PHENOLOGY MODELING PLATEFORM (PMPConsole) – *Version 5.5.0.7*

15th of December, 2012
Isabelle CHUINE, Gauthier BOAGLIO

This document deals exclusively with technical considerations and procedures related to the usage of PMP in console mode : ***PMPConsole.exe****. For a deeper understanding on how PMP works and the philosophy behind it, please refer to the "PMP's general documentation"...*

MAIN POINTS :

>> Configuring one or more fits (generating the input XML file)
>> Running the program
>> Batch processing [TODO]
>> Configuring the program's defaults (Advanced users only)

Configuring one or more fits

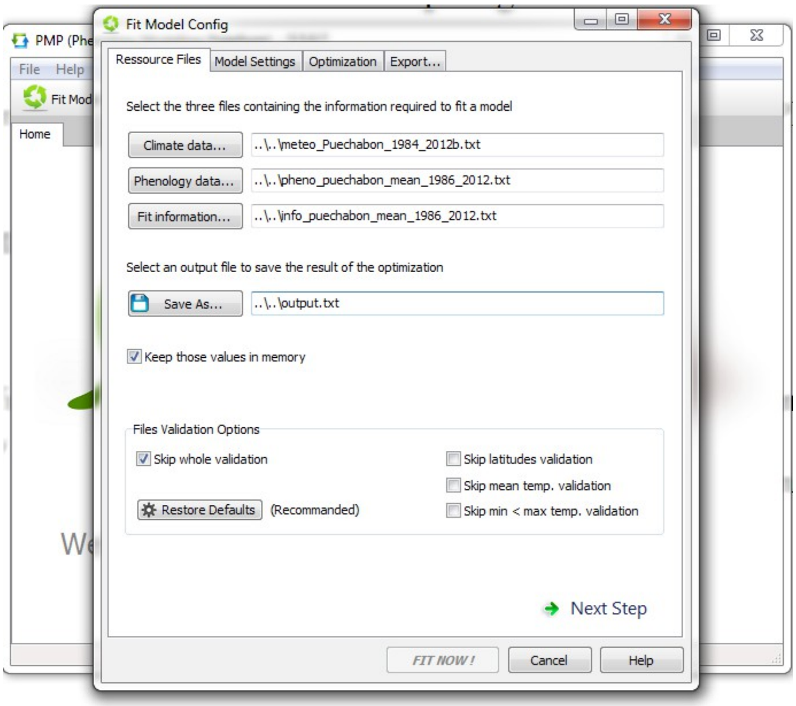
First of all, PMPConsole requires an input file. This file is unique, but can describe several fits to be run (at the same time). This way, only one instance of the program is run and handles the CPU usage for the best (each fit is run in a separated thread which means that the computing load is distributed on the computer's CPU cores).

Configuring > Using the graphical wizard

For ease of use, this file can entirely be generated using the Graphical version of PMP, by following those steps :

[*Generally speaking, see the* ***PMP's general documentation*** *on how to use the graphical interface, for each step*]

