CBM-Toolbox Script for Reversal Learning Data

This script fits reinforcement learning models to choice data from reversal learning using the Computational Behavioral Modeling toolbox by Piray et al., 2019 that applies hierarchical Bayesian inference during concurrent model fitting. In order to run successfully, you need add the cbm-toolbox ('codes') to your matlab-path, and have the 'model_RL' script in the Reversal_Vol_CBM folder. The model script contains a standard Rescorla Wagner model-space with Single Update, individually weighted Double Update and full Double update, with one or two learning rates (win/loss), combined with a softmax (1 beta).

1) Prepare data for cbm

adapt the paths for your computer

```
clc
clear all
close;
cd('/Users/larawieland/Owncloud/Shared/Reversal_Vol');
addpath('/Users/larawieland/Documents/ECN/Promotion/Modeling/cbm/codes'); % adapt
% addpath('C:\Users\katthagt\ownCloud\Shared\NegSym\Modeling\hgfToolBox_v4.15');
input_path = '/Users/larawieland/Owncloud/Shared/Reversal_Vol/data/T1/'; %behavioral da

% load data
input =  dir([input_path, '*WS.mat']);
data = {};
nmodels = 6;
nsubjects = length(input);

% subject loop to store all individual choice data in cohensive input file needed for c
for sub = 1:length(input)
data{sub,1}.code    = input(sub).name(12:19);
data{sub,1}.T       = input(sub).name(18:19);
load([input_path input(sub).name], 'R', 'A');
R = R(~isnan(A));  A = A(~isnan(A)); R(R==-1)=0; %check whether your model needs losses
data{sub,1}.actions = A;
data{sub,1}.outcome = R;
end
```

2) check if model is running correctly with random parameters (no actual fitting)

Note: The cbm files were modified, so that one learning script can fit different models depending on the model number, that is the third input now in the standard fitting function loglik = model_RL(parameters,subj, model_no). Check section 3 for amount of free parameters for each model.

Model numbers (model_no):

1 --> SU 1 alpha, 2 --> SU 2 alphas

3 --> iDU 1 alpha, 2 --> iDU 2 alphas

4 --> DU 1 alpha, 2 --> DU 2 alphas

```
% loglik = model_RL(parameters,subj1)
```

```
subj1 = data{1};
parameters = randn(1,3);
% F1 = model_RL(parameters,subj1,2),
[F1, traj] = model_RL_DU2al(parameters,subj1)
```

```
F1 = -152.7861
traj = struct with fields:
     a: [159×1 double]
     o: [159×1 double]
     q: [159×2 double]
    da: [1×318 double]
```

3) use cbm for initial model fitting and group prior estimation

The function 'cbm_lap' fits every model to each subject data separately (i.e. in a non-hierarchical fashion, so that only (random) individuals are fitted first without any (fixed) group information)

It uses Laplace approximation needing a normal prior for every parameter (see manual) with:

mean = 0 and variance = 6.25.

All parameters are normalized (-Inf to Inf), thus parameters between 0:1 need to be sigmoid-transformed (--> logit-space) and parameters > 0 must be exponential-transformed (log-space) in the beginning of the learning model-file.

