

## 1.calculateStructureVolume.m(slevin)

A script used to calculate the volume of a structure

It is not a function, if we want to use it, we need to **rewrite** it into a function.

It is well commented and has some code about parsing RT Structure Dicom file.

## 2. mysql.m (Veda) – provides documentation for the c++ helper function mysql.cpp along with usage examples.

Mysql is used to make calls to the database rt\_hn\_v6 in the server and get information from the dicom files.

Parameters

-----

None

Returns

-----

None

Keep?

-----

Should be kept to provide docs on mysql call format

## 3. mysql.cpp (Veda) – c++ helper function that allows calls to SQL database from matlab. Developed by <http://www.courant.nyu.edu/~almgren/mysql/mysql.cpp>.

Requires libmysqlclient-dev on Ubuntu (installed through sudo apt-get install on 14.04) + libraries from oracle website (apt libraries for ubuntu:  
<https://dev.mysql.com/downloads/repo/apt/>)

The function must be compiled prior to usage, requires MEX setup with valid c++ compiler on system.

Usage: mysql(command)

Parameters

-----

command : string

A string containing a sql command, such as 'use DATABASE\_1'

Returns

-----

Data from sql database

Keep?

-----

Should be kept to ensure mysql access for dicom info

#### 4. untitled.m (Veda)

untitled.m - provides a demo of getting a specific patient's CT images and CT info from sql.

Assembles a CT block using the z dimension of a patient's position from set of dicom CT images.

Appears to create part of an isodose contour, does not display.

Important Parameters

-----

fk\_patient\_id : int - 1, used to test data of a single patient in script

imageBlock : 3D matrix, an assembled CT block of width x height x numCTs

doseGrid : 3D matrix [w x h x numCTs] , data to create isodose contour (dose.dcm in UCLA)

imageBlockRGB - dose grid of dims [w x h x numCTs] converted to RGB by repeating each slice 3x.

tempDoseMask : 4D matrix, binarized doseGrid where 1 indicates region where dose of specific percentage was administered.

doseOutline : 4D matrix, perimeter of regions from tempDoseMask where dosage was administered.

Results

-----

No returned or saved variables or saved figures.

Keep?

-----

Needs to be rewritten to become a demo script.

#### 5.centroidDistance.m(Slevin)

It is a script used to calculate the centroid Distance between regions of interest. Such like PTV VS bladder or PTV vs rectum.  
Need to be rewritten into a function.  
It also build a connection with database and use the data from database to calculate.  
Well commented and ease to read.

## **6.imrotate\_3D.m (Veda)**

Function to rotate image volume A by angle degrees in a counterclockwise direction around its center point in the x,y-plane. Short, well commented.

Inputs

-----

InMat – 3D array to rotate  
degrees – double, degrees to rotate InMat by.

Returns

-----

OutMat – Rotated 3D array

Keep?

-----

MATLAB dependency report reports this function is not called by any other function-likely to need to be deleted unless other use is found.

## **7.ROIVolumes\_DB.m (Slevin)**

A script calculate volumes of ROI.

Well commented

Build a connection with DB and use the data from DB and dicom file

The algorithm is same as that of calculateStructureVolume

## **8.ROI\_Volume\_Surveyors.m (Slevin)**

A script calculate the volumes of ROI

It has use **geometric formula** to calculate the volumes of ROI

It is different from the algorithm of ROI\_Volume

## **9.getOvh.m(slevin)**

It is well commented

The steps to get OVH :

- 1.Create Mapping of ROI Number and names
- 2.Determine the number of items in each ROI
- 3.Extract Row spacing and column spacing from CT Dicom file
- 4.Get PTV and OAR contours and solid ROI masks(by using getContoursFull function)
- 5.Smooth contours of PTV
- 6.Separate OAR into intersecting and non-intersecting parts
- 7.Get the position of each Non zero voxel in the organ mask
- 8.Calculate minimum distance from each OAR(intersecting sub and nonintersecting sub) point to the PTV outline
- 9.Based on the distance and bin width to get OVH graph

## **10.getContoursFull.m(Slevin, Natalie)**

A Function returns a 3D contour block and solid ROI mask.

Assume that CT images and related structure file are in the same sub-folder.

Inputs

-----

structureSetInfo– parsed dicom data

ROI\_Item\_no– the ROI number

Width – the width of contour block

Height – the height of contour block

Patient\_number – the patient ID in the database

Returns

-----

roiBlock – a solid ROI mask( a 3D matrix)

contourBlock -- a 3D contour block( a 3D matrix)

- SlicePositionMap – holds image position (in z direction) for each CT image slice
- PlaneContourData: holds contour points for particular plane and particular ROI
- Xx, xy, yx, yy – projections (cosines) to give orientation
  - Xx = projection of x on x axis, xy = projection of y on x axis, etc.
  - Helpful: <http://dicomiseasy.blogspot.com/2013/06/getting-oriented-using-image-plane.html>
- Sx, sy – x and y position from upper left hand corner of image
- DelJ, delI – pixel spacing in x and y direction
- A = 2x2 – multiplies projections by respective del spacing
- B = 2x1 – displacement - difference between contour position and reference upper left hand corner position
- V = solved linear equation (A\b), giving pixel coordinates (x and y, for a given z) \*\*used to make contourBlock
- RoiBlock – poly2mask takes col and row coordinates (found from v) to make solid structure (3D matrix)

