

Quantitative Radiomics System Decoding the Tumor Phenotype

John Quackenbush and Hugo Aerts



DANA-FARBER/BRIGHAM AND WOMEN'S



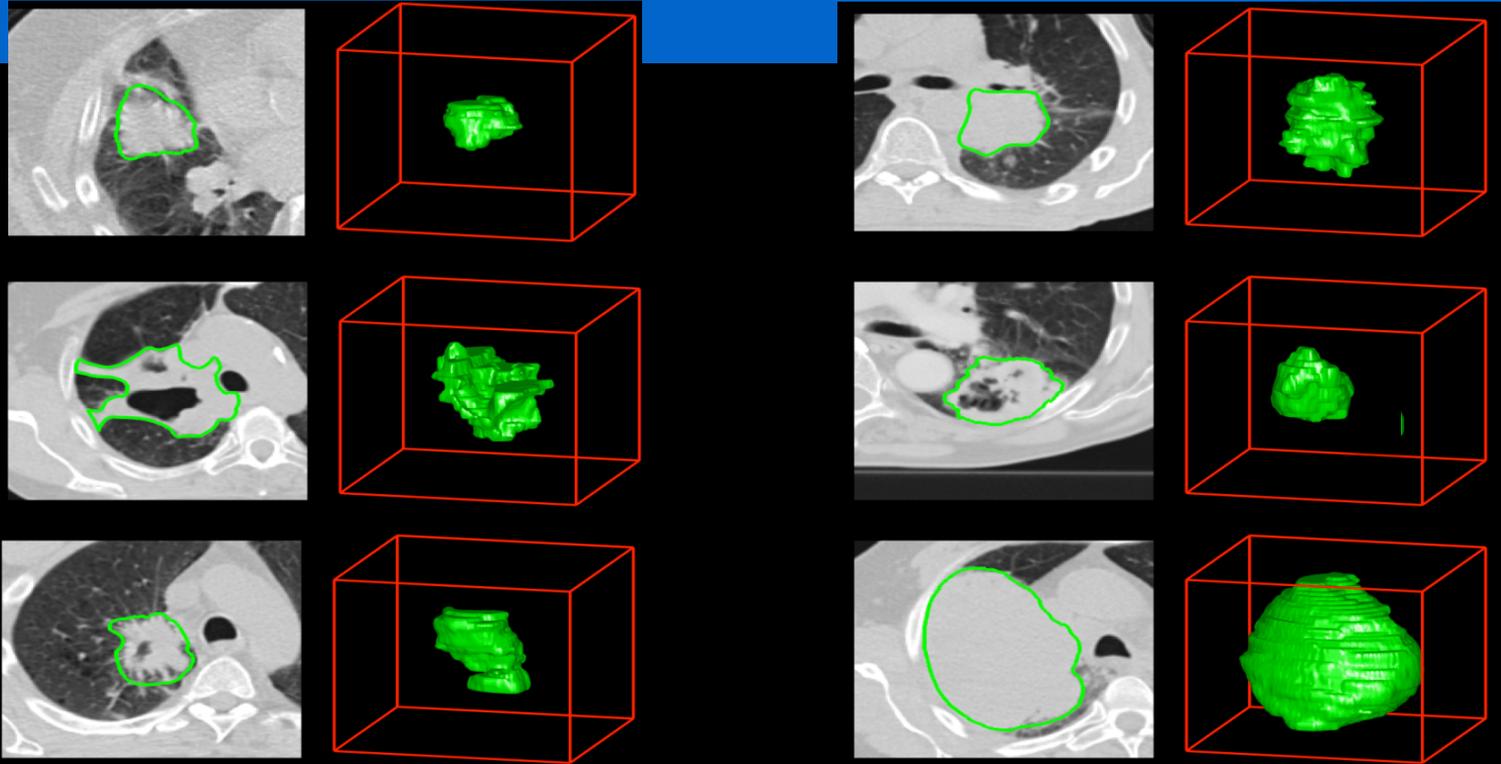
CANCER CENTER



The Radiomics Hypothesis

- **The tumor's structural phenotype reflects its molecular and clinical properties.**
- This is an extension of the clinical use of radiology in which features like volume and density are used to make predictions.
- In Radiomics, we extend this clinical use by extracting and using quantitative image features.
- We extend the use to making predictions of clinical and molecular features.

