
pyult

Release 0.0.0

Motoki Saito

Jul 27, 2022

CONTENTS:

1	Quickstart	3
1.1	Installation	3
1.2	Necessary files: Exported files by Articulate Assistant Advanced	3
1.3	Terminology	3
1.4	Generate dataframes and pictures automatically from ultrasound files	4
1.5	Additional preprocesses	5
1.5.1	Flipping of images	5
1.5.2	Reduction of y-axis resolution	5
1.5.3	Cropping (Trimming) of the four sides of images	6
1.5.4	Fitting spline curves	6
1.5.5	Parallel processing	6
2	Installation	7
2.1	Supported systems and versions	7
2.2	Linux	7
3	About	9
3.1	Authors	9
3.2	Contact	9
3.3	Citation	9
3.4	Acknowledgements	10
	Python Module Index	11
	Index	13

pyult helps to preprocess ultrasound files exported by Articulate Assistant Advanced (AAA), an ultrasound-recording software. It can produce dataframes, images, and videos from ultrasound files.

QUICKSTART

1.1 Installation

First, you need to install *pyult*. *pyult* has been tested only on 64-bit Linux systems.

```
pip install --user pyult
```

1.2 Necessary files: Exported files by Articulate Assistant Advanced

pyult provides a series of functions to preprocess ultrasound-related files exported by [Articulate Assistant Advanced \(AAA\)](#), which is a product by Articulate Instruments Ltd¹.

Exported files by AAA for each recording usually consist of the main ultrasound file with the extension *.ult*, a parameter file (which usually ends with *...US.txt*), a meta-information text file containing information such as prompts (words/phrases displayed on a screen during recording) with the extension *.txt*, and an audio file (*.wav*).

Additionally, TextGrid files may be necessary, if you would like to include segment-information in the dataframes produced by *pyult*. TextGrid files can be produced by using forced-alignment programs.

1.3 Terminology

ult file

Files that have the extension “.ult”, e.g. recording01.ult. A ult file has a vector of brightness values for recorded ultrasound images.

ustxt file

Files that have the ending “US.txt”, e.g. recording01US.txt. A ustxt file has the information (parameters) necessary to (re)construct ultrasound images or videos out of a vector of brightness values from a ult file.

txt file

Text files that have the extension “.txt” but not “US.txt”, e.g. recording01.txt (not recording01US.txt). A txt file has the information about the participant, the prompt, and the date of the recording.

wav file

Audio files that have the extension “.wav”, e.g. recording01_Track0.wav.

recording

In this package (and thus this documentation), ‘recording’ refers to a set of the relevant files sharing the same

¹ Articulate Instruments Ltd. (2012). Articulate Assistant Advanced User Guide: Version 2.14. Edinburgh, UK: Articulate Instruments Ltd.

basename. For example, the recording “foobar01” refers to a set of “foobar01.ult”, “foobar01US.txt”, “foobar01.txt”, “foobar01.wav”, and so on. Its basename is “foobar01” in this case.

session

Usually, more than one item or participant are recorded. They are exported separately by AAA. In this package, ‘session’ refers to the directory that contains multiple recordings. For example, one session may contain “foobar01.ult”, “foobar01US.txt”, “foobar01.wav”, ..., “foobar02.ult”, “foobar02US.txt”, “foobar02.wav”, ... and so on.

```
session_01
├── foobar01.txt
├── foobar01US.txt
├── foobar01.ult
├── foobar01_Track0.wav
├── foobar02.txt
├── foobar02US.txt
├── foobar02.ult
└── foobar02_Track0.wav
```

1.4 Generate dataframes and pictures automatically from ultrasound files

pyult can generate dataframes and pictures from ultrasound recordings (exported files) by the following command:

```
python -m pyult
```

This command requires at least two arguments: path to the target directory and what task the script should do. They can be provided by `-d` (–directory) and `-t` (–task) options as below:

```
python -m pyult -d /foo/bar -t df
```

The target directory must have all the necessary files with corresponding names, i.e. the main ultrasound files (xxx.ult), parameter files (xxxUS.txt), meta-information files (xxx.txt), and audio files (xxx_Track0.wav) for each recording. “US” of parameter files and track numbers of audio files are ignored. Therefore, the target directory should have such a structure as below:

```
session_01
├── recording_01.txt
├── recording_01US.txt
├── recording_01.ult
├── recording_01_Track0.wav
├── recording_02.txt
├── recording_02US.txt
├── recording_02.ult
└── recording_02_Track0.wav
```

For tasks, “df”, “raw”, “squ”, “fan” and “video” are available for now. “df” produces dataframes for each recording. “raw”, “squ”, and “fan” produces images in the raw rectangle, square, and fan-shapes for each. “video” produces videos for each recording.

With “-t df”, the package generates dataframes in the long format. These dataframes have brightness values, corresponding x- and y-values, frames, and time at least. If TextGrid files are provided, corresponding segments and words

are included, too. These dataframes are expected to be further used by another program such as R to carry out statistical analyses, e.g. fitting Generalized Additive Mixed-effects Models.

