DeerLab: A comprehensive toolbox for analyzing dipolar EPR spectroscopy data

Luis Fábregas Ibáñez[1], Gunnar Jeschke[1], Stefan Stoll[2]

[1]ETH Zurich, Laboratory of Physical Chemistry, Vladimir-Prelog-Weg 2, 8093 Zurich, Switzerland
[2]University of Washington, Department of Chemistry, Seattle, WA 98195, USA

Supporting Information

1 DeerLab scripts

All DeerLab scripts used for the figures in the main text are provided here. These were written using DeerLab (release 0.9.0) in MATLAB R2019b.

Script 1: `figure3_regularization.m` - DeerLab script employed for generating Figure 2 of the main text.

```
%---------------------------------------------------------------------------
% Figure 2 - Regularization approaches
%---------------------------------------------------------------------------
% This DeerLab script was used to make Fig.2 of the paper:
%
% Luis Fabregas Ibanez, Gunnar Jeschke, Stefan Stoll
% DeerLab: A comprehensive toolbox for analyzing dipolar EPR spectroscopy data
% Magnetic Resonance, 2020
%
% DeerLab version:
%   Release: 0.9.0
%---------------------------------------------------------------------------

clear,clc,clf

% Figure 2a
%----------------------------------------------------------------------

%Fix random numbers for reproduceable results
rng(0)

%Generate ground truth distance distribution
[rfine,P] = groundtruth('figure2');

%Define a shorter distance axis for the fit
r = linspace(2.5,5,300);


%Simulate dipolar signal
t = linspace(-0.5,4,500);
lambda = 0.2;
V = dipolarsignal(t,rfine,P,lambda,'noiselevel',0.01);

%Prepare dipolar kernel
K = dipolarkernel(t,r,lambda);

%Determine regularization parameter for to avoid computing twice
alpha = selregparam(V,K,'tikh','aic');

%Get Tikhonov fit
Ptikh = fitregmodel(V,K,r,'tikh',alpha);
Vtikh = K*Ptikh;

%Get TV fit
Ptv = fitregmodel(V,K,r,'tv','aic');
Vtv = K*Ptv;

%Get OBIR fit
Pobir = obir(V,K,r,'tikh',alpha,'NoiseLevelAim',0.01);
Vfit = K*Pobir;

%Plot results

subplot(221)
plot(t,V,'k.',t,Vfit,t,Vtikh,t,Vtv,'LineWidth',1.5)
axis tight
set(gca,'fontsize',13)
xlabel('Time [\mus]'),ylabel('V(t) [\mus^{-1}]')

subplot(222)
plot(rfine,P,'k',r,Pobir,r,Ptikh,'-',r,Ptv,'-','LineWidth',1.5)
axis tight
xlim([3 4.5])
set(gca,'fontsize',13)
xlabel('Distance [nm]'),ylabel('P(r) [nm^{-1}]')
legend('truth','OBIR','Tikh','TV')

% Figure 2b
%----------------------------------------------------------------------
```

```matlab

rng(9)

%Simulate dipolar signal
t = linspace(-0.5,4,600);
lambda = 0.2;
V = dipolarsignal(t,rfine,P,lambda,'noiselevel',0.025);

%Prepare dipolar kernel
K = dipolarkernel(t,r,lambda);

%Determine regularization parameter for to avoid computing twice
alpha = selregparam(V,K,'tikh','aic');

%Get Tikhonov fit
Ptikh = fitregmodel(V,K,r,'tikh',alpha);
Vtikh = K*Ptikh;

%Get TV fit
Ptv = fitregmodel(V,K,r,'tv','aic');
Vtv = K*Ptv;

%Get OBIR fit
Pobir = obir(V,K,r,'tikh',alpha,'NoiseLevelAim',0.025);
Vfit = K*Pobir;

%Plot results

subplot(223)
plot(t,V,'k.',t,Vfit,t,Vtikh,t,Vtv,'LineWidth',1.5)
axis tight
set(gca,'fontsize',13)
xlabel('Time [\mus]'),ylabel('V(t) [\mus^{-1}]')

subplot(224)
plot(rfine,P,'k',r,Pobir,r,Ptikh,'-',r,Ptv,'-','LineWidth',1.5)
axis tight
xlim([3 4.5])
set(gca,'fontsize',13)
xlabel('Distance [nm]'),ylabel('P(r) [nm^{-1}]')
legend('truth','OBIR','Tikh','TV')
```

Script 2: **figure5_multigauss.m** - DeerLab script employed for generating Figure 3 of the main text.

```matlab
%--------------------------------------------------------------------------
% Figure 3 - Multi-Gauss model fitting
%--------------------------------------------------------------------------
% This DeerLab script was used to make Fig.3 of the paper:
%
% Luis Fabregas Ibanez, Gunnar Jeschke, Stefan Stoll
% DeerLab: A comprehensive toolbox for analyzing dipolar EPR spectroscopy data
% Magnetic Resonance, 2020
%
% DeerLab version:
%   Release: 0.9.0
%--------------------------------------------------------------------------

clear,clc

rng(1)

% Generate data
%-------------------------
t = linspace(-0.25,4,300); % time axis, us
lam = 0.35;
B = bg_exp(t,0.08);

[r,P] = groundtruth('figure3');


% Generate dipolar signal with noise
V = dipolarsignal(t,r,P,lam,B) + whitegaussnoise(t,0.01,'rescale');

% Run multi-Gauss fitting
%-------------------------
NGauss = 5; % maximum number of Gaussians
```

```
[Pfit,param,Pci,paramci,Nopt,metrics,Peval] = fitmultimodel(V,t,r,@dd_gauss,NGauss,'aic','...
    background',@bg_exp,'confidencelevel',0.95,'multistart',1);
K = dipolarkernel(t,r,param(end-1),bg_exp(t,param(end)));

Vfit = K*Pfit;

fprintf('The optimal number of Gaussians is: %i \n',Nopt)

% Plot results
%-------------------------
figure(8),clf

figure(1),clf
hold on
plot(t,V,'k.','LineWidth',1)
plot(t,Vfit,'b','LineWidth',1.5)
plot(t,(1-param(end-1))*bg_exp(t,param(end)),'r','LineWidth',1.5)
box on,legend('model','fit')
xlabel('time (\mus)'),ylabel('S(t)')
axis tight
set(gca,'fontsize',14)

figure(2),clf
hold on
plot(r,P,'k','LineWidth',1.5)
plot(r,Pfit,'b','LineWidth',1.5)
fill([r fliplr(r)], [Pci(:,1); flipud(Pci(:,2))],'b','Linestyle','none','facealpha',0.25)
box on, axis tight
legend('model','optimal fit')
xlabel('distance (nm)'), ylabel('P(r)')
set(gca,'fontsize',14)
% xlim([2.5 4.5])

figure(3),clf
hold on
ax = 1:length(metrics);
w = exp(-(metrics - min(metrics))/2)/sum(exp(-(metrics - min(metrics))/2))
plot(ax,metrics - min(metrics),'-o','LineWidth',1.5)
box on,axis tight
ylabel('AIC')
xlabel('number of Gaussians in model')
set(gca,'fontsize',14)

figure(4),clf
hold on
plot(r,Peval + 2*(1:NGauss).','b-','LineWidth',1.5)
box on,axis tight
set(gca,'ytick',2:2:2*NGauss,'yticklabel',1:NGauss)
xlabel('distance (nm)')
ylabel('#Gaussians in model')
set(gca,'fontsize',14)
% xlim([2.5 4.5])
ylim(2*[0.8 7.5])
```

