# Engineering a Peer-to-Peer Collaboratory for Tissue Microarray Research *

Cristina Schmidt and Manish Parashar
*The Applied Software Systems Laboratory*
*Department of Electrical and Computer Engineering, Rutgers University*
{*cristins,parashar*}*@caip.rutgers.edu*

Wenjin Chen and David J. Foran
*Center for Biomedical Imaging & Informatics*
*University of Medicine and Dentistry*
*New Jersey - Robert Wood Johnson Medical School*
*foran@umdnj.edu, wenjin@eden.rutgers.edu*

## Abstract

*This paper presents the design of a prototype peer-to-peer collaboratory for imaging, analyzing, and seamlessly sharing tissue microarrays (TMA), correlated clinical data, and experimental results across a consortium of distributed clinical and research sites. The overarching goal of this project is to facilitate cooperative oncology research aimed at improved understanding of the underlying mechanisms of disease onset and progression while simultaneously providing new insight into therapy planning and treatment. Key components of the collaboratory include a specification of metadata schematics for characterizing TMA specimens and abstracting their interpretations, an framework for automated and accurate analysis of digitized TMAs and a peer-to-peer infrastruture for indexing and discovery of TMA data and metadata, and a novel, optimized decentralized search engine that supports flexible querying with search guarantees and bounded costs. Prototype implementations of the automated TMA analysis component and the storage/discovery component and their evaluations are presented.*

## 1. Introduction

The tissue microarray (TMA) technique enables researchers to extract small cylinders of tissue from histological sections and arrange them in a matrix configuration on a recipient paraffin block such that hundreds can be analyzed simultaneously. A key advantage of TMA technology is that it allows amplification of limited tissue resources by providing the means for producing large numbers of small core biopsies, rather than a single section. Using this technology, a carefully planned array can be constructed with cases from pathology tissue block archives, such that a 20-year survival analysis can be performed on a cohort of 600 or more patients using only a few microliters of antibody. Another major advantage of the TMA technique is the fact that each specimen is treated in an identical manner. Therefore, reagent concentrations are consistent across discs within each specimen, as are the incubation times, temperatures and washing conditions. Using conventional protocols, a study composed of 300 tissue samples would involve processing of 300 hundred slides, i.e. at least 20 batches of 15 slides. Using TMA the entire cohort can be processed on a single slide. As a result, TMA technology holds great potential for reducing the time and cost associated with conducting research in tissue banking, proteomics, and outcome studies. However capturing, organizing, analyzing and characterizing, and sharing TMA data presents a number of significant challenges.

A key challenge is the analysis and evaluation of TMA samples. Currently, the primary methods used to evaluate the arrays involve manual, interactive review of TMA samples while they are subjectively evaluated and scored. An alternate, but less utilized approach for evaluation is to sequentially digitize each specimen for subsequent semi-quantitative assessment [7]. Both procedures ultimately involve the interactive evaluation of TMA samples which is a slow, tedious process that is prone to error.

Another challenge is the large volume of TMA data. Today, TMAs can contain from tens to hundreds of samples (0.6 to 2mm in diameter) arranged on a single slide. A digitized TMA specimen containing just 400 discs can easily approach 18GB in size. Given the increasing number of institutions and investigators utilizing TMA technology it is likely that modern facilities may easily generate tens of thousands of entries and terabytes of data. Clearly archiving, indexing and cataloging and mining this data across the TMA research community is a significant challenge and

* The research presented in this paper is supported in part by NSF via grants numbers ACI 9984357 (CAREERS), EIA 0103674 (NGS), EIA-0120934 (ITR), ANI-0335244 (NRT), CNS-0305495 (NGS), by DOE ASCI/ASAP (Caltech) via grant number 82-1052856 and by NIH contract 1 RO1 LM007455-01A1 from the National Library of Medicine. Additional support was provided by the Cancer Institute of New Jersey.

centralized solutions quickly become infeasible.

Finally, the increasing popularity of TMA has lead to more and more medical and research institutions being interested and conducting research in this area. While the exact focus of the research conducted by each of these groups may differ in terms of the patient group, the type of cancer, and/or the nature of the staining, being able to share data and meta-data has many advantages. Sharing experimental results and clinical outcomes data could lead to huge benefits in drug discovery and therapy planning. While some leading institutions are developing data management systems for TMA data, these systems are only minimally useful if the data isn't accessible to others in the scientific community. While there are ongoing efforts focused on developing standards to represent TMA data (e.g. [2]), existing efforts on sharing microarray data is based on centralized databases (e.g. http://ihome.cuhk.edu.hk/~b400559/arraysoft_public.html). However, the size of the data involved as well as issues of ownership can quickly limit the scalability and feasibility of this approach.

This paper presents the design of a prototype peer-to-peer collaboratory for imaging, analyzing, and seamlessly sharing tissue microarrays (TMA), correlated clinical data, and experimental results across a consortium of distributed clinical and research sites. Key components of the collaboratory addressed in this paper include:

**Specification of Semantic Metadata Schematics for TMA**: A key requirement for effective sharing of TMA data and metadata is the definition of semantic schemas for describing the TMA sample, the patient parameters, the evaluations conducted and the observed results. We propose an XML schema that is sufficiently rich to capture these dimensions and can be effectively parsed and presented using conventional technologies.

