

A Software for Medical Image Labeled Data Analysis

Ezequiel de la Rosa, Umamaheswaran Raman Kumar

January 11, 2018

1 INTRODUCTION

The use of labeled medical images plays, currently, a key role in medicine and bio-imaging. Several applications dependant on labeled data can be mention, such as anatomical and physiological parameters measurement with diagnostic, follow up or therapy decision support purposes. The use of real time surgical procedures requires, as well, the identification and segmentation of determined organs/tissues in order to give to the surgeon feedback about the intervention. On the other hand, from a technical imaging point of view, the use of automatized tools and softwares for fast and user-free anatomical and pathological structures segmentation is raising considerably. Thus, in all these applications, labeled images are crucial for developing robust and precise systems.

Working with labeled data requires, as well, specific evaluation tools for assessing performances, similarities, and in order to qualify the wellness or reliability of a segmentation against others. For instance, when deciding weather to use an algorithm instead of other for inclusion in a medical software, some informative metrics are needed to help us take this decision.

In this project, a general software for evaluating medical labeled image data it is proposed. It considers the different scenarios that can be found when working with annotated (labeled) images tackling, mainly, the problems of data unification and performance assessment. The proposed software is easy and fast to use, friendly to the user and fully automatic, allowing the rapid and simultaneous evaluation of big amounts of labeled volumes.

2 OBJECTIVES

2.1 MAIN GOAL

To develop a software application with Graphic-User-Interface able to work with labeled volume data for inter-rater comparison, evaluation and ground truth generation.

2.2 SPECIFIC GOALS

The software will tackle two typical problems encountered when working with labeled data:

- Generation of unique ground truth volumes from several raters annotations.
- Comparison between labeled generated and ground truth data by means of typical performance metrics.

3 METHODS

3.1 SOFTWARE PROPOSAL

In this project, a software with functions and features for dealing with labeled data is proposed under a C++, ITK and VTK environment. The main idea is that the user can read, simultaneously, labeled data (annotations) coming from different raters, segmentation algorithms or ground truths. Afterwards, the software should automatically perform the analysis over all the volumes coming from different origins, without any extra user work or interaction with the software. The proposed software is mainly able to address two different scenarios related with labeled data and medical image ground truth generation:

3.1.1 GROUND TRUTH UNIFICATION

When new medical image datasets are generated with the aim of conducting specific experiments (such as in experimental, pre-clinical, and clinical research), it is sometimes necessary to provide manual segmentations/annotations of the images involved in the dataset. The purposes of dividing the images in regions and in labelling them can follow several goals. For instance, the measurement of clinical and physiological indexes, the measurement of lesion sizes, the tracking of an organ in real time interventions, among others. In order to avoid biased results, and considering that among doctors and physicians always exist differences in the manual labelling, the segmentation process is assigned to generally more than one professional. Thus, with different segmentations coming from different professionals, it is possible to understand the variability among the observers and is possible to generate, as well, more accurate ground truths. A current method able to unify the annotations coming from different human raters is STAPLE (Simultaneous truth and performance level estimation) [4], represented in Fig. 3.1. The algorithm is based on an expectation-maximization iterative process and after considering a set of segmentations, it computes an estimate (probabilistically) of the real true segmentation. Thus, it produces a performance measure for the different segmentations involved in the set. It is also important to highlight that the algorithm can

be used not only for human raters segmentations, but also for unifying automatic algorithm-based segmentations. In fact, segmentation algorithms have been proposed by using STAPLE technique [2].

