IAD Optical Analysis Suite

User and Technical Manual

Diego Olvera

June 25, 2025

Contents

1 Introduction

The **IAD Optical Analysis Suite** is a Windows-based application designed to automate the batch processing of reflectance and transmittance data using the Inverse Adding-Doubling (IAD) method.

The tool provides:

- A graphical user interface (GUI) for simplified operation
- Automated execution of `iad.exe`
- Extraction of reduced scattering (μ_s') and absorption (μ_a) coefficients
- Power-law fitting based on Mie scattering theory

This application leverages Scott Prahl's open-source implementation of `iad.exe`, wrapped in a user-friendly and extendable Python interface.

2 Installation

System Requirements

- Windows 10 or later
- No Python installation required (bundled)
- Compatible with Python 3.11 if run from source

To Run the App

1. Download or clone the full directory structure
2. Place your `.rxt` files into the `iad_inputs/` folder
3. Double-click `iad_shell.py` or run from terminal with:

```
python iad_shell.py
```

Folder Structure

```
iad_shell/
 core/
 gui/
 docs/
 iad_inputs/
 iad_outputs/
 config.yaml
 iad.exe
 iad_shell.py
```

3 Workflow Diagram

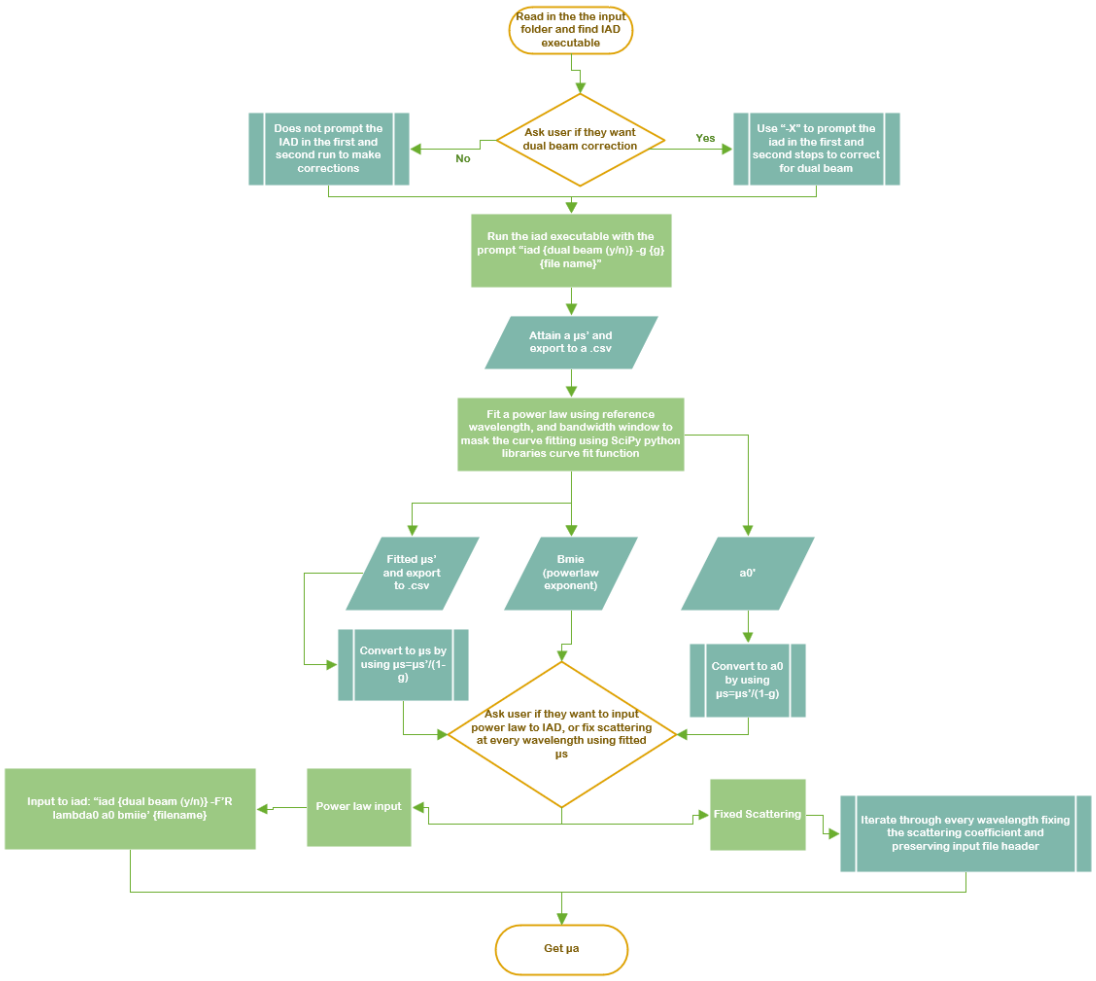

Figure 1: High-level workflow of the IAD Optical Analysis Suite

4 Using the Application

The IAD Optical Analysis Suite is structured around a 3-step process visible in the GUI. These steps correspond to distinct computational phases in the pipeline, each represented by a button and a status update region.

The application must be started with all required files present in the directory:

- `iad.exe` must be in the root folder
- `.rxt` files must be in `iad_inputs/`
- `config.yaml` (optional) sets analysis parameters

Each step is described below in detail.

Analysis Settings Overview

These parameters are configured by the user in the GUI and are used throughout the IAD analysis pipeline, especially in scattering estimation and final absorption coefficient extraction.

- **Anisotropy Factor (g)**
  The average cosine of the scattering angle in the medium.

  $$g = \langle \cos(\theta) \rangle \quad \text{where } 0 \leq g < 1$$

