Published in final edited form as:

Proc SPIE. ; 8319: .

Managing and Querying Whole Slide Images

Fusheng Wang^{a,b}, Tae W. Oh^a, Cristobal Vergara-Niedermayr^{c,*}, Tahsin Kurc^{a,b}, and Joel Saltz^{a,b}

^aCenter for Comprehensive Informatics, Emory University, Atlanta, Georgia, USA

^bDepartment of Biomedical Informatics, Emory University, Atlanta, USA

^cThe Cancer Institute of New Jersey, Piscataway, New Jersey, USA

Abstract

High-resolution pathology images provide rich information about the morphological and functional characteristics of biological systems, and are transforming the field of pathology into a new era. To facilitate the use of digital pathology imaging for biomedical research and clinical diagnosis, it is essential to manage and query both whole slide images (WSI) and analytical results generated from images, such as annotations made by humans and computed features and classifications made by computer algorithms. There are unique requirements on modeling, managing and querying whole slide images, including compatibility with standards, scalability, support of image queries at multiple granularities, and support of integrated queries between images and derived results from the images. In this paper, we present our work on developing the Pathology Image Database System (PIDB), which is a standard oriented image database to support retrieval of images, tiles, regions and analytical results, image visualization and experiment management through a unified interface and architecture. The system is deployed for managing and querying whole slide images for *In Silico* brain tumor studies at Emory University. PIDB is generic and open source, and can be easily used to support other biomedical research projects. It has the potential to be integrated into a Picture Archiving and Communications System (PACS) with powerful query capabilities to support pathology imaging.

Keywords

Whole slide image; image data management; Pathology Analytical Imaging Standards; PACS

1. INTRODUCTION

Digital pathology images, such as whole-slide images (WSI) generated by scanning microscope slides, at diagnostic resolution enable the microscopic examination of tissue specimens taken from patients to support clinical diagnosis and biomedical research. Meanwhile, computerized pathology image analysis offers a means of rapidly carrying out quantitative, reproducible measurements of micro-anatomical features in WSI. Despite the successful and widespread use of imaging data and computerized processes in other areas of healthcare such as Radiology, the transition of digital pathology in clinical diagnosis has only begun recently. There are several unique challenges to the wide adoption of digital pathology, including storage and management of large image datasets, lack of standardized data exchange mechanisms and interfaces, and complex queries on image datasets and image analysis results. These challenges demand an effective infrastructure for modeling,

Further author information: (Send correspondence to Fusheng Wang, fusheng.wang@emory.edu, Telephone: +1 404-712-0057).

*Work done while employed at Center for Comprehensive Informatics, Emory University

managing, querying and sharing of whole slide images and analytical results in a standard oriented approach.

As a first step towards building such an infrastructure, DICOM Working Group 26 has developed supplements 122 and 145 for formal representations of specimens and whole slide images.¹ In supplement 145, images at multiple resolutions can be represented as a hierarchy of split tiles at different resolution levels. This approach provides two major benefits: image regions at different resolutions could be rapidly retrieved for viewing, and DICOM-based standardized representation of data for archiving and exchange could be performed through an institution's existing Picture Archiving and Communication System (PACS), thereby obviating the difficulties in reading whole slide images generated by different scanners in proprietary formats. However, there are no current implementations of pathology extended PACS. In addition, such systems will not handle various whole slide image formats in reality. Traditional PACS systems also lack essential capabilities required by whole slide imaging applications, and the flexibility to support biomedical research.

It is necessary to support retrieval of images, tiles, regions, analytical results, image visualization and study management for effective management and use of digitized high resolution pathology images. Figure 1 illustrates different granularities for digital slides. Whole slide images are often represented in a pyramid structure consisting of multiple images at different resolutions. The baseline image has the highest resolution. Whole-slide images generated at diagnostic resolution are very large: a typical whole slide image may contain 100,000x100,000 pixels. Analyzing and viewing whole slide images are often constrained by computer memory or screen size. Thus, a whole slide image is usually partitioned into much smaller sized *tiled images* (or *tiles*) such that a tile can fit into computer memory for image analysis or visualization. A *region* in an image is an area of interest to be extracted, displayed or processed. For example, a region can be represented by a 2-dimensional window containing objects of interest, such as a tumor, a pseudopalisade, a nucleus of certain characteristics, etc. The retrieval of images or regions at multiple levels and resolutions is required in applications that involve whole slide images. Such requirements are not considered in traditional radiology based image management systems. Common types of queries in whole slide imaging applications are summarized below.

