News about VocalTractLab

Institute of Acoustics and Speech Communication,
Chair of Speech Technology and Cognitive Systems,
Technische Universität Dresden

25.11.2021

Motivation

VocalTractLab Python

- Installation: pip install VocalTractLab

- Safe interaction with the VTL backend

- Multi core support for all functions that need it

- No GUI, but improved visualizations for all objects
  - ➢ Publication-ready plots in a single line

- Advanced functionalities
  - ➢ Exports audio at any sample rates
  - ➢ Variable normalization
  - ➢ Target estimation
  - ➢ G2P, text-to-speech capabilities

Motivation

VocalTractLab Python

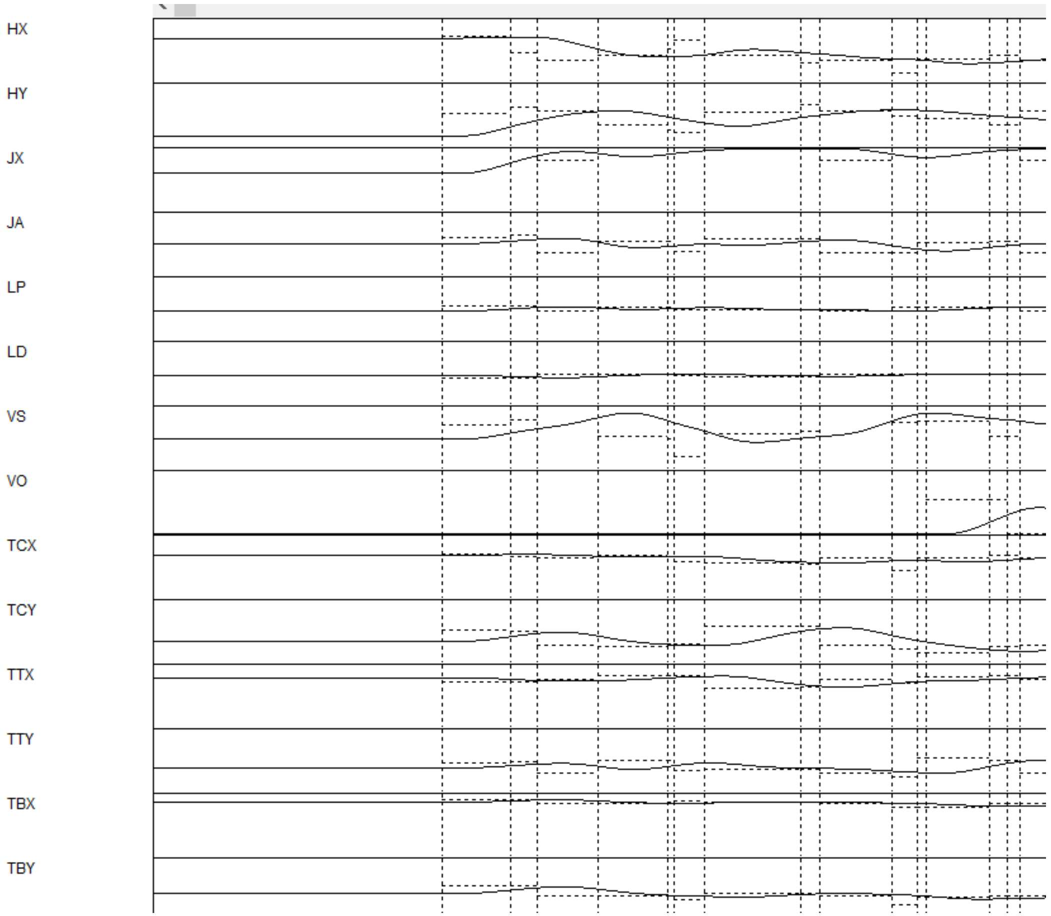
- Installation: pip install VocalTractLab

- Safe interaction with the VTL backend

- Multi core support for all functions that need it

- No GUI, but improved visualizations for all objects
  - Publication-ready plots in a single line

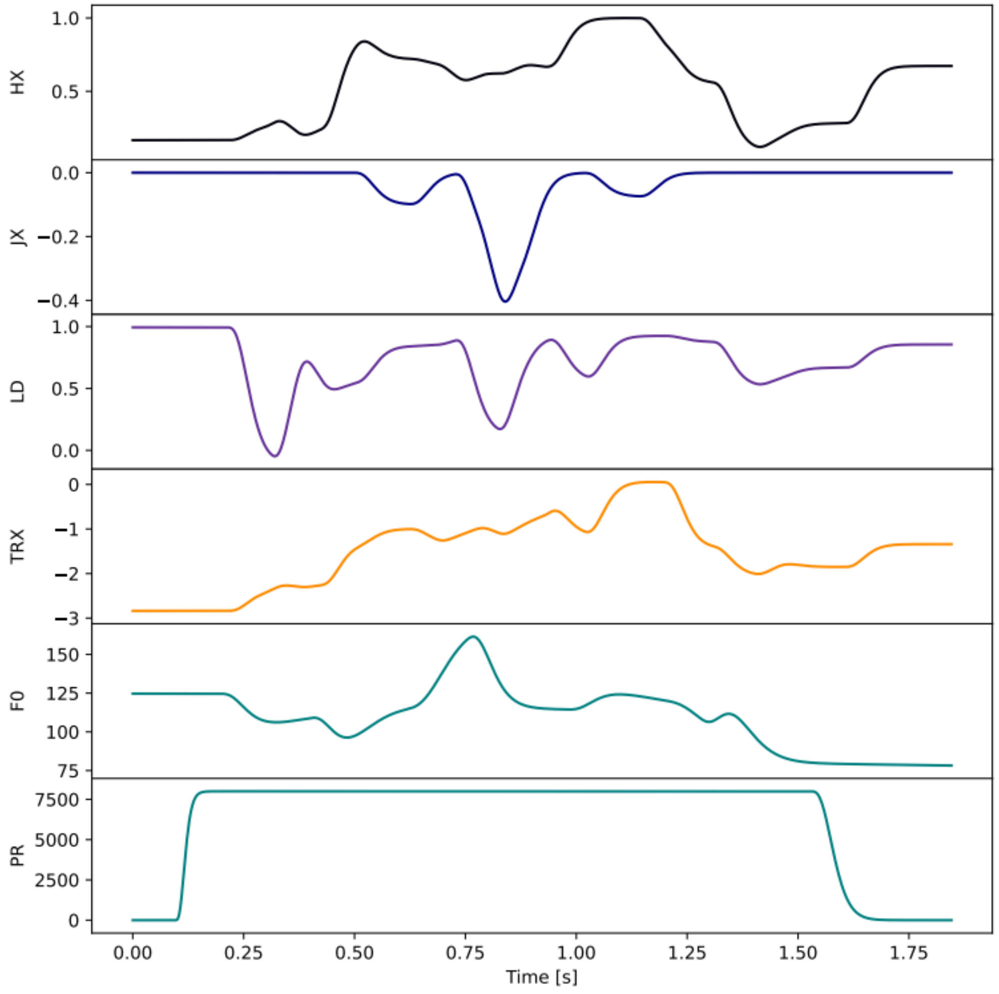
- Advanced functionalities
  - Exports audio at any sample rates
  - Variable normalization
  - Target estimation
  - G2P, text-to-speech capabilities



Motivation

VocalTractLab Python

- Installation: pip install VocalTractLab
- Safe interaction with the VTL backend
- Multi core support for all functions that need it
- No GUI, but improved visualizations for all objects
  - Publication-ready plots in a single line
- Advanced functionalities
  - Exports audio at any sample rates
  - Variable normalization
  - Target estimation
  - G2P, text-to-speech capabilities

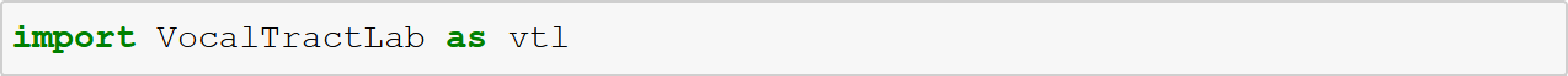


Outline

- **Typical VTL objects**
  - Segment sequence
  - Tube state
  - Transfer function
  - Sub-, supra-glottal and tract sequences
  - Targets and target scores