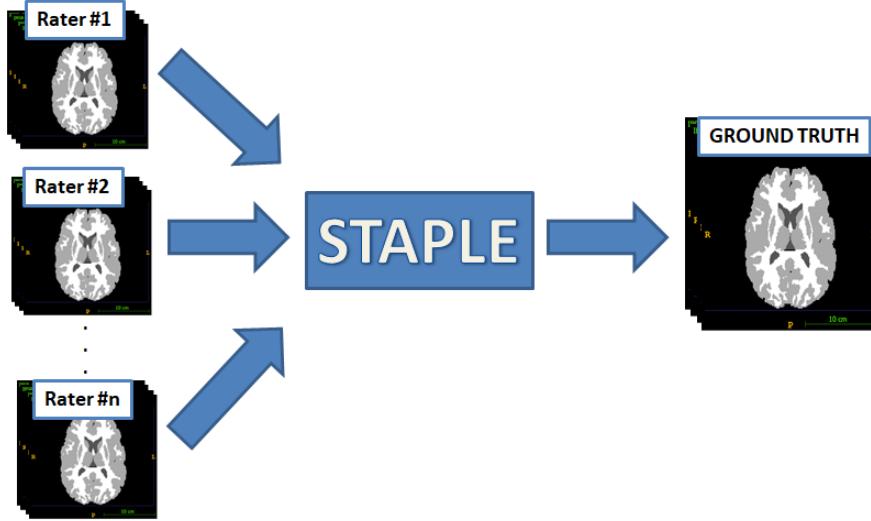


Figure 3.1: STAPLE data unification.

3.1.2 INTER-RATER COMPARISON

When testing the performance of segmentation algorithms with the aim of assessing the agreement with the ground truth data, several measures can be computed. In this project, some common indexes are incorporated with the aim of allowing the user to compare, automatically, several labeled data cases against the ground truth. The indexes that our software computes are the ones listed below, where S represents a source image and T a target one [3] based on the figure 3.2.

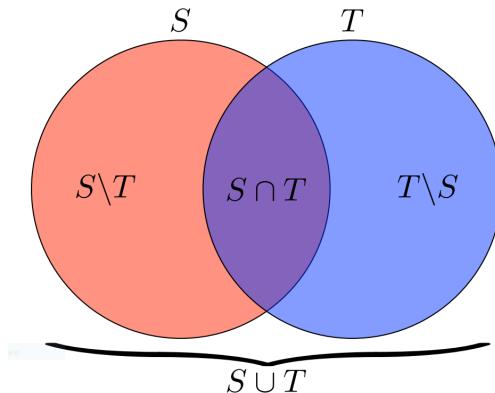


Figure 3.2: Venn diagram for understanding overlapping measures and errors [3].

3.1.3 OVERLAP MEASURES

Target Overlap :

$$TO_R = \frac{|S_r \cap T_r|}{|T_r|} \quad (3.1)$$

Jaccard Coefficient :

$$JC = 2 \frac{|S_r \cap T_r|}{|S_r \cup T_r|} \quad (3.2)$$

Dice Coefficient :

$$DC = 2 \frac{|S_r \cap T_r|}{|S_r| + |T_r|} \quad (3.3)$$

Volume similarity:

$$VS = 2 \frac{|S_r| - |T_r|}{|S_r| + |T_r|} \quad (3.4)$$

3.1.4 OVERLAP ERROR MEASURES

False Negative Error:

$$FN = \frac{|T_r \setminus S_r|}{|T_r|} \quad (3.5)$$

False Positive Error:

$$FP = \frac{|S_r \setminus T_r|}{|S_r|} \quad (3.6)$$

It is evident from these equations that overlap metrics can achieve values in between 0 (minimum, worst performance) and 1 (maximum, best performance), except for the *Volume Similarity*, which can reach values in between -1 (under-estimation of the volume) and 1 (over-estimation). Besides, a zero value of this last metric means a perfect performance. On the other side, overlap error metrics can obtain values between 0 and 1, representing the former value the best performance and the latter one the worst performance.

It is important to highlight that all the equations provided here work with individual labeled regions. The corresponding ones for working with more than one label are found in [3]. In this project, single and multi-label measures are proposed, thus the user can choose how to compute them.

3.2 IMPLEMENTATION

The software is implemented in C++ using *Qt* for GUI(Graphical User Interface), *VTK(Visualization Toolkit)* for visualizing 3d segmentation volumes and *ITK(Insight Toolkit)* for handling and processing the volumes. The below subsections describe the HLD(High Level Design) with Use Case Diagram, GUI(Graphical User Interface) and LLD(Low Level Design) using sequence diagrams for the software application developed.

3.2.1 USE CASE DIAGRAM

Figure 3.3 shows the use case diagram of the application where the main features are represented by bubbles, the association between the user and the system are represented by normal arrows and the dependencies between the features are represented by dotted arrows.