**Mechanisms and Tools for Automated TMA Analysis**: As mentioned above, current procedures for TMA analysis ultimately involve the interactive evaluation of TMA samples which is a slow, tedious process that is prone to error. Recent studies showed that having a pathologist score the specimens yields results that are subjective, difficult to reproduce, and do not reflect subtleties. Reliable quantitative measurements will allow investigators to make accurate predictions about patient outcomes and response to therapy. But for the most part, the promise of TMAs remains unrealized because scientists lack methods of high throughput, automated quantitative evaluation. To address this issue, we propose a prototype framework for automatically imaging, analyzing, and archiving tissue microarrays.

**Peer-to-Peer Infrastructure for Indexing and Discovery of TMA Data and Metadata**: In addition to the algorithmic and software development that is required for analyzing tissue microarrays, reliable tools are also needed to enable individual groups to dynamically acquire and seamlessly share imaged specimens and correlated metadata. However scalable information discovery in the absence of global knowledge of naming conventions remains a fundamental problem in large, decentralized, distributed environments. This is due to the heterogeneous nature and large volume of data and resources, their dynamism and the dynamism of the sharing environment. As a result, an information indexing and discovery system has to be efficient, fault-tolerant and self-organizing. Further, in the case of TMA data, security as well as the ability of each research group to maintain ownership as well as access control capabilities to their data is critical.

As a part of the TMA collaboratory we propose Squid, a P2P information indexing and discovery infrastructure. Each peer (e.g. research institution) in this system maintains ownership of its data and only publishes (in a controlled manner) metadata describing its data, which can then be discovered and search externally. The key innovation is a dimension reducing indexing scheme that effectively maps the multidimensional metadata information space to physical peers. Note that access to TMA data in this system is always controlled by the owner of the data.

**Flexible Query Engine with Search Guarantees**: A key requirement for the TMA collaboratory is the ability to flexibly and efficiently search TMA data and metadata across peer site using keywords, partial keywords, wildcards and ranges. Further, the underlying query engine should guarantee that all existing data elements that match a query are found with bounded costs. The Squid query engine supports such complex queries and guarantees that all existing data elements that match a query will be found with bounded costs in terms of number of messages and number of nodes involved.

The rest of this document is organized as follows. Section 2 present the overall architecture of the collaboratory. Section 3 presents the automated TMA analysis system. Section 4 describes the Squid P2P indexing and discovery infrastructure and the Squid query engine. Section 5 discusses the overall integration of the collaboratory. Section 6 presents related work in P2P and TMA data sharing research. Section 7 concludes this paper.

## 2. System Architecture

A schematic overview of the overall architecture of the prototype collaboratory is presented in Figure 1. The *data gathering module* collects the data that is processed and shared. There are three major sources of data: the TMA slides, which are constructed as presented in Figure 2, the clinical history of the donors, and the information related with the construction and preparation of the TMA slides.

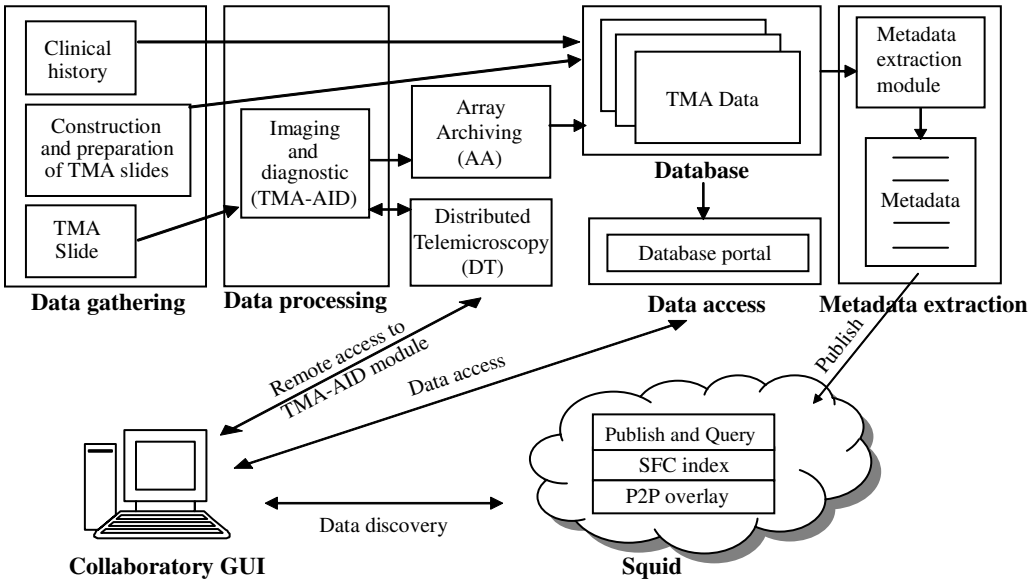



**Figure 1. A schematic overview of the peer-to-peer TMA collaboratory.**

The *data processing module* employs imaging and analyzing algorithms to extract relevant data from the TMA slide (i.e., the TMA-AID system). The data gathered is stored in the database using the Array Archiving (AA) subsystem. The TMA-AID system can be accessed remotely using the Distributed Telemicroscopy (DT) subsystem. The *data access module* enable remote access to the data stored in the local database.

The *metadata extraction module* extracts metadata describing the shared data from the local database. The metadata is published in *Squid* P2P storage and discovery system. Finally the *collaboratory GUI* allows users to flexibly search TMA data and metadata in Squid and access it through the Database Portal.

