
Open Platform of Transparent Analysis Tools for fNIRS

解析ツール作成のためのステップガイド

2011.11.22

Version 1.03

Copyright(c) 2019,

National Institute of Advanced Industrial Science and Technology

本書は Platform for Optical Topography Analysis Tools (以下 POTATo と略す) における解析ツールの作成手順を簡単な例を用いて説明します。3 つの例を通して、解析ツール作成に必要な概要を理解することが目的です。このマニュアルは、Matlab[®] による関数作成の経験のある方を対象に作成しています。

- 解析ツールとは POTATo では
実際の解析処理を行うプログラムを「解析ツール」と呼びます。
POTATo のプラグイン形式に対応した解析ツールは、フダールを特定のフォルダにコピーするだけで POTATo に認識され利用可能となります。
- 解析ツールを作成する利点
POTATo のプラグイン形式に対応した解析ツールを作成することにより、既に POTATo 上にある他の解析ツールと簡単に組み合わせた解析処理を簡単に行うことが可能となります。WEB などを通して公開することにより、他の研究者が簡便に利用することも可能となります。

MATLAB[®] は The Mathworks 社の登録商標です。

1. 概要

解析ツール作成の手順は大きく 2 つの Step からなります。

Step1：メイン処理関数の作成

メイン処理関数は解析ツールの具体的な処理が記述された関数（mファイル）です。

Step2：POTATo への登録用関数の作成

メイン処理関数を POTATo 上で動作させるための関数です。これは後述する「Plugin Wizard」によりほぼ自動的に簡単に作成することができます。

メイン関数の実装イメージを図 1 に示します。実際の POTATo の実行時には解析ツールは POTATo の一つの機能として利用できます。図 2 に概念図を示します。

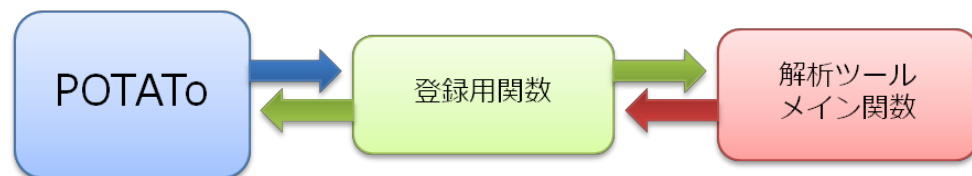


図 1. 解析ツールの POTATo への実装イメージ



図 2. POTATo 実行時の解析ツールのイメージ

このステップガイドでは 3 つの例を説明します。1 つ目は「Hello world!」です。解析には関係ありませんが、解析ツール作成の枠組みを用いて、Matlab コマンドプロンプトに Hello world という文字を表示させます。2 つ目は「平均値の算出」です。実際のデータから平均値を算出し、Matlab コマンドプロンプトに出力します。3 つ目は「データの追加」です。Oxy-Hb と Deoxy-Hb のデータから、両者の差を算出し、新たに Oxy-Deoxy というデータを追加します。

