the network if the replication implementation method is employed.

Despite a number of advantages a distributed database system possesses, there are still some problems inherent to having data located at multiple sites. These problems are largely derived from transparency issues, including security issues, high cost, complexity in managing transactions, communication delay, and difficulties in consistency and concurrency controls.

2.2 Database Models

A database model is a collection of logical constructs used to represent the data structure and the data relationship found within the database [19]. Database model provides a way in which the stored data is organized for quick retrieval or update. Research in the area of database has resulted in many models, including Hierarchical model, Network model, Relational model, Object oriented model, Semistructured model, Associative model, Context model and some hybrids of the mentioned models. Each of the models encountered some limitations in being able to represent the data. Of all the models, the relational model attained the most popularity and remains as popular at the present time mainly because of its simplicity, ease of use and manipulation via a standard query language (SQL).

2.2.1 Hierarchical Model

Hierarchical DBMSs were popular from the 1970s due to the introduction of IBM's Information Management System (IMS) DBM [20]. Hierarchical data models organize data into a tree structure. In this kind of structures, record information can be repeated, usually in different branches of the tree. Data are gathered in a series of records, which have a set of field values attached to it. All of the instances of a specific record are represented as a record type which is equivalent to a table in the relational model and with the individual records being the equivalent of rows. These record types can be

linked together in the hierarchical model by using parent-child relationships, i.e., the 1:N mapping between record types. For example, an organization might store information about an employee, such as name, telephone number, address, department, salary, etc. The organization might also store information about an employee's children, such as name, gender, and date of birth. The employee and children data form a hierarchy, where the employee data represents the parent segment and the children data represents the child segment. This model has a limitation: a child segment can have only one parent segment.

2.2.2 Network Model

The network model was formally defined by the Conference on Data Systems Languages (CODASYL) in 1971, and the database standard was established by DBTG (Data Base Task Group) and improved by SPARC (Standard Planning And Requirements Committee). As mentioned in the previous example, some data can be more naturally modeled with more than one parent per child. The network model allows many-to-many (N:N) relationships in data modeling. The set construct, the basic data modeling construct in the network model, can be composed of a set name, an owner record type, and a member record type. A member record type can have more than one set, so that the multi-parent concept is supported. An owner record type can also be a member or owner in another set. The data model is a simple network in which the link and intersection record types may exist. In this model, the complete network of relationships can be represented by pairwise sets; in each set some record type could be owner and one or more record types could be members.

2.2.3 Relational Model

The relational database model is based on the relational algebra. RDBMS (relational database management system) was based on the relational model developed by E.F. Codd [21]. A relational database allows the definition of data structures, storage and retrieval operations and integrity constraints. In such a database the data and relations between them are organized in tables. A table is a collection of records and each record in a table contains the same fields. Certain fields may be designated as keys or foreign keys, so that searches for specific values of that field can be speed up by key indexing. If fields in different tables take values from the same set, a join operation can be performed to select related records in these tables by matching values in those fields. Operations can be performed by joining multiple tables on multiple fields. Because the complex relationships revealed by the joining operation are only specified at retrieval time, relational database system is classed as dynamic database management system.

2.2.4 Object Model

Object-oriented databases (OODBs) combine object-oriented programming language (OOPL) systems and persistent systems. “The power of the OODB comes from the seamless treatment of both persistent data, as found in databases, and transient data, as found in executing programs" [22]. Object-oriented models add database functionality to object programming languages, while retaining native language compatibility, extending the semantics of object programming languages, such as C++, Smalltalk and Java, to provide full-featured database programming capability. In this model, database application and development are unified, so that applications require less code and code bases are easier to maintain. Because of the one-to-one mapping of programming language objects to database objects, this model has no performance overhead to store or

retrieve a web or hierarchy of interrelated objects, and thus provides higher performance management of objects with complex interrelationships. This makes object DBMSs better suited to support applications such as telecommunications service applications, World Wide Web document structures, design and manufacturing systems, and hospital patient record systems.

2.2.5 Object-relational Model

By inheriting the robust transaction- and performance-management features of relational models and the flexibility of object-oriented models, object-relational database management systems (ORDBMSs) add new object storage capabilities to the relational systems, and thus integrate the management of conventional fielded data, complex objects and diverse binary media such as audio, video and images. ORDBMS servers can execute complex operations in data manipulation to retrieve and transform complicated objects. Utilization lead by such big-named vendors as IBM, Informix and Oracle, ORDBMS query and procedural languages and call interfaces are extensions of RDBMS languages and interfaces, which include SQL3, ODBC, JDBC and proprietary call interfaces.

2.2.6 Semistructured Model

Originally, semistructured data were studied in the context of integration of a large volume of heterogeneous sources. The term semistructured was first used in the original equipment manufacturer (OEM) model [23]. The most popular example of semistructured data is XML. The unified idea about semistructured data is the representation of data as some graph-like or tree-like structure that contains labels showing semantics to its underlying structure. In semistructured data model, the information normally associated with a schema is contained within the data, which is

sometimes called ``self-describing". Usually, there is no clear separation between the data and the schema, and the degree to which it is structured depends on the application. Such databases subsume the modeling power of flat relational databases, entity nesting features of nested databases, and the cyclic referencing power of objects. Because of the rapid development of Web-based information transferring, semistructured databases are being intensively studied.

2.2.7 Associative Model

The associative model represents the first database architecture designed to reflect the structure of data in the real world and the way that people perceive and process information. Associative model has been proposed and developed by the Chief Executive of LazySoft Ltd, Simon Williams [24]. This model divides real-world things into two categories: entity and association. Entities are things that have discrete and independent existence, whose existence does not depend on any other thing. While associations are things whose existence depends on one or more other things, such that if any of the other related things ceases to exist, then the thing itself ceases to exist or becomes meaningless. For instance, a person is an entity, but the role that person plays as programmer, football player, or philosopher is an association. If the person dies, so does the programmer, football player, and philosopher. However, if the person stops playing football, only that association dies, not the person or other associations. Associative data takes the form of the subject-verb-object syntax like English sentences. For example, "John likes bananas" and "Mike sends email" are associative data, with "John", "bananas", "Mike" and "email" as entities, and "likes" and "sends" as associations between these entities.

2.2.8 Context Model

Context model combines some capabilities of object-oriented, network and semistructured models. It is a flexible model with which any type of database structure can be used depending on task. This kind of data contains a set of predefined types and user defined types. The predefined types include character strings, texts, digits, pointers (links, references) and aggregate types (structures). These predefined data types are grouped into three main categories: regular, virtual and reference. A regular field can be either atomic (i.e., no inner structure) or composite (i.e., may have a complex structure and its type is described in the scheme of file). Further, the composite fields are divided into static and dynamic. While the type of a static composite field is stored in the files schema and is permanent, the description of the type of a dynamic composite field is stored within the record and can vary from record to record.

Similar to a network database, context database has fields storing a place where this information can be found, i.e., a pointer (link, reference) which can point to a record in this or another file. A context database without composite or pointer fields is essentially a relational database. With static composite and pointer fields, context database become object-oriented. With dynamic composite fields, a context database becomes what is now known as a semistructured database. In contrast to pure object-oriented databases, context databases are not so coupled to the programming language and doesn't support object methods directly. Instead, method invocation is partially supported through the concept of virtual fields.

Like a regular field, a virtual field can be read or written. However, this field is not physically stored in the database; therefore it does not have a type described in the file scheme. A read operation on a virtual field is performed by the DBMS’s invoking a

subroutine associated with the field. Similarly, a write operation on a virtual field invokes an appropriate subroutine to update the value of the field. The current value of virtual fields is maintained by a run-time process; it is not preserved between sessions. In object-oriented terms, virtual fields function as public methods that don't take arguments. From the DBMS point of view, virtual fields provide transparent interface to other methods through reading and writing methods.

2.3 Web-based DBMS Applications

The power and maturity of the database technology have made the database systems and related applications among the most popular aspects in the modern society. The advantages of the traditional data management field are represented by its effective data organizing, storing, and retrieving techniques. However, it seems that the popularity of databases in the traditional management systems is taken over by the rapid growth of Internet technology and the World Wide Web (i.e., WWW or the Web). The Web's success stems largely from its simplicity and compatibility. Its simplicity makes users to easily publish or retrieve information from the Internet via the hypertext interface. It is compatible with other existing protocols, such as gopher, ftp, and telnet, etc. Moreover, it provides users with the ability to browse complex documents, such as multimedia files and photograph images, on an open environment available to many different platforms, requiring little or no cooperation between information provider and users. Although some concerned issues such as privacy and security are still challengeable, it is expected that the Internet and Intranet will be the major vehicle for future information exchange. With the application of the Internet or Intranet, database application developers can be freed from network maintenance. Once a database is linked to the Web, it becomes accessible from any corner of the world where an Internet-connected computer is available.

Because the Web browsing capability is accommodated by almost all platforms, it is unnecessary to develop corresponding application interfaces to different platforms. Consequently, more and more researchers and database developers are building Web-based database applications.

2.3.1 Client/Server Architecture

The term client/server was first introduced in 1980s in reference of personal computers on a network. The actual client/server model did not start gaining acceptance until late 1980s. The client/server software architecture is a versatile, message-based and modular infrastructure that is intended to improve usability, flexibility, interoperability and scalability as compared to centralized, mainframe, time sharing computing [25]. The client/server architecture emerged to overcome the limitations of file sharing architectures, i.e., file sharing architectures work only if shared usage is low, update contention is low, and the volume of data to be transferred is low. This approach introduces a database server to replace the file server. By using a relational database management system (RDBMS), user queries could be answered directly. This architecture reduces network traffic by providing a query response rather than total file transfer. It improves multi-user updating through a graphic user interface (GUI) front end to a shared database. In client/server architectures, Remote Procedure Calls (RPCs) or standard query language (SQL) statements are typically used for communications between the client and server [26].

Two tier architectures. With two tier client/server architectures, the user system interface is usually located in the user's desktop environment and the database management services are usually in a server that is a more powerful machine for serving many clients. Processing management is split between the user system interface

environment and the database management server environment. The database management server provides stored procedures and triggers [26].

This architecture functions well with a distributed computing when work groups are defined as less than 100 people interacting on a LAN simultaneously. Because the server maintains a connection via "keep-alive" messages with each client, even when no work is being done, the performance of this architecture deteriorate rapidly if too many users make requests simultaneously. Another limitation of the two tier architecture is that implementation of processing management services using vendor proprietary database procedures restricts flexibility and choice of DBMS for applications. The current implementations of this architecture also limit the flexibility in repartitioning program functionality from one server to another since additional manually regenerating procedural code must be added.

Three tier architectures. The three tier architecture is also referred to as the multi-tier architecture which overcomes the limitations of the two tier architecture. In this architecture, a middle tier is added between the client environment and server environment (Fig. 2-1). Although there are a variety of ways of implementing this middle tier, such as transaction processing monitors, message servers, or application servers, the functions of the middle tier are the same: queuing, application execution, and database staging. The middle layer also adds scheduling and prioritization for work in progress. The three tier client/server architecture has been shown to improve performance for groups with a large number of users (in the thousands) and improves flexibility when compared to the two tier approach [27]. Flexibility in partitioning can be easily achieved, i.e., as simple as "dragging and dropping" application code modules onto different

computers. The three tier architecture has major advantages, as access to any of the servers can be obtained using any of the APIs or protocols provided on the client side. But this architecture has a disadvantage in that its development environment is more difficult to use than that of the visually oriented development of two tier applications [27].

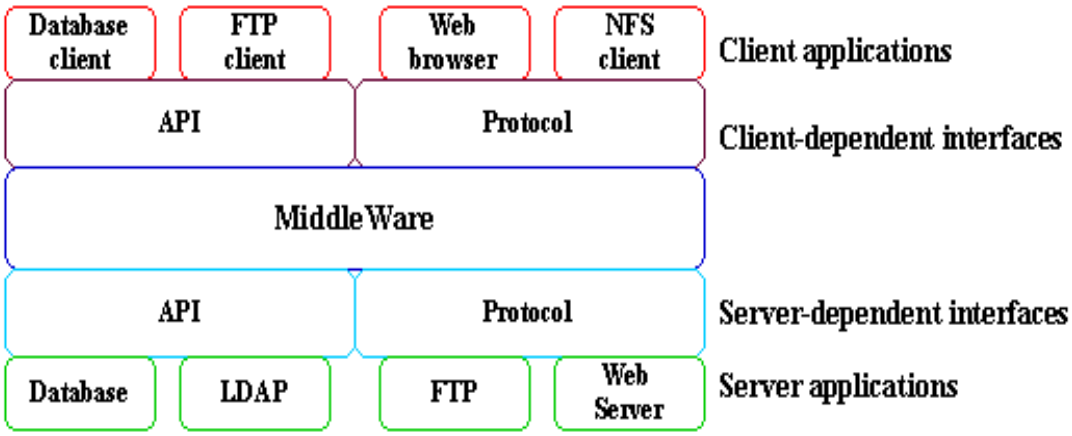

Figure 2-1. Three tier client/server architecture

2.3.2 Java and Web-based Application

Java and the Web. Essentially the Java story begins in about 1990 when Patrick Naughton, a programmer at Sun Microsystems, made a proposal which lead to the company to focus on consumer electronics applications first of all [28]. In 1992, a demo was made to program a video recorder all at the touch of a button. The idea was to integrate the smallest electronic device with cyberspace using a standard language all of it own which would have Internet connectivity at its very heart. At the beginning the code was named "Oak". But then, Sun scrapped the team (by now named *FirstPerson*) in 1994, and just plugged away at their traditional “high end server” market. However very shortly after this, the emergence of Mosaic Web Browser (the first web browser to be able to view pictures) and the sudden growth of the World Wide Web made the folk at

Sun reconsider. Having seen the potential of Mosaic, and its commercial elaboration by Netscape, they challenged to put Java into an explicitly Internet language. Thus, the first real public incarnation of Java was introduced: as applets appearing on web pages. An applet is a mini web application, a small entity of interactivity that executes on your machine within a web browser. By the end of 1994, an experimental web browser written in Java, called WebRunner and later renamed as HotJava, was developed to host those applets [28].

At the early stage of the development, the Web encountered two major obstacles: portability and security. The Internet consists of diverse and distributed computer units in various platforms. It is essential for the programs to be high portable. In another aspect, if any program is allowed to run on a user's machine without restrictions, it will be impossible to protect the user's data and resources from unauthorized intrusion. Java's characteristics, as shown in the following, make it a suitable programming language in dealing with these problems and attractive choice in developing Web-based applications.

- **Simple and Familiar:** It looks quite similar to programmers using such languages as C and C++, but with many of the elements which sometimes prove troublesome (e.g., pointers and memory deallocation) removed.
- **Object-oriented:** A Java program is essentially the weaving-together of various objects (mixtures of code and data) which are instantiations of various "classes" in a very strict way.
- **Architecture-neutral:** Because it is an interpreted language, its execution requires only an interpreter or just-in-time compiler, known as the Java Virtual Machine (JVM). So, as long as JVM is present for a machine, Java programs should theoretically run.
- **Portable:** Theoretically, a java program should be written once and work on all machines because of the reasons described above, though this is still not true in practice.

