ClaritySpecra User Manual

Table of Contents

1. Introduction

The Raman Spectrum Analysis Tool is a comprehensive software package for importing, analyzing, and identifying Raman spectra. This application is designed for researchers, geologists, materials scientists, and other professionals who work with Raman spectroscopy data.

1.1 Key Features

- Import Raman spectra from various file formats
- Baseline correction and peak detection
- Spectrum database management
- Advanced search and matching algorithms
- Hey Classification integration for mineral identification
- Customizable visualization options
- Export capabilities for reports and results

2. Installation

2.1 System Requirements

- **Operating System**: Windows 10/11, macOS 10.14+, or Linux
- **Processor**: 1.6 GHz or faster
- **RAM**: 4 GB minimum (8 GB recommended)
- **Storage**: 500 MB free space
- **Display**: 1280 x 800 or higher resolution

2.2 Installation Steps

1. Download the installer package from the official website
2. Run the installer and follow the on-screen instructions
3. Launch the application from your Start menu, Applications folder, or desktop shortcut

2.3 Manual Installation (Advanced Users)

If you prefer to install from source:

1. Ensure Python 3.8+ is installed on your system
2. Clone the repository: git clone https://github.com/username/raman-analyzer.git
3. Navigate to the project directory: `cd raman-analyzer`
4. Install required dependencies: `pip install -r requirements.txt`
5. Run the application: python raman_analysis_app.py

2.4 Database Installation

The database is too large for me to have on GitHub (25MB limit). There two options:

1. Download a precompiled database with Hey Index classification already down
    1. This is the easiest: goto https://drive.google.com/file/d/1LZ8Tp0jGij4VUVILiDvyl9RFyf7H8KaW/view?usp=drive_link
    2. Put this file in the same folder as all the *.py files. ClaritySpec will recognize and and load it (don't change the filename).
2. Build the database yourself
    1. You only need a bunch of spectra in a folder.
        1. You can get these from rruff.info
        2. Goto the Database tab and in the Database management area, click on Batch Import Spectra
            1. don't worry if this looks like it is not responding, depending on the database size (RRUFF is about 6,000 spectra) it could take 20 or 30 minutes. You can can see the *.pkl file changing size as the database is changing. That is your indication that things are ok.
        3. Once that is done, you can add the Hey Classification by clicking the Update Hey Classification. This is pretty quick.

3. Getting Started

3.1 Application Interface Overview

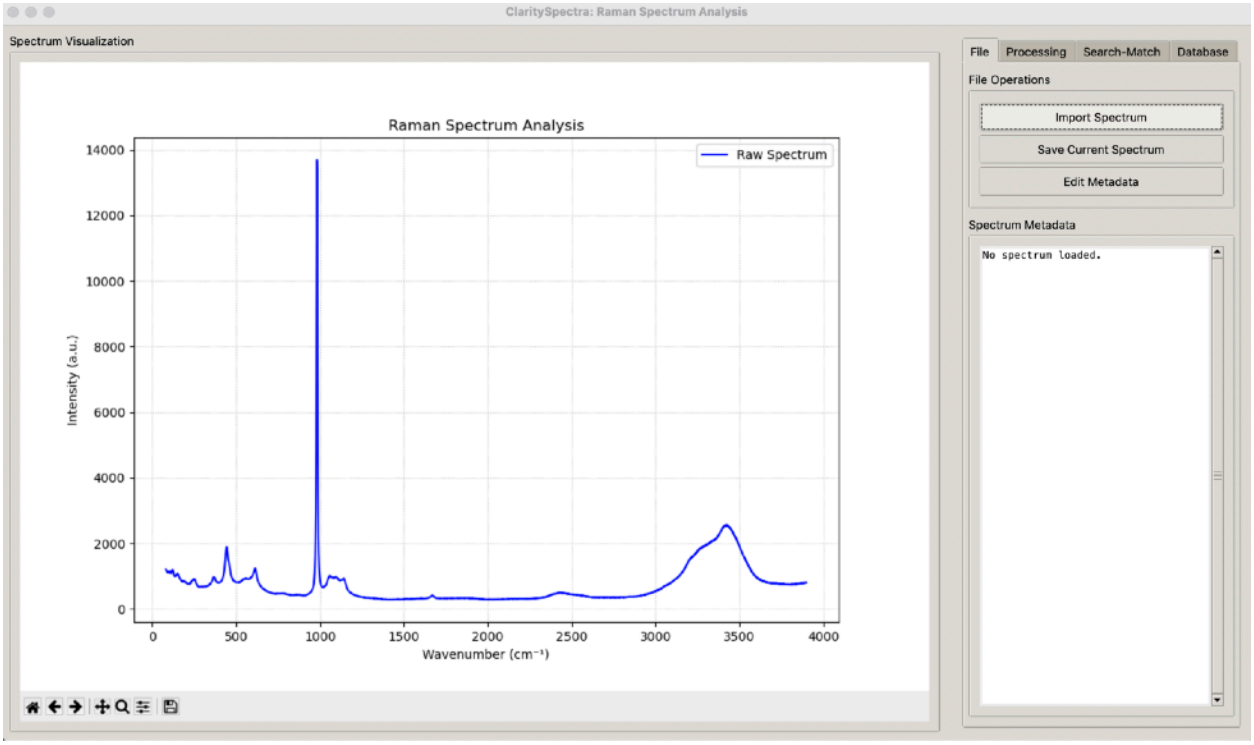
When you first open the application, you'll see a window divided into two main sections:

