

Queuing Data for Analysis

Draft Version

Version 2.001

Copyright Michael L. Johnson, 2009

All Rights Reserved

Queuing Data for Analysis

One approach to analyzing large groups of data without the requirement of analysis of individual data series is that of queuing batch jobs. This command is located under the File menu as Queue Batch Jobs, as depicted in Figure 1.

It should be noted here that under this command, every file listed for analysis will be analyzed in precisely the same manner.

The first step in the process is to be certain that all of the concentration data series files are in the same subdirectory and that the algorithm is pointing at that directory by using the command **File**→**Change Default Folder (CD)**.

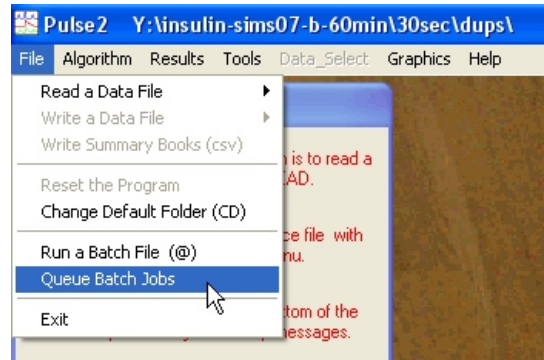


Figure 1

Every time the program is executed, a command history file is automatically created by the program. This history file includes all commands utilized during the previous fit. Thus, you will want to select a data set and run that file in the precise manner in which you intend to run all of your files. It is imperative that after the analysis is complete that the “Store Results” box is selected.

After the analysis is complete and results stored, go under **Results** and click on **Command History** and then select **Rename**. Select a file name that you will remember such as Queue.tpl and be certain that you store the renamed file in your working directory with a file extension being *.tpl. This extension stands for template and will be automatically recognized by the queuing system.

The step requires that you go to the **File** menu and select “**Queue Batch Jobs**”. Select the box indicated "Select a Template File" (as shown in Figure 2) and then select the file which you created.

After an appropriate file has been selected, a new box will appear which indicates "**Edit a Template File**". Select that box and a new text editor window will appear as shown in Figure 3. There are several specific items which must be modified before the analysis may be properly performed.

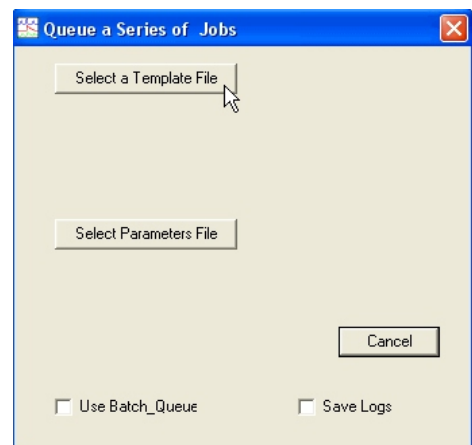


Figure 2

```
?! command file created by hormone_20090118 on 01/23/2009 at 10:09:15
Algorithm AutoDecon
READ_FIX
Y:\insulin-sims07-b-60min\30sec\dups\60m30sdup-0001.fix
<EOF>
DECON_AutoInit
DECON_Automatic
    5.000000E-02
    1
        1.06440
    1
        0.250000
    1
    1
        4.67118
STORE_RESULTS
WRITE_NATIVE Y:\insulin-sims07-b-60min\30sec\dups\60m30sdup-0001_.dat
```

Figure 3

The initial line of the file is simply a “comment” line as will remain undisturbed as will most other lines. The second line indicates which algorithm has been selected and the third, the type of file being read (in this example it is a *.fix formatted set). The next line (the 4th line) includes the path for where the data has been stored along with the actual file name of the “test” file (in this case: 60m30sdup-0001.fix).

As the program reads and performs exactly what is included in the *.tpl file, it is necessary to now substitute the file name (but not extension) with a code recognized by the program .

These substitutions include:

%~n1 (this is the most commonly used substitution meaning we are only substituting the name of the file)

%~dpn1 (directory/path/name)

%~d1 (directory)

