Chapter 5

PESSCARA: An Example Infrastructure for Big Data Research

Panagiotis Korfiatis and Bradley Erickson

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/63815

Abstract

Big data requires a flexible system for data management and curation which has to be intuitive, and it should also be able to execute non-linear analysis pipelines suitable to handle with the nature of big data. This is certainly true for medical images where the amount of data grows exponentially every year and the nature of images rapidly changes with technological advances and rapid genomic advances. In this chapter, we describe a system that provides flexible management for medical images plus a wide array of associated metadata, including clinical data, genomic data, and clinical trial information. The system consists of open-source Content Management System (CMS) that has a highly configurable workflow; has a single interface that can store, manage, enable curation, and retrieve imaging-based studies; and can handle the requirement for data auditing and project management. Furthermore, the system can be extended to interact with all the modern big data analysis technologies.

Keywords: big data, data analysis, content management system, curation, 3D imaging, workflows, REST API

1. Introduction

Big data is the term applied for data sets that are large and complex, rendering traditional analysis methods inadequate. 'Large' can be defined in many ways, including both the number of discrete or atomic elements, but also, the actual size in terms of bytes can also be important [1]. A single image can be viewed as being one datum, but in other cases may be viewed to have multiple data elements (i.e. each pixel). An image can be as small as 10s of bytes, but typically is megabytes, but can be several orders of magnitude larger. Furthermore, most

research requires many images, and usually further processing on each image must be done, yielding an enormous amount of data to be managed. For example, generating filtered versions of one 15 MB image can lead to several GB depending on the filters that been applied. Additionally, when the information is combined with metadata like genomic information or pathology imaging, the data increase exponentially in size [2–4].

Current popular non-medical imaging applications are as simple as determining if a certain animal is present in a picture. In some cases, medical imaging applications can be as simple: is there a cancer present in this mammogram? In most cases, though, the task is more complex: is the texture of the liver indicating hepatic steatosis, or is the abnormality seen on this brain MRI due to a high grade glioma, multiple sclerosis, a metastasis, or any of a number of other causes. In some respects, the problem is similar, but other aspects are different. The stakes are also much higher.

Medical image assessment nearly always requires other information about the patient-demographic data as well as information about family members that might help with genetically related diseases, or individual history of prior trauma or other disease. There are well-developed ontologies for describing these various entities though these are rarely used in routine clinical practice. Thus, as with other medical data mining efforts, collecting, transforming, and linking the medical record information to the images is a substantial and non-trivial effort [5].

Finally, once one has the images and appropriate medical history collected, the actual processing of the image data must begin. In many cases, multiple image types can be collected for a part of the body, and 'registering' these with each other is essential, such that a given x, y, z location in one image is the same tissue as in another image. Since most body tissues deform, this transformation is non-trivial. And tracking the tissues through time is even more challenging, particularly if the patient has had surgery or experienced other things that substantially changed their shape. Once the images are registered, one can then begin to apply more sophisticated algorithms to identify the tissues and organs within the image, and once the organs are known, one can then begin to try to determine the diagnosis.

One of the challenging tasks when dealing with big data when there are multiple associations, like medical images and metadata originating from a variety of sources, is management and curation [6]. Without proper organization, it is very challenging to extract meaningful results [7]. Big data analytics based on well-organized and linked data sets plays a significant role in aiding the exploration and discovery process as well as improving the delivery of care [8–10].

In this chapter, we describe a system we have constructed based on years of experience attempting to perform the above analysis. We believe that this system has unique properties that will serve as a basis for moving medical imaging solidly into the 'big data' world, including flexible means to represent complex data, a highly scalable storage structure for data, graphical workflows to allow users to efficiently operate on large data sets, and integration with GPU-based grid computers that are critical to computing on large image sets [11].

http://dx.doi.org/10.5772/63815

2. Unique requirements of medical image big data

2.1. Image data formats: DICOM, NIfTI, others

Most people are familiar with photographic standards for image files—JPEG, TIFF, PNG, and the like. These are designed to serve the needs of general photography, including support for RGB colour scheme, compression that saves space at the cost of perfect fidelity, and a simple header describing some of the characteristics of the photograph and camera.

Medical images share some similarity with photographic images—indeed in some cases, such as endoscopy, ophthalmology, or skin photographs use standard photographic methods. Pathology images are similar, but typically have much larger number of pixels—often billions of pixels for an image of an entire slide. Radiologic images are unique in that most are grey scale only and with a larger number of grey scales (16 bits or 65,536 grey levels) than photographic images. The result was that standards for photographic images did not support the needs of the early digital imaging modalities (which were mostly in radiology). The American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) recognized the increasing need for standards for exchanging digital images and developed the ACR-NEMA standard for medical images, which was released in 1985. The third version of ACR-NEMA dropped previously described hardware connection methods and focused on an information model, and exchange method that was generalized to non-radiology images and was designed to be used over standard networks. This third version was therefore renamed from 'ACR-NEMA' to 'DICOM' (Digital Communications in Medicine) [12]. The DICOM standard continues to evolve to support new imaging modalities and capabilities, and also new technical capabilities (e.g. RESTful interfaces). For many years, DICOM defined each image as its own 'object' and thus its own file. While was fine for radiographics images, it was more problematic for multi-slice image techniques like CT and MR that naturally produce images that are effectively three dimensional (3D). DICOM does support 3D image formats and also image annotation methods, but adoption of these has been slow, leading to use of other file formats for imaging research [13].

An early popular file format for medical image research was the Analyze© file format which had one small (384 bytes) header file, and a separate file which consisted of only image pixel data. The header proved too limiting for some uses, specifically its representation of image orientation, and was extended, resulting in the Neuroimaging Informatics Technology Initiative (NIfTI) file format (see http://brainder.org/2012/09/23/the-nifti-file-format/). There are other formats including Nearly Raw Raster Data (NRRD) (see http://teem.sourceforge.net/nrrd/index.html) that are also used in medical image research.