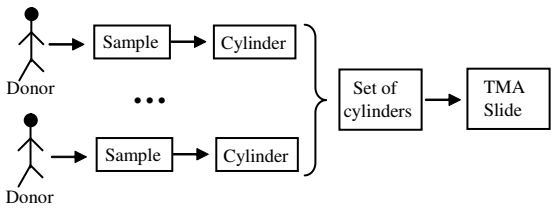



**Figure 2. TMA slide preparation.**

## 3. TMA-AID (Automated Imaging and Diagnostic) System

Traditionally, diagnostic pathology and cancer research was conducted by staining and analyzing hundreds upon hundreds of individual tissue sections-all in the name of one part, of one experiment. Increasingly such low-throughput monotony is giving way to "omics"-style science thanks to tissue microarrays. By reducing the amount of time and effort to process them this technology is accelerating the pace of research for oncologists, drug discoverers, and other scientists seeking to make sense of the data being generated out of genomics and proteomics laboratories.

In order to facilitate large-scale, multi-institutional studies a quick, reliable means for processing tissue microarrays is needed. We have developed a web-based prototype which features automated imaging, registration and intelligent archiving of tissue microarrays.

The system consists of a robotic microscope interfaced with a JAVA-based micro-controller and imaging workstation. The system utilizes a combination of sophisticated image processing and pattern recognition strategies to coregister specimens while the software directs a robotic microscope to systematically image specimens at multiple optical magnifications, delineate array discs, extract spectral and spatial signatures and populate the databases with the resulting data including pointers to imaged arrays. The system features a visually intuitive interface which enables local and remote users to manipulate the configuration of digitized arrays in order to facilitate new experimental designs and data assimilation.

Due to slight variations in specimen preparation and the possibility of mechanical distortion, software was developed to automatically compensate for these variations. Optical and mechanical system is automatically calibrated and measured by the TMA to ensure accurate stage location and measurements. An entropy-based, fast auto-focusing was developed to ensure image quality [3].

Algorithms which can automatically recover the grid structure of the array from a very low-resolution image map

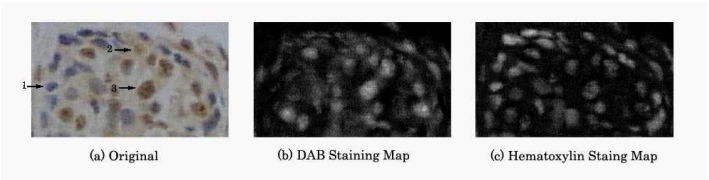



**Figure 3. Result of color decomposition. (a) One sub-field of original disc image. (b) The DAB staining map. (c) The hematoxylin staining map. Please note that some hematoxylin stained nuclei (arrow 1) are absent in the DAB staining map and some DAB stained nuclei (arrow 2) are absent in hematoxylin map. The majority of nuclei, however, bear a combination of the two stains (arrow 3).**

and locate and index each disc with proper column and row indexes has already been developed for the system. The size of tissue discs in the image map is estimated based upon the approximate core diameter of the physical array and the system while taking into account the scan settings at the time of acquisition. The image map is convolved with a template and a two-step, top-hat peak detection strategy applied to determine local maximums of the convolution output. Spatial constraints are applied in order to ensure that there is only a single candidate center point for each disc on the microarray [4].

When only a single stain is used to prepare a specimen, the integrated pixel density, i.e. luminance, within the corresponding microscopic image, can serve as a measure of target units present, which relates to the amount of specific antigen molecules or binding sites. When two or more dyes are used, however, one stain is often used to reveal the histological context within the specimen. In these cases each of the colors within the specimen can contribute to the luminance of the image. In this case, proper color separation should be performed before taking the luminance measurement.

We have developed a means for color decomposition which can reliably detect and characterize the staining characteristics within tissue microarray specimens. During the course of feasibility studies, we conducted experiments using specimens which had been stained with DAB chromogen and counter-stained with hematoxylin. The tissue microarray images in these studies exhibited multiple shades and combinations of these two dye colors.

Utilizing this novel color decomposition strategy, each of the discs within the arrays are split into DAB and Hematoxylin staining maps based upon their profiles in $L^*h_{1/v}{}^*C_{1/v}{}^*$ color space [5]. Figure 3 shows an original stained section of a disc and the corresponding output images after they have undergone quantitative analysis. As shown the staining characteristics of each nucleus appears as a continuous representation of staining intensity for each of the dyes. It is interesting to note that the protocol that we have described is able to unveil and quantify the underlying staining characteristics of even those cells which suffer from visual masking due to the counter-staining.

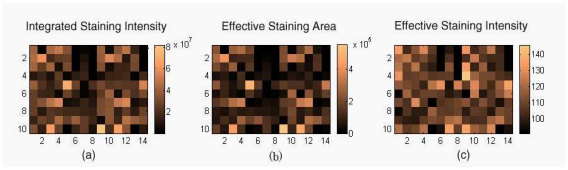



**Figure 4. (a) integrated staining intensity, (b) effective staining area and (c) effective staining intensity of all discs on one TMA specimen.**

The software automatically generates the following measures for each tissue disc: 1) integrated staining intensity which is computed as sum of the DAB staining intensity over the entire disc; 2) effective staining area which includes only those pixels which express above the threshold ; and 3) effective staining intensity which is computed as the average staining intensity divided by the effectively stained pixels. Using the color maps which are displayed to the right of each sub-figure, Figure 4 illustrates the results of measures (1-3) as they were computed over a specimen of 10x14 discs. The system scores discs exhibiting an effective staining area below 1000 pixels as non-stained. These discs are automatically assigned with 0 effective staining intensity.

