DOI 10.3233/ACP-CLO-2010-0551
IOS Press

Ultra-fast processing of gigapixel Tissue MicroArray images using high performance computing

Yinhai Wang [a,b], David McCleary [c], Ching-Wei Wang [d], Paul Kelly [e], Jackie James [b], Dean A. Fennell [a,b] and Peter Hamilton [a,*]

[a] *Centre for Biomedical Informatics, Queen's University Belfast, Belfast, UK*
[b] *Centre for Cancer Research and Cell Biology, Queen's University Belfast, Belfast, UK*
[c] *i-Path Diagnostics Ltd., Belfast, UK*
[d] *Graduate Institute of Biomedical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan*
[e] *Department of Pathology, Royal Group of Hospitals, Belfast, UK*

Abstract. *Background*: Tissue MicroArrays (TMAs) are a valuable platform for tissue based translational research and the discovery of tissue biomarkers. The digitised TMA slides or TMA Virtual Slides, are ultra-large digital images, and can contain several hundred samples. The processing of such slides is time-consuming, bottlenecking a potentially high throughput platform.

Methods: A High Performance Computing (HPC) platform for the rapid analysis of TMA virtual slides is presented in this study. Using an HP high performance cluster and a centralised dynamic load balancing approach, the simultaneous analysis of multiple tissue-cores were established. This was evaluated on Non-Small Cell Lung Cancer TMAs for complex analysis of tissue pattern and immunohistochemical positivity.

Results: The automated processing of a single TMA virtual slide containing 230 patient samples can be significantly speeded up by a factor of *circa* 22, bringing the analysis time to one minute. Over 90 TMAs could also be analysed simultaneously, speeding up multiplex biomarker experiments enormously.

Conclusions: The methodologies developed in this paper provide for the first time a genuine high throughput analysis platform for TMA biomarker discovery that will significantly enhance the reliability and speed for biomarker research. This will have widespread implications in translational tissue based research.

Keywords: Cluster, dynamic load balancing, high performance computing, parallel processing, Tissue MicroArray, TMA, virtual slide

1. Introduction

Tissue MicroArrays have become a very important tool in the evaluation and discovery of tissue biomarkers that are clinically relevant and support diagnostic classification, prognosis, or in defining sensitivity or resistance to patient targeted therapies [13]. Having up to several hundred tissue samples on a single glass slide reduces to a single assay what would otherwise be an expensive, time consuming and technically variable experiment. Techniques such as immunohistochemistry (IHC) or fluorescence *in situ* hybridisation (FISH) can obtain a simultaneous view of protein or nucleotide sequence expression across a wide cohort of patients with different clinical outcomes. For this reason, the approach has been termed "high throughput". While this is true, in that it is a single assay platform employed for multiple samples, the subsequent analysis of biomarker expression on TMAs is still based on visual inspection and scoring by a trained pathologist. Whilst critical for successful biomarker analysis using TMAs it represents a significant bottleneck in many studies. In addition to the time it takes to manually score hundreds of tissue cores, there are also issues associated with inter- and intra-observer reproducibility

[*] Corresponding author: Peter Hamilton, G64 Health Science Building, Queen's University Belfast, 97 Lisburn Road, Belfast, BT9 7BL, UK. Tel.: +44 28 9097 2802; Fax: +44 28 9097 2776; E-mail: p.hamilton@qub.ac.uk.

of scoring due to the subjectivity of visual interpretation by the naked eye. It is for these reasons that computerised image analysis has once again come to the fore, as a means of supplementing biomarker evaluation by pathologists using TMAs.

Until recently image analysis of TMAs would have been impracticable, since recording separate digital images of each individual core using a standard camera would have been enormously time consuming. However, the advent of virtual microscopy and high resolution scans of entire glass slides has allowed an entire TMA slide to be scanned in a few minutes, completely capturing the biomarker densitometric and location information in the form of a single digital image. This provides an ideal platform to explore the use of computer-vision algorithms [5,8,18] for the automated analysis of tissue biomarkers within TMAs and the opportunity to develop a truly high throughput platform for biomarker discovery in tissues.

A number of commercial systems are currently available which provide computer-based analysis of TMAs using generic algorithms for nuclear/cytoplasmic segmentation and quantitation of immunohistochemistry. One of the major technical challenges in using virtual slides is the size of the images generated. Scanning a typical region of 25 mm × 15 mm occupied by TMA tissue samples on a glass slide at 40× magnification can result in an image with 100,000 × 60,000 pixels [1], corresponding to 20 GB of uncompressed data. At this resolution, an individual tissue core of approximately 0.6 mm in diameter would be approximately 9 mega-pixels. Analysing tissue structure and biomarker density in images of this size on multiple cores is computationally intensive and time consuming. However, by analysing multiple cores simultaneously, using high performance computing (HPC) one could theoretically significantly speed up biomarker quantitation and TMA analysis. The discrete nature of a TMA and its component tissue samples lends itself perfectly to independent and highly parallelised analysis.

Others have considered this in the context of Grid-based computing [6,17,19] which is a highly distributed form of computing using a decentralised model. Whilst providing certain speed advantages, Grid-based computing can be difficult to control, manage and configure for dedicated experiments [14]. This arises from the fact that it tends to incorporate heterogeneous collections of computers, with widely different capabilities, managed by different organisations, widely distributed geographically, with inconsistent connections and bandwidth. In this study we have explored an alternative approach using a dedicated high performance computer cluster specifically designed for the high performance analysis of TMAs. The benefits of cluster-based computing are that the computer architecture is specifically designed to manage parallel processing with consistency across processors in the cluster and fast connections among nodes. These benefits promised to provide a convenient and highly rapid approach to automated TMA analysis and this was tested using a number of algorithms on TMAs with novel biomarkers in lung cancer.

2. Materials and methods

2.1. High performance computing (HPC)

This study utilised the HPC Centre at the Queen's University of Belfast which currently houses a Hewlett-Packard (HP) BladeSystem c7000 enclosure with multiple blade servers. Each blade consists of 2 Intel Xeon E5420 quad-core processors at 2.5 GHz and 10–16 GB of shared memory. There are altogether >9000 processor cores and 18 TB (terabytes) of shared memory available to use. The blades use gigabit Ethernet interconnections and a fibre channel (FC) storage area network (SAN) connection to hard drives. Currently, the total size of hard drives is 250 GB. This cluster system also uses a 64 bit Microsoft Windows HPC Server 2008 operating system and a Linux server.

2.2. System architecture

The cluster based HPC platform was developed for (i) the rapid analysis of TMA virtual slides, and (ii) the management of virtual slides. The schematic overview of the overall system architecture is presented in Fig. 1. It hosts five functional modules, namely a *Parallel Processing* module, an *Image File Access* module, an *Analytic* module, a *Digital Slide Serving* module and a *Digital Slide Viewing* module. These are detailed below.

2.2.1. Parallel Processing module

A *Parallel Processing* module was developed to allow the simultaneous analysis of multiple tissue cores on the HP BladeSystem cluster. A centralised dynamic load balancing parallel strategy was developed. This *Parallel Processing* module was programmed in C/C++ language and a Microsoft implementation of the Message Passing Interface (MPI) which is based on MPICH2 (Argonne National Laboratory).

Load balancing refers to the technique to distribute workload evenly across a set of processing units/cores.