- **Distributed:** A program can create objects from classes stored on different web servers. Only the appropriate classes are downloaded for a particular group of actions at a particular time.

- **Secure:** Since a program from the Internet is allowed to run on a local machine, a whole layer of Java's architecture is devoted to security, so that it formalizes the way to deal with possible security problems.

Because of these features, Java has being used widely in Web applications, and thus enhanced the development of the Internet technology.

Servlet and user interface. In recent years, there has been a strong movement toward a more dynamic approach for Wed page development. Web pages can be created on the fly, customized to meet users' requirements, and linked with database servers to provide accurate and updated information. There are many technologies for creating dynamic Web pages. Some of the technologies available involve creation of a Web page on the fly, based on the previous page selections a user makes. These techniques include Active Server Pages (ASP), common gateway interface (CGI) scripting, Hypertext Processor (PHP), Server-Side Includes (SSI), JavaScript, Servlet, JavaServer Pages (JSP), and Microsoft FrontPage, etc. Among these techniques, Servlet represents one of the most desirable for creating dynamic Web pages without sacrificing convenience, portability, and security.

Servlets are Java technology's answer to the CGI programming. They are programs that run on a Web server, functioning as a middle layer between requests coming from a Web browser or other HTTP client and the databases or applications on the HTTP server. Servlets have the following functions [29]:

- Read any data from the user. Data can be entered in a form, from Java applet or from a client program.

- Look up other information about the request that is embedded in the HTTP request. For example, information on browser capability, cookies, the host name of the requesting client, and so on.

- Generate results. This process may include requesting a database, executing a remote method invocation (RMI) or a CORBA call, invoking a legacy application or computing the response directly.

- Format the result inside a document. That is to tell the browser what kind result should be returned.

- Send the document back to the client. The returned document may come in text format, binary format (as in images), compressed format, etc.

Theoretically, servlets can not be restricted to Web or application servers that handle HTTP requests, but can be used other types of servers as well, such as FTP servers or mail servers. In practice, however, such applications of servlets have not caught on yet.

Compared to "traditional" CGI, servlets have advantages in efficiency, convenience, capability, portability, security and cost:

- Efficient. With CGI programming, a new process is started from each HTTP request, and the overhead of starting a process could dominate the execution time if the CGI program itself is relatively short. When there are N (N > 1) simultaneous requests to the same CGI program, the code for the CGI program will be loaded into memory N times. With servlets, the JVM stays running and handles each request with a lightweight Java thread, so that there would be N threads but only a single copy of program class. Moreover, termination of a CGI program upon completing a request execution makes it difficult to cache computations, keep connections and

perform other optimizations which rely on persistent data. Servlets solve this problem by remaining in memory the completed responses.

- Convenient and inexpensive. Servlets have an extensive infrastructure whose capabilities include automatically parsing and decoding HTTP form data, reading and setting HTTP headers, handling cookies, tracking sessions, and many other high-level utilities. There are many free or inexpensive Web servers available that are good for personal use or low-volume sites. Adding servlet support to the available server costs very little extra.
- Powerful. Servlets support some capabilities that are impossible or difficult with regular CGI. For instance, servlets can communicate directly with server so that tasks such as translating URLs to concrete pathnames can be accomplished, while CGI can not, at least without using a server-specific API. Servlets can share data, making it easier to implement database connection pooling or other optimization operations.
- Portable. Because servlets are written in Java language and follows a standard API, servlets can run virtually unchanged on various servers, such as Sun's Java Web Server, Apache Tomcat, Microsoft Internet Information Server (IIS), IBM WebSphere, and StarNine WebStar, and so forth. Servlets are being supported directly or by a plug-in on every major Web servers, and now included in J2EE. Thus, industry support for servlets is becoming even more pervasive.

- Secure. Traditional CGI programs are executed on general-purpose operating system shells, and the programmers must carefully and completely filter out some shell-specially-treated characters. Otherwise, troubles could be constantly incurred with widely used CGI libraries. Another source of problems of traditional CGI programs comes from the languages used, e.g., C and C++, which lack such functions as automatic checking of array bounds to avoid buffer overflows. Servlets suffer from none of these problems. Even if a servlet executes a remote system call to invoke a program on the local operating system, it does not use a shell to do so. And array bounds checks and other memory protection features are a central part of Java programming language.

Because of these advantages, servlet technique is used in the implementation of the present project--Web-based human tumor suppressor gene database.

Java JDBC. Officially, JDBC is not an acronym and thus does not stand for anything. Unofficially, "Java Database Connectivity" is commonly used as the long form of the term [29]. JDBC is a standard Java API for connecting to relational databases. By using JDBC, a wide range of different SQL databases can be accessed with exactly the same Java syntax, though SQL syntax of querying and executing transaction may vary with SQL extensions. JDBC provides a standard way to access relational databases from Java applications, applets, and servlets, so that it becomes unnecessary to develop a separate program for accessing databases with vendor-specific drivers. With JDBC, it is possible to access databases by using Java programming language rather than SQL.

Moreover, performance can be improved by using JDBC for connection pooling and data batch loading.

CHAPTER 3
DATA ACQUISITION AND WEB SITE CONSTRUCTION

In the construction of tumor suppressor gene database (TSGDB), the initial tasks include the acquisition of raw data for each tumor suppressor gene, and the development of user-friendly web pages for querying the database and for displaying genes. The data must be accurate and most updated, so that the database, once published, will contain reliable information for cancer studies and other reference purposes.

3.1 Data Acquisition

Once the database development is initiated, the first question that has to be dealt with is: among the huge number of discovered genes which gene is a tumor suppressor gene? Then, once a tumor suppressor gene is picked out, what kinds of gene characters should be included in the gene database? Such basic questions must be answered in constructing the database.

3.1.1 Online Search for TSGs

Currently, there are many databases and other information sources online. After preliminary information retrieving and screening, the search effort was focused on the following information resources:

- NCBI databases, including nucleotide, protein databases.
- OMIM – Online Mendelian Inheritance in Man.
- GeneCards.
- SWISS-PROT.
- SOURCE – Stanford Online Universal Resource for Clones and ESTs.

The NCBI database is a comprehensive resource for gene and protein sequences, protein structures, literatures, and other related information about organisms. In acquiring the raw data, the NCBI nucleotide databases were intensively searched by using some key words. For example, if a gene belongs to a (putative) tumor suppressor gene or related to a tumor suppressor gene, the query result will show the accession number (Fig. 3-1). However, even though some returned gene entry contains the key works (e.g., words "tumor suppressor gene"), the gene itself is not necessarily a tumor suppressor gene. Therefore, for each chosen gene, confirmation must be conducted by further reading related publications in order to maintain the data's accuracy.

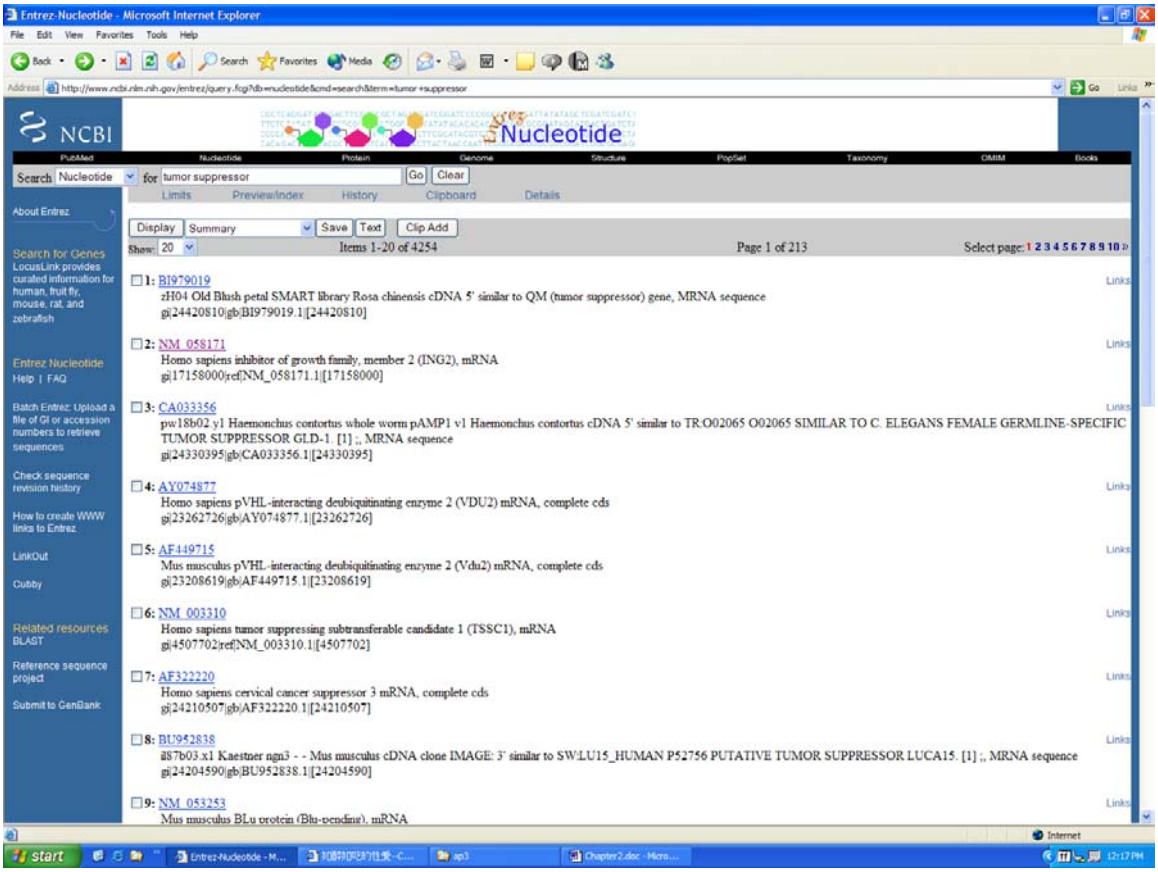

Figure 3-1. Sample query result of the NCBI nucleotide database.

OMIM is a database created and maintained by Dr. Victor A. McKusick and his colleagues at Johns Hopkins University, and developed for the World Wide Web by NCBI [30]. This database is a catalog of human genes and genetic disorders, and contains textual information, references and copious links to MEDLINE and sequence records in the Entrez system, as well as links to additional related resources at NCBI and elsewhere. The search of this database was focused on the contents of gene entries to decide whether the current gene is a tumor suppressor gene.

GeneCards is a database created and manipulated by Weizmann Institute of Science, Israel [31]. This database contains information on human genes, their products and their involvement in diseases. It offers concise information about the functions of all human genes that commonly have an approved symbol, which is particularly useful for people who wish to find information about genes of interest in the context of functional genomics and proteomics. These characters of the database were used to find tumor suppressor genes and other related information such as gene aliases and gene chromosome locations.

SWISS-PROT is a protein sequence database that provides a high level of annotations (such as the description of the function of a protein, its domains structure, post-translational modifications, variants, etc.), a minimal level of redundancy and high level of integration with other databases [32]. This database was used to validate protein information and to compare protein variants from different organism sources.

SOURCE compiles information from some major publicly accessible databases, including UniGene, dbEST, Swiss-prot, GeneMap99, RHdb, GeneCards and LocusLink, to provide a unique scientific resource that pools publicly available data commonly

sought after for any clone, GenBank accession number, or gene [33]. This database accommodates human, mouse and rat genes. It was specifically designed to facilitate analysis of the large data sets that biologists can now produce using genome-scale experimental approaches. This database was used to collect gene information on expression cell or tissue types and gene functions, and to validate chromosome locations.

On the basis of the above information sources, after removing the repeats with various names, 178 tumor suppressor genes were finalized. These genes represent the ever-discovered (putative) tumor suppressors from human and other organisms, including mouse, rat, fish, nematode, insect, plant and yeast (Table 3-1).

Table 3-1. Tumor Suppressor Genes and Their Source Organisms

Source	Gene name						
Homo sapiens	101F6	ABR	ADPRTL3	ANP32C	ANP32D	APC2	APC
	ARHGAP8	ARHI	ATM	ATP8A2	AXUD1	BAP1	BECN1
	BIN1	BRCA1	BRCA2	BTG2	C1orf11	C5orf4	C5orf7
	CACNA2D 2	CACNA2D 3	CAP-1	CARS	CAV1	CD14	CD81
	CDC14B	CDC23	CDK2AP1	CDKN1A	CDKN1C	CDKN2A	CDKN2B
	CHDH	CLCA2	CREBL2	CTNNA1	CUL2	DAB2	DBC2
	DCC	DD5	DDX26	DEC1	DLC1	DLEC1	DLEU1
	DLEU2	DLG1	DMBT1	DOC-1	DPC4	DPH2L	EGR1
	FABP3	FAT	FGL1	FHIT	FLJ10506	FOXD1	FOXP1
	FUS1	FUS2	GAK	GAS1	GAS11	GLTSCR1	GLTSCR2
	GRLF1	HDAC3	HEMK	HIC1	HRG22	HSAL2	HTS1
	HYAL1	HYAL2	IGFBP7	IGSF4	ING1	ING1L	ING4
	LAPSER1	LATS1	LATS2	LDOC1	LOH11CR2A	LRP1B	LUCA3
	MAD	MAFB	MAP2K4	MAPKAPK3	MAPRE3	MCC	MDC
	MEN1	ML-1	MRVI1	MTAP	MXI1	NAP1L4	NBR2
	NF2	NORE1B	NPR2L	OVCA2	PDGFRL	PHEMX	pHyde
	PIK3CG	PINX1	PLAGL1	PRDM2	PTCH	PTEN	PTPN13
	PTPRG	RASSF1	RBBP7	RBM6	RBX1	RECK	RFP2
	RIS1	RPL10	RPS29	RRM1	S100A2	SEMA3B	SF1
	SFRP1	SLC22A1 L	SLC26A3	SMARCA4	ST7	ST7L	ST13
	ST14	STIM1	TCEB2	THW	TP53	TP63	TRIM8
	TSG101	TSSC1	TSSC3	TSSC4	VHL	WFDC1	WIT-1
	WNT4	WT1	WWOX				
Mus musculus	ARF	Cables	Ciao1-pending		CW17R	DLGH1	DLGH3
	DNAJA3	Timp3	Vhlh				

Table 3-1. Continued

Source	Gene name
Arabidopsis thaliana	AT1G14320
Xiphophorus hellerii	CDKN2X
Caenorhabditis elegans	DAF-18 GLD-1
Drosophila melanogaster	D-APC FT l(2)tid l(3)mbn l(3)mbt
Saccharomyces cerevisiae	GRC5
Nicotiana tabacum	NtRb1
Rattus norvegicus	Tsc2

3.1.2 Gene Feature Selection

Since database information varies from gene to gene, some gene features are not available to most genes. For instance, protein structure information can only be found for a few tumor suppressor genes. In such situation, the gene feature selection should guarantee the selected feature could be found in all of the TSGs with only few exceptions, so that genes can be compared with the same set of characters. To achieve this purpose, the following features were chosen to serve as database fields:

- Gene name: Most names of the tumor suppressor genes are HUGO/GDB (The Human Genome Organization/The Genome Database) committee approved.