In most cases, each file format is able to represent the relevant information fairly well. There are many tools to convert between the various formats. The main advantage of these alternative formats is that a complete three or more dimensional data set is stored in a single file, compared to the popular 2D DICOM option which can requires many 10s to 1000s of files. Which file is selected is largely driven by the applications one expects to use, and the file formats they support.

2.2. Connecting images with image-specific metadata and other data

One of the major concerns when managing big data originating from medical practice is the data privacy. Data privacy is a critical issue for all people, but in most jurisdictions, there are specific requirements for how medical and health information must be kept private. One of the early comprehensive regulations on medical data privacy was the Health Insurance Portability and Accountability Act (HIPAA) [14]. It specified what data were considered private and could not be exposed without patient consent, and penalties for when such data breeches occurred. In the case of textual medical data, even a casual reader can quickly determine if protected Health Information (PHI) is within a document.

Medical images are more difficult to assess because DICOM images contain tags as part of the header that are populated with PHI during the normal course of an imaging examination. Releasing such medical images with that information in tact without patient consent would represent a breech of HIPAA. Removing these tags, and inserting some other identifier such as for research is straightforward to do in most cases. However, in some cases, vendors may also place PHI in non-standard locations of the header or may include it as part of the pixel information in the image. In some cases, this is done for compatibility with older software. In other cases, hospitals have been known to put PHI in fields that were designated for other purposes, to address their unique workflow needs. It is these exceptional cases that make de-identification more challenging. Fortunately, putting PHI into non-standard locations is declining as awareness of these problems is becoming better known.

Medical images may also contain PHI that is 'burned into' pixels—that is, the displayed image shows the PHI. While easily recognized by humans, it is more difficult for computers to recognize such PHI. One may use Optical Character Recognition algorithms, but they may have false negatives and positives due to the actual image contents looking like a character, or obscuring a character. Fortunately, the practice of burning in PHI is also declining.

When study of big data is conducted for clinical purposes, it may be appropriate to perform the research directly on medical records with the true medical record identifiers. This avoids the need for de-identification, which can be slow and expensive for some types of data. The medical record number usually makes it easy to tie various pieces of information for a subject together. However, having PHI directly accessible by computer systems beyond the Electronic Health Record (EHR) [15,16] represents increased risk of HIPAA or equivalent violation and therefore is discouraged.

Working on de-identified data substantially reduces the risk of releasing PHI during the course of big data research. This means that the de-identification step must be tailored for the type of data and that the de-identification also be coordinated so that the same study identifier is used. While not complex in concept, implementation can be more difficult if there is a strong need for rapid data access. The challenge is that when a new patient arrives in an emergency room, their true identity may not be known for some time, but medical tests and notes will be generated with a 'temporary ID'. How and when that temporary ID is changed to the final ID can be very different, and in some cases, a single temporary ID cannot be used in all systems.

http://dx.doi.org/10.5772/63815

Misidentified patients (e.g. same name) and correction of their data are similar problems. And cases where there is more than one subject (e.g. the foetus in a mother) also represent challenges that are manageable but must be considered up front. Obstetrical ultrasound images are nearly always of the foetus, but usually are collected under the identifier of the mother. In the case of twins, it can be challenging to know which foetus is seen on a given image, and such a notation is usually done by annotating the image (burning into pixels) rather than in a defined tag that is reliably computed.

2.3. Computational environment

Currently, there is no standard or expected computational environment used for image and metadata analysis. Researchers utilize a variety of operating systems, programming languages, and libraries (and versions of libraries). Furthermore, the tools can be deployed as command line executable, GUIs or more recently as web-based applications. There is a plethora of computational tools available but setting them up and maintaining them poses challenges. Setting up the appropriate environment is challenging since the user has to anticipate all the specific libraries and parameters that will be used during later computational steps. This is made more challenging because not all tools are available on any single platform. There is also an expectation of sharing data and algorithms, which also complicates long-term support of a platform.

Computation on medical images is very different from computation on other data types [17]. The fundamental unit in a medical image is the pixel, and the operations are those used in image processing elsewhere: filtering, artefact correction, registration/alignment, and segmentation to name a few [18]. Medical image analysis techniques are aimed in quantification of disease, image enhancement, detection of changes, or more generally dealing with medical image based problems originating from different imaging modalities utilizing digital image analysis techniques [18,19]. While these computations are unique to imaging, later steps that include classification and characterization or more generally analytical methods are similar to other big data efforts originating from different fields [20].

3. PESSCARA design

We have developed the Platform to Enable Sharing of Scientific Computing Algorithms and Research Assets (PESSCARA) to address the challenges we see with big data in medical imaging. The central component of PESSCARA is a Content Management System (CMS) that stores image data and metadata as objects. The CMS we chose is TACTIC (http://www.southpawtech.com), an open-source CMS with a Python API to access objects [21]. The Python API allows efficient development and testing of image processing routines on large sets of image objects [22]. TACTIC manages both project data and files, with project data stored in the database and files stored in the file system. TACTIC can store any type of data and image data format, including file formats commonly used in medical research, such as Analyze, NRRD, NifTI, and DICOM. The properties assigned to the image objects can be used to select the subset

of images to be processed, define the way that images are processed, and to capture some or all of the results of processing. TACTIC also has a workflow engine that can execute a series of graphically defined steps. Finally, it has project management facilities that can address planning, data auditing, and other aspects of project management.