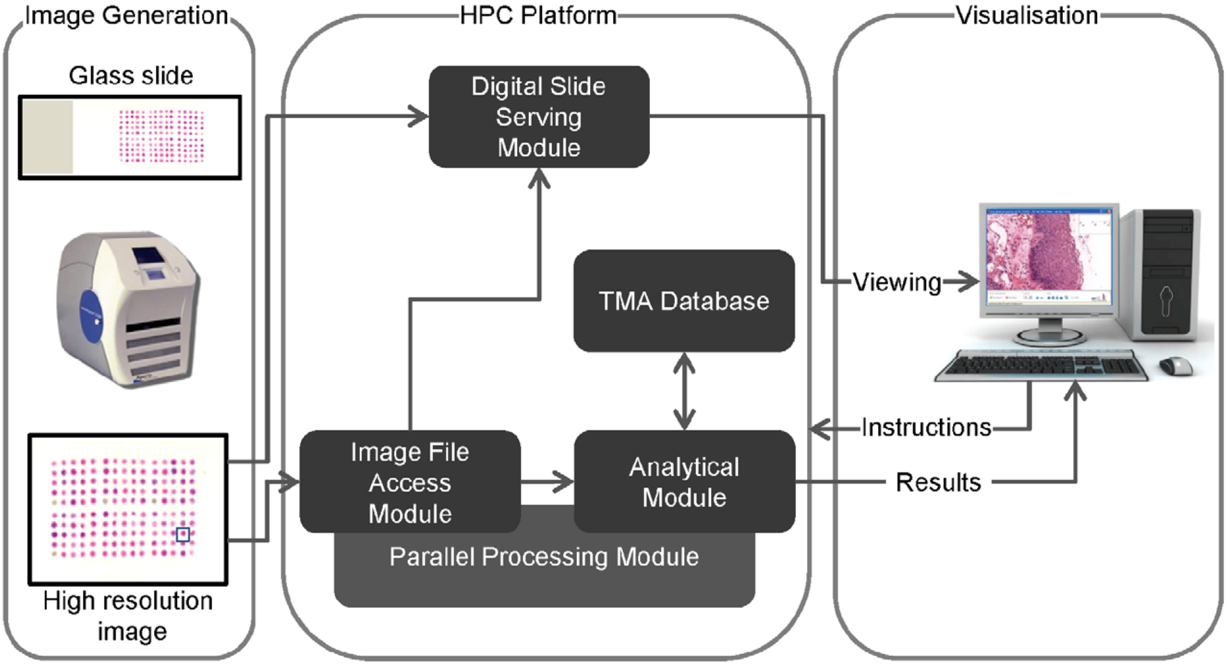


Fig. 1. System architecture of the HPC Platform for TMA Analysis. Image Generation (left) utilises scanning technology to generate high resolution images of the glass TMA slide. This is utilised within the HPC Platform (centre) which comprises a number of interacting functional modules. Visualisation (right) of imagery and data generated from HPC analysis can be achieved remotely used web-based technology based on PathXL platform (i-Path Diagnostics Ltd.). (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

Two approaches to load balancing were investigated to determine relative efficiency in TMA analysis. (i) Static load balancing: assuming there are p processor-cores, a static load balancing approach assigns each processor-core every pth TMA core in a round-robin fashion. Each processor core is responsible for accessing the hard drives for loading and saving TMA images. (ii) Centralised dynamic load balancing (Fig. 2): here a dedicated master processor-core is configured to dispatch image processing tasks to worker processor-cores (P0 to Pp), where worker processor-cores perform the analytical tasks. A new TMA-core processing task is only assigned when a worker processor is idle and requesting tasks from the master processor-core. In dynamic load balancing, only worker processor cores access hard drives for loading and saving, whereas the master processor-core is only responsible for managing work load amongst worker cores.

2.2.2. *Image File Access module*

Virtual slides produced from different scanners use variety of compression techniques and different file formats [15]. These virtual slides are often not interoperable. A *File Access* module was adapted from the *PathXL* framework (i-Path Diagnostics Ltd.) so that virtual slides can be handled regardless of file formats and compression used. Currently, the *File Access* module supports virtual slides scanned using Aperio ScanScope series scanners, Hamamatsu NanoZoomer scanners and Carl Zeiss Mirax Scanners.

The *File Access* module is able to load virtual slides into vendor-format-independent image data for viewing and further processing and output regions of virtual slides in standard JPEG files. The proposed *Image File Access* module unifies pixel format into a vendor independent sequence of red–green–blue pixels (Fig. 3), which is essential for the design of virtual slide viewing and analysis functionalities (introduced in Sections 2.2.3–2.2.5). When it is required to output certain regions of a virtual slide (e.g., a TMA core), the output is saved as a *.jpg* file using the JPEG compression algorithm from Intel JPEG Library (version [2.0.18.50]).

2.2.3. *Analytic module*

The *Analytic* module was designed to accommodate any algorithm, or set of algorithms for the analysis of TMA cores. Two sets of analytic algorithms were implemented and evaluated in the current study: (i) tissue core texture pattern measurements used for histological sub-typing and (ii) automated quantitation of biomarker IHC density on TMA core images.

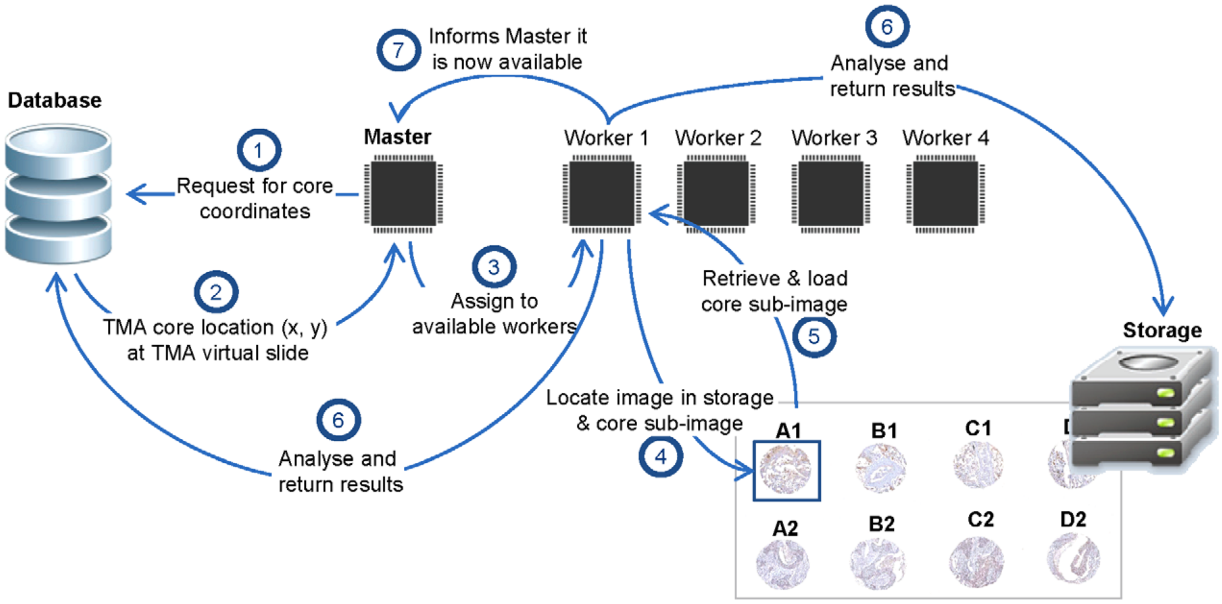

Database
1 Request for core coordinates
Master
2 TMA core location (x, y) at TMA virtual slide
3 Assign to available workers
7 Informs Master it is now available
Worker 1
Worker 2
Worker 3
Worker 4
6 Analyse and return results
Retrieve & load core sub-image
5
Locate image in storage & core sub-image
4
6 Analyse and return results
Storage
A1
B1
C1
A2
B2
C2
D2

Fig. 2. Diagram illustrating the centralised dynamic load balancing approach for parallelise image processing tasks. The steps are numbered to illustrate the workflow. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

Texture features are widely used in tissue imaging for supervised and unsupervised learning [7,20] across a variety of applications, both for tissue pattern recognition and immuno-quantitation [9]. Tissue texture computation allows for the recognition of pattern changes associated with malignancy and facilitates the automatic identification and segmentation of tumour regions in tissue samples and TMAs [4,20]. For high throughput biomarker evaluation using HPC this is an essential approach for selecting tumour regions within which biomarker IHC can be measured. In this study, six popular statistical moments based texture features were implemented in the *Core Analytic* module and used to evaluate the performance of the HPC platform. Their mathematical formulae are listed in Table 1. These were combined to form a classifier for 100 × 100 pixel tiles, allowing the identification of tumour regions in lung TMA samples.

An automated IHC quantification method was developed specifically for lung cancer TMA analysis (see Section 2.4) and integrated seamlessly into the HPC platform. This allows objective, rapid and continuous assessment of biomarker expression and quantitative analysis. The algorithm has a number of functions:

(a) The removal of carbon particle objects from each TMA core image using static gray-level thresholding at the value of 40.

