pre_processing

September 20, 2021

1 Classification of arrhytmia heartbeat

- Paper: https://www.nature.com/articles/s41597-020-0386-x
- Chinese Database: https://physionetchallenges.org/2021/

1.1 Importing packages

[2]:
```
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import glob
import math
import os
import sys
import loess
from scipy.io import loadmat
from scipy import signal
from statsmodels.nonparametric.smoothers_lowess import lowess
```

1.2 Loading records and files

[3]:
```
files = os.listdir(os.path.abspath('Datasets/ECGData'))
```

[4]:
```
df_attrib = pd.read_excel(os.path.abspath('Datasets/AttributesDictionary.xlsx'))
df_condnames = pd.read_excel(os.path.abspath('Datasets/ConditionNames.xlsx'))
df_diagno = pd.read_excel(os.path.abspath('Datasets/Diagnostics.xlsx'))
df_rhyt = pd.read_excel(os.path.abspath('Datasets/RhythmNames.xlsx'))
```

1.3 Describing files

[5]:
```
df_attrib
```

[5]:

	Attributes	Type	ValueRange	\
0	FileName	String	NaN	
1	Rhythm	String	NaN	
2	Beat	String	NaN	
3	PatientAge	Numeric	0-999	

```
4            Gender      String       MALE/FEMAL
5   VentricularRate     Numeric            0-999
6         AtrialRate     Numeric            0-999
7        QRSDuration  Numeric -             0-999
8         QTInterval     Numeric            0-999
9        QTCorrected     Numeric            0-999
10             RAxis     Numeric         -179~180
11             TAxis     Numeric         -179~181
12          QRSCount     Numeric            0-254
13            QOnset     Numeric  16 Bit Unsigned
14           QOffset     Numeric  17 Bit Unsigned
15           TOffset     Numeric  18 Bit Unsigned

                     Description
0   ECG data file name(unique ID)
1                    Rhythm Label
2          Other conditions Label
3                              Age
4                           Gender
5         Ventricular rate in BPM
6              Atrial rate in BPM
7            QRS duration in msec
8             QT interval in msec
9   Corrected QT interval in msec
10                          R axis
11                          T axis
12                       QRS count
13            Q onset(In samples)
14           Q offset(In samples)
15           T offset(In samples)
```

[6]:
```
df_condnames
```

[6]:
```
   Acronym Name                                  Full Name
0          1AVB            1 degree atrioventricular block
1          2AVB            2 degree atrioventricular block
2         2AVB1  2 degree atrioventricular block(Type one)
3         2AVB2  2 degree atrioventricular block(Type two)
4          3AVB            3 degree atrioventricular block
5           ABI                              atrial bigeminy
6           ALS                              Axis left shift
7           APB                     atrial premature beats
8           AQW                             abnormal Q wave
9           ARS                           Axis right shift
10          AVB                     atrioventricular block
11          CCR                countercolockwise rotation
12           CR                       colockwise rotation
```

13	ERV	Early repolarization of the ventricles
14	FQRS	fQRS Wave
15	IDC	Interior differences conduction
16	IVB	Intraventricular block
17	JEB	junctional escape beat
18	JPS	J point shift
19	JPT	junctional premature beat
20	LBBB	left bundle branch block
21	LBBBB	left back bundle branch block
22	LFBBB	left front bundle branch block
23	LRRI	Long RR interval
24	LVH	left ventricle hypertrophy
25	LVHV	left ventricle high voltage
26	LVQRSAL	lower voltage QRS in all lead
27	LVQRSCL	lower voltage QRS in chest lead
28	LVQRSLL	lower voltage QRS in limb lead
29	MI	myocardial infarction
30	MIBW	myocardial infraction in back wall
31	MIFW	Myocardial infgraction in the front wall
32	MILW	Myocardial infraction in the lower wall
33	MISW	Myocardial infraction in the side wall
34	PRIE	PR interval extension
35	PWC	P wave Change
36	QTIE	QT interval extension
37	RAH	right atrial hypertrophy
38	RAHV	right atrial high voltage
39	RBBB	right bundle branch block
40	RVH	right ventricle hypertrophy
41	STDD	ST drop down
42	STE	ST extension
43	STTC	ST-T Change
44	STTU	ST tilt up
45	TWC	T wave Change
46	TWO	T wave opposite
47	UW	U wave
48	VB	ventricular bigeminy
49	VEB	ventricular escape beat
50	VFW	ventricular fusion wave
51	VPB	ventricular premature beat
52	VPE	ventricular preexcitation
53	VET	ventricular escape trigeminy
54	WAVN	Wandering in the atrioventricalualr node
55	WPW	WPW

