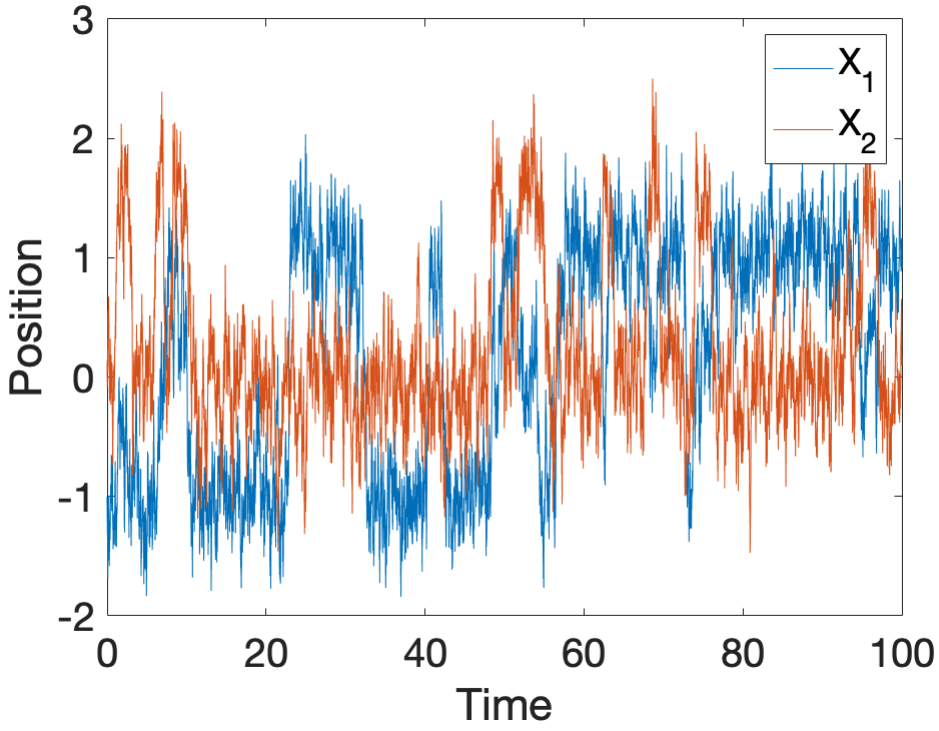
Analysis of the Simulation Data from Triple-Well Potential

```
clearvars;
addpath(genpath('../'))
```

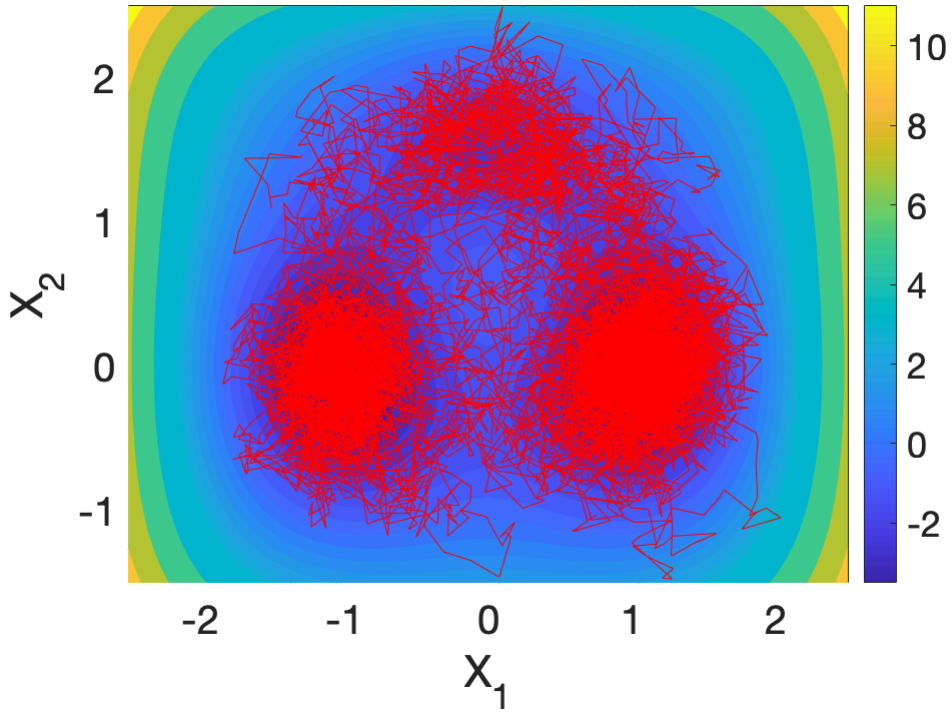
Generating simulation data

```
rng(1)
x_0 = [-1,0]';
par.T_max = 100;
par.dt = 1e-2;
sigma = 0.8; % 1.2,1.5
[x_out, t_out] = generating_simulation_
```

```
figure;
plot(t_out,x_out)
set(gca,'FontSize',20)
legend('X_{1}','X_{2}')
xlabel('Time')
ylabel('Position')
```




```
figure;
fc = fcontour(@triple_well_potential,[-
fc.LevelList = [-10:0.5:3,3:2:50];
colorbar
hold on
plot(x_out(1,:),x_out(2,:),"Color",'r',
xlabel('X_{1}')
ylabel('X_{2}')
set(gca,'FontSize',20)
```




```
data = x_out(:,1:5:end)';
%save
%{
save('../Data/double_well.mat','x_out',
%}
```