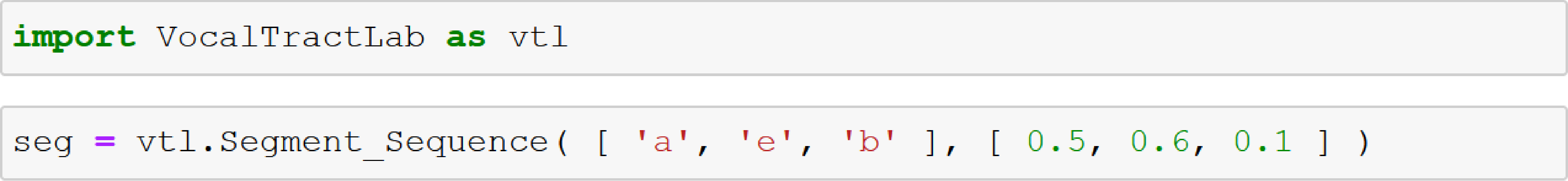
  - Gestural score
- **Advanced functionalities**
  - Target estimation
  - Text-to-speech
- **Useful general functions**

Segment Sequence



```python
import VocalTractLab as vtl
```

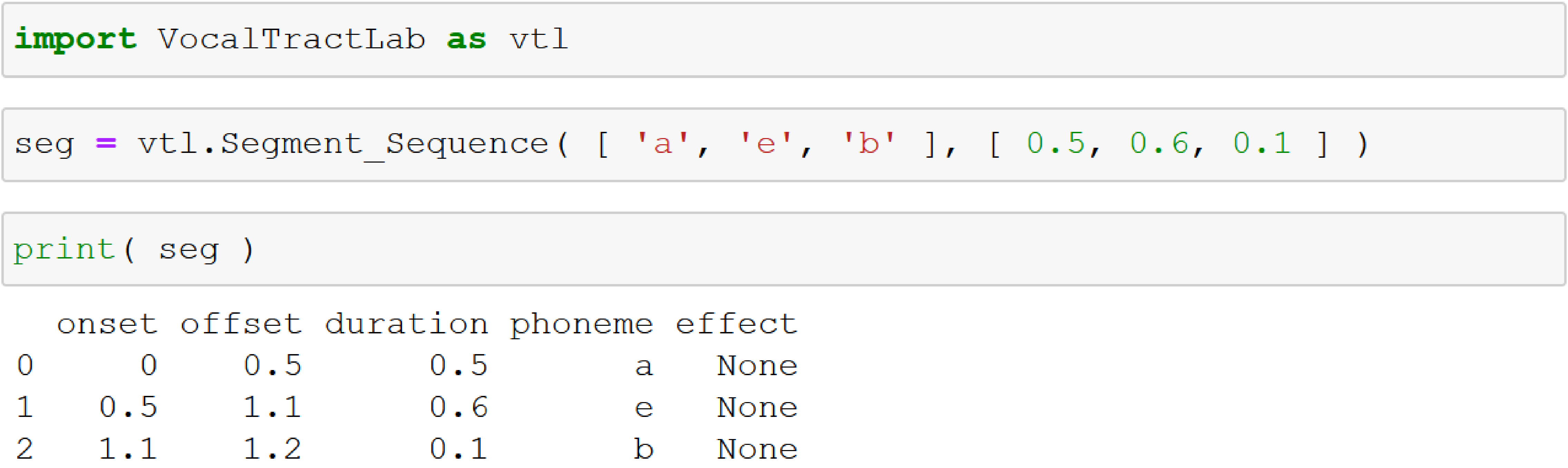
Segment Sequence



```
import VocalTractLab as vtl
```

```
seg = vtl.Segment_Sequence( [ 'a', 'e', 'b' ], [ 0.5, 0.6, 0.1 ] )
```

Segment Sequence



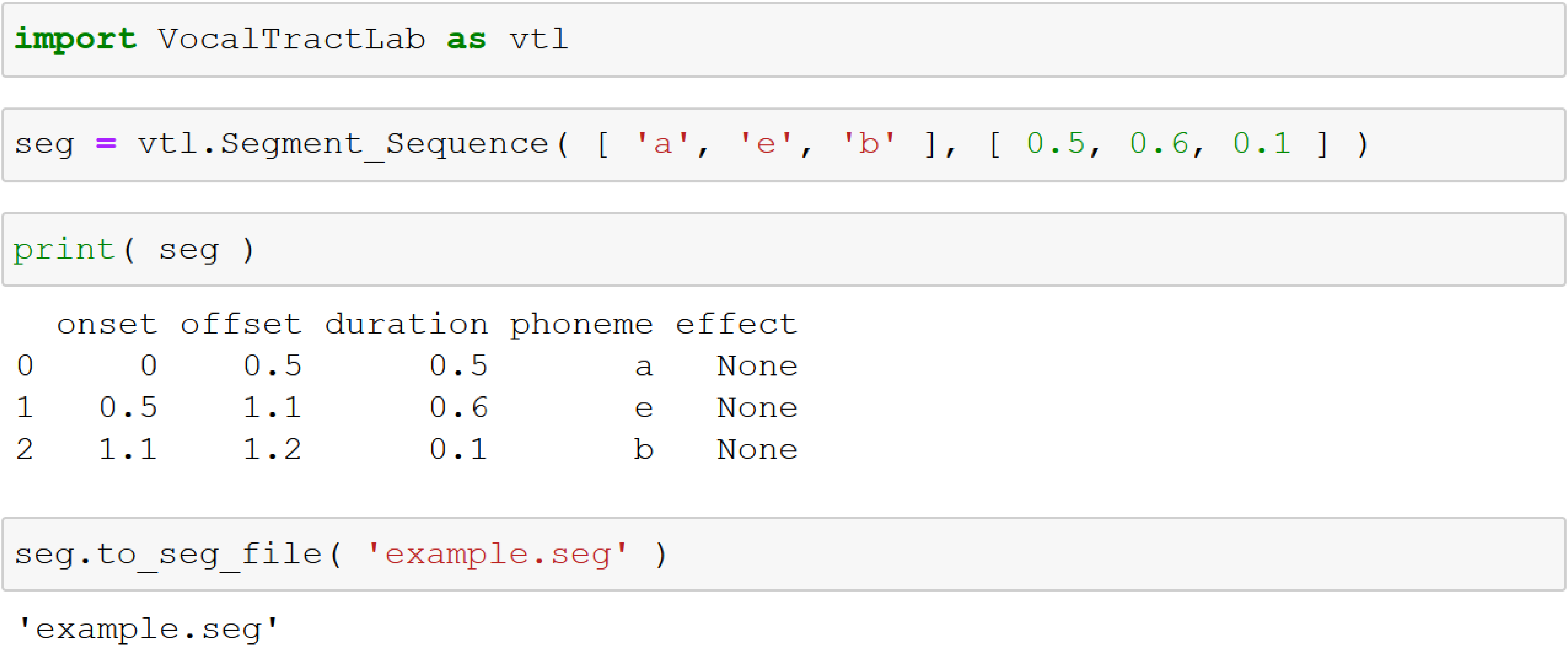
```python
import VocalTractLab as vtl
```

```python
seg = vtl.Segment_Sequence( [ 'a', 'e', 'b' ], [ 0.5, 0.6, 0.1 ] )
```

```python
print( seg )
```

```
  onset offset duration phoneme effect
0     0    0.5      0.5       a   None
1   0.5    1.1      0.6       e   None
2   1.1    1.2      0.1       b   None
```

Segment Sequence



```
import VocalTractLab as vtl
```

```
seg = vtl.Segment_Sequence( [ 'a', 'e', 'b' ], [ 0.5, 0.6, 0.1 ] )
```

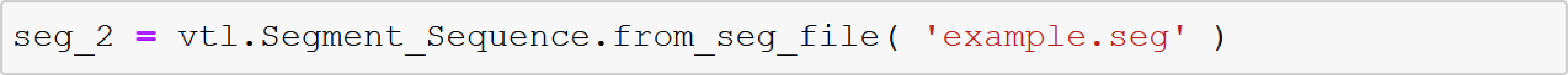
```
print( seg )
```

```
  onset offset duration phoneme effect
0     0    0.5      0.5       a   None
1   0.5    1.1      0.6       e   None
2   1.1    1.2      0.1       b   None
```

```
seg.to_seg_file( 'example.seg' )
```

