Journal of
Medical Imaging

MedicalImaging.SPIEDigitalLibrary.org

Quantitative imaging feature pipeline: a web-based tool for utilizing, sharing, and building image-processing pipelines

Sarah A. Mattonen
Dev Gude
Sebastian Echegaray
Shaimaa Bakr
Daniel L. Rubin
Sandy Napel

SPIE.

Sarah A. Mattonen, Dev Gude, Sebastian Echegaray, Shaimaa Bakr, Daniel L. Rubin, Sandy Napel, "Quantitative imaging feature pipeline: a web-based tool for utilizing, sharing, and building image-processing pipelines," *J. Med. Imag.* **7**(4), 042803 (2020), doi: 10.1117/1.JMI.7.4.042803

Quantitative imaging feature pipeline: a web-based tool for utilizing, sharing, and building image-processing pipelines

Sarah A. Mattonen,[a,b,c] Dev Gude,[a] Sebastian Echegaray,[a] Shaimaa Bakr,[d] Daniel L. Rubin,[a,e,f] and Sandy Napel[a,d,e,*]

[a]Stanford University, Department of Radiology, Stanford, California, United States
[b]The University of Western Ontario, Department of Medical Biophysics, London, Ontario, Canada
[c]The University of Western Ontario, Department of Oncology, London, Ontario, Canada
[d]Stanford University, Department of Electrical Engineering, Stanford, California, United States
[e]Stanford University, Department of Medicine, Stanford, California, United States
[f]Stanford University, Department of Biomedical Data Science, Stanford, California, United States

Abstract. Quantitative image features that can be computed from medical images are proving to be valuable biomarkers of underlying cancer biology that can be used for assessing treatment response and predicting clinical outcomes. However, validation and eventual clinical implementation of these tools is challenging due to the absence of shared software algorithms, architectures, and the tools required for computing, comparing, evaluating, and disseminating predictive models. Similarly, researchers need to have programming expertise in order to complete these tasks. The quantitative image feature pipeline (QIFP) is an open-source, web-based, graphical user interface (GUI) of configurable quantitative image-processing pipelines for both planar (two-dimensional) and volumetric (three-dimensional) medical images. This allows researchers and clinicians a GUI-driven approach to process and analyze images, without having to write any software code. The QIFP allows users to upload a repository of linked imaging, segmentation, and clinical data or access publicly available datasets (e.g., The Cancer Imaging Archive) through direct links. Researchers have access to a library of file conversion, segmentation, quantitative image feature extraction, and machine learning algorithms. An interface is also provided to allow users to upload their own algorithms in Docker containers. The QIFP gives researchers the tools and infrastructure for the assessment and development of new imaging biomarkers and the ability to use them for single and multicenter clinical and virtual clinical trials. [DOI: 10.1117/1.JMI.7.4.042803]

Keywords: medical image analysis; radiomics; machine learning; feature extraction; processing pipeline.

Paper 19234SSR received Sep. 9, 2019; accepted for publication Feb. 26, 2020; published online Mar. 14, 2020.

1 Introduction

The field of quantitative imaging is rapidly growing, especially in the area of radiomics and machine learning. Radiomics aims to extract quantitative image features from medical images to identify valuable biomarkers of underlying cancer biology.[1–4] These features in combination with machine learning algorithms can be used for diagnosis and to predict clinical outcomes and/or treatment response.[5–7] In addition, association of these imaging features with cancer genomics or other patient information may further describe the fundamental biology.[8–10] Currently, quantitative image analysis tools are being developed for all disease sites and several imaging modalities to assess outcomes, diagnoses, and/or responses.[11–15] However, current

[*]Address all correspondence to Sandy Napel, E-mail: snapel@stanford.edu

radiomics tools are lacking sufficient evaluation and validation, and there is a lack of translation of these tools into the clinical workflow. This is in part due to the lack of available shared software algorithms and architectures to fully compare and evaluate these quantitative imaging tools across institutions. Similarly, researchers must also have expertise in writing software code to perform many image analysis tasks, including radiomic feature extraction and machine learning. Currently, many open-source pipelines, including Slicer Radiomics,[16] Orange,[17] and KNIME,[18] only process a single image at a time or only perform one quantitative imaging task, such as feature extraction or machine learning.

Therefore, what is critically needed is a user-friendly platform for sharing and assessing quantitative imaging algorithms. The quantitative imaging feature pipeline (QIFP) is an open-source, web-based platform that allows users access to a wide range of image processing and analysis tools without requiring writing code. Users are also able to upload their own algorithms in Docker containers, which allows the system to evolve and to support code that has been written in a variety of languages. This pipeline gives researchers the tools and infrastructure needed to assess and compare the value of combinations of quantitative image features. For example, researchers may want to create a pipeline that first performs segmentation of a region of interest, then performs feature extraction, and finally trains a machine learning classifier to predict an outcome of interest. The QIFP system allows users to complete these tasks in a single pipeline. It can also allow for the widespread development, assessment, and dissemination of new imaging biomarkers, including the opportunity for external validation of existing software pipelines. This system can also be used to facilitate incorporating quantitative imaging tools into single and multicenter clinical and virtual clinical trials specifically involving image processing, radiomics, and/or machine learning. The QIFP could be used as a central webserver where multiple institutions could upload de-identified imaging data and perform standardized image-processing pipelines.

2 Architecture

Figure 1 shows the architecture of the QIFP, which uses the Common Workflow Language (CWL) execution model and the CWL standard for defining tools and workflows. Simple CWL (json) formatted definitions of tools or workflows can be imported into or exported from the QIFP system. The QIFP leverages Docker for ease of sharing algorithms that have been written in a variety of languages and on a variety of platforms.[19] The entire QIFP is also available as a Docker version for installation on a local server to run within an institutional firewall. The server needs to have at least 4 cores and 64 GB of memory. A detailed user guide is available on the QIFP website[20] under documentation, which provides details on how to perform this installation, including the required Docker-composed file.

The QIFP system, a web application written in java, runs under a Tomcat webserver. When running the QIFP through the webserver, there are no configuration or minimum bandwidth

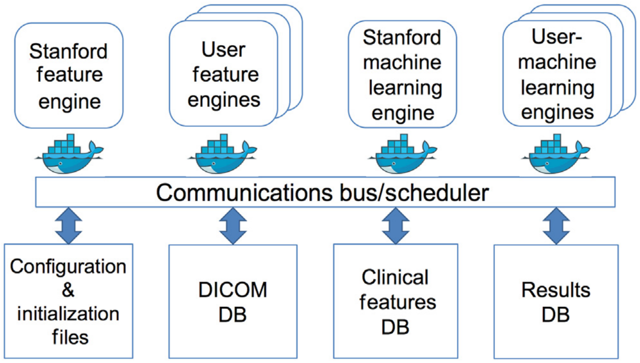

Stanford feature engine
User feature engines
Stanford machine learning engine
User-machine learning engines
Communications bus/scheduler
Configuration & initialization files
DICOM DB
Clinical features DB
Results DB

Fig. 1 QIFP architecture. The top half of the figure represents a Docker image capable of given task, such as feature extraction or machine learning. The lower half shows the connections to run configuration options files and to various local databases, such as DICOM images/segmentations, clinical features, or workflow results.

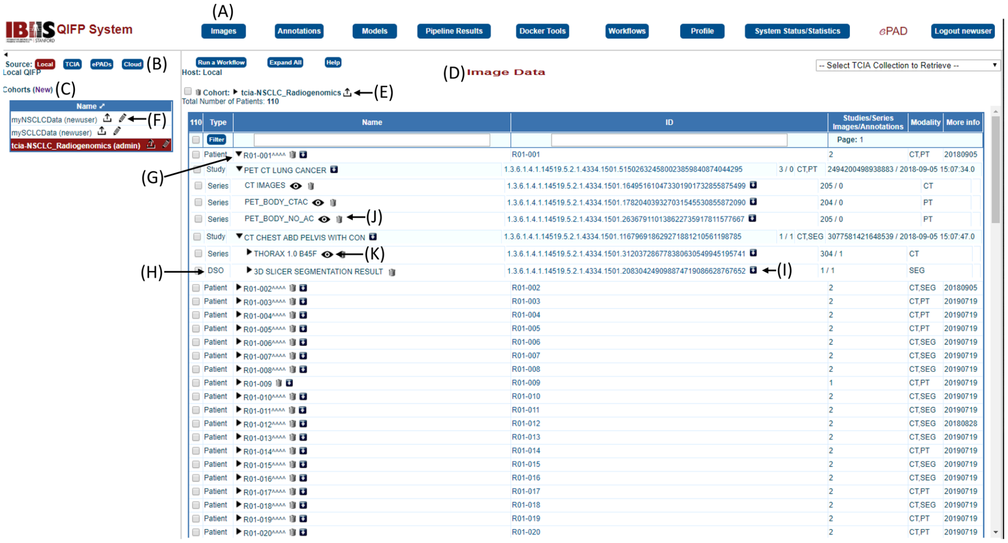
requirements; however, data upload and download speed will be determined by the user's local network. The QIFP schedules and monitors tools to be executed by a workflow. Each of the blocks in the top half of Fig. 1 represents a Docker image capable of, e.g., feature extraction, image conversion, or machine learning. The system acquires the image, semantic, and clinical data from one of various sources [e.g., user's computer, local database, the electronic Imaging Device (ePAD) system,[21] and the Cancer Imaging Archive (TCIA)[22]]. The appropriate Docker images are then scheduled to run with the input images and clinical data or with the output of a previously scheduled Docker image as input. After each tool has completed, the system stores the output of the tool in the local database and schedules the next tool to be run as defined by the workflow. Once the workflow has completed, it sends an email to the user with a link to the results. The lower half of Fig. 1 shows connections to run configuration options files and various local databases.

3 Interface

The QIFP is an open-source, web-based system publicly accessible at Ref. 20. After logging into the system with a distinct username and password, users will see the QIFP interface as shown in Fig. 2. Users can request an account on the main login page for the QIFP. Image cohorts are displayed on the left-hand panel and users can choose any of the top menu functions, as described below.