(b) The separation of DAB brown colour channel using the exact colour deconvolution method proposed by study [16].

(c) The subsequent quantification of IHC (DAB – brown) staining using a dynamic Otsu's method [12]. It is used for the determination of an optimum threshold for each single TMA core based on histogram distribution.

Using dynamic load balancing only, biomarker image analysis was carried out on all three virtual TMA slides. Following image load and decompression, the *Analytic* module was called and performed on each TMA core, generating a quantitative score. This score was then used to generate a corresponding mark-up image showing positively stained regions superimposed in red. These mark-up images were eventually saved as *.jpg* files using *compression quality* of 100.

Similar to the *Image File Access* module, the *Analytic* module is also integrated with the *Parallel Processing* module for acceleration. The *Analytic* module sits on the head node of the HPC platform. Depending on the choice of either static or centralised dynamic load balancing approach from the *Parallel Processing* module, algorithms implemented in the *Analytic* module are dispatched to all allocated processor-cores and processed in parallel until all tasks are complete.

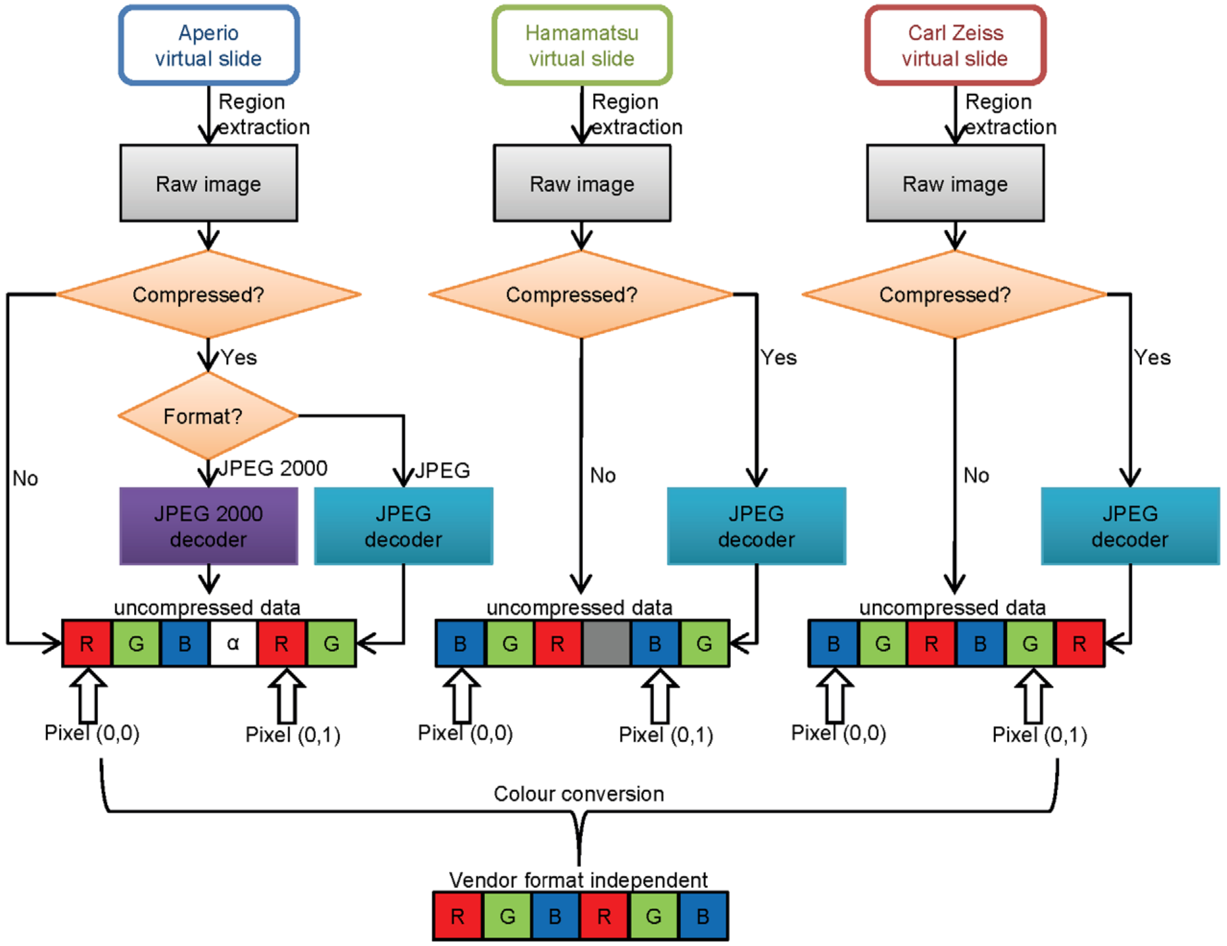


Fig. 3. Flowchart of how the File Access module loads virtual slides. Currently Virtual slides produced using Aperio, Hamamatsu and Carl Zeiss scanners are supported. Regions of these virtual slides are initially loaded into memory and subsequently decompressed using their corresponding decompression method. Finally, these uncompressed data are converted into vendor format independent RGB format. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

Table 1

Equations for the 6 texture features to be calculated on each TMA core

Texture feature	Expression
Average intensity	$m = \sum_{i=0}^{L-1} z_i p(z_i)$
Average contrast	$\sigma = \sqrt{\mu_2(z)}$
Smoothness	$R = 1 - 1/(1 + \sigma^2)$
3rd moment	$\mu_3 = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i)$
Uniformity	$U = \sum_{i=0}^{L-1} p^2(z_i)$
Entropy	$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)$

Notes: z_i is a random variable indicating intensity, $p(z)$ is the histogram of the intensity level in a region, L is the number of possible intensity levels.

2.2.4. *Digital Slide Serving module*

For TMA virtual slides to be viewed remotely, a *Digital Slide Serving* module developed on the *PathXL Server* (i-Path Diagnostics Ltd.) was utilised. This transfers image data to an on-line viewer using a region-on-demand process and utilising the decoding capabilities of the *Image File Access* module which is image format and vendor independent. Images are served as standard JPEG image over standard TCP/IP protocols and viewed by end-users via a standard browser.

2.2.5. *Digital Slide Viewing module*

End users are able to view TMA virtual slide through a unified client web interface, *PathXL Client* (i-Path Diagnostics Ltd.). It works directly with the

Digital Slide Serving module by requesting a spatial region of the virtual slide at a specific magnification, followed by placing returned images in the appropriate position on screen. It is platform-independent and operates within all standard web-browsers. In addition, end users are able to request the HPC platform to perform certain instructions including analysis of TMA slides which are implemented by the *Core Analytic* module. *PathXL* currently communicates with the *Core Analytic* module through the Microsoft HPC Job Manager (Microsoft Inc.). Finally, end users are able to access processing results for the last step either through standard output files from the *HPC Job Manager* or direct remote hard drive access.

2.3. TMA core database

A comprehensive TMA database (TMAX) was built to allow the storage of extensive metadata associated with TMA experiments [2,10], including patient, diagnostic, pathological and clinical information, virtual TMA slide identifier, tissue core location, image analysis results, classification data, etc. TMAX is also designed to support data exchange standards across platforms by generating XML based metadata [2]. The TMAX database resides on the networked hard drives on the HP BladeSystem.

Using a TMA de-arraying algorithm, key information on the layout of the TMA is defined and stored in the database and must be retrieved in order to process a given virtual TMA on the HPC platform. The *Tissue* table contains virtual slide file name (*FileName*) and absolute path on server (*FileDir*). The *Core* table stores location information for each of the TMA cores, including the X and Y coordinates (in pixels) of the bounding box TMA core's top left corner, as well as the *Width* and *Height* (in pixels) for each core.

