

"Level" Remake Documentation

List of Contents

1. "Level" and "R"
 2. R Setup
 - a. Installing R
 - b. Installing necessary packages
 - i. Mac
 - ii. Windows
 - iii. Linux
 - c. Installing X11 (for Mac only)
 3. Running "level.r"
 4. Using "level.r"
 5. How "level" works
-

1. "level" and "R"

The "level" program is based on another program called "Wavmod" created by Tom Carrell while at the Speech Research Lab.

The implementation of the level program relies on an open-source, statistical software called "R." This software is based on "S," another commercial statistical software package. The "level" program is simply a script for R.

2. R Setup

Before you can run the level script, there are a few **one-time** installations you need to perform. These are explained below.

2a. Installing R

Download R (it's free) from the website <http://cran.r-project.org>. There are versions for Linux, Windows and Mac.

2b. Installing necessary packages

R offers many additional functionality through add-on packages. In order for the "level" program to work, you will need to install the tuneR package. Please follow the instructions below depending on your platform.

[Mac]

1. Launch R.
2. Type `install.packages()` (or use the menu command equivalent)
3. In the Install dialog, apparently the defaults sometimes work. If they don't, choose:

- * Other Repository
- * In Other Location
- * Get List

Then select tuneR from the list and click Install Selected.

[Windows]

1. Launch R.
2. Choose Packages from the menu, and Install Packages.
3. Select one of the mirror sites, then tuneR package from the package list. Installation is automatically done.

[Linux]

1. Download the tuneR package(tuneR_0.2-1.tar.gz) from the R project website.
(one of the mirror sites: <http://cran.cnr.berkeley.edu/src/contrib/Descriptions/tuneR.html>)
2. Uncompress and untar the file with `tar xvfz tuneR_0.2-1.tar.gz` in an appropriate location.
3. Install the package with R CMD `INSTALL tuneR` from the command line. You either need to be in the directory where tuneR has been untarred, or specify the full path and filename for tuneR to do this.

2c. Installing X11 (for Mac only)

The implementation of the level program makes use of Tcl/Tk for the user interface. The Mac version of R contains Tcl/Tk binaries for the open source X Windows System, also called X11. Therefore, you need to install X11.

Mac OS 10.4 owners can install X11 by using the Tiger DVD installer disk. You may also get the .dmg file from:

<http://www.apple.com/support/downloads/x11formacosx.html>.

Note: X11 is only supported on Mac OS X versions.

3. Running "level.r"

You are now ready to run the script which implements the level program. For both Windows and Mac, a system script file has been made that will allow you to simply open it and the level program will automatically come up. All the necessary files will be in a zip file called "level.zip". Unzip it and follow the instructions below.

[Windows]

The Windows system script will be zipped with two other files. Browse to the "windows" folder and open "level.bat" to start the "level" script. **Note:** If you installed R in a folder other than the default, you will have to edit "level.bat" so that it points to the R executable in you system.

[Mac]

The Mac system script will also be zipped with one other file. Browse to the “mac” folder and double-click “launchLevel.app.” This will launch R, X11, and run the level script. **Note that this script application was developed and tested on a PowerMac machine running Mac OS X 10.4, Tiger.**



Figure 1: Double-click to start the level program

Then the user interface for the program will appear.

