20200403_FS_Trial_end_speed_profiles_IMSHOW()

April 3, 2020

1 This script is designed to graph speed profiles before approaching the beacon using imShow.

It should take beacons file, find time, match with position, then read however rows of position before and after and plot it.

imports

```
In [61]: import pandas as pd
         import matplotlib.pyplot as plt
         import matplotlib as mpl
         import seaborn as sns
         from matplotlib.patches import Circle
         import matplotlib.tri as tri
         import numpy as np
         from scipy.spatial.transform import Rotation as R
         import math

         root = 'C:/Users/Fabian/Desktop/Analysis/Multiple_trial_analysis/Data/Raw/'
         figures = 'C:/Users/Fabian/Desktop/Analysis/Multiple_trial_analysis/Figures/'

         #Data with beacon metadata
         beacon = pd.read_csv(root+'beacons 20200128-151826.txt',sep=" ", header=None)
         beacon2 = pd.read_csv(root+'beacons 20200128-160013.txt',sep=" ", header=None)

         beacon_Day86_fs2 = pd.read_csv(root+'beacons 20200128-160013.txt',sep=" ", header=None
         beacon_Day86_fs1 = pd.read_csv(root+'beacons 20200128-151826.txt',sep=" ", header=None

         beacon_Day87_fs2 = pd.read_csv(root+'beacons 20200129-153534.txt',sep=" ", header=None
         beacon_Day87_fs1 = pd.read_csv(root+'beacons 20200129-161806.txt',sep=" ", header=None

         beacon_Day88_fs2 = pd.read_csv(root+'beacons 20200130-102126.txt',sep=" ", header=None
         beacon_Day88_fs1 = pd.read_csv(root+'beacons 20200130-111741.txt',sep=" ", header=None

         beacon_Day89_fs2 = pd.read_csv(root+'beacons 20200130-161126.txt',sep=" ", header=None
         beacon_Day89_fs1 = pd.read_csv(root+'beacons 20200130-151829.txt',sep=" ", header=None
```

```
beacon_Day90_fs2 = pd.read_csv(root+'beacons 20200203-154441.txt',sep=" ", header=None
beacon_Day90_fs1 = pd.read_csv(root+'beacons 20200203-145842.txt',sep=" ", header=None

beacon_Day91_fs2 = pd.read_csv(root+'beacons 20200204-125552.txt',sep=" ", header=None
beacon_Day91_fs1 = pd.read_csv(root+'beacons 20200204-133905.txt',sep=" ", header=None

beacon_Day92_fs2 = pd.read_csv(root+'beacons 20200205-143220.txt',sep=" ", header=None
beacon_Day92_fs1 = pd.read_csv(root+'beacons 20200205-151052.txt',sep=" ", header=None

beacon_Day93_fs2 = pd.read_csv(root+'beacons 20200206-133529.txt',sep=" ", header=None
beacon_Day93_fs1 = pd.read_csv(root+'beacons 20200206-125706.txt',sep=" ", header=None

Day46_fs1 = pd.read_csv(root+'position 20190923-174441.txt',sep=" ", header=None)
Day46_fs2 = pd.read_csv(root+'position 20190923-171112.txt',sep=" ", header=None)
Day47_fs1 = pd.read_csv(root+'position 20191001-112411.txt',sep=" ", header=None)
Day47_fs2 = pd.read_csv(root+'position 20191001-115127.txt',sep=" ", header=None)
Day48_fs1 = pd.read_csv(root+'position 20191002-115000.txt',sep=" ", header=None)
Day48_fs2 = pd.read_csv(root+'position 20191002-111038.txt',sep=" ", header=None)
Day51_fs1 = pd.read_csv(root+'position 20191106-170809.txt',sep=" ", header=None)
