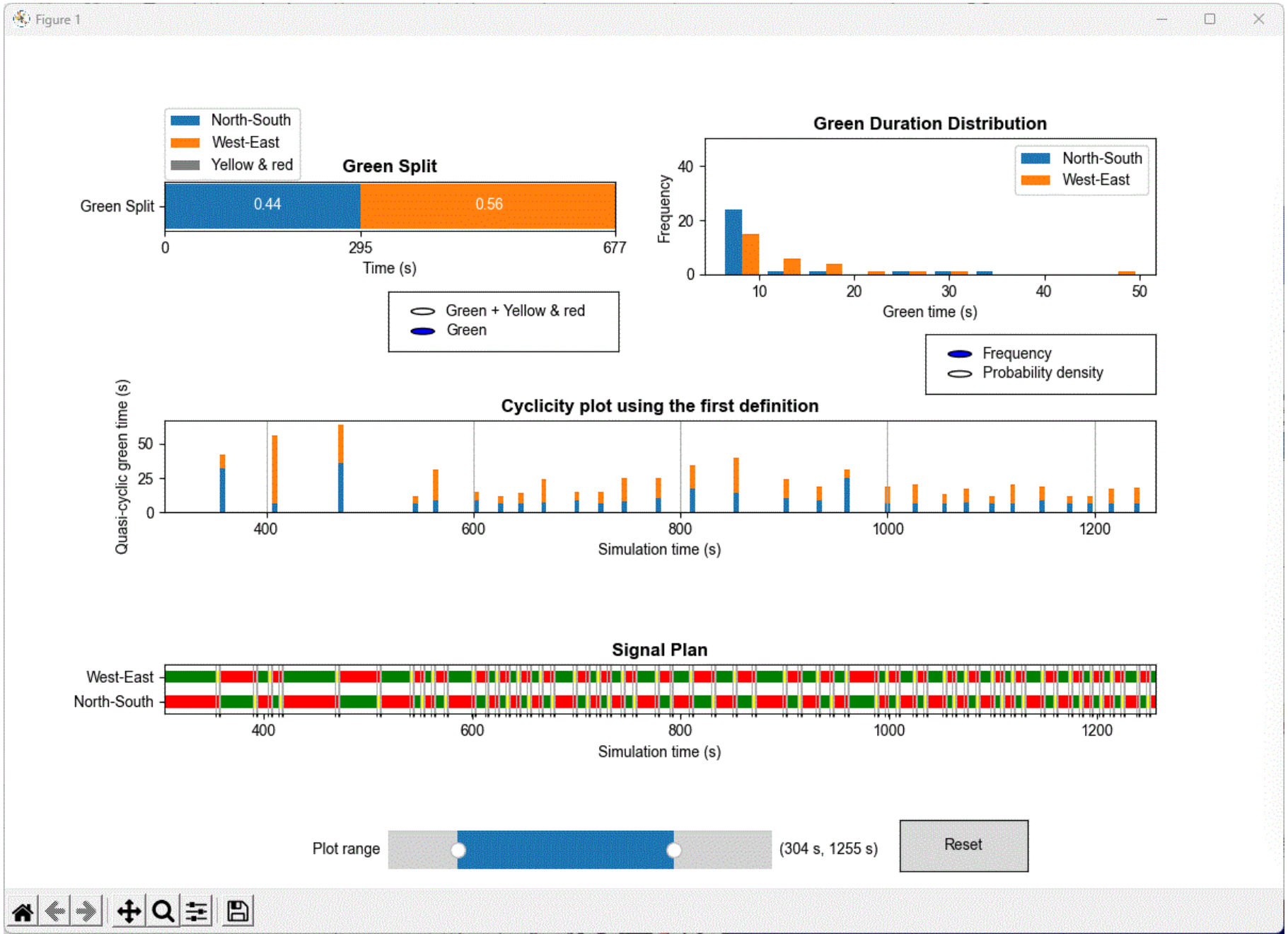
Using plot functions `clusterPlot_TLS` for signalisation analysis based on SUMO output

1) General: `clusterPlot_TLS`
2) How to get the data?
3) What are required definitions?
4) What optional settings are available?
5) Example with an important notes
6) Developer's notes (for those interested in contributing to the repo)



Usage example for `clusterPlot_TLS`

1) General: `clusterPlot_TLS`

- The plot function is for a single signalised intersection that is modelled in SUMO
- The plot function is written in Python with the use of some 3rd party libraries (please refer to requirements.txt):
  - Numpy
  - Pandas
  - Matplotlib
  - Lxml (for importing .xml file as a Dataframe)

- MIT License
- This software is intended to be freely shared among Oguchi Lab members for research purposes

2) How to get the data?