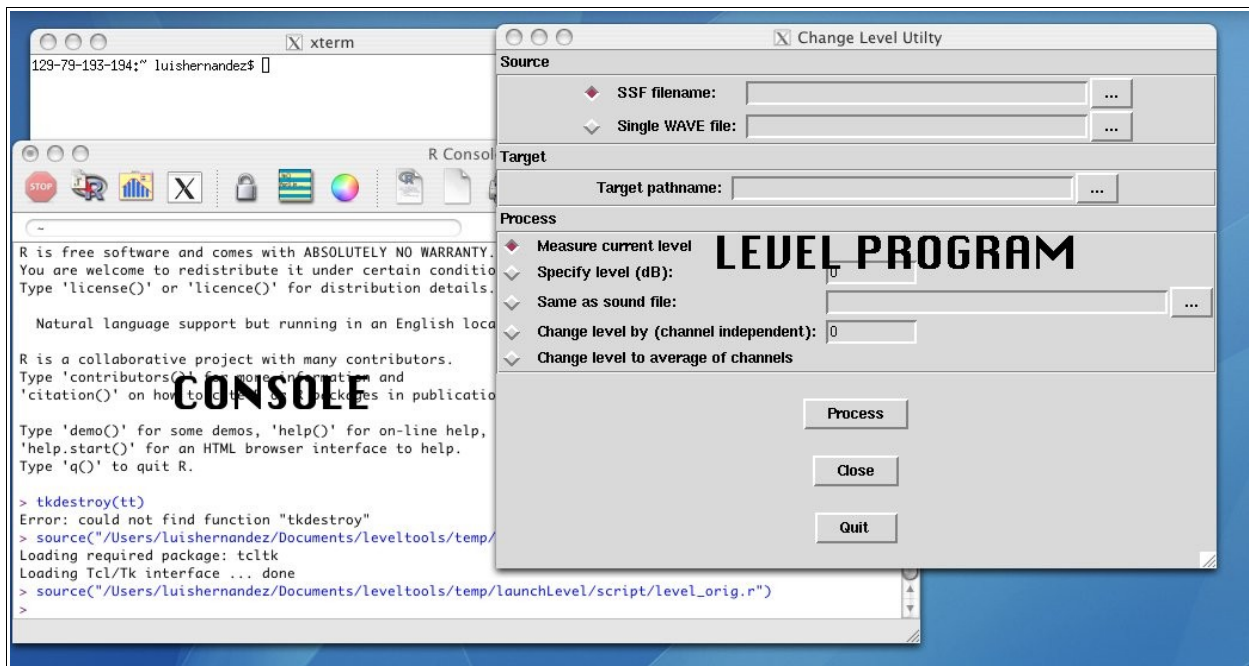


Figure 2: The user interface for the level program

4. Using "level.r"

As Figure 2 shows, the “level” program consists of three parts, the “Source,” “Target,” and “Process” section. The “Process” button will take the input given in the three sections and carry out the action wanted. Note that **all output, including status messages, will appear in the R console window.** Now we will discuss what each of the sections do.

I. The “Source” section allows you to either load a SSF file containing a list of audio files or load a single WAVE file.

II. The “Target” section allows you to choose an output folder if you plan to change the audio files in any way.

III. The central part of the program is in the “Process” section. There are five options. The first one allows you to read a profile of the WAVE files specified. The other four will change the amplitude level of the audio files and save them to the target folder you specified in the the “Target” section (it will do nothing if no target path is given). An important difference between this implementation of the level program is that **stereo files are also supported**. For stereo files, both left and right channels will be manipulated independently so that they have the same mean amplitude level.

i. Choose and type the desired value in the “Specify level (dB)” box to change the amplitude of audio file(s) to a specific dB RMS (Root Mean Square value in decibels) value.

ii. The “Same as sound file” option will calculate the dB RMS of a monaural file and change the channel(s) dB RMS of all other files to that value.

iii. The “Change level by (channel independent)” option allows you to enter a positive or negative dB RMS difference that will be applied to each channel of the audio files specified.

iv. The final option, “Change level to average of channels,” will work only for stereo and will set the dBRMS of both channels to their average dBRMS.

Figure 3 shows an example of a test run for a list of monaural files, where we change the dB RMS of all the files to 60 dB RMS.

```

=====
dB Process report for: /Users/luishernandez/Documents/testfiles.ssf , 06-28-2007
=====
File(s)      dB RMS  dB Peak  Samples  Out dB RMS
  Out dB Peak  Clipped peaks
-----
...sine48.wav 81.278  84.288  32001   60.000
 63.009      0
***Monoaural File

...sine35.wav 69.235  72.247  32001   60.000
 63.009      0
***Monoaural File

...sine32.wav 51.163  54.185  32001   60.000
 63.021      0
***Monoaural File

...sine46.wav 57.190  60.206  32001   60.001
 63.015      0
***Monoaural File

...osine1.wav 63.213  66.227  32001   60.000
 63.015      0
***Monoaural File

...sine36.wav 81.278  84.288  32001   60.000
 63.009      0
***Monoaural File

...osine4.wav 75.257  78.268  32001   60.000
 63.009      0
***DONE ****
>

```

Figure 3: Example output in console from processing a list of audio files.

4. How "level" works

The level program's main function is to modify the average amplitude of a sound file. When given an audio file, the level program figures out its Root Mean Square (RMS), which is then converted to dB. The dB conversion takes the log base 10 of the RMS and then multiplies it by 20. Not only does the dB representation scale the range of RMS values to within -66 to 90.309 dB, but it's also a good model of how the ear perceives changes in volume. Nevertheless, all amplitude-modifying computations are done using the RMS.

The RMS is defined as following for a list of n values $\{x_1, x_2, \dots, x_n\}$:

$$x_{\text{rms}} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}}$$

The RMS is the appropriate measure of the average for a signed representation of an audio signal because **the sign** on individual signal values represents in which direction the air is being displaced and is not at all an indication of the loudness of the sound. By squaring each sample, we are ignoring the sign.

The way we perform the amplitude modification of an audio file is by multiplying each sample of the audio file by a specific factor. This factor is the ratio of our wanted RMS to the RMS of the audio file (you can derive the proof of this statement using the RMS formula and some Algebra). For example, say we have a signed 16-bit (a range of [-32767, 32767]) monaural audio file with an RMS of 10,000 and we want it to change it to an RMS of 8,000. The factor we would use to multiply each sample of the original file would be:

$$8,000/10,000 = 0.8$$