Day52_fs2 = pd.read_csv(root+'position 20191107-174215.txt',sep=" ", header=None)
Day52_fs1 = pd.read_csv(root+'position 20191107-183857.txt',sep=" ", header=None)
Day53_fs2 = pd.read_csv(root+'position 20191108-142321.txt',sep=" ", header=None)
Day53_fs1 = pd.read_csv(root+'position 20191108-145125.txt',sep=" ", header=None)
Day66_fs1 = pd.read_csv(root+'position 20191118-161325.txt',sep=" ", header=None)
Day66_fs2 = pd.read_csv(root+'position 20191118-171209.txt',sep=" ", header=None)
Day72_fs1 = pd.read_csv(root+'position 20191127-122008.txt',sep=" ", header=None)
Day72_fs2 = pd.read_csv(root+'position 20191127-132223.txt',sep=" ", header=None)


Day79_fs2 = pd.read_csv(root+'position 20200121-154004.txt',sep=" ", header=None)
Day79_fs1 = pd.read_csv(root+'position 20200121-161359.txt',sep=" ", header=None)

Day80_fs2 = pd.read_csv(root+'position 20200122-141738.txt',sep=" ", header=None)
Day80_fs1 = pd.read_csv(root+'position 20200122-133022.txt',sep=" ", header=None)

Day81_fs2 = pd.read_csv(root+'position 20200123-141930.txt',sep=" ", header=None)
Day81_fs1 = pd.read_csv(root+'position 20200123-150059.txt',sep=" ", header=None)

Day82_fs2 = pd.read_csv(root+'position 20200124-151642.txt',sep=" ", header=None)
Day82_fs1 = pd.read_csv(root+'position 20200124-160826.txt',sep=" ", header=None)

Day83_fs2 = pd.read_csv(root+'position 20200126-183810.txt',sep=" ", header=None)
Day83_fs1 = pd.read_csv(root+'position 20200126-180200.txt',sep=" ", header=None)

Day84_fs2 = pd.read_csv(root+'position 20200127-205615.txt',sep=" ", header=None)
Day84_fs1 = pd.read_csv(root+'position 20200127-155645.txt',sep=" ", header=None)
```

```
Day85_fs2 = pd.read_csv(root+'position 20200128-112255.txt',sep=" ", header=None)
Day85_fs1 = pd.read_csv(root+'position 20200128-104637.txt',sep=" ", header=None)

Day86_fs2 = pd.read_csv(root+'position 20200128-160013.txt',sep=" ", header=None)
Day86_fs1 = pd.read_csv(root+'position 20200128-151826.txt',sep=" ", header=None)

Day87_fs2 = pd.read_csv(root+'position 20200129-153534.txt',sep=" ", header=None)
Day87_fs1 = pd.read_csv(root+'position 20200129-161806.txt',sep=" ", header=None)

Day88_fs2 = pd.read_csv(root+'position 20200130-102126.txt',sep=" ", header=None)
Day88_fs1 = pd.read_csv(root+'position 20200130-111741.txt',sep=" ", header=None)

Day89_fs2 = pd.read_csv(root+'position 20200130-161126.txt',sep=" ", header=None)
Day89_fs1 = pd.read_csv(root+'position 20200130-151829.txt',sep=" ", header=None)

Day90_fs2 = pd.read_csv(root+'position 20200203-154441.txt',sep=" ", header=None)
Day90_fs1 = pd.read_csv(root+'position 20200203-145842.txt',sep=" ", header=None)

Day91_fs2 = pd.read_csv(root+'position 20200204-125552.txt',sep=" ", header=None)
Day91_fs1 = pd.read_csv(root+'position 20200204-133905.txt',sep=" ", header=None)

Day92_fs2 = pd.read_csv(root+'position 20200205-143220.txt',sep=" ", header=None)
Day92_fs1 = pd.read_csv(root+'position 20200205-151052.txt',sep=" ", header=None)

Day93_fs2 = pd.read_csv(root+'position 20200206-133529.txt',sep=" ", header=None)
Day93_fs1 = pd.read_csv(root+'position 20200206-125706.txt',sep=" ", header=None)
```