- **Left panel**: Spectrum visualization area
- **Right panel**: Control tabs (File, Processing, Search, Database)

3.2 Navigation

The right panel contains four tabs that organize the application's functionality:

- **File**: Import, save, and manage spectrum metadata
- **Processing**: Background subtraction and peak finding operations
- **Search-Match**: Search database for matching spectra
- **Database**: Add, remove, and browse spectra in the database



4. Importing Spectra Into The Database

4.1 Supported File Formats

The application supports various text-based formats containing Raman spectrum data:

- CSV (comma-separated values)
- TXT (tab or space-delimited)
- Custom formats with metadata headers (lines starting with # have meta information)

4.2 Importing a Single Spectrum

1. Click the **File** tab in the right panel
2. Click the Import Spectrum button
3. Browse to locate your spectrum file
4. Select the file and click **Open**

The spectrum will appear in the visualization panel, and any metadata from the file will be displayed in the Metadata section.

4.3 Batch Import

For importing multiple spectra at once:

1. Go to the **Database** tab

2. Click Batch Import Spectra
3. Select a folder containing multiple spectrum files
4. Monitor the progress in the import window
5. Click **Close** when the import is complete

5. Spectrum Processing

5.1 Background Subtraction

To remove background fluorescence from your spectrum:

- Select the **Processing** tab
- Adjust the baseline parameters:
  - **λ (lambda)**: Controls smoothness (higher values = smoother baseline)
  - **p**: Controls asymmetry (lower values = more fitting to peaks)
- Click Subtract Background
  - If you don't like how the background fit, then import the data again.
  - A fast way to import the data again is: **File > Import Spectrum**, that way you are not switching between tabs.

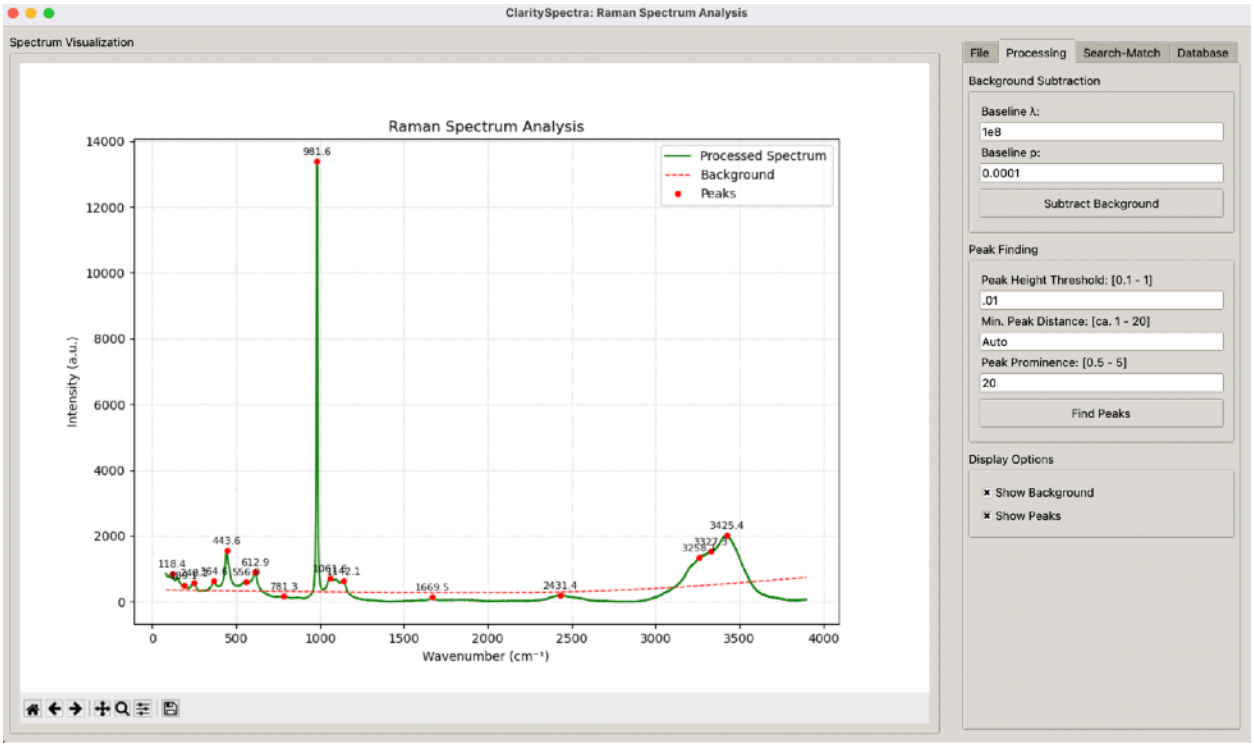
The processed spectrum will appear in the visualization panel.

