

PYSOUNDANALYSER MANUAL

Version 0.1

Samuele Carcagno
sam.carcagno@gmail.com

Copyright ©2012–2013 Samuele Carcagno.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Disclaimer: This document comes with NO WARRANTY whatsoever of being correct in any of its parts. This document is work in progress.

Contents

1	Introduction	1
2	User Interface	2

Chapter 1

Introduction

Chapter 2

User Interface

Load Sound

The “Load Sound” button allows you to load a wav file into the program. Currently only 16 and 32 bit wav files with one or two channels are supported. Note that the entire wav file is loaded in memory, this is fine and fast for wav files of short durations (tens of seconds), but longer sound files are going to consume huge amounts of RAM and may even halt your computer. If you need to work on long sound files, please use other software, like audacity.

Save As

The “Save as” button allows you to save PSA sound objects as wav files. Currently it is only possible to save sounds as 16 or 32 bit wav files with one or two channels. If you choose a mono (1-channel) format, and multiple PSA sound objects have been selected for saving, they will be summed together before saving. If you choose a stereo (2-channels) format, “right” and “left” PSA sound objects will be saved to their respective channels in the wav file; if multiple “right” or “left” PSA sound objects have been selected, they will be summed before saving.

Cut

The “Cut” button allows you to remove segments of a sound waveform. The starting and ending points of the segments to be cut off can be specified in seconds, milliseconds, or sample numbers. When working with sample numbers, note that PSA indexing works exactly like numpy indexing. Examples:

```

>>> import numpy as np #import numpy
>>> sig = np.arange(10) #generate a 10-elements array
>>> x[0:5] #Select the first five samples
           #indexing starts from 0
array([0, 1, 2, 3, 4])
>>> sig[7:10] #select last 3 samples
array([7, 8, 9])
>>> sig[1:3] #select the second and third sample
array([1, 2])
>> sig[1:2] #select the second sample
array([1])
>>> sig[1:1] #note that if the start and end point
              #are the same nothing is selected
array([], dtype=int64)

```