David Morris v21/10/2024

**OPTIMAT Flow Chart**

|  |  |  |
| --- | --- | --- |
| **Code steps** | **Output folders** | **Check Steps** |
| A. OPTIMATSort | i. SortedDM | 1. Machine Learning |
| B. Dicom2Nifti | ii. MATDM | 2. ML SEG applied to FF |
| C. Machine Learning | iii. FFDM |  |
| D. Error checking | iv. Input |  |
| E. OPTIMATSEG2FF | v. Output |  |
|  | vi. FFnifti |  |
|  | vii. ROInifti |  |
|  | viii. CSV |  |

A. OPTIMATSort

ii. \*\_MATDM

iii. \*\_FFDMDM

i. \*\_SortedDM

B. Dicom2Nifti

C. Machine Learning

D. Error Checking *klearning i*

E. OPTIMATSEG2FF

iv. \*\_Input

v. \*\_Output

vi. \*\_Output**FFnifti**

vii. \*\_Output**ROInifti**

viii. \*\_Output**CSV**

1. Check Rotation is consistent

2. Check ROI is correct on FF map

Raw Data

Final Output

Overview:

* Naming conventions
* Notes to explain original context of code development and application, and terms specific to the UK Biobank MRI data.
* List of Code (Alphabetical)
* List of Outputs (Roman numerals)
* Check steps (Numerical)
* Final output
* Raw data locations
* See Example

**Raw Data**: Assume folder \* which contains an individual folder for each patient labelled as its EID (electronic ID for each UK Biobank participant). This structure is reproduced for all folders generated by OPTIMAT Sort

**Note 1:** this code pipeline was developed to work with bulk MRI data downloaded from UK Biobank and stored on University of Edinburgh servers. Code may need modification depending on the structure of the data that you are working with.

**Note 2:** In the lists below, we sometimes refer to the following:

* “sequences 7, 8, 15, 16, 19 & 20”. These relate to the MRI volumes (i.e. anatomical region covered by the scan) and image types (i.e. water image or fat image) provided by UK Biobank.
* “Spine/Head/Neck/Epi/TH/Dia”. These refer to the region of bone marrow being segmented by the deep learning code. Spine = vertebrae; Head = femoral head; Neck = femoral neck; Epi = femoral ephiphysis; TH = femoral total hip region; Dia = femoral diaphysis.
  + Each femoral volume is from the left (non-dominant) leg.
  + The Neck and Epi regions are combined to generate the TH (total hip region), as presented in our published papers (e.g. <https://doi.org/10.1016/j.csbj.2023.12.029>).
* An erosion step is applied to the segmentation volume generated by the UNet, in which the edge region is eroded by one voxel to ensure no bone tissue is included in the segmentation volume. This step is applied to the Spine, Head, Neck and Epi regions (and therefore also to TH), but not to the Diaphysis region. This is because the Diaphysis is so small that erosion can cause the volume to become too small and/or non-contiguous.

**List of Code**

1. OPTIMATSort
   * Code uploaded to GitHub in William Cawthorn’s OPTIMAT repository
     + Original location on University of Edinburgh servers was: OPTIMAT:\DMM\_CODE\ OPTIMATSort (matlab code 2019b)
   * Select the folder containing downloaded images
     + Initial location hard coded in line 10
   * If code aborts edit at line 23 to restart folder list
   * Line 28 must include the name of all non-dicom files in directory
   * Only sequences 7, 8, 15, 16, 19 & 20 are sorted to save space
   * Default smoothing hard coded at line 97
2. Dicom2Nifti
   * All the code is recommended to run on a ubuntu GPU server
   * Code (filename CW\_Dicom2Nifti\_MLcode) to be made available by Chengjia Wang (<https://github.com/chengjiawang>)
   * Open code (this is the path for the code in its original location on the University of Edinburgh server): OPTIMAT:\CW\_Dicom2Nifti\_MLCode\data\convDCM2niftiAll\_optimat\_bigloop.py
   * Set relative paths by modifying values of the following variables:
     + root\_path: where the output folders of step A located
     + out\_root\_path: where the output nifty files located, following the same subdirectory structure with the input folders
     + folder\_keyword: the folder keyword of the input files after the last “\_” in the file name
   * in the terminal, run “python convDCM2niftiAll\_optimat\_bigloop.py”
3. Machine Learning (code available from <https://github.com/chengjiawang/OPTIMAT_NET>).
   * In the same folder with Step B
   * open the file “train\_optimat\_xxx.py” where the xxx represent different training methods:
     + “\_dsup”: deep supervision based training suitable for small structures: Dia, Epi
     + “\_Attention”: attention based training suitable for relatively larger structures: Head, Neck, Spine
   * Check “options/options.py” file for all useful options for running the code:  
     an example: “python train\_optimat\_dsup.py --dataset Dia --gpu 1 --visualize\_segmentations --batch\_size 2 --model unet3D\_attention\_dsup --continue\_train”
     + –dataset: set the object for training (Dia/Epi/Head/Neck/Spine)
     + --gpu: set which gpu to use
     + --visualize\_segmentations: if present, display the segmentation using tensorboard
     + –batch\_size: batch size of training (input X volumes at the same time)
     + –model: which unet model to use
     + --continue\_train: whether to refine the best model obtained presently or train from scratch
   * The input folder is set by: --dataroot option, should be the output folder of Step Bb
   * The model output folder is set by: --model\_output, is the folder where the trained unet stored
   * After training, when applying the model to real data:
     + Use gennifti1file\_all\_dcm2niix.py
     + Set group name, input and output folder, patient group as in Step B
     + set the output folder of the segmentations using “output\_root” variable
     + run “python gennifti1file\_all\_dcm2niix.py”
4. Error checking
   * Initial error check: EverCheckNaive1.py
   * Set input and output paths as in Step C using the variable with the same name
   * run “python EverCheckNaive1.py”
     + The empty files will be removed from the ML output folder
     + Three csv files will be generated:
       1. “Empty\_files.csv”: all empty files after the first round of errer checking
       2. “Ini\_empty\_files.csv”: all empty files before the first round of errer checking
       3. “modified\_files.csv”: all files that are modified in the first round of errer checking
   * Setup and run “UpdateShapeStat.py”:
     + This will updating the shape statistics of the manual segmentations
     + The statistics will be saved at the same folder with the code
   * Setup and run “EverCheckNaive\_std.py” for the second round of error checking:
     + This will check the segmentation based on the updated statistics
     + Run “EverCheckNaive\_shapeplot.py” if you want to plot the statistics as histograms
     + Strange output will be recorded in “outlier\_files.csv”
5. OPTIMATSEG2FF
   * Code uploaded to GitHub in William Cawthorn’s OPTIMAT repository
     + Original location on University of Edinburgh servers was: OPTIMAT:\DMM\_CODE\ OPTIMATSEG2FF (matlab code 2019b)
   * First select window is all regions to be applied (single or multiple)
     + Initial location hard coded in line 4
   * Second select window is folder contain fat fraction maps (FFDM)
     + Initial location hard coded in line 6
   * If code aborts edit at line 23 to restart from that number file
   * Based on UK biobank naming from initial download

