





audiomath: A neuroscientist's sound toolkit

N. Jeremy **Hill**, Scott W.J.**Mooney**, Glen T. **Prusky** - Burke-Blythedale Pediatric Neuroscience Research Collaboration Heliyon, 2021

Apresentação: Nancy Sotero Silva









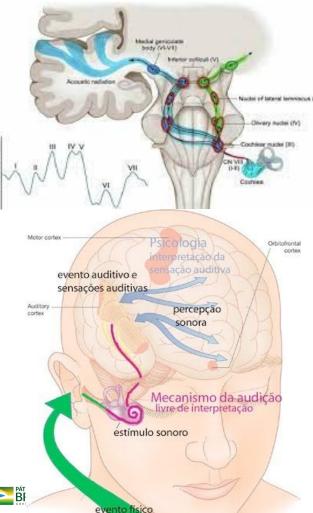


Avaliação do córtex auditivo

- Eletrofisiologia
- Event-related potentials
- Apresentação de estímulos específicos

○Parâmetros○Tipo de estímulo○Duração○Frequência○Taxa de Apresentação

- Sincronização estímulo/resposta
- Protocolo de apresentação













Estímulos auditivos

- Uso de softwares específicos para avaliação clínica
- Pesquisa: necessidade de manipulação dos parâmetros
- Dificuldade de portabilidade entre sistemas
- Sistemas de integração multimodal: desafio a mais
 - Exemplo: fNIRS + EEG
 - NISTim
 - PsychoPy











Problema

Encontrar uma ferramenta que possa gerar ou manipular estímulos auditivos, ser compatível com os *softwares* de apresentação/sincronização dos instrumentos de aquisição do sinal cerebral.

Estímulo de fala - parâmetros

Duração

ISI

Rise decay

Proporção de estímulos

EEG - P300 (paradigma de *oddball*)

fNIRS - sincronização temporal









audiomath - biblioteca open source

Heliyon 7 (2021) e06236

Contents lists available at ScienceDirect

Heliyon

iournal homepage: www.cell.com/helivon



Research article

CelPress

audiomath: A neuroscientist's sound toolkit

N. Jeremy Hill 4, Scott W.J. Mooney b,c, Glen T. Prusky b,c,d

- Stratton VA Medical Center, Albany, NY, USA
 Burke Neurological Institute, White Plains, NY, USA
- Bythedale Children's Hospital, Valhalla, NY, USA
- ⁴ Department of Physiology and Biophysics, Weill Cornell Medicine, New York, NY, USA

ARTICLE INFO

Keywords: Auditory stimuli Python Software library Audio latency Audio jitter

ABSTRACT

In neuroscientific experiments and applications, working with auditory stimuli demands software tools for generation and acquisition of raw audio, for composition and tailoring of that material into finished stimuli, for precisely timed presentation of the stimuli, and for experimental session recording. Numerous programming tools exist to approach these tasks, but their differing specializations and conventions demand extra time and effort for integration. In particular, verifying stimulus timing requires extensive engineering effort when developing new applications.

This paper has two purposes. The first is to present aud/comach (https://pppi.org/project/aud/omath), a sound software library for Python that prioritizes the needs of neuroscientists. It minimizes programming effort by providing a simple object-oriented interface that unifies functionally for audio generation, manipulation, visualization, decoding, encoding, recording, and playback. It also incorporates specialized tools for measuring and optimizing stimulus timing.

The second purpose is to relay what we have learned, during development and application of the software, about the twin challenges of delivering stimuli precisely at a certain time, and of precisely measuring the time at which stimuli were delivered. We provide a prime on these problems and the possible approaches to them. We then report audio latency measurements across a range of hardware, operating systems and settings, to illustrate the ways in which hardware and software factors interact to affect stimulus presentation performance, and the resulting pitfalls for the programmer and experimenter. In particular, we highlight the potential conflict between demands for low latency, low variability in latency ("filter"), cooperativeness, and orbustness. We report the ways in which audiomath can help to map this territory and provide a simplified path toward each application's particular priority.

By unifying audio-related functionality and providing specialized diagnostic tools, audiomath both simplifies and potentiates the development of neuroscientific applications in Python.





- Facilitar para programadores em Python atividades de sintetização, gravação, manipulação, edição, visualização e execução de ondas sonoras.
- Criação motivada pela necessidade de Ferramentas para desenvolvimento e apresentação de estímulos auditivos na pesquisa neurocientífica.
- Python escolhida pela maturidade, modularidade, potência e minimalismo da linguagem, abrangência, possibilidade de pacotes de terceiros.







4 Tarefas Principais realizadas com uso do audiomath

- 1. Gerar estímulo auditivo: extrair arquivo de audio, gravar ou sintetizar
- 2. Composição do som: manipulação dos dados, extração de características e padrões
- 3. Apresentação do estímulo para o sujeito: sincronização, manipulação de volume, latência
- 4. Capturas de som gravação, respostas verbais, outros eventos

Design x Execução

Recursos principais para Design: manipulação, geração, visualização, decodificação, codificação, gravação, e reprodução.







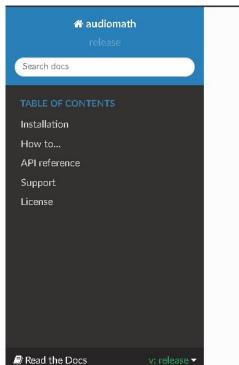




Objetivo

 Fornecer funcionalidade em certo nível certo nível, evitando que programador se preocupe com detalhes na implementação do código.

Integra outras bibliotecas



- Play sounds with more-precise latency, via PsychPortAudio
- Record a sound into memory
- · Record a sound directly to file
- Target a particular input or output device
- Manipulate the operating-system's overall volume settings
- Measure audio latency
- API reference
 - o The audiomath package
 - The sound class
 - The synth class
 - The Player class
 - The oueue class
 - The Fader class
 - The Delay class
 - o The Recorder class
 - o The ffmpeg class
 - o The sox class
 - o The audiomath.PortAudioInterface Sub-module
 - o The audiomath.PsychToolboxInterface Sub-module
 - o The audiomath.Signal sub-module
 - o The audiomath.System/olume Sub-module
- Support
- License











Programa para padronização de estímulo auditivo usando audiomath

- 1. Instalação das bibliotecas de terceiros: audiomath, librosa
- 2. **Importação das classes utilizadas:** random (Py); Sound, Player, Concatenate (audiomath); StretchAndShift (audiomath + librosa)
- 3. Definir funções Manipulação de canais
- 4. Leitura de dados Importação de arquivos de audio
- 5. Entrada de dados Definição de parâmetros (frequências, duração, ISI)
- 6. **Estrutura de decisão** verificação de canais
- 7. Funções e Classes Manipulação dos estimulos
- 8. **Estrutura de repetição** criação e manipulação de listas
- 9. **Saída de dados** print conferência de informações do arquivo de audio pré e pós manipulação; execução do estímulo final direto no *hardware*
- **10.** Escrita de dados Criação de arquivo sonoro em disco











Obrigada!

nancy.sotero@edu.isd.org.br















