WH002-read-and-separate

November 4, 2019

[210]:
```
from os import listdir
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.cluster.hierarchy as sch
import pylab
```

[3]:
```
files = listdir('data')
print(files)
```

```
['017.TXT', '074.TXT', '041.TXT', '009.TXT', '014.TXT', '004.TXT', '046.TXT',
'099.TXT', '081.TXT', '065.TXT', '025.TXT', '072.TXT', '042.TXT', '066.TXT',
'024.TXT', '015 coughing after approx 90s.TXT', '028.TXT', '069.TXT', '090.TXT',
'036.TXT', '084.TXT', '022.TXT', '052.TXT', '047.TXT']
```

[4]:
```
with open('data/'+files[0]) as f:
    first_line = f.readline()
    print(first_line)
```

```
-0.210  -0.111  7.010   21.260  10.071  23.000  0.000   0.992
```

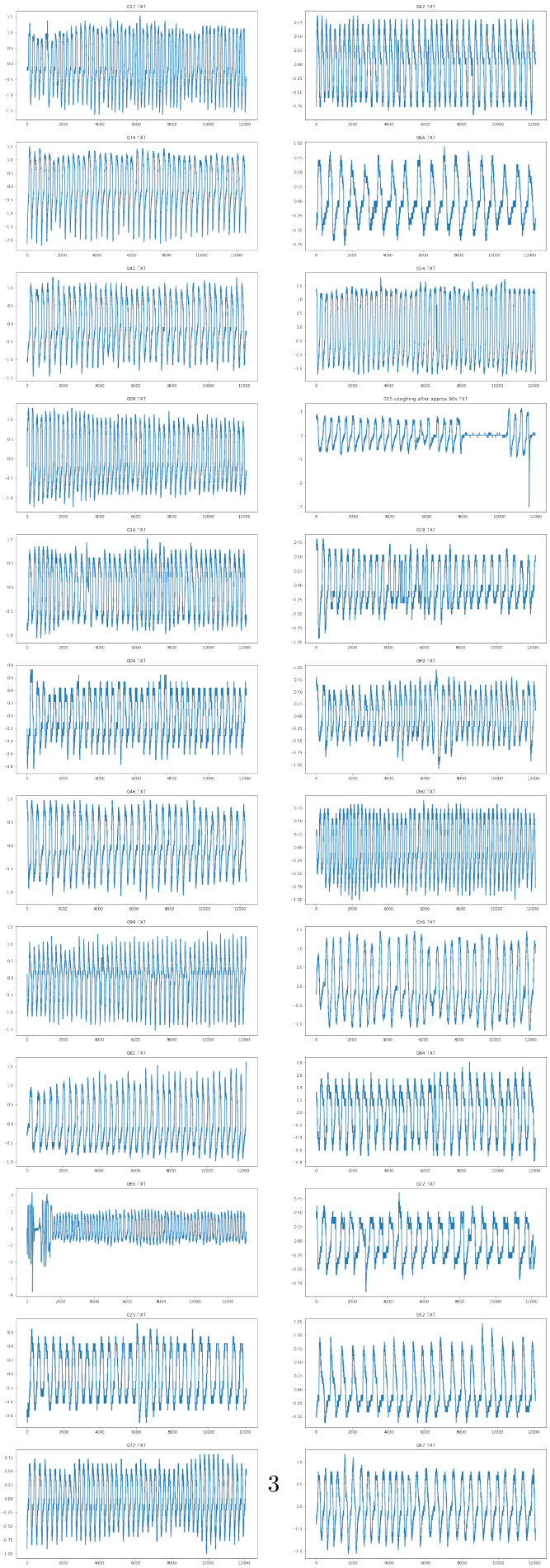
[5]:
```
data = np.genfromtxt('data/'+files[0], delimiter='\t')
```

[6]:
```
data[0]
```

[6]:
```
array([-0.21 , -0.111,  7.01 , 21.26 , 10.071, 23.   ,  0.   ,  0.992])
```