[7]:
```
df_rhyt
```

[7]:

	Acronym Name	Full Name
0	SB	Sinus Bradycardia
1	SR	Sinus Rhythm
2	AFIB	Atrial Fibrillation
3	ST	Sinus Tachycardia
4	AF	Atrial Flutter
5	SI	Sinus Irregularity
6	SVT	Supraventricular Tachycardia
7	AT	Atrial Tachycardia
8	AVNRT	Atrioventricular Node Reentrant Tachycardia
9	AVRT	Atrioventricular Reentrant Tachycardia
10	SAAWR	Sinus Atrium to Atrial Wandering Rhythm

[8]:
```
df_diagno
```

[8]:

	FileName	Rhythm	Beat	PatientAge	Gender	\
0	MUSE_20180113_171327_27000	AFIB	RBBB TWC	85	MALE	
1	MUSE_20180112_073319_29000	SB	TWC	59	FEMALE	
2	MUSE_20180111_165520_97000	SA	NONE	20	FEMALE	
3	MUSE_20180113_121940_44000	SB	NONE	66	MALE	
4	MUSE_20180112_122850_57000	AF	STDD STTC	73	FEMALE	
…	…	…	…	…	…	
10641	MUSE_20181222_204306_99000	SVT	NONE	80	FEMALE	
10642	MUSE_20181222_204309_22000	SVT	NONE	81	FEMALE	
10643	MUSE_20181222_204310_31000	SVT	NONE	39	MALE	
10644	MUSE_20181222_204312_58000	SVT	NONE	76	MALE	
10645	MUSE_20181222_204314_78000	SVT	NONE	75	MALE	

	VentricularRate	AtrialRate	QRSDuration	QTInterval	QTCorrected	\
0	117	234	114	356	496	
1	52	52	92	432	401	
2	67	67	82	382	403	
3	53	53	96	456	427	
4	162	162	114	252	413	
…	…	…	…	…	…	
10641	196	73	168	284	513	
10642	162	81	162	294	482	
10643	152	92	152	340	540	
10644	175	178	128	310	529	
10645	117	104	140	312	435	

	RAxis	TAxis	QRSCount	QOnset	QOffset	TOffset
0	81	-27	19	208	265	386
1	76	42	8	215	261	431
2	88	20	11	224	265	415
3	34	3	9	219	267	447
4	68	-40	26	228	285	354

```
…       …    …        …      …       …       …
10641    258    244        32     177      261      319
10642    110    -75        27     173      254      320
10643    250     38        25     208      284      378
10644     98    -83        29     205      269      360
10645    263    144        19     208      278      364

[10646 rows x 16 columns]
```

1.4 Checking NaN values

[9]:
```
df_diagno.isnull().sum()
```

[9]:
```
FileName           0
Rhythm             0
Beat               0
PatientAge         0
Gender             0
VentricularRate    0
AtrialRate         0
QRSDuration        0
QTInterval         0
QTCorrected        0
RAxis              0
TAxis              0
QRSCount           0
QOnset             0
QOffset            0
TOffset            0
dtype: int64
```

[10]:
```
df_diagno['Rhythm'].unique()
```

[10]:
```
array(['AFIB', 'SB', 'SA', 'AF', 'SR', 'ST', 'SVT', 'AT', 'AVNRT',
       'SAAWR', 'AVRT'], dtype=object)
```

[11]:
```
df_diagno.groupby(['Rhythm']).count()['FileName']
```

[11]:
```
Rhythm
AF        445
AFIB     1780
AT        121
AVNRT      16
AVRT        8
SA        399
SAAWR       7
SB       3889
```

```
SR       1826
ST       1568
SVT       587
Name: FileName, dtype: int64
```