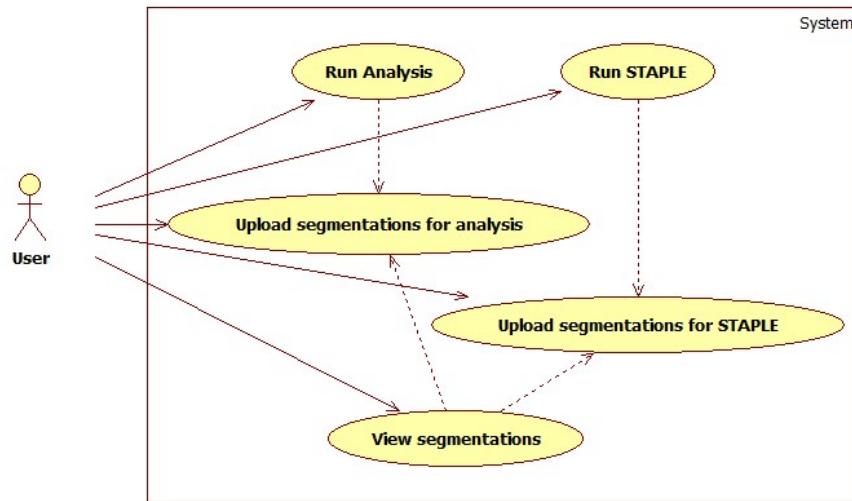


Figure 3.3: Application Use Case diagram.

3.2.2 GRAPHICAL USER INTERFACE (GUI)

Figure 3.4 shows the user interface developed in Qt for the application. The regions highlighted in red boxes are explained below in the same order.

1. **Menu bar :** The menu bar is created using `QMenuBar` class. It has 4 actions corresponding to the 4 menu items given below.

- *File -> Browse -> Analysis* : Opens the file browser window to browse for the parent folder containing the segmentations in different sub-folders and populate the file names in the analysis list widgets corresponding to each sub-folder. It is mandatory to have a sub-folder with name '*Ground Truth*' or else the files will not be read.
- *File -> Browse -> STAPLE* : Opens the file browser window to browse for the parent folder containing the segmentations from different raters or different segmentation algorithms in sub-folders and populate the file names in the STAPLE list widgets corresponding to each sub-folder.
- *Run -> Analysis* : Call the corresponding action to calculate all the metrics and save it in an excel file.
- *Run -> STAPLE* : Call the corresponding action to unify different observer segmentation using STAPLE and save as separate '.nii' files.

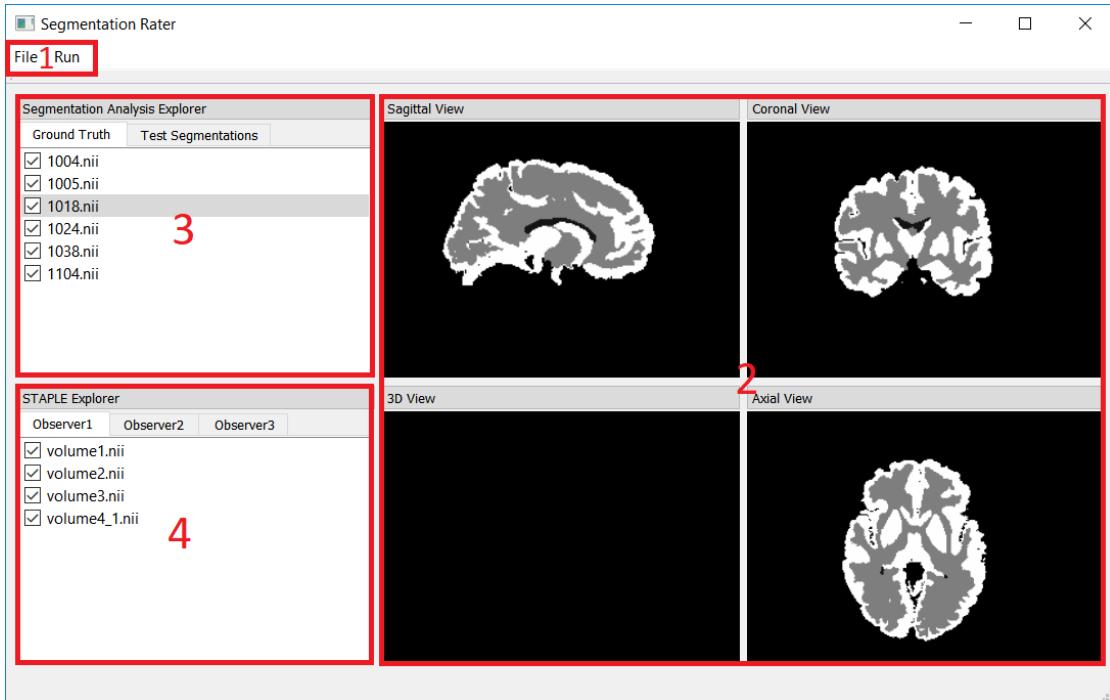


Figure 3.4: Application Graphical User Interface.