Load the Data and Estimate Number of Clusters by EPI

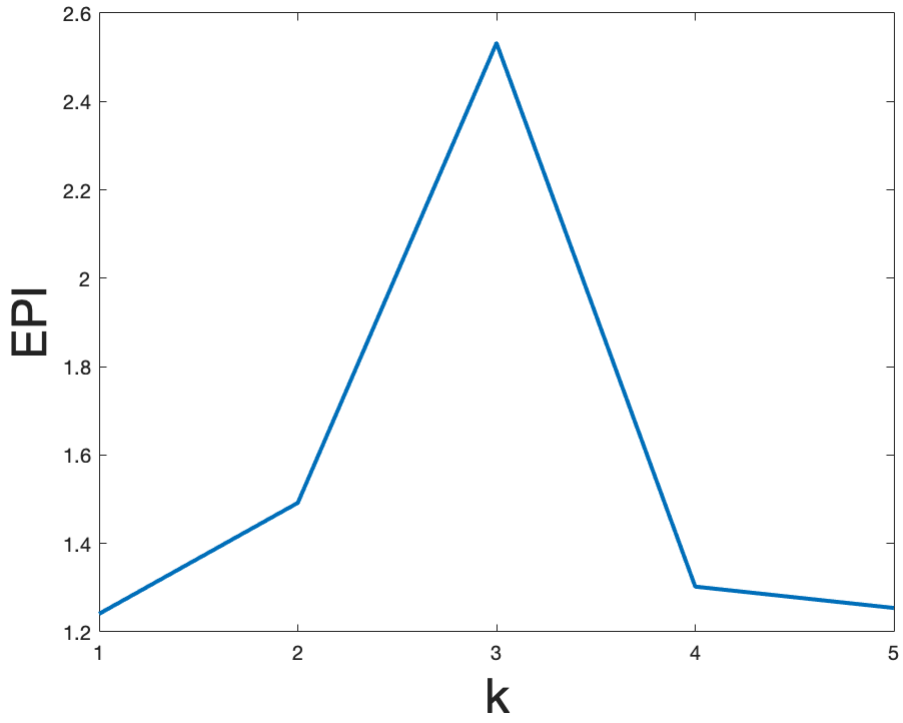
From EPI plot, we intend to seek for the index where peak occurs, which serves

as the candidate for the choice of cluster numbers

```
out = EstClusterNum(data,[]);
```

```
Computed P-values 500 of 2001 datapoint
Computed P-values 1000 of 2001 datapoir
Computed P-values 1500 of 2001 datapoir
Computed P-values 2000 of 2001 datapoir
Mean value of sigma: 0.61654
Minimum value of sigma: 0.4928
Maximum value of sigma: 0.78183
```

```
figure;
plot(out.ratio(1:5), 'linewidth',2.0)
xlabel('k', 'FontSize', 24);
ylabel('EPI', 'FontSize', 24);
xticks(0:30);
```