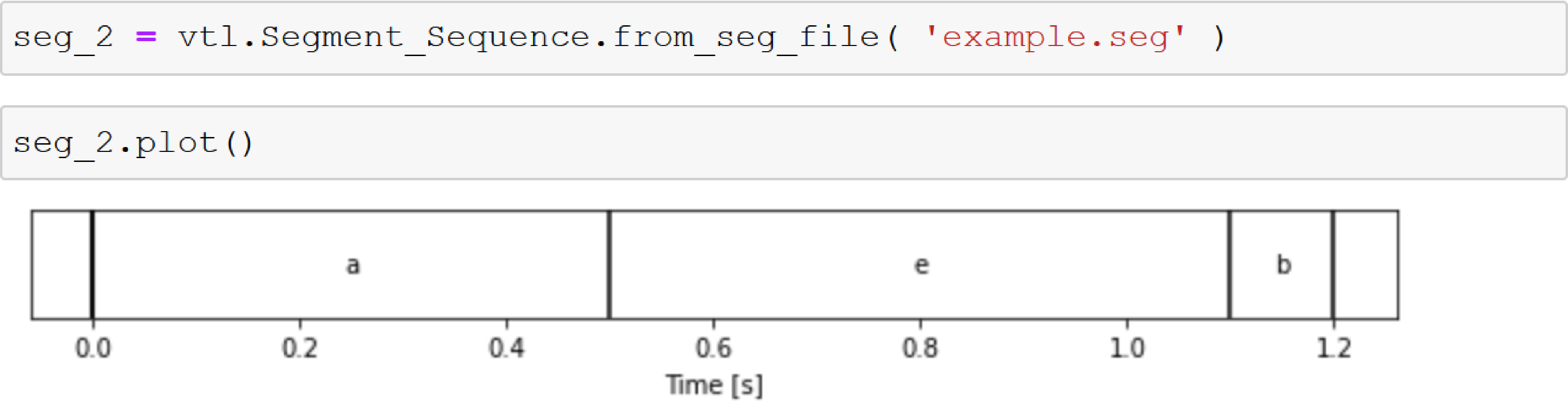
```
'example.seg'
```

Segment Sequence



```
seg_2 = vtl.Segment_Sequence.from_seg_file( 'example.seg' )
```

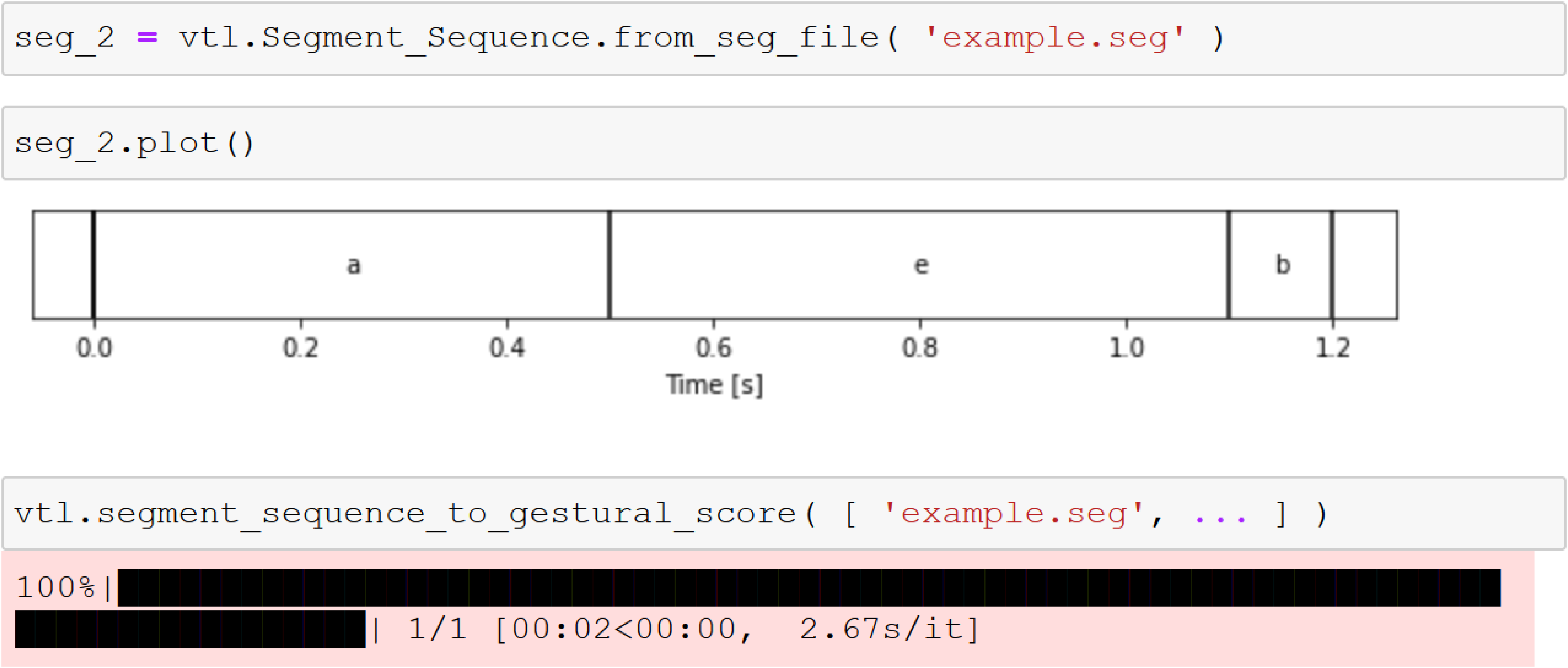
Segment Sequence



```
seg_2 = vtl.Segment_Sequence.from_seg_file( 'example.seg' )
```

```
seg_2.plot()
```

Segment Sequence



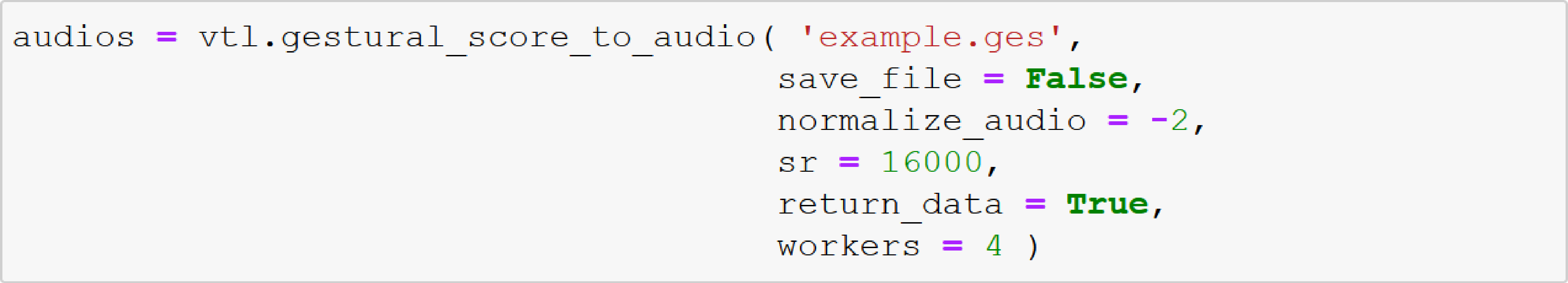
```
seg_2 = vtl.Segment_Sequence.from_seg_file( 'example.seg' )
```

```
seg_2.plot()
```

```
vtl.segment_sequence_to_gestural_score( [ 'example.seg', ... ] )
```

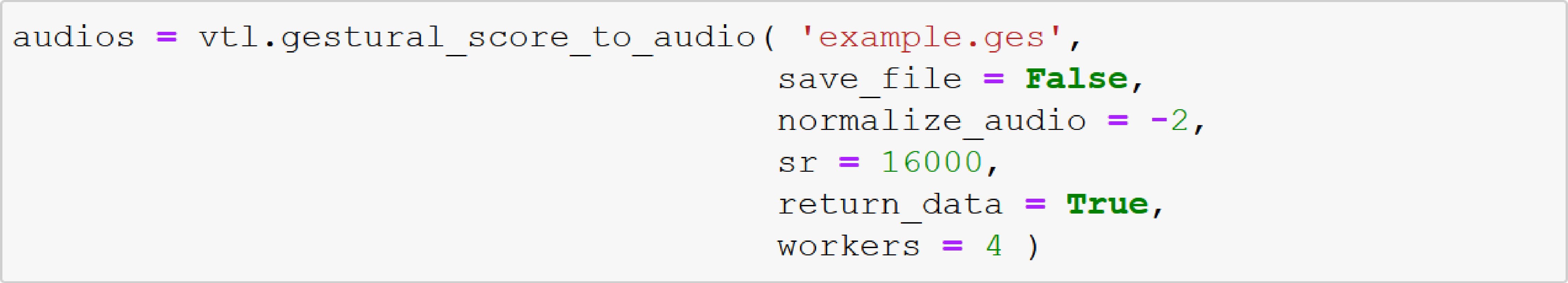
```
100%|██████████████████████████████████████████████████████████████████
█████████████████| 1/1 [00:02<00:00,  2.67s/it]
```