crucial function: cbm_lap(data, @model, priors, output file,[ ], model_no)

```
v                 = 6.25;    % variance is the same for all priors
% name output files
fname_RL = {};
fname_RL{1}='SU1al.mat';  fname_RL{2}='SU2al.mat'; fname_RL{3}='iDU1al.mat';
fname_RL{4}='iDU2al.mat'; fname_RL{5}='DU1al.mat'; fname_RL{6}='DU2al.mat';

% set amount of priors for each model
prior(1)     = struct('mean', zeros(2,1),'variance',v);          % 2 free parameters (a
prior(2)     = struct('mean', zeros(3,1),'variance',v);          % 3 free parameters (a
prior(3)     = struct('mean', zeros(3,1),'variance',v);          % 3 free parameters (a
prior(4)     = struct('mean', zeros(4,1),'variance',v);          % 4 free parameters (a
prior(5)     = struct('mean', zeros(2,1),'variance',v);          % 2 free parameters (a
prior(6)     = struct('mean', zeros(3,1),'variance',v);          % 3 free parameters (a

models = {'@model_RL_SU1al', '@model_RL_SU2al','@model_RL_iDU1al', '@model_RL_iDU2al',

cbm_lap(data, @model_RL_SU1al,  prior(1), char(fname_RL(1))); % last input for model nu
```

```
cbm_lap                                            03-Mar-2020 14:48:39
=======================================================================
Number of samples: 49
Number of parameters: 2

Number of initializations: 14
-----------------------------------------------------------------------
```

```
Subject: 01
Subject: 02
Subject: 03
Subject: 04
Subject: 05
Subject: 06
Subject: 07
Subject: 08
Subject: 09
Subject: 10
Subject: 11
Subject: 12
Subject: 13
Subject: 14
Subject: 15
Subject: 16
Subject: 17
Subject: 18
Subject: 19
Subject: 20
Subject: 21
Subject: 22
Subject: 23
Subject: 24
Subject: 25
Subject: 26
Subject: 27
Subject: 28
Subject: 29
Subject: 30
Subject: 31
Subject: 32
Subject: 33
Subject: 34
Subject: 35
Subject: 36
Subject: 37
Subject: 38
Subject: 39
Subject: 40
Subject: 41
Subject: 42
Subject: 43
Subject: 44
Subject: 45
Subject: 46
Subject: 47
Subject: 48
Subject: 49
done :]
```

```
cbm_lap(data, @model_RL_SU2al,  prior(2), char(fname_RL(2))); % last input for model nu
```

```
cbm_lap                                       03-Mar-2020 14:48:48
==================================================================
Number of samples: 49
Number of parameters: 3

Number of initializations: 21
------------------------------------------------------------------
Subject: 01
Subject: 02
Subject: 03
Subject: 04
```

```
Subject: 05
Subject: 06
Subject: 07
Subject: 08
Subject: 09
Subject: 10
Subject: 11
Subject: 12
Subject: 13
Subject: 14
Subject: 15
Subject: 16
Subject: 17
Subject: 18
Subject: 19
Subject: 20
Subject: 21
Subject: 22
Subject: 23
Subject: 24
Subject: 25
Subject: 26
Subject: 27
Subject: 28
Subject: 29
Subject: 30
Subject: 31
Subject: 32
Subject: 33
Subject: 34
Subject: 35
Subject: 36
Subject: 37
Subject: 38
Subject: 39
Subject: 40
Subject: 41
Subject: 42
Subject: 43
Subject: 44
Subject: 45
Subject: 46
Subject: 47
Subject: 48
Subject: 49
done :]
```

```
cbm_lap(data, @model_RL_iDU1al, prior(3), char(fname_RL(3))); % last input for model nu
```

```
cbm_lap                                           03-Mar-2020 14:49:04
======================================================================
Number of samples: 49
Number of parameters: 3

Number of initializations: 21
----------------------------------------------------------------------
Subject: 01
Subject: 02
Subject: 03
Subject: 04
Subject: 05
Subject: 06
Subject: 07
Subject: 08
```

```
Subject: 09
Subject: 10
Subject: 11
Subject: 12
Subject: 13
Subject: 14
Subject: 15
Subject: 16
Subject: 17
Subject: 18
Subject: 19
Subject: 20
Subject: 21
Subject: 22
Subject: 23
Subject: 24
Subject: 25
Subject: 26
Subject: 27
Subject: 28
Subject: 29
Subject: 30
Subject: 31
Subject: 32
Subject: 33
Subject: 34
Subject: 35
Subject: 36
Subject: 37
Subject: 38
Subject: 39
Subject: 40
Subject: 41
Subject: 42
Subject: 43
Subject: 44
Subject: 45
Subject: 46
Subject: 47
Subject: 48
Subject: 49
done :]
```

```
cbm_lap(data, @model_RL_iDU2al, prior(4), char(fname_RL(4)));% last input for model nu
```

```
cbm_lap                                           03-Mar-2020 14:49:20
======================================================================
Number of samples: 49
Number of parameters: 4

Number of initializations: 28
----------------------------------------------------------------------
Subject: 01
Subject: 02
Subject: 03
Subject: 04
Subject: 05
Subject: 06
Subject: 07
Subject: 08
Subject: 09
Subject: 10
Subject: 11
Subject: 12
```

```
Subject: 13
Subject: 14
Subject: 15
Subject: 16
Subject: 17
Subject: 18
Subject: 19
Subject: 20
Subject: 21
Subject: 22
Subject: 23
Subject: 24
Subject: 25
Subject: 26
Subject: 27
Subject: 28
Subject: 29
Subject: 30
Subject: 31
Subject: 32
Subject: 33
Subject: 34
Subject: 35
Subject: 36
Subject: 37
Subject: 38
Subject: 39
Subject: 40
Subject: 41
Subject: 42
Subject: 43
Subject: 44
Subject: 45
Subject: 46
Subject: 47
Subject: 48
Subject: 49
done :]
```

```
cbm_lap(data, @model_RL_DU1al,  prior(5), char(fname_RL(5))); % last input for model nu
```

```
cbm_lap                                         03-Mar-2020 14:49:47
====================================================================
Number of samples: 49
Number of parameters: 2

Number of initializations: 14
--------------------------------------------------------------------
Subject: 01
Subject: 02
Subject: 03
Subject: 04
Subject: 05
Subject: 06
Subject: 07
Subject: 08
Subject: 09
Subject: 10
Subject: 11
Subject: 12
Subject: 13
Subject: 14
Subject: 15
Subject: 16
```

```
Subject: 17
Subject: 18
Subject: 19
Subject: 20
Subject: 21
Subject: 22
Subject: 23
Subject: 24
Subject: 25
Subject: 26
Subject: 27
Subject: 28
Subject: 29
Subject: 30
Subject: 31
Subject: 32
Subject: 33
Subject: 34
Subject: 35
Subject: 36
Subject: 37
Subject: 38
Subject: 39
Subject: 40
Subject: 41
Subject: 42
Subject: 43
Subject: 44
Subject: 45
Subject: 46
Subject: 47
Subject: 48
Subject: 49
done :]
```

```
cbm_lap(data, @model_RL_DU2al,  prior(6), char(fname_RL(6))); % last input for model nu
```

```
cbm_lap                                          03-Mar-2020 14:49:56
======================================================================
Number of samples: 49
Number of parameters: 3

Number of initializations: 21
----------------------------------------------------------------------
Subject: 01
Subject: 02
Subject: 03
Subject: 04
Subject: 05
Subject: 06
Subject: 07
Subject: 08
Subject: 09
Subject: 10
Subject: 11
Subject: 12
Subject: 13
Subject: 14
Subject: 15
Subject: 16
Subject: 17
Subject: 18
Subject: 19
Subject: 20
```

```
Subject: 21
Subject: 22
Subject: 23
Subject: 24
Subject: 25
Subject: 26
Subject: 27
Subject: 28
Subject: 29
Subject: 30
Subject: 31
Subject: 32
Subject: 33
Subject: 34
Subject: 35
Subject: 36
Subject: 37
Subject: 38
Subject: 39
Subject: 40
Subject: 41
Subject: 42
Subject: 43
Subject: 44
Subject: 45
Subject: 46
Subject: 47
Subject: 48
Subject: 49
done :]
```