2.4. Lung TMA samples

Having constructed the HPC approach to TMA analysis, the system was subsequently evaluated on a series of lung TMAs as part of on-going non-small cell lung cancer research programme within the Centre for Cancer Research & Cell Biology at QUB. Lung specimens were taken from 116 patients. These specimens were paraffin-fixed and subsequently sampled to construct 3 TMA blocks, namely TMA1, TMA2 and TMA3. For each whole-tissue block, 3–4 cores were taken and subsequently placed horizontally adjacent within a same TMA block. For TMA1, 3 additional controls from the same block were also taken and placed in a same TMA block. TMA1 and TMA2 also included a number of other types of controls. TMA3 does not have any controls. All TMA cores were sampled at the diameter of 0.6 mm. These 3 TMA blocks contained 490 tissue cores and used to generate a 5 µm tissue sections. Using IHC (DAB), two of the sections were stained for the BCL-2 family proteins NOXA (Q13794) and BAK (Q16611) [3,11].

Each TMA slide was scanned using an Aperio ScanScope CS scanner (Aperio Technologies Inc., San Diego, CA, USA) with the objective of 20×/0.75 Plan Apo, which gives the magnification of 40× and the resolution of 0.25 µm/pixel. After scanning, TMA virtual slides were compressed using the standard libjpeg library (Independent JPEG Group – www.ijg.org) for lossy compression. There are currently no studies which have looked at the impact of compression on measurements in tissue pathology and this is something that our group is exploring in detail – both across compression types and compression quality levels. Previous studies on quantitative IHC, appear to have used a variety of compression quality levels (although often not stated). For this reason we wanted to include different compression quality levels to determine impact if any on processing speed and used compression values of 70 and 30 in this study. Further work will be required to determine if compression has a negative impact on accurate measurement in Digital Pathology. Details of the three virtual slides are listed in Table 2.

2.5. Evaluation of performance

Processing time and *Speedup* were used as performance measurements. *Processing time* was measured by the number of seconds to perform an operation. *Speedup* is defined as the ratio of fastest sequential execution time and parallel execution time:

$$\mathit{Speedup} = \frac{\mathit{Sequential\ Execution\ Time}}{\mathit{Parallel\ Execution\ Time}}. \tag{1}$$

Given an image analysis algorithm and a specific TMA virtual slide, the sequential processing time running on one processor core was firstly recorded. Following this, processing time using multiple processor-cores was calculated, compared with sequential processing time and *Speedup* calculations made. If static load balancing approach is used, one processor-core is initially allocated, whereas for centralised dynamic

Table 2
TMA virtual slides details

	TMA1	TMA2	TMA3
IHC marker	NOXA	BAK	NOXA
Number of patients	31	37	48
Number of TMA cores per sample	4	3	3
Number of tissue cores	232	114	144
Segmented number of tissue cores	229	114	144
Number of controls	124	9	0
Virtual slide dimension (pixels)	160,290 × 65,017	61,505 × 54,619	80,436 × 51,100
Uncompressed image size (giga-bytes)	19.3	9.4	11.5
IJG-JPEG compression quality	30	70	30
IJG-JPEG compression ratio	38.65	24.71	35.32
File size (mega-bytes)	512	389	333

load balancing approach, two processor-cores are used so that one processor-core could act as the master node and the other one be the worker node.

For individual experiments, the impact of HPC was evaluated on three processes: *Image Loading, Image Analysis* and *Image Saving*:

- *Image Loading*: Loading the entire TMA virtual slide and its component TMA cores at original resolution into the system memory followed by JPEG decompression.
- *Image Analysis*: Calculation of all algorithmic derived features from every TMA core on a virtual slide.
- *Image Saving*: Compressing every TMA core image from a virtual slide using JPEG compression and saving them to the HPC's hard disk.

Finally, the impact of image compression on parallel processing of TMA samples was examined. All TMA core images were JPEG-compressed using 411 sampling at the *compression quality* of 100 (maximum) and 0 (minimum) and the impact of HPC induced processing speed analysed.

3. Results

For the evaluation of the performance of the HPC platform, all three virtual TMA slides from the lung TMA dataset were tested. These virtual slides contain altogether 490 TMA cores (with 232, 114 and 114 cores, respectively). During TMA de-arraying process, 3 TMA cores were not successfully segmented which gave the total of 487 segmented TMA cores. These three TMA virtual slides were tested independently.

3.1. Loading and storing digital slides on HPC platform

The processing time for the *Image Loading* and *Image Saving* of virtual slides is significantly reduced when using multiple processor cores. Figure 4A–C shows that the processing time to load a TMA virtual slide is rapidly reduced to less than 4 s by comparison to as much as 80 s using standard sequential code with a maximum *Speedup* of 21.60. However, the benefit of assigning more processor-cores peaks between 10 and 25, where speed improvement stabilises. *Saving* images (Fig. 4D–F) is computationally more intensive than *Loading*, with sequential code processing taking between 150 and 286 s, depending on the size of the TMA image and number of tissue cores. Again, *Image Saving* significantly benefits from parallelisation and increasing the number of processor-cores, reduced time to <10 s and as much as 58.17 times faster than sequential code.

3.2. Static load balancing vs. centralised dynamic load balancing

A comparison of static and centralised dynamic load balancing showed little difference in processing time, as shown in the example using texture feature calculation on TMA1 (Fig. 5). Centralised dynamic load balancing is slightly faster when more than 10 processor-cores were used. As results shown in Table 3, if the Centralised Dynamic Load Balancing is used, it could save $>4\%$ of processing time in *Loading*, and $>16\%$ of processing time in *Saving*. However in situations when the number of processor-cores are limited (<10 processor-cores per TMA slide), static load balancing is a better choice as no processor-cores are sacrificed for scheduling tasks as what happens using dynamic load balancing.