  Affects the conversion from reduced scattering (μ_s') to total scattering (μ_s):

  $$\mu_s = \frac{\mu_s'}{1 - g}$$

  Typical value: 0.8–0.95 for biological tissues.
  Implication: Higher g means more forward scattering. If $g \to 1$, the denominator approaches zero, causing instability.

- **Reference Wavelength (λ_0) [nm]**
  The central wavelength used as a baseline for Mie scattering fitting:

  $$\mu_s'(\lambda) = a_0 \left(\frac{\lambda}{\lambda_0} \right)^{-b_{\text{mie}}}$$

  The value of μ_s' at this wavelength becomes the fitted a_0.
  Default: 700 nm.
  User setting: Any wavelength in your data range.

- **Fit Window Min/Max [nm]**
  Defines the range of wavelengths over which the Mie power law is fitted to the data. Only the values of μ_s' within this window are used in curve fitting.
  Good practice: Choose a window where the signal is clean and monotonic and absorption is low (e.g., 600–750 nm).

- **Use Dual Beam Correction**
  When enabled, the app adds the `-X` flag to all `iad.exe` calls, enabling dual-beam correction. This assumes measurements have already been normalized using a reference beam. It may improve accuracy in spectrophotometric setups.

4.1 Step 1: Run Initial Analysis

This step performs the first-pass IAD run for every input file:

- The GUI scans the `iad_inputs/` directory for all `.rxt` files
- Each file is passed to `iad.exe` using a command-line call:

```
iad.exe [-X] -g <g_value> <input_file>
```

- The optional `-X` flag applies dual-beam correction
- The resulting `.txt` file is copied into `iad_outputs/` and the original deleted

From these IAD output files:

- The program extracts all reduced scattering values $\mu_s'(\lambda)$
- A single CSV file, `scattering_values_multi.csv`, is generated, with:
  - Rows = wavelengths
  - Columns = one per input sample

This step also:

- Updates the GUI with green indicators for executable/input presence
- Logs all activity in real time in the bottom scroll window
- Visually advances progress to Step 2

4.2 Step 2: Fit Mie Power Law

This step fits a Mie-based power law to each sample's scattering curve:

$$\mu_s'(\lambda) = a_0 \left(\frac{\lambda}{\lambda_0} \right)^{-b_{\text{mie}}}$$