[7]:
```
fig, axs = plt.subplots(len(files)//2,2,figsize=(24, len(files)//2 * 6))
p=0
for i in range (2):
    for j in range(len(files)//2):
        if p < len(files):
            data = np.genfromtxt('data/'+files[p], delimiter='\t')
            print(files[p],data.shape)
            axs[j,i].plot(data[:,0])
            axs[j,i].set_title(files[p])
            p+=1
```

```
017.TXT (12069, 8)
074.TXT (12549, 8)
041.TXT (12090, 8)
009.TXT (12092, 8)
014.TXT (12079, 8)
004.TXT (12076, 8)
046.TXT (12336, 8)
099.TXT (12114, 8)
081.TXT (12168, 8)
065.TXT (13092, 8)
025.TXT (12049, 8)
072.TXT (12107, 8)
042.TXT (12075, 8)
066.TXT (12301, 8)
024.TXT (12082, 8)
015 coughing after approx 90s.TXT (11971, 8)
028.TXT (12155, 8)
069.TXT (12111, 8)
090.TXT (12143, 8)
036.TXT (12261, 8)
084.TXT (12060, 8)
022.TXT (12219, 8)
052.TXT (12099, 8)
047.TXT (12042, 8)
```


017.TXT
042.TXT
074.TXT
066.TXT
041.TXT
024.TXT
009.TXT
015 coughing after approx 90s.TXT
014.TXT
028.TXT
004.TXT
069.TXT
046.TXT
090.TXT
099.TXT
036.TXT
081.TXT
084.TXT
065.TXT
022.TXT
025.TXT
052.TXT
072.TXT
047.TXT

[104]:
```
def find_start(signal_data,pos):
            flag = True
            for i in range(1, 4):
                if signal_data[pos - i] > 0:
                    flag = False
            if flag:
                for i in range(1, 4):
                    if signal_data[pos + i] < 0:
                        flag = False
            return (flag,pos)
```

[105]:
```
def separate_cycles(signal_data):
    data_min = 0
    min_list=[]
    flag = False
    for i in range(signal_data.shape[0]-5):
        if signal_data[i]<0:
            flag,pos = find_start(signal_data,i)
        if flag:
            min_list.append(i)
            flag = False
    return min_list
```

[153]:
```
def split_cycles(min_list,raw_data):
    cycles=[]
    for  i in range(len(min_list)-1):
        d = np.zeros(50)
        c = np.zeros(50)
        j = 0
        l = min_list[i+1]-min_list[i]
        for k in range(min_list[i],min_list[i+1]-1):
            m = int(((k-min_list[i])*50.0)/l)
            d[m]+=raw_data[k]
            c[m]+=1
        for k in range(50):
            d[k] = d[k]/c[k]
        if True in np.isnan(np.array(d)):
            d = []
        else:
            cycles.append(d)
    return cycles
```