- Study level retrieval: retrieval of a set of images based on study metadata. For example, retrieve a list of whole slide images for a given set of study IDs;
- Image level retrieval: retrieval of whole slide images based on metadata. For example, retrieve the whole slide image based on a patient ID;
- Tile level retrieval: retrieval of tile images based on metadata. For example, retrieve a tile image based on a tile location (e.g., x, y coordinates) at a specific resolution;
- Region level retrieval: retrieval of regions in a whole slide image. For example, given the coordinates of a rectangle, return the region contained in the rectangle from image I;
- Integrated queries of images and derived results. For example, retrieve all tiles containing nuclei such that the mean area of nuclei is larger than 400 square pixels.

The requirement of support for integrated queries of images and results stems from the increasing power of computed aided pathology image based diagnosis. For example, the morphology of brain tumor nuclei from pathology whole slide images is a pivotal attribute used to classify brain tumors (e.g., astrocytoma, oligodendroglioma). Nuclei appear to be round shaped with smooth regular texture in oligodendrogliomas, whereas they are generally more elongated with rough and irregular texture in astrocytomas. The classifications of brain

tumor nuclei based on morphology are also linked to genetic and gene expression classifications.^{2,3,4} Modeling and querying analytical results has been extensively studied in the PAIS project.^{5,6,7,8} With integrative queries, users can retrieve and examine regions of an image closely with specific image morphology characteristics.

Our Contribution

In our work, we design and implement a data management architecture for managing whole slide images, with the following capabilities:

1. Standard oriented data model compatible with related standard models or open models;
2. Management of whole slide images for biomedical studies, by tracking metadata of images and studies;
3. Management of image tiles at multiple resolutions;
4. Support of comprehensive queries on studies, images, tiles, and regions, and advanced integrated queries of images and analytical results;
5. A convenient data loading tool for data loading and a Web querying tool for querying the database; and
6. A unified interface and architecture that bring all the capabilities together.

2. DATA MODEL

2.1 Background

One of the goals of the Pathology Image Database System (PIDB) is data model compatibility with existing models. The Open Microscopy Environment (OME) project^{9,10} has developed a data model and a database system that can be used to represent, exchange, and manage microscopy image data and metadata. OME provides a data model of common specification for storing details of microscope setup and image acquisition. We build the model by extending OME model to support whole slide images.

Pathology Analytical Imaging Standards (PAIS)^{5,6,7,8} is a project on developing open data model and database to support the modeling, management and queries of large scale analytical pathology image results, such as markups, features and annotations generated by humans or computer algorithms. Managing and querying images is beyond the scope of PAIS. PIDB model complements PAIS and is compatible with PAIS, which enables support of powerful integrative queries of images and results.

In DICOM extension for pathology,¹ a pyramid based model is used to represent tiles of images at different resolutions as image series. PIDB model is compatible with DICOM with similar representations of tiles, thus PIDB has the potential of being used as the backend of a pathology PACS system.

The caBIG Life Sciences Domain Analysis Model (LS DAM) provides foundational component for achieving semantic interoperability among the various applications across caBIG¹¹ to support basic and pre-clinical research. The PIDB model borrows from components already defined in LS DAM.

2.2 PIDB Model

The model of PIDB is designed to represent image information (whole slide images and tiled images), image acquisition information, project and study information, and is compatible with PAIS, OME and LS DAM. The logical model contains 15 entities and is

illustrated in Figure 2 (attributes ignored). Entities on image acquisition information include INSTRUMENT – slide scanning instrument, PATIENT – subject of the specimen, and SPECIMEN – specimen and its characteristics. Entities on images include IMAGE – original whole slide image, TILEDIMAGE – tiled image, TILESET – level of tiles in the tile pyramid, LOCATION – a folder where the image or tile is stored, SERVER – the computer which store the images, and ACTIVITYSTATUS – the status of the image such as if the image is analyzed. Figure 3 shows IMAGE related entities with attributes. IMAGE is extended from OME with attributes *resolution*, *thumbnail* and *imagereference_uid*. TILESET represents different sets of tiles for an image at different tile sizes. The semantics of *imagereference_uid* in IMAGE is compatible with that of WHOLESIDEIMAGEREFERENCE entity in the PAIS model, and the semantics of *name* attribute of TILESET is compatible with that of REGION entity in PAIS model. Such semantics defines how the identifiers are generated, and makes it possible to link PIDB images to PAIS image and tile references. Study related entities inherit and extend OME model entities such as PROJECT, EXPERIMENT, EXPERIMENTER, EXPERIMENTALGROUP and DATASET, and LS DAM entity EXPERIMENTALSTUDY. The complete model can be found at PIDB Wiki.¹²