1.5 Grouping rhythms

[ ]:
```
df_1 = df_diagno[df_diagno['Rhythm'].isin(['AF','AFIB'])]
df_1['Rhythm_grouped'] = ['AFIB' for _ in range(len(df_1))]

df_2 = df_diagno[df_diagno['Rhythm'].isin(['SVT','AT',␣
↪'SA','SAAWR','ST','AVNRT','AVRT'])]
df_2['Rhythm_grouped'] = ['GSVT' for _ in range(len(df_2))]

df_3 = df_diagno[df_diagno['Rhythm'].isin(['SB'])]
df_3['Rhythm_grouped'] = ['SB' for _ in range(len(df_3))]

df_4 = df_diagno[df_diagno['Rhythm'].isin(['SR','SI'])]
df_4['Rhythm_grouped'] = ['SR' for _ in range(len(df_4))]

df_total = [df_1,df_2,df_3,df_4]
df = pd.concat(df_total)
```

[13]:
```
df.groupby(['Rhythm_grouped']).count()['FileName']
```

[13]:
```
Rhythm_grouped
AFIB    2225
GSVT    2706
SB      3889
SR      1826
Name: FileName, dtype: int64
```

[48]:
```
df
```

[48]:

	FileName	Rhythm	Beat	PatientAge	Gender
0	MUSE_20180113_171327_27000	AFIB	RBBB TWC	85	MALE
4	MUSE_20180112_122850_57000	AF	STDD STTC	73	FEMALE
6	MUSE_20180114_075026_69000	AFIB	TWC	80	FEMALE
15	MUSE_20180113_133901_16000	AFIB	STTC	67	FEMALE
22	MUSE_20180116_123940_90000	AFIB	TWC	81	MALE
…	…	…	…	…	…
9878	MUSE_20180209_123628_18000	SR	NONE	66	MALE
9879	MUSE_20180209_122145_99000	SR	NONE	63	FEMALE
9887	MUSE_20180209_132636_13000	SR	NONE	46	MALE
9888	MUSE_20180210_123100_20000	SR	TWC	57	FEMALE
9891	MUSE_20180209_132228_29000	SR	NONE	71	FEMALE

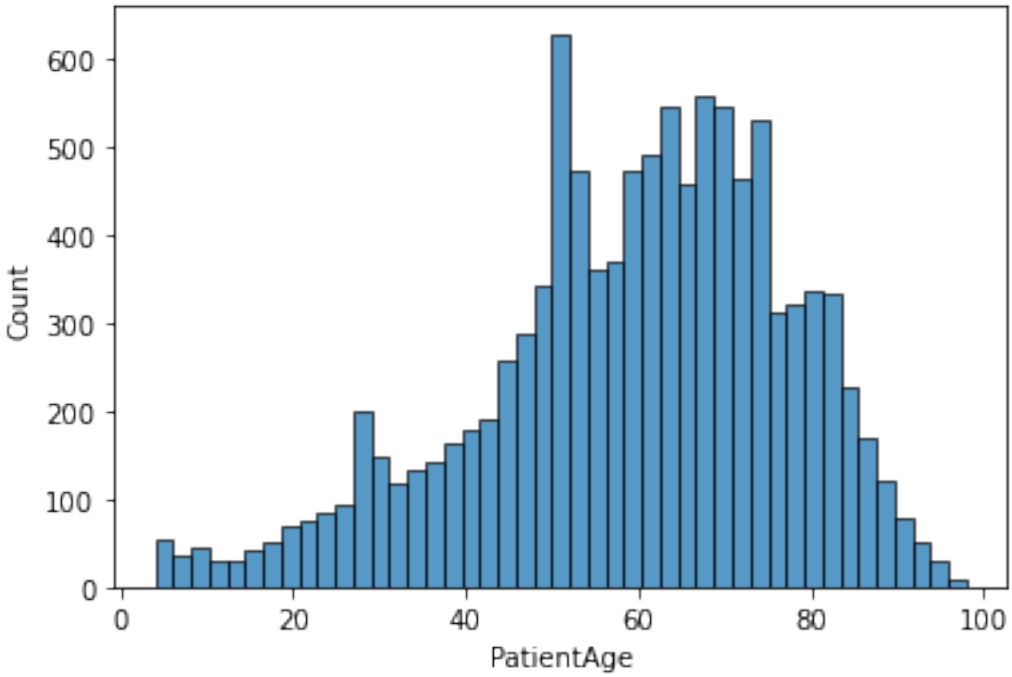
```
      VentricularRate  AtrialRate  QRSDuration  QTInterval  QTCorrected  \
0                 117         234          114         356          496
4                 162         162          114         252          413
6                  98          86           74         360          459
15                 72          65           90         416          455
22                 79         150           92         404          463
…                 …           …            …           …            …
9878               73          73           96         416          458
9879               72          72           72         402          440
9887               78          78          100         348          396
9888               68          68           96         400          425
9891               90          90           68         360          440

      RAxis  TAxis  QRSCount  QOnset  QOffset  TOffset Rhythm_grouped
0        81    -27        19     208      265      386          AFIB
4        68    -40        26     228      285      354          AFIB
6        69     83        17     215      252      395          AFIB
15       -1    -15        12     228      273      436          AFIB
22      -15    -51        13     218      264      420          AFIB
…     …    …       …      …       …      …            …
9878     43     42        12     209      257      417            SR
9879     13     45        12     214      250      415            SR
9887     73     73        13     218      268      392            SR
9888     18     60        11     210      258      410            SR
9891     70     48        15     226      260      406            SR

[10646 rows x 17 columns]
```