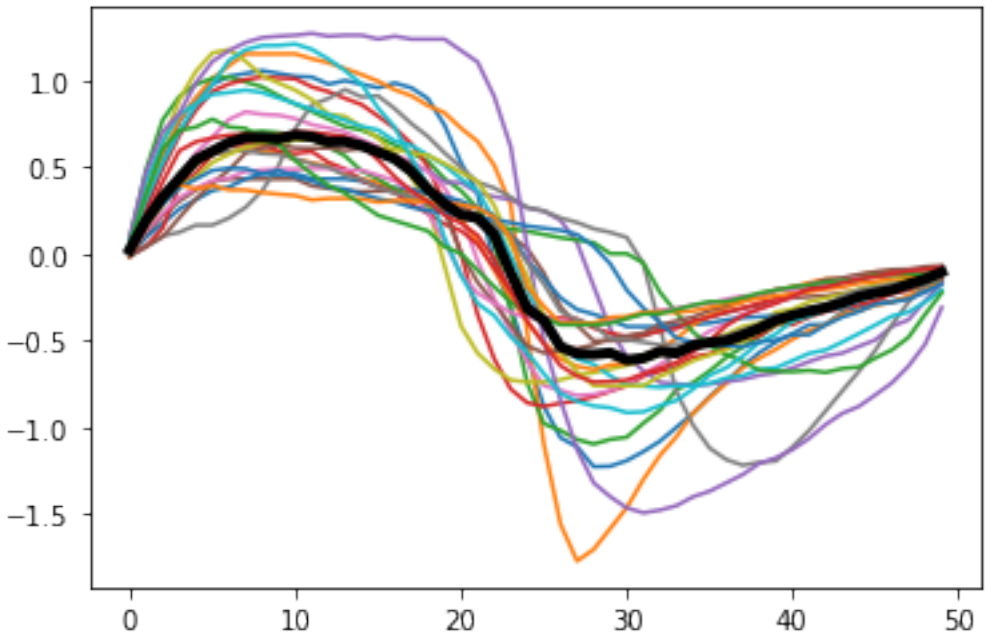
[154]:
```
def get_file(file_number):
    data = np.genfromtxt('data/'+files[file_number], delimiter='\t')[:,0]
    #data_average = data.mean()
```

```
    #data = data - data_average
    mins = separate_cycles(data)
    cycles = split_cycles(mins,data)
    return cycles
```

[155]:
```
cycles = get_file(6)
```

```
/home/ric/Projects/Python/env/lib/python3.7/site-
packages/ipykernel_launcher.py:13: RuntimeWarning: invalid value encountered in
double_scalars
  del sys.path[0]
```

[224]:
```
for c in cycles:
    plt.plot(c)
plt.plot(np.median(np.array(cycles),axis=0),linewidth=4, color='black')
plt.show()
```



[202]:
```
def sig_plot():
    median_plots = []
    fig, axs = plt.subplots(len(files)//2,2,figsize=(24, len(files)//2 * 6))
    p=0
    for i in range (2):
        for j in range(len(files)//2):
            if p < len(files):
                cycles = get_file(p)
                for c in cycles:
```

```
                    axs[j,i].plot(c)
                m = np.median(np.array(cycles),axis=0)
                axs[j,i].plot(m,linewidth=4, color='black')
                axs[j,i].set_title(files[p])
                median_plots.append(m)
                p+=1
    return median_plots
```

[203]:
```
def difference(a,b):
    count = 0
    for i in range(a.shape[0]-1):
        #print(a[i],b[i])
        count += np.absolute(a[i]-b[i])
    return count/a.shape[0]
```

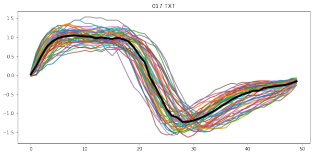
[214]:
```
def dendogram_heat_map(distance_data,map_data,tags):
    """ Create s heat map with associated dendogram froma distance N*N numpy␣
 ↪matrix and a scipy
    scipy.cluster.hierarchy linkage 2d array"""
    # Adapted from http://stackoverflow.com/users/208339/steve-tjoa
    fig = pylab.figure(figsize=(12,12))

    # Plot first dendrogram.
    ax1 = fig.add_axes([0.09,0.1,0.2,0.6])#set position
    den1 = sch.dendrogram(map_data, orientation='right',labels=tags)
    # No axis lables
    ax1.set_xticks([])
    #ax1.set_yticks([])

    # Plot second dendrogram.
    ax2 = fig.add_axes([0.31,0.74,0.61,0.2])#set position
    den2 = sch.dendrogram(map_data,labels=tags)
    # No axis lables
    #ax2.set_xticks([])
    ax2.set_yticks([])

    # Plot distance matrix as heat map.
    heat_map = fig.add_axes([0.32,0.1,0.6,0.6])
    idx1 = den1['leaves']
    idx2 = den2['leaves']
    # Reorder the distance data so that it matches with the dendogram order
    distance_data = distance_data[idx1,:]
    distance_data = distance_data[:,idx2]
    hm = heat_map.matshow(distance_data, aspect='auto', origin='lower',␣
 ↪cmap=pylab.cm.YlGnBu)
    # No Y axis
    heat_map.set_yticks([])
```