Although the integrated staining intensity represents the overall staining level of each disc, given the heterogeneity of tissue specimens, it is only efficient in representing the staining level of the desired target when justified by the effective staining area. Figure 5 shows an example of two discs having different effective stained areas but similar stained color (effective staining intensity) resulting in different integrated staining intensities.

A Distributed Telemicroscopy (DT) subsystem [6] has been integrated with the TMA analysis prototype to enable individuals to control each other's robotic scopes from remote locations. Once a user logs into the system and issues the "Registration" command, the remote microscope automatically begins acquiring digital images of slightly overlapped frames of the tissue microarray sample in a raster pattern. The client application receives scaled version of the images and automatically stitches them together, in the

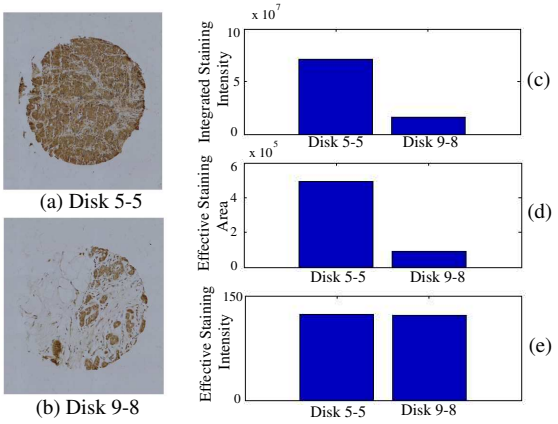



**Figure 5. Due to heterogeneity of tissue composition, two discs having different integrated staining intensities can have the same effective staining intensity. Shown (a) (b) are two discs located at 5th row 5th column and 9th row 8th column, respectively, having the same effective staining intensity (e), but different integrated staining intensities (c) due to differences in effective stained area (d).**

same raster pattern, giving rise to an image map. The unsupervised registration protocol is subsequently performed on the image map and the registration results appear in a heads-up display for the user as shown on the client interface with the recovered grid structure superimposed on the original map image. During preliminary performance studies the unsupervised TMA registration software was shown to reliably complete the low-resolution scan and successfully recover the grid structures of the samples, despite variations in the staining and rotation of the arrays. Upon completion of the registration process the software directed the robotic scope to sequentially digitize each core at multiple resolutions, archived the images, and populated an Oracle8i database with the location of each digitized specimen.

The Array Archiving (AA) subsystem was designed to provide the means for dynamically populating the database with new cases including image metrics and correlated profiles. This subsystem will continue to incorporate new standards for data exchange of tissue microarrays as they continue to evolve and gain acceptance [2].

The database is organized as shown in Figure 6. The physical specimen layer (PSL) of the database records information related to the construction and preparation of the physical TMA sample. These data are referred to as ”array profile”. A visually intuitive interface, array profile editor, has been developed and integrated into this layer to facilitate the design, editing and managing array profiles. The digital sample layer (DSL) of the database stores archived digital images including the image map and imaged tissue discs (at multiple resolutions). High-resolution images of tissue

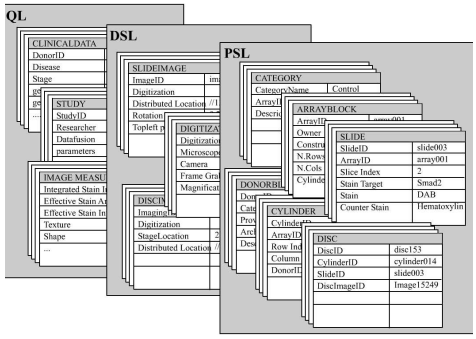



**Figure 6. Organization of the database. The major entities are highlighted in gray while auxiliary tables are rendered in black and white**

discs are broken down into subregions in order to facilitate network access. The third layer of the database, the quantification layer (QL), provides a data structure, which supports automated segmentation and computation of protein expression across each disc. The constituent entities of the AA subsystem were designed to be generalizable, to provide the underlying structure which can support a broader range of imaging applications.

## 4. Squid - Information Storage and Discovery System

This section presents Squid, a P2P information discovery system. Unlike most existing information discovery systems, Squid supports complex queries containing partial keywords, wildcards, and range queries and guarantees that all existing data elements that match a query will be found with bounded costs in terms of number of messages and number of nodes involved. The key innovation is a dimension reducing indexing scheme that effectively maps the multidimensional information space to physical peers.

The architecture of the Squid P2P information retrieval system is based on data-lookup systems [8, 13], and essentially implements an Internet-scale distributed hash table (DHT). The key difference is in the way we map the data elements[1] to the DHT space. In existing systems this is done using a hashing function that uniformly distributes data elements to nodes, and as a result a data element can be retrieved only if its exact identifier is known. In contrast, Squid uses a dimension-reducing mapping called Hilbert Space Filling Curve (SFC) [11], which is recursive and enables complex queries.

1 We will use the term ’data element’ to represent a piece of information that is indexed and can be discovered. A data element can be a document, a file, an XML file describing a resource, etc.

### 4.1. Publishing data

To support keyword searches, data elements in Squid are associated with a sequence of descriptive keywords. These keywords form a multidimensional keyword space where data elements are points in the space and the keywords are the coordinates. The keywords can be viewed as base-$n$ numbers, for example $n$ can be 10 for numeric keywords or 26 if the keywords are english words. An examples of a keyword space is shown in Figure 7.