[14]:
```
df_diagno['Gender'].unique()
```

[14]:
```
array(['MALE', 'FEMALE'], dtype=object)
```

[15]:
```
sns.histplot(df_diagno['PatientAge'])
```

[15]:
```
<AxesSubplot:xlabel='PatientAge', ylabel='Count'>
```



1.6 Loading a patient's ECG

[38]:
```
patient_denoised = pd.read_csv('Datasets/ECGDataDenoised/'+ files[0])
patient_denoised.columns =␣
 ↪['I','II','III','aVR','aVL','aVF','V1','V2','V3','V4','V5','V6' ]
patient_denoised
```

[38]:

	I	II	III	aVR	aVL	aVF	V1	V2	\
0	-150.75	-336.810	-114.980	251.410	-23.8830	-217.52	441.74	646.47	
1	-136.69	-315.560	-108.630	233.450	-19.1260	-204.36	436.06	656.31	
2	-123.74	-296.230	-103.090	217.230	-14.8150	-192.29	432.27	666.14	
3	-112.57	-279.750	-98.611	203.530	-11.2130	-181.87	431.08	676.31	
4	-103.74	-266.700	-95.229	192.840	-8.5498	-173.50	432.76	687.31	
…	…	…	…	…	…	…	…	…	
4994	-190.32	52.396	170.760	74.543	-170.8900	107.63	310.73	398.59	
4995	-178.98	65.438	171.870	62.325	-165.6500	114.41	302.32	398.80	
4996	-171.04	74.181	172.080	53.876	-161.4900	118.93	297.25	401.02	
4997	-165.46	79.570	171.300	48.203	-157.8400	121.64	293.97	403.86	
4998	-160.69	83.461	169.810	43.624	-154.1300	123.48	290.95	406.44	

	V3	V4	V5	V6
0	642.56	35.389	-269.510	-532.21
1	665.95	76.572	-225.830	-495.39
2	688.05	113.510	-186.000	-461.84

```
3     708.47  144.300 -151.680 -432.56
4     726.98  167.550 -124.030 -408.07
…       …       …       …       …
4994  538.11  230.970   87.684 -249.88
4995  541.18  245.270  110.940 -226.42
4996  544.16  254.480  129.340 -207.20
4997  546.18  259.510  144.140 -190.64
4998  547.24  262.570  157.620 -174.67

[4999 rows x 12 columns]
```

[50]:
```
patient_normal = pd.read_csv('Datasets/ECGData/'+ files[0])
patient_normal
```