Gestural Scores

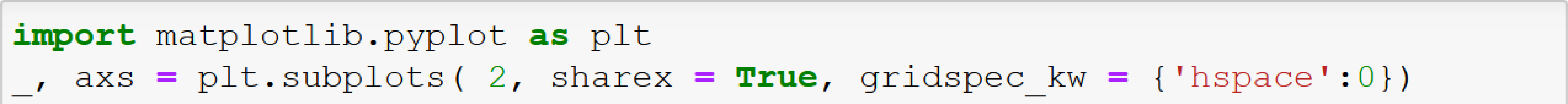


```
audios = vtl.gestural_score_to_audio( 'example.ges',
                                     save_file = False,
                                     normalize_audio = -2,
                                     sr = 16000,
                                     return_data = True,
                                     workers = 4 )
```

Gestural Scores

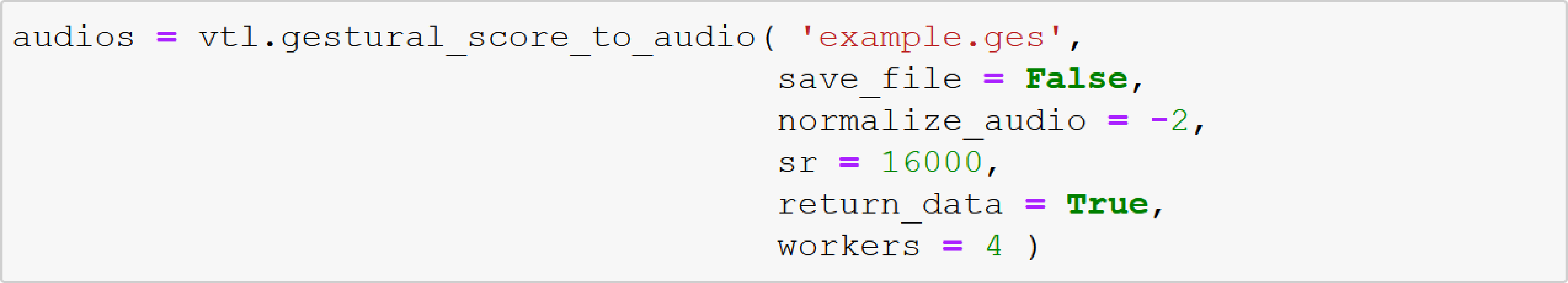


```
audios = vtl.gestural_score_to_audio( 'example.ges',
                                     save_file = False,
                                     normalize_audio = -2,
                                     sr = 16000,
                                     return_data = True,
                                     workers = 4 )
```

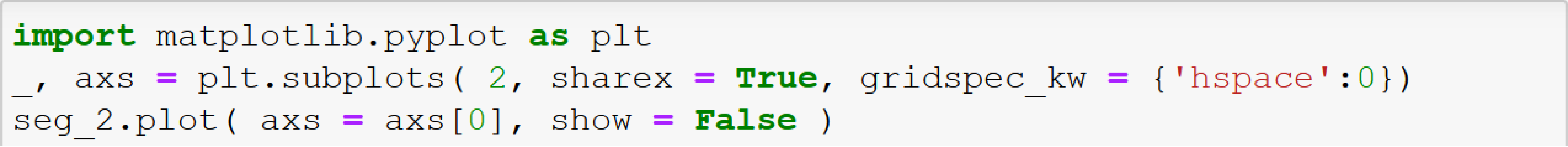


```
import matplotlib.pyplot as plt
_, axs = plt.subplots( 2, sharex = True, gridspec_kw = {'hspace':0})
```

Gestural Scores

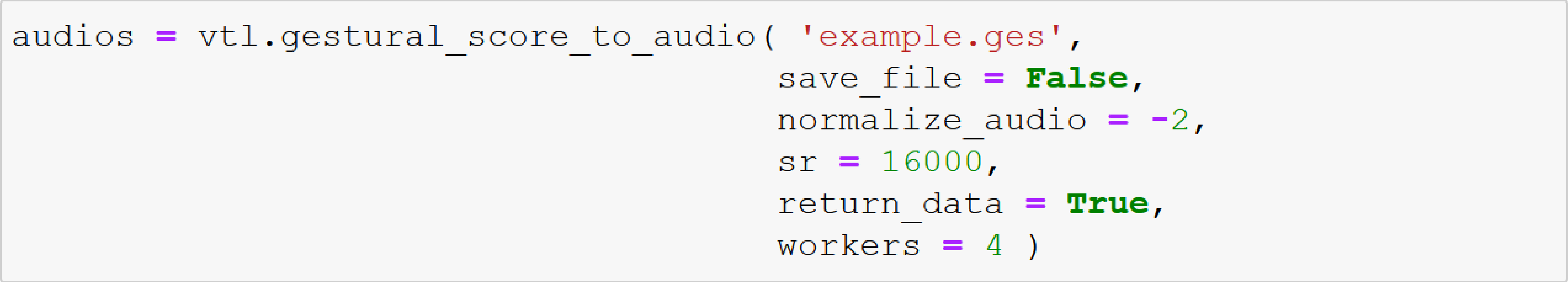


```
audios = vtl.gestural_score_to_audio( 'example.ges',
                                     save_file = False,
                                     normalize_audio = -2,
                                     sr = 16000,
                                     return_data = True,
                                     workers = 4 )
```

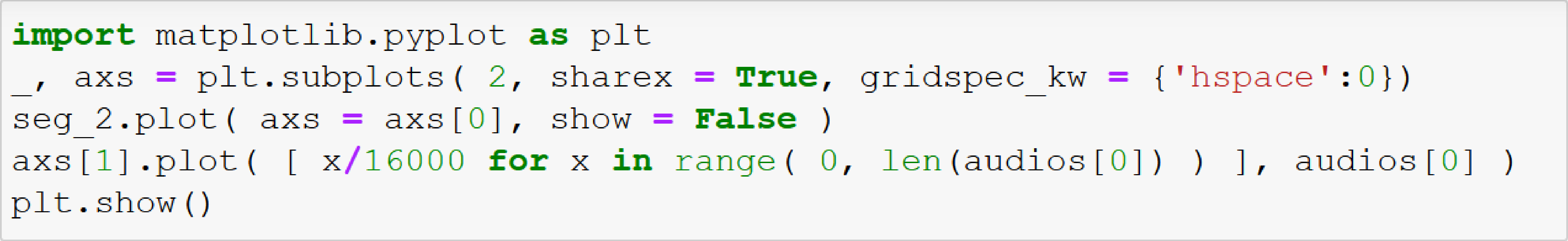


```
import matplotlib.pyplot as plt
_, axs = plt.subplots( 2, sharex = True, gridspec_kw = {'hspace':0})
seg_2.plot( axs = axs[0], show = False )
```

Gestural Scores



```
audios = vtl.gestural_score_to_audio( 'example.ges',
                                      save_file = False,
                                      normalize_audio = -2,
                                      sr = 16000,
                                      return_data = True,
                                      workers = 4 )
```

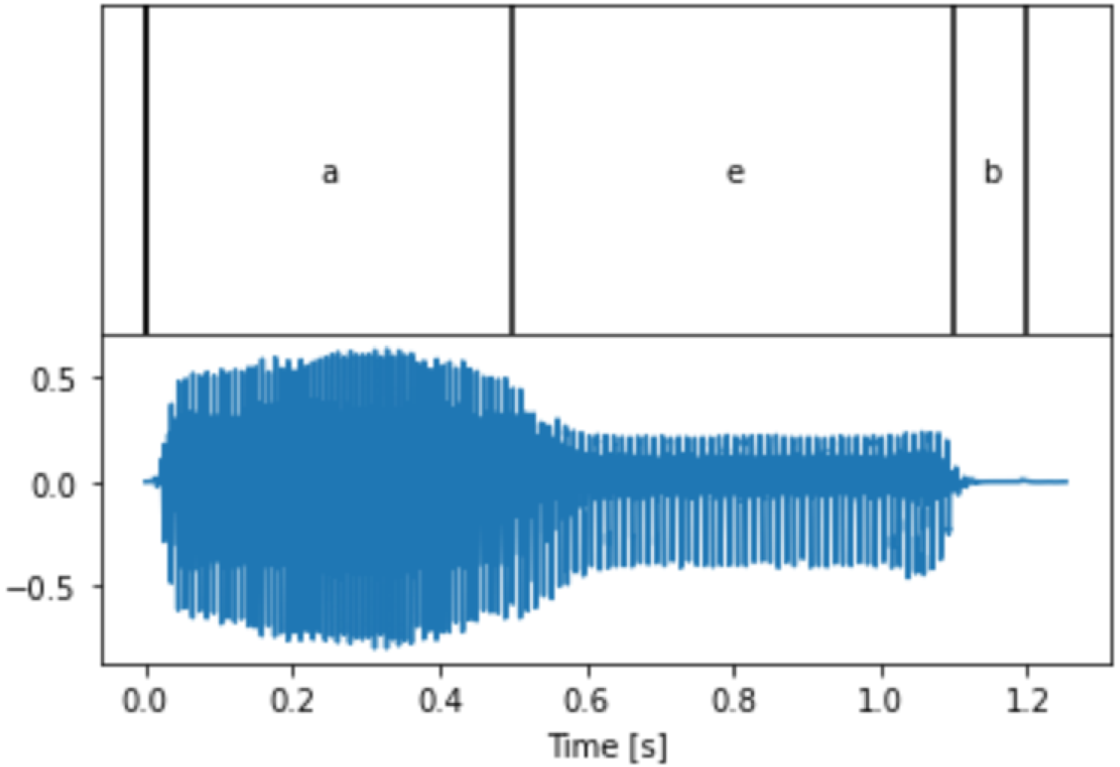


```
import matplotlib.pyplot as plt
_, axs = plt.subplots( 2, sharex = True, gridspec_kw = {'hspace':0})
seg_2.plot( axs = axs[0], show = False )
axs[1].plot( [ x/16000 for x in range( 0, len(audios[0]) ) ], audios[0] )
plt.show()
```

