vector(l50)

array(l50,z)  
**L50=nb of resampled time blocks**

**Time dependent specific roughness**

**Specific roughness estimation**

**Resampling**1500 Hz → 50 Hz

**Low modulation rate weighting**

**Modulation rate estimation**identify harmonic complex {f0, nf0} based on the maximum energy

**High modulation rate weighting**

**Representative roughness**

**Time dependent roughness**

**For each time block l**

**For each frequency band z**

array(l,z,k)  
**with L=nb of time blocks   
 Z=CBF=53=nb of demi critical bands   
 K=demi block size due to DFT =256**

**Specific loudness**loudness\_ecma(signal, sb, sh)

**For each prominent peak i**

**Prominence computation**identify maximum difference level between each peak and the surrounding spectrum values  
+ only keep the 10 maximum values

**Peak picking**scipy.signal.find\_peaks(Phi\_E)

**Noise reduction of the envelope spectrum**

**Envelope calculation**Hilbert transform

**Downsampling**48 kHz → 1500 Hz

**Scaled power spectrum calculation**DFT using specific von-hann window  
+ weighting according to the specific loudness levels

**Segmentation into band-pass signals**\_rectified\_band\_pass\_signals(signal, sb, sh)  
found also in loudness\_ecma calculation