Representative CT images of lung cancer



Tumors are different

Medical imaging can capture these phenotypic differences

Advantages and Disadvantages

Advantages of Imaging:

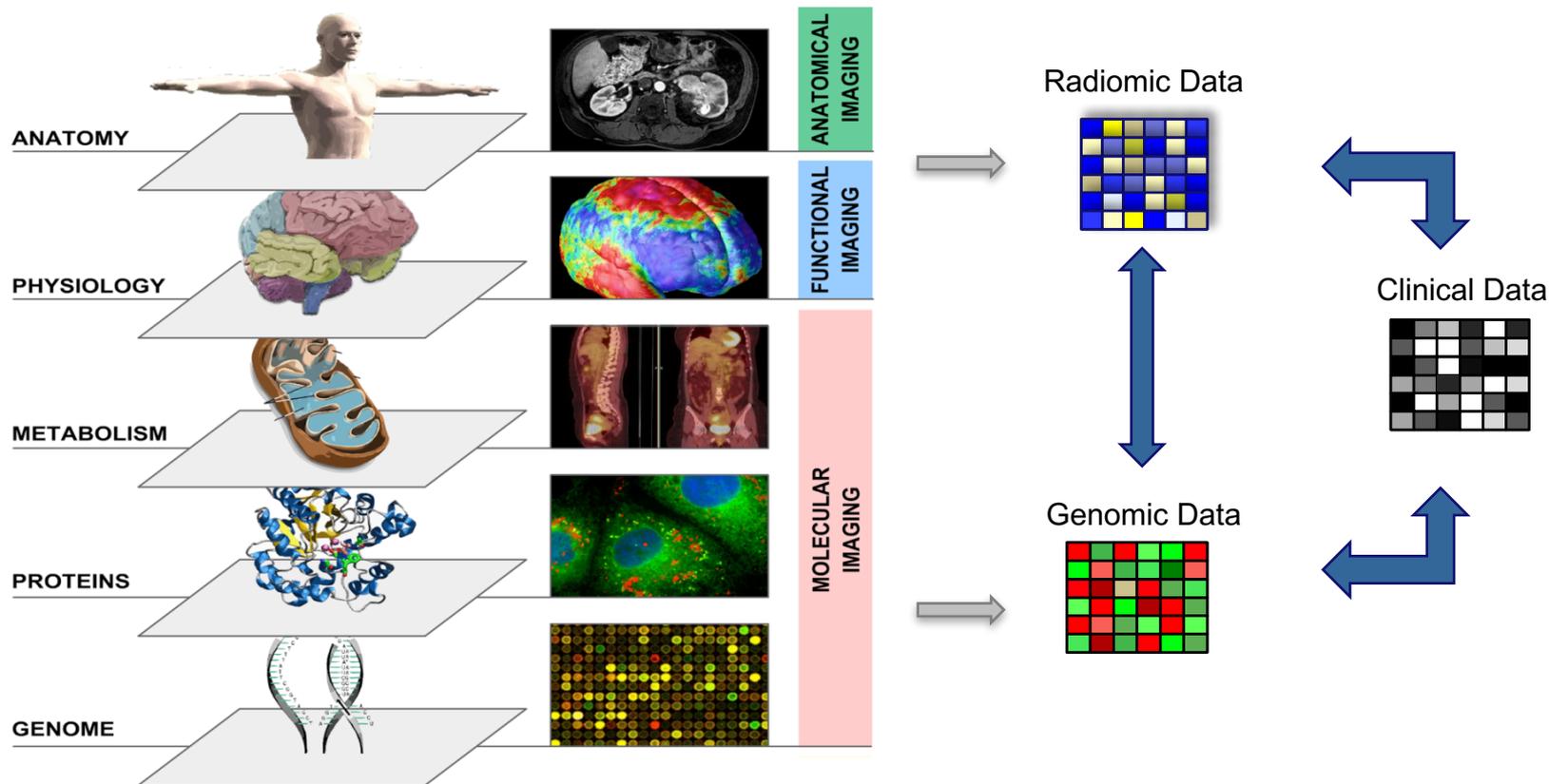
- Performed non-invasively
- Provides 3D picture of the entire cancer
- Already performed in clinical practice
- Multiple times during treatment for diagnosis, staging, radiation oncology planning, response assessment
- Captures a cancer's appearance over time and space

Advantages and Disadvantages

Disadvantages of Imaging:

- Probes the cancer at the macroscopic level
- Often qualitative not quantitative
- Very heterogeneous acquisition protocols:
 - comparisons between patients difficult
 - comparisons same patient in time difficult
- Storage of only reconstructed images (not the raw data)

Multi-level patient data



RADIOMIC TECHNOLOGIES



ENGINEERED FEATURES

Engineered features are those extracted from physician-annotated images of ROI's. They measure quantities including intensity, shape, texture and wavelet. They are then tested for stability across cohorts. Pyradiomics allows for the extracting of such features.

