Texture — User Guide

Local Image Features Extraction
— LIFEx —

C. Nioche, F. Orlhac, I. Buvat

Part I
Protocol description

Chapter 1

How to perform textural analysis and estimate conventional parameters? – step by step –

1.1 Prerequisite / Current knowledge / IBSI

1.1.1 Prerequisite

Textural analysis and measurement of conventional parameters from images are based on a number of rules related to the image series, ROI and pixel values, as described below:

Image series

- Textural analysis is currently supported for PET, MR, CT, US and NM. Calculations for Computed radiography (CR), Digital Radiography (DX), Radio Fluoroscopy (RF), Slide Microscopy (SM) and Optical Images are available only if the images are in TIFF or JPG formats;

- Texture indices are calculated in a ROI for one or several layers (1 layer = 1 image series). To properly interpret the index values, it is important to know that the index are calculated for the selected image series (active layer) and, by default, for the voxel sampling of the selected image series.

ROI:

- The ROI should include at least 64 voxels for 3D ROI and 16 voxels for 2D ROI;

- The ROI should be a single piece (contiguous voxels). It is possible to have a region including several clusters of voxels, in which case only the largest cluster is used for calculating textural indices;

- The ROI definition is an important step, as edges can largely affect textural index values as well as conventional index values (Orlhac et al 2017).

- Using scripts may produce numerous ROIs. To check that the resulting ROI are compatible with texture index calculation and before running such calculations that can be time-consuming, you can use the CheckTex button after the ROI creation and before proceeding with textural index calculation. This will check whether the ROI includes a single cluster and contain a number of voxels greater than that required for consistent textural feature calculation. If one of these two conditions is not met, a warning message will be displayed.

Input parameters:

- Number of grey levels: this is the number of grey levels (must be >0) used to resample the ROI content. Can typically be 16, 32, 64, 128. The initial pixel values will be resampled into that number of grey levels;

- Size of bin: the width of bin;

- Resampling: can be absolute (between fixed min and max values regardless of the ROI content) or relative (between the min and max values in the ROI); See Orlhac et al Plos One 2015 for the impact of resampling approach onto tissue classification;

- Min, max bound: to be set for absolute resampling only: min and max bounds define the range of resampled values.

1.1.2 Current knowledge

In PET images. By default, texture matrices are computed after a resampling step with 64 discrete values, between 0 and 20 SUV units, without spatial resampling [Orlhac 2015]. Six texture indices have been shown to be robust to segmentation and are relatively independent one from another:

- GLCM_Homogeneity (from GLCM),

- GLCM_Entropy (from GLCM),

- GLRLM_Short-Run Emphasis (GLRLM_SRE, from GLRLM),

- GLRLM_Long-Run Emphasis (GLRLM_LRE, from GLRLM),

- GLZLM_Low Gray-level Zone Emphasis (GLZLM_LGZE, from GLZLM),

- GLZLM_High Gray-level Zone Emphasis (GLZLM_HGZE, from GLZLM).

Other indices. A set of 32 texture indices, 4 histogram indices, 3 shape indices and 4 conventional PET features are computed. For the resampling step, the user can choose the number of discrete values (default: 64) and the range (default: 0-20 SUV units) and the spatial resampling (default: dimension of voxel in the images).

Figure 1.1: Texture feature extract (Advanced settings of First Order features - Morphological - IntensityBased)

Figure 1.2: Texture feature extract (Advanced settings of First Order features - LocalIntensityBased - IntensityHistogram)

1.1 Prerequisite / Current knowledge / IBSI

Figure 1.3: Texture feature extract (Advanced settings of First Order features - LocalIntensityHistrogram)

Figure 1.4: Texture feature extract (Advanced settings of Second Order features - IntensityBasedRIM)

Figure 1.5: Texture feature extract (Advanced settings of Second Order features - GLCM - GLRM)

1.1.3 The image biomarker standardisation initiative (IBSI)

LIFEx participates in the image biomarker standardisation initiative. The primary aim of this initiative is to improve reproducibility of quantitative image analyses. LIFEx has been validated on the digital phantom provided by IBSI in nifti format for direct benchmarking of feature values, without image processing. This phantom has been designed specifically for the purpose of benchmarking software for radiomic feature calculation.
(see www.cancerdata.org/resource/doi:10.17195/candat.2016.08.1)

Figure 1.6: Texture feature extract (Advanced settings of Second Order features - NGTDM - GLSZM)

1.2 Step by step

This section describes the steps needed to get textural and conventional index estimates.

1. Reading of the DICOM images for which index should be calculated (link);

2. Drawing or loading the ROI (link);

3. Definition of the input parameters: default values can be changed, and parameters without default values should be set;

4. Definition of the output parameters of interest: index to be calculated should be selected;

5. Running the calculation: click RUN in the textural index setting window;

6. The folder containing the csv file listing the results is open.

1.3 Graphic User Interface of Texture protocol
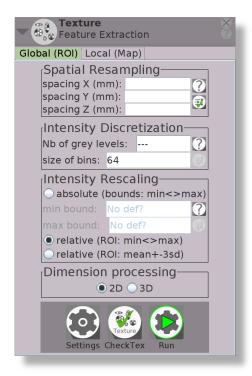
This section describes how to get textural and conventional index estimates.