Script 3: `figure7_bileveloptimization.m` - DeerLab script employed for generating Figure 5 of the main text.

```
%-------------------------------------------------------------------------------
% Figure 5 - One-step vs Two-step Analysis
%-------------------------------------------------------------------------------
% This DeerLab script was used to make Fig.5 of the paper:
%
% Luis Fabregas Ibanez, Gunnar Jeschke, Stefan Stoll
% DeerLab: A comprehensive toolbox for analyzing dipolar EPR spectroscopy data
% Magnetic Resonance, 2020
%
% DeerLab version:
%   Release: 0.9.0
%-------------------------------------------------------------------------------

%Fix random numbers for reproduceable results
rng(1)

% Load example data
[r,P] = groundtruth('figure5');

% Loop over all four cases of truncation
```

```matlab
for i=1:4

    switch i
        case 1
            % Plot 1: truncation 1/4
            t = linspace(-0.5,8,300);
        case 2
            % Plot 2: truncation 2/4
            t = linspace(-0.5,3.2,300);
        case 3
            % Plot 3: truncation 3/4
            t = linspace(-0.5,1.5,300);
        case 4
            % Plot 4: truncation 4/4
            t = linspace(-0.5,0.7,300);
    end

    % Simulate the dipolar signal
    B = bg_exp(t,0.09);
    lam = 0.3;
    V = dipolarsignal(t,r,P,lam,B) + whitegaussnoise(t,0.005,'rescale');

    % Create function handle depending on r and param from the custom model
    fcnhandle = @(t,param) mymodel(t,param,r,V,K);

    % One-step analysis
    %-----------------------------------------------------------------------
    % Fit the background and distribution simultaneously
    parafit = fitparamodel(V,fcnhandle,t,[0.05,0.05],'Lower',[0 0.01],'Upper',[1 1],'TolFun',1e...
    -5);
    [Vfit,Pfit,Bfit,alphaopt] = mymodel(t,parafit,r,V);
    Bfit = (1-parafit(1))*Bfit;

    % Two-step analysis
    %-----------------------------------------------------------------------
    % Fit the background...
    [Bfit2,lam2] = fitbackground(V,t,@bg_exp);
    KB2 = dipolarkernel(t,r,lam2,Bfit2);
    % ...and then the distribution
    Pfit2 = fitregmodel(V,KB2,r,'tikhonov','aic');
    Vfit2 = KB2*Pfit2;

    % Plot the results
    %-----------------------------------------------------------------------
    subplot(3,4,i)
    plot(t,V,'k.',t,Vfit,'b-',t,Bfit,'b--',t,(1-lam)*B,'k--','LineWidth',1.5)
    axis tight
    set(gca,'fontsize',14)
    ylims = ylim;
    xlabel('t [\mus]')
    ylabel('V(t)')

    subplot(3,4,i+4)
    plot(t,V,'k.',t,Vfit2,'r-',t,(1-lam2)*Bfit2,'r--',t,(1-lam)*B,'k--','LineWidth',1.5)
    axis tight
    set(gca,'fontsize',14)
    ylim(ylims)
    xlabel('t [\mus]')
    ylabel('V(t)')

    subplot(3,4,i+8)
    plot(r,P,'k',r,Pfit,'b',r,Pfit2,'r','LineWidth',1.5)
    axis tight
    xlim([3.5 6.5])
    set(gca,'fontsize',14)
    xlabel('r [nm]')
    ylabel('P(r) [nm^{-1}]')

    drawnow
end

% Definition of the parametric dipolar signal model
%----------------------------------------------
function [Vfit,Pfit,Bfit,alphaopt] = mymodel(t,param,r,V)

% Extract parameters from the upper-level
lambda = param(1);
```

```
k = param(2);

% Generate the fitted kernel
Bfit = bg_exp(t,k);
K = dipolarsignal(t,r,lambda,Bfit);

% Find the solution of the lower-level via Tikhonov regularization
[Pfit,alphaopt] = fitregmodel(V,K,r,'tikhonov','aic');

% Get the time-domain fit
Vfit = K*Pfit;

end
```