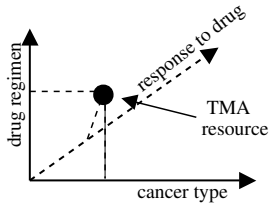



**Figure 7. A 3-Dimensional keyword space for storing TMA resources (e.g. metadata files for TMA data, and the locations of data), using the attributes: cancer type, drug regimen and response to drug.**

An SFC [11] is a continuous mapping from a $d$-dimensional space to a 1-dimensional space, generated recursively. Figure 9 (b) shows an example of Hilbert SFC in a 2-dimensional space. Aditional details about the use of Hilbert SFC in Squid can be found in [12].

In Squid, SFCs are used to generate the 1-$d$ index space from the multi-dimensional keyword space. Applying the Hilbert mapping to this multi-dimensional space, each data element can be mapped to a point on the SFC. Any range query or query composed of keywords, partial keywords, or wildcards, can be mapped to regions in the keyword space and the corresponding clusters (segments on the SFC curve) in the SFC.

The 1-dimensional index space is mapped onto an overlay network of peers. In our current implementation we use the Chord [13] overlay network topology. In Chord each node has a unique identifier ranging from 0 to $2^m$-1. These identifiers are arranged as a circle modulo $2^m$. Each node maintains information about (at most) $m$ neighbors, called *fingers*, in a *finger table*. The finger table is used for efficient routing and enables data lookup with O(log N) cost [13], where N is the number of nodes in the system. The finger table is constructed when a node joins the overlay, and it is updated any time a node joins or leaves the system. The cost of a node join/leave is O($\log^2$N).

In our implementation, node identifiers are generated randomly. Each data element is mapped, based on its SFC-based index or key, to the first node whose identifier is equal to or follows the key in the identifier space. This node is

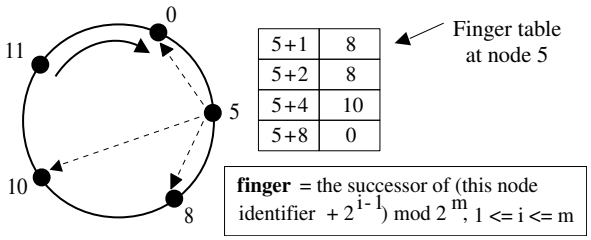



**Figure 8. Example of the overlay network. Each node stores the keys that map to the segment of the curve between itself and the predecessor node.**

called the *successor* of the key. An example of an overlay network with 5 nodes and an identifier space from 0 to $2^4$-1 is shown in Figure 8.

To summarize, publishing a data element in Squid consists of the following steps: attach keywords that describe the content of the data element, use the SFC-mapping to construct the index of the data element, and using this index, store the element at the appropriate node in the overlay. Figure 9 illustrates the publishing process.

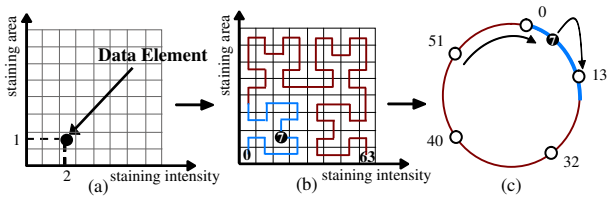



**Figure 9. The process of publishing a data element: (a) the data element (2, 1) is viewed as a point in a multidimensional space; (b) the data element is mapped to the index 7, using Hilbert SFC; (c) the data element is stored in the overlay (an overlay with 5 nodes and an identifier space from 0 to $2^6$-1) at node 13, the successor of the index 7.**

### 4.2. The Query Engine

The primary function of the query engine is to efficiently process user queries. The expected result of a query is the complete set of data elements that match the user's query.

As described above, data elements in the system are associated with a sequence of up to $d$ keywords, where $d$ is the dimensionality of the keyword space. The queries can consist of a combination of keywords, partial keywords, or wildcards. For example (30-40, breast cancer, Arimidex, *) is a valid query, specifying data regarding patients 30 to 40 years old, with breast cancer, treated with Arimidex and with any response to the treatment.

Processing a query consists of two steps: translating the keyword query to relevant clusters of the SFC-based index

space, and querying the appropriate nodes in the overlay network for data-elements.

If the query consists of complete keywords (no wildcard or range) it will be mapped to at most one point in the index space, and the node containing the matching data-element is located using the overlay's lookup protocol. If the query contains partial keywords, wildcards and/or ranges, the query identifies a region in the keyword space, which corresponds to a set of points in the index space. For example, in Figure 10 (a), the query (*, 4) identifies 8 data elements. The index (curve) enters and exits the region three times, defining three segments of the curve or clusters.

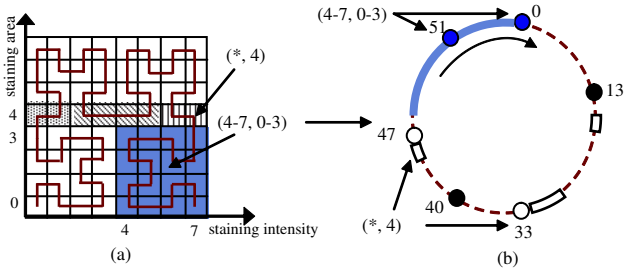



**Figure 10. Processing the query (*, 4): (a) the query defines a rectangular region in the 2-dimensional keyword space, and 3 clusters (3 segments on the SFC curve); (b) the clusters (the solid part of the circle) are stored at nodes 33 and 47, so these nodes will be queried.**

Once the clusters associated with a query are identified, straightforward query processing consists of sending a query message for each cluster, using the lookup mechanism provided by Chord. Figure 10 illustrated the query processing: the range query (4-7, 0-3) defines a rectangular region in the 2-dimensional keyword space, and identifies one cluster. The cluster is stored into the overlay at nodes 51 and 0, so these two nodes will be queried.