To assist communication with the computational environment, we developed a Python library (tiPY) that facilitates input and output from TACTIC (**Figure 1**). PESSCARA is the first system that provides the research community with an environment suitable to deal with the requirements of medical image analysis while supporting the spirit of open and accountable research.

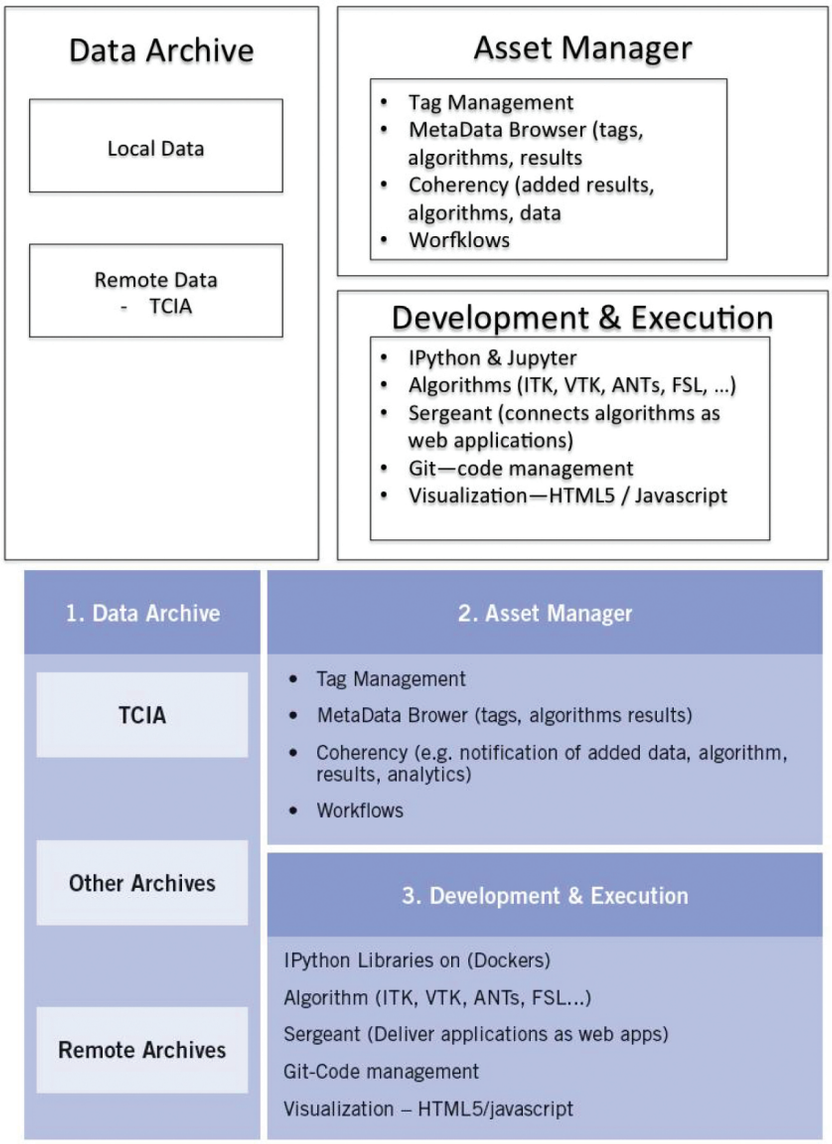

Figure 1. PESSCARA architecture. Most image analysis systems consist only of a data archive. PESSCARA includes this and allows for both federated and local data archives. PESSCARA also has an Asset Manager that allows flexible tagging of data, easy browsing of the data, and a workflow engine for processing data based on tags. Workflows and components of workflows are created in the development environment, and workflows are also executed in that same environment.

http://dx.doi.org/10.5772/63815

3.1. Databases vs content management

Databases are widely used for storing data. Although the main technology behind a CMS is essentially a database, in a CMS the content is not just a retrievable object, but also is an asset with properties. Such an object can be examined and displayed based on its properties, and based on those properties, it can be related to any other asset in the CMS. These additional capabilities make a CMS an excellent tool to use for big data research, since such data are complex and require metadata in order to assure proper processing and interpretation, thus leading to meaningful information [6,23].

PESSCARA is designed to link image and associated metadata with the computational environment. It allows users to focus on the content rather than database tables and gives great flexibility in assigning meaning to the various assets. Content in our example (discussed later in this chapter) consists of image data, metadata, biomarker information, notes, and tags.

TACTIC tracks the content creation process, which in the case of medical image research means the original acquired image, and all of its subsequent processing steps until the final measured version. TACTIC allows tracking of data check-in and checkout by providing a mechanism to identify changes; it also employs a versioning system to record the history of the changes to specific content. It also includes user logins and authentication, allowing tracking of who performed certain steps and when. Our adaptation of TACTIC for medical image research purposes was straightforward because medical images are digital content.

PESSCARA has a very flexible data-handling schema (**Figure 2**) that can easily address the heterogeneous data that are a part of 'big data', so it can adapt as new requirements emerge. It is easy to add other components to this schema to address other needs, for instance when genomic data need to be processed, rather than simply included as data.