3.1 *Images Menu*

The QIFP maintains its own image repository and can connect to other image sources. Figure 2 shows the QIFP user interface when the "Images" menu (along the top row) and the "Local"



Cohort: ▸ tcia-NSCLC_Radiogenomics
Total Number of Patients: 110

110	Type	Name	ID	Studies/Series Images/Annotations	Modality	More info
□	Filter			Page: 1		
□	Patient	▾ R01-001^^^^	R01-001	2	CT,PT	20180905
□	Study	▾ PET CT LUNG CANCER	1.3.6.1.4.1.14519.5.2.1.4334.1501.515026324580023859840874044295	3 / 0	CT,PT	2494200498938883 / 2018-09-05 15:07:34.0
□	Series	CT IMAGES	1.3.6.1.4.1.14519.5.2.1.4334.1501.164951610473301901732855875499	205 / 0	CT	
□	Series	PET_BODY_CTAC	1.3.6.1.4.1.14519.5.2.1.4334.1501.178204039327031545530855872090	204 / 0	PT	
□	Series	PET_BODY_NO_AC	1.3.6.1.4.1.14519.5.2.1.4334.1501.263679110138622735917811577667	205 / 0	PT	
□	Study	▾ CT CHEST ABD PELVIS WITH CON	1.3.6.1.4.1.14519.5.2.1.4334.1501.116796918629271881210561198785	1 / 1	CT,SEG	3077581421648539 / 2018-09-05 15:07:47.0
□	Series	▸ THORAX 1.0 B45F	1.3.6.1.4.1.14519.5.2.1.4334.1501.312037286778380630549945195741	304 / 1	CT	
□	DSO	▸ 3D SLICER SEGMENTATION RESULT	1.3.6.1.4.1.14519.5.2.1.4334.1501.208304249098874719086628767652	1 / 1	SEG	
□	Patient	▸ R01-002^^^^	R01-002	2	CT,SEG	20180905
□	Patient	▸ R01-003^^^^	R01-003	2	CT,PT	20190719
□	Patient	▸ R01-004^^^^	R01-004	2	CT,PT	20190719
□	Patient	▸ R01-005^^^^	R01-005	2	CT,PT	20190719
□	Patient	▸ R01-006^^^^	R01-006	2	CT,SEG	20190719
□	Patient	▸ R01-007^^^^	R01-007	2	CT,SEG	20190719
□	Patient	▸ R01-008^^^^	R01-008	2	CT,SEG	20190719
□	Patient	▸ R01-009	R01-009	1	CT,PT	20190719
□	Patient	▸ R01-010^^^^	R01-010	2	CT,SEG	20190719
□	Patient	▸ R01-011^^^^	R01-011	2	CT,SEG	20190719
□	Patient	▸ R01-012^^^^	R01-012	2	CT,SEG	20180828
□	Patient	▸ R01-013^^^^	R01-013	2	CT,SEG	20190719
□	Patient	▸ R01-014^^^^	R01-014	2	CT,PT	20190719
□	Patient	▸ R01-015^^^^	R01-015	2	CT,SEG	20190719
□	Patient	▸ R01-016^^^^	R01-016	2	CT,SEG	20190719
□	Patient	▸ R01-017^^^^	R01-017	2	CT,PT	20190719
□	Patient	▸ R01-018^^^^	R01-018	2	CT,SEG	20190719
□	Patient	▸ R01-019^^^^	R01-019	2	CT,PT	20190719
□	Patient	▸ R01-020^^^^	R01-020	2	CT,PT	20190719

Fig. 2 (A) QIFP interface with the "Images" menu displayed. (B) This example shows the data sources available, with the "Local" data source selected in red. (C) Three local cohorts are available ("myNSCLCData," "mySCLCData," and "TCIA NSCLC_Radiogenomics") with the latter selected. (D) The list of patients in the TCIA NSCLC_Radiogenomics cohort is displayed in the "Image Data" section. (E) Clicking the arrow next to the cohort name allows user to upload images and/or segmentations. (F) Clicking the pencil will allow you to edit details regarding the cohort, including adding or removing other users to this cohort. (G) Clicking the triangle next to the patient name expands all available data for that patient, including studies, series, and annotations. (H) The annotation for this patient (3-D Slicer Segmentation Result) is a DICOM segmentation object. (I) Data can also be downloaded with the down arrow next to the ID or (J) deleted by clicking on the trash can next to the name. (K) Clicking the eye symbol next to an image series opens it in an ePAD image viewer within the window for viewing and potential annotation.

source are selected, revealing a list of available cohorts in the left-hand panel and the available patient images/annotations in the right-hand panel. Other available sources include: The Cancer Imaging Archive (TCIA),[22] an instance of the ePAD[21] image annotation and storage system, and either the Google or Amazon S3 cloud services [Fig. 2(B)]. QIFP can also be configured to query data from any DICOM compliant local PACS. When QIFP is running on a local server, local cohorts can be created by transferring them from the other sources to the QIFP, or by manually uploading images, segmentations, and/or annotations by clicking on the upload button next to cohort name [Fig. 2(E)]. This will allow users to select a file type to upload and browse to files on their local computer. For example, users can upload a zip file of DICOM or Neuroimaging Informatics Technology Initiative (NIfTI) images and segmentations. Finally, owners of a local cohort can add other QIFP users to the cohort to allow them to access the data and workflows associated with it [Fig. 2(F)].

Once a cohort has been selected, the QIFP shows the list of DICOM images available in that cohort in the main right-hand panel and lists information on the patient, study, and series. A link to an ePAD image viewer (a freely available open-source DICOM viewer[21]) is also provided to quickly visualize an image series [Fig. 2(K)] and annotate images. Any new annotations (e.g., segmentation seed points) created in ePAD will then be available in the QIFP. Any annotations associated with a specific series are also displayed. Users can select individual series, studies, or patients to process by selecting the box next to each one. Otherwise, users can select the whole cohort by clicking the box next to the cohort name at the top of the screen, or a subset of patients by clicking on each patient and holding down the shift key to select all patients in between. Selected cohorts can then be processed by one of many processing tools and pipelines, described in Secs. 3.5 and 3.6, respectively.

3.2 *Annotations Menu*

The "Annotations" menu lists all the available image annotations or segmentations for a given cohort. Annotation files can be stored as annotation and image markup (AIM)[23] files or DICOM segmentation objects (DSO). Users can also upload other segmentation file types, e.g., NIfTI file format as described above [Fig. 2(E)],[24] and save them to the local cohort. Only users who are members of the local cohort have access to these segmentations.

3.3 *Models Menu*

Whenever a new predictive model is created through a machine learning workflow, the user has the option to save it and include it in future workflows. The "Models" menu contains a list of previously constructed predictive models that are available in the QIFP system.

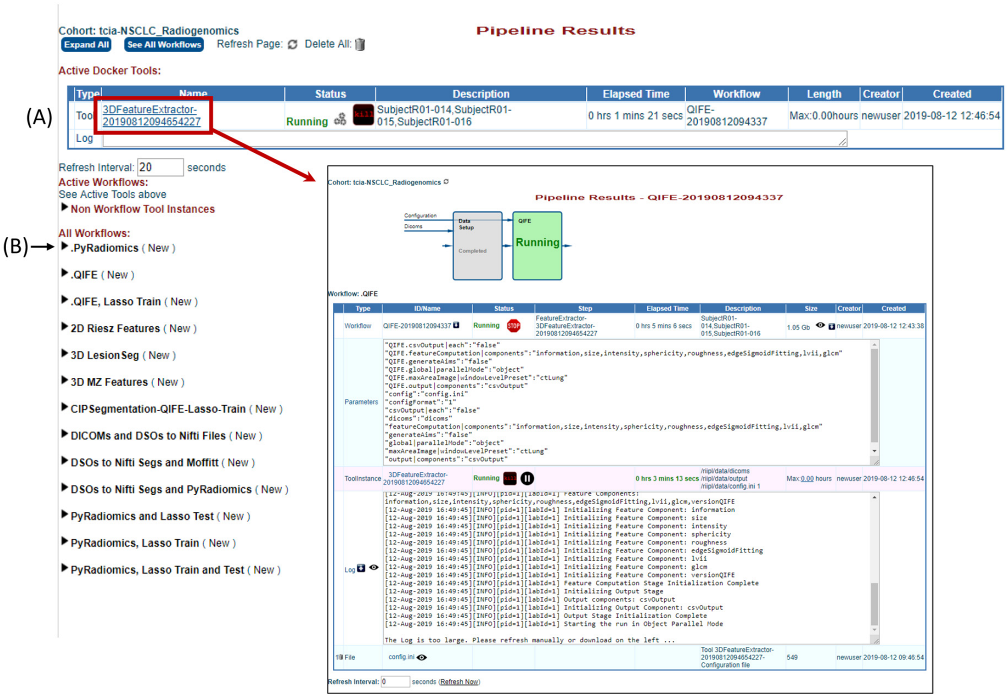
3.4 *Pipeline Results Menu*

Users can see results for past and progress for currently running workflows under the "Pipeline Results" menu. Figure 3 shows an example of a workflow in progress. The cohort name is listed on the top left corner of the page and actively running workflows are displayed under "Active Docker Tools."

While a workflow is running, the Pipeline Results page displays relevant information, such as the elapsed time, the parameters used, log-file entries, and overall status of the workflow. All workflows have an ID name which is based on the tool name (e.g., QIFE), and the date and time that the workflow was started. Users receive an email once the workflow has completed. A log file is also available to provide information on the completed workflow, if it finds an error, and why it may have failed.

3.5 *Docker Tools Menu*

The "Docker Tools" menu provides access to all available Docker tools in the system, as well as a "Tool Help." Currently, the QIFP system has a range of tools available for quantitative image



Cohort: tcia-NSCLC_Radiogenomics

Pipeline Results

Expand All | See All Workflows | Refresh Page: | Delete All:

Active Docker Tools:

Type	Name	Status	Description	Elapsed Time	Workflow	Length	Creator	Created
Tool	3DFeatureExtractor-20190812094654227	Running kill	SubjectR01-014,SubjectR01-015,SubjectR01-016	0 hrs 1 mins 21 secs	QIFE-20190812094337	Max:0.00hours	newuser	2019-08-12 12:46:54
Log								

Refresh Interval: 20 seconds

Active Workflows:
See Active Tools above
▶ Non Workflow Tool Instances

All Workflows:
▶ .PyRadiomics (New)
▶ .QIFE (New)
▶ .QIFE, Lasso Train (New)
▶ 2D Riesz Features (New)
▶ 3D LesionSeg (New)
▶ 3D MZ Features (New)
▶ CIPSegmentation-QIFE-Lasso-Train (New)
▶ DICOMs and DSOs to Nifti Files (New)
▶ DSOs to Nifti Segs and Moffitt (New)
▶ DSOs to Nifti Segs and PyRadiomics (New)
▶ PyRadiomics and Lasso Test (New)
▶ PyRadiomics, Lasso Train (New)
▶ PyRadiomics, Lasso Train and Test (New)

Cohort: tcia-NSCLC_Radiogenomics

Pipeline Results - QIFE-20190812094337

Workflow: .QIFE

Type	ID/Name	Status	Step	Elapsed Time	Description	Size	Creator	Created
Workflow	QIFE-20190812094337	Running STOP	FeatureExtractor-3DFeatureExtractor-20190812094654227	0 hrs 5 mins 6 secs	SubjectR01-014,SubjectR01-015,SubjectR01-016	1.05 Gb	newuser	2019-08-12 12:43:38
Parameters								
ToolInstance	3DFeatureExtractor-20190812094654227	Running kill		0 hrs 3 mins 13 secs	/riipl/data/dicoms /riipl/data/output /riipl/data/config.ini 1	Max:0.00 hours	newuser	2019-08-12 12:46:54
Log								
1 File	config.ini				Tool 3DFeatureExtractor-20190812094654227-Configuration file	549	newuser	2019-08-12 09:46:54

Parameters:

```
"QIFE.csvOutput|each":"false"
"QIFE.featureComputation|components":"information,size,intensity,sphericity,roughness,edgeSigmoidFitting,lvii,glcm"
"QIFE.generateAims":"false"
"QIFE.global|parallelMode":"object"
"QIFE.maxAreaImage|windowLevelPreset":"ctLung"
"QIFE.output|components":"csvOutput"
"config":"config.ini"
"configFormat":"1"
"csvOutput|each":"false"
"dicoms":"dicoms"
"featureComputation|components":"information,size,intensity,sphericity,roughness,edgeSigmoidFitting,lvii,glcm"
"generateAims":"false"
"global|parallelMode":"object"
"maxAreaImage|windowLevelPreset":"ctLung"
"output|components":"csvOutput"
```