Script 4: **figure9_multipathway.m** - DeerLab script employed for generating Figure 7 of the main text.

```
%--------------------------------------------------------------------------
% Figure 7(a) - Fitting a 4-pulse DEER signal with "2+1 arifact"
%--------------------------------------------------------------------------
% This DeerLab script was used to make Fig.7a of the paper:
%
% Luis Fabregas Ibanez, Gunnar Jeschke, Stefan Stoll
% DeerLab: A comprehensive toolbox for analyzing dipolar EPR spectroscopy data
% Magnetic Resonance, 2020
%
% DeerLab version:
%   Release: 0.9.0
%--------------------------------------------------------------------------

clear,clc,clf

rng(1)

% Load example data
[r,P] = groundtruth('figure7a');

%Construct time-axis
t = linspace(-0.1,4.5,500);

% Multipathway parameters
Ts2 = max(t) - 0.005;
lambdas = [0.5 0.25 0.1];
pathways = ex_ovl4pdeer(t,[lambdas Ts2]);

% Background parameters
kappa = 0.25;
strfact = 0.93;
Bmodel = @(t,lam) bg_strexp(t,[kappa strfact],lam);

%Genrate signal
K = dipolarkernel(t,r,pathways,Bmodel);
B = dipolarbackground(t,pathways,Bmodel);
V = K*P + whitegaussnoise(t,0.004);

% Fit the 4pDEER signal with the "2+1 artifact"
[Vfit,Pfit,Bfit] = fitsignal(V,t,r,'P',@bg_strexp,@ex_ovl4pdeer);

% Plot
subplot(221)
plot(t,V,'k.',t,Vfit,'r',t,B,'k--',t,Bfit,'r--','LineWidth',1.5)
axis tight, grid on, box on
set(gca,'fontsize',14)
xlabel('t [\mus]'),ylabel('V(t)')
legend('V_{exp}','V_{fit}','B_{truth}','B_{fit}','Location','eastout')

subplot(222)
plot(r,P,'k',r,Pfit,'r','LineWidth',1.5)
axis tight
set(gca,'fontsize',14)
xlabel('r [nm]'),ylabel('P(r) [nm^{-1}]')
legend('P_{truth}','P_{fit}','Location','eastout')
xlim([1.5 5])

%--------------------------------------------------------------------------
% Figure 7(b) - Fitting a 5-pulse DEER signal
%--------------------------------------------------------------------------
```

```
% This DeerLab script was used to make Fig.7b of the paper:
%
% Luis Fabregas Ibanez, Gunnar Jeschke, Stefan Stoll
% DeerLab: A comprehensive toolbox for analyzing dipolar EPR spectroscopy data
% Magnetic Resonance, 2020
%
% DeerLab version:
%   Release: 0.9.0
%-------------------------------------------------------------------------------

% Fix random numbers for reproduceable results
rng(1)

% Load example data
[r,P] = groundtruth('figure7b');

% Define time-axis
t = linspace(-0.1,9,500);

% Multipathway parameters
Ts2 = 4.6;
lambdas = [0.65 0.75 0.25];
pathways = ex_5pdeer(t,[lambdas Ts2]);

% Background parameters
kappa = 0.1;
strfact = 0.88;
Bmodel = @(t,lam) bg_strexp(t,[kappa strfact],lam);

% Simulate the 5pDEER dipolar signal
K = dipolarkernel(t,r,pathways,Bmodel);
B = dipolarbackground(t,pathways,Bmodel);
V = K*P + whitegaussnoise(t,0.005);

% Fit the 5pDEER signal
[Vfit,Pfit,Bfit] = fitsignal(V,t,r,'P',@bg_strexp,@ex_5pdeer);

% Plot
subplot(223)
plot(t,V,'k.',t,Vfit,'b',t,B,'k--',t,Bfit,'b--','LineWidth',1.5)
axis tight,box on, grid on
xlabel('t [\mus]'),ylabel('V(t)')
set(gca,'fontsize',14)
legend('V_{exp}','V_{fit}','B_{truth}','B_{fit}','Location','eastout')

subplot(224)
plot(r,P,'k',r,Pfit,'b','LineWidth',1.5)
axis tight,box on, grid on
xlabel('r [nm]'),ylabel('P(r) [nm^{-1}]')
set(gca,'fontsize',14)
xlim([2.5 5])
legend('P_{truth}','P_{fit}','Location','eastout')
```

Script 5: **figure10_globalfit.m** - DeerLab script employed for generating Figure 8 of the main text.

```
%-------------------------------------------------------------------------------
% Figure 8 - Global fit of signals from different experiments
%-------------------------------------------------------------------------------
% This DeerLab script was used to make Fig.8 of the paper:
%
% Luis Fabregas Ibanez, Gunnar Jeschke, Stefan Stoll
% DeerLab: A comprehensive toolbox for analyzing dipolar EPR spectroscopy data
% Magnetic Resonance, 2020
%
% DeerLab version:
%   Release: 0.9.0
%-------------------------------------------------------------------------------

clc,clear,clf

% Generate ground truth distance distribution
[r,P] = groundtruth('figure8');

% Define background model
kappa = 0.25;
Bmodel = @(t,lam)bg_exp(t,kappa,lam);
```

```

% Simulate a 4pDEER signal  (main pathway at start of signal, secondary pathway at the end)
rng(1)
t4p = linspace(-0.3,3.2,300);
pathways(1,:) = [0.8 NaN];
pathways(2,:) = [0.2 0];
V4p = dipolarkernel(t4p,r,pathways,Bmodel)*P + whitegaussnoise(t4p,0.01);

% Simulate a 5pDEER signal (main pathway at start of signal, secondary pathway at the middle)
rng(2)
t5p = linspace(-0.3,6.5,300);
pathways(1,:) = [0.4 NaN];
pathways(2,:) = [0.4 0];
pathways(3,:) = [0.2 max(t5p)/2];
V5p = dipolarkernel(t5p,r,pathways,Bmodel)*P  + whitegaussnoise(t5p,0.01);

% Collect signals for global-fit
Vs = {V4p,V5p};
ts = {t4p,t5p};

% Run global fit
[Vfit,Pfit] = fitsignal({V4p,V5p},{t4p,t5p},r,'P',@bg_exp,{@ex_4pdeer,@ex_5pdeer});
V4pfit = Vfit{1};
V5pfit = Vfit{2};

% Plot results
clf
subplot(121),hold on
plot(t4p,V4p,'k.',t4p,V4pfit)
plot(t5p,V5p + 0.5,'k.',t5p,V5pfit + 0.5)
axis tight, grid on, box on
xlabel('t [\mus]'),ylabel('V(t)')
legend('4pDEER data','4pDEER fit','5pDEER data','5pDEER fit','Location','best')

subplot(122)
plot(r,P,'k',r,Pfit,'b')
axis tight, grid on, box on
xlabel('r [nm]'),ylabel('P(r) [nm^{-1}]')
legend('truth','global fit')
```