The compatibility enables semantics linkage of different databases, and provides integrated query support. For example, by querying a PAIS database and a PIDB database, we can retrieve images, tiles or regions of images based on certain characteristics such as features or classifications, as demonstrated later in query examples.

The physical model is derived from the logical model with entity tables and additional relationship tables for many-to-many relationships. There are 17 tables in the physical model. Figure 4 shows the example screenshots of IMAGE and TILEDIMAGE tables respectively.

In the model, images are represented as references to physical storage paths of images in file systems or as a BLOB data type (binary large object) stored in database. The reference based approach is used to manage large scale whole slide images, with typical sizes of a few hundred MBs or a few GBs, and tiled images, with typical sizes of a few MBs to tens of MBs. Thumbnails are small in size and can be managed uniformly inside a database as BLOB objects to support direct SQL query access.

3. WHOLE SLIDE IMAGE DATA MANAGEMENT ARCHITECTURE

The architecture of PIDB is shown in Figure 5 together with the PAIS database. Original whole slide images are scanned from slides, and the images can be tiled into smaller images for further analysis by image analysis algorithms. Image Loading Tool loads images and tiled images along with provenance information into Image Database. Analytical results are converted into PAIS XML documents through PAIS Document Generator. The PAIS data loading tool first submits compressed PAIS documents into a staging area in the PAIS database, and then maps XML data into the PAIS database. The Image Loading Tool extracts metadata from images, folders, and metadata configuration file, and populates the tables. The Image Database also provides a few user defined functions for extracting regions from images. The Image Database is co-located with PAIS database to support integrated queries. Image viewers, image analysis tools or data sharing tools can query the databases either through standard SQL query protocols such as JDBC or Web Services from an application server.

3.1 Image Loading Tool

Images and tiled images are first stored in a file system, with predefined folder structures and naming conventions. The Image Loading Tool supports three image folder structures and detects them automatically. One folder structure is shown in Figure 6. Under a root folder (*images*), there are multiple dataset folders (*dataset*). Each dataset folder has multiple data folders (*data1*, *data2*), and each data folder contains whole slide images. Tiled images are contained in a tile folder (*20Xtiles*) under the root folder. The tiled image folder name begins with a resolution of (*20X*) concatenated with term *tiles*. Image filenames also come with naming conventions, thus essential provenance information can be extracted. For example, the tiled image “TCGA-02-0001-01Z-00-DX1.svs-0000000000-0000000000.tiff” has a patient id *0001*, and is located at (0,0), where x and y coordinates are delimited by a hyphen. Such naming conventions, together with additional provenance information such as users, projects, etc, are specified in an XML configuration file and parsed by the Image Loading Tool.

The Image Loading Tool also automatically extracts image related metadata such as format, dimension, size, etc, and creates thumbnails during the loading process. In summary, the loading process performs the following tasks: i) collects image files and tiled image files based on folder structure conventions, ii) generate provenance metadata from configuration file and file naming conventions, iii) extract image metadata by parsing images, and iv) process images to generate thumbnails to be loaded into the Image Database. Java Advanced Image IO package¹³ is used for parsing tiled images and generating thumbnails during the loading process. For efficiency, thumbnails can also be pre-generated.

3.2 User Defined Functions

While images can be retrieved from image repository for manipulation with image processing tools, integrating image manipulation into a database engine could provide seamless integration with a unified expressive query interface through SQL and improved efficiency by reducing overhead of data communication. User defined functions (UDF) in DBMSs provide an opportunity for such close integration. A UDF can return a single value such as a BLOB data object or tabular value. Its implementation could be in multiple programming languages, such as Java or C/C++. Below are two example functions we have implemented to extract regions from a whole slide image (*getRegionImage*) and tiled image (*getRegionImageTile*) respectively.