Dynamical Analysis and Output

```
tic;
par.choice_distance = 'euclid';
par.K_cluster = 3; %selected based on E
par.trials = 2; % number of random trai
par.initial = 'random';
% the main function of MuTrans
Output = DynamicalAnalysis (data, par);
```

```
Computed P-values 500 of 2001 datapoint
Computed P-values 1000 of 2001 datapoir
Computed P-values 1500 of 2001 datapoir
```

```
Computed P-values 2000 of 2001 datapoir
Mean value of sigma: 0.61654
Minimum value of sigma: 0.4928
Maximum value of sigma: 0.78183
J_new = 1.8788
J_new = 1.5264
J_new = 1.4310
J_new = 1.3600
J_new = 1.3389
J_new = 1.3370
J_new = 1.3361
J_new = 1.3356
J_new = 1.3350
J_new = 1.3344
J_new = 1.3344
J_new = 2.1691
J_new = 1.7141
J_new = 1.6497
J_new = 1.6321
J_new = 1.6197
J_new = 1.6060
```

```
J_new = 1.5878
J_new = 1.5664
J_new = 1.5358
J_new = 1.4971
J_new = 1.4413
J_new = 1.3711
J_new = 1.3485
J_new = 1.3424
J_new = 1.3387
J_new = 1.3374
J_new = 1.3361
J_new = 1.3352
J_new = 1.3350
J_new = 1.3348
J_new = 1.3344
J_new = 1.3344
E_best = 0.1221

 Iteration  Func-count       f(x)
     0           1          1.33688
     1          10          1.27642
```

2	12	1.27471
3	14	1.2667
4	16	1.26264
5	17	1.26189
6	18	1.25815
7	20	1.25667
8	21	1.25579
9	22	1.25547
10	23	1.25414
11	25	1.25349
12	26	1.25284
13	27	1.25228
14	28	1.25154
15	29	1.25109
16	30	1.25065
17	31	1.25026
18	32	1.24992
19	33	1.24958

Iteration	Func-count	f(x)
20	34	1.24936

21	35	1.24908
22	36	1.24893
23	37	1.24869
24	38	1.24855
25	39	1.24842
26	40	1.24827
27	41	1.2482
28	42	1.24806
29	43	1.24804
30	44	1.24794
31	46	1.24788
32	47	1.24784
33	48	1.24779
34	49	1.24777
35	50	1.24773
36	51	1.24772
37	52	1.2477
38	53	1.24769
39	54	1.24766

Iteration	Func-count	f(x)

40	55	1.24766
41	56	1.24764
42	57	1.24764
43	58	1.24762
44	59	1.24762
45	60	1.24761
46	61	1.2476
47	62	1.2476
48	63	1.24759
49	64	1.24759
50	65	1.24758
51	66	1.24758
52	67	1.24758
53	68	1.24758
54	69	1.24757
55	70	1.24757
56	71	1.24757
57	72	1.24757
58	73	1.24757
59	74	1.24756

Iteration	Func-count	f(x)
60	75	1.24756
61	76	1.24756
62	77	1.24756
63	78	1.24756
64	79	1.24756
65	80	1.24756
66	81	1.24756
67	82	1.24755
68	83	1.24755
69	84	1.24755
70	85	1.24755
71	86	1.24755
72	87	1.24755
73	88	1.24755
74	89	1.24755
75	90	1.24755
76	91	1.24755
77	92	1.24755
78	93	1.24755
79	94	1.24754

Iteration	Func-count	f(x)
80	95	1.24754
81	96	1.24754
82	97	1.24754
83	98	1.24754
84	99	1.24754
85	100	1.24754
86	101	1.24754
87	102	1.24754
88	103	1.24754
89	104	1.24754
90	105	1.24754
91	106	1.24754
92	107	1.24754
93	108	1.24754
94	109	1.24754
95	110	1.24754
96	111	1.24754
97	112	1.24754
98	113	1.24754

99	114	1.24754

Iteration	Func-count	f(x)
100	115	1.24754
101	116	1.24754
102	117	1.24753
103	118	1.24753
104	119	1.24753
105	120	1.24753
106	121	1.24753
107	122	1.24753
108	123	1.24753
109	124	1.24753
110	125	1.24753
111	126	1.24753
112	127	1.24753
113	128	1.24753
114	129	1.24753
115	130	1.24753
116	131	1.24753
117	132	1.24753

