Time scale algorithm
~ *basics to applications* ~
* Training *

Yuko Hanado（ Retired NICT, Japan）

Bureau International des Poids et Mesures

BIPM Summer School: “Time scale algorithms"
September 11, 2025

Purpose of this training:

- To learn how to use the Python program "*TAgen_pred.py*"
  - ※ *TAgen_pred.py* is a simulation tool for TAp [*Ref. 2*].
  - ※ TAp is "*TA with prediction*", which is an advanced and practical TA.
- To get familiar with "*TAgen_pred.py*" through an exercise
- To understand how TAp changes depending on calculation conditions

※ For exercise, you need to install Python on your computer and download the necessary materials. *(pp.10-13.)*

Contents

Contents

1. Overview

■ What shall we do in this training?

[1] Calculate TAp using the actual clock data.

Refer to *[Ref. 2]*

1) TAp is the accumulation of prediction errors.

$$TAp(t_1) \equiv \sum_i w_i(t_1) \cdot \{h_i(t_1) - \hat{x}_i(t_1)\} \quad \text{Here} \sum_i w_i(t_1) = 1$$

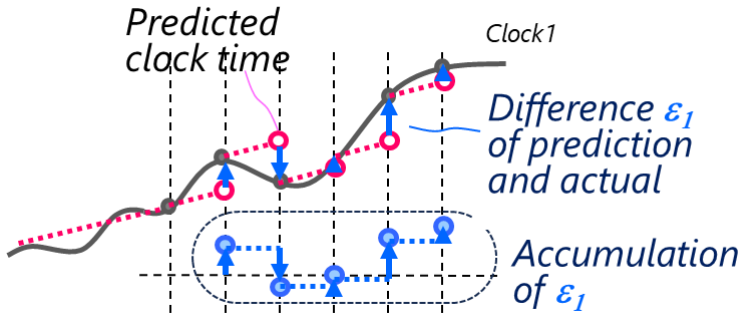
Eq. (3)

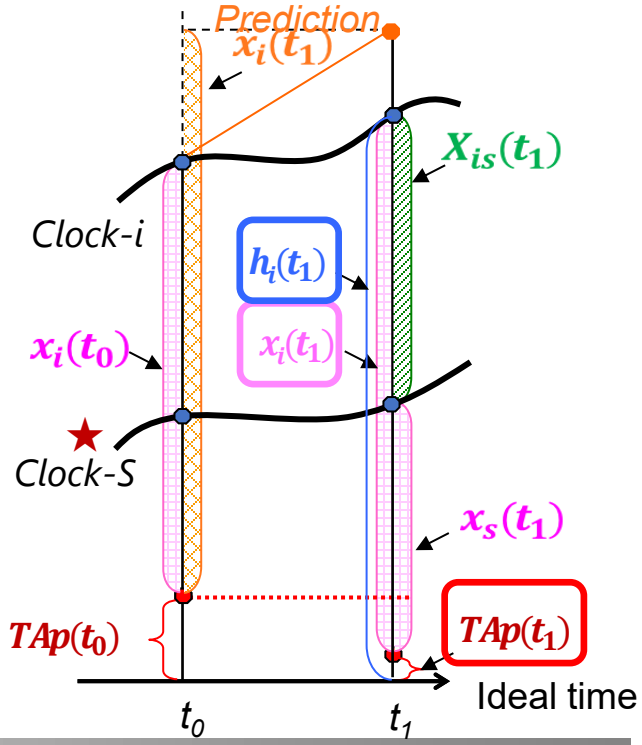
2) TAp can be defined as the time shifted by x_i from the time of $clock_i$.

$$x_i(t) \equiv h_i(t) - TAp(t) \Rightarrow TAp(t) = h_i(t) - x_i(t)$$

Eq. (5)

"Calculate TAp" means "Calculate x" !





Bureau International des Poids et Mesures

1. Overview

■ What shall we do in this training? (Cont.)

[1] Calculate *TAp* using the actual clock data. (Cont.)

Refer to *[Ref. 2]*

3) The algorithm of *TAp* is the procedure that computes the time series of $x(t)$.

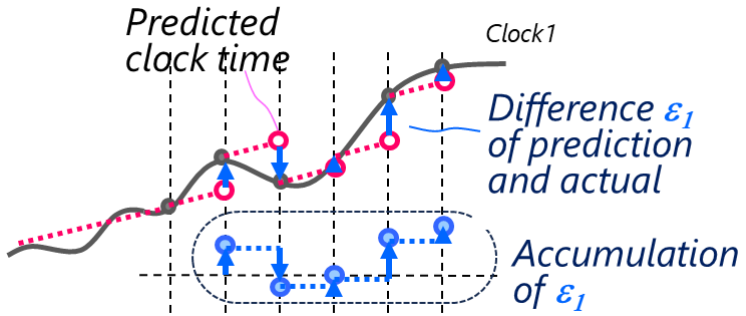
$$\hat{x}_i(t_k) = x_i(t_{k-1}) + \hat{y}_i(t_{k-1}) \cdot (t_k - t_{k-1}) \quad \text{Eq. (10)}$$

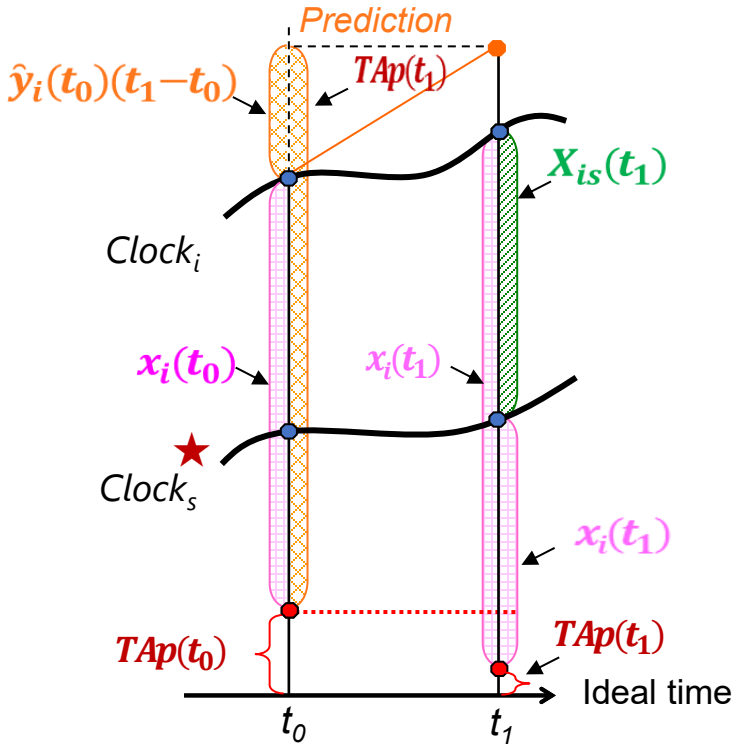
$$x_s(t_k) = \sum_i w_i(t_k)\{\hat{x}_i(t_k) - X_{is}(t_k)\} \quad \text{Eq. (9)}$$

$$x_i(t_k) = x_s(t_k) + X_{is}(t_k) \quad \text{Eq. (11)}$$

$$\hat{y}_i(t_k) = \frac{x_i(t_k) - x_i(t_k - Trate)}{T_{rate}} \quad \text{Eq. (12)}$$

:Measured :Predicted :Output
:Calculated from previous x_i





1. Overview

■ What shall we do in this training? (Cont.)

[2] Calculate "$UTC(k) - TAp$" for evaluation, and
Compare its quality (stability, continuity) with "$UTC(k) - clock_i$".

Refer to [*Ref. 2*]

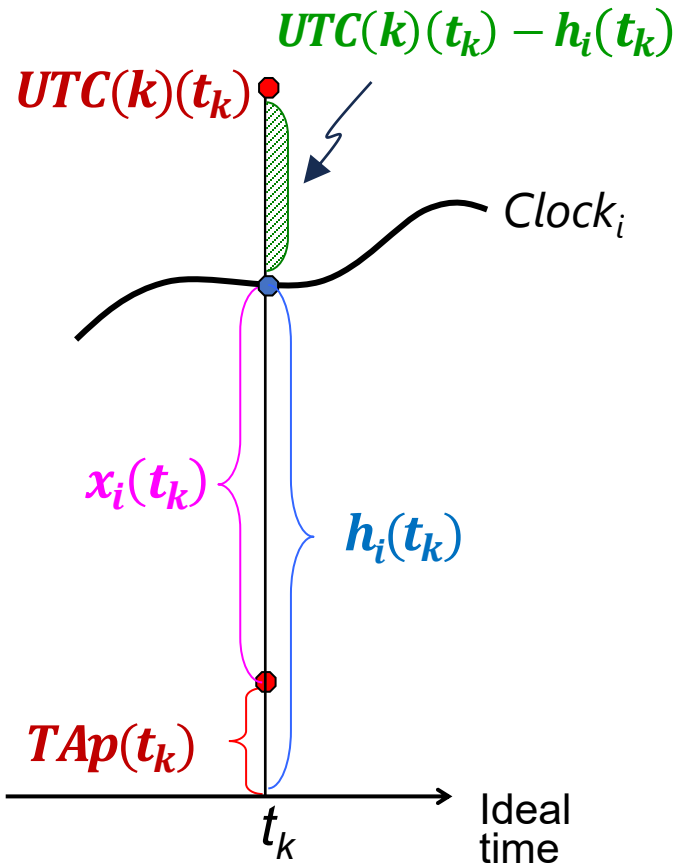
$$UTC(k)(t_k) - TAp(t_k)$$
$$= [UTC(k)(t_k) - h_i(t_k)] + [h_i(t_k) - TAp(t_k)]$$

Eq. (13)

Given by measurement

Calculated x_i

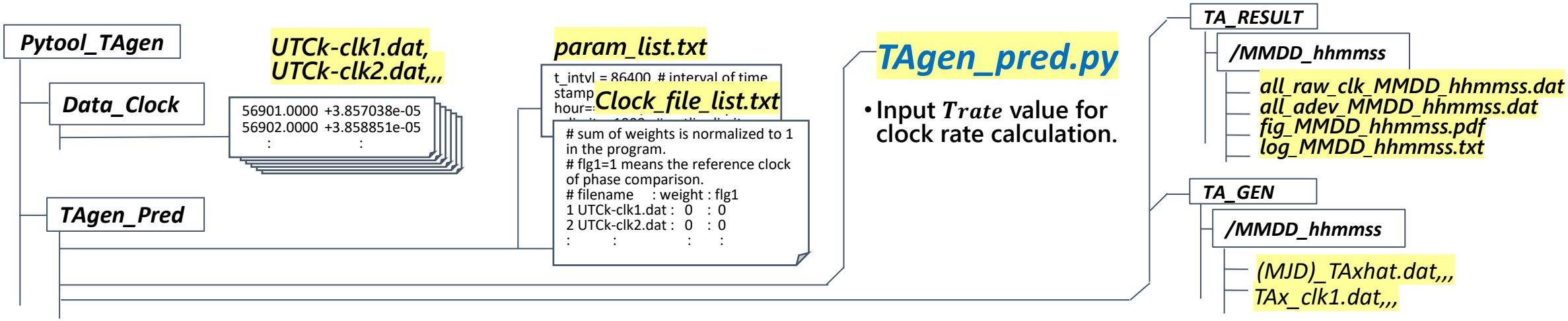
This is to be done with *TAgen_pred.py* !



1. Overview

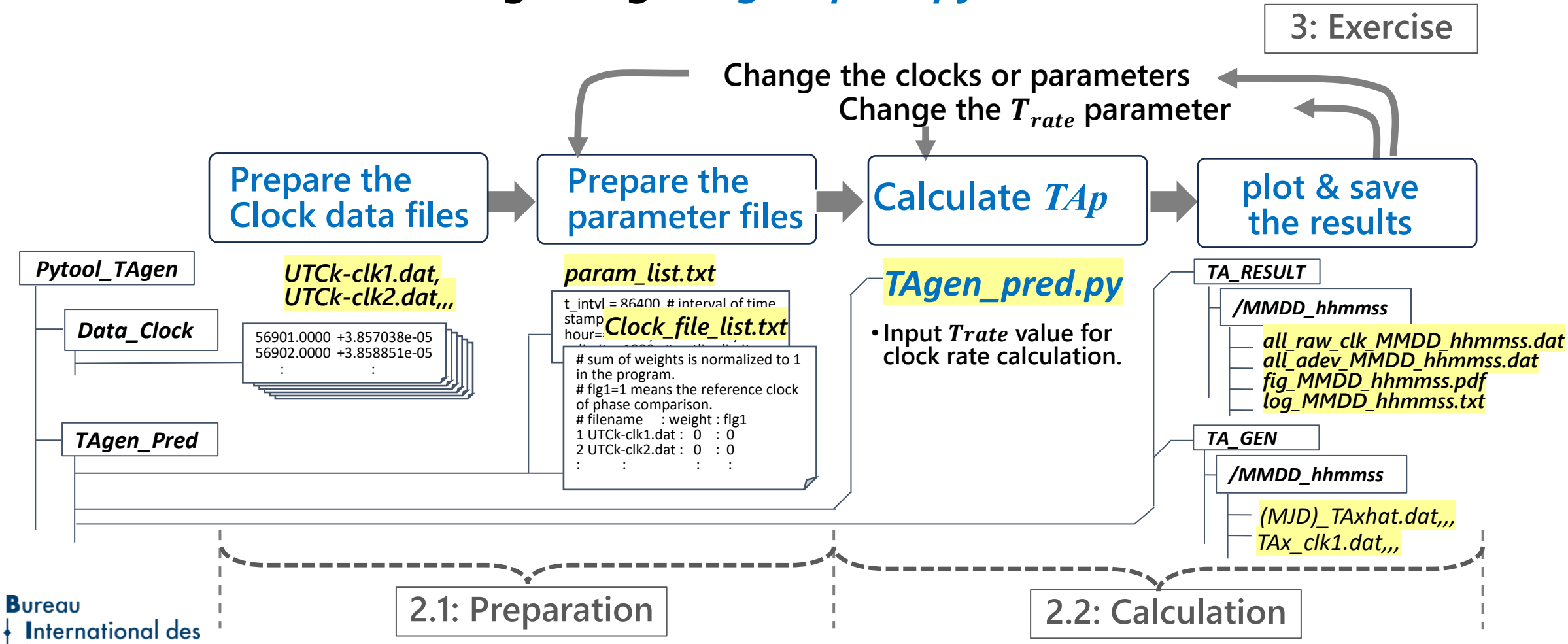
■ What is *TAgen_pred.py?*

- ***TAgen_pred.py*** is a simulation tool used for making TAp which is the "TA with prediction".
- It reads the clock data "***UTC(k)-clock$_i$***"s, calculates TAp using the given starting weights, and displays the results of the "***UTC(k)-TAp***" and its Allan deviation (*ADEV*).
- You can find further information about the principle of the algorithm in the tutorial lecture *[Ref. 2]* and about the details of the program in the user's manual *[Ref. 3]*.



1. Overview

■ Process of the training using "*TAgen_pred.py*"



Contents

2.1 Preparation

[1] Install *Python* in your computer

- Set the path to the working directory.
- Following Python modules should be installed in advance;

  - *numpy*
  - *matplotlib*
  - *allantools*

For example, from the command prompt;

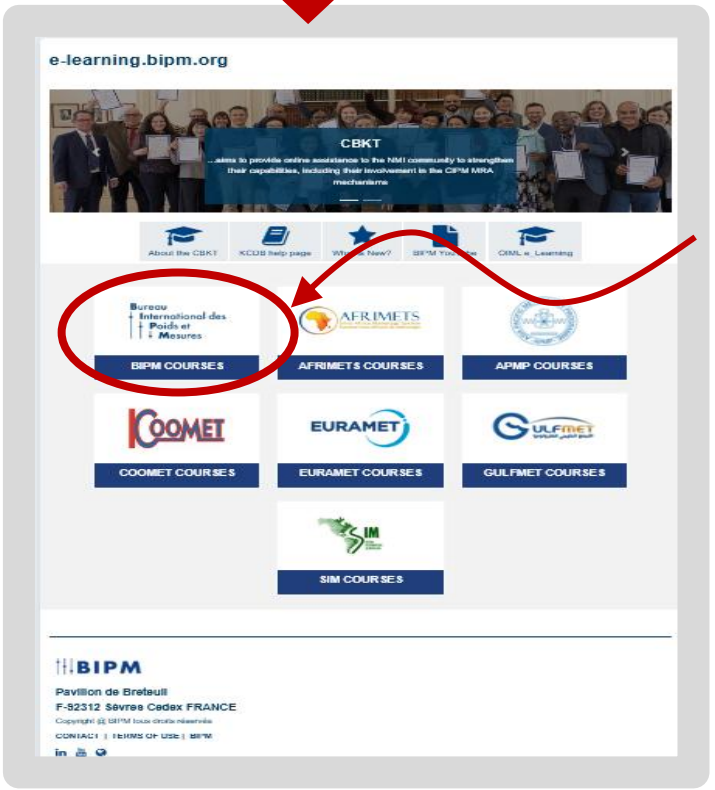
```
C:\Mywork>python -m pip install numpy
```

※ ***"Spyder"*** **is not necessary for running the tools, but is useful for debugging and modifying them.**

https://www.spyder-ide.org/

[2] Download the files

(1) **Go to the "*BIPM e-learning platform*".**
https://e-learning.bipm.org/

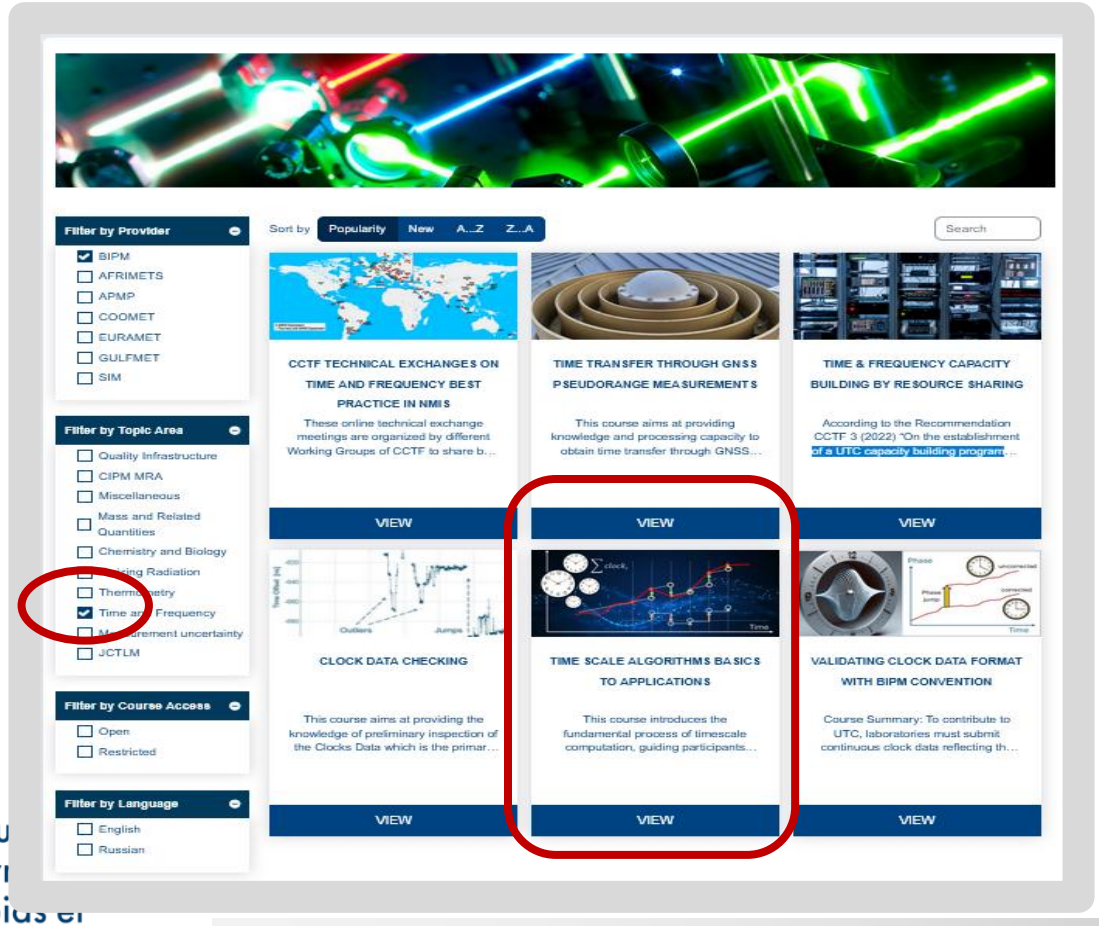


(2) **Select "*BIPM*"**

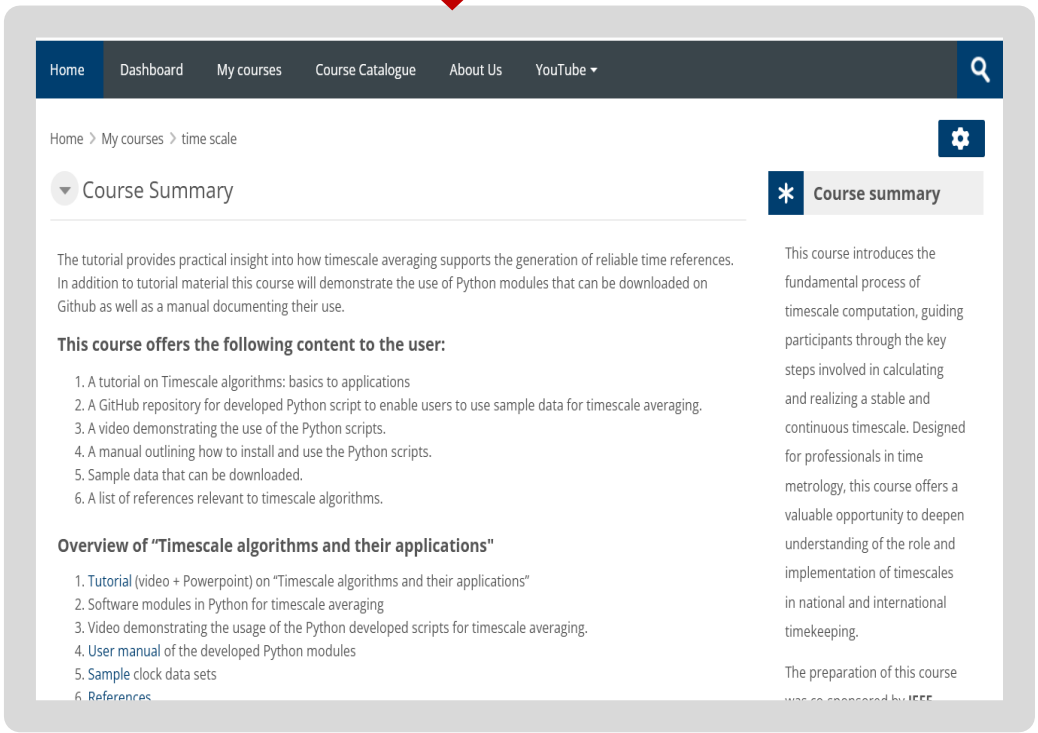
2.1 Preparation

[2] Download the files

(3) Select "*Time scale algorithms*" *[Ref. 1]*



(4) Make your account and log in the course.

