Tissue MicroArray: a Distributed Grid Approach for Image Analysis

Federica VITI[a], Ivan MERELLI[a,1], Antonella GALIZIA[b], Daniele D'AGOSTINO[b], Andrea CLEMATIS[b] and Luciano MILANESI[a]

[a] *Institute for Biomedical Technologies, National Research Council, Segrate (Milan), Italy*

[b] *Institute for Applied Mathematics and Information Technology, National Research Council, Genoa, Italy*

Abstract. The Tissue MicroArray (TMA) technique is assuming even more importance. Digital images acquisition becomes fundamental to provide an automatic system for subsequent analysis. The accuracy of the results depends on the image resolution, which has to be very high in order to provide as many details as possible. Lossless formats are more suitable to bring information, but data file size become a critical factor researchers have to deal with. This affects not only storage methods but also computing times and performances. Pathologists and researchers who work with biological tissues, in particular with the TMA technique, need to consider a large number of case studies to formulate and validate their hypotheses. It is clear the importance of image sharing between different institutes worldwide to increase the amount of interesting data to work with. In this context, preserving the security of sensitive data is a fundamental issue. In most of the cases copying patient data in places different from the original database is forbidden by the owner institutes. Storage, computing and security are key problems of TMA methodology. In our system we tackle all these aspects using the EGEE (Enabling Grids for E-sciencE) Grid infrastructure. The Grid platform provides good storage, performance in image processing and safety of sensitive patient information: this architecture offers hundreds of Storage and Computing Elements and enables users to handle images without copying them to physical disks other than where they have been archived by the owner, giving back to end-users only the processed anonymous images. The efficiency of the TMA analysis process is obtained implementing algorithms based on functions provided by the Parallel IMAge processing Genoa Library (PIMA(GE)^2 Lib). The acquisition of remotely distributed TMA images is made using specialized I/O functions based on the Grid File Access Library (GFAL) API. In our opinion this approach may represent important contribution to tele-pathology development.

Keywords. Tissue Microarray, Grid platform, image analysis

---

[1] Corresponding Author: Ivan Merelli; E-mail: ivan.merelli@itb.cnr.it

Introduction

In recent years, microarray technology has increased its potential and now it produces a large amount of genomic, transcriptomic and proteomic data. The data become interesting and informative only if they are screened, filtered, statistically analyzed, correlated to previously existing information and, above all, scientifically validated, to produce accurate and biological consistent predictions.

Gene expression microarrays are used, for example, to determine which genes are differentially expressed in a pathological tissue and in a healthy one. They rely on DNA/oligonucleotides probes, synthesized or spotted on a glass coated chip, which are hybridized with retro transcribed RNA from different cells. This is a useful high throughput technique to deliver a first large-scale screening of the transcriptomic products of cells. The output is a large amount of gene data, which must be analyzed to identify the most representatives transcripts for the cell conditions: only these genes will be valuated and studied.

The Tissue MicroArray technique represents a good validation of these selected data [1]. In fact, while microarrays use tens of thousands of probes as input, and give back only some tens of gene sequences (those expressed) as output [2], TMA can take hundreds of tissues as input to evaluate a single biological entity in a parallel way, concerning many tissues of the same paraffin block [3]. Researchers can check gene expression microarray output using probes to highlight results directly on tissues, validating the different quantity of genes or proteins in case-control samples. This technique allows to analyse at once many tissues, with a decrease in costs and time and with a statistical enrichment of biological profiles. Accurate analysis of the biological structure present in each sample of TMA can be obtained using different molecular biology techniques: most common analyses are IHC (immunohistochemistry), ISH-RNA and ISH-DNA (in situ hybridization for RNA and DNA), and FISH (fluorescent in situ hybridization).

The combinatorial explosion in analyzing data from TMA experiments places a strain on computational and storage resources. From each TMA block we can process slides with different techniques and handle a large number of high resolution images. Moreover, considering that a TMA experiment can include hundreds of tissue samples it is clear the importance of an infrastructure not only to assure patient data privacy, but also as a resource for storage and computing to process images. Grid platforms could solve these problems by providing storage facilities and high computational power under a solid security architecture.

At present, pathology institutes store images and associated metadata - clinical situation, sample treatment information, paraffin block creation and slide reaction type - relating them to patients through anonymous identifiers. This is a good method for a local storage, but is not safe for sharing data with the scientific community. The Grid middleware enables the development of a secure method of storage and analysis in order to promote TMA images and metadata sharing.

