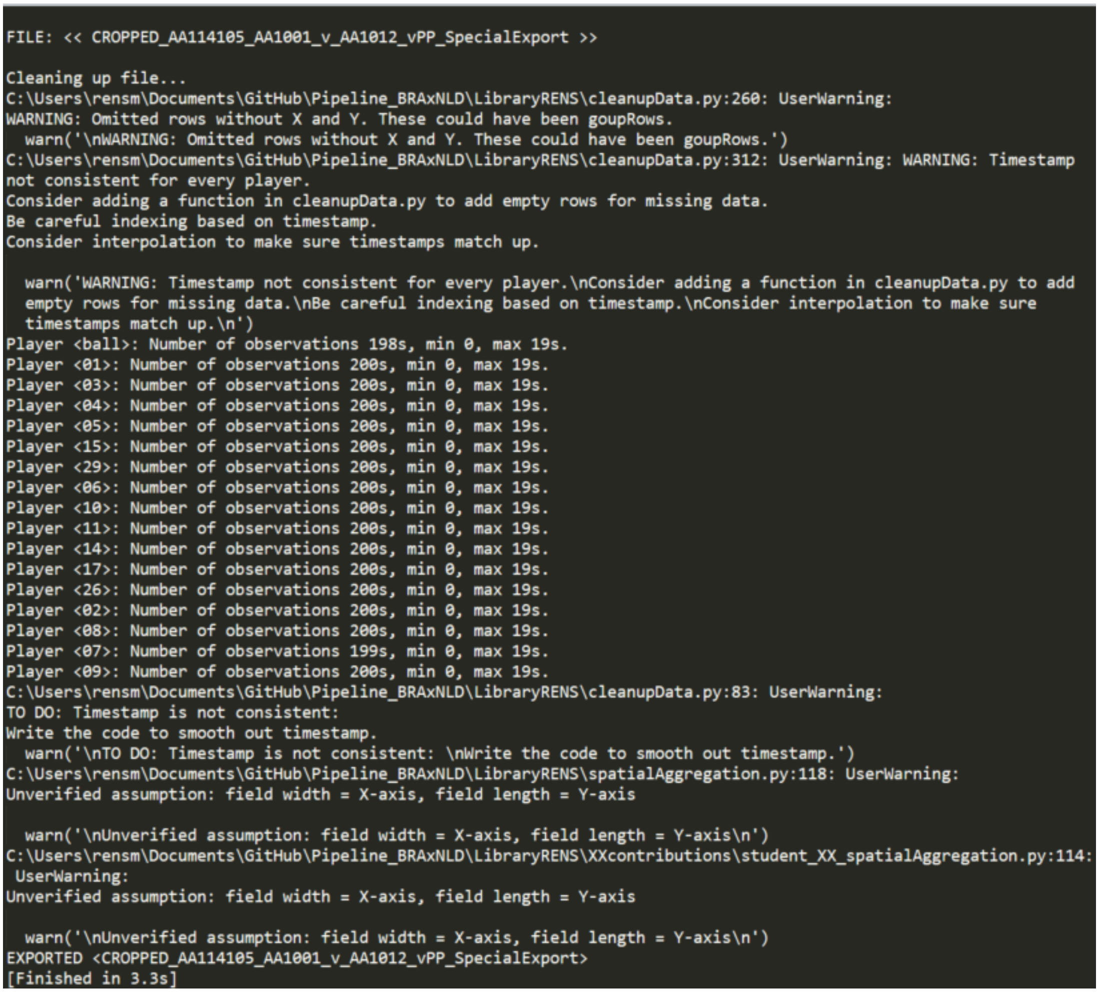
Updated 13-03-2018

Walkthrough – how to use the pipeline

Below, you find a short walkthrough describing step-by-step how to use the pipeline on your own database.