The node that initiated the query can not know if a cluster is stored in the network or not, or if multiple clusters are stored at the same node, to make optimizations. The number of clusters can be very high, and sending a message for each cluster is not a scalable solution. We optimized the query processing by considering the recursive nature of the SFC. Details about the optimization can be found in [12].

### 4.3. Experimental Evaluation

The performance of Squid is evaluated using a simulator. As the overlay network configuration and operations are based on Chord [13], its maintenance costs are of the same order as in Chord. An evaluation of the query engine is presented below.

This experiment measures the scalability of the system. The system size increases from 1000 nodes to 5400 nodes, and the number of stored keys (unique keyword combinations) increases from $2*10^5$ to $10^6$. Each key may be associated with one or more data elements. 2-dimensional (2D), and a 3-dimensional (3D) keyword spaces were evaluated, using the following types of queries:

**Q1**: Queries with one keyword or partial keyword. For example (her2, *) and (her2, *, *) are valid Q1 queries, for cases where a Her2 marker was used.

**Q2**: Queries with two to three keywords or partial keywords. For example (breast cancer, arimidex, *) is a query for breast cancer cases, treated with Arimidex, and any response to drug.

**Q3**: Range queries:

**Q3_1**: (keyword, range, *).

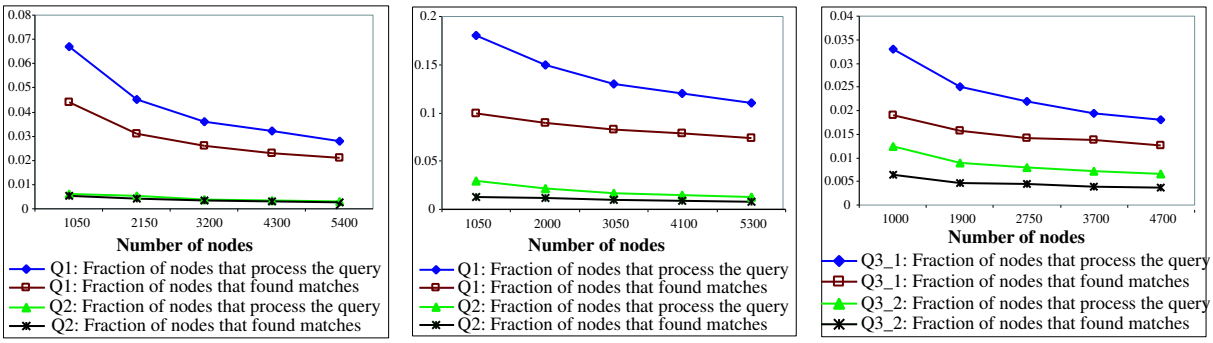
**Q3_2**: (range, range, range).

A set of queries of each type were tested. The queries were chosen such that the number of matches represents the same fraction of the total data regardless of the size of the system (number of nodes) and the quantity of data. For each query we measured the number of nodes that process it (refine it and search for matches) and the number of nodes that found matching data (data nodes). The results were averaged and normalized.

As seen in Figure 11 (a), (b) and (c) the number of processing and data nodes is a small fraction of the total nodes and increases at a slower rate than the system size. For a 2D keyword space, the average number of processing nodes is below 8%, and the number of data nodes is below 5%, and these percentages decrease as the system size increases (number of nodes and data), demonstrating the scalability of the system. The number of data nodes is close to the number of processing nodes, indicating that the query optimizations effectively reduce the number of nodes involved. Also, Q2 queries are more efficient than Q1 queries, which is expected. This is because query optimization and pruning are more effective when both keywords are at least partially known.

The 2D and 3D results follow a similar pattern, the only difference is the magnitude of the results. As described in Section 4, data elements that share a specific keyword will typically be mapped to disjoint fragments on the curve (clusters). In the 3D case the number of such fragments is larger than in the 2D case - 3 keywords result in a "longer" curve. Consequently, the results obtained for the 3D case for all the metrics have the same pattern as the 2D case but a larger magnitude.

Note that even under these conditions, the results are good. A keyword search system like Gnutella (http://gnutella.wego.com) would have to query the entire network using some form of flooding to guarantee that all the matches to a query are returned, while in the




(a) (b) (c)

**Figure 11. (a) Results for 2D keyword space for query types Q1 and Q2; (b) Results for 3D keyword space for query types Q1 and Q2; (c) Results for 3D keyword space for range queries.**

case of a data lookup system such as Chord [13], one would have to know all the matches a priori and look them up individually.

We also implemented the system on Project JXTA (http://www.jxta.org), a general-purpose peer-to-peer framework. The overlay network (e.g. Chord) and Squid are implemented as event-driven JXTA services.

The system was evaluated on a Linux cluster consisting of 64 1.6 GHz Pentium IV machines and an 100Mbps Ethernet interconnection. Each of the 64 acted as a node in the overlay.

