Contest Log Analyzer

Version: 0.35.21-Beta Date: 2025-08-15

--- Revision History ---

[0.35.21-Beta] - 2025-08-15

Changed

- Updated "Key Features" and "Available Reports" to include the

new hourly animation report.

[0.31.0-Beta] - 2025-08-11

Added

- Initial release of the README.md file.

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.
-

Animated Hourly Replay: Generates an MP4 video that visualizes the entire contest on an hour-by-hour basis, showing cumulative scores, QSO rates, and band-by-band totals for up to three logs.
- **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.

Usage

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

Basic Syntax

```
python main_cli.py --report <ReportID|all|chart|text|plot> <LogFile1> [<LogFile2>...] [option
```

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.
  ```

- **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/k
  ```

Available Reports

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

Animation Reports (`animations/`)

- `hourly_animation`: Hourly Rate Animation

Chart Reports (`charts/`)

- `chart_point_contribution`: Point Contribution Breakdown (Comparative)
- `chart_point_contribution_single`: Point Contribution Breakdown (Single Log)
- `qso_breakdown_chart`: QSO Breakdown by Run/S&P

Plot Reports (`plots/`)

- `cumulative_difference_plots`: Cumulative Difference Plot
- `point_rate_plots`: Cumulative Point Rate Plot
- `qso_rate_plots`: Cumulative QSO Rate Plot

Text Reports (`text/`)

- `comparative_continent_summary`: Comparative Continent Summary
- `comparative_score_report`: Comparative Score Report
- `continent_breakdown`: Continent Breakdown by Run/S&P
- `continent_summary`: Continent Summary
- `missed_multipliers`: Missed Multipliers
- `multiplier_summary`: Multiplier Summary
- `multipliers_by_hour`: Multipliers by Hour
- `qso_comparison`: QSO Comparison Summary
- `rate_sheet`: Rate Sheet (per hour)
- `rate_sheet_comparison`: Rate Sheet Comparison
- `score_report`: Score Report
- `summary`: QSO Summary by Run/S&P

License

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