Log:

```
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Feature Components:
information,size,intensity,sphericity,roughness,edgeSigmoidFitting,lvii,glcm,versionQIFE
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Feature Component: information
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Feature Component: size
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Feature Component: intensity
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Feature Component: sphericity
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Feature Component: roughness
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Feature Component: edgeSigmoidFitting
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Feature Component: lvii
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Feature Component: glcm
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Feature Component: versionQIFE
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Feature Computation Stage Initialization Complete
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Output Stage
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Output components: csvOutput
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Initializing Output Component: csvOutput
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Output Stage Initialization Complete
[12-Aug-2019 16:49:45][INFO][pid=1][labId=1] Starting the run in Object Parallel Mode

The Log is too large. Please refresh manually or download on the left ...
```

Refresh Interval: 0 seconds (Refresh Now)

Fig. 3 Example of the "Pipeline Results Menu" with a feature extraction workflow in progress. This menu shows all active and completed tools and workflows for the cohort selected prior to invoking this menu. (A) Currently active Docker tool (red box) and the results in progress, including status, elapsed time, parameters used, and access to a log file, pointed to by red arrow. Workflows shown below "Active Docker Tools" are organized by type; for example, clicking the arrow (B) will show all workflows using the PyRadiomics Tool completed or in progress. Clicking on the workflow name, as shown in the red box, will open a new webpage with additional details on that workflow.

analysis, including preprocessing, segmentation, feature extraction, and machine learning prediction models (Table 1). What follows provides a detailed explanation of the tools currently available on the QIFP.

3.5.1 *Preprocessing tools*

There are many different formats for images, segmentations, and annotations and not all tools can process all formats. Therefore, QIFP contains tools that can be used for file conversion. For example, there are tools to convert between image types (e.g., DICOM, NIfTI) and segmentation types (e.g., DICOM-RT, NIfTI, and DSO). There is also a tool if a user wishes to validate DICOM files to ensure all required DICOM tags are present prior to processing.[25]

3.5.2 *Segmentation*

Currently, there are several segmentation algorithms implemented as Docker tools available on the QIFP. There are two-dimensional (2-D) (2D LesionSeg) and three-dimensional (3-D) (3D LesionSeg) level set-based tumor segmentation tools, which take as input the image and a polygon or long axis line within the lesion.[26,27] There is also a Chest Imaging Platform (CIP) Lesion Segmentation tool for DICOM or NIfTI files written by the Applied Chest Imaging Laboratory (Brigham and Women's Hospital). This tool takes an input image and one or more seed points on the lesion and outputs a segmentation of the lesion of interest.[28] AIM files are required to provide these inputs and specific example templates are provided on QIFP when running the workflow.

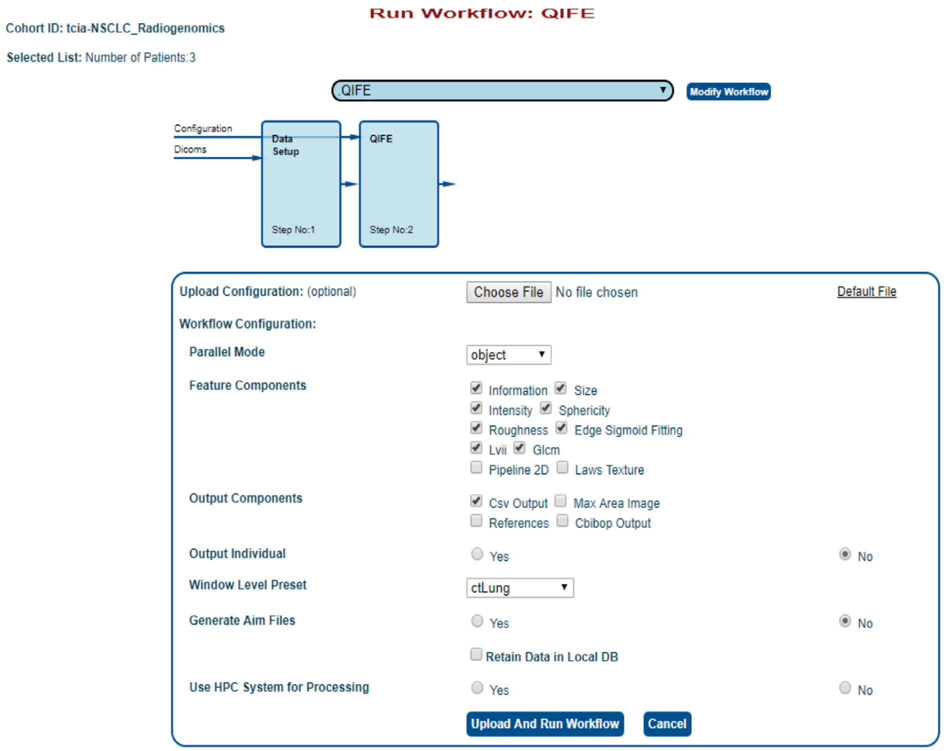
Table 1 Docker tools currently available on the QIFP.

Tool type	Tool name
Feature extraction	PyRadiomics
	2-D JJVector feature extractor
	3-D feature extractor (Mu Zhou)
	3-D feature extractor (QIFE)
	Moffitt feature extractor
	SIFT feature extractor
Machine learning prediction engines	LASSO train prediction engine
	LASSO test prediction engine
	LASSO randomization prediction engine
Preprocessing tools	Analyze segmentation to NIfTI conversion
	DICOM-RT to DSO conversion
	NIfTI to DSO conversion
	DICOMs to NIfTI conversion
	DSO to Nifiti conversion
	DICOM validation tool
Segmentation	Lung segmentation
	Tumor segmentation
	2-D lesion segmentation
	3-D lesion segmentation
	CIP DICOM 3-D segmentation
	CIP NIfTI 3-D segmentation
Other	CoLiAGe feature map
	Delta features

3.5.3 *Feature extraction*

There are several different feature extraction modules that are currently available within the QIFP system. Stanford's Quantitative Image Feature Extraction (QIFE) tool allows for the extraction of size, shape, intensity, texture, and law's features.[29] Figure 4 shows an example of how to configure a workflow containing this tool. Users can view, edit, and upload their own configuration file or manually select workflow options through the checkboxes provided. Figure 5 shows a completed workflow. All the files produced by the workflow are available for download through a link provided at the bottom of the results, including the log file, the resultant feature file in a comma-separated values (CSV) format, and the configuration file used for that run.

The PyRadiomics tool is another feature extraction engine that has the option to extract higher order wavelet features along with the traditional features on the original images.[16] Additional feature extraction tools include 2-D Riesz features[30] and scale-invariant feature



Cohort ID: tcia-NSCLC_Radiogenomics

Selected List: Number of Patients:3

Upload Configuration: (optional) No file chosen

Workflow Configuration:

Parallel Mode: object

Feature Components: ☑ Information ☑ Size ☑ Intensity ☑ Sphericity ☑ Roughness ☑ Edge Sigmoid Fitting ☑ Lvii ☑ Glcm □ Pipeline 2D □ Laws Texture

Output Components: ☑ Csv Output □ Max Area Image □ References □ Cbibop Output

Output Individual: ○ Yes ● No

Window Level Preset: ctLung

Generate Aim Files: ○ Yes ● No

□ Retain Data in Local DB

Use HPC System for Processing: ○ Yes ○ No

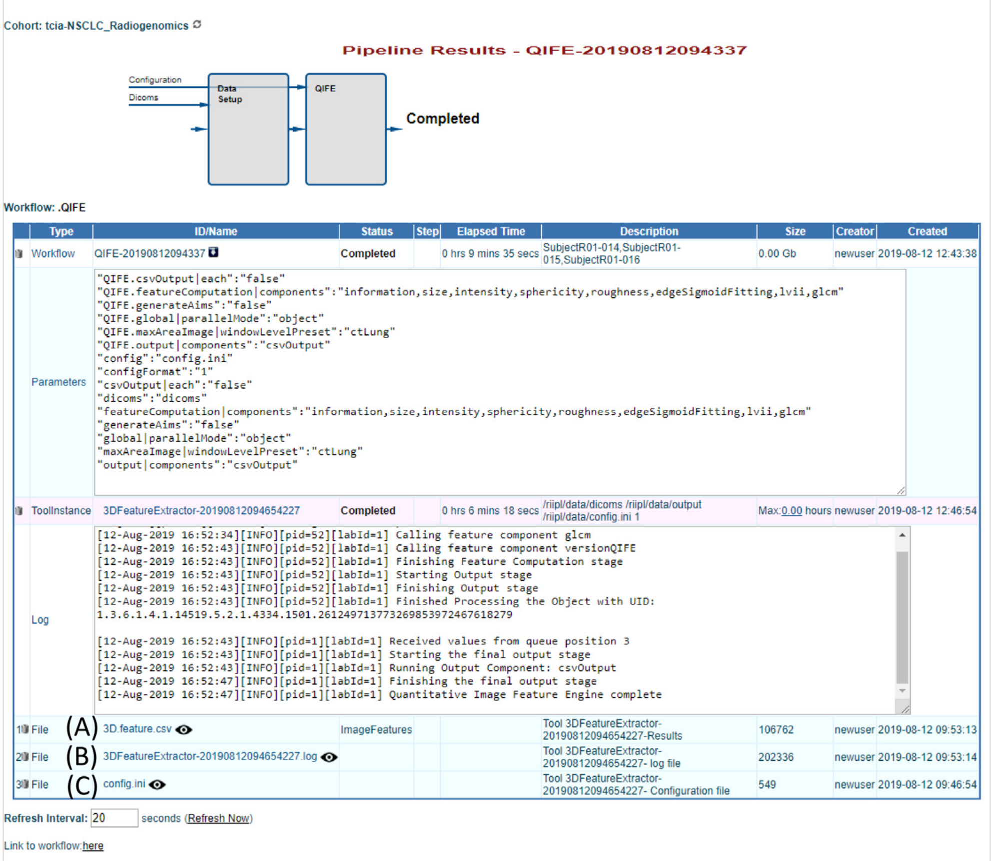
Fig. 4 Example feature extraction workflow using the QIFE. Users can upload a configuration file or manually select configuration options in the interface shown.

transform (SIFT) features.[31] In general, each feature extraction module has its own set of user-configurable parameters to ensure the workflow is configured to best suit the required data type and analysis.

Different feature extraction modules may compute radiomic features differently, and for this reason may arrive at different values for what might appear to be the same feature.[32] Common differences may be specifications for directional sampling of voxels for texture features, algorithms used for surface area calculations, and intensity discretization. We refer the user to the manuscripts describing QIFE[29] and PyRadiomics[16] for feature definitions. Also, since each Docker tool will contain a version of the tools from a specific point of time, the version code and Docker ID for each instance of the tool is recorded for each output workflow.

