



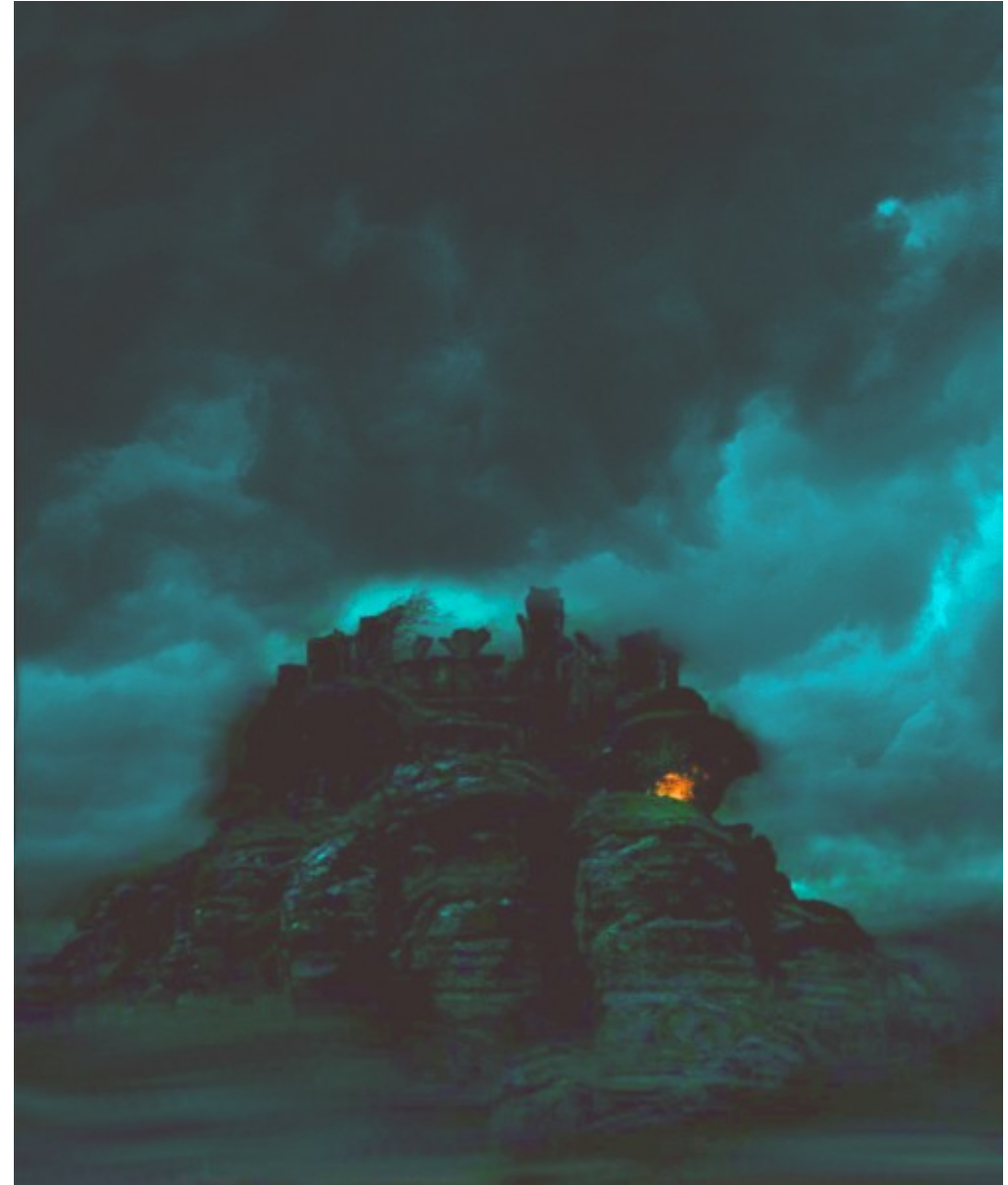
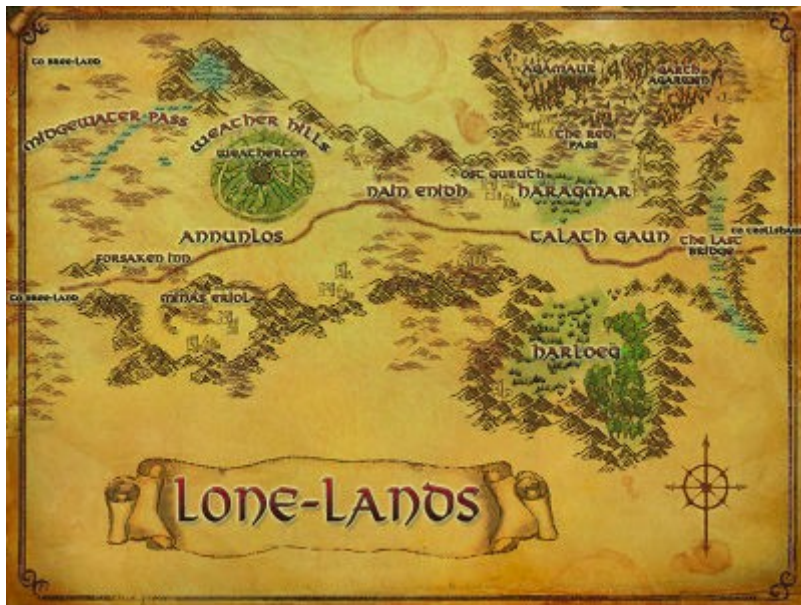
WeatherHills post-processing system