[50]:
```
            I      II     III     aVR     aVL     aVF      V1      V2      V3  \
0    -214.72 -229.36  -14.64  224.48 -102.48 -122.00  614.88  814.96  912.56
1    -200.08 -209.84   -9.76  204.96  -97.60 -112.24  605.12  819.84  927.20
2    -190.32 -195.20   -4.88  195.20  -92.72 -102.48  600.24  829.60  956.48
3    -165.92 -165.92    0.00  165.92  -82.96  -82.96  590.48  844.24  971.12
4    -161.04 -156.16    4.88  161.04  -82.96  -78.08  585.60  844.24  985.76
…       …       …       …       …       …       …       …       …       …
4995 -117.12    9.76  126.88   53.68 -122.00   68.32  439.20  634.40  800.32
4996 -102.48   14.64  117.12   43.92 -112.24   63.44  424.56  634.40  819.84
4997  -92.72   29.28  122.00   34.16 -107.36   73.20  424.56  644.16  829.60
4998  -92.72   34.16  126.88   29.28 -112.24   78.08  424.56  644.16  829.60
4999  -87.84   29.28  117.12   29.28 -102.48   73.20  424.56  653.92  834.48

          V4      V5      V6
0     126.88 -239.12 -507.52
1     165.92 -195.20 -463.60
2     204.96 -151.28 -429.44
3     239.12 -117.12 -400.16
4     273.28  -78.08 -370.88
…       …       …       …
4995  278.16    0.00 -273.28
4996  297.68   24.40 -248.88
4997  312.32   43.92 -229.36
4998  312.32   58.56 -204.96
4999  312.32   63.44 -200.08

[5000 rows x 12 columns]
```

1.7 Records that do not meet the premises below are excluded.

- lead II = lead I + lead II
- lead aVR + aVL + aVF = 0

Authors: "It is well known that the right hand electrode and left hand electrode could have their

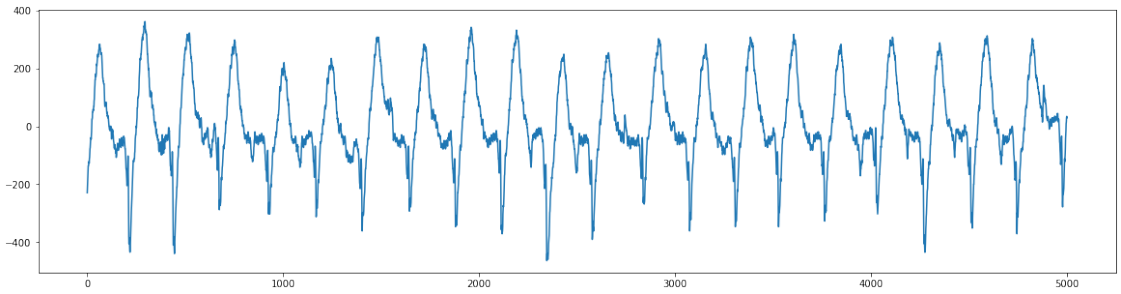
positions switched by operators without a change on corresponding ECG data. Moreover, some of the electrodes could slip off during the test resulting in ECGs displaying a straight line."

[40]:
```
#for i in range(len(patient1)):
#    if (patient1.II[i] != (patient1.I[i] + patient1.III[i])) or ((patient1.
 ↪aVR[i] + patient1.aVL[i] + patient1.aVF[i]) != 0):
#        patient1.drop(i, inplace=True)
#patient1.reset_index(drop=True, inplace=True)
```