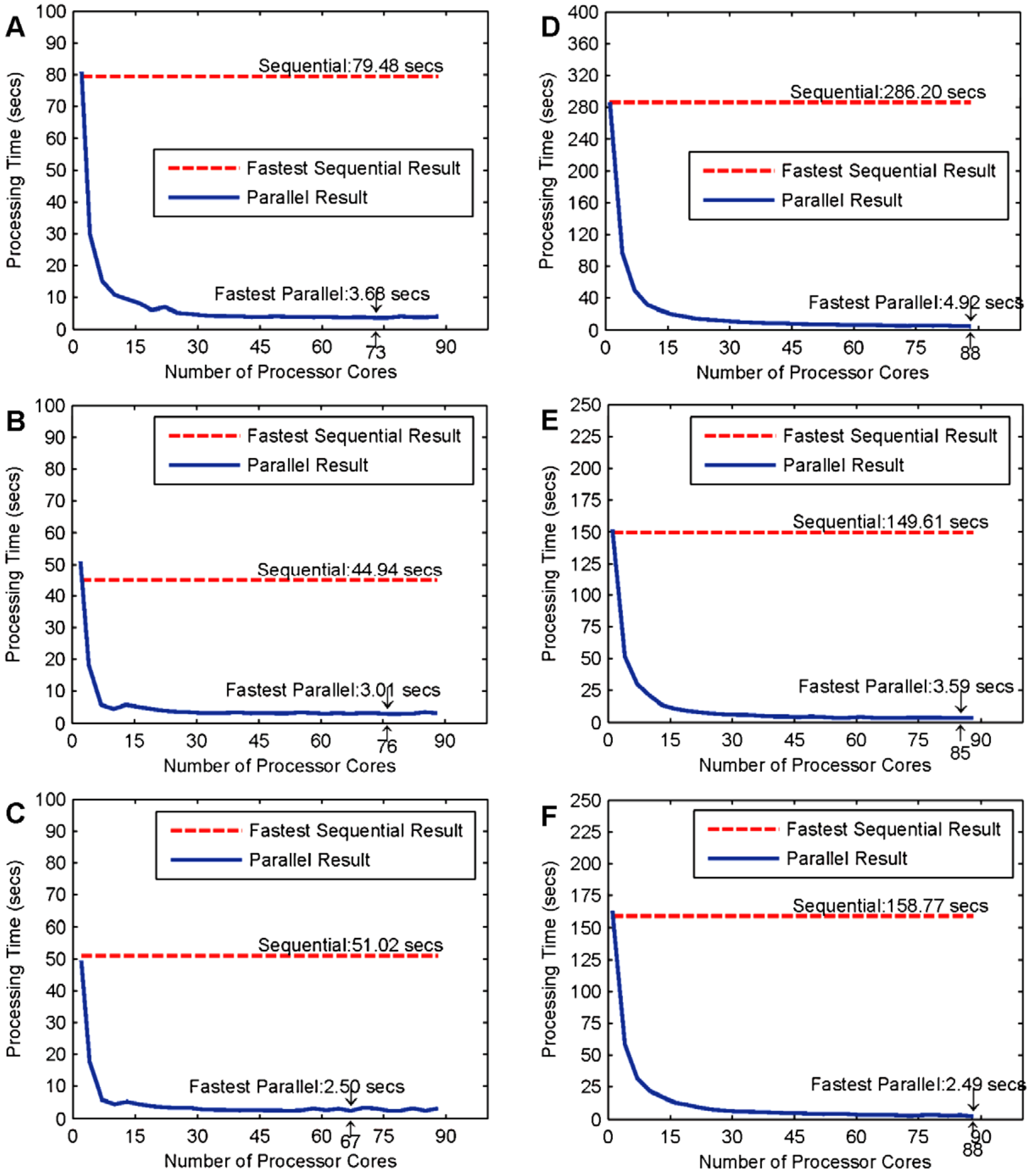


Fig. 4. Results for the time in seconds taken to load (A–C) and save (D–F) virtual slides. Time is plotted against the number of processor cores used. The dashed line shows the corresponding time taken using the fastest sequential code. Loading times for (A) TMA1, (B) TMA2 and (C) TMA 3 are plotted together with saving time for (D) TMA1, (E) TMA2 and (F) TMA 3. It is evident that increasing the number processing cores significantly speeds up both loading and saving of digital images. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

3.3. *TMA texture measurements on HPC platform*

Tumour region identification using texture computation is illustrated in Fig. 6. The time taken to calculate texture features across a TMA virtual slide is small compared to the time required with *Loading* and *Saving* a whole virtual slide and its component TMA cores. The average time for sequential code on

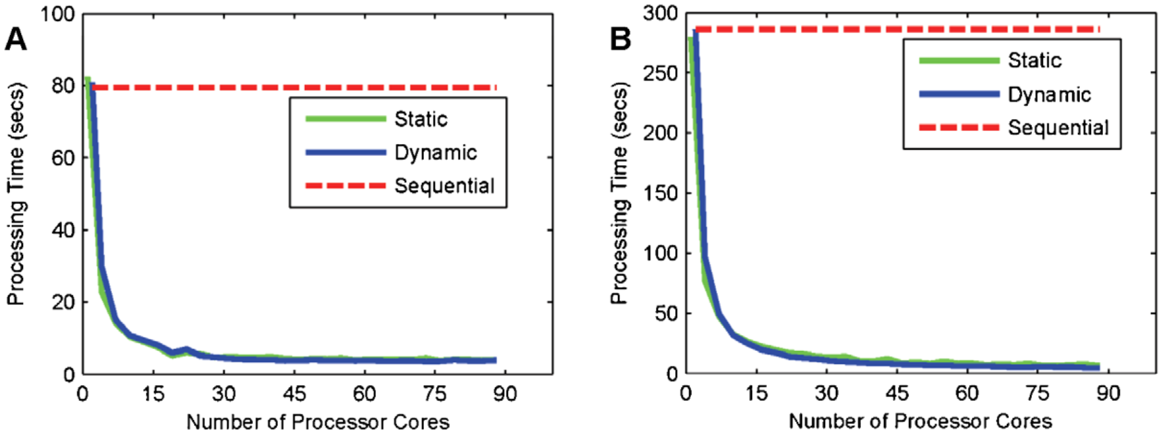


Fig. 5. Comparison of processing time between static load balancing and centralised dynamic load balancing. This comparison was using the calculation of texture features on TMA1 as an example. (A) Loading time, (B) Saving time. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

Table 3

Percent of processing time saved using centralised dynamic load balancing over static load balancing

		Centralised dynamic load balancing (s)	Static load balancing (s)	% of time saved using centralised dynamic load balancing (%)
TMA1	Loading	4.85	5.07	4.36
	Saving	10.21	12.42	17.85
TMA2	Loading	3.559	4.20	15.65
	Saving	6.08	7.71	21.10
TMA3	Loading	3.18	3.39	6.02
	Saving	6.46	7.72	16.33

1 processor core for the calculation of texture features on the entire TMA virtual slide is about 5 s (Fig. 7). Using 10 or more processor-cores, reduces time taken to <1 s[1] regardless of whether static load balancing or centralised dynamic load balancing approach was applied.

Without the proposed HPC platform, if we take the average of sequential processing time of 261.91 s per slide, to perform texture feature calculation for 90 virtual slides, it would take 23,580 s (6.55 h) traditionally using 1 processor core. However we estimated that when process these virtual slides in parallel over 9000 processor-cores, it will only take the amount of time for the processing of the slowest slide, which is 8.72 s in our experiment (Table 4).

When low (0) and high (100) image compression qualities were applied, experiments showed only slight increase in processing time when analysing high compression qualities. An example using TMA1 is shown in Fig. 8. When 2 processor-cores were used, the iteration for *Saving* a JPEG with compression quality of 100 (363.56 s) is 77.40 s slower than the iteration using a compression quality of 0 (286.15 s). During compression and decompression, both images with low and high compression qualities go through a same encoding/decoding procedure. Therefore only fractional time differences would occur as the result of these numerical calculations. However, the image with high compression quality has a larger amount of data (in bytes) to be accessed from the hard drive, and results in an increased demand on processors. Having a number of processor working in tandem can off-set the processing time required. The differences in processing time between low and high compression quality images de-

[1] Processing time in Fig. 7 may appear to be negative as the result of how it is calculated. In our test, the processing time for texture feature calculation is obtained as the time difference between two separate runs. The first iteration calculated the overall processing time for *Image Loading*, *texture feature calculation* and *Image Saving*, whereas the second iteration only calculated the overall processing time for *Image Loading* and *Image Saving*.

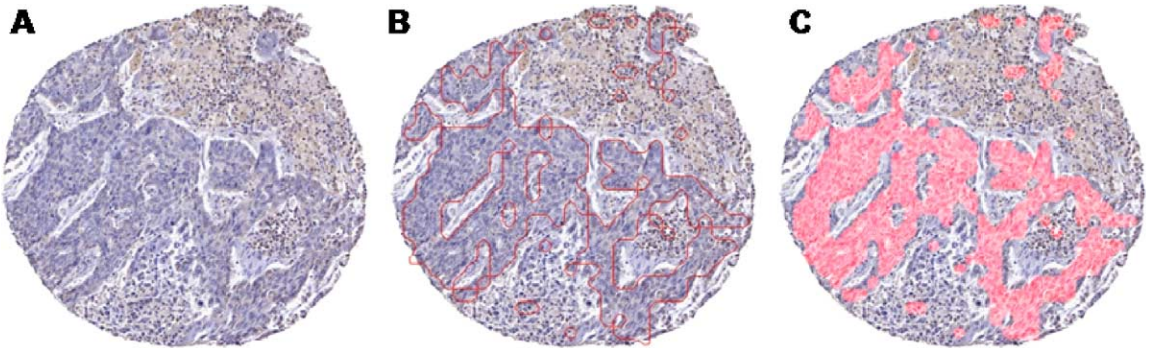


Fig. 6. Examples of tumour region identification using texture feature calculation. Figure A is a TMA core image, Figure B and C marked tumour regions using contour and overlapping pseudo-colour. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

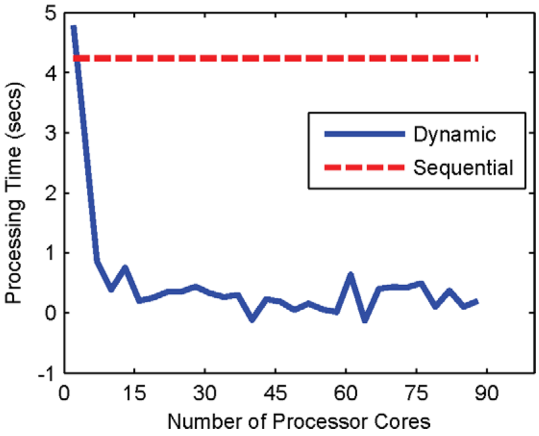


Fig. 7. Runtime for calculating texture features (without loading and saving) for TMA2. The solid line indicates the texture calculation time using centralised dynamic load balancing, whereas the dashed line is the time taken for sequential code using 1 processor core. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

creases rapidly with increasing numbers of processor-core and became negligible when using our HPC platform. This finding suggests with the use of multiple processing cores, JPEG compression quality has little influence in processing time, and in theory higher quality image could be used without impacting on speed of analysis.