>> Configuring the input and output files



Select the three files containing the information required to fit a model

Climate data... ..\..\meteo_Puechabon_1984_2012b.txt

Phenology data... ..\..\pheno_puechabon_mean_1986_2012.txt

Fit information... ..\..\info_puechabon_mean_1986_2012.txt

Select an output file to save the result of the optimization

Save As... ..\..\output.txt

☑ Keep those values in memory

Files Validation Options

☑ Skip whole validation

□ Skip latitudes validation

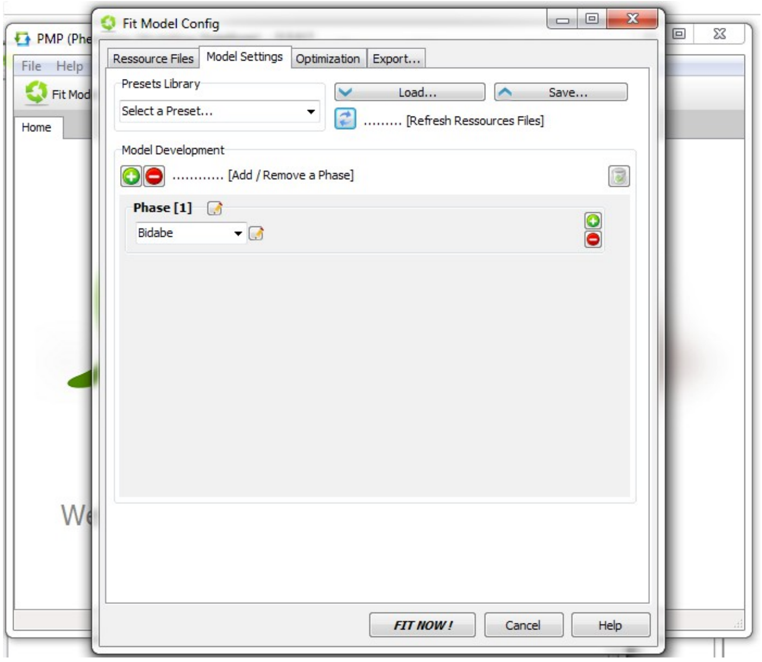
□ Skip mean temp. validation

□ Skip min < max temp. validation

(Recommanded)

Note : Do not worry about the relative paths, they will be turned to absolute ones at XML generating time (while pressing the “Save As” button / “Export” tab).

>> Configuring a model



Presets Library

Select a Preset...

.... [Refresh Ressources Files]

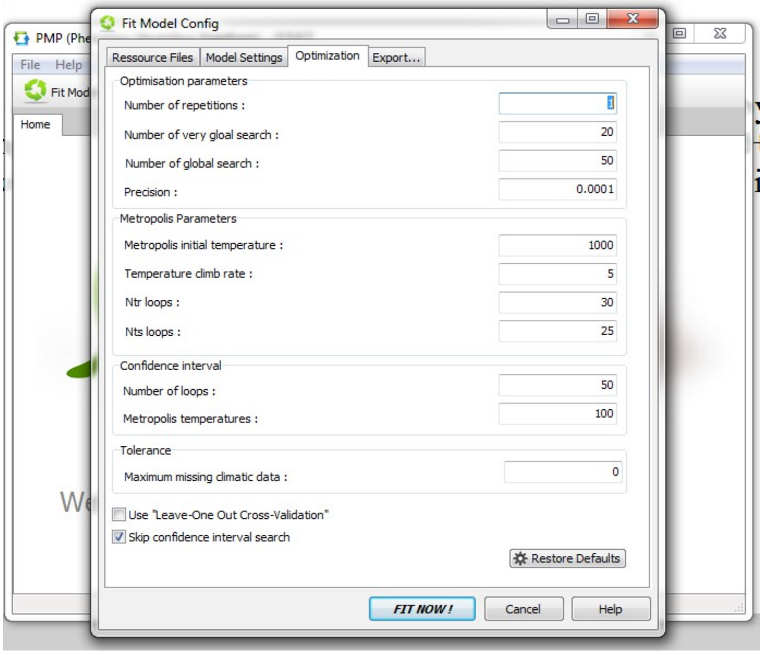
Model Development

.... [Add / Remove a Phase]

Phase [1]

