Experiments with Topological Data Analysis on Signals

November 3, 2017

1 Persistent Homology of Sliding Windows

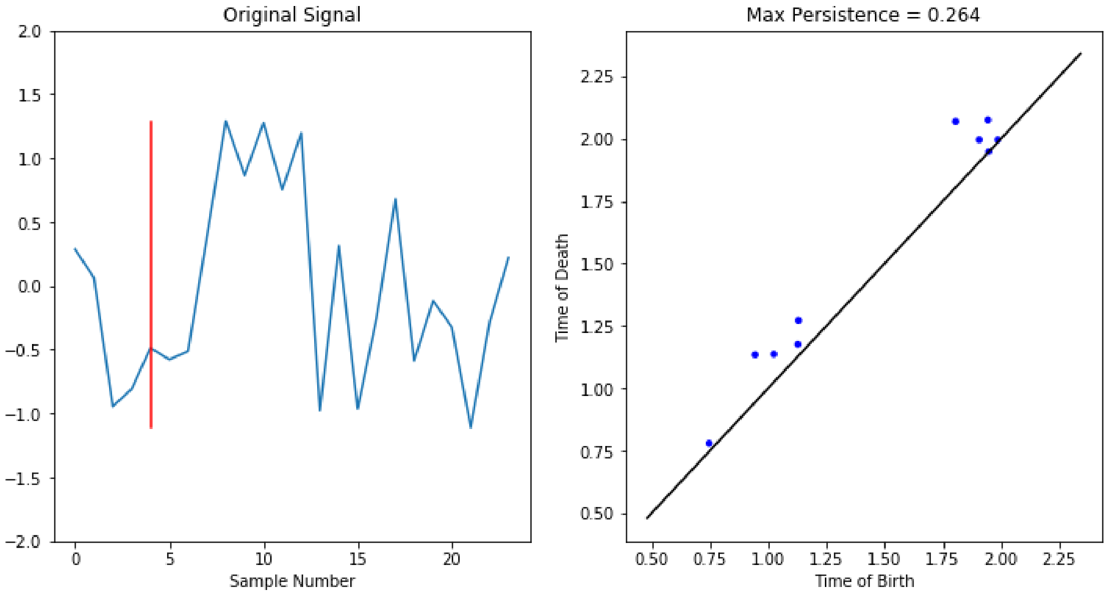
1.1 Part 1: Introduction to Topological Data Analysis

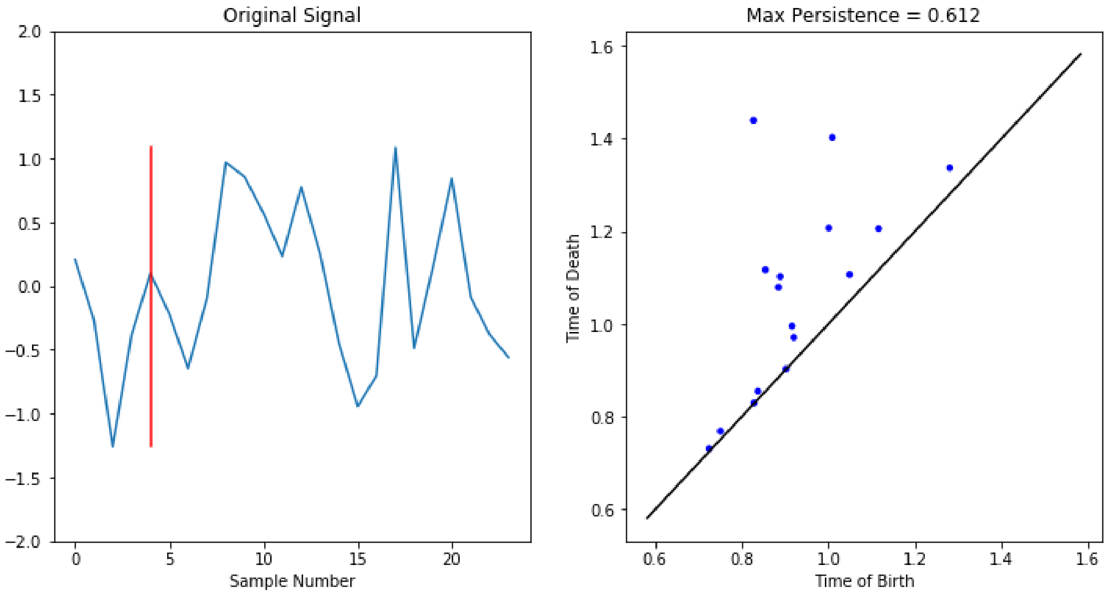
Topological Data Analysis is a new approach towards high dimensional data analysis and machine learning. Based on methods from applied algebraic topology, this method deals with high dimensional data by exploring the topological properties of the data, such as the number of connected components or the number of holes formed and disappeared after the space has been filled in with simplicial complexes. In this particular problem topological data analysis tries to quantify the periodicity of the biological signal by using a sliding window. Continuous snippets of the signal are taken, and the points in each of these snippets are input into a vector, so that each snippet is discretized by a point in the high dimensional space. By examining the circularity of the points in this high dimensional space, we can move back to the signal and have a conclusion about how 'periodic' the signal is based on a numerical score.

