
Open Platform of Transparent Analysis Tools for fNIRS

解析の拡張

国立研究開発法人 産業技術総合研究所

目次

1. 解析機能.....	2
1.1. POTAToにおける解析	2
1.2. 解析処理	2
2. 解析に関わるデータ構造	3
2.1. 解析データ.....	3
2.2. レシピ	3
3. フィルタの拡張	5
3.1. 関数インタフェース	5
3.2. createBasicInfoサブ関数	6
3.3. getArgument サブ関数.....	7
3.4. writeサブ関数.....	8
4. 補助関数.....	12
4.1.1. uc_dataload関数.....	12
4.1.2. nan_fcn関数	12

1. 解析機能

1.1. POTATo における解析

ここでは Platform for Optical topography Analysis Tools(POTATo)の Research モードの解析準備(Preprocessor)における単体データ解析の拡張方法を説明します。

POTATo では解析の手順をレシピと呼んでおり、ファイルも可能です。またレシピは複数の解析関数(フィルタ)から構成されており、変数や実行順序を変更可能です。

- レシピは再利用・配布可能
- レシピは編集可能
- 解析関数(フィルタ)を追加可能

ここでは解析関数の追加に関して説明します。

簡単な解析関数の追加方法は別途【解析ツール作成のためのステップガイド.pdf】をご参照ください。ここでは、関連する内容も含め POTATo の内部構造に近い説明を行います。

なお、ここではプログラムコードの作成を前提としていますので、プログラムサイドからの説明になります。

1.2. 解析処理

Research モードの解析準備(Preprocessor)における単体データ解析では解析データ(Analysis データ)内になるレシピ(解析手順)を編集し、解析を実施します。

その結果、POTATo データを出力します。

解析データは、実験データ(Raw データ)の名前とレシピを持っています。

このレシピを編集するため、POTATo は解析関数のリストとその特徴を取得します。その後、ユーザの操作により解析関数を加えたり削除したりすることによりレシピを編集します。

解析の実施では解析データから解析用の M-ファイルを作成します。M-ファイルの作成では、実験データ名からデータの読込処理を、レシピに登録されている関数からフィルタ処理をファイルに書き出します。