The Present



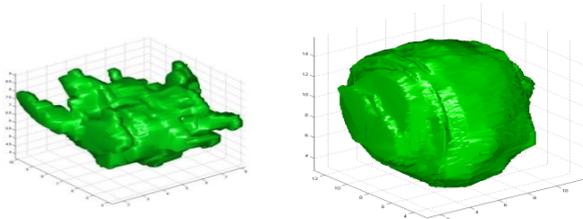
DEEP LEARNING

Deep learning features refers to feature vectors in neural nets' hidden layers. Deep radiomics involves utilizing convolutional neural nets - or convnets - in extracting these features and building prognostic models for supervised and unsupervised learning.

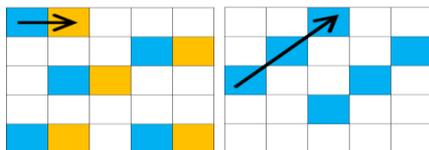
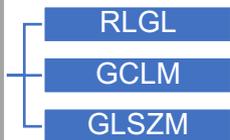
The Future

Radiomic Engineered Feature Set (current release ~1600 features)

Shape



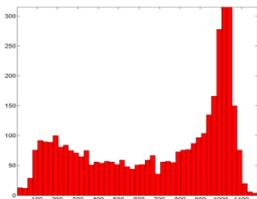
Texture



GLCM

GLRL

Stats



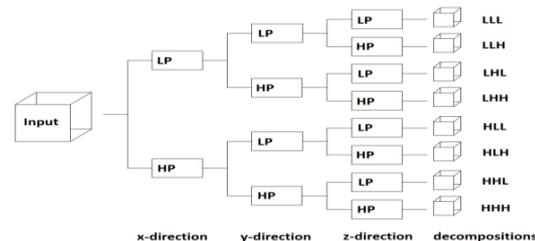
$$\text{Kurtosis: } Kur = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y \left\{ \left[\frac{I(x,y) - \mu}{\sigma} \right]^4 \right\} - 3$$

$$\text{Entropy: } H = - \sum_{i=1}^{XY} P(i) \cdot \log_2 P(i)$$

LoG



Wavelet



Radiomic features to capture tumor phenotypic details

Imaging-Genomics across cancer types



*Aerts *et al.* Nature Comm. 2014

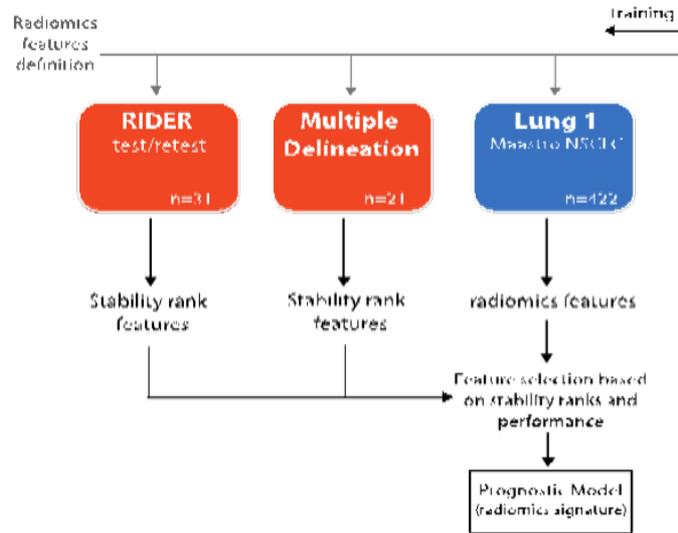
- Radiomics analysis on CT imaging of >1000 patients with Lung or H&N cancer
- Developed and validated a prognostic radiomics signature that can be applied across cancer types
- Imaging-Genomics analysis showed strong correlations between radiomics and genomics data

*Aerts *et al.* Nature Comm. 2014



This image cannot currently be displayed.