2. **Visualization widget :** This widget is a *QVTKWidget* that allow us to view the volumes when any volume is selected in the list widget. The renderer of the widget is connected to the renderer of *vtkResliceViewer* which provides additional functionalities to the viewer like windowing, scrolling through the slices and zooming.
3. **Segmentation analysis tab widget :** The explorer tab widget for viewing segmentation volumes with ground truth is created using *QTabWidget*. For each sub folder a separate *QListWidget* is added to the tab widget. The user is allowed to check/uncheck only the ground truth volumes which should be used for generating analysis metrics.
4. **STAPLE tab widget :** The tab widget is very similar to the analysis tab widget with the only difference being that, the user is allowed to select volumes on all the tabs which he wants to be included in the final STAPLE algorithm.

3.2.3 SEQUENCE DIAGRAM

The sequence diagram clearly shows the control flow between the objects interacting with each other in an orderly fashion and their life time within the application. This section gives the sequence diagrams for few of the use cases shown in figure 3.3. The function names written in all the sequence diagrams do not reflect the actual names of the function from the code but only for understanding purpose as the actual control flow includes more complexities.

1. Browse for analysis/STAPLE segmentation volumes:

Figure 3.5 shows the control flow of 'Browse' menu for importing segmentation volumes for both analysis and STAPLE actions.

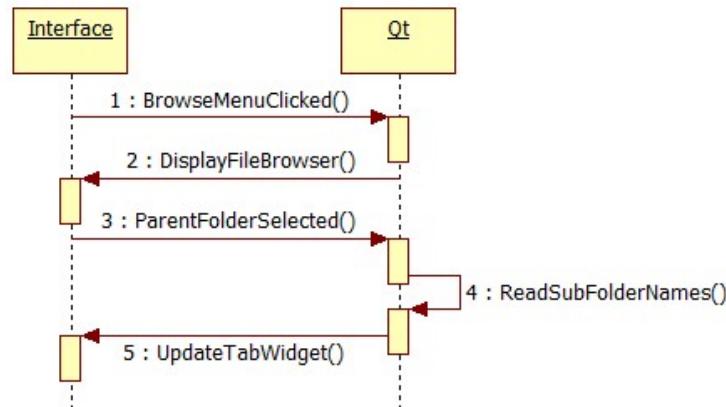


Figure 3.5: Sequence diagram for Browse menu.

2. Visualizing segmentation volumes:

Figure 3.6 shows the control flow when the user selects a volume to visualize it in the visualizer. It clearly depicts the abstraction provided when using a GUI application.

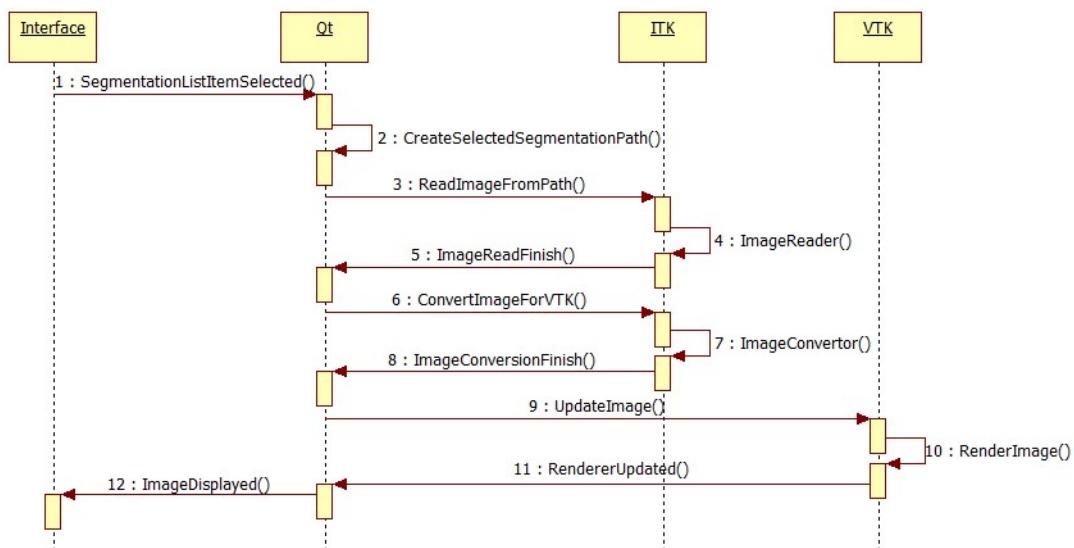


Figure 3.6: Sequence Diagram for visualizing segmentation volumes.

