The procedure of installing the HHSA tool is simply adding the folder: \hhsa_tools as well as its subfolder \IFS to MATLAB search paths.

Example 1: *HHSA via core functions with scripts directly*

For the first example, we perform HHSA on a series of real EEG signals acquired from channel Fz during a trial in a visual working memory task. The data is simply a MATLAB row vector. With the exception of sampling rate and the maximum number of IMFs both in the first and second layer EMD, all the required and optional parameters are set in a MATLAB structure “S”. The following example code (for details, see ‘neurohhsa_ex1.m’ in the main directory of the tool) calculates the two-layer EMD with the enhanced Masking EMD algorithm:

```
load hhsa_ex1_data.mat    % load the example data at the subdirectory \scripts
fs=1000;    % specify the sampling rate
TNM=-1; TNM2=-1; % determine the maximum number of IMFs automatically
S.ifmethod='quad';
    % use “Direct Quadrature” to estimate 1st layer instantaneous frequency (IF).
S.ifmethod2='quad';    % use “Direct Quadrature” to estimate 2nd layer IF.
S.NEnsemble=0;    % set the order of Masking EMD for 1st layer EMD
S.NEnsemble2=0;    % set the order of Masking EMD for 2nd layer EMD
```

S.ENoise=2; % set the masking amplitude factor for 1^{st} layer EMD

S.ENoise2=2; % set the masking amplitude factor for 2^{nd} layer EMD

S.shiftLevel=1; % set the upsampling level, the new sampling rate is:

% (2^ S.shiftLevel)*(original sampling rate)

[fm, am, FM, AM, IMF, IMF2] =

multi_EMD_DCM_SV(data,fs,TNM,TNM2,S);

The function: "multi_EMD_DCM_SV" returns the IF and amplitude functions of first layer IMFs first, followed by the instantaneous AM frequencies of second layer IMFs, the AM amplitude functions of second layer IMFs, the first layer IMFs, and the second layer IMFs. With these returning results, the HHS can be obtained. The following codes calculate HHS for each IMF from time point 501 to 2500 with time resolution 500, from frequency 1 Hz to 128 Hz in log2 scale (i.e. 2^0-2^7), and from AM frequency 0.5 Hz to 64 Hz also in log2 scale (i.e. 2^{-1}-2^6):

S.dyadic=1; S.collapse=1; S.dyad_btw=8;

ntp0=501; ntp1=2500; tres=500; fw0=0; fw1=7; fres=[]; Fw0=-1; Fw1=6;

Fres=[];

[All_nt, fscale, Fscale]= **nspplotf3d_tres3x**(fm(:,1:end-1), FM(:,1:end,1:end-1), AM(:,1:end,1:end-1), ntp0, ntp1, fres, Fres, fw0, fw1, Fw0, Fw1, tres, S);

The function “nspplotf3d_tres3x” returns a four-dimensional array All_nt, which gives the HHS spectral power. In most cases, the last dimension of All_nt, corresponding to the first layer IMFs, should be collapsed to obtain the 3D full HHS (i.e. summing over the 4th dimension of All_nt). 2D marginal HHS can further be obtained by taking marginal sum over any one of the three dimensions of the 3D HHS. For instance, the 2D (AM frequency, carrier frequency) HHS can be acquired by summing marginally on the third dimension, i.e. the dimension of time points, of the 3D HHS. The following codes demonstrate how to generate the 3D HHS as well as the 2D (AM frequency, frequency) HHS from the outputs of the function nspplotf3d_tres3x:

```
All_nt3=sum( All_nt, 4); % obtain the 3D full HHS

All_nt2=sum( All_nt3, 3); % get the 2D (AM frequency × frequency) HHS
```

These 3D or 2D HHS can further be plotted with some MATLAB built-in graphic functions (e.g. “imagesc”, and “contour”). The following figure shows the 2D (AM frequency × frequency) HHS represented by the script ‘spmm_tfplot_r.m’ in the software tool.

Fig. 1

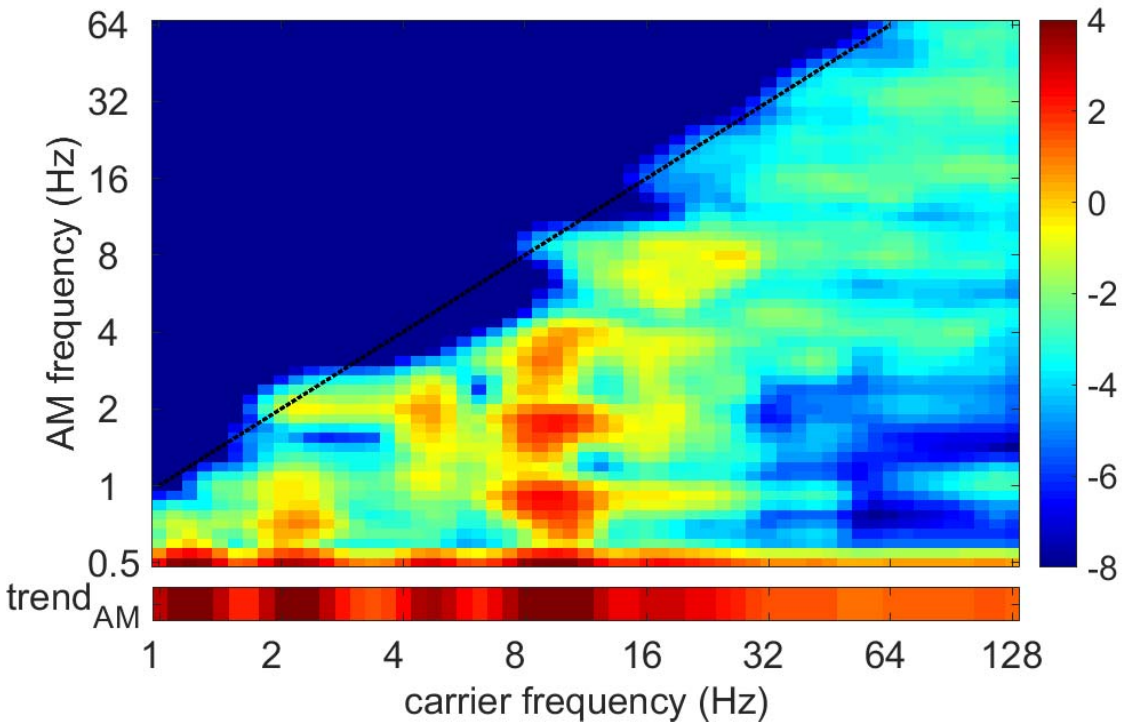

Figure 1. 2D (AM frequency × carrier frequency) HHS for data in example 1. In this result, the HHS AM power was smoothed by a Gaussian 2D filter first, and represented by log10 scale further.