如何将MIT-BIH数据库里的dat、hea、atr数据代入matlab程序进行读取画图?

Mr.Snow

2 人赞同了该回答

```
% This programm reads ECG data which are saved in format 212.
% (e.g., 100.dat from MIT-BIH-DB, cu01.dat from CU-DB,...)
% The data are displayed in a figure together with the annotations.
% The annotations are saved in the vector ANNOT, the corresponding
% times (in seconds) are saved in the vector ATRTIME.
% The annotations are saved as numbers, the meaning of the numbers can
% be found in the codetable "ecgcodes.h" available at physionet.org.
%
% ANNOT only contains the most important information, which is displayed
% with the program rdann (available on physionet.org) in the 3rd row.
% The 4th to 6th row are not saved in ANNOT.
%
%
%    created on Feb. 27, 2003 by
%    Robert Tratnig (Vorarlberg University of Applied Sciences)
%    (email: rtratnig@gmx.at),
%
%    algorithm is based on a program written by
%    Klaus Rheinberger (University of Innsbruck)
%    (email: klaus.rheinberger@uibk.ac.at)
%


%------ SPECIFY DATA ------------------------------------------------------
%------ 指定数据文件 -------------------------------------------------------
   % .dat 格式，ECG 数据
SAMPLES2READ=2048;     % 指定需要读入的样本数
            % 若.dat文件中存储有两个通道的信号:
            % 则读入 2*SAMPLES2READ 个数据

%------ LOAD HEADER DATA --------------------------------------------------
%------ 读入头文件数据 -----------------------------------------------------
%
% 示例：用记事本打开的117.hea 文件的数据
%
```

```
%      117 2 360 650000
%      117.dat 212 200 11 1024 839 31170 0 MLII
%      117.dat 212 200 11 1024 930 28083 0 V2
%      # 69 M 950 654 x2
%      # None
%
%-----------------------------------------------------------------------
 % 在Matlab命令行窗口提示当前工作状态
%
% 【注】函数 fprintf 的功能将格式化的数据写入到指定文件中。
% 表达式：count = fprintf(fid,format,A,...)
% 在字符串'format'的控制下，将矩阵A的实数数据进行格式化，并写入到文件对象fid中。该函数返回所写入数据的字节数 count。
% fid 是通过函数 fopen 获得的整型文件标识符。fid=1，表示标准输出（即输出到屏幕显示）；fid=2，表示标准偏差。
%
signalh= fullfile(PATH, HEADERFILE);    % 通过函数 fullfile 获得头文件的完整路径
fid1=fopen(signalh,'r');    % 打开头文件，其标识符为 fid1 ，属性为'r'--“只读”
z= fgetl(fid1);             % 读取头文件的第一行数据，字符串格式
A= sscanf(z, '%*s %d %d %d',[1,3]); % 按照格式 '%*s %d %d %d' 转换数据并存入矩阵 A 中
nosig= A(1);    % 信号通道数目
sfreq=A(2);     % 数据采样频率
clear A;        % 清空矩阵 A ，准备获取下一行数据
for k=1:nosig           % 读取每个通道信号的数据信息
    z= fgetl(fid1);
    A= sscanf(z, '%*s %d %d %d %d %d',[1,5]);
    dformat(k)= A(1);           % 信号格式; 这里只允许为 212 格式
    gain(k)= A(2);              % 每 mV 包含的整数个数
    bitres(k)= A(3);            % 采样精度（位分辨率）
    zerovalue(k)= A(4);         % ECG 信号零点相应的整数值
    firstvalue(k)= A(5);        % 信号的第一个整数值 (用于偏差测试)
end;
fclose(fid1);
clear A;

%------ LOAD BINARY DATA -------------------------------------------------
%------ 读取 ECG 信号二值数据 ---------------------------------------------
%
if dformat~= [212,212], error('this script does not apply binary formats different to 212.'); end;
```

```
