RF Data Analysis - Documentation

Lukka Ladmann

July 8, 2024

1 Directory Structure

The project directory is organized as follows:

- `Files_for_SW_Dev_application/`: Contains the main RF data files.

- `plots/`: Contains the generated plots.

- `carriers/`: Contains the JSON files with carrier information.

- `rtsa_decoder/`: Contains the RTSA decoder files.

- `coding_exercise_SW_dev.ipynb`: Jupyter notebook with the analysis code.

2 Code Description

The code begins by calculating a noise threshold using multiple standard deviations above the mean from provided noise samples. It then processes each data file by generating and plotting a heatmap, flattening the matrix to create a max hold array, and downsampling this array. Gradients of the downsampled max hold are calculated and smoothed to make adjacent gradients more detectable. Carriers are detected by identifying significant gradients within specified ranges, and for each detected carrier, the code adjusts indices for downsampling, calculates time ranges, and determines the maximum power value. Finally, it compiles this information into a JSON file for each data file.

3 Running the Analysis

To run the analysis, follow these steps:

1. Install the necessary Python packages:

   `pip install -r requirements.txt`

2. Ensure the Jupyter notebook `coding_exercise_SW_dev.ipynb` is in the root of the directory.

3. Open the Jupyter notebook:

   ```
   jupyter notebook coding_exercise_SW_dev.ipynb
   ```

4. Run the cells in the notebook to perform the analysis.

4   Acknowledgment

The creation of plots and a bit of code cleaning were assisted by ChatGPT-4.