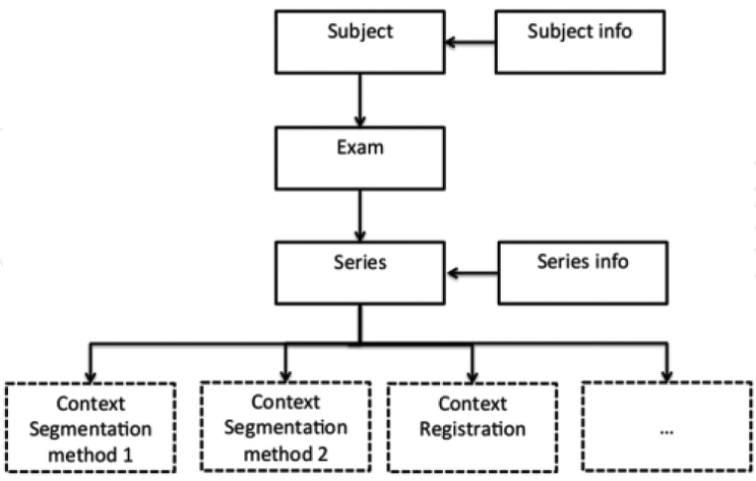

Figure 2. Data-handling schema. PESSCARA allows tags to be created for any object of group of objects. We established the basic organization of PESSCARA to have consistent tags at the Subject, Exam, Series, and Image level. There is also a 'study' level tag that equates to the institutional research board identifier, or essentially the project number. Each of these has a context that has permitted methods and workflows that can be applied.

All the data are available through a Representational State (REST) API designed to scale based on the requests issued from the analytical applications. Some of this is a part of TACTIC, though more of the management of computational tasks is through other components like sergeant and the grid engine (see **Figure 1**).

3.2. Workflow

When dealing with a large number of assets (data and metadata of any kind), it is crucial to have a mechanism that can automate and efficiently execute a specific series of actions on the data. In general, the workflows in medical imaging research tend to be linear and simple to implement. For example, a data importation/curation task typically begins by classifying the incoming image data based on their type, converting the data to a format suitable for subsequent analyses, placing new images on a queue for human quality control where the system then displays selected images and enables the reviewer to approve or reject them.

PESSCARA supports such workflows, which may be developed either as Python code, or developed graphically using the provided tool (**Figure 3**). PESSCARA users may design workflows and set the events that trigger workflows and define the users who are allowed to perform human steps. Tasks within the workflow can be calls to REST APIs, Python code, or notifications.

The workflows can be initialized based on events that can be either automated or manually controlled by a user or a prespecified group.

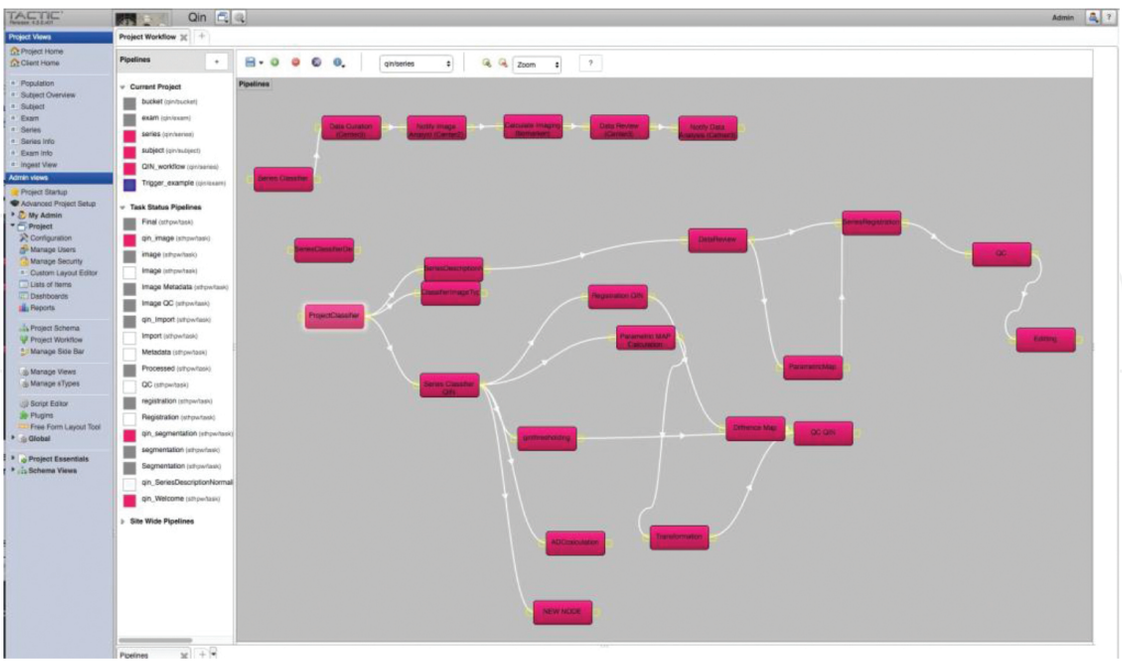

Figure 3. Snapshot of the pipeline creation tool. The pipeline workflow is used to depict the steps that a particular series need to undergo.

http://dx.doi.org/10.5772/63815

3.3. Grid computing

PESSCARA currently leverages the power of grid computing utilizing *sergeant* (https://github.com/potis/sergeant), which is an open-source tool that enables the deployment of code as web apps. This enables easy scalability, since the web app can be hosted on a cloud-based infrastructure design. Sergeant offers the ability to interact with each web app through a REST API, making it easier for people to utilize an application without the hustle of setting up and configuring binaries or executable. In the case of PESSCARA, a 'step' can be a call to sergeant, which in turn, could launch a grid job that might result in the processing of a large group of images utilizing the grid engine. This is, in fact, a common thing for us to do in our research efforts.

Cloud computing has been emerging as a good way to address computational challenges in modern big data research. This is because it is a way that a small research laboratory can access large computers, and the pay-as-you-go model provides flexibility for any size user. Cloud computing also addresses one of the challenges relating to transferring and sharing data, because data sets and analysis results held in the cloud can be shared with others just by providing credentials so they may also access the instance in the cloud.

The PESSCARA design allows us to leverage such cloud-computing resources. PESSCARA is engineered to support architectures such as MapReduce, Spark, and Storm [24–26] that are popular constructs in cloud computing. These technologies enable researchers to utilize data for fast analysis, with the end goal to translate scientific discovery into applications for clinical settings.