The user interface (see figure 1.7 p.12) includes 3 sub-parts:

1. The "Spatial Resampling" and "Intensity Resampling" and "Intensity Discretization"

2. A panel to set advanced parameters (see figure **??** p.**??**)

3. The "Run" button to start the calculations

1.4 Intensity quantization and current limits

In PET, several studies have already investigated the impact of resampling settings. Default values for number of grey levels, bin width, resampling approach, and bounds, are set based on these studies.
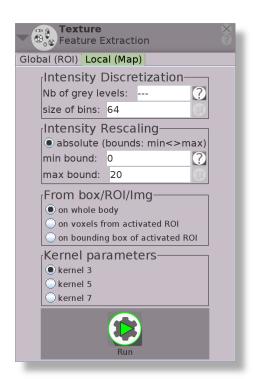
11

Figure 1.7: Display of textural GUI

In MR, CT, US, NM, CR, DX, RF, there is a lack of studies regarding the resampling settings. Studies are on-going to define robust parameter values. Meanwhile, users should interpret the results with caution.

Three types of intensity resampling are available:

Absolute resampling bounds: min⟺max   ROI values are rescaled between the *min* and *max* values as set in the parameter box. These values should be set in the appropriate units depending on the image type (Hounsfield units, SUV, #).
All voxels are accounted for in the texture index calculation.

Relative resampling of ROI values: min⟺max   ROI values are automatically rescaled between the *min* and *max* values of the ROI content. All voxels are accounted for in the texture index calculation. Voxels with an initial value less than *min* are set to *min*. Voxels with an initial value greater than *max* are set to *max*.

Relative resampling of ROI values: mean-3*Sd⟺mean+3*Sd   ROI values are automatically rescaled between mean-3*Sd⟺mean+3*Sd of the ROI content, where mean and Sd are the mean and standard deviation of the voxels included in the ROI. All voxels are accounted for in the texture index calculation. Voxels with an initial value less than $mean - 3 * Sd$ are set to $mean - 3 * Sd$. Voxels with an initial value greater than $mean + 3 * Sd$ are set to $mean + 3 * Sd$.

Summary regarding intensity resampling

- if relative min-max intensity resampling is selected:

  - boundMin = min value of ROI
  - boundMax = max value of ROI
  - binSize OR nbGreyLevels to be set

- if relative mean±3sd intensity resampling is selected:

  - boundMin = mean value of ROI - 3 sd of ROI
  - boundMax = mean value of ROI + 3 sd of ROI
  - binSize OR nbGreyLevels to be set

- if absolute intensity resampling is selected:

  - boundMin = get min text input
  - boundMax = get max text input
  - binSize OR/AND nbGreyLevels to be set

calculation of intensity resampling:

- if binSize set and nbGreyLevels not set:
  nbGreyLevels = ((boundMax - boundMin) / binSize) + 1

- if nbGreyLevels set and binSize not set:
  binSize = (boundMax - boundMin) / (nbGreyLevels - 1)

- discretizationValueOfVoxel = Floor ((nbGreyLevels * (valueOfVoxel - boundMin) / (boundMax - boundMin)) + 1);

Recommendation   In PET, we do not recommend to use the relative resampling. As shown in [Orlhac 2015], relative resampling results in a high correlation between many texture indices (TI) and the metabolic volume, that is mostly removed when using absolute resampling. Also, changes in TI can be counter-intuitive when using relative resampling instead of absolute resampling [Orlhac 2017]. In PET, resampling using 64 bins between 0 and 20 SUV units (e.g., size bin equal to 0.3) most often yields robust results (0-20 being a frequent range of SUV in oncology with FDG), although the maximum value may need to be changed in case of uptake actually higher than SUV=20.

1.5   Spatial resampling and current limits

Textural feature values can be significantly affected by the voxel size (Orlhac et al J Nucl Med 2016, Shafiq-Ul-Hassan et al Med Phys 2017). The voxel size to be used to calculate the textural feature values can be set in the "spacing X, Y, Z" fields in the textural feature dialog box. Unit is millimeter.

To avoid any memory (RAM) saturation, the values should be set in the range defined by 0.2*Min (Xspacing, Yspacing, Zspacing) (lower limit) and 5*Max (Xspacing, Yspacing, Zspacing) (upper limit). If this is not appropriate for you, please do not hesitate to contact us at contact@lifexsoft.org. In case of RAM saturation, an error message will be displayed: the voxel size in which the value should be performed should then be increased, or the RAM should be increased and the LIFEx application restarted.

1.5 Spatial resampling and current limits

A polynomial of degree 5 (Lagrangian polynomial) is used for interpolating the 3D voxel value within the ROI and performing the textural feature calculation at the requested voxel size. The original min and max values are restored after interpolation so that the interpolation does not change these extreme values.

Images acquired with an orientation matrix (image orientation patient (IOP)) different from the identity matrix are reoriented in the Dicom reference coordinate system. The voxel size (including the slice thickness) can vary according to the tilt of the images and no longer corresponds to the native voxel size.

1.6 Calculation and Results

2D/3D processing: It is possible to set how radiomic features are calculated: either in 2D on each 2D transaxial slice, or in 3D. Radiomic feature calculations in 2D can be performed when using 3D ROI. If the selection is 2D, the results will be identical to those executed if the region was divided into as many 2D ROI as there are axial sections. In the resulting file, there will be one row of results for each axial slice. A column is added in the result array named "z location (2D only)" and contains the position of the analyzed slice (in voxel unit).

A column is added in the result table named "z location (2D only)" and contains the position of the analyzed slice (in voxel unit). This feature is also available using scripts with the addition of the following lines:

```
# dimension calculation (3D is default) [ 2D || 3D ]
LIFEx.texture.DimensionProcessing=2D
```

Calculation: Calculation starts when the RUN button is clicked. All ROI selected on the reference image series are used for calculation.

Error messages: Error messages will be displayed if the prerequisites for calculation are not met. Once the prerequisites are met, calculations should run smoothly. If not, please contact@lifexsoft.org.