5) hierarchical Bayesian inference: concurrent model fitting

Here, we use the parameters from step 3 as priors for concurrent model-fitting (all models are fitted within one step). Using hierarchical bayesian inference, random (individuals') parameters inform the fixed (group) parameters according to the likelihood of this model being the best model for the respective subject ("responsibility").

Central function: cbm_hbi(data,nmodels,fcbm_maps,fname_hbi);

```
% 2nd input: a cell input containing function handle to models (== @ before the name o
models = {@model_RL_SU1al, @model_RL_SU2al, @model_RL_iDU1al, @model_RL_iDU2al, @model_

% 3rd input: another cell input containing file-address to files saved by
% cbm_lap, note that they corresponds to models (so pay attention to the order)
fcbm_maps = {'SU1al.mat','SU2al.mat','iDU1al.mat','iDU2al.mat','DU1al.mat','DU2al.mat'

% 4th input: a file address for saving the output
fname_hbi = 'hbi_RL_fullmodelspace.mat';

cbm_hbi(data,models,fcbm_maps,fname_hbi);
```

```
cbm_hbi_hbi                                         03-Mar-2020 14:50:35
Running hierarchical bayesian inference (HBI)...
```

```
HBI has been initialized according to
 SU1al.mat [for model 1]
 SU2al.mat [for model 2]
 iDU1al.mat [for model 3]
 iDU2al.mat [for model 4]
 DU1al.mat [for model 5]
 DU2al.mat [for model 6]

Number of samples: 49
Number of models: 6
======================================================================
Iteration 01
Iteration 02
 model frequencies (percent)
 model 1: 18.1| model 2: 19.5| model 3: 17.9| model 4: 26.6| model 5: 9.3| model 6: 8.5|
                                                           dL:   51.38
                                                           dm:   73.42
                                                           dx:    0.50
Iteration 03
 model frequencies (percent)
 model 1: 17.7| model 2: 15.2| model 3: 20.4| model 4: 36.2| model 5: 6.6| model 6: 3.9|
                                                           dL:   15.26
                                                           dm:    9.64
                                                           dx:    0.30
Iteration 04
 model frequencies (percent)
 model 1: 16.3| model 2: 10.6| model 3: 20.9| model 4: 45.4| model 5: 5.0| model 6: 1.7|
                                                           dL:    9.68
                                                           dm:    9.22
                                                           dx:    0.29
Iteration 05
 model frequencies (percent)
 model 1: 15.4| model 2: 7.4| model 3: 19.1| model 4: 53.1| model 5: 4.3| model 6: 0.7|
                                                           dL:    7.13
                                                           dm:    7.69
                                                           dx:    0.23
Iteration 06
 model frequencies (percent)
 model 1: 14.9| model 2: 5.5| model 3: 15.5| model 4: 59.5| model 5: 4.4| model 6: 0.3|
                                                           dL:    6.02
                                                           dm:    6.38
                                                           dx:    0.20
Iteration 07
 model frequencies (percent)
 model 1: 14.4| model 2: 5.5| model 3: 10.7| model 4: 65.0| model 5: 4.3| model 6: 0.1|
                                                           dL:    4.23
                                                           dm:    5.53
                                                           dx:    0.21
Iteration 08
 model frequencies (percent)
 model 1: 14.1| model 2: 5.9| model 3: 6.0| model 4: 70.0| model 5: 4.0| model 6: 0.0|
                                                           dL:    4.19
                                                           dm:    4.95
                                                           dx:    0.21
Iteration 09
 model frequencies (percent)
 model 1: 13.9| model 2: 6.4| model 3: 2.2| model 4: 73.8| model 5: 3.7| model 6: 0.0|
                                                           dL:    4.46
                                                           dm:    3.81
                                                           dx:    0.27
Iteration 10
 model frequencies (percent)
 model 1: 13.5| model 2: 6.5| model 3: 0.4| model 4: 75.8| model 5: 3.8| model 6: 0.0|
                                                           dL:    3.42
```

```
                                                                 dm:    1.99
                                                                 dx:    0.33
Iteration 11
 model frequencies (percent)
 model 1: 13.2| model 2: 6.5| model 3: 0.0| model 4: 76.4| model 5: 3.9| model 6: 0.0|
                                                                 dL:    2.03
                                                                 dm:    0.63
                                                                 dx:    0.18
Iteration 12
 model frequencies (percent)
 model 1: 12.9| model 2: 6.5| model 3: 0.0| model 4: 76.7| model 5: 3.9| model 6: 0.0|
                                                                 dL:    0.32
                                                                 dm:    0.28
                                                                 dx:    0.22
Iteration 13
 model frequencies (percent)
 model 1: 12.7| model 2: 6.5| model 3: 0.0| model 4: 76.9| model 5: 3.9| model 6: 0.0|
                                                                 dL:    0.24
                                                                 dm:    0.20
                                                                 dx:    0.03
Iteration 14
 model frequencies (percent)
 model 1: 12.5| model 2: 6.5| model 3: 0.0| model 4: 77.1| model 5: 3.9| model 6: 0.0|
                                                                 dL:    0.20
                                                                 dm:    0.16
                                                                 dx:    0.01
Iteration 15
 model frequencies (percent)
 model 1: 12.4| model 2: 6.5| model 3: 0.0| model 4: 77.2| model 5: 3.9| model 6: 0.0|
                                                                 dL:    0.17
                                                                 dm:    0.14
                                                                 dx:    0.01
                                                                Converged :]
```