3.4. Multi-site synchronization

Content synchronization is an important requirement for multi-centre clinical trials and settings with multiple collaborators. TACTIC offers a powerful mechanism to synchronize data among servers hosting the databases and users, ensuring that changes are always up to date and that the correct version of the content is used. Encryption and decryption through a public- and private-key mechanism are used for all data transfers.

This is a particularly important feature for scientists, since 'data' include not just the raw data, but also all the metadata (which can be at least as laborious to create) and processed versions of data. PESSCARA achieves this via the content management system using the object capabilities, meaning that the visibility of what is shared and synchronized is very flexible and straightforward to administer.

We decided NOT to use this synchronization for algorithms, primarily because other tools such as github (www.github.com) already provide this capability, and specialized capabilities like merging of code—something that is not as easily done with a CMS, unless a special module was written for 'code' objects. Since github has already done this, we preferred to let users select the tool of their choice for code sharing and management.

4. Using PESSCARA

4.1. Data importation, curation, editing

PESSCARA incorporates dcm4che (http://www.dcm4che.org/) for DICOM connectivity and the Clinical Trial Processor (CTP) (https://www.rsna.org/ctp.aspx) for DICOM de-identification. The dcm4che module is an open-source Java library used as the DICOM receiver. The receiver can receive the images from a picture archiving and communications system or directly from the particular imaging modality.

Subsequently, CTP is used to de-identify the data for compliance with HIPAA. Tags that should be removed from the DICOM object are configured through a lookup table. In addition, CTP provides a log of all actions, which meets the logging requirements in 21 CFR part 11. During the de-identification process, a table with the correspondence between patient identifier and research identifier is kept and securely maintained. This table is useful for adding information to the patient dataset, such as tags from the pathology reports and survival information. In addition, when data corresponding to follow-up studies of patients who have been de-identified are included, CTP will assign the same research identifiers. Although CTP is capable of removing PHI, it can appear in many unexpected locations (e.g. burned-in pixel values). For this reason, PESSCARA is typically configured to place imported images in a 'quarantine' zone until the assigned user reviews the data. In our case, an important step of image importation is converting images from DICOM to NIfTI because most image processing packages do not deal well with native DICOM files. The tiPY library includes a routine to perform this conversion.

Once data have been imported into TACTIC and some initial workflows have been completed (i.e. for image series classification, or querying databases to gather additional information such as genomics or survival information), TACTIC workflow places the object on a queue for data quality inspection. At this point, information missing can be added manually, and poor quality items can be censored.

The project management element of PESCARA enables project managers to monitor resource usage and progress. This can allow tracking of resources used to support accurate billing and know individual effort. One can also assign total expected counts and thus calculate fractional completion.

To ensure data security, PESSCARA regularly backs up all parameter files used by CTP, dcm4che, the virtual machine running TACTIC, and the file storage area. This exists as just another workflow and thus is flexible in what is included, frequency, and how it is performed.

4.2. Creating image processing modules/dockers

Distribution of image analysis algorithms, particularly when developed in small research laboratories, is challenging since currently there is not standardized image analysis development environment. When the user employs the PESSCARA infrastructure, they are working with a standardized environment that usually enables easy deployment of the algorithm.

http://dx.doi.org/10.5772/63815

However, for algorithms that are not easy to be implemented in the PESSCARA environment (i.e. the LINUX host running PESSCARA), there is support for docker containers (http://www.docker.com) to perform 'steps' of a workflow.

Just as sergeant is able to 'request' execution of steps through a REST API that might result in submission of jobs to a grid engine, it is possible to 'request' the instantiation of a docker container that could perform a given step. The benefit of a docker container is that the execution environment is defined by the docker creator and is allowed to be different from the host environment. Virtual machines also have this benefit, but virtual machines require much more computer resource to execute. A disadvantage is that currently Microsoft Windows and Apple OS X applications are not supported; though, Windows support has been announced.

For development purposes, PESSCARA supports a majority of tools used in the image processing community, including ITK, Slicer3D, FSL, and others. However, for algorithm development, Python is the preferred language for PESSCARA. Python is a very approachable, readable language that includes a number of powerful tools including Numpy, Matplotlib, scikit-learn, nipype, RPy, and pandas. The Jupyter Notebook development framework extends Python and is at the core of a substantial shift in the methodology of science, enabling iteration, documentation, and sharing of science. This philosophy is in perfect alignment with PESSCARA. It promotes reproducible research (i.e. provenance tracking of the entire history from input data, algorithms used, intermediate calculations, and results). Its interactive capabilities means that code that code already run can have its results used rather than re-running the code.

While Python is the 'first language' of PESSCARA, there are many libraries and developers that depend on other languages, including non-Python tools such as ITK, FSL, ANTs, Slicer, and others. Furthermore, Jupyter enables development in many different languages including R, C++, and Julia. [27].

A Jupyter Notebook (which includes code, data, and results) can be easily shared by simply giving the URL and login credentials to your audience. In addition, the Results/Output and comments (including LaTex and Markdown) can be integrated into the Notebook to document what has been done in a long-term and shareable way.

The basic model for such 'shared science' is import/export. The user often starts by importing other investigators' Notebooks, but they may also start their own. They can then develop in their own 'sandbox', and when they feel they have something to share, they can 'export' it, which makes it publicly visible and available to be imported by others. Exporting the code in conventional Python format is also supported. They can also save all code and results as HTML for publishing on the web, or as PDF as a 'final' document to be saved in an electronic laboratory notebook [28].

Based on this architecture, the algorithms can be utilized by a variety of cloud services and important characteristic to consider when large amount of data are involved.

