Program start

ema2wav can be started either

- using the **GUI**
  - via binary (.dmg)
  - via console
- using command-line (see documentation)
- by importing as a python module (custom script/notebook/**Google colab notebook**; see documentation)

Installation (Anaconda - script version)

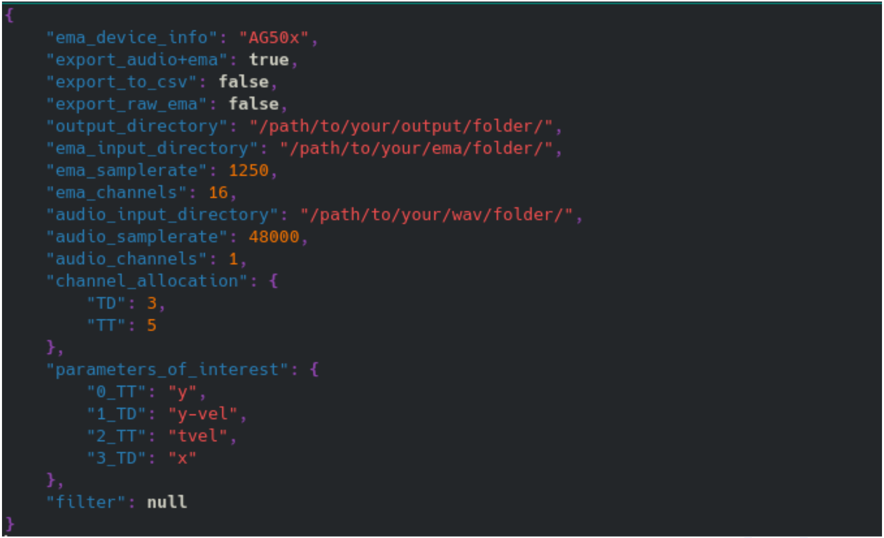
- open Terminal/Console/Anaconda Prompt
- create conda environment:
  ```
  conda create -name ema_env
  ```
- activate the environment:
  ```
  conda activate ema_env
  ```
- install the dependencies:
  ```
  pip install -r
  path/to/ema2wav/folder/src/requirements.txt
  ```

Run ema2wav

- for Mac: run ema2wav_ app.app (see GitHub documentation)
- script version (GUI, from terminal, ema2wav src directory assumed):

