

モデル間相互比較研究のための
汎用データ解析ツール
jstream_tools
Version 1.1.3

ユーザーマニュアル

令和1年8月
日本気象株式会社

1. 概要

解析ツール `jstream_tools` は、大気化学輸送モデルの出力データから必要な情報を抽出し、テキストファイルや `netCDF` ファイルに出力します。また、抽出されたファイルを元に時系列統計処理も行います。

2. インストール

2.1. python のインストール

本ツールはスクリプト言語 `python` をベースに作成されています。`python` がインストールされていない場合は、インストールして下さい。`python` がインストールされているかどうかは、以下のコマンドを実行し、バージョン番号が返ってくるかどうかでわかります。

```
$ python --version
```

以降、管理者権限でインストールする場合は、管理者でログインして # で始まるコマンドを実行してください。管理者権限がない場合は、# を \$ に置き換えて、同様に実行してください。

`python` のパッケージインストーラ `pip` がインストールされていない場合は、以下を実行して下さい。

- i. `get-pip.py` をダウンロードして下さい。
- ii. ファイルをダウンロードしたディレクトリに移動して、以下を実行して下さい。

```
# python get-pip.py
```

さらに、`jstream_tools` に必要な `python` パッケージを以下のようにインストールして下さい。

```
# pip install numpy  
# pip install pandas  
# pip install netCDF4
```

2.2. 解析ツール (jstream_tools) のインストール

2.2.1. 配布ファイルの展開

```
$ tar zxvf jstream_tools-1.1.3.tar.gz
$ cd jstream_tools-1.1.3
```

2.2.2. jstream_tools のインストール

以下のコマンドを実行して下さい。

```
$ make
# make install
```

インストール先の情報は、files.txt に出力されます。以下に例を示します。

```
パッケージ本体      python デフォルトの site-packages 内
  管理者権限のある場合: /usr/local/lib/python2.7/site-packages/jstream_tools/
  管理者権限のない場合: $HOME/.local/lib/python2.7/site-packages/jstream_tools/
実行コマンド      実行ファイル用ディレクトリ内
  管理者権限のある場合: /usr/local/bin/
  管理者権限のない場合: $HOME/.local/bin
```

コマンドは重複を避けるために jstream_get_point_data のような冗長な名前になっています。他のコマンド名と重ならないように、簡単な名前にリンクすることをお勧めします。

```
(例) $ ln -sf ~/.local/bin/jstream_get_point_data ~/bin/gpd
```

2.2.3. jstream_tools のアンインストール

インストールを行った directory において、以下を実行して下さい。

```
# make uninstall
```

インストール情報ファイル files.txt がなくなっている場合は、2.2.2 のインストールコマンドを再実行して files.txt を作成して下さい。

2.2.4. 補足事項

解凍したパッケージ directory (jstream_tools) 内のファイルを使ってツールを実行することもできます。必要なライブラリ gctp.so は以下を実行して作成して下さい。

```
$ f2py -c gctp.pyf gctp.f
```

解析ツールは以下のように実行することができます。

```
$ python get_point_data.py
```

3. jstream_tools の使い方

基本的なコマンドは以下の5つです。

コマンド名	機能	デフォルトパラメータファイル
<code>jstream_get_point_data</code>	地上地点時系列データ抽出	<code>gpd_params.txt</code>
<code>jstream_get_vert_data</code>	鉛直分布時系列抽出	<code>gvd_params.txt</code>
<code>jstream_get_grid_data</code>	グリッド時系列抽出	<code>ggd_params.txt</code>
<code>jstream_stats_point_data</code>	地上地点統計量計算	<code>spd_params.txt</code>
<code>jstream_stats_grid_data</code>	グリッド統計量計算	<code>sgd_params.txt</code>

コマンドの第一引数にパラメータファイル名を与えます。

(例：`$ jstream_get_point_data gpd_params_1.txt`)