6) Plot the hbi parameters

As in step 4, display the fitted parameters as implemented in the toolbox function: 'cbm_hbi_plot(fname_hbi, model_names, param_names, transform, model of interest)'

1st input is the file-address of the file saved by cbm_hbi

2nd input: a cell input containing model names

3rd input: another cell input containing parameter names of the winning model

4th input: another cell input containing transformation function associated with each parameter of the winning model

5th input: optional, indicates model of interest

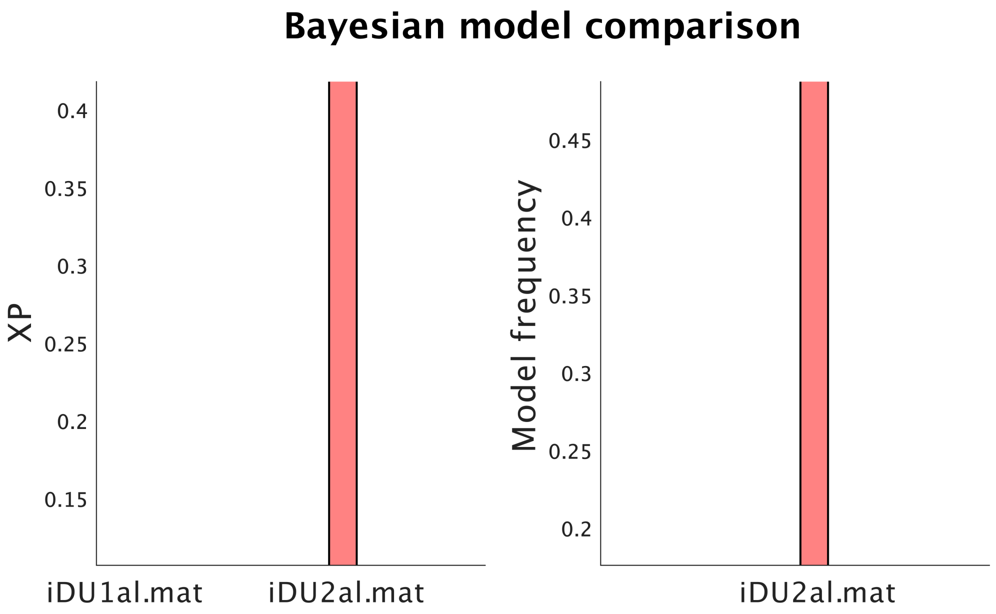
```
fname_hbi = 'hbi_RL_fullmodelspace.mat'; % 1st input is the file-address of the file sa
hbi = load('hbi_RL_fullmodelspace.mat');
cbm = hbi.cbm;
cbm.output
```

