

# MEDS Developer Guide

## IMEDS V 3.1/AutoMEDS

**Eve-Marie Devaliere**

### **Preface**

I'd first like to give a little more details on the IMEDS development. The IMEDS development has been an ever growing task; we started with only one data type: wave and two analysis types: temporal-correlation and quantile-quantile analysis. The data structure, however, was thought to be able to handle various data types that can be decomposed into components and attributes, following object oriented programming concepts. In the wave example, the component can be a wind-sea or a young swell and the attributes height, period and direction.

We then added two more data types: wind and water-level and two more analysis types: Peak Analysis (available for wind and wave) and Extreme Analysis (available for water-level).

As any software projects growing on the fly, things could have been planned differently would we have known what we now know. This document will give some information about the code itself and highlight parts of the code that I think could use some improvement.

IMEDS is the desktop version with the GUI and AutoMEDS is the automated version that can be run on the fly after a model runs. Both IMEDS and AutoMEDS run on the same set of core scripts. We will mostly refer to IMEDS in this documentation but everything that is not concerning the preprocessors and the display modules is true for AutoMEDS as well.

## Overview

As of version 3.1 IMEDS handles the following data types:

- Waves (from spectrum or bulk statistics)
- Wind
- Water-level

Four different types of analyses are available:

- Temporal-Corelation (TC) – for all data types
- Quantile-Quantile (QQ) - for wind and waves (no point for water-level)
- Peak Analysis (PK) – for wind and waves
- Extremes Analysis (EA) – for water-level (The water-level PK equivalent)

For more details on the different analysis types, please refer to Appendix 1.

We also built IMEDS so we could get performance scores by folding performances values in three different ways, referred as ‘type’ in the code and upcoming figures:

- ‘station’ : This is when we have many stations and want to obtain a performance score by combining the performance scores from each station. This is either for one month or for a single event.
- ‘month’: This is when we have one station covering many months and want to get a performance score by folding the performance scores from each month.
- ‘combo’: This is when we have many stations for many months and want to fold by months, by stations and both.

## Code Organization

The following figures (1 through 5) show the code organization of the main functions. Other functions, called by many programs, which I will refer to as ‘tool functions’, are not present on these diagrams.

The first diagram represents the global code organization. Three groups of functions have been identified and are presented in the subsequent diagrams:

- Pre-processing functions: Those are the functions responsible for the pre-processing. They take raw data from various observation data-sets or model output and transform it in an IMEDS friendly format.
- Part 1 processing functions: Those functions are processing each station for each month/event independently and create and save the corresponding statistics and plots if desired.
- Part 2 processing functions: Those functions are the ones folding the stations or months or both together. They are dependent upon the analysis type and the run type.

- Display functions: The display functions are the ones used in the different display module. Most of them are common to the three display modules but for the PK and EA related ones.

Optional functions (other than depending on the analysis type, data type or run type) are shown in italic and grey.