- Aliases: Most listed gene has one or more aliases which could be used as the major name in some other databases.
- Gene source: Where the interest gene is isolated.
- Expression cell/tissue types: Normalized expression distribution of cell or tissue types.
- Chromosome location: Usually with information on chromosome number and regions.
- Gene structure: Description of gene size and/or organization in the whole genome.
- Protein size: Amino acid number of a usually deduced protein sequence.
- Gene functions: Description of how the gene acts as a tumor suppressor, either through direct control of cell proliferation or indirect affecting the metabolism.
- References: Major information source for the above feature information.

These gene features, except for aliases and chromosome number for few non-human sources, are available to all tumor suppressor genes. The field "References" in gene web pages shows only the most influential publications and databases. This field was not included as a data field in the database.

3.2 Web Page Construction

3.2.1 TSGDB Homepage Creation

The database homepage and sub-pages for the genes were created with HTTP format. The homepage contains three parts: a brief description of the Tumor Suppressor Gene Database (TSGDB) and its usage; a display function to show all the related information of each individual gene, which is realized by a pull-down window containing the names of 178 genes; and a TSGDB query function area at which gene names can be retrieved by querying the database with some features (Fig. 3-2).

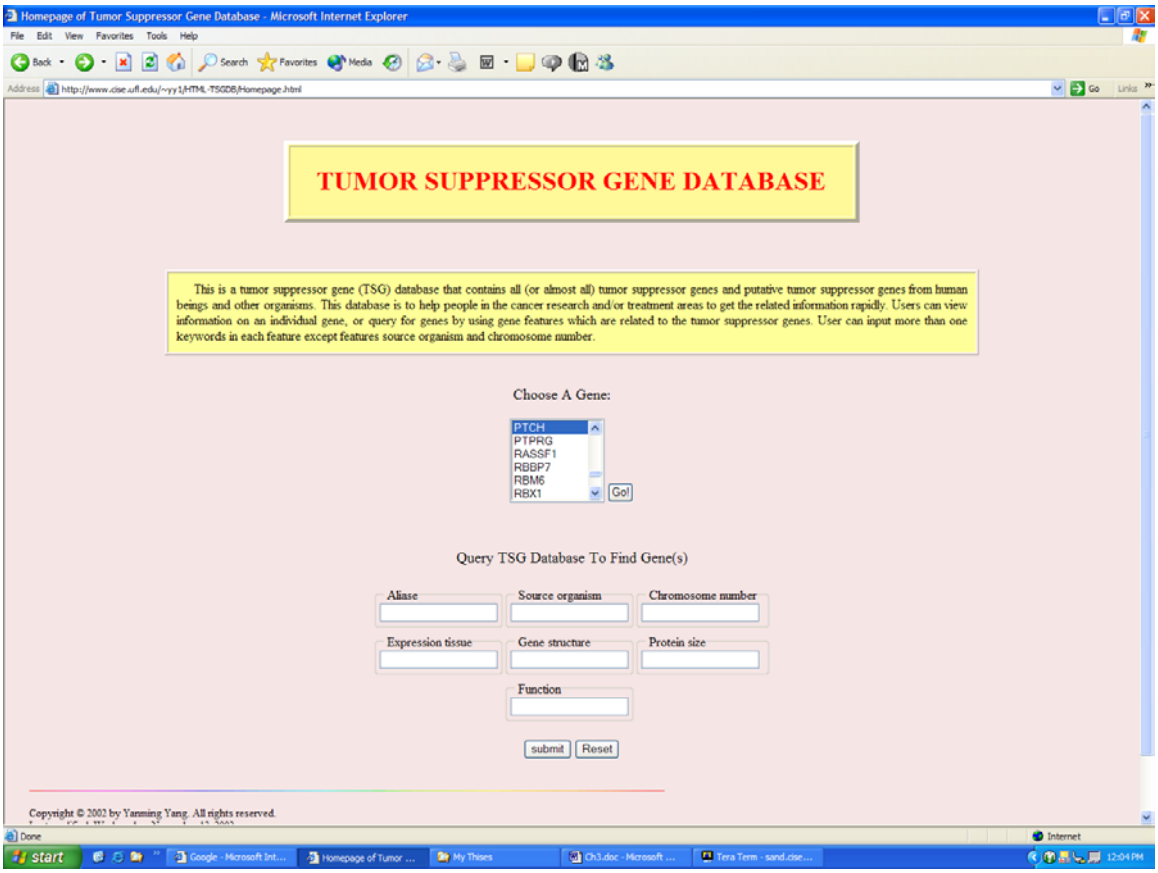

Figure 3-2. TSGDB home page showing the major functions of the database.

The individual gene selection window allows 6 genes visible at the same time, and scrolling the window bar can reveal more genes. The database query function allows users to find tumor suppressor genes if only some features are known, or just some features are intentionally set. For example, to find that chromosome number 11 contains which tumor suppressor genes, to discover which genes are expressed in the tissues of lung or mammary gland, and so forth. Search scope can be widened by putting more query entries in each feature, and can be narrowed down by entering more features at the same query. The relationships between various parameters are "AND", while within the same parameter, the relationships between various input entries are "OR", as expressed in relational algebra.

3.2.2 Construction of Individual Gene Web Pages

Once a gene is selected, it will be linked to the particular web page for that gene. The selection and linkage function was achieved by using the HTML form method, a JavaScript-driven function. JavaScript is a general-purpose scripting language that was originally derived from Netscape's LiveScript, and later developed by Sun Microsystems with Java programming language [34]. Unlike Java, which can be used to develop entirely standalone applications, JavaScript works primarily with Web pages, and usually included within HTML files and interpreted line-by-line by the browser without the need to execute an application. The JavaScript and partial form content for the pull-down window functions were shown as in Figure 3-3.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN
"html://www.w3c.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title> Homepage of Tumor Suppressor Gene Database </title>

<script LANGUAGE="JavaScript">
<!-- Hide JavaScript
function MsgBox(textstring) { alert(textstring) }
function goToLink(form){ location.href=form.options[form.selectedIndex].value; }
function new_window(url) { link = window.open(url,"Link","toolbar=1, location=1,
       directories=1, status=1,menubar=1,scrollbar=1, resizable=1, width=1000,
       height=500, left=100, top=120"); }
// - End JavaScript - -->
</script>
......
<!-- Drop-down menu navigation as following -->
<form name="URLmenu">
   <SELECT name="choices" size=6>
      <OPTION value="101F6.html">101F6
      <OPTION value="ABR.html">ABR
      <OPTION value="ADPRTL3.html">ADPRTL3
      ......
      <OPTION value="WT1.html">WT1
      <OPTION value="WWOX.html">WWOX
   </SELECT>
   <INPUT type="button" value="Go!" OnClick="goToLink(this.form.choices)">
</form>
```

Figure 3-3. JavaScript and form method for pull-down window function

In each individual gene page, the above described features were included, i.e., gene name, aliases, source organism, expressed cell/tissue type, chromosome location, gene structure, protein size, gene functions and major related references (Fig. 3-4).

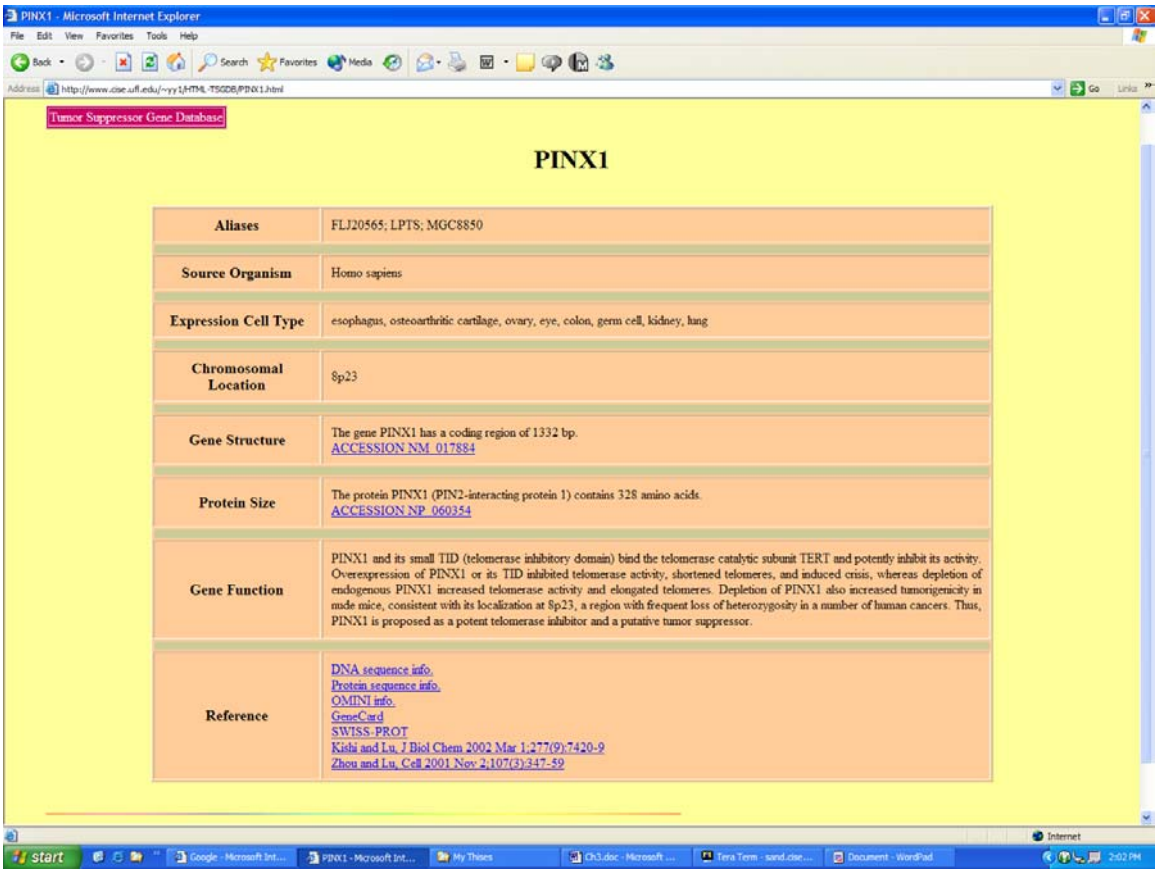

Figure 3-4. A sample Web page for a tumor suppressor gene.

Within each page, all the reference links are alive. If users need other information related to the gene, they can dig further from these linked Web sites, each of which contain more detailed information on certain aspects.

CHAPTER 4
IMPLEMENTATION OF TSGDB WITH RELATIONAL MODEL

Among the data models described in Chapter 2, relational model was selected to implement the database for tumor suppressor genes. The relation data model was originally introduced by Dr. Codd in 1970 [35]. It is currently the most popular data model. The mathematical simplicity and ease of visualization of the relational data model have contributed to its success. In addition, the relational data model is not as biased towards a particular kind of structure as the hierarchical model.

4.1 Relational Model Concept

The relation data model is based on the mathematics of set theory. This model structures the logical view of data around two mathematical constructs: domains (a.k.a. data types) and relations. The name relational comes from "relation" as known and widely used in mathematics, although the definition of relation is slightly extended in database theory.

A domain is simply a set of values, together with its associated operators, which is equivalent to the notion of types in programming languages. A relation over the domains D_1, D_2, ... D_n is simply a subset of the Cartesian product, the usual notation is R "included in" $D_1 \times D_2 \times ... \times D_n$. An element of the Cartesian set is called a tuple. A database can be considered as a collection of "relation valued" variables, i.e., variables whose value is a relation, a.k.a. table, together with the set of integrity constraints that the data must satisfy [36]. Each domain that defines a relation is associated with a string label, i.e., a column name. Then, a column is the association between a column name and a domain.

A relation header is then a set of columns. Then a tuple becomes the mapping between each column in the relation header and a value. And a relation is a set of tuples. Because column names are unique in a "relation header," the positional ordering in the mathematical definition becomes inessential; therefore each data value in a tuple can be identified by its column name. This essentially makes programming convenient.

Besides the structure of data, the relational model also defines the means for data manipulation (relational algebra and relational calculus) and the means for specifying and enforcing data integrity (integrity constraints). That's the basics of the relational model. In spite of its apparent simplicity, the relational model is very rich and powerful, and is a wonderful tool for doing real software engineering as well as theoretical research.

In another words, with relational data model, databases are represented as groups of related tables. The relation itself corresponds to the familiar notion of a table: a relation is a collection of tuples, each of which contains values for a fixed number of attributes (columns). Because of their resemblance to an unstructured sequence of records, relations are sometimes referred to as flat files. Theoretically, each tuple in a relation must be unique, i.e., there can be no duplicates. Columns of a table are also called attributes. Other commonly used terms for attribute include “property” and “field”. A tuple (table row) is an instance of an entity or relationship or whatever is represented by the relation.

Another important term in a relational model is called “key”. A key is a single attribute or a combination of attributes whose values uniquely identify the tuples of the relation. This means that each row must have a different value for the key attribute(s).

The relational model requires that every relation have a key and the following conditions must be satisfied:

- No two tuples may have the same key value.
- Every tuple must have a value for the key attribute (the null value is not allowed for the key fields).
- Duplicate tuples are not permitted. If two tuples are entered with the same value for each and every attribute, they are considered to be the same tuple.
- No ordering of tuples within a relation is assumed.

However, some restrictions, especially the last two restrictions, are sometimes circumvented. Duplicated tuples can be entered by assigning unique line or tuple numbers to each entry, thus assuring that it is unique. Or even identical tuples are entered for the reason of query efficiency. Ordering of tuples also can be controlled by some ordering methods.

4.2 Implementation of A Standalone Database

On the basis of the relational model rules, the database for tumor suppressor genes was created. This database was implemented with SQL*Plus of the Oracle Relational Database Management System, which is an interactive user interface to the Oracle DBMS. SQL*Plus delivers a full implementation of SQL and PL/SQL with a rich extensions, including functions in configuring database environment.

4.2.1 Table Creation

A schema is a description of the database, rather than the database itself. The tumor suppressor gene database only contains a single table--TGS. This schema is composed of eight data fields. Therefore, the database schema can be shown as following (Fig. 4-1):

TSG

Name	Alias	Source	Celltype	Chromosome	Structure	Protein	Function

Figure 4-1. TSG schema.

To create the table TSG, the following conversion of relational data definition and restrictions to Oracle syntax was used (Fig. 4-2):

```
CREATE TABLE TSG (
       name          VARCHAR(20)          NOT NULL,
       alias         VARCHAR(200),
       source        VARCHAR(50)          NOT NULL,
       celltype      VARCHAR2(1000),
       chromosome    VARCHAR(50),
       structure     VARCHAR2(2000)       NOT NULL,
       protein       VARCHAR2(2000)       NOT NULL,
       function      VARCHAR2(4000)       NOT NULL,
       PRIMARY KEY (name)
);
```

Figure 4-2. SQL statements defining the TSG schema

Because the Oracle SQL*Plus is a DOS/UNIX manipulated system, tables can be created directly from the UNIX shell once logging into the Oracle DBMS. The above table creation description was input as command lines to get the table TSG created.

4.2.2 Data Treatment and Bulk-loading

After the table was created, the data are to be loaded into the table. The raw data collected for constructing the Web pages of each tumor suppressor gene have to be treated before the loading since the data definitions of each field in the table schema

constrain the application of the original gene feature statements. For example, some feature of a gene may not exist (null value), and the accession number in the structure and protein references may confuse the query execution. Also, during the bulk-loading, each line of the data is considered a separate tuple of the table, thus data for each gene must be put in a single line. Moreover, bulk-loading controlling process requires fields be delimited with a special symbol. Therefore, vacant feature must be treated carefully to avoiding field displacement.

The Oracle DBMS bulk loader was used to input the data. The application of Oracle bulk loader requires two files: a control file and a data file. A control file specifies how data should be loaded into the database, while the data file specifies what data should be loaded. The data file was constructed by refining raw data as described above. The control file was created as following (Fig. 4-3).

