Example of performing JOINT model (IMPULSED + transcytolemmal water exchange)

This example shows how to simultaneously fit mean cell size $\bar{d}$ and intracellular water lifetime % τ_{in} based on simulated data.

This script calls mati.JOINT and sample simulated data in the folder 'Data/simulation'

Reference

The main reference that should be cited when using the code in this script is

1. Jlang J, et al. Simultaneous Quantification of Transcytolemmal Water Exchange and Mean Cell Size Using Temporal Diffusion Spectroscopy. (under review)

Comments or questions?

Please send your comments or questions to Xiaoyu Jiang (xiaoyu.jiang@vumc.org) or Junzhong (JZ) Xu (junzhong.xu@vanderbilt.edu)

Contents

Preliminary

```
clear variables

% Set data folder and general parameters
file_dir = fullfile('Data','simulation') ;
load(fullfile(file_dir,'Pars_sim.mat')) ; % load arrays of diameter and permeability

% Choose ground-truth parameters of interest
snr = 0 ;               % set SNR level (0: no noise)
diameter = 10 ;     % cell size diameter, ranging from 10 to 20 um
tau = 50 ;              % intracellular water lifetime, ranging from 50 ms to inf
Pm = 1/(tau*6/diameter-diameter/10) ;       % calculate membrane permeability based on tau and diameter

% Load data
try
    load(fullfile(file_dir, sprintf('d=%02d_tau=%d.mat',diameter, tau)),'signal','pulse','structure') ;
catch
    sprintf('no such data file...please check the values of diameter and tau')
end

% Select a subset of dMRI data
index = find(pulse.tdiff~=50 & pulse.tdiff~=70) ;     % excluding data with diffusion time tdiff of 50ms and 70ms
pulse = pulse(pulse.tdiff~=50 & pulse.tdiff~=70) ;
signal = signal(index);

% Select a specific JOINT model
structure.modelName = 'joint_vin_d_Dex_Din_kin' ; % set the fitting model
structure.geometry = 'sphere' ;

% Create a JOINT object
joint = mati.JOINT(structure, pulse) ;

% Create an ImageData object
signal = mati.Physics.AddRicianNoise(signal,snr) ;
img(1,1,1,1:length(index))=signal;
data = mati.ImageData(img,snr) ;
```

Fit dMRI signals using a JOINT model

Create a FitPars object

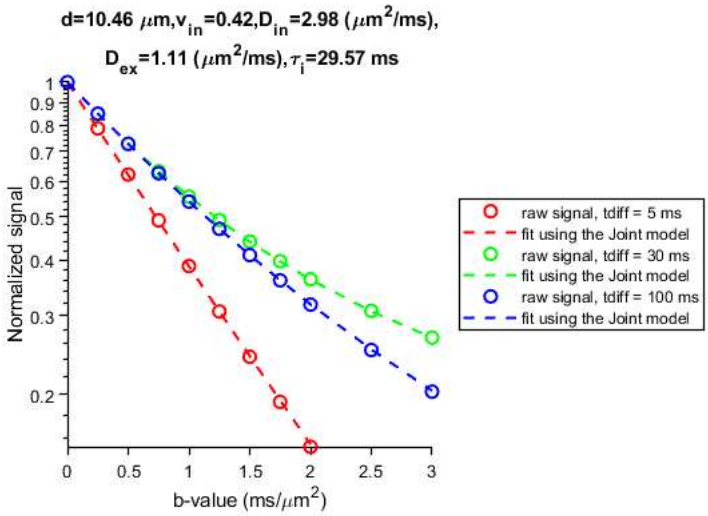
```
fitopts.solverName = 'fmincon';         % {'lsqcurvefit' , 'lsqnonlin' , 'fmincon'}
fitopts.options = optimoptions(fitopts.solverName,'Display','off') ;
if snr~=0
    fitopts.noiseModel = 'none' ; %{'none','standard';'logLikelihood'}
else
    fitopts.noiseModel = 'none' ;
end
fitopts.flag.multistart = 'y' ; fitopts.flag.parfor = 'y' ; fitopts.flag.deivim = 'n' ;
fitopts.NumStarts = 5 ;
fitpars = mati.FitPars(joint, fitopts) ;

% Fit the model to data
fitout = fitpars.Fit(data) ;
```

Display the fitting results

```
tdiff=unique(joint.pulse.tdiff);
nf=length(tdiff);
color='rgbkmcy';
```

```
figure(1);
signal_fit = joint.FcnSignal({fitout.vin, fitout.d, fitout.Dex, fitout.Din, fitout.kin}, joint) ;
for i=1:nf
    index=find(joint.pulse.tdiff==tdiff(i));
    plot(joint.pulse.b(index),signal(index),[color(i) 'o'],'linewidth',1.5,'DisplayName',['raw signal, tdiff = ' num2str(tdiff(i)) ' ms']);
    hold on;plot(joint.pulse.b(index),signal_fit(index),[color(i) '--'],'linewidth',1.5,'DisplayName','fit using the Joint model');
end
% title(sprintf('d=%0.2f \\mum,v_i_n=%0.2f,\n D_i_n=%0.2f (\\mum^2/ms),\n D_e_x=%0.2f (\\mum^2/ms),\n \\tau_i=%0.2f ms',fitout.d,fitout.vin,fitout.Din,fitout.Dex,1/
title(sprintf('d=%0.2f \\mum,v_i_n=%0.2f,D_i_n=%0.2f (\\mum^2/ms),\n D_e_x=%0.2f (\\mum^2/ms),\\tau_i=%0.2f ms',fitout.d,fitout.vin,fitout.Din,fitout.Dex,1/fitout.k
set(gca, 'Box', 'off', 'TickDir', 'out', 'TickLength', [.02 .02],'LineWidth', 1);
set(gca,'FontName','Aria','FontSize',10);pbaspect([1 1 1]);
set(gca, 'YScale', 'log');
legend('location','EastOutside');
xlabel('b-value (ms/\mum^2)');
ylabel('Normalized signal')
hold off;
```



Published with MATLAB® R2020b