1.1 now need to add a function which takes time from beacon and matches it with position information in position

1.2 Get speed list!

In [29]:
```
import numpy

time = numpy.arange(0.01, len(Day86_fs1[0]), 0.01)
time= np.array(Day86_fs1[0][1:]-Day86_fs1[0][0])
def calculateSpeed_list(x,y,time):
    travel=0
    speed=[]
    for i in range(len(y)-1):
        dist = math.sqrt((x[0+i] - x[1+i])**2 + (y[0+i] - y[1+i])**2)/time[0+i]
        speed.append(dist)
    return (speed)

speed_list = calculateSpeed_list(Day86_fs1[1],Day86_fs1[3],time)
speed_list=np.array(speed_list)
speed_list
```

Out[29]:
```
array([1.28241777e+00, 4.49595123e-01, 3.30493744e-01, ...,
```

```
9.77006484e-08, 1.31449350e-07, 1.47630570e-07])
```

1.3 Freguency 100hz

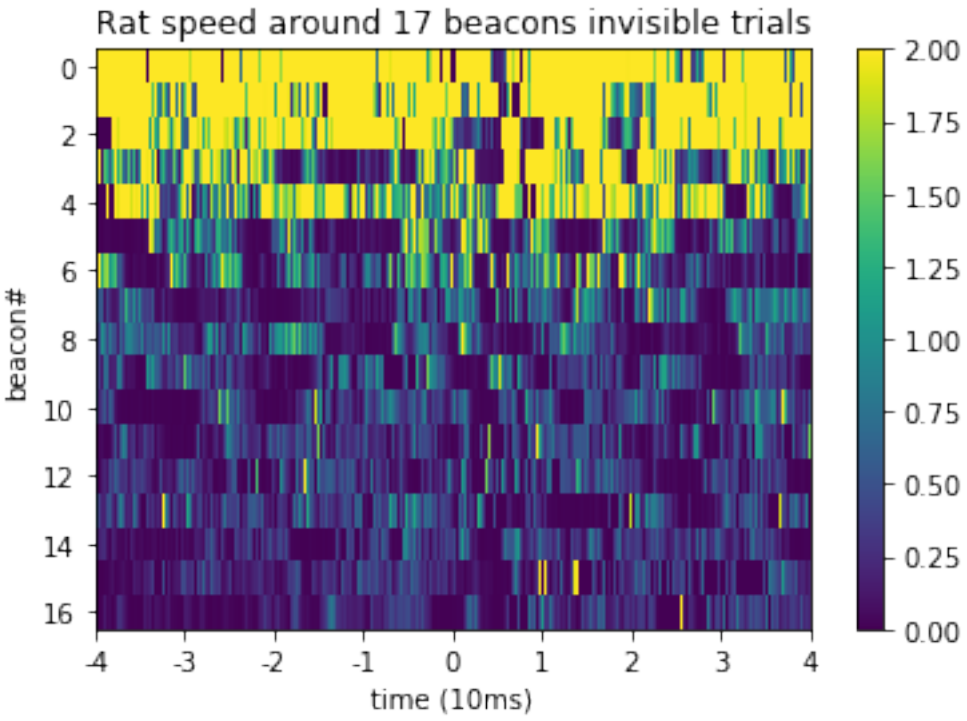
```
In [30]: speed_list.shape
```

```
Out[30]: (107974,)
```

```
In [68]: def Speed_before_beacon (session,seconds_back,position_data,beacon_data,max_speed,zero
             """function plots a given amount of seconds before beacon reached, takes Beacon f

             # still
             time_since_list =((np.array(beacon_data [0]))-(np.array(position_data[0]))[0])
             enum = list(np.arange(0,len(list(time_since_list)),1))
             index=[]
             beacon_times = np.array(beacon_data[0])


             for i in enum:
                 index.append(np.abs(beacon_times[i]-np.array(position_data[0])).argmin() )

             if height== False:
                 num=3
             else:
                 num=2


             time = numpy.arange(0.01, len(position_data[0]), 0.01)
             speed_list = calculateSpeed_list(position_data[1],position_data[num],time)
             speed_list = [element * 100000 for element in speed_list] # to get to cm/s speed.
             time = numpy.arange(-seconds_back,seconds_back,.01)
             fig, ax = plt.subplots()

             # involve gaussian smoothening.
             if visible == True:
                 index = index[::2]
                 eye='visible trials'
             elif visible == False:
                 index = index[1::2]
                 eye = 'invisible trials'
             else:
                 index=index
                 eye = 'all trials'

             list_of_speed_lists=[]
             for i in index:
                 if i <= seconds_back*100:
                     i = seconds_back*100+1
```