3. Run analysis:

Figure 3.7 shows the control flow of '*Run -> Analysis*' menu option and it is very similar to '*Run -> STAPLE*' menu option, but the only difference being that the Qt requests ITK for running the STAPLE algorithm instead of requesting for the metrics.

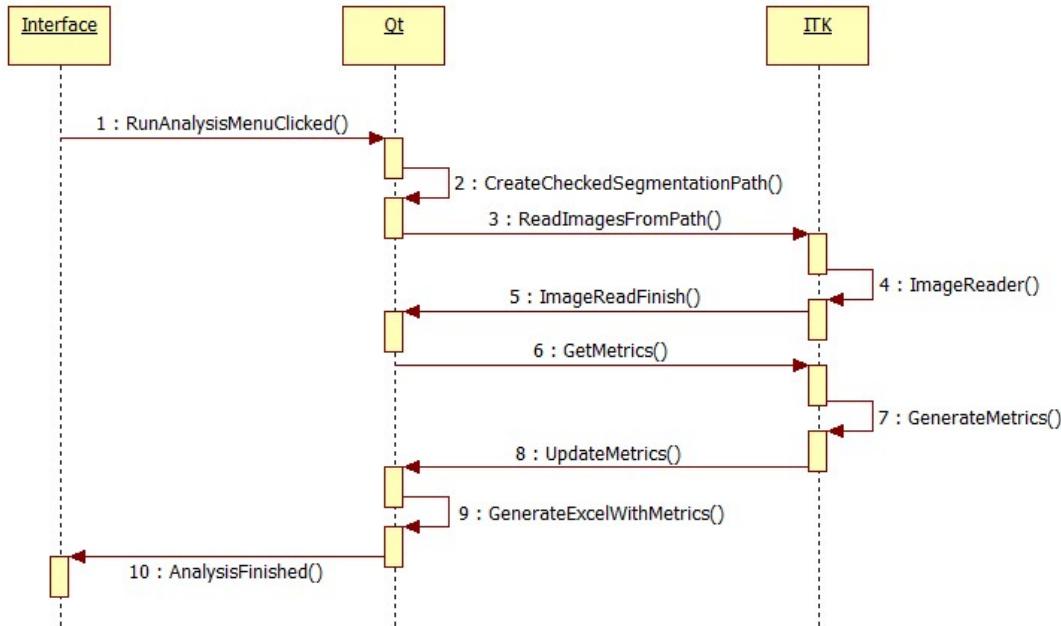


Figure 3.7: Sequence Diagram for Run Analysis menu.

4 RESULTS

4.1 GROUND TRUTH UNIFICATION

Once the STAPLE algorithm is run over the selected cases, the software creates, automatically, a folder containing the *.nii* exported volume, located in the same path where the labeled data is. The STAPLE volume obtained for a 'toy' labeled data example can be appreciated on Fig 4.1, either with the original volumes that generated it. The idea of using a toy volume is for understanding purposes, since it is easier to interpret the obtained volume. In the figure, it is worth to say that since STAPLE added an extra label for those pixels that cannot take a decision (for instance, equal probability of belonging to each class), when showing it on ITK it can be seen an extra color. However, if we look to the intensity values displayed on the left panel, it is observed the agreement provided by STAPLE (in the pixel being pointed, when the majority volumes provided higher probability for label #2, STAPLE provided the same label, although it is displayed with a different color).