```
LOAD DATA
INFILE test1.dat
INTO TABLE TSG
FIELDS TERMINATED BY '|'
(name, aliase, source, celltype, chromosome, structure, protein, function)
```

Figure 4-3. Bulk loader control file.

The Oracle bulk loader is called “sqlldr”. It is a DOS/UNIX level command which should be issued directly from the DOS/UNIX shell. The following command was used to execute the bulk loading:

sqlldr yy1@oradb control=bulk_load.ctl log=bulk_load.log bad=bulk_load.bad

The log file and the bad file respectively store information on the loading procedure and unloaded, if any, part of the data file, which can be used for debugging in case something goes wrong during loading.

4.2.3 SQL Manipulation

Once the data were loaded successfully, the database is subject to SQL manipulations, such as querying, inserting, deleting, renaming, and so forth. Because SQL is based on set and relational operations with certain modifications and enhancements, a typical SQL query has the form:

SELECT $A_1, A_2, \ldots, A_i, \ldots, A_n$

FROM $r_1, r_2, \ldots, r_j, \ldots, r_m$

WHERE P

A_i represents attributes; r_j represents relations; and P is a predicate.

This query is equivalent to the relational algebra expression:

$$\prod_{A_1, A_2, \ldots, A_i, \ldots, A_n} (\sigma_P (r_1 \times r_2 \times \ldots \times r_j \times \ldots \times r_m))$$

The result of an SQL query is a relation. Web-based users access the database only with the SELECT clause to query a result, while the database owner can use all the SQL utilities to maintain and update the database.

The SELECT clause corresponds to the projection operation of the relational algebra. It is used to list the desired attributes in the result of the query. The WHERE clause corresponds to the selection predicate of the relational algebra. It consists of a predicate involving attributes of relations that appear in the FROM clause. For example, with the following query command, the result shows gene names and source organisms of non-human genes (Fig. 4-4):

For the updating purpose, new genes (tuples in the table) can be inserted or the existing tuples can be modified by using standard SQL commands. If the database needs to be expanded to include information in other categories (relations), e.g., adding tables

DISEASE and ORGANIZATION, the database schema can be modified and new table data fields can be defined.

```
SQL> SELECT name, source FROM TSG WHERE not source like '%Homo%';

NAME                 SOURCE
-------------------- ----------------------------
ARF                  Mus musculus
AT1G14320            Arabidopsis thaliana
Cables               Mus musculus
CDKN2X               Xiphophorus hellerii
Ciao1-pending        Mus musculus
CW17R                Mus musculus
DAF-18               Caenorhabditis elegans
D-APC                Drosophila melanogaster
DLGH1                Mus musculus
DLGH3                Mus musculus
DNAJA3               Mus musculus

NAME                 SOURCE
-------------------- ----------------------------
FT                   Drosophila melanogaster
GLD-1                Caenorhabditis elegans
GRC5                 Saccharomyces cerevisiae
l(2)tid              Drosophila melanogaster
l(3)mbn              Drosophila melanogaster
l(3)mbt              Drosophila melanogaster
NtRb1                Nicotiana tabacum
Timp3                Mus musculus
Tsc2                 Rattus norvegicus
Vhlh                 Mus musculus

21 rows selected.
```

Figure 4-4. Sample result of an SQL query to TSGDB

CHAPTER 5
IMPLEMENTATION OF A WEB-BASED DATABASE SYSTEM

5.1 Building Web-based Information System

With the development of Web-based information systems, static HTML pages which are usually stored on the file system of the connected machines seem no longer to satisfactorily meet the newly arisen requirements of the information world. The explosive growth of information volume makes it impossible to store the desired HTML files into the disks of the Web users' machines in the form of HTML pages. So, instead of static pages, an application program is needed to run the Web server for receiving client requests, retrieving the relevant data from the source, and packing them into HTML format. Even the recently emerged "semistructured" XML databases, which store data into the XML format, need an application program to connect to the DBMS for retrieving the XML files or fragments [37]. In another point of view, if a database system remains isolated from the Web, the scope of users will become very limited and the significance of the database will be dramatically undermined. Therefore, connecting databases to the Web and offering users with dynamically generated HTML pages constitute the primary features of a modern informatics system.

5.1.1 System Architecture

The traditional coupling of a Web client and a Web server becomes no longer suitable for dynamically constructing Web pages. Instead, a third part must be added to the system--an application program that runs on the Web server and serves data from an underlying database. This system scheme is also referred to as Web-powered database.

The architecture of the current system, i.e., the Web-based database system for tumor suppressor genes, is indicated in the following (Fig. 5-1).

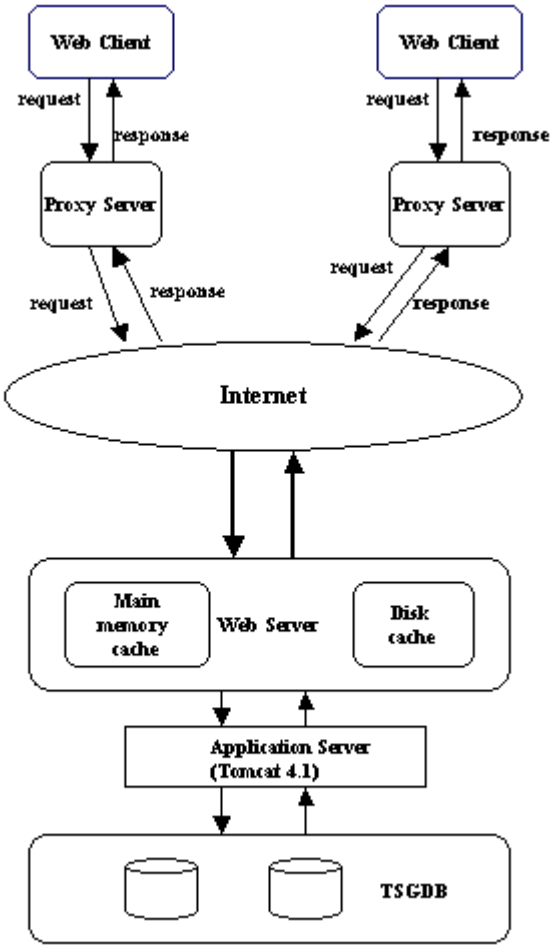

Figure 5-1. Architecture of Web-powered TSGDB

In this architecture, there are three tiers: the database back-end (TSGDB), the Web/application server (Apache/Tomcat Server) and the Web clients. In order to generate dynamic content, the Web server must execute the application program--a servlet in this system. The servlet acquires user requests, connects to the TSGDB, executes the client query, gets the results and packs them in HTML format in order to return to the user. The underlying database in this particular system is driven by the Oracle DBMS system with the thin driver.

5.1.2 Using Servlet as A Middle-layer Application

Servlet technology represents a new generation in solving dynamic Web page construction and cooperation between users and underlying databases. Because of the advantages analyzed in Chapter 2, a Java servlet is created in the current project to accomplish the relevant tasks. This servlet, named Params, extends Java class HttpServlet, and acts as a middle layer between a request coming from the Web browser and the TSGDB. The functions of this servlet include:

- Read request data sent by users.
- Call utility routines to process user's input.
- Make connection to the database.
- Query the database with the processed user input.
- Retrieve query result
- Pack the query result to HTML form and return it to the user.

Partial of the servlet program file is shown as following (Fig. 5-2):

```
public class Params extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse
                    response) throws ServletException, IOException {

                String str = null;
                String s = "Error Message: ";
                String s1= "";
                Connection conn = null;
                Statement stmt = null;
                ResultSet result = null;

                //Load the Oracle JDBC driver
                try{
                Class.forName("oracle.jdbc.driver.OracleDriver");
                }catch (Exception e1){s = s + e1;}

                //Connect to Oracle database
                try{
                String sourceURL = "jdbc:oracle:thin:@oracle:1521:oradb";
                String username="yy1";
                String password=TSGDBUtilities.getPassword();
                // get actual connection
                conn = DriverManager.getConnection(sourceURL, username,
```

Figure 5-2. Partial code of the interface servlet program.

```
      password);
}catch(Exception e){s1 = s1 + e;}

response.setContentType("text/html");
PrintWriter out = response.getWriter();

String title = "Input Query Parameters";
String ali = request.getParameter("aliase");
String src = request.getParameter("source");
String chrom = request.getParameter("chromosome");
String cel = request.getParameter("cell");
String gen = request.getParameter("gene");
String protn = request.getParameter("protein");
String fnc = request.getParameter("function");
String query = null;
String [] aliArray = new String[50];
String [] cellArray = new String[50];
String [] geneArray = new String[50];
String [] protnArray = new String[50];
String [] fncArray = new String[50];

//load the requested data to arrays
aliArray = TSGDBUtilities.parseToArray(aliArray, ali);
cellArray = TSGDBUtilities.parseToArray(cellArray, cel);
geneArray = TSGDBUtilities.parseToArray(geneArray, gen);
protnArray = TSGDBUtilities.parseToArray(protnArray, protn);
fncArray = TSGDBUtilities.parseToArray(fncArray, fnc);

if (!ali.equals("") || !src.equals("") || !chrom.equals("") ||
!cel.equals("") || !gen.equals("") || !protn.equals("") ||
!fnc.equals(""))
       query = TSGDBUtilities.listNames(aliArray, cellArray,
        geneArray, protnArray, fncArray, src, chrom);

if (query != null){
   out.println(ServletUtilities.headWithTitle(title) +
   "<BODY BGCOLOR=\"#FDF5E6\">\n" +
   "<H2 ALIGN=CENTER>" + title + "</H2>\n" +
   "<table><tr><td width=\"50%\"></td><td>" +
   "<UL>\n" +
   "  <LI><B>Aliase</B>: "
   + request.getParameter("aliase") + "\n" +
   "  <LI><B>Source</B>: "
   + request.getParameter("source") + "\n" +
   "  <LI><B>Chromosome</B>: "
   + request.getParameter("chromosome") + "\n" +
   "  <LI><B>Cell Type</B>: "
   + request.getParameter("cell") + "\n" +
   "  <LI><B>Gene Structure</B>: "
   + request.getParameter("gene") + "\n" +
   "  <LI><B>Protein</B>: "
   + request.getParameter("protein") + "\n" +
   "  <LI><B>Function</B>: "
   + request.getParameter("function") + "\n" +
                    "</UL>\n" + "</td><td></td></td></tr> <tr><tr>
        <tr></table>" + "<p><p><p><p>" +
       "<center> <font size=\"6\" color=\"#0000FF\"> Query Result
       </font><p>" + "\n" + "<font color=\"#FF3300\">" +
     "================================================" +
     "</font></center><p>" + "</BODY></HTML>");
```

Figure 5-2. Partial code of the interface servlet program (continued).

```
            try {
               if (conn != null){
                   stmt = conn.createStatement();
                   result = stmt.executeQuery(query);

                   while (result.next()){
                               int i  = 1;
                               out.println("<center><font size=\"4\" > " +
                          result.getString(i) + "</font></center>");
                   }
                   }
                   }catch(Exception e2) {};
            }else
                   out.println("<BODY BGCOLOR=\"#FDF5E6\">\n" + "<font
                size=\"4\" > " + "No Parameters Input. Try Again." +
               "</BODY></HTML>");