Bidabe

>> Setting up the optimization initial parameters



Optimisation parameters

Number of repetitions : 1

Number of very gloal search : 20

Number of global search : 50

Precision : 0.0001

Metropolis Parameters

Metropolis initial temperature : 1000

Temperature climb rate : 5

Ntr loops : 30

Nts loops : 25

Confidence interval

Number of loops : 50

Metropolis temperatures : 100

Tolerance

Maximum missing climatic data : 0

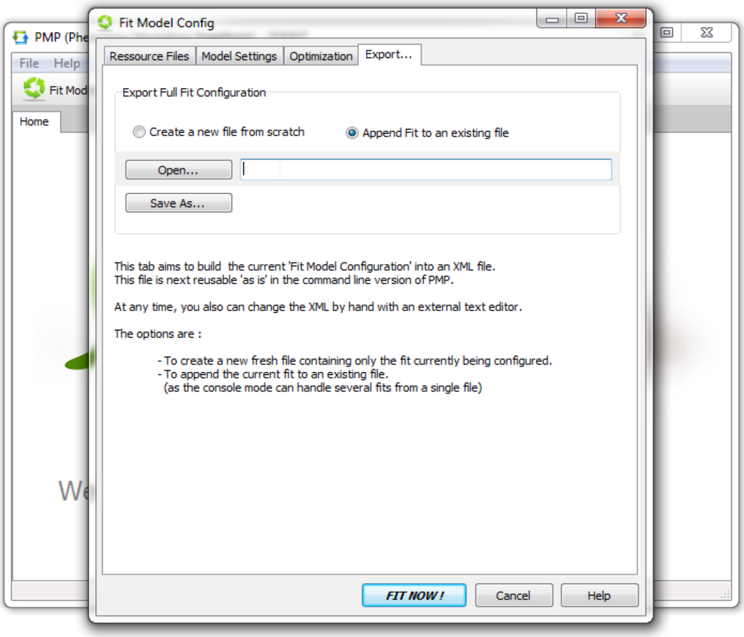
□ Use "Leave-One Out Cross-Validation"

☑ Skip confidence interval search

>> Finally, export as XML

Here are the important points: the final XML file can be generated

- "from scratch", that is to say that we obtain a new fresh file containing only one fit description (the one previously configured). This is "Option 1".
- "appended", which means that the user is asked for an existing XML file. PMP then, adds the newly configured fit at the following of the fits already present in the given file. And this is "Option 2".



Export Full Fit Configuration

○ Create a new file from scratch ● Append Fit to an existing file

This tab aims to build the current 'Fit Model Configuration' into an XML file.
This file is next reusable 'as is' in the command line version of PMP.

At any time, you also can change the XML by hand with an external text editor.

The options are :

- To create a new fresh file containing only the fit currently being configured.
- To append the current fit to an existing file.
  (as the console mode can handle several fits from a single file)

To make the second option more clear, here is the (truncated) structure of the generated file before and after having appended a fit:

Before:

```
<Fits>
  <Fit>
    <FilesPath>
      <MeteoFile value="D:\02_DEV\PMP\1\meteo_Puechabon_1984_2012b.txt"/>
      <PhenoFile value="D:\02_DEV\PMP\1\pheno_puechabon_mean_1986_2012.txt"/>
      <FitInfoFile value="D:\02_DEV\PMP\1\info_puechabon_mean_1986_2012.txt"/>
      <OutputFile value="D:\02_DEV\PMP\1\output.txt"/>
    </FilesPath>
    <Optimization>
             Some params...
    </Optimization>
    <Model name="Meta Model">
             Some model description...
      </Model>
  </Fit>
</Fits>
```

After:

```
<Fits>
  <Fit>
    <FilesPath>
      <MeteoFile value="D:\02_DEV\PMP\1\a_weather_file.txt"/>
      <PhenoFile value="D:\02_DEV\PMP\1\a_pheno_file.txt"/>
      <FitInfoFile value="D:\02_DEV\PMP\1\an_info_file.txt"/>
      <OutputFile value="D:\02_DEV\PMP\1\an_output.txt"/>
    </FilesPath>
    <Optimization>
             Some params...
    </Optimization>
    <Model name="Meta Model">
             Some model description...
       </Model>
  </Fit>
  <Fit>
    <FilesPath>
      <MeteoFile value="D:\02_DEV\PMP\2\another_weather_file.txt"/>
      <PhenoFile value="D:\02_DEV\PMP\2\another_pheno_file.txt"/>
      <FitInfoFile value="D:\02_DEV\PMP\2\another_info_file.txt"/>
      <OutputFile value="D:\02_DEV\PMP\2\another_output.txt"/>
    </FilesPath>
    <Optimization>
             Some params for last configured fit...
    </Optimization>
    <Model name="Meta Model">
             Some model description for last configured fit...
       </Model>
  </Fit>
</Fits>
```

Configuring > Manually

By the other hand, one can find more handy to configure the input XML file by hand. This operation can be performed using any plain text editor (We would recommend Notepad++ for its syntactic coloration making easier the reading and editing of XML files).
To fully understand how to modify this file, here is the description of each of its inner tags :