signald= fullfile(PATH, DATAFILE);           % 读入 212 格式的 ECG 信号数据
fid2=fopen(signald,'r');
A= fread(fid2, [3, SAMPLES2READ], 'uint8')';  % matrix with 3 rows, each 8 bits long, = 2*12bit
fclose(fid2);
% 通过一系列的移位（bitshift）、位与（bitand）运算，将信号由二值数据转换为十进制数
M2H= bitshift(A(:,2), -4);        %字节向右移四位，即取字节的高四位
M1H= bitand(A(:,2), 15);          %取字节的低四位
PRL=bitshift(bitand(A(:,2),8),9);     % sign-bit   取出字节低四位中最高位，向右移九位
PRR=bitshift(bitand(A(:,2),128),5);   % sign-bit   取出字节高四位中最高位，向右移五位
M( : , 1)= bitshift(M1H,8)+ A(:,1)-PRL;
M( : , 2)= bitshift(M2H,8)+ A(:,3)-PRR;
if M(1,:) ~= firstvalue, error('inconsistency in the first bit values'); end;
switch nosig
case 2
   M( : , 1)= (M( : , 1)- zerovalue(1))/gain(1);
   M( : , 2)= (M( : , 2)- zerovalue(2))/gain(2);
   TIME=(0:(SAMPLES2READ-1))/sfreq;
case 1
   M( : , 1)= (M( : , 1)- zerovalue(1));
   M( : , 2)= (M( : , 2)- zerovalue(1));
   M=M';
   M(1)=[];
   sM=size(M);
   sM=sM(2)+1;
   M(sM)=0;
   M=M';
   M=M/gain(1);
   TIME=(0:2*(SAMPLES2READ)-1)/sfreq;
otherwise  % this case did not appear up to now!
   % here M has to be sorted!!!
   disp('Sorting algorithm for more than 2 signals not programmed yet!');
end;
clear A M1H M2H PRR PRL;
fprintf(1,'\\n$> LOADING DATA FINISHED \n');

%------ LOAD ATTRIBUTES DATA ----------------------------------------------
atrd= fullfile(PATH, ATRFILE);      % attribute file with annotation data
fid3=fopen(atrd,'r');
A= fread(fid3, [2, inf], 'uint8')';
```

```
fclose(fid3);
ATRTIME=[];
ANNOT=[];
sa=size(A);
saa=sa(1);
i=1;
while i<=saa
    annoth=bitshift(A(i,2),-2);
    if annoth==59
        ANNOT=[ANNOT;bitshift(A(i+3,2),-2)];
        ATRTIME=[ATRTIME;A(i+2,1)+bitshift(A(i+2,2),8)+...
                bitshift(A(i+1,1),16)+bitshift(A(i+1,2),24)];
        i=i+3;
    elseif annoth==60
        % nothing to do!
    elseif annoth==61
        % nothing to do!
    elseif annoth==62
        % nothing to do!
    elseif annoth==63
        hilfe=bitshift(bitand(A(i,2),3),8)+A(i,1);
        hilfe=hilfe+mod(hilfe,2);
        i=i+hilfe/2;
    else
        ATRTIME=[ATRTIME;bitshift(bitand(A(i,2),3),8)+A(i,1)];
        ANNOT=[ANNOT;bitshift(A(i,2),-2)];
   end;
   i=i+1;
end;
ANNOT(length(ANNOT))=[];       % last line = EOF (=0)
ATRTIME(length(ATRTIME))=[];   % last line = EOF
clear A;
ATRTIME= (cumsum(ATRTIME))/sfreq;
ind= find(ATRTIME <= TIME(end));
ATRTIMED= ATRTIME(ind);
ANNOT=round(ANNOT);
ANNOTD= ANNOT(ind);
```

```
%------ DISPLAY DATA --------------------------------------------------
figure(1); clf, box on, hold on
plot(TIME, M(:,1),'r');
if nosig==2
    plot(TIME, M(:,2),'b');
end;
for k=1:length(ATRTIMED)
    text(ATRTIMED(k),0,num2str(ANNOTD(k)));
end;
xlim([TIME(1), TIME(end)]);
xlabel('Time / s'); ylabel('Voltage / mV');
string=['ECG signal ',DATAFILE];
title(string);
fprintf(1,'\\n$> DISPLAYING DATA FINISHED \n');

% -------------------------------------------------------------------
fprintf(1,'\\n$> ALL FINISHED \n');
```