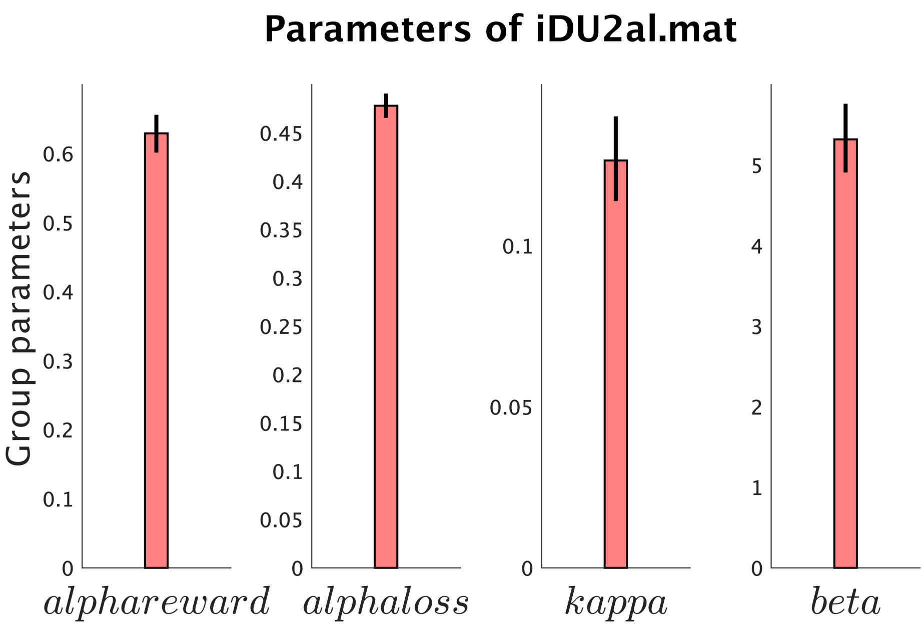
```
ans = struct with fields:
                    parameters: {6×1 cell}
                responsibility: [49×6 double]
```

```
                 group_mean: {[0.5749 1.6501]  [0.0294 -3.4633 1.0423]  [4.3103e-05 1.0953e-05 2.3220
   group_hierarchical_errorbar: {[0.1881 0.3432]  [0.1264 1.2730 0.6276]  [0.1414 0.1413 0.1414]  [0.117
            model_frequency: [0.1238 0.0650 2.2483e-05 0.7720 0.0391 8.7308e-05]
            exceedance_prob: [0 0 0 1 0 0]
   protected_exceedance_prob: [NaN NaN NaN NaN NaN NaN]
```

```
[M,I]=max(cbm.output.exceedance_prob);    % find the model with the highest exceedance p

param_names(1).p         = {'alpha', 'beta'};
param_names(1).transform= {'sigmoid', 'exp'};
param_names(2).p         = {'alpha reward', 'alpha loss', 'beta'};
param_names(2).transform= {'sigmoid', 'sigmoid', 'exp'};
param_names(3).p         = {'alpha', 'kappa', 'beta'};
param_names(3).transform= {'sigmoid', 'sigmoid', 'exp'};
param_names(4).p         = {'alpha reward', 'alpha loss', 'kappa', 'beta'};
param_names(4).transform= {'sigmoid','sigmoid','sigmoid','exp'};
param_names(5)    = param_names(1);
param_names(6)    = param_names(2);

cbm_hbi_plot(fname_hbi, fname_RL, param_names(I).p, param_names(I).transform, I) % this
```

```
There is no protected exceedance probability as cbm_hbi_null has not been executed
Plotting exceedance probability instead...
```





4) Check the fitted parameters

Note that the parameters are normalized before saving, and not to be back-transformed to make sense in the model.

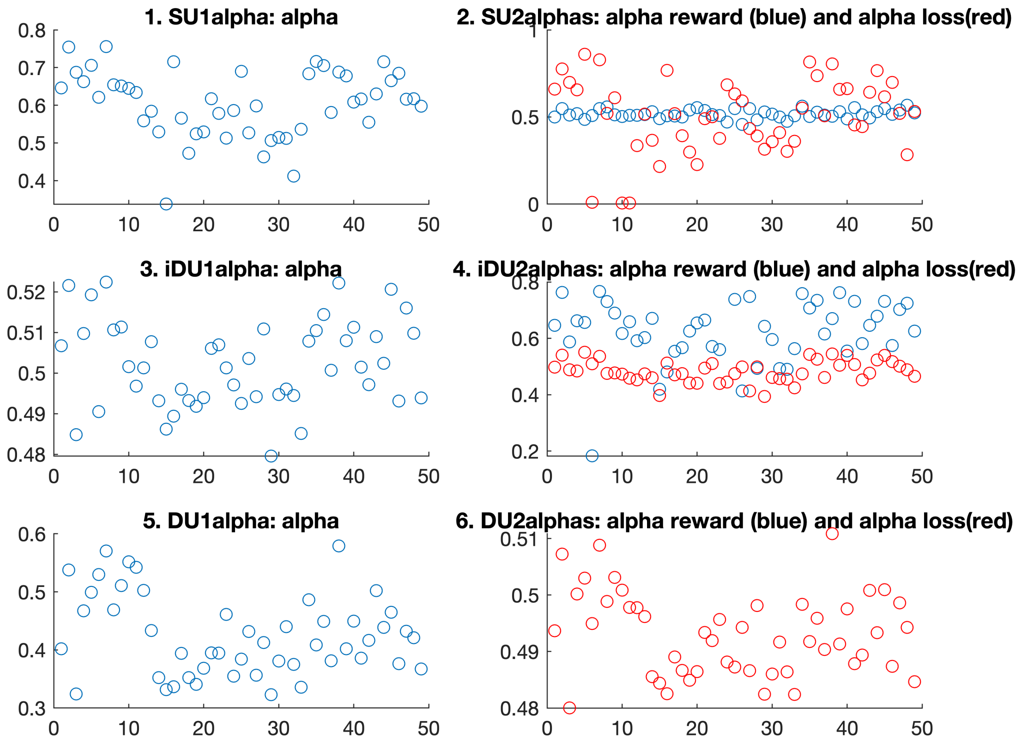
params contains all fitted parameters (subjects, parameters, model_no):