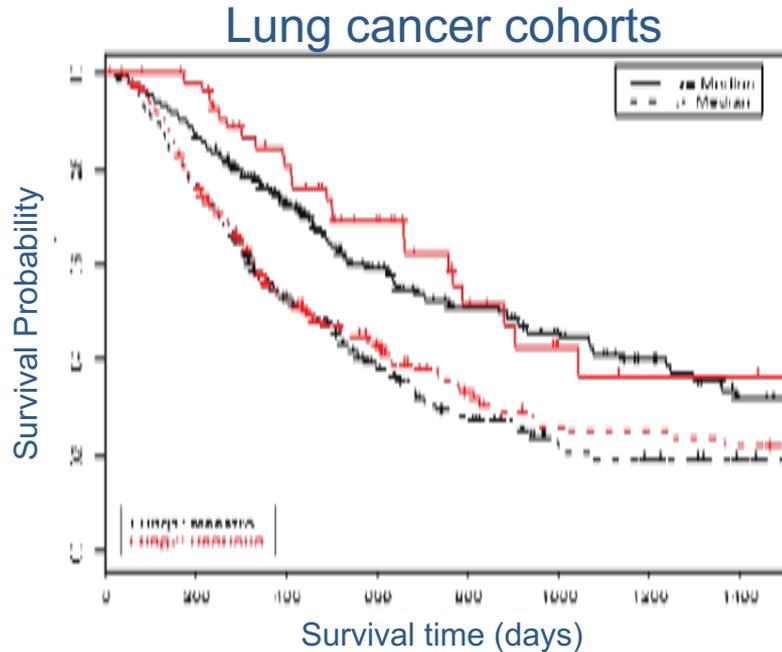
Radiomics CT Workflow



Radiomics Signature:

- 1 “Statistics Energy”
- 2 “ShapeCompactness”
- 3 “Gray Level Nonuniformity”
- 4 Wavelet “Gray Level Nonuniformity HLH”

Radiomics CT Signature Performance



Performance Model:

- CI = 0.65 on the Lung2 Validation Dataset (n=225)

Somatic Mutations & Imaging Phenotypes in Lung Adenocarcinoma

Somatic Mutations & Imaging Phenotypes in Lung Adenocarcinoma

PROFILE

(n = 213)

EGFR+ (38/17.8%)

KRAS+ (89/41.7%)

TIANJIN

(n = 257)

EGFR+ (115/45%)

KRAS+ (20/8.1%)

HARVARD - RT

(n = 162)

EGFR+ (19/11.7%)

KRAS+ (59/36.4%)

MOFFITT

(n = 131)

EGFR+ (11/8%)

KRAS+ (47/35.8%)

Mixed histologies

Mixed treatments

Diagnostic CT's

OncoMap panel

Adenocarcinomas

Tx. Surgery

Diagnostic CT's

EGFR and KRAS

Mixed histologies

Tx. Radiotherapy

RT - planning CT's

EGFR and KRAS

Mixed histologies

Tx. Surgery

Diagnostic CT's

EGFR, KRAS, P53 and
STK11

Integrative analysis in 764 patients with lung adenocarcinoma

Question: Can imaging features predict KRAS/EGFR mutational status?