5.2 Peak Finding

To detect peaks in your spectrum:

- In the **Processing** tab, adjust the peak detection parameters:
  - **Peak Height Threshold**: Minimum intensity for peak detection
  - **Min. Peak Distance**: Minimum separation between peaks
  - **Peak Prominence**: How much a peak stands out from surrounding baseline
- Click Find Peaks
- Keep changing the values until you are satisfied
- Numbers in the [ ] are suggested values

Detected peaks will be marked on the spectrum with red dots and labeled with their wavenumber positions.



5.3 Display Options

Customize the visualization with the display options:

- **Show Background**: Toggle display of the estimated background
- **Show Peaks**: Toggle display of detected peaks

6. Database Management

6.1 Adding Spectra to Database

After importing and processing a spectrum:

1. Go to the **Database** tab
2. Enter a name for the spectrum in the **Spectrum Name** field
3. Click Add Current Spectrum

The spectrum and its metadata will be saved to the database.

6.2 Viewing the Database

To browse the database contents:

1. In the Database tab, click View/Search Database
2. A new window will open showing all database entries
3. Use the search field to filter entries
4. Select an entry to view its details

6.3 Editing Metadata

To edit metadata for a spectrum:

1. Select a spectrum in the database viewer
2. Click Edit Metadata
3. Modify the fields in the metadata editor
4. Click **Save Metadata** when finished

6.4 Hey Classification

The application integrates Hey Classification for mineral identification:

1. Click **Update Hey Classification** to update all database entries
2. Or use the metadata editor to lookup Hey Classification for individual entries

7. Search and Matching

7.1 Basic Search

To find spectra similar to your current spectrum:

1. Go to the **Search** tab
2. Set the desired number of matches and similarity threshold
3. Select a matching algorithm:
    1. **Combined**: Uses correlation, peak matching, and MSE (recommended)
    2. **Correlation**: Based on spectral correlation coefficient only
    3. **Peak Matching**: Compares peak positions only (first using Find Peaks for this to work)
4. Click Search Match

7.2 Advanced Search

For more targeted searches:

1. Select the Advanced Search sub-tab
2. Enter specific peak positions to search for
3. Set the peak tolerance (in cm^{-1})
4. Select a Hey Classification filter if desired
5. Click Advanced Search

7.3 Interpreting Results

Search results appear in a new window with three tabs:

1.
2. **Spectral Comparison**: Visual comparison between query and match
3. **Correlation Analysis**: Heatmap showing regional correlation
4. **Report**: Detailed text report of the match

The left panel shows all matches sorted by similarity score.

Search-Match Performance and Usage Comparisons

Execution Speed:

- Fastest: Correlation search (simple matrix operation)
- Medium: Peak-based search (set operations)
- Slowest: ML-based search (requires PCA computation)
- Moderate: Combined search (multiple calculations but optimized)

Memory Usage:

- Lowest: Peak-based search (only uses peak positions)
- Medium: Correlation search (full spectra but simple operations)
- Highest: ML-based search (stores transformed data in PCA space)
- Medium-High: Combined search (requires multiple metrics)