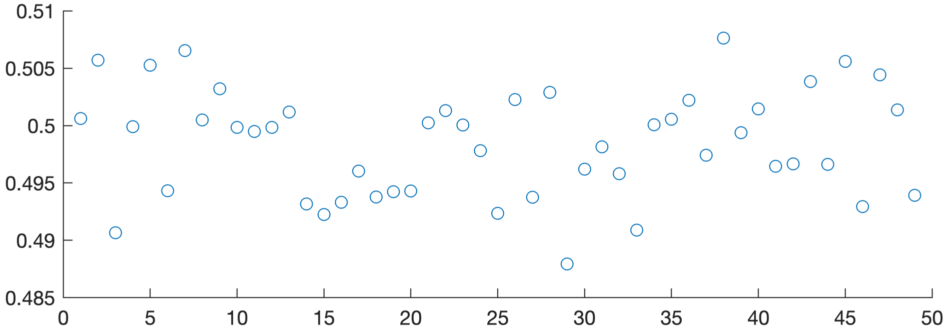
1st column: alpha_both   2nd alpha_reward   3rd alpha_loss   4th kappa   5th beta

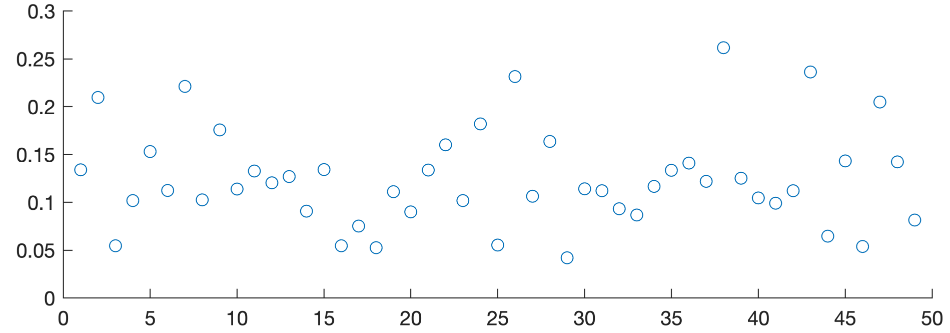
You can pick between the parameters from the individually fitted models and the ones from hierarchical Bayesian inference with "whichfit"