Results ("patient" file): Results of textural and conventional index are stored in an comma separated values (csv) filed named by default:
`patientName_patientID_Texture_time.csv` ; the `time` variable is an approximate estimate of date and time and is different for each created file (atomic variable).

Open a CSV file in Excel: Excel reads CSV files by default but in most cases when you open a CSV file in Excel, you see scrambled data that's impossible to read.

- open a new Excel document and navigate to the Data tab;

- click "From Text";

- navigate to the CSV file you wish to open and click "Import";

- from the newly-opened window, choose "Delimited". Then click "Next";

- check the box next to the type of delimiter - comma. Then click "Next";

- Click "Finish" ; That's it; you have just imported a CSV file to Excel!

The Result file is always created is the folder `workdir/patientName/RoiValue`.

Results (session file): A session file makes it possible to add columns in a result file for multi-patients study. It also makes it possible to close the application without loosing any result. For each new calculation, the application will update the session file if you check that option on the settings (advanced parameters) frame.

If the session file does not exist yet, it is automatically created in the `workdir` folder (see LIFEx_UserGuide). The folder including the session file corresponds to the folders in which all patient folders are created. The directory `workdir` can be changed in the LIFEx preferences window (see Bottom toolbar p.27 in LIFEx UserGuide).

15

1.6 Calculation and Results

You can create several session files, by modifying the file name. Only the `TextureSession.csv` is created by the application if it does not exist. If the file already exists, new results are appended to this file.

Cautious: updating a previous csv file with additional index values is relevant only if the previously calculated index values are unchanged. For instance, file updating can be relevant for multi-patient studies, or multi-ROI studies, in which always the same indices are calculated.

1.7　Scripting series

Scripting procedure for texture calculation without user interaction

1.7.1　Rationale

When you have a large number of ROI and/or a large number of patients (`session` in script) for which you want to calculate usual indices (SUV, Volume, ...) and/or histogram-based or textural features, it can be convenient to run a script that will do all the calculations for you without any user interaction. ROI have to be prepared before-hand. You can then run the script many times if you want to perform the calculations using different sets of parameters.

For instance: You have 10 ROI per patient and a set of 100 patients to be processed. You are interested in studying the impact of the *nbGreyLevels* parameter on the 10*100 ROI, by setting this parameter to 64, 128 and 256. You can create three script files that will all be identical except for the line that sets the *nbGreyLevels* parameter. Running the script will calculate all indices for you automatically and store them in csv files. You will be able to run all three scripts using a single command line, see the « Can I run several script files at once? » p.20.

Another example: You have 10 ROI per patient and a set of 100 patients to be processed. You are interested in studying the impact of the *spatial resampling* parameter on the 10*100 ROI, by setting this parameter to 2x2x2 mm and to 4x4x4 mm. You can write and run a first script by setting the *spatial resampling* parameter to 2x2x2 mm. Then duplicate the script and change 2x2x2 mm by 4x4x4 mm to produce a second script to be run. Running the two scripts will calculate all indices for you automatically and store them in csv files. Setting the spatial resampling parameter to 2x2x2 mm in the script file can be done as follows:

```
# SpatialResampling of Img0 of session0
  LIFEx.texture.Session0.Img0.ZSpatialResampling=2
  LIFEx.texture.Session0.Img0.YSpatialResampling=2
  LIFEx.texture.Session0.Img0.XSpatialResampling=2
```

Using scripts may produce numerous ROIs. To check that the resulting ROI are compatible with texture index calculation and before running such calculations that can be time-consuming, you can use the CheckTex button after the ROI creation and before proceeding with textural index calculation. This will check whether the ROI includes a single cluster and contain a number of voxels greater than that required for consistent textural feature calculation. If one of these two conditions is not met, a warning message will be displayed.

1.7.2　What is a script file?

A script file makes it possible to run all operations and calculations without any user interaction. You can prepare it in advance. You can have as many script files as you want, with names that you choose and you can save them and modify them. They are simple text files.

1.7.3 How to write a script file?

The script file describes a list of functions to be executed within LIFEx.

- a) Reading the patient (series) file;

- b) Perform filter on series

- b) Reading one or several ROI files for that patient;

- c) Setting the parameters involved in the index calculation;

- d) Calculation of the different indices;

- e) Writing results in a new or existing csv file;

- f) Closing all files;

- g) Back to step a) if needed.