3.5.4 *Machine learning tools*

Machine learning tools allow for the use of radiomic features with or without clinical features to predict an outcome or clinical parameter of choice (e.g., overall survival, specific gene mutation). The QIFP contains a least absolute shrinkage and selection operator (LASSO)[33] tool, written using the open-source R software[34] (Vienna, Austria). This tool can be configured for training and/or testing classification and regression models. To run a machine learning workflow, the user must also upload the corresponding clinical data and indicate the clinical parameter that they want to predict. An example file demonstrating how the clinical data should be organized is provided when the user sets up a workflow (Fig. 6). As with the feature extraction modules, each machine learning module has its own set of user-defined configuration parameters that can be used to customize the workflow. For example, the configuration file can specify the model type (binomial, Cox, etc.), the elastic-net mixing parameter alpha (LASSO to ridge), feature standardization, and number of folds for cross-validation. Future work is ongoing to add additional classifiers, hyperparameter tuning methods, and unsupervised machine learning techniques.



Cohort: tcia-NSCLC_Radiogenomics

Pipeline Results - QIFE-20190812094337

Workflow: .QIFE

Type	ID/Name	Status	Step	Elapsed Time	Description	Size	Creator	Created
Workflow	QIFE-20190812094337	Completed		0 hrs 9 mins 35 secs	SubjectR01-014,SubjectR01-015,SubjectR01-016	0.00 Gb	newuser	2019-08-12 12:43:38

Parameters

```
"QIFE.csvOutput|each":"false"
"QIFE.featureComputation|components":"information,size,intensity,sphericity,roughness,edgeSigmoidFitting,lvii,glcm"
"QIFE.generateAims":"false"
"QIFE.global|parallelMode":"object"
"QIFE.maxAreaImage|windowLevelPreset":"ctLung"
"QIFE.output|components":"csvOutput"
"config":"config.ini"
"configFormat":"1"
"csvOutput|each":"false"
"dicoms":"dicoms"
"featureComputation|components":"information,size,intensity,sphericity,roughness,edgeSigmoidFitting,lvii,glcm"
"generateAims":"false"
"global|parallelMode":"object"
"maxAreaImage|windowLevelPreset":"ctLung"
"output|components":"csvOutput"
```

Type	ID/Name	Status	Step	Elapsed Time	Description	Size	Creator	Created
ToolInstance	3DFeatureExtractor-20190812094654227	Completed		0 hrs 6 mins 18 secs	/riipl/data/dicoms /riipl/data/output /riipl/data/config.ini 1	Max:0.00 hours	newuser	2019-08-12 12:46:54

Log

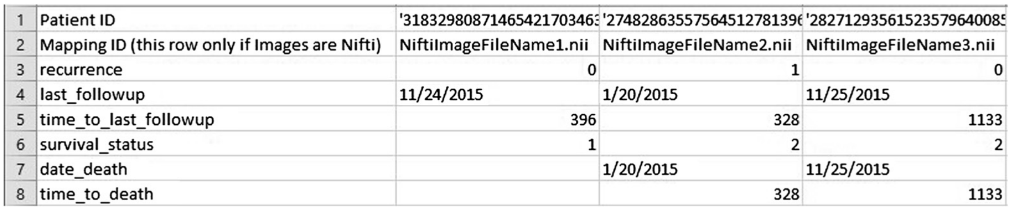
```
[12-Aug-2019 16:52:34][INFO][pid=52][labId=1] Calling feature component glcm
[12-Aug-2019 16:52:43][INFO][pid=52][labId=1] Calling feature component versionQIFE
[12-Aug-2019 16:52:43][INFO][pid=52][labId=1] Finishing Feature Computation stage
[12-Aug-2019 16:52:43][INFO][pid=52][labId=1] Starting Output stage
[12-Aug-2019 16:52:43][INFO][pid=52][labId=1] Finishing Output stage
[12-Aug-2019 16:52:43][INFO][pid=52][labId=1] Finished Processing the Object with UID:
1.3.6.1.4.1.14519.5.2.1.4334.1501.261249713773269853972467618279

[12-Aug-2019 16:52:43][INFO][pid=1][labId=1] Received values from queue position 3
[12-Aug-2019 16:52:43][INFO][pid=1][labId=1] Starting the final output stage
[12-Aug-2019 16:52:43][INFO][pid=1][labId=1] Running Output Component: csvOutput
[12-Aug-2019 16:52:47][INFO][pid=1][labId=1] Finishing the final output stage
[12-Aug-2019 16:52:47][INFO][pid=1][labId=1] Quantitative Image Feature Engine complete
```

Type	ID/Name	Status	Step	Elapsed Time	Description	Size	Creator	Created
1 File	(A) 3D.feature.csv	ImageFeatures			Tool 3DFeatureExtractor-20190812094654227-Results	106762	newuser	2019-08-12 09:53:13
2 File	(B) 3DFeatureExtractor-20190812094654227.log				Tool 3DFeatureExtractor-20190812094654227- log file	202336	newuser	2019-08-12 09:53:14
3 File	(C) config.ini				Tool 3DFeatureExtractor-20190812094654227- Configuration file	549	newuser	2019-08-12 09:46:54

Refresh Interval: 20 seconds (Refresh Now)

Link to workflow:here

Fig. 5 Example of a completed feature extraction workflow. The output components are displayed at the bottom. The files for this workflow include (A) the extracted features, (B) the log file describing the results of the workflow, and (C) the configuration file used to run the workflow. Clicking on any of the file names will allow the user to view and/or download them.



1	Patient ID	'31832980871465421703463	'27482863557564512781396	'28271293561523579640085
2	Mapping ID (this row only if Images are Nifti)	NiftiImageFileName1.nii	NiftiImageFileName2.nii	NiftiImageFileName3.nii
3	recurrence	0	1	0
4	last_followup	11/24/2015	1/20/2015	11/25/2015
5	time_to_last_followup	396	328	1133
6	survival_status	1	2	2
7	date_death		1/20/2015	11/25/2015
8	time_to_death		328	1133

Fig. 6 Example clinical file required in all LASSO training workflows.

3.6 *Workflows Menu*

This menu allows the user to visualize all currently available workflows, which are also categorized according to their type, including feature extraction, segmentation, or prediction workflows (Table 2). User can also create a new workflow from modifications of any existing workflow. Workflows have already been created to run individual or combinations of Docker tools. For example, there is a workflow to run only the PyRadiomics feature extractor and another that will first run PyRadiomics followed by a machine learning engine. Workflows can be customized to include any of the tools listed in Table 1.

3.6.1 *Creating and customizing workflows*

The simplest way to create a new workflow for existing Dockers tools on the QIFP is to run the existing "user configurable workflow." This workflow allows users to manually select the

Table 2 Workflows currently available on the QIFP.

Workflow type	Workflow descriptions
QIFE 3-D/2-D features	All workflows that include the Stanford feature extraction code (QIFE)
PyRadiomics 3-D features	All workflows that include the PyRadiomics feature extraction code
Other 3-D features	All workflows that include feature extraction code other than QIFE and PyRadiomics
2-D features	All workflows that include feature extraction code for 2-D images
Prediction	All workflows that include the LASSO prediction tools
Image conversion	All workflows that include an image and/or segmentation conversion tool
Segmentation	All workflows that contain a segmentation tool
Other workflows	All workflows that do not fall into one of the above categories (e.g., semantic features)
All workflows	All workflows available on the QIFP

Docker Tools to run, with the option to add/remove components. The configured workflow can then be saved with a new name for future use. Any existing workflow can also be customized by clicking on the "Modify Workflow" bottom beside the workflow name (an example is shown in Fig. 4).

3.7 *Other Menus*

The remaining buttons will provide access to user profile information and preferences (Profile) as well as usage statistics and event logs for completed QIFP actions (System Status/Statistics).

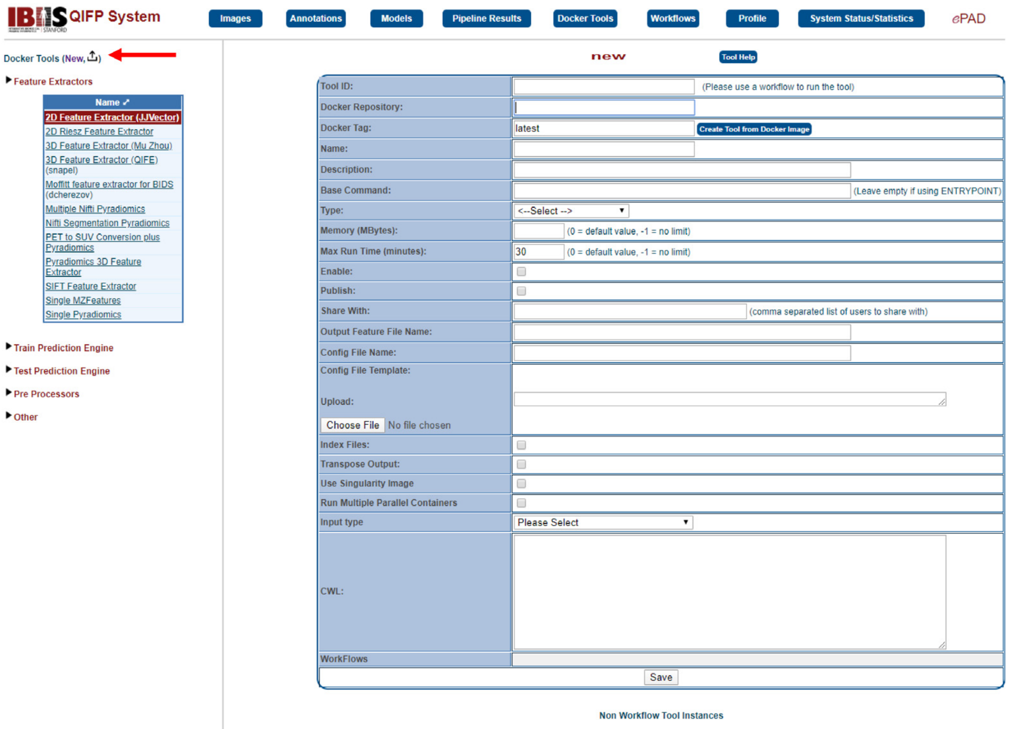
4 User-Supplied Tools

4.1 *Creating and Uploading Tools*

Users are able to upload their own tools to the QIFP by encapsulating them in a Docker container and storing them on DockerHub. For each Docker tool added to the QIFP, a Linux command should be indicated to describe the required inputs and outputs of the tool. For ease of incorporation into the QIFP system, tools can be created in two formats. The first being a tool that works on a single patient/series and a single segmentation or annotation. This type is simpler to implement, since the program does not need to figure out which segmentation refers to which series or to aggregate features/results for all patients. When using this type of tool on a whole cohort, the QIFP system will call the Docker image multiple times and run a separate Docker image for each case. The second option is to create a tool that processes multiple series/segmentations (i.e., an entire cohort). Each patient could have one or more series, and each series could have one or more segmentations/annotations. This type is more difficult to implement and requires using index files that contains file references and the feature results, if any, need to be aggregated into a single file. However, this type of tool is more efficient since only one Docker image is run for the entire data set. Figure 7 demonstrates how to upload a Docker tool and what information is required.

