

```

1 function [Dist,D,k,w]=dtw(t,r)
2 %Dynamic Time Warping Algorithm
3 %Dist is unnormalized distance between t and r
4 %D is the accumulated distance matrix
5 %k is the normalizing factor
6 %w is the optimal path
7 %t is the vector you are testing against
8 %r is the vector you are testing
9 D1=imread('001','tif');D2=imread('002','tif');D3=imread('003','tif');D4=imread('004','tif');D5=imread('005','tif');D6=imread('006','tif');D7=imread('007','tif');D8=imread('008','tif');D9=imread('009','tif');D10=imread('010','tif');D11=imread('011','tif');D12=imread('012','tif');D13=imread('013','tif');D14=imread('014','tif');D15=imread('015','tif');D16=imread('016','tif');D17=imread('017','tif');D18=imread('018','tif');D19=imread('019','tif');D20=imread('020','tif');D21=imread('021','tif');D22=imread('022','tif');D23=imread('023','tif');D24=imread('024','tif');D25=imread('025','tif');D26=imread('026','tif');D27=imread('027','tif');D28=imread('028','tif');D29=imread('029','tif');D30=imread('030','tif');D31=imread('031','tif');D32=imread('032','tif');D33=imread('033','tif');D34=imread('034','tif');D35=imread('035','tif');D36=imread('036','tif');D37=imread('037','tif');D38=imread('038','tif');D39=imread('039','tif');D40=imread('040','tif');D41=imread('041','tif');D42=imread('042','tif');D43=imread('043','tif');D44=imread('044','tif');D45=imread('045','tif');D46=imread('046','tif');
10 %this is reading the images as matrix, for the MOD09A1 dataset, a total of 46 images was readed,
11 %assume that the images are n*m
12 for i=1:n
13     for j=1:m
14         t=[D1(i,j),D2(i,j),D3(i,j),D4(i,j),D5(i,j),D6(i,j),D7(i,j),D8(i,j),D9(i,j),D10(i,j),D11(i,j),D12(i,j),D13(i,j),D14(i,j),D15(i,j),D16(i,j),D17(i,j),D18(i,j),D19(i,j),D20(i,j),D21(i,j),D22(i,j),D23(i,j),D24(i,j),D25(i,j),D26(i,j),D27(i,j),D28(i,j),D29(i,j),D30(i,j),D31(i,j),D32(i,j),D33(i,j),D34(i,j),D35(i,j),D36(i,j),D37(i,j),D38(i,j),D39(i,j),D40(i,j),D41(i,j),D42(i,j),D43(i,j),D44(i,j),D45(i,j),D46(i,j)];
15     %constructing the NDVI time-series
16     r=[0.151144,0.0475944,0.175165,0.157775,0.164598,0.172529,0.155204,0.142113,0.136211,0.145350,0.140974,0.133808,0.133333,0.150101,0.182812,0.182927,0.242059,0.204676,0.238500,0.260517,0.276132,0.331067,0.406653,0.440842,0.415762,0.609840,0.694531,0.326094,0.391149,0.334303,0.409154,0.439298,0.402376,0.385945,0.341566,0.338080,0.308387,0.282548,0.265436,0.257268,0.233965,0.165222,0.190283,0.0517611,0.197170,0.040494];
17     %r is the standard NDVI time-series value (get from the sampled data)
18     %now perform the DTW function
19     [rows,N]=size(t);
20     [rows,M]=size(r);
21     %for n=1:N
22     %     for m=1:M
23     %         d(n,m)=abs(t(n)-r(m));
24     %     end
25     %end
26     d=abs(repmat(t(:),1,M)-repmat(r(:)',N,1))/10; %this replaces the nested for loops from above
27 Thanks Georg Schmitz

```

```

45 %d is abs DTW value divided 10 for the convenience of results outputs
46 D=zeros(size(d));
47 D(1,1)=d(1,1);
48
49 for n=2:N
50     D(n,1)=d(n,1)+D(n-1,1);
51 end
52 for m=2:M
53     D(1,m)=d(1,m)+D(1,m-1);
54 end
55 for n=2:N
56     for m=2:M
57         D(n,m)=d(n,m)+min([D(n-1,m),D(n-1,m-1),D(n,m-1)]);
58     end
59 end
60 Dist=D(N,M);
61 n=N;
62 m=M;
63 k=1;
64 w=[];
65 w(1,:)=[N,M];
66 while ((n+m)~=2)
67     if (n-1)==0
68         m=m-1;
69     elseif (m-1)==0
70         n=n-1;
71     else
72         [values,number]=min([D(n-1,m),D(n,m-1),D(n-1,m-1)]);
73         switch number
74             case 1
75                 n=n-1;
76             case 2
77                 m=m-1;
78             case 3
79                 n=n-1;
80                 m=m-1;
81         end
82     end
83     k=k+1;
84     w=cat(1,w,[n,m]);
85 end
86 new(i,j)=Dist;
87 end
88 end

```

```
89  A=im2uint16(new);%the results image DN value are hexadecimal, and the results should divide
90  by 6553.6 to get the final results of DTW distance
91  imwrite(A,'DTWDIST.tif','tif');%produce DTW distance image, as shown in Figure 6
```