Here is an example of script file based on a example provided with LIFEx. You can save it as `TextureScript.txt`. The lines starting with # are not interpreted by LIFEx, they only include comments.

```
##########################################################################
# filter (only one activate at same script) #
##########################################################################
#filter: Laplacian of Gaussian (LoG) (LIFEx>=7.0.0) [ true || false ]
LIFEx.filter.LoG.Enable=false
#filter: Laplacian of Gaussian (LoG) [ 3D || 2D ]
LIFEx.filter.LoG.DimensionProcessing=3D
#filter: Laplacian of Gaussian (LoG) [ Reflect || Periodic || Edge || Zero ]
LIFEx.filter.LoG.PaddingMethod=Reflect
#filter: Laplacian of Gaussian (LoG) [ unit mm ]
LIFEx.filter.LoG.Sigma.z=7
LIFEx.filter.LoG.Sigma.y=2
LIFEx.filter.LoG.Sigma.x=2

#filter: Mean (LIFEx>=7.0.0) [ true || false ]
LIFEx.filter.Mean.Enable=false
#filter: Diameter Kernel Size (vx) [ integer value ]
LIFEx.filter.Mean.DiameterKernelSize=3
#filter: PaddingMethod [ Reflect || Periodic || Edge || Zero ]
LIFEx.filter.Mean.PaddingMethod=Reflect

#filter: Wavelet (LIFEx>=7.0.0) [ true || false ]
LIFEx.filter.Wavelet.Enable=false
#filter: Transform along Z axis(only for 3D) [ true || false ]
LIFEx.filter.Wavelet.ZTransform=true
#filter: Family [ Coiflets || Biorthogonal || Daubechies || Haar || Reverse biorthogonal || Symlets ]
LIFEx.filter.Wavelet.Family=Coiflets
#filter: Order [ integer value ] cf. documentation for values
LIFEx.filter.Wavelet.Order=1
#filter: Level [ constant integer value ] cf. documentation for values
LIFEx.filter.Wavelet.Level=1
```

```
#filter: PaddingMethod [ Reflect || Periodic || Edge || Zero ]
LIFEx.filter.Wavelet.PaddingMethod=Reflect

#filter: Laws (LIFEx>=7.0.0) [ true || false ]
LIFEx.filter.Laws.Enable=false
#filter: Kernel Size (vx) [ L3 || L5 || E3 || E5 || S3 || S5 || W5 || R5 ]
LIFEx.filter.Laws.ZKernelSize=L3
LIFEx.filter.Laws.YKernelSize=L3
LIFEx.filter.Laws.XKernelSize=L3
#filter: PaddingMethod [ Reflect || Periodic || Edge || Zero ]
LIFEx.filter.Laws.PaddingMethod=Reflect


########################################################################
# texture #
########################################################################
#texture: Common (LIFEx>=5.1.0)
LIFEx.texture.BinSize=0
LIFEx.texture.NbGrey=64
LIFEx.texture.SessionCsv=.../ScriptDemoMultiSeries/TextureResults.csv

#texture: Absolute (LIFEx>=5.1.0) [ true || false ]
LIFEx.texture.ButtonAbsolute=true
LIFEx.texture.MinBound=0
LIFEx.texture.MaxBound=20

#texture: RelativeMeanSd (LIFEx>=5.1.0) [ true || false ]
LIFEx.texture.ButtonRelativeMeanSd=false

#texture: RelativeMinMax (LIFEx>=5.1.0) [ true || false ]
LIFEx.texture.ButtonRelativeMinMax=false

#texture: DistanceWithNeighbours (LIFEx>=5.1.0)
LIFEx.texture.GLCM.DistanceWithNeighbours=1

#texture: dimension calculation (3D is default) (LIFEx>=5.1.0) [ 3D || 2D ]
LIFEx.texture.DimensionProcessing=3D

# Patient0
LIFEx.texture.Session0.Img0=.../ScriptDemoMultiSeries/PT0img.nii.gz
# SpatialResampling of Img0 of session0 (0 = no spatial resampling = native spacing voxels)
LIFEx.texture.Session0.Img0.ZSpatialResampling=0
LIFEx.texture.Session0.Img0.YSpatialResampling=0
LIFEx.texture.Session0.Img0.XSpatialResampling=0
LIFEx.texture.Session0.Roi0=.../ScriptDemoMultiSeries/R1.nii.gz
LIFEx.texture.Session0.Roi1=.../ScriptDemoMultiSeries/R2.nii.gz

LIFEx.texture.Session1.Img0=.../ScriptDemoMultiSeries/PT1img.nii.gz
# SpatialResampling of Img0 of session1 (0 = no spatial resampling = native spacing voxels)
LIFEx.texture.Session1.Img0.ZSpatialResampling=2
LIFEx.texture.Session1.Img0.YSpatialResampling=2
LIFEx.texture.Session1.Img0.XSpatialResampling=2
```

`LIFEx.texture.Session1.Roi0=.../ScriptDemoMultiSeries/C1.nii.gz`

Here are the functions that are executed when running this text file:

- Reading the images (`PT0img.nii.gz`) of patient0 (=`Session0`)
- no spatial resampling (because values=0)
- Reading `R1.nii.gz` ROI of patient0
- Reading `R2.nii.gz` ROI of patient0
- Reading the setting of the Common, Absolute, RelativeMeanSd, RelativeMinMax
- Calculating all texture features of `R1.nii.gz` and `R2.nii.gz` on `PT0img.nii.gz`
- Writing the results in an existing csv session file named `TextureResults.csv`
- Closing all ROI and patient0 data

then:

- Reading the images `PT1img.nii.gz` of patient1 (=`Session1`)
- Do a spatial resampling with 2x2x2 new spacing voxel
- Reading `C1.nii.gz` ROI of patient1
- Reading the setting of the Common, Absolute, RelativeMeanSd, RelativeMinMax
- Calculating all texture features on `C1.nii.gz` on `PT1img.nii.gz`
- Writing the results in an existing csv session file named `TextureResults.csv`
- Closing all ROI and patient1 data

1.7.4 How to run a script file.

Before running a script:

1. Please check that you have no session file in the destination directory that you have specified in your script file;

2. The result filenames should NOT be `*TextureSession.csv`

Running the script :

- Please drag the script file in the panel used to load patient images.
- Alternatively, you can use the « browser + LocalDisk » interface to find and load you script file from your local disk.

1.7.5 Can I run several script files at the same time?

You can indeed run several script files one after another automatically. To do so, please select all script files and drag them in the panel used to put the patient image files. The script files will be executed one after another.

Beware: you can not load your script files using the « browser » interface, which supports only one script file at a time.