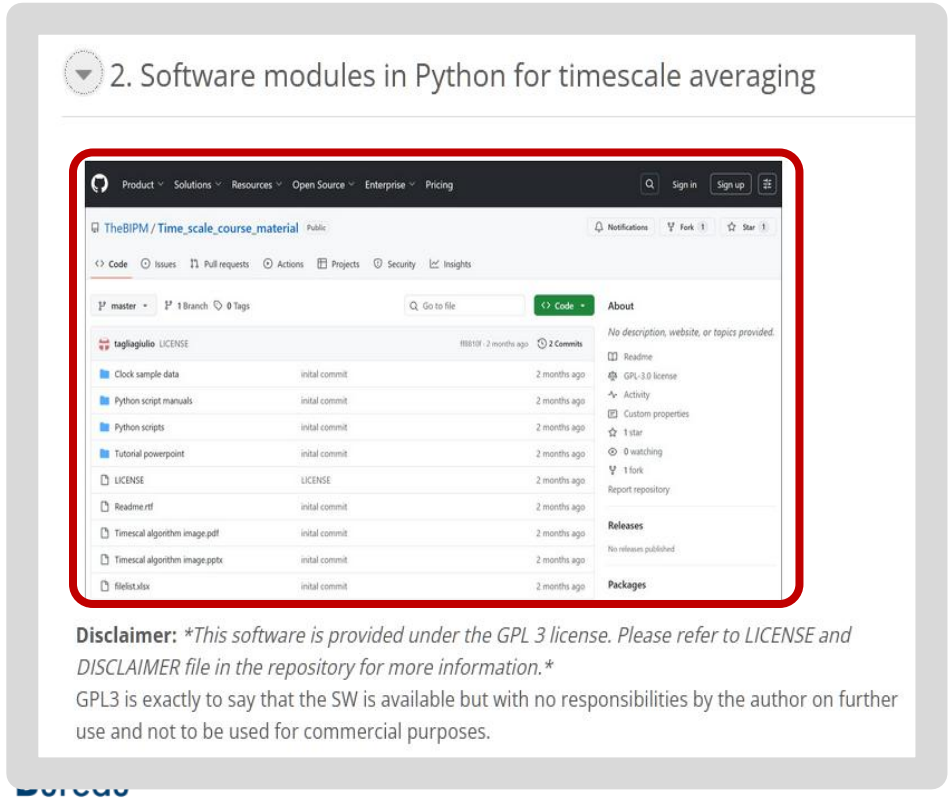


https://e-learning.bipm.org/course/view.php?id=96

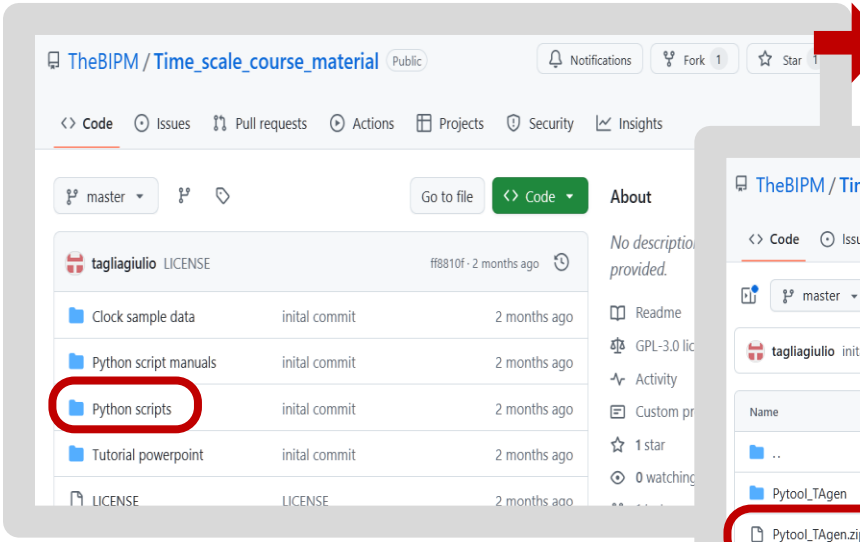
2.1 Preparation

[2] Download the files

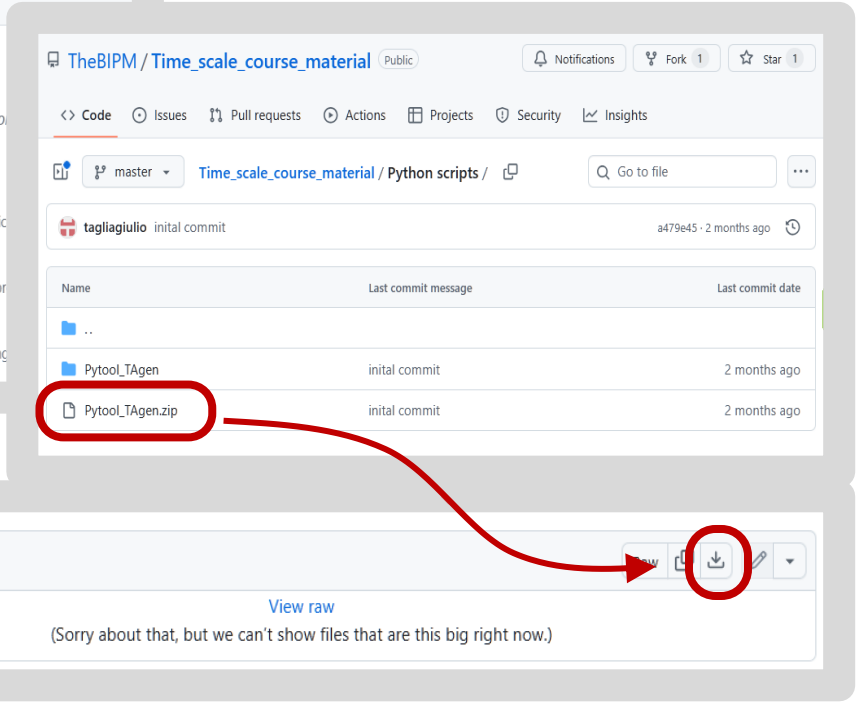
(5) Open "*2. Software modules in Python for timescale averaging*".



(6) Open "*Python Scripts*".



(7) Get "*Pytool_TAgen.zip*" and unzip it on your PC.



2.1 Preparation

"BIPM UTC Summer School"

/committees/cb/cbkt/cbkt-tf-utc-2025 - BIPM

	Dinner and networking (Optional dinner paid by participants)	

Sample Data

The school will include hands-on sessions where students can use their own lab data, or sample data, to follow in guided demonstrations of web-based applications.

Sample for these sessions can be found here:

01 "Validating clock data against BIPM convention" session
02 "Processing of CGGTTS data" session
03 "Web based GNSS calibration code" session
04 "Clock data checking" and "Time scale algorithms" sessions

Python scripts and manuals

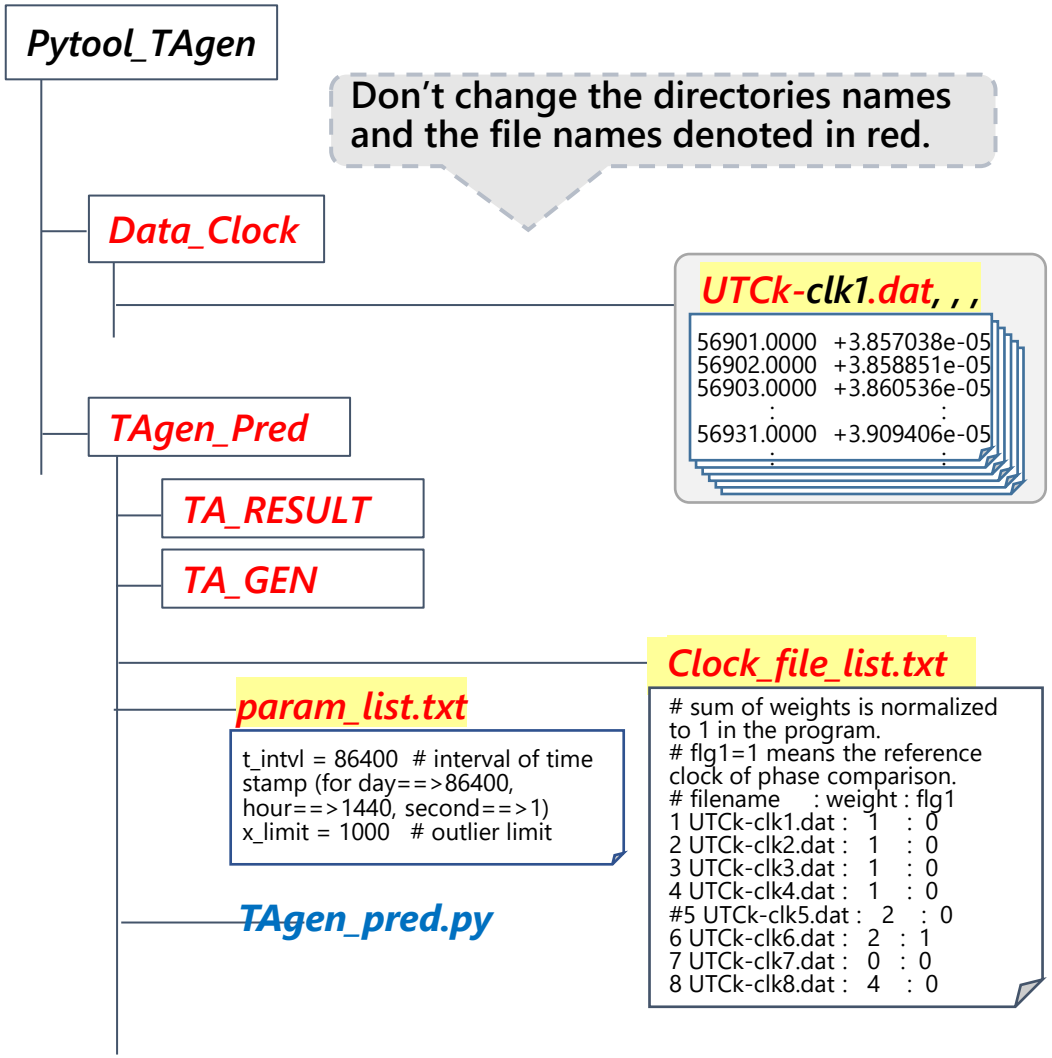
- Pytool_Tagen.zip
- manual_TA_pred
- manual_TA_auto_

2.1 Preparation

[3] Check the directories and files

- **Clock measurement data :**
  *"/Pytool_TAgen/Data_Clock/**UTCk-****.dat**"* *(p.14, 47)*
- **Clock-data name list file :**
  *"/Pytool_TAgen/TAgen_Pred/**clock_file_list.txt**"* *(p.15, 45)*
- **Parameter list file :**
  *"/Pytool_TAgen/TAgen_Pred/**param_list.txt**"(p.16, 46)*

※ ***Details of each file are provided in Appendix-1.***



Check time

Appendix-1 : A-1.2.3 "UTCk_(clock$_i$).dat"

■ "*/Pytool_TAgen/Data_Clock/**UTCk-(clock$_i$).dat***"

Outline

- "***UTCk-(clock$_i$).dat***" are the measurement clock data as the materials of ***TAp*** calculation.
- This data files should be prepared in advance.

Details / Rules of use

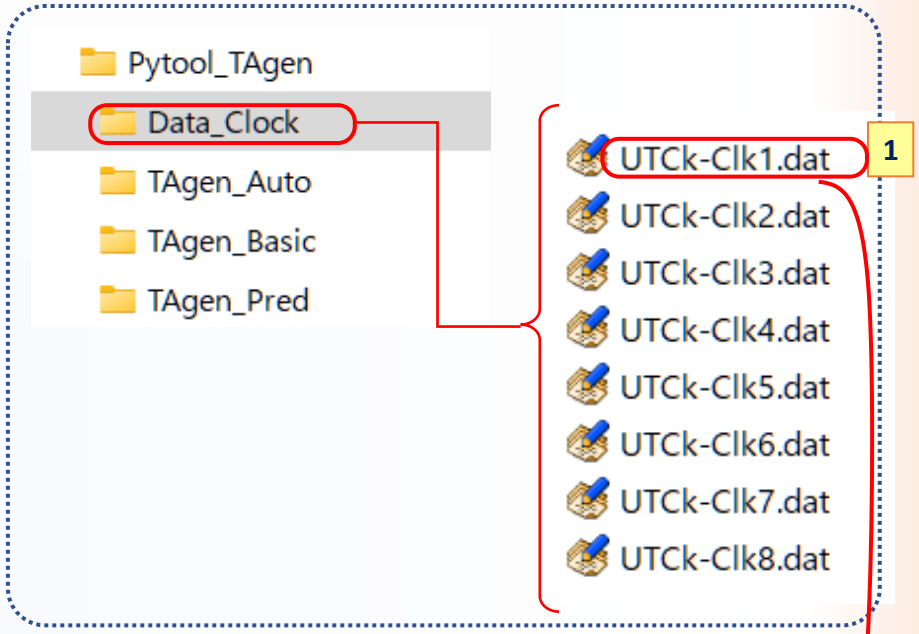
- One file contains data from a single clock.
- The file name should be "***UTCk-****.dat***", where the four characters "****" identify each clock. 1
- 1st column : observation time stamp 2
- 2nd column : measured time difference of "***UTC(k) – clock$_i$***" (in second) 3

Note

※ The Interval of the 1st column should be specified as ***t_intvl*** in "***param_list.txt***". *(p.16, 46)* 4

※ This sampling interval must be constant and continuous.

※ Sufficient data is required for each clock to start the calculation. *(p.24, 63)*



Appendix-1 : A-1.2.1 "clock_fie_list.txt"

■ "/Pytool_TAgen/TAgen_Pred/**clock_file_list.txt**"

Outline

- "***clock_file_list.txt***" provides the conditions of each clock in the averaging process.
- This table should be prepared in advance.

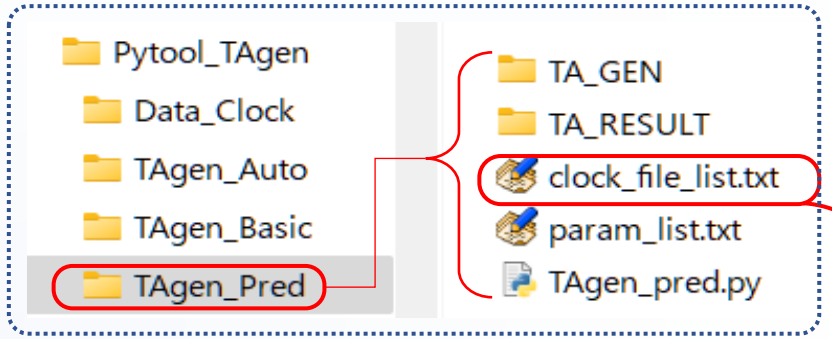
Details / Rules of use

- The line starting with "#" is ignored. 1
- 1st column : file No. (defines plot color)
- 2nd column : data file name
- 3rd column : clock weight*1
- 4th column : flag of the reference clock *2.

Note

*1 : Each weight is normalized in the program. 2

*2 : The clock with *"flg1=1"*, the reference clock named as ***Refclk*** hereafter, is used as the base clock to get the time difference between clocks. 3




```
# sum of weights is normalized to 1 in the program.
# flg1=1 means the reference clock of phase comparison.
# filename     : weight : flg1
1 UTCk-clk1.dat :  1  : 0
2 UTCk-clk2.dat :  1  : 0
3 UTCk-clk3.dat :  1  : 0
4 UTCk-clk4.dat :  1  : 0
#5 UTCk-clk5.dat :  2  : 0
6 UTCk-clk6.dat :  2  : 1
7 UTCk-clk7.dat :  0  : 0
8 UTCk-clk8.dat :  4  : 0
```

1 • *"clk5"* is ignored by "#".

2 • ***Clk5** is ignored from all the calculation*
• ***clk7** is excluded from the averaging*

3 • *"clk6"* is to be the *"Refclk"* (reference clock).

Appendix-1 : A-1.2.2 "param_list.txt"

■ *"/Pytool_TAgen/TAgen_Pred/**param_list.txt**"*

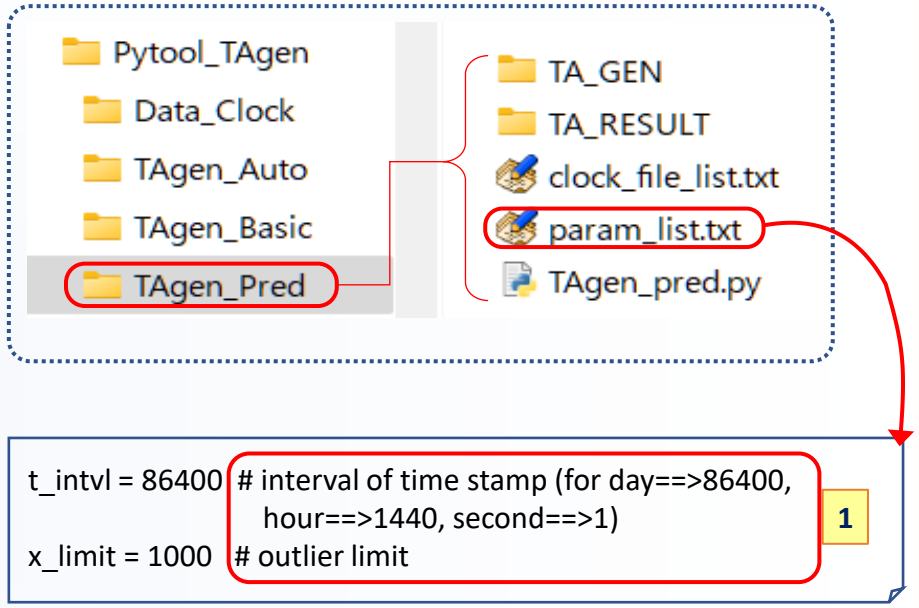
Outline

- *"**param_list.txt**"* provides the parameters for the ***TAp*** calculation.
- Parameters that are fixed within the program but can be changed on demand are stored here.
- This table should be prepared in advance.

Details / Rules of use

- These parameters are used with the same names in the program.
- The text after "#" provides the explanation of the parameter. 1

- ***t_intvl*** : interval of time stamp
  (for day → *86400*, hour → *1440*, second → *1*)
- ***x_limit*** : outlier limit



Contents

2.2 Calculation : [1] First, run the program

[1] Run "/TAgen_Basic/TAgen_pred.py"

※ Case (1) : via "Spyder"

1. Select the following Python program from the menu of "***File*** --> ***open*** -> ***/Pytool_TAgen/TAgen_Basic/TAgen_pred.py***". 1
2. Click "▶" or "***Run***". 2

※ The result is shown in the "*Console*" window. 3

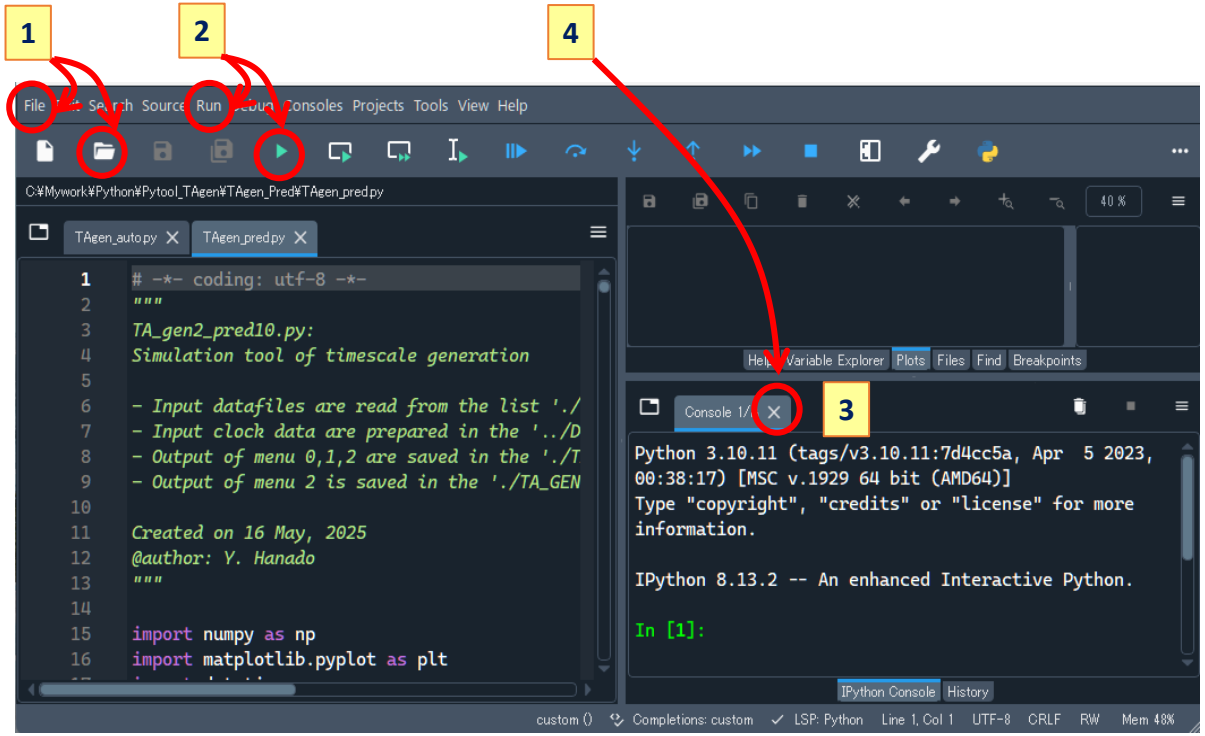
※ You can exit the program by clicking "***X***" on the console window. 4

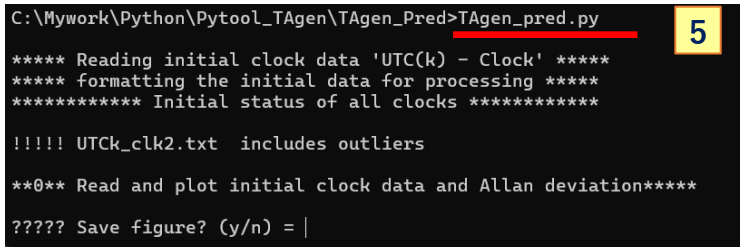
※ Case (2) : via "Command Prompt"

1. Go to the directory in advance including the "***Pytool_TAgen/TAgen_Pred/***" in advance.
2. Run "***TAgen_pred.py***" --> You can proceed with the program in the command prompt console. 5

※ The graph is shown in another window.

※ You can exit the program by "***Ctrl+Z***".





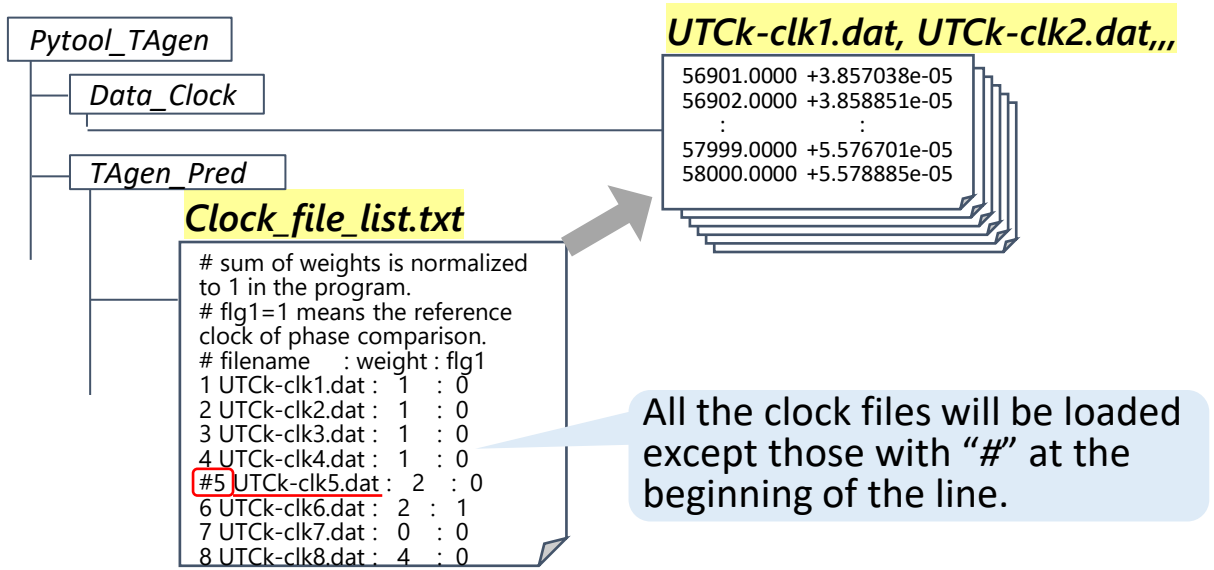
2.2 Calculation : [1] First, run the program

■ Initial processing is performed first

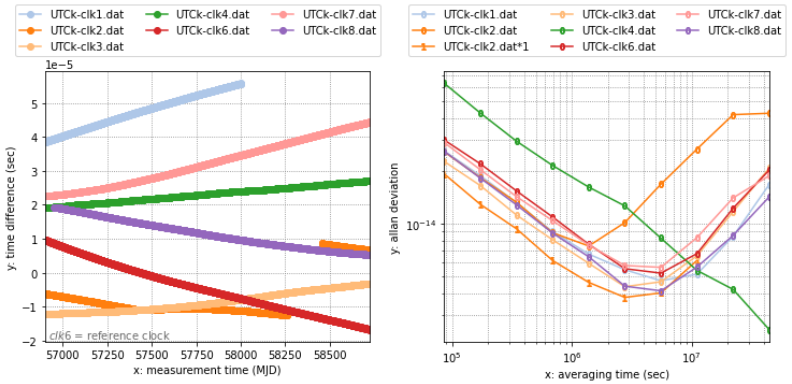
(1) Read "*param_list.txt*" and "*clock_file_list.txt*" *(p.45, 46)*

(2) Read the assigned "*UTCk_(clock$_i$).dat*" *(p.47)*

- Alert of outlier : outlier limit is set in the "***param_list.txt***". 1




```
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17)
[MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.13.2 -- An enhanced Interactive Python.

runfile('C:/Mywork/Python/Pytool_TAgen/TAgen_Pred/TAgen_pred.
py', wdir='C:/Mywork/Python/Pytool_TAgen/TAgen_Pred')

..... Reading initial data of 'UTC(k) - Clock'
..... formatting the initial data for processing

************ Initial status of all clocks ************

!!!!! UTCk-clk2.dat  includes outliers 1
```




```
**0** Read and plot initial clock data and Allan deviation*****

????? Save figure? (y/n) =
```

Bureau International des Poids et Mesures

2.2 Calculation : [1] First, run the program

■ Initial processing (cont.)