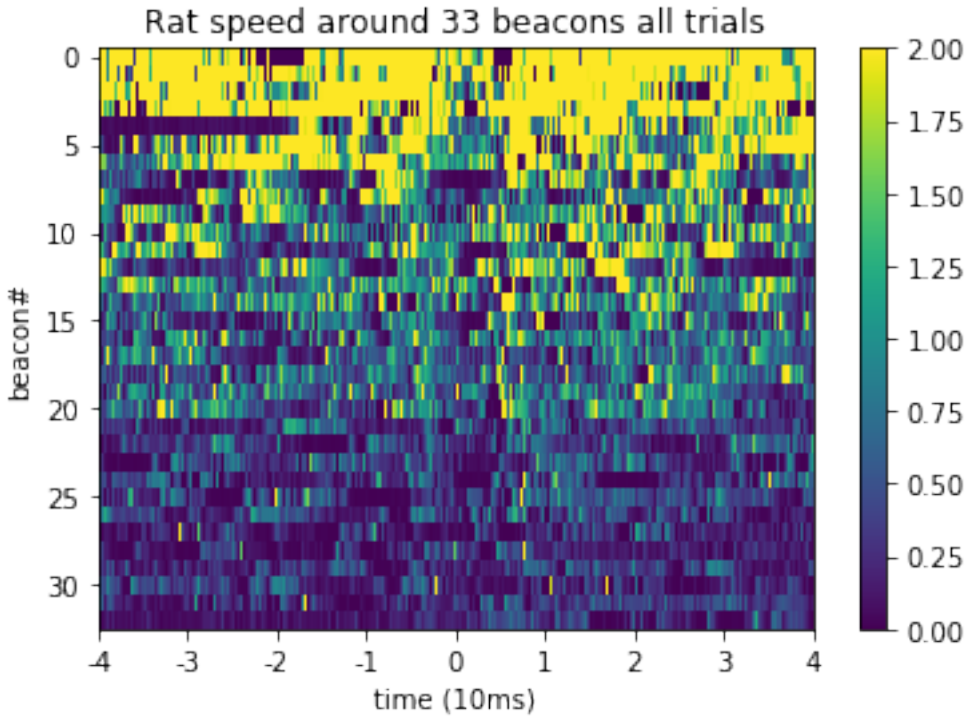
```
            else:
                list_of_speed_lists.append(speed_list[i-(seconds_back*100):i+(seconds_back

        array = np.array(list_of_speed_lists).reshape(len(index)+zero,seconds_back*200) #

        im = ax.imshow(array, cmap=plt.get_cmap('viridis'),aspect='auto',vmin=0, vmax=max_
        fig.colorbar(im)

        ax.set(xlabel='time (10ms)', ylabel='beacon#',
               title='Rat speed around %s beacons %s' %(len(index),eye))
        #ax.grid()
        x_label_list = ['-5','-4','-3','-2','-1','0','1','2','3','4','5']
        ax.set_xticklabels(x_label_list)
        ax.set_alpha (alpha = .5)
        print('%s seconds in session %s, with %s max speed filter showing %s beacons  %s

        plt.savefig('%s_imshow_speed_%s_sec._before_and _after_%s_beacon_%s.png'%(figures
        plt.show()

Speed_before_beacon ('86_fs1',10,Day86_fs1,beacon_Day86_fs1,2,zero=0,visible=False,hei
```

```
20 seconds in session 86_fs1, with 2 max speed filter showing 17 beacons  invisible trials
```



```
In [34]: Speed_before_beacon ('86_fs2',10,Day86_fs2,beacon_Day86_fs2,2,visible=all,height=Fals
```

```
20 seconds in session 86_fs2, with 2 max speed filter showing 33 beacons  all trials
```