1.7.6 Advanced execution scripting of grey-level resampling.

When you set the quantization (or grey-level resampling) parameters, you can leave one of the two sets to 0. Its value will then be automatically calculated based on the other parameter setting.

```
LIFEx.texture.BinSize=3.125
LIFEx.texture.NbGrey=128.0
```

For instance:

if $BinSize = 0$ then $BinSize = (boundMax - boundMin)/(nbGreyLevels - 1)$

if $nbGreyLevels = 0$ then $nbGreyLevels = ((boundMax - boundMin)/binSize) + 1$

if $BinSize = 0$ and $nbGreyLevels = 0$ then $nbGreyLevels = 64$ by default
$boundMin$ and $bounsMax$ are the minimum and maximum values in the processed ROI.

Discretization has been set with:

$$discritizedValue = floor((nbGreyLevels * \frac{originalValue - boundMin}{boundMax - boundMin} + 1)$$

1.7.7 Advanced execution scripting of spatial resampling.

When you set the spatial resampling parameters, you can leave the three parameters $ZSpatialResampling$, $YSpatialResampling$, $XSpatialResampling$ set to 0 (or delete rows). This is equivalent to performing the calculation using the original voxel size of the images.

If you want to change the voxel size of the image, set the parameters to a value expressed in <u>millimeters</u>. The three mandatory values ($ZSpatialResampling$, $YSpatialResampling$, $XSpatialResampling$) can be different.

1.7.8 Main remarks relevant to the writing of a script in LIFEx:

- What are the image formats that can be managed using LIFEx scripts?

  - image files can be in NIfTI-1 format (.nii or .nii.gz). In this case, the complete pathway to the patient image file should be given in the script;

  - to load DICOM images, you must define the root directory of these images (without the final filename); All files included in this directory will be loaded;

- What are the ROI formats that can be managed using LIFEx scripts?

  - the NIfTI-1 and RTStruct are supported in LIFEx scripts. In both cases, the file name should include the full path to the file. In the case of RTStuct, all ROI are loaded without exception.

- Syntax of all pathways of files in LIFEx scripts:

  - the syntax of pathway <u>must not contain accents</u> and <u>better when not contain space</u>.

  - the syntax of pathway must not necessarily conform to your system rules:

* Windows: Unit:/Directory/File.extension or Unit:/Your Directory/ for series of DICOM images ; example C:/Home/Users1/File1
* Linux: /Your Directory/Your File.extension
* MacOs: /Your Directory/Your File.extension

1.8   Local Texture (Map) feature

Intensity discretization and intensity rescaling menus of the local texture map menus have the same meaning as in the global feature menu (see section 1.4 p.11 ; section 1.5, p.13).

Then, you have to decide whether you want to get a map of the textural feature in the whole image (which we do not recommend as it would be quite long to calculate) or only for the voxels included in a ROI:

- we recommend to calculate the textural feature map in a ROI only. For that, please select the ROI from which you want to get the parametric map.

- if you select « on voxels from activated ROI » in the menu, you will get the parametric texture map in the selected ROI.

- if you select on bounding box of activated ROI in the menu, you will get the parametric texture map in a rectangle including the ROI, which might be nicer for display purpose.

The kernel parameter is the size of the box the features are calculated from. If you select 3, in each and every voxel K, the features will be calculated from a 3 x 3 x 3 box centered on that voxel K and the feature parametric map corresponding to a given feature will display that feature value in voxel K. The box is moved by one voxel and the calculation is repeated. This is repeated on all voxels in your VOI to get a parametric map of the feature. Doing that, you can get a parametric map of Entropy, Homogeneity, aso.

When clicking the run button, all parametric maps available will be calculated. Then, on the right hand side of your image display window, you have buttons making it possible to display the parametric map of your choice, on top of the original image.
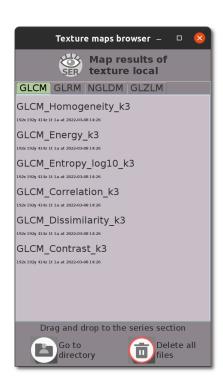
Figure 1.8: Texture feature extract - GLCM Map Results

Part II
Texture Features

Chapter 1
GLCM definition

1.1 GLCM

The grey level co-occurrence matrix (GLCM) [Haralick] takes into account the arrangements of pairs of voxels to calculate textural indices. The GLCM is calculated from 13 different directions in 3D with a δ-voxel distance ($\|\vec{d}\|$) relationship between neighboured voxels. The index value is the average of the index over the 13 directions in space (X, Y, Z). Seven textural indices are computed from this matrix. An entry (i, j) of GLCM for one direction is equal to:

$$GLCM_{\Delta x, \Delta y}(i,j) = \frac{1}{Pairs_{ROI}} \sum_{p=1}^{N-\Delta x} \sum_{q=1}^{M-\Delta y} \begin{cases} 1 \text{ if } (I(p,q) = i, I(p+\Delta x, q+\Delta y) = j) \\ \quad \text{and } I(p,q), I(p+\Delta x, q+\Delta y) \in ROI \\ 0 \text{ otherwise} \end{cases}$$

(1.1)

where $I(p, q)$ corresponds to voxel (p, q) in an image (I) of size $N * M$. The vector $\vec{d} = (\Delta x, \Delta y)$ covers the 4 directions (D1, D2, D3, D4, Figure 1.1) in 2D space or 13 directions (D1, D2, ..., D13, Figure 1.2) in 3D space and $Pairs_{ROI}$ corresponds to the number of all voxel pairs belonging to the region of interest (ROI).