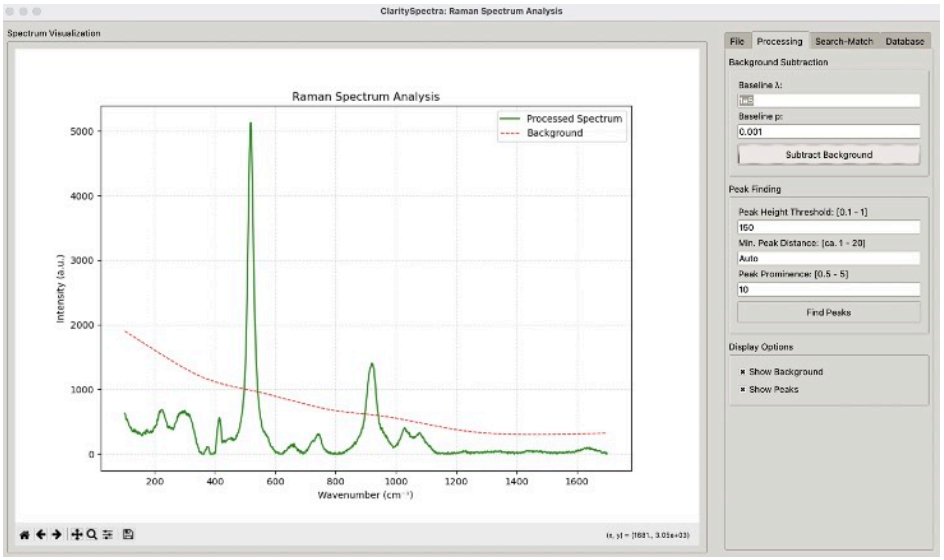
Effectiveness by Scenario:

Scenario	Best Method
Clean, high-quality spectra	Correlation or Combined
Noisy spectra with clear peaks	Peak-based
Spectra with baseline issues	Peak-based or ML-based
Complex mixtures	ML-based or Combined
General-purpose	Combined (hence "Recommended")

The application intelligently handles the choice of search algorithm based on user selection, and falls back to the Combined method if ML-based search is selected but scikit-learn is unavailable.

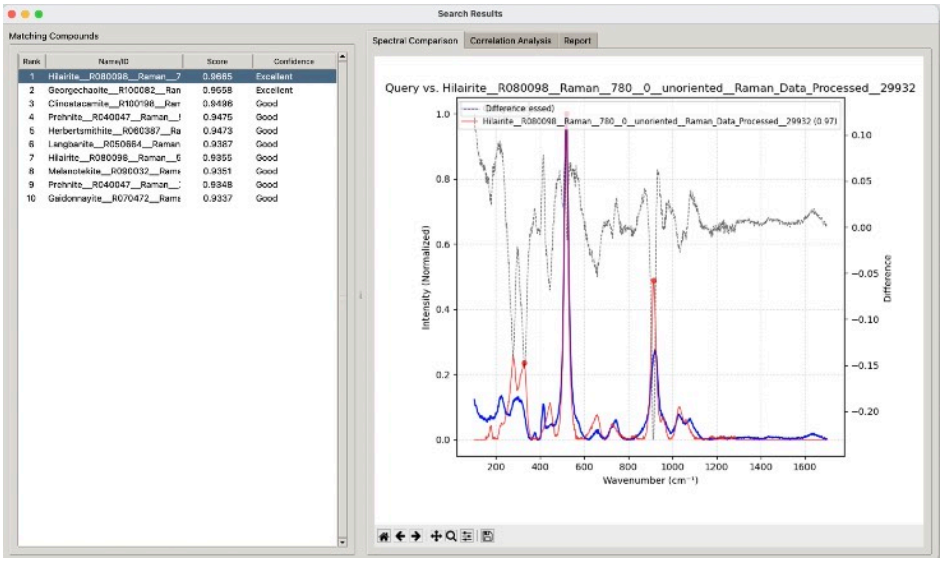
Example 1. A Mineral

For this example, I'll use a mineral that has broad peaks and a moderately high background.