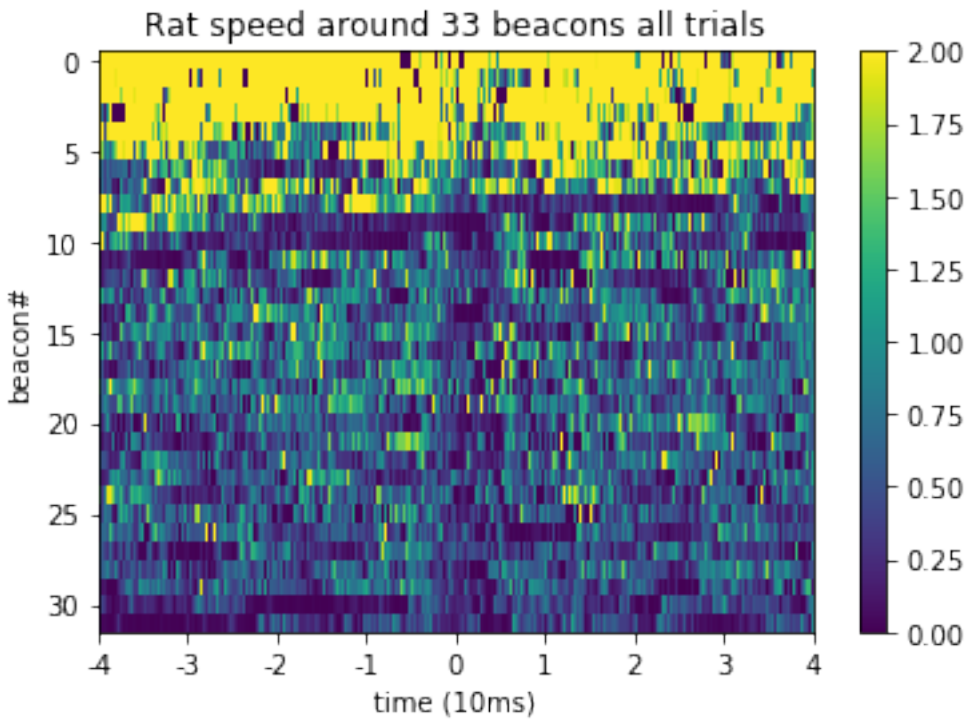



```
In [9]: speed_list.shape
```

```
Out[9]: (107974,)
```

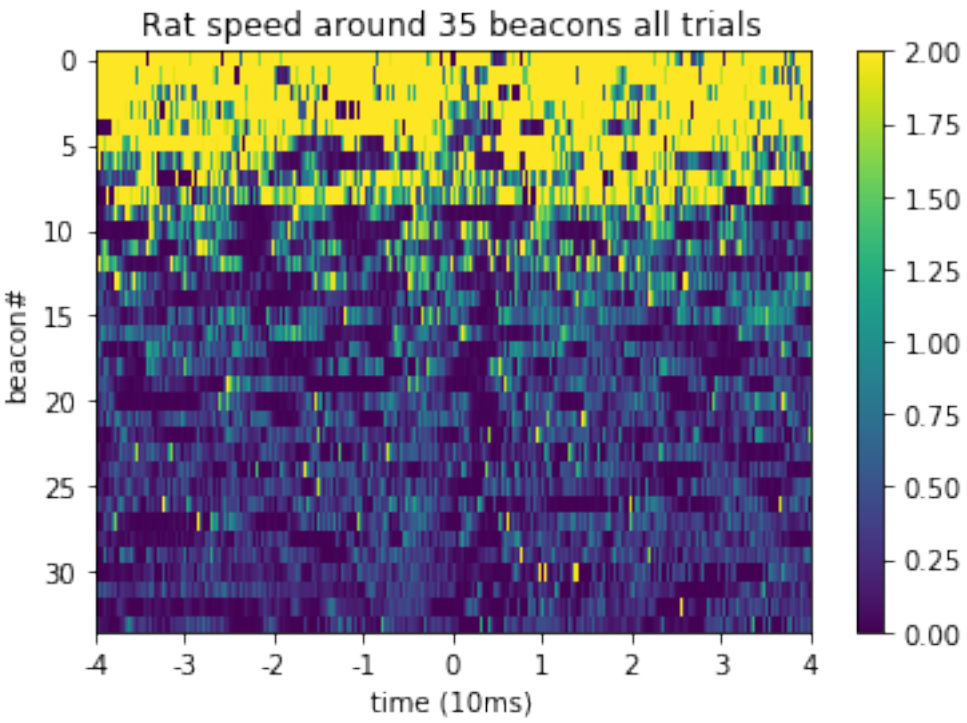
```
In [69]: Speed_before_beacon ('87_fs2',10,Day87_fs2,beacon_Day87_fs2,2,visible=all,height=Fals
```

```
20 seconds in session 87_fs2, with 2 max speed filter showing 33 beacons  all trials
```