[41]:
```
#patient1.reset_index(drop=True, inplace=True)
#patient1
```

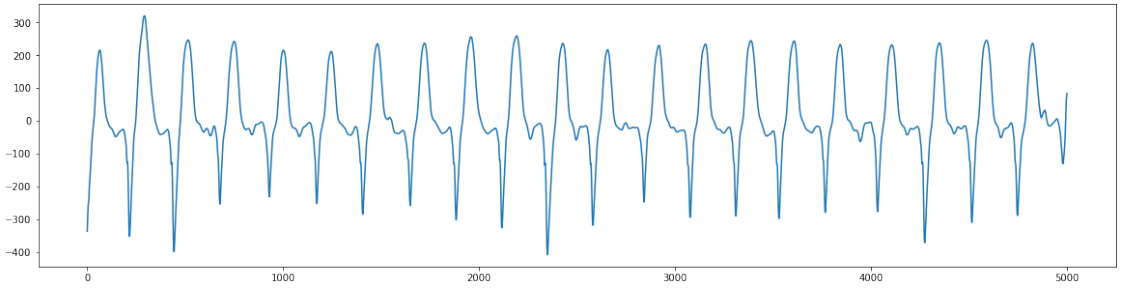
[51]:
```
plt.figure(figsize=(20,5))
patient_normal['II'].plot()
```

[51]: <AxesSubplot:>



[43]:
```
plt.figure(figsize=(20,5))
patient_denoised['II'].plot()
```

[43]: <AxesSubplot:>



1.8 Denoise signals

- Butterworth low pass filter

- LOcally WEighted Scatterplot Smoothing (LOWESS) curve fitting
- Non local means(NLM)

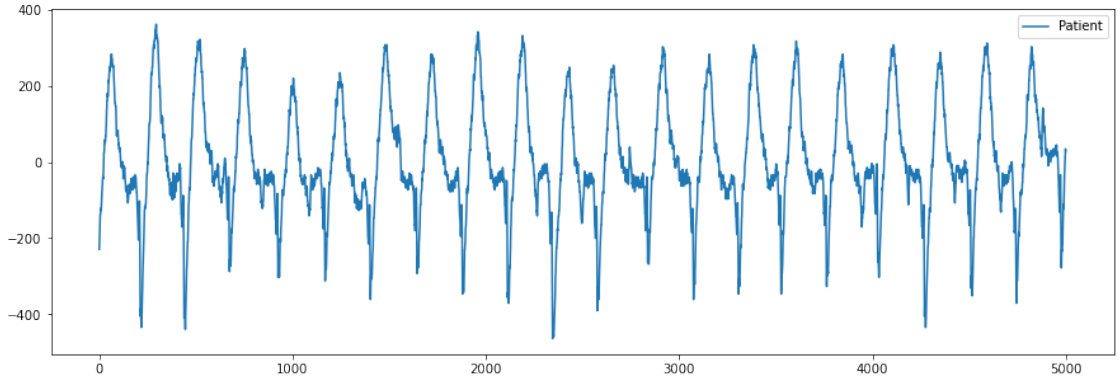
[44]:
```
def NLM_1dDarbon(signal,Nvar,P,PatchHW):
    if isinstance(P,int): # scalar has been entered; expand into patch sample␣
↪index vector
        P = P-1 #Python start index from 0
        Pvec = np.array(range(-P,P+1))
    else:
        Pvec = P # use the vector that has been input
    signal = np.array(signal)
    #debug = [];
    N = len(signal)

    denoisedSig = np.empty(len(signal)) #NaN * ones(size(signal));
    denoisedSig[:] = np.nan
    # to simpify, don't bother denoising edges
    iStart = PatchHW+1
    iEnd = N - PatchHW
    denoisedSig[iStart: iEnd] = 0

    #debug.iStart = iStart;
    #debug.iEnd = iEnd;

    # initialize weight normalization
    Z = np.zeros(len(signal))
    cnt = np.zeros(len(signal))

    # convert lambda value to  'h', denominator, as in original Buades papers
    Npatch = 2 * PatchHW + 1
    h = 2 * Npatch * Nvar**2

    for idx in Pvec: # loop over all possible differences: s - t
        # do summation over p - Eq.3 in Darbon
        k = np.array(range(N))
        kplus = k + idx
        igood = np.where((kplus >=0) & (kplus < N)) # ignore OOB data; we could␣
↪also handle it
        SSD = np.zeros(len(k))
        SSD[igood] = (signal[k[igood]] - signal[kplus[igood]])**2
        Sdx = np.cumsum(SSD)

        for ii in range(iStart,iEnd): # loop over all points 's'
            distance = Sdx[ii + PatchHW] - Sdx[ii - PatchHW-1] #Eq 4;this is in␣
↪place of point - by - point MSE
            # but note the - 1; we want to icnlude the point ii - iPatchHW
```

```
            w = math.exp(-distance/h) # Eq 2 in Darbon
            t = ii + idx # in the papers, this is not made explicit

            if t>0 and t<N:
                denoisedSig[ii] = denoisedSig[ii] + w * signal[t]
                Z[ii] = Z[ii] + w
                #cnt[ii] = cnt[ii] + 1
                #print('ii',ii)
                #print('t',t)
                #print('w',w)
                #print('denoisedSig[ii]', denoisedSig[ii])
                #print('Z[ii]',Z[ii])
     # loop over shifts

    # now apply normalization
    denoisedSig = denoisedSig/(Z + sys.float_info.epsilon)
    denoisedSig[0: PatchHW+1] =signal[0: PatchHW+1]
    denoisedSig[ - PatchHW: ] =signal[- PatchHW: ]
    #debug.Z = Z;

    return denoisedSig   #,debug
```