作成した M-File を実行することにより、解析後の POTATo データが取得できます。

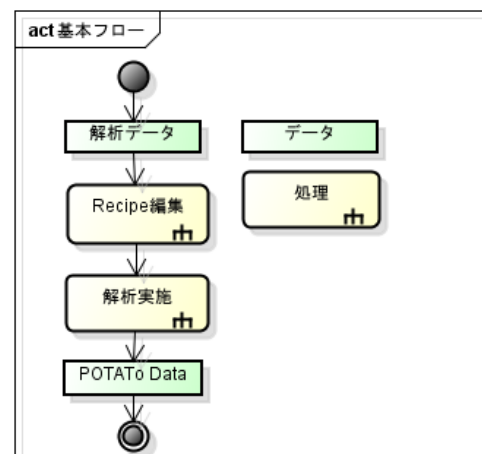


図 1.1 解析処理

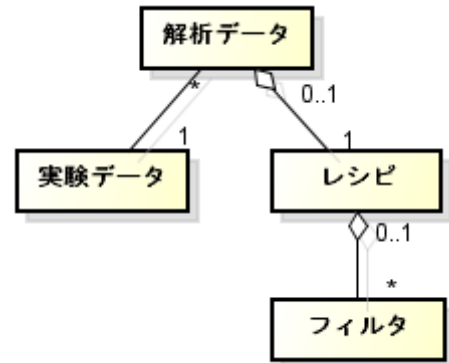
2. データ構造

2.1. 解析データ

解析を実施するために必要なデータは解析データです。
解析データの構造を右図に示します。

解析データは1つの実験データ(旧バージョンは複数可)と1つのレシピを持っています。レシピは複数のフィルタから出来ています。

ここでフィルタには、連続データに対して行うフィルタ、ブロッキング用フィルタ、区間データに対して行うフィルタの3種類があります。



具体的にはレシピは解析データ(AnalysisData)以下に
AnalysisData.data.filterdata に構造体として保存されます。

図 2.1 解析実施に必要なデータ

2.2. レシピ

レシピは解析手順を示しますが以下のような構造体です。

表 2.1 レシピ構造体

フィールド名	内容	例
dummy	空構造体为了避免のための ダミーフィールド	'No Effective Data'
HBdata	連続データに対するフィルタリスト	{ FilterData, FilterData }
block_enable	ブロック有効フラグ	1
BlockPeriod	ブロック期間	[5 15]
TimeBlocking	ブロック化フィルタ	{ FilterData }
BlockData	区間化データに対するフィルタリスト	{ FilterData, FilterData }

全てのフィールドは対象とする解析を実施しない場合省略できますが、レシピは空でない構造体にする必要があります。そこで、解析がなく、レシピ構造体が空になる場合は dummy フィールドで何かの値を設定します。

解析は、最初に連続データに対してフィルタを適用し、次にブロッキングを行い、最後に区間データに対してフィルタを適用します。ここでフィルタに関するデータ、フィルタデータ(FilterData)に関しては後述します。

ここで、ブロッキングに関しては複数の設定があります。block_enable はブロック化の有効/無効を示します。block_enable==0 の場合はブロック化以降の処理を行いません。

BlockPeriod はブロッキングの共通設定項目で、この項目がない場合もブロック化以降の処理

を行いません。

TimeBlockign はブロッキング用のフィルタですが、省略時はデフォルトのブロッキング処理が行われます。なおデフォルトの処理では全刺激の種類に対してブロッキングを行います。

3. フィルタの拡張

3.1. 関数インタフェース

フィルタは POTATo 内の PluginDir フォルダ以下に PlugInWrap_*.m というファイル名で作成します。POTATo はサブフォルダ以下も検索しますので、フォルダを分けることで整理することができます。また、p コード化したファイルも検索しますので、必要があれば p コード化してください。

この関数は以下のようなインタフェースを持ちます。

PlugInWrap_*('subfname', [arg1, arg2, ...])

ここで subfname にサブ関数名が入り、arg1, arg2... はサブ関数の引数です。

用意すべきサブ関数は以下 3 つになります。

表 3.1 フィルタ:サブ関数

サブ関数名	内容
createBasicInfo	基本情報設定
getArgument	フィルタデータを作成する
write	解析を実施するための M-File を作成する

それぞれのサブ関数の引数や用途は決まっており、ここでは各サブ関数について説明します。

なお、これらの関数の骨格となるコードは、他のフィルタ関数をコピーするか、“P3_wizard_plugin”の Filter Plug-in で作成できます。

また、関数ヘルプ用のコメント(M-ファイルの最初のコメント)はヘルプとして利用されますので、正しく入力することを推奨します。

3.2. createBasicInfo サブ関数

createBasicInfo サブ関数はフィルタ関数の基本的な情報を返します。POTATo でレシピを編集する際など、システムの制御に利用します。なお、結果である基本情報は固定値にしてください。

シンタックス	info=createBasicInfo	
機能	指定した name, ID に対応する AO 関連データを更新する	
出力	info	基本情報(構造体)

ここで、基本情報構造体は以下のフォーマットです。

表 3.2 フィルタ基本情報

フィールド名	内容	例
name	フィルタ名	'MyFilter'
region	利用可能領域を示す配列 2: 連続データに対応 3: 区間データに対応 -1: ブロッキング(この場合配列不可)	[2 3]
DispKind	表示種別	0
Description	M-File 作成時のフィルタの説明	'MyFilter: band-pass'

ここで name は表示するフィルタ名です。name は POTATo 内でユニークであるべきです。もし同一の名前が記載されると起動時、毎回警告が出力されます。

region はそのフィルタが実行可能なデータを示します。通常、2(連続データ)もしくは3(区間データ)を記載します。両方のデータに対応する場合は[2, 3]と複数記載できます。