3.4. TMA biomarker quantification using HPC platform

Visual inspection of the final mark-up images showed good concordance between algorithm-based segmentation and regions of positive immunoreactions, indicating the success of this algorithmic approach. The measurement of density within these regions allows for the evaluation of biomarker expression and its relationship to clinical outcome. Examples of the biomarker quantification results are shown in Fig. 9.

The speed of generating these results is also significantly improved when using multiple processor-cores. When using only 1 processor core, the runtime for TMA1, TMA2 and TMA3 are respectively 1697.70 s (28 min), 763.45 s (13 min) and 1003.30 s (17 min), whereas with the use of the proposed HPC platform, their runtime was reduced rapidly to be 76.53, 34.44 and 52.82 s (Fig. 10 and Table 5), which gave the maximum *Speedup* of 22.17. When considering the average time for the processing of one TMA core, the HPC platform gave the average processing time of 0.34 s, comparing with the average 7.11 s using only 1 processor-core. Similar to the calculation of texture features, more TMA virtual slides can be processed simultaneously using different set of 100 processor cores. Over 90 slides are able to be processed for biomarker quantification in 76.53 s (which is also the time to process the slowest TMA virtual slide). However it would traditionally take about 103,932 s (28.87 h) using only one processor core.

4. Discussion

TMAs are a key tool in the search for new tissue biomarkers but are hampered by the need to visually score immunohistochemistry results on hundreds, possibly thousands of individual samples. In recent years, there has been a renewed interest in the development of computer-based image analysis for this purpose. Quantitative immunohistochemistry is made easier by the inter-specimen consistency that is achievable in TMAs

Table 4

Statistics for the overall processing time for texture feature calculation including *Loading*, *Texture* and *Saving* from the test of the 3 TMA virtual slides

	TMA1	TMA2	TMA3
Time taken using the fastest sequential code	371.30 s	198.80 s	215.62 s
Shortest parallel processing time	8.72 s	7.01 s	5.73 s
Number of processor-cores used	88	64	88
Maximum *Speedup*	42.58	28.36	37.63

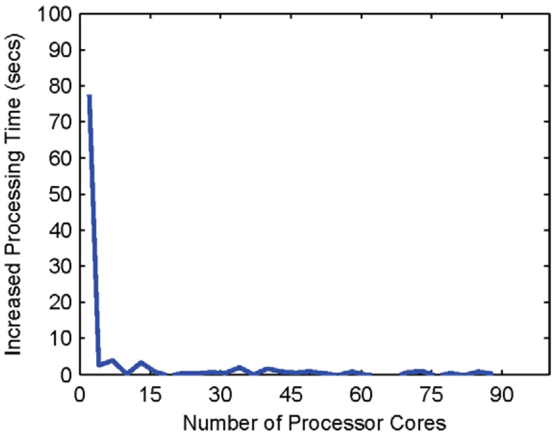


Fig. 8. Increased amount of processing time in Saving when using high compression quality. This figure shows the increased processing time using the compression quality of 100 over compression quality of 0 for TMA1, this plot shows the difference = (processing time for compression quality 100) − (processing time for compression quality 0). (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

which are stained as a single assay together with internal controls which can act as quantitative baseline. And while DAB staining is not stoichiometric, it is hoped that image measurement at least gives a more reliable evaluation of immunohistochemical detection of biomarkers than subjective visual interpretation. In this study, a novel centralised high performance computing approach was introduced for the rapid parallel analysis of TMAs using virtual slides.

The bespoke software developed as part of this study, allowed individual processor cores to retrieve and load tissue core sub-images from the main virtual TMA image, analyse these tissue core images using defined algorithms, generate important quantitative data from the tissue cores, and to do this in a highly parallelised fashion. While this software was developed and run on a dedicated high performance *HP BladeSystem Cluster* with >9000 processor cores, it can easily operate on clusters with a much lower specification, even down to a standard dual-core platform common now on any standard desktop or laptop PC architectures. The functional components of the platform (i.e., Image File Access, Parallel Processing, Analytic and Web serving modules) were designed to allow the system to be used immediately in a practical setting and this was evaluated using some typical examples. The proposed HPC architecture allows easy integration of other modules with additional functionality such as the support of virtual slide and image formats and other TMA core analytic algorithms. The *TMAX* database can also be extended if necessary.

Two approaches to load balancing, namely static load balancing and dynamic load balancing, were explored for the purpose of distributing image processing workloads evenly across processor-cores. Static load balancing best suits problems where image processing tasks and processing time for each TMA core are known *a priori*. Centralised dynamic load balancing approach performs better when image complexity and processing requirements across processors vary. Given that TMA core images are often heterogeneous resulting in different processing requirements, we found centralised dynamic load balancing approach to be most effective although the difference was marginal. However, if we were to transfer this to a platform where there was a limited number of processor-cores available (e.g., 2–8 processors), static load balancing would outperforms dynamic load balancing (as shown in Fig. 5). This is mainly due the fact that no processor cores are sacrificed for scheduling tasks in static load balancing.

Experiments using a variety of TMA samples and algorithms of varying complexity showed significant speed improvements when using multiple processors. For the analysis of single virtual TMA images with between 150 and 250 tissue cores, this study showed that image loading can speeded up by as much as 21.60 times when using HPC and image saving by 58.17. The overall performance for processing a TMA slide can be speeded up by 42.58. The most significant *Speedup* is achieved with *circa* 20 processor cores operating in parallel. Making available further processor cores only adds minimal benefit across the range of operations

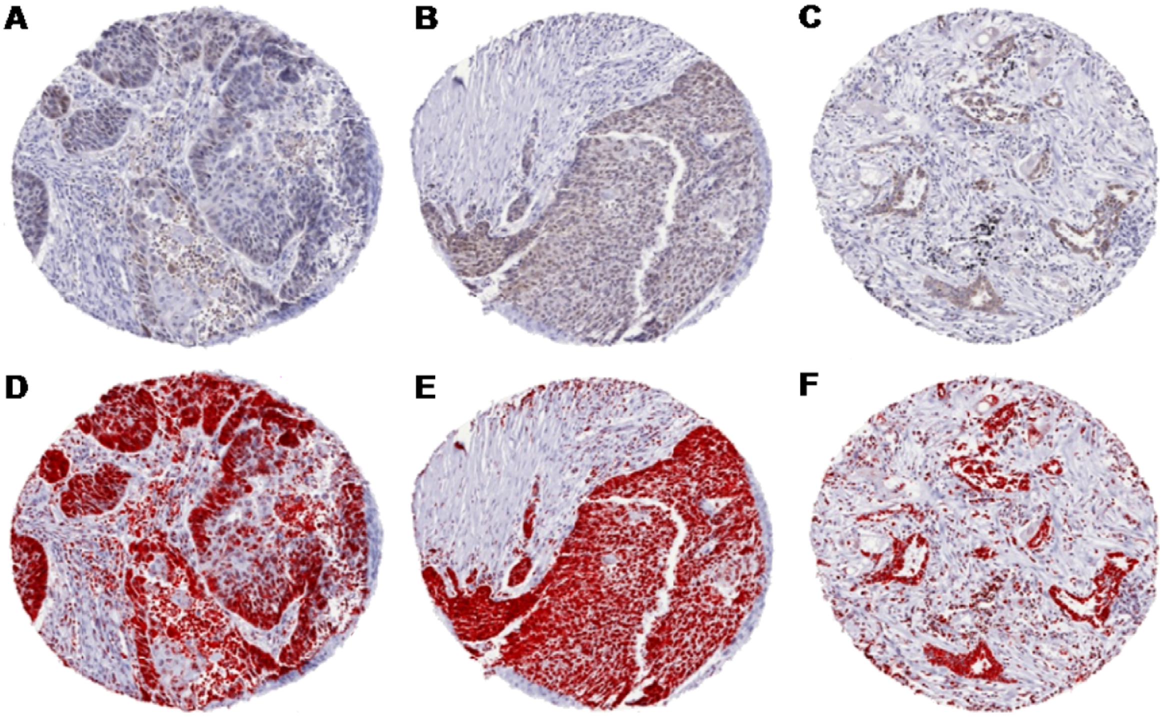


Fig. 9. Examples of biomarker quantification. A, B and C show 3 TMA core images, whereas D, E and F show their marked up DAB regions using overlapping red colour. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