118	133	1.24753
119	134	1.24753

Iteration	Func-count	f(x)
120	135	1.24753
121	136	1.24753
122	137	1.24753
123	138	1.24753
124	139	1.24753
125	140	1.24753
126	141	1.24753
127	142	1.24753
128	143	1.24753
129	144	1.24753
130	145	1.24753
131	146	1.24753
132	147	1.24753
133	148	1.24752
134	149	1.24752
135	150	1.24752
136	151	1.24752

```
   137          152          1.24752
   138          153          1.24752
   139          154          1.24752

 Iteration  Func-count        f(x)
   140          155          1.24752
   141          156          1.24752
   142          157          1.24752
   143          158          1.24752
   144          159          1.24752
   145          160          1.24752
   146          161          1.24752
   147          162          1.24752
   148          163          1.24752

Optimization completed: The first-order
than options.OptimalityTolerance = 1.00

Elapsed time is 144.648490 seconds.
```

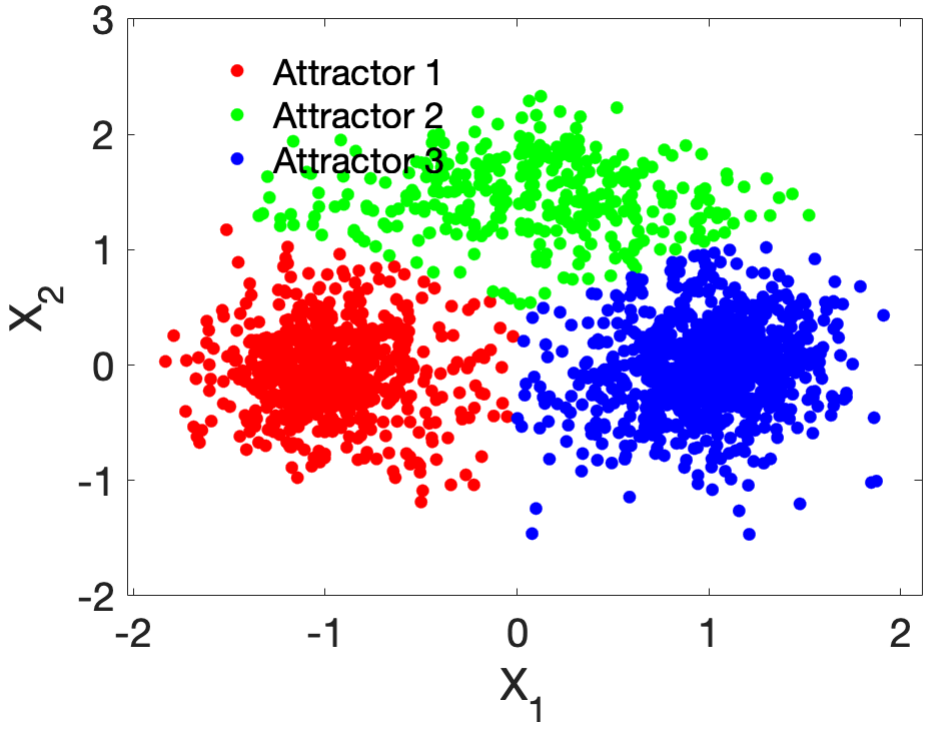
```
toc;
```

```
Elapsed time is 144.666496 seconds.
```

```
class_order = Output.class_order;
rho_class = Output.rho_class;
perm_class = Output.perm_class;
P_perm = Output.P_perm;
P_hat = Output.P_hat;
P_appr_perm = Output.P_appr_perm;
P_rho = Output.P_rho;
labs_perm = Output.labs_perm;
data_perm = Output.data_perm;
mu_hat = Output.mu_hat;
k = Output.k;
H = Output.H;
```

```
figure;
gscatter(data_perm(:,1),data_perm(:,2),
legend('Attractor 1','Attractor 2','Att
xlabel('X_{1}')
ylabel('X_{2}')
set(gca,'FontSize',20)
```