- *getRegionImage*(imageuid, outputformat, x0, y0, width, height): extract a region from a whole slide image within a given window. *imageuid* specifies the image UID, *outputformat* can be JPG, PNG or TIF, and *x0*, *y0*, *width*, *height* are the top left x, y coordinates and the width and height of the window. This UDF is implemented in C language and relies on OpenSlide library¹⁴ to process images.
- *getRegionImageTile*(tiledimagename, outputformat, x0, y0, width, height): extract a region from a tiled image within a given window, where *tiledimagename* specifies the tile name. This UDF is implemented in Java and relies on Java Advanced Imaging library for extracting a region from a tiled image.

An example SQL query using a UDF is shown in Figure 8(a), which extracts a region as JPG format from a window with top left (x, y) coordinates (4096, 16384), height 800 and width 800 from image “TCGA-27-1836-01Z-DX2_20x”.

4. SUPPORT OF POWERFUL QUERIES

A significant advantage of PIDB is the support of comprehensive queries on images, image analytical results and provenance. Most queries can be expressed in SQL directly, and some complex queries will combine SQL queries with additional processing.

PIDB supports two types of query interfaces, SQL queries and RESTful Web Service based queries. The latter wraps SQL queries into HTTP requests for easy Web access. SQL queries include standard SQL queries and user defined functions for advanced operations.

4.1 Example Queries

PIDB supports queries at study, image, tile, and region levels, and provides integrated queries across images and analytical results, as demonstrated in a few example queries below.

- Study/patient level retrieval: return all image uids of a study named “In Silico brain tumor study”.
- Image level retrieval: retrieval of whole slide images based on metadata. For example, retrieve the original whole slide image based on image UID “TCGA-06-0152-01Z-00-DX6_20x”.
- Tile level retrieval: retrieval of tiled images based on metadata. For example, retrieve a tiled image with name “TCGA-06-0152-01Z-00-DX6-0000016384-0000004096”.
- Region level retrieval: retrieval of regions on a whole slide image. For example, extract a region at location (4096, 16384) with height 800 and width 800 from whole slide image “TCGA-27-1836-01Z-DX2_20x”. This query is supported with UDF *getRegionImage*.
- Integrated queries of derived results and images. For example, overlay all boundaries of nuclei segmented by an image analysis algorithm on top of the region generated in the query above. An example query is shown in Figure 7(a). As another example, return regions containing nuclei whose area and eccentricity features are closest to the mean area and eccentricity, as shown in Figure 7(b).

PIDB provides flexible query writing through SQL queries and some necessary programming if needed. Some queries could be much complex for users to develop. For example, Figure 4.1 shows two SQL queries to support overlay queries mentioned above – the first one retrieves the region from the Image Database, and the second query returns markup boundaries contained in the window from the PAIS Database. There is also additional processing of the SQL query results. The complexity of building queries could be a major hurdle for end users. To make it convenient for users to use the database, we have developed a RESTful Web Service based framework to build a set of Web based query interfaces.

4.2 RESTful Query Interfaces

REST is a software architecture style for distributed hypermedia systems such as the Web.¹⁵ REST is lightweight in representation, and can be used to build applications easily. In RESTful Web Services, data is viewed as resources which can be identified by their URIs. Normally implemented on the HTTP, RESTful Web Services are very efficient on transporting data over the Web. We define a set of common queries as RESTful Web Services. A service is defined with the syntax *http://hostname/databasename/functionpath;parameters*, in which *hostname* is the host name of the application server,

databasename represents the database, *functionpath* represents the function, and parameters are appended with semicolons as the delimiter. For example, the query shown in Figure 9 returns a region from a database named “TCGA”s with an image uid, window location and output image format specified as parameters.

