PlatiPy

Backend

CLI

* Used for booting the PlatiPy CLI

# DICOM (Folder)

## Io (Subfolder)

* Crawl.py
  + Can get DICOM information from the description
* Nifty\_to\_rtstruct.py
* Nifti\_to\_series.py
  + Converts Nifti image to DICOM image series
* Rtstruct\_to\_nifti
  + Read\_dicom\_image
    - Returns a stik.Image
  + Read\_dicom\_struct\_file
    - Returns pydicom.dataset (rtstruct)

## Tests (Subfolder)

* Test\_convert.py
* Used to convert dicom files into nifty files (unsure of this process for now)

# Imaging (Folder)

## Dose

* DVH.py
  + Def calculate\_dvh
    - Calculates the dose-volume histogram
    - Returns dose\_points, counts

## Generation

* Augment.py
  + Apply\_augmentation(image, augmentation, mask=[])
    - Image has to be sitk.Image
    - Augmentation has to be a DeformableAugment object
    - Returns the image\_derformed (sitk.image) and dvf
  + Generate\_random\_augmentation
    - Input parameters(ct\_image, masks)
    - Augmentation types are; Shift, Contract, expand
* Dvf.py
  + DVF = deformable vector field
  + Generate\_field\_shift(mask\_image, vector\_shift, gaussian\_smooth)
    - Takes the array of the mask\_image (has to be sitk object?)
    - Generates a template deformation field
    - Copy the mask\_image and then apply it to the deformation template
    - Applies gaussian smooth
    - Returns the mask\_image\_shift, dvf\_tfm, dvf\_template
  + Generate\_field\_asymmetric\_contract
  + Generate\_field\_asymmetric\_extend
  + Generate\_field\_Expand
  + Generate\_field\_Radial\_blend
* Image.py
  + Insert\_sphere
  + Insert\_cylinder
  + Insert\_sphere\_image
  + Insert\_cylinder\_image
* Mask.py
  + Get\_bone\_mask
    - Automatically generates binary mask of bones from a CT image
  + Get\_external\_masks
    - Generate binary mask of patient external contour

## Label

## Projects

## Registration

Linear.py

* Def alignment\_Registration
  + Procedure that can align images in a single step. Uses the images centre-of-mass to estimate the shift and rotation needed for alignment
  + Takes in a fixed image, moving image

## Tests

## Utils

## Visualization

* Animation.py
* Utils.py
  + Class VisualiseContour
  + Class VisualiseScalaryOverlay
  + Class VisualiseVectorOverlay
  + Class VisualiseComparisonOverlay
  + Class VisualiseBoundingBox
  + Def return\_slice(axis, index)
  + Def subsample\_vector\_field
    - Prepares a slice tuple to use for extracting a slice for rendering
  + Def vector\_image\_Grid
  + Def reorientate\_vector\_field
  + Def generate\_comaprison\_colormix
    - Function used to take in two images and then defines a colorspace
  + Def project\_onto\_arbitrary\_plane
* Visualiser.py
  + Class ImageVisualiser
    - Visualising images
    - Overlaying contours
    - Scalars+bounding boxes
    - Def Add\_contour
      * Contour – must be either contour mask or dict containing contour masks.
    - Def Add\_scalar\_overlay
    - Def add\_vector\_overlay
      * Overlays a vector field on to the existing image
    - Def add\_comparison\_overlay
      * Adds a comparison image on the existing image
    - Def add\_bounding\_box
      * Adds a bounding box to draw
    - Def show
      * Renders the image with all overlays
    - Def display\_slice
    - Def \_overlay\_comaprison

# Image Fusion Troubles

* ImageVisualiser Data Class takes in an image
* Contours -> must be sitk.Image or dict but not sure how to get the contours
  + Contour mask or dict container containing contour masks
  + What is considered as dict or contour mask
* Having troubles taking in dcm files
* Unsure if the image are taken as dcm or nci.gz files