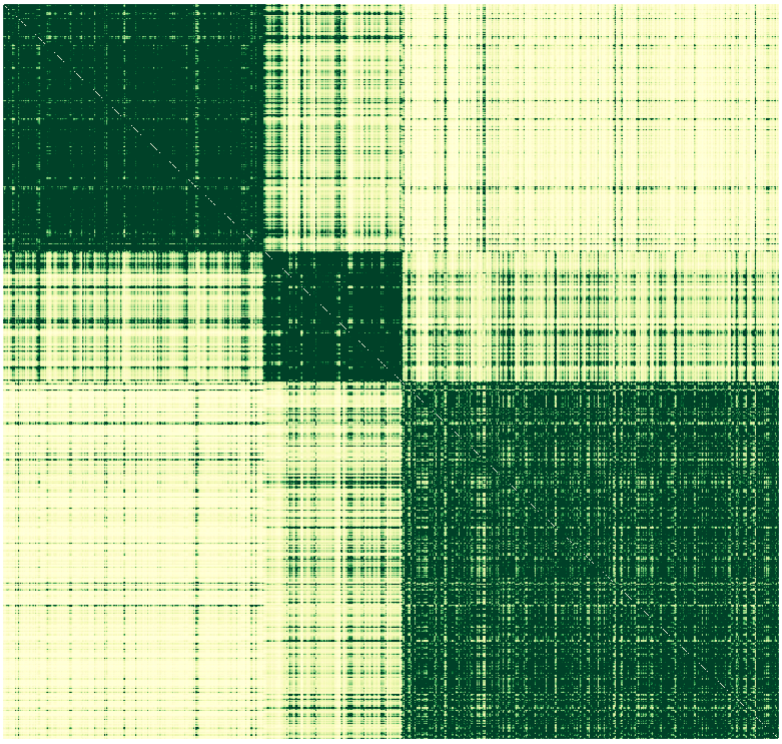
```
legend boxoff
```



Plot the Cell-Cell Scale rwTPM

```
max_P = 0.2* max(max(P_rho));
c_lim = [0 max_P];
cmp = 'ylgn';

figure('rend','painters','pos',[10 10 5
colormap(brewermap([],cmp))
imagesc(P_perm);
axis off
```

```
set(gca,'xtick',[],'ytick',[]);
caxis(c_lim)
```

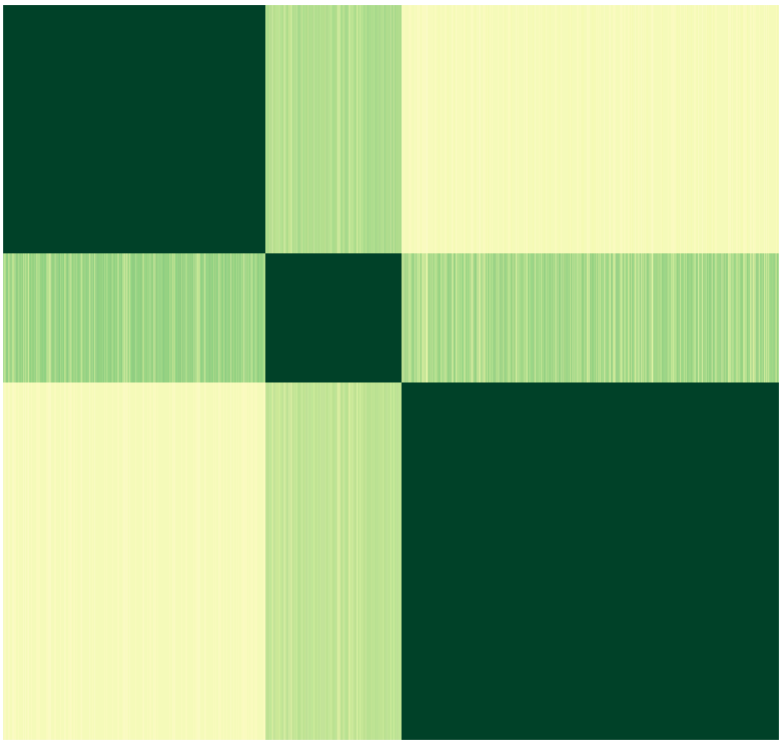


```
%colorbar;
```

Plot the Cluster-Cluster Scale rwTPM

```
figure('rend','painters','pos',[10 10 5
colormap(brewermap([],cmp))
imagesc(P_appr_perm);
axis off
set(gca,'xtick',[],'ytick',[]);
```

```
caxis(c_lim)
```

