function [Ahat, QHat,lambdahat, H\_Theta\_Hat, SHat, T\_hat, dTau, Res, Cost, XHat, YHat]= HMC\_binding(steps ,tol ,T ,a1, a2)

Tb = T;

gtol = tol;

bsteps = steps;

Res = zeros(1, steps);

dTau = zeros(1, steps);

Cost = zeros(1, steps);

%dtau,

while tol >= 0.05 & steps > 0 & T >= 10

l = max(length(a1), length(a2));

if length(a1) > length(a2)

dx = (a2(length(a2)) - a2(1))/length(a2);

a2 = [a2, a2+dx:dx:a2+(l - length(a2))\*dx];

else

dx = (a1(length(a1)) - a1(1))/length(a1);

a1 = [a1, a1+dx:dx:a1+(l - length(a1))\*dx];

end

[f1, gof, output] = fit(a1', a2', 'poly3', 'Normalize','on','Robust','Bisquare');

% dx = (a1(length(a1)) - a1(1))/T;

x = a1(1):(a1(length(a1))-a1(1))/(T-1):a1(length(a1));

y = f1(x);

if steps == 1 | steps == 50

figure,

plot(f1, a1, a2);

hold on

plot(x, y);

% hold on

% plot(g1, b1, b2);

% hold on

% plot(m, n);

end

b = zeros(1, T);

bphi = b;

sigma = b;

normb = b;

normy = b;

Qhat = zeros(T-1);

alphac = ones(size(x))./(1+exp(-(x-mean(x))/max(x)));%sigmoid(x)

for t = 1:T-1

b(t) = x(t+1) - x(t);

bphi(t) = alphac(t+1) -alphac(t);

sigma(t) = std(x(1:t+1));

normb(t) = sqrt(2\*(T-1)\*b(t)\*sigma(t));

normy(t) = sqrt(2\*(T-1)\*y(t)\*sigma(t));

end

a = sigma'\*sigma;

Qhat = inv(a(1:T-1,1:T-1)).\*(normb(1:T-1)./normy(1:T-1).\*y(1:T-1)-b(1:T-1));

lambda = normb(1:T-1)./normy(1:T-1);

%dphi = abs((alphac(2:T)'-alphac(1:T-1)')/dx./(mean(a(1:T-1,:),2)-mean([zeros(1,T-1);a(1:T-2,:)],2)));

%bphi = abs((alphac(2:T)'-alphac(1:T-1)')./(mean(a(1:T-1,:),2)-mean([zeros(1,T-1);a(1:T-2,:)],2)));

dphi = (alphac(2:T)'-alphac(1:T-1)')/dx;

bphi = alphac(2:T)'-alphac(1:T-1)';

steps = steps -1;

% tol = abs(mean(alphac)-mean(mean(lambda(min(unique(round(alphac\*100)),size(lambda, 1)), min(unique(round(bphi\*10000)), size(lambda,2))))));

% tol = abs(mean(alphac)-mean(mean(lambda(min(unique(alphac),size(lambda, 1)), min(unique(bphi), size(lambda,2))))));

% Res(50-steps) = mean(lambda(1:T) -alphac(1:T));

% Cost(50-steps) = T\*log(T);

% dTau(50-steps) = 1/T;

if tol < 0.05 | steps == 0 | T <= 10

if steps < 49

if (Res(50-steps) == Res(50-steps-1)) | ((Res(50-steps) == Res(50-steps-2)))

break;

end

end

else

T = T - 10;

end

end

%T = T +10;

%alphac = alphac + [bphi(1),bphi',bphi(length(bphi))];

alphac = alphac + [bphi(1),bphi'];

Xhat = alphac\*100;

%xhat = alphac;

Yhat = f1(Xhat);

alphachat = ones(size(Xhat))./(1+exp(-(Xhat-mean(Xhat))/max(Xhat)));%sigmoid(x)

Bhat = zeros(1, T-1);

Bphihat = Bhat;

Sigmahat = Bhat;

normBhat = Bhat;

normYhat = Bhat;

for t = 1:T-1

Bhat(t) = Xhat(t+1) - Xhat(t);

Bphihat(t) = alphachat(t+1) -alphachat(t);

Sigmahat(t) = std(Xhat(1:t+1));

normBhat(t) = sqrt(2\*(T-1)\*Bhat(t)\*Sigmahat(t));

normYhat(t) = sqrt(2\*(T-1)\*Yhat(t)\*Sigmahat(t));

end

Ahat = Sigmahat'\*Sigmahat;

QHat = inv(Ahat).\*(normBhat(1:length(Ahat))./normYhat(1:length(Ahat)).\*normYhat(1:length(Ahat))-Bhat);