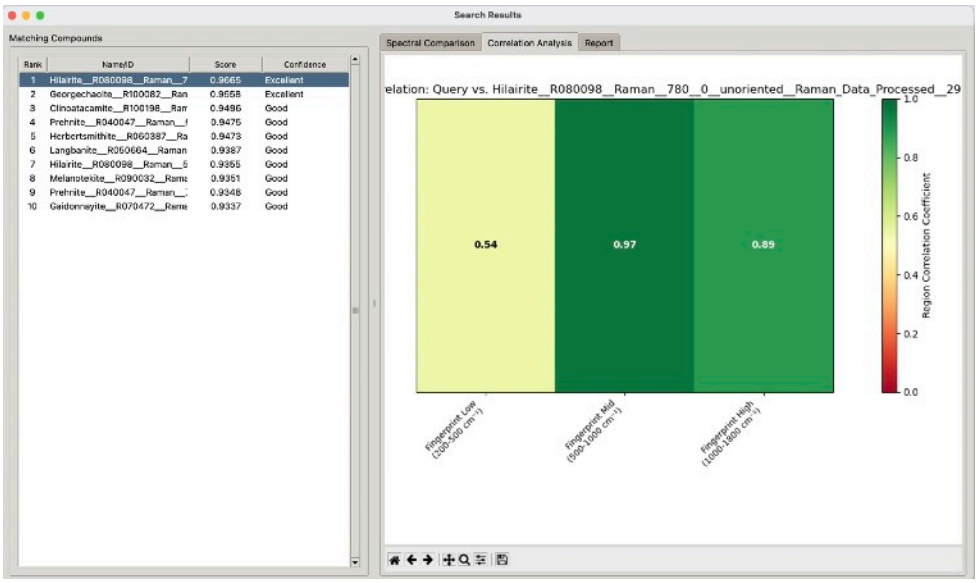


The figure above shows the background fitting and resulting spectrum. You can see the parameters I used in the figure.

Next, I went through all the different types of search methods, and found that the ML tool results in the best fitting. This was expected as the data is noisier than one would like. Similar minerals like georgechaoite and gaidonayite also appeared in the result list (these are all similar minerals).



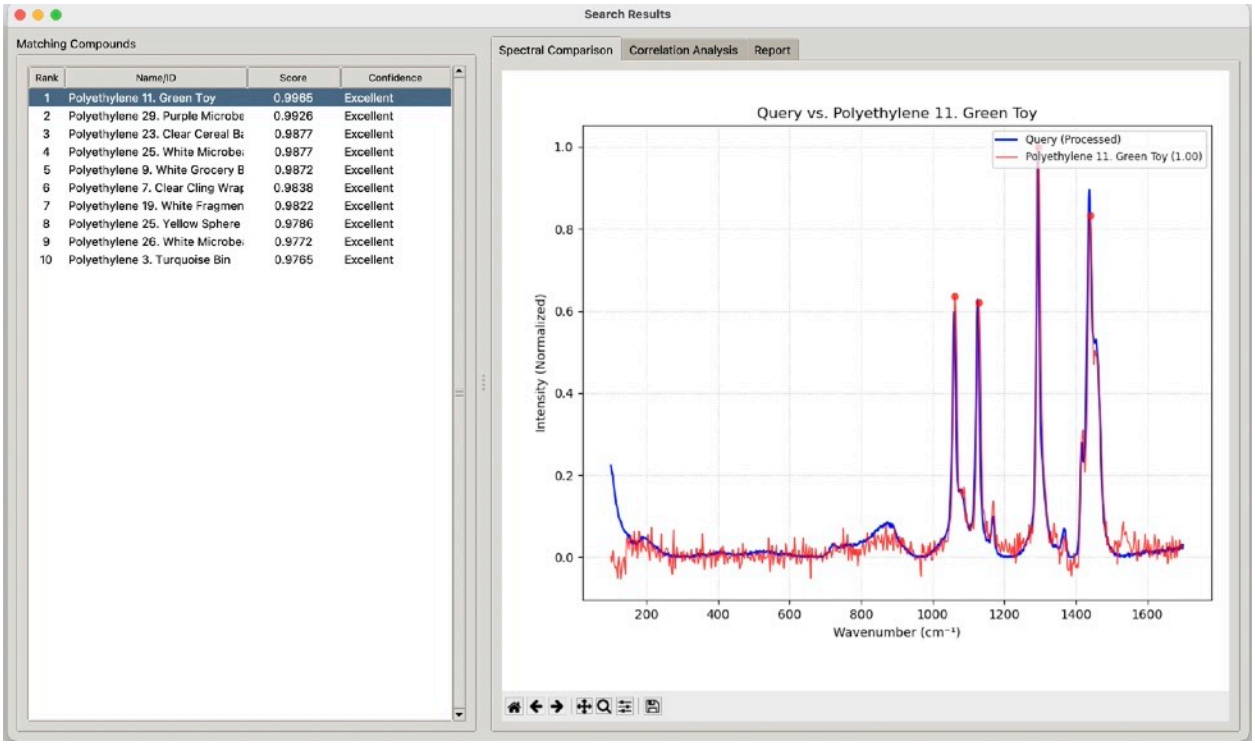
The heat map also shows a good fit in the med and mid finger print regions. The lattice modes didn't fit all that well, but this was a non dominant part of the spectrum.