We have defined a set of commonly used queries for PIDB as specified in Figure 10 (parameters ignored). The queries include:

- Metadata queries (/images/list), which return image UIDs, tilenames, patient UIDs, and the name of a tile containing a point based on certain parameters. For example, the query */images/list/imageuids;patientid=0152* returns image UIDs of a patient with patient UID “0152”.
- Image queries (/images/image), which return the image of a tile, a region, or the whole slide image. For example, the query */images/image / tile;tilename=TCGA-06-0152-01Z-00-DX6-0000016384-0000004096* returns the tiled image.
- Thumbnail queries (/images/thumbnail), which return the thumbnail of a tile, a region, or the whole slide image. For example, the query */images/thumbnail/ tile;imageuid=TCGA-06-0152-01Z-00-DX6 20x;format=png* returns the thumbnail of a tiled image in PNG format.
- Overlay queries (/images/overlay), which return an overlay of markups on a tiled image or a region within a window.
- Similarity queries (/images/similarity), which retrieve top-K similar regions based on features or classification labels. For example, the query */images/similarity/ feature;paisuid=TCGA-06-0195-01Z-00-DX1_20x_20x_NS-MORPH_1;rowsize=5;format=png* retrieves a set of similar region icons arranged in five icons per row for PAIS document “TCGA-06-0195-01Z-00-DX1_20x_20x_NS-MORPH_1”.

5. IMPLEMENTATION AND EVALUATION

PIDB system is implemented on IBM DB2 Enterprise Edition 9.7.2, running on CentOS 5.5. Region functions are implemented using OpenSlide Version 3.2.4 and Java Advanced Image IO 1.1.2. The Image Loading Tool is implemented in Java with Spring and Hibernate Frameworks. Web APIs are implemented using jersey 1.10, a JAX-RS (JSR 311) reference implementation for building RESTful Web services, together with JFreeChart 1.0.14 for generating necessary images. Web APIs run on Apache Tomcat 6.0.33. The implementation is system neutral, and can be easily ported to different systems.

PIDB is open source and will be released for public. The software package includes PIDB schema and index scripts, user defined functions, the Image Data Loading Tool, and Web API tool. It requires IBM DB2, OpenSlide, Java Development Kit (1.6.x and above), and Apache Tomcat Web Server (1.6.x and above). More comprehensive information can be found at PIDB wiki.¹²

We have a PIDB deployment at Emory University for managing whole slide images used in an *In Silico* brain tumor study.¹⁶ The database runs on PowerEdge T410 Linux server with four quadcore CPUs, 16GB memory, and a 7200rpm hard drive. Currently we have 474 whole slide images in the PIDB database, along with the corresponding PAIS database which manages all analytical results such as spatial markups and nuclear features and classifications.

PIDB supports efficient queries. A special query based on *getRegionImage* function is tested and the performance is shown in Figure 5. Figure 11(a) shows query time for a few whole slide images at different sizes where regions to be extracted have varying dimensions. Queries with regions smaller than 2048x2048 pixels all take less than 1 second. Queries with region dimensions at 4096x4096 take 3-6 seconds. Note that there is no obvious correlation between query performance and the size of the original whole slide images (SVS, in TIFF/BigTIFF format). This is a significant benefit as retrieval of regions can be very efficient even for large images. Figure 11(b) shows query performance with varying dimensions of regions for one whole slide image with size at about 400MB. We can see a sublinear query time over the area of the region.

6. DISCUSSION

While whole slide images are often managed as simple files, the lack of normalization of metadata and provenance makes it difficult to support queries and data exchange. Modeling and managing pathology images in databases provides immediate benefits on the value and usability of data through standardized data representation, data normalization, and provenance tracking. The database provides powerful query capabilities, which are otherwise difficult or cumbersome to support by other approaches such as programming languages.

Commercial slide scanner vendors often provide a whole slide image management system, for example, Aperio's Spectrum¹⁷ is a Web-based digital pathology information management system. Such systems are often closed and come with limited query capabilities, and it is difficult for users to specify comprehensive queries with standard query languages or APIs. PIDB provides an open source solution with comprehensive data model and query capabilities, which can also be extended by users.

PACS systems take a store and push approach for archiving and retrieving images, which are stored on a central server. Thus when a request is sent to view or process an image, the whole image has to be downloaded to the client machine for client applications. PIDB provides retrievals at multiple level granularities to reduce data to be downloaded. For example, we can efficiently retrieve any region within a couple of seconds.

In PIDB, we manage original whole slide images, and support major virtual slide formats supported by OpenSlide.¹⁴ With extension of DICOM for whole slide images, existing PACS systems have to be extended to support additional metadata and the pyramid based image tiling model, which are demonstrated by the metadata and data model of PIDB. Thus PIDB could provide a reference for DICOM extensions for whole slide images. PIDB can also be extended with DICOM specific metadata and additional UDFs to support DICOM based whole slide images. Especially, the comprehensive query support, especially the integrated queries of image data and analytical results, makes PIDB a powerful medical imaging geographical information system (GIS), which is not possible in traditional PACS systems. While a simple solution could be to provide two systems running in parallel – one as PACS for data exchange and one as medical imaging GIS (PIDB + PAIS), a viable solution could be to extend traditional PACS systems with such query capabilities.