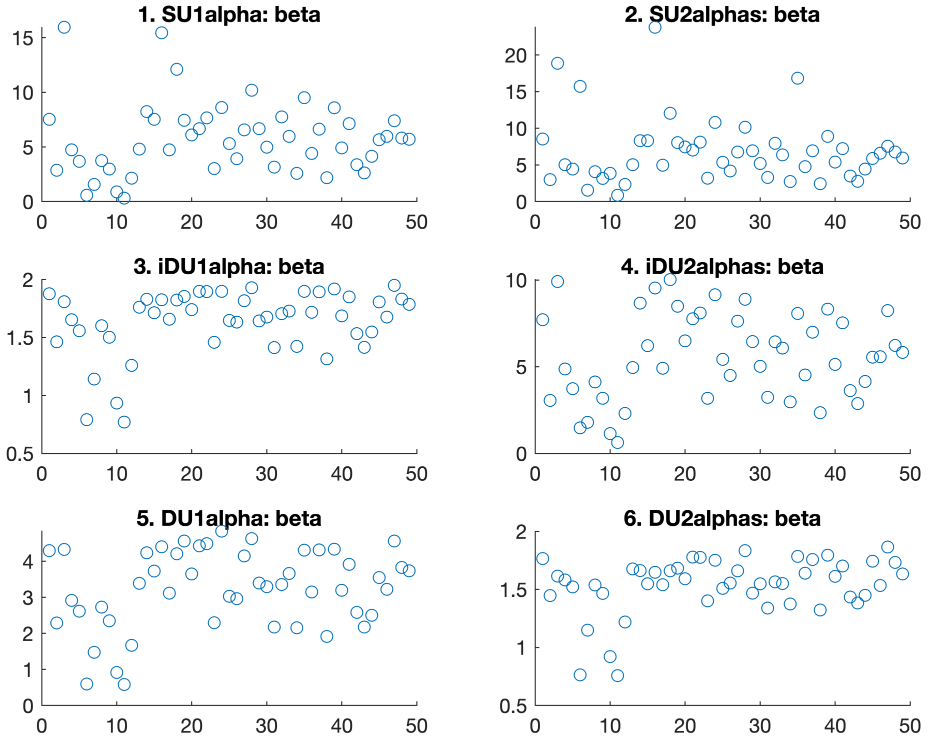
```
params = NaN(nsubjects, 6, nmodels);
whichfit = 2; % for cbm, 2 for hbi

if whichfit == 2
     fname_hbi = load('hbi_RL_fullmodelspace.mat');
     cbm = fname_hbi.cbm;
end

for i = 1:nmodels
if whichfit == 1
    model_name = load(fname_RL{i});
    cbm = model_name.cbm;
    nd_params = cbm.output.parameters;
else
    nd_params = cbm.output.parameters{i};
    params(:,6,i) = cbm.output.responsibility(:,i);
end

if mod(i,2)
params(:,1,i)= 1./(1+exp(-nd_params(:,1)));  % alpha
    if (i == 1 | i == 5)
```

```
params(:,5,i)= exp(nd_params(:,2));     % beta
    elseif i==3
params(:,4,i)= 1./(1+exp(-nd_params(:,2))); % kappa
params(:,5,i)= exp(nd_params(:,3));        % beta
    end
elseif ~mod(i,2)
params(:,2:3,i)= 1./(1+exp(-nd_params(:,1:2))); % alpha_win, alpha_loss
    if (i == 2 | i == 6)
params(:,5,i)= exp(nd_params(:,3));     % beta
    elseif i==4
params(:,4,i)= 1./(1+exp(-nd_params(:,3))); % kappa
params(:,5,i)= exp(nd_params(:,4));        % beta
    end
end
clear nd_params
end

% % Plot parameters

for i = 1:3     %number of max free parameters
figure; hold on;
if i == 1       % alpha
for j = 1:nmodels
    subplot(3,2,j);
    if mod(j,2)
    scatter((1:nsubjects),params(:,1,j));
    else
    scatter((1:nsubjects),params(:,2,j)); hold on;
    scatter((1:nsubjects),params(:,3,j), 'r'); hold off;
    end
    if j==1
    title('1. SU1alpha: alpha');
    elseif j==2
    title('2. SU2alphas: alpha reward (blue) and alpha loss(red)');
    elseif j==3
    title('3. iDU1alpha: alpha');
    elseif j==4
    title('4. iDU2alphas: alpha reward (blue) and alpha loss(red)');
    elseif j==5
    title('5. DU1alpha: alpha');
    elseif j==6
    title('6. DU2alphas: alpha reward (blue) and alpha loss(red)');
    end

end
elseif i ==2 % kappa
for j = 1:nmodels
    if j==3
    subplot(2,1,1)
    title('3. iDU1alpha: kappa');
    scatter((1:nsubjects),params(:,4,j));
    elseif j==4
    subplot(2,1,2)
    title('4. iDU2alphas: kappa');
    scatter((1:nsubjects), params(:,4,j));
    end
end

elseif i ==3 % beta
    for j = 1:nmodels
    subplot(3,2,j);
        scatter((1:nsubjects), params(:,5,j));
        if j==1
        title('1. SU1alpha: beta');
        elseif j==2
        title('2. SU2alphas: beta');
        elseif j==3
        title('3. iDU1alpha: beta');
        elseif j==4
        title('4. iDU2alphas: beta');
        elseif j==5
        title('5. DU1alpha: beta');
        elseif j==6
        title('6. DU2alphas: beta');
        end
    end
end
    hold off;
end
```