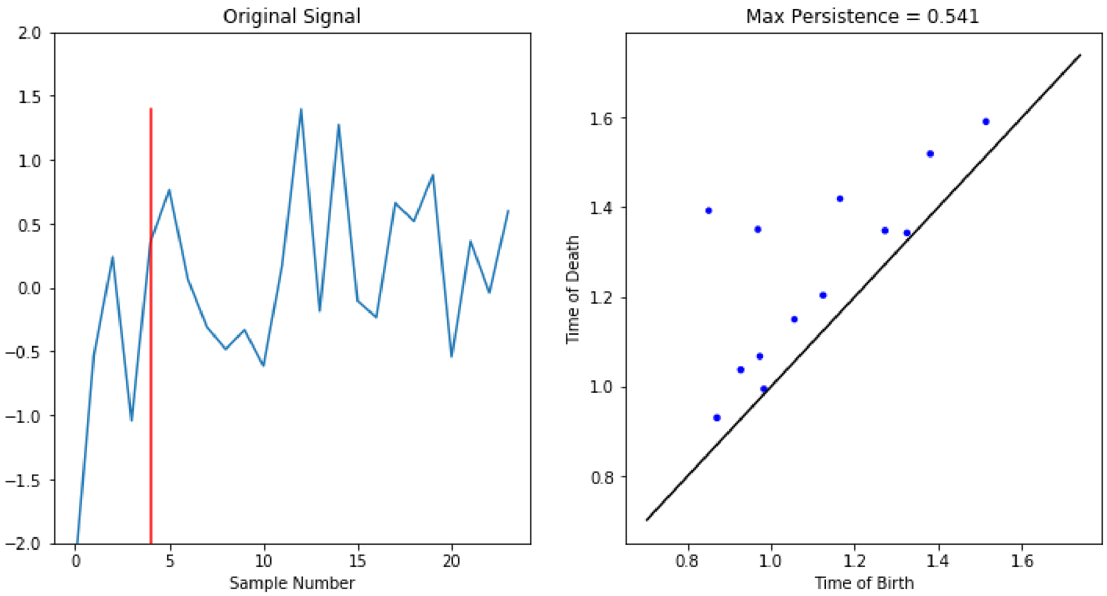
Based on a method by Jose Perea et al. (2016), and an implementation from https://github.com/ctralie/TUMTopoTimeSeries2016, we can construct a preliminary topological processing of the raw RNA data.