Make the pipeline work with the excerpt data:

1. Install Python (ask Google) and download an editor (e.g., Sublime, Spyder).
2. Download the files in the master on https://github.com/Rens88/Pipeline_BRAxNLD
3. Open "process_Template.py"
4. Change the string input for 'folder'. This folder should contain your data repository dedicated to the analysis. In this 'root' folder there should be a subfolder with the string <Data>. After running the pipeline it will additionally contain a folder <Output> and <Figs>.
5. Copy the excerpt from GitHub ("CROPPED_AA114105_AA1001_v_AA1012_vPP_SpecialExport.csv") to the subfolder <Data> in your 'folder'.
6. In "process_Template.py", Build and Run the script. You should now have a <Cleaned> folder in your <Data> subfolder of 'folder'. The <Cleaned> folder contains the standardized data that can be used for the analysis. NB: Cleaning only happens once. If you want to clean the data again, delete the cleaned file from the <Cleaned> folder. Also, you should see the following output:



```
FILE: << CROPPED_AA114105_AA1001_v_AA1012_vPP_SpecialExport >>

Cleaning up file...
C:\Users\rensm\Documents\GitHub\Pipeline_BRAxNLD\LibraryRENS\cleanupData.py:260: UserWarning:
WARNING: Omitted rows without X and Y. These could have been goupRows.
  warn('\nWARNING: Omitted rows without X and Y. These could have been goupRows.')
C:\Users\rensm\Documents\GitHub\Pipeline_BRAxNLD\LibraryRENS\cleanupData.py:312: UserWarning: WARNING: Timestamp
not consistent for every player.
Consider adding a function in cleanupData.py to add empty rows for missing data.
Be careful indexing based on timestamp.
Consider interpolation to make sure timestamps match up.

  warn('WARNING: Timestamp not consistent for every player.\nConsider adding a function in cleanupData.py to add
  empty rows for missing data.\nBe careful indexing based on timestamp.\nConsider interpolation to make sure
  timestamps match up.\n')
Player <ball>: Number of observations 198s, min 0, max 19s.
Player <01>: Number of observations 200s, min 0, max 19s.
Player <03>: Number of observations 200s, min 0, max 19s.
Player <04>: Number of observations 200s, min 0, max 19s.
Player <05>: Number of observations 200s, min 0, max 19s.
Player <15>: Number of observations 200s, min 0, max 19s.
Player <29>: Number of observations 200s, min 0, max 19s.
Player <06>: Number of observations 200s, min 0, max 19s.
Player <10>: Number of observations 200s, min 0, max 19s.
Player <11>: Number of observations 200s, min 0, max 19s.
Player <14>: Number of observations 200s, min 0, max 19s.
Player <17>: Number of observations 200s, min 0, max 19s.
Player <26>: Number of observations 200s, min 0, max 19s.
Player <02>: Number of observations 200s, min 0, max 19s.
Player <08>: Number of observations 200s, min 0, max 19s.
Player <07>: Number of observations 199s, min 0, max 19s.
Player <09>: Number of observations 200s, min 0, max 19s.
C:\Users\rensm\Documents\GitHub\Pipeline_BRAxNLD\LibraryRENS\cleanupData.py:83: UserWarning:
TO DO: Timestamp is not consistent:
Write the code to smooth out timestamp.
  warn('\nTO DO: Timestamp is not consistent: \nWrite the code to smooth out timestamp.')
C:\Users\rensm\Documents\GitHub\Pipeline_BRAxNLD\LibraryRENS\spatialAggregation.py:118: UserWarning:
Unverified assumption: field width = X-axis, field length = Y-axis

  warn('\nUnverified assumption: field width = X-axis, field length = Y-axis\n')
C:\Users\rensm\Documents\GitHub\Pipeline_BRAxNLD\LibraryRENS\XXcontributions\student_XX_spatialAggregation.py:114:
 UserWarning:
Unverified assumption: field width = X-axis, field length = Y-axis

  warn('\nUnverified assumption: field width = X-axis, field length = Y-axis\n')
EXPORTED <CROPPED_AA114105_AA1001_v_AA1012_vPP_SpecialExport>
[Finished in 3.3s]
```

Start editing the pipeline to suit your own analysis

The aim of this generic pipeline is to be able to easily combine all separate efforts. If you make your edits as much as possible within the dedicated sections, it will be much easier to merge new code with the existing pipeline. Therefore, try to stick to the input and output restrictions imposed by the templates.