7. RELATED WORK

The Open Microscopy Environment (OME) project¹⁰ has developed a data model and a database system that can be used to represent, exchange, and manage image data and metadata. The OME provides a data model of common specification for storing details of microscope setup and image acquisition, and recently starts the support of whole slide images in a limited scope. Cell Centered Database (CCDB)^{18,19} is a system and data model

developed to capture image analysis output, image data, and information on the specimen preparation and imaging conditions that generated the image data. The CCDB implements an ontology link to support semantic queries and data source federation. DICOM Working Group 26 developed a DICOM-based standard for storing whole slide images. As a first step, two supplements 122 and 145 are developed for formal representations for specimens and whole slide images.¹

Pathology image analysis produces large scale spatially derived information, and indeed, pathology analytical results are spatially oriented and most queries are GIS flavor queries. To manage algorithm results, human annotations and provenance, and to efficiently support complex queries, we developed a system called PAIS (Pathology Analytical Imaging Standards).^{6,7,8,9} The PAIS data model consists of 62 UML classes, representing information on markups (spatial shapes representing regions, cellular or subcellular objects), annotations (including calculated features or observations such as classifications associated with markups), and provenance (image references, analysis, algorithm and parameters). Markups could be represented as geometric shapes to represent the boundaries of segmented objects, such as tumor regions, blood vessels, and nuclei. PAIS employs a spatial DBMS based implementation for managing data, and provides three major types of tables: i) *Spatial tables* for representation of markup objects with geometric shapes, ii) *Feature and observation tables* to capture calculated features, such as area, perimeter, and eccentricity, and descriptive observations, such as classifications of regions or nuclei; and iii) *Provenance tables* to represent image references, subject, specimen, the invoked algorithms, etc. Such spatial oriented implementation provides powerful “GIS” like query support. PIDB is fully compatible with PAIS to support integrated queries.

8. CONCLUSION

Digital pathology is transforming the pathology domain into a new era, and a major need is to manage and query whole slide images to facilitate image visualization, exchange, and data analysis. The Pathology Image Database System provides a standard compatible data model and a unified query interface for querying and retrieving images and analytical results at multiple level granularities and resolutions. PIDB provides efficient comprehensive query support for not only metadata based queries but also powerful GIS-like queries. PIDB is used to support brain tumor studies at Emory University *In Silico* Brain Tumor Research Center. The system is open source and has the potential to be integrated with PACS with powerful query capabilities to support pathology imaging.

Acknowledgments

This work was supported in part by Contracts HHSN261200800001E and N01-CO-12400 from the National Cancer Institute, R24HL085343 from the National Heart Lung and Blood Institute, by Grants R01LM011119-01 and R01LM009239 from the National Library of Medicine, RC4MD005964 from National Institutes of Health, NIH NIBIB BISTI P20EB000591, and PHS Grant UL1RR025008 from the Clinical and Translational Science Awards program. IBM provides academic license for DB2 in this project.

REFERENCES

1. Dicom wg-26. 2011. <http://medical.nema.org/DICOM/minutes/WG-26/>
2. Cooper L, Kong J, Gutman D, Wang F, et al. An integrative approach for in silico glioma research. Biomedical Engineering, IEEE Transactions on. Oct.2010 57:2617–2621.
3. Kong J, L. C, Moreno C, Wang F, Kurc T, Saltz J, Brat D. In Silico Analysis of Nuclei in Glioblastoma using Large-scale Microscopy Images Improves Prediction of Treatment Response. [EMBC]. Aug-Sep;2011