パラメータファイルの書式は、JSON と INI の混合形式です。セミコロン (;) から行末までは無視されます。

パラメータファイルの例は、パッケージディレクトリ、および、インストールディレクトリの `jstream_tools/sample_params` 内にあります。

第一引数を省略した場合は、デフォルトとして上表の名前のファイルを読み込もうとします。

3.1. jstream_get_point_data

日にちごとの netCDF ファイルから、指定地点（複数）における指定変数のデータを抽出します。複数日の計算結果が連続的にまとめられ、変数毎のファイルが出力されます。出力ファイルの time は、元の netCDF ファイルと同じ定義です。指定地点が netCDF ファイルの領域外にある場合は、その地点のデータは nan として出力されます。

パラメータファイル（デフォルト名 `gpd_params.txt`）の説明

`point_file`: 地点名、経度、緯度が記録された csv 形式のファイル名（ア）
`cctm_basedir`: cctm 出力の netCDF ファイルが存在するディレクトリ
`CONC_fileprefix`: cctm 出力の濃度 netCDF ファイル名の日付部より前の部分
`CONC_filepostfix`: cctm 出力の濃度 netCDF ファイル名の日付部より後の部分（イ）
`AD_fileprefix`: PM2.5 関連の変数を出力する場合、AERODIAM ファイル名の日付部より前の部分
`AD_filepostfix`: PM2.5 関連の変数を出力する場合、AERODIAM ファイル名の日付部より後の部分（イ）
`cctm_dateformat`: cctm 出力の netCDF ファイルの日付形式
(例) `Ymd <- 20180327, Yj <- 2018086`
`mcip_basedir`: mcip 出力の netCDF ファイルが存在するディレクトリ
`METCRO2D_fileprefix`: METCRO2D ファイル名の先頭部分（ウ）
`METCRO3D_fileprefix`: METCRO3D ファイル名の先頭部分（ウ）
`METCRO_filepostfix`: METCRO ファイル名の末尾部分（イ）
`mcip_dateformat`: mcip 出力の netCDF ファイルの日付形式
`cmaq_version`: CMAQ のバージョン（例：5.1）（エ）
`chemmech`: CMAQ の化学メカニズム（例：saprc07tc_ae6_aq）（エ）
`select_vars`: 出力する変数名（オ）
`outfile_prefix`: 出力ファイルの変数名より前の名前
`init_date`: 最初の日付。年/月/日 の形式
`end_date`: 最後の日付。年/月/日 の形式

(ア) 地点ファイルの書式は以下の例に従って下さい。

```
name, lon, lat <— 一行目は変更しない
dot_0_0, 138.5252, 34.28518 <— 地点名、経度、緯度
dot_1_0, 138.5805, 34.2859
dot_0_1, 138.5243, 34.33089
dot_1_1, 138.5797, 34.3316
```

cro_0_0, 138.5524, 34.30838

center, 139.8, 34.0

- (イ) ファイル名が日付で終了する場合は、指定する必要はありません。
- (ウ) 通常は、METCRO2D、または、METCRO3D です。
- (エ) 本ツールが対応している CMAQ のバージョンと化学メカニズムの組み合わせは、本パッケージの `jstream_tools/spec_def_files/` 以下のディレクトリ名を参照してください。

`cmaq_version` が 4.7.1, 5.0.1, 5.0.2 の場合は、化学メカニズムの区別はなく、`chemmech` の値は無視されます。

- (オ) netCDF ファイルに元々含まれている変数 (素変数) については、netCDF ファイルの変数名 (例: O3, AXYL1J) を指定します。

複数の素変数に依存する派生変数については、下表の CMAQ ソースコードに定義されています。

バージョン	ディレクトリ	定義ファイル
4.7.1	<code>models/TOOLS/src/combine/</code>	<code>spec_def.[dep,conc],v</code>
5.0.2	<code>tools/combine/</code>	<code>spec_def.[dep,conc]</code>
5.1	<code>scripts/tools/combine/spec_def_files/</code>	<code>species_*.txt</code>
5.2, 5.2.1	<code>POST/combine/scripts/spec_def_files/</code>	<code>SpecDef_*.txt</code>