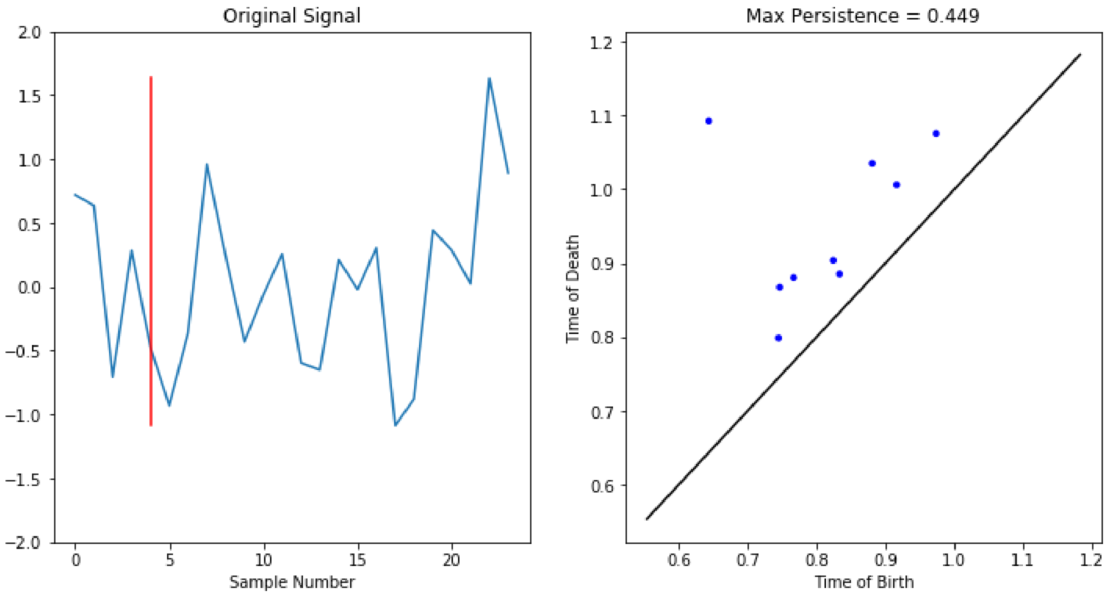
```
In [3]: ##Do all of the imports and setup inline plotting
        %matplotlib inline
        import numpy as np
        import matplotlib.pyplot as plt
        from sklearn.decomposition import PCA
        from mpl_toolkits.mplot3d import Axes3D
        import scipy.interpolate as interp

        from TDA import *

        ##Setup the sliding window code
        def getSlidingWindow(x, dim, Tau, dT):
            N = len(x)
            NWindows = int(np.floor((N-dim*Tau)/dT)) #The number of windows
            if NWindows <= 0:
                print("Error: Tau too large for signal extent")
                return np.zeros((3, dim))
            X = np.zeros((NWindows, dim)) #Create a 2D array which will store all windows
            idx = np.arange(N)
            for i in range(NWindows):
                #Figure out the indices of the samples in this window
                idxx = dT*i + Tau*np.arange(dim)
                start = int(np.floor(idxx[0]))
                end = int(np.ceil(idxx[-1]))+2
                if end >= len(x):
                    X = X[0:i, :]
                    break
                #Do spline interpolation to fill in this window, and place
                #it in the resulting array
                X[i, :] = interp.spline(idx[start:end+1], x[start:end+1], idxx)
            return X

In [4]: ##Reading the signal
        import pandas as pd
        gene = pd.read_csv('CopyOfRNA_Avg_Raw_Data.csv')

In [5]: ##Plotting the sliding window projection and observing the topological events and how th
        for i in range(4):
            signal = np.asarray(gene.loc[i,:][1:])

            #Step 1: Setup the signal
            x = pd.to_numeric(signal) #The final signal

        #Step 2: Do a sliding window embedding
            dim = 4
            Tau = 1
            dT = 0.5
            X = getSlidingWindow(x, dim, Tau, dT)
            extent = Tau*dim

        #Step 3: Do Rips Filtration
            PDs = doRipsFiltration(X, 1)

        #Step 4: Perform PCA down to 2D for visualization
            pca = PCA(n_components = 2)
            Y = pca.fit_transform(X)
            eigs = pca.explained_variance_

        #Step 5: Plot original signal and the persistence diagram
            fig = plt.figure(figsize=(12, 6))
            ax = plt.subplot(121)
            ax.plot(x)
            ax.set_ylim((-2, 2))
            ax.set_title("Original Signal")
            ax.set_xlabel("Sample Number")
        #ax.hold(True)
            ax.plot([extent, extent], [np.min(x), np.max(x)], 'r')
            ax2 = fig.add_subplot(122)
            I = PDs[1]
            plotDGM(I)
            plt.title("Max Persistence = %.3g"%np.max(I[:, 1] - I[:, 0]))
            plt.savefig('Persistence_{0}.jpg'.format(i))
```

```
/Applications/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:30: DeprecationWarning
spline is deprecated in scipy 0.19.0, use Bspline class instead.
```









The next step will be obtaining the score of periodicity for various signals based on the methods suggested by the paper, and explore different ways to potentially filter the signals on the high dimensional space with topological filters.

2 References

Perea, J. A., Deckard, A., Haase, S. B., & Harer, J. (2015). SW1PerS: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. BMC Bioinformatics, 16, 257. http://doi.org/10.1186/s12859-015-0645-6