SAIPy Package – Updates overview (release v2.0.0)

Here we provide a summary of changes since the first release (v1.0.0) and a list of new features, fixes, and structural changes. The trained models: **CREIME_RT, Dynapicker_v2,** and **PolarCap**, located in the **saved_models** folder, are preserved as originally in the first SAIPy release.

1. Structure of the Saipy project for the application of seismogram analysis.

```
saipy/
├── __init__.py
├── user_settings.py          # New
├── data/
│   ├── __init__.py
│   ├── base.py
│   └── realdata.py           # New functions
├── models/
│   ├── __init__.py
│   ├── creime.py
│   ├── dinapicker.py
│   └── polarcap.py
├── modules/
│   ├── __init__.py
│   ├── multi_stations.py     # New
│   ├── phaseclassification.py
│   └── pytorchtools.py
└── utils/
    ├── __init__.py
    ├── networktools.py       # New
    ├── packagetools.py       # New functions
    ├── picktools.py          # New functions
    └── visualization.py      # New functions
```

2. New Additions

- **user_settings.py**: includes user-defined functions related to the data format and the file name structures, which are used in the SAIPy's modules to load data, label plots and keys in dictionaries to store results.:
  - **set_filename**. *# Required*
  - **date_format_object**. *# Required*
  - **read_non_seismic_format**. *# Optional*

- **modules/multi_stations.py**: includes functions for event detection using multiple stations and the results are saved in a .csv file format. The clustering for the P arrival time association is based on temporal proximity of detections and arrival time pattern

in different stations (by default). It is possible to edit the code in case to prefer only temporal proximity for P arrival association.

- **utils/networktools.py**: includes the functions:
  - **make_monitor_multistations**: to automatically load, pre-process, and analyze data of every station in the defined seismic network. The results are stored into a dictionary for a posterior event detection by network.
    network_detect: to make the process for event detection by clustering of P arrival associated between multiple stations (minimum stations is set to 3 by default).

- New functions in **/data/realdata.py:**
  - **load_streams**: load obspy streams in 3 channels for the current station.

- New functions in **/utils/packagetools.py:**
  - **monitor**: replaces monitor1 and monitor2 (functions used in version 1). This function works with a stream Obspy as input data.

- New functions in **/utils/picktools.py:**
  - **phase_picking2**: uses input data in numpy array format. In this version the function does not include any filters, since the data are previously preprocessed.

- New functions in **/utils/visualizations.py:**
  - **plot_data.**
  - **plot_dynapicker_output.**
  - **plot_polarcap_output.**
  - **plot_multi_stations.**
  - **plot_overview.**

3. Modifications

i) Preprocessing process in **data/realdata.py** was modified to remove trend and mean, filter in a user-selected range (by default: 1-45 Hz), and to resample to 100 Hz only when is necessary.

ii) Function name changes:
- In **utils/pakagetools.py:**
  - data_windows —> windows_for_creime.
  - polarity_estimation —>polarity_classification.
- In **models/dynapicker.py:**
  - load_model —> load_dynapicker.

iii) Path to saved models is required as input for **CREIME_RT**, **PolarCAP,** and **load_dynapicker** functions (from **models/creime.py, models/polarcap.py,** and **models/dynapicker.py**, respectively).