**List of Outputs**

1. All dicom files from sequences 7, 8, 15, 16, 19 & 20 in their own folder to allow for conversion to NIfTI format for machine learning in a reasonable time
   * Name is folder’s name from Raw data\Folder for each sequence named Seriesnumber\_Seriesname\_EID
2. Matlab format (.MAT) of sequences in case we need to reprocess the Fat Fractions
3. Fat fraction maps in .MAT format: 7&8 give *Back*, 15&16 give *Hip*, 19&20 give *Leg*
   * Name is Edinburgh ID\_*Region*\_FF\*
   * \* gives level of smoothing, default 2 (Matlab Gaussian)
4. NIfTI files fed into the machine learning orientation needs to be consistent with v
   * Name is EID\_Region\_Group(no longer relevant =0) *– Chengjia?*
5. NIfTI outputs from the machine learning orientation needs to be consistent with iv
   * Name is EID\_Segmenation\_Group(no longer relevant =0)\_ML *– Chengjia?*
   * Segmentation is Spine from input back, Head/Neck/Epi from Hip and Dia from Leg
6. NIfTI FF maps orientation needs to be consistent with vii
   * Name is as III ID\_*Region*\_FF\* plus \_DMM = ID\_*Region*\_FF\*\_DMM
7. NIfTI ROI orientation needs to be consistent with vii
   * Name is as v EID\_Segmenation\_Group)\_ML plus \_DMM = EID\_Segmenation\_Group\_ML\_DMM
   * Additional nifty EID\_Segmenation\_Group\_MLr\_DMM cleaning the outer four slices at each end of the volume to prevent regions in fold over artefact regions & restricting Dia to 9 slices
     + EID\_Segmenation\_Group\_MLr corresponding MAT file
   * Additional nifty EID\_Segmenation\_Group\_MLre\_DMM eroding the clean region by one edge voxel (3x3 Gaussian smooth)
     + EID\_Segmenation\_Group\_MLre corresponding MAT file
8. CSV for each input segmentation creates output folder from input folder eg \*\_Output
   * Output name eg 1000189\_1000189\_Back\_FF2\_1000189\_Spine\_G0\_MLr
     + ID 1000189
     + Name of Fat fraction map used 1000189\_Back\_FF2
     + Name of regions used 1000189\_Spine\_G0\_MLr as saved in ROInifti

r and re as defined in vii

* + Contents of CSV
    - Column 1 – Name (Edinburgh ID)
    - Column 2 – Location (**S**pine, **H**ead, **N**eck, **E**pi, **D**ia) *ADDED TotalHip* ***T***
    - Column 3 – Region (r or re)
    - Column 4 – Mean (Mean FF in region)
    - Column 5 –SD (standard deviation of FF in region)
    - Column 6 –Median (Median of FF in region)
    - Column 7 – IQR (Inter quartile range of FF in region)
    - Column 8 – Size (Voxels in region)

**Check Steps**: Using MRICRO to overlay Nifti images or IMageJ to go side by side

1. Input is fat image and output is a binary mask that should overlay the specified segmentation
2. The ROI a binary mask and FF is the smoothed Fat Fraction map and the ROI should overly the specified segmentation

**N.B.** The image orientations will not be the same and the raw dicom images and 1 will be different to 2 but consistent within 1 &2

**Final output**: CSVs need to be sorted in Excel/R/etc us re for Spine/Head/Neck/Epi/TH segmentations and r for Dia

**REVISION FOR TotalHip segmentations:**

* + Code revised in OPTIMAT:\HandoverDMM2023OPTIMAT\OPTIMATcode2023
    - OPTIMATSort2023 version used in all analysis
    - OPTIMATSEG2FF2022 original version pre 2022 no Total hip – used to date
    - OPTIMATSEG2FF\_TH2022 analyses only TH regions – Used to date
    - OPTIMATSEG2FF2023 – All region together tested but not used yet

**Example save 2023**: OPTIMAT:\HandoverDMM2023OPTIMAT\Exampledata

* + This folder (on the University of Edinburgh servers) shows examples of the outputs generated using this code. This cannot be publicly shared because it includes UK Biobank participant data, for which access requires users to have an approved UK Biobank project.