4.3. Creating and executing workflows

As noted above, workflow is critical in modern science. One must be able to execute the research process consistently. When dealing with 'big data', efficiency is also essential. In the following section, we show a multi-centre implementation of a workflow created with PESSCARA (**Figure 4**). The application will be aimed at developing imaging biomarkers for differentiating between progression and pseudoprogressions in case of glioblastoma multiforme (a type of malignant brain tumour) using large data sets and then applying the findings from a large data set to a live clinical trial and ultimately routine clinical practice.

We see PESSCARA having two configurations: one for development and one for clinical trials or practice. The development configuration includes the CMS system with the data used for development as well as a large batch-oriented computational environment. Once the code and the workflows have been established, the clinical configuration is created containing only the workflows and the computational environed to support them.

Following is an example of how the two configurations of PESSCARA can work.

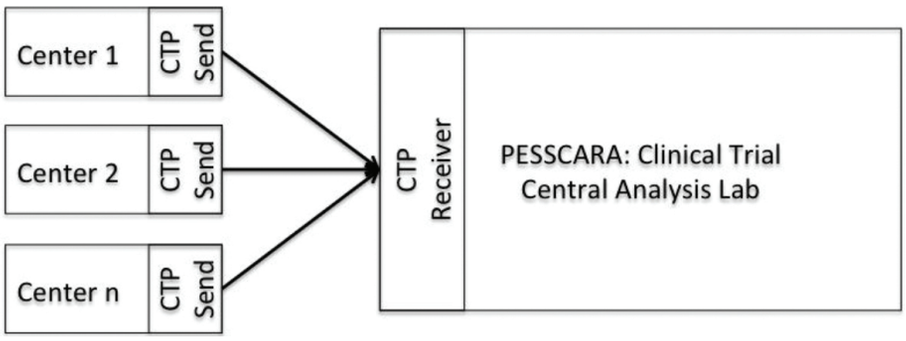

Figure 4. Translation of workflows created with PESSCARA for a multi-centre set-up. Each of n Centers collects image data and sends via CTP software. The same CTP software also acts as a receiver at the Central Analysis Lab, where CTP sends it to PESSCARA for analysis. We expect there would be a separate instance of PESSCARA for a clinical trial to minimize the chance that a developer would alter data or impact performance.

Researchers from the participating institutions can use the PESSCARA development configuration to develop the image analysis algorithms as well as the workflows necessary to compute the image-based biomarker. Typically, data from multiple centres are used for analysis. Both development and clinical trial configurations will typically have an input process where data are reviewed for quality and then stored. In the case shown in **Figure 5**, we imagine that Center 3 is responsible for curating the data, and after that, Center 2 will perform visual QC of image quality and automated image segmentation. Center 3 then reviews Center 2's work, and Center 1 is notified that data analysis is complete. In the development configuration, there is a loop where Center 1 along with other centres may refine the analysis, and further computational models/biomarkers are tested. When the workflow is completed and the supporting web app established the PESSCARA, clinical trial configuration is created.