The experiment measured the Squid overhead at a node. Three sets of queries were used, the first containing wildcards, the second containing ranges and the third containing both wildcards and ranges. The query processing overheads at the Squid layer were measured at each node and averaged. The results are plotted in Figure 12. The measured overhead includes times for cluster refinements and subclusters lookup. As Figure 12 shows, the overhead grows slowly and at a much smaller rate than the system size. This demonstrates that Squid can effectively scale to large numbers of nodes while maintaining acceptable query processing times. As expected, the routing times are high for queries with wildcards as they involve a larger number of clusters and correspondingly larger number of nodes.

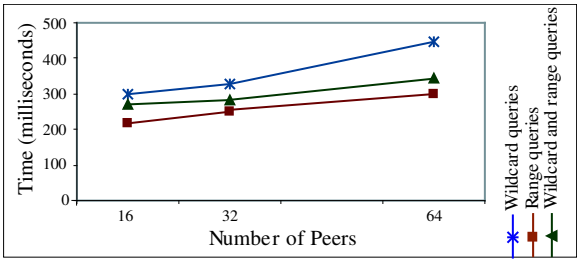



**Figure 12. Query processing overhead at a node.**

# 5. System Integration

## 5.1. The P2P Infrastructure

The participating peers in the P2P infrastructure typically run on machines at hospitals, research centers and universities. Specialized software agents at each local site extract metadata from the local database, and publish it in the P2P storage and discovery system. However, rather than storing the data, only references to the data described by the metadata are stored. This behavior is desired because access to data is typically restricted based on access credentials. The Squid P2P infrastructure thus enables global discovery (with some access control restrictions) of metadata while allowing the peers to maintain ownership and locally control access to their data.

Since peers typically run on dedicated machines, the machines will be likely to be robust and stay alive for longer periods of time, and the P2P system can become quite stable. While this property is not necessary for Squid, it can be exploited to reduce the maintenance costs of the overlay network.

## 5.2. Metadata extraction

A part of the data available locally is indexed using metadata, and will be used in queries. Certain image files that don't have very suggestive metadata associated with them are not used. The metadata associated with a piece of data is extracted from the local database by a software agent, who then publishes it to Squid along with references to the data. The agent checks the database for changes at regular intervals, looking for new data. The process is illustrated in Figure 13.

An example of the XML metadata extracted from a database record (a case) is presented in Figure 14. The values of the attributes are used as keywords. Squid constructs the index using the keywords. The index is then used to lo-

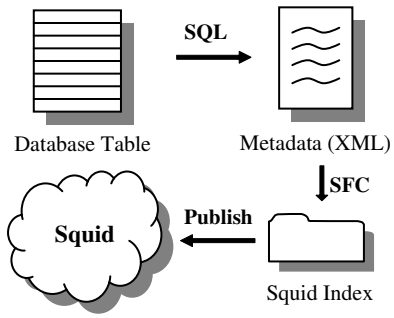



**Figure 13. The process of publishing data in Squid: each piece of data has associated an XML file with metadata. The metadata is used to publish the XML and the location of the data in Squid.**

cate the peer node where the XML metadata will be stored, together with the address of the database containing more information about this case (e.g. images, case history, etc).

```
<tma_entry>
    <CaseOrigin>New Jersey</CaseOrigin>
    <Age>61</Age>
    <CancerType>breast cancer</CancerType>
    <Marker>Her2</Marker>
    <DrugRegimen>Arimidex</DrugRegimen>
    <ResponseToTreatment>high</ResponseToTreatment>
<tma_entry>
```

**Figure 14. Example of metadata (XML) extracted from the database.**

### 5.3. Searching for Data

The system is queried through a friendly graphical user interface (GUI). The user query is presented to Squid as an XML file. Squid parses the document, extracts the user query and resolves it. The results are presented to the user and consist of links to relevant data in databases maintained

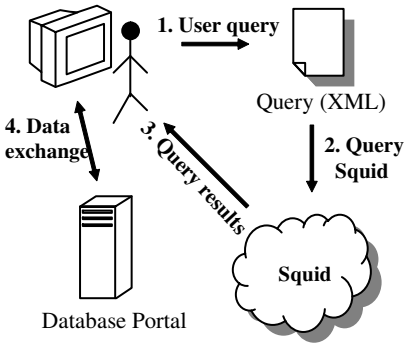



**Figure 15. Searching information using Squid.**

by hospitals, research centers, etc. The user can then contact the owners of the data to obtain required permission to access the data using the database portal. Note that the access to the data is outside Squid and is subject to the hospital's (or research center's) regulations. Figure 15 illustrates this process.

```
<tma_query>
    <CaseOrigin>*</CaseOrigin>
    <Age>30-40</Age>
    <CancerType>breast cancer</CancerType>
    <Marker>*</Marker>
    <DrugRegimen>chemo</DrugRegimen>
    <ResponseToTreatment>high</ResponseToTreatment>
<tma_query>
```

**Figure 16. Example of a user query.**

An example of a possible user query is presented in Figure 16. In the example, the user is interested in data about patients between 30 and 40, with breast cancer, who are treated with chemo, and have a high response to the treatment. The marker type and the case origin can be anything.

## 6. Related Work

A key component of a collaboratory for TMA research is a scalable and flexible P2P storage and discovery system. Data to be shared is owned by hospitals and research centers that would prefer to maintain ownership and control of their data. As a result, a P2P sharing infrastructure is a natural solution. The number of participanting sites can be large, and the quantity of data to be shared is also very large, making scalabilty of the P2P system a critical requirement. Further, the system must provide search guratees so that even "rare" data that exists in the sytem must be a found by a matching query. Finally, the P2P discovery system has to be flexible, to allow queries with ranges and wildcards. Squid P2P information storage and discovery system has all above mentioned properties. The rest of this section summarizes the current landscape of P2P storage/discovery systems.

