

---

## Table of Contents

.....	1
Import data .....	1
Bandpass using Gammatone Filterbank .....	1
Segment the data as needed (nonoverlapping) .....	1

```
addpath('toolboxes/');  
nsegments_in = 20; % select number of segments to process for testing
```

## Import data

```
[x, fs] = audioread('resources/heli_and_boat_short/  
boat2_short.wav'); %assume 44.1kHz  
%[x, fs] = audioread('resources/welcome16k.wav'); %assume 44.1kHz  
x = mean(x,2); % col vector  
  
% Resample to around 8KHz  
x = resample(x,2,11);  
fs = fs*2/11;  
xlen = length(x);  
  
% Construct final window  
ham_t = .25; %250 ms duration window  
ham_N = floor(ham_t*fs);  
w = hamming(ham_N);  
wshift = 4; %4hz  
exp_modulator = exp(1j*2*pi*wshift.*(1:ham_N)); %mod by 4 hz  
exp_modulator = exp_modulator.';  
w = w.*exp_modulator;
```

## Bandpass using Gammatone Filterbank

```
% Make the center frequency vector  
LOW_CF = 200;  
HIGH_CF = 4000;  
NUMCHANS = 18;  
CFS = iosr.auditory.makeErbCFs(LOW_CF,HIGH_CF,NUMCHANS);
```

## Segment the data as needed (nonoverlapping)

```
segmentlen = fs;  
nsegments_total = floor(xlen/segmentlen);  
  
nsegments = min(nsegments_in,nsegments_total); % for testing  
  
start_pos = 1;  
  
% Operate on each time segment  
for segmentind = 1:nsegments
```

---

```

end_pos = start_pos + segmentlen - 1;

x_segment = x(start_pos:end_pos);

BM = iosr.auditory.gammatoneFast(x_segment,CFS,fs); %operate on
every col

for channum = 1:NUMCHANS

    % calculate envelope and downsample
    envt = envelope(BM(:,channum)); %operate on every col
    envt = downsample(envt, 100);

    % normalize
    envt = envt./mean(abs(envt));

    % bp filter
    bp_sig = log10(abs(filter(w, 1, envt)));

    % threshold
    bp_sig(bp_sig>0) = 0;
    bp_sig(bp_sig<(-30)) = -30;

    out_chann(:,channum) = bp_sig;
end

out(:, :, segmentind) = out_chann;

start_pos = start_pos + segmentlen;

end

for segmentind = 1:nsegments
    figure;
    imagesc(out(:, :, segmentind));
    title(['mod spectrogram for frame number ' num2str(segmentind)]);
    xlabel('filterbank frequencies');
    ylabel('acoustic frequencies');
    axis xy; colormap(jet);
    colorbar;
end

```

```

Warning: Integer operands are required for colon operator when used as
index
Warning: Integer operands are required for colon operator when used as
index
Warning: Integer operands are required for colon operator when used as
index
Warning: Integer operands are required for colon operator when used as
index
Warning: Integer operands are required for colon operator when used as
index