DispKind はそのフィルタの種別を示します。BookMark など、表示するフィルタリストの絞込みに使われていましたが、現在は普通のフィルタでは使われていません。

ただし、Developer-Mode で利用する 1st-Level-Analysis 用のフィルタを作成する場合は F_1stLvlAna を指定してください。

3.3. getArguments サブ関数

解析を行うための引数設定を行います。

シンタックス	fdata=getArgument(fdata, mfile)	
機能	解析を行うためのフィルタデータを設定する	
入力	varargin	その他の引数。 現行バージョンの POTATo には varargin{1}に フィルタ実施直前のデータを作成するためのスクリ プトファイルが入っている。
入出力	fdata	フィルタデータ

POTATo 内の GUI より呼び出されます。新規作成時は初期データのみ与えられますが、更新時、fdata には以前の fdata が設定されます。

キャンセルする場合は return 前に fdata=[];と設定します。

フィルタデータのフォーマットは以下の通りです。

表 3.3 フィルタデータ

フィールド名	内容	備考
name	フィルタ名、基本情報と同じ。	変更不可。
wrap	関数名	変更不可。
argData	解析で利用するデータ。	構造体。 自由に作成可能。

ここでフィルタデータは POTATo 内では enable フィールドを持ちます。enable は 'on' もしくは 'off' の値をとり、フィルタの有効/無効を示しています。

レシピ内(参照:表 2.1 レシピ構造体)のフィルタデータはこの enable フィールドが付加されています。

また、引数設定前の POTATo データを取得したい場合、以下のように利用してください。

```
mfile=varargin{1};  
[data, hdata]=scriptMeval(mfile, 'data', 'hdata');
```

ここで、この関数で得られる POTATo データは周辺のレシピが変更されることにより変更されます。変更時には引数設定直前の POTATo データが取得できます。

3.4. write サブ関数

解析実施時、POTATo はレシピより一旦 M-File を作成し、その後作成した M-File を実行することで解析を実施します。

そのため、write サブ関数では解析を実施するための M-File 作成を行います。

シンタックス	str = write(region, fdata)	
機能	AO: 描画処理のための文字列作成	
入力	region	領域を示す文字列。
		連続データ: 'HBdata'、区画データ: 'BlockData'
	fdata	フィルタデータ
出力	str	Axis-Area 内の AO 描画処理で実行する文字列

ここで、M-File の作成時の POTATo とフィルタの関係を示します。

あらかじめ POTATo は 1. getArgument をフィルタに実施し、FilterData を作成します。この時 FilterData はレシピ内に保存されます。

次に POTATo は POTATo 内の補助関数 make_mfile 関数を用いて作成する M ファイルを開きます。

その後、レシピに従い M ファイルを作成します。このときフィルタに対して createBasicInfo を用いて、ヘルプ用のヘッダ等も作成します。

POTATo がフィルタ実施用コードの記載が必要になると、フィルタに対して 3.write を実施します。

フィルタの write サブ関数内で M ファイルに直接書き込みをする場合、make_mfile を用いて書き込みを実施します。

その後、POTATo はフィルタ write サブ関数の戻り値、str を書き込んだり、残りの M ファイルの作成を行ったりします。最後に make_mfile に 4close を実施し、必要に応じて作成された M ファイルを実施したり、開いたりします。