Script 6: `figure11_modelfree_titration.m` - DeerLab script employed for generating Figure 9 of the main text.

```
%-------------------------------------------------------------------------
% Figure 11 - Titration curve with parameter-free distributions
%-------------------------------------------------------------------------
% This DeerLab script was used to make Fig.11 of the paper:
%
% Luis Fabregas Ibanez, Gunnar Jeschke, Stefan Stoll
% DeerLab: A comprehensive toolbox for analyzing dipolar EPR spectroscopy data
% Magnetic Resonance, 2020
%
% DeerLab version:
%   Release: 0.8.beta
%   Commit: e875bbc3ff613cf26896d9aa42ecea64069b6656
%-------------------------------------------------------------------------

clc, clear,clf

%Fix random numbers for reproduceable results
rng(2)

%Prepare equilibrium of type:
%   A + L <-> B
KD = 5.65;  % dissociation constant, uM
Ctot = 1; % total protein concentration, uM
L = [0.3 1 3 10 30 100 300]; % total ligand concentration, uM
nDataSets = numel(L);

% Calculate molar fractions
%    a (for A = protein without ligand)
%    b (for B = protein with bound protein)
Kb = 1/KD;
for q = 1:nDataSets
    xbound_ = roots([Kb*Ctot -(Kb*L(q) + Kb*Ctot + 1) Kb*L(q)]);
    xbound(q) = xbound_(xbound_<=1 & xbound_>=0);
```

```
end

%Generate ground truth distance distributions for each fraction
[rtruth,P1] = groundtruth('figure10_1');
[¬,P2] = groundtruth('figure10_2');

% Generate different dipolar time-axes
tmax = 1 + randi(6,nDataSets,1);
for i=1:length(tmax)
    ts{i} = linspace(-0.1,tmax(i),300);
end

% Generate different background decays, noise levels, and modulation depths
% for each dataset
ks = linspace(0.2,1.15,nDataSets)*0.25;
ks = ks(randperm(nDataSets));
noiselevels = 0.005 + 0.02*rand(nDataSets,1);
lams = linspace(0.75,1.15,nDataSets)*0.45;
lams = lams(randperm(nDataSets));

%Generate set of dipolar signals
for i = 1:nDataSets

    % Prepare the sum distance distribution from corresponding molar fractions
    P = xbound(i)*P1  + (1 - xbound(i))*P2;
    Ps{i} = P;
    %Prepare corresponding dipolar kernel
    K = dipolarkernel(ts{i},rtruth,lams(i),bg_exp(ts{i},ks(i)));
    %Simulate dipolar signal
    Vs{i} = K*Ps{i} + whitegaussnoise(ts{i},noiselevels(i),'rescale');

end

%Define the fit distance axis
r = linspace(2,6,300);

% Define start values & boundaries for fit parameters
% param0 =  [b ks lams];
param0 =  [linspace(0,1,nDataSets) 0.25*ones(1,nDataSets) 0.25*ones(1,nDataSets)];
lower  =  [zeros(1,nDataSets) 0*ones(1,nDataSets) 0*ones(1,nDataSets)];
upper  =  [ones(1,nDataSets) 1*ones(1,nDataSets) 1*ones(1,nDataSets)];

% Gobal fit of the parametric time-domain model with parameter-free distributions
parfit = fitparamodel(Vs,@(t,p,idx)mymodel(t,p,idx,ts,r,Vs,Ps,rtruth,nDataSets),ts,param0,'...
    Lower',lower,'Upper',upper,'tolfun',1e-8,'globalweights',ones(nDataSets,1));

%Get the fitted molar fractions
xbound_fit = parfit(1:nDataSets);
xunbound_fit = 1 - xbound_fit;

%Control label switching
if mean(diff(xbound_fit))<0
    tmp = xunbound_fit;
    xunbound_fit = xbound_fit;
    xbound_fit = tmp;
end

% Re-calculate time and distance-domain fits
for i=1:nDataSets
    [Vfit{i},Pfit{i},B{i}] = mymodel(ts{i},parfit,i,ts,r,Vs,Ps,rtruth,nDataSets);
    B{i} = (1 - parfit(2*nDataSets + i))*B{i};
end

%Extrapolate the molar fraction values for plotting
Lfine = logspace(min(log10(L)),max(log10(L)),50);
for q = 1:numel(Lfine)
    p = [Kb*Ctot -(Kb*Lfine(q)+Kb*Ctot+1) Kb*Lfine(q)];
    xbound_ = roots(p);
    xbound(q) = xbound_(xbound_≤1 & xbound_≥0);
end
xunbound = 1-xbound;

%Plot input data & time-domain fits
figure(5),clf,hold on
for j=1:length(ts)
    i = length(ts)+1-j;
    plot(ts{i},Vs{i}+0.5*(j-1),'k.',ts{i},Vfit{i}+0.5*(j-1),'r',ts{i},B{i}+0.5*(j-1),'b--','...
     LineWidth',1.5,'MarkerSize',8)
    text(4,0.5*j,sprintf('%.2f \\muM',L(i)))
end
axis tight,box on
xlabel('Time [\mus]'),ylabel('V(t)')
set(gca,'FontSize',15,'yticklabel',[])

%Plot ground truth distributions & distance-domain fits
r1 = r(r<8);
r2 = r(r≥1);
rjoined = [r1 r2];
figure(4),clf
for i=1:length(Pfit)
    subplot(1,8,i)
    tmp = Pfit{i};
    P1 = tmp(1:numel(r1));
    P2 = tmp(numel(r1)+1:end);
    plot(r1,P1/mean(diff(r2)),'r',r2,P2/mean(diff(r2)),'b',rtruth,Ps{i}/sum(Ps{i})/mean(diff(...
     rtruth)),r1,P2/mean(diff(r2))+P1/mean(diff(r2)),'LineWidth',1.5)
    axis tight,box on
    xlim([2 6])
    title(sprintf('c_{ligand}=%.2f \\muM',L(i)))
end
legend('bound','unbound','truth','sum')
xlabel('distance [nm]'),ylabel('P(r)')
set(gca,'FontSize',15,'yticklabel',[])

%Plot titration curve with ground truth and fitted points (log scale)
figure(7),clf
plot(log10(L),xbound_fit,'r.',log10(L),xunbound_fit,'b.',log10(Lfine),xbound,'r',log10(Lfine),...
     xunbound,'b','LineWidth',1.5,'MarkerSize',25)
axis tight,box on
xlabel('log(Ligand conc.)'),ylabel('Molar fraction')
set(gca,'fontsize',15,'ytick',0:0.2:1,'yticklabel',0:0.2:1)
legend('fit x_{bound}','fit x_{unbound}','truth x_{bound}','truth x_{unbound}')
ylim([0 1])
ytickformat('%.1f')

%Function definition of the global fit with parameter-free distributions
function [Vfit,Pfit,B] = mymodel(t,p,idx,ts,r,Vs,Ps,rtrue,nDataSets)

persistent Ks Pjoined
% Make the dipolar kernels and fitted distributions as persistent variables
% since they do not need to be calculated for each signal again (since the
% fit is global). Speed enhancement = number of globally fitted signals

% Extract parameters
k = p(nDataSets+1:2*nDataSets);
lam = p(2*nDataSets+1:end);

% Define fit axis for first component...
r1 = r(r < 8);
% ...and second component..
r2 = r(r ≥ 1);
% ...and concatenate them into a single vector
rjoined = [r1 r2];

% Run this section only for the first signal, skip for the rest
if idx==1

    % Loop over all signals being globally fitted
    for i=1:numel(Vs)

        % Prepare the dipolar kernels for both components
        K1 = dipolarkernel(ts{i},r1);
        K2 = dipolarkernel(ts{i},r2);
        % Concatenate them in the distance-dimension
        Kjoined = [K1 K2];

        % Normalize the time-dimension
        [¬,t0] = min(abs(ts{i}));
        Kjoined = Kjoined./Kjoined(t0,:);

        % Introduce modulation depth and background information
        Kjoined = (1-lam(i) + lam(i)*Kjoined).*bg_exp(ts{i},k(i));

        % Weight the distance-dimension according to the molar fractions
        amps = [ones(1,numel(r1))*p(i) ones(1,numel(r1))*(1 - p(i))];
        Kjoined = amps.*Kjoined;

        % Save the joined dipolar kernel for the current signal
        Ks{i} = Kjoined;
    end
    % Global Tikhonov regularization using the joined dipolar kernels
    rng(2)
    alpha = 25;
    Pjoined = fitregmodel(Vs,Ks,rjoined,'tikh',alpha);
    % Normalize integral
    Pjoined = Pjoined/sum(Pjoined);
end

% Extract the fits for each of the components from the joined vector
Pfit2 = Pjoined(end-numel(r2)+1:end);
Pfit1 = Pjoined(1:numel(r1));
% Normalize their integrals
Pfit1 = Pfit1/sum(Pfit1);
Pfit2 = Pfit2/sum(Pfit2);

% Compute the time-domain fit of the current dipolar signal
Vfit = Ks{idx}*[Pfit1; Pfit2];

% Return the fits scaled by their respective molar fractions
Pfit = [p(idx)*Pfit1; (1-p(idx))*Pfit2];

% Compute the fitted background to return as output
B = bg_exp(ts{idx},k(idx));

% Plot intermediate results
subplot(121),hold on
if idx==1
    cla
end
plot(t,Vs{idx}+idx/2,'k.',t,Vfit+idx/2,'r')
ylim([0.5 5.5])
axis tight, grid on, box on
xlabel('t [\mus]')
ylabel('V(t)')
legend('data','fit')
subplot(122)
Ptruth = Ps{idx}/sum(Ps{idx})/mean(diff(rtrue));
Pfrac1 = p(idx)*Pfit1/mean(diff(r1));
Pfrac2 = (1-p(idx))*Pfit2/mean(diff(r2));
plot(rtrue,Ptruth,r1,Pfrac1,r2,Pfrac2)
axis tight, grid on, box on
xlabel('r [nm]')
ylabel('P(r) [nm^{-1}]')
legend('truth (sum)','fraction 1','fraction 2')
drawnow

end
```