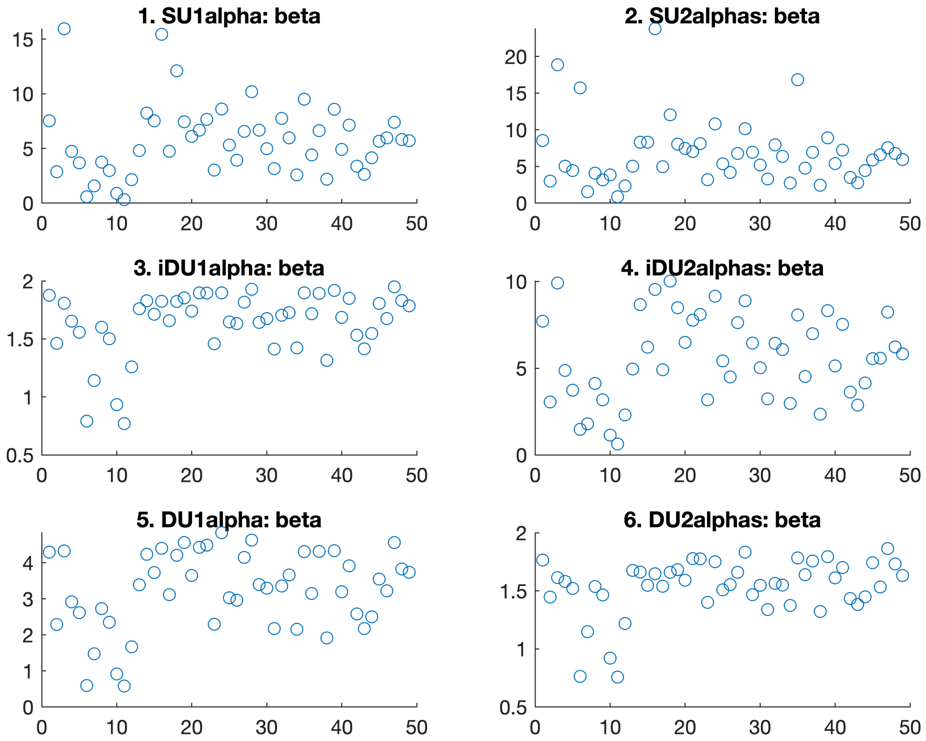




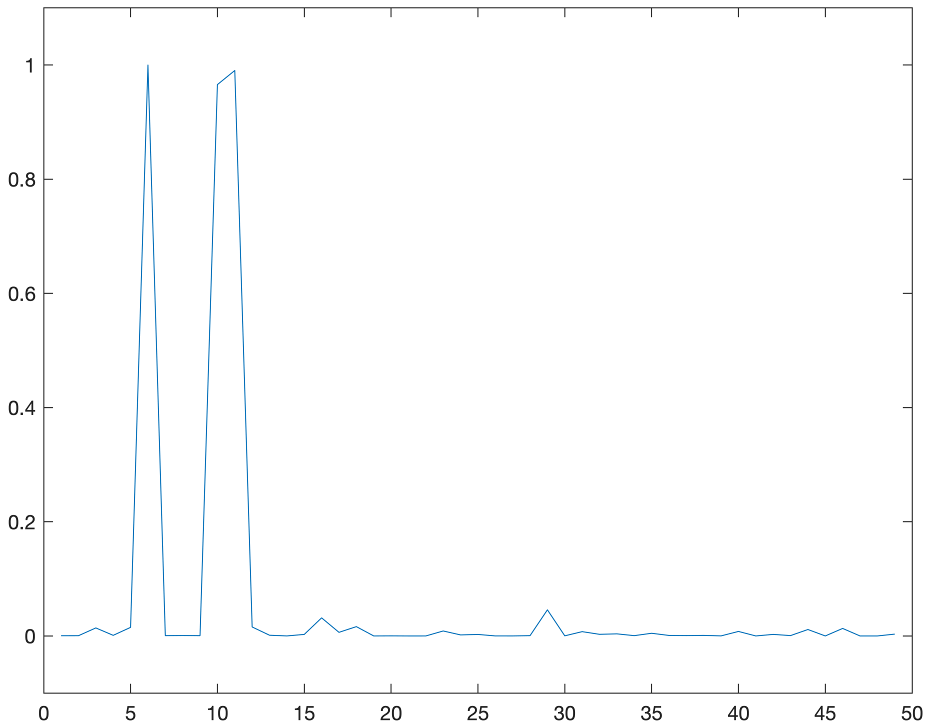


5. Compare group parameters and model comparison

```
hold off; figure
```



```
parameters_RL = cbm.output.parameters{1};
parameters_dualRL = cbm.output.parameters{2};
responsibility = cbm.output.responsibility;
plot(responsibility(:,2)); ylim([-.1 1.1]);
```



```
% estimate protected exceedance probabilities (taking into account the null hypothesis
fname_hbi = 'hbi_RL_fullmodelspace.mat'; % file-address of the file saved by cbm_hbi
cbm_hbi_null(data,fname_hbi); % updates the upper file now containing  the pxps
```

```
cbm_hbi_hbi                                      03-Mar-2020 14:52:23
Running hierarchical bayesian inference (HBI)- null mode...

HBI has been initialized according to
 SU1al.mat [for model 1]
 SU2al.mat [for model 2]
 iDU1al.mat [for model 3]
 iDU2al.mat [for model 4]
 DU1al.mat [for model 5]
 DU2al.mat [for model 6]

Number of samples: 49
Number of models: 6
======================================================================
Iteration 01
Iteration 02
                                                           dL:   26.12
                                                           dx:    0.26
Iteration 03
                                                           dL:    2.39
                                                           dx:    0.10
Iteration 04
                                                           dL:    0.63
                                                           dx:    0.04
Iteration 05
                                                           dL:    0.22
                                                           dx:    0.02
```

```
Iteration 06
                                                                    dL:     0.08
                                                                    dx:     0.01
                                                                  Converged :]
```

6. Plot model responsibility by patient group

(numbering based on the Leipzig data)

```
fname_hbi = load('hbi_RL_fullmodelspace.mat');
cbm = fname_hbi.cbm;
resp = cbm.output.responsibility;

group = group'; group_idx = str2double(group);

for i = 1:nmodels
HC(i)  = [mean(resp(group_idx==0,i))];
PDI(i) = [mean(resp(group_idx==5,i))];
AUD(i) = [mean(resp(group_idx==10,i))];
Szm(i) = [mean(resp(group_idx==20,i))];
Szu(i) = [mean(resp(group_idx==21,i))];
BED(i) = [mean(resp(group_idx==30,i))];
OCD(i) = [mean(resp(group_idx==40,i))];
end

figure
for i = 1:nmodels
subplot(3,2,i)
scatter(1:40,resp(group_idx==20,i))
end



%
% figure
% subplot(2,2,1)
% bar(HC)
% title('HC N=89')
% subplot(2,2,2)
% bar(AUD)
% title('AUD N=42')
% subplot(2,2,3)
% bar(BED)
% title('BED N=22')
% subplot(2,2,4)
% bar(OCD)
% title('OCD N=30')
```