5 Example Workflow

This section provides a step-by-step example of how to run a workflow on the QIFP using publicly available data on the TCIA: the "NSCLC Radiogenomics" dataset[35] processed by the


IBIIS QIFP System
Images
Annotations
Models
Pipeline Results
Docker Tools
Workflows
Profile
System Status/Statistics
ePAD
Docker Tools (New,)
Feature Extractors
Name
2D Feature Extractor (JJVector)
2D Riesz Feature Extractor
3D Feature Extractor (Mu Zhou)
3D Feature Extractor (QIFE) (snapel)
Moffitt feature extractor for BIDS (dcherezov)
Multiple Nifti Pyradiomics
Nifti Segmentation Pyradiomics
PET to SUV Conversion plus Pyradiomics
Pyradiomics 3D Feature Extractor
SIFT Feature Extractor
Single MZFeatures
Single Pyradiomics
Train Prediction Engine
Test Prediction Engine
Pre Processors
Other
new
Tool Help
Create Tool from Docker Image
Choose File
Save
Non Workflow Tool Instances

Tool ID:	(Please use a workflow to run the tool)
Docker Repository:	
Docker Tag:	latest
Name:	
Description:	
Base Command:	(Leave empty if using ENTRYPOINT)
Type:	<--Select -->
Memory (MBytes):	(0 = default value, -1 = no limit)
Max Run Time (minutes):	30 (0 = default value, -1 = no limit)
Enable:	□
Publish:	□
Share With:	(comma separated list of users to share with)
Output Feature File Name:	
Config File Name:	
Config File Template: Upload: No file chosen	
Index Files:	□
Transpose Output:	□
Use Singularity Image	□
Run Multiple Parallel Containers	□
Input type	Please Select
CWL:	
WorkFlows	

Fig. 7 Example of how to add a new Docker Tool. After selecting the Docker Tool menu, click on the "New" option at the top of the left-hand (indicated by the arrow) to add the requirements for a user-supplied tool.

PyRadiomics feature extraction module, followed by an LASSO predictive modeler. The goal of this example is to predict recurrence (a binary outcome) in this cohort of lung cancer patients. The clinical information for this example can be directly downloaded from the TCIA website.[36] Users can request an account on the main login page for the QIFP.[37]

5.1 *Selecting the Cohort and Workflow*

To run a workflow, the user must first select the cohort that they would like to analyze by clicking on "Images" menu at the top, choosing the "TCIA" source, and then clicking on the "tcia: NSCLC Radiogenomics" cohort. This example will process 75 patients (R01-001 through ROI-075). To select these, click on the checkbox next to R01-001 then scroll down to R01-075 then press and hold SHIFT while clicking on the checkbox next to R01-075. Next, click on "run workflow" at the top left-hand corner of the window and select the workflow of interest, in this example "PyRadiomics 3D Features → PyRadiomics, LASSO Train" workflow (Fig. 8). This workflow will first perform feature extraction using the PyRadiomics Docker tool and then perform LASSO training to build a predictive model.

5.2 *Configuring and Running the Workflow*

Figure 9 shows what information must be provided to successfully run the workflow. Specific text is shown to the left of the block diagram with arrows indicating which of these inputs are required for each specific tool. This includes uploading a file of clinical data, which includes the outcome of interest and a link to the image patient ID. For this example, extract columns A (case ID) and AE (recurrence) from the TCIA clinical data file. All rows for cases beginning with "AMC" can be deleted, so that only "R01" cases remain. The "Case ID" header must also be renamed to "Patient ID," then these two columns can be saved as a new CSV file. This will be the clinical data file uploaded to run this workflow. In this case, the CSV file must be transposed, and the target feature "recurrence" can be selected from the drop-down menu

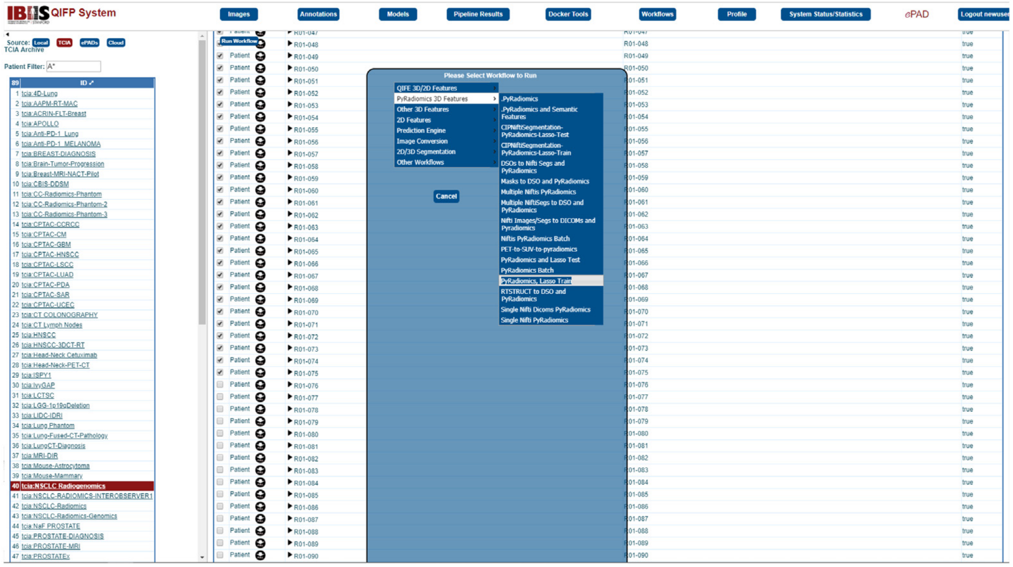
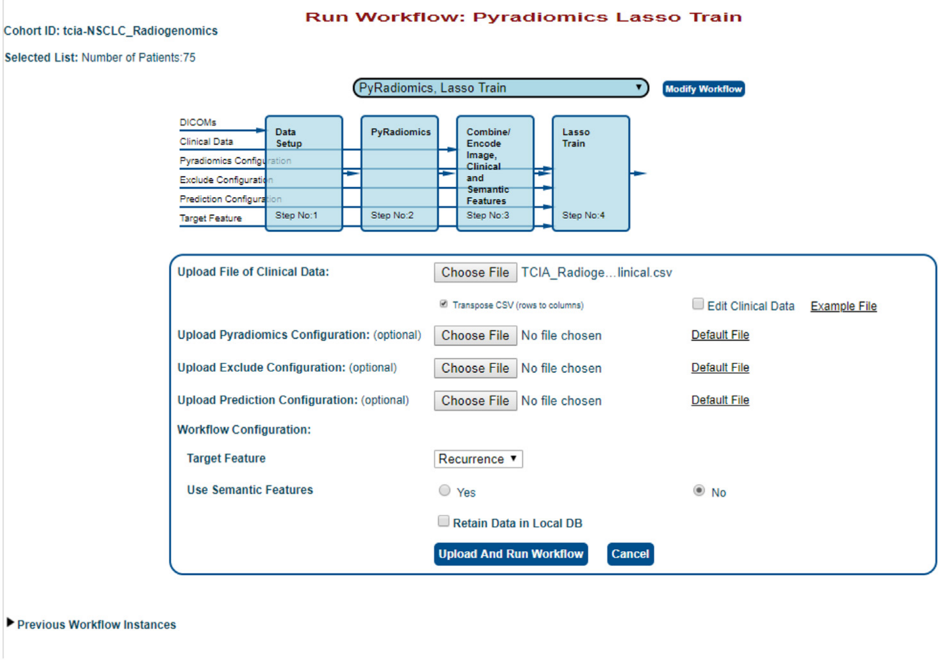


Fig. 8 Example execution of a feature extraction and machine learning training workflow using the TCIA Radiogenomics cohort.


Run Workflow: Pyradiomics Lasso Train

Cohort ID: tcia-NSCLC_Radiogenomics

Selected List: Number of Patients:75

PyRadiomics, Lasso Train

Upload File of Clinical Data: Choose File TCIA_Radioge…linical.csv

☑ Transpose CSV (rows to columns) ☐ Edit Clinical Data Example File

Upload Pyradiomics Configuration: (optional) Choose File No file chosen Default File

Upload Exclude Configuration: (optional) Choose File No file chosen Default File

Upload Prediction Configuration: (optional) Choose File No file chosen Default File

Workflow Configuration:

Target Feature: Recurrence ▾

Use Semantic Features ○ Yes ● No

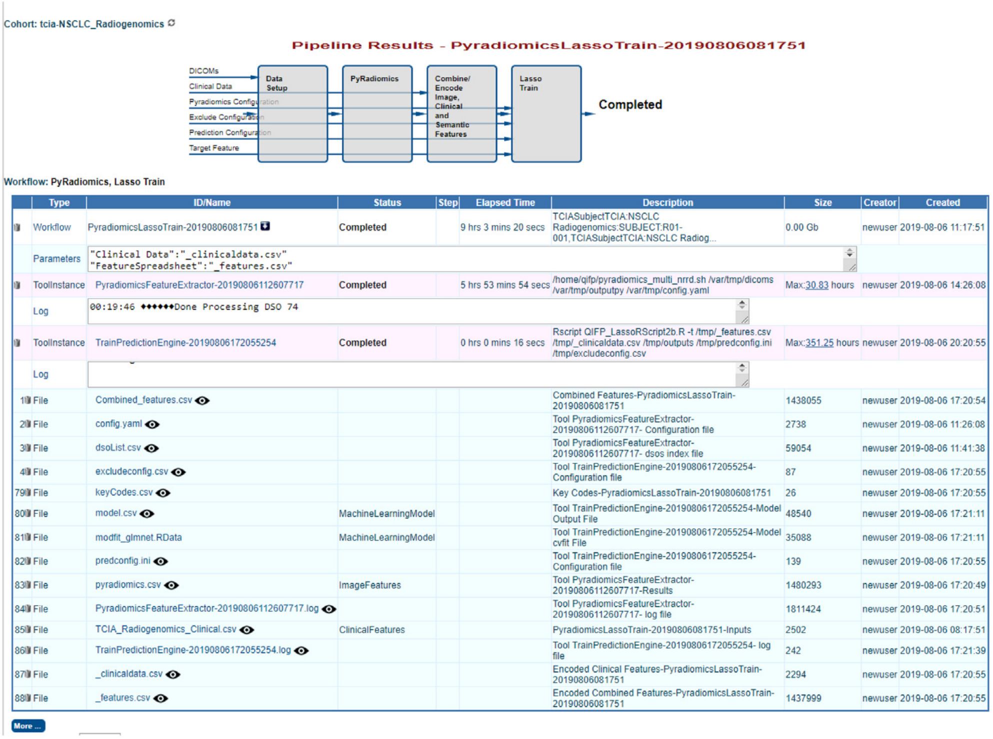
☐ Retain Data in Local DB

Fig. 9 Input screen for the feature extraction and machine learning training workflow applied to the TCIA cohort. The required input requirements are shown to the left of the block diagram. Users have the ability to upload these required files and are provided with example files to illustrate formatting and default files used.

provided. Configuration files for the feature extraction component and prediction engines can also be uploaded here; however, this example will use default configuration options. Users are also provided with output and processing options for each workflow. Selecting "retain data in local DB" will save the data in a local cohort named "tcia-NSCLC_Radiogenomics" and this will avoid having to redownload the images from TCIA for future processing. After all the selections have been made, clicking on "Upload and Run Workflow" will start the workflow. The status of the workflow can be tracked under the "Pipeline Results" menu. An email will be sent to the user once the workflow has completed with a direct link to the results of the workflow.