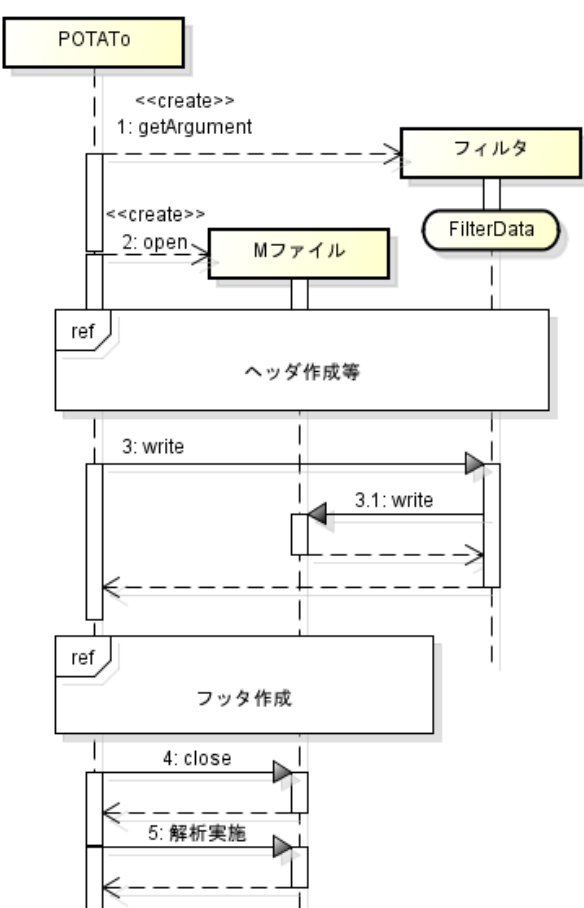


図 3.1 解析実施に必要なデータ

典型的な write サブ関数の例を示します。

フィルタがMファイルを作成する際、適切なインデント(行頭の空白)をいれたり、統一されたルール沿ってコードの区切り文いれたりすることで読みやすいMファイルが作成できます。このようにファイルを整形するためのツールとしてmake_mfile関数が提供しています。make_mfileのシンタックスは以下のようになります。

```
make_mfile('サブ関数名', 引数);
```

主な利用方法は以下になります。

表 3.4 make_mfile

サブ関数	引数	内容
write	Str	文字列 Str を M-File に出力
with_indent	Str	インデント付きで 文字列 Str を M-File に出力
code_separator	level	level に応じた区切り文を出力 1～10 の整数で小さいほど重要な区切りとする
indent_fcn	'up'	インデントを上げる(空白文字を増やす)
	'down'	インデントを下げる

ここでStrは整形済みのchar配列としてください。cellで記載することも可能ですが、非推奨です。

writeサブ関数は通常以下の様に開始します。

```
function str = write(region, fdata) %#ok
% input : fdata
str='';
bi=createBasicInfo;
```

ここで、strで結果を返すとインデント調整ができないので、初期値として空白で返すよう予め設定しておきます。

基本情報はgetArgumentにより作成されたfdataのバージョンとwriteサブ関数実行時のバージョンを比較すること、またM-Fileにwriteサブ関数のバージョンを残すことに利用します。

次にフィルタ用のヘッダを記載することを推奨しています。

コードの区切りをレベル3にし、フィルタ名、writeサブ関数のバージョン表示を行います。

```
% *****
%   Header Area
% *****
make_mfile('code_separator', 3);
make_mfile('with_indent', ['% == ' fdata.name ' ==']);
make_mfile('with_indent', sprintf('%% Version %f', bi.Version));
make_mfile('code_separator', 3);
make_mfile('with_indent', '');
```

次に実際の実行部分を記載します。実行部分はtry-catch文で括るべきです。

```
% *****
%   Exexute Area
% *****
make_mfile('with_indent', 'try');
make_mfile('indent_fcn', 'down'); % インデントを下げる

<<< ここに実行部分を記載 >>

make_mfile('indent_fcn', 'up'); % インデントを戻す
make_mfile('with_indent', 'catch');
make_mfile('indent_fcn', 'down');
% エラー処理
make_mfile('with_indent', 'error(lg(lasterr));');
make_mfile('indent_fcn', 'up');
make_mfile('with_indent', 'end');
make_mfile('with_indent', '');
```

最後に実行部分を記載する。実行部分を記載する際、変数の有効範囲(スコープ)に注意してください。

writeサブ関数内で使える変数は引数のregion, fdataおよびフィルタ基本情報です。ここで引数が格納されているフィルタデータ(fdata)は作成するMファイル中には存在しません。