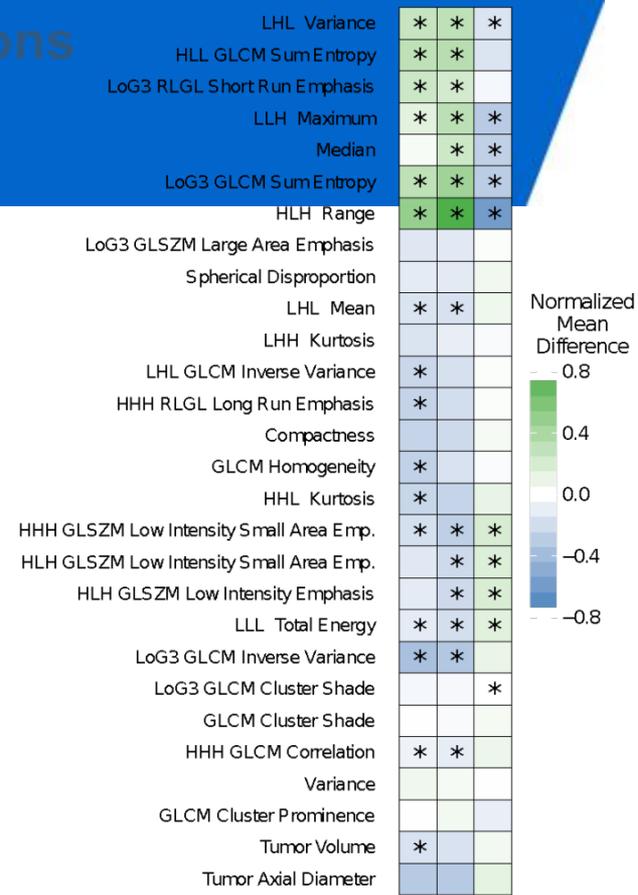
Somatic genotype - imaging phenotype associations

Approach:

- Use complete integrated dataset to incorporate diversity of genotypic and phenotypic variations across the datasets.
- *A priori*, 26 features were selected based on robustness (test-retest) and independence
- These features were evaluated for differences between mutational subtypes (FDR of 5%).

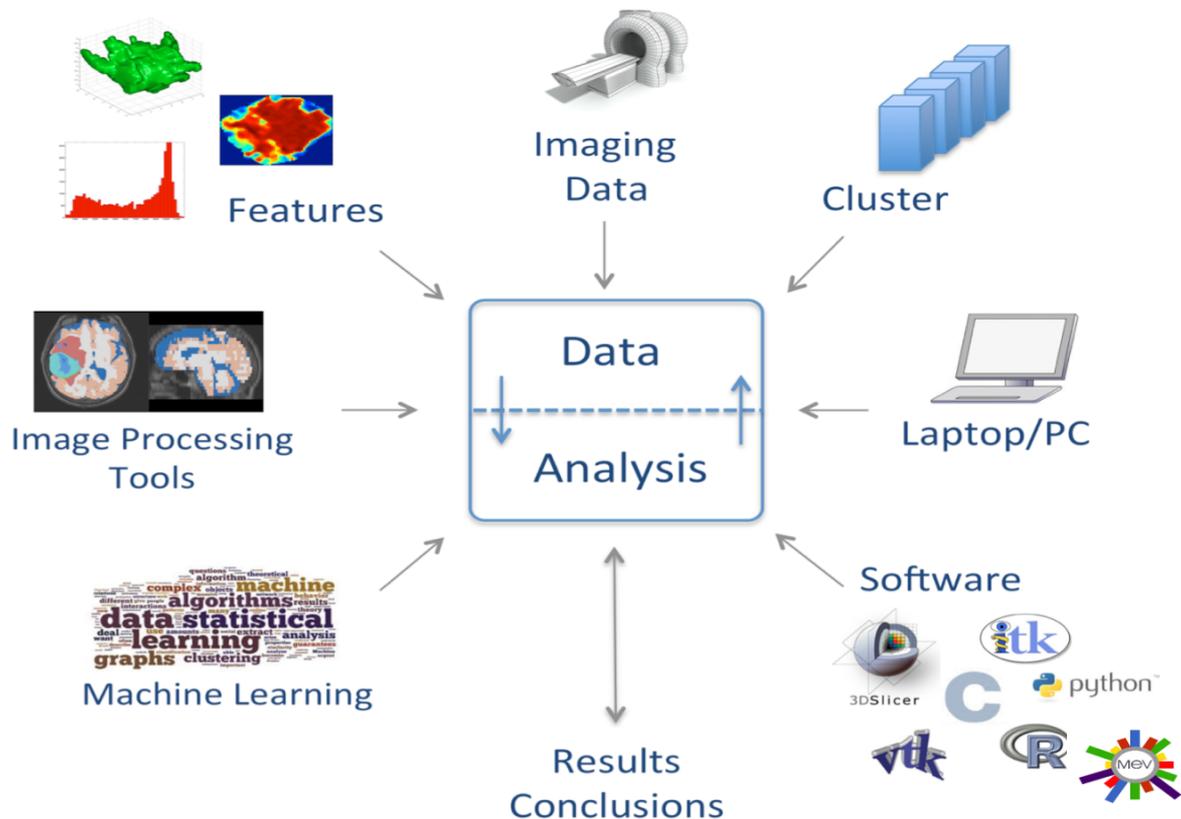
Results:

- We found 16 radiomic features to be significantly associated with EGFR mutations, and 10 features associated with KRAS mutations.
- EGFR mutated tumors has a higher radiographic heterogeneity, a smaller volume, and presenting overall lower density.
- KRAS mutated tumors were more likely to be homogeneous, with a similar size compared to non-KRAS mutated tumors.