            try{
            if (conn != null)
                   conn.close();
            }catch(Exception err){};
   }
```

Figure 5-2. Partial code of the interface servlet program (continued).

The servlet is driven by Apache Tomcat 4.1 which is a Web server supporting servlets and Java Server Pages (JSP). To make the servlet work, the CLASSAPTH was set to include "/usr/local/java/tomcat/common/lib/servlet.jar" and "/usr/local/libexec/oracle-client/product/8.1.6/jdbc/lib/classes12.zip". In addition, a non-SSL Coyote HTTP/1.1 connector port and a Coyote/JK2 AJP 1.3 connector port were redefined.

5.2 Query Mechanism and Result Formatting

5.2.1 Transferring User Inputs to Queries

The database for tumor suppressor genes can be queried by inputting some gene features at home Web page of the database. The "submit" button, once pushed, takes "post" action on the servlet Params which is driven by Tomcat at the machine "rain" (IP address: 128.227.205.19). The servlet program functions through some interface programs in the javax.servlet.http package, which include sub-interfaces

HttpServletRequest and HttpServletResponse. HttpServletRequest extends the ServletRequest interface to provide request information for HTTP servlets, while HttpServletResponse extends the ServletResponse interface to supplies HTTP-specific functionality in sending a response. In the servlet program, an HttpServletRequest object "request" and an HttpServletResponse object "response" were created for utilizing the relevant methods of these classes. User-input parameters are transformed into strings by calling getParameter() method of the HttpServletRequest class, which takes the value, as the argument, of input names of the HTTP form function. The available parameters that a user can choose include gene aliases, chromosome location, organism source for the gene, normally expressed tissue types, gene structure (e.g., nucleotide numbers), protein (e.g., amino acid numbers), and gene functions.

The user input requests can not be used directly as query entries to retrieve information from the database since the format of the input is not compatible with the SQL query format. Therefore, user inputs must be processed before they are entered into the querying procedure. Some utility programs were created to accomplish the desired processing. These utility programs are Java classes, including ServletUtilities, and TSGDBUtilities (Appendices A.1 and A.2).

In processing the user inputs, the fist step is to scan the input stream so that it can be parsed into "meaningful" tokens. For example, if the input stream for the parameter "Aliases" is like "MDB, AAPC; SEN4?p53 GUD'P70, ~ ZNF162", only strings MDB, AAPC, SEN4, p53, GUD, P70 and ZNF162 are taken into account because other symbols in the stream do not contribute significant values to the gene feature. The next step includes the packing of the "meaningful" tokens into an array so that they can be used in

the construction of SQL query string. This task is accomplished by calling the method parseToArray of the TSGDBUtilities class. On the consideration of the nature of parameters "Source organism" and "Chromosome number", only one choice is allowed, i.e., users only can input one source organism and/or one chromosome number at once. Extra inputs for these features will be trimmed off. Thus, no array packing is needed for these features.

5.2.2 Establishing Connections to TSGDB

The servlet interface is also responsible for connecting to the underlined database via the JDBC functionality. To make the connection, the Oracle JDBC thin driver was used, which is a Type 4 driver. It uses Java sockets to connect directly to Oracle DBMS, based on its own implementation of a TCP/IP version of Oracle's SQL*Net. Written entirely in Java and platform-independent, it can actually connect to any Oracle database of version 7.2 and higher. It does not need Oracle software for client side, but requires a TCP/IP listener on the server side. Prior to accessing the data from the database, the Java servlet imports the JDBC classes, registers the JDBC driver, and then establishes the connectivity with the database. For class registration, Class.forName is called with the argument "oracle.jdbc.driver.OracleDriver".

5.2.3 Querying Database and Formatting Results

The query to the gene database is only limited to find gene names from the relevant features. Once a gene name is revealed, a complete set of information can be retrieved from the "Choose A Gene" function of the homepage. To query for other features of the genes does not make sense in a practical view.

Queries to be sent to the database take a general form of relational algebra as:

Π_{name} (σ <A11 OR A12 …, OR A1n> AND <A21 OR A22 …, OR A2m> …, AND <A71 OR A72 …, OR A7t> (TSG))

Where A11, A12… A1n represent "meaningful" inputs to feature "Aliases"; A21, A22 …A2m indicate inputs for feature "Expression tissues", and so on. Note that attribute choices for "Source organism" and "Chromosome number" only allow a single phrase for each, i.e., the extra inputs will be ignored by the processing program. The relationship between multiple phrases in the same feature input is "OR", while it is an "AND" relationships between multiple features. As an example for a user input as in Figure 5-3, the assembled query string is like the following with the SQL format:

SELECT name FROM TSG WHERE ((Aliases LIKE '%FPC%' or Aliases LIKE '%ATDC%' or Aliases LIKE '%TP63%') and (Tissue LIKE 'heart' or 'Tissue LIKE '%lung%') and Chromosome like '11%').

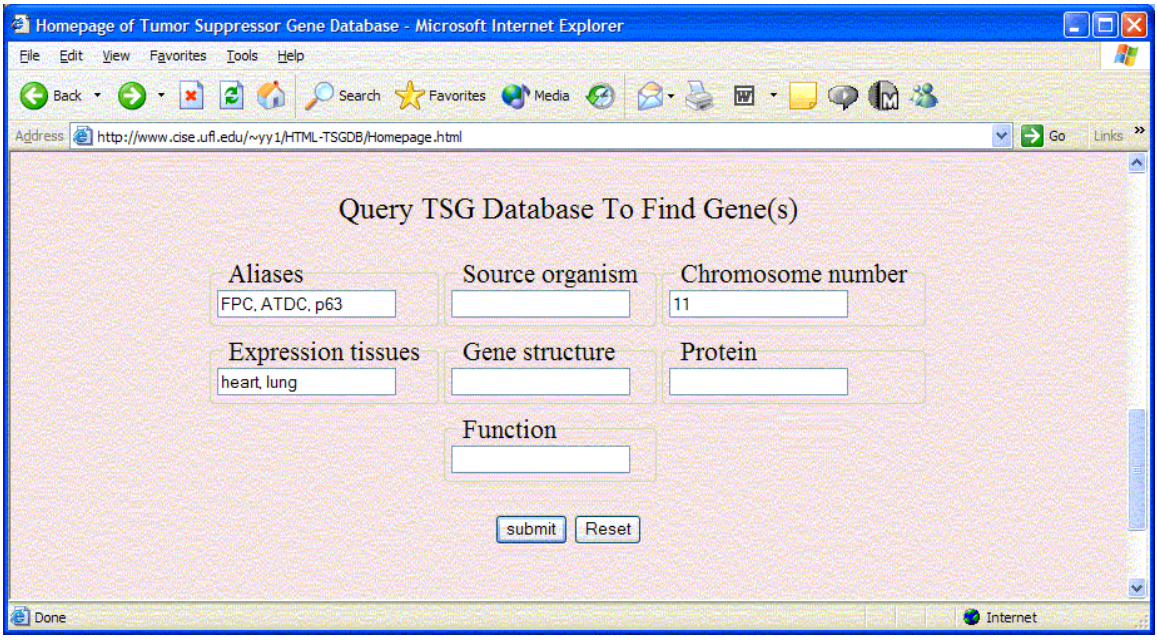

Figure 5-3. A sample query input for the TSGDB.

The returned result from a query is packed into a ResultSet object, and is then retrieved one by one via calling getString() method of the object. Dynamic generation of

HTTP-formatted pages was realized by using some functionalities of the servlet Params, which include the method headWithTitle() of the ServletUtilities class and getParameter() method of HttpServletRequest to display the Web page head and query parameters. The search result shows all parameter-matched genes, if there is any, in a list. In the case of empty input, i.e., the user did not put any parameters for gene features but executed the query, a simple page showing the message "No Parameters Input. Try Again." will be displayed. At such situation, no database connection is initiated for the sake of saving computing resources. A typical dynamically composed Web page showing a query result is like the following (Fig. 5-4).

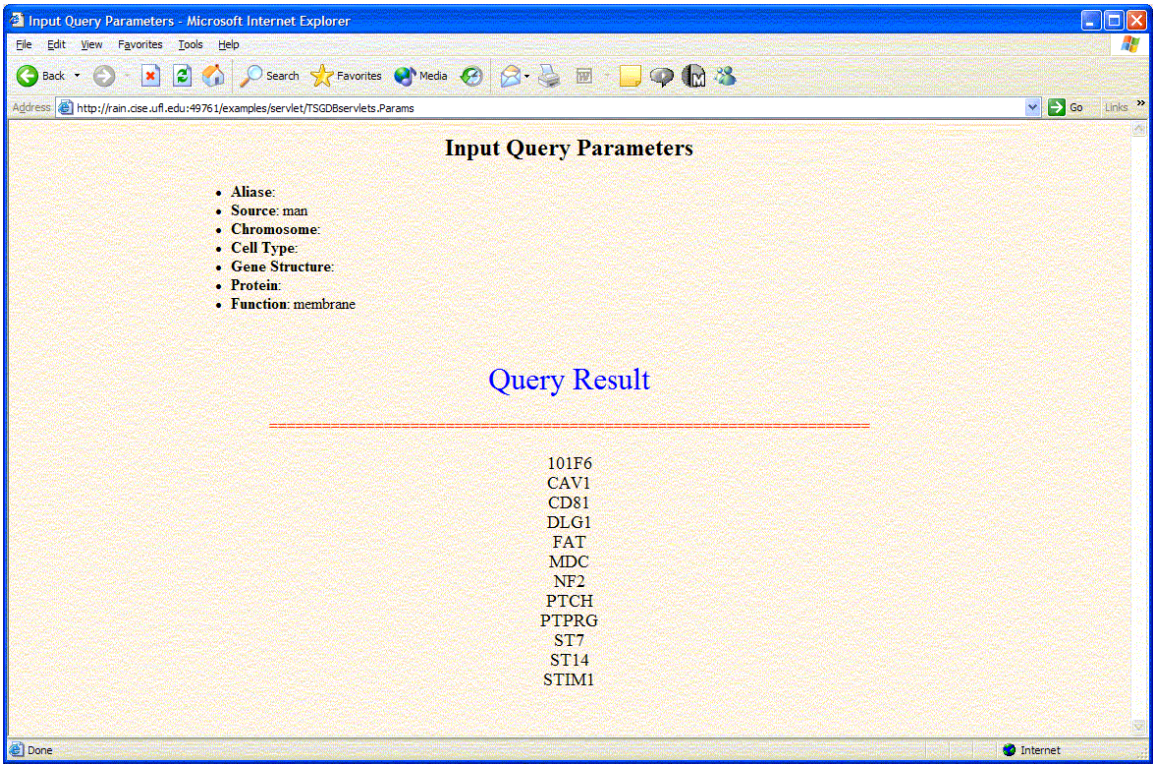

Figure 5-4. A sample Web page displaying the result of a typical query.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

The web-based database system for tumor suppressor genes (TSGDB) is a newly-developed informatics system aimed to provide a comprehensive information source for medical studies in caner-related areas and for general users with interests in cancer suppressing mechanism and theories in molecular biology and genetics. This database contains 178 tumor suppressor genes, ever discovered so far, from human beings, mice, fruitflies and other organisms. The information scope of each gene covers gene's aliases, chromosome location, expression tissues, gene structure and functions, etc. All the information in the database is based on scientific publications and Web-published, verified data from other database sources. Therefore, the information contained by this database is valid, accurate and reliable.

The database was designed to contain eight most important features for each gene, and was implemented with relational data model for the system management. The underlying database is run by Oracle SQL*Plus system, and incorporated into a Web-powered database system through a middle-layered application program. The interactive operations between users and the database are achieved by the servlet application program. This servlet acts like a common gateway interface (CGI) but with many advantages, whose functions include to collect users' input requests, process users' requests as parameters, construct SQL query string, connect to the database, and format and return query results to users. The servlet application is driven by Apache Tomcat, and the JDBC is performed by Oracle thin driver. The overall performance level of the

database system is very high in terms of query processing speed. In conclusion, TSGDB is a highly informative, performance-effective and reliable Web-based informatics system that can be used for scientific research and general knowledge acquisition in cancer related fields.

Because of the limitations in time and data resources, the database system was left with many aspects for further improving. For instance, more relations (tables in the SQL implementation) could be added so that within the database multiple relationships could be found between a particular gene and other relevant data, such as disease symptoms, population distributions, etc. Also, the web site could be further improved so that it looks more professional than the current appearance does. These aspects are all included in the future work scheme, and could be completed.

APPENDIX
UTILITY PROGRAMS AND HUMAN GENE FUNCTIONS

A.1 ServletUtilities

The ServletUtilities class has following functions: to set the title of a HTML file, to finalize the DOCTYPE string and to get an integer value. The program is shown as following:

```
package TSGDBservlets;
import javax.servlet.*;
import javax.servlet.http.*;


public class ServletUtilities {
	public static final String DOCTYPE = "<!DOCTYPE HTML PUBLIC \"-
		//W3C//DTD HTML 4.0 " + "Transitional//EN\">";

	public static String headWithTitle(String title) {
		return(DOCTYPE + "\n" + "<HTML>\n" + "<HEAD><TITLE>" + title
			 + "</TITLE></HEAD>\n");
	}

	 /** Read a parameter with the specified name, convert it to an int, and
	 *  return it. Return the designated default value if the parameter
	 *  doesn't exist or if it is an illegal integer format. */

	public static int getIntParameter(HttpServletRequest request,
			String paramName, int defaultValue) {
		String paramString = request.getParameter(paramName);
		int paramValue;
		try {
			paramValue = Integer.parseInt(paramString);
		} catch(NumberFormatException nfe) { // null or bad format
			paramValue = defaultValue;
		}
		return(paramValue);
	}
}
```

A.2 TSGDBUtilities

The functions of this program include packing chosen meaningful tokens from those returned by the scanning function, packing those tokens into array and producing an SQL formatted query string. The major parts of the program are shown as following:

```
public static String listNames(String[] ali, String[] cel, String[] gen,
                        String[] protn, String[] fn, String s, String chr){

      String src = s.trim();
      String chrom = chr.trim();
      int i = 0, j = 0, k = 0, l = 0, m = 0;
      String query = "SELECT name FROM TSG WHERE (";

      while (ali[i] != null){
         query = query + "aliase like '" + ali[i] + ";%' or aliase like '" +
            ali[i] + " %' or aliase like '% " + ali[i] + ";%' or aliase like '%
            " + ali[i] + " %' or aliase like '% " + ali[i] + "' or aliase like
            '" + ali[i] + "' or aliase like '% " + ali[i] + " %' or aliase like
            '" + ali[i] + " %'" ;
         i++;
         if(ali[i] != null)
            query = query + " or ";
      }

      if (cel[j] != null)
         if (ali[0] != null)
            query = query + ") and (";
      while (cel[j] != null){
         query = query + "celltype like '%" + cel[j] + "%' " ;
         j++;
         if(cel[j] != null)
            query = query + " or ";
      }

      if (gen[k] != null)
         if (ali[0] != null || cel[0] != null)
            query = query + ") and (";
      while (gen[k] != null){
         if (gen[k].equalsIgnoreCase("bp"));  //do nothing
         else
            query = query + " structure like '% " + gen[k] + " %' " ;
         k++;
         if(gen[k] != null && !gen[k].equalsIgnoreCase("bp"))
            query = query + " or ";
      }

      if (protn[l] != null)
         if (ali[0] != null || cel[0] != null || gen[0] != null)
            query = query + ") and (";
      while (protn[l] != null){
         if  (protn[l].equalsIgnoreCase("aa")  ||protn[l].equalsIgnoreCase("a.a.")
             || protn[l].equalsIgnoreCase("amino acids"));
         else
            query = query + " protein like '% " + protn[l] + " %' " ;
         l++;
         if(protn[l] != null && (!protn[l].equalsIgnoreCase("aa") &&
            !protn[l].equalsIgnoreCase("a.a.") &&
            !protn[l].equalsIgnoreCase("amino acids")))
            query = query + " or ";
      }

      if (fn[m] != null)
         if (ali[0] != null || cel[0] != null || gen[0] != null || protn[0] !=
            null)
            query = query + ") and (";

      while (fn[m] != null){
         query = query + " function like '%" + fn[m] + "%' " ;
```

```
      m++;
      if(fn[m] != null)
         query = query + " or ";
   }

   if (!src.equals("")){
      String substr = null;
      if (src.indexOf(' ') < 0)
         substr = src;
      else
         substr = src.substring(0, src.indexOf(' '));
      if (ali[0] != null || cel[0] != null || gen[0] != null || protn[0] !=
         null || fn[0] != null)
         query = query + ") and (";
      if (substr.equalsIgnoreCase("homo") || substr.equalsIgnoreCase("human")
         || substr.equalsIgnoreCase("man")
          || substr.equalsIgnoreCase("men") )
         query = query + " source like '%Homo%' ";
      else if ( substr.equalsIgnoreCase("mouse") ||
             substr.equalsIgnoreCase("mice") )
         query = query + " source like '%Mus%' ";
      else if (substr.equalsIgnoreCase("rat") ||
             substr.equalsIgnoreCase("rats"))
         query = query + " source like '%Rattus%' ";
      else if (substr.equalsIgnoreCase("Nicotiana") ||
             substr.equalsIgnoreCase("tobacco"))
         query = query + " source like '%Nicotiana%' ";
      else if (substr.equalsIgnoreCase("yeast") ||
             substr.equalsIgnoreCase("Saccharomyces"))
         query = query + " source like '%Saccharomyces%' ";
      else if (substr.equalsIgnoreCase("fruitfly") ||
             substr.equalsIgnoreCase("fruit fly") ||
             substr.equalsIgnoreCase("Drosophila") )
         query = query + " source like '%Drosophila%' ";
      else if (substr.equalsIgnoreCase("nematode") ||
             substr.equalsIgnoreCase("Caenorhabditis"))
         query = query + " source like '%Caenor%' ";
      else if (substr.equalsIgnoreCase("Arabidopsis") )
         query = query + " source like '%Arabidopsis%' ";
      else if (substr.equalsIgnoreCase("fish") ||
             substr.equalsIgnoreCase("Xiphophorus"))
         query = query + " source like '%Xipho%' ";
      else
         query = query + " source like '%" + substr + "%' ";
   }

   if (!chr.equals("")){
      if (ali[0] != null || cel[0] != null || gen[0] != null || protn[0] !=
         null || fn[0] != null || !src.equals(""))
         query = query + ") and (";
      if (chr.equalsIgnoreCase("x") )
         query = query + " chromosome like 'X%' or chromosome like 'x%' ";
      else if (chr.equalsIgnoreCase("y") )
         query = query + " chromosome like 'Y%' or chromosome like 'y%' ";
      else
         query = query + " chromosome like '" + chr + "p%' or chromosome
                 like '" + chr + "q%' or chromosome like '" + chr + " %'  ";
   }
   query = query + ")";
   return query;
}
```