The fitting behavior includes:

- λ_0 is the reference wavelength (default: 600 nm)
- a_0 is extracted from each sample's curve at λ_0
- b_{mie} is estimated using nonlinear least-squares regression over a wavelength window
- The fit window is defined by `fit_min` and `fit_max` (e.g., 600–750 nm)

For each sample:

- R^2 is calculated as the coefficient of determination
- The smoothed scattering curve is generated

Outputs include:

- `powerlaw_summary.csv` — table of `sample`, a_0, b_{mie}, and R^2
- `powerlaw_smoothed.csv` — wavelength-indexed table of fitted curves

The GUI updates:

- Progress bar (Step 2 of 3)
- Logs each fit and any failed samples
- Displays the summary table with auto-refresh

4.3 Step 3: Final Analysis

This final stage synthesizes all previous computations to estimate optical absorption (μ_a) and scattering (μ_s') coefficients for each sample across wavelengths. It executes one final call to `iad.exe` for each sample using either fitted models or fixed scattering values.

Analysis Modes

The user selects one of two modes from a dropdown menu in the GUI:

a) **Power Law Mode** (Default):
In this mode, the app uses the previously fitted Mie model parameters a_0 and b_{mie} to reconstruct $\mu_s'(\lambda)$ for each wavelength in the measurement range:

$$\mu_s'(\lambda) = a_0 \left(\frac{\lambda}{\lambda_0}\right)^{-b_{\text{mie}}} \qquad \mu_s(\lambda) = \frac{\mu_s'(\lambda)}{1-g}$$

This provides a continuous, smooth scattering curve. It is preferred when noise is present in raw data and spectral fitting has succeeded.

b) **Fixed Scattering Mode**:
Uses raw μ_s' values directly from `scattering_values_multi.csv`, line by line per wavelength. This method runs IAD individually at each wavelength using tabulated scattering input, bypassing the need for fitting. It is preferred for high-fidelity raw data or when fitting fails.

Backend Execution

For each sample and wavelength, `iad.exe` is invoked in one of the following formats:

- **Power Law Mode:**

```
iad.exe [-X] -F R|P <lambda0> <a0> <b_mie> <input_file>
```

Here:

  - `-F` = Fixed scattering mode (R = reflectance, P = transmittance)
  - `R|P` = Specifies which measurement is fixed
  - `<lambda0>` = Reference wavelength
  - `<a0>` and `<b_mie>` = Parameters from Step 2
  - `[-X]` = Optional dual beam correction

- **Fixed Mode:**

```
iad.exe [-X] -F <scattering_coefficient> <input_file>
```

Each run is performed per wavelength using a temporary `.rxt` file with raw reflectance, transmittance, and fixed μ_s' value.