EGFR+ vs. EGFR-
EGFR+ vs. KRAS+
KRAS+ vs. KRAS-

Radiomics Informatics Platform



[www.radiomics](http://www.radiomics.com)



RADIOMICS

[HOME](#)

[SCIENCE](#)

[CODE](#)

[DATA](#)

[TEAM](#)

Quantitative Radiographic Phenotyping

CODE

Radiomic feature extraction and analysis tools

ALL

ENGINEERED FEATURES

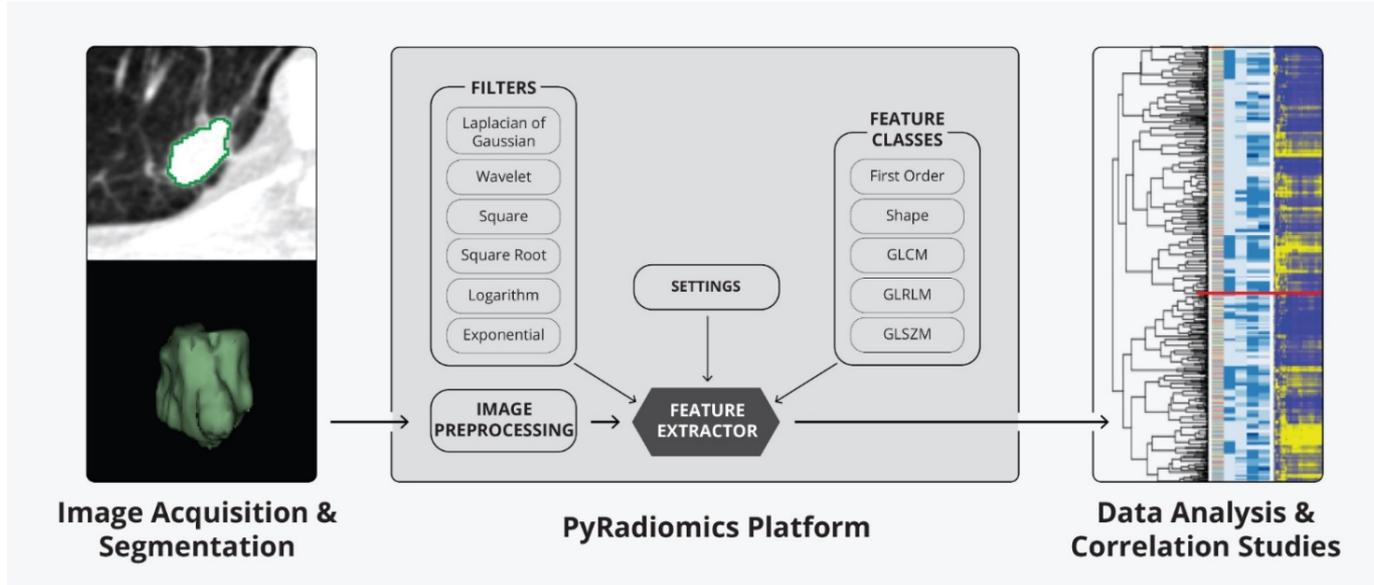
DEEP LEARNING

 python
+
∞ RADIOMICS

 3DSlicer
+
∞ RADIOMICS



PyRadiomics



Open source python package for Radiomic Feature extraction

- Aims to establish a reference standard
- increase awareness of radiomics capabilities
- Supports extraction in 2D and 3D

Extracting engineered features in python

- Easy to use
 - Does not require extensive programming expertise
- Readable source code
 - Supporting users and developers
- Modular design
 - Easy addition of feature classes / filters
 - Pipeline defined in one module for easy usage
- Extracts ~2000 engineered features per image
- Continuous Testing to ensure correct calculation