http://dx.doi.org/10.5772/63815

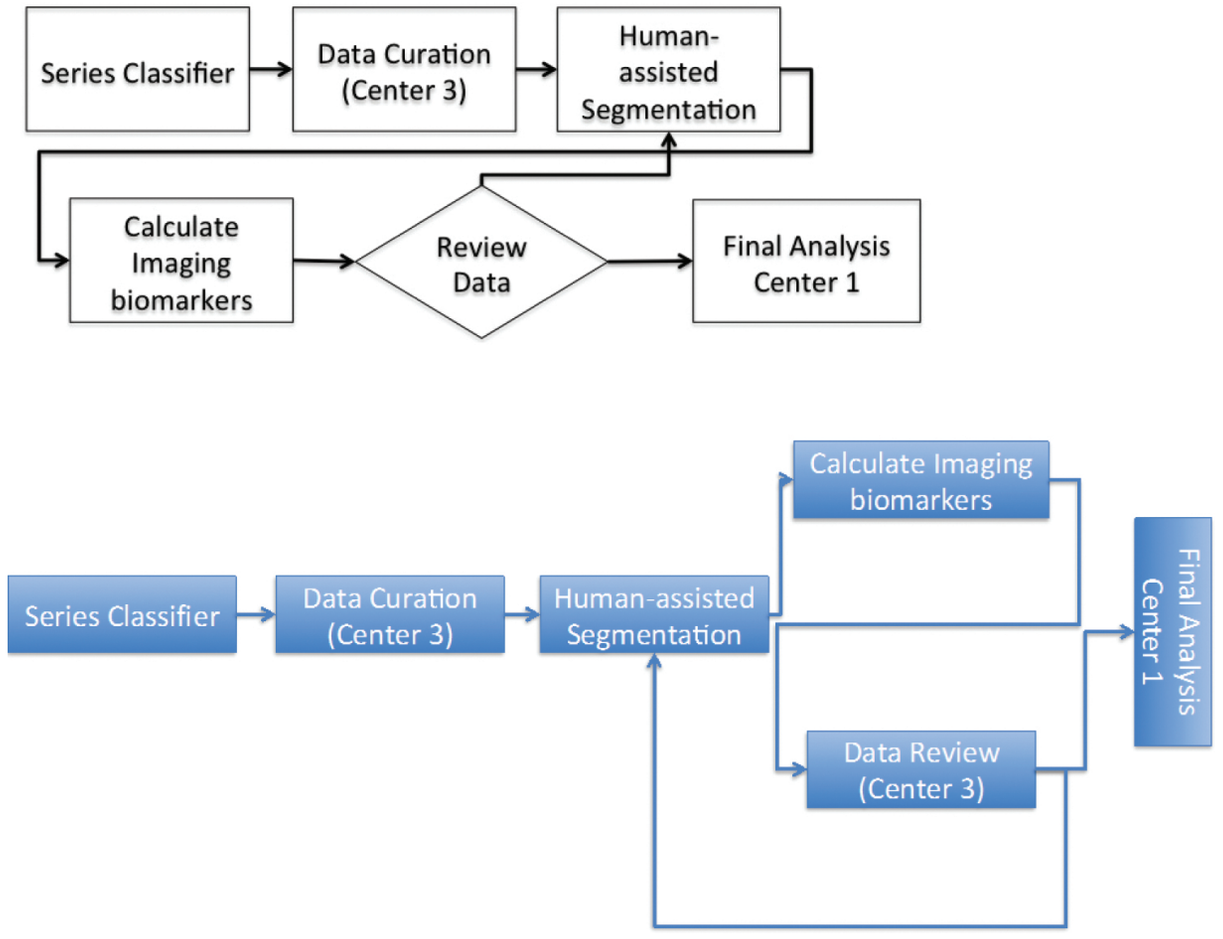

Figure 5. Example workflow. In this case, images are first identified by the Series Classifier. Once they are labelled, Data Curation is performed, in this example at a remote centre (Center 3). Then, human-assisted segmentation is performed, and biomarkers and then computed. This is again reviewed by a human, and if acceptable, the measurements are sent to the central data collection.

The clinical trial configuration is focused on efficient calculation of a biomarker developed via the above mechanism, and in some cases, it also provides a mechanism for immediate delivery of the biomarker result. As with development, when a subject has been identified in Center 1 as suitable for the study, it is forwarded to the PESSCARA DICOM receiver set-up for this study. The dataset PHI are de-identified through use of the CTP functionality and a preconfigured CTP configuration file. All the received files are placed in a folder, where they are 'ingested'. The metadata are also forwarded to the system utilizing the tiPY library. A configuration file exists in the receiving pool to assign the proper tags to the data to be ingested, such as institutional review board number, data type, and project name. The ingesting process will create a new entry inside TACTIC or will update the information if the data already exist. Once the data have been injected, a Series workflow is triggered. The first step of the workflow is a classifier step, which routes the data for a specific study to the right pipeline—for instance, that an image series designed to measure perfusion is sent to an algorithm that calculates perfusion. Subsequently, DICOM field tags are extracted and a normalized series description is assigned to each object (e.g. 'Axial', 'T1', and 'Post-Contrast' might all be assigned to an axial

postcontrast T1 image). If the classifier finds all the required series (T1 weighted postcontrast and perfusion in this case), a notification is send to the centre responsible for data curation). Otherwise, a notification/report of the data missing is send to the predesigned contact person in the originating centre.

Once data curation is finished, a notification is sent to centre 2 where the tumour segmentation is performed. The Image analyst can get the data either through the web page or through a link, to perform the tumour segmentation task. Once this is completed, the step(s) responsible for perfusion analysis computation as well as the registration of the tumour ROI to the perfusion image is executed. Once the data are reviewed and found acceptable, the imaging biomarkers extracted from perfusion are assigned to the appropriate tags for that examination. Once this step in completed, the data metadata and all analytics extracted are available for analysis utilizing any kind of 'big data' analysis methodology. This may be simply stored for later group analysis or may be made available for immediate clinical decision-making. All the data and metadata created during the execution of the workflow are backed up to a different server for protection over data loss.

4.4. Current status and next steps

Currently, the system is under development with further optimization needed to enhance its security features. Additionally, further resources are needed to provide the users with more resources for faster testing and support for algorithms with higher computational requirements. The system has been undergoing rapid development—the documentation and training resources have not kept up.

We hope that the next phases will see further connections of PESSCARA with non-imaging data repositories; improvements in the workflow engine enable a wider variety of algorithms on a wider variety of platforms and greater connections to clinical systems.

We do intend to provide the basic system as open access tools through github so researchers will be able to set the same environment locally with more resources. We also hope to provide a simple demonstration environment (http://www.PESSCARA.org) that will allow prospective users to test the PESSCARA environment.

5. Conclusion

Big data techniques will lead to an improved model of healthcare delivery with the potential to achieve better clinical outcomes and increased efficiency. However, appropriate infrastructure is needed to enable the data collection and curation especially in case of heterogeneous (with respect to data) environments such as healthcare.

PESSCARA aims to minimize the requirements for data downloading and transfer, since data and metadata are hosted within the same infrastructure. Code development also can be performed through a web interface making the system easy to use for inexperienced users. Perhaps even more important is that researchers can share their algorithms—the analysis

http://dx.doi.org/10.5772/63815

performed, and the subsequent results—which is a significant step toward reproducible research. When big data originate from multimodal data that have complex connections to other data, the use of a CMS is a must. PESSCARA is and will continue to meet the unique demands of big data research in medical imaging by leveraging a good CMS that is effectively connected to powerful computational resources, and an algorithm development environment designed for code and result sharing. 'Shared Science' is the future of science, and PESSCARA is one tool for medical imaging to participate in this new world of big data and shared science.

Acknowledgements

This work was supported by NIH Grant CA160045.

Author details

Panagiotis Korfiatis and Bradley Erickson*

*Address all correspondence to: bje@mayo.edu

Department of Radiology, Mayo Clinic, Rochester, MN, United States

References

[1] Jagadish HV. Big data and science: myths and reality. Big Data Res. 2015 June;2(2):49–52.

[2] Tan SS, Gao G, Koch S. Big data and analytics in healthcare. Methods Inf Med. 2015 November 27;54(6):546–7. PubMed PMID: 26577624.

[3] Belle A, Thiagarajan R, Soroushmehr SM, Navidi F, Beard DA, Najarian K. Big data analytics in healthcare. Biomed Res Int. 2015;2015:370194. PubMed PMID: 26229957. PMCID: PMC4503556.