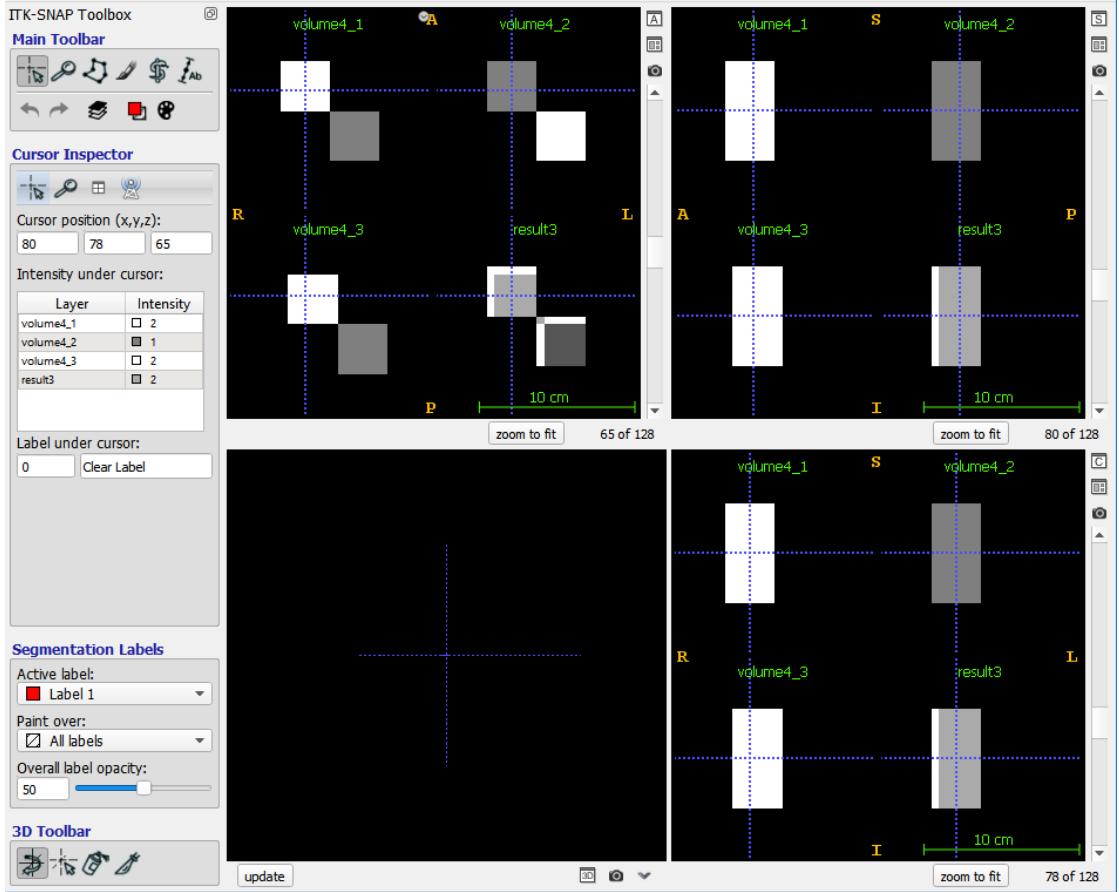


Figure 4.1: Labeled volumes corresponding to three simulated raters with their corresponding STAPLE unified ground truth.

Moreover, for showing a real medical image data case, we apply STAPLE in the proposed software for a dataset of labeled volumes corresponding to the 2012 STACOM (Statistical Atlases and Computational Models of the Heart) MICCAI challenge [1]. We consider three original segmentations of a myocardial infarcted heart provided by different raters, belonging to late gadolinium enhance MRI cases. The STAPLE result obtained for one patient can be appreciated on Fig 4.2. In the image, it is shown in ITK-Snap a small over-segmentation provided by one rater (small 'island' where the cursor is). The labels, shown in the left panel, specify that only the method #1 (first rater) considers those pixels as scar tissue, meanwhile for the two other raters (methods #3 and #4), those pixels are healthy tissue. As a result, in this higher probability situation of being 0 (healthy) against infarcted (1), STAPLE chose the highest likelihood option.