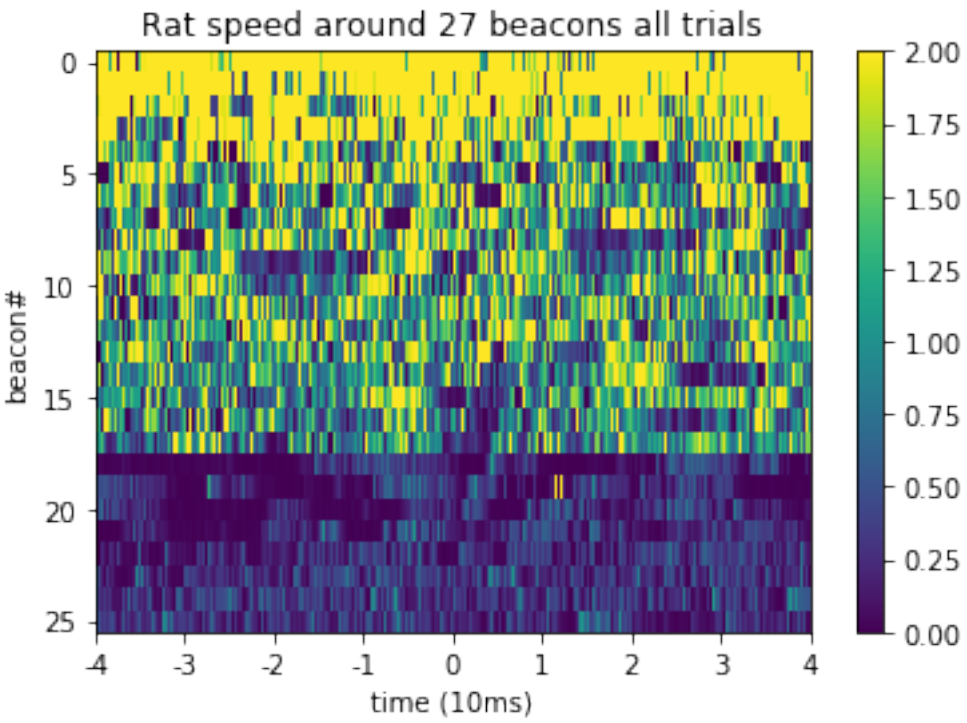

```
In [70]: Speed_before_beacon ('87_fs1',10,Day86_fs1,beacon_Day86_fs1,2,visible=all,height=Fals
```

```
20 seconds in session 87_fs1, with 2 max speed filter showing 35 beacons  all trials
```