Preliminary Note : In the following, ids are used to ensure that the chaining of the phases, functions and parameters are correct, but are not mandatory, since the XML nodes are read in their order of appearance in the file.

```
<Fits>
  <Fit>
    <FilesPath>
      <MeteoFile value="D:\02_DEV\PMP\1\meteo_Puechabon_1984_2012b.txt"/>
      <PhenoFile value="D:\02_DEV\PMP\1\pheno_puechabon_mean_1986_2012.txt"/>
      <FitInfoFile value="D:\02_DEV\PMP\1\info_puechabon_mean_1986_2012.txt"/>
      <OutputFile value="D:\02_DEV\PMP\1\output.txt"/>
    </FilesPath>
    <Optimization>
      <NbRepetitions value="1"/>
      <NbVeryGlobalSearch value="20"/>
      <NbGlobalSearch value="50"/>
      <MetPrecision value="0.0001"/>
      <MetropolisTemperature value="1000"/>
      <TemperatureClimbRate value="5"/>
      <NtrLoops value="30"/>
      <NtsLoops value="25"/>
      <NbLoopsConfInt value="50"/>
      <TemperatureConfInt value="100"/>
      <MaxMissingTemp value="0"/>
      <SkipFisher value="True"/>
      <LeaveOneOut value="False"/>
    </Optimization>
    <Model name="Meta Model">
      <Phases>
        <Phase id="0" usedToFit="True" phenoDataColumn="1">
          <T0 mode="0" value="0" t0DataColumn="-1" photomode="0"/>
          <TempSamplingFunc value=""/>
          <SStar fixed="False" value="0"/>
          <PhaseLinkMode value="0"/>
          <OverlapFunction name="Negative Exponential">
            <Param id="0" fixed="False" name="z" value="0.00000"/>
            <Param id="1" fixed="False" name="w" value="0.00000"/>
          </OverlapFunction>
          <Functions>
            <Function id="0" name="GDD" weatherDataColumn="0">
              <UseDayLengthAsVar value="False"/>
              <VarLinkMode value="0"/>
              <Weight value="1"/>
              <Param id="0" fixed="False" name="Tb" value="0.00000"/>
            </Function>
          </Functions>
        </Phase>
      </Phases>
    </Model>
  </Fit>
</Fits>
```

For a given fit (tag <Fit>), the following must be configured :

>> The tag <FilesPath> must contain four sub-tags and the values of those tags must be different for each fit.

<MeteoFile> : The path of the input file containing the weather data.
<PhenoFile> : The path of the input file containing the phenological data.
<FitInfoFile> : The path of the file specifying which part of the input data will be used for fitting.
<OutputFile>: The path of the file containing the results as soon as the fitting process is over.

>> Then comes the "optimization" information (tag <Optimization>) :

[*Those parameters are mainly related to the Metropolis algorithm settings and are more deeply described in PMP's general doc.*]

- <NbRepetitions> : The number of repetitions to perform in order to achieve the fit.
- <NbVeryGlobalSearch> : The number of loops over which the Metropolis algorithm will try to fit in a very large scope.
- <NbGlobalSearch> : The number of loops over which the Metropolis algorithm will try to fit in a more restrained scope.
- <MetPrecision> : The decimal precision used by the Metropolis algorithm.
- <MetropolisTemperature> : The initial temperature for the algorithm.
- <NtrLoops> : The number of "ntr" loops the algorithm will run over.
- <NtsLoops> : The number of "nts" loops the algorithm will run over.
- <NbLoopsConfInt> : The number of loops used to compute the confidence interval (has no effect unless <SkipFisher> is set to "*False*").
- <TemperatureConfInt> : The initial temperature used to compute the confidence interval (has no effect unless <SkipFisher> is set to "*False*").
- <LeaveOneOut> : If the process has to be run in "Leave-One-Out" mode or not.

>> Finally comes the description of the model to be fitted itself (tag <Model>) :

A model is described as a list of phases, and each phase as follows :

A <Phase> has two main attributes :

- If it is used to fit (usedToFit).
- Which column it must use (phenoDataColumn) from the phenological information file (index of the column in the file **minus 3**).

And the following additional parameters :

- <T0> tells how to obtain the the starting date value.
  <T0 mode="*0*" value="*0*" t0DataColumn="*-1*" photomode="*0*"/>, where mode can be (0:fitted, 1:end of the previous phase, 2:fixed to a constant, 3:fixed to a varying date from the phenological information file), value is the fixed value of "T0" if mode is set to "*2*", t0DataColumn is the column to be used from the phenological information file if mode is set to "*3*" (index of the column in the file **minus 3**), and photomode tells if the "T0" value has to be converted into a photoperiod (0:No, 1:Yes, increasing, 2:Yes, decreasing).