作成するMファイル内で使える変数はPOTAToデータです。

表 3.5 変数とスコープ

スコープ	変数名	種類	内容
write サブ関数内	region	引数	領域を示す文字列。
			連続データ: 'HBdata'、区画データ: 'BlockData'
	fdata	引数	フィルタデータ(表 3.3 フィルタデータ)
	bi	定数	フィルタ基本情報(表 3.2 フィルタ基本情報) createBasicInfo で取得
作成	dataname	入力	計算対象実験データファイル名(セル)
M ファイル内	hdata	計算対象	POTATo データ
	data		フィルタ実行直前
	chdata	計算結果	POTATo データ
	cdata		ブロック化直前の連続データ(ブロック化後のみ)
	bhdata	計算結果	POTATo データ
	bdata		解析完了後の区間データ(通常なし)

writeサブ関数内の変数はサブ関数内で定義することにより、追加されます。ここで変更したデータは他に影響を及ぼしません。

作成Mファイル内の変数は他のフィルタにより変更されたり追加されたりします。もし、作成するフィルタが他のフィルタを必要とする場合、適切な説明やエラー文を入れることを推奨します。

4. 補助関数

フィルタを実施する上で利用可能な補助関数を示します。

表 4.1 補助関数リスト

関数名	サブ関数名	内容
uc_dataload	–	データ読込
nan_fcn	‘mean’	NaN データを除いた mean(x,dim)
	‘std0’	NaN データを除いた std(x,0,dim)
	‘std1’	NaN データを除いた std(x,1,flag)を計算。
make_mfile	–	M ファイル作成用ツール。 3.4 write サブ関数 で利用する。
msglistbox	–	msgbox と同じ。 ただしメッセージ部分がリストボックスになり 多くの文字列を表示できる。
uihelp	–	ヘルプを表示する

4.1.1. uc_dataload 関数

実験データから POTATo 連続データを読み込む。

[data, hdata] = uc_dataload(‘ファイル名’);

ここで、M ファイル作成時、ファイル名は dataname{1}で与えられる。

4.1.2. nan_fcn 関数

POTATo データには NaN が混在している場合があります。POTATo データにおける NaN は多くの場合、単に無効なデータという意味です。

そのためこれらのデータを無視する関数を提供します。

シンタックス	[data, n] = nan_fcn(fcn,data, dim)	
機能	NaN を無視した平均値・標準偏差の計算。	
入力	fcn	関数タイプ: ‘mean’ → mean(data, dim)を実施 ‘std0’ → std(data,0,dim)を実施 ‘std1’ → std(data,1,dim)を実施
	data	計算対象データ
	dim	対象次元
	data	結果
	n	計算に利用した有効なデータ数。

NaN 設定の経緯:

POTATo の Motion Check(体動)フィルタは体動と判断した時点の測定データに対してフラグを立てます。連続データの場合はその時刻、チャンネルに対してフラグを立て、区間データの場合はブロック、チャンネルに対してフラグを立てます。

このフラグの扱いは解析方法の指針により異なります。対応方法のひとつとして、フラグが立っているデータを無効にするため、対応箇所のデータを NaN に設定する方法があります。

4.1.3. uihelp 関数

フィルタプラグインのヘルプは uihelp 関数により表示されます。作成するフィルタプラグインは以下のヘルプ機能に対応することを推奨します。

シンタックス	h = uihelp(fname)	
機能	関数のヘルプ表示	
入力	fname	関数名
出力	h	uihelp の figure ハンドル

uihelp は入力関数からヘルプドキュメントを取得し表示します。

また、PDF や html によるヘルプドキュメントがある場合はヘルプドキュメントを開くためのボタンを有効化します。

PDFやhtmlのヘルプドキュメントの検索は関数と同じパスの以下のファイルを検索し、最初に見つかったものを開きます。

XXX/関数名.pdf
XXX/関数名.html
関数名.pdf
関数名.html

ここで、XXX は POTATo 内の言語設定名で、現在は日本語 “ja”です。