```
/**
 * Save a String to a file with the given filename.
 * @return boolean did the save work.*/
public static String writFile(String fileName, String msg){
   try{
      FileWriter out=new FileWriter(fileName);
      out.write(msg);
      out.close();
   }catch (Exception e){
      //System.out.println(e);
      System.exit(-1);
   }
   return fileName;
}


//parse and put input string into array
public static String [] parseToArray (String[] stringArray, String string){

   try{
      int index = 0;
      Scanner scan = new Scanner(writFile("myfile.txt", string));
          /*string is the requested data from the form input */
      String s = scan.getNextToken().tokenValue;
      System.out.println(string);
      while (s != null){
         System.out.println(s);
         //s = scan.getNextToken().tokenValue;
         char c = s.charAt(s.length() - 1);
         if (c != '.' && c != '"'){
            if(((s.charAt(0) >= 65 && s.charAt(0) <= 90)||(s.charAt(0) >= 97
                 && s.charAt(0) <= 122)||(s.charAt(0) >= 49 && s.charAt(0)
             <=
                 57))) { //the fist character on belong A-Z or a-z or 1-0

               stringArray[index++] = s;
               int ind = index-1;
               System.out.println("stringArray[" + ind + "] is:  " +
                         stringArray[ind]);
            }
         }
         else { //Token ended with "." or """
            String substr = null;
            if (s.charAt(0) == '"')
              substr = s.substring(1, s.length()-1);
            else substr = s.substring(0, s.length()-1);
            if ((substr.charAt(0) >= 65 && substr.charAt(0) <= 90)||
                 (substr.charAt(0) >= 97 && substr.charAt(0) <=
                 122)||(substr.charAt(0) >= 49 && substr.charAt(0) <= 57)) {
                 /* the fist character on belong A-Z or a-z or 1-0  */
               stringArray[index++] = substr;
               int js = index - 1;
               System.out.println("stringArray[" + js + "] is:  " +
                                  stringArray[js]);
            }
         }
         s = scan.getNextToken().tokenValue;
         if (s == null) break;
      }
   }catch (Exception e){}
   return stringArray;
```

A.3 Human TSGs and Their Functions

GENE NAME	FUNCTION SUMMARY
101F6	101F6 encodes an integral plasma membrane protein with six trans-membrane helices. Both termini of each helix are in the cytoplasm.
ABR	brain homology to guanine nucleotide GTPase-activating proteins. involved in reciprocal translocations with ABL oncogene in Philadelphia chromosome-positive chronic myelogenous leukemia. Rho family regulate coordinate cellular signaling breakpoint cluster associated medulloblastoma.
ADPRTL3	encoding an ADP-ribosyltransferase-like protein. altered in several types of solid malignant tumors, a candidate tumor suppressor gene.
ANP32C	stimulate transformed focus formation cotransfected into NIH 3T3 cells tumor promoters. modulate the oncogenic prostate cancer. inhibits several types of cancers tumorigenic. Tumor suppressor function of 25 amino acid region divergent PP32.
ANP32D	suppressor Gene ANP32D is 90% identical to pp32 and tumorigenic. Like ANP32C, ANP32D can stimulate transformed focus formation when cotransfected into NIH 3T3 cells along with tumor promoters. The PP32 region is truncated in ANP32D.
APC2	similar cullins implicated ubiquitination G1 phase cyclins and cyclin-dependent kinase inhibitors. essential apc2 mutants arrest at metaphase and are defective degradation of Pds1p. ubiquitin ligase targets cell cycle regulators for degradation.
APC	encodes a large protein with multiple cellular functions and interactions, signal transduction in the wnt-signaling pathway, mediation of intercellular adhesion, stabilization of the cytoskeleton regulation of the cell cycle and apoptosis.
ARHGAP8	putative tumor suppressor gene. In a bacterial virulence study, Yersinia YopE toxicity was demonstrated to be linked to its Rho GTPase activating protein activity. YopE blocked polarization of the yeast cytoskeleton and cell cycle progress ion.
ARHI	ARHI seems to be a putative imprinted tumor suppressor gene whose function is abrogated in ovarian and breast cancers.
ATM	involved in signal transduction, cell cycle control and DNA repair. as a tumor suppressor gene. can activate abl1 and sapk, phosphorylates p53, nfkbia, brca1, ctip, nibrin (nbs1), terf1, rad9 t-cell development, gonad neurological function.
ATP8A2	tumor suppressing and negative control of cell proliferation.
AXUD1	exogenously expressed AXIN1 (the wild-type Axin gene). The Wnt signaling pathway negatively regulated by axin, axis formation in early development and impaired regulation of this signaling pathway. product has a tumor suppressor function.
BAP1	binds RING finger domain, BRCA1. ubiquitin hydrolase, deubiquitinating BAP1 and BRCA1 co-expressed murine breast. development remodeling. overlapping patterns. subnuclear distribution. enhances BRCA1-mediated inhibition of breast cancer cell growth.
BECN1	promotes autophagy in yeast with disruption apg6/vps30, MCF7 breast. carcinoma autophagy-promoting beclin-1MCF7 in cells with MCF7, inhibiting cellular proliferation, in clonigenicity tumorigenesis mice. autophagy gene inhibit tumorigenesis, decreased levels in human breast

	carcinoma.
BIN1	c-Myc-interacting adapter cell death properties. malignant progression stanch c-Myc activation malignancy cell cycle control. inhibited malignant cell transformation by MYC. reduced or undetectable in some carcinoma cell lines breast tumors.
BRCA1	associates RNA polymerase II. Mutations inhibit cancers. regulating the onset of mitosis. Activating Chk1 kinase expression, phosphorylation, cellular localization Cdc25C and Cdc2/cyclin B kinase MRE11/RAD50/NBS1. multisubunit component polII. transcriptional regulator. inhibitory transcription-coupled repair.
BRCA2	dominant oncogene. transcription factor activating transcription, promotes homologous recombination, break repair. associated activation of double-strand break repair and/or homologous recombination. hereditary breast and/or ovarian cancer.
BTG2	expressed quiescent cells overexpression decrease growth rate and clonability of NIH 3T3 cells. Btg2/Tis21 inactivation in ES cells disruption of DNA damage-induced G2/M arrest increase cell death. cell cycle control cellular response to DNA damage.
C1orf11	localization of this gene to a chromosomal region that is prone to deletions in human cancers makes it potential candidate tumor suppressor.
C5orf4	C5orf4 has a transcript of 3.1 kb, encoding a putative 144-amino-acid protein. The gene product is a putative tumor suppressor.
C5orf7	cDNA is 6.3 kb and encodes a protein of 1417 amino acids with a predicted molecular mass of approximately 155 kDa. By analyzing myeloid leukemia cells, C5ORF7 is evaluated as a candidate tumor suppressor gene.
CACNA2D2	showing very frequent allele loss and occasional homozygous deletions in lung, breast, and other cancers. The gene product is a calcium channel protein which plays an important role in excitation-contraction coupling.
CACNA2D3	the gene product is a calcium channel protein which plays an important role in excitation-contraction coupling.
CAP-1	binds specifically cytosolic domain of CD40. signal transducer associated with the cytoplasmic domain of the 75 kda tumor necrosis factor receptor (tnf-r2), and also binds to cd40 and the lymphotoxin-beta receptor.
CARS	alterations in this region have been associated with the Beckwith-Wiedemann syndrome, Wilms tumor, rhabdomyosarcoma, adrenocortical carcinoma, and lung, ovarian, and breast cancer. Alternative splicing of this gene results in two transcript variants.
CAV1	suppressor negative regulator Ras-p42/44 MAP kinase cascade. Loss of heterozygous. in breast, ovarian, prostate, colorectal carcinoma, uterine sarcomas leiomyomas. structural component of caveolar membranes signal transduction, endocytosis potocytosis.
CD14	receptor lipopolysaccharide-binding protein:lipopolysaccharide complex (LBP:LPS). Glycoprotein CD14 on the surface of human macrophages is important for the recognition and clearance of apoptotic cells triggering inflammatory responses.
CD81	integral membrane tetraspanin glycoprotein muscle cell fusion myotube maintenancee signal transduction. inhibits NK cells. antibodies blocks NK cell activation, cytokine production, cytotoxic granule release, and proliferation.
CDC14B	tyrosine phosphatase similar to Saccharomyces cerevisiae Cdc14. functionally equivalent yeast CDC14 Human CDC14p was expressed as recombinant protein, and shown to have kinetic properties characteristic of dual specific phosphatases.
CDC23	anaphase-promoting complex (APC). formation cyclin B-ubiquitin conjugate ubiquitin-mediated proteolysis of B-type cyclins. contain the TPR (tetratricopeptide repeat), a protein domain

	important for protein-protein interaction.
CDK2AP1	negatively regulate CDK2 activity interacts with DNA polymerase alpha/primase, and mediates the phosphorylation of the large p180 subunit. Therefore, the protein may play a role in DNA replication during S phase of the cell cycle.
CDKN1A	kinase inhibitor cyclin-CDK2 or -CDK4 complexes regulator of cell cycle progression at G1 mediates the p53-dependent cell cycle G1 phase arrest response stress stimuli. interacts antigen (PCNA) S phase DNA replication and DNA damage.
CDKN1C	tight-binding inhibitor of several g1 cyclin/cdk complexes (cyclin e-cdk2, cyclin d2-cdk4, and cyclin a-cdk2). It is also a negative regulator of cell proliferation. role in maintenance of the nonproliferative state throughout life.
CDKN2A	inhibitors CDK4 kinase. stabilizer p53 MDM1 CDK inhibitor isoforms and the ARF functionality in cell cycle G1 control through the regulatory roles of CDK4 and p53 in cell cycle G1 progression mutated or deleted tumors suppressor gene.
CDKN2B	adjacent CDKN2A cyclin-dependent kinase inhibitor complex with CDK4 or CDK6, and prevents the activation of the CDK kinases, cell growth regulator cycle G1 progression induced by TGF beta.
CHDH	gene product is a Member of the GMC oxidoreductase family of FAD-containing proteins.
CLCA2	outward rectifying conductance of anions calcium sensitive chloride conductance protein family expressed predominantly in trachea and lung cystic fibrosis. adhesion molecule mediate vascular arrest and colonization suppressor gene for breast cancer.
CREBL2	bZip domain of CREB. basic DNA binding leucine zipper motif dimerization. CREBL2 encodes a protein with DNA binding capabilities. The occurrence of CREBL2 deletion in malignancy suggests that CREBL2 may act as a tumor suppressor gene.
CTNNA1	cytoplasmic domain cadherins, producing a complex actin filament network, and cadherins cell-adhesion properties. The protein can associate with both e- and n-cadherins, thus may play a crucial role in cell differentiation.
CUL2	cullin homolog 2 is member of the cullin family of proteins, and it may target other proteins for ubiquitin-dependent proteolysis. the protein can forms a stable complex with the vhl tumor suppressor.
DAB2	expressed ovarian epithelial 83% identity with the mouse mitogen-responsive phosphoprotein down-regulation of DAB2 ovarian carcinogenesis. The protein is component of the csf-1 signal transduction pathway (by similarity).
DBC2	the protein is involved in axon outgrowth and myoblast fusion.
DCC	encoding similarity to cell adhesion molecules such as N-CAM. The DCC gene may play a role in the pathogenesis of human colorectal neoplasia, perhaps through alteration of the normal cell-cell interactions controlling growth.
DD5	progestin induced HECT (homology to E6-AP carboxyl terminus) E3 ubiquitin-protein ligases, targeting specific proteins for ubiquitin-mediated proteolysis. a locus disrupted in a variety of cancers. regulation of cell proliferation or differentiation.
DDX26	motif Asp-Glu-Ala-Asp/His, are putative RNA helicases. translation initiation, nuclear and mitochondrial splicing, and ribosome and spliceosome assembly. loss of heterozygosity (LOH). DEAD box conserved in evolution.
DEC1	esophageal squamous cell carcinomas. Gene expression reduced carcinomas and thus esophageal squamous cell carcinomas. It may function as a tumor suppressor associated with esophageal cancer.

DLC1	hepatocellular carcinoma. mutation analysis candidate tumor suppressor gene based on its location and homology to rhoGAP. human liver cancer, as well as for prostate, lung, colorectal, and breast cancers.
DLEC1	spliced DLEC1 transcripts contain disrupted coding regions nonfunctional proteins. Aberrant transcription of DLEC1 carcinogenesis of the lung, esophagus, and kidney. involved in carcinogenesis of the lung, esophagus, and kidney.
DLEU1	DLEU1 (leukemia associated gene 1) is a candidate tumor suppressor, which is frequently deleted in B-cell chronic lymphocytic leukemia (B-CLL).
DLEU2	presence of a putative tumor suppressor gene on chromosome 13q14, deleted in patients with B-cell chronic lymphocytic leukemia (B-CLL). This gene may function as a tumor suppressor.
DLG1	homolog of Drosophila discs large tumor suppressor 1 membrane-associated guanylate kinase (MAGUK) family structural and signaling roles. It binds to the cytoskeletal protein 4.1. it is a candidate tumor suppressor gene.
DMBT1	scavenger-receptor cysteine-rich (SRCR) homozygous deletions or lack of expression in glioblastoma multiforme, medulloblastoma, and in gastrointestinal and lung cancers. Deletions in this region were associated with endometrial cancers.
DOC-1	exhibiting loss heterozygosity reduction expression malignant hamster oral keratinocytes, associates DNA polymerase alpha/primase and mediates the phosphorylation large p180 catalytic subunit regulator of DNA replication S phase cell cycle.
DPC4	pancreatic cancers mediator of signal transduction by tgf-beta (transforming growth factor) superfamily binding of the smad2/smad4/fast-1 complex to dna, an activation function required for smad1 or smad2 to stimulate transcription.
DPH2L	down-regulated differentiation apoptosis treatment all-trans RA N-(4-hydroxyphenyl)retinamide and during cell-cycle arrest. Stable expressionDPH2L-specific anti-sense construct inhibition of cell proliferation. global protein synthesis regulation.
EGR1	transcriptional regulator recognizing and binding DNA sequence 5'-cgcccccgc-3'(egr-site). mitogenesis and differentiation. controls transforming growth factor-beta-1 (TGFB1) gene expression, expression of TGFB1 inhibited human cancer cell growth.