Workflow of Fixed mode processing

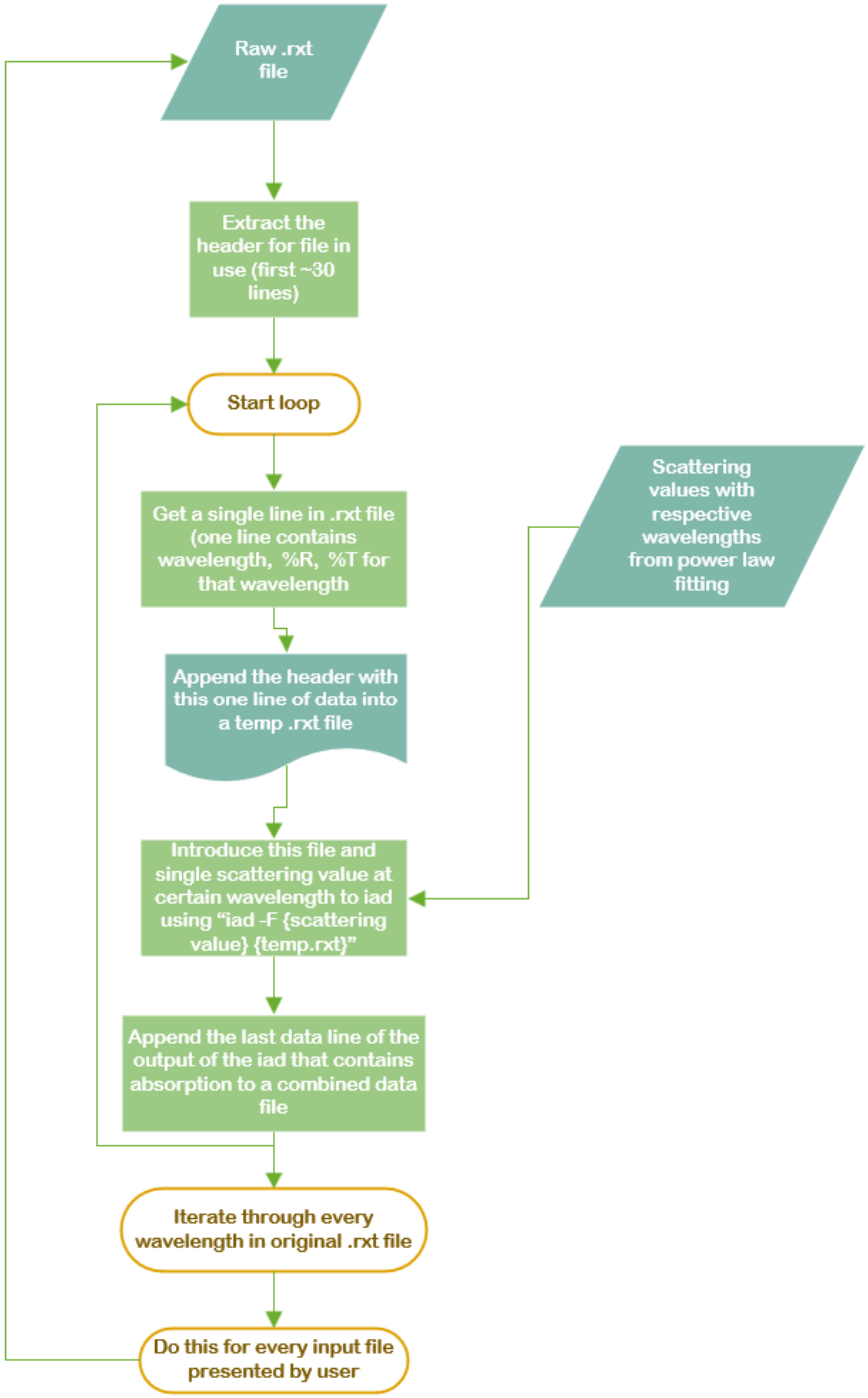

Figure 2: Flow diagram of data processing for fixing scattering at every wavelenghth using the IAD program

All temporary files (e.g., `temp_single.rxt`) are automatically managed by the app and deleted after each execution.

Outputs

After running IAD for all samples:

- `combined_output.txt`: Contains a unified table of results across all samples and wavelengths:
  - **Sample** – name parsed from file
  - **Wavelength** – in nm
  - **Reflectance**, **Transmittance** – measured input values
  - **Absorbance** – estimated as $-\log_{10}(R+T)$
  - μ_s', μ_a – final optical coefficients [1/mm]
- `iad_analysis_results.xlsx`: Multi-tab Excel spreadsheet with:
  - Raw scattering values
  - Power law fits
  - Smoothed μ_s'
  - Final IAD results

GUI Behavior

Upon successful completion of Step 3:

- The progress bar advances to 3/3
- The status label shows `"Analysis complete"`
- Plotting options become available in the GUI
- Users can click:
  - **Show Plots** — to view raw/fitted/absorbance data
  - **Export to Excel** — for spreadsheet analysis
  - **Generate Report** — to produce an HTML summary
  - **Open Output Folder** — to browse the generated files

The GUI also displays warnings if:

- Power law parameters are missing
- Input files are malformed or empty
- `iad.exe` is missing from the run directory