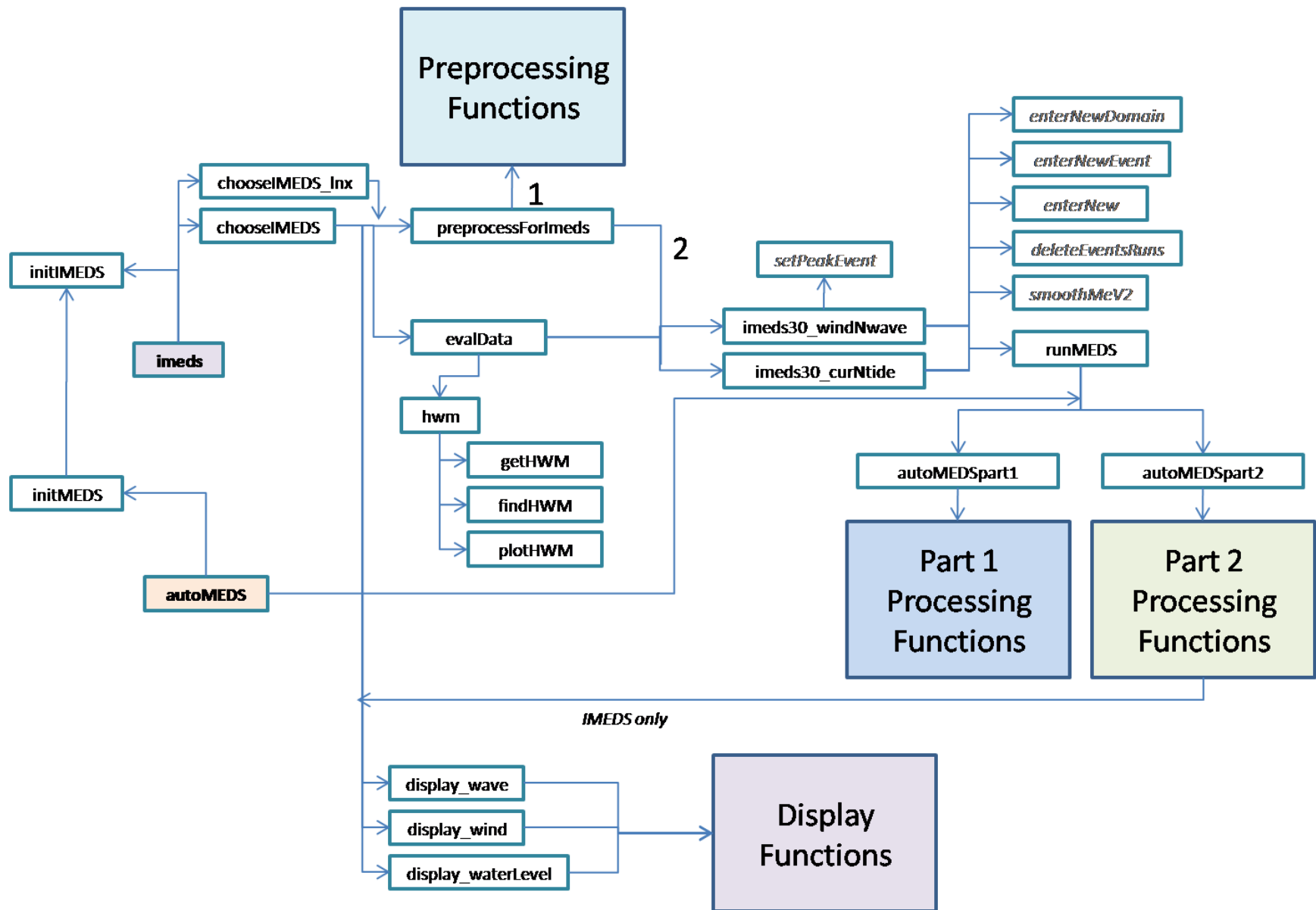


Figure 1 : Overall IMEDS Code Organization

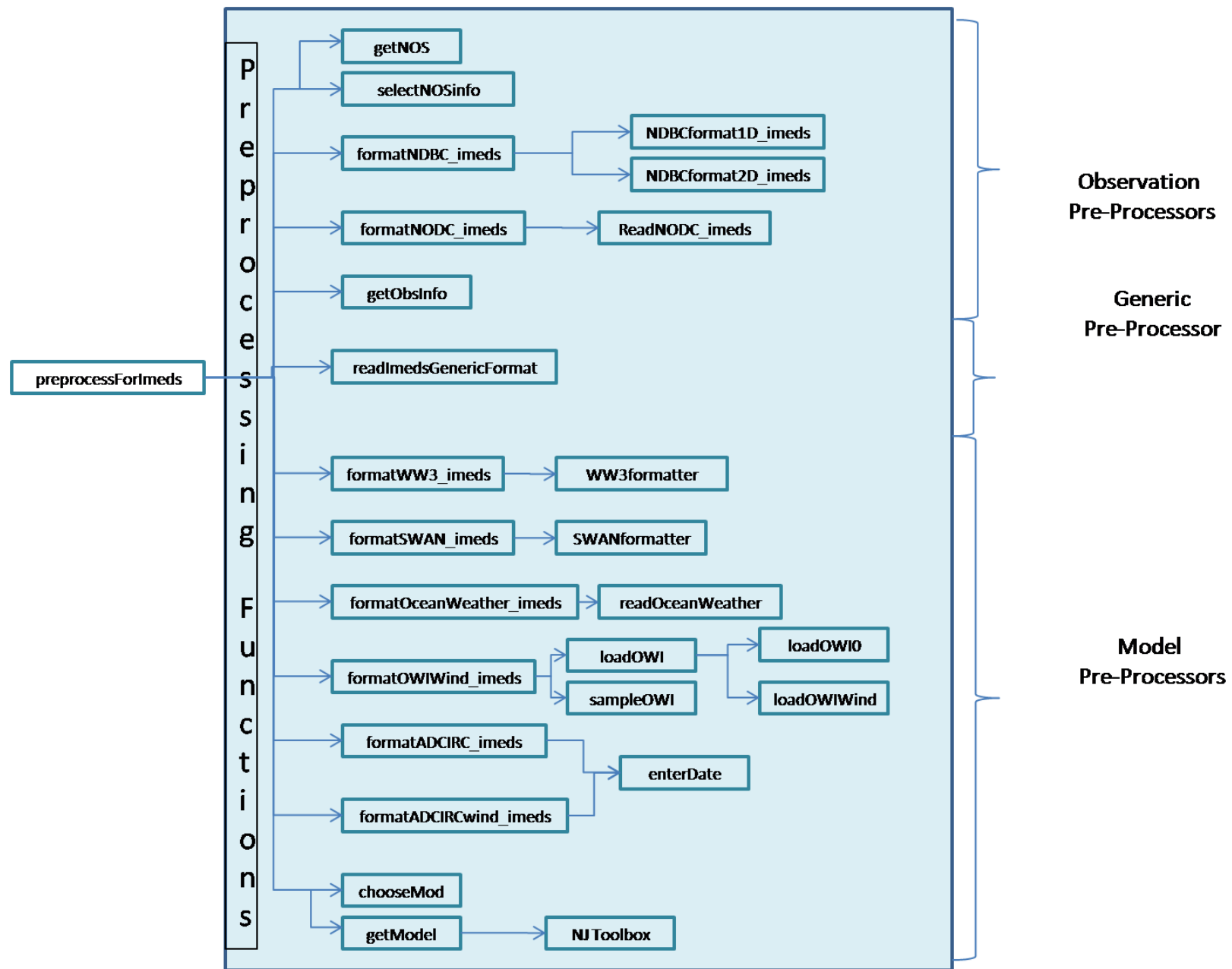


Figure 2 : IMEDS Pre-processing functions

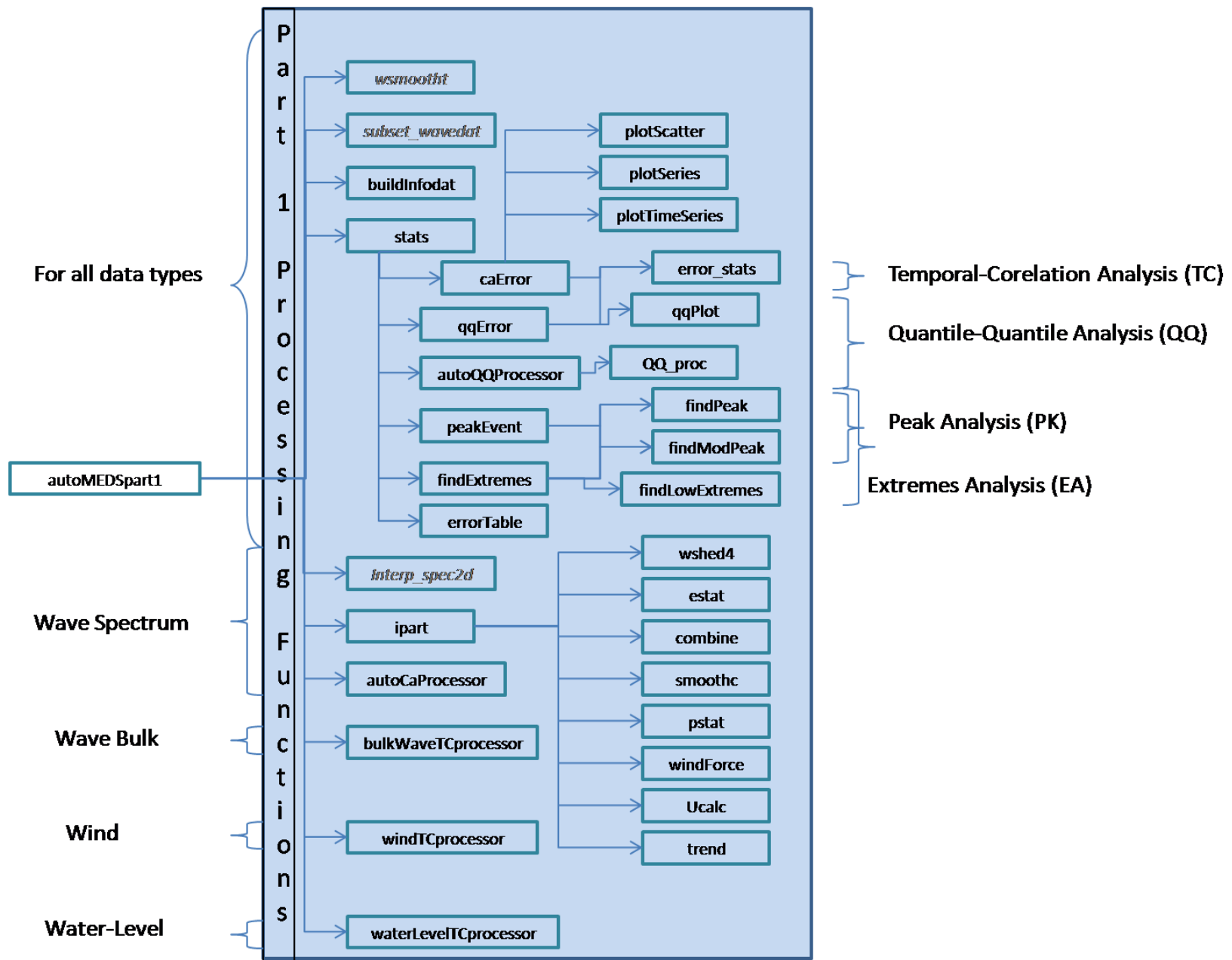


Figure 3: IMEDS Part 1 Processing Functions

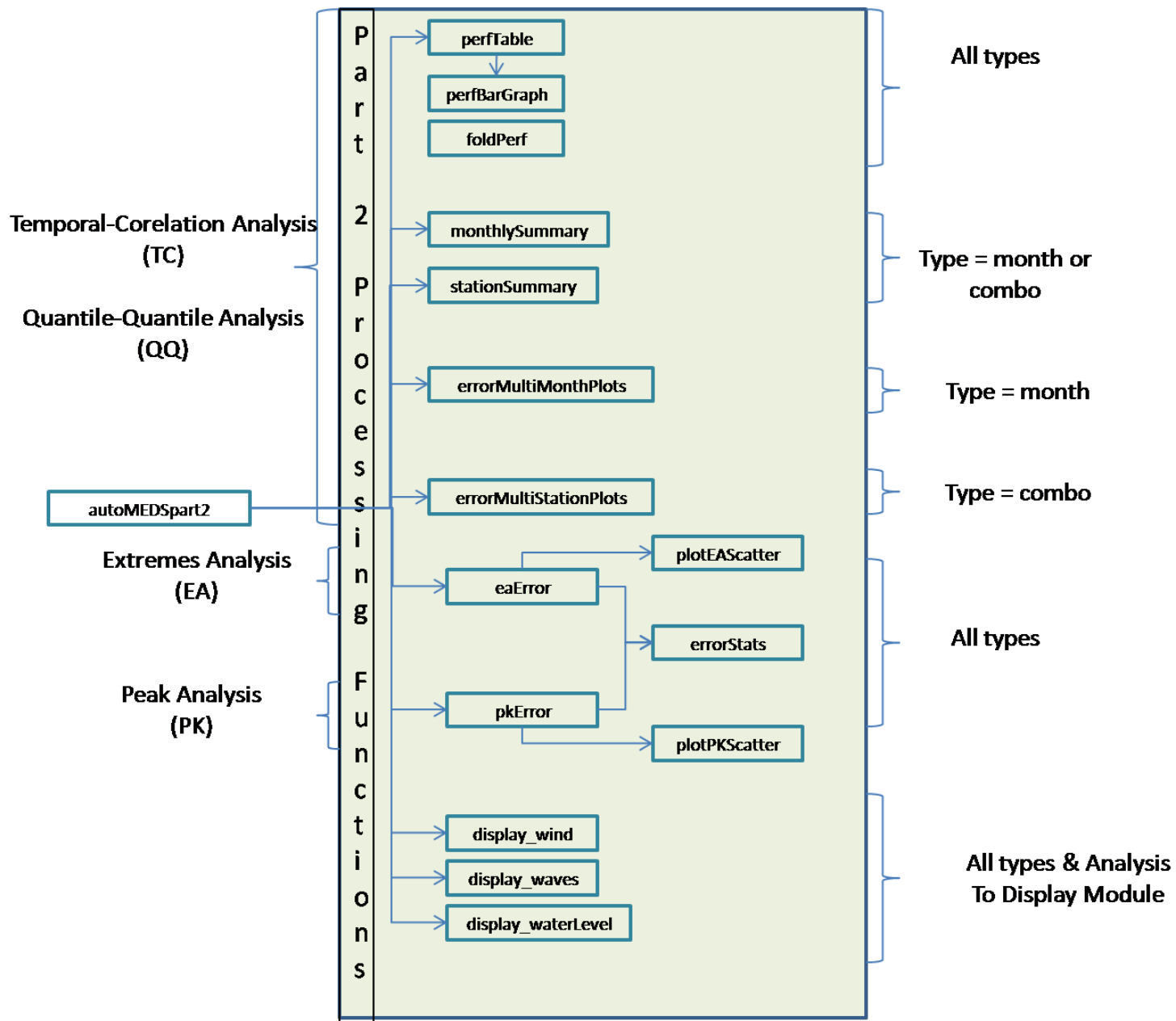


Figure 4: IMEDS Part 2 Processing Functions

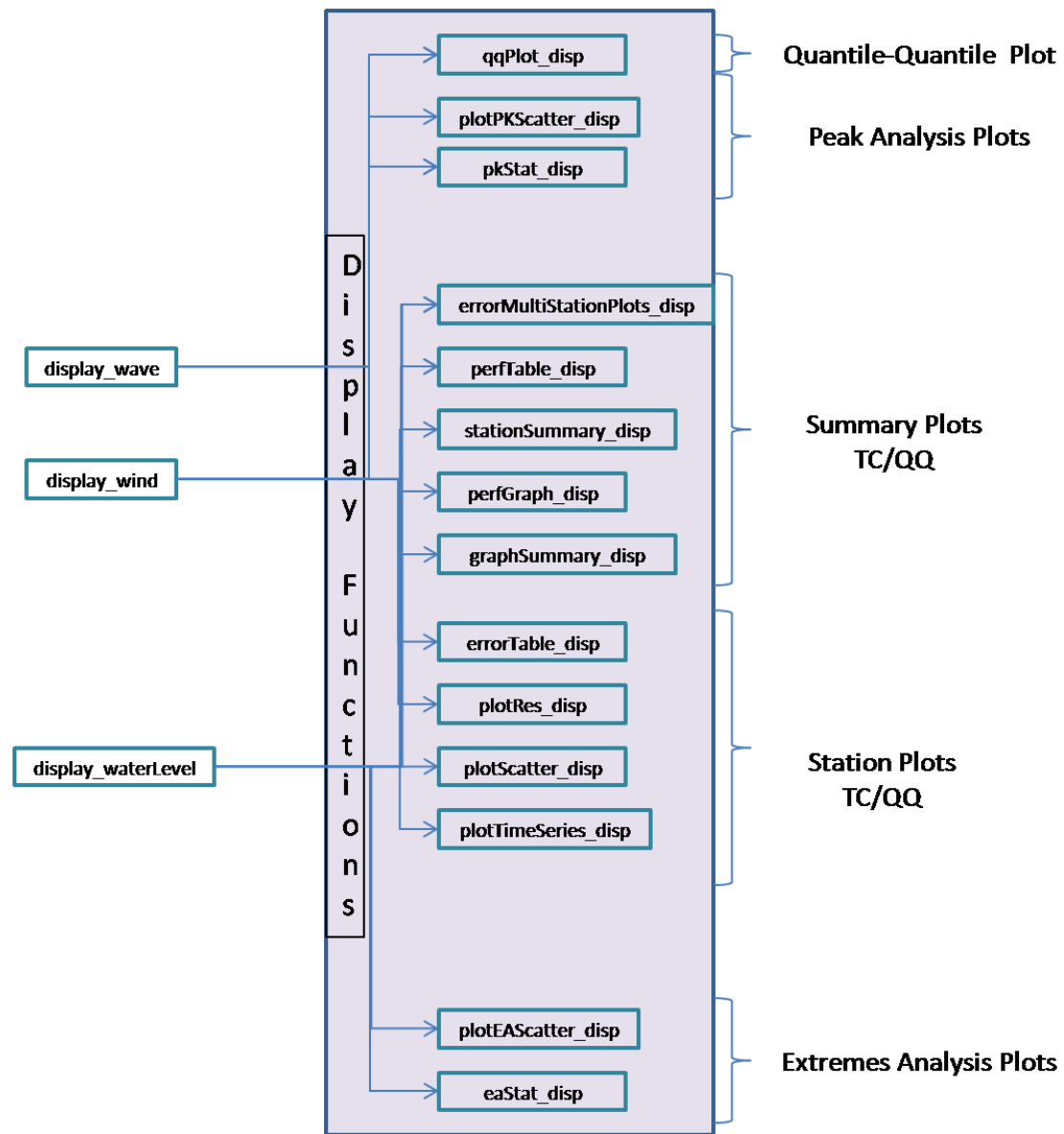


Figure 5: IMEDS Display Functions



Following the above logic, the IMEDS src directory holds the following subdirectories:

- lpart : wave processing functions developed by Wave Force Technologies. Only the Pcode of the program is available here.
- Toolboxes: external toolboxes used in IMEDS
  - M\_map: for mapping purposes
  - NJ\_toolbox: for extracting data from a remote netCDF file
- preProcess: holds the preprocessing functions