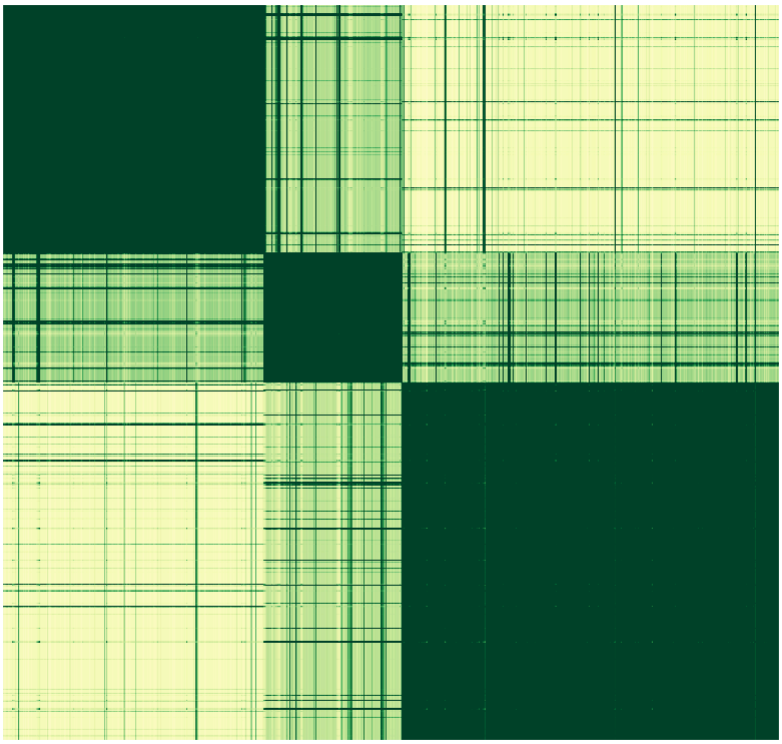


```
%colorbar;
```

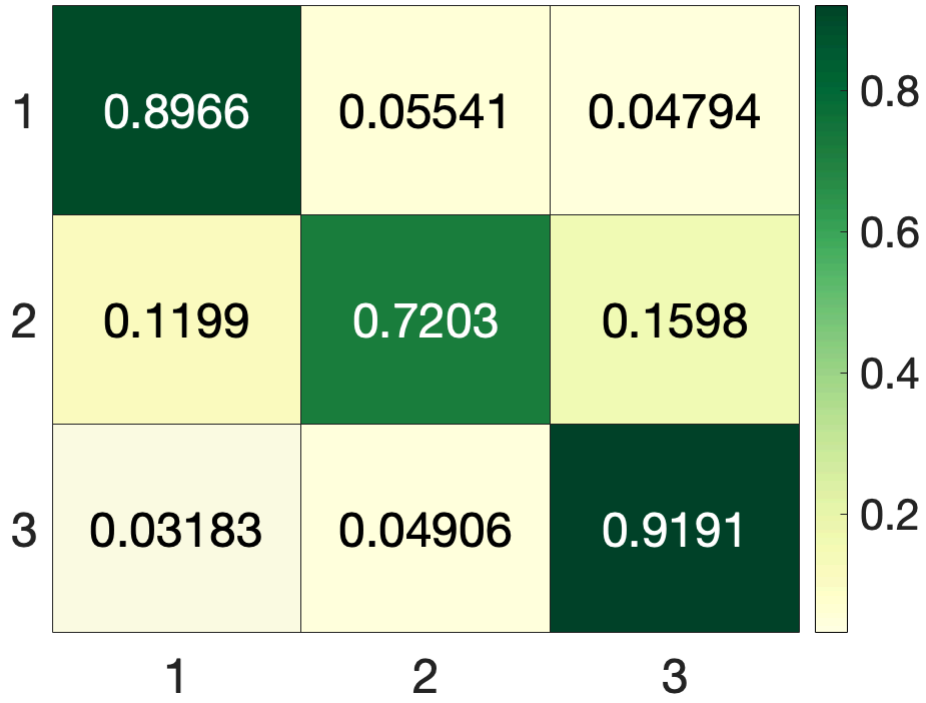
Plot the Cell-Cluster Scale rwTPM

```
figure('rend','painters','pos',[10 10 5
colormap(brewermap([],cmp))
imagesc(P_rho);
axis off
```

```
set(gca,'xtick',[],'ytick',[]);
caxis(c_lim)
%colorbar;
box off
```