FABP3	responsible in the modulation of cell growth and proliferation. Gene FABP3 encodes fatty acid-binding protein 3 that functions to arrest growth of mammary epithelial cells. candidate tumor suppressor gene for human breast cancer.
FAT	ortholog Drosophila fat controlling cell proliferation cadherin superfamily adhesion molecule signaling receptor, large transmembrane protein of nearly 4600 residues with 34 tandem cadherin repeats, five EGF-like repeats laminin A-G domain.
FGL1	fibrinogen homologous to carboxy terminus of the fibrinogen beta- and gamma- subunits cysteines of fibrinogens and fibrinogen related proteins. hepatocellular carcinomas. down-regulated hepatocellular carcinoma.
FHIT	diadenosine 5',5'"-P1,P3-triphosphate hydrolase purine metabolism. esophageal, stomach, and colon carcinomas. cleaves a-5'-ppp-5'a to yield amp and adp, and may function as a tumor suppressor for specific tissues.
FLJ10506	WD repeat heterotrimeric or multiprotein complexes. cell cycle progression, signal transduction, apoptosis, and gene regulation glioblastoma tumorigenesis of glial and other tumors.
FOXD1	forkhead domain, the intronless gene belongs to the forkhead family of transcription factors. The gene binds freac-3 and freac-4 to their cognate sites, resulting in bending of the DNA at an angle of 80-90 degrees.
FOXP1	transcription factors, winged helix DNA-binding motif, C(2)H(2) zinc finger, nuclear localization

	signals, coiled-coil regions, PEST transactivation domains. transcriptional repressor. role in the specification and differentiation of lung epithelium.
FUS1	The FUS1 may function as a tumor suppressor, inhibiting colony formation, causing g1 arrest and ultimately inducing apoptosis in homozygous 3p21.3 120-kb region-deficient cells.
FUS2	NAT activity formation of a covalent NAT/acetyl-coA intermediate acetyl-coA nucleophile acetyl group is transferred to the substrate by nucleophilic attack. cytoplasmic expression consistent action of characterized NATs NATs to susceptibility.
GAK	transcriptional target p53 tumor suppressor. GAK partner cyclin G and CDK5. auxilin homolog uncoating of clathrin- coated vesicles by hsc70 in non-neuronal cells. Its expression oscillates slightly during the cell cycle, peaking at G1.
GAS1	growth arrest p53-dependent mechanism. suppresses growth tumorigenicity of human tumor overexpression MDM2 site of deletion in myeloid malignancies suppresses DNA synthesis. Overexpression block cell proliferation in lung and bladder carcinoma.
GAS11	loss of heterozygosity involving chromosome 16q24.3 is common in breast and prostate cancer and suggests the presence of a tumor suppressor gene. The gene encodes a protein belonging to the family of cytoskeleton-associated proteins.
GLTSCR1	expression GLTSCR1 transcript heart, brain, placenta, skeletal muscle, and pancreas, lung, liver, and kidney. glioma tumor suppressor.
GLTSCR2	the product of this gene is a putative tumor suppressor.
GRLF1	GTPase and middle domains (residues 1-1228), tumors including pancreatic carcinomas and gliomas. Glioblastoma/astrocytoma loss chromosomal region encompassing p190-A development of these tumors.
HDAC3	histone deacetylase/acuc/apha represses transcription tethered to a promoter. participate regulation of transcription zinc-finger transcription factor YY1. down-regulate p53 function and thus modulate cell growth and apoptosis.
HEMK	suppressor mutations abrogated growth defects of the hemK knockout strain threonine alanine change codon 246 of polypeptide chain release factor (RF) 2, translational termination. HemK methylates RF1 and RF2 tryptic fragment GGQ motif methylation.
HIC1	zinc-finger transcription expressed underexpressed tumor cells p53 binding 5' flanking region, activation wild-type suppression of G418 selectability of cultured brain, breast and colon cancer cells following insertion of the gene.
HRG22	homology N-terminal BTB/POZ and C-terminal C(2)H(2) zinc finger domains. highlighted a conserved GLDLSKK/R peptide. conserved GLDLSKK/R motif related to interaction motif (PXDLSXK/R), except replacement of proline by a glycine. transcriptional repressor.
HSAL2	developmental transcription factors. homology to a region-specific homeotic gene (SAL) in Drosophila. expression P2 (the distal promoter, upstream of exon 1A) undetectable in some malignant populations as opposed to their normal human counterparts.
HTS1	suppress tumorigenicity Hela nude mice, association "HeLa tumor suppression" C-terminal region similarity Rab 3 family of small GTP binding proteins. binds SH3 domain of c-Abl kinase, regulator of MAPK1/ERK2 kinase.
HYAL1	lysosomal hyaluronidase degrades hyaluronan, glycosaminoglycans of the extracellular matrix. Hyaluronan proliferation, migration and differentiation. associated with tumor suppression.
HYAL2	similar to hyaluronidases degrade hyaluronan, proliferation, migration and differentiation. lysosomal hyaluronidase hydrolyzes high molecular weight hyaluronan, GPI-anchored receptor oncogenic virus Jaagsiekte sheep retrovirus.

IGFBP7	insulin-like growth factor modulating signaling of the TGF-beta family, as does alpha-inhibin. Bind IGFs. Expression abrogated breast cancer progression concomitant with loss of heterozygosity growth-suppressing factor IGF-binding protein.
IGSF4	deletion associated lung and breast neuroblastoma. expression reduced or absentin A549 NSCLC, hepatocellular carcinoma (HCC) and pancreatic cancer (PaC) cell lines promoter methylation suppresses tumor formation by A549 cells in nude mice.
ING1	growth inhibitory regulate apoptosis linkage between DNA repair, apoptosis, and chromatin remodeling multiple HAT.ING1.PCNA protein complexes. Regulation p53-dependent transcriptional activation histone acetyltransferase cell-cycle arrest apoptosis.
ING1L	similar ING1 interaction with p53. ING1L contains PHD-type zinc-fingers. Its functions are shown to be similar to ING1, i.e. it may interfere with signals transmitted through p53 and p33 (ING1).
ING4	this gene is a homolog to the tumor suppressor p33 ING1, and also contains a PHD-finger. It may act in chromatin-mediated transcriptional regulation.
LAPSER1	resembling transcription factors (P-Box, Q-rich and multiple leucine zippers). expressed highest levels prostate and testis Over-expression inhibited cell growth and colony-forming efficiencies of most cancer.
LATS1	lats tumor suppressor suppresses tumor growth rescues all developmental defects embryonic lethality phosphorylated complexes CDC2 inhibits proliferation. down-regulates Cyclin A and Cyclin B CDC2 kinase MCF-7 tumor formation athymic nude mice.
LATS2	PAPA repeat dipeptide proline-alanine protein-protein interactions serine/threonine kinase domain. The Drosophila 'large tumor suppressor' closely related to the LATS1 proteins, followed by Drosophila Lats.
LDOC1	leucine zipper-like motif proline-rich SH3-binding down-regulated in some cancer cell lines. transcriptional regulation. development and/or progression of some cancers.
LOH11CR2A	chromosomal deletions lung and breast carcinomas, heterozygosity tumor susceptibility genes Amino acid substitutions in the coding region of LOH11CR2A indicates that this gene is unlikely to be involved in the tumorigenic process.
LRP1B	lipoprotein receptor similar to LRP Hemi- and homozygous deletion LRP1B urothelial cancers. LRP1B tumor suppressor gene in urothelial cancers.
LUCA3	similar in structure to hyaluronidases degrade hyaluronan glycosaminoglycans extracellular matrix. Hyaluronan cell proliferation, migration and differentiation.
MAD	encodes MAX dimerization protein, competing with MYC for binding to MAX to form a sequence-specific DNA-binding complex. transcriptional repressor.
MAFB	Maf transcription factors cellular differentiation hematopoietic up-regulated MafB myeloid and monocytic differentiation. transcriptional regulator lineage-specific hematopoiesis by repressing ETS1-mediated transcription of erythroid-specific myeloid.
MAP2K4	kinase kinases mapk8 (jnk1) and mapk9 (jnk2) mapk14 (p38) c-Jun NH2-terminal kinase activation, signaling pathway distinct from DPC4, p16, p53, and BRCA2. MKK4/SEK1 transfected metastasis suppressor gene.
MAPKAPK3	Ser/Thr protein kinase mitogen-activated extracellular signal-regulated kinases (ERKs) integration point signals activated by ERK, p38 MAP kinase and Jun N-terminal kinase E47, a basic helix-loop-helix transcription factor.
MAPRE3	Gene MAPRE3 encodes a protein which is a member of the RP/EB family. This protein localizes to the cytoplasmic microtubule network and binds APCL, a homolog of the adenomatous polyposis coli tumor suppressor gene.

MCC	colorectal neoplasia sporadic and familial tumors. colorectal cancers related somatically acquired point mutations in MCC amino acid substitutions. candidate for the putative colorectal tumor suppressor gene.
MDC	disintegrin and metalloprotease (ADAM) domain 11, membrane-anchored proteins, fertilization, muscle development, and neurogenesis. tumor suppressor gene for human breast cancer based on its location chromosome 17q21 deletion mapping.
MEN1	menin neoplasia type 1. inhibits JunD familial endocrine (fmen1) werner syndrome parathyroid glands gastro-intestinal anterior pituitary cutaneous lesions hypergastrinemia peptic ulcer zollinger-ellison syndrome zes hyperparathyroidism hyperinsulinemia.
ML-1	associated with tumorigenesis, antisense cDNA construct, transfected into nontumorigenic, anchorage-independent growth (AIG) cells, was sufficient to convert these cells into a tumorigenic phenotype.
MRVI1	disrupted by mouse AIDS-related virus (MRV). endoplasmic reticulum Jaw1, lymphoid-restricted expression down-regulated during myeloid differentiation. MRV integration at Mrvi1 induces myeloid leukemia by altering expression.
MTAP	methylthioadenosine phosphorylase polyamine metabolism salvage of both adenine and methionine. Methylthioadenosine phosphorylase is deficient in many cancers due to codeletion of MTAP.
MXI1	MYC family of transcription factors negatively regulate MYC function oncogenic transcription factor, inhibits the transcriptional activity of MYC by competing for MAX, another basic helix-loop-helix protein that binds to MYC required for its function.
NAP1L4	nucleosome assembly protein (NAP) interact core and linker histones. shuttle between the cytoplasm and nucleus histone chaperone. Beckwith-Wiedemann syndrome, Wilms tumor, rhabdomyosarcoma, adrenocortical carcinoma, and lung, ovarian breast.
NBR2	close proximity on chromosome 17 to tumor suppressor gene BRCA1. The head to head coordinated through a bi-directional promoter. balance disrupted, abnormal cell growth and differentiation.
NF2	cytoskeletal components moesin, ezrin, and radixin, merlin bridging membrane intracellular cytoskeletion. mutated majority of schwannomas and meningiomas.
NORE1B	RASSF1, located at 3p21.3 inactivated in cancers in lung, breast, bladder and renal cell carcinomas. down-regulated in several cancer cell lines. gene silencing methylation of the CpG islands. Ras association suggests Ras-like signaling pathways.
NPR2L	NPR2L gene resides in a 120-kb critical region for a lung cancer tumor suppressor gene on chromosome 3p21.3. NPR2L is proposed as a candidate for functional tumor suppressor gene studies.
OVCA2	mRNA expressed epithelial cells ovary decreased expression of these two genes contributes to ovarian tumorigenesis and should be considered candidate tumor suppressor genes.
PDGFRL	similarity ligand binding domain of platelet-derived growth factor receptor beta. Mutations deletion associated sporadic hepatocellular carcinomas, colorectal cancers, and non-small cell lung cancers. tumor suppressor.
PHEMX	subtransferable fragments located imprinted gene domain of 11p15.5, Alterations associated Beckwith-Wiedemann syndrome, Wilms tumor, rhabdomyosarcoma, adrenocortical carcinoma, and lung, ovarian, and breast cancer. hematopoietic tetraspanin superfamily.
pHyde	AdRSVpHyde inhibited prostate cancer DU145 and LNCaP reduced DU145 tumors in nude mice AdRSVpHyde induced apoptosis and stimulated p53 expression. tumor suppressor gene induction of apoptosis.

PIK3CG	subunit p110 gamma of phosphoinositide 3-OH-kinase-gamma (PI3K gamma) pi3/pi4-kinase phosphorylates phosphoinositides on the 3-hydroxyl group of the inositol ring. extracellular signals, E-cadherin-mediated adherens junctions cytotoxicity in NK cells.
PINX1	TID (telomerase inhibitory domain) bind telomerase catalytic subunit TERT inhibit its activity. Overexpression inhibited telomerase activity, shortened telomeres, and induced crisis Depletion increased tumorigenicity in nude mice.
PLAGL1	Pleomorphic adenoma gene-like 1 C2H2 zinc finger protein with transactivation and DNA-binding activity. PLAGL1 has been shown to exhibit antiproliferative activities and regulate apoptosis and cell cycle arrest. candidate tumor suppressor gene.
PRDM2	PRDI-BF1-RIZ homology zinc finger histone/protein methyltransferase bind retinoblastoma estrogen receptor TPA-responsive element (MTE) of the heme-oxygenase-1 transcriptional regulation neuronal differentiation and pathogenesis of retinoblastoma.
PTCH	ransmembrane represses transcription TGF-beta Wnt signaling proteins. receptor for sonic hedgehog (shh), indian hedgehog (ihh) and desert hedgehog (dhh). associate smoothened protein (smo) to transduce the hedgehog's proteins signal.
PTEN	phosphatase on tyrosine, serine and threonine residues. mutations glioma, prostate, kidney and breast carcinoma cell lines or tumor specimens. oncogenesis of multiple human cancers. embryonic development and tumor suppression.
PTPN13	tyrosine phosphatase (PTP) signaling molecules cell growth, differentiation, mitotic cycle, and oncogenic transformation. five PDZ domains, leucine zipper motif. interact dephosphorylates Fas receptor IkappaBalpha.
PTPRG	tyrosine phosphatase (PTP) family extracellular region, a single transmembrane region, and two tandem intracytoplasmic catalytic domains, receptor-type PTP. carbonic anhydrase-like (CAH) carcinoma and lung carcinoma.
RASSF1	Ras association domain family. Loss of expression methylation of the CpG islandRASSF1A promoter Reexpression reduced colony formation, suppressed anchorage-independent growth, and inhibited tumor formation in nude mice. silencing hypermethylation.
RBBP7	retinoblastoma binding protein 7 WD-repeat regulates cell proliferation histonedeacetylase complexes, mSin3 co-repressor chromatin assembly acetyltransferase nucleosomal DNA interact BRCA1 tumor-suppressor.
RBM6	RNA-binding motifs zinc-finger poly(g) homopolymers lung cancer cell lines. Exclusion of exon 5 frameshift truncated protein of 520 amino acids instead of 1123 amino acids normal lung tissue, relative amount shorter transcript much greater.