2. Hello World!

最初の例として Hello World と出力するプログラムを作成し、POTATo に組込みます。

Step1 : メコン関数の作成

メコン関数は以下に示すコードになります。

```
function P3P_HelloWorld

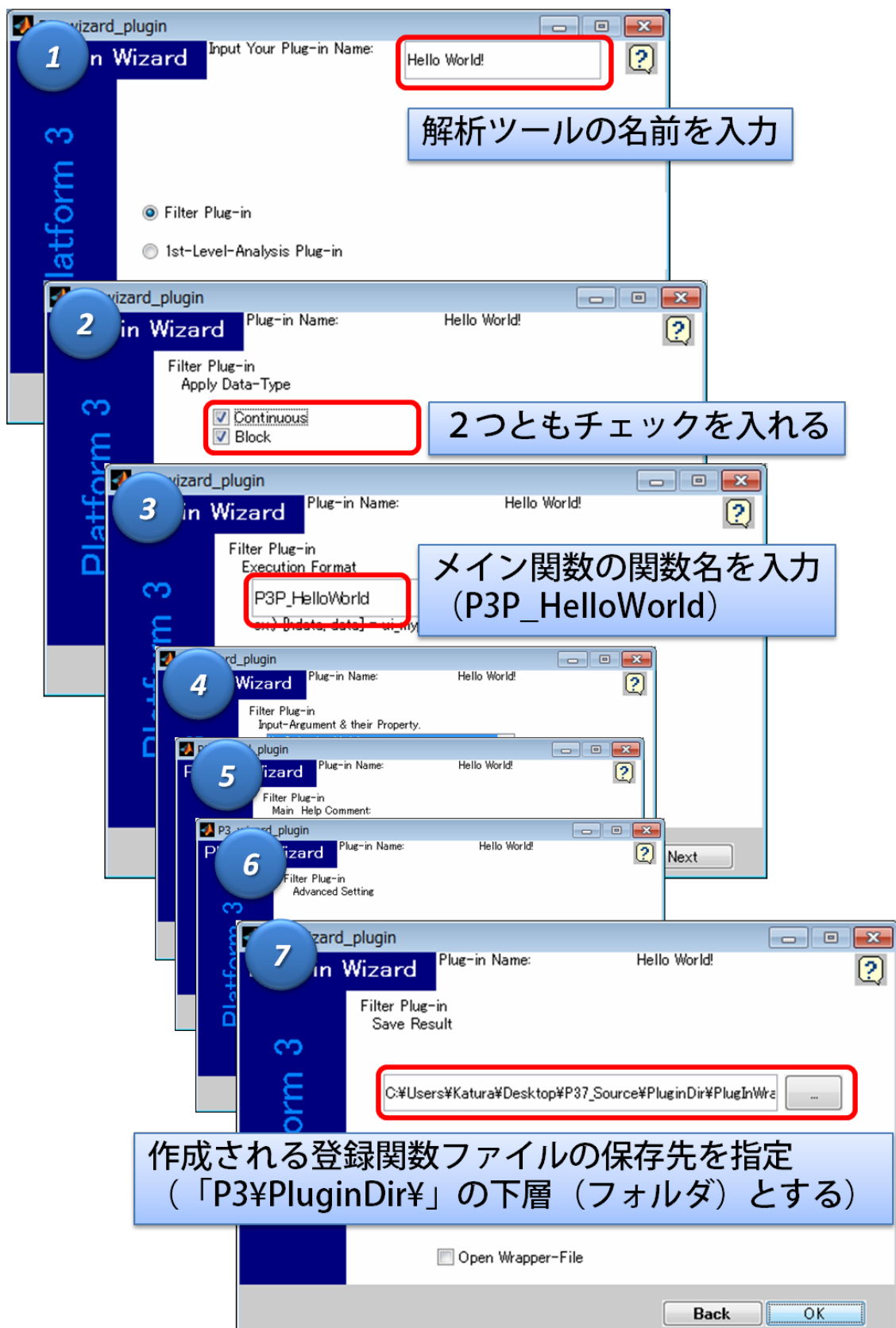
% Sample Program 1 for Filter Plug-in : Hello World!

disp('-----');
disp('  Hello Filter Plug-in World!  ');
disp('-----');
```

1 行目は関数の宣言です。POTATo での命名規則は特にありません。後述の登録関数作成時に関数名を利用します。今回作成するメコン関数では引数は用いないので、関数名のみを記述します。
2 行目にはコメントがあります。任意で設定してください。HTML 版のヘルプが存在しない場合には、この部分のコメントがヘルプメッセージとして表示されます。
4 行目以降がメコン部分です。ここでは disp 命令を用い、Matlab コマンドラインにメッセージを表示させます。
以上のコードを作成し、プラグイン名 : P3P_HelloWorld.m として保存します。保存先は POTATo フォルダ内の「PluginDir」フォルダの直下、またはその中に作成した任意のフォルダとします。POTATo では起動時に PluginDir 内をスキャンし、解析ツールを準備します。

Step2 : 登録用関数の作成

作成したメコン関数を POTATo に登録するための関数を作成します。登録関数の詳細は多くの事項があるため別のドキュメントで説明しますが、メコン関数を登録するためには詳細な情報は必要ありません。
登録関数の作成は、「Plugin wizard」機能を利用して自動的に作成します。まず、POTATo の「ツール」メニューから「Plugin Wizard」を選択し、ウケザード機能を立ち上げます。
次のページにウケザードの流れを示します。ウケザードは全 7 ステップです。入力が必要なのはステップ 1, 2, 3, 7 の 4 つの画面です。以下の赤枠で示した部分が入力部分です。(画面サイズは適宜縮小しています)



まず、①では解析ツールの表示名を入力します。次に②の **Data Type** では **Continuous, Block** の両方にチェックを入れます。③ではメコン関数の呼び出し命令を入力します。ここでは、メコン関数フーグル名、

P3P_HelloWorld

入力します。④、⑤、⑥は何もせずに **Next** を押します。最後に⑦では作成される登録関数のフーグル保存先を指定します。先ほどのメコン関数と同じ場所を指定してください。フーグル名はデフォルトのままにします。（**POTATo** ではフーグル名の先頭が「**PlugInWrap_**」であるものを登録関数フーグルとして識別します。）

