# JASONにおけるHDF5ファイルとExternal Command機能によるPythonの活用

## 関連製品：核磁気共鳴装置(NMR)

現在、JASON[1]ではHDF5ファイルの階層構造とBeautifulJASONの活用により、Python[2]を用いたNMRスペクトル解析が効率的に行えます。
本資料では、以下の2点について紹介します：

1. HDF5ファイルの概要とJASONでの応用
2. External Command機能の活用例

### 1. HDF5ファイルの概要とJASONでの応用例

**HDF5ファイルの概要**

HDF5（Hierarchical Data Format version 5）は、複雑な構造を持つ大量のデータを効率的に保存・管理するためのファイル形式です。科学技術分野で広く使われています。主な特徴は以下の通りです：

- **階層構造** ファイル内にデータセットやグループを階層的に格納でき、ディレクトリ形式に近い構造を持ちます。これにより、大量データの整理が容易になります。
- **大容量データ対応** 大量のデータを効率的に保存・アクセスできるため、科学計算や画像解析などの分野で重宝されています。
- **クロスプラットフォーム互換性** Python、C++、MATLABなど、複数の言語で扱えるため、OS間の高い互換性を備えています。
- **ランダムアクセス** 必要なデータ部分のみを効率よく読み書きでき、大規模データ処理に適しています。

**NMRデータにおけるHDF5ファイルの利点**

- **NMRスペクトル解析** 多次元データを高速かつ部分的に抽出できるため、スムーズな解析が可能です。
- **階層型構造の活用** 例として、JasonDocument/NMR/NMRData/0/DataPoints や SpecInfo などのパスから、必要なデータを取得できます。
- **メタデータの利用** SpecInfo[3] 内に格納されている属性（例: SW、Spectrometer Frequencies、Spectrum Refなど）を活用することで、解析に必要な条件を簡単に取得できます。

**JASON+HDF5ファイルの応用例**

ここでは、HDF5ファイルの特徴を活かしたPythonによる簡易データ解析とレポート出力の例を紹介します。ワークフローは以下の通りです：

1. NMRデータの読み込み
2. データの処理・解析
3. グラフやレポートの自動生成

各ステップの内容と使用するPythonライブラリはTable 1にまとめています。この例は、HDF5形式のファイル構造を理解し、スペクトル解析を試す際の参考になります。

Table 1. NMRスペクトル解析ワークフロー例

| ステップ | 処理内容 | 使用ライブラリ | 代表コード例（抜粋） |
|---|---|---|---|
| ① データ読み込み | jjh5ファイルからNMRデータを取得 | h5py, numpy | real_data = f[path]['0'][()]<br>ppm = (frequencies / spectrometer_freq)[::-1] |
| ② データ処理 | ノイズ除去・スムージング | scipy.signal, scipy.ndimage | denoised_data = medfilt(real_data, kernel_size=5)<br>smoothed_data = gaussian_filter(real_data, sigma=2) |
| ③ 範囲解析 | ppm範囲で積分値を計算 | numpy | mask = (ppm >= r[0]) & (ppm <= r[1])<br>sum(data_to_analyze[mask]) |
| ④ グラフ作成 | データの可視化 | matplotlib.pyplot | plt.plot(ppm, real_data)<br>plt.savefig("original_plot.png") |
| ⑤ ノイズ評価 | 標準偏差でノイズ量を比較 | numpy | np.std(original_data)<br>noise_reduction = (...) |
| ⑥ 統計解析 | 平均・最大・最小値を算出 | numpy | np.mean(data)<br>np.max(data) |
| ⑦ PDFレポート作成 | 結果・グラフをレポートにまとめる | reportlab.platypus | doc = SimpleDocTemplate(...)<br>flowables.append(Image(...)) |

**NMRデータの読み込みとppm軸の計算**

Figure 1に示す関数では、.jjh5 [4]形式のNMR測定データから信号の実部データと化学シフト（ppm軸）を計算して取得します。$^{13}$C NMRなど他のスペクトルにも応用可能で、ファイル構造や属性の違いに注意すれば汎用的な読み込み関数として展開できます。



日本電子株式会社
本社・昭島製作所
〒196-8558 東京都昭島市武蔵野3−1−2 TEL：(042) 543−1111(大代表) FAX：(042) 546−3353
www.jeol.co.jp ISO 9001・ISO 14001 認証取得