これらの定義は、本パッケージの `jstream_tools/spec_def_files/` の中に、バージョン、および、化学メカニズムごとに `spec_def.csv` というファイルにまとめられています。

`spec_def.csv` には、独自の派生変数を定義することができます。書式は既存の派生変数に従ってください。なお、バックスラッシュ (\) を用いて、定義式の途中で改行することができます。

素変数と派生変数の両方に登録されている変数は、素変数が優先されます。

(例: O3 は素変数に単位 ppm で記録されていますが、`spec_def.csv` にも素変数を 1000 倍した単位 ppb の派生変数としても登録されています。このプログラムは素変数を優先して、単位 ppm の値を出力します。)

派生変数の定義式の解読処理には、Louis Fischer による `Mathematical expressions parser` を元に改造したクラスを使用しています。

派生変数の定義については、以下の論文もご参照ください。

Supplement of Geosci. Model Dev., 10, 1703-1732, 2017

<http://www.geosci-model-dev.net/10/1703/2017/>

doi:10.5194/gmd-10-1703-2017-supplement

多くの変数に依存する変数（PM25_UNSPEC1 など）は、計算に時間がかかりますので、一度、少ない地点数や日数で計算時間を確認することをお勧めします。

(カ) 例えば、`outfile_prefix = test_` で、`select_vars = ["O3","NO"]` の場合には、出力ファイルは、`test_O3.csv` と `test_NO.csv` になります。

3.2. jstream_get_vert_data

日にちごとの netCDF ファイルから、指定地点（1回の実行あたり1箇所）における指定変数の鉛直分布データを抽出します。複数日の計算結果が連続的にまとめられ、変数毎のファイルが出力されます。

出力ファイルの time は、元の netCDF ファイルと同じ定義です。

パラメータファイル（デフォルト名 gvd_params.txt）の説明

point_file: 地点名、経度、緯度が記録された csv 形式のファイル名。(ア)