It's best to explore and test different approaches and use the tools as a guide.

Example 2. Plastic

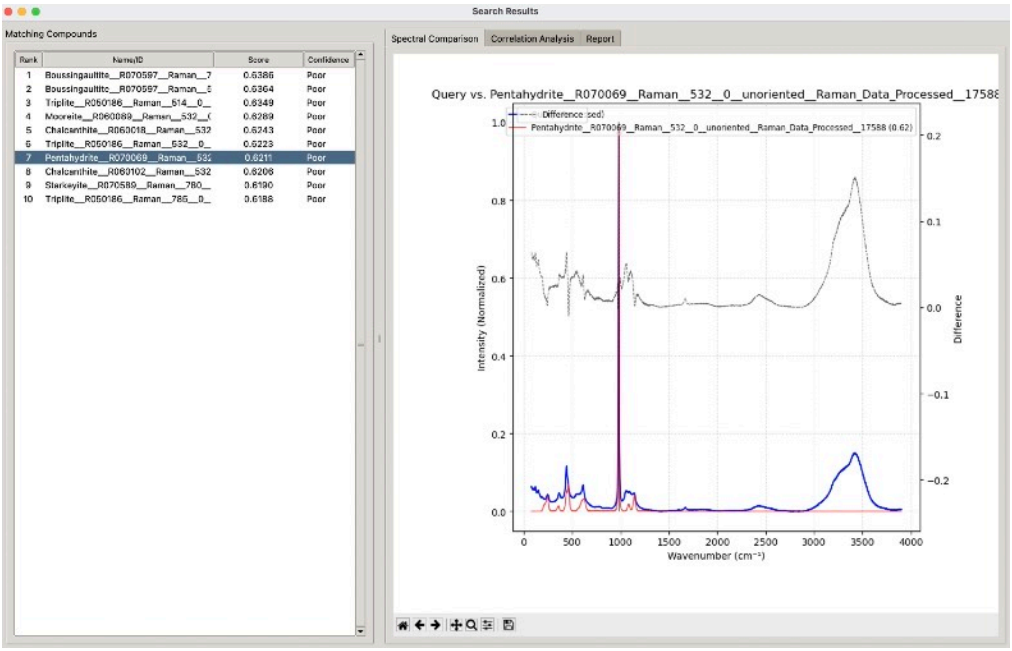
After background subtraction and peak searching, the ML search tool found a near perfect fit to polyethylene. This particular piece was found inside a turtle.

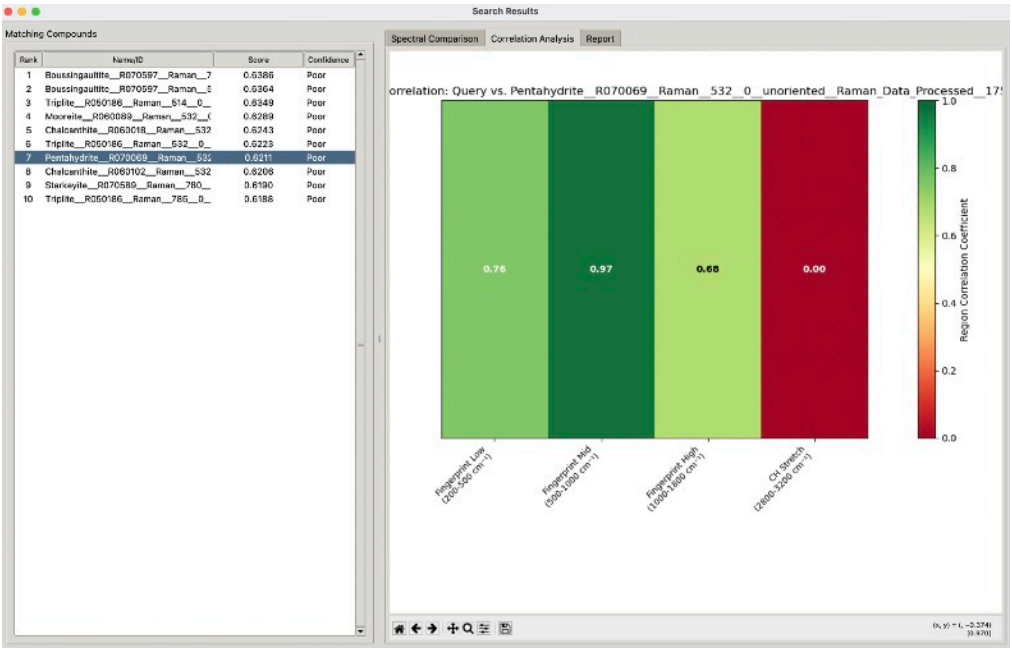


Example 3. A Liquid

For this example, I want to see the H2O / OH vibrational modes and the anion vibrational modes. I'm not expecting a good fit because there are no liquid Raman spectra in my database. But, maybe I can find something similar and see what my liquid is mostly made of.