Karolina Stanisławska

Seychelles, 24 February 2016



WeatherHills post-processing system



Components

- PointForecast
- QualityAssurance
- ObservationImporter
- GridForecastImporter
- Data parsers and utility scripts



PointForecast

point_forecast.py: produces point forecasts for given stations/POIs from a wrfout file and saves them as csv files

xml_converter.py: reads point forecasts and creates an xml file compatible with the yr.no format, to be displayed as meteograms

weights_linear_regression.py: produces linear regression weights based on comparison of observations and forecasts

weights_bilinear.py

weights_nearest.py

forecast_stats.py: generate timeseries of forecast error and running RMS and save them to a csv file



QualityAssurance

qa.py: runs the quality check

| time | station_id | variable | value | q_range | q_step | q_persistence | q_spatial |
|------------------------|------------|------------|-------|---------|--------|---------------|-----------|
| 2010-07-02 11:00:00+00 | 908 | rel_hum | 106 | 3 | 8 | 8 | 8 |
| 2010-07-02 11:00:00+00 | 908 | temp | 3 | 0 | 0 | 0 | 0 |
| 2010-07-02 11:00:00+00 | 908 | wind_dir | 132 | 0 | 0 | 0 | 0 |
| 2010-07-02 11:00:00+00 | 908 | wind_speed | 3.2 | 0 | 0 | 0 | 0 |
| 2010-07-02 12:00:00+00 | 908 | rel_hum | 106 | 3 | 8 | 8 | 8 |
| 2010-07-02 12:00:00+00 | 908 | temp | 3.3 | 0 | 0 | 0 | 0 |
| 2010-07-02 12:00:00+00 | 908 | wind_dir | 183 | 0 | 0 | 0 | 0 |
| 2010-07-02 12:00:00+00 | 908 | wind_speed | 1.7 | 0 | 0 | 0 | 0 |
| 2010-07-02 13:00:00+00 | 908 | rel_hum | 103 | 0 | 8 | 0 | 0 |
| 2010-07-02 13:00:00+00 | 908 | temp | 3.4 | 0 | 0 | 0 | 0 |
| 2010-07-02 13:00:00+00 | 908 | wind_dir | 156 | 0 | 0 | 0 | 0 |
| 2010-07-02 13:00:00+00 | 908 | wind_speed | 3 | 0 | 0 | 0 | 0 |



ObservationImporter

import_files.py + *parsers*: import observations to database using different parsers

```
class ExampleObservationParser(SeparatedTextObservationParser):

    def __init__(self):
        super(ExampleObservationParser, self).__init__(separator=',', headers='^[^,]*[a-zA-Z]+^[^,]*([,]*[a-zA-Z]+)*$')
        self.meta['provider_ref'] = 'met.sc'

    def parse_header_line(self, line):
        if not super(ExampleObservationParser, self).parse_header_line(line):
            return False
        if line != '':
            parts = self.separator.split(line.strip())
            self.field_list = [('time', parse_time, parts.index('DD/MM/YYYY HH:MM')), ('station_ref', str, parts.index('Site'))]
            for key_in, key_out in FIELD_ALIASES.iteritems():
                if key_in in parts:
                    self.field_list.append((key_out, float_or_no_data, parts.index(key_in)))
        return True

    def parse_data_line(self, line):
        return super(ExampleObservationParser, self).parse_data_line(line.strip())
```



GridForecastImporter

gfi.py: import wrfout files to the database



The database

```
karolina@lilmachine: ~/Downloads/pycharm-4.0.6/bin x
File Edit View Search Terminal Help
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | grid | table | karolina
public | grid_forecast | table | karolina
public | grid_forecast_data | table | karolina
public | grid_point | table | karolina
public | observation | table | karolina
public | observation_quality | table | karolina
public | point_forecast_data | table | karolina
public | provider | table | karolina
public | schedule | table | karolina
public | station | table | karolina
public | wind_power | table | karolina
(11 rows)
(END)
```