Script 7: `figure12_sensitivityanalysis.m` - DeerLab script employed for generating Figure 11 of the main text.

```
%--------------------------------------------------------------------------
% Figure 11 - Uncertainty estimation methods
%--------------------------------------------------------------------------
% This DeerLab script was used to make Fig.11 of the paper:
%
% Luis Fabregas Ibanez, Gunnar Jeschke, Stefan Stoll
% DeerLab: A comprehensive toolbox for analyzing dipolar EPR spectroscopy data
% Magnetic Resonance, 2020
%
% DeerLab version:
%   Release: 0.9.0
%--------------------------------------------------------------------------

clear,clc,clf

% Fix random numbers for reproduceable results
rng(1)

% Load data for the examples
```

```matlab
[r,P] = groundtruth('figure11');

% Construct time-axis
t = linspace(-0.2,4,300);
% Background parameters
kappa = 0.25;
lambda = 0.35;
B = bg_exp(t,lambda*kappa);
% Construct dipolar kernel
KB = dipolarkernel(t,r,lambda,B);
% Noise standard deviation
sig = 0.02;
% Generate dipolar signal
V = KB*P + whitegaussnoise(t,sig);

% Fit the signal
[Vfit,Pfit,Bfit,parfit,modelci,paramci,stat] = fitsignal(V,t,r,'P',@bg_exp,@ex_4pdeer,[],'...
    RegParam',0.5);

%Get confidence intervals on Pfit
Pci = modelci.Pfit;

%Plot signal and fit
subplot(231)
hold on
plot(t,V,'k.',t,Vfit,'m','Linewidth',1.5)
axis tight, box on
xlabel('t [\mus]'),ylabel('V(t)')
legend('V_{exp}','V_{fit}','Location','eastout')
text(2.5,0.9,sprintf('\\chi^2 = %.3f',stat.chi2red))
paramci = [paramci.ex; paramci.bg];
parfit = [parfit.ex; parfit.bg];

% Generate covariance-based distributions of the fit parameters
%-------------------------------------------------------------------

% Student's t-distribution critical value (95%)
z = 1.968;
paramsig = abs(paramci(:,1) - parfit)/z;

%Gaussian distribution for modulation depth
parVals1 = linspace(0.15,0.55,300);
P1 = exp(-(parfit(1) - parVals1).^2./(2*paramsig(1)^2));
%Gaussian distribution for decay rate
parVals2 = linspace(0.05,0.5,300);
P2 = exp(-(parfit(2) - parVals2).^2./(2*paramsig(2)^2));


% Generate distribution of residuals
%-------------------------------------------------------------------

%Compute fit residuals
residuals = (V - Vfit)/std(V - Vfit);

% Standard Gaussian distribution
resVals = linspace(-3,3,200);
stdGauss = exp(-resVals.^2/2);

%Fit the Gaussian to the histogram of residuals (for display)
yhist = histcounts(residuals,'normalization','probability');
p = fminsearch(@(p)norm(yhist - p*stdGauss.'),1);
stdGauss = stdGauss*max(yhist);

% Plot histogram of residuals
subplot(234),cla,hold on
histogram(residuals,'normalization','probability','FaceColor','w')
plot(resVals,stdGauss,'m','linewidth',2)
axis tight, box on
legend('Histogram','Std. Gaussian','Location','eastout')


% Bootstrap analysis
%-------------------------------------------------------------------

fcn = @(V) bootfcn(V,r,t);
[bootci,stats] = bootan(fcn,V,Vfit,1000,'verbose',true);