cctm_basedir: cctm 出力の netCDF ファイルが存在するディレクトリ

CONC_fileprefix: cctm 出力の濃度 netCDF ファイル名の日付部より前の部分

CONC_filepostfix: cctm 出力の濃度 netCDF ファイル名の日付部より後の部分

3.1(イ)と同様

AD_fileprefix: PM2.5 関連の変数を出力する場合、AERODIAM ファイル名の日付部より前の部分。

AD_filepostfix: PM2.5 関連の変数を出力する場合、AERODIAM ファイル名の日付部より後の部分。3.1(イ)と同様。

cctm_dateformat: cctm 出力の netCDF ファイルの日付形式

(例) Ymd <- 20180327, Yj <- 2018086

mcip_basedir: mcip 出力の netCDF ファイルが存在するディレクトリ

METCRO2D_fileprefix: METCRO2D ファイル名の先頭部分

3.1(ウ)と同様

METCRO3D_fileprefix: METCRO3D ファイル名の先頭部分

3.1(ウ)と同様

METCRO_filepostfix: METCRO ファイル名の末尾部分

3.1(イ)と同様

mcip_dateformat: mcip 出力の netCDF ファイルの日付形式

cmaq_version: CMAQ のバージョン (例: 5.1) 3.1(エ)と同様

chemmech: CMAQ の化学メカニズム (例: saprc07tc_ae6_aq) 3.1(エ)と同様

select_vars: 出力する変数名。3.1(オ)と同様

outfile_prefix: 出力ファイルの変数名より前の名前。3.1(カ)と同様。

init_date: 最初の日付。年/月/日 の形式

end_date: 最後の日付。年/月/日 の形式

(ア) 地点ファイルの書式は、3.1(ア)と同様ですが、複数地点が記載されている場合は、最初の1地点のみが抽出対象となります。

3.3. jstream_get_grid_data

日にちごとの netCDF ファイルから、指定変数のデータを抽出します。

複数日の計算結果が連続的にまとめられ、指定変数のみを含む netCDF ファイルが 1 つ出力されます。

出力ファイルの TFLAG は入力ファイルのものと同じです。

パラメータファイル（デフォルト名 `ggd_params.txt`）の説明

`cctm_basedir`: cctm 出力の netCDF ファイルが存在するディレクトリ

`CONC_fileprefix`: cctm 出力の濃度 netCDF ファイル名の日付部より前の部分

`CONC_filepostfix`: cctm 出力の濃度 netCDF ファイル名の日付部より後の部分
3.1(イ)と同様

`AD_fileprefix`: PM2.5 関連の変数を出力する場合、AERODIAM ファイル名の日付部より前の部分。

`AD_filepostfix`: PM2.5 関連の変数を出力する場合、AERODIAM ファイル名の日付部より後の部分。3.1(イ)と同様。

`cctm_dateformat`: cctm 出力の netCDF ファイルの日付の形式

(例) `Ymd <- 20180327, Yj <- 2018086`

`mcip_basedir`: mcip 出力の netCDF ファイルが存在するディレクトリ

`METCRO2D_fileprefix`: METCRO2D ファイル名の先頭部分
3.1(ウ)と同様

`METCRO3D_fileprefix`: METCRO3D ファイル名の先頭部分
3.1(ウ)と同様

`METCRO_filepostfix`: METCRO ファイル名の次元 (2D, 3D) より後の部分
3.1(イ)と同様

`mcip_dateformat`: mcip 出力の netCDF ファイルの日付形式

`cmaq_version`: CMAQ のバージョン (例 : 5.1) 3.1(エ)を参照

`chemmech`: CMAQ の化学メカニズム (例 : `saprc07tc_ae6_aq`) 3.1(エ)を参照

`combined_file`: 出力される netCDF ファイルの名前

`combined_file_type`: 出力される netCDF ファイルの形式。(ア)
`cctm, metcro2d, metcro3d` のいずれか。

`select_vars`: 出力する変数名。3.1(オ)を参照。

`init_date`: 最初の日付。年/月/日 の形式

`end_date`: 最後の日付。年/月/日 の形式

(ア) ツールへの入力ファイルとしては、netCDF のバージョンは 3 でも 4 でも受け付けます。一方、出力ファイルはバージョン 3 に限定しています。

3.4. jstream_stats_point_data

`jstream_get_point_data` または `jstream_get_vert_data` の出力ファイルに対し、日平均 (dmean)、日最高値 (dmax)、日最低値 (dmin)、8 時間値 (8hr)、時刻平均値 (diurnal) を計算してテキストファイルに出力します。

日の区切りは、指定のタイムゾーンで行われます。

出力ファイルの時間ラベルは以下のように定義されます。時間ラベルは LST です。CTM 出力の慣例に従い、時間データは後 1 時間平均を表します (例: 0 時データ = 0:00-1:00 の平均)。

日統計値 (dmean, dmax, dmin) date = 20130722 の場合
20130722 00,01,02,...,23LST の統計値
(タイムゾーンが Asia/Tokyo の場合は、20130721 15UTC,...,20130722 14UTC)
8 時間値 (8hr) time = 20130722_16 の場合
20130722 09,10,11,...,16LST の平均値
(タイムゾーンが Asia/Tokyo の場合は、20130721 00,01,...,07UTC)
時刻平均値 (diurnal_?) Hour = 0 の場合
指定期間の 00LST (0:00LST - 1:00LST) の平均値
(タイムゾーンが Asia/Tokyo の場合は、15UTC (15:00UTC -- 16:00UTC))

パラメータファイル (デフォルト名 `spd_params.txt`) の説明

`basedir`: `jstream_get_[point,vert]_data` の出力ファイルが置かれている directory

`fileprefix`: `jstream_get_[point,vert]_data` の出力ファイル名の変数名の前の部分

`select_vars`: 統計処理を行なう変数

`day_type`: 時刻平均値を計算する日のタイプ。現バージョンでは以下の 3 種類。(ア)

0 - 全日、1 - 祝日を除く平日、2 - 土日祝

`outfile_prefix`: 出力ファイル名の先頭部。(イ)

`timezone`: タイムゾーンの名称。(ウ)