5.3 *Saving the Prediction Model*

The resultant model parameters can be found in the model.csv file provide as output from the training workflow (Fig. 10). In this example, LASSO selected six features for the final model. Figure 10 shows the output from the workflow, including the configuration files, feature extraction, and model results. To save the model and test it on a new cohort, users must go to the "Models" menu and click on "New" on the top of the left-hand panel. Users can name the model and select the appropriate workflow instance and tool instance from the training workflow that



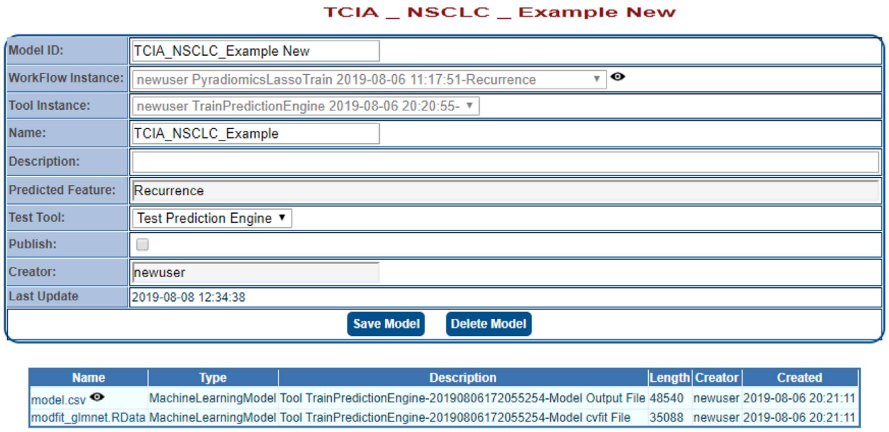
Cohort: tcia-NSCLC_Radiogenomics

Pipeline Results - PyradiomicsLassoTrain-20190806081751

Workflow: PyRadiomics, Lasso Train

	Type	ID/Name	Status	Step	Elapsed Time	Description	Size	Creator	Created
	Workflow	PyradiomicsLassoTrain-20190806081751	Completed		9 hrs 3 mins 20 secs	TCIASubjectTCIA:NSCLC Radiogenomics:SUBJECT:R01-001,TCIASubjectTCIA:NSCLC Radiog...	0.00 Gb	newuser	2019-08-06 11:17:51
	Parameters	"Clinical Data":"_clinicaldata.csv" "FeatureSpreadsheet":"_features.csv"							
	ToolInstance	PyradiomicsFeatureExtractor-20190806112607717	Completed		5 hrs 53 mins 54 secs	/home/qifp/pyradiomics_multi_nrrd.sh /var/tmp/dicoms /var/tmp/outputpy /var/tmp/config.yaml	Max:30.83 hours	newuser	2019-08-06 14:26:08
	Log	00:19:46 ♦♦♦♦♦♦Done Processing DSO 74							
	ToolInstance	TrainPredictionEngine-20190806172055254	Completed		0 hrs 0 mins 16 secs	Rscript QIFP_LassoRScript2b.R -t /tmp/_features.csv /tmp/_clinicaldata.csv /tmp/outputs /tmp/predconfig.ini /tmp/excludeconfig.csv	Max:351.25 hours	newuser	2019-08-06 20:20:55
	Log								
1	File	Combined_features.csv				Combined Features-PyradiomicsLassoTrain-20190806081751	1438055	newuser	2019-08-06 17:20:54
2	File	config.yaml				Tool PyradiomicsFeatureExtractor-20190806112607717- Configuration file	2738	newuser	2019-08-06 11:26:08
3	File	dsoList.csv				Tool PyradiomicsFeatureExtractor-20190806112607717- dsos index file	59054	newuser	2019-08-06 11:41:38
4	File	excludeconfig.csv				Tool TrainPredictionEngine-20190806172055254-Configuration file	87	newuser	2019-08-06 17:20:55
79	File	keyCodes.csv				Key Codes-PyradiomicsLassoTrain-20190806081751	26	newuser	2019-08-06 17:20:55
80	File	model.csv	MachineLearningModel			Tool TrainPredictionEngine-20190806172055254-Model Output File	48540	newuser	2019-08-06 17:21:11
81	File	modfit_glmnet.RData	MachineLearningModel			Tool TrainPredictionEngine-20190806172055254-Model cvfit File	35088	newuser	2019-08-06 17:21:11
82	File	predconfig.ini				Tool TrainPredictionEngine-20190806172055254-Configuration file	139	newuser	2019-08-06 17:20:55
83	File	pyradiomics.csv	ImageFeatures			Tool PyradiomicsFeatureExtractor-20190806112607717-Results	1480293	newuser	2019-08-06 17:20:49
84	File	PyradiomicsFeatureExtractor-20190806112607717.log				Tool PyradiomicsFeatureExtractor-20190806112607717- log file	1811424	newuser	2019-08-06 17:20:51
85	File	TCIA_Radiogenomics_Clinical.csv	ClinicalFeatures			PyradiomicsLassoTrain-20190806081751-Inputs	2502	newuser	2019-08-06 08:17:51
86	File	TrainPredictionEngine-20190806172055254.log				Tool TrainPredictionEngine-20190806172055254- log file	242	newuser	2019-08-06 17:21:39
87	File	_clinicaldata.csv				Encoded Clinical Features-PyradiomicsLassoTrain-20190806081751	2294	newuser	2019-08-06 17:20:55
88	File	_features.csv				Encoded Combined Features-PyradiomicsLassoTrain-20190806081751	1437999	newuser	2019-08-06 17:20:55

Fig. 10 Output screen for the feature extraction and machine learning training workflow applied to the TCIA cohort (Fig. 9).



TCIA _ NSCLC _ Example New

Model ID:	TCIA_NSCLC_Example New
WorkFlow Instance:	newuser PyradiomicsLassoTrain 2019-08-06 11:17:51-Recurrence
Tool Instance:	newuser TrainPredictionEngine 2019-08-06 20:20:55-
Name:	TCIA_NSCLC_Example
Description:	
Predicted Feature:	Recurrence
Test Tool:	Test Prediction Engine
Publish:	□
Creator:	newuser
Last Update	2019-08-08 12:34:38

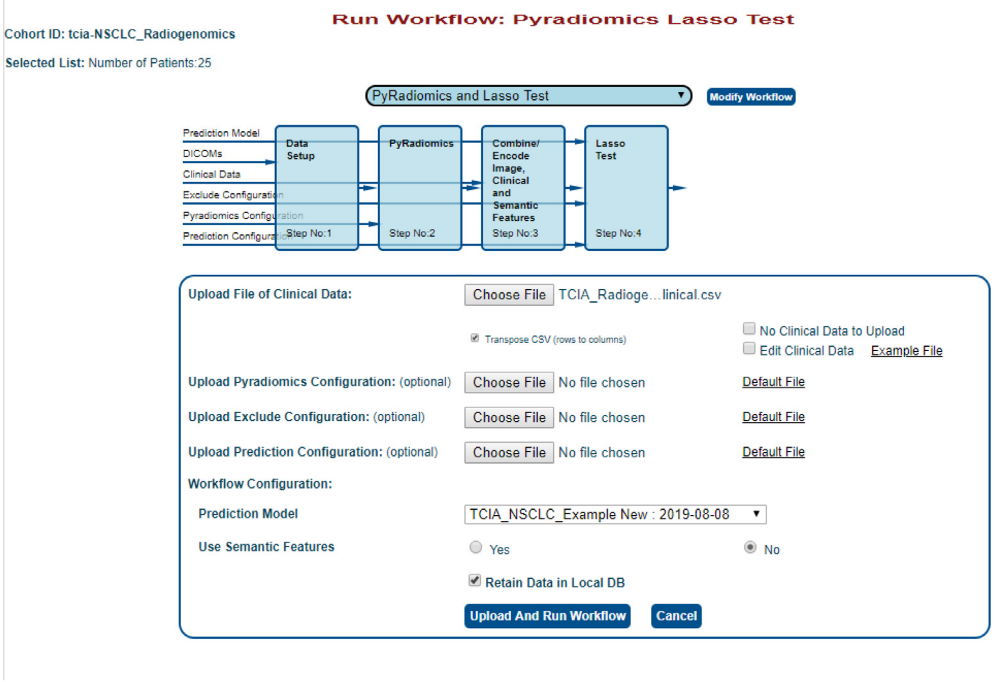
Name	Type	Description	Length	Creator	Created
model.csv	MachineLearningModel	Tool TrainPredictionEngine-20190806172055254-Model Output File	48540	newuser	2019-08-06 20:21:11
modfit_glmnet.RData	MachineLearningModel	Tool TrainPredictionEngine-20190806172055254-Model cvfit File	35088	newuser	2019-08-06 20:21:11

Fig. 11 Output screen allowing the saving of the model produced by the workflow applied to the TCIA cohort.

was just completed. Once a model is saved, it will appear under the "Models" menu on the left-hand side (Fig. 11).

5.4 *Testing the Prediction Model*

To test this model on a new cohort of patients, patients are selected as described in Sec. 5.1. For this example, testing will be done on 25 different patients from the same TCIA NSCLC radiogenomics cohort (patients R01-076 through R01-100) by running the workflow "PyRadiomics



Run Workflow: Pyradiomics Lasso Test

Cohort ID: tcia-NSCLC_Radiogenomics

Selected List: Number of Patients:25

PyRadiomics and Lasso Test ▾ | Modify Workflow

Upload File of Clinical Data: Choose File TCIA_Radioge…linical.csv

☑ Transpose CSV (rows to columns)

□ No Clinical Data to Upload

□ Edit Clinical Data Example File

Upload Pyradiomics Configuration: (optional) Choose File No file chosen Default File

Upload Exclude Configuration: (optional) Choose File No file chosen Default File

Upload Prediction Configuration: (optional) Choose File No file chosen Default File

Workflow Configuration:

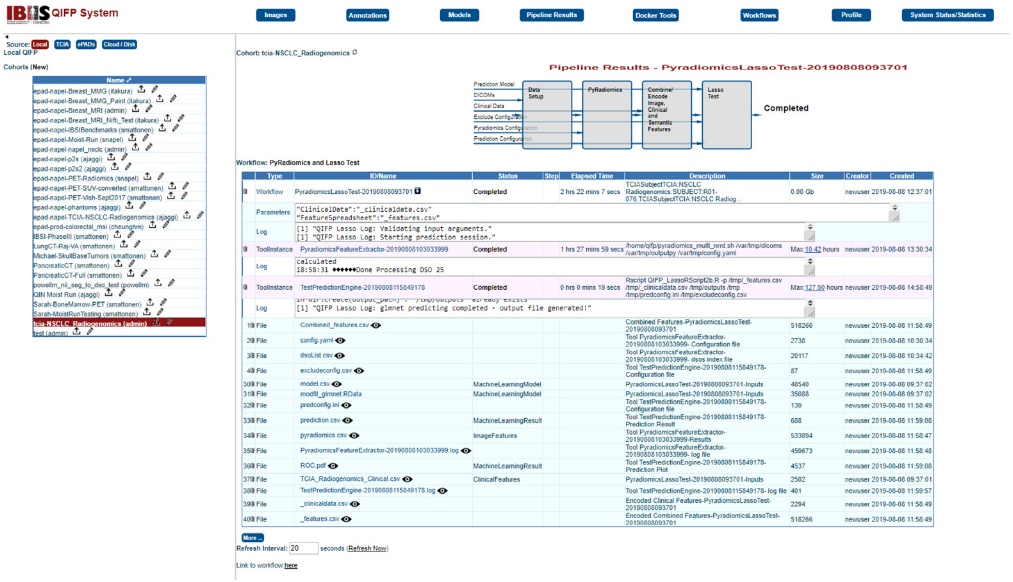
Prediction Model: TCIA_NSCLC_Example New : 2019-08-08 ▾

Use Semantic Features: ○ Yes ● No

☑ Retain Data in Local DB

Upload And Run Workflow | Cancel

Fig. 12 Input screen for the feature extraction and machine learning testing workflow applied to the TCIA cohort.