Existing information storage/discovery systems can be broadly classified as unstructured, hybrid or structured. Unstructured systems as Gnutella (http://gnutella.wego.com) support complex queries (including wildcards and ranges), but they do not offer any search guarantees, since they are using flooding techniques to process queries. Hybrid systems, such as Napster (http://napster.com), use centralized directories to provide guarantees, which can limit their scalability.

Structured systems can be further categorized in "data lookup" and "structured keyword" systems. Data lookup systems [13, 8, 10] guarantee that if information exists in the system, it will be found by the peers within a bounded number of hops. These systems build on structured overlays and essentially implement Internet-scale Distributed Hash

Tables (DHT). Information is located using unique and globally known data identifiers and complex queries are not supported. Structured keyword search systems extend data lookup systems with search capabilities. The Squid system, presented in this article, falls into this category. Other approaches that fall in this category include PeerSearch [14], Reynolds and Vahdat [9] and Andrzejak and Xu [1].

Squid differs from these approaches in that it uses an SFC-based indexing scheme to map data elements to peers using keywords, and consequently, when resolving a query only the data elements that match all the keywords in the query are retrieved. It also supports flexible searching using partial keywords, wildcards, and range queries. Andrzejak and Xu [1] propose a discovery system based on Hilbert SFC. Unlike Squid [12], this system uses the inverse SFC mapping, from a 1-dimensional space to a $d$-dimensional space, to map a resource to peers based on a single attribute (e.g. cancer type). Squid uses SFC's to encode the $d$-dimensional keyword space to a 1-dimensional index space. This way we can map and search a resource using multiple attributes.

To our knowledge, there is no other large-scale distributed system for sharing TMA data. Until recently, no standard existed to represent TMA data. Each institution stored the data in local databases, in a custom format, making data sharing impossible. However, as part of a recent initiative, the TMA community is developing an XML-based, open TMA data exchange specification [2].

## 7. Summary and Conclusions

In this paper we presented the design of a prototype peer-to-peer collaboratory for imaging, analyzing, and seamlessly sharing tissue microarrays (TMA), correlated clinical data, and experimental results across a consortium of distributed clinical and research sites. Specifically, we described the design and evaluations of the TMA AID system for automated TMA analysis, the TMA metadata extraction module and the Squid P2P storage/discovery infrastructure.

TMA research is becoming increasingly popular and the TMA research community already spans medical and research institutions across the US and overseas. We believe that enabling a peer-to-peer sharing of experimental results and clinical outcomes data across this community could lead to huge benefits in drug discovery and therapy planning. The peer-to-peer collaboratory presented in this paper is an initial prototype of such a sharing infrastructure.

## References


[1] A. Andrzejak and Z. Xu. Scalable, efficient range queries for grid information services. In *Proceedings of the 2nd IEEE International Conference on Peer-to-Peer Computing (P2P 2002)*, pages 33–40, Sept. 2002.

[2] J. Berman, M. Edgerton, and B. Friedman. The tissue microarray data exchange specification: A community-based, open source tool for sharing tissue microarray data. *BMC Medical Information and Decision Making*, 3(1):5, May 2003.

[3] W. Chen and D. Foran. Computer-assisted microscopy (cam): robotic microscopy with applications in unsupervised specimen analysis and telemedicine. Submitted for publication, 2004.

[4] W. Chen, D. J. Foran, and M. Reiss. Unsupervised imaging, registration and archiving of tissue microarrays. In *Proceedings of the AMIA Symposium*, pages 136–139, 2002.

[5] W. Chen, M. Reiss, and D. J. Foran. Unsupervised tissue microarray analysis for cancer research and diagnostics. *IEEE Transactions on Information Technology in Biomedicine, accepted for publication*, 2004.

[6] D. Foran, D. Comaniciu, P. Meer, and L. Goodell. Computer-assisted discrimination among malignant lymphomas and leukemia using immunophenotyping, intelligent image repositories, and telemicroscopy. *IEEE Transactions on Information Technology in Biomedicine*, 4(4):265–273, Dec. 2000.

[7] N. Mucci, G. Akdas, S. Manely, and M. Rubin. Neuroendocrine expression in metastatic prostate cancer: evaluation of high throughput tissue microarrays to detect heterogeneous protein expression. *Human Pathology*, 31(4):406–414, 2000.

[8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM*, pages 161–172, San Diego, CA, Aug. 2001.

[9] P. Reynolds and A. Vahdat. Efficient peer-to-peer keyword searching. In *Proceedings of ACM/IFIP/USENIX Middleware Conference*, volume 2672, pages 21–40, Rio de Janeiro, Brazil, June 2003.

[10] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for largescale peer-to-peer systems. In *Proceedings of IFIP/ACM Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, Heidelberg, Germany, Nov. 2001.

[11] H. Sagan. *Space-Filling Curves*. Springer-Verlag, 1994.

[12] C. Schmidt and M. Parashar. Flexible information discovery in decentralized distributed systems. In *Proceedings of the 12th High Performance Distributed Computing (HPDC)*, pages 226–235, Seattle, WA, June 2003.

[13] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of ACM SIGCOMM*, pages 149–160, San Diego, CA, Aug. 2001.

[14] C. Tang, Z. Xu, and M. Mahalingam. Peersearch: Efficient information retrieval in peer-to-peer networks. Technical Report HPL-2002-198, HP Labs, 2002.