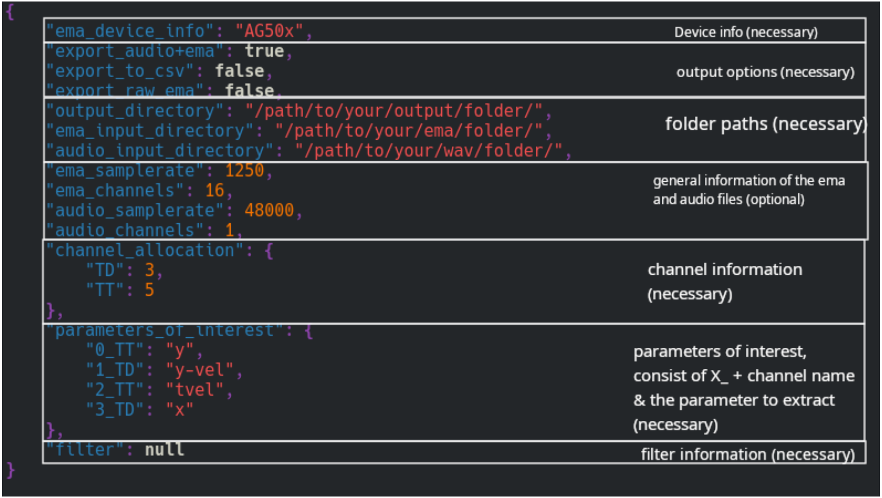
  ```
  python ema2wav_app.py
  ```

User input



```
{
    "ema_device_info": "AG50x",
    "export_audio+ema": true,
    "export_to_csv": false,
    "export_raw_ema": false,
    "output_directory": "/path/to/your/output/folder/",
    "ema_input_directory": "/path/to/your/ema/folder/",
    "ema_samplerate": 1250,
    "ema_channels": 16,
    "audio_input_directory": "/path/to/your/wav/folder/",
    "audio_samplerate": 48000,
    "audio_channels": 1,
    "channel_allocation": {
        "TD": 3,
        "TT": 5
    },
    "parameters_of_interest": {
        "0_TT": "y",
        "1_TD": "y-vel",
        "2_TT": "tvel",
        "3_TD": "x"
    },
    "filter": null
}
```

User input - parts of the input file



```
{
    "ema_device_info": "AG50x",
    "export_audio+ema": true,
    "export_to_csv": false,
    "export_raw_ema": false,
    "output_directory": "/path/to/your/output/folder/",
    "ema_input_directory": "/path/to/your/ema/folder/",
    "audio_input_directory": "/path/to/your/wav/folder/",
    "ema_samplerate": 1250,
    "ema_channels": 16,
    "audio_samplerate": 48000,
    "audio_channels": 1,
    "channel_allocation": {
        "TD": 3,
        "TT": 5
    },
    "parameters_of_interest": {
        "0_TT": "y",
        "1_TD": "y-vel",
        "2_TT": "tvel",
        "3_TD": "x"
    },
    "filter": null
}
```

Device info (necessary)

output options (necessary)

folder paths (necessary)

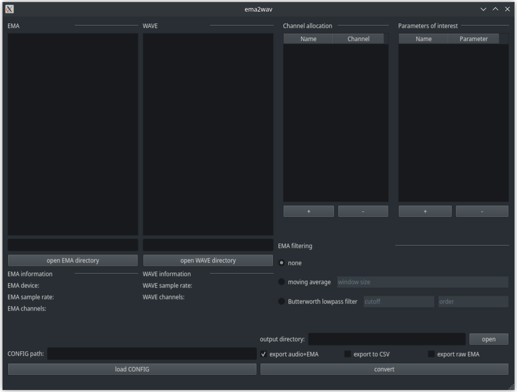
general information of the ema and audio files (optional)

channel information (necessary)

parameters of interest, consist of X_ + channel name & the parameter to extract (necessary)

filter information (necessary)

Start of the conversion process (GUI)



Start of the conversion process (manual)

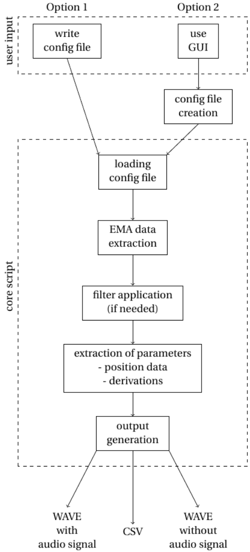
- from terminal (ema2wav src directory assumed)

  ```
  python convert.py config.json
  ```

- as Python module:

  ```
  import ema2wav_core as ec
  config_file =
  "/path/to/your/config_file.json"
  ec.ema2wav_conversion(config_file)
  ```

Conversion process



Conversion process - reshaping of the data

sample 1									sample 2					sample n
channel 1				channel 2				channel n	channel 1				channel n	⋯
x	y	z	⋯	x	y	z	⋯	⋯	x	y	z	⋯	⋯	⋯