```

```
% Plots
%-------------------------------------------------------------------

subplot(232)
cla, hold on
plot(r,P,'k',r,Pfit,'r')
fill([r fliplr(r)],[Pci{1}(:,1); flipud(Pci{1}(:,2))],'r','facealpha',0.2,'linestyle','none')
fill([r fliplr(r)],[Pci{2}(:,1); flipud(Pci{2}(:,2))],'r','facealpha',0.4,'linestyle','none')
axis tight, box on
xlabel('r [nm]'),ylabel('P(r) [nm^{-1}]')
legend('P_{truth}','P_{fit}','95%-CI','50%-CI','Location','eastout')
ylim([0 1.5])

subplot(235)
cla,hold on
Pbootci = bootci{2};
plot(r,P,'k',r,Pfit,'b')
fill([r fliplr(r)],[Pbootci.ci95(:,1); flipud(Pbootci.ci95(:,2))],'b','facealpha',0.2,'...
    linestyle','none')
fill([r fliplr(r)],[Pbootci.ci50(:,1); flipud(Pbootci.ci50(:,2))],'b','facealpha',0.4,'...
    linestyle','none')
axis tight, grid off, box on
xlabel('r [nm]'),ylabel('P(r) [nm^{-1}]')
legend('P_{truth}','P_{fit}','95%-CI','50%-CI','Location','eastout')
ylim([0 1.5])

subplot(233)
paramboot = stats{1};
hold on
normfact = max(paramboot(1).boothist.bins);
histogram('BinCounts',paramboot(1).boothist.bins(1:2:end-1), 'BinEdges', paramboot(1)...
    .boothist.edges(1:2:end),'FaceColor','w');
plot(parVals1,P1*normfact,'r','Linewidth',1.5)
plot(paramboot(1).bootdist.values, paramboot(1).bootdist.pdf/max(paramboot(1).bootdist.pdf).*...
    normfact,'b','Linewidth',1.5)
axis tight, grid off, box on
xlabel('\lambda')
ylabel('Probability')
legend('Boot. Histogram','Standard Distribution','Boot. Distribution','Location','eastout')
xlim([0.3 0.4])

subplot(236)
paramboot = stats{1};
hold on
normfact = max(paramboot(2).boothist.bins);
histogram('BinCounts',paramboot(2).boothist.bins(1:2:end-1), 'BinEdges', paramboot(2)...
    .boothist.edges(1:2:end),'FaceColor','w');
plot(parVals2,P2*normfact,'r','Linewidth',1.5)
plot(paramboot(2).bootdist.values, paramboot(2).bootdist.pdf/max(paramboot(2).bootdist.pdf).*...
    normfact,'b','Linewidth',1.5)
axis tight, grid off, box on
xlabel('\kappa [\mus^{-1}]')
ylabel('Probability')
legend('Boot. Histogram','Standard Distribution','Boot. Distribution','Location','eastout')
xlim([0.15 0.35])

%Bootstrapped function
function [parfit,Pfit,Vfit] = bootfcn(V,r,t)
    [Vfit,Pfit,¬,parfit] = fitsignal(V,t,r,'P',@bg_exp,@ex_4pdeer,[],'RegParam',0.5);
    parfit = [parfit.ex; parfit.bg];
end
```