In this paper we present a new approach for processing remote TMA images using the Enabling Grids for E-sciencE (EGEE) Grid infrastructure. The image analysis is performed using the Parallel IMAGE processing GEnoa Library (PIMA(GE)^2 Lib). The

privacy of the sensible data is ensured through the use of specialized I/O functions based on the Grid File Access Library (GFAL) API, that is provided by the EGEE framework.

1. The Grid platform

In this work we considered the use of the Enabling Grids for E-sciencE (EGEE) Grid infrastructure, a wide area platform for scientific applications that relies on the Globus Toolkit, as middleware, which operates as an ideal communication layer between the different Grid components. Let us provide a short overview of EGEE and its way of managing data.

1.1. The EGEE distributed environment

This Grid infrastructure is a network of several Computing Elements (CE), that are gateways for the Worker Nodes (WN) on which jobs are performed, and a number of Storage Elements (SE) on which the data are stored. Through the User Interface (UI) it is possible to submit jobs, monitor their status and retrieve the outputs if jobs are terminated successfully or resubmit them in case of failure.

Due to the use of remote computational resources, the Grid communication software must offer an efficient security system. Security is guaranteed by Grid Security Infrastructure (GSI) which uses public key cryptography (asymmetric cryptography) to recognize users. GSI certificates are encoded in the X.509 format, a standard established by the Internet Engineering Task Force (IETF) [4], and accompany each job to authenticate the user. Moreover, in order to submit jobs, users must be members of a Virtual Organization (VO), a group of Grid users with similar interests and requirements who are able to work collaboratively and share resources (data, software, etc.), regardless of geographical location. This is another aspect of Grid supervision because users request accounting must be authorized by the VO manager.

The distribution of the computational load is performed by the Resource Broker (RB) delegated for routing each job on an the available CEs. Jobs are submitted from a UI using the Job Description Language (JDL) scripts, which specify the necessary directives to perform the task. Important features specified by JDL are the files to be accessed on the SE, the data that have to be exchanged between the UI and the WN and the software requirements for the job execution.

1.2. The data and metadata handlers

The EGEE middleware provides a set of tools in order to manage data in a remotely distributed way. They provide accessibility to the physical location of the files and to the LCG File Catalog (LFC) system used to associate a physical identifier to the Logical File Name (LFN). The LFC contains a GUID (Globally Unique Identifier) as an identifier for a physical file and combines logical and physical mappings for the file in the same

database. To upload a file on an SE users must specify the hostname of the Grid node where they want to store images: once the file has been uploaded the server gives back an id that univocally corresponds to the file. At the same time the file is registered in the LFC in order to be accessible through its LFN. Both, the GUID and the correspondent LFN can be used to access data. Uploaded files are visible from everyone belonging to the same VO, even if the owner can manage permissions, avoiding the access by non-authorized users.

A very important feature is that in the EGEE Grid it is possible to handle the access to files archived on the SEs by performing computations without copying data on the WNs. This is possible using GFAL API, a library available with interface for different programming languages (C, Java, etc.) that allows to access files which are located in remote SEs, anywhere in the Grid. In this way we avoid to have multiple physical copies of raw data without compromising the security level of sensitive data.

Presently we disregard all the metadata information. We plan to manage the metadata using the AMGA system [5]. It is based on a client/server architecture, and it can be easily integrated into the Grid environment. The metadata are recorded in an AMGA table in relation to the file's GUID. In the same way as the other Grid components, the AMGA catalogue preserves data security because it supports VO authentication.

2. The use of PIMA(GE)^2 Lib

We implemented the image processing algorithms using the functions of PIMA(GE)^2 Lib [6] library functions. This library has been developed in order to satisfy the scientist's requirements in the image processing field, ensuring robust and high performance execution. It implements the most common image processing operations, according to the classification provided in Image Algebra [7].