- `<TempSamplingFunc>` is the function used for sampling the temperatures. Its `value` can be `""` (None: the sampling is set to "daily"), or one of the following if the sampling has to be "hourly" : `"tempheur"` (triangular algorithm) or `"tempheur_shark"` (shark algorithm).
- `<Sstar>` is the critical sum for this phase and can be fitted if `fixed` is set to *`"False"`* , or fixed otherwise, in which case its `value` must be specified.
- `<PhaseLinkMode>` is the way the phase is linked to the previous one in the chain. This is, of course, ignored for the very first function of the chain. The value can be 0:Sequential, 1:Alternating, 2:Overlap Multiplication, 3:Overlap Composition. If the chosen mode is not Sequential, an overlapping function must be specified (tag `<OverlapFunction>`).
- `<OverlapFunction>` can be any of the "overlap compatible" functions (See PMP's general documentation, again). Each of those functions has at least one parameter (Excepted *`"Identity"`* which has none). Each parameter (tag `<Param>`) can be fixed (*`fixed="True"`*) to a specified `value`, or fitted.
- `<Functions>` : the chain of functions for the phase. Contains one or more `<Function>` sub-tags.

A `<Function>` has two main attributes :

- Its `name` (which can be any of the functions available in PMP).
- Which column it must use (`weatherDataColumn`) from the weather data file (index of the column in the file **minus 4**) as input variable.

And the following additional parameters :

- `<UseDayLengthAsVar>` : If one want to use the day length instead of the weather data grabbed from the weather data file (as mentioned previously, using the `weatherDataColumn`) as input variable.
- `<VarLinkMode>` : The way the function is linked (operator) to the previous one in the chain of functions of the phase (ignored for the very first function in the chain). Its `value` can be 0:Multiplicative, 1:Additive, 2:Composition.
- `<Weight>` : Used to apply multiplicative factor to the result of the function (default is 1.0).
- `<Param>` : Each function has at least one parameter. Each parameter can be fixed (*`fixed="True"`*) to a specified `value`, or fitted (See the PMP's general documentation for a full description of the parameters of each function).

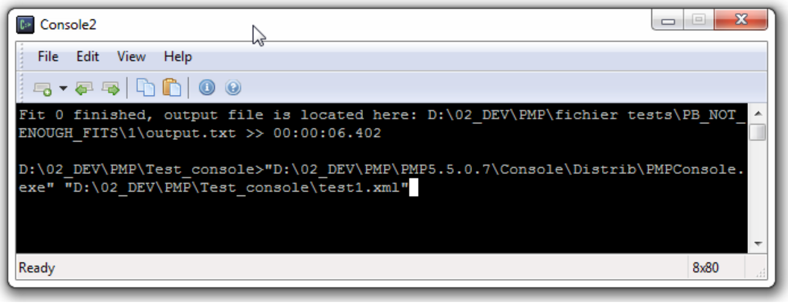
Running the program

First requirement : you must open a Windows Command Line Prompt, then type the path of the PMPConsole.exe executable, or simply drag it from its location on your hard drive and drop it into the console.

Then, their are two ways of running PMPConsole :

>> By passing directly the XML input file as an argument to the executable (+ press enter)
[*This mode is the one, if you planned to run PMP on a plateform which leaks a graphical interface or if you want to execute PMPConole from a batch file*]

Note : Again, this can be done by a simple drag & drop of the XML in the console (must press the space bar once before).

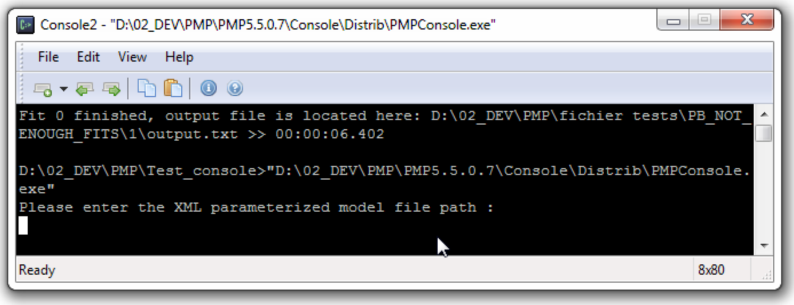


```
Fit 0 finished, output file is located here: D:\02_DEV\PMP\fichier tests\PB_NOT_
ENOUGH_FITS\1\output.txt >> 00:00:06.402

D:\02_DEV\PMP\Test_console>"D:\02_DEV\PMP\PMP5.5.0.7\Console\Distrib\PMPConsole.
exe" "D:\02_DEV\PMP\Test_console\test1.xml"
```