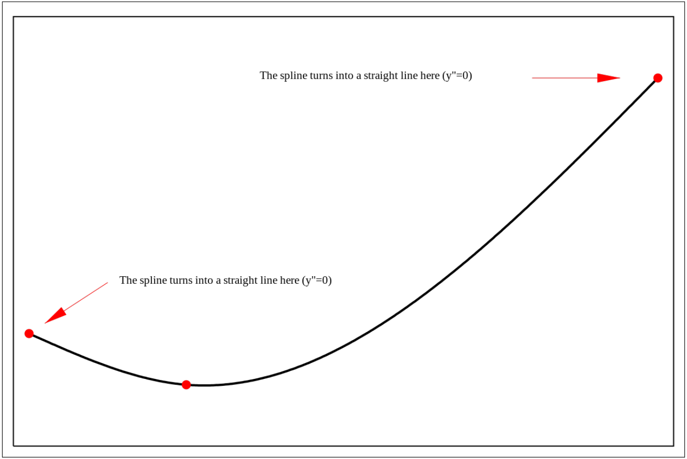
↓

sample n							
	x	y	z	phi	theta	rms	empty
channel 1	⋯	⋯	⋯	⋯	⋯	⋯	⋯
channel 2	⋯	⋯	⋯	⋯	⋯	⋯	⋯
channel n	⋯						

sample 2							
	x	y	z	phi	theta	rms	empty
channel 1	value	value	value	value	value	value	value
channel 2	value	value	value	value	value	value	value
channel n	⋯						

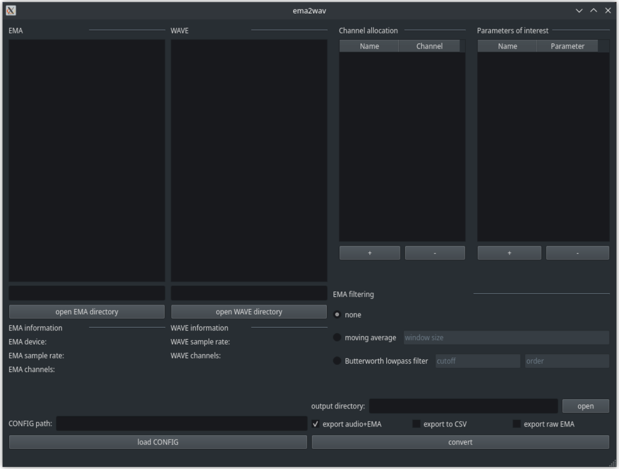
sample 1							
	x	y	z	phi	theta	rms	empty
channel 1	value	value	value	value	value	value	value
channel 2	value	value	value	value	value	value	value
channel n	⋯						

Conversion process - Interpolation



(https://en.wikipedia.org/wiki/Spline_interpolation/media/File:Cubic_splines_three_points.svg

Conversion process - GUI approach



EMA filtering

● none

○ moving average

○ Butterworth lowpass filter

☑ export audio+EMA

□ export to CSV

□ export raw EMA

Conversion process - GUI approach (2)

- open the folders containing .pos and .wav files
- channel allocation: enter name channel number
- parameters of interest: enter channel name parameters
- select filter (if necessary)
- open output folder
- start the conversion

Conversion process - GUI approach (3)

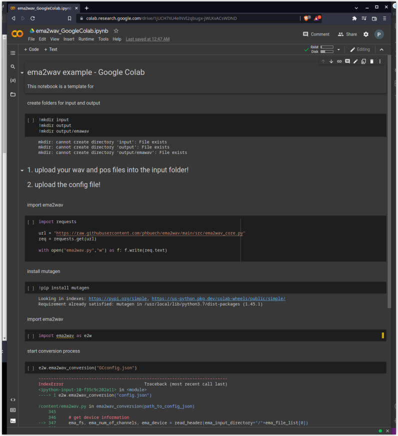
- config file as documentation
- can be used for replicating the conversion process (load CONFIG)

Conversion process - Google Colab approach

- Google Colab
- execute Jupyter Notebooks online
- free easy-to-use
- see example notebook

Conversion process - Google Colab (1)

- open the example notebook in Google Colab


ema2wav_GoogleColab.ipynb
colab.research.google.com/drive/1jUCH7XU4elNVI2qbugx-jWLKvACsWDND
ema2wav_GoogleColab.ipynb
File Edit View Insert Runtime Tools Help Last saved at 12:47 AM
Comment
Share
P
+ Code + Text
RAM
Disk
Editing

ema2wav example - Google Colab

This notebook is a template for

create folders for input and output

```
[ ] !mkdir input
    !mkdir output
    !mkdir output/emawav
```

```
mkdir: cannot create directory ‘input’: File exists
mkdir: cannot create directory ‘output’: File exists
mkdir: cannot create directory ‘output/emawav’: File exists
```

1. upload your wav and pos files into the input folder!