### パスの指定方法例

HDF5ファイルは、フォルダーとファイルが階層的に構成されたデータベースのような形式です。以下にパス指定の例を示します：

path = "JasonDocument/NMR/NMRData/0/DataPoints"

real_data = f[path]['0'][()] # 実部データ

**補足** JasonDocument/NMR/NMRData/0/DataPoints はファイル内の階層構造を表す文字列です。f[path][ '0' ][()] は、指定したパス内の '0' というキーに対応するデータを取得する操作です。SpecInfo などの属性情報も同様にパス指定でアクセス可能です。

### 生成されたレポート例

Figure 2に、Table 1ワークフローから生成されたレポート例を示します。Numpy、Scipy、ReportLabについては、それぞれのユーザーズガイドなどをご参照ください。

```python
import h5py
import numpy as np

def extract_nmr_data(jjh5_file):
    with h5py.File(jjh5_file, 'r') as f:
        # 実部スペクトルデータの取得
        # → NMR信号の"実部"成分を取り出します
        path = "JasonDocument/NMR/NMRData/0/DataPoints"
        real_data = f[path]['0'][()]  # 実部 (Real)

        # 測定条件(メタ情報)を取得してppm軸を計算します
        # → スペクトルの横軸(ppm)を作るために必要な情報です
        specinfo_path = "JasonDocument/NMR/NMRData/0/SpecInfo"
        attrs = f[specinfo_path].attrs

        # Sweep Width(SW): 測定された周波数範囲(Hz)
        sweep_width = attrs["SW"][0]

        # 観測周波数(MHz): 測定装置の中心周波数
        spectrometer_freq = attrs["SpectrometerFrequencies"][0]

        # スペクトルの中心位置(Hz)
        spectrum_ref = attrs["SpectrumRef"][0]

        # ppm軸の計算
        num_points = len(real_data)
        start_freq = spectrum_ref - (sweep_width / 2)
        stop_freq = spectrum_ref + (sweep_width / 2)
        frequencies = np.linspace(start_freq, stop_freq, num_points)

        # ppm軸に変換(MHzで割る)し左から右に向かって増えるように反転
        ppm = (frequencies / spectrometer_freq)[::-1]

        return ppm, real_data
```

Figure 1. NMRデータの抽出 スクリプトの抜粋


Figure 2. レポート出力例

## 2. External Command 機能の活用例

### External Command 機能の概要

JASONのExternal Command機能は、JASON GUIのProcessingタブから外部のPythonスクリプトなどを呼び出して、NMRデータを直接処理できる機能です。自動化やカスタム処理に便利です。ここでは、JASON公式サイトからダウンロード可能な「External NMR Processing Scripts」を用いたPythonの使用例を紹介します。

### サンプルコードについて

以下のJASON公式サイトからサンプルコードをダウンロードできます：

https://www.jeoljason.com/resources-external-nmr-processing-scripts/

Figure 3に、External NMR Processing Scriptsディレクトリ内のPythonスクリプト一覧を示します。


Figure 3. External NMR Processing Scripts



JEOL 日本電子株式会社
本社・昭島製作所
〒196-8558 東京都昭島市武蔵野3−1−2 TEL：(042) 543−1111(大代表) FAX：(042) 546−3353
www.jeol.co.jp ISO 9001・ISO 14001 認証取得

### jasonParEdit.py コードについて

jasonParEdit.pyは、JASONデータ内のパラメータ（例：SW、SpectrumRefなど）を編集するためのPythonスクリプトです。GUIから呼び出して値を変更できます。 Figure 4は、jasonParEdit.pyコードの一部です。このスクリプトでは、「SpectrometerFrequencies」「SW」「SpectrumRef」パラメータを任意に変更できます。スクリプト上部には使用方法の説明も記載されていますので、併せてご確認ください。

```
from argparse import ArgumentParser

# コマンドライン引数の処理を行うための準備
# → スクリプト実行時にファイル名や編集したいパラメータなどを指定できるようにします
parser = ArgumentParser(description="JASONデータファイル内のパラメータを編集するツールです。")

# 編集対象のファイル名（HDF5形式のJASONファイル）
parser.add_argument( "-f", "--filename", action="store", default="$TMPFILE", help="編集対象のJASON HDF5ファイルのパス")

# 編集したいパラメータ名（例：SpectrometerFrequenciesなど）
parser.add_argument("-p", "--par", action="store", default="SpectrometerFrequencies", help="編集対象のパラメータ名")

# 編集する次元（0〜7のインデックス）
# → 多次元データの場合どの次元の値を変更するかを指定します
parser.add_argument( "-d", "--dim", action="store", type=int, default=0, help="編集対象の次元インデックス(0〜7)")

# 新しい値（float型）
# → 指定したパラメータに設定する新しい数値
parser.add_argument( "-v", "--vNum", action="store", type=float, default=500.0, help="パラメータに設定する新しい値")

# 引数を解析して変数に格納
args = parser.parse_args()
```