Custom Plots

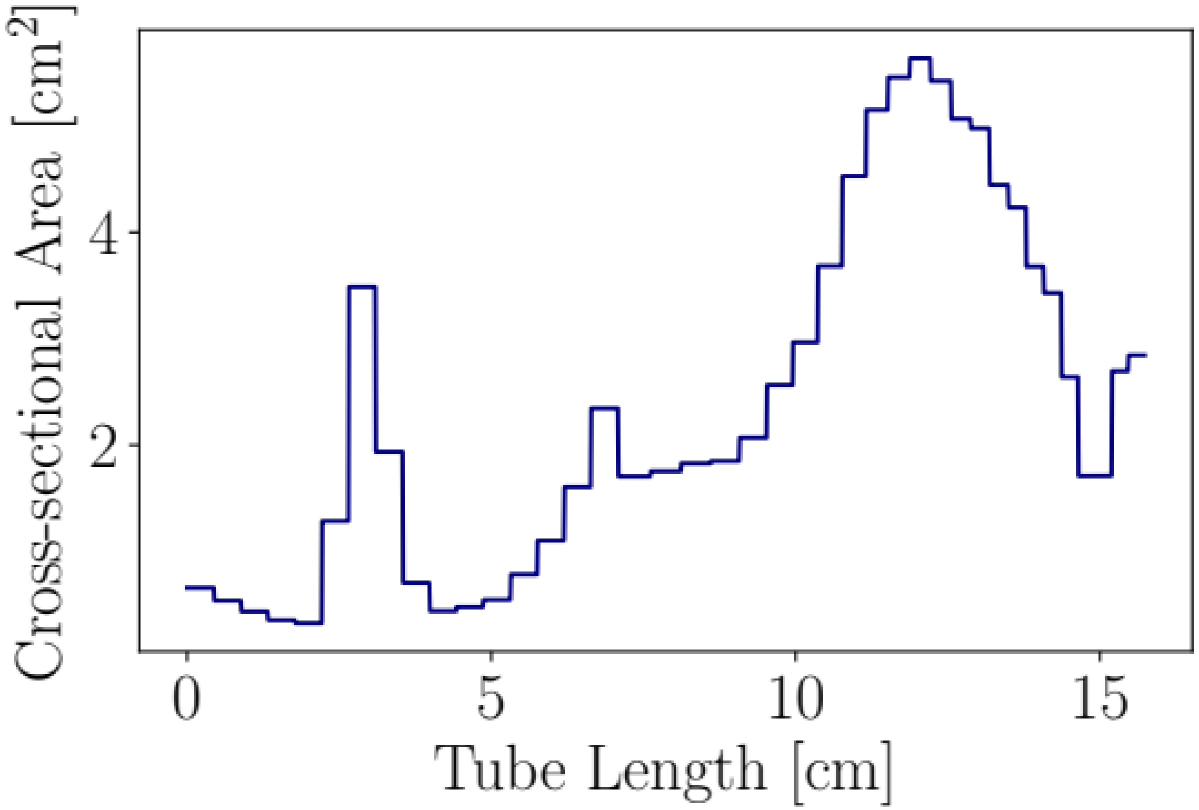
- This is possible with every VTL object

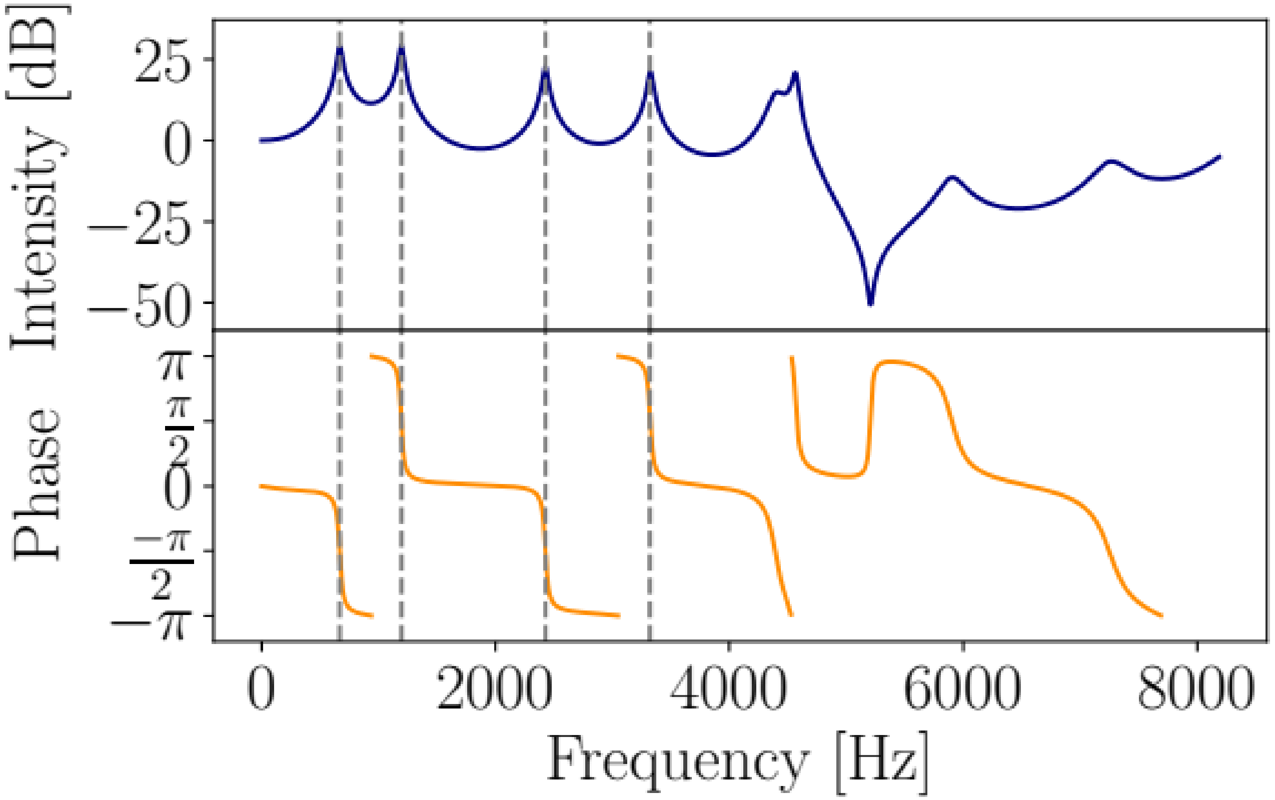
➢ Makes highly custom plots easy



Custom Plots

- This is possible with every VTL object





Tract Sequences

- **In the GUI:**

- File that contains
  - 19 supra-glottal parameters (per state)
  - 11* sub-glottal parameters (per state)

- Only way to drive synthesis at the motor level

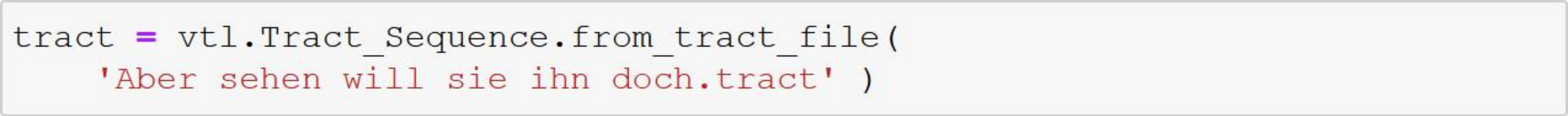
- **VTL Python:**

- Object
  - Supra-glottal sequence
  - Sub-glottal sequence

- Can be obtained from
  - Tract sequence files
  - Automatically from shapes
  - Custom from any data