2. upload the config file!

import ema2wav

```
[ ] import requests

    url = "https://raw.githubusercontent.com/phbuech/ema2wav/main/src/ema2wav_core.py"
    req = requests.get(url)

    with open("ema2wav.py","w") as f: f.write(req.text)
```

install mutagen

```
[ ] !pip install mutagen
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: mutagen in /usr/local/lib/python3.7/dist-packages (1.45.1)
```

import ema2wav

```
[ ] import ema2wav as e2w
```

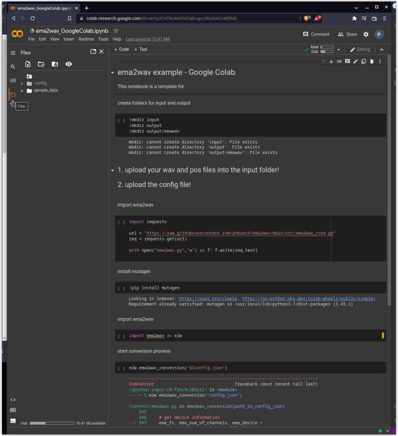
start conversion process

```
[ ] e2w.ema2wav_conversion("GCconfig.json")
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-10-f35c9c202a11> in <module>
----> 1 e2w.ema2wav_conversion("config.json")

/content/ema2wav.py in ema2wav_conversion(path_to_config_json)
    345
    346     # get device information
--> 347     ema_fs, ema_num_of_channels, ema_device = read_header(ema_input_directory+"/"+ema_file_list[0])
```

Conversion process - Google Colab (2)

- open the files folder



ema2wav example - Google Colab

This notebook is a template for

create folders for input and output

```
[ ] !mkdir input
    !mkdir output
    !mkdir output/emawav
```

```
mkdir: cannot create directory 'input': File exists
mkdir: cannot create directory 'output': File exists
mkdir: cannot create directory 'output/emawav': File exists
```

1. upload your wav and pos files into the input folder!

2. upload the config file!

import ema2wav

```
[ ] import requests

    url = "https://raw.githubusercontent.com/phbuech/ema2wav/main/src/ema2wav_core.py"
    req = requests.get(url)

    with open("ema2wav.py","w") as f: f.write(req.text)
```

install mutagen

```
[ ] !pip install mutagen
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: mutagen in /usr/local/lib/python3.7/dist-packages (1.45.1)
```

import ema2wav

```
[ ] import ema2wav as e2w
```

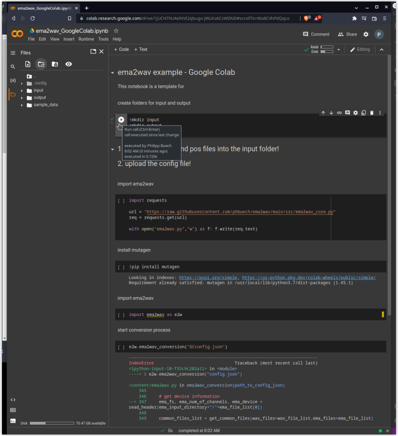
start conversion process

```
[ ] e2w.ema2wav_conversion("GCconfig.json")
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-10-f35c9c202a11> in <module>
----> 1 e2w.ema2wav_conversion("config.json")

/content/ema2wav.py in ema2wav_conversion(path_to_config_json)
    345
    346     # get device information
--> 347     ema_fs, ema_num_of_channels, ema_device =
```

Conversion process - Google Colab (3)

- run the first code snippet


ema2wav_GoogleColab.ipynb
colab.research.google.com/drive/1jUCH7XU4eINVI2qbugx-jWLKvACsWDND#scrollTo=Wa8CVhPdQqLo
ema2wav_GoogleColab.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Comment
Share
P
Files
.config
input
output
sample_data
+ Code
+ Text
RAM
Disk
Editing
Disk
70.47 GB available

ema2wav example - Google Colab

This notebook is a template for

create folders for input and output

```
!mkdir input
!mkdir output
```

Run cell (Ctrl+Enter)
cell executed since last change

executed by Philipp Buech
8:02 AM (0 minutes ago)
executed in 0.729s

1 nd pos files into the input folder!