Easy customization

- All settings combined in 1 parameter file
 - Yaml structured text

```
20 setting:
21   binWidth: 25
22   label: 1
23   interpolator: 'sitkBSpline' # This is an enumerated value, here None is not allowed
24   resampledPixelSpacing: # This disables resampling, as it is interpreted as None, to enable it, specify spacing in x, y, z as [x, y, z]
25   verbose: True
26   weightingNorm: # If no value is specified, it is interpreted as None
27
28 # Input images to use: original for unfiltered image and/or any other filters, see documentation of featureextractor.py
29 # for possible values
30 inputImage:
31   Original: {} # for dictionaries / mappings, None values are not allowed, '{}' is interpreted as an empty dictionary
32
33 # Featureclasses, from which features must be calculated. If a featureclass is not mentioned, no features are calculated
34 # for that class. Otherwise, the specified features are calculated, or, if none are specified, all are calculated.
35 featureClass:
36   shape: # for lists none values are allowed, in this case, all features are enabled
37   firstorder: [] # specifying an empty list has the same effect as specifying nothing.
38   glcm:
39   glrlm:
40   glszm:
41
```

Hosted on GitHub

The screenshot shows the GitHub repository page for `Radiomics/pyradiomics`. At the top, there is a navigation bar with the repository name, a search bar, and links for Pull requests, Issues, Marketplace, and Gist. Below this, the repository name is displayed along with statistics: 15 Unwatch, 35 Unstar, and 21 Fork. A secondary navigation bar includes links for Code, Issues (10), Pull requests (2), Projects (0), Wiki, Settings, and Insights.

The main content area features the repository description: "Open-source python package for the extraction of Radiomics features from 2D and 3D images and binary masks." Below the description are tags for various topics: radiomics, cancer-imaging-research, medical-imaging, computational-imaging, nci-qin, tcia-dac, python, radiomics-features, and docker. A bar at the top of the file list shows 661 commits, 10 branches, 4 releases, and 10 contributors.

At the bottom of the screenshot, a table lists recent commits:

Commit Message	Author	Time Ago
DOCS: Update requirements in documentation	JoostJM	30 minutes ago
ENH: Add mask correction and customizable tolerance		9 days ago
DOCS: Restructure documentation on customization		9 days ago
MATH: Remove SumVariance, rename SumVariance2		a month ago
DOCS: Update requirements in documentation		30 minutes ago
BUG: Fix incorrect name for settings in feature extractor		2 days ago
Tests by building against the base Docker		2 months ago
BUG: notebook in python 3 failing		8 days ago
ENH: Define return values for 0-division errors		8 days ago
ENH: Add mask correction and customizable tolerance		9 days ago

Extensive documentation

pyradiomics
latest

Search docs

Home

- Installation
- Usage
- Pipeline Modules
- Radiomic Features
- Developers
- FAQs
- Release Notes

Welcome to pyradiomics documentation!

This is an open-source python package for the extraction of Radiomics features from medical imaging. With this package we aim to establish a reference standard for Radiomic Analysis, and provide a tested and maintained open-source platform for easy and reproducible Radiomic Feature extraction. By doing so, we hope to increase awareness of radiomic capabilities and expand the community. The platform supports both the feature extraction in 2D and 3D.

If you publish any work which uses this package, please cite the following publication: *Joost JM van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina GH Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pleper, Hugo JWL Aerts, "Computational Radiomics System to Decode the Radiographic Phenotype"; Submitted 2017*

Note

This work was supported in part by the US National Cancer Institute grant 5U24CA194354, QUANTITATIVE RADIOMICS SYSTEM DECODING THE TUMOR PHENOTYPE.

Table of Contents

- Installation
 - Get the code
 - Installation on your system
- Usage
 - Instruction Video
 - Example
 - Command Line Use
 - Interactive Use
 - PyRadiomics in 3D Slicer
 - Using feature classes directly
 - Setting Up Logging
 - Customizing the Extraction
- Pipeline Modules
 - Feature Extractor
 - Image Processing and Filters
 - General Info Module
 - Feature Class Base
 - Global Toolbox Functions
- Radiomic Features
 - First Order Features
 - Shape Features

Read the Docs v: latest

pyradiomics
latest

Search docs

Home

- Installation
- Usage
- Pipeline Modules

Radiomic Features

- First Order Features
- Shape Features
- Gray Level Co-occurrence Matrix (GLCM) Features
- Gray Level Size Zone Matrix (GLSZM) Features
- Gray Level Run Length Matrix (GLRLM) Features

Developers

- FAQs
- Release Notes

Uregestry

Tired of your current domain name registrar? Try something better. [Click here](#) to get exclusive perks.