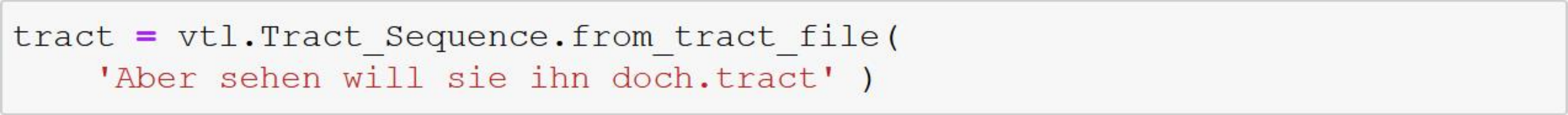
- Can be turned into (via MP)
  - Audio, SVG, limited states, ...
  - Transfer function objects
  - Tube state objects

Tract Sequences

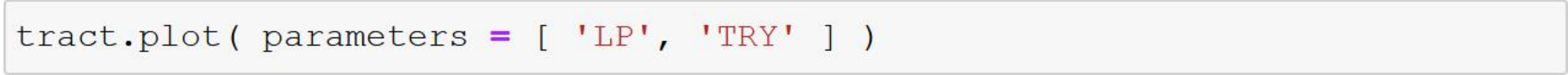


```
tract = vtl.Tract_Sequence.from_tract_file(
    'Aber sehen will sie ihn doch.tract' )
```

Tract Sequences

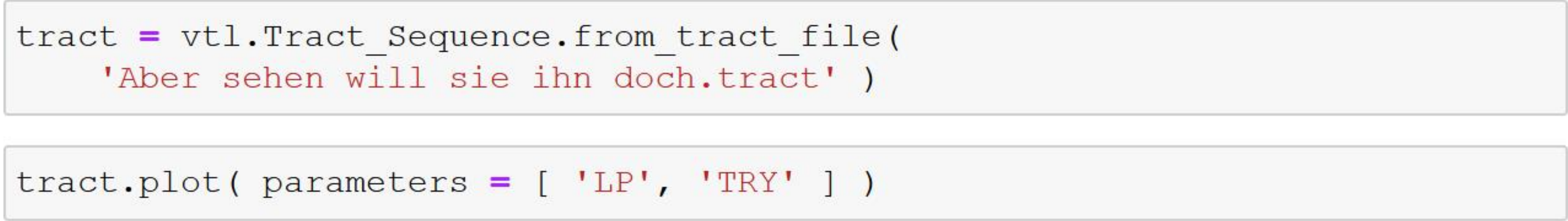


```
tract = vtl.Tract_Sequence.from_tract_file(
    'Aber sehen will sie ihn doch.tract' )
```



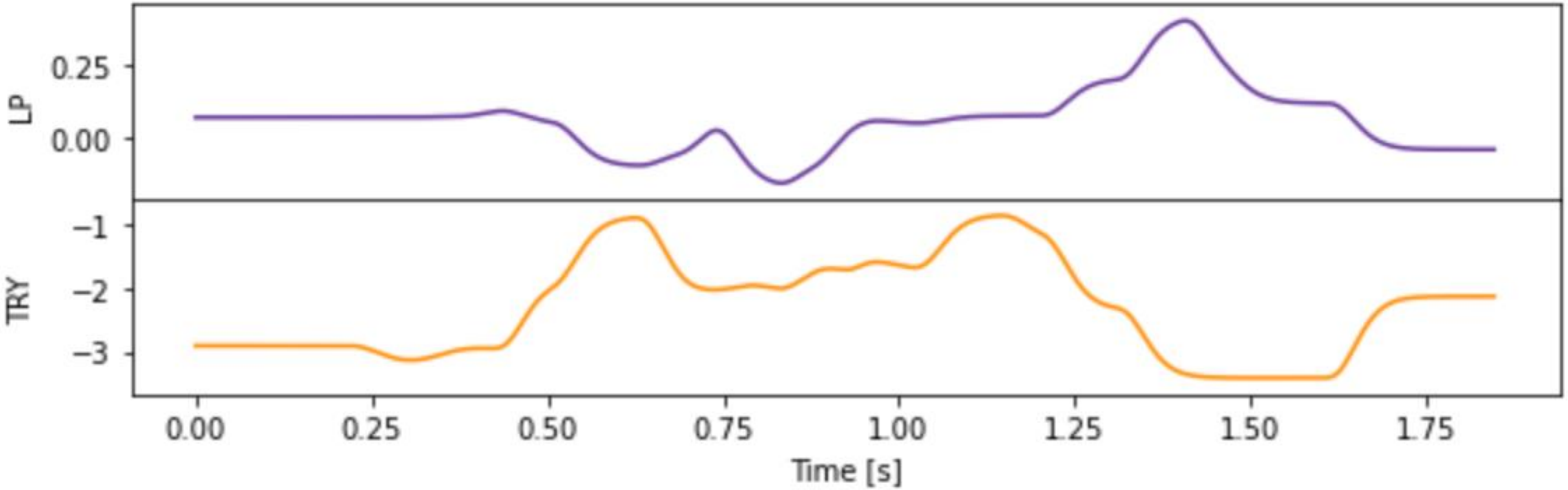
```
tract.plot( parameters = [ 'LP', 'TRY' ] )
```

Tract Sequences

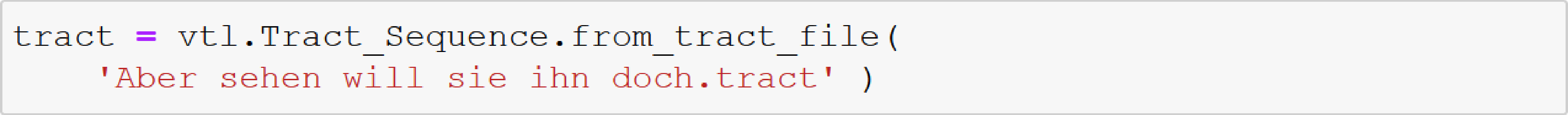


```
tract = vtl.Tract_Sequence.from_tract_file(
    'Aber sehen will sie ihn doch.tract' )
```

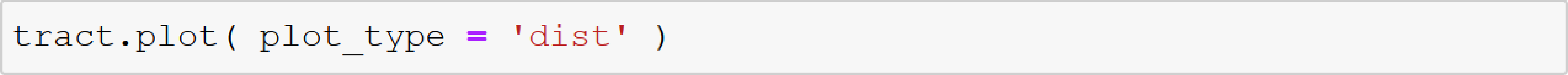
```
tract.plot( parameters = [ 'LP', 'TRY' ] )
```



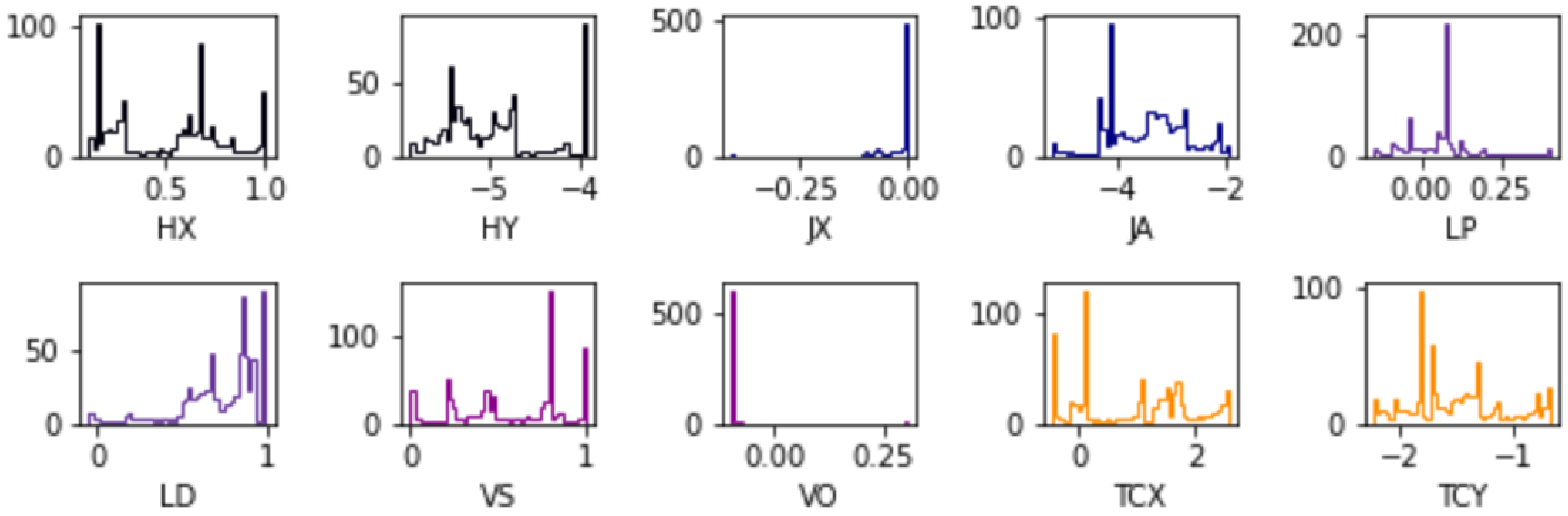
Tract Sequences



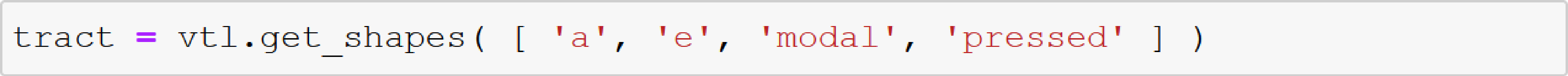
```
tract = vtl.Tract_Sequence.from_tract_file(
    'Aber sehen will sie ihn doch.tract' )
```



```
tract.plot( plot_type = 'dist' )
```