(3) Plot all the raw data of "*UTC(k) – clock$_i$*" *(p.57)*

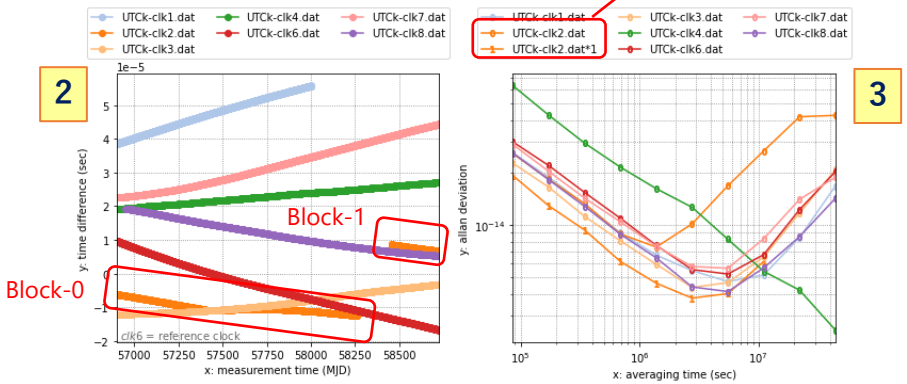
- **Left fig. : Time difference of "*UTC(k) – clock$_i$*".** 2
  - *MJD*s are adopted for the maximum range of all data.
- **Right fig. : *ADEV* of "*UTC(k) – clock$_i$*".** 3
  - *ADEV* is calculated by "***AllanTools***" [*Ref. 4*].
  - If the data has discontinuous and divided into several blocks, the *ADEV* is plotted for each block and marked with *"*1", "*2",,,* .

  *("**UTCk_clk2.dat**" shows the ADEV of [block-0], and "**UTCk_clk2.dat*1**" shows the ADEV of [block-1].)* 4

The purpose of this plotting is to confirm the condition of each clock data.

```
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17)
[MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.13.2 -- An enhanced Interactive Python.

runfile('C:/Mywork/Python/Pytool_TAgen/TAgen_Pred/TAgen_pred.
py', wdir='C:/Mywork/Python/Pytool_TAgen/TAgen_Pred')

..... Reading initial data of 'UTC(k) - Clock'
..... formatting the initial data for processing

************ Initial status of all clocks ************

!!!!! UTCk-clk2.dat  includes outliers
```

4




```
**0** Read and plot initial clock data and Allan deviation*****

????? Save figure? (y/n) =
```

Bureau International des Poids et Mesures

2.2 Calculation : [1] First, run the program

■ Initial processing (cont.)

(4) Save the results, if required *(p.52, 54, 56, 57)*

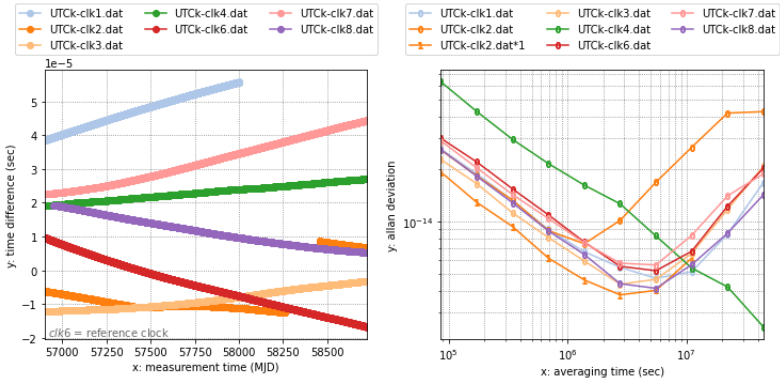
- If input *"y"* for "*Save figure (y/n) = "* , 5
  four results are saved under the following directory
  "./TA_RESULT/(time)/" with a time stamp of data saving.

  - ***Log_(time).txt*** : **history of the process**
  - ***fig _(time).pdf*** : **both figures**
  - ***all_raw_clk_(time).dat***: ***"UTC(k)-clock$_i$"*** **(left fig)**
  - ***all_adev_(time).dat*** : ***ADEV* of *"UTC(k)-clock$_i$"*** **(right fig)**

※ ***(time)*** is "***MMDD_hhmmss***" of operation time-stamp

- ***MMDD*** : month and date
- ***hhmmss*** : hour, minute and second

Check time



0 Read and plot initial clock data and Allan deviation*****

????? Save figure? (y/n) = y 5

::::: save logfile = ./TA_RESULT/0717_093343/log_0717_093343.txt
::::: save figfile = ./TA_RESULT/0717_093343/fig_0717_093343.pdf
::::: save alldata = ./TA_RESULT/0717_093343/all_raw_clk_0717_093343.dat
::::: save ADEVdat = ./TA_RESULT/0717_093343/all_adev_0717_093343.dat

==========
Menu of the data processing
---
0. Plot the "UTC(k)-Clock" data
1. Assign the MJD range
2. Make TA with daily prediction
and fixed clock weights
10. Exit
==========
????? Input menu number ? >>

2.2 Calculation : [2] Select the processing menu

[2] Select the processing menu

0. *"Plot the "UTC(k)-Clock" data"*

- Read all the data assigned in "***clock_file_list.txt***", and plot the time differences vs ***UTC(k)*** and their *ADEV*s.
- **This operation initializes all the processing to date.**

1. *"Assign the MJD range"*

- The specified range will be used for all subsequent processing unless "*menu-0*" is selected.

2. *"Make TA with daily prediction and fixed clock weights"* *1

1

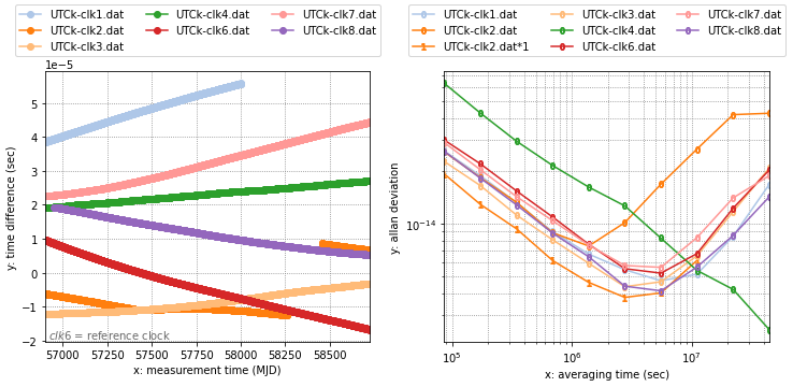
- Calculate the ***TAp*** using daily prediction.
- The clock weights are fixed depending on the values*2 given in "***clock_file_list.txt***".

10. *"Exit"*

- Exit the program.

*1 : *See **TAp** in [Ref. 2].*

*2 : *Normalized in the program.*



0 Read and plot initial clock data and Allan deviation*****

????? Save figure? (y/n) = y

::::: save logfile = ./TA_RESULT/0717_093343/log_0717_093343.txt
::::: save figfile = ./TA_RESULT/0717_093343/fig_0717_093343.pdf
::::: save alldata = ./TA_RESULT/0717_093343/all_raw_clk_0717_093343.dat
::::: save ADEVdat = ./TA_RESULT/0717_093343/all_adev_0717_093343.dat

```
=======================================
    Menu of the data processing
---------------------------------------
 0.  Plot the "UTC(k)-Clock" data
 1.  Assign the MJD range
 2.  Make TA with daily prediction
         and fixed clock weights
 10.  Exit
=======================================
????? Input menu number ? >>  2
```

1

Bureau International des Poids et Mesures

2.2 Calculation : [3] Select "*menu-2*"

[3] Select "*menu-2*" 1

➡ Calculate *TAp* and plot "*UTC(k)-TAp*"

■ Outline of the process

(1) Preparation

- Read the parameter tables ➔ [*1][*2]
- Read the "***UTC(k)-clock$_i$***" data ➔ [*3]
- Prepare initial parameters ➔ [*4][*6]
- Prepare initial data set ➔ [*5][*7]

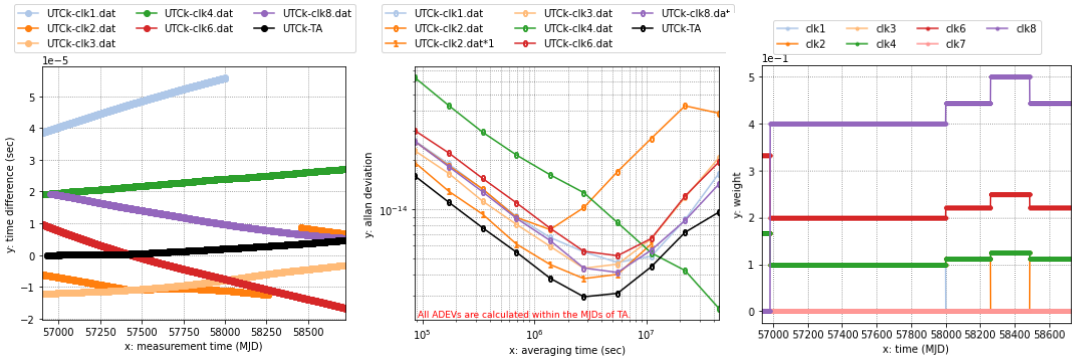
(2) Calculation of daily *TAp*

- Iterative daily calculation ➔ [*8]
  - Set the normalized clock weights
  - Calculate the prediction $\hat{x}_i$
  - Calculated the time series of x_i

(3) Plot & save the results

- Calculate "***UTC(k)-TAp***" ➔ [*9]
- Plot and save "***UTC(k)-TAp***" and its *ADEV* ➔ [*10][*11]

```
=======================================
    Menu of the data processing
---------------------------------------
 0.  Plot the "UTC(k)-Clock" data
 1.  Assign the MJD range
 2.  Make TA with daily prediction
        and fixed clock weights
10.  Exit
=======================================
????? Input menu number ? >>  2   1

???? Span in days (try 10 or 30 or 90) =  30

***** Calculating a timescale for  56932
                    :
                    :
***** Calculating a timescale for  58726
```




```
**2** make a timescale with prediction *****

????? Save figure? (y/n) = y

::::: save logfile =  ./TA_RESULT/0731_155524/log_0731_155524.txt
::::: save figfile =  ./TA_RESULT/0731_155524/fig_0731_155524.pdf
::::: save alldata =  ./TA_RESULT/0731_155524/all_raw_clk_TA_0731_155524.dat
::::: save ADEVdat =  ./TA_RESULT/0731_155524/all_adev_0731_155524.dat
::::: save weights =  ./TA_RESULT/0731_155524/all_weight_0731_155524.dat
::::: save wgt_fig =  ./TA_RESULT/0731_155524/wgt_fig_0731_155524.pdf
```

2.2 Calculation : [3] Select "*menu-2*"

(1) Preparation

[*1] Read the parameter table ← *param_list.txt* (p.46)

[*2] Read the clock parameters ← *clock_file_list.txt* (p.45)

[*3] Read the assigned clock data ← *UTCk_(clock$_i$).dat* (p.47)

[*4] Fix the [$MJD1, MJD2$] range
- This range is automatically set as wide as possible.

[*5] Prepare X_{is} (=$clock_i$ - $Refclk$) → *all_clk-(Refclk).dat* (p.48)
- $Refclk$ is assigned in "clock_file_list.txt"

[*6] Set T_{rate} for $\hat{y}_i(t_k)$ ← T_{rate} *from console* 2 (p.63)

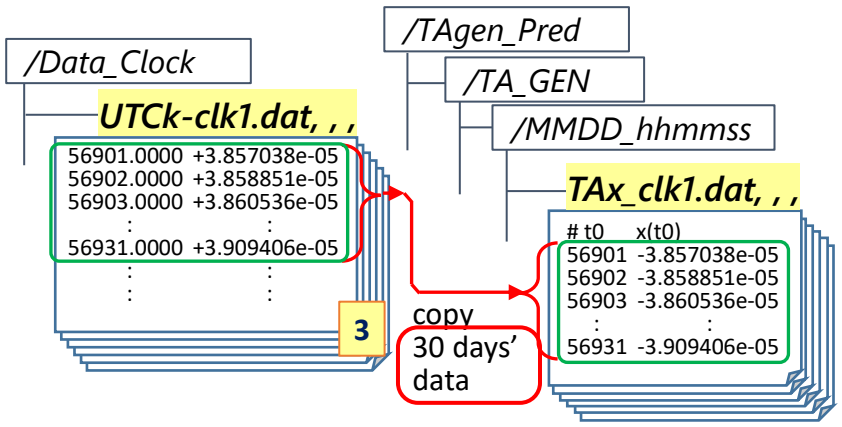
※ In the case of Cs-clocks, *30 days* will be suitable.

[*7] Copy "$clock_i$-*UTC(k)*"s as substitutes for the initial x_i to start the calculation → *TAx_(clock$_i$).dat* 3 (p.50)
- It's because no past x_i at the beginning.

$clock_i(t) - UTC(k)(t) \rightarrow clock_i(t) - TAp(t) \equiv x_i(t) \quad (t < start\ date)$

```
=======================================
    Menu of the data processing
---------------------------------------
 0.  Plot the "UTC(k)-Clock" data
 1.  Assign the MJD range
 2.  Make TA with daily prediction
        and fixed clock weights
10.  Exit
=======================================
????? Input menu number ? >> 2

!!!!! UTCk-clk2.dat  includes outliers

*2** Set the span for clock rate calculation.
??2?? Span in days (try 10 or 30 or 90) = 30
```
2

$$\hat{y}_i(t_k) = \frac{x_i(t_k) - x_i(t_k - Trate)}{T_{rate}}$$ Eq. (12) 2 (p.5)



2.2 Calculation : [3] Select "*menu-2*"

(2) Calculation of daily *TAp*

[*8] Calculate $x_i(t_k)$ for "$TA_0 \le t_k \le MJD2$"

- The *TAp* calculation start date is "$TA_0 = MJD1 + T_{rate} + 1$". 4

[*8.1] Set the clock weights

- The given initial weights in *"clock_file_list.txt"* are used after normalization.
- If the clock data is not normal (missing or outlier), its weight becomes zero for the date.

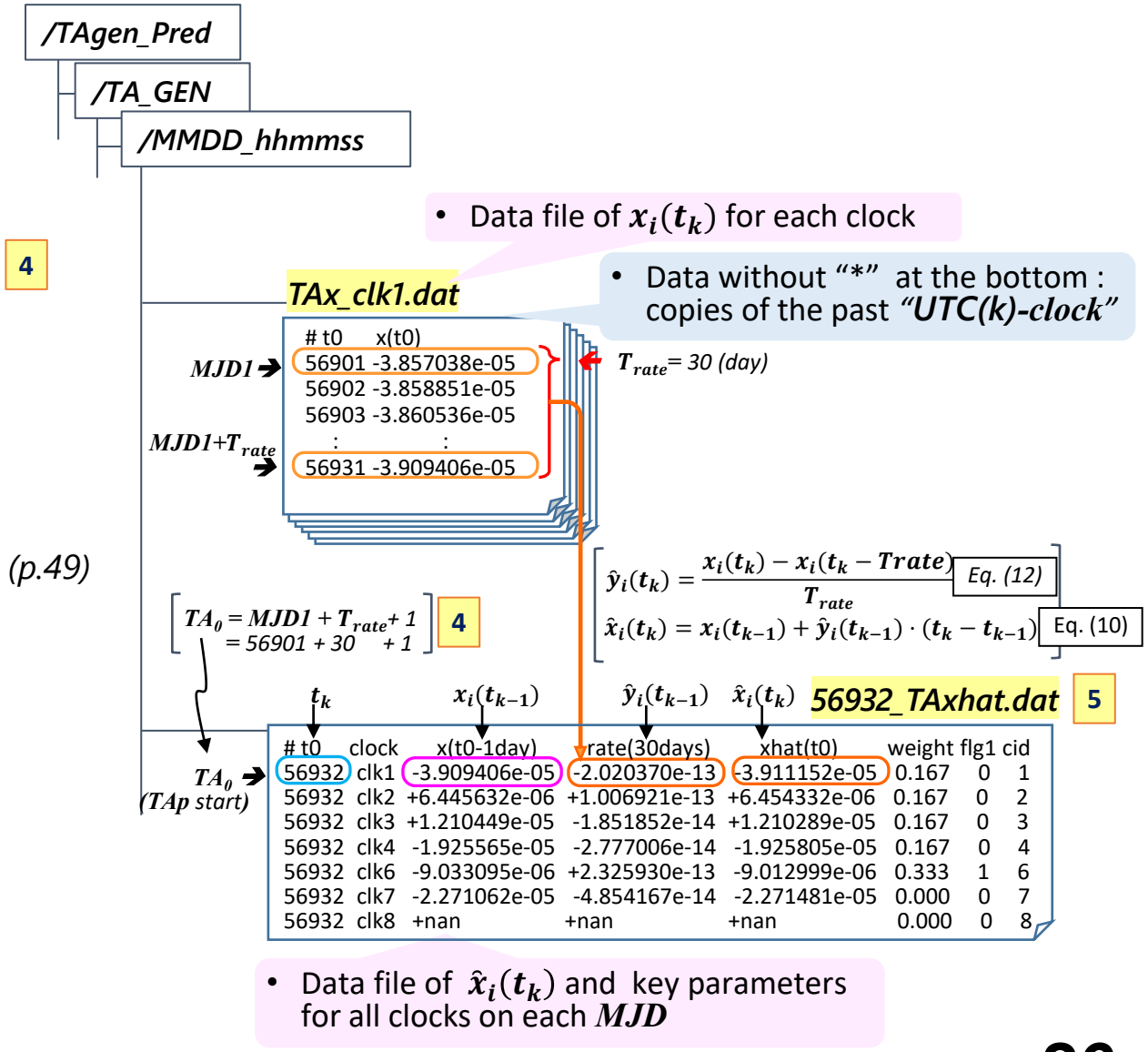
[*8.2] Calculate $\hat{x}_i(t_k)$

➔ *(MJD)_TAxhat.dat* 5 *(p.49)*

- $\hat{x}_i(t_k)$ is calculated by *Eq.(10)* using the past x_i.
- $\hat{y}_i(t_{k-1})$ is calculated by *Eq. (12)* as the slope of x_i over the latest past T_{rate} period.

$$\hat{x}_i(t_k) = x_i(t_{k-1}) + \hat{y}_i(t_{k-1}) \cdot (t_k - t_{k-1}) \quad \text{Eq. (10)} \quad (p.5)$$

$$\hat{y}_i(t_k) = \frac{x_i(t_k) - x_i(t_k - T_{rate})}{T_{rate}} \quad \text{Eq. (12)} \quad (p.5)$$



2.2 Calculation : [3] Select "*menu-2*"

(2) Calculation of daily TAp (cont.)

[*8] Calculate $x_i(t_k)$ for "$TA_0 \le t_k \le MJD2$" (Cont.)

[*8.3] Calculate the daily $x_i(t_k)$

- Calculate $x_i(t_k)$ by *Eq.(9)(11) (p.5).* 6

$$x_s(t_k) = \sum_i w_i(t_k)\{\hat{x}_i(t_k) - X_{is}(t_k)\} \quad \textit{Eq. (9)}$$

$$x_i(t_k) = x_s(t_k) + X_{is}(t_k) \quad \textit{Eq. (11)}$$

- The calculated $x_i(t_k)$ are accumulated in "***TAx_(clock$_i$).dat*** with the star-mark " * " at the end of the line.

➔ ***TAx_(clock$_i$).dat*** 7 *(p.50)*

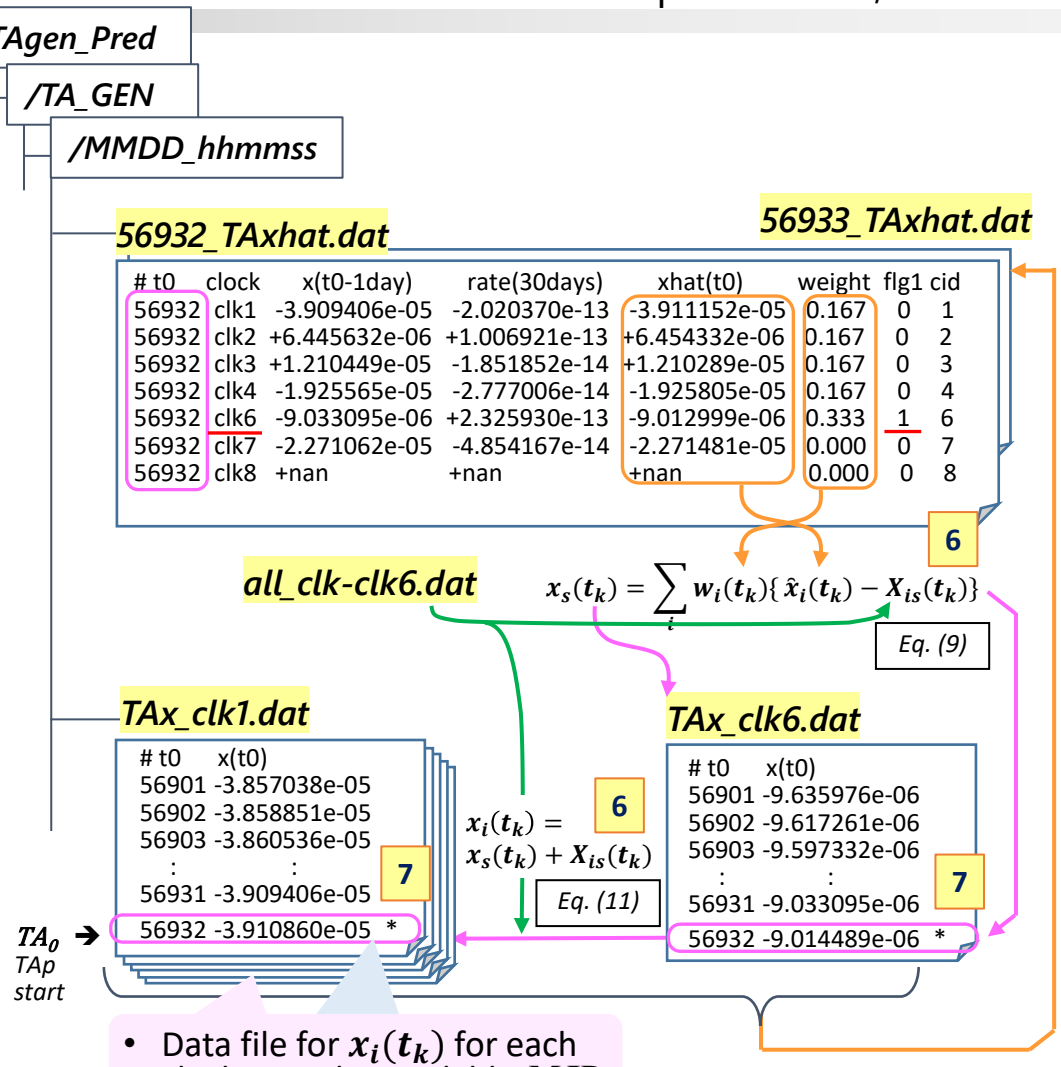
(3) Plotting and saving the results

[*9] Calculate "*UTC(k)-TAp*" for evaluation

- TAp can be compared with other timescales or clocks via common reference, for example ***UTC(k)***.

$$UTC(k) - TAp = [UTC(k) - h_i] + [h_i - TAp] \quad \textit{Eq. (13)} \quad \textit{(p.6)}$$

"*UTCk-(clock$_i$).dat*" *Calculated* x_i



- Data file for $x_i(t_k)$ for each clock over the available MJD
- Data without a start-mark "*" at the end of the line is the copy of the past *"UTC(k)-clock"*.