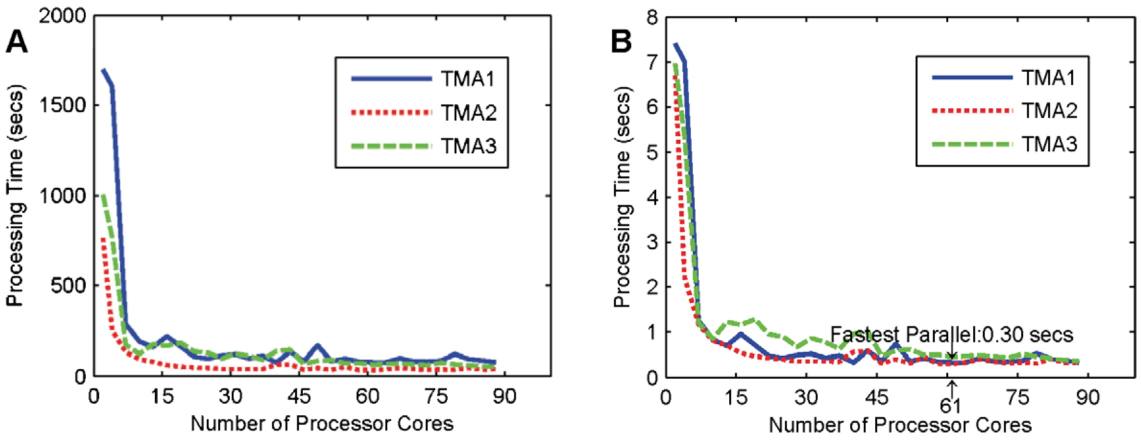


Fig. 10. Results for the amount of time for IHC quantification for all 3 TMA virtual slides. (A) Overall processing time for 3 TMA virtual slides, (B) average processing time per TMA core. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/ACP-CLO-2010-0551.)

Table 5

Statistics for the overall processing time for biomarker quantification over the 3 TMA virtual slides

	TMA1	TMA2	TMA3
Time taken using the fastest sequential code	1697.70 s	763.45 s	1003.30 s
Shortest parallel processing time	76.53 s	34.44 s	52.82 s
Number of processor-cores used	40	61	88
Maximum *Speedup*	22.19	22.17	19.01

that need to be performed on a single slide. The maximum theoretical speedup value can be explicitly calculated using Amdahl's law [6] and Gustafson's law [7]. A simple explanation in our case is that with the increased number of processor cores, the inter processor-core communication increases, e.g., the sending and receiving of tissue-core coordinates and acknowledgement messages for synchronisation. Such communications are a lot more time consuming by comparison with on-chip processing. Therefore the speedup gained with the increased amount of processor-cores eventually levels off.

The real benefit of using hundreds of processor cores comes when multiple virtual TMA slides need to be processed simultaneously. In high throughput laboratories, examining multiple biomarkers across TMA libraries, this could be of real benefit. Many TMA virtual slides can be loaded and saved simultaneously given there are enough processor cores. With the HP cluster used in the current study there are more than 9000 processor cores available. If we were to generously allocate 100 processor-cores for the processing of one TMA virtual slide, over 90 virtual TMA slides could be processed simultaneously without queuing, with the entire job taking <2 min by estimate, representing a *Speedup* of approximately 3285.90 times. This certainly provides an opportunity for the rapid evaluation of complex algorithms over large amount of TMA data.

Publications regarding the performance of the analysis of TMA virtual slides are rare, which makes performance comparison difficult. However, we have collected benchmarks from a small number of studies [6, 21]. They are summarised in Tables 6 and 7 and compared with our results.

Study [6] examined the role of parallel processing in TMA image analysis and tested the performance of data transfer between the remote storage and the computing nodes using Grid computing. They estimated data transfer rates of 1 megabyte ranging from 1.2 to 6.5 s (Table 6). In the current cluster based HPC platform and storage area network, the hard drives are communicating with the blade server through high speed fibre channel interconnect technology, data transfer latency is a lot smaller than its counterparts using Grids. We calculated data transfer time in the following way:

$$T_{transfer} \approx 0.5 \times (T_{Load} + T_{DeCom} + T_{Com} + T_{Save}), \tag{2}$$

where T_{Load} is the time for loading every TMA image data from a virtual slide into memory from the cluster's networked hard drives, T_{DeCom} is the time for JPEG decompression, T_{Com} is the time for JPEG compression of all TMA core images and T_{Save} is the time to store TMA core images at their full resolutions onto the cluster's hard drives.

As summarised in Table 6, our approach takes 0.0077 s (7.7 ms) to transfer per megabyte of data, which significantly outperformed all benchmarks given by [6]. As shown in Section 3.3, the amount time used for data transfer takes up a large portion of the overall processing time. The significant *Speedup*s in data transfer time results in the *Speedup* in overall processing time, and it clearly exhibited the superiority of using the proposed high performance clusters over traditional Grids.

The time taken to carry out image analysis of the tissue core depends on the complexity of the algorithm. In the current study, the biomarker IHC quantification algorithm is inherently more complex than image texture computation and takes considerably longer to run. It is therefore difficult to directly compare results from different studies when the algorithms used are different. Yang et al. [18] used a Grid computing approach to analysis breast TMA samples and recorded an average speed of 0.64 s per tissue core (Table 7). In the current study, using <100 processor cores, texture feature computation took 0.04 s per TMA core with biomarker IHC quantification algorithm taking 0.34 s per TMA core. When consider the use of all 9000 processor cores with the allocation of 100 processor cores for each TMA virtual slide, our conservative estimation would indicate that for texture imaging the processing time for each TMA core would be 0.0006 s (0.6 ms) and for IHC quantitation, 0.005 s (5 ms), representing *Speedup* values of 3285.90 and 1461.79, respectively.

While the absolute value of these comparisons is not really meaningful, it indicates that at least the cluster based HPC approach is competitive with other studies of this type and at most provides a faster, more reliable

Table 6
Data transfer speed comparison between this study and Galizia et al.

	Method	Data transfer time per MB (s)
Galizia et al.	FTP	1.22
	LCG2	6.51
	GFAL3	2.77
Our approach	Sequential	0.1423
	Parallel	*0.0077*

Notes: FTP – File transfer protocol; LCG – Large hadron collider computing grid; GFAL – Grid file access library.

Table 7

Computation performance comparison between the this study and Yang et al.

	No. TMA slides	No. TMA cores	Method	Processing time for all data (s)	Processing time per TMA core (s)	*Speedup*
Yang et al.	n/a	3,744	Sequential	18,144,000	4,846.20	7,560
			Grid	2,400	0.64	
Our approach (average per slide)	1	162.3	Sequential (Texture)	262.20	1.61	36.51
			Parallel (Texture)	7.15	*0.04*	
			Sequential (Quantification)	1,155	7.11	20.93
			Parallel (Quantification)	54.60	*0.34*	
Our approach (theoretical estimation)	90	14,610	Sequential (Texture)	23,580	1.61	3,285.90
			Parallel (Texture)	8.72	*0.0006*	
			Sequential (Quantification)	103,932	7.11	1,461.79
			Parallel (Quantification)	76.53	*0.005*	

and manageable approach for high throughput TMA analysis. It is well recognised in the HPC community that cluster-based approaches are faster since the processor technology is consistent with fast communication between processors. The benefit of cluster-based HPC also lies in the ability to manage load balancing strategies and the architectural design of all functional modules. In building robust and reliable architectures for high throughput analysis of TMAs this is the preferred model.

With increased demand for biomarker discovery as part of tissue-based research programmes, drug discovery and clinical trials, the demand for TMA generation and analysis is increasing. The approach outlined in this paper represents a powerful yet practical approach to the genuine high throughput analysis of TMA using a combination of virtual microscopy, image analysis and HPC to provide rapid quantitative data on tissue samples. This approach represents the key to allow fast and reliable evaluation of biomarkers as a means of defining their relationship to diagnosis, clinical outcome and response to therapy.