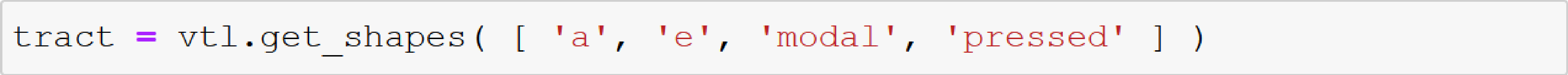
Target Scores



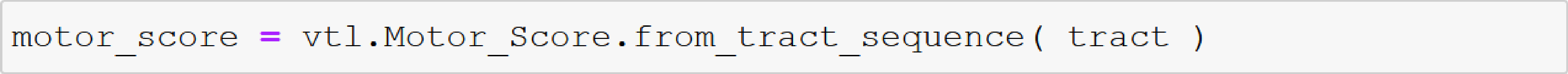
```
tract = vtl.get_shapes( [ 'a', 'e', 'modal', 'pressed' ] )
```

Target Scores

TECHNISCHE UNIVERSITÄT DRESDEN

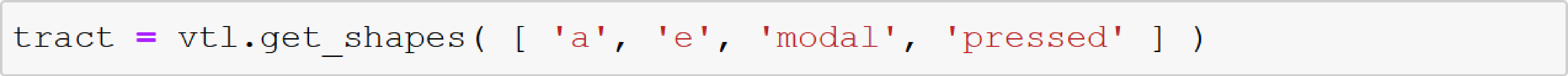


```
tract = vtl.get_shapes( [ 'a', 'e', 'modal', 'pressed' ] )
```

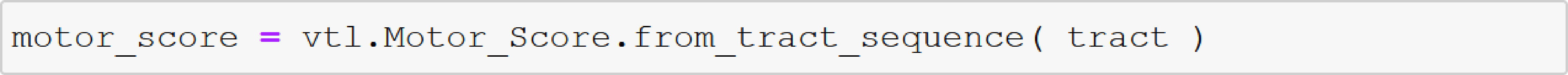


```
motor_score = vtl.Motor_Score.from_tract_sequence( tract )
```

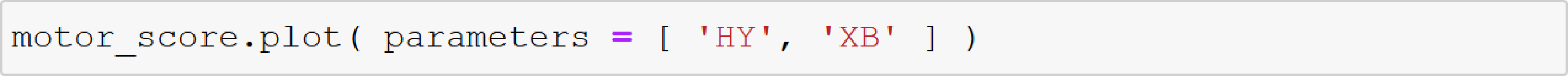
Target Scores



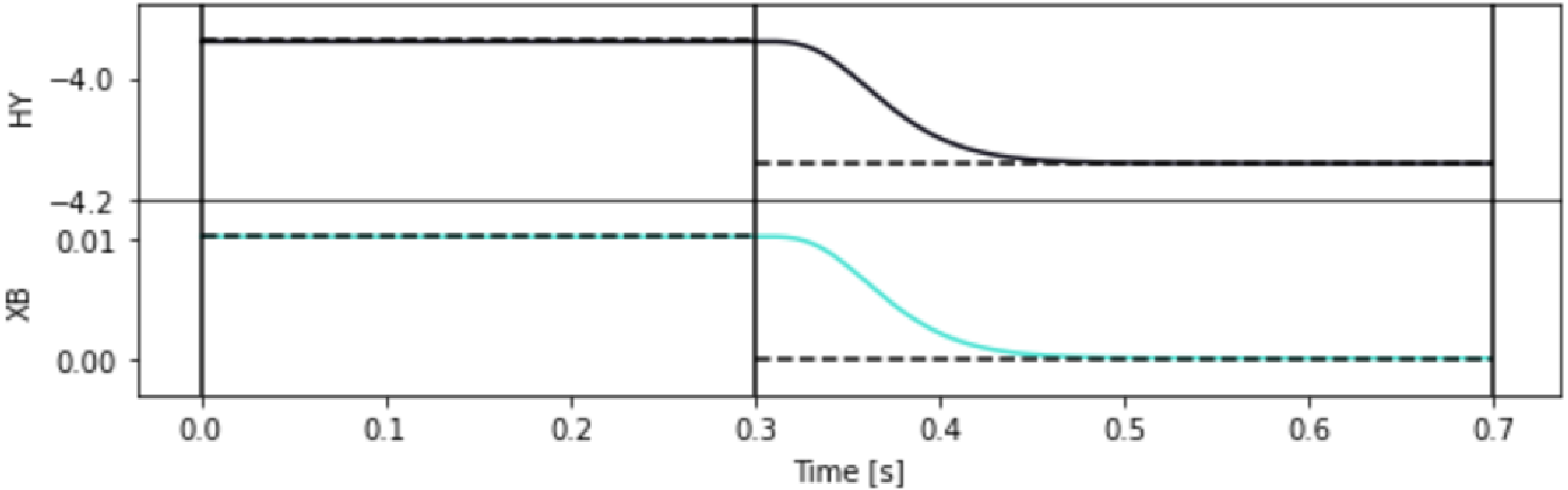
```
tract = vtl.get_shapes( [ 'a', 'e', 'modal', 'pressed' ] )
```



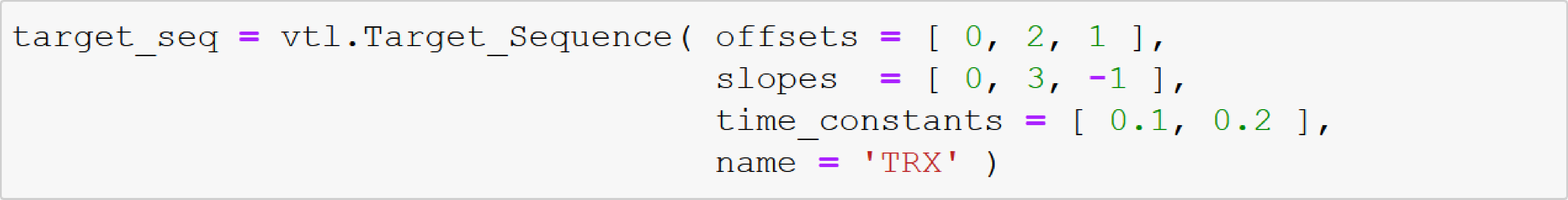
```
motor_score = vtl.Motor_Score.from_tract_sequence( tract )
```



```
motor_score.plot( parameters = [ 'HY', 'XB' ] )
```

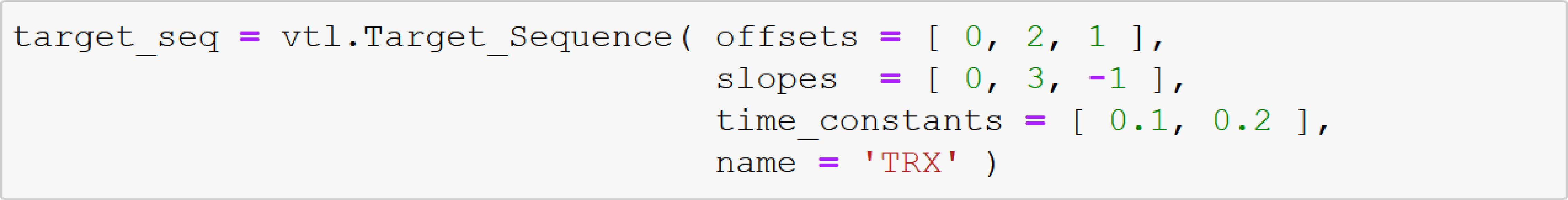


Target Scores

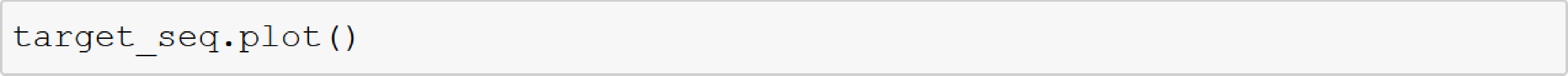


```
target_seq = vtl.Target_Sequence( offsets = [ 0, 2, 1 ],
                                  slopes  = [ 0, 3, -1 ],
                                  time_constants = [ 0.1, 0.2 ],
                                  name = 'TRX' )
```