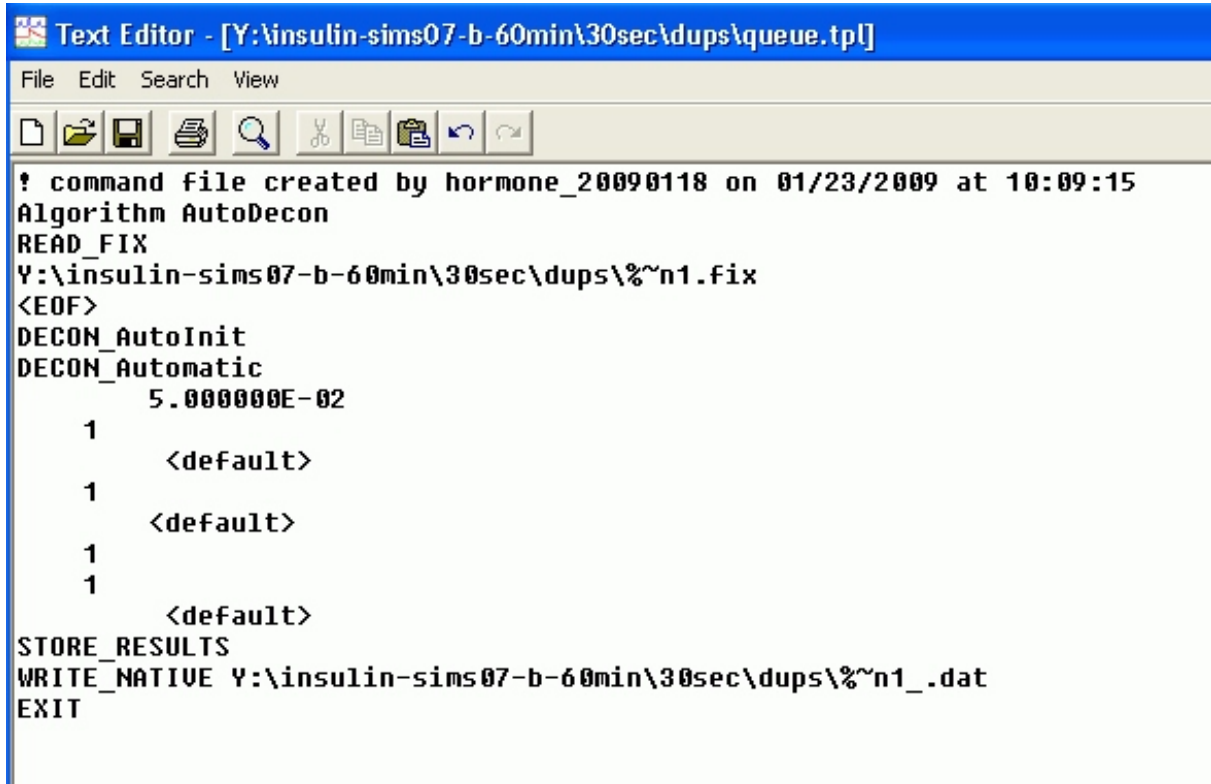
%~p1 (path)

%~nx1 (name/extension)

%~x1 (extension)

For the example provided here, the name will be substituted where the data file is

READ_FIX and where it is Written (i.e. WRITE_NATIVE). Figure 4 shows the edited files with the correct nomenclature. An important substitution required is for the values calculated by the algorithm which will change with various data files. Those lines should now contain <default>. This allows the program to accept the preliminary values for that particular data series for the initial fit. Notice now that we have added an EXIT on the last line of the file to ensure that the program

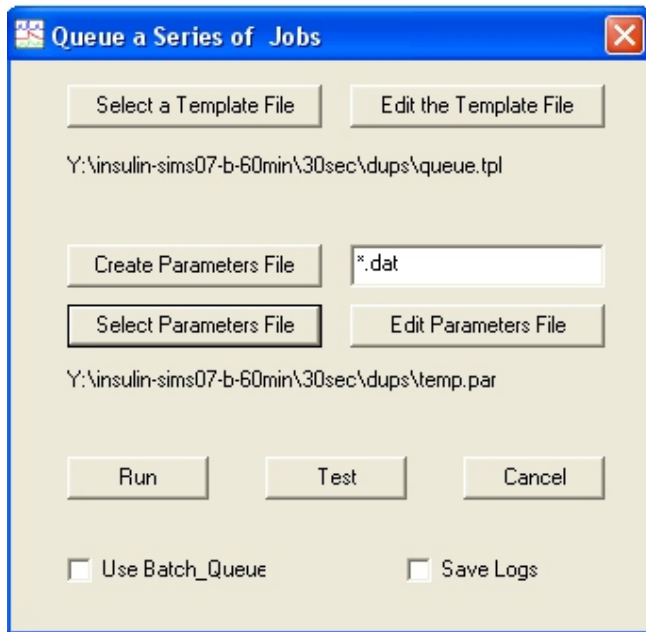


```
Text Editor - [Y:\insulin-sims07-b-60min\30sec\dups\queue.tpl]
File Edit Search View
! command file created by hormone_20090118 on 01/23/2009 at 10:09:15
Algorithm AutoDecon
READ_FIX
Y:\insulin-sims07-b-60min\30sec\dups\%~n1.fix
<EOF>
DECON_AutoInit
DECON_Automatic
    5.000000E-02
    1
    <default>
    1
    <default>
    1
    1
    <default>
STORE_RESULTS
WRITE_NATIVE Y:\insulin-sims07-b-60min\30sec\dups\%~n1_.dat
EXIT
```

Figure 4

does not get snagged up in some infinite loop. Now, go under File and Save this edited file.

The next step is to now to Select a Parameter file. This file will simply be a listing of every file with the same extension in the current subdirectory. Click on the Box Select a Parameter File and again select a name fairly easy to remember with an extension this time ending in *.par (indicating that this is a parameter file). For this example the name temp.par has been used.



The screen should now appear as it does in Figure 5. There is currently no data in the file since we have only created a file name. To create the data file, go into the white bounding box and select what format of data in which you have stored your files. For this example, the *.dat must be changed to *.fix since the data is in a fixed format. After selecting the data format, simply click on the “Create Parameters File” and the file should now contain all of the files with the selected extension. This can be verified by clicking on “Edit Parameters File”.

The user may now Test the queuing process (meaning only the first file will be actually run as a test) or may be bold and select Run.

Figure 5

Now the data files will be analyzed automatically by the program without further user intervention!