```

---

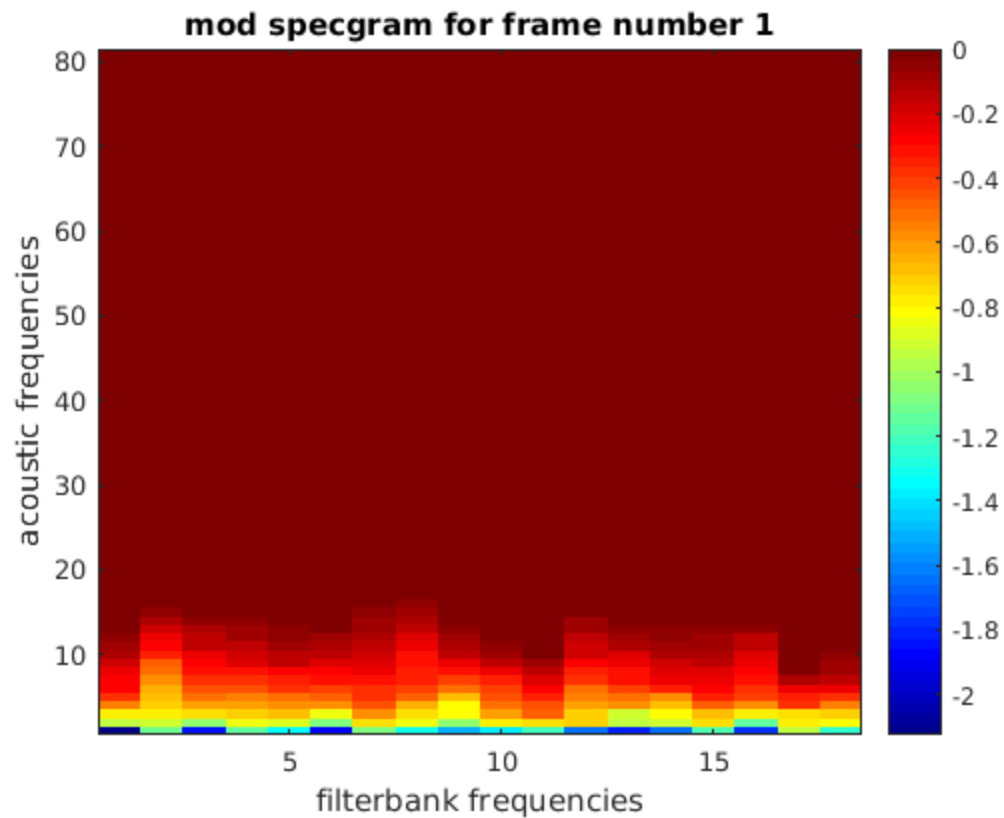
*Warning: Integer operands are required for colon operator when used as index*