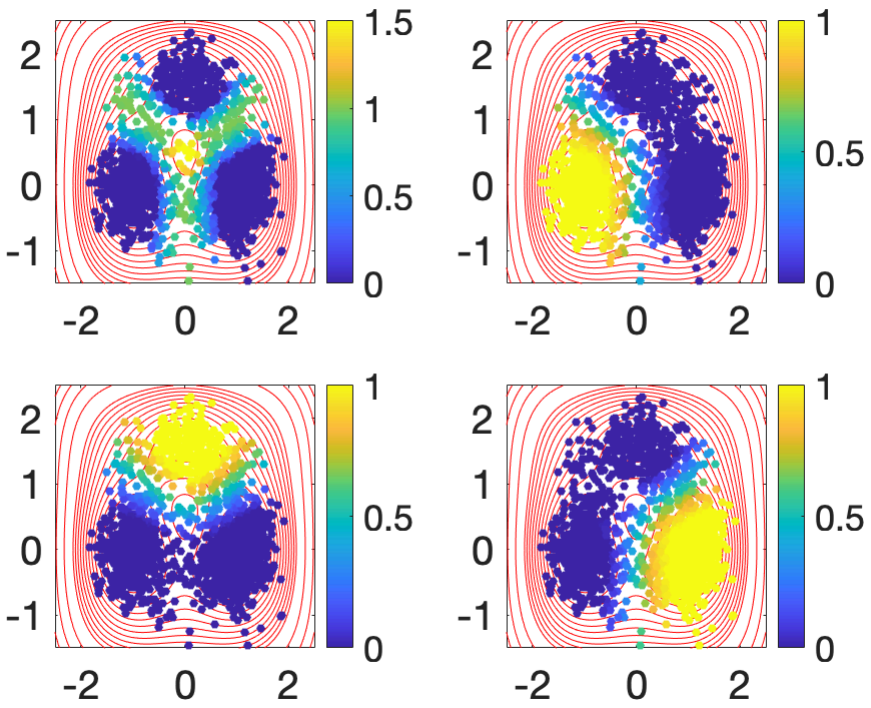
```
figure
heatmap(P_hat,'Colormap',colormap(brewe
```




```
figure
subplot(2,2,1)
fc = fcontour(@triple_well_potential,[-
fc.LevelList = [-5:0.5:3,3:2:15];
hold on
scatter(data_perm(:,1),data_perm(:,2),2
colorbar
caxis([0 1.5])
set(gca,'FontSize',20)

subplot(2,2,2)
```

```
fc = fcontour(@triple_well_potential,[-
fc.LevelList = [-5:0.5:3,3:2:15];
hold on
scatter(data_perm(:,1),data_perm(:,2),2
colorbar
caxis([0 1])
set(gca,'FontSize',20)

subplot(2,2,3)
fc = fcontour(@triple_well_potential,[-
fc.LevelList = [-5:0.5:3,3:2:15];
hold on
scatter(data_perm(:,1),data_perm(:,2),2
colorbar
caxis([0 1])
set(gca,'FontSize',20)

subplot(2,2,4)
fc = fcontour(@triple_well_potential,[-
fc.LevelList = [-5:0.5:3,3:2:15];
hold on
```

```
scatter(data_perm(:,1),data_perm(:,2),2
colorbar
caxis([0 1])
set(gca,'FontSize',20)
```




```
function v = triple_well_potential(x,y)
    v = 3*exp(-x.^2-(y-1/3).^2)-3*exp(-
end

function dv = triple_well_gradient_neg(
    dv = zeros(2,1);
    dv(1) = -6*x(1)*exp(-x(1)^2-(x(2)-1
```

```
    dv(2) = -6*(x(2)-1/3)*exp(-x(1)^2-(
    dv(1) = -dv(1);
    dv(2) = -dv(2);
end
```