Matlab functions used (need to find equivalent in Python):

Dicominfo – search pydicom --> dicominfo.py

<http://pydoc.net/pydicom/0.9.9/dicom.examples.DicomInfo/>

Poly2mask

## 11.ovhParameters.m / ovhParameters2.m (Slevin)

Same process as getOvh.m

It calculate ovh for ptv and bladder and rectum

## **12.findX.m(Slevin)**

A function estimates the XI values at which the dependent variable Y reach or cross a specified target level YI

Inputs

-----

X : a vector containing a sequence of monotone increasing values of the independent variable

Y: a vector containing the discrete values of the dependent variable

Returns

-----

XI : at which Y reach YI

IDEXACT:  $Y(\text{IDEXACT}) = YI$

## **13.ovh\_init.m(Slevin)**

A script to fetch the information from dicom file

The process is used in getOvh.m file.

Could help us to know how to get the data from structureset.dcm

## **14.ovh\_init2.m(Slevin)**

It is just a copy of getOvh.m file

## **15.emdforDVH.m (Veda)**

Function that returns the chi square distance for the dvhvolumes of one patient and the linearly interpolated dvhvolumes of another patient based of the first patient's histogram.

Inputs

-----

patient1 : int , PatientID of first patient (query patient) to compare  
patient2 : int, PatientID of second patient (historical case) to compare  
roiName : string, ROI to compare between patients

Returns

-----

d : scalar, the distance between two DVHs

Keep?

-----

Yes- already a function, so may have use. Dependency report shows it is used to precalculate differences between DVHs (test\_insertDVHDistance). does it actually calculate earth mover's distance though?

#### **16. test\_insertDVHDistance.m (Veda)**

Updates dvhDistance field in mysql database for a given ROI table. The table used in the script is 'pharynx' , which does not exist in the database rt\_hn\_7 (though the similar tounguePairwise, cochleaPairwise, etc does exist)

Keep?

-----

Possibly- effectively wraps a for loop for the function emdforDVH.

#### **17. test\_dvhComulative.m (Veda)**

Script that returns nothing, gets cumulative sum for dvhDose field for a given patient and given ROI pairwise table.

Keep?

-----

MATLAB dependency report reports this function is never used by another function, so probably do not keep.

#### **18. test\_emdforDVH.m (Veda)**

emdforDVH.m but in script form, not function form.

Keep?

-----

MATLAB reports near duplicate of emdforDVH (lines that vary are unused variables or commented out lines) so probably can be deleted.

## 19. ChiforOVH.m (Veda)

Uses the same formula as emd for dvh, except chi square distance is divided by 2 and not 200 as in emdforDVH. Also calculates chi square distance for OVH histograms, not DVH histograms as in emdforDVH.

Inputs

-----

patient1 : int , PatientID of first patient (query patient) to compare  
patient2 : int, PatientID of second patient (historical case) to compare  
roiName : string, ROI to compare between patients

Returns

-----

d : scalar, the distance between two OVHs

Keep?

-----

Yes- already a function, so may have use. Dependency report shows it is used to precalculate differences between OVHs (test\_insertOVHChi). does it actually calculate earth mover's distance though?

## 20. emd.m (Natalie)

Computes the flow that minimizes an overall cost function subject to a set of constraints

Used in emdCalculator.m

Inputs

-----



F1, F2 - feature vectors of respective signatures – e.g. OVH

W1, W2 - weights of the features

Func - function which computes the ground distance between two feature vectors

Returns

-----

X - the flow which minimizes the cost function

Fval – the value of this flow

## **21. PT\_ROIvolumes.m (Veda)**

Gets the percent overlap, and intersecting volume between the PTV (in this script example, prostate) and each of two OARs for a single patient. Well commented. The data is then inserted back into the SQL database after processing. Commented out part where the PTV contour block is displayed.

Inputs

-----

None, is script. PatientID (string) is set in script, SOPClassUID (string) is MediaStorageSOPClassUID from dicom file 'RP....dcm' and is also hardcoded in script.

Returns

-----

None. Resulting percent overlap and intersecting volumes are saved in SQL database.

Keep?

-----

Keep, though should become function with patientID and SOP as input params, and possibly split display of contour from rest of function.

## **22. PT\_getContoursFull2.m (Veda)**

Gets a matrix with the contour for a given PTV (all points inside the contour become ones, everything else is zeros) Also returns the contour slices (not filled in, just the points) for a given patient. Contains the entirety of PT\_getContoursFull.m commented out at end of script.