Local QIFP

Cohorts (New)

Name
epad-napel-Breast_MMG (itakura)
epad-napel-Breast_MMG_Paint (itakura)
epad-napel-Breast_MRI (admin)
epad-napel-Breast_MRI_Nifti_Test (itakura)
epad-napel-IBSIBenchmarks (smattonen)
epad-napel-Moist-Run (snapel)
epad-napel-napel_nsclc (admin)
epad-napel-p2s (ajaggi)
epad-napel-p2s2 (ajaggi)
epad-napel-PET-Radiomics (snapel)
epad-napel-PET-SUV-converted (smattonen)
epad-napel-PET-Vish-Sept2017 (smattonen)
epad-napel-phantoms (ajaggi)
epad-napel-TCIA-NSCLC-Radiogenomics (ajaggi)
epad-prod-colorectal_msi (cheunghm)
IBSI-PhaseIII (smattonen)
LungCT-Raj-VA (smattonen)
Michael-SkullBaseTumors (smattonen)
PancreaticCT (smattonen)
PancreaticCT-Full (smattonen)
powellm_nii_seg_to_dso_test (powellm)
QIN Moist Run (ajaggi)
Sarah-BoneMarrow-PET (smattonen)
Sarah-MoistRunTesting (smattonen)
tcia-NSCLC_Radiogenomics (admin)
test (admin)

Cohort: tcia-NSCLC_Radiogenomics

Pipeline Results - PyradiomicsLassoTest-20190808093701

Workflow: PyRadiomics and Lasso Test

	Type	ID/Name	Status	Step	Elapsed Time	Description	Size	Creator	Created
	Workflow	PyradiomicsLassoTest-20190808093701	Completed		2 hrs 22 mins 7 secs	TCIASubjectTCIA:NSCLC Radiogenomics:SUBJECT:R01-076,TCIASubjectTCIA:NSCLC Radiog...	0.00 Gb	newuser	2019-08-08 12:37:01
	Parameters	"ClinicalData":"_clinicaldata.csv" "FeatureSpreadsheet":"_features.csv"							
	Log	[1] "QIFP Lasso Log: Validating input arguments." [1] "QIFP Lasso Log: Starting prediction session."							
	ToolInstance	PyradiomicsFeatureExtractor-20190808103033999	Completed		1 hrs 27 mins 59 secs	/home/qifp/pyradiomics_multi_nrrd.sh /var/tmp/dicoms /var/tmp/outputpy /var/tmp/config.yaml	Max:10.42 hours	newuser	2019-08-08 13:30:34
	Log	calculated 18:58:31 ♦♦♦♦♦♦Done Processing DSO 25							
	ToolInstance	TestPredictionEngine-20190808115849178	Completed		0 hrs 0 mins 19 secs	Rscript QIFP_LassoRScript2b.R -p /tmp/_features.csv /tmp/_clinicaldata.csv /tmp/outputs /tmp /tmp/predconfig.ini /tmp/excludeconfig.csv	Max:127.50 hours	newuser	2019-08-08 14:58:49
	Log	[1] "QIFP Lasso Log: glmnet predicting completed - output file generated!"							
1	File	Combined_features.csv				Combined Features-PyradiomicsLassoTest-20190808093701	518266	newuser	2019-08-08 11:58:49
2	File	config.yaml				Tool PyradiomicsFeatureExtractor-20190808103033999- Configuration file	2738	newuser	2019-08-08 10:30:34
3	File	dsoList.csv				Tool PyradiomicsFeatureExtractor-20190808103033999- dsos index file	20117	newuser	2019-08-08 10:34:42
4	File	excludeconfig.csv				Tool TestPredictionEngine-20190808115849178-Configuration file	87	newuser	2019-08-08 11:58:49
30	File	model.csv	MachineLearningModel			PyradiomicsLassoTest-20190808093701-Inputs	48540	newuser	2019-08-08 09:37:02
31	File	modfit_glmnet.RData	MachineLearningModel			PyradiomicsLassoTest-20190808093701-Inputs	35088	newuser	2019-08-08 09:37:02
32	File	predconfig.ini				Tool TestPredictionEngine-20190808115849178-Configuration file	139	newuser	2019-08-08 11:58:49
33	File	prediction.csv	MachineLearningResult			Tool TestPredictionEngine-20190808115849178-Prediction Result	688	newuser	2019-08-08 11:59:08
34	File	pyradiomics.csv	ImageFeatures			Tool PyradiomicsFeatureExtractor-20190808103033999-Results	533894	newuser	2019-08-08 11:58:47
35	File	PyradiomicsFeatureExtractor-20190808103033999.log				Tool PyradiomicsFeatureExtractor-20190808103033999- log file	459673	newuser	2019-08-08 11:58:48
36	File	ROC.pdf	MachineLearningResult			Tool TestPredictionEngine-20190808115849178-Prediction Plot	4537	newuser	2019-08-08 11:59:08
37	File	TCIA_Radiogenomics_Clinical.csv	ClinicalFeatures			PyradiomicsLassoTest-20190808093701-Inputs	2502	newuser	2019-08-08 09:37:01
38	File	TestPredictionEngine-20190808115849178.log				Tool TestPredictionEngine-20190808115849178- log file	401	newuser	2019-08-08 11:59:57
39	File	_clinicaldata.csv				Encoded Clinical Features-PyradiomicsLassoTest-20190808093701	2294	newuser	2019-08-08 11:58:49
40	File	_features.csv				Encoded Combined Features-PyradiomicsLassoTest-20190808093701	518266	newuser	2019-08-08 11:58:49

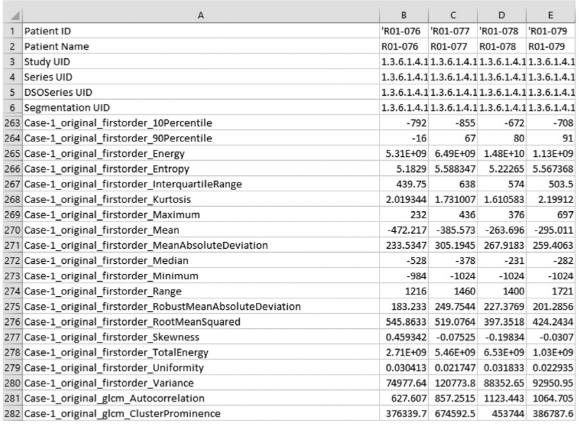
More ...

Refresh Interval: 20 seconds (Refresh Now)

Link to workflow:here

Fig. 13 Output screen for the feature extraction and machine learning testing workflow applied to the TCIA cohort (Fig. 12).

(a)



	A	B	C	D	E
1	Patient ID	'R01-076	'R01-077	'R01-078	'R01-079
2	Patient Name	R01-076	R01-077	R01-078	R01-079
3	Study UID	1.3.6.1.4.1	1.3.6.1.4.1	1.3.6.1.4.1	1.3.6.1.4.1
4	Series UID	1.3.6.1.4.1	1.3.6.1.4.1	1.3.6.1.4.1	1.3.6.1.4.1
5	DSOSeries UID	1.3.6.1.4.1	1.3.6.1.4.1	1.3.6.1.4.1	1.3.6.1.4.1
6	Segmentation UID	1.3.6.1.4.1	1.3.6.1.4.1	1.3.6.1.4.1	1.3.6.1.4.1
263	Case-1_original_firstorder_10Percentile	-792	-855	-672	-708
264	Case-1_original_firstorder_90Percentile	-16	67	80	91
265	Case-1_original_firstorder_Energy	5.31E+09	6.49E+09	1.48E+10	1.13E+09
266	Case-1_original_firstorder_Entropy	5.1829	5.588347	5.22265	5.567368
267	Case-1_original_firstorder_InterquartileRange	439.75	638	574	503.5
268	Case-1_original_firstorder_Kurtosis	2.019344	1.731007	1.610583	2.19912
269	Case-1_original_firstorder_Maximum	232	436	376	697
270	Case-1_original_firstorder_Mean	-472.217	-385.573	-263.696	-295.011
271	Case-1_original_firstorder_MeanAbsoluteDeviation	233.5347	305.1945	267.9183	259.4063
272	Case-1_original_firstorder_Median	-528	-378	-231	-282
273	Case-1_original_firstorder_Minimum	-984	-1024	-1024	-1024
274	Case-1_original_firstorder_Range	1216	1460	1400	1721
275	Case-1_original_firstorder_RobustMeanAbsoluteDeviation	183.233	249.7544	227.3769	201.2856
276	Case-1_original_firstorder_RootMeanSquared	545.8633	519.0764	397.3518	424.2434
277	Case-1_original_firstorder_Skewness	0.459342	-0.07525	-0.19834	-0.0307
278	Case-1_original_firstorder_TotalEnergy	2.71E+09	5.46E+09	6.53E+09	1.03E+09
279	Case-1_original_firstorder_Uniformity	0.030413	0.021747	0.031833	0.022935
280	Case-1_original_firstorder_Variance	74977.64	120773.8	88352.65	92950.95
281	Case-1_original_glcm_Autocorrelation	627.607	857.2515	1123.443	1064.705
282	Case-1_original_glcm_ClusterProminence	376339.7	674592.5	453744	386787.6

(b)

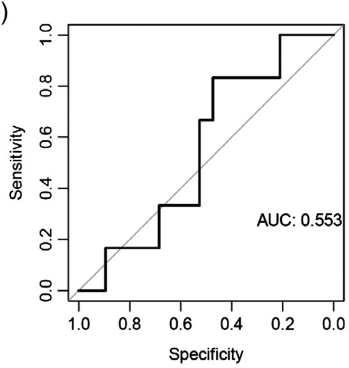


Fig. 14 Resulting features and area under the ROC curve produced by the feature extraction and machine learning testing workflow applied to the TCIA cohort (Fig. 13). (a) The output pyradiomics.csv file displays all features in the rows and all images in the columns. The first six rows identify the annotation, rows 7 to 262 contain metadata and were removed from the figure. Quantitative features start at row 263 and a subset of 20 of the 900 features is shown. (b) The area under the ROC curve is 0.55. Note that this example is for illustrative purposes only, the cohorts have not been preselected, standardized, or balanced for the outcome of interest, and the performance of the classifier has not been optimized for this dataset, including assessing performance with time-to-event analysis.