The operations provided by the PIMA(GE)^2 Lib have been grouped in 'conceptual objects', that are not intended to prescribe how an operation is performed but to underline the operation similarity and to help in the definition of an effective and efficient interface. Furthermore, even from the user's point of view, the conceptual objects allow an easier management of the library operations, since the user is no longer involved with a large number of functions, but he/she has to consider and handle a small set of well-defined objects. Operations are grouped together according to different rules, such as the nature similarity, or the data structures processed: they are collected together following 'algorithmic pattern'. The single image processing operation can be called by instantiating algorithmic pattern with the proper parameters, including the function to be applied to the data elements. Following this approach the core of PIMA(GE)^2 Lib is represented by a set of eight objects:

- I/O Operations: operations that perform the I/O and memory management;
- Point Operations: operations that elaborate one image applying a unary function, i.e. square root, absolute value, sine;
- Image Arithmetic: operations that elaborate two image, i.e. addition, product, etc;

- Reduce Operations: operations that elaborate one image calculating a collective value, i.e. the maximum, minimum pixel value of an image;
- Geometric Operations: operations that elaborate one image applying a Region of Interest (ROI) or a domain function, i.e. rotation, translation and scaling;
- Neighbourhood Operations: operations that elaborate one image applying a convolution function that involves a neighbourhood of each pixel: percentile, median;
- Morphology Operations: operations that elaborate one image applying a convolution function that combines together two arithmetic functions: Gaussian convolution, erosion, dilation;
- Differential Operations: operations that elaborate one image and perform differential operators, i.e. Hessian, gradient, Laplacian.

The PIMA(GE)^2 Lib operations can be performed both in a sequential and in data parallel fashion. Till now we considered only a sequential use of the library operations, however we plan to exploit the EGEE resources for parallel executions, like MPI, when considering more time consuming implementations.

In this work, we added specialized functions to the object of I/O operations in order to ensure the privacy preservation. It has been shown in the literature [8] that GFAL API for the Grid platform provides a secure system for image manipulation. We moved the library software to the Grid environment acquiring properties of distribution, extensibility and dynamicity. Once data have been accessed they can be processed as usual by PIMA(GE)^2 Lib operations.

3. Testing our approach

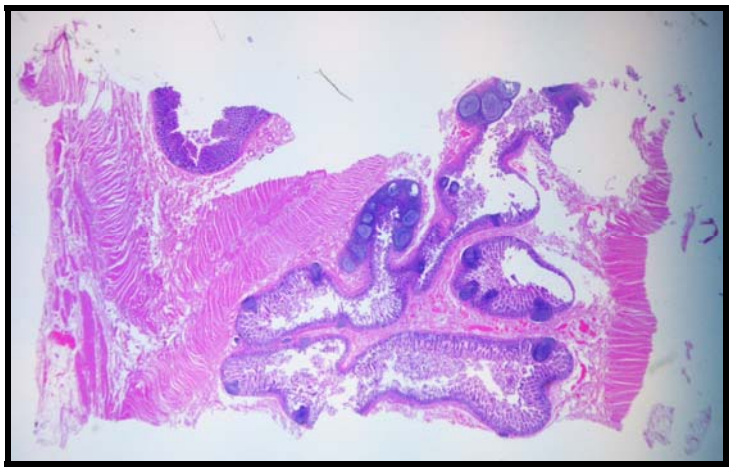
We used a modified version of PIMA(GE)^2 Lib to perform algorithms of edges detection and segmentation of images stored on Grid SE. The implemented code uses GFAL API to open images in tiff and bmp format: in fact a standard format for saving TMA images does not exist at the moment, but these two packaging methods ensure that the image are stored in a lossless form. Both image management and analysis submission are possible from the UI. The JDL scripts describe how the job has to be computed on the remote WN, copying the image to its memory buffer, processing data using the selected algorithm and retrieving the output on the UI.

As a test case we stored 10 images in a set of SEs. The algorithm we implemented accesses the data distributed on the Grid, acquires one image at a time using the new I/O functions, processes it on a remote WN, then produces the resulting image, a masked image from which it is not possible to reconstruct original data, to ensure patient privacy, and which is more informative than the raw images concerning pathology.

An example of our analysis is shown in Figure 1. Figure 1.A is the input image and it represents an haematoxylin-eosin reaction: this coloration differentiates basic and acid entities in the tissue slice. Basic elements (like cytoplasm) are highlighted in a red shade

while acid elements (like nuclei) are in blue. Figure 1.B represents the result of a first attempt of the edge detection algorithm.

A.



B.