Script 8: **groundtruth.m** - Function for generation of the ground truth distributions

```matlab
function [r,P] = groundtruth(figurename)

%Generate the ground truth distance distribution for a given figure
switch figurename

    case 'figure2'
        r = linspace(1.5,8.2,300);
        P = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
                0.001 0.001 0.002 0.004 0.006 0.009 0.014 0.022 0.034 0.052 0.078 0.111 0.148 0.189...
     0.241 0.314 0.428 0.595 0.811 1.046 1.252 1.379 ...
                1.399 1.322 1.184 1.033 0.903 0.807 0.739 0.688 0.644 0.608 0.591 0.612 0.691 0.843...
     1.078 1.387 1.739 2.086 2.366 2.527 2.538 2.395 2.124 ...
                1.77 1.393 1.049 0.777 0.588 0.47 0.393 0.333 0.272 0.207 0.144 0.091 0.051 0.026 0...
    .012 0.005 0.002 0.001 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
                0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
                0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

    case 'figure3'
        r = linspace(2,6,300);
        P = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.001 ...
                0.001 0.002 0.003 0.006 0.009 0.013 0.02 0.028 0.039 0.052 0.067 0.085 0.106 0.128 ...
    0.151 0.175 0.198 0.219 0.238 0.252 0.263 0.271 0.275 ...
                0.278 0.281 0.286 0.295 0.307 0.322 0.34 0.359 0.375 0.387 0.392 0.389 0.378 0.359 ...
    0.336 0.311 0.287 0.266 0.252 0.246 0.249 0.263 0.29 ...
                0.328 0.381 0.446 0.522 0.609 0.701 0.796 0.888 0.973 1.048 1.11 1.157 1.189 1.207 ...
    1.211 1.205 1.189 1.167 1.139 1.108 1.075 1.04 1.002 ...
                0.961 0.919 0.874 0.828 0.783 0.74 0.701 0.665 0.633 0.605 0.579 0.558 0.539 0.526 ...
    0.519 0.521 0.532 0.553 0.584 0.621 0.664 0.709 0.754 ...
                0.796 0.836 0.872 0.905 0.934 0.959 0.977 0.989 0.991 0.983 0.965 0.939 0.906 0.87 ...
    0.834 0.799 0.768 0.74 0.714 0.69 0.665 0.638 0.609 ...
                0.579 0.547 0.516 0.487 0.46 0.436 0.414 0.394 0.375 0.357 0.338 0.318 0.298 0.277 ...
    0.256 0.237 0.218 0.201 0.184 0.169 0.156 0.142 0.13 ...
                0.117 0.106 0.094 0.083 0.073 0.063 0.055 0.047 0.041 0.036 0.033 0.03 0.028 0.026 ...
    0.025 0.025 0.024 0.023 0.022 0.021 0.02 0.019 0.017 ...
                0.016 0.015 0.013 0.012 0.011 0.01 0.009 0.009 0.008 0.007 0.006 0.005 0.004 0.004 ...
    0.003 0.002 0.002 0.001 0.001 0.001 0 0 0 0 0 0 0 0 0 ...
                0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0 0 0];

    case 'figure5'
        r = linspace(1.5,8.2,300);
        P = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
                0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0.0001 0.0001 0.0002 0.0005 0.0009 0.0016 0.0028 ...
                0.0045 0.007 0.0104 0.0144 0.0189 0.0236 0.0282 0.0325 0.0367 0.0409 0.0458 0.0523 ...
    0.062 0.0765 0.0973 0.125 0.1584 0.1954 0.2343 0.2748 ...
                0.3184 0.3668 0.4193 0.4725 0.5205 0.5575 0.5804 0.5898 0.5899 0.5865 0.5841 0.5837...
     0.583 0.5783 0.5676 0.553 0.54 0.5346 0.54 0.5546 ...
                0.5737 0.5914 0.6046 0.6153 0.6305 0.6601 0.7127 0.7908 0.8891 0.9955 1.0964 1.1814...
     1.2456 1.2878 1.3075 1.3027 1.2716 1.2152 1.1389 ...
                1.0522 0.9661 0.8911 0.8354 0.8035 0.795 0.8035 0.8172 0.8221 0.8072 0.7681 0.7088 ...
    0.6385 0.5668 0.5002 0.4407 0.3869 0.3368 0.2896 ...
                0.2456 0.2057 0.17 0.1379 0.1086 0.0821 0.0592 0.0408 0.0271 0.0175 0.0111 0.0069 0...
    .0041 0.0024 0.0013 0.0006 0.0003 0.0001 0 0 0 0 ...
                0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
                0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

    case 'figure7a'
        r = linspace(1,5,300);
        P = [zeros(1,78) 1e-05 1e-05 2e-05 4e-05 7e-05 0.00012 0.00018 0.00027 0.00041 0.00059 ...
    0.00083 0.00116 0.00155 0.00206 0.00274 0.00355 ...
                 0.00457 0.00585 0.0073 0.00904 0.01104 0.01323 0.01571 0.01856 0.02173 0.02547 0...
    .03007 0.03533 0.04172 0.04934 0.0577 0.06706 0.07681 ...
                 0.08654 0.0958 0.10367 0.11042 0.11518 0.11791 0.11947 0.11959 0.11928 0.11905 0...
    .11948 0.12132 0.12414 0.1285 0.13389 0.13963 0.14543 ...
                 0.15017 0.15395 0.15589 0.15572 0.15428 0.1511 0.14695 0.14243 0.13779 0.13377 0...
    .13033 0.12806 0.12744 0.12809 0.13095 0.13664 0.1444 ...
                 0.1555 0.17 0.18652 0.20571 0.22611 0.24666 0.26651 0.28395 0.29946 0.31158 0...
    .32023 0.32678 0.33051 0.33235 0.33323 0.33308 0.33255 ...
                 0.33191 0.33129 0.331 0.33101 0.33159 0.33308 0.33533 0.33893 0.34438 0.35112 0...
    .35994 0.37048 0.38185 0.39397 0.40554 0.41641 0.42575 ...
             0.43347 0.4404 0.44658 0.45339 0.46102 0.47045 0.4822 0.49547 0.511 0.52862 0...
    .54772 0.56916 0.59354 0.62006 0.65006 0.68336 0.71845  ...
             0.75562 0.79282 0.8293 0.86356 0.89365 0.92048 0.94172 0.95645 0.96671 0.97004 0...
    .96677 0.9595 0.94686 0.93088 0.91343 0.89495 0.87738  ...
             0.86087 0.84664 0.83578 0.82754 0.82383 0.82594 0.83221 0.84516 0.86443 0.88715 0...
    .91393 0.94054 0.96526 0.98442 0.99319 0.99426 0.98268 ...
             0.96025 0.93246 0.89944 0.86838 0.84053 0.82031 0.80964 0.80467 0.80724 0.81305 0...
    .81947 0.82451 0.82437 0.81978 0.80704 0.78386 0.75334 ...
             0.71125 0.65859 0.60011 0.53468 0.46771 0.40152 0.33914 0.28407 0.2342 0.1926 0...
    .15826 0.1282 0.10369 0.08326 0.06527 0.05051 0.03853  ...
             0.0283 0.02051 0.01473 0.01012 0.00699 0.00485 0.00323 0.00222 0.00153 0.00102 0...
    .00069 0.00047 0.0003 0.00019 0.00012 7e-05 4e-05 2e-05  ...
             1e-05 1e-05 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];


    case 'figure7b'
        r = linspace(2,6,200);
        P = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
    1e-05 2e-05 3e-05 5e-05 8e-05 0.00014 0.00023  ...
            0.00041 0.00074 0.00131 0.00225 0.00366 0.00565 0.00838 0.01219 0.0178 0.02633 0...
    .03922 0.05771 0.08221 0.11188 0.14518 0.18168 0.2248  ...
            0.28275 0.36735 0.49053 0.65741 0.85896 1.06947 1.25245 1.37376 1.39808 1.35365 1...
    .24516 1.10999 0.97984 0.87289 0.79319 0.73507 0.68932  ...
            0.64908 0.61427 0.59408 0.59927 0.63937 0.73502 0.89383 1.11895 1.40177 1.71917 2...
    .03425 2.30415 2.49212 2.53524 2.50154 2.32238 2.05226  ...
            1.72641 1.3876 1.07637 0.82189 0.63535 0.51027 0.42821 0.36843 0.31529 0.25996 0...
    .20167 0.14516 0.09623 0.05855 0.03262 0.01664 0.00776  ...
            0.0033 0.00128 0.00045 0.00014 4e-05 1e-05 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  ...
            0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

    case 'figure8'
        r = linspace(2.5,6,200);
        P = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.0001 0.0002 0.0003 0.0005 0...
    .0008 0.0013 0.002 0.003 0.0044 0.0064 0.0087 ...
            0.0116 0.0148 0.0183 0.022 0.0256 0.0292 0.0326 0.0358 0.0391 0.0426 0.0467 0.0518 ...
    0.0592 0.0692 0.0824 0.0992 0.1209 0.1463 0.1741 ...
            0.2035 0.234 0.2658 0.2994 0.3352 0.3737 0.415 0.4567 0.4959 0.5306 0.5588 0.5767 0...
    .5862 0.5898 0.5892 0.5866 0.5846 0.5839 0.5835 ...
            0.5825 0.5787 0.5709 0.5606 0.5496 0.5398 0.5356 0.5379 0.5458 0.5582 0.5731 0.587 ...
    0.5985 0.6081 0.6169 0.6288 0.6506 0.6845 0.7321 ...
            0.7941 0.8713 0.9533 1.0347 1.1112 1.1779 1.2291 1.2679 1.2939 1.307 1.3032 1.2819 ...
    1.246 1.197 1.137 1.0689 1.0012 0.9378 0.8821 ...
            0.8384 0.8121 0.7991 0.7975 0.8046 0.8154 0.8204 0.8155 0.7983 0.7675 0.7209 0.668 ...
    0.6123 0.5568 0.5045 0.4573 0.4135 0.3724 0.3333 ...
            0.2962 0.2612 0.2284 0.198 0.1701 0.1448 0.1212 0.0992 0.0789 0.0609 0.0462 0.034 0...
    .0244 0.0171 0.012 0.0084 0.0057 0.0038 0.0024 ...
            0.0015 0.0009 0.0005 0.0003 0.0001 0.0001 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

    case 'figure10_1'

        r = linspace(2,8,300);
        P = [0 0 0 0 0 0 0 1e-05 1e-05 2e-05 3e-05 4e-05 7e-05 0.0001 0.00014 0.00019 0.00025 0...
    .00033 0.00042 0.00053 0.00068 0.00087 0.00111   ...
            0.00139 0.00169 0.00197 0.00219 0.00234 0.0024 0.0024 0.00239 0.0024 0.00247 0.0026...
     0.00277 0.00295 0.00308 0.00314 0.00311 0.00302  ...
            0.00288 0.00274 0.00263 0.00256 0.00256 0.00264 0.00284 0.00318 0.00365 0.00422 0...
    .00485 0.00545 0.00595 0.00631 0.00654 0.00665 0.00669  ...
            0.00668 0.00666 0.00664 0.00664 0.00666 0.00671 0.00682 0.007 0.00727 0.0076 0...
    .00797 0.00831 0.00859 0.00881 0.00899 0.0092 0.00949  ...
            0.00986 0.01033 0.01088 0.01153 0.01229 0.0132 0.01423 0.01535 0.01647 0.0175 0...
    .01836 0.019 0.01938 0.01948 0.01932 0.01894 0.01842  ...
            0.01787 0.01735 0.01692 0.01662 0.01651 0.01663 0.01702 0.01767 0.01847 0.01927 0...
    .01985 0.02001 0.01966 0.01888 0.01788 0.01697 0.01636  ...
            0.01613 0.01621 0.01642 0.01657 0.01651 0.01613 0.01533 0.01406 0.01237 0.0104 0...
    .00837 0.00649 0.00491 0.00366 0.0027 0.00196 0.00138  ...
            0.00094 0.00061 0.00037 0.00022 0.00012 7e-05 4e-05 2e-05 1e-05 1e-05 0 0 0 0 0 0 0...
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  ...
            0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  ...
            0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

    case 'figure10_2'