(ア) カレンダー処理には K.Tsunoda による `jholiday.py` を利用しています。

(イ) 出力ファイル名は以下ようになります。

```
[outfile_prefix]_[変数名]_[dmean,dmax,dmin,8hr,diurnal_[0,1,2]].csv
```

(ウ) 使用可能なタイムゾーン名は以下のコマンドで表示される。

```
$ python
>>> import pytz
>>> print(pytz.common_timezones)
```

3.5. jstream_stats_grid_data

`jstream_get_grid_data` の出力ファイルに対し、日平均 (`dmean`)、日最高値 (`dmax`)、日最低値 (`dmin`)、8 時間値 (`8hr`)、時刻平均値 (`diurnal`) を計算して `netCDF` ファイルを出力します。

日の区切りは、指定のタイムゾーンで行われます。

出力ファイルの `TFLAG` は以下のように定義されます。時間ラベルは UTC です。CTM 出力の慣例に従い、時間データは後 1 時間平均を表します (例: 0 時データ=0:00-1:00 の平均)。

日統計値 (`dmean`, `dmax`, `dmin`) `TFLAG = 2013203, 150000` の場合
 `20130723 00,01,02,...,23LST` の統計値
 (タイムゾーンが `Asia/Tokyo` の場合は、`20130722 15UTC,...,20130723 14UTC`)
8 時間値 (`8hr`) `TFLAG = 2013203, 70000` の場合
 `20130722 09,10,11,...,16LST` の平均値
 (タイムゾーンが `Asia/Tokyo` の場合は、`20130722 00,01,...,07UTC`)
時刻平均値 (`diurnal_?`) `TFLAG = 2013203, 150000` の場合
 指定期間の `00LST (0:00LST - 1:00LST)` の平均値
 (タイムゾーンが `Asia/Tokyo` の場合は、`15UTC (15:00UTC -- 16:00UTC)`)

パラメータファイル (デフォルト名 `sgd_params.txt`) の説明

`basedir`: `jstream_get_grid_data` の出力ファイルが置かれている `directory`
`infile`: `jstream_get_grid_data` の出力ファイル名。(ア)
`stats`: 統計処理の種類 `dmean`, `dmax`, `dmin`, `8hr`, `diurnal` から選択 (複数可)
`day_type`: 時刻平均値を計算する日のタイプ。現バージョンでは以下の 3 種類。
 0 - 全日、1 - 祝日を除く平日、2 - 土日祝
`timezone`: タイムゾーンの名称。3.3 (ウ)を参照。

(ア) 本コマンドの出力ファイル名は以下のようになります。

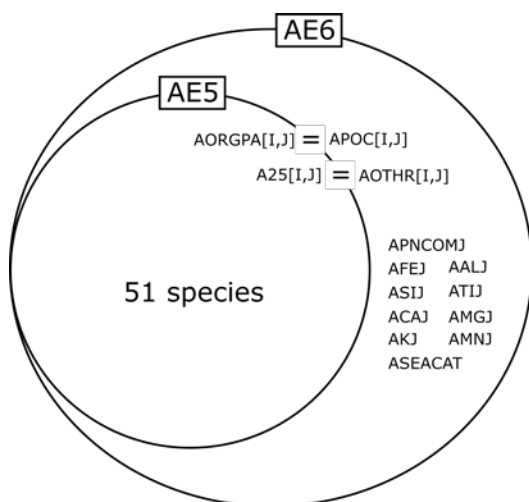
`[infile の basename]_[dmean,dmax,dmin,8hr,diurnal_[0,1,2]].nc`

4. 派生変数の Aerosol メカニズムの関係

ATOTJ や PM10 などの派生変数は `jstream_tools/spec_def_files/spec_def.csv` に定義されている。派生変数の定義式は、CMAQ 計算で使用した化学メカニズムに依存するが、CMAQ バージョン 5.0.2 以下では、`spec_def.csv` は共通である。