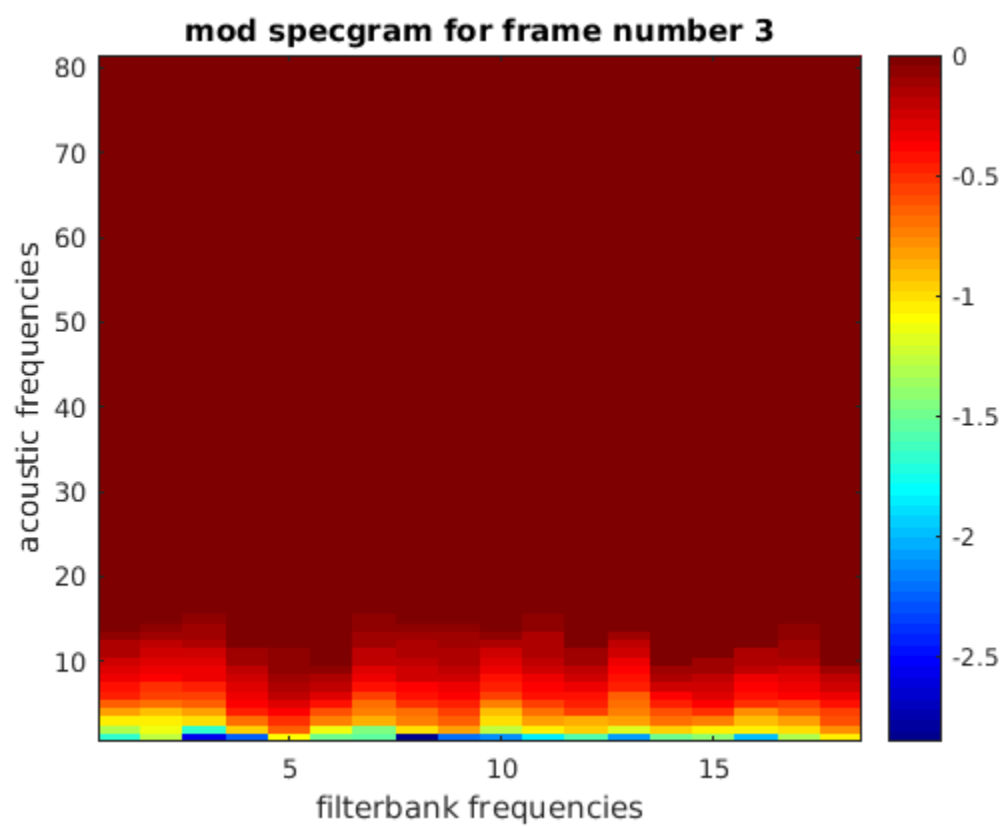
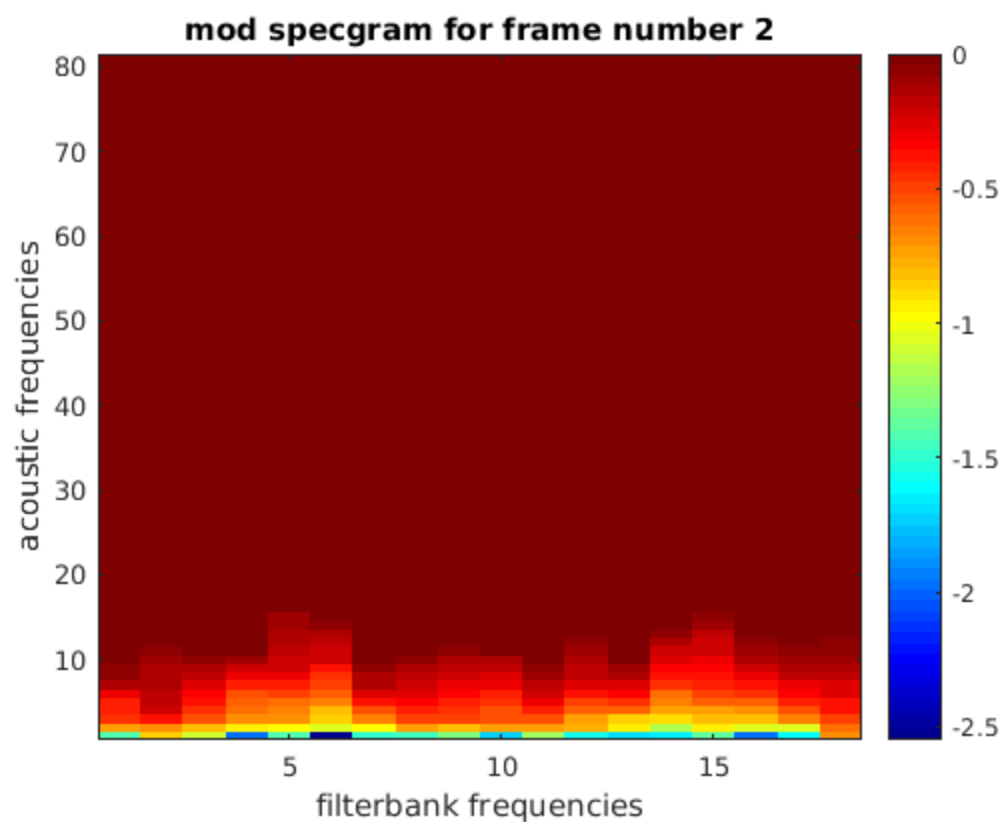
*Warning: Integer operands are required for colon operator when used as index*