Cost-effective creation of a transposed matrix. To reduce runtime calculation of GLCM matrix involving 26 directions in 3D (8 directions in 2D) the transposed matrix accounting for 13 directions calculated in 3D (and 4 in 2D) is added to the original matrix.
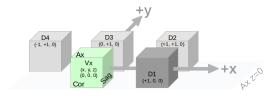

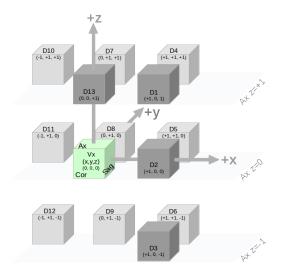Figure 1.1: 2D Voxel (green) and associated 4 directions

Figure 1.2: 3D Voxel (green) and associated 13 directions

The GLCM describes the distribution of co-occurring pixel values at a given offset. This offset (δ) set to 1 by default, meaning that only neighbour voxels are used to calculate GLCM. It is possible to change this default value from 1 to 10 in the "Distance with neighbours" box, which is in the "Settings" dialog box, GLCM subpart.

If you specify a distance of 5 voxels the matrix will be built not with the nearest neighbor (dist = 1) but with the neighbor with 5 voxels of distance and without taking into account the intermediate voxels of distance 2, 3, 4.

Be careful: if the distance is too large for small ROIs, some entries of the GLCM matrix will not have any meaning. This is detected and the corresponding results in the output csv file are noted NaN! or Num!

GLCM

1.2 Save results for each direction

Textural indices corresponding to each direction can be saved independently (13 directions in 3D and 4 directions in 2D). To do that, please tick the "SaveGLCM in file" option in the "Output" sub-dialog box.

All other features exist and are defined on the Image biomarker standardisation initiative (IBSI) document attached in the LIFEx package. Feel free to take the reference IDs of the features in order to make faster searches in this document.

Chapter 2
NGLDM definition

2.1 NGLDM

The neighborhood grey-level difference matrix (NGLDM) [Amadasum1989] corresponds to the difference of grey-levels between one voxel and its 26 neighbours in 3 dimensions (8 in 2D). Three texture indices can be computed from this matrix. An element $(i, 1)$ of NGLDM corresponds to the probability of occurrence of level i and an element $(i, 2)$ is equal to:

$$NGLDM(i, 2) = \sum_p \sum_q \left\{ \begin{array}{l} |\overline{M}(p, q) - i| \text{ if } I(p, q) = i \\ 0 \text{ else} \end{array} \right. \tag{2.1}$$

where $\overline{M}(p, q)$ is the average of intensities over the 26 neighbour voxels of voxel (p, q).

NGLDM

All other features exist and are defined on the Image biomarker standardisation initiative (IBSI) document attached in the LIFEx package. Feel free to take the reference IDs of the features in order to make faster searches in this document.

Chapter 3
GLRLM definition

3.1 GLRLM

The grey-level run length matrix (GLRLM) [Xu] gives the size of homogeneous runs for each grey level. This matrix is computed for the 13 different directions in 3D (4 in 2D) and for each of the 11 texture indices derived from this matrix, the 3D value is the average over the 13 directions in 3D (4 in 2D). The element (i, j) of GLRLM corresponds to the number of homogeneous runs of j voxels with intensity i in an image and is called $GLRLM(i, j)$ thereafter.

vs. pyRadiomic. We must highlight that comparisons of results with other software supporting texture analysis should be performed with great care. The calculation of the texture indices resulting from the matrix GLRLM can differ between software. For instance, in pyRadiomics (v1.1.1), after the calculation of the matrix GLRLM and before the extraction of the textural indices, the matrix is cropped (grey-level axis of GLRLMs cropped between minimum and maximum grey-levels and run-length axis of GLRLMs cropped to maximum run-length). This moves indexes (i, j) of the matrix and thus the values of the resulting textural indices.

In LIFEx, we do not shift the index so that it corresponds to grey level i, and j corresponds to the number of run j and we use the formulations defined below.

GLRLM

All other features exist and are defined on the Image biomarker standardisation initiative (IBSI) document attached in the LIFEx package. Feel free to take the reference IDs of the features in order to make faster searches in this document.

Chapter 4
GLZLM definition

4.1 GLZLM

The grey-level zone length matrix (GLZLM) [Thibault] provides information on the size of homogeneous zones for each grey-level in 3 dimensions (or 2D). It is also named Grey Level Size Zone Matrix (GLSZM). From this matrix, 11 texture indices are computed. Element (i, j) of GLZLM corresponds to the number of homogeneous zones of j voxels with the intensity i in an image and is called $GLZLM(i, j)$ thereafter.

All other features exist and are defined on the Image biomarker standardisation initiative (IBSI) document attached in the LIFEx package. Feel free to take the reference IDs of the features in order to make faster searches in this document.

GLZLM

Chapter 5

First order features from Histogram

5.1 Histogram calculation

To build a histogram *HISTO*, it is necessary to determine a bin width ("bin" parameter). The indices derived from the histogram will depend on this bin width parameter.

This dependence, similar to that found in texture index calculations, is often overlooked in publications.

In LIFEx, with the absolute model the histogram is built a number of bins equal to that entered by the user in the "number of grey level" and "size of bin" (or bin width or

bw, see eq.5.1) fields of the resampling menu.

$$bw = \frac{max - min}{nbGreyLevel} \tag{5.1}$$

where *max* is maximum of intensity (in ROI), *min* is minimum of intensity (in ROI) and *nbGreyLevel* is the number of grey level.

In LIFEx, with the relative model the histogram is built only with "number of grey level" fields of the resampling menu that entered by the user and min and max are extracted values of each ROI.

5.2 DISCRETIZED_AUC_CSH

reflects the cumulative intensity of histograms ($p(i)$) produce by a per cent volume of a ROI (derived from ROI semi-automatic ROI delineation methods) with an intensity above a certain threshold is plotted against that threshold value (j), which is varied from 1 to number of grey [Van Velden 2011].