Doing a Combined search, you can see that only sulfate minerals are shown, indicating, that the spectrum is a sulfate salt saturated solution. Heatmap plot shows really good fitting at the mid finger print region, but poor fitting at the CH/OH region (because the data base didn't go out that far). Clicking on the report will reveal all the details of the Match results.





8. Data Export

8.1 Exporting Spectra

To save the current spectrum:

1. Go to the **File** tab
2. Click Save Current Spectrum
3. Choose a location and format
4. Click **Save**

8.2 Exporting Reports

After performing a search, you can export the results:

1. Go to the **Report** tab in the results window
2. Click one of the export buttons:
    1. **Export as PDF**: Creates a formatted PDF report
    2. **Export as Text**: Saves as plain text
    3. **Export as CSV**: Creates a spreadsheet-compatible file

9. Troubleshooting

9.1 Common Issues

Application Won't Start And If It Does, I Don’t See The Controls

- Ensure your system meets the minimum requirements
- If a window pops-up, expand the window as some systems don’t read my default window sizes

Import Errors

- Verify the file format is supported
- Check that the file is not corrupted
- Ensure file contains both wavenumber and intensity columns

Processing Issues

- Try different baseline parameters for difficult spectra
- Use "Auto" settings for peak detection on first attempt
- For noisy spectra, consider preprocessing with smoothing so that ClaritySpec can find the peaks

9.2 Getting Help

For additional support:

- Check this document :)
- Contact me acelestian@nhm.org but please allow me a bit of time to get back to you.

10. Appendices

10.1 Keyboard Shortcuts

These may not work on all systems.

- **Ctrl+I**: Import spectrum
- **Ctrl+S**: Save spectrum
- **Ctrl+B**: Subtract background
- **Ctrl+P**: Find peaks
- **Ctrl+F**: Search/match
- **Ctrl+D**: Add to database

10.2 File Format Specifications

The application works best with files in the following format:

```
# NAME: Quartz
# RRUFFID: R040031
# IDEAL CHEMISTRY: SiO2
# HEY CLASSIFICATION: D. Silicates - Tectosilicates
# LOCALITY: Brazil
# DESCRIPTION: Colorless, transparent
128.0 352
152.3 1245
206.5 867
...
```

Where the first column is wavenumber (cm^{-1}) and the second column is intensity.

10.3 Algorithm Descriptions

Baseline Correction

The application uses Asymmetric Least Squares Smoothing (ALS) for baseline correction, which iteratively fits a smoothed curve to the spectrum's baseline points.

Peak Finding

Peak detection uses the SciPy `find_peaks` function with customizable parameters for height, distance, and prominence.

Spectrum Matching

The combined matching algorithm uses a weighted combination of:

- Correlation coefficient (60%)
- Mean squared error (20%)
- Peak position overlap (20%)

Technical Details of Core Algorithms

Background Subtraction Using Asymmetric Least Squares (ALS)

The background subtraction method implemented in this software uses the Asymmetric Least Squares algorithm, which is particularly effective for removing fluorescence backgrounds from Raman spectra.

Mathematical Foundation

The ALS algorithm works by minimizing the following penalized least squares function:

$$S = \sum_i w_i(y_i - z_i)^2 + \lambda \sum_i [(\nabla^2 z)_i]^2$$

Where:

- y_i is the observed spectrum intensity at point i
- z_i is the estimated baseline at point i
- w_i is an asymmetric weight (different for points above vs. below the baseline)
- λ is the smoothness parameter (controls how smooth the baseline is)
- ∇^2 is the second-order difference operator (approximates second derivative)

Algorithm Implementation

1. Initialize weights w_i = 1 for all points
2. For a set number of iterations (typically 10): a. Solve the penalized least squares problem for z using the current weights b. Update weights asymmetrically:
    - If y_i > z_i (point above baseline): w_i = p (small value, e.g., 0.01)
    - If y_i ≤ z_i (point at/below baseline): w_i = 1-p (large value, e.g., 0.99) c. Repeat with new weights