2. upload the config file!

import ema2wav

```
[ ] import requests

    url = "https://raw.githubusercontent.com/phbuech/ema2wav/main/src/ema2wav_core.py"
    req = requests.get(url)

    with open("ema2wav.py","w") as f: f.write(req.text)
```

install mutagen

```
[ ] !pip install mutagen

    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Requirement already satisfied: mutagen in /usr/local/lib/python3.7/dist-packages (1.45.1)
```

import ema2wav

```
[ ] import ema2wav as e2w
```

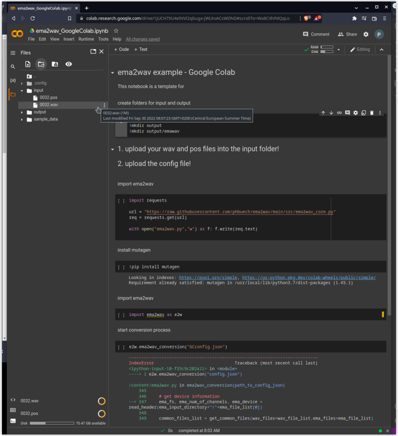
start conversion process

```
[ ] e2w.ema2wav_conversion("GCconfig.json")

    ---------------------------------------------------------------------------
    IndexError                                Traceback (most recent call last)
    <ipython-input-10-f35c9c202a11> in <module>
    ----> 1 e2w.ema2wav_conversion("config.json")

    /content/ema2wav.py in ema2wav_conversion(path_to_config_json)
        345
        346     # get device information
    --> 347     ema_fs, ema_num_of_channels, ema_device =
    read_header(ema_input_directory+"/"+ema_file_list[0])
        348
        349     common_files_list = get_common_files(wav_files=wav_file_list,ema_files=ema_file_list)
```

✓ 0s completed at 8:02 AM

Conversion process - Google Colab (4)

- upload your .pos and .wav files


ema2wav_GoogleColab.ipynb
colab.research.google.com/drive/1jUCH7XU4eINVI2qbugx-jWLKvACsWDND#scrollTo=Wa8CVhPdQqLo
ema2wav_GoogleColab.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
..
.config
input
0032.pos
0032.wav
output
sample_data
0032.wav
0032.pos
Disk
70.47 GB available

0032.wav (1M)
Last modified Fri Sep 30 2022 08:07:23 GMT+0200 (Central European Summer Time)

ema2wav example - Google Colab

This notebook is a template for

create folders for input and output

```
!mkdir output
!mkdir output/emawav
```

1. upload your wav and pos files into the input folder!

2. upload the config file!

import ema2wav

```
[ ] import requests

    url = "https://raw.githubusercontent.com/phbuech/ema2wav/main/src/ema2wav_core.py"
    req = requests.get(url)

    with open("ema2wav.py","w") as f: f.write(req.text)
```

install mutagen

```
[ ] !pip install mutagen

    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Requirement already satisfied: mutagen in /usr/local/lib/python3.7/dist-packages (1.45.1)
```