4. Cooper LA, Kong J, Gutman DA, Wang F, Gao D, Appin C, Cholleti SR, Pan TC, Sharma A, Scarpace L, Mikkelsen T, Kurc T, Moreno CS, Brat DJ, Saltz JH. Integrated morphologic analysis for the identification and characterization of disease subtypes. *J Am Med Inform Assoc.* (In Press).
5. Wang F, Kong J, Cooper L, Pan T, Tahsin K, Chen W, Sharma A, Niedermayr C, Oh TW, Brat D, Farris AB, Foran D, Saltz J. Data model and database for high-resolution pathology analytical image informatics. *Journal of Pathology Informatics.* 2011; 2(32)
6. Wang F, Kong J, Gao J, Vergara-Niedermayr C, Adler D, Cooper L, Chen W, Kurc T, Saltz J. High Performance Analytical Pathology Imaging Database for Algorithm Evaluation. [MICCAI/MICCAI-DCI]. 2011
7. Foran DJ, Yang L, Chen W, Hu J, Goodell LA, Reiss M, Wang F, Kurc TM, Pan T, Sharma A, Saltz JH. Imageminer: a software system for comparative analysis of tissue microarrays using content-based image retrieval, high-performance computing, and grid technology. *JAMIA.* 2011; 18(4): 403–415. [PubMed: 21606133]
8. Pathology analytical imaging standards.. 2011. <https://web.cci.emory.edu/confluence/display/PAIS>
9. Goldberg I, Allan C, Burel J-M, Creager D, Falconi A, Hochheiser H, Johnston J, Mellen J, Sorger P, Swedlow J. The Open Microscopy Environment (OME) Data Model and XML File: Open Tools for Informatics and Quantitative Analysis in Biological Imaging. *Genome Biol.* 2005; 6(R47)
10. The Open Microscopy Environment.. 2011. <http://www.openmicroscopy.org>
11. Life Sciences Domain Analysis Model (LS DAM).. 2011. [https://wiki.nci.nih.gov/display/LS/Life+Sciences+Domain+Analysis+Model+\(2011\)](https://wiki.nci.nih.gov/display/LS/Life+Sciences+Domain+Analysis+Model+(2011))
12. Pathology Image Database System Wiki.. 2011. <https://web.cci.emory.edu/confluence/display/PIDB>
13. Java Advanced Imaging.. 2011. <http://java.sun.com/products/java-media/jai/iio.html>
14. OpenSlide.. 2011. <http://openslide.org/>
15. Fielding, RT. PhD thesis. University of California; Irvine: 2000. Architectural Styles and the Design of Network-based Software Architectures.
16. Emory in silico brain tumor research center.. 2011. <https://wiki.nci.nih.gov/display/ISCRE/Emory+University+In+Silico+Brain+Tumor+Research+Center>
17. Aperio Spectrum.. 2011. <http://www.aperio.com>
18. Martone ME, Tran J, Wong WW, Sargis J, Fong L, Larson S, Lamont SP, Gupta A, Ellisman MH. The cell centered database project: An update on building community resources for managing and sharing 3d imaging data. *Journal of Structural Biology.* 2008; 161(3):220–231. [PubMed: 18054501]
19. OpenCCDB.. 2011. <http://openccdb.org>

Figure 1.
Image Granularities

Figure 2.
Overview of PIDB Logical Model

Figure 3.
Image Related Entities

IMAGEREFERENCE_UID	FILESIZE	RESOLUTION	ACQUISITIONDATE	FORMAT	NAME
TCGA-06-0152-01Z-00-DX6_20x	693487031	20x	Sep 21, 2009 12:00...	2	intgen.org_GBM.tissue_images.6.0.0/TCGA-06-0152-01Z-00-
TCGA-06-0152-01Z-00-DX7_20x	538737981	20x	Sep 21, 2009 12:00...	2	intgen.org_GBM.tissue_images.6.0.0/TCGA-06-0152-01Z-00-
TCGA-06-0648-01Z-00-DX1_20x	408185129	20x	Sep 21, 2009 12:00...	2	intgen.org_GBM.tissue_images.6.0.0/TCGA-06-0648-01Z-00-
TCGA-06-0648-01Z-00-DX2_20x	122089111	20x	Sep 21, 2009 12:00...	2	intgen.org_GBM.tissue_images.6.0.0/TCGA-06-0648-01Z-00-
TCGA-08-0352-01Z-00-DX1_20x	200301225	20x	Sep 21, 2009 12:00...	2	intgen.org_GBM.tissue_images.6.0.0/TCGA-08-0352-01Z-00-
TCGA-08-0352-01Z-00-DX2_20x	343370037	20x	Sep 21, 2009 12:00...	2	intgen.org_GBM.tissue_images.6.0.0/TCGA-08-0352-01Z-00-