“raw”, “squ”, and “fan” all produce images for each frame for each recording but in different shapes. Ultrasound images are inherently rectangle. Typical fan shapes need to be created from these rectangle images with interpolation. With “raw”, *pyult* produces rectable shapes of (raw) images. With “fan”, *pyult* interpolates images and creates fan-shaped images. “squ” can be used to produce square shapes of images.

Usually, corresponding recording parameters from parameter files (i.e. xxxUS.txt) are used to reconstruct fan-shaped pictures from a vector of brightness values. However, the resultant images tend to be small. If you would like to magnify the size of output fan-shaped images, you can do so by the option “-m (-magnify)”:

```
python -m pyult -d /foo/bar -t fan -m 4
```

which makes lengths of x- and y-axis about 4 times bigger than original lengths. But please note that magnification of images takes much longer time and so please use it carefully.

“video” produces videos from the main ultrasound file, parameter files, and audio files, for each recording.

For the time being, the package cannot handle multiple tasks at the same time. This feature is to be implemented soon.

1.5 Additional preprocesses

1.5.1 Flipping of images

Horizontal and vertical directions of images can be flipped before producing them as png files. Give “x” for horizontal flip, “y” for vertical flip, and “xy” for flipping in both directions. Therefore, for example, if you would like to flip all the produced images along x-axis (hotizontally), then...:

```
python -m pyult -d /foo/bar -t df -f x
```

1.5.2 Reduction of y-axis resolution

Ultrasound images are inherently very thin rectangles with greater ranges for y-axis (greater height). In other words, ultrasound images have much more information for vertical direction than horizontal direction. Therefore, in some cases, reduction of y-axis length does not harm at all but can contribute a lot to reduce overall data size.

Currently, *pyult* supports to reduce y-axis length by taking every *n*-th pixel along y-axis. Accordingly, if you would like to take every 3rd pixel along y-axis to compress the size of the produced dataframes into one third of the original, then...:

```
python -m pyult -d /foo/bar -t df -r 3
```

Likewise, you can produce fan-shaped images with its y-axis length one fifth of the original by the following:

```
python -m pyult -d /foo/bar -t fan -r 5
```

1.5.3 Cropping (Trimming) of the four sides of images

Sometimes, some regions of ultrasound images are not given the main focus. Recording movements of the tongue, for example, very bright parts on the bottom of images (which are assumed to represent skin parts) can be trimmed to focus on the interested parts (e.g. tongue surface).

Cropping of images can be achieved by the option “-cr (-crop)”. Minimum and maximum values along x- and y-axis should be given in the format such as “minX,maxX,minY,maxY” without any space. So, for example, the following command produces the cropped images, where x-axis values start at the 10th pixel and ends at the 50th pixel of the original images, and where y-axis starts at the 120th pixel and ends at the 600th pixel of the original images:

```
python -m pyult -d /foo/bar -t fan -cr 10,50,120,600
```

1.5.4 Fitting spline curves

Recording tongue movements, the main attention is sometimes given only to the tongue surface positions. Although *pyult* is designed for the analysis of the whole ultrasound images, rather than focusing on the tongue surfaces, it is also possible with *pyult* to find and fit spline curves on the tongue surfaces. For the spline fitting, simply feed “-s (-spline)” as below:

```
python -m pyult -d /foo/bar -t fan -s
```

1.5.5 Parallel processing

Preprocessing by *pyult*, introduced above, can be carried out in parallel for each recording. Please note that the package is implemented only with parallelization for each recording, not within one recording. Therefore, if you have 2 recordings like the following:

```
session_01
├── recording_01.txt
├── recording_01US.txt
├── recording_01.ult
├── recording_01_Track0.wav
├── recording_02.txt
├── recording_02US.txt
├── recording_02.ult
└── recording_02_Track0.wav
```

then you can parallelize the preprocessing by 2 cores at most (with -co or -cores):

```
python -m pyult -d /foo/bar -t fan -co 2
```

INSTALLATION

2.1 Supported systems and versions

pyult has been tested only on 64-bit Linux systems.

2.2 Linux

```
pip install --user pyult
```


3.1 Authors

pyult has mainly been developed by [Motoki Saito](#) with help by [Konstantin Sering](#).

3.2 Contact

If you want to contact the main developer of the package (Motoki Saito), please email to
motoki [dot] saito [at] uni [minus] tuebingen [dot] de

3.3 Citation

I would be more than happy if this package could help your work in any form. In that case, I would be very grateful if you would cite this package:

Saito, Motoki. (2020). Pyult: Preprocessing utilities for ultrasound data in Python. <https://doi.org/10.5281/zenodo.4022838>

For BibTex:

```
@misc{pyult,  
  author      = {Motoki Saito},  
  title       = {Pyult: Preprocessing utilities for ultrasound data in Python},  
  year        = {2020},  
  doi         = {10.5281/zenodo.4022838},  
  url         = {https://doi.org/10.5281/zenodo.4022838}  
}
```

3.4 Acknowledgements

(To be updated)

PYTHON MODULE INDEX

p

pyult, 1

INDEX

M

module

 pyult, 1

P

pyult

 module, 1