

PHOENIX Stellar Atmosphere Analysis Pipeline

This pipeline provides a set of tools for analyzing PHOENIX stellar atmosphere models, calculating quality metrics, and optimizing model parameters.

Features

- **Quality Calculation:** Calculate quality metrics by comparing synthetic PHOENIX spectra to observed spectra
- **Global Minimum Finding:** Find the optimal T_{eff} and $\log g$ values using multiple optimization algorithms
- **Periodicity Analysis:** Detect periodicities in the data and recommend additional sampling points
- **Neural Periodicity Analysis:** Advanced pattern detection using neural networks and machine learning
- **Z-Scale Optimization:** Analyze models with different z-scale values to find the optimal z-scale
- **Comprehensive Reporting:** Generate detailed reports and visualizations

Installation

Prerequisites

- Python 3.7 or higher
- PHOENIX model files in H5 format
- Observed spectrum file (default: `uves_spectra_fomalhaut.csv`)

Install from source

```
git clone https://github.com/yourusername/phoenix-analysis-pipe.git
cd phoenix-analysis-pipe
pip install -e .
```

Usage

Command Line Interface

```
python phoenix_analysis_pipe.py \
  --nlte-dir /path/to/nlte/models \
  --z-scale-dir /path/to/z_scale/models \
  --quality-dir /path/to/quality/output \
  --image-dir /path/to/image/output \
  --report-dir /path/to/report/output
```

With Configuration File

```
python phoenix_analysis_pipe.py --config config.json
```

Options

- `--nlte-dir`: Directory containing NLTE model files (H5 format)
- `--z-scale-dir`: Directory containing models with different z-scales
- `--quality-dir`: Directory for storing quality calculation results
- `--image-dir`: Directory for storing image outputs
- `--report-dir`: Directory for storing reports and predictions
- `--config`: Path to JSON config file containing pipeline parameters
- `--recalculate-all`: Recalculate quality for all NLTE models
- `--skip-z-scale-min`: Skip z-scale minimum calculation
- `--skip-prediction`: Skip prediction analysis
- `--skip-neural-prediction`: Skip neural prediction analysis
- `--original-spectrum`: Path to original spectrum file (default: `uves_spectra_fomalhaut.csv`)
- `--threads`: Number of threads to use (default: number of CPU cores - 1)
- `--log-level`: Logging level (DEBUG, INFO, WARNING, ERROR, CRITICAL)

Input File Requirements

NLTE Model Files

- H5 files with naming convention: `nlte_TEFF_LOGG_ZSCALE.h5`
- Example: `nlte_07500_4.00_p0.0.h5`
- All files in the NLTE directory should have the same z-scale

Z-Scale Model Files

- H5 files with naming convention: `nlte_TEFF_LOGG_ZSCALE.h5`
- For each teff/logg pair, at least 4 different z-scale values are needed for reliable optimization

Observed Spectrum File

- CSV file with columns for wavelength (WL) and flux (Flux)
- Default: `uves_spectra_fomalhaut.csv`

Output

The pipeline generates several outputs:

1. **Quality Calculation Results**: JSON files with quality metrics for each model
2. **Global Minimum Analysis**: Reports and visualizations of the optimal Teff and logg values
3. **Periodicity Analysis**: Detected periodicities and recommended sampling points
4. **Neural Periodicity Analysis**: Advanced pattern detection and additional sampling recommendations
5. **Z-Scale Optimization**: Optimal z-scale values for each teff/logg pair
6. **Comprehensive Reports**: Summary reports combining insights from all analyses

Contributing

Contributions are welcome! Please feel free to submit a Pull Request.

License

This project is licensed under the MIT License - see the LICENSE file for details.