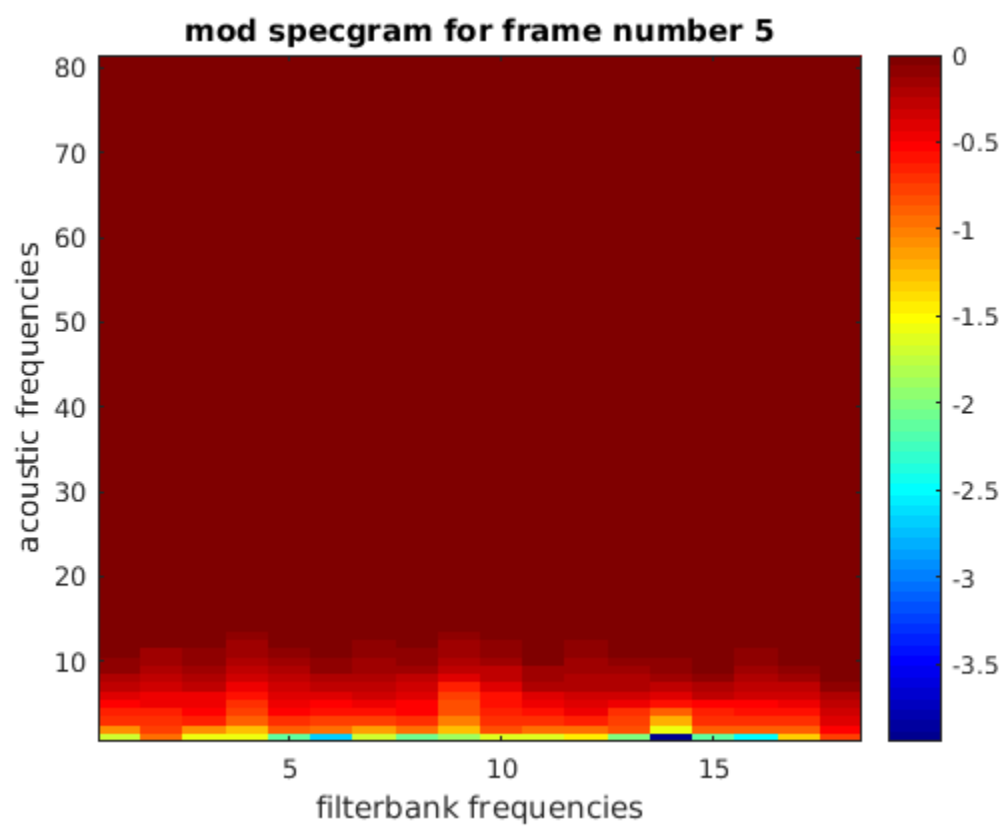
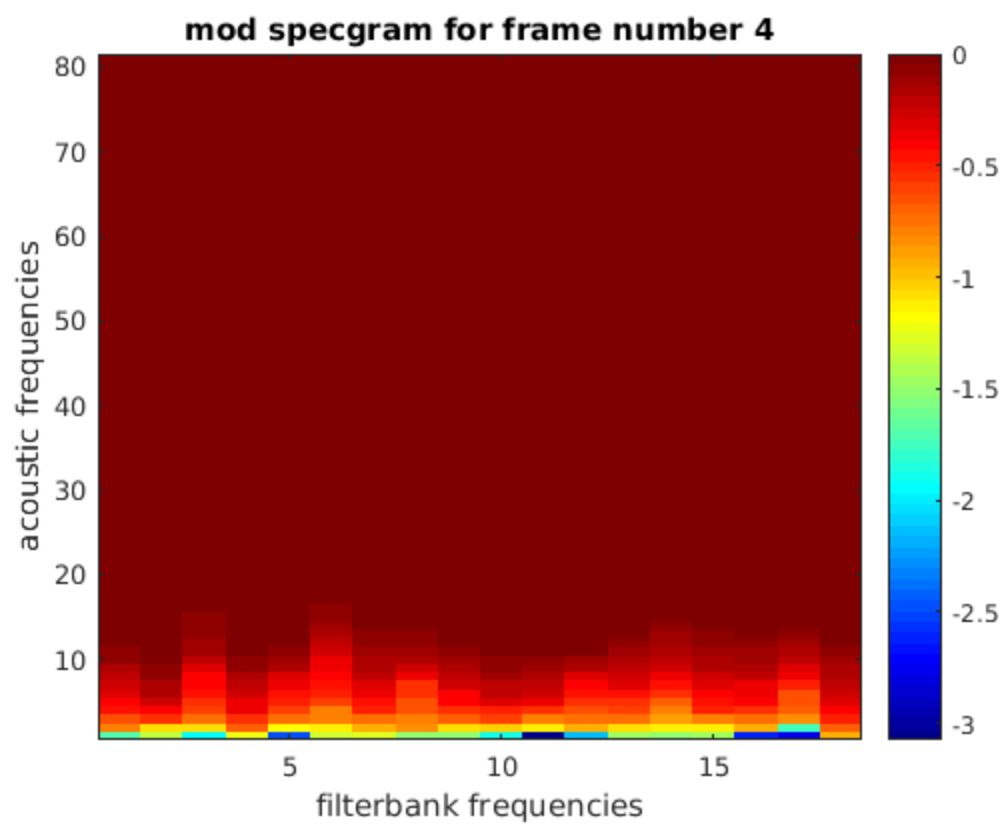
*Warning: Integer operands are required for colon operator when used as index*