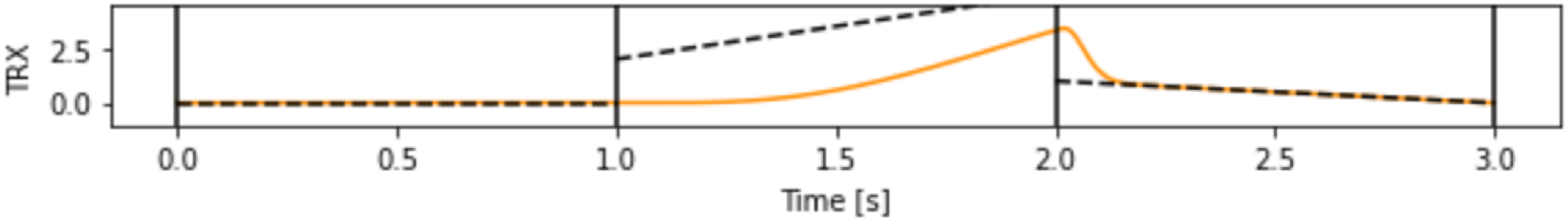
Target Scores



```
target_seq = vtl.Target_Sequence( offsets = [ 0, 2, 1 ],
                                  slopes  = [ 0, 3, -1 ],
                                  time_constants = [ 0.1, 0.2 ],
                                  name = 'TRX' )
```

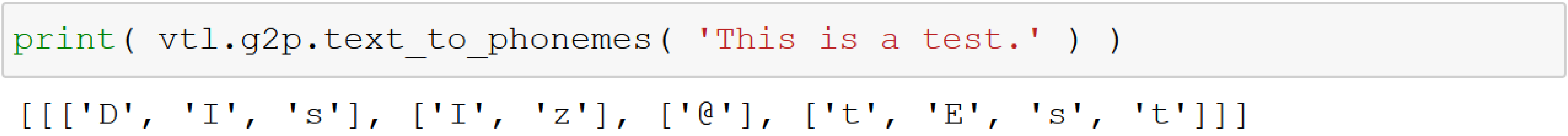


```
target_seq.plot()
```



Advanced Functions

- **g2p:**

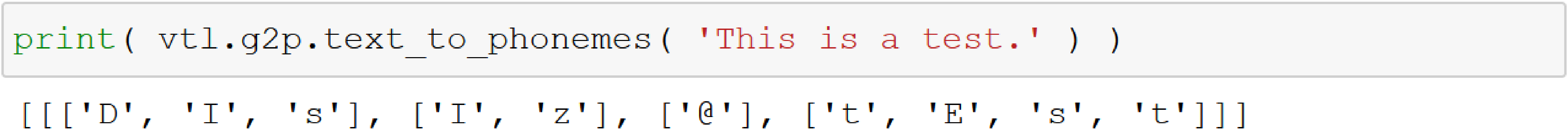


```
print( vtl.g2p.text_to_phonemes( 'This is a test.' ) )
```

```
[[['D', 'I', 's'], ['I', 'z'], ['@'], ['t', 'E', 's', 't']]]
```

Advanced Functions

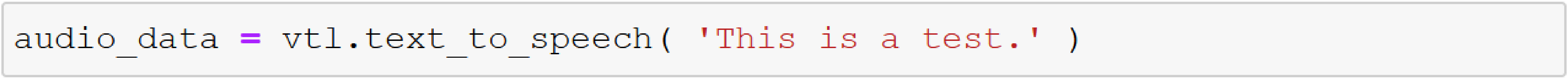
- **g2p:**



```
print( vtl.g2p.text_to_phonemes( 'This is a test.' ) )
```

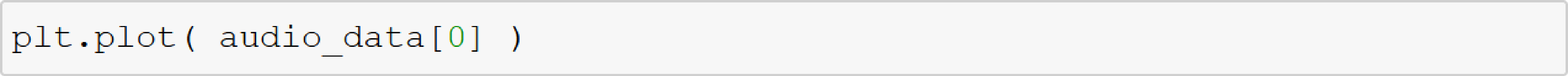
```
[[['D', 'I', 's'], ['I', 'z'], ['@'], ['t', 'E', 's', 't']]]
```

- **tts:**

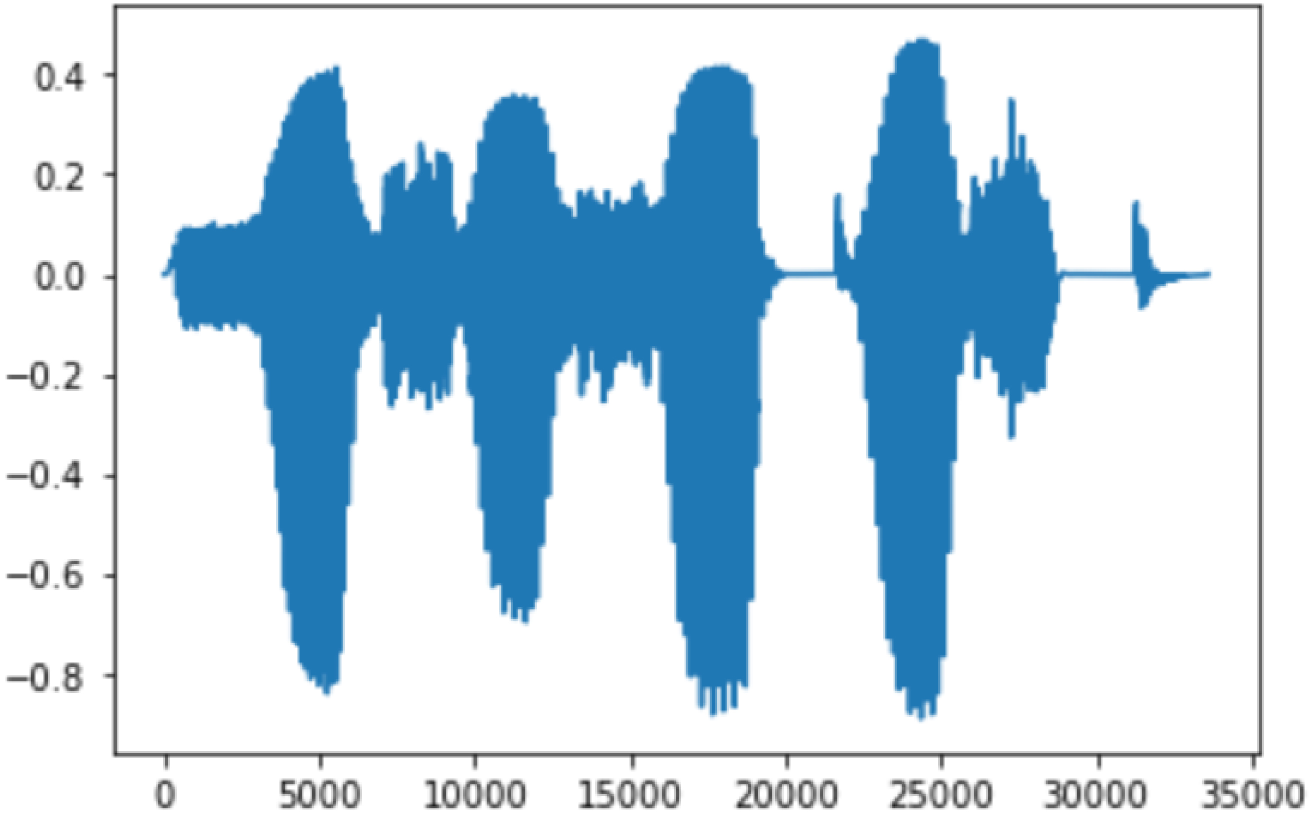


```
audio_data = vtl.text_to_speech( 'This is a test.' )
```

Advanced Functions



```
plt.plot( audio_data[0] )
```


0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0
5000
10000
15000
20000
25000
30000
35000