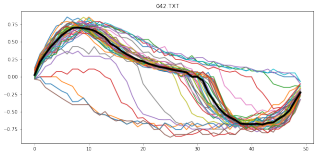
```
    heat_map.set_xticks([])



    # Plot colorbar.
    axcolor = fig.add_axes([0.93,0.1,0.02,0.6])
    pylab.colorbar(hm, cax=axcolor)
```

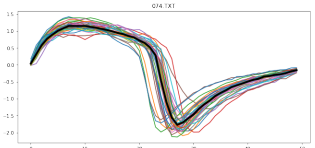
[215]:
```
def difference(a,b):
    count = 0
    for i in range(a.shape[0]-1):
        #print(a[i],b[i])
        count += np.absolute(a[i]-b[i])
    return count/a.shape[0]
```

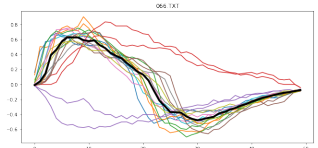
[216]:
```
cycles = sig_plot()
```

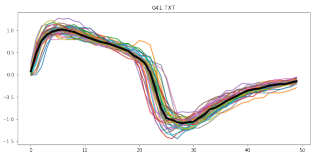
```
/home/ric/Projects/Python/env/lib/python3.7/site-
packages/ipykernel_launcher.py:13: RuntimeWarning: invalid value encountered in
double_scalars
  del sys.path[0]
```