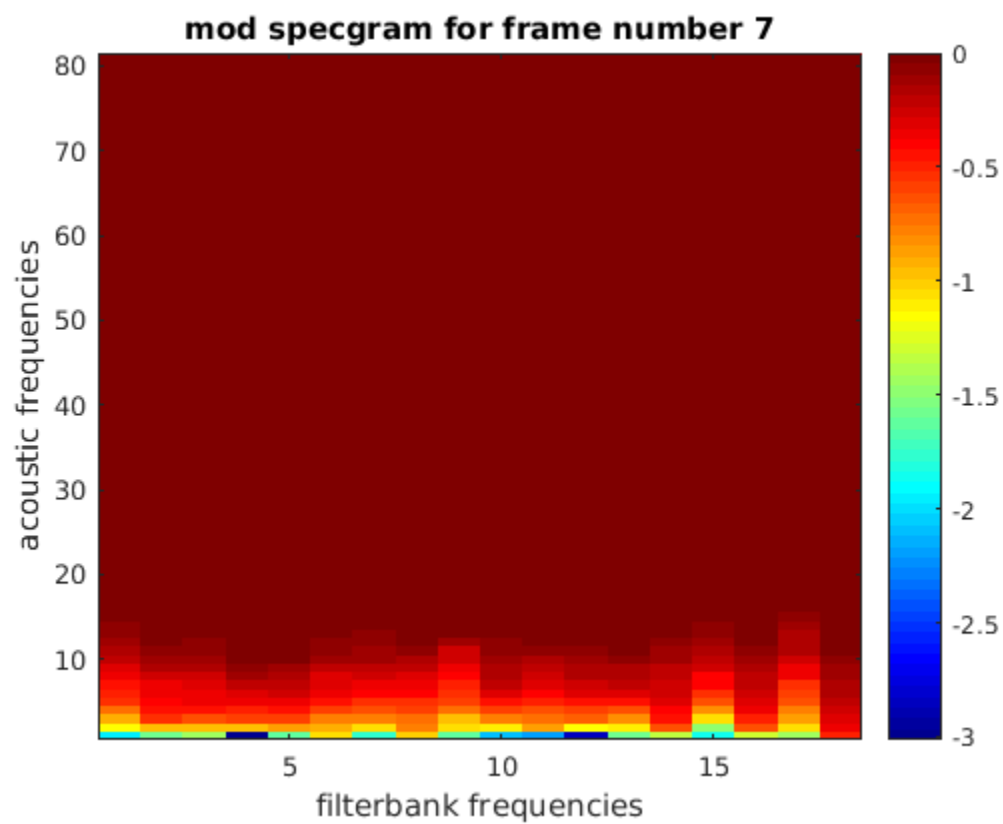
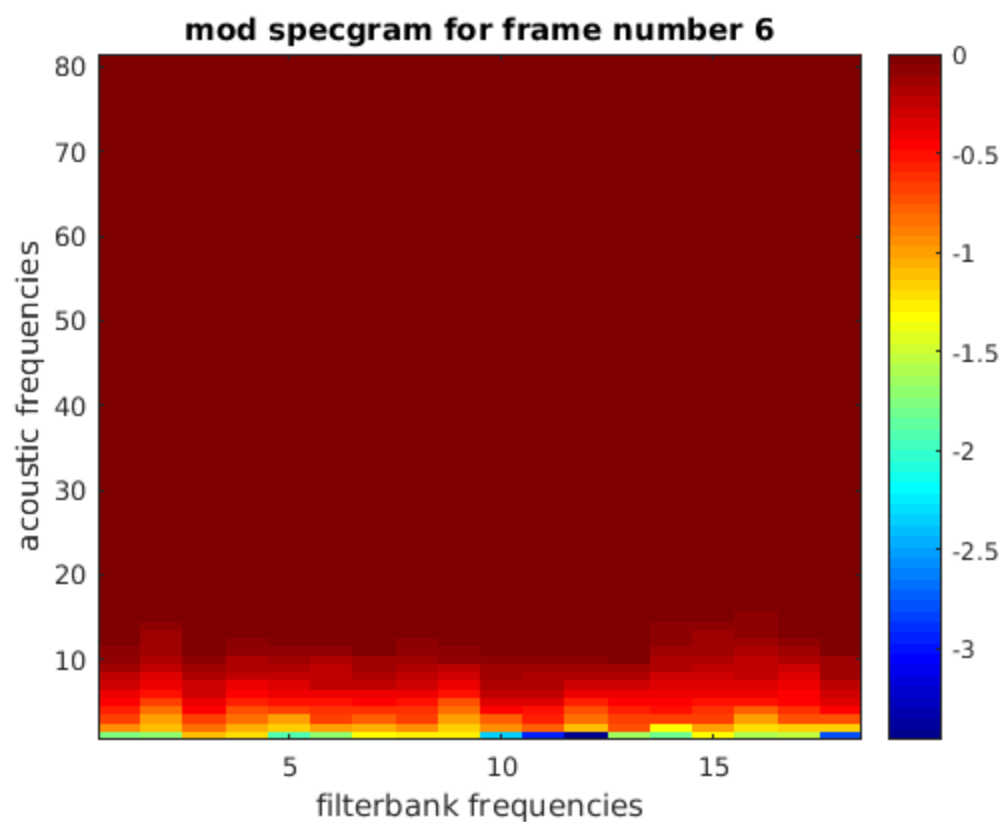
*Warning: Integer operands are required for colon operator when used as index*

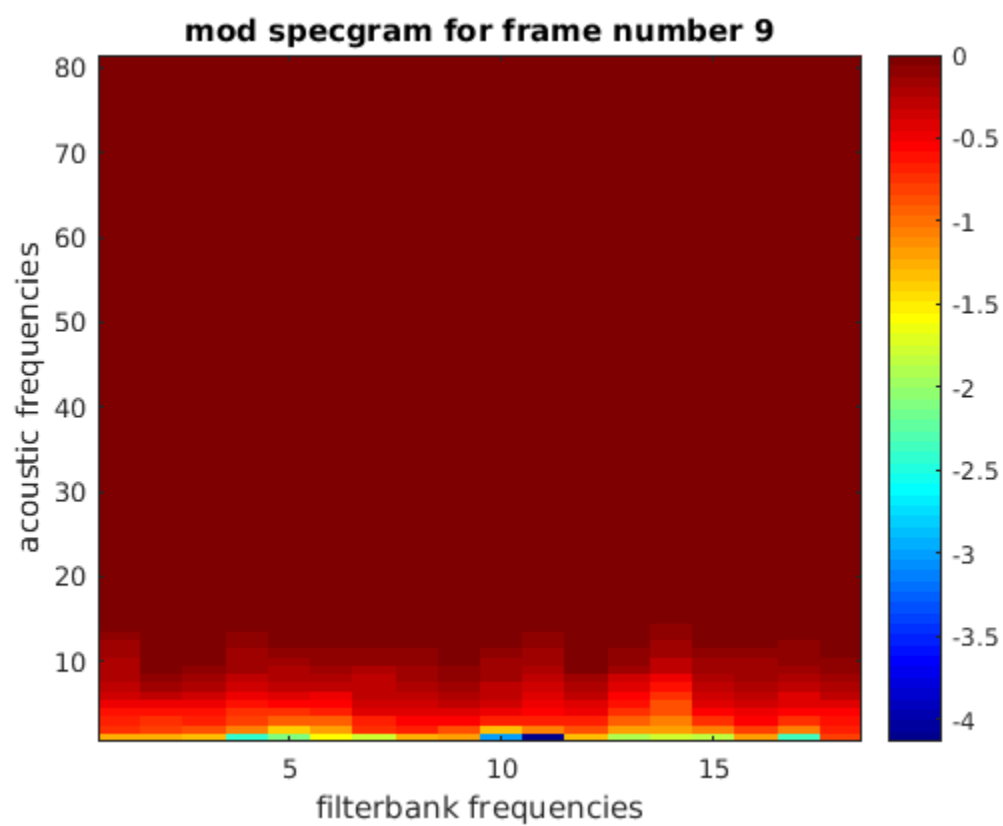
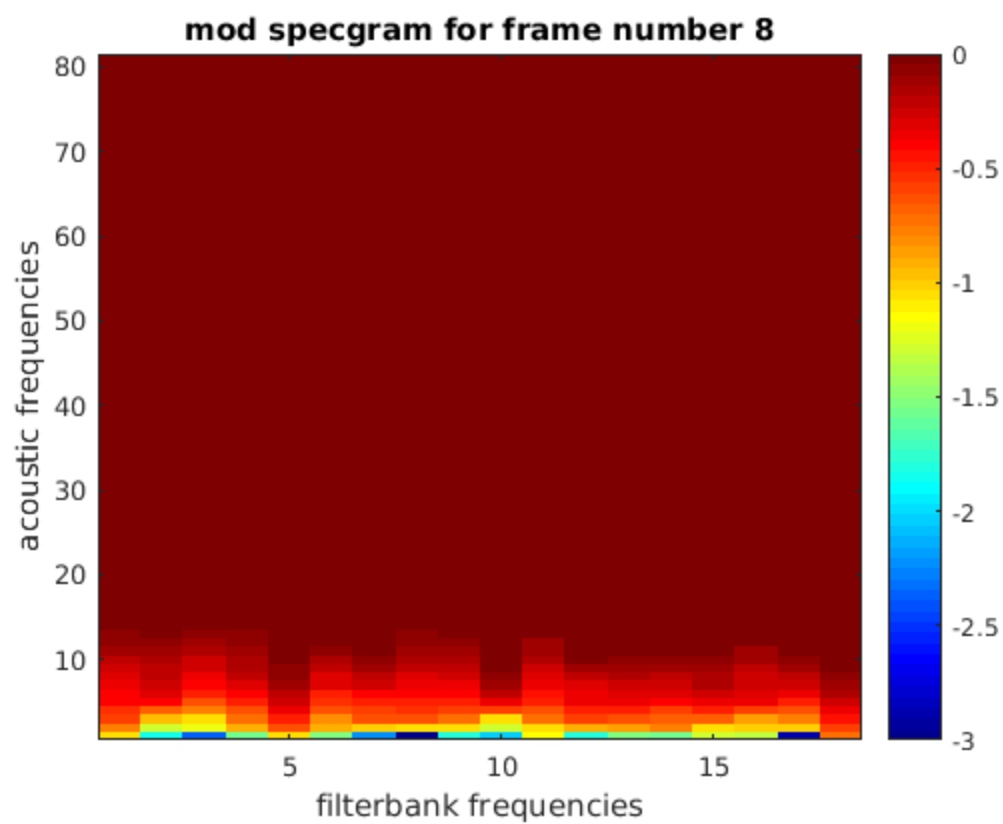
*Warning: Integer operands are required for colon operator when used as index*

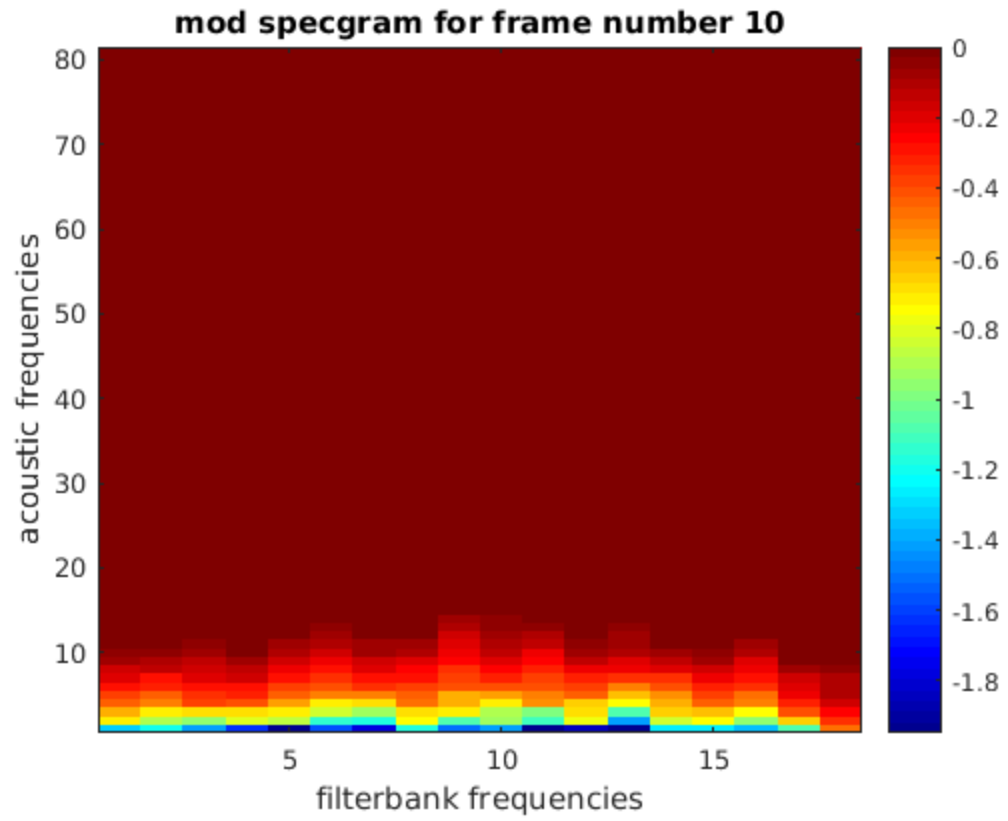












*Published with MATLAB® R2017a*