>> By using the “quite” interactive mode

First run the program in a shell without passing any parameter (+ press enter), then type or paste the path of the XML input file :


Console2 - "D:\02_DEV\PMP\PMP5.5.0.7\Console\Distrib\PMPConsole.exe"
File
Edit
View
Help
Ready
8x80

```
Fit 0 finished, output file is located here: D:\02_DEV\PMP\fichier tests\PB_NOT_
ENOUGH_FITS\1\output.txt >> 00:00:06.402

D:\02_DEV\PMP\Test_console>"D:\02_DEV\PMP\PMP5.5.0.7\Console\Distrib\PMPConsole.
exe"
Please enter the XML parameterized model file path :
```

Batch processing

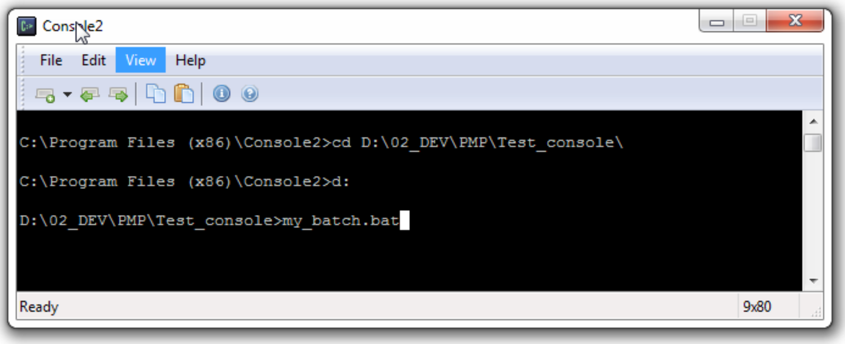
Their is also a more handy way which allows to run several instances of PMPConsole, and avoids to re-type (or re-drop) the whole paths each time we run the same fits : “batch” processing.

>> First, create a new empty text file, with the **“.bat”** extension
>> Second, put one line per instance of PMPConsole you planned to run. Example :

```
"D:\02_DEV\PMP\PMP5.5.0.7\Console\Distrib\PMPConsole.exe" "D:\02_DEV\PMP\Test_console\test7.xml"
"D:\02_DEV\PMP\PMP5.5.0.7\Console\Distrib\PMPConsole.exe" "D:\02_DEV\PMP\Test_console\test8.xml"
```

Note : The paths can be relative to your .bat file's location, not necessarily absolute.

>> Third, save the file
>> Fourth, go to your .bat location and run it (this will execute its content one line after another) :



```
C:\Program Files (x86)\Console2>cd D:\02_DEV\PMP\Test_console\

C:\Program Files (x86)\Console2>d:

D:\02_DEV\PMP\Test_console>my_batch.bat
```

Configuring the program's defaults

(Advanced users only)

You should do this only if you pretty well know what you're doing !

PMPConsole comes with a configuration file which can be found at "**System\Presets\Optimisation\functions.xml**" and looks like this :
*[And we recommend to not remove or modify the backuped one "**functions.xml.bak**", you could need it to restore the first one]*

```
<Functions>

      <Function name="Wang">
            <Param name="Topt" id="0" value="5" step="15" min="0" max="30"/>
            <Param name="Tmin" id="1" value="1" step="15" min="-40" max="10"/>
            <Param name="Tmax" id="2" value="40" step="15" min="20" max="50"/>
      </Function>

      <Function name="Triangular">
            ...
      </Function>

      ...

</Functions>
```

This file contains a description for each parameter of each function available in PMP. Their `id` and their `name` should not be changed, but their initial `value` and range (`min` and `max`) do.
The `name` of the functions should not be changed either.

Enjoy ;-)