The area under of this new histogram ($DISCRETIZED\_AUC\_CSH$ feature) is a quantitative index of tracer uptake heterogeneity and/or heterogeneous response where lower values correspond with increased heterogeneity. In this case $DISCRETIZED\_AUC\_CSH$ is independent of value max (here, number of grey).

$$DISCRETIZED\_AUC\_CSH = \sum_{j} \sum_{i \geq j} p(i) * bw \tag{5.2}$$

where $p(i)$ is the probability of occurrence of voxels with intensity i, j is the number of grey and *bw* the width of one bin (see eq.5.1).

First order features from Histogram

All other features exist and are defined on the Image biomarker standardisation initiative (IBSI) document attached in the LIFEx package. Feel free to take the reference IDs of the features in order to make faster searches in this document.

Chapter 6
Conventional Indices

Contexe The values (*Value*) relating to this part are extracted from the native values of the images.

6.1 *CONVENTIONAL_Q1, _Q3*

A quartile is a type of quantile which divides the number of data points into four more or less equal parts, or quarters. Due to the fact that the data needs to be ordered from smallest to largest in order to compute quartiles, quartiles are a form of Order statistic.

CONVENTIONAL_Q1: The first quartile (Q1) is defined as the middle number between the smallest number and the median of the data set. It is also known as the lower quartile or the 25th empirical quartile and it marks where 25% of the data is below or to the left of it (if data is ordered on a timeline from smallest to largest).

$CONVENTIONAL\_Q3$:   The third quartile (Q3) is the middle value between the median and the highest value of the data set. It is also known as the upper quartile or the 75th empirical quartile and 75% of the data lies below this point.

6.2 $CONVENTIONAL_{calciumAgatstonScore}$

reflects the $CONVENTIONAL_{calciumAgatstonScore}$ of the ROI. This feature is only available on CT images: see UserGuide of LIFEx "Calcium Quantitation - Agatston Score" for more information.

6.3 $CONVENTIONAL\_TLG(mL)$

is the Total Lesion Glycolysis defined as the product of $SUVmean$ by $Volume$ in mL. [LARSON 1999]

$$CONVENTIONAL\_Mean = V \cdot \frac{1}{N} \sum_i SUV_i \tag{6.1}$$

6.4 $CONVENTIONAL\_RIM$

is the Radial Intensity Mean "$CONVENTIONAL\_RIM\_mean$" of successive layers of voxels (envelope) from the outside of the region to the inside. Each layer is 1 voxel thick. These envelopes are getting smaller and smaller (3D erosion of 1 voxel) up to the center of the ROI.

Other statistics are extracted from these envelopes, in particular:

- $CONVENTIONAL\_RIM_{min}$: minimum value of voxel values from the envelopes

- $CONVENTIONAL\_RIM_{stdev}$: standard deviation value of voxel values from the envelopes

- $CONVENTIONAL\_RIM_{max}$: maximum value of voxel values from the envelopes

- $CONVENTIONAL\_RIM_{volume}(mL)$: volume in milliliter unit of all voxels from the envelopes

- $CONVENTIONAL\_RIM_{volume}(vx)$: volume in voxel unit of all voxels from the envelopes

- $CONVENTIONAL\_RIM_{sum}$: sum of voxel values from the envelopes

Conventional Indices

Example of results for 6 envelopes:
$CONVENTIONAL\_RIM_{mean}$ = 1.2684131|1.2192162|1.2974288|1.3191646|1.271777|1.226076|

The first value 1.2684131 is SUV_{mean} (if SUV is unit of voxel value), of the 1-voxel thick envelope of the ROI, 1.2192162 the mean of the envelope after an erosion of 1 voxel, ..., 1.226076 the mean of the smallest envelope (center of ROI).

All other features exist and are defined on the Image biomarker standardisation initiative (IBSI) document attached in the LIFEx package. Feel free to take the reference IDs of the features in order to make faster searches in this document.

Discretized Indices

Contexe The values (*Value*) relating to this part are extracted from the discretized values of the images.

7.1 *DISCRETIZED_Q1, _Q3*

A quartile is a type of quantile which divides the number of data points into four more or less equal parts, or quarters. Due to the fact that the data needs to be ordered from smallest to largest in order to compute quartiles, quartiles are a form of Order statistic.

DISCRETIZED_Q1: The first quartile (Q1) is defined as the middle number between the smallest number and the median of the data set. It is also known as the lower quartile or the 25th empirical quartile and it marks where 25% of the data is below or to the left of it (if data is ordered on a timeline from smallest to largest).

DISCRETIZED_Q3: The third quartile (Q3) is the middle value between the median and the highest value of the data set. It is also known as the upper quartile or the 75th empirical quartile and 75% of the data lies below this point

7.2 *DISCRETIZED_CalciumAgatstonScore*

reflects the *DISCRETIZED_CalciumAgatstonScore* of the ROI. This feature is only available on CT images: see UserGuide of LIFEx "Calcium Quantitation - Agatston Score" for more information.

7.3 $DISCRETIZED\_TLG(mL)$

is the Total Lesion Glycolysis defined as the product of *SUVmean* by *Volume* in mL. [LARSON 1999]

$$DISCRETIZED\_Mean = V \cdot \frac{1}{N} \sum_i SUV_i \tag{7.1}$$

7.4 *DISCRETIZED_RIM*

is the Radial Intensity Mean "*DISCRETIZED_RIM_mean*" of successive layers of voxels (envelope) from the outside of the region to the inside. Each layer is 1 voxel thick. These envelopes are getting smaller and smaller (3D erosion of 1 voxel) up to the center of the ROI.