NB: ‘XX’ refers to your initials that you can use to name your code.

The process template exists of the following sections:

1. Initialization
2. Analysis
3. Preparation
4. Import existing data
5. Compute new attributes
6. Export (work in progress)
7. Visualization (work in progress)

Section 01 User input

1. Make a copy of “process_Template.py” and name it as you like (e.g., “process_AnalysisXX.py”)
2. Create a copy of the subfolder <LibraryRENS\XXcontributions> and its contents. Rename it as you like. In this folder, you should find the most important modules which you may need to adjust to your own dataset. If you want to contribute to the pipeline, changes that you made in these modules are the easiest to be incorporated.
3. Change the user input (‘studentFolder’) to correspond to the name you gave the copy of the folder <XXcontributions>.
4. Start with a folder dedicated to the analysis (e.g., <XX repository>) that has a folder with the CSV files of the matches/trials that you want to analyze: <...\XX repository\Data> (see point 4. above).
5. Change the ‘datatype’.
    a. “FDP” (football data project) should work with inMotio data (with some dependency on how you exported it).
6. Change the rest of the user input, including the column header strings of the raw data and any attributes that are already in the data.

Section 02 Initialization

This section is off-limits. It simply loads some of the modules (automatically if you add them to your XXcontributions).

Section 03 Analysis

From here onward, each section is handled file by file.

Section 04 Preparation

This is the only section where you should use ‘dataType’, as preparing the data can be rather dataset specific. Nevertheless, try to make the preparation as generic as possible. When using ‘dataType’ “FDP”, you should be able to use inMotio data.

1. If your ‘dataType’ is different (or if it doesn’t work for some reason), the first module to write ‘dataType’ specific functions in “cleanupData.py”. In “student_XX_cleanUp.py” you can

make any changes that will be automatically incorporated in the main pipeline. Stick to making edits in this function only as much as possible.

   a. Note that although "cleanupData.py" is somewhat slow, it will only run when no cleaned file exists for the current file.

2. To organize your output, you may be able to dissect your filename. There is a template in <XXcontributions> called "student_XX_dissectFilename.py". You could use this to write a new way to dissect the filename (hint: use regular expressions). However, this function is only called when you're NOT using 'dataType' = "FDP". Feel free to make a minor change to "dissectFilename.py" to call your specific function whilst still using "FDP" as your 'dataType'.

Section 05 Import existing data

This section can only be accessed once the data is clean. If the cleaning goes well, then there should be no need to make changes in this section. In here, the dictionary keys of 'rawPanda' are always:

# 'Ts' --> Timestamp

# 'X' --> X-position

# 'Y' --> Y-position

# 'PlayerID' --> Player identification. NB: Ball-rows should be 'ball' and Match-rows should be 'groupRow' (to indicate CentroidTeamA)

# 'TeamID' --> Team idenfitification

Section 06 Compute new attributes

Here you can compute new attributes based on spatial aggregation. In the <XXcontributions> there is a template that you can use to compute new attributes and directly embed them in the pipeline. Try to -as much as possible – restrict making changes in your version of "student_XX_spatialAggregation.py".

Section 07 Export (work in progress)

Currently, I've included a shortcut that simply exports the data without aggregating it temporally. Soon (before the end of March), the temporal aggregation should be function as well. Then, the export will go through this section.

Section 08 Visualization (work in progress)

Soon, there will be some basic functionality of the visualization, which will help you visualizing any newly computed attributes.