4.4 Other Features

- **Real-Time Logging**: all steps are logged with timestamps and color-coded severity
- **Dynamic Plotting**:
  - Raw $\mu_s'(\lambda)$
  - Smoothed μ_s' from Mie fit
  - Absorbance spectra
  - Overlays with fit windows
- **Reports**: one-click generation of:
  - Excel workbook (`.xlsx`)
  - Styled HTML report with fit table and output summary
- **Status Indicators**: checks for:
  - Presence of `iad.exe`
  - Available input files
  - Fit and output files from previous steps

5 Configuration

config.yaml Fields

Key	Description
`g_value`	Anisotropy factor (g, typically 0.8)
`reference_wavelength`	Reference wavelength for fitting (e.g., 600 nm)
`Power Law or Fixed`	Introduce power law parameters or loop iteratively through each wavelength
`use_dual_beam`	Enable `-X` dual beam correction flag

Table 1: Configuration Parameters

6 Input File Format

Overview

Each sample must be provided as a plain text file with a `.rxt` extension. These files contain reflectance and transmittance measurements across a range of wavelengths and are the primary input to the IAD pipeline.

File Placement

Place all `.rxt` files into the `iad_inputs/` directory prior to launching the application.

Structure of an .rxt File

Each file contains:

- **Header** (typically 30 lines): Includes sample metadata, integration times, instrument information, etc.
- **Data Section**: Begins with a labeled line and includes tab-separated wavelength data.

Example Structure

```
IAD1
#
# Sample header
# by Scott Prahl     {also ignored}
#
1.42       # Index of refraction of sample
1.0      # Index of refraction of top slide
1.19       # [mm] Thickness of sample
0.0       # [mm] Thickness of slides
0.22       # [mm] Diameter of illumination beam
0.99      # Reflectance of calibration standard

0         # [mm] Number of spheres used during each measurement

          # Refection sphere properties (ignored because 0 spheres is selected)
203.2     # [mm] Sphere Diameter
25.4      # [mm] Sample Port Diameter
12.7      # [mm] Empty Port Diameter
1.0       # [mm] Detector Port Diameter
0.96 # Reflectivity of the sphere wall

          # Transmission sphere properties (ignored because 0 spheres is selected)
203.2     # [mm] Sphere Diameter
25.4      # [mm] Sample Port Diameter
12.7         # [mm] Empty Port Diameter
1       # [mm] Detector Port Diameter
0.96 # Reflectivity of the sphere wall

2         # [mm] Number of measurements

#wave refl trans
750 0.313 0.763
748 0.313 0.763
746 0.312 0.763
744 0.312 0.762
742 0.311 0.762
740 0.31 0.761
738 0.31 0.76
```

```
736 0.311 0.76
734 0.312 0.761
...      ...      ...
```

Required Format Rules

- Header must be present, but its contents are ignored by IAD — the app reads it but only retains the last 3-column data line.
- Data section must start with: `#wave  refl  trans`
- Data must be tab- separated
- Each line must include exactly three numeric values:
  1. Wavelength in nm
  2. Reflectance (unitless, $R \in [0, 1]$)
  3. Transmittance (unitless, $T \in [0, 1]$)

Notes on Units

- Reflectance and transmittance values are fractions (not percent), e.g., 0.27 not 27%.
- Wavelengths should cover a range wide enough to support meaningful fitting (e.g., 400–750 nm).

Error Handling

If an input file:

- Has fewer than 30 lines, the app assumes the entire file is data and proceeds
- Has malformed data lines (e.g., missing columns), those lines are skipped
- Is missing from the folder, it will be ignored with a warning

Batch Behavior

The app will process all `.rxt` files found in `iad_inputs/` during:

- Step 1 (initial IAD run)
- Step 3 (final IAD run — fixed or power law)

Each file is treated as a separate sample throughout the workflow.

7 Data Visualization and Plotting

The IAD Optical Analysis Suite includes built-in interactive plotting tools to help users visualize and validate the output of each analysis stage. All plots are generated inside the GUI using DearPyGUI, and can optionally be opened in standalone Matplotlib windows for export or publication.

Accessing the Plotting Interface

After any analysis step (initial, power law, or final), click the **"Show Plots"** button in the GUI's *Results & Visualization* section. This expands a plot control panel and canvas.

You may choose from several plot types using radio buttons. A refresh button and "Open in Matplotlib" option are available for exporting high-resolution figures.

Plot Types and Descriptions

- **Raw Scattering**
  Displays the unprocessed $\mu_s'(\lambda)$ curves extracted from IAD output files.
  - Data Source: `scattering_values_multi.csv`
  - One curve per sample
  - Useful for assessing spectral shape and noise
- **Power Law Smoothed**
  Shows the fitted Mie scattering models for each sample:

$$\mu_s'(\lambda) = a_0 \left(\frac{\lambda}{\lambda_0} \right)^{-b_{\text{mie}}}$$

  - Data Source: `powerlaw_smoothed.csv`
  - Reflects smoothing from nonlinear regression
  - Fit range and shape are determined by user-set parameters
- **Absorbance**
  Plots final absorbance curves estimated from the IAD final output.
  - Data Source: `combined_output.txt`
  - Includes one curve per sample
  - Highlights absorption bands and overall trends
- **Raw + Fitted Scattering (with Fit Window)**
  Overlay of raw μ_s' data and fitted curves, with shaded region showing the fit window:
  - Data Sources: `scattering_values_multi.csv`, `powerlaw_smoothed.csv`
  - Fit window is defined by `fit_min` and `fit_max`
  - Useful for comparing model quality and fit accuracy

Additional Features

- **Legend and axis labels** are automatically generated
- **Data Tooltips** are available in Matplotlib version
- **Absorbance parse errors** are logged in the GUI if formatting is invalid
- GUI logs show file sources and sample names as each plot is drawn

Common Use Cases

- Verifying clean spectral response before fitting
- Inspecting the accuracy of the power law smoothing
- Identifying outliers or low-quality measurements
- Comparing absorbance across pigment types or sample conditions

Screenshot Example

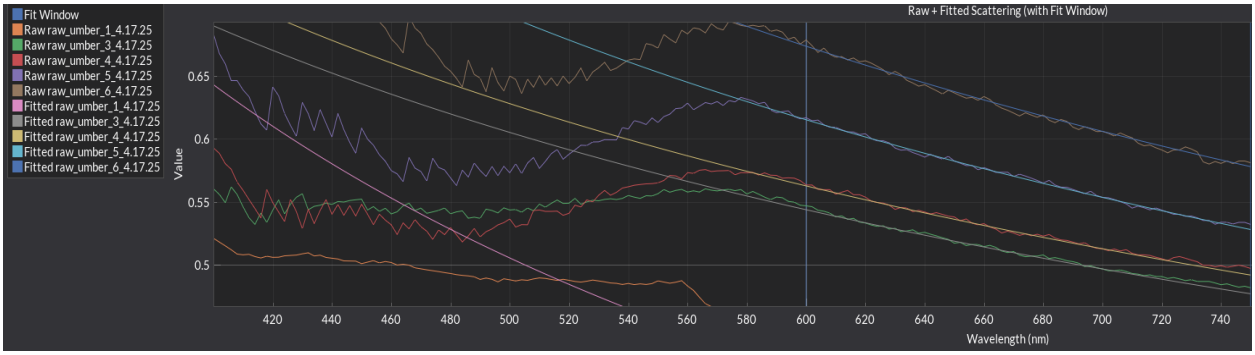

Figure 3: Plot selection and overlay options inside the GUI. Each sample's curve is color-coded and automatically labeled.