import ema2wav

```
[ ] import ema2wav as e2w
```

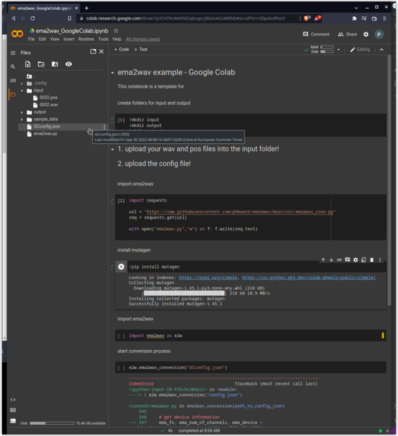
start conversion process

```
[ ] e2w.ema2wav_conversion("GCconfig.json")

    ---------------------------------------------------------------------------
    IndexError                                Traceback (most recent call last)
    <ipython-input-10-f35c9c202a11> in <module>
    ----> 1 e2w.ema2wav_conversion("config.json")

    /content/ema2wav.py in ema2wav_conversion(path_to_config_json)
        345
        346     # get device information
    --> 347     ema_fs, ema_num_of_channels, ema_device =
    read_header(ema_input_directory+"/"+ema_file_list[0])
        348
        349     common_files_list = get_common_files(wav_files=wav_file_list,ema_files=ema_file_list)
```

✓ 0s completed at 8:02 AM

Conversion process - Google Colab (5)

- upload the GC config file



GCconfig.json (393)
Last modified Fri Sep 30 2022 08:08:16 GMT+0200 (Central European Summer Time)

ema2wav example - Google Colab

This notebook is a template for

create folders for input and output

```
[1]  !mkdir input
     !mkdir output
```

1. upload your wav and pos files into the input folder!

2. upload the config file!

import ema2wav

```
[2]  import requests

     url = "https://raw.githubusercontent.com/phbuech/ema2wav/main/src/ema2wav_core.py"
     req = requests.get(url)

     with open("ema2wav.py","w") as f: f.write(req.text)
```

install mutagen

```
!pip install mutagen

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting mutagen
  Downloading mutagen-1.45.1-py3-none-any.whl (218 kB)
     |████████████████████████████████| 218 kB 10.9 MB/s
Installing collected packages: mutagen
Successfully installed mutagen-1.45.1
```

import ema2wav

```
[ ]  import ema2wav as e2w
```

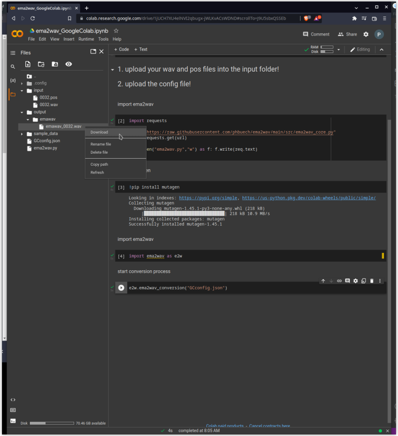
start conversion process

```
[ ]  e2w.ema2wav_conversion("GCconfig.json")

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-10-f35c9c202a11> in <module>
----> 1 e2w.ema2wav_conversion("config.json")

/content/ema2wav.py in ema2wav_conversion(path_to_config_json)
    345
    346     # get device information
--> 347     ema_fs, ema_num_of_channels, ema_device =
```

4s completed at 8:04 AM

Conversion process - Google Colab (6)

- run the other code snippets and download converted files



1. upload your wav and pos files into the input folder!

2. upload the config file!

import ema2wav

```
[2] import requests
    https://raw.githubusercontent.com/phbuech/ema2wav/main/src/ema2wav_core.py"
    equests.get(url)

    en("ema2wav.py","w") as f: f.write(req.text)
```

en

```
[3] !pip install mutagen

    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Collecting mutagen
      Downloading mutagen-1.45.1-py3-none-any.whl (218 kB)
         |████████████████████████████████| 218 kB 10.9 MB/s
    Installing collected packages: mutagen
    Successfully installed mutagen-1.45.1
```