Parameter Effects

- **λ (Lambda)**: Controls baseline smoothness
    - Higher values (e.g., 1e7) create smoother baselines but may underfit
    - Lower values (e.g., 1e3) allow more flexibility but may follow peaks
    - Typical range: 1e5 to 1e7 for Raman spectra
- **p (Asymmetry parameter)**: Controls fitting asymmetry
    - Smaller values (e.g., 0.001) strongly favor fitting points below the curve
    - Larger values (e.g., 0.1) allow more influence from points above (peaks)
    - Typical range: 0.001 to 0.05 for Raman spectra

Implementation Efficiency

The implementation uses sparse matrices for computational efficiency:

- The second derivative operator is represented as a sparse difference matrix D
- The weighted least squares problem becomes a sparse linear system
- Solution via sparse solvers (spsolve) greatly reduces computation time and memory usage

Peak Finding Algorithm

The peak detection algorithm leverages SciPy's `find_peaks` function with specialized parameter handling for Raman spectra.

Algorithm Steps

1. Determine if processed (background-subtracted) or raw spectrum should be used
2. Calculate default parameter values if not specified by user:
    - Height threshold = 5% of maximum intensity
    - Distance = 1% of spectrum length (in data points)
    - Prominence = 2% of maximum intensity
3. Call SciPy's `find_peaks` function with these parameters
4. Extract and store peak information:
    - Indices in the original spectrum
    - Corresponding wavenumber values
    - Peak heights
    - Additional properties (prominence, width, etc.)

Parameter Explanations

- **Height**: Minimum intensity value to be considered a peak
    - Controls sensitivity to minor peaks
    - Adaptive default: 5% of spectrum maximum
- **Distance**: Minimum separation between peaks
    - Prevents detecting multiple points from the same peak
    - Adaptive default: 1% of spectrum length
- **Prominence**: How much a peak stands out relative to surrounding baseline
    - More robust than height for distinguishing real peaks
    - Adaptive default: 2% of spectrum maximum
- **Width**: Optional constraint on peak width
    - Useful for filtering out noise spikes vs. true Raman bands
    - Not set by default, allowing all peak widths

Advanced Considerations

The algorithm incorporates several technical enhancements:

- Automatic parameter calculation based on spectrum characteristics
- Wavenumber mapping to associate peak indices with actual Raman shifts

- Property storage for advanced filtering in search algorithms
- Scale-invariant defaults that work across diverse intensity scales

Search-Match Algorithm

The search-match functionality implements a sophisticated multi-metric approach that combines spectral correlation, peak matching, and intensity distribution similarity.

Core Matching Metrics

1. Spectral Correlation Coefficient

- Pearson correlation coefficient between query and database spectra
- Measures overall spectral shape similarity
- Spectrum normalization (0-1 range) for scale invariance
- Wavenumber interpolation for consistent comparison
- Mathematical formula: $r = cov(X,Y)/(\sigma_X \cdot \sigma_Y)$

2. Mean Squared Error (MSE)

- Quantifies point-by-point intensity differences
- Converted to similarity score using: $score = 1/(1+10 \cdot MSE)$
- More sensitive to local differences than correlation
- Complements correlation by focusing on absolute differences

3. Peak Position Matching

- Jaccard similarity between peak sets: $|A \cap B|/|A \cup B|$
- Peaks are rounded to integer wavenumbers for comparison
- Tolerance parameter allows for slight peak shifts
- More robust to baseline issues than full-spectrum metrics

Combined Scoring System

The final match score is a weighted combination:

- 60% Correlation coefficient
- 20% MSE-derived similarity
- 20% Peak position matching

This weighting scheme provides balanced sensitivity to overall spectral shape, intensity differences, and characteristic peak positions.

Search Optimization

Several optimizations enable efficient searching:

1. Query spectrum is processed only once before comparison
2. Database spectra are interpolated to query spectrum wavenumbers

3. Normalization is applied dynamically during comparison
4. Early filtering based on threshold reduces computation for large databases
5. Vectorized operations for performance on large datasets

Advanced Search Capabilities

The advanced search enhances this core algorithm with:

1. Peak position filtering (search by specific peaks of interest)
2. Metadata-based filtering (Hey Classification)
3. Customizable tolerance for peak matching
4. Peak subset matching (match specific regions of interest)

This comprehensive approach enables nuanced spectral matching even in challenging cases with baseline issues, intensity variations, or partial spectral overlap.