[4] Langer SG. Challenges for data storage in medical imaging research. J Digit Imaging. 2011 April;24(2):203–7. PubMed PMID: 20544372. PMCID: PMC3056978.

[5] Prather JC, Lobach DF, Goodwin LK, Hales JW, Hage ML, Hammond WE. Medical data mining: knowledge discovery in a clinical data warehouse. Proceedings of AMIA Annual Fall Symposium. 1997:101–5. PubMed PMID: 9357597. PMCID: PMC2233405.

[6] Lynch C. Big data: how do your data grow? Nature. 2008 September 4;455(7209):28–9. PubMed PMID: WOS:000258890200019 [English].

[7] Mathew P, Pillai A. Big data challenges and solutions in healthcare: a survey. In: Snášel V, Abraham A, Krömer P, Pant M, Muda AK, editors. Innovations in Bio-Inspired Computing and Applications. Advances in Intelligent Systems and Computing. 424: Springer International Publishing; 2016. p. 543–53.

[8] Raghupathi W, Raghupathi V. Big data analytics in healthcare: promise and potential. Health Inf Sci Syst. 2014;2:3. PubMed PMID: 25825667. PMCID: PMC4341817.

[9] Trifonova OP, Il'in VA, Kolker EV, Lisitsa AV. Big data in biology and medicine: based on material from a joint workshop with representatives of the international Data-Enabled Life Science Alliance, July 4, 2013, Moscow, Russia. Acta Nat. 2013 July;5(3): 13–6. PubMed PMID: 24303199. PMCID: PMC3848064.

[10] Margolis R, Derr L, Dunn M, Huerta M, Larkin J, Sheehan J, et al. The National Institutes of Health's Big Data to Knowledge (BD2K) initiative: capitalizing on biomedical big data. J Am Med Inform Assoc. 2014 November–December;21(6):957–8. PubMed PMID: 25008006. PMCID: PMC4215061.

[11] Schadt EE, Linderman MD, Sorenson J, Lee L, Nolan GP. Computational solutions to large-scale data management and analysis. Nat Rev Genet. 2010 September;11(9):647–57. PubMed PMID: 20717155. PMCID: PMC3124937.

[12] National Electrical Manufacturers Association, American College of Radiology. Digital imaging and communications in medicine (DICOM). Washington, DC: National Electrical Manufacturers Association; 1998. vol. 1–8, p. 10–5.

[13] Larobina M, Murino L. Medical image file formats. J Digit Imaging. 2014 April;27(2): 200–6. PubMed PMID: 24338090. PMCID: PMC3948928.

[14] Services USDoHH. U.S. Department of Health & Human Services [cited 2016 01/01/2016]. Available from: http://www.hhs.gov/ocr/privacy/hipaa/understanding/summary/.

[15] Hamilton B. Electronic Health Records. 3rd ed. New York: McGraw-Hill; 2013.

[16] Carter JH, American College of Physicians. Electronic Health Records: A Guide for Clinicians and Administrators. 2nd ed. Philadelphia: ACP Press; 2008. vol. xxi, 530 p.

[17] Duncan JS, Ayache N. Medical image analysis: Progress over two decades and the challenges ahead. IEEE Trans Pattern Anal. 2000 January;22(1):85–106. PubMed PMID: WOS:000085472300005 [English].

[18] Dhawan AP. Medical Image Analysis. Hoboken, NJ, Piscataway, NJ: Wiley-Interscience, IEEE Press; 2003. vol. xv, 315p.

[19] Costaridou L. Medical Image Analysis Methods. Boca Raton: CRC Press/Taylor & Francis; 2005. 489p.

[20] Mohammed EA, Far BH, Naugler C. Applications of the MapReduce programming framework to clinical big data analysis: current landscape and future trends. BioData Min. 2014;7:22. PubMed PMID: 25383096. PMCID: PMC4224309.

[21] Technology S. TACTIC Digital Asset and Workflow Software [cited 2014 01/01/2016].

[22] Korfiatis PD, Kline TL, Blezek DJ, Langer SG, Ryan WJ, Erickson BJ. MIRMAID: a content management system for medical image analysis research. Radiographics. 2015 September–October;35(5):1461–8. PubMed PMID: 26284301. PMCID: PMC4613872.

[23] Chen JC, Chen YG, Du XY, Li CP, Lu JH, Zhao SY, et al. Big data challenge: a data management perspective. Front Comput Sci-Chi. 2013 April;7(2):157–64. PubMed PMID: WOS:000317303800001 [English].

[24] Richter AN, Khoshgoftaar TM, Landset S, Hasanin T, editors. A multi-dimensional comparison of toolkits for machine learning with big data. In: 2015 IEEE International Conference on Information Reuse and Integration (IRI), 13–15 August, 2015.

[25] Sun Z, Chen F, Chi M, Zhu Y. A spark-based big data platform for massive remote sensing data processing. In: Zhang C, Huang W, Shi Y, Yu PS, Zhu Y, Tian Y, et al., editors. Data Science. Lecture Notes in Computer Science. 9208: Springer International Publishing; 2015. p. 120–6.

[26] Dean J, Ghemawat S. Mapreduce: simplified data processing on large clusters. Commun ACM. 2008 January;51(1):107–13. PubMed PMID: WOS:000251994700031 [English].

[27] Perez F, Granger BE. IPython: A system for interactive scientific computing. Comput Sci Eng. 2007 May–June;9(3):21–9. PubMed PMID: WOS:000245668100005 [English].

[28] Shen H. Interactive notebooks: sharing the code. Nature. 2014 November 6;515(7525): 151–2. PubMed PMID: 25373681.

IntechOpen