import ema2wav

```
[4] import ema2wav as e2w
```

start conversion process

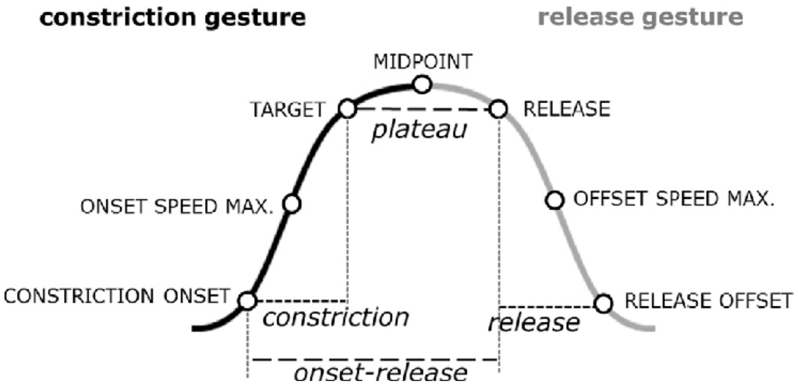
```
e2w.ema2wav_conversion("GCconfig.json")
```

Annotation & Measurements in Praat

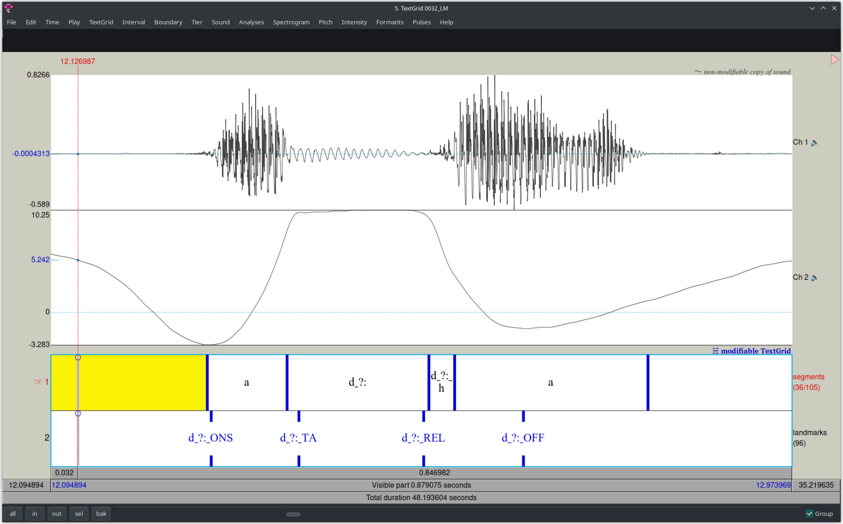
Praat tweaks

- Display default settings in Praat are not suitable for annotating EMA data
- Options for the best annotation experience:
  - disable the spectrogram
  - Change sound scaling in the editor window: (Sound > Sound scaling... > select ′by window and channel′)
  - Mute channels (if necessary): (Sound > Mute channels... > (ranges) > enter 2:X)

Landmark annotations



Gestural landmarks and intragestural intervals for a typical gestural complex (Tilsen, 2014)


5. TextGrid 0032_LM
File
Edit
Time
Play
TextGrid
Interval
Boundary
Tier
Sound
Analyses
Spectrogram
Pitch
Intensity
Formants
Pulses
Help
12.126987
~ non-modifiable copy of sound
0.8266
-0.0004313
-0.589
Ch 1
10.25
5.242
0
-3.283
Ch 2
☵ modifiable TextGrid
1
a
d_?:
d_?:_
h
a
segments
(36/105)
2
d_?:_ONS
d_?:_TA
d_?:_REL
d_?:_OFF
landmarks
(96)
0.032
0.846982
12.094894
12.094894
Visible part 0.879075 seconds
12.973969
35.219635
Total duration 48.193604 seconds
all
in
out
sel
bak
☑ Group

- Annotations of gestural landmarks using Point Tiers in Praat
- retrieve timing information as usual
- measure amplitude information by calling ′Get value at time′ applied to the Sound object
- see **measurement_example.praat**

Remarks

- never use the entire file for frequency/intensity/spectral measurements! Instead: extract the the audio track and use that for acoustic measurements
- make use of a good documentation of your files