```

```
        r = linspace(2,8,300);
        P = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
            1e-05 2e-05 3e-05 5e-05 8e-05 0.00013 0.0002 0.00029 0.00039 0.00049 0.00059 0...
    .00069 0.00079 0.00088 0.00098 0.00111 0.0013 0.00158 0.00199 ...
            0.00255 0.00326 0.00406 0.00492 0.00581 0.00676 0.0078 0.00895 0.01015 0.01129 0...
    .01222 0.01286 0.01317 0.01322 0.01318 0.0131 0.01308 ...
            0.01307 0.01301 0.01282 0.01251 0.01219 0.012 0.01204 0.01229 0.01269 0.01311 0...
    .01345 0.01369 0.01398 0.0145 0.01547 0.01701 0.01907 ...
            0.02142 0.02376 0.02582 0.02744 0.02857 0.02921 0.02925 0.02883 0.02776 0.0262 0...
    .02431 0.02235 0.02055 0.01912 0.01819 0.01785 0.01791 ...
            0.01821 0.0184 0.01826 0.0176 0.01642 0.01491 0.01329 0.01175 0.01036 0.00911 0...
    .00796 0.00688 0.00586 0.00493 0.0041 0.00335 0.00267 ...
            0.00205 0.00151 0.00106 0.00071 0.00046 0.00029 0.00018 0.00011 7e-05 4e-05 2e-05 1...
    e-05 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
            0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
            0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
     0 0];


    case 'figure11'
        r = linspace(2,6,200);
        P = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
    0 0.00011 0.001 0.00211 0.00484 0.00946 0.01733 ...
            0.02933 0.0471 0.06927 0.0982 0.13102 0.16621 0.20087 0.23114 0.25357 0.26817 0...
    .27546 0.27997 0.28738 0.30286 0.32483 0.35254 0.37696 ...
            0.39035 0.38721 0.36533 0.33176 0.29486 0.26356 0.24795 0.25155 0.28147 0.33788 0...
    .42578 0.53866 0.67285 0.8141 0.94745 1.06055 1.14316 ...
            1.1928 1.20993 1.20172 1.1734 1.13239 1.08437 1.03168 0.97241 0.9089 0.84055 0...
    .77312 0.71144 0.65749 0.61237 0.57398 0.54391 0.52432 ...
            0.52054 0.53732 0.57641 0.63207 0.6982 0.76504 0.82662 0.88086 0.92737 0.96396 0...
    .98637 0.98887 0.96906 0.92979 0.87785 0.82404 0.7746 ...
            0.73248 0.695 0.65705 0.61488 0.5694 0.52212 0.47876 0.44062 0.4078 0.37856 0.35101...
     0.32164 0.29127 0.25971 0.23081 0.20392 0.17904 ...
            0.15817 0.13797 0.1191 0.10191 0.08472 0.06954 0.05621 0.04489 0.03673 0.03193 0...
    .02828 0.02564 0.025 0.02364 0.02213 0.02063 0.01913 ...
            0.01662 0.01512 0.01262 0.01112 0.00961 0.009 0.00761 0.00611 0.0046 0.004 0.0026 0...
    .002 0.001 0.001 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
            0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

    otherwise
        error('This figure has not associated data.')
end

%Normalize the distributions
dr = mean(diff(r));
P = P/sum(P)/dr;

%Return as column vector
P = P(:);

end
```