3D Features → PyRadiomics and Lasso Test." The same clinical data file can be uploaded that was used for testing, since it includes all patients in the TCIA dataset (Fig. 12). Once again, the default feature extraction and prediction configurations will be used. The model that was saved in Sec. 5.3 above can be selected from the drop-down menu and then the workflow can be started. Once the workflow has completed, the output files will be displayed (Fig. 13), including a list of the resulting model's features and their coefficients, and an area under the receiver operating characteristic (ROC) curve describing performance (Fig. 14).

6 Limitations and Future Work

Although the QIFP is equipped with several preprocessing and feature extraction tools, there is a limited number of machine learning tools available. Future work will include the addition of new feature selection methods and classifiers, including unsupervised machine learning techniques, as well as methods for hyperparameter tuning. Another limitation is that there are currently no cross-validation modules, including random sampling or bootstrapping; however, this is an area of ongoing work. There are also no deep learning tools available on the QIFP; however, since Docker easily allows for sharing algorithms, it would be relatively easy to dockize a pretrained neural net. Finally, the QIFP does not have any built-in data visualization or harmonization tools, important in quantitative imaging and, therefore, another area of future work.

7 Conclusions

The QIFP is an open-source, web-based platform that allows users to access, share, and build configurable quantitative image processing pipelines for both planar and volumetric medical images. The QIFP gives researchers the tools and infrastructure for the assessment and evaluation of new imaging biomarkers in single and multicenter clinical and virtual clinical trials. This includes performing all aspects of quantitative imaging, from segmentation to feature extraction and machine learning. The QIFP currently has 68 registered users across 18 institutions and companies in the United States, Canada, and Europe. Any researcher can request an account on the QIFP system using the link provided on the QIFP login page. A detailed user guide is also available on the QIFP website.[38]

Disclosures

Dr. Sandy Napel is on the Medical Advisory Board for Fovia Inc., a Scientific Advisor for EchoPixel Inc., and a Scientific Advisor for RADLogics Inc. There are no other potential conflicts of interest to disclose.

Acknowledgments

The authors would like to acknowledge grant funding (U01 CA187947 and U01 CA190214) from the National Institutes of Health (NIH) National Cancer Institute (NCI). They would also like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship.

References

1. R. J. Gillies, P. E. Kinahan, and H. Hricak, “Radiomics: images are more than pictures, they are data,” *Radiology* **278**(2), 563–577 (2015).
2. P. Lambin et al., “Radiomics: extracting more information from medical images using advanced feature analysis,” *Eur. J. Cancer* **48**(4), 441–446 (2012).
3. S. Napel et al., “Quantitative imaging of cancer in the postgenomic era: radio(geno)mics, deep learning, and habitats,” *Cancer* **124**(24), 4633–4649 (2018).
4. R. Li et al., *Radiomics and Radiogenomics: Technical Basis and Clinical Applications*, CRC Press, New York (2019).
5. S. S. Yip and H. J. Aerts, “Applications and limitations of radiomics,” *Phys. Med. Biol.* **61**(13), R150 (2016).
6. M. L. Giger, “Machine learning in medical imaging,” *J. Am. Coll. Radiol.* **15**(3), 512–520 (2018).
7. C. Parmar et al., “Machine learning methods for quantitative radiomic biomarkers,” *Sci. Rep.* **5**, 13087 (2015).
8. E. Sala et al., “Unravelling tumour heterogeneity using next-generation imaging: radiomics, radiogenomics, and habitat imaging,” *Clin. Radiol.* **72**(1), 3–10 (2017).
9. R. Thawani et al., “Radiomics and radiogenomics in lung cancer: a review for the clinician,” *Lung Cancer* **115**, 34–41 (2018).
10. J. Wu et al., “Radiomics and radiogenomics for precision radiotherapy,” *J. Radiat. Res.* **59**(Suppl. 1), i25–i31 (2018).
11. Y.-Q. Huang et al., “Development and validation of a radiomics nomogram for preoperative prediction of lymph node metastasis in colorectal cancer,” *J. Clin. Oncol.* **34**(18), 2157–2164 (2016).
12. H. J. Aerts et al., “Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach,” *Nat. Commun.* **5**, 4006 (2014).
13. Y. Huang et al., “Radiomics signature: a potential biomarker for the prediction of disease-free survival in early-stage (I or II) nonsmall cell lung cancer,” *Radiology* **281**(3), 947–957 (2016).
14. M. Vallières et al., “A radiomics model from joint FDG-PET and MRI texture features for the prediction of lung metastases in soft-tissue sarcomas of the extremities,” *Phys. Med. Biol.* **60**(14), 5471–5496 (2015).
15. S. A. Mattonen et al., “[18F] FDG positron emission tomography (PET) tumor and penumbra imaging features predict recurrence in non-small cell lung cancer,” *Tomography* **5**(1), 145–153 (2019).
16. J. J. Van Griethuysen et al., “Computational radiomics system to decode the radiographic phenotype,” *Cancer Res.* **77**(21), e104–e107 (2017).
17. J. Demšar et al., “Orange: data mining toolbox in Python,” *J. Mach. Learn. Res.* **14**(1), 2349–2353 (2013).
18. M. R. Berthold et al., “KNIME-the Konstanz information miner: version 2.0 and beyond,” *ACM SIGKDD Explor. Newsletter* **11**(1), 26–31 (2009).

19. C. Boettiger, “An introduction to Docker for reproducible research,” *ACM SIGOPS Oper. Syst. Rev.* **49**(1), 71–79 (2015).
20. S. Napel, D. L. Rubin, and D. Gude, “Quantitative imaging feature pipeline,” http://qifp.stanford.edu/ (accessed 3 March 2020).
21. D. L. Rubin et al., “ePAD: an image annotation and analysis platform for quantitative imaging,” *Tomography* **5**(1), 170–183 (2019).
22. K. Clark et al., “The cancer imaging archive (TCIA): maintaining and operating a public information repository,” *J. Digital Imaging* **26**(6), 1045–1057 (2013).
23. D. S. Channin et al., “The caBIG™ annotation and image markup project,” *J. Digital Imaging* **23**(2), 217–225 (2010).
24. R. Cox et al., “A (sort of) new image data format standard: NIfTI-1: WE 150,” *NeuroImage* **22**, e1440 (2004).
25. D. Clunie, “PixelMed Java DICOM Toolkit,” 2015, http://www.dclunie.com/pixelmed/software/.
26. A. Hoogi et al., “Adaptive local window for level set segmentation of CT and MRI liver lesions,” *Med. Image Anal.* **37**, 46–55 (2017).
27. A. Hoogi et al., “Adaptive estimation of active contour parameters using convolutional neural networks and texture analysis,” *IEEE Trans. Med. Imaging* **36**(3), 781–791 (2017).
28. K. Krishnan et al., “An open-source toolkit for the volumetric measurement of CT lung lesions,” *Opt. Express* **18**(14), 15256–15266 (2010).
29. S. Echegaray et al., “Quantitative image feature engine (QIFE): an open-source, modular engine for 3D quantitative feature extraction from volumetric medical images,” *J. Digital Imaging* **31**(4), 403–414 (2018).
30. A. Depeursinge et al., “Rotation–covariant texture learning using steerable Riesz wavelets,” *IEEE Trans. Image Process.* **23**(2), 898–908 (2014).
31. B. Rister, M. A. Horowitz, and D. L. Rubin, “Volumetric image registration from invariant keypoints,” *IEEE Trans. Image Process.* **26**(10), 4900–4910 (2017).
32. M. McNitt-Gray et al., “Standardization in quantitative imaging: a multi-center comparison of radiomic features from different software packages on digital reference objects and patient datasets,” *Tomography* in press (Minor revisions requested) (2020).
33. R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. R. Stat. Soc. Ser. B* **58**(1), 267–288 (1996).
34. R Core Team, “R: a language and environment for statistical computing,” R Foundation for Statistical Computing, Vienna, Austria, 2013.
35. S. Bakr et al., “A radiogenomic dataset of non-small cell lung cancer,” *Sci. Data* **5**, 180202 (2018).
36. S. Napel and S. K. Plevritis, “NSCLC Radiogenomics,” https://wiki.cancerimagingarchive.net/display/Public/NSCLC+Radiogenomics (accessed 3 March 2020).
37. S. Napel, D. L. Rubin, and D. Gude, “Quantitative imaging feature pipeline—system login,” https://qifp.stanford.edu/qifp/ (accessed 3 March 2020).
38. S. Napel, D. L. Rubin, and D. Gude, “Quantitative imaging feature pipeline—documentation,” http://qifp.stanford.edu/index.php/2017/03/16/documentation/ (accessed 3 March 2020).

Sarah A. Mattonen is an assistant professor at Western University in the Departments of Medical Biophysics and Oncology. She received her PhD in Medical Biophysics from Western University in 2016 and completed her postdoctoral training at Stanford University in the Department of Radiology in 2019. She has received funding from the Natural Sciences and Engineering Research Council of Canada. Her research interests include radiomics and machine learning for translational cancer imaging.

Dev Gude is a research software developer at Stanford University working on the Quantitative Imaging Feature Pipeline project. He received his MSEE degree from the University of Houston and BTech degree from the Indian Institute of Technology, Bombay. He has been a consultant developing software for enterprise web applications for many years.

Sebastian Echegaray received his BS and MS degrees from St. Mary's University in San Antonio Texas in 2008 and 2010, respectively, and his PhD in EE from Stanford University in 2017. He was formerly a postdoctoral fellow at Stanford, working under Dr. Sandy Napel. He is currently the head of technology at Listo Unlimited Inc., where he is leading the development of new tools and algorithms to provide better financial services to the underserved population.

Shaimaa Bakr received her BSc degree from the American University in Cairo and her MS degree in electrical engineering from Rensselaer Polytechnic Institute, advised by Professor Richard Radke. She is currently a PhD candidate at Stanford University in the Radiological Image and Information Processing Laboratory (RIIPL), led by Professor Sandy Napel.

Daniel L. Rubin received his MD degree from Stanford University in 1985 and his MS degree in biomedical informatics in 2000 and is currently a professor of biomedical data science, radiology, and medicine (biomedical informatics) and, by courtesy, of computer science and ophthalmology at Stanford University. He is director of biomedical informatics of the Stanford Cancer Institute and leads the Laboratory of Quantitative Imaging and Artificial Intelligence (QIAI), where his team develops methods and tools to integrate imaging and other nonimage data and leverage them to create applications that enable precision medicine and precision health.

Sandy Napel received his BSES degree from SUNY Stony Brook in 1974 and his MSEE and PhD degrees in EE from Stanford University in 1976 and 1981, respectively. He was formerly VP of engineering at Imatron Inc., and is currently professor of radiology and, by courtesy, of electrical engineering and medicine (Biomedical Informatics Research) at Stanford University. He coleads the Stanford Radiology 3D and Quantitative Imaging Lab and leads the Radiology Department's Division of Integrative Biomedical Imaging Informatics, where he is developing techniques for linkage of image features to molecular properties of disease.