%QHat = inv(ahat)\*(bhat(1:length(ahat))./(yhat(1:length(ahat)).^2-bhat);

lambdahat = abs(normBhat(1:length(Ahat))./normYhat(1:length(Ahat)));

% H\_theta = (alphac - alphachat)/dtau;

H\_theta\_hat = lambdahat' \* Bphihat;

SHat = meshgrid(Bphihat , QHat);

tau\_hat = 1/(2\*lambdahat(1));

[f2, gof2, output2] = fit(Xhat', x', 'poly3', 'Normalize','on','Robust','Bisquare');

XHat = f2(Xhat);

YHat = f1(f2(Xhat));

alphacHat = ones(size(XHat))./(1+exp(-(XHat-mean(XHat))/max(XHat)));%sigmoid(x)

BHat = zeros(1, T-1);

BphiHat = BHat;

SigmaHat = BHat;

normBHat = BHat;

normYHat = BHat;

for t = 1:T-1

BHat(t) = XHat(t+1) - XHat(t);

BphiHat(t) = alphacHat(t+1) -alphacHat(t);

SigmaHat(t) = std(XHat(1:t+1));

normBHat(t) = sqrt(2\*(T-1)\*BHat(t)\*SigmaHat(t));

normYHat(t) = sqrt(2\*(T-1)\*YHat(t)\*SigmaHat(t));

end

AHat = SigmaHat'\*SigmaHat;

Q\_Hat = inv(AHat).\*(normBHat(1:length(AHat))./normYHat(1:length(AHat)).\*normYHat(1:length(AHat))-BHat);

%QHat = inv(ahat)\*(bhat(1:length(ahat))./(yhat(1:length(ahat)).^2-bhat);

Lambdahat = abs(normBHat(1:length(AHat))./normYHat(1:length(AHat)));

% H\_theta = (alphac - alphachat)/dtau;

H\_Theta\_Hat = Lambdahat' \* BphiHat;

S\_Hat = meshgrid(BphiHat , Q\_Hat);

T\_hat(1)= 1/(2\*Lambdahat(1));

for j = 2: length(Lambdahat)

T\_hat(j) = T\_hat(j-1) + 1/Lambdahat(j);

end

%T\_hat = T\_hat + 1/(2\*lambdahat(length(lambdahat)));

DTau = ones(size(T\_hat))/T\_hat;

%H = reshape(H\_theta\_hat, length(Lambdahat),length(Lambdahat));

%C = mean(mean(S\_Hat,1),2);

%c = abs(abs(C/max(abs(C)))-1)\*255;

% steps = bsteps;

% b1 = a1(length(a1):-1:1);

% b2 = a2(length(a2):-1:1) + [0, 200, 400, 400, 0, 0, -300, -400, -1000, -1000, -2000, -3000, -1000, -100, -100];

% dm = (b1(length(b1)) - b1(1))/Tb;

% [g1, gog, outputg] = fit(b1', b2', 'poly3', 'Normalize','on','Robust','Bisquare');

% m = b1(1):dm:b1(length(b1));

% n = g1(b1(1):dm:b1(length(b1)));

%

% gRes = zeros(1, steps);

% gdTau = zeros(1, steps);

% gCost = zeros(1, steps);

% [g1, gog, outputg] = fit(b1', b2', 'poly3', 'Normalize','on','Robust','Bisquare');

%

% while gtol >= 0.05 & steps > 0 & Tb >= 10

% dm = (b1(length(b1)) - b1(1))/Tb;

%

% m = b1(1):dm:b1(length(b1));

% n = g1(b1(1):dm:b1(length(b1)));

% if steps == 1 | steps == 50

% figure,

% plot(g1, b1, b2);

% hold on

% plot(m, n);

% % hold on

% % plot(g1, b1, b2);

% % hold on

% % plot(m, n);

% end

% gb = zeros(1, Tb);

% gbphi = gb;

% gsigma = gb;

% gnormb = gb;

% gnormn = gb;

% gQhat = zeros(Tb-1);

% galphac = ones(size(m))./(1+exp(-(m-mean(m))/max(m)));%sigmoid(x)

% for t = 2:Tb

% gb(t-1) = m(t+1) - m(t);

% gbphi(t-1) = galphac(t+1) - galphac(t);

% gsigma(t-1) = std(m(2:t+1));

% gnormb(t-1) = sqrt(-2\*(Tb-1)\*gb(t-1)\*gsigma(t-1));

% gnormn(t-1) = sqrt(2\*(Tb-1)\*n(t)\*gsigma(t-1));

% end

% ga = gsigma'\*gsigma;

% gQhat = -inv(ga(1:Tb-1,1:Tb-1)).\*(gnormb(1:Tb-1)./gnormn(1:Tb-1).\*n(1:Tb-1)-gb(1:Tb-1));

% glambda = gnormb(1:Tb-1)./gnormn(1:Tb-1);

% %dphi = abs((alphac(2:T)'-alphac(1:T-1)')/dx./(mean(a(1:T-1,:),2)-mean([zeros(1,T-1);a(1:T-2,:)],2)));

% %bphi = abs((alphac(2:T)'-alphac(1:T-1)')./(mean(a(1:T-1,:),2)-mean([zeros(1,T-1);a(1:T-2,:)],2)));

% gdphi = (galphac(2:Tb)'-galphac(1:Tb-1)')/dm;

% gbphi = galphac(2:Tb)'-galphac(1:Tb-1)';

% steps = steps -1;

% gtol = abs(mean(galphac)-mean(mean(glambda(min(unique(round(abs(galphac\*100))),size(glambda, 1)), min(unique(round(abs(gbphi\*10000))), size(glambda,2))))));

% % tol = abs(mean(alphac)-mean(mean(lambda(min(unique(alphac),size(lambda, 1)), min(unique(bphi), size(lambda,2))))));

% gRes(50-steps) = gtol;

% gCost(50-steps) = Tb\*log(Tb);

% gdTau(50-steps) = 1/Tb;

% if gtol < 0.05 | steps == 0 | Tb <= 10

% if steps < 49

% if (gRes(50-steps) == gRes(50-steps-1)) | ((gRes(50-steps) == gRes(50-steps-2)))

% break

% end

% end

% else

% Tb = Tb - 10;

% end

% end

% Tb = Tb +10;

% galphac = galphac + [gbphi(1),gbphi',gbphi(length(gbphi))];

% Mhat = galphac\*100;

% %xhat = alphac;

% Nhat = g1(Mhat);

% galphachat = ones(size(Mhat))./(1+exp(-(Mhat-mean(Mhat))/max(Mhat)));%sigmoid(x)

% gBhat = zeros(1, Tb-1);

% gBphihat = gBhat;

% gSigmahat = gBhat;

% gnormBhat = gBhat;

% gnormNhat = gBhat;

% for t = 2:Tb

% gBhat(t-1) = Mhat(t+1) - Mhat(t);

% gBphihat(t-1) = galphachat(t+1) -galphachat(t);

% gSigmahat(t-1) = std(Mhat(2:t+1));

% gnormBhat(t-1) = sqrt(-2\*(Tb-1)\*gBhat(t-1)\*gSigmahat(t-1));

% gnormNhat(t-1) = sqrt(2\*(Tb-1)\*Nhat(t)\*gSigmahat(t-1));

% end

% gAhat = gSigmahat'\*gSigmahat;

% gQHat = -inv(gAhat).\*(gnormBhat(1:length(gAhat))./gnormNhat(1:length(gAhat)).\*gnormNhat(1:length(gAhat))-gBhat);

% %QHat = inv(ahat)\*(bhat(1:length(ahat))./(yhat(1:length(ahat)).^2-bhat);

% glambdahat = abs(gnormBhat(1:length(gAhat))./gnormNhat(1:length(gAhat)));

%

% % H\_theta = (alphac - alphachat)/dtau;

% gH\_theta\_hat = glambdahat' \* gBphihat;

% gSHat = meshgrid(gBphihat , gQHat);

%

% gtau\_hat = 1/(2\*glambdahat(1));

% [g2, gog2, outputg2] = fit(Mhat', m', 'poly3', 'Normalize','on','Robust','Bisquare');

% MHat = g2(Mhat);

% NHat = g1(g2(Mhat));

%

% galphacHat = ones(size(MHat))./(1+exp(-(MHat-mean(MHat))/max(MHat)));%sigmoid(x)

% gBHat = zeros(1, Tb-1);

% gBphiHat = gBHat;

% gSigmaHat = gBHat;

% gnormBHat = gBHat;

% gnormMHat = gBHat;

% for t = 2:Tb

% gBHat(t-1) = MHat(t+1) - MHat(t);

% gBphiHat(t-1) = galphacHat(t+1) - galphacHat(t);

% gSigmaHat(t-1) = std(MHat(1:t+1));

% gnormBHat(t-1) = sqrt(-2\*(Tb-1)\*gBHat(t-1)\*gSigmaHat(t-1));

% gnormNHat(t-1) = sqrt(2\*(Tb-1)\*NHat(t)\*gSigmaHat(t-1));

% end

% gAHat = gSigmaHat'\*gSigmaHat;

% gQ\_Hat = -abs(inv(gAHat).\*(gnormBHat(1:length(gAHat))./gnormNHat(1:length(gAHat)).\*gnormNHat(1:length(gAHat))-gBHat));

% %QHat = inv(ahat)\*(bhat(1:length(ahat))./(yhat(1:length(ahat)).^2-bhat);

% gLambdahat = abs(gnormBHat(1:length(gAHat))./gnormNHat(1:length(AHat)));

%

% % H\_theta = (alphac - alphachat)/dtau;

% gH\_Theta\_Hat = -gLambdahat' \* gBphiHat;

% gS\_Hat = meshgrid(gBphiHat , gQ\_Hat);

% gT\_hat(1)= 1/(2\*gLambdahat(1));

% for j = 2: length(gLambdahat)

% gT\_hat(j) = gT\_hat(j-1) + 1/gLambdahat(j);

% end

% %T\_hat = T\_hat + 1/(2\*lambdahat(length(lambdahat)));

% gDTau = ones(size(gT\_hat))/gT\_hat;

% %H = reshape(H\_theta\_hat, length(Lambdahat),length(Lambdahat));

% %C = mean(mean(S\_Hat,1),2);

% %c = abs(abs(C/max(abs(C)))-1)\*255;

%

% % [dxx, dyy] = gradient(gH\_Theta\_Hat);

figure,

subplot(3,2,1)

surf(XHat(1:T-1),YHat(1:T-1), H\_Theta\_Hat)

hold on

%surf(MHat(Tb:-1:2),NHat(Tb:-1:2), gH\_Theta\_Hat(Tb-1:-1:1, Tb-1:-1:1))

%hold on

contour(XHat(1:T-1),YHat(1:T-1),H\_Theta\_Hat, 'showtext', 'on')

%hold on

%contour(MHat(Tb:-1:2),NHat(Tb:-1:2), gH\_Theta\_Hat, 'showtext', 'on')

title('position')

subplot(3,2,2)

surf(XHat(1:T-1),YHat(1:T-1), Q\_Hat)

hold on

contour(XHat(1:T-1),YHat(1:T-1), Q\_Hat, 'showtext', 'on')

title('momentum degradation')

%subplot(4,2,3)

%hold on

%surf(MHat(Tb:-1:2),NHat(Tb:-1:2), gQ\_Hat)

%hold on

%contour(MHat(Tb:-1:2),NHat(Tb:-1:2),gQ\_Hat, 'showtext', 'on')

%title('momentum reprogram')

subplot(3,2,3)

plot3(XHat(1:T-1),YHat(1:T-1), S\_Hat)

hold on

plot3(XHat(1:T-1),YHat(1:T-1), YHat(1:T-1)./y(1:T-1))

%hold on

%plot3(MHat(Tb:-1:2),NHat(Tb:-1:2), gS\_Hat(1,Tb-1:-1:1))

%hold on

%plot3(MHat(Tb:-1:2),NHat(Tb:-1:2), NHat(Tb:-1:2)./n(Tb:-1:2))

title('action potential S: rate and residule ratio')

subplot(3,2,4)

plot3(XHat(1:T-1),YHat(1:T-1),Q\_Hat + H\_Theta\_Hat)

%hold on

%plot3(MHat(Tb:-1:2),NHat(Tb:-1:2),gQ\_Hat + gH\_Theta\_Hat)

title('Total Hamilton')

subplot(3,2,5)

surf(meshgrid(Q\_Hat,H\_Theta\_Hat))

hold on

contour(meshgrid(Q\_Hat,H\_Theta\_Hat)-min(min(meshgrid(Q\_Hat,H\_Theta\_Hat))), 'showtext', 'on')

title('Hamilton')

%surf(meshgrid(gQ\_Hat,gH\_Theta\_Hat))

%hold on

%contour(meshgrid(gQ\_Hat,gH\_Theta\_Hat)-min(min(meshgrid(gQ\_Hat,gH\_Theta\_Hat))), 'showtext', 'on')

%title('Hamilton')

subplot(3,2,6)

plot(x',y)

xlabel('mRNA numbers')

ylabel('protein numbers')

hold on

plot(x', YHat)

%hold on

%plot(m(Tb:-1:2)',NHat(Tb:-1:2))

%xlabel('mRNA numbers')

%ylabel('protein numbers')

%title('fitted proteins and predicted proteins')

Res = abs(alphachat(1:T-1)-lambdahat(1:T-1));

Cost = T\_hat.\*log(T\_hat);

%H(:,:,1:size(H,3))

dTau = tau\_hat;

end