- Use the data output of the type 'TLS Switch States' (URL to SUMO website)
  - Add the following xml definition in an additional file:

```
</additional>
      <timedEvent type="SaveTLSSwitchStates" source="J" dest="TLSrecord.xml"/>
</additional>
```

  - `source`: id of TLLogic object, `dest`: output path

- The exported data will be in .xml file format

- To pipeline the data to the function, import the .xml file as a Pandas DataFrame

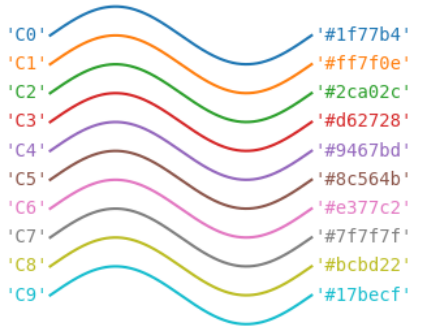
```
tlsdf = pandas.read_xml('TLSrecord.xml')
```

3) What are required definitions?

There are 3 positional argument to `clusterPlot_TLS`

Variable name	Object type	Description	Object instantiation example
`tlsdf`	`pandas.DataFrame`	Traffic signal data in 2)	`tlsdf = pandas.read_xml('TLSrecord.xml')`
`stageIndices`	`list`	Grouping and mapping of traffic movements to user-defined stage indices. This is a `list` of `int` for stage indices which starts from 0. Each element represents the defined stage of the corresponding movement. The order of traffic movements follows TLS state definition of SUMO.	`stageIndices = [0,0,0,1,1,1,0,0,0,1,1,1,1,0,1,0]`
`stageNames`	`list`	Names for the defined stages. This is a `list` of `str` labels, whose positions in this `list` correspond to the stage indices defined in `stageIndices`. The length of this `list` has to correspond to the total number of stage indices.	`stageNames = ['North-South','West-East']`

4) What optional settings are available?



- There are currently 4 optional keyword arguments to `clusterPlot_TLS`

Variable name	Object type	Description	Default value	Object instantiation example
`bar_colours`	`list`	List of bar colours for all plots except for the signal plan plot. This is a `list` of `str` for matplotlib colours. Note that the length of this `list` has to equal that of `stageNames`.	The default matplotlib colour cycle shown above	`bar_colours = ['c','m']`
`num_bins`	`int` : > 1	The number of bins for the histogram plot of green time distribution.	10	`num_bins = 6`
`cyclicity_type`	`int` : 1 or 2	Specify which type of cyclicity plot is to be created. Type 1: A vertical bar is stacked until the same stage is recurred. Type 2: A vertical bar is stacked until all defined stages are in the bar.	1	`cyclicity_type = 2`
`stage_type`	`Str` : 'mode' or 'first'	Which movements should represent the stage states? or should it be the statistical mode of the stage movements? At this moment, either using statistical mode or using the first movement from the left can be chosen to be the representative one.	'mode'	`stage_type = 'mode'`

5) Example with important notes

- An example can be found in `main.py` in the repo:

```
#Usage example
import pandas as pd
import matplotlib.pyplot as plt
from plot_functions import clusterPlot_TLS
tlsdf = pd.read_xml('TLSrecord.xml')
stageIndices = [0,0,0,1,1,1,0,0,0,1,1,1,1,0,1,0]
stageNames = ['North-South','West-East']
time_slider, button, radio1, radio2 = clusterPlot_TLS(tlsdf, stageIndices, stageNames,
                                                      bar_colours = ['c','m'])

plt.show()
```

Important note 1:
Reference to all 4 returned values have to be kept, otherwise the interactive plot functionality will not work.

Important note 2:
Depending on your python console, you will need to call `plt.show()`

6) Developer's notes

- Please note the following disparities between the terminology used in this code/documentation and the terminology used in SUMO.

This code/documentation	SUMO	Description
subStage	phase	
movement	link	
stage	-	User-defined, grouping of movements