2.2 Calculation : [3] Select "*menu-2*"

(3) Plotting and saving the results (Cont.)

[*10] Plot the results 12

- Left : time difference vs ***UTC(k)*** *(p.20, 57)*
- Middle : *ADEV*s of left data *(p.20, 57)*
- Right : clock weights *(p.58)*

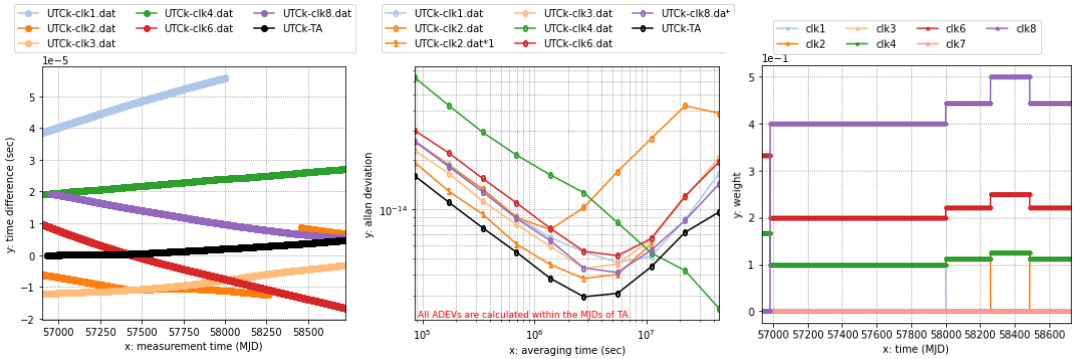
• Only the clocks with non-zero weights are plotted.

[*11] Save the results, if required *(p.21, pp.52-58))* 8

• Results are saved under "***./TA_RESULT/(time).***

- ***Log_(time).txt*** : history of the process
- ***fig _(time).pdf*** : all figures
- ***all_raw_clk_TA_(time).dat*** : "***UTC(k)-clock$_i$***" (left fig)
- ***all_adev_(time).dat*** : *ADEV* of "***UTC(k)-clock$_i$***" (right fig)
- ***all_weight_(time).dat*** : daily calculated weights
- ***wgt_fig_(time).dat*** : figure of all weights

```
=======================================
    Menu of the data processing
---------------------------------------
 0.  Plot the "UTC(k)-Clock" data
 1.  Assign the MJD range
 2.  Make TA with daily prediction
        and fixed clock weights
10.  Exit
=======================================
????? Input menu number ? >>  2

???? Span in days (try 10 or 30 or 90) =  30

***** Calculating a timescale for  56932
                    :
                    :
***** Calculating a timescale for  58726
```




```
**2** make a timescale with prediction *****

????? Save figure? (y/n) = y   8

::::: save logfile =  ./TA_RESULT/0731_155524/log_0731_155524.txt
::::: save figfile =  ./TA_RESULT/0731_155524/fig_0731_155524.pdf
::::: save alldata =  ./TA_RESULT/0731_155524/all_raw_clk_TA_0731_155524.dat
::::: save ADEVdat =  ./TA_RESULT/0731_155524/all_adev_0731_155524.dat
::::: save weights =  ./TA_RESULT/0731_155524/all_weight_0731_155524.dat
::::: save wgt_fig =  ./TA_RESULT/0731_155524/wgt_fig_0731_155524.pdf
```

Contents

3. Exercise : Case-1 : Calculate using good clocks

■ Case-1 : Calculate using good clocks

Basic operation & Replacement of clocks

[1] Set "*clock_file_list.txt*"

- Backup the current file if needed.
- Open the file.
- Select all clocks
  = All weights are set to "1"
- Save the file.

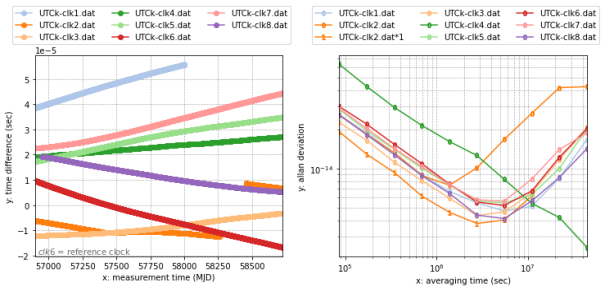
"clock_file_list.txt"

```
# sum of weights is
normalized to 1 in the
program.
# flg1=1 means the
reference clock of phase
comparison.
# filename     : weight : flg1
1 UTCk-clk1.dat :  1   : 0
2 UTCk-clk2.dat :  1   : 0
3 UTCk-clk3.dat :  1   : 0
4 UTCk-clk4.dat :  1   : 0
5 UTCk-clk5.dat :  1   : 0
6 UTCk-clk6.dat :  1   : 1
7 UTCk-clk7.dat :  1   : 0
8 UTCk-clk8.dat :  1   : 0
```

[2] Run "*TAgen_pred.py*" and confirm the processing

1

```
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17)
[MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.13.2 -- An enhanced Interactive Python.

runfile('C:/Mywork/Python/Pytool_TAgen/TAgen_Pred/TAgen_
pred.py',
wdir='C:/Mywork/Python/Pytool_TAgen/TAgen_Pred')

..... Reading initial data of 'UTC(k) - Clock'
..... formatting the initial data for processing

************ Initial status of all clocks ************

!!!!! UTCk-clk2.dat  includes outliers
```




```
**0** Read and plot initial clock data and Allan deviation*****

????? Save figure? (y/n) = y  2  (Cont.)
```

- Open the program and run it. 1
- Input "*y*". 2
- Take note the **directory name.** 3
- Exit the program. 4

(Cont.)

```
????? Save figure? (y/n) = y  3

::::: save logfile
=  ./TA_RESULT/0807_164038/log_0807_164038.txt
::::: save figfile
=  ./TA_RESULT/0807_164038/fig_0807_164038.pdf
::::: save alldata
=  ./TA_RESULT/0807_164038/all_raw_clk_0807_164038.dat
::::: save ADEVdat
=  ./TA_RESULT/0807_164038/all_adev_0807_164038.dat

==========================================
     Menu of the data processing
------------------------------------------
 0.  Plot the "UTC(k)-Clock" data
 1.  Assign the MJD range
 2.  Make TA with daily prediction
          and fixed clock weights
10.  Exit
==========================================
????? Input menu number ? >> 10  4
```

Bureau International des Poids et Mesures

3. Exercise : Case-1 : Calculate using good clocks

■ Case-1 : (cont.)

[3] Confirm the clock status

- Open "*./TA_RESULT/(MMDD_hhmmss)/fig_(MMDD_hhmmss).pdf*" (Refer the recorded directory name). 5
- Confirm the clock status : In this case, data missings at *clock1* and *clock8,* and *clock2* has a data missing and a gap. As for the Allan deviation, *clock4* seems to be worse than other clocks.

  ➔ I want to exclude *clock1, 2, 4, 8* from the ***TAp*** calcluation.

[4] Select the good clocks for *TAp*

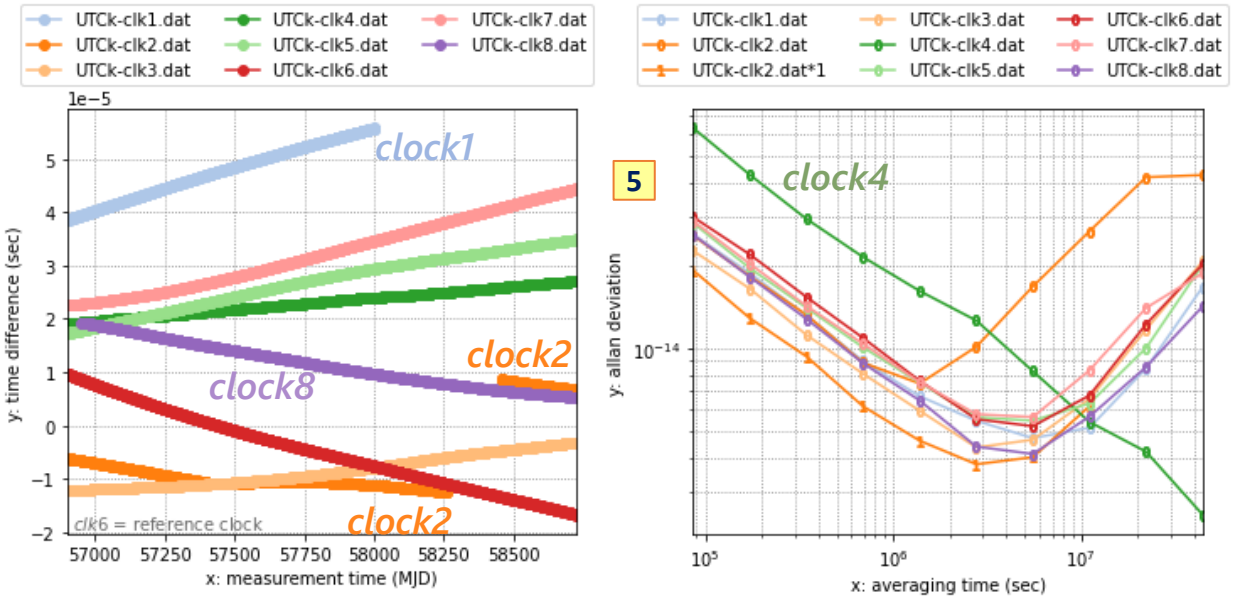
- Backup the file if needed.
- Input "*0*" to the weights of *clock1, 2, 4, 8*. 6

  These clocks are excluded from the ***TAp*** calculation.
- Save the file.

"clock_file_list.txt"

```
# sum of weights is
normalized to 1 in the
program.
# flg1=1 means the
reference clock of phase
comparison.
# filename     : weight : flg1
1 UTCk-clk1.dat :  0  : 0
2 UTCk-clk2.dat :  0  : 0
3 UTCk-clk3.dat :  1  : 0
4 UTCk-clk4.dat :  0  : 0
5 UTCk-clk5.dat :  1  : 0
6 UTCk-clk6.dat :  1  : 1
7 UTCk-clk7.dat :  1  : 0
8 UTCk-clk8.dat :  0  : 0
```

6



Bureau International des Poids et Mesures

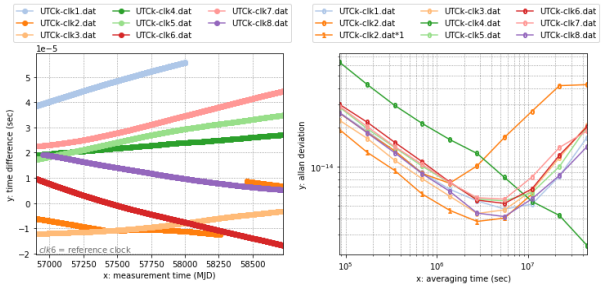
3. Exercise : Case-1 : Calculate using good clocks

■ Case-1 : (Cont.)

[5] Calculate *TAp*

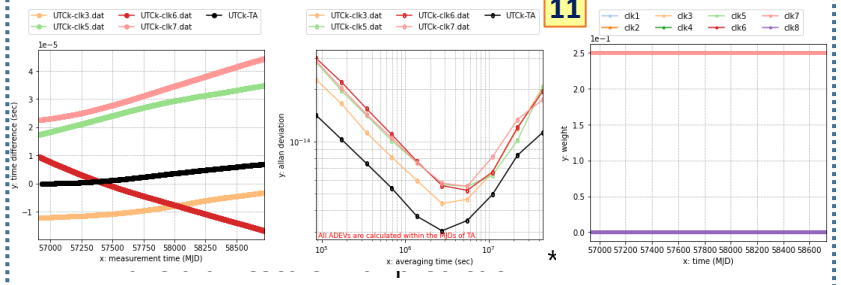
- Run "*TAgen_pred.py*" again.
- Saving can be skipped. 7
- ※ In the initial plotting, only the clocks marked with "#" are excluded.
- Select "*menu* ***2***". 8
- Input "***30***" for T_{rate}. 9
  …. Calculation … 10
- Confirm the plotting. 11
- ※ In a *TAp* calculation, the clocks marked with "#" or zero weight are excluded.
- Save the results and take note the directory name. 12
- Exit the program.

```
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17)
[MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.13.2 -- An enhanced Interactive Python.

runfile('C:/Mywork/Python/Pytool_TAgen/TAgen_Pred/TAgen_
pred.py',
wdir='C:/Mywork/Python/Pytool_TAgen/TAgen_Pred')

..... Reading initial data of 'UTC(k) - Clock'
..... formatting the initial data for processing

************ Initial status of all clocks ************

!!!!! UTCk-clk2.dat  includes outliers
```




```
**0** Read and plot initial clock data and Allan deviation*****

????? Save figure? (y/n) = n  7

=======================================
     Menu of the data processing
---------------------------------------
  0.  Plot the "UTC(k)-Clock" data
  1.  Assign the MJD range
  2.  Make TA with daily prediction
          and fixed clock weights
 10.  Exit
=======================================
????? Input menu number ? >> 2  8
```

```
????? Input menu number ? >> 2  8

????? Span in days (try 10 or 30 or 90) = 30  9

***** Calculating a timescale for  56932
                    :
                10  :
***** Calculating a timescale for  58726
```




```
????? Save figure? (y/n) = y

::::: save logfile                                    12
=  ./TA_RESULT/0807_182508/log_0807_182508.txt
::::: save figfile
=  ./TA_RESULT/0807_182508/fig_0807_182508.pdf
::::: save alldata
=  ./TA_RESULT/0807_182508/all_raw_clk_TA_0807_182508.dat
::::: save ADEVdat
=  ./TA_RESULT/0807_182508/all_adev_0807_182508.dat
::::: save weights
=  ./TA_RESULT/0807_182508/all_weight_0807_182508.dat
::::: save wgt_fig
=  ./TA_RESULT/0807_182508/wgt_fig_0807_182508.pdf

=================================
     Menu of the data processing
---------------------------------
  0.  Plot the "UTC(k)-Clock" data
```

Bureau International des Poids et Mesures

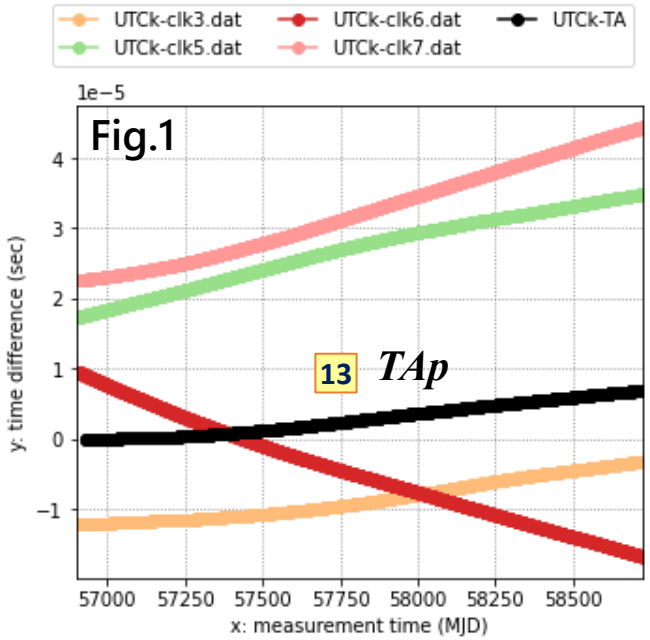
3. Exercise : Case-1 : Calculate using good clocks

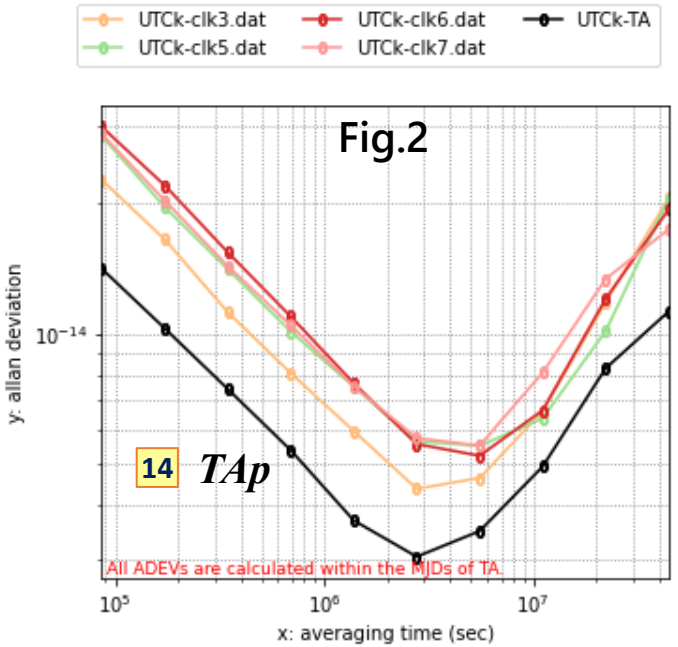
■ Case-1 : (cont.)

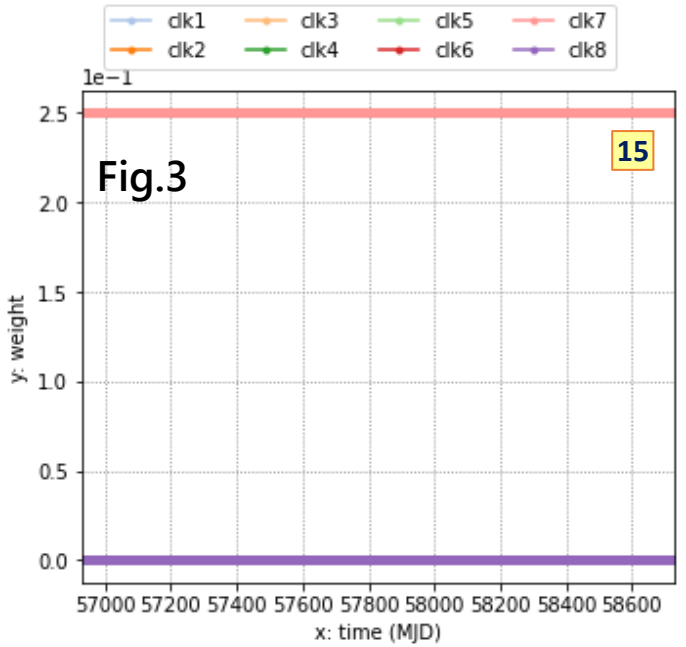
Through this exercise, you confirmed the ordinally operation of making ***TAp***.

[6] Evaluate the results

- Open "*./TA_RESULT/(MMDD_hhmmss)/fig_(MMDD_hhmmss).pdf*" and "***wgt_fig_(MMDD_hhmmss).pdf***".
- In Fig.1, the time differences vs ***UTC(k)***, ***TAp*** shows good continuity and flatness. 13
- In Fig.2, the Allan deviation of Fig.1 data, ***TAp*** shows better stability than any of all clocks. 14
- The weights of *clock3, 5, 6, 7* are *0.25* because their initial weights are the same. (Refer "***all_weights_(MMDD_hhmmss).dat***" 15







Bureau International Poids et Mesures

3. Exercise : Case-2 : If all the clocks are used

■ Case-2 : If all the clocks are used

Confirm the merit of *TAp*

[1] Set the weights of all clocks to "*1*" and calculate *TAp*

"clock_file_list.txt"

```
# sum of weights is normalized
to 1 in the program.
# flg1=1 means the reference
clock of phase comparison.
# filename     : weight : flg1
1 UTCk-clk1.dat :  0  : 0
2 UTCk-clk2.dat :  0  : 0
3 UTCk-clk3.dat :  1  : 0
4 UTCk-clk4.dat :  0  : 0
5 UTCk-clk5.dat :  1  : 0
6 UTCk-clk6.dat :  1  : 1
7 UTCk-clk7.dat :  1  : 0
8 UTCk-clk8.dat :  0  : 0
```

"clock_file_list.txt"

```
# sum of weights is normalized
to 1 in the program.
# flg1=1 means the reference
clock of phase comparison.
# filename     : weight : flg1
1 UTCk-clk1.dat :  1  : 0
2 UTCk-clk2.dat :  1  : 0
3 UTCk-clk3.dat :  1  : 0
4 UTCk-clk4.dat :  1  : 0
5 UTCk-clk5.dat :  1  : 0
6 UTCk-clk6.dat :  1  : 1
7 UTCk-clk7.dat :  1  : 0
8 UTCk-clk8.dat :  1  : 0
```

3. Exercise : Case-2 : If all the clocks are used

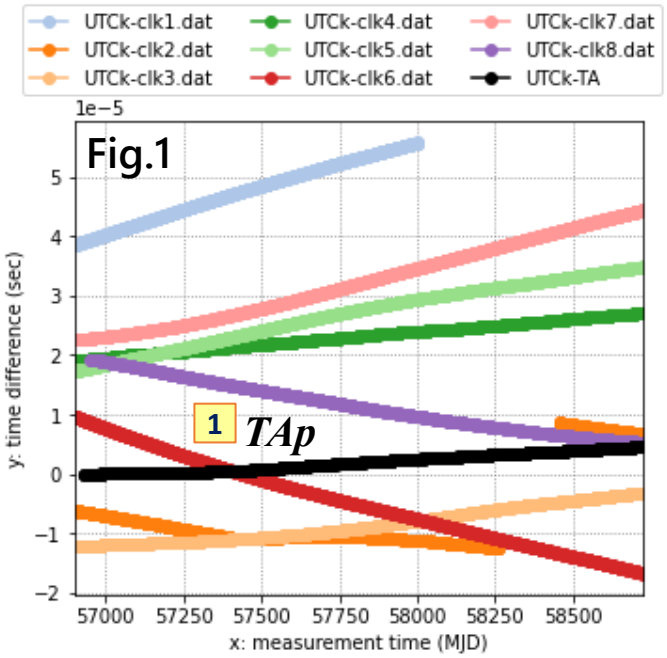
■ Case-2 : If all the clocks are used

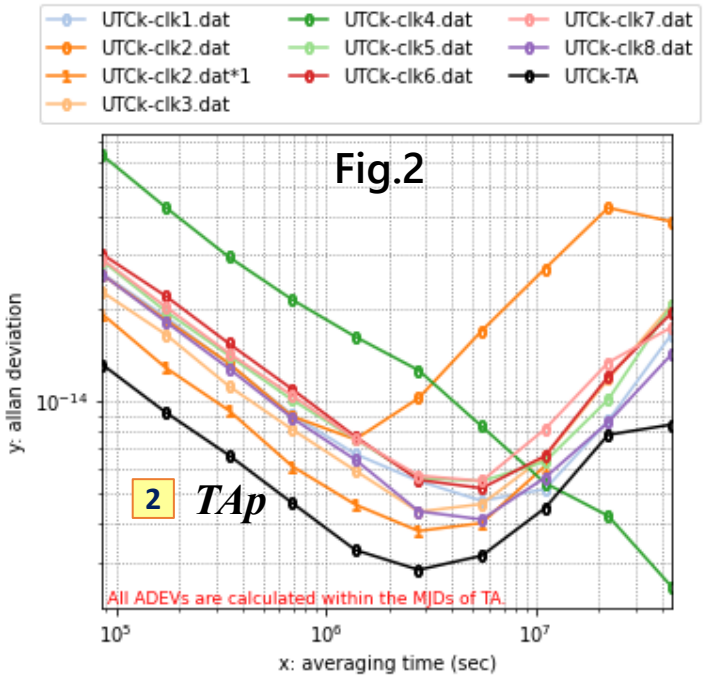
Confirm the merit of *TAp*

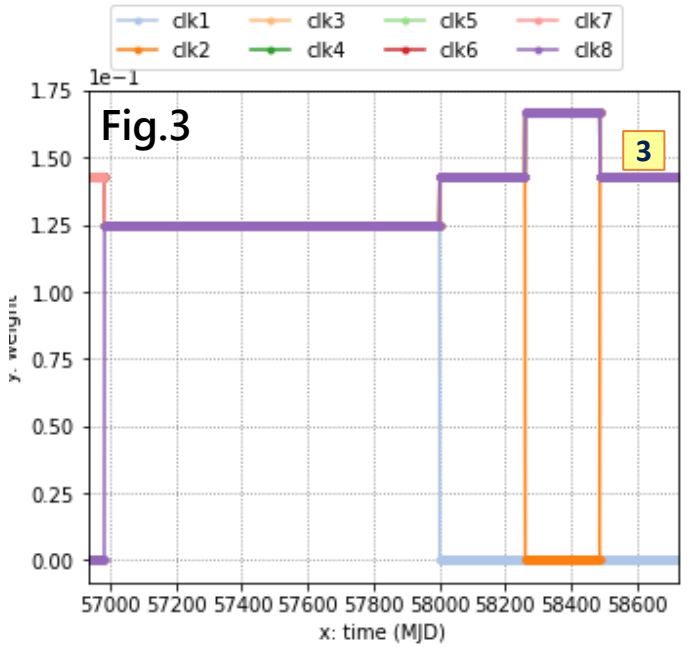
[1] Set the weights of all clocks to "*1*" and calculate *TAp*

- In Fig.1, ***TAp*** seems to be continuous and flat. 1
- In Fig.2, ***TAp*** seems to have almost better stability than any clocks. 2
- In Fig.3, *clock1, 2, 8* have zero weights during certain periods. 3
  Acoordingly, the weights of the other clocks change.

