Filename: "README.md"

Contest Log Analyzer

Version: 0.26.3-Beta Date: 2025-08-03

A Python-based tool for in-depth analysis and comparison of amateur radio contest logs. This application processes standard Cabrillo files to generate detailed reports, charts, and visualizations, providing deep insights into operator strategy and performance.

Key Features

- **Data-Driven Architecture**: Uses simple JSON files to define the rules, scoring, and exchange formats for each contest, making the tool highly extensible.
- **Run/S&P Heuristics**: A sophisticated, multi-pass heuristic analyzes QSO timing and frequency to classify each contact as "Run," "Search & Pounce," or "Unknown," providing a clear picture of operating strategy.
- **Unique vs. Common QSO Analysis**: The analyzer precisely identifies "unique" QSOs (worked by only one of two logs) and "common" QSOs (worked by both), breaking them down by Run/S&P status to reveal strategic advantages.
- **Cumulative Difference Plots**: Goes beyond traditional rate graphs by presenting QSO and Point rate data in "Cumulative Difference Plots," which visualize performance trends and momentum shifts much more clearly.
- **Annotated CSV Output**: Generates detailed, "annotated" CSV files from the processed logs, perfect for loading into Excel or other tools for custom analysis and prototyping.
- **Contest-Specific Scoring**: A modular system calculates QSO points based on the official rules for supported contests (ARRL-DX, ARRL-SS, CQ-WPX, CQ-WW).
- **Dynamic Reporting Engine**: A flexible, "plug-and-play" system for generating a wide variety of text, plot, and chart-based reports.

Installation

This project uses `conda` for environment and package management.

1. Clone the Repository

```
git clone https://github.com/kd4d/Contest-Log-Analyzer.git "Contest-Log-Analyzer"
cd "Contest-Log-Analyzer"
```

2. Create and Activate Conda Environment

It is recommended to use Miniforge and create a dedicated environment.

```
conda create --name contest-analyzer python=3.11 -y
conda activate contest-analyzer
```

3. Install Dependencies

Update the base packages and then install the required libraries.

```
conda update --all -y
conda install pandas matplotlib seaborn -y
```

4. Set Up Environment Variable

The program requires the `CONTEST_DATA_DIR` environment variable to be set to the location of your data directory.

- **Windows (Temporary):**

  ```
  set CONTEST_DATA_DIR="C:\path\to\your\Contest-Log-Analyzer\data"
  ```

- **macOS/Linux (Temporary):**

  ```
  export CONTEST_DATA_DIR="/path/to/your/Contest-Log-Analyzer/data"
  ```

5. Download Data Files

Place the necessary data files in a central `data/` directory.

- **Required for all contests:** `cty.dat` (from country-files.com)
- **Required for ARRL DX:** `ARRLDXmults.dat`
- **Required for ARRL SS:** `SweepstakesSections.dat`

Usage

The analyzer is run from the command line using `main_cli.py` .

Basic Syntax

```
python main_cli.py --report <ReportID|all> <LogFile1> [<LogFile2>...] [options]
```

Examples

- **Generate all available reports for two logs:**

  ```
  python main_cli.py --report all Logs/2024/cq-ww-cw/k3lr.log Logs/2024/cq-ww-cw/kc1xx.log
  ```

- **Generate a specific report (Score Summary) for a single log:**

  ```
  python main_cli.py --report score_report Logs/2024/cq-ww-cw/k3lr.log
  ```

- **Generate a Missed Multipliers report for CQ WW Zones:**

  ```
  python main_cli.py --report missed_multipliers --mult-name Zones Logs/2024/cq-ww-cw/k3lr.log Logs/2024/cq-ww-cw/kc1xx.log
  ```

Available Reports

All generated files are saved to a structured directory under `reports_output/YYYY/CONTEST_NAME/` .

Text Reports (`text/`)

- `summary` : High-level overview of QSO counts (Run, S&P, Unknown).
- `score_report` : Comprehensive score breakdown by band for a single log.
- `rate_sheet` : Detailed hourly QSO rates per band for a single log.
- `rate_sheet_comparison` : Side-by-side hourly rate comparison for multiple logs.
- `qso_comparison` : Detailed pairwise breakdown of Total, Unique, and Common QSOs.
- `missed_multipliers` : Comparative report showing multipliers missed by each station.
- `multiplier_summary` : Detailed breakdown of QSOs per multiplier.
- `multipliers_by_hour` : Shows new multipliers worked each hour of the contest.
- `continent_summary` : Total QSOs per continent for a single log.
- `comparative_continent_summary` : Side-by-side comparison of QSOs per continent.
- `continent_breakdown` : Detailed QSOs per continent broken down by Run/S&P status.

Plots (`plots/`)

- `qso_rate_plots` : Cumulative QSO rate line graphs.
- `point_rate_plots` : Cumulative point rate line graphs.
- `cumulative_difference_plots` : Plot showing the running QSO or Point difference between two logs.

Charts (`charts/`)

- `qso_breakdown_chart` : Stacked bar chart comparing unique QSO counts for two logs.
- `chart_point_contribution` : Side-by-side pie charts comparing point sources.
- `chart_point_contribution_single` : Per-band pie charts showing point sources for one log.

License

This project is licensed under the **Mozilla Public License, v. 2.0**.