Read the Docs v: latest

values implies more asymmetry about the mean while a lower value indicates a peak near the mean value and less variation about the mean.

`getClusterShadeFeatureValue()` [\[source\]](#)

4. Cluster Shade

$$\text{cluster shade} = \sum_{i=1}^{N_x} \sum_{j=1}^{N_x} (i+j - \mu_x(i) - \mu_y(j))^3 p(i,j)$$

Cluster Shade is a measure of the skewness and uniformity of the GLCM. A higher cluster shade implies greater asymmetry about the mean.

`getClusterTendencyFeatureValue()` [\[source\]](#)

5. Cluster Tendency

$$\text{cluster tendency} = \sum_{i=1}^{N_x} \sum_{j=1}^{N_x} (i+j - \mu_x(i) - \mu_y(j))^2 p(i,j)$$

Cluster Tendency is a measure of groupings of voxels with similar gray-level values.

`getContrastFeatureValue()` [\[source\]](#)

6. Contrast

$$\text{contrast} = \sum_{i=1}^{N_x} \sum_{j=1}^{N_x} (i-j)^2 p(i,j)$$

Contrast is a measure of the local intensity variation, favoring values away from the diagonal ($i = j$). A larger value correlates with a greater disparity in intensity values among neighboring voxels.

`getCorrelationFeatureValue()` [\[source\]](#)

7. Correlation

$$\text{correlation} = \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_x} p(i,j)ij - \mu_x(i)\mu_y(j)}{\sigma_x(i)\sigma_y(j)}$$

Correlation is a value between 0 (uncorrelated) and 1 (perfectly correlated) showing the linear dependency of gray level values to their respective voxels in the GLCM.

Note

When there is only 1 discrete gray value in the ROI (flat region), σ_x and σ_y will be 0. In this case, an arbitrary value of 1 is returned instead. This is assessed on a per-angle basis.

`getDifferenceAverageFeatureValue()` [\[source\]](#)

Slicer Extension

The screenshot displays the 3D Slicer 4.7.0-2017-01-06 interface. The main window shows a brain MRI with a segmentation overlay. The left sidebar contains the 'Parameters' panel with the following settings:

- Input Image Volume: brain1_image
- Input LabelMap: brain1_label
- Input Segmentation: None
- Features: firstorder glm shape glrjm glszm
- Toggle Features: All Features (selected), No Features

The bottom right panel shows a table of feature data:

	A	B	C	D	
1	Label	Input image type	Feature Class	Feature Name	Value
2	brain1_label_label_1	general	info	BoundingBox	(162, 84, 11, 47, 70, 7)
3	brain1_label_label_1	general	info	GeneralSettings	{'distances': [1], 'additionalInfo': True, 'enableCExtensions': True, 'force2D': False, 'interpolator': 'sitkBSpline', 'resampledPixelSpacing': None, 'label': 1, 'norm
4	brain1_label_label_1	general	info	ImageHash	5c9ce3ca174f0f8324ae4d277e0fe82dc5ac566
5	brain1_label_label_1	general	info	ImageCaption	{}

The bottom left panel shows the 'Python Interactor' with the following code:

```
>>> starting storescp process
(u'Starting E:/Builds/Slicer2013R/Slicer-build/./DCMTK-build/bin/Release/storescp.exe with ', ['5778', '--accept-all', '--output-directory', u'C:\\Users\\Joost\\Documents\\SlicerDICOMDatabase\\incoming', '--exec-sync', '--exec-on-reception', u'E:/Builds/Slicer2013R/Slicer-build/./DCMTK-build/bin/Release/dcmdump.exe --load-short --print-short --print-filename --search PatientName "C:\\Users\\Joost\\Documents\\SlicerDICOMDatabase\\incoming/#?*"'])
process E:/Builds/Slicer2013R/Slicer-build/./DCMTK-build/bin/Release/storescp.exe now in state Starting
```

Thank you for your attention

Visit us at

www.radiomics.io

Look at our code at

github.com/radiomics

Watch the movie at

www.youtube.com/watch?v=oN-qZx8VjQc