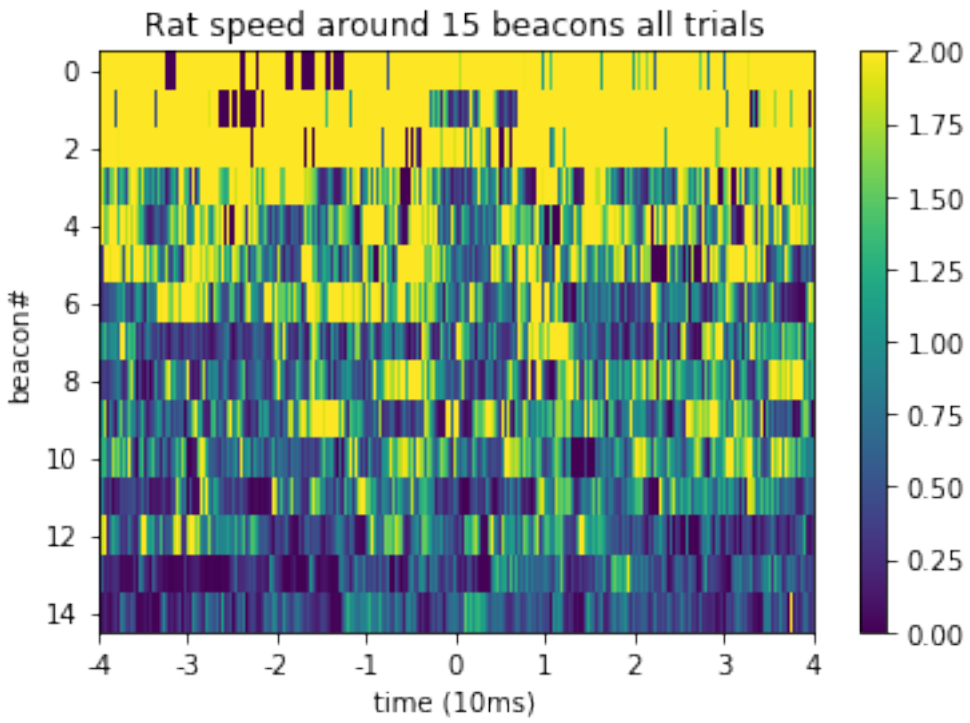

```
In [71]: Speed_before_beacon ('90_fs1',10,Day90_fs1,beacon_Day90_fs1,2,visible=all,height=Fals
```

```
20 seconds in session 90_fs1, with 2 max speed filter showing 27 beacons  all trials
```



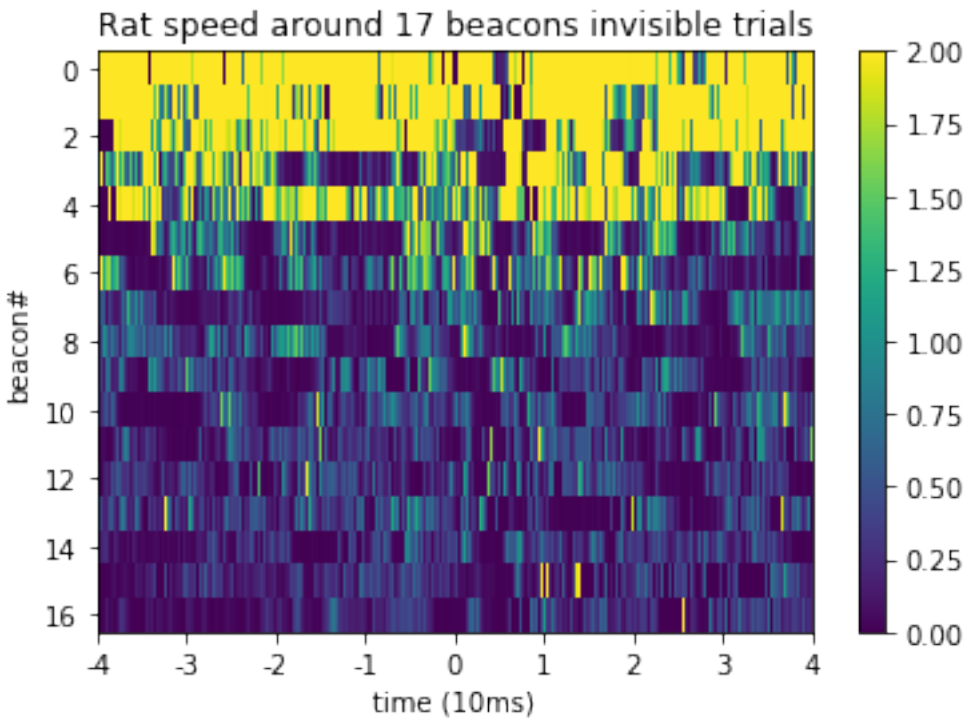

```
In [72]: Speed_before_beacon ('90_fs2',10,Day90_fs2,beacon_Day90_fs2,2,zero=0,visible=all,heigl
```

```
20 seconds in session 90_fs2, with 2 max speed filter showing 15 beacons  all trials
```




```
In [67]: Speed_before_beacon ('86_fs1',10,Day86_fs1,beacon_Day86_fs1,2,zero=,visible=False,heig
```

```
20 seconds in session 86_fs1, with 2 max speed filter showing 17 beacons  invisible trials
```



1.4 Plot speed over the whole session

In [ ]: