```
function out=PreDREEGplot(dreegstruc,out,N)
%PREDREEGplot Calculates 3D tSNE calculations and settings for the output of DREEG
%
% Out can be blank or a previous generated out structure
% N (default = 1000) is number of iterations of tSNE
%
%   Authors:
%       Michael Caiola (Michael.Caiola@fda.hhs.gov)
%       Meijun Ye (Meijun.Ye@fda.hhs.gov)
%
%   Disclaimer: This software and documentation (the "Software") were
%   developed at the Food and Drug Administration (FDA) by employees of
%   the Federal Government in the course of their official duties.
%   Pursuant to Title 17, Section 105 of the United States Code,
%   this work is not subject to copyright protection and is in the
%   public domain. Permission is hereby granted, free of charge, to any
%   person obtaining a copy of the Software, to deal in the Software
%   without restriction, including without limitation the rights to
%   use, copy, modify, merge, publish, distribute, sublicense, or sell
%   copies of the Software or derivatives, and to permit persons to
%   whom the Software is furnished to do so. FDA assumes no
%   responsibility whatsoever for use by other parties of the Software,
%   its source code, documentation or compiled executables, and makes
%   no guarantees, expressed or implied, about its quality,
%   reliability, or any other characteristic. Further, use of this code
%   in no way implies endorsement by the FDA or confers any advantage
%   in regulatory decisions. Although this software can be
%   redistributed and/or modified freely, we ask that any derivative
%   works bear some notice that they are derived from it, and any
%   modified versions bear some notice that they have been modified.
%   The Software is not intended to make clinical diagnoses or to be
%   used in any way to diagnose or treat subjects for whom the EEG is
%   taken.

if nargin<3
    N=1000;
end
%remake g
[a,~]=find(isnan(dreegstruc.F));
dreegstruc.names(a,:)=[];
g=zeros(size(dreegstruc.names,1),1);
[~,gin]=unique(dreegstruc.names(:,1),'stable');
for i=1:length(gin)-1
    g(gin(i):gin(i+1)-1)=i;
end
g(gin(end):end)=i+1;
if or(nargin==1, isempty(out))
    D=parallel.pool.DataQueue;
    h = waitbar(0, 'Doing Stuff');
    afterEach(D, @nUpdateWaitbar);
    p=1;
```

```
    %reduce
    n=dreegstruc.min;
    F=dreegstruc.F;
    if size(F,1)==380
        F=F';
    end
    F(a,:)=[];
    loss=zeros(N,1);
    r=zeros(size(F,1),3,N);

    parfor i=1:N
        r(:,:,i)=1e-4*randn(size(F,1),3);
        [~,loss(i)]=tsne(F,"NumDimensions",3,'Standardize',true,'InitialY',r(:,:,i));
        send(D, i);
        disp("tsne computed!")
    end
    close(h)
    [~,minin]=min(loss);
    options=statset('MaxIter',1000,'TolFun',1e-7);
    [Z,minloss]=tsne(F,"NumDimensions",3,'Algorithm','Exact',...
        'Standardize',true,'InitialY',r(:,:,minin),'Options',options,'Verbose',1);
    %plot

    %remove outliers
    bnd=[mean(Z)-std(Z);mean(Z)+std(Z)];
    onesdidx=zeros(size(Z,1),1);
    for i=1:3
        onesdidx=onesdidx+or(Z(:,i)<bnd(1,i),Z(:,1)>bnd(2,i));
    end
    onesd=dreegstruc.names(~onesdidx,:);
    bnd=[mean(Z)-2*std(Z);mean(Z)+2*std(Z)];
    twosdidx=zeros(size(Z,1),1);
    for i=1:3
        twosdidx=twosdidx+or(Z(:,i)<bnd(1,i),Z(:,1)>bnd(2,i));
    end
    twosd=dreegstruc.names(~twosdidx,:);
    %stats
    subjects=split(dreegstruc.names(:,1),'\');
    subjects=double(subjects(:,7));
    T=table('Size',[3,16],'VariableTypes',["string","double","double","double",...
        "double","double","double","double","double","double","double","double"...
        ,"double","double","double","double"],'VariableNames',["Data","Subjects",...
        string(num2str(n))+"-min Recordings","1","2","3","4","5","6","7","8"...
        ,"9","10","Avg # of Rec","Std # of Rec","Med # of Rec"]);
    T{:,1}=["Original";"1std";"2std"];
    T{:,2}=[numel(unique(subjects));numel(unique(subjects(~onesdidx)));numel(unique↙
(subjects(~twosdidx)))];
    T{:,3}=[length(dreegstruc.names);sum(~onesdidx);sum(~twosdidx)];
    [~,a1]=unique(subjects(~onesdidx),'stable');
    [~,a2]=unique(subjects(~twosdidx),'stable');
    [~,a0]=unique(subjects,'stable');
```

```
    a=diff([a0;T{1,3}+1]);
    b=diff([a1;T{2,3}+1]);
    c=diff([a2;T{3,3}+1]);
    for i=1:10
        T{:,i+3}=[sum(a==i);sum(b==i);sum(c==i)];
    end
    T{:,14}=[mean(a);mean(b);mean(c)];
    T{:,15}=[std(a);std(b);std(c)];
    T{:,16}=[median(a);median(b);median(c)];
    %output
    out.numsubjects=numel(unique(subjects));
    out.allnames=dreegstruc.names;
    out.onesd=onesd;
    out.twosd=twosd;
    out.stats=T;
    out.persubject1=T{2,2}/T{1,2};
    out.perrecording1=T{2,3}/T{1,3};
    out.persubject2=T{3,2}/T{1,2};
    out.perrecording2=T{3,3}/T{1,3};
    out.minloss=minloss;
    out.minloss_approx=loss(minin);
    out.Z=Z;
    out.F=F;
else
    Z=out.Z;
end
c=lines(max(g));
figure;
scatter3(Z(:,1),Z(:,2),Z(:,3),15,c(g,:),'filled');
title('tSNE')
hold on
scatter3(mean(Z(:,1)),mean(Z(:,2)),mean(Z(:,3)),50,'k','filled');
%plotcube(2*std(Z),mean(Z)-std(Z),.1);
%plotcube(4*std(Z),mean(Z)-2*std(Z),.1);
%%
 function nUpdateWaitbar(~)
        waitbar(p/N, h);
        p = p + 1;
 end
end
```