References

[1] Aperio Technologies, Inc., Digital slides and third party data interchange, 2006.

[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 Informatics and Decision Making* **3**(1) (2003), 5.

[3] T. Chittenden, E.A. Harrington, R. O'Connor, C. Remington, R.J. Lutz, G.I. Evan et al., Induction of apoptosis by the Bcl-2 homologue Bak, *Nature* **374**(6524) (1995), 733.

[4] J. Diamond, N.H. Anderson, P. Bartels, R. Montironi and P.W. Hamilton, The use of morphological characteristics and texture analysis in the identification of tissue composition in prostatic neoplasia, *Human Pathology* **35**(9) (2004), 1121–1131.

[5] K.A. DiVito and R.L. Camp, Tissue microarrays – automated analysis and future directions, *Breast Cancer Online* **8** (2005), online.

[6] A. Galizia, F. Viti, D. D'Agostino, I. Merelli, L. Milanesi and A. Clematis, Experimenting grid protocols to improve privacy preservation in efficient distributed image processing, in: *Parallel Computing: Architectures, Algorithms and Applications*, C. Bischof et al., eds, John von Neumann Institute for Computing, Julich, 2007, pp. 139–146.

[7] R. Kamalov, M. Guillaud, D. Haskins, A. Harrison, R. Kemp, D. Chiu et al., A Java application for tissue section image analysis, *Computer Methods and Programs in Biomedicine* **77**(2) (2005), 99.

[8] K. Kayser, D. Radziszowski, P. Bzdyl, R. Sommer and G. Kayser, Towards an automated virtual slide screening: theoretical considerations and practical experiences of automated tissue-based virtual diagnosis to be implemented in the Internet, *Diagnostic Pathology* **1** (2006), 10.

[9] A.M. Mahmoud, A.G. Philipp, D. James, S. Osama, M. Perry, M. Rodolfo et al., Epigenetic events, remodelling enzymes and their relationship to chromatin organization in prostatic intraepithelial neoplasia and prostatic adenocarcinoma, *BJU International* **99**(4) (2007), 908–915.

[10] S. Manley, N.R. Mucci, A.M. De Marzo and M.A. Rubin, Relational database structure to manage high-density tissue microarray data and images for pathology studies focusing on clinical outcome: the prostate specialized program of research excellence model, *American Journal of Pathology* **159**(3) (2001), 837–843.

[11] E. Oda, R. Ohki, H. Murasawa, J. Nemoto, T. Shibue, T. Yamashita et al., Noxa, a BH3-only member of the Bcl-2 family and candidate mediator of p53-induced apoptosis, *Science* **288**(5468) (2000), 1053–1058.

[12] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man and Cybernetics* **9**(1) (1979), 62.

[13] J. Packeisen, E. Korsching, H. Herbst, W. Boecker and H. Buerger, Demystified. Tissue microarray technology, *Molecular Pathology* **56**(4) (2003), 198–204.

[14] B. Rajkumar and M. Manzur, *Gridsim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing*, Wiley, New York, 2002.

[15] M.G. Rojo, G.B. García, C.P. Mateos, J.G. García and M.C. Vicente, Critical comparison of 31 commercially available digital slide systems in pathology, *International Journal of Surgical Pathology* **14**(4) (2006), 285–305.

[16] A.C. Ruifrok and D.A. Johnston, Quantification of histochemical staining by color deconvolution, *Analytical Quantitative Cytology and Histology* **23**(4) (2001), 291–299.

[17] C. Schmidt, M. Parashar, W. Chen and D.J. Foran, Engineering a peer-to-peer collaboratory for tissue microarray research, in: *Proceedings of CLADE Workshop*, at HPDC 13th, Honolulu, HI, 2004, pp. 64–73.

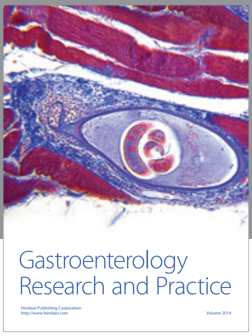
[18] D.G. Soenksen, Automated microscopic inspection of tissue microarrays using virtual microscopy, *Genomics Proteomics Technology* **8**(1) (2003), 28–31.

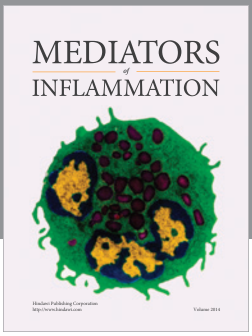
[19] F. Viti, I. Merelli, A. Galizia, D. D'Agostino, A. Clematis and L. Milanesi, Tissue MicroArray: a distributed grid approach for image analysis, *Studies in Health Technology Informatics* **126** (2007), 291–298.

[20] Y. Wang, D. Crookes, O.S. Eldin, W. Shilan, P. Hamilton and J. Diamond, Assisted diagnosis of cervical intraepithelial neoplasia (CIN), *IEEE Journal of Selected Topics in Signal Processing* **3**(1) (2009), 112.

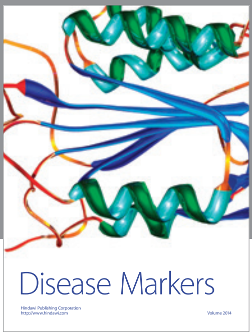
[21] L. Yang, W. Chen, P. Meer, G. Salaru, M.D. Feldman and D.J. Foran, High throughput analysis of breast cancer specimens on the grid, in: *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2007*, N. Ayache, S. Ourselin and A. Maeder, eds, Springer, Berlin/Heidelberg, 2007, pp. 617–625.

The Scientific World Journal


Gastroenterology Research and Practice

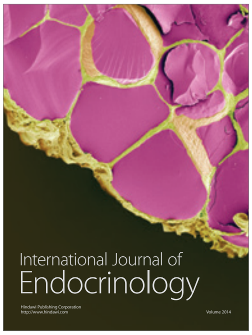

MEDIATORS of INFLAMMATION

Journal of Diabetes Research


Disease Markers

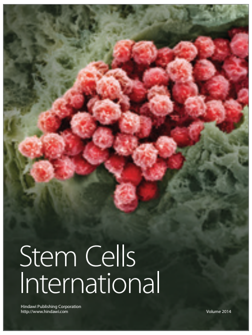
Journal of Immunology Research

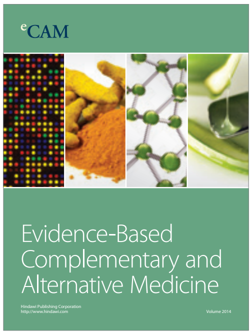
PPAR Research


International Journal of Endocrinology

BioMed Research International

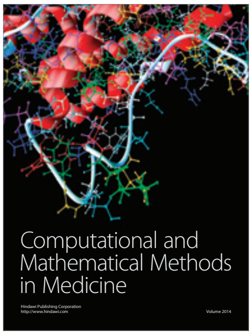
Journal of Ophthalmology

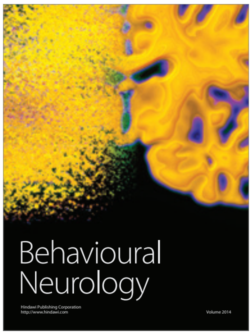

Stem Cells International


eCAM
Evidence-Based Complementary and Alternative Medicine

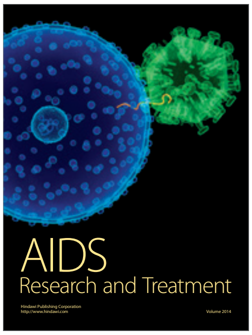
Journal of Obesity

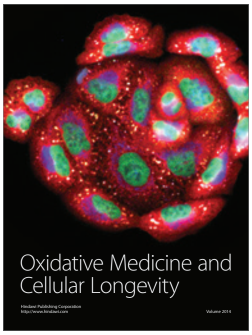
Journal of Oncology


Computational and Mathematical Methods in Medicine


Behavioural Neurology

Parkinson's Disease


AIDS Research and Treatment


Oxidative Medicine and Cellular Longevity