#### Inputs

-----

StructureSetInfo : struct, Contains similar field names to dicominfo struct from structureset.dcm in UCLA\_PR\_5

ROI Number : Scalar. Each ROI is assigned a scalar number for ROISequence in StructureSetInfo (fields are Item\_1, Item\_2 .. ).

width : scalar, width of a CT image

height : scalar, height of a CT image

PatientID : scalar

ctStudyInstanceUID : scalar, StudyInstanceUID in structureset.dcm

ctSeriesInstanceUID : scalar, SeriesInstanceUID in structureset.dcm

#### Returns

-----

ptvBlock – 3D matrix of [h x w x number\_of\_ct\_images ] dimensions. All points inside the coordinates of the contour are 1, all points outside are 0.

contourBlock – 3D matrix with same dimensions as ptvBlock. All points on the contour coordinates are 1, rest are 0.

#### Keep?

-----

Probably keep. Much of the function is duplicated in PT\_getContoursFull with the main difference being Full2 uses data from SQL and PT\_getContoursFull function uses data from local storage.

### **23. SIM.m (Natalie)**

Finds the alpha, beta, and gamma component for each ROI in an ROI index.

These components are in the x (alpha), y, or z direction and are the difference in location of the PTV centroid to the ROI centroid over the hypotenuse or total distance between centroids

Should be written as a function

#### **24. EmdCalculator.m (Natalie)**

Uses emd.m to find the emd value between two patients for a certain ROI

Input

-----

Patient1, Patient2 – number value for current and historical patient

RoiName – string of ROI name

Return

-----

EmdValue – earth mover's distance between the ROIs of the two patients

Already a function

#### **25. dvh.m (Natalie)**

Plots DVH (dose volume histogram) for bladder, rectum, and PTV (planning treatment volume)

Finds 50, 75, and 90 dose for bladder, rectum and PTV

#### **26. PR\_testSimilarity.m (Natalie)**

Cycles through all patients and finds PatientID, distance, volume, alpha, beta, gamma, overlap, distance mean, dose mean, dose max, ROI volume to calculate EMD and SIM

-Match determination code is commented out

#### **27. getDVHdb.m (Natalie)**

Function which retrieves DVH dose and volume for a certain patient and ROI

Input

-----

Patient\_id – number of patient's ID

RoiName – string of ROI name

Return

-----

DvhDose - DVH dose double

DvhVolume - DVH volume double

#### **28. Gdm.m (Natalie)**

Finds the ground distance matrix between two signatures

Input

-----

F1, F2 – features of respective signatures

Func – function which computes the ground distance between two feature vectors

Return

-----

F - ground distance matrix

Uses gdf.m

Already a function

### **29. Gdf.m (Natalie)**

Finds the ground distance between two vectors (uses norm function)

Input

-----

V1, V2 – feature vectors

Return

-----

E – ground distance between vectors

Already a function

### **30. PT\_centroidDistances.m (Veda)**

Calculates centroid distances using same procedure as centroidDistances.m.

PT\_centroidDistances.m uses the PT dataset, while centroidDistances.m uses the UCLA\_PR dataset. Because of this, some of the SQL calls are different between the two files.

Keep (Use?)

-----

Procedure of either file will be needed in Python conversion.

### **31. PT\_getContoursFull.m (Veda)**

Calculates centroid distances using same procedure as getContoursFull.m.

PT\_getContoursFull.m uses the PT dataset, while getContoursFull.m uses the UCLA\_PR dataset.

Because of this, some of the SQL calls and loading of data are different between the two files.

Keep (Use?)

-----

Procedure of either file will be needed in Python conversion for OVH.

### **32. getOVHdb.m (Veda)**

Gets MATLAB numerical arrays for OVH volume and OVH distance from SQL server.

Inputs

-----

patient\_id : scalar, patient ID in SQL server

roiName : string, ROI name in SQL server

Returns

-----

ovhDose : 1D Array, contains distances (x values) of OVH

ovhVolume : 1D Array, contains percentage of volume in ROI (y values)

### **33. GetContoursExp.m (Veda)**

Gets contour mask for a single image from a CT scan block. Written in script form.

Keep (Use?)

-----

Better version (functionality for entire CT block) is already written in getContoursFull.m .

### **34. GetContoursExp2.m (Veda)**

Gets contour mask for a single image from a CT scan block. Written in script form. Varies from getContoursExp.m in that getContoursExp2 is written slightly more efficiently than the original version.

Keep (Use?)

-----

Better version is already written in getContoursFull.m .

### **35. getContoursExp3.m (Veda)**

Gets contour mask for a full CT scan block. Written in script form. Varies from getContoursExp2.m in that getContoursExp2 only gets contours for a single image from the image block.

Keep (Use?)

-----

Better version is already written in getContoursFull.m .

### **36. getContoursExp4.m (Veda)**

Gets contour mask for a full CT scan block. Written in script form. Varies from getContoursExp3.m in that getContoursExp4 also displays volume for an ROI block.

Keep (Use?)

-----

Better version is already written in getContoursFull.m .

### **37. getContoursExp5.m (Veda)**

Gets contour mask for a full CT scan block. Written in script form. Varies from getContoursExp4.m in that getContoursExp5 corrects the spacing of the first and last image in an image block (they get ½ spacing compared to non first and last images)

Keep (Use?)

-----

Better version is already written in getContoursFull.m .

-----

### 38. **getCTImageBlock.m** (Piyush)

Contouring for the entire CT block in the file. *strcat* joins all the files in the *containing\_folder*.