- display: holds the functions for the display modules
- process: all the MEDS processing functions (start, init, part1, part2), including the plotting functions
- tools: functions used throughout IMEDS

## Improvements

Some parts of the code could, I believe, use some improvements:

### The NetCDF extractor

- This version allows for one time vector only (It will however warn the user if more than one time variable is available) - This needs to be handled more gracefully in the next version. However, if different time vectors are available, an issue might be raised too as the IMEDS internal format only holds one time vector. In IMEDS, it is for now possible to treat wind and waves together (or separately) and water-level on its own, because of the nature of observations. As a matter of fact, the observation files hold only waves (and potentially wind) or water-level information, not both. Since models like CH3D output both bulk wave parameters and water-level information one would need to run the NetCDF extractor twice for CH3D. (Once to extract water-level information where water-level stations are available, and once to extract wave information where wave stations are available). This is something that may need to be reconsidered in the future if this is not acceptable.
- The only extraction method developed so far is a closest neighbor search, more options, such as bilinear interpolation should be added in future versions.
- The module dynamically creates a Wavedat file (internal IMEDS format) from the data extracted from the NetCDF files using a saved Matlab structure to know which SURF variable to match with which IMEDS variable. This may have to be revisited if naming conventions haven't been set and a given SURF variable name can differ across models. We will also need to add a wind option, and eventually a spectra option if wave spectrum is stored in NetCDF files.
- Variables from the 'SURF models', such as currents, water temperature are not part of IMEDS processing (yet). Therefore, the model data extraction will be limited for now.
- There is a need for another njToolbox function returning a time series at a specified point - for now the getData function from the njToolbox was used but that assumes that the array shape of the data set is known. Therefore, for now, we had to use time/lat/lon keywords to dynamically figure out which dimension of the data array each of the coordinates and time attributes were using.

- Work needs to be done on dynamically parsing the SURF XML file from the OpenDAP server listing the available models and their location so the preprocessor can hold the full list of available models.

- Another issue was found at the last minute. Some netCDF files are not processed correctly by njGetDimensions. It appears that the Java function getGrids doesn't return any variable for those files... specifically this is called at line 27 (vars=grid.getGrids();) in getDimensions.m in the @mDataset folder... An example of file bringing the issue can be downloaded there:

[http://testbedapps.sura.org/thredds/fileServer/inundation\\_scan/umass/fvcom\\_extratropical/2007/case2\\_wave\\_only/output/sci\\_cold\\_waveonly\\_0001.nc](http://testbedapps.sura.org/thredds/fileServer/inundation_scan/umass/fvcom_extratropical/2007/case2_wave_only/output/sci_cold_waveonly_0001.nc)

The CH3D files seem to always work fine...

