

## Introduction

This project archives the results for the following publication:

Inducing musical-interval learning by combining task practice with periods of stimulus exposure alone

## Abstract

A key component of musical expertise is the ability to discriminate between and identify musical intervals, such as a perfect 4th and a major 3rd. This ability is necessary for playing by ear, performance self-correction, and the analysis of music. However, even musicians can struggle with ‘ear training’ and mastery is typically associated with years of musical experience. Here we report a training regimen that yields learning on musical-interval discrimination and generalization to musical-interval identification across four training sessions. Critically, the regimen combines periods of practice on musical-interval discrimination and periods of stimulus exposure without practice within each session. Continuous practice yields no improvement at all. The Practice+Exposure regimen generates improvement in both musicians and non-musicians, providing a practical means for the rapid acquisition of an aural skill in music that has long been viewed to take many years to learn.

## Setup

To setup the repository you will need to install Anaconda Python 2.7. Once you’re finished run the following commands on the command line, replacing [project directory] with the location of the folder containing this project.

```
cd [project directory]
conda env create -f environment.yml
source activate apmusic_2017_12_11
```

In windows these command can be run on the command line that is installed with anaconda. You will also need to replace the last line as follows:

```
activate apmusic_2017_12_11
```

This will probably take some time to run on your machine as various python and R software packages are installed.

## Usage

Once installed, to run the analyses or re-create a figure you will need to have the open conda environment ready. If you haven't closed the window you opened during "Setup", you can use that, otherwise, re-open the command line and run the following

```
cd [project directory]
source activate apmusic_2017_12_11
```

Again, in Windows these command can be run on the command line that is installed with anaconda and you will need to replace the last line with `activate apmusic_2017_12_11` (no `source`).

Each figure panel from the paper generally has two corresponding files. They are either python or R source files, and they re-create the figure pdf and run the corresponding analyses, respectively. These files are located under the **plots** and **analyses** subfolders. There are some figures with auxillary analyses (e.g. figure 2A has a secondary analysis by foil instead of regimen). All analysis associated with a figure include that figure panel as a prefix of the filename, and are listed under the analyses directory.

To run one of these files, call either **Rscript** or **python** (as appropriate for the file type) from the same command line you used above.

For example, to re-run the analyses corresponding to Figure 1A you can call the following from the command line.

```
python analyses/fig1A.py
```

And to re-create Figure 1A's pdf:

```
Rscript plots/fig1A.R
```

## Recreating the preprocessed data

The scripts for each figure and analysis load some preprocessed data. All preprocessed data files are listed in **preprocessing/files.txt**. If you want to recreate one of these files you can do so by finding the script with the same name prefix as the file.

For example, the preprocessed data file **preprocessing/data/discrim\_2017-12-31.csv** can be recreated (with a new date suffix) by calling

```
Rscript preprocessing/discrim.R
```

All preprocessed files are stored under the **preprocessing/data** subfolder.

**WARNING:** some of the preprocessing steps can be time consuming to run.

To start all analyses over from scratch, simply delete all of the preprocessing data files under the `preprocessing/data` directory. You will have to rerun the preprocessing scripts in the following order

1. Aggregate discrimination data - `preprocessing/discrim.R`
2. Aggregate classification data - `preprocessing/class.R`
3. Multi-level model fitting - `preprocessing/multi_level_sample.py`
4. Multi-level model fitting with musical experience - `preprocessing/music_multi_level_sample.py`
5. Each figure's preprocessing scripts - `preprocessing/fig1A.py`, `preprocessing/fig1B.py` etc...

If you re-process any data you will have to update the file name listed in `preprocessing/files.txt` so that any scripts relying on this data will use the newly created files.

## Editing the scripts

To edit the R files you can use R Studio and to edit python you can use Spyder. Alternatively you can use a decent text editor to edit both types of files:

- Emacs
- Vim
- Sublime

## Regimen Abbreviations

The training regimens have been given abbreviations in the code: AP (Practice+Exposure), LA (All-Practice), SA (Practice+Silence), and P (Exposure+Silence).

## Comparison to Published Figures and Statistics

The resulting pdfs from the analysis were imported into Adobe Illustrator and the aesthetics modified for improved legibility and consistency. Thus the published figures will differ somewhat from those produced by these scripts: however, the actual data and reported statistics as represented in the final paper should remain *unchanged*. Small variations in model fits and the resulting statistics may occur due to the MCMC sampling if any of the model samples are re-processed.