Figure 4. jasonParEdit.py スクリプトの抜粋

### 操作方法

操作手順はTable 4にまとめています。JASON GUI上での設定および入力例はFigure 5をご参照ください。

Table 4. 操作方法

| ステップ | 処理内容 |
|---|---|
| ① | JASON GUIのProcessingタブに「External Command」を追加します |
| ② | 「External Command」タブのCmd: に実行するPythonのフルパスを指定します |
| ③ | 「External Command」タブのArguments: に実行するPythonスクリプトのフルパス（ここでは、jasonParEditを指定）を指定します |
| ④ | ここでは、jasonParEditで実行したい処理内容を指定します。例えば、SW幅を指定したい場合（5000 Hzへ変更する処理）：<br>C:¥Users¥externalNMRProcessing¥python¥jasonParEdit.py -f $TMPFILE -p SW -d 0 -v 5000.0 |
| ⑤ | Processingタブの設定内容を確認後「Appy」をクリックします |

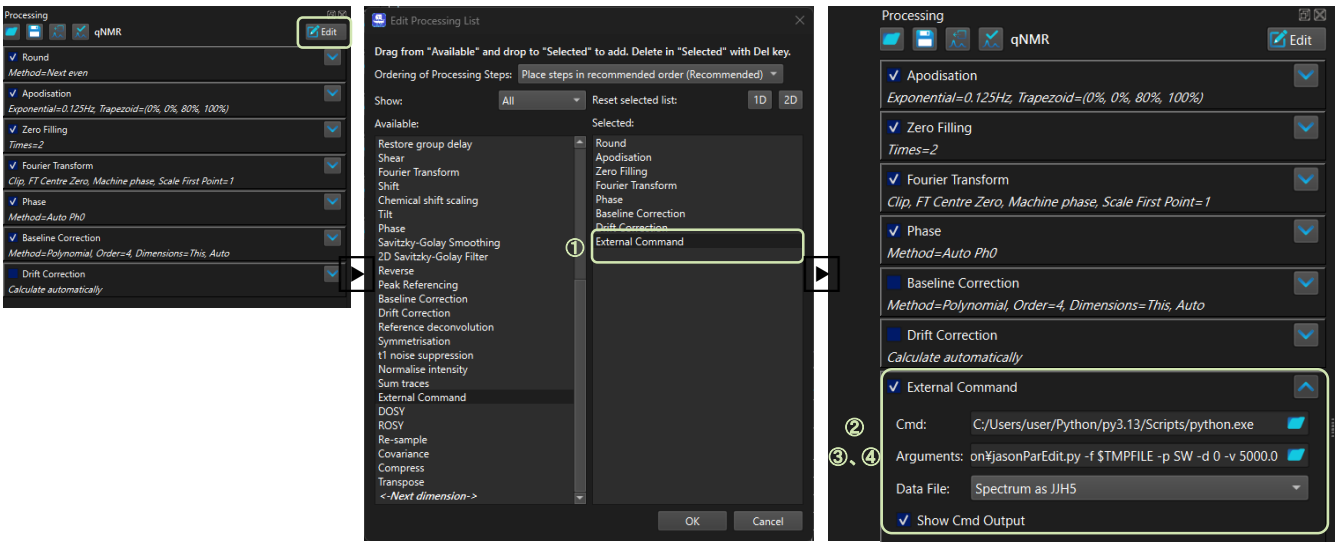


Figure 5. jasonParEdit.py 設定例



JEOL 日本電子株式会社
本社・昭島製作所
〒196-8558 東京都昭島市武蔵野3−1−2 TEL：(042) 543−1111(大代表) FAX：(042) 546−3353
www.jeol.co.jp ISO 9001・ISO 14001 認証取得