[52]:
```
Fs=500
fp=50
fs=60
rp=1
rs=2.5
wp=fp/(Fs/2)
ws=fs/(Fs/2)

[n,wn]= signal.buttord(wp,ws,rp,rs)
[bz,az] = signal.butter(n,wn)
LPassDataFile= signal.filtfilt(bz,az,patient_normal['II'])

t = range(0,len(LPassDataFile))
yy2 = lowess(t,LPassDataFile,frac=0.1)
#yy2 = signal.savgol_filter(LPassDataFile,2,1)
BWRemoveDataFile = (LPassDataFile-yy2[:,1])
#BWRemoveDataFile = (LPassDataFile-yy2)
Dl1=BWRemoveDataFile

for k in range(2,len(Dl1)-1):
    Dl1[k]=(2*Dl1[k]-Dl1[k-1]-Dl1[k+1])/math.sqrt(6)

NoisSTD = 1.4826*np.median(np.abs(Dl1-np.median(Dl1)))
Denoising_Patient_1= NLM_1dDarbon(BWRemoveDataFile,(1.5)*(NoisSTD),5000,10)
```

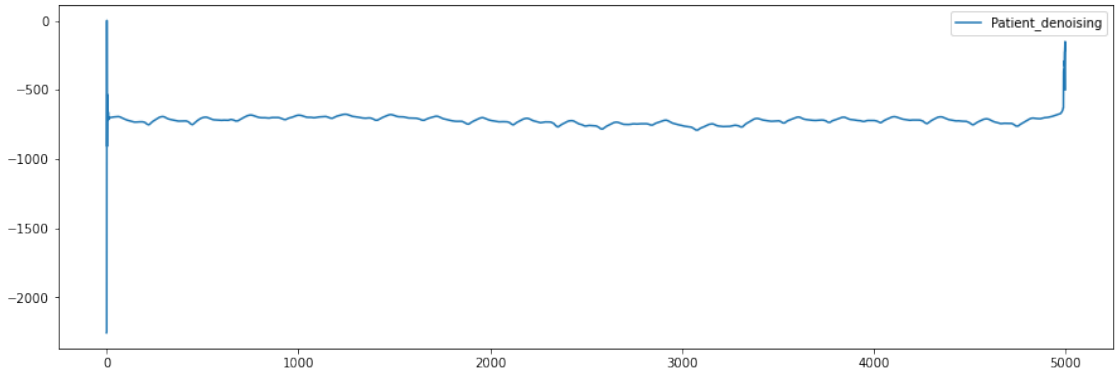
[54]:
```
plt.figure(figsize=(15,5))
#plt.xlim(1000,1500)
patient_normal['II'].plot()
plt.legend(['Patient'])
```

[54]: <matplotlib.legend.Legend at 0x24ea473d3d0>



[55]:
```
plt.figure(figsize=(15,5))
#plt.xlim(1000,1500)
df_denoised = pd.DataFrame(Denoising_Patient_1,columns=['II'])
df_denoised['II'].plot()
plt.legend(['Patient_denoising'])
```

[55]: <matplotlib.legend.Legend at 0x24ea47a90d0>



[ ]: