

# Tuna ユーザーマニュアル

川井拓真

2018 年 3 月 18 日

## 目次

1	IAD 解析	2
1.1	基本的な手順 . . . . .	2
1.2	UI 各部の説明 . . . . .	3
2	ソースコードから起動する方法	5
2.1	Windows . . . . .	5
2.2	Mac/Linux . . . . .	5

# 1 IAD 解析

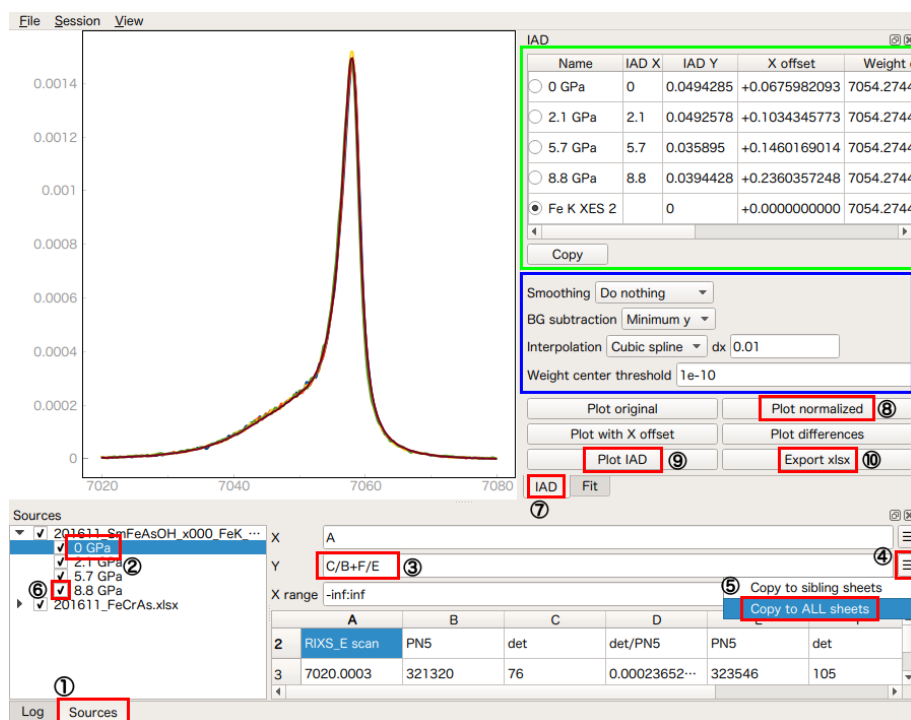


図 1 IAD 解析画面

## 1.1 基本的な手順

1. 測定データの入ったファイルを Tuna のウィンドウにドラッグ&ドロップして開く。複数のファイルを同時にドラッグしたり、連続して開いたりできる。テキストファイル, CSV, MS Excel, LibreOffice 等のフォーマットに対応している。
2. 左下の **Sources** タブ (①) をクリックする。
3. 左下のツリービューで、適当なシートを選択する (②)。図 1 は 2 つのエクセルファイルを開いた場合の画面である。
4. 読み込むべきスペクトルの  $x$ ,  $y$  座標にあたる列を右下の **X**, **Y** の欄 (③) に記入する。 **X**, **Y** はいずれも任意の数式を記入できる。また 1 シートに複数のスペクトルが記録されている場合、カンマ (,) で区切って複数の数式を記入できる。
5. 他のシートも同様に **X**, **Y** を記入する。すべてのシートに同じ式を使う場合、右端のメニューボタン (④) を押してメニューから **Copy to ALL sheets** (⑤) を選択す

ると全シートにコピーされる。

6. 全シートに式を記入したら、適当なシートのチェックを外し、チェックし直す (⑥).  
これで式の再読み込みが行われる。また、使わないシートはチェックを外しておく。
7. 右上ペイン下部の **IAD** タブ (⑦) をクリックする。
8. **Plot normalized** ボタン (⑧) をクリックしてスペクトルが正しく読み込まれていることを確認する。
9. **Plot IAD** ボタン (⑨) をクリックして結果を確認する。
10. 必要なら、**Export xlsx** ボタン (⑩) で検算用のエクセルファイルを書き出す。

## 1.2 UI 各部の説明

- **右上テーブル** 解析結果を表示する。選択範囲をエクセル等にコピーペーストできる。
  - **Name** 各スペクトルの名前。読み込んだファイル名またはシート名をもとに作成される。ラジオボタンで基準スペクトルを選択する。基準スペクトルの IAD 値は必ず 0 となる。
  - **IAD X** IAD 値をプロットする際の横軸の値。例えば圧力変化を測ったなら、圧力値を入力する。
  - **IAD Y** IAD 値の計算結果。
  - **X offset** 重心を合わせるために横シフトした量。
  - **Weight center** 重心の値。
  - **Peak x** ピーク位置  $x$  座標。
  - **Peak y** ピーク位置  $y$  座標。
- **Copy** テーブルの内容をすべてコピーする。エクセル等にペーストできる。
- **Smoothing** スムージングの方法を指定する。現在, Savitzky-Golay フィルタ (scipy.signal.savgol\_filter) のみ実装している。
- **BG subtraction** バックグラウンド除去の方法を指定する。
  - **Minimum y** スペクトルの最小値をバックグラウンドとして定数引く。
  - **Left edge** スペクトルの左端から **deltaX** の範囲で平均値をとり、これをバックグラウンドとして定数引く。
  - **Right edge** スペクトルの右端から **deltaX** の範囲で平均値をとり、これを

バックグラウンドとして定数引く.

- **interpolation** 内挿の方法を指定する.
  - **dX** 内挿の間隔を指定する.
    - **Cubic spline** `scipy.interpolate.CubicSpline(x, y)`
    - **B-spline** `tck=scipy.interpolate.splrep(x, y, w), scipy.interpolate.splev(x, tck)` **w** には **x**, **y**, **y<sub>max</sub>** を変数とした任意の式を記入できる. 測定値の誤差が  $d$  で与えられる場合,  $w = 1/d$  とする. ピークの高いところは測定値に合わせるが, 低いところはスムーズにしたい場合, 例えば `w=1/(ymax*0.003*(1.00001-y/ymax))` 等とすると良い.
    - **Linear** `scipy.interpolate.interp1d(x, y, "linear")`
    - **Pchip** `scipy.interpolate.PchipInterpolator(x, y)`
    - **Akima** `scipy.interpolate.Akima1DInterpolator(x, y)`
    - **Krogh** `scipy.interpolate.KroghInterpolator(x, y)`
    - **Barycentric** `scipy.interpolate.BarycentricInterpolator(x, y)`
  - **Weight center threshold** 各スペクトルと基準スペクトルの重心の差がこの値より小さくなるよう, **X offset** を調整する. 例えば 1 等じゅうぶん大きな値に設定しておくと, スペクトルの横シフトを行わずに IAD 値を計算する.
- 
- **Plot original** 各シートの **Y** 欄に記入した式の評価結果 (生の測定値) をそのままプロットする.
  - **Plot normalized** 各スペクトルに対し, スムージング, バックグラウンド除去, 内挿を行ったあと,  $\sum y_i = 1$  となるようノーマライズしてプロットする.
  - **Plot with X offset** 各スペクトルに対し **Plot normalized** と同じ手続を行ったあと, 重心のばらつきが **Weight center threshold** より小さくなるよう横方向にずらしてプロットする.
  - **Plot differences** 各スペクトルに対し **Plot with X offset** と同じ手続を行ったあと, 基準スペクトルとの差をプロットする.
  - **Plot IAD** 各スペクトルに対し **Plot with X offset** と同じ手続を行ったあと, IAD 値を計算し, **IAD X** とあわせてプロットする.
  - **Export xlsx** 検算用のエクセルファイルを書き出す.

## 2 ソースコードから起動する方法

### 2.1 Windows

1. <https://github.com/takumak/tuna/archive/master.zip> をダウンロードして展開する。もしくは `git clone https://github.com/takumak/tuna.git` してもいい。
2. <https://www.python.org/downloads/windows/> から Python をダウンロードしてインストールする。Python 3.6.5rc1 や Python 3.7.0b2 など、バージョン番号の後ろに rc や a, b がついているものはテスト版なので、Python 3.6.4 など、バージョン番号のみのものを選ぶ。exe ファイルを作りたい場合、pyinstaller の相性問題により Python 3.5 系の最新版を推奨する。x86 と x86-64 などは何でもいいが、Windows x86-64 executable installer を推奨する。
3. <https://www.lfd.uci.edu/~gohlke/pythonlibs/> から numpy と scipy をダウンロードして `tuna\build\win` に入れる。Python 3.5 系をインストールした場合は cp35, 3.6 系なら cp36 がファイル名に含まれているものを選ぶ。x86 を選択した場合は win32, x86-64 を選択した場合は win\_amd64 がファイル名に含まれているものを選ぶ。どのファイルをダウンロードすべきかわからない場合、後で `python build.py` を実行したらダウンロードすべきファイルを指示されるので、それをダウンロードする。
4. コマンドプロンプトを開いて以下のコマンドを実行する。

```
cd tuna-master\build\win
python build.py
cd ..\..\
.\build\win\venv\Scripts\activate.bat
python src\tuna.py
```

### 2.2 Mac/Linux

ターミナルで以下のコマンドを実行する。事前に `pyenv` を導入していれば、`tools/make_virtualenv.bash` が自動的にテスト済みのバージョンを導入する

`((pyenv install 3.5.5;pyenv local 3.5.5)).`

```
$ git clone https://github.com/takumak/tuna.git
$ cd tuna
$ ./tools/make_virtualenv.bash
$ ./tools/run_in_virtualenv.bash
```