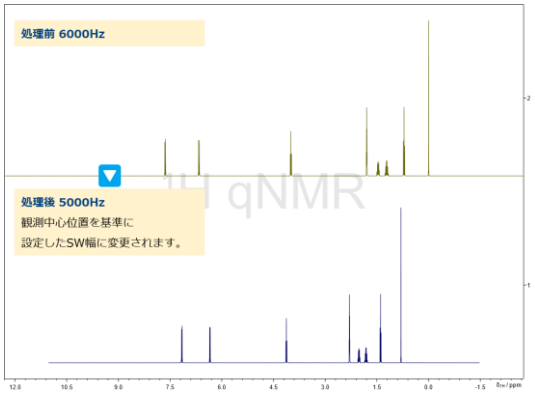


Figure 6. jasonParEdit.py SW出力例

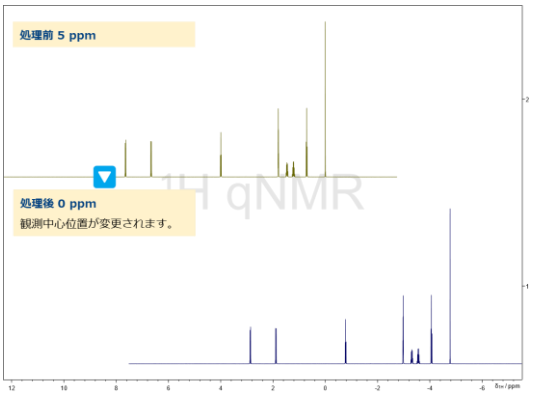


Figure 7. jasonParEdit.py SpectrumRef出力例

**出力結果**

Figure 6に、処理後のスペクトル出力例を示します。JASON GUIに反映されたスペクトルは、元データとの比較が可能です。

**エラーメッセージについて**

スクリプト実行時に設定に誤りがある場合、以下のような明確なエラーメッセージが表示されます：

- 未対応のパラメータ名：Error: parameter: XXX is not supported by this script.
- 存在しない属性：Error: parameter: XXX does not exist.
- 次元の指定ミス：Error: dimension index must be in range of 0–7.

これにより、Applyボタンを押した後に問題が発生しても、原因をすぐに特定できるようになっており、安全に設計されています。

**jasonParEdit.py: SpectrumRef の実施例**

Figure 7では、jasonParEdit.pyを用いた「SpectrumRef」の調整例を紹介します。操作方法はTable 4と同様です。Argumentsへのフルパス指定の例は以下の通りです（観測中心を0.0 ppmに変更する場合）：C:¥Users¥externalNMRProcessing¥python¥jasonParEdit.py –f $TMPFILE –p SpectrumRef –d 0 –v –0.0

## 補足：HDF5ファイル構造の探索

HDF5ファイルの構造が不明な場合は、以下のような探索コードを活用することで、格納されているデータの種類や位置を確認できます：

構造探索コード例：ファイル内のデーター覧を確認可能

```
import h5py

with h5py.File("your_file.jjh5", "r") as f:
    f.visit(print)
```

属性情報の確認：測定条件などのメタ情報（周波数、スイープ幅など）を取得可能

```
with h5py.File("your_file.jjh5", "r") as f:
    for key in f["JasonDocument/NMR/NMRData/0/SpecInfo"].attrs:
        print(key, f["JasonDocument/NMR/NMRData/0/SpecInfo"].attrs[key])
```

## まとめ

JASON × HDF5 × Python により、NMRデータの効率的な処理・解析が可能です。

- 階層構造を活用することで、必要な情報を柔軟に抽出できます。
- External Command機能により、Pythonなど外部プログラムとの連携が容易です。
- 独自の処理やレポート生成を組み込むことで、解析の自由度が向上します。
- HDF5構造の理解と探索コードの活用が応用の鍵になります。

**参照**

JASON公式サイト: https://www.jeoljason.com/
External NMR Processing Scripts: https://www.jeoljason.com/resources-external-nmr-processing-scripts/
Python公式ドキュメント: https://docs.python.org/ja/3/
h5pyライブラリ: https://www.h5py.org/
NumPyライブラリ: https://numpy.org/
ReportLabライブラリ: https://www.reportlab.com/documentation/

[1] JEOL Analytical Software Network
[2] Pythonは、Python Software Foundationの商標または登録商標です。
[3] SpecInfoは、NMR測定に関するメタ情報（測定条件など）を格納する構造です。SWはスイープ幅（測定範囲）、SpectrumRefは中心周波数、SpectrometerFrequenciesは観測周波数を意味します。
[4] .jjh5はJASONにおける.hdf5の形式です。



JEOL 日本電子株式会社
本社・昭島製作所
〒196-8558 東京都昭島市武蔵野3－1－2 TEL：(042) 543－1111(大代表) FAX：(042) 546－3353
www.jeol.co.jp ISO 9001・ISO 14001 認証取得