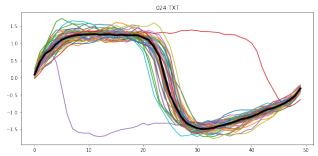

017.TXT

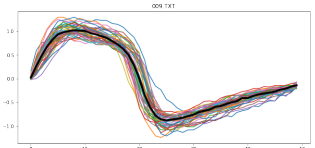

042.TXT

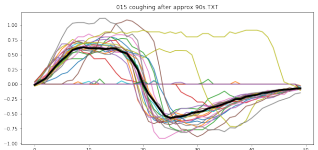

074.TXT

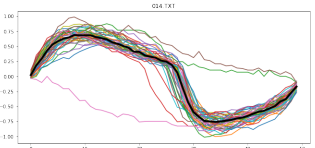

066.TXT

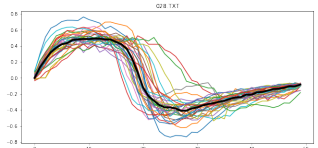

041.TXT

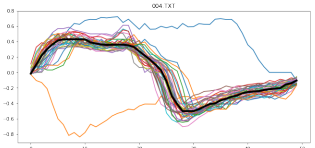

024.TXT

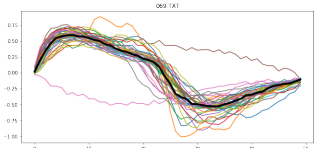

009.TXT

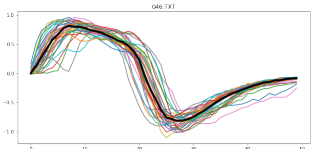

015 coughing after approx 90s.TXT

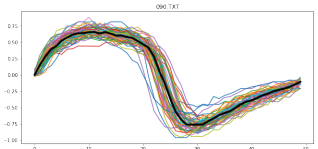

014.TXT

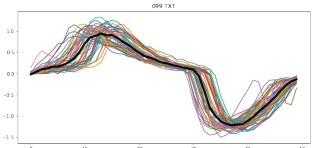

028.TXT

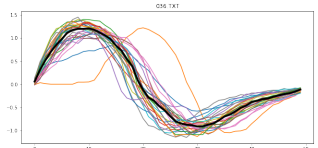

004.TXT

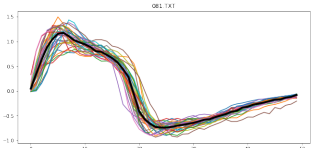

069.TXT

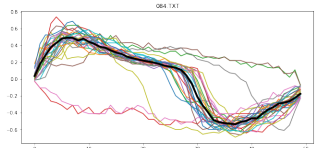

046.TXT

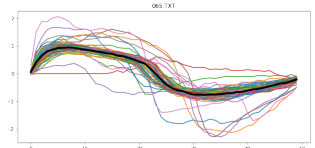

090.TXT

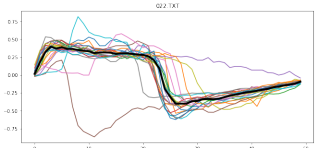

099.TXT

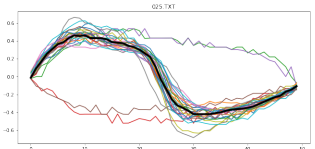