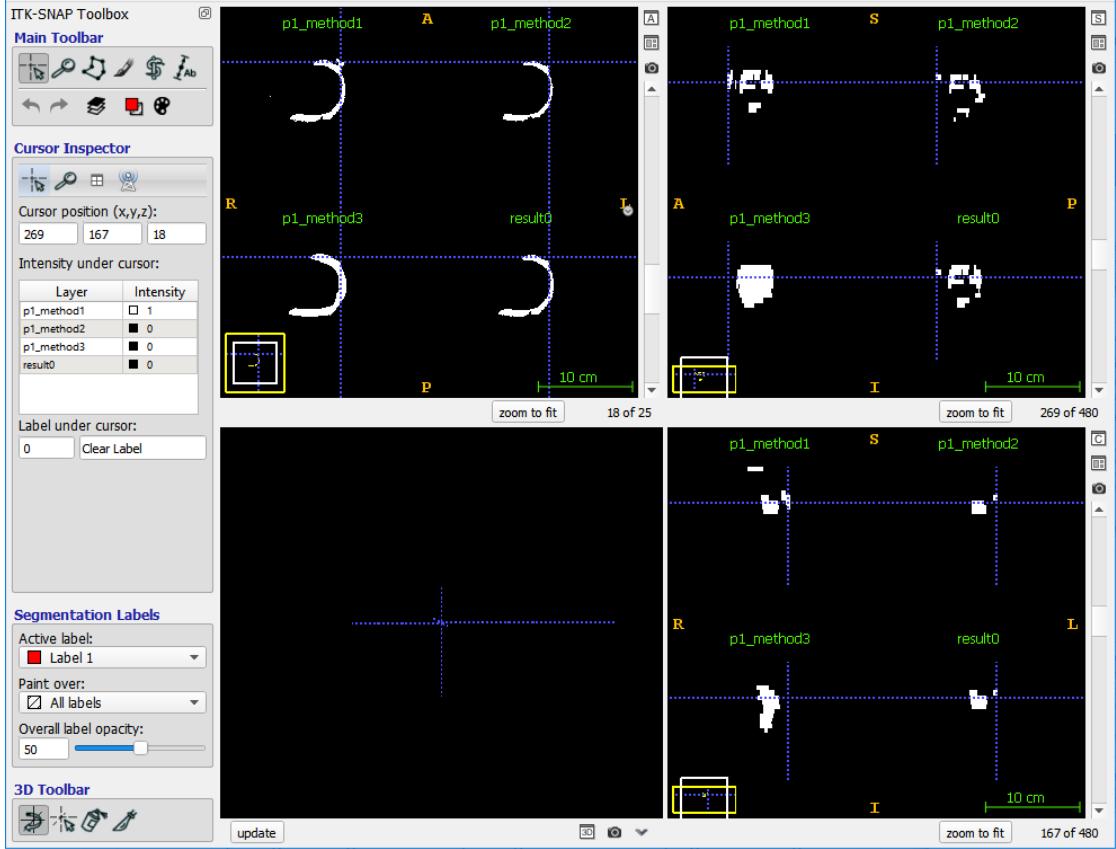


Figure 4.2: Labeled volumes corresponding to three real raters with their corresponding STAPLE unified ground truth. Volumes were taken from [1].

4.2 SIMILARITY METRICS AND AGREEMENT

Again, results are automatically exported in external folders that can be found in the directory passed for reading the images. On Fig 4.3, the exported .xls file can be appreciated. We can see in the red box, the similarity metrics obtained when comparing the segmentations (coming from method #1) against the ground truth data. In each column, the different metrics are displayed, meanwhile in each row, the result obtained for each considered volume (case) of the dataset .

On the other side, in the green box (Excel sheets) the same table before described can be found, but now containing the results of comparing the ground truth with the segmentations of the method #2. For each extra folder of segmentations (methods/rater) added in the image directory, one extra sheet with results is provided.

SegmentationMeasures [Modo de compatibilidad] - Microsoft Excel

	A	B	C	D	E	F	G	H	I	J
1	Ground Truth Volume	Segmentation Volume	Dice Coefficient	Jaccard Coefficient	Mean Overlap	Total Overlap	Union Overlap	False Positive Error	False Negative Error	Volume Similarity
2	p1_gtruth.nii	p1_method2.nii	0.316399407	0.187930206	0.316399407	0.589431405	0.187930206	0.783763872	0.410568595	0.92642501
3	p2_gtruth.nii	p2_method2.nii	0	0	0	0	0	1	1	0.143551402
4	p3_gtruth.nii	p3_method2.nii	0.132545932	0.07097681	0.132545932	0.15975957	0.07097681	0.886745907	0.84024043	0.340682415
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										

Figure 4.3: Sequence Diagram for Run Analysis menu.

5 PROJECT MANAGEMENT

In order to handle the project time deadlines and goals, we organized a tentative plan of meetings and activities to conduct during the working time. The timetables we have fixed for our project and how they were reached can be seen on the tables (5.1 and 5.2). It is important to point out that setting the tools needed in the project (i.e., building QT, ITK and VTK) took a considerable amount of time, much more than the expected one.