- ***TAp*** **could maintain continuity even if some clocks with data missing were included.**







Bureau International Poids et Mesures

3. Exercise : Case-3 : Change the bias of weights

■ Case-3 : Change the bias of the weights

Confirm the effect of bias of the weights

[A] *clock4* : *clock7* = ***1 : 1*** 1

[1] Set clock parameters
[2] Run the program
[2] Select "*menu-2*"
[3] Input "*30*" to the Trate
[4] Save the results
(Take note the directory name)
[5] Exit the program

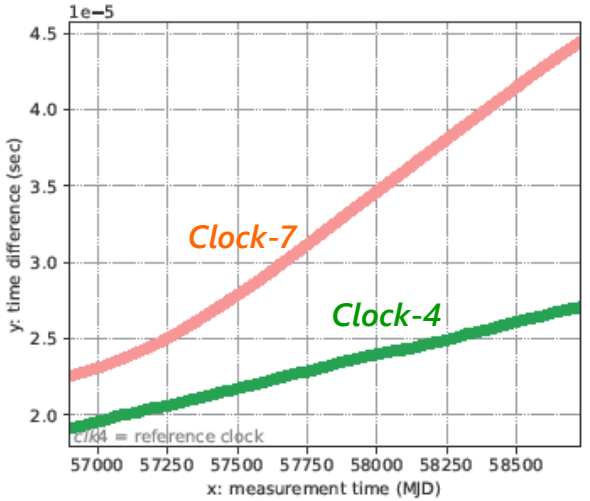
"clock_file_list.txt"

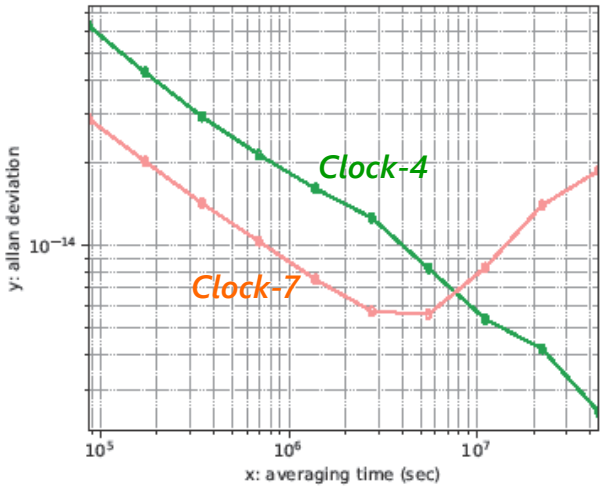
```
# sum of weights is normalized
to 1 in the program.
# flg1=1 means the reference
clock of phase comparison.
# filename     : weight : flg1
1 UTCk-clk1.dat :  0  : 0
2 UTCk-clk2.dat :  0  : 0
3 UTCk-clk3.dat :  0  : 0
4 UTCk-clk4.dat :  1  : 0
5 UTCk-clk5.dat :  0  : 0
6 UTCk-clk6.dat :  0  : 1
7 UTCk-clk7.dat :  1  : 0
8 UTCk-clk8.dat :  0  : 0
```

[B] *clock4* : *clock7* = ***4 : 1***
([1]~[5] are the same)

[C] *clock4* : *clock7* = ***1 : 4***
([1]~[5] are the same)

- Predict the differences between these *TAps* in advance...



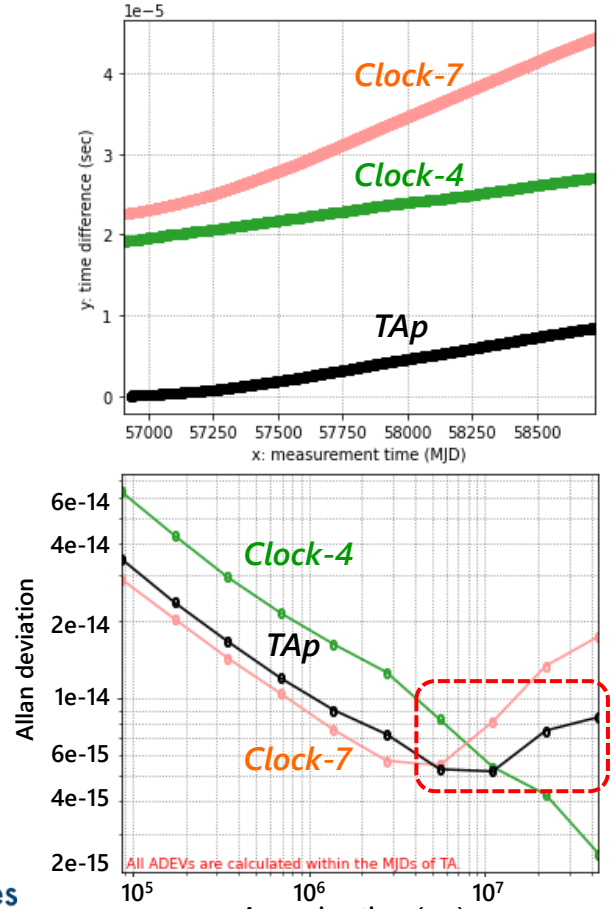


Bureau International des Poids et Mesures

3. Exercise : Case-3 : Change the bias of weights

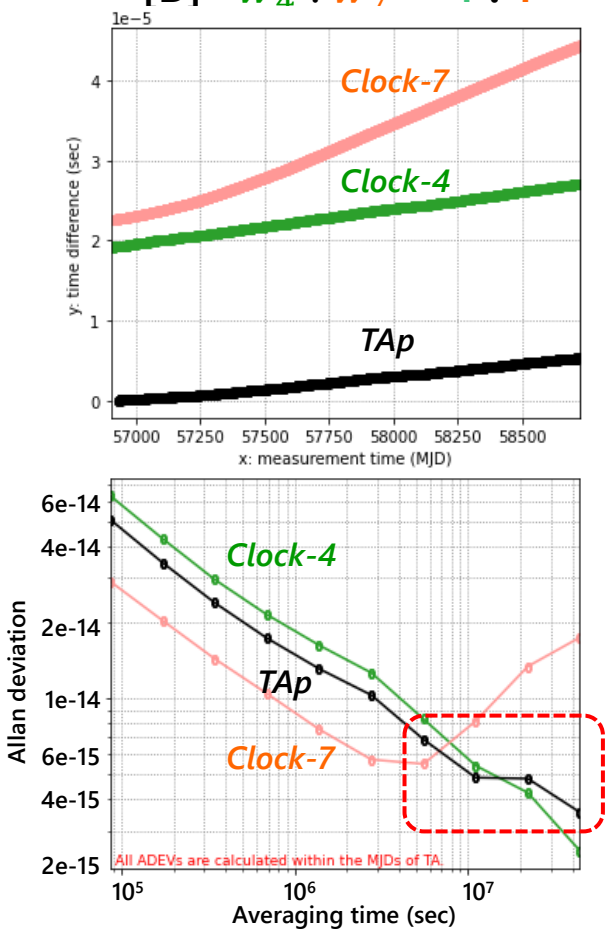
- *TAp* becomes closer to the clock with higher weight.

[A] $w_4 : w_7 = 1 : 1$



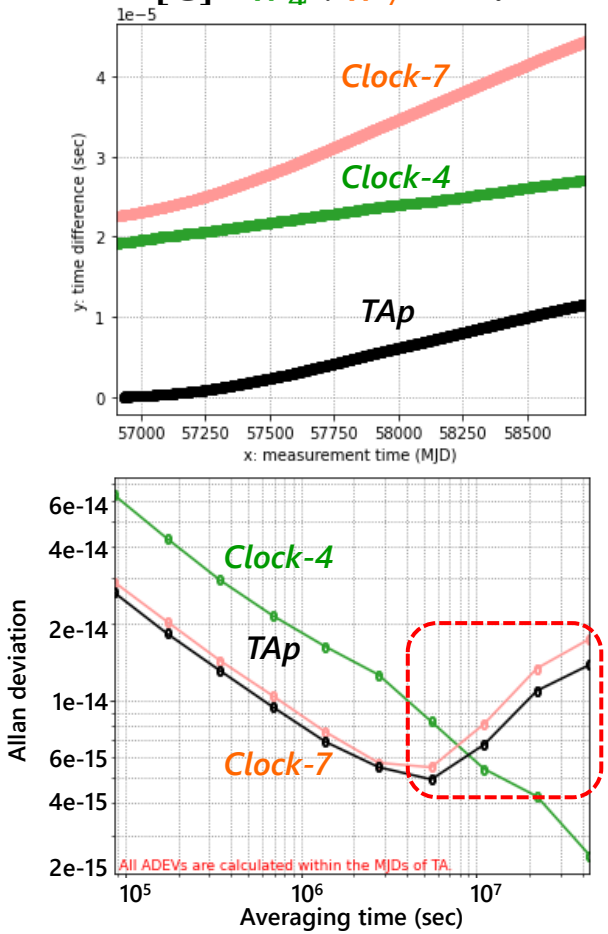
/0808_152121

[B] $w_4 : w_7 = 4 : 1$



/0808_152345

[C] $w_4 : w_7 = 1 : 4$



/0808_152734

Bureau International des Poids et Mesures

Contents

4. Summary and Comments

Summary

- This training aims to help participants understand the principles of "*TA with prediction (TAp)*" and how to make and evaluate it using the tutorial Python program "*TAgen_pred.py*".
- Participants learnt how to use the program through the lecture and exercise of basic operation process.
- Further information about *TAp* and the program are provided in *[Ref. 2]* and *[Ref. 3]*.

4. Summary and Comments

■ Comments

- This "*TAgen_pred.py*" was developed by a beginner in Python programming (i.e. me), so there may be bugs or areas that are not fully developed. If you find any issues, please contact me via BIPM.
- If you are interested in modifying the program, please refer to the user guide *[Ref. 3]* which provides information on the program's construction and flow.
- For "*TA by basic averaging* (*TAb*)", which is more basic and simpler form of *TA*, another Python program "*TAgen_basic.py*" is provided.
- For more optimal and practical *TAp*, another program "*TAgen_auto.py*" is provided. It is almost the same as "*TAgen_pred.py*", except that it adopts a dynamic weighting.
- These programs and manuals are provided on the BIPM CBKT e-learning web-site *[Ref. 1],* so please refer to them if you are interested.

Thank you very much for your kind attention!

※ Reference

[1] "*Time scale algorithms Basics to Applications*", BIPM course on the BIPM e-learning platform (https://e-learning.bipm.org/)
[2] Y. Hanado, *"Time scale algorithms ~basics to applications~ Lecture", BIPM Summer School, September 11, 2025*
[3] "*User's Manual of "TAgen_pred.py"*, BIPM course on the BIPM e-learning platform (https://e-learning.bipm.org/)
[4] Web site of *"AllanTools"* (*https://allantools.readthedocs.io/en/latest/*)

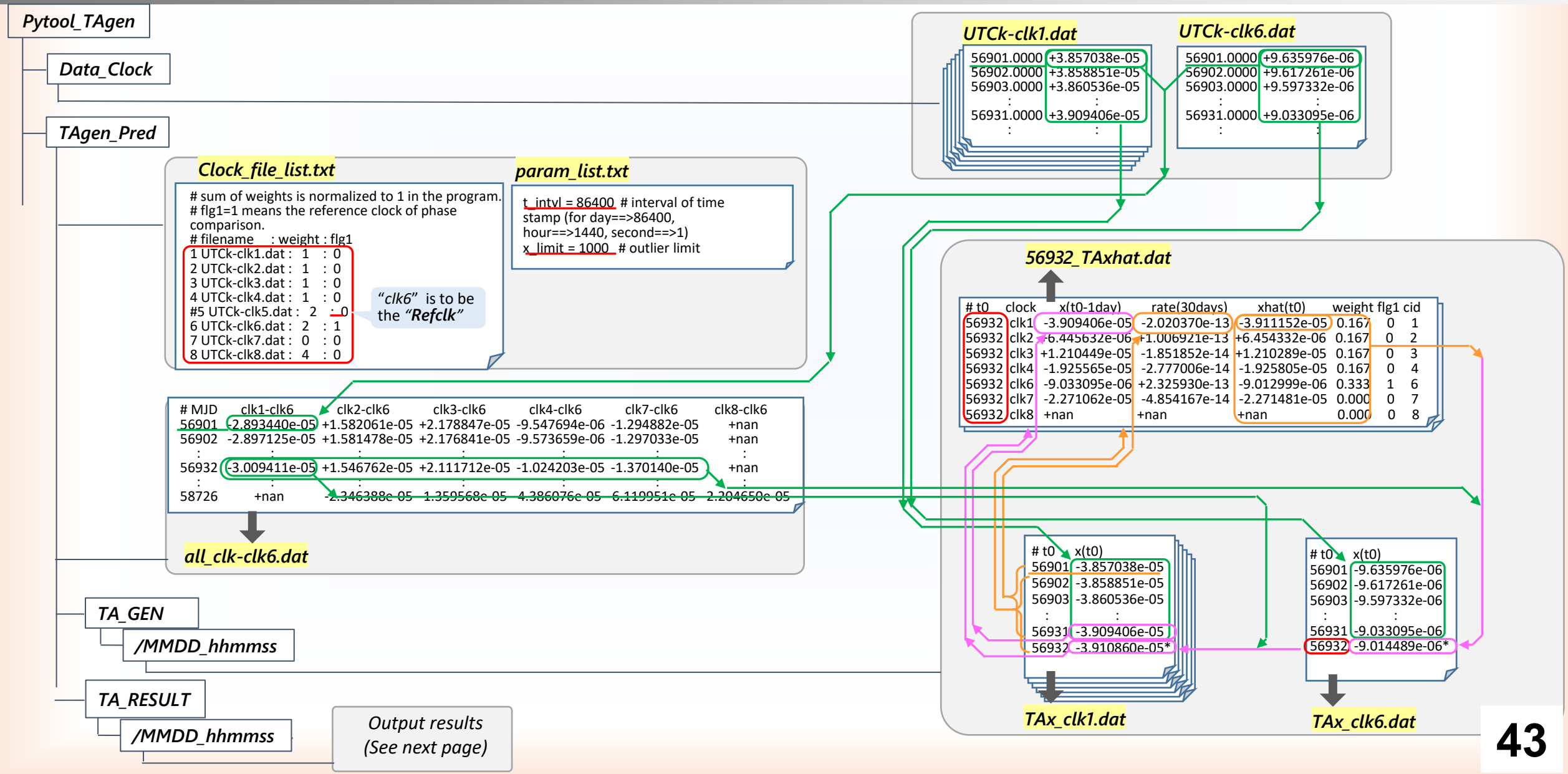
Appendix-1 : Input / Output files

A-1.1. Relation of the files

A-1.2. Files used in the calculation

A-1.3. Output files of results

A-1.1 Relation of the files (1)



A-1.1 Relation of the files (2)

Pytool_TAgen/TAgen_pred/TA_RESULT/0717_093410/

Clock_file_list.txt

```
# sum of weights is normalized
to 1 in the program.
# flg1=1 means the reference
clock of phase comparison.
# filename    : weight : flg1
1 UTCk-clk1.dat :  1   : 0
2 UTCk-clk2.dat :  1   : 0
3 UTCk-clk3.dat :  1   : 0
4 UTCk-clk4.dat :  1   : 0
#5 UTCk-clk5.dat :  2   : 0
6 UTCk-clk6.dat :  2   : 1
7 UTCk-clk7.dat :  0   : 0
8 UTCk-clk8.dat :  4   : 0
```

UTCk-clk1.dat

```
56901.0000  +3.857038e-05
56902.0000  +3.858851e-05
      :           :
57999.0000  +5.576701e-05
58000.0000  +5.578885e-05
```

all_raw_clk_TA_0731_155524.dat

fname	UTCk-clk1.dat	UTCk-clk2.dat	UTCk-clk3.dat	UTCk-clk4.dat	UTCk-clk6.dat	UTCk-clk7.dat	UTCk-clk8.dat	UTCk-TA
Weight	1.0	1.0	1.0	1.0	2.0	0.0	4.0	0.0
flag	0	0	0	0	1	0	0	3
cid	1	2	3	4	6	7	8	0
56901	+3.857038e-05	-6.184638e-06	-1.215249e-05	+1.918367e-05	+9.635976e-06	+2.258480e-05	+nan	+nan
:	:	:	:	:	:	:	:	:
56931	+3.909406e-05	-6.445632e-06	-1.210449e-05	+1.925565e-05	+9.033095e-06	+2.271062e-05	+nan	+nan
56932	+3.910934e-05	-6.452386e-06	-1.210189e-05	+1.925726e-05	+9.015229e-06	+2.271663e-05	+nan	+7.400889e-10
:	:	:	:	:	:	:	:	:
58726	+nan	+6.687379e-06	-3.180825e-06	+2.708426e-05	-1.677650e-05	+4.442301e-05	+5.270004e-06	+4.684945e-06

all_adev_0731_155524.dat

Tau(s)	UTCk-clk1.dat	UTCk-clk2.dat	UTCk-clk2.dat*1	UTCk-clk3.dat	UTCk-clk4.dat	UTCk-clk6.dat	UTCk-clk8.dat	UTCk-TA
MJDspan	56932-58000	56932-58260	58457-58726	56932-58726	56932-58726	56932-58726	56952-58726	56932-58726
86400	+2.593622e-14	+2.572997e-14	+1.914455e-14	+2.260179e-14	+6.282088e-14	+2.997311e-14	+2.576366e-14	+1.587481e-14
172800	+1.879802e-14	+1.838375e-14	+1.284166e-14	+1.647945e-14	+4.260514e-14	+2.195515e-14	+1.811728e-14	+1.101474e-14
:	:	:	:	:	:	:	:	:
11059200	+5.182775e-15	+2.695475e-14	+6.159347e-15	+6.611689e-15	+5.426419e-15	+6.643764e-15	+5.645967e-15	+4.474511e-15
22118400	+8.639638e-15	+4.250378e-14	+nan	+1.179217e-14	+4.242674e-15	+1.202004e-14	+8.546454e-15	+7.239207e-15
44236800	+1.646562e-14	+3.832687e-14	+nan	+2.051915e-14	+2.502899e-15	+1.933946e-14	+1.418509e-14	+9.593108e-15

log_0731_155524.txt

```
# Data log at 2025.07.31_15h55m_24s
#
**0** Read and plot initial clock data and Allan deviation*****
::::: Input clocklist   : clock_file_list.txt
::::: Input clock file  : ['UTCk-clk1.dat', 'UTCk-clk2.dat', 'UTCk-clk3.dat', 'UTCk-
clk4.dat', 'UTCk-clk6.dat', 'UTCk-clk7.dat', 'UTCk-clk8.dat']
::::: data time interval (t_intval): 86400(sec)
::::: Outlier limit (x_limit))     : 1000.0

**2** make a timescale with predicton *****
::::: saved TAxhat data : ./TA_GEN/0731_155500/(MJD)_TAxhat.dat
::::: saved TAx data : ./TA_GEN/0731_155500/TAx_(clock).dat
::::: interval for rate estimation (TArate_span)     : 30(days)
```

all_weight_0731_155524.dat

# MJD	clk1	clk2	clk3	clk4	clk6	clk7	clk8
56932.0	0.167	0.167	0.167	0.167	0.333	0.000	0.000
:	:	:	:	:	:	:	:
56983.0	0.100	0.100	0.100	0.100	0.200	0.000	0.400
:	:	:	:	:	:	:	:
58001.0	0.000	0.111	0.111	0.111	0.222	0.000	0.444
:	:	:	:	:	:	:	:
58261.0	0.000	0.000	0.125	0.125	0.250	0.000	0.500
:	:	:	:	:	:	:	:
58488.0	0.000	0.111	0.111	0.111	0.222	0.000	0.444
:	:	:	:	:	:	:	:
58726.0	0.000	0.111	0.111	0.111	0.222	0.000	0.444

wgt_fig_0731_155524.pdf

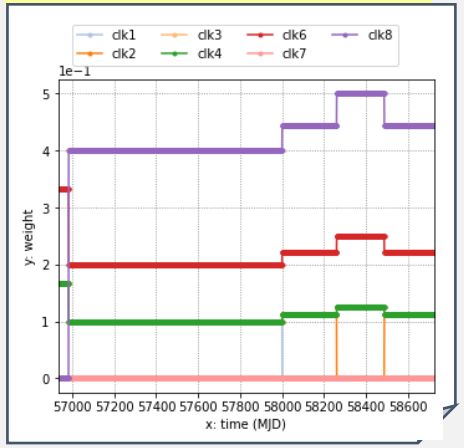
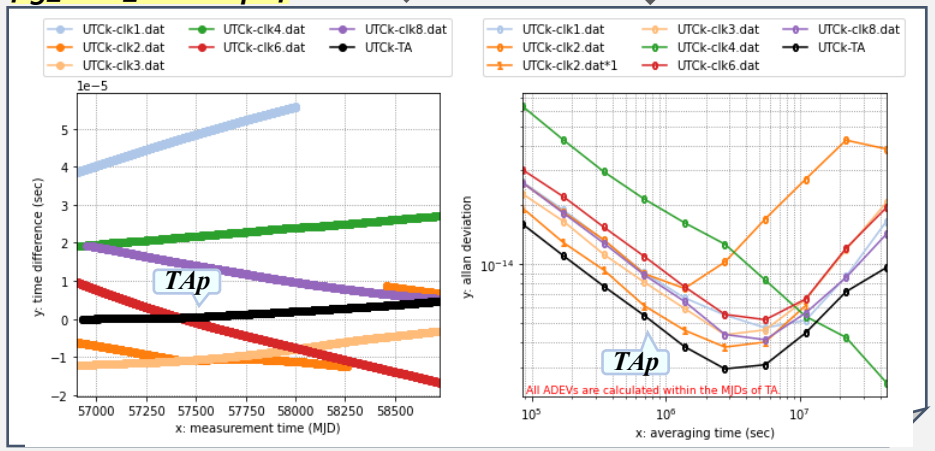


fig_0731_155524.pdf



Appendix-1 : Input / Output files

A-1.1. Relation of the files

A-1.2. Files used in the calculation

A-1.3. Output files of results

A-1.2.1 *"clock_fie_list.txt"*

■ *"/Pytool_TAgen/TAgen_Pred/**clock_file_list.txt**"*

Outline

- "***clock_file_list.txt***" provides the conditions of each clock in the averaging process.
- This table should be prepared in advance.