Other statistics are extracted from these envelopes, in particular:

- $DISCRETIZED\_RIM_{min}$: minimum value of voxel values from the envelopes

- $DISCRETIZED\_RIM_{stdev}$: standard deviation value of voxel values from the envelopes

- $DISCRETIZED\_RIM_{max}$: maximum value of voxel values from the envelopes

- $DISCRETIZED\_RIM_{volume}(mL)$: volume in milliliter unit of all voxels from the envelopes

- $DISCRETIZED\_RIM_{volume}(vx)$: volume in voxel unit of all voxels from the envelopes

- $DISCRETIZED\_RIM_{sum}$: sum of voxel values from the envelopes

Example of results for 6 envelopes:
$DISCRETIZED\_RIM_{mean} = 1.3|1.2|1.3|1.3|1.2|1.1|$

Discretized Indices

The first value 1.3 is SUV_{mean} (if SUV is unit of voxel value), of the 1-voxel thick envelope of the ROI, 1.2 the mean of the envelope after an erosion of 1 voxel, ..., 1.1 the mean of the smallest envelope (center of ROI).

All other features exist and are defined on the Image biomarker standardisation initiative (IBSI) document attached in the LIFEx package. Feel free to take the reference IDs of the features in order to make faster searches in this document.

Part III
References

Chapter 1
Useful references

1.1 Bibliography

Journal papers involving the use of LIFEx are listed in
https://www.lifexsoft.org/index.php/resources/publications-journal-papers

1.1.1 2018

F. Orlhac, S. Boughdad, C. Philippe, H. Stalla-Bourdillon, C. Nioche, L. Champion, M. Soussan, F. Frouin, V. Frouin, I. Buvat. A post-reconstruction harmonization method for multicenter radiomic studies in PET. J Nucl Med. 2018 doi: 10.2967/jnumed.117.199935. (link)

1.1 Bibliography

1.1.2   2017

F. Orlhac, C. Nioche, M. Soussan, I. Buvat. Understanding changes in tumor textural indices in PET: a comparison between visual assessment and index values in simulated and patient data. J Nucl Med. 2017 Mar;58(3):387-392. (link)

1.1.3   2016

F. Orlhac, B. Thézé, M. Soussan, R. Boisgard, I. Buvat. Multi-scale texture analysis: from 18F-FDG PET images to pathological slides. J Nucl Med 57: 1823-1828, 2016. (link)

1.1.4   2015

F. Orlhac, M. Soussan, K. Chouahnia, E. Martinod, I. Buvat. 18F-FDG PET-derived textural indices reflect tissue-specific uptake pattern in non small cell lung cancer. Plos One 10(12):e0145063, 2015. (link)

I. Buvat, F. Orlhac, M. Soussan. Tumor texture analysis in PET: where do we stand? J Nucl Med 56: 1642-1644, 2015. (link)

1.1.5   2014

M. Soussan, F. Orlhac, M. Boubaya, L. Zelek, M. Ziol, V. Eder, I. Buvat. Relationship between tumor heterogeneity measured on FDG-PET/CT and pathological prognostic factors in invasive breast cancer. Plos One 9: e94017, 2014. (link)

F. Orlhac, M. Soussan, JA. Maisonobe, CA. Garcia, B. Vanderlinden, I. Buvat. Tumor texture analysis in 18F-FDG-PET: relationships between texture parameters, histogram indices, SUVs, metabolic volumes and total lesion glycolysis. J Nucl Med 55: 414-422, 2014. (link)

1.1.6   2012

Vanderhoek M, Perlman S, Jeraj R. Impact of the definition of peak standardized uptake value on quantification of treatment response. J Nucl Med. 2012;53:4-11. (link)

1.1.7   2011

Floris H. P. van Velden, Patsuree Cheebsumon, Maqsood Yaqub, Egbert F. Smit, Otto S. Hoekstra, Adriaan A. Lammertsma, Ronald Boellaard. Evaluation of a cumulative SUV-volume histogram method for parameterizing heterogeneous intratumoural FDG uptake in non-small cell lung cancer PET studies Eur J Nucl Med Mol Imaging (2011) 38:1636?1647. DOI 10.1007/s00259-011-1845-6 (link)

1.1.8   2009

Thibault G, Fertil B, Navarro C, et al. Texture indexes and gray level size zone matrix. application to cell nuclei classification. Pattern Recognition Inf Process. 2009;140:145 (link)

Useful references

46

Wahl R, Jacene H, Kasamon Y and Lodge MA. From RECIST to PERCIST:evolving considerations for PET response criteria in solid tumors. J Nucl Med. 2009;50:122S-50S. (link)

1.1.9 2004

Xu D-H, Kurani AS, Furst JD, Raicu DS. Run-length encoding for volumetric texture. Paper presented at: The 4th IASTED International Conference on Visualization, Imaging and Image Processing: VIP; September 6-8, 2004; Marbella, Spain (link)

1.1.10 1999

Larson S, Erdi Y, Akhurst T et al. Tumor treatment response based on visual and quantitative changes in global tumor glycolysis using PET-FDG Imaging. The visual response score and the change in total lesion glycolysis. Clin Positron Imaging. 1999;42:159-171. (link)

1.1.11 1989

Amadasun M, King R. Textural features corresponding to textural properties. IEEE Trans Syst Man Cybern.1989;19:1264-1274 (link)

1.1.12 1973

Haralick RM, Shanmugam K, Dinstein I. Textural features for image classification. IEEE Trans Syst Man Cybern.1973;3:610-621. (link)

Useful references