Table 5.1: Weekly tasks

Week	Main Tasks
Week 1	Project delimitation Software features discussion Check feasibility with different tools (Matlab, C++ and MeVisLab)
Week 2	Distribution of the project tasks for POC Set up Qt environment Build VTK and ITK for MinGW applications
Week 3	Familiarize with VTK and ITK Implement 3D volume read and write
Week 4	Basic QT GUI implementation Connect VTK reslice viewer with Qt widget to display 3D volumes as slice STAPLE algorithm implementation
Week 5	Similarity metrics implementation for label images Write to excel files Report
Week 6	Running experiments with real data. Report

Table 5.2: Number of working hours

Team Member	Tasks	Week 1-3(hours)	Week 4-6(hours)
Umamaheswaran	Learning	10	12
	Environment Setup	20	0
	Coding	5	12
	Experiments	15	25
	Presentation & Report	3	10
Ezequiel	Learning	15	10
	Environment Setup	4	0
	Coding	5	8
	Experiments	10	15
	Report	3	15

Deadlines were mutually decided and scheduled by considering the amount of time required for each task. Faced problems and troubleshooting were reported. For handling the project communication between team-mates and to monitor its evolution, the following tools were used: I) Google Drive for version control and file sharing, II) GoogleDocs for presentation, III) ShareLatex for report writing and IV) Hangouts of google as communication platform.

6 CONCLUSION & FUTURE WORK

In this project, a basic software application for accelerating medical labeled data assessment is proposed. The software is based on ITK, VTK and C++ and allows, automatically, to unify labeled data using STAPLE algorithm as well as allows to conduct labeled data agreement evaluation. The GUI includes a 4 view slicing image visualizer that allows to inspect volumes in the three axis (the original one and two reconstructions). Some features that can be highlighted about the developed software are listed below:

- Fast and easy to use.
- Automatically operates over all the volumes considered in one folder, and over all the folders which are assumed to come from different raters.
- Allows visualization of the data by means of four display windows (original slices and projections are visualized).
- Incorporates the widely used STAPLE algorithm.
- Allows to assess segmentation agreement by typical measures.
- Volumes generated using STAPLE are automatically exported in *.nii* format.
- Similarity metrics are automatically exported in *.xls* format. This allows fast plots/diagrams development in posterior analysis, as well as fast statistical results assessment. It is worth to say that this format can be easily read by most programming languages.
- Similarity metrics are divided by sheets in the *.xls* file, making easier posterior analysis.
- The code implementation allows to easily incorporate new features and metrics.

6.1 FUTURE WORK

In this first software proposal, basic tools for working with labeled data were implemented. Some improvements that can be consider as future work are:

- Inclusion of other similarity metrics, such as Haussdorf distance, relative volume difference, average boundary distance, among others.
- Inclusion of plots for assessing results correlation (Pearson, Spearman, Kendall) and data distribution.

- Automatized measurement of areas and volumes on the labeled data. With this metrics, correlations and statistical tests can be performed.
- Bland-Altman plots inclusion for assessing data agreement.
- Inclusion of inter-rater agreement or concordance metrics. Some useful metrics that could be included are Cohen's kappa, Scott's pi and Fleiss' kappa.
- Automatic statistical tests conduction in order to look for statistical significant differences among parameters. Parametric (such as t-test) and non-parametric (such as Mann-Whitney U-test) tests might be considered.
- Visualization windows can be exploited by showing, for instance, image differences (false positive and false negative areas highlighted). Besides, the real volumes could be visualized.

REFERENCES

- [1] Rashed Karim et al. “Infarct segmentation challenge on delayed enhancement MRI of the left ventricle”. In: *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer. 2012, pp. 97–104.
- [2] Xiaofeng Liu et al. “iSTAPLE: improved label fusion for segmentation by combining STAPLE with image intensity.” In: *Medical Imaging: Image Processing*. 2013, 86692O.
- [3] NJ Tustison and JC Gee. “Introducing Dice, Jaccard, and other label overlap measures to ITK”. In: *Insight J* (2009), pp. 1–4.
- [4] Simon K Warfield, Kelly H Zou, and William M Wells. “Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation”. In: *IEEE transactions on medical imaging* 23.7 (2004), pp. 903–921.