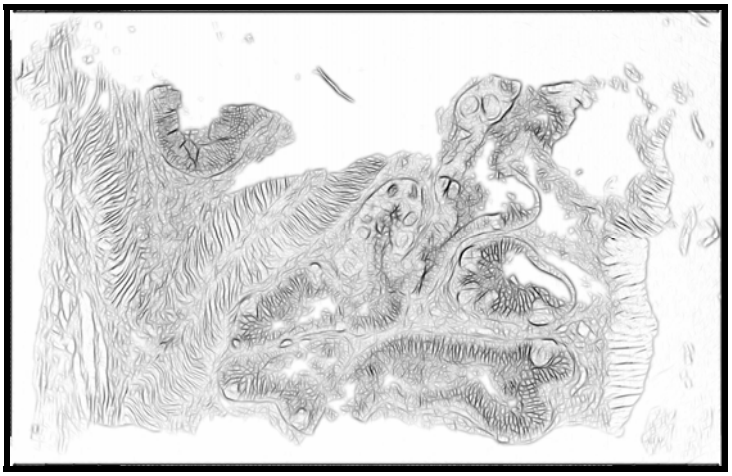


Figure 1. Input (A) and output (B) of remote image analysis using the edge detection algorithm

We are also developing functions which work on thresholds and segmentations. PIMA(GE)^2 Lib provide interesting operations to help detection of cellular structures and tissue architectures, i.e. Morphology and Differential objects.

4. Conclusions

In this paper we presented an approach for processing remote TMA images in the EGEE environment. We enabled the use of PIMA(GE)^2 Lib in the EGEE Grid by implementing a set of specialized functions in order to access distributed medical images without moving them and preserving the privacy of sensible data. In particular we implemented an edge extraction algorithm.

Up to now images have not been associated with any supplementary information and analyses have been performed on manually chosen images. Without the associated metadata images become meaningless. Users should have the possibility of choosing images to work with according to their associated information. The community of pathologists (API, Association of Pathology Informatics) [9] developed an XML language to facilitate TMA data exchange [10] and we are creating a metadata classification compliant with this acquired specification.

Positive test results about the feasibility of our work brings us to consider future developments including features to automatic data access using metadata and the implementation of more informative image elaboration algorithms.

Acknowledgments

Thanks are due to the Italian FIRB project LITBIO (Laboratory for Interdisciplinary Technologies in BIOinformatics), (http://www.litbio.org).
Thanks also to EU BioinfoGRID no.:026808 project.
A special thanks to BioLab Laboratory of the Department of Computer Science, Control Systems and Telecommunications of University of Genoa.

References

[1] S. Mousses, L. Bubendorf, U. Wagner, G. Hostetter, J. Kononen, R. Cornelison, N. Goldberger, A.G. Elkahloun, N. Willi, P. Koivisto, W. Ferhle, M. Raffeld, G. Sauter, O.P. Kallioniemi, Clinical validation of candidate genes associated with prostate cancer progression in the CWR22 model system using tissue microarrays, *Cancer Research* **62** (2002), 1256-1260.

[2] M. Schena, D. Shalon, R.W. Davis, P.O. Brown, Quantitative monitoring of gene expression patterns with a complementary DNA microarray, *Science* **270(5235)** (1995), 467-470.

[3] J. Kononen, L. Bubendorf, A. Kallioniemi, M. Barlund, P. Schraml, S. Leighton, J. Torhorst, M.J. Mihatsch, G. Sauter, O.P. Kallioniemi, Tissue microarrays for high-hroughput molecular pro_ling of tumor Specimens, Nature Medicine **4** (1998), 844-847.

[4] http://www.globus.org/security/overview.html

[5] N. Santos, B. Koblitz, Metadata services on the grid, *Proc. of Advanced Computing and Analysis Techniques*, ACAT'05, Zeuthen, Berlin, May 2005.
[6] A. Clematis, D. D'Agostino, A. Galizia, The Parallel IMAGE GEnoa processing Library: PIMA(GE)^2 Lib, *Technical Report*, IMATI-CNR-Ge, N.22/2006.
[7] G. Ritter, J.Wilson, Handbook of Computer Vision Algorithms in Image Algebra, *CRC Press*, 2nd edition, Inc, 2001.
[8] S. Bagnasco, F. Beltrame, B. Canesi, I. Castiglioni, P. Cerello, S.C. Cheran, M.C. Gilardi, E. Lopez Torres, E. Molinari, A. Schenone, L. Torterolo, Early Diagnosis of Alzheimer's Disease Using a Grid Implementation of Statistical Parametric Mapping Analysis, *HealthGRID 2006*, Valencia, June 2006.
[9] http://www.pathologyinformatics.org/
[10] J.J. Berman, M.E. Edgerton, B.A. Friedman, The tissue microarray data exchange specification: A community-based, open source tool for sharing tissue microarray data, *BMC Medical Informatics and Decision Making* **3(5)** (2003).