036.TXT

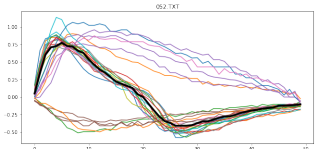

081.TXT

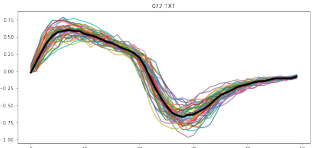

084.TXT

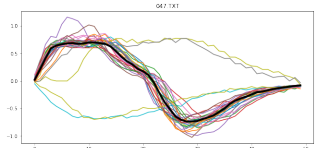

065.TXT


022.TXT


025.TXT


052.TXT

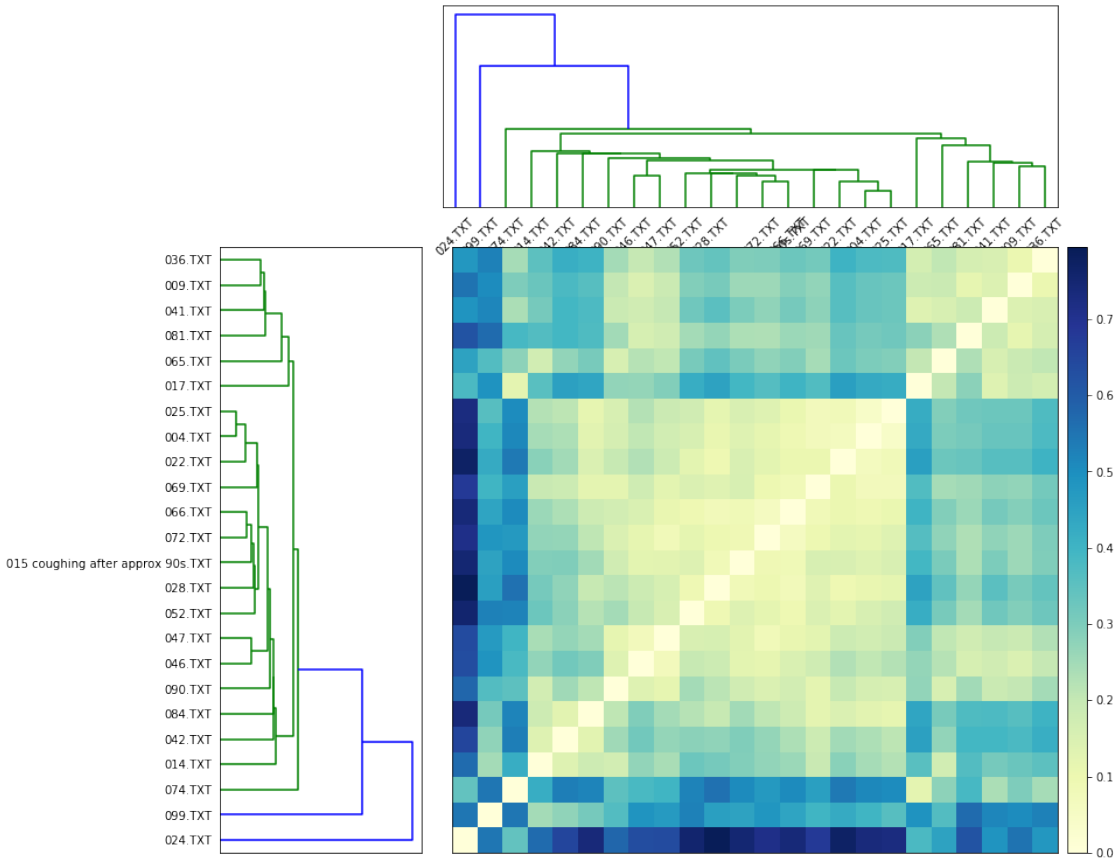

072.TXT


047.TXT

[217]:
```
distance = np.zeros((len(cycles),len(cycles)))
for i in range(len(cycles)):
    for j in range(len(cycles)):
        distance[i,j]=differance(cycles[i],cycles[j])
kmap = sch.linkage(distance, method='single')
```

```
/home/ric/Projects/Python/env/lib/python3.7/site-
packages/ipykernel_launcher.py:5: ClusterWarning: scipy.cluster: The symmetric
non-negative hollow observation matrix looks suspiciously like an uncondensed
distance matrix
  """
```

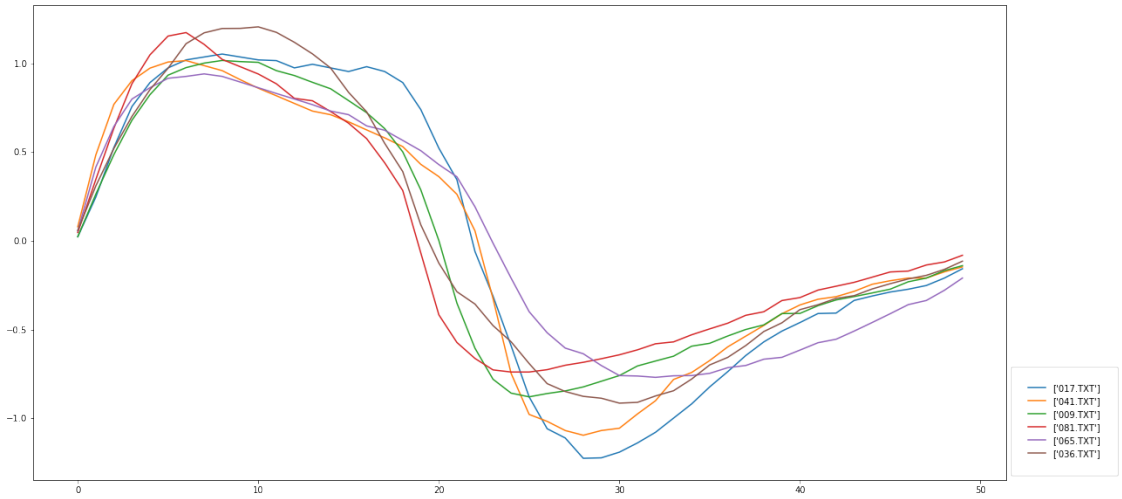
[218]:
```
dendogram_heat_map(distance,kmap,files)
```