以上で作成作業は完了です。**POTATo** を再起動し、作成した解析ツールを認識させます。メコン画面右側の **FilterList** 内に、作成した解析ツールが登録されていることを確認し、**Recipe** に追加します。次に **Recipe** を実行し、作成した解析ツールの動作を確認します。**Recipe** を実行する方法はいくつかありますが、ここでは **Draw** ボタンを押し、**Recipe** を実行させます。正しく動作した場合は、通常通りの **Figure** が描画され、**Matlab** コマンドドラゴンには「**Hello World!**」の文字が表示されます。

3. 平均値の算出

2つめの例では実際のデータを用い、平均値を算出します。この例を通し、解析フィルタで扱うデータについて理解します。

Step1：メロン関数の作成

メロン関数は以下に示すコードになります。

```
function P3P_Calc_Average(data)
% Sample Program: Calculate average

timePeriod = 10:100;
channel = [2 4 10];
dataType = 1;

res = mean(mean(mean(data(timePeriod, channel, dataType))));

disp('-----');
disp(sprintf('average = %f', res));
disp('-----');
```

1 行目は関数の宣言です。今回作成するメロン関数では実際のデータを計算に利用します。この場合は、「data」を入力引数とします。解析ツールの作成では「data」という変数名は特別な意味を持ちます。すなわち、POTATOのRecipe実行時に処理されているデータの変数名が「data」となっています。ですので、ここでの入力引数名はほかの名前に変更してはいけません。

2 行目にはコメントがあります。任意で設定してください。HTML版のヘルプが存在しない場合には、この部分のコメントがヘルプメッセージとして表示されます。

4 行目以降がメロン部分です。この解析ツールでは Continuous データのみを対象とすることにします。Continuous データの配列は、

[time point, channel, kind(data type)]

の 3 次元行列 (Double) です。time point は時系列のサンプリング点数、channel は計測チャンネル、kind(data type) は Oxy-Hb, Deoxy-Hb, Total-Hb の 3 つに対応します。例えば、10Hz で 1 分間、24 チャンネルの装置で計測したデータの行列サイズは、[600, 24, 3] となります。

上記のコードでは、サンプリング点 10～100 の期間を、チャンネル番号 2, 4, 10 の Oxy-Hb (kind == 1) のデータを平均し一つの値として res を算出しています。

以上のコードを作成し、フグ□ル名：P3P_Calc_Average.m として保存します。保存先は POTATo フォルダ内の「PluginDir」フォルダの直下、またはその中に作成した任意のフォルダとします。POTATo では起動時に PluginDir 内をスキャンし、解析ツールを準備します。

Step2：登録用関数の作成

作成したメ□ン関数を POTATo に登録するための関数を作成します。

登録関数の作成は、「Plugin wizard」機能を利用して自動的に作成します。

まず、POTATo の「ツール」メニューから「Plugin Wizard」を選択し、ウケザード機能を立ち上げます。ウケザードの流れを示す図は Hello World! と同様ですので省略します。

まず、①では解析ツールの表示名を入力します。次に②の Data Type では、今回の解析ツールでは Continuous のみを対象とすることとしましたので、Continuous のみにチェックを入れます。③ではメ□ン関数の呼び出し命令を入力します。ここでは、メ□ン関数名と入力引数を含む、

P3P_Calc_Average(data)

を入力します。④、⑤、⑥は何もせずに Next を押します。最後に⑦では作成される登録関数のフグ□ル保存先を指定します。先ほどのメ□ン関数と同じ場所を指定してください。フグ□ル名はデフォルトのままにします。(POTATo ではフグ□ル名の先頭が「PluginWrap_」であるものを登録関数フグ□ルとして識別します。)

以上で作成作業は完了です。POTATo を再起動し、作成した解析ツールを認識させます。メ□ン画面右側の FilterList 内に、作成した解析ツールが登録されていることを確認し、Recipe に追加します。次に Recipe を実行し、作成した解析ツールの動作を確認します。Matlab コマンドドラ□ンには算出された平均値が表示されましたか？

4. データの追加

最後の例では実際のデータを用い、新しいデータを作成し追加します。この例を通し、解析フィルタで扱うデータおよびヘッダデータの一部について理解します。

Step1：メロン関数の作成

メロン関数は以下に示すコードになります。

```
function [data, hdata]=P3P_Add_NewDataType(data,hdata)

% Sample Program: Add new data type

d_Oxy = data(:, :, 1);
d_Deoxy = data(:, :, 2);
d_OmD = d_Oxy - d_Deoxy;

data(:, :, end+1) = d_OmD;
hdata.TAGs.DataTag{end+1} = 'Oxy - Deoxy';
```