TILENAME	X	Y	HEI	W	FILESIZE	RESOLUTION	LOCATION_ID	TILEDTIME
TCGA-06-0152-01Z-00-DX6-0000016384-0000004096	16,384	4,096	4,096	4,096	49130082	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000028672-0000024576	28,672	24,576	4,096	4,096	62238820	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000032768-0000004096	20,480	12,288	4,096	4,096	58827070	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000036864-0000036864	36,864	28,672	4,096	4,096	17083732	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000000000-0000032768	0	32,768	4,096	4,096	1372144	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000008192-0000020480	8,192	20,480	4,096	4,096	58374600	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000024576-0000004096	24,576	4,096	4,096	4,096	13713008	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000000000-0000000000	0	0	4,096	4,096	23334700	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000020480-0000004096	20,480	4,096	4,096	4,096	21721472	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000040960-0000045056	40,960	45,056	4,096	4,096	810300	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000028672-0000016384	28,672	16,384	4,096	4,096	61866340	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000016384-0000008192	16,384	8,192	4,096	4,096	54909482	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000008192-0000028672	8,192	28,672	4,096	4,096	60855854	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000032768-0000004096	32,768	40,960	4,096	4,096	17487976	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000004096-0000004096	4,096	40,960	4,096	4,096	10910226	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000012288-0000045056	12,288	45,056	4,096	4,096	913898	20X		1 Dec 29, 2010
TCGA-06-0152-01Z-00-DX6-0000024576-0000004096	24,576	40,960	4,096	4,096	14834982	20X		1 Dec 29, 2010

Figure 4.
Example IMAGE and TILEDIMAGE Tables

Figure 5.
The Architecture of Pathology Image Database System

```
/data/images      (ROOT FOLDER)
  /dataset
    /data1/
      image11.svs
      ...
    /data2/
      image21.svs
      ...
  /20Xtiles
    /tileset/
      tileimage11.tif
      ...
```

Figure 6.
An Example Folder Structure for Images

Figure 7.
Examples of Overlay Query and Similarity Query

```
SELECT PI.GETREGIONIMAGE('TCGA-27-1836-01Z-DX2_20x', 'JPG',  
    4096, 16384, 800, 800)  
FROM PAIS.WHOLESIDEIMAGEREFERENCE  
WHERE pais_uid = 'TCGA-27-1836-01Z-DX2_20x_20x_NS-MORPH_1';
```

(a) Extract a Region

```
SELECT pais.plgn2str(polygon) AS boundary  
FROM pais.markup_polygon  
WHERE pais_uid= 'TCGA-27-1836-01Z-DX2_20x_20x_NS-MORPH_1'  
    AND tilename = 'TCGA-27-1836-01Z-DX2_0000004096-0000016384'  
    AND DB2GSE.ST_Contains( DB2GSE.ST_Polygon(  
        CAST('polygon( (4096 16384, 4896 16384, 4896 17184, 4096  
        17184, 4096 16384))' AS VARCHAR(30000) ), 100), polygon) = 1;
```

(b) Retrieve Boundaries of Nuclei

Figure 8.
SQL Queries for Overlaying Markups on an Image

```
http://www.openpais.org/tcga/images/image/window;  
imageuid=TCGA-27-1836-01Z-DX2_20x;x=4096;y=16384;w=800;h=800;format=JPG
```

Figure 9.
An Example Web Query URL

URL Path Name		Function
/images/list	/imageuids	List of image uids
	/tilenames	List of tilenames
	/patients	List of patient UIDs
	/tilenameofpoint	Return a tilename containing a point
/images/image	/tile	Tile image
	/window	Region image
	/wsi	Whole slide image
/images/thumbnail	/tile	Thumbnail of a tile
	/wsi	Thumbnail of a whole slide image
/images/overlay	/tile	Overlap of markups and a tile image
	/window	Overlap of a markups and the image of a window
/images/similarity	/feature	List top K similar subregion images based on features
	/cluster	List top K similar subregion images based on classifications

Figure 10.
Common Image Queries in RESTful Web Services

(a) Varied Images and Window Dimensions

(b) Varied Region Dimensions (Same Image, 400MB)

Figure 11.
Query Performance on Retrieving Regions