8 Output Files

- `scattering_values_multi.csv` – Extracted μ_s' across wavelengths
- `powerlaw_summary.csv` – Fitted a_0, b_{mie}, and R^2 values
- `powerlaw_smoothed.csv` – Smoothed curves per sample
- `combined_output.txt` – Final IAD results: μ_a, μ_s', absorbance
- `analysis_report.html` – Summary report with parameters and tables
- `iad_analysis_results.xlsx` – Exported spreadsheet of all results

9 Equation Reference

Power Law (Mie Scattering)

$$\mu_s'(\lambda) = a_0 \left(\frac{\lambda}{\lambda_0} \right)^{-b_{\text{mie}}}$$

Conversion to Total Scattering

$$\mu_s = \frac{\mu_s'}{1 - g}$$

10 Known Limitations

- Only ‘.rxt‘ inputs are supported
- One ‘.txt‘ output per input file is assumed
- Windows-only at the moment (due to `iad.exe`)
- No built-in validation of ‘.rxt‘ format

11 Appendix

Sample Screenshot

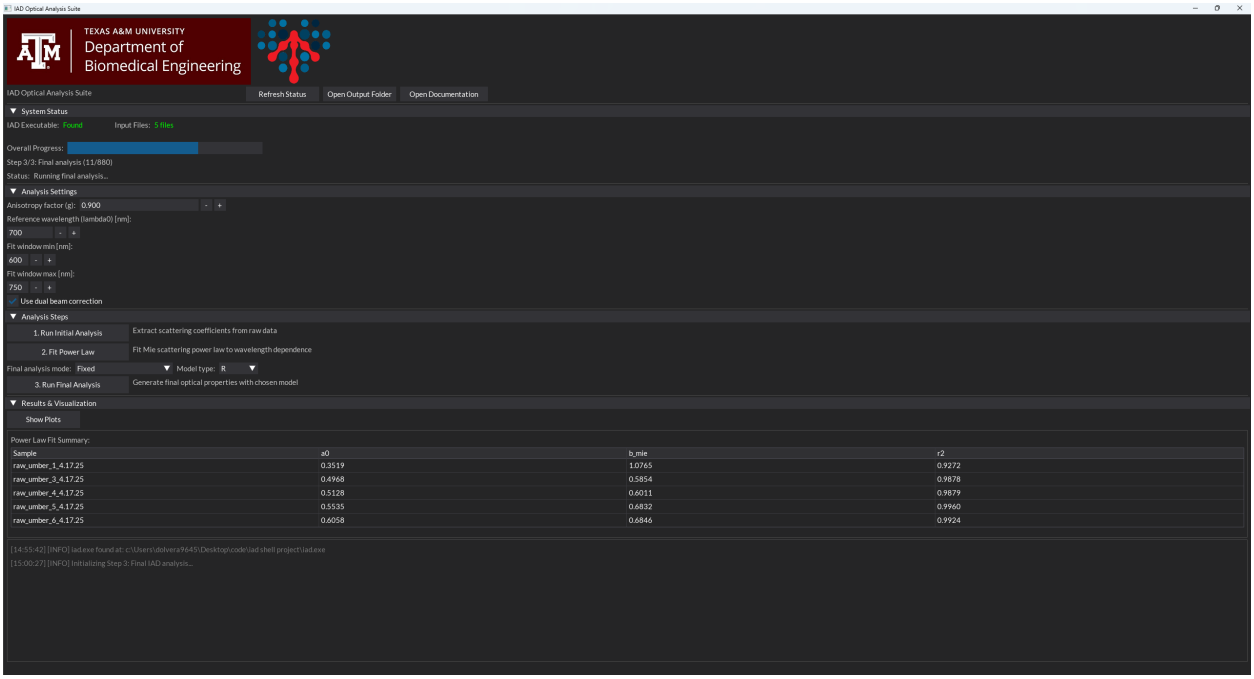

Figure 4: GUI running on Windows with analysis in progress

Power Law Fit Table

Sample	a_0	b_{mie}	R^2
raw_umber_1	1.231	0.926	0.982
raw_umber_3	1.151	0.879	0.975
...	...	...	...

Table 2: Example Power Law Fit Results

12 Credits

Developed by Diego Olvera
Inspired by Scott Prahl’s open-source IAD model