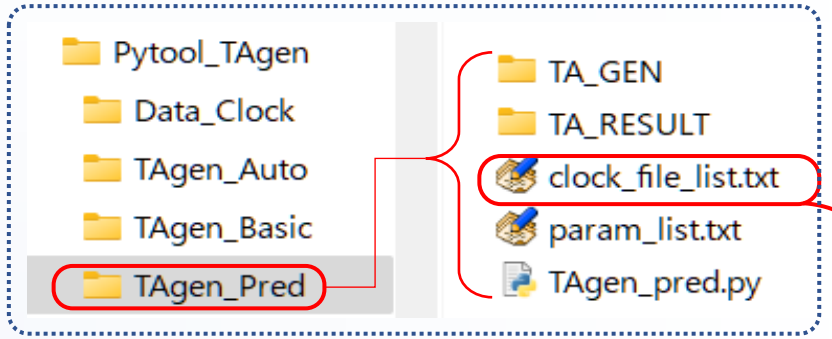
Details / Rules of use

- The line starting with "**#**" is ignored. 1
- 1st column : file No. (defines plot color)
- 2nd column : data file name
- 3rd column : clock weight*1
- 4th column : flag of the reference clock *2.

Note

*1 : Each weight is normalized in the program. 2

*2 : The clock with *"**flg1=1**"*, the reference clock named as "***Refclk***" hereafter, is used as the base clock to get the time difference between clocks. 3




```
# sum of weights is normalized to 1 in the program.
# flg1=1 means the reference clock of phase comparison.
# filename     : weight : flg1
1 UTCk-clk1.dat :  1  : 0
2 UTCk-clk2.dat :  1  : 0
3 UTCk-clk3.dat :  1  : 0
4 UTCk-clk4.dat :  1  : 0
#5 UTCk-clk5.dat :  2  : 0
6 UTCk-clk6.dat :  2  : 1
7 UTCk-clk7.dat :  0  : 0
8 UTCk-clk8.dat :  4  : 0
```

1 • "***clk5***" is ignored by "#".

2 • ***Clk5*** *is ignored from all the calculation*
• ***clk7*** *is excluded from the averaging*

3 • "***clk6***" is to be the "***Refclk***" (reference clock).

A-1.2.2 *"param_list.txt"*

■ *"/Pytool_TAgen/TAgen_Pred/**param_list.txt**"*

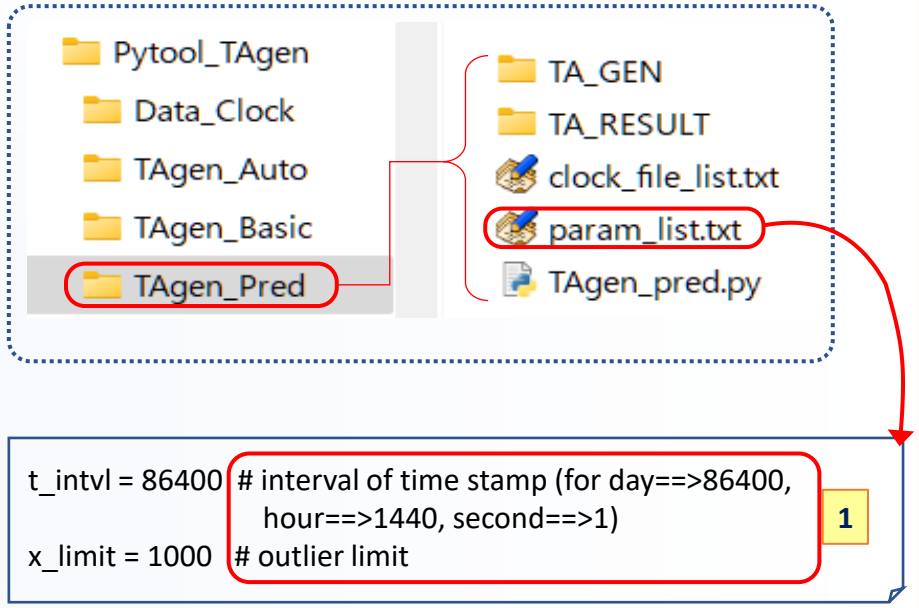
Outline

- *"**param_list.txt**"* provides the parameters for the ***TAp*** calculation.
- Parameters that are fixed within the program but can be changed on demand are stored here.
- This table should be prepared in advance.

Details / Rules of use

- These parameters are used with the same names in the program.
- The text after "#" provides the explanation of the parameter. 1

- ***t_intvl*** : interval of time stamp
  (for day → *86400*, hour → *1440*, second → *1*)
- ***x_limit*** : outlier limit



A-1.2.3 "*UTCk_(clock$_i$).dat*"

■ "*/Pytool_TAgen/Data_Clock/**UTCk-(clock$_i$).dat***"

Outline

- "***UTCk-(clock$_i$).dat***" are the measurement clock data as the materials of ***TAp*** calculation.
- This data files should be prepared in advance.

Details / Rules of use

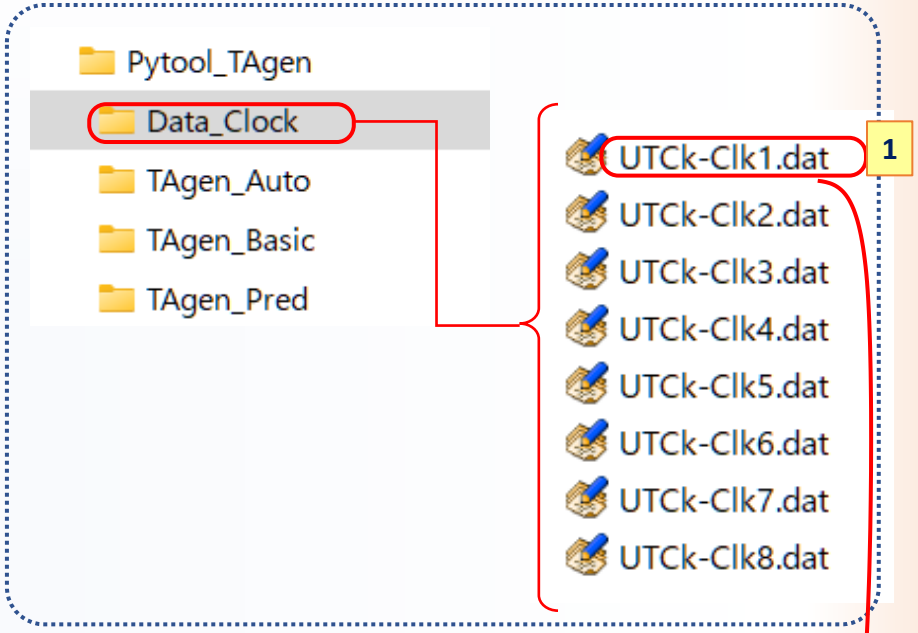
- One file contains data from a single clock.
- The file name should be "***UTCk-****.dat***", where the four characters "****" identify each clock. 1
- 1st column : observation time stamp 2
- 2nd column : measured time difference of "***UTC(k) – clock$_i$***" (in second) 3

Note

※ The Interval of the 1st column should be specified as ***t_intvl*** in "***param_list.txt***". *(p.46)* 4

※ This sampling interval must be constant and continuous.

※ Sufficient data is required for each clock to start the calculation. *(p.63)*



A-1.2.4 *"all_clk_(Refclk).dat"*

■ *"/Pytool_TAgen /TAgen_Pred/**all_clk_(Refclk).dat**"*

Outline

- *"**all_clk_(Refclk).dat**"* is the merged data of time difference "***clock$_i$ - Refclk***" calculated from the *"**UTCk-(clock$_i$).dat**"* files.
- This file is an intermediate product during *"menu-2"*.

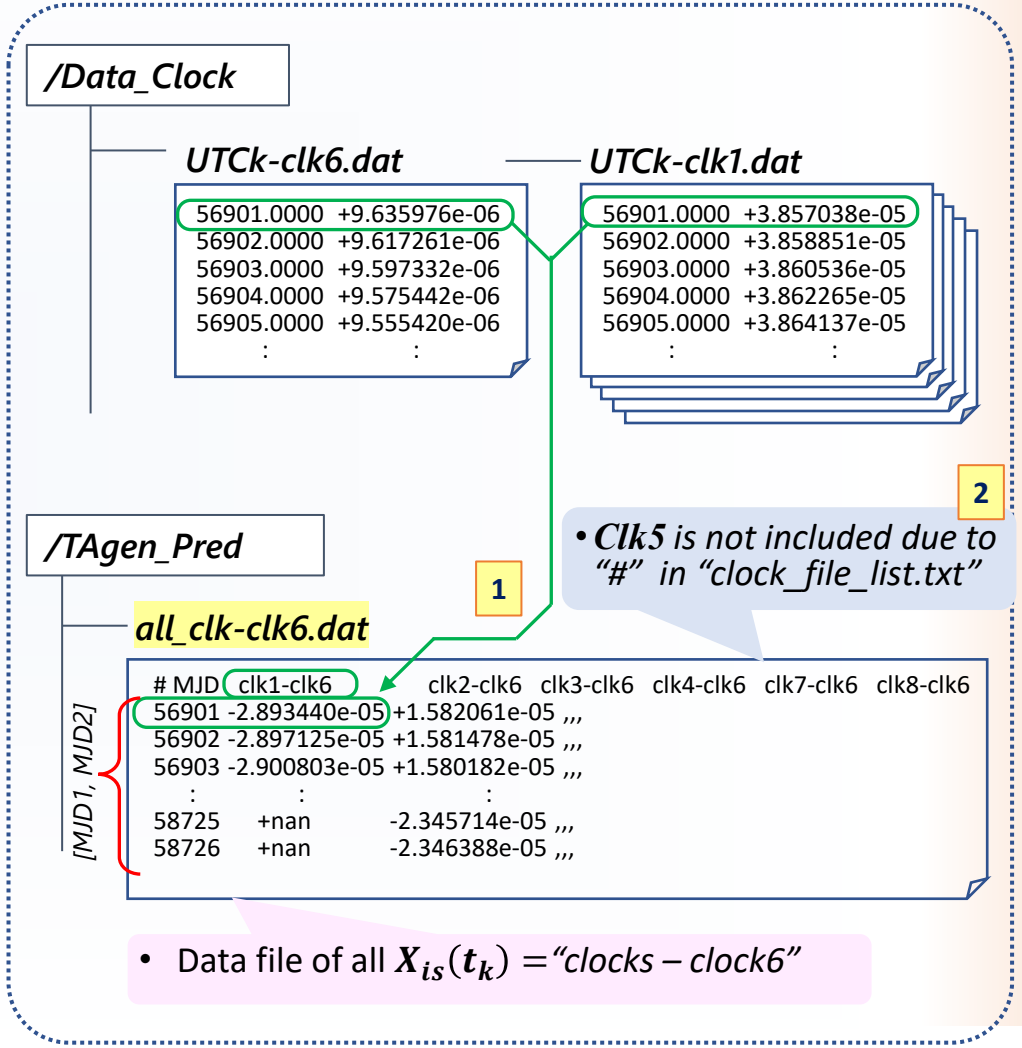
Details / Rules of use

- ***Refclk*** is assigned by the flg1 in *"clock_file_list.txt"*.

- 1st column : observation time stamp
- 2nd column : measured time difference of "***clock$_i$ - Refclk***" 1
  :

Note

※ The clock with "**#**" in *"**clock_file_list.txt**"* is excluded. *(p.45)* 2



A-1.2.5 "(MJD)_TAxhat.dat"

■ "/Pytool_TAgen/TAgen_Pred/TA_GEN/(time)/**(MJD)_TAxhat.dat**"

Outline

- "***(MJD)_TAxhat.dat***" is the intermediate product for checking the ***TAp*** calculation. The main result is $\hat{x}_i(t_k)$.
- These data are intermediate products during "*menu-2*" : one file is for one day result.

Details / Rules of use

1

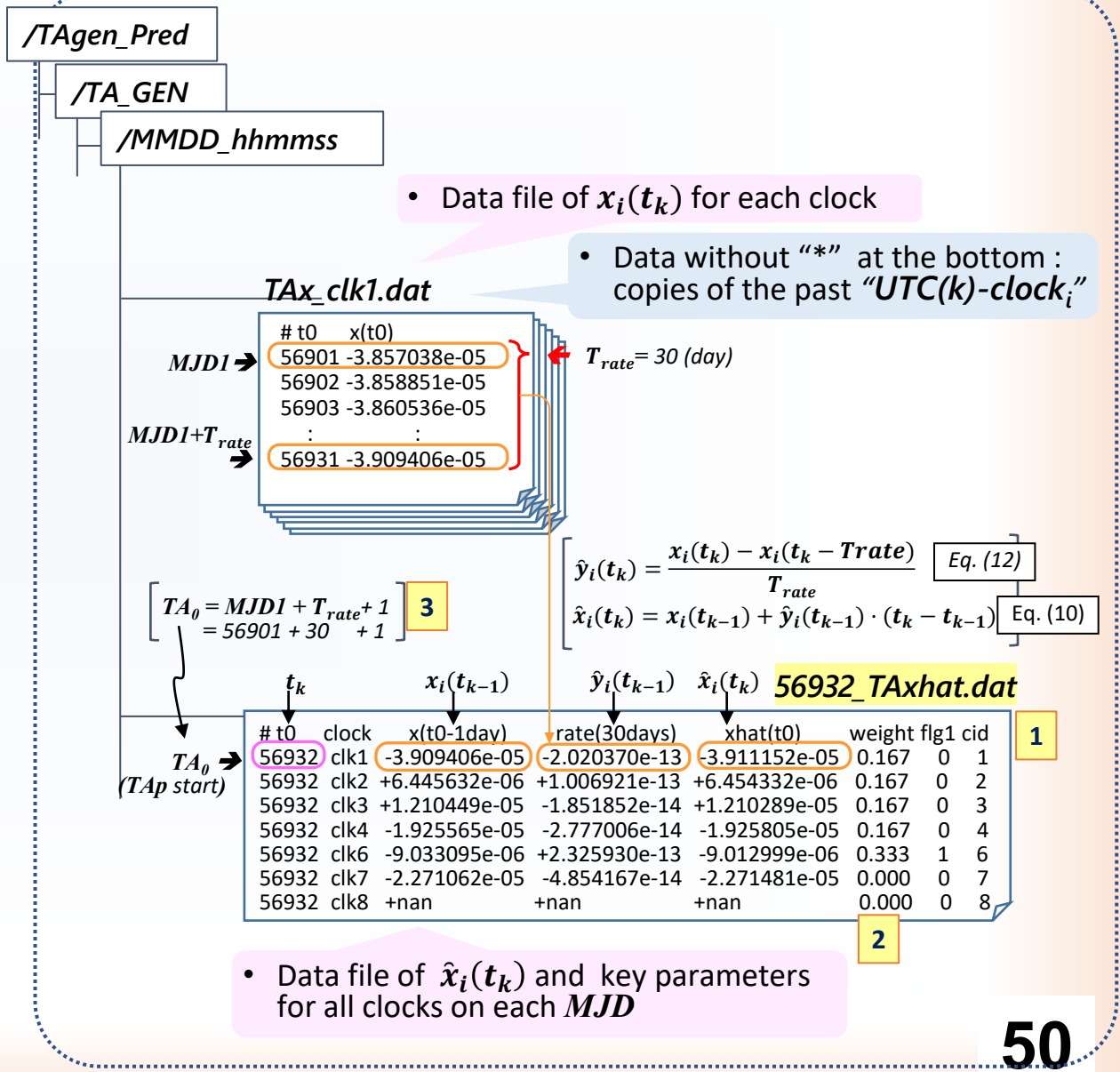
- 1st column : t_k: MJD to determine ***TAp***
- 2nd column : clock name
- 3rd column : $x_i(t_{k-1})$: x_i on the previous day of t_k
- 4th column : $\hat{y}_i(t_{k-1})$ (*Eq. (12) on p.5*)
- 5th column : $\hat{x}_i(t_k)$ (*Eq. (10), on p.5*)
- 6th column : normalized clock weight 2
- 7th column : ***flg1***
- 8th column : clock ID

Note

※ T_{rate} is given from the console at "*menu-2*".

※ The ***TAp*** start date is determined by "$TA_0 = MJD1 + T_{rate} + 1$". *(p.64)* 3

※ ***MJD1*** is the oldest *MJD* when the available clock data is provided, or is specified by "*menu-1*".



A-1.2.6 "*TAx_(clock$_i$).dat*"

■ "*/Pytool_TAgen/TAgen_Pred/TA_GEN/(time)/**TAx_(clock$_i$).dat***"

Outline

- "***TAx_(clock$_i$).dat***" is the time series of x_i of *clock$_i$* which is the targeted product of ***TAp*** calculation.

Details / Rules of use

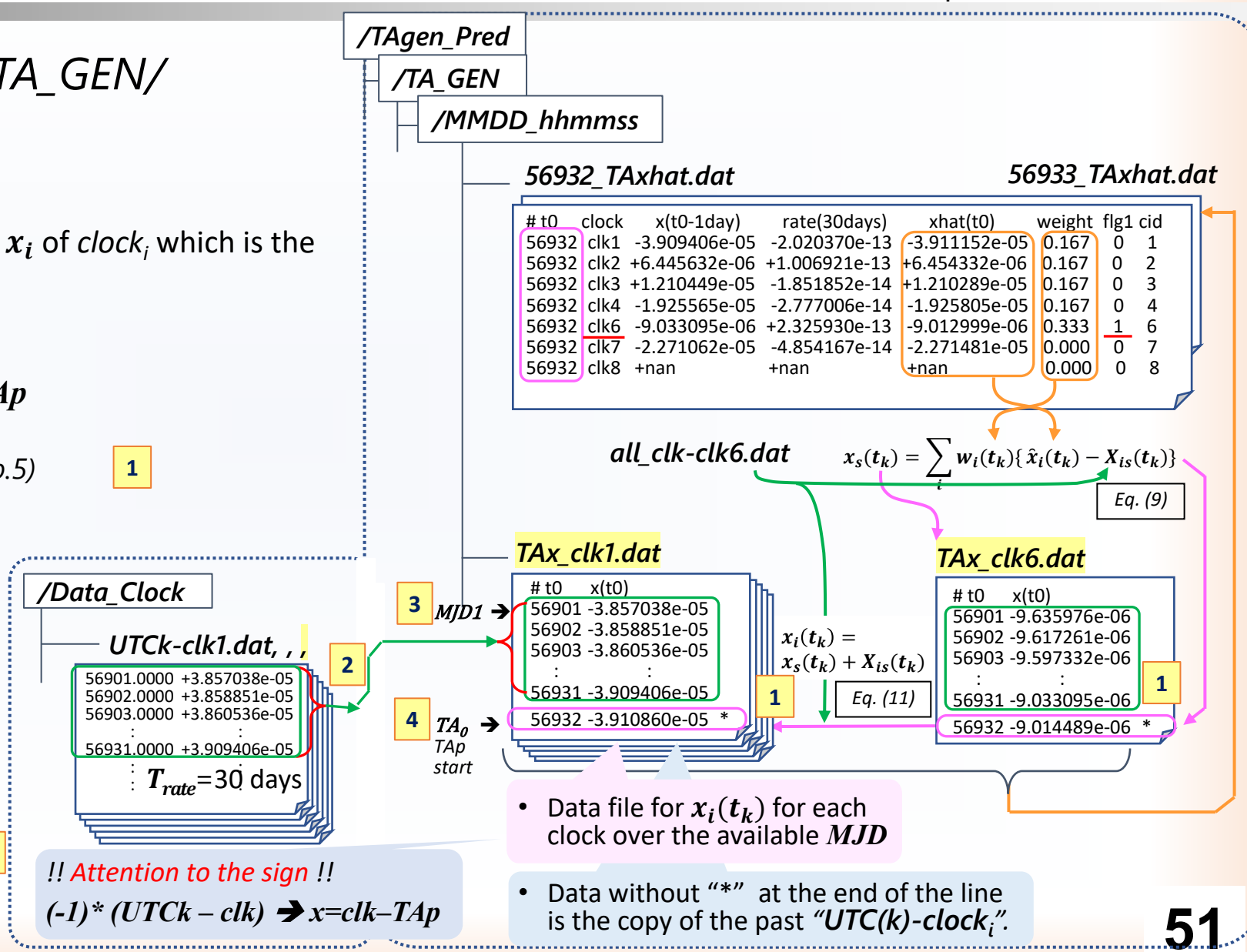
- 1st column : t_k: MJD to determine ***TAp***
- 2nd column : $x_i(t_k)$ of *clock$_i$*
  (derived from Eq.(9)(11) on p.5) 1

Note

※ Data without " * " at the bottom are the copies of the past "***UTC(k)-clock$_i$***". 2

※ The first date ***MJD1*** is the oldest date when the available clock data is provided, or is specified by "*menu-1*". 3

※ The ***TAp*** start date is determined by "$TA_0 = MJD1 + T_{rate} + 1$". *(p.64)* 4



Appendix-1 : Input / Output files

A-1.1. Relation of the files

A-1.2. Files used in the calculation

A-1.3. Output files of results

A-1.3.1 *"all_raw_clk_(time).dat"*

■ *"/Pytool_TAgen/TAgen_Pred/TA_RESULT/ (time)/**all_raw_clk_(time).dat**"*

Outline

- *"**all_raw_clk_(time).dat**"* is the combined data of *'**UTC(k)-clock$_i$**'*, which is used for the left plot of *"**fig_(time).pdf**"* (*p.57*)
- This data file is generated at the end of *Initial processing* on demand.

Details / Rules of use

- 1st line : clock file name
- 2nd line : *weight* : copy of the 3rd column of *"clock_file_list.txt"*
- 3rd line : *flag* : copy of the 4th column of *"clock_file_list.txt"*
- 4th line : *cid* : copy of the 1st column of *"clock_file_list.txt"*
- 5th line ~ : copy of the *"**UTC(k)-clock$_i$**"* of *"**UTCk_(clock$_i$).dat**"* (*p.47*)
- *(time)* : ***MMDD_hhmmss*** : time stamp of data saving
  (***MMDD***: month & date, ***hhmmss*** : hour & minute & second)

Note

※ The clock with *"#"* in *"**clock_file_list.txt**"* (*p.45*) is excluded.

Pytool_TAgen/TAgen_pred/TA_RESULT/0717_093343/

Clock_file_list.txt

```
# sum of weights is normalized
to 1 in the program.
# flg1=1 means the reference
clock of phase comparison.
# filename    : weight : flg1
1 UTCk-clk1.dat :  1   : 0
2 UTCk-clk2.dat :  1   : 0
3 UTCk-clk3.dat :  1   : 0
4 UTCk-clk4.dat :  1   : 0
#5 UTCk-clk5.dat :  2   : 0
6 UTCk-clk6.dat :  2   : 1
7 UTCk-clk7.dat :  0   : 0
8 UTCk-clk8.dat :  4   : 0
```

UTCk-clk1.dat

```
56901.0000  +3.857038e-05
56902.0000  +3.858851e-05
      :           :
57999.0000  +5.576701e-05
58000.0000  +5.578885e-05
```

all_raw_clk_0717_093343.dat

fname	UTCk-clk1.dat	UTCk-clk2.dat	UTCk-clk3.dat	UTCk-clk4.dat	UTCk-clk6.dat	UTCk-clk7.dat	UTCk-clk8.dat
Weight	1.0	1.0	1.0	1.0	2.0	0.0	4.0
flag	0	0	0	0	1	0	0
cid	1	2	3	4	6	7	8
56901	+3.857038e-05	-6.184638e-06	-1.215249e-05	+1.918367e-05	+9.635976e-06	+2.258480e-05	+nan
56902	+3.858851e-05	-6.197517e-06	-1.215115e-05	+1.919092e-05	+9.617261e-06	+2.258759e-05	+nan
:	:	:	:	:	:	:	:
58725	+nan	+6.693075e-06	-3.187783e-06	+2.707895e-05	-1.676407e-05	+4.440954e-05	+5.273111e-06
58726	+nan	+6.687379e-06	-3.180825e-06	+2.708426e-05	-1.677650e-05	+4.442301e-05	+5.270004e-06

A-1.3.2 *"all_raw_clk_TA_(time).dat"*

■ *"/Pytool_TAgen/TAgen_Pred/TA_RESULT/ (time)/**all_raw_clk_TA_(time).dat**"*

Outline

- "***all_raw_clk_TA_(time).dat***" is the combined data of "***UTC(k)-clock$_i$***" and "***UTC(k)-TAp***".
- This data file is generated at the end of "*menu-2*" on demand.