RBX1	interacts with cullins. ubiquitination reaction by heterodimerizing with cullin-1 to catalyze ubiquitin polymerization. regulation of protein turn-over. The gene product is a VHL tumor suppressor complex subunit and SCF ubiquitin ligase subunit.
RECK	transformation suppressor glycoprotein invasion and metastasis metalloproteinase-9 negatively regulates MMP-9 by suppressing secretion and by direct inhibition of its enzymatic activity. regulate MMP-2 and MT1-MMP. cysteine-rich.
RFP2	member of the tripartite motif (TRIM) family. The TRIM motif includes three zinc-binding domains, a RING, a B-box type 1 and a B-box type 2, and a coiled-coil region. The gene may act as a tumor suppressor.
RIS1	upregulated Ras-senescence. transcriptional factor Ets2 RIS1 functions as a highly specific marker of Ras-induced senescence and is a candidate tumor-suppressor gene.
RPL10	ribosomal protein 60S subunit L10E family nontumorigenic derivative QM mRNA was modulated between tumorigenic and nontumorigenic cell lines. Thus the QM gene was proposed as an

	attractive tumor-suppressor candidate gene.
RPS29	ribosomal protein small 40S ribosomal subunit S14P family of ribosomal proteins. The protein C2-C2 zinc finger-like domain that can bind to zinc, can enhance the tumor suppressor activity of Ras-related protein 1A (KREV1).
RRM1	large subunit (M1) of ribonucleotide-diphosphate reductase, the heterodimeric enzyme rate-limiting deoxyribonucleotide Alterations Beckwith-Wiedemann syndrome, Wilms tumor, rhabdomyosarcoma, adrenocrotical carcinoma lung, ovarian breast.
S100A2	S100 calcium binding protein A2 EF-hand calcium-binding motifs. regulation cell cycle progression and differentiation. Chromosomal rearrangements and altered expression of breast cancer. This protein may have a tumor suppressor function.
SEMA3B	semaphorin/collapsin guidance of growth cones during neuronal development. High loss of heterozygosity (LOH) short arm of chromosome 3 diminished tumorigenicity in nude mice. SEMA3B anchorage independence of HEY cells.
SF1	transcriptional repressor and splicing factor. polymerase II holoenzyme factors EWS Ewing's sarcoma tumors. atherosclerosis proliferating arterial SMC (smooth muscle cells). role for zfm1 in controlling proliferation expression of pro-inflammatory.
SFRP1	cysteine-rich domain ligand-binding Frizzled proteins. silencing of hypermethylated genes methylation dense CpG islands and histone hypermethylated in colorectal cancer and gastric cancer. loss of tumor suppressor construction molecular marker panel.
SLC22A1L	subtransferable candidate metal-tetracycline/h+ antiporter. imprinted Alterations Beckwith-Wiedemann syndrome, Wilms tumor, rhabdomyosarcoma, adrenocortical carcinoma, and lung, ovarian, and breast cancer. fetal kidney.
SLC26A3	solute carrier downregulated in adenoma glycoprotein mucosa gastrointestinal tract. down-regulated in colon tumors. DRA down-regulation positively colonic tumor progression polyp to adenocarcinoma. Mutations congenital chloride diarrhea.
SMARCA4	SWI-SNF chromatin remodeling bind BRCA1 regulate the expression tumorigenic protein CD44. implicated in growth control interaction with the tumor suppressor pRb and a negative regulator of proliferation. function as a tumor suppressor.
ST7	ST7 prostate-cancer-derived PC3 proliferation abrogated tumorigenicity. differential expression in cancer cells. transmembrane protein. The level of this protein lower in tumor derived cell lines compared to normal cells. tumor suppressor.
ST7L	WNT2B allelic loss rearrangements breast cancer germ cell tumors squamous cell carcinoma of head neck, non-small cell lung cancer gastrointestinal stromal/smooth muscle tumors meningioma, melanoma, acute megakaryoblastic leukemia Kaposi's sarcoma.
ST13	heat shock 70kD mediates HSP70 and HSP90. assembly process of glucocorticoid receptor, assistance of multiple molecular chaperones. The expression downregulated in colorectal carcinoma tissue suggesting that is a candidate tumor suppressor gene.
ST14	matriptase epithelial-derived, integral membrane serine protease Kunitz-type HAI-1, activated sphingosine 1-phosphate cleaves activates hepatocyte scattering factor urokinase plasminogen activator expression breast, colon, prostate, and ovarian tumors.
STIM1	transmembrane Alterations associated Beckwith-Wiedemann syndrome, Wilms tumor, rhabdomyosarcoma, adrenocrotical carcinoma, and lung, ovarian, and breast cancer. induces growth arrest and degeneration of the human tumor cell lines G401.
TCEB2	elongin B. heterotrimer transcriptionally active regulatory subunits. von Hippel-Lindau (VHL) tumor elongin-binding domain of VHL is frequently mutated in cancers, loss of elongin binding causes tumorigenesis VHL ubiquitination functions.

THW	down-regulated in mammary carcinoma pancreas metastases location of the THW gene on chromosome 6q. LOH for this region has been reported in malignant melanoma and prostate, pancreas, uterine and mammary carcinomas.
TP53	regulating cell cycle. transition G0 to G1 transformation and malignancy DNA-binding oligomerization transcription tetramer inhibit growth invasion. Mutants loss of tumor suppressor activity. Alterations somatic mutations germline mutations Li-Fraumeni syndrome. apoptosis.
TP63	homology to p53 multiple isotypes. transactivate p53 reporter genes and induce apoptosis. dominant-negative agents toward transactivation by p53 and p63, growth-suppression and apoptosis, upregulated p21waf-1. gene mutations in epidermal tumors.
TRIM8	tripartite motif (TRIM) zinc-binding RING B-box coiled-coil nuclear bodies expression induced IFN-g in epithelial and lymphoid cells. Co-expression SOCS-1 decreases stability and levels expression.
TSG101	deletion in NIH3T3 lung metastases in nude mice ubiquitin-conjugating coiled-coil stathmin cytosolic phosphoprotein cell growth and differentiation negative growth regulator Mutations and alternative splicing breast cancer.
TSSC1	PMID 9403053 tumor-suppressing subtransferable imprinted gene domain of 11p15.5. Alterations associated with the Beckwith-Wiedemann syndrome, Wilms tumor, rhabdomyosarcoma, adrenocortical carcinoma, and lung, ovarian, and breast cancer.
TSSC3	subtransferable imprinted Alterations Beckwith-Wiedemann syndrome, Wilms tumor, rhabdomyosarcoma, adrenocortical carcinoma, and lung, ovarian, and breast cancer. Fas-mediated apoptosis. imprinted in placenta, liver and fetal tissues maternal allele.
TSSC4	subtransferable imprinted Alterations Beckwith-Wiedemann syndrome, Wilms tumor, rhabdomyosarcoma, adrenocortical carcinoma, and lung, ovarian, and breast cancer. malignancies and disease that involve this region.
VHL	autosomal multicentric or bilateral, and manifest benign neoplasms retinal, cerebellar, and spinal hemangioblastoma, renal cell carcinoma, pheochromocytoma, and pancreatic VEGF gatekeeper kidney cells hypoxia-inducible ubiquitin-dependent proteasome.
WFDC1	WAP-type four disulfide core domain loss of heterozygosity (LOH) in prostate, breast and hepatocellular cancers and Wilms' tumor, in particular in hepatocellular carcinoma (HCC). growth inhibitory properties ps20 tumor suppressor gene.
WIT-1	expressed coordinately with WT1. tumor suppressor gene that has been implicated in Wilms tumor, mesodermal origin. same temporal and cell-restricted expression pattern. Methylation of this gene is implicated in chemoresistant acute myeloid leukemia.
WNT4	secreted signaling proteins oncogenesis cell fate and patterning embryogenesis. influence the sex-determination cascade. identity Wnt4 protein of mouse and rat. DAX1 antagonize the testis-determining factor, female development breast tissue.
WT1	Wilms' zinc-finger transcription proto-oncogenes. binds consensus sequence functions in kidney gonad proliferation differentiation. regulate cell proliferation mesenchyme-to-epithelium metanephric crescentic glomerulonephritis or mesangialsclerosis.
WWOX	protein degradation, transcription, and RNA splicing. oxidoreductase. ovarian cancer chromosomal translocations and homozygous deletions Alternative WWOX transcripts were expressed at high levels involvement WWOX gene in breast cancer progression.

LIST OF REFERENCES

[1] ComputerScope Ltd., 2001. "NUA Internet how many online?" URL: http://www.nua.ie/surveys/how_many_online/index.html. Accessed: 12/8/02

[2] Baylor College of Medicine, 1999. "Tumor gene database" URL: http://condor.bcm.tmc.edu/ermb/tgdb/tgdb.html. Accessed: 12/8/02

[3] Wjst M, Immervoll T, 1998. "An internet linkage and mutation database for the complex phenotype asthma" *Bioinformatics* 14:827-828. URL: http://cooke.gsf.de/asthmagen. Accessed: 12/8/02

[4] Shafer R, 2002. "Stanford HIV RT and protease sequence database" URL: http://hivdb.stanford.edu/. Accessed: 12/8/02

[5] The Institute of Medical Genetics, University of Wales, 2002. "The human gene mutation database" URL: http://www.uwcm.ac.uk/uwcm/mg/hgmd0.html. Accessed: 11/18/02

[6] Reeves P, 1998. "Bacterial polysaccharide gene database" URL: http://www.microbio.usyd.edu.au/BPGD/default.htm. Accessed: 11/18/02

[7] Baylor College of Medicine, 1999. "Breast cancer gene database" URL: http://condor.bcm.tmc.edu/ermb/bcgd/bcgd.html. Accessed: 11/18/02

[8] Crasto C, Marenco L, Miller P, Shepherd G, 2002. "Olfactory receptor database: a metadata-driven automated population from sources of gene and protein sequences" *Nucleic Acids Research* 1**:**354-360

[9] Bioinformatics.org, 2002. "Cardiovascular gene database" URL: http://bioinformatics.org/project/?group_id=137. Accessed: 11/18/02

[10] Bouchard C, 2002. "Obesity gene map database" URL: http://obesitygene.pbrc.edu/. Accessed: 11/18/02

[11] Mégy K, 2002. "Cardiac gene database" URL: http://polyc.cnrs-mrs.fr/Card_Gene/index.html. Accessed: 11/18/02

[12] Pergament E, Fiddler M, 2001. "Tumor suppressor genes" URL: http://www.intouchlive.com/home/frames.htm?http://www.intouchlive.com/cancergenetics/tsg.htm&3. Accessed 12/17/02

[13] American for Medical Progress, 2001. “Reflections on the past, challenges for the future” URL: www.amprogress.org/Cancerresearch/cancerresearchmain.cfm. Accessed: 11/18/02

[14] Knudsen A, Jr., 1971. “Mutation and cancer: statistical study of retinoblastoma” Proc *Natl Acad Sci USA* 68:820-823

[15] Fung Y, Murphree A, T’Ang A, Qian J, Hinrichs S, Benedict W, 1987. “Structural evidence for the authenticity of the human retinoblastoma gene” *Science* 236:1657-1661

[16] Lee W, Bookstein R, Hong F, Young L, Shew J, Lee E, 1987. “Human susceptibility gene: cloning, identification, and sequence” *Science* 235:1394-1399

[17] Fong E, Sheppard C, Harvill K, 2001. “Design, implementation and management of distributed databases--an overview” In *High Performance Web Databases* edited by Purba S, Auerbach Publications

[18] Brueggen D, Lee S, 2001. “Data communications requirements of distributed database systems” In *High Performance Web Databases* edited by Purba S, Auerbach Publications

[19] Kara-Zaitri C, 2000, “Database models” URL: http://www.staff.brad.ac.uk/ckarazai/Lecture3.pdf. Accessed: 11/17/02

[20] IBM, 2002. “IMS family” URL: http://www.ibm.com/software/success/cssdb.nsf/topstoriesFM?OpenForm&Site=dmims. Accessed: 11/17/02

[21] Codd, E. 1985. "Does your DBMS run by the rules?" *ComputerWorld,* October 21

[22] Rao B, 1994. *Object-Oriented Databases: Technology, Applications, and Products* McGraw-Hill, New York.

[23] Quass D, Rajaraman A, Sagiv Y, Ullman J, Widom J, 1995. “Querying semistructured heterogeneous information” In *Proceedings of the Fouth International Conference on Deductive and Object-Oriented Database (DOOD),* edited by Ling T, Mendelzon A, and Vieille L, Singapore, Springer-Verlag

[24] Lazy Software Ltd, 2002. “The associated model of data” URL: http://www.lazysoft.com/associativemodel/default.htm. Accessed: 11/17/02

[25] Son S, Yoon I, Kim C, 1998. **“**A component-based client/server application development environment using Java Technology of Object-Oriented Languages” *TOOLS Proceedings* 28:168 -179

[26] Schussel G, 1995. "Client/server past, present and future" URL: http://news.dci.com/geos/dbsejava.htm. Accessed: 11/17/02

[27] Edelstein H, 1994. "Unraveling client/server architecture." *DBMS 7,* 5: 34(7).

[28] Hudson P, 2000. http://www.herts.ac.uk/ltdu/technology/history_of_java.html. Accessed: 11/15/02

[29] Hall M, 2001. *Core Servlets and JavaServer Pages* Sun Microsystems Press/Prentice Hall PTR, Upper Saddle River

[30] McKusick V, Brylawski B, 2002. "Online mendelian inheritance in man" URL: http://www.ncbi.nlm.nih.gov/Omim/. Accessed: 12/10/02

[31] Weizmann Institute of Science, 2001. "GeneCards" URL: http://genome-www.stanford.edu/genecards/index.html. Accessed: 11/18/02

[32] ExPasy Molecular Biology Server, 2002. "SWISS-PROT and TrEMBL" URL: http://us.expasy.org/sprot/. Accessed: 12/10/02

[33] Genetics Department of Stanford University, 2001. "SOURCE" URL: http://genome-www5.stanford.edu/cgi-bin/SMD/source/sourceSearch. Accessed: 12/10/02

[34] Holzschlag ME, 2000. *Special Edition Using HTML 4,* QUE Publishing, Indianapolis

[35] Codd E, 1970. "A relational model of data for large shared data banks" *Communications of the ACM* 13:377-387. URL: http://www.acm.org/classics/nov95. Accessed: 11/17/02

[36] Cunningham & Cunningham , Inc., 2002. "Relational model" URL: http://c2.com/cgi/wiki?RelationalModel. Accessed: 11/17/02

[37] Bourret R, 2002. "XML and databases" URL: http://www.rpbourret.com/xml/XMLAndDatabases.htm. Accessed: 11/17/02

BIOGRAPHICAL SKETCH

Yanming Yang was born in Changyi of Shandong Province, China. He received a B.S. in forestry from Shandong Agricultural University in 1982, an M.S. in plant physiology from the Chinese Academy of Science in 1985, and a Ph.D. in molecular biology from the University of Arkansas in 1996. Yanming and his family moved to Gainesville, Florida, in the spring of 1998 to work as a post-doctorate research associate at the Department of Plant Pathology, University of Florida. Yanming began his study in computer science in the post-baccalaureate program at the Department of Computer and Information Science and Engineering, and was enrolled in the MS program in May of 2001.

Currently, Yanming still works as a post-doctorate scientist at the Department of Plant Pathology, and his research is involved in gene manipulation of plant viruses, incorporation of foreign genes into plant genome to generate disease resistance, and informatics in plant genomes. His research interests include bioinformatics, molecular biology, computational biology, and database systems management.