## The display modules

It would be very useful if one could find a way to combine the 3 display modules into one. If this is kept in Matlab, I would suggest getting away from the radio buttons and putting pull down menus instead, as the entries can easily be created on the fly.... Since PK and EA don't seem like they will ever be run at the same time, the module could have three tabs: TC, QQ and PK/EA depending on the data type

## Handling of More Data Types

IMEDS also needs to handle more data types. This is pretty straightforward and new entries need to be made in initIMEDS. One of the complications would be with the associated display module, which is one more reason why this should be reviewed

## Information Structures

IMEDS has 3 main structures to pass information along during the run.

- AMsetup/stp : This is the structure that is defined during the Run Setup in IMEDS (and with the setup file in AutoMEDS)
- imeds: This structure helps keeping track of :
  - paths,
  - information used by the GUIs,
  - the different components for each data-type and the corresponding attributes (new entries have to be entered here if a new data type is added to IMEDS)
  - markers and colors associated to each compo/attribute pair
  - time-zones

This is defined in initIMEDS.m

- infoDat: This is the structure that mostly holds information about a certain station, such as station name, month and year. It also has some of the imeds structure information about attributes and components setup because this information was needed after the fact in programs where the imeds structure was not available. This is defined in buildInfodat.m and autoMEDSpart1.m

As mentioned in the above description, some information that should belong in one structure has been entered or duplicated in another because the programs were set up in such a way that

wouldn't allow for the first structure to be present. This, I think, should be revisited for a more robust set of scripts.

## **autoMEDSpart2**

I think this part can be better streamlined. The different loops could be reorganized to only load each file once.

## **Appendix 1: IMEDS Analyses Types**

The four statistical approaches available in IMEDS are Temporal Correlations (TC), Quantile-Quantile (QQ), Peak Event (PE) and Extreme Analysis (EA). The results of each of these are folded into a Performance Score computation. Each of these computations is briefly described below.

### **Temporal Correlation (TC) Analysis**

The TC analysis is a direct comparison of time-paired data attributes. The TC analysis provides an indication of how well the hindcast quantities match the observed quantities in absolute time. The following metrics are used to quantify the TC errors:

For non-directional data (speed, time, height and period) the error metrics are:

- RMS error
- Bias
- Scatter index.

For directional data the error metrics are:

- Circular correlation
- Circular bias

### **Quantile-Quantile (QQ) Analysis**

The QQ analysis provides information on how the distribution of data attributes compare between observation and model results. Quantile-Quantile distributions computed for both data sets (observed and modeled) are statistically compared to compute a set non-directional error metrics (see list above).

### **Peak Event (PE) Analysis**

The PE analysis extends the IMEDS capability by isolating and computing statistics on event peak data. User-provided thresholds (constants or Standard Deviation multipliers) are used to identify event peaks in wind and wave records. A standard up-crossing analysis is used to isolate time series segments that contain relevant data. Corresponding peaks are identified in the test data using a user-provided time search threshold. Only the bulk (full-spectrum) statistics are used for this analysis. The attributes extracted from each peak event form data pairs that are then used to compute the standard non-directional error metrics (see list above).

## Extremes Analysis (EA)

Similar to the IMEDS Peak Event analysis for winds and waves, the water-level Extreme Analysis (EA) identifies peaks (and lows) and computes error statistics on the extreme highs and lows during the tidal cycle and/or over the passage of a storm.

## Performance Scores

This above analyses result in a set of error metrics that quantify the hindcast skill in reproducing the physical attributes at each observation station. For a 1-year wave study with 6 stations this can result in a database of 3,500 independent measures of model skill for each hindcast run. A performance scoring method was developed to reduce the error metric database into a small set of performance indicators for overall skill assessment. Performance scores are computed by normalizing the wave component metrics to mean quantities and averaging them across metrics, months and stations with contributions weighted by sample size. The resulting non-dimensional performance scores range from 0 (uncorrelated) to 1 (perfect correlation) and relate to the fraction of the mean that is not impacted by error. So a performance score of 0.8 can be interpreted that error levels are within approximately  $(1-0.8)*100 = 20\%$  of the attribute means.