1行目は関数の宣言です。今回作成するメロン関数では実際のデータを計算に利用し、結果をデータとして返します。また、実際のデータ以外にもヘッダ情報も利用するので、出力を[data, hdata]と指定し、入力引数を「data, hdata」とします。解析ツールの作成では「data」および「hdata」という変数名は特別な意味を持ちます。すなわち、POTATOのRecipe実行時に処理されているデータの変数名が「data」となっており、付随する情報全てが「hdata」（構造体）となっています。ですので、ここでの入力引数名はほかの名前に変更してはいけません。

2行目にはコメントがあります。任意で設定してください。HTML版のヘルプが存在しない場合には、この部分のコメントがヘルプメッセージとして表示されます。

4行目以降がメロン部分です。この解析ツールではOxy-HbのデータとDeoxy-Hbのデータの差分を算出し、新たにOxy-Deoxyというデータを追加します。Oxy-Hbデータはd_Oxyに、Deoxy-Hbデータはd_Deoxyという変数に代入されます（4, 5行目）。そして、d_OmDにOxy-HbとDeoxy-Hbの差分が代入されます（6行目）。

今回の解析ツールで **Continuous** のみを対象とすることとします。したがって、**d_Oxy**, **d_Deoxy**, **d_OmD** の配列サイズは同一で

[time point, channel]

となります。次に、得られた結果を元の **data** 配列に追加します。入力時の **data** の配列サイズは

[time point, channel, 3]

です。これに対し、ここでは「**end+1**」という指定方法で 3 次元目にデータを追加します（8 行目）。この結果、**data** の配列サイズは、

[time point, channel, 4]

になります。これで **data** の変更は完了しました。しかし、このままでは **Draw** ボタンで描画しても結果は反映されません。それは、**Draw** ボタンでの描画時に、**data** のほかに、**hdata** から情報を読み込み利用しているからです。**Hdata** 内には **data** に関連する全ての情報が含まれていますが、この解析ツールのデータ追加に関連して変更しなければならない項目は 1 つだけです。**data** に含まれる **data type (kind)** の名前を保存している、

hdata.TAGs.DataTag

です。初期状態ではこの変数の内容は

{‘Oxy’, ‘Deoxy’, ‘Total’}

となっています。この名前を利用して **Draw** ボタンでは描画が行われます。そこで、今回追加した 4 番目のデータの名前を追加する必要があります。サンプルコードの 10 行目では、先ほどと同様に **end+1** によって、

‘Oxy-Deoxy’

という名前を追加しています。

以上でメロン関数の作成は完了です。関数内で変更された **data**, **hdata** はそのままでは破棄されてしまいますが、この関数では戻り値（出力）として **[data, hdata]** を指定しましたので、変更内容は **Recipe** 処理に引き継がれていきます。

Step2：登録用関数の作成

作成したメロン関数を **POTATO** に登録するための関数を作成します。登録関数の作成は、「**Plugin wizard**」機能を利用して自動的に作成します。**POTATO** の「ツール」メニューから「**Plugin Wizard**」を選択し、ウケザード機能を立ち上げます。ウケザードの流れを示す図は **Hello World!** と同様ですので省略します。

まず、①では解析ツールの表示名を入力します。次に②の **Data Type** では、今回の解析ツールでは **Continuous** のみを対象とすることとしましたので、**Continuous** のみにチェックを入れます。③ではメロン関数の呼び出し命令を入力します。ここでは、

`[data, hdata]=P3P_Add_NewDataType(data, hdata)`

を入力します。④、⑤、⑥は何もせずに **Next** を押します。最後に⑦では作成される登録関数のフーネル保存先を指定します。先ほどのメロン関数と同じ場所を指定してください。フーネル名はデフォルトのままにします。（POTATo ではフーネル名の先頭が「**PlugInWrap_**」であるものを登録関数フーネルとして識別します。）

以上で作成作業は完了です。POTATo を再起動し、作成した解析ツールを認識させます。メロン画面右側の **FilterList** 内に、作成した解析ツールが登録されていることを確認し、**Recipe** に追加します。次に **Recipe** を実行し、作成した解析ツールの動作を確認します。**Layout** から **Line plot** を選択し、**Draw** ボタンを押してみてください。**DataType (kind)** リストに新しいデータが追加されていますか？そのリストから新しいデータを選択すると、**Oxy - Deoxy** の波形が表示されましたか？