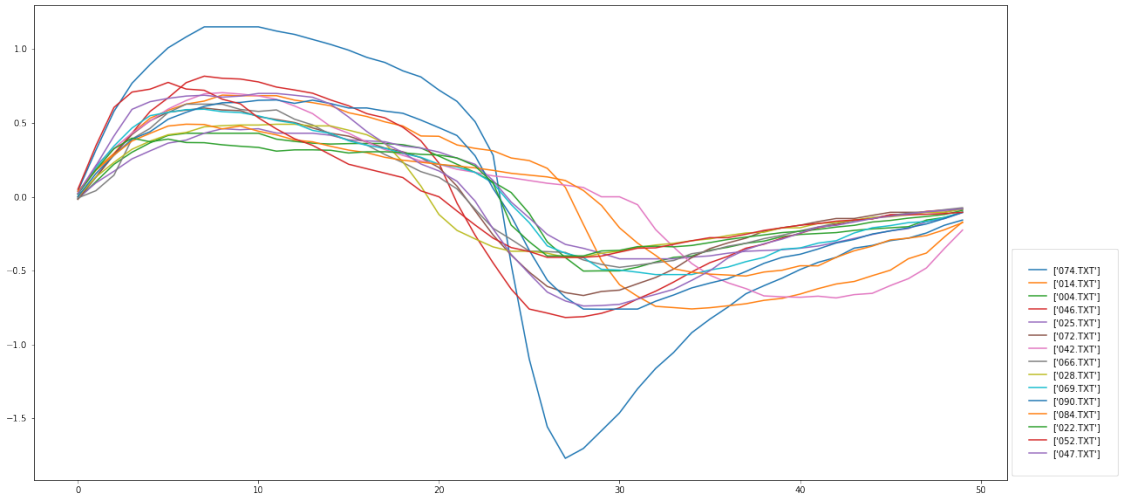
[ ]:
```
'069.TXT','066.TXT','072.TXT','028.TXT','052.TXT','047.TXT','046.TXT','090.
↪TXT','084.TXT','042.TXT','014.TXT','074.TXT']
```

[244]:
```
def plt_group(in_list):
    plt.figure(figsize=[20,10])
    for i in range(len(cycles)):
        if files[i] in in_list:
            plt.plot(cycles[i],label=[files[i]])
    plt.legend(loc=3,borderpad=2,bbox_to_anchor=(1, 0))
    plt.show()
```

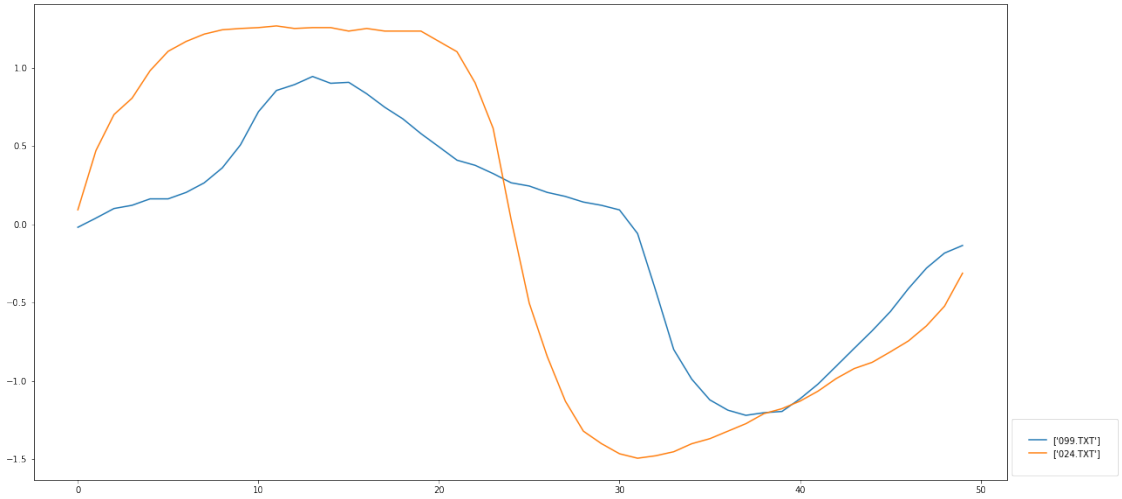
[247]:
```
plt_group(['036.TXT','009.TXT','041.TXT','081.TXT','065.TXT','017.TXT'])
```



[251]:
```
plt_group(['025.TXT','004.TXT','022.TXT','069.TXT','066.TXT','072.TXT','028.
↪TXT','052.TXT','047.TXT','046.TXT','090.TXT','084.TXT','042.TXT','014.
↪TXT','074.TXT'])
```



[253]:
```
plt_group(['099.TXT','024.TXT'])
```



[ ]:
```

```