Details / Rules of use

- The rules except for the last column data are the same as "***all_raw_clk_(time).dat***". *(p.52)*
- The last column:
  - The values of *weight, flag* and *cid* are not used in the program.
  - ***TAp*** calculation starts from "$MJD1 + T_{rate} + 1day$".

Note

※ The clock with "**#**" in "***clock_file_list.txt***" *(p.45)* is excluded.

Clock_file_list.txt

```
# sum of weights is normalized
to 1 in the program.
# flg1=1 means the reference
clock of phase comparison.
# filename     :  weight  : flg1
1 UTCk-clk1.dat :  1      : 0
2 UTCk-clk2.dat :  1      : 0
3 UTCk-clk3.dat :  1      : 0
4 UTCk-clk4.dat :  1      : 0
#5 UTCk-clk5.dat :  2     : 0
6 UTCk-clk6.dat :  2      : 1
7 UTCk-clk7.dat :  0      : 0
8 UTCk-clk8.dat :  4      : 0
```

UTCk-clk1.dat

```
56901.0000  +3.857038e-05
56902.0000  +3.858851e-05
      :           :
57999.0000  +5.576701e-05
58000.0000  +5.578885e-05
```

Pytool_TAgen/TAgen_pred/TA_RESULT/0731_155524/

all_raw_clk_TA_0731_155524.dat

TAp calculation starts from $MJD1 + T_{rate} + 1$ = 56901 +30 +1

fname	UTCk-clk1.dat	UTCk-clk2.dat	UTCk-clk3.dat	UTCk-clk4.dat	UTCk-clk6.dat	UTCk-clk7.dat	UTCk-clk8.dat	UTCk-TA
Weight	1.0	1.0	1.0	1.0	2.0	0.0	4.0	0.0
flag	0	0	0	0	1	0	0	3
cid	1	2	3	4	6	7	8	0
56901	+3.857038e-05	-6.184638e-06	-1.215249e-05	+1.918367e-05	+9.635976e-06	+2.258480e-05	+nan	+nan
:	:	:	:	:	:	:	:	:
56931	+3.909406e-05	-6.445632e-06	-1.210449e-05	+1.925565e-05	+9.033095e-06	+2.271062e-05	+nan	+nan
56932	+3.910934e-05	-6.452386e-06	-1.210189e-05	+1.925726e-05	+9.015229e-06	+2.271663e-05	+nan	+7.400889e-10
:	:	:	:	:	:	:	:	:
58726	+nan	+6.687379e-06	-3.180825e-06	+2.708426e-05	-1.677650e-05	+4.442301e-05	+5.270004e-06	+4.684945e-06

A-1.3.3 *"all_adev_(time).dat"*

■ *"/Pytool_TAgen/TAgen_Pred/TA_RESULT/ (time)/**all_adev_(time).dat**"*

Outline

- ***"all_adev_(time).dat"*** is the Allan deviation of the data of ***"all_raw_clk_(time).dat"*** or ***"all_raw_clk_TA_(time).dat"***.
- This data file is saved at the end of *Initial processing* or *"menu-2"* on demand.

Details / Rules of use

- 1st line : clock file name
- 2nd line : ***MJDspan*** : the period used in the *ADEV* calculation.
- 3rd line ~ : Allan deviation calculated by "***AllanTools***" [*Ref. 4*].
- *(time)* : ***MMDD_hhmmss*** : time stamp of data saving
  (***MMDD*** : month & date, ***hhmmss*** : hour & minute & second)

1

Note

※ If there are not sufficient samples, *ADEV* cannot be calculated and is expressed as '***nan***'. 2

※ If the data is outlier (> ***x_limit, nan***), the data sequence is divided using the outlier as boundary. 3

※ If the data has discontinuous and divided into several blocks, the *ADEV* is plotted for each block and marked with ***"*1", "*2"***,,, . 4

*("UTCk_clk2.dat" shows the ADEV of [block-0], and "UTCk_clk2.dat*1" shows the ADEV of [block-1].)*

Pytool_TAgen/TAgen_pred/TA_RESULT/0717_093410/

all_raw_clk_TA_0731_155524.dat

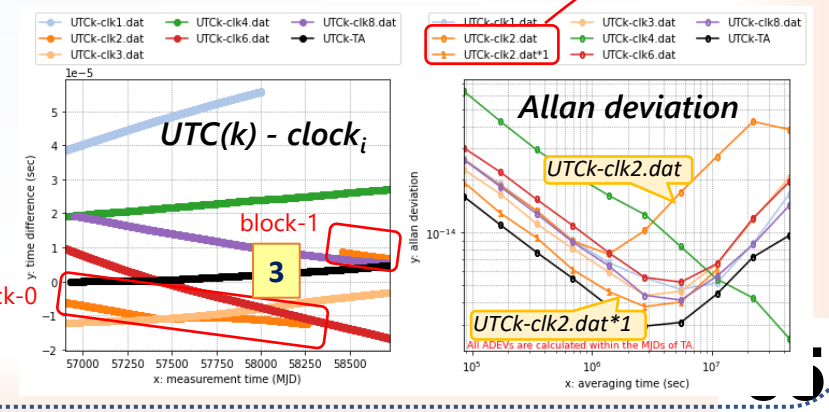
all_adev_0731_155524.dat

1

Tau(s)	UTCk-clk1.dat	UTCk-clk2.dat	UTCk-clk2.dat*1	UTCk-clk3.dat	UTCk-clk4.dat	UTCk-clk6.dat	UTCk-clk8.dat	UTCk-TA
MJDspan	56932-58000	56932-58260	58457-58726	56932-58726	56932-58726	56932-58726	56952-58726	56932-58726
86400	+2.593622e-14	+2.572997e-14	+1.914455e-14	+2.260179e-14	+6.282088e-14	+2.997311e-14	+2.576366e-14	+1.587481e-14
172800	+1.879802e-14	+1.838375e-14	+1.284166e-14	+1.647945e-14	+4.260514e-14	+2.195515e-14	+1.811728e-14	+1.101474e-14
:	:	:	:	:	:	:	:	:
11059200	+5.182775e-15	+2.695475e-14	+6.159347e-15	+6.611689e-15	+5.426419e-15	+6.643764e-15	+5.645967e-15	+4.474511e-15
22118400	+8.639638e-15	+4.250378e-14	+nan	+1.179217e-14	+4.242674e-15	+1.202004e-14	+8.546454e-15	+7.239207e-15
44236800	+1.646562e-14	+3.832687e-14	+nan	+2.051915e-14	+2.502899e-15	+1.933946e-14	+1.418509e-14	+9.593108e-15

4 2

fig_0717_093410.pdf 4



A-1.3.4 *"all_weight_(time).dat"*

■ *"/Pytool_TAgen/TAgen_Pred/TA_RESULT/ (time)/**all_weight_(time).dat**"*

Outline

- *"**all_weight_(time).dat**"* is the list of clock weights actually used in the ***TAp*** calculation.
- Each weight is normalized to be proportional to the initial value in the *"**clock_file_list.txt**". (p.45)*
- This data file is generated at the end of *"menu-2"* on demand.

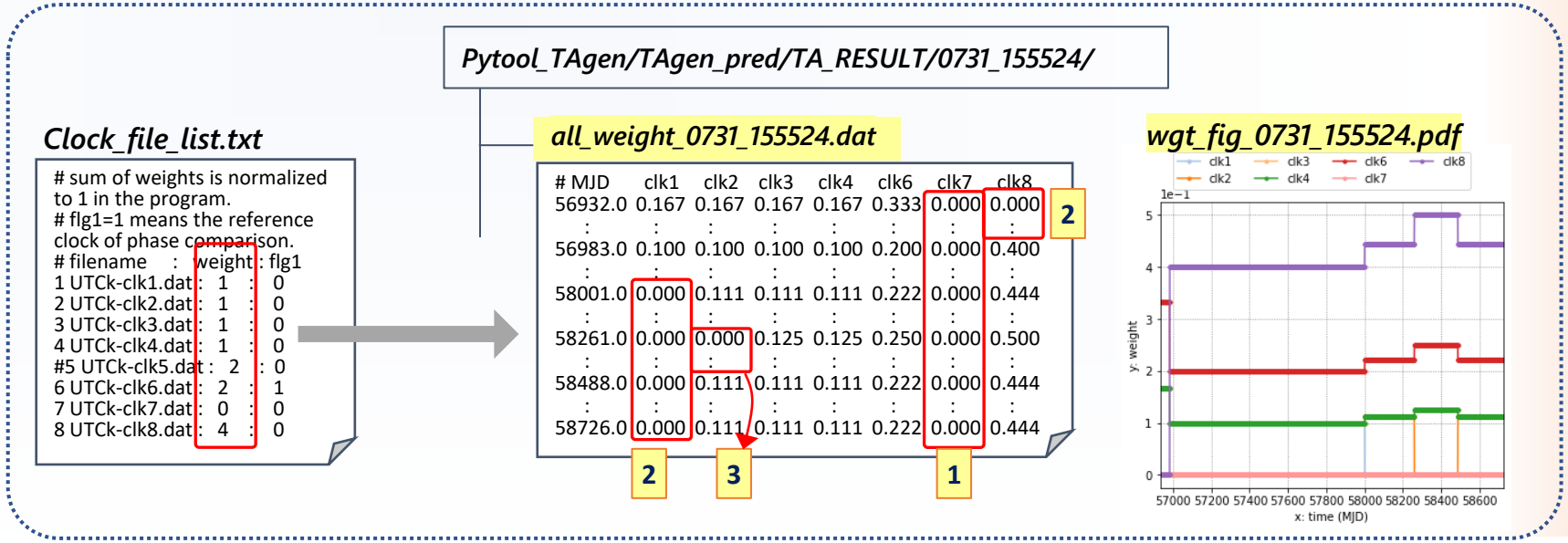
Details / Rules of use

- 1st line : clock name
- 2nd line ~ : clock weights actually used in the ***TAp*** calculation
- *(time)* : ***MMDD_hhmmss*** : time stamp of data saving
  (***MMDD*** : month & date ***hhmmss*** : hour & minute & second)

Note

※ The reason of zero weight:

1. Initial setting
2. Missing of clock data
3. Anomaly of clock data



A-1.3.5 *"log_(time).txt"*

■ *"/Pytool_TAgen/TAgen_Pred/TA_RESULT/ (time)/**log_(time).txt**"*

Outline

- *"**log_(time).dat**"* is the history of operation process.
- This file is generated at the end of *initial processing* or *"menu-2"* on demand.

Details / Rules of use

- Operated menu processes are serially added to this log file.
- The recorded parameters are actually used in the calculation.
- *(time)* : ***MMDD_hhmmss*** : time stamp of data saving
  (***MMDD*** : month & date ***hhmmss*** : hour & minute & second)

Note

※ The serially recording is reset when *"menu-0"* is selected.

Pytool_TAgen/TAgen_pred/TA_RESULT/0731_155524/

log_0731_155524.txt

```
# Data log at 2025.07.31_15h55m_24s
#
**0** Read and plot initial clock data and Allan deviation*****
::::: Input clocklist   : clock_file_list.txt
::::: Input clock file  : ['UTCk-clk1.dat', 'UTCk-clk2.dat', 'UTCk-clk3.dat', 'UTCk-clk4.dat', 'UTCk-clk6.dat', 'UTCk-clk7.dat', 'UTCk-clk8.dat']
::::: data time interval (t_intval): 86400(sec)
::::: Outlier limit (x_limit))     : 1000.0

**2** make a timescale with predicton *****
::::: saved TAxhat data : ./TA_GEN/0731_155500/(MJD)_TAxhat.dat
::::: saved TAx data : ./TA_GEN/0731_155500/TAx_(clock).dat
::::: interval for rate estimation (TArate_span)      : 30(days)
```

A-1.3.6 *"fig_(time).pdf"*

■ *"/Pytool_TAgen/TAgen_Pred/TA_RESULT/ (time)/**fig_(time).pdf**"*

Outline

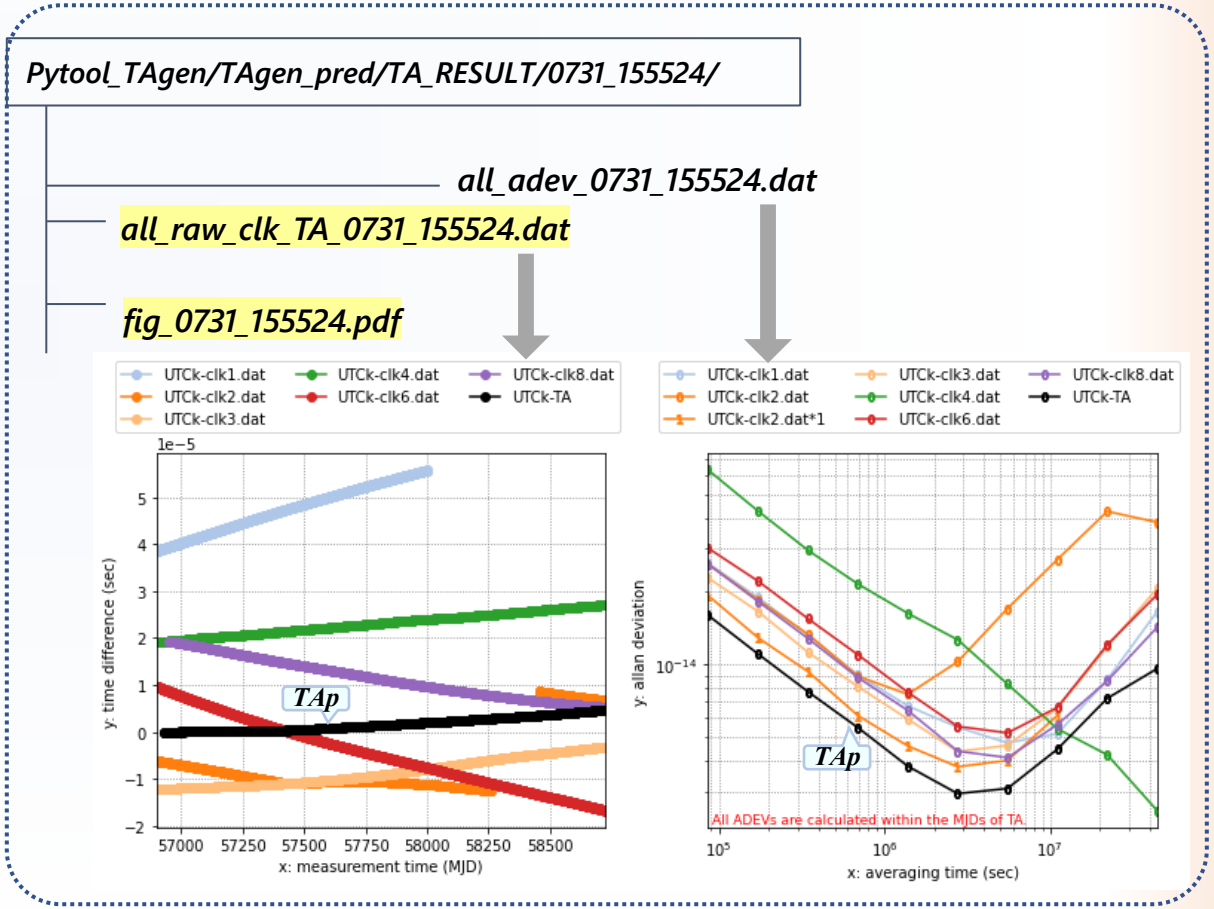
- *"**fig_(time).pdf**"* is the plots of time differences vs ***UTC(k)*** and their Allan deviations.
- This figure file is generated at the end of *Initial processing* or *"menu-2"*.

Details / Rules of use

- Left figure :
  - time differences of ***clock$_i$*** or ***TAp*** vs ***UTC(k)***
  - ***"all_raw clk_(time).dat"*** or ***"all_raw clk_TA_(time).dat"*** is used for this plot. *(p.52 or 53)*
- Right figure :
  - Allan deviations of the data for the left plot
  - ***"all_adev (time).dat"*** is used for this plot. *(p.54)*
- *(time)* : ***MMDD_hhmmss*** : time stamp of data saving
  (***MMDD*** : month & date ***hhmmss*** : hour & minute & second)

Note

- Only the clocks with non-zero weights are plotted.
- The black line shows the results of ***TAp***.



A-1.3.7 *"wgt_fig_(time).pdf"*

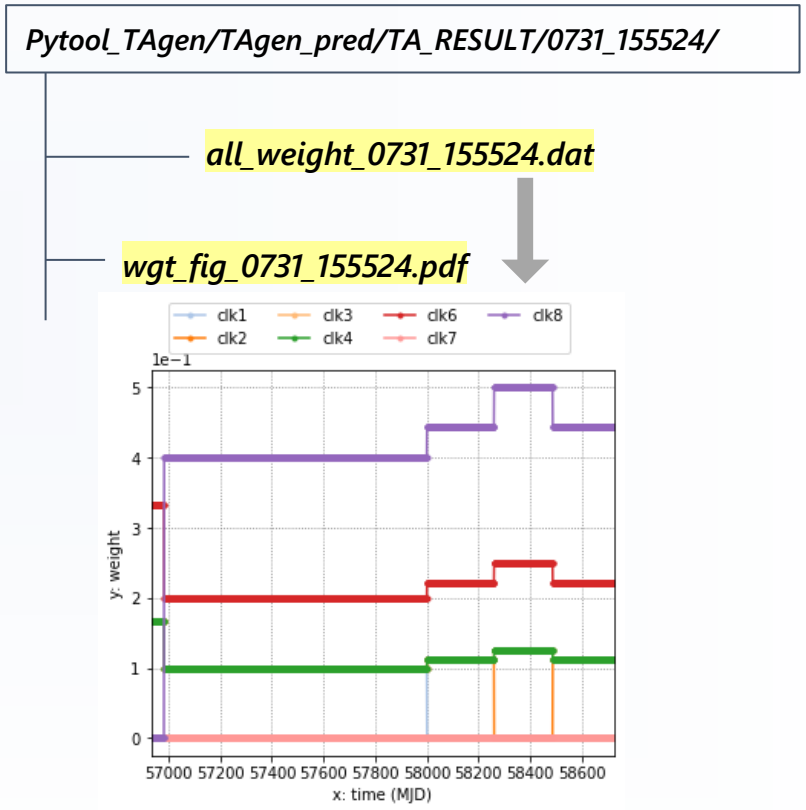
■ *"/Pytool_TAgen/TAgen_Pred/TA_RESULT/ (time)/**wgt_fig_(time).pdf**"*

Outline

- *"**wgt_fig_(time).pdf**"* is the plots of actually used clock weights.
- This figure file is generated at the end of *"menu-2"*.

Details / Rules of use

- ***"all_weight_(time).dat"*** (*p.55*) is used for this plot.
- *(time)* : ***MMDD_hhmmss*** : time stamp of data saving
  (***MMDD*** : month & date ***hhmmss*** : hour & minute & second)



Appendix-2： Detailed process of “*menu-2*”

A-2. Detailed process of *"menu-2"*

■ Outline of the process

" [*No.]" are shown in the following pages

[1] Preparation

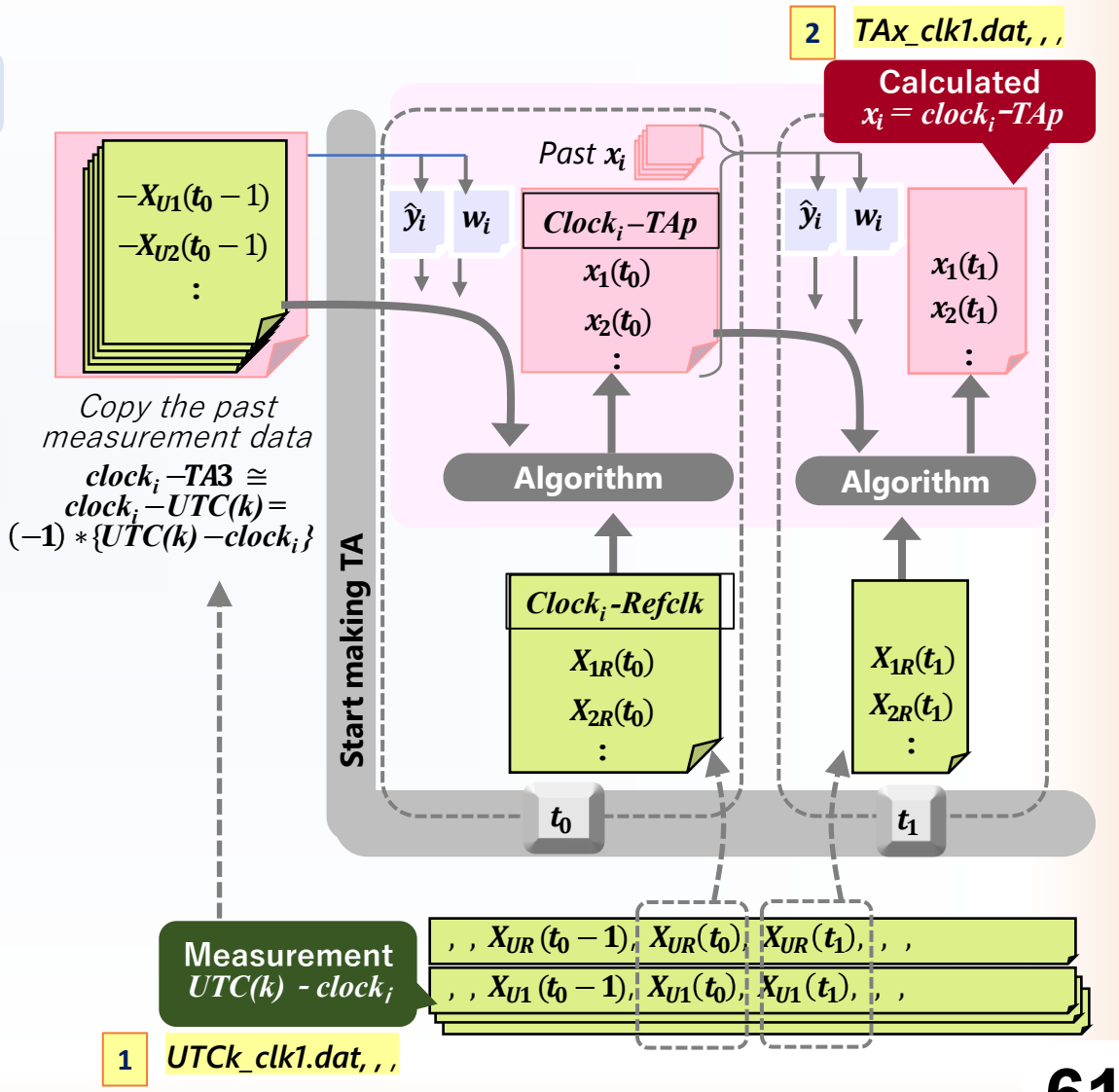
- Read the parameter tables and clock data ➔ [*1][*2][*3]
  - Parameter table : "***param_list.txt***"
  - Clock table : "***clock_file_list.txt***"
  - Measurement data file of "***UTC(k)-clock$_i$***" : "***UTCk_(clock$_i$).dat***" 1
- Prepare the initial parameters ➔ [*4][*6]
  - ***MJD*** range, parameters for the rate calculation
- Prepare the initial data set ➔ [*5][*7]
  - Calculate the "***clock$_i$ - Refclk***" from "***UTC(k)-clock$_i$***".
  - Copy the past "***clock$_i$ -UTC(k)***" to the data file of x_i :

[2] Daily calculation of *TAp*

- Iterative daily calculation ➔ [*8]
  - Set the normalized clock weights
  - Calculate the prediction $\hat{x}_i$ of each clock
  - Calculate the daily x_i and accumulate it to "***TAx_(clock$_i$).dat***" 2

[3] Plotting & saving the results

- Calculate "***UTC(k)-TAp***" for evaluation ➔ [*9]
- Plot and save "***UTC(k)-TAp***" and its *ADEV* ➔ [*10][*11]
  - The Result files are saved under *"./TA_RESULT/"*.



A-2. Detailed process of *"menu-2"*

[1] Preparation

[*1] Read the parameters from "*param_list.txt*" *(p.46)*

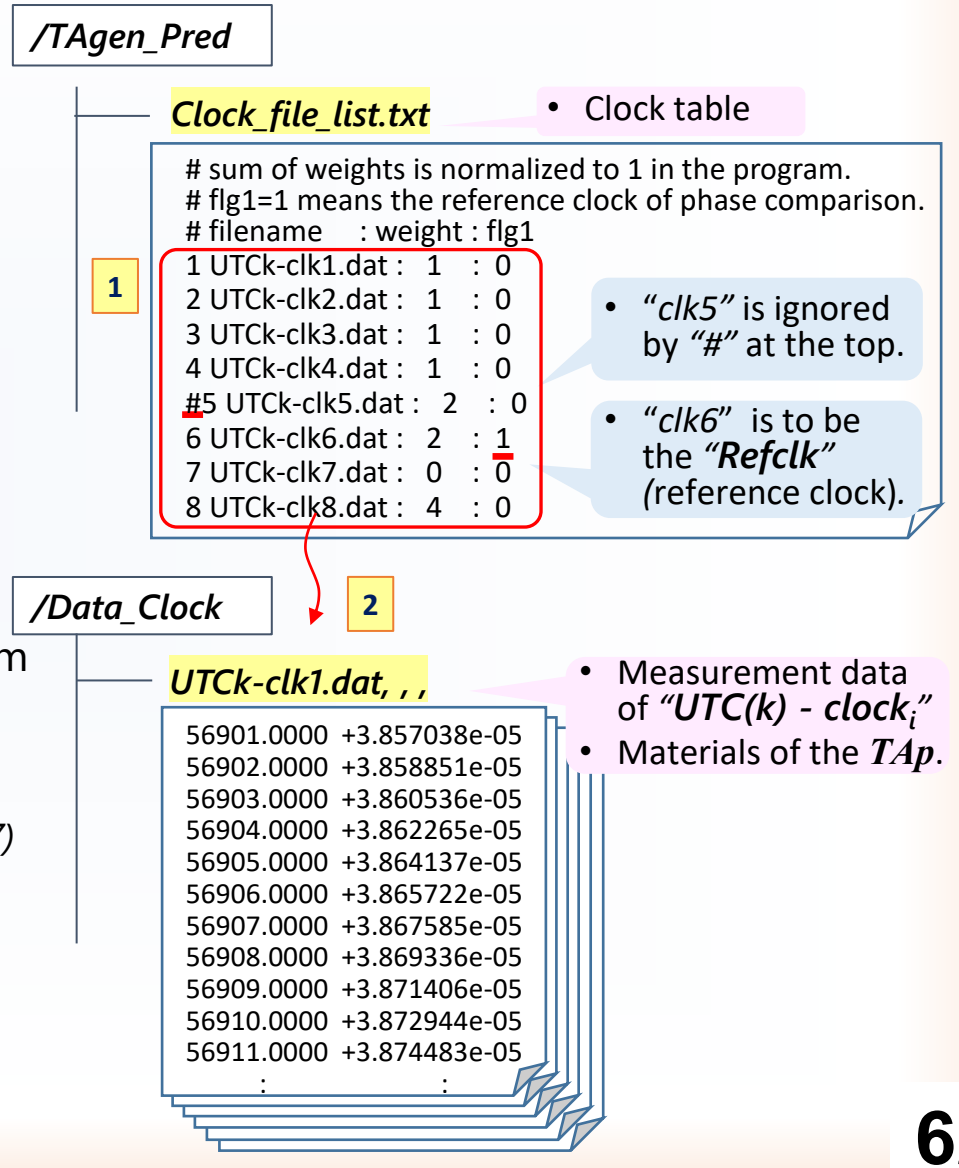
- Parameters that are fixed within the program but can be changed on demand are provided from this parameter list.

[*2] Read the parameters from "*clock_file_list.txt*" *(p.45)* 1

- This parameter list provides the conditions of each clock in the averaging process.
- The clocks expected to be used in the ***TAp*** calculation are assigned here with some weights.
- Clocks with ***weight=0*** are not ignored in this process but excluded from the averaging.

[*3] Read the assigned clock data "*UTCk-(clock$_i$).dat*" *(p.47)*

- These measurement clock data are used as the materials of ***TAp*** calculation. 2
- The assignment is done in *"**clock_file_list.txt**"*.



A-2. Detailed process of *"menu-2"*

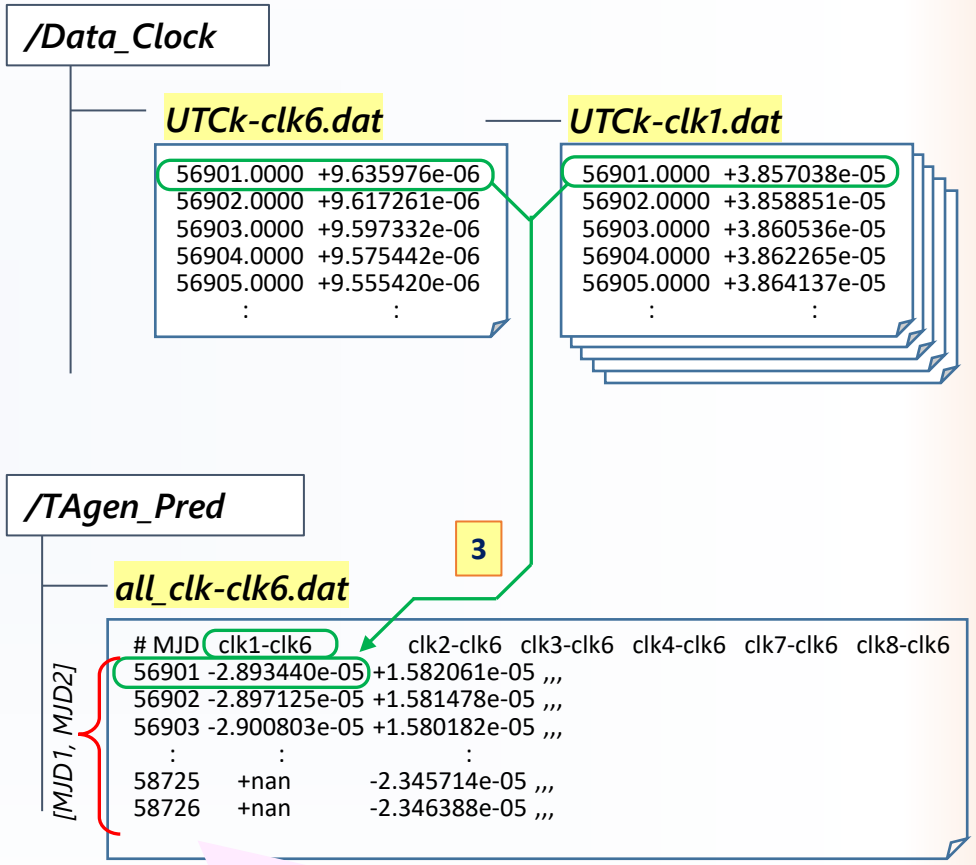
[1] Preparation (Cont.)

[*4] Fix the available MJD range for TAp calculation

- Set the MJD range $[MJD1, MJD2]$, as wide as possible.
- The MJD with enough number of clocks can be available.
  - ※ Mathematically, TAp can be calculated al least one clock.
  - ※ However, concentrating the weight on a single clock reduces the advantages of TAp.
  - ※ To avoid this problem, actual algorithms usually impose restrictions on the maximum weights and minimum number of clocks.
- If $[MJD1, MJD2]$ is defined by *"menu-1"* in advance, this range is adopted as a priority.

[*5] Prepare the measurement data set of $X_{is}(t_k)$ 3

- Calculate the measurement time difference of "$clock_i$ - $Refclk$" from "***UTCk-(clock$_i$).dat***".
- Results are saved in the file "***./all_clk-(Refclk).dat***". *(p.48)*



- Data file of all $X_{is}(t_k)$ = "$clock_i - clock6$" except for the $clock_i$ with the mark "#" in "clock_file_list.txt"

A-2. Detailed process of *"menu-2"*

[1] Preparation (Cont.)

[*6] Enter the parameter T_{rate}, as the span for calculating the rate, from the console 4

- The equation of rate $\hat{y}_i$ is given by *Eq.(12)* (*p.5*).
- Any value for T_{rate} is permitted within the calculation MJD span.

※ The recommended values (*10,30,90 days*) will be suitable to observe the effects of the different parameters.

[*7] Prepare the initial data set of x_i in "*TAx_(clock$_i$).dat*" to start the TAp calculation

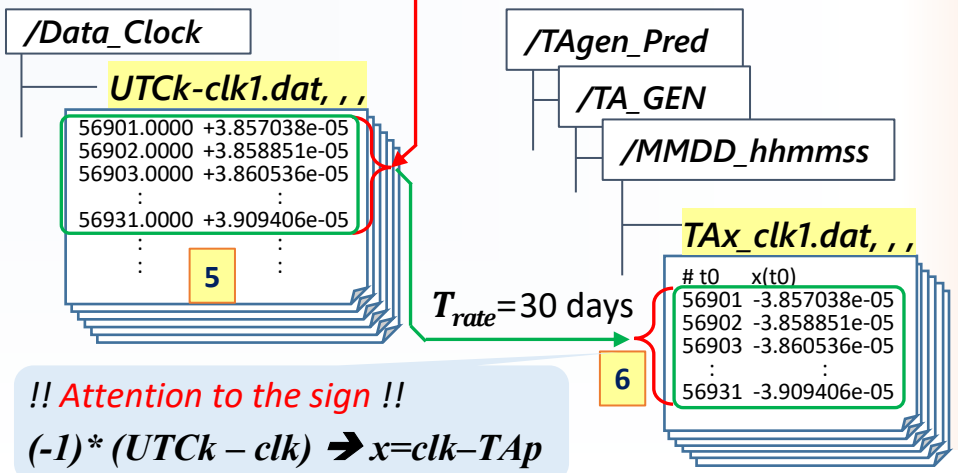
- As shown in *Eq. (10)* (*p.5*), the past x_i and $\hat{y}_i$ are required to start the TAp calculation. In particular, $\hat{y}_i$ needs the x_i over the latest T_{rate} days.
- As no past x_i are at the beginning, "***UTC(k)-clock$_i$***"s are used as substitutes for the initial x_is. 5

$$clock_i(t) - UTC(k)(t) \rightarrow clock_i(t) - TAp(t) \equiv x_i(t) \quad (t < start\ date)$$

- This "***clock$_i$ -UTC(k)***" within the latest past T_{rate} range are copied in "***TAx_(clock$_i$).dat***". *(p.50)* 6

```
=======================================
    Menu of the data processing
---------------------------------------
 0.  Plot the "UTC(k)-Clock" data
 1.  Assign the MJD range
 2.  Make TA with daily prediction
         and fixed clock weights
10.  Exit
=======================================
????? Input menu number ? >> 2

!!!!! UTCk-clk2.dat  includes outliers

*2** Set the span for clock rate calculation.
??2?? Span in days (try 10 or 30 or 90) = 30
```

4



!! Attention to the sign !!

(-1) (UTCk – clk) ➔ x=clk–TAp*

A-2. Detailed process of *"menu-2"*

[2] Daily calculation of TAp

[*8] Calculate $x_i(t_k)$ for "$TA_0 \le t_k \le MJD2$"

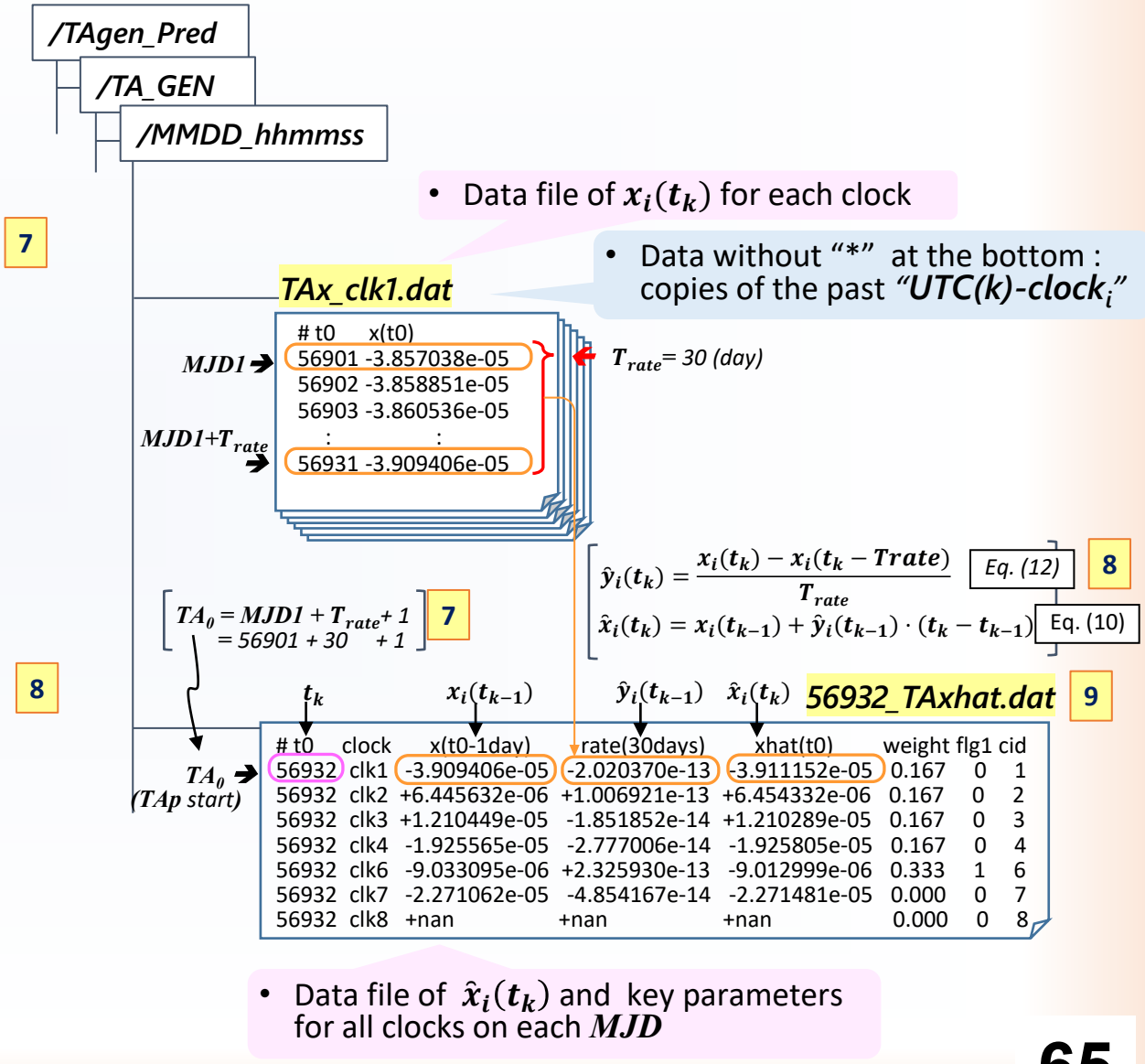
- The TAp calculation start date is "$TA_0 = MJD1 + T_{rate} + 1$". 7

[*8.1] Set the weights for the available clocks

- The given initial weights in "***clock_file_list.txt***" are used after normalization.
- If the measurement data for a clock is not normal (missing or outlier), the weight of this clock becomes zero for that MJD. 8

[*8.2] Calculate the prediction $\hat{x}_i(t_k)$

- $\hat{x}_i(t_k)$ is calculated by *Eq.(10) (p.5)* using the past x_i.
- $\hat{y}_i(t_{k-1})$ is calculated by *Eq.(12) (p.5)* as the slope of x_i over the latest past T_{rate} period.
- The daily calculated results are saved in the files "*(MJD)_TAxhat.dat*". *(p.49)* 9



A-2. Detailed process of *"menu-2"*

[2] Daily calculation of TAp (Cont.)

[*8] Calculate $x_i(t_k)$ for "$TA_0 \le t_k \le MJD2$" (Cont.)

[*8.3] Calculate the daily $x_i(t_k)$

- Calculate $x_i(t_k)$ of each clock by *Eq. (9)(11).* *(p.5)* 10
- The calculated time series of $x_i(t_k)$ are accumulated in the files "***TAx_(clock$_i$).dat*** with the mark "*****" at the end of the line. *(p.50)* 11

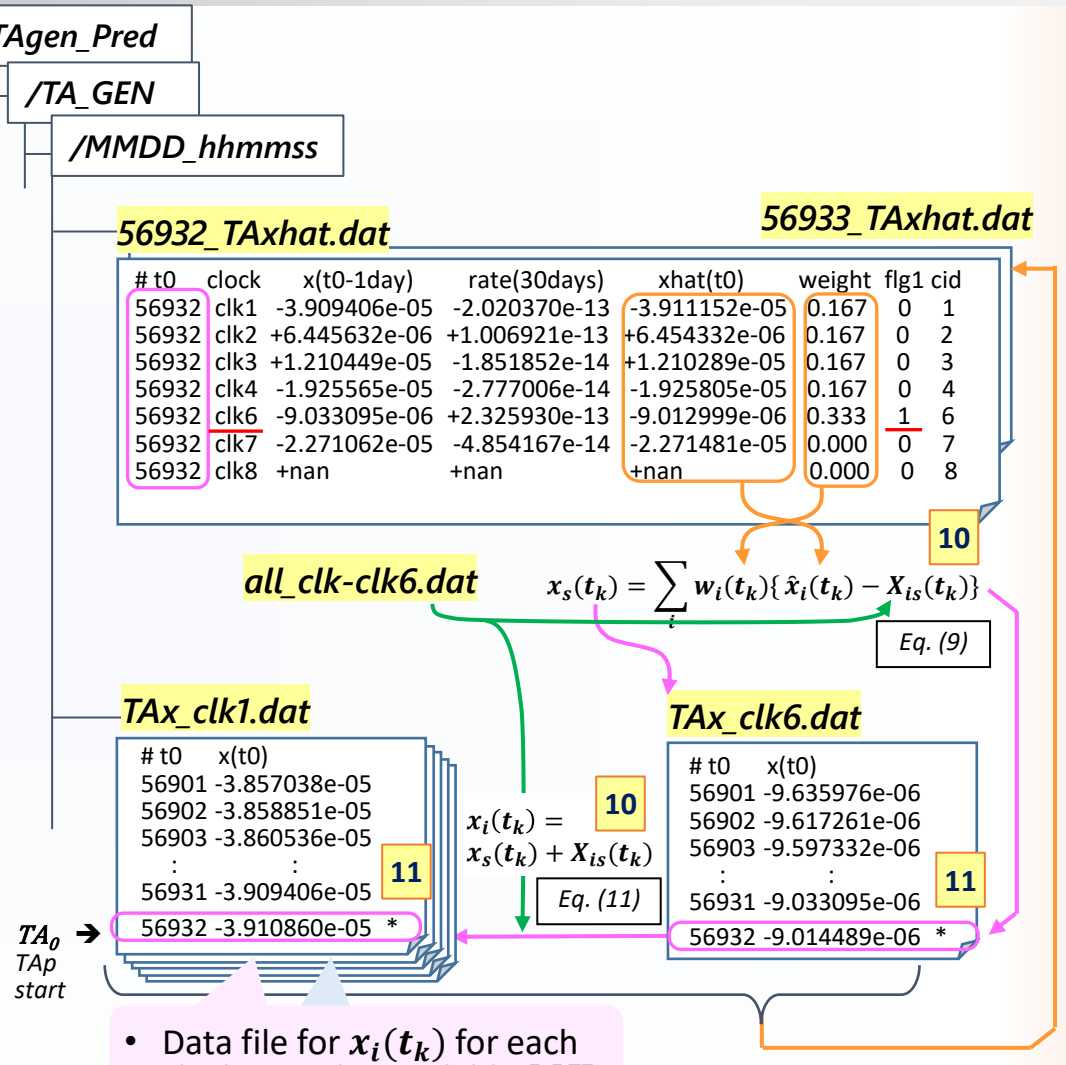
[3] Plotting and saving the results

[*9] Calculate "*UTC(k)-TAp*" for evaluation

- TAp can be compared with other timescales or clocks via common reference, for example ***UTC(k)***.
- "***UTC(k)-TAp***" is calculated by *Eq.(13).* *(p.6)*

$$UTC(k) - TAp = [UTC(k) - h_i] + [h_i - TAp] \quad \text{Eq. (13)}$$

"UTCk-(clock$_i$).dat" Calculated x_i



- Data file for $x_i(t_k)$ for each clock over the available MJD
- Data without "*" at the end of the line is the copy of the past "*UTC(k)-clock$_i$*".

A-2. Detailed process of *"menu-2"*

[3] Plotting and saving the results (Cont.)

[*10] Plot the results 12

- Left : time difference vs ***UTC(k)*** *(p.20, 57)*
- Middle : *ADEV*s of left data *(p.20, 57)*
- Right : clock weights *(p.58)*

• Only the clocks with non-zero weights are plotted.

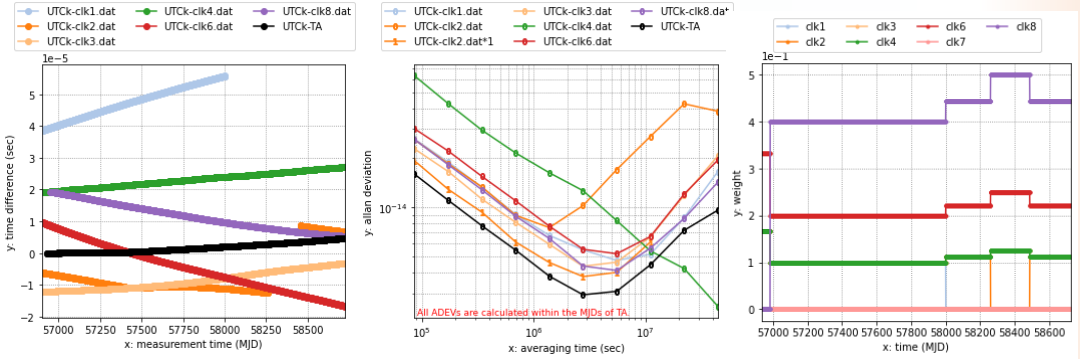
[*11] Save the results, if required *(p.21, pp.52-58))* 13

• Results are saved under ***"./TA_RESULT/(time)."***

- ***Log_(time).txt*** : history of the process
- ***fig _(time).pdf*** : all figures
- ***all_raw_clk_TA_(time).dat*** : "***UTC(k)-clock$_i$***" (left fig)
- ***all_adev_(time).dat*** : *ADEV* of "***UTC(k)-clock$_i$***" (right fig)
- ***all_weight_(time).dat*** : daily calculated weights
- ***wgt_fig_(time).dat*** : figure of all weights

```
????? Span in days (try 10 or 30 or 90) =  30

***** Calculating an timescale for  56932
                    :
***** Calculating an timescale for  58726
```

12




```
**2** make a timescale with prediction *****

????? Save figure? (y/n) = y
```

13

```
::::: save logfile =  ./TA_RESULT/0731_155524/log_0731_155524.txt
::::: save figfile =  ./TA_RESULT/0731_155524/fig_0731_155524.pdf
::::: save alldata =  ./TA_RESULT/0731_155524/all_raw_clk_TA_0731_155524.dat
::::: save ADEVdat =  ./TA_RESULT/0731_155524/all_adev_0731_155524.dat
::::: save weights =  ./TA_RESULT/0731_155524/all_weight_0731_155524.dat
::::: save wgt_fig =  ./TA_RESULT/0731_155524/wgt_fig_0731_155524.pdf

=======================================
    Menu of the data processing
---------------------------------------
 0.  Plot the "UTC(k)-Clock" data
 1.  Assign the MJD range
 2.  Make TA with daily prediction
         and fixed clock weights
 10.  Exit
=======================================
????? Input menu number ? >>
```