CMAQ の Aerosol メカニズムは大きく、Aero5 (AE5)系と Aero6 (AE6)系に分類される。が、CMAQ の

CMAQ バージョン	Aerosol メカニズム	spec_def.csv
4.7.1	AE4, AE5, AE5ca, AE5st	共通
5.0	AE5, AE6	共通
5.0.1	AE5, AE6	共通
5.0.2	AE5, AE5ca, AE6	共通
5.1	AE6, AE6i	メカニズム別
5.2	AE6, AE6nvPOA, AE6i, AE6invPOA	メカニズム別



CMAQ バージョン	spec_def.csv 内変数	CMAQ で使用したメカニズム	
		AE5	AE6
4.7.1	AORGPA[I,J], A25[I,J]	そのまま	—
5.0.1	APOC[I,J], AOTHR[I,J]	AORGPA -> APOC	そのまま
5.0.2		A25 -> AOTHR	

5. netCDF ファイルのバージョンについて

5.1. IOAPI と netCDF の関係

IOAPI は、netCDF に機能を追加したデータフォーマットで、CMAQ の入出力に採用されています。基本的に netCDF 用のツールで IOAPI ファイルも読むことができます。

ところで、IOAPI は、netCDF のインストール時に、`config` 実行において `--disable-netcdf-4` のオプションが指定されていることが前提になっています。これは、IOAPI が netCDF4 に固有の関数には対応していないからです。

netCDF >= v4.4.2 の問題

netCDF-Fortran の v4.4.2 以降では、関数 `CALL NC*()` が `IERR=NF_*()` に置換されています。IOAPI 3.2 では、この変更に対応していますが、IOAPI < 3.2 では対応していないため、netCDF >= 4.4.2 とはリンクすることができません。

そのため、以下が有効な組み合わせとなります。

netCDF-Fortran >= 4.4.2	IOAPI 3.2
netCDF-Fortran < 4.4.2	IOAPI 3.1

5.2. モデルユーザーごとの環境

CMAQ ユーザーは、必ず IOAPI をインストールしています。よって、netCDF 3 または netCDF 4 (`--disable-netcdf-4`) をインストールしているはずですが。

また、CAMx ユーザーは、CAMx 固有の出力フォーマットを `camx2ioapi` で IOAPI に変換することができます。そのため、IOAPI をインストールしており、CMAQ ユーザーと同じく netCDF 3 または netCDF 4 (`--disable-netcdf-4`) をインストールしている可能性が高いです。

一方、WRFChem ユーザーは、IOAPI を使う必要がないので、netCDF 3 または netCDF 4 (`--disable-netcdf-4`、または、フルの netCDF4) であると考えられます。

netCDF と IOAPI のバージョンとモデルユーザーの関係は下表のように整理できます。

バージョン間の関係

netCDF	IOAPI	想定ユーザ	備考
3.*	<= 3.1	CMAQ	
< 4.4.2 <code>--disable-netcdf-4</code>	<= 3.1	CAMx	
>= 4.4.2 <code>--disable-netcdf-4</code>	3.2	WRFChem	IOAPI <= 3.1 では、netCDF 4.4.2 で導入された関数にリンクできない
4.* (default)	-	WRFChem	IOAPI は、netCDF4 固有の関数には対応していない

5.3. jstream_tools における対応

IOAPI は汎用性が低く、netCDF のバージョンアップへの対応も遅いので、jstream_tools の出力フォーマットは IOAPI ではなく netCDF とします。また、出力ファイルの netCDF のバージョンは 3 とします。

なお、入力ファイルは、netCDF3, netCDF4 (--disable-netcdf-4), netCDF4 (full)のいずれにも対応しています。

6. jstream_tools が依存する第三者によるプログラム

jstream_tools は、python 本体および標準的なパッケージの他に以下の第三者によるプログラムに依存しています。

目的	ファイル名	パッケージ名	著作権者
地理座標 変換	gctp.f	general cartographic transformation package (version 2.0.2)	U.S. Geological Survey
数式解読	parser.py	mathematical expressions parser	Louis Fischer
祝日判定	jholiday.py		K. Tsunoda