QMBCFN

Version 1.74

Stefan Meinel

June 7, 2013
(minor correction on June 27, 2023)

Contents

1 Introduction

QMBCFN is an application with a graphical user interface intended for correlated least-χ^2 fitting. QMBCFN allows very general types of fits, where every single fit point has its own arbitrary function of the parameters, entered by the user as a string. Thus, it is useful for applications where there are not too many fit points, and where the flexibility of having an individual function for each point is useful. QMBCFN allows both unconstrained fits and Bayesian fits with Gaussian priors for the fit parameters.

1.1 Compiling QMBCFN

1.1.1 Software Requirements

- For OS X, the necessary developer tools can be obtained for example by installing Apple Xcode.
- Qt version 4.x. On most Linux distributions, this is already installed or available through the package management. Note that the *development* packages are also needed (these usually have `-dev` or `-devel` in the package name). Free open source editions for Linux/X11, OS X and Windows are also available from http://qt-project.org/
- GNU Scientific Library, version ≥ 1.13 (see http://www.gnu.org/software/gsl/). On most linux distributions this is available through the package management. Note that the *development* packages are also needed (these usually have `-dev` or `-devel` in the package name). For OS X, installation via http://www.macports.org/ is recommended.

1.1.2 Adjusting `QMBCFN.pro` (if necessary)

This step is only necessary if libraries such as the GNU Scientific Library are installed in unusual locations (in the following example, `/path/to/mylib` and `/path/to/myinclude`). In this case, go to the directory with the QMBCFN source code, and add the following lines at the end of the file `QMBCFN.pro`, for each library where needed:

```
LIBS += -L/path/to/mylib
QMAKE_CXXFLAGS += -I/path/to/myinclude
```

1.1.3 Compiling on Linux

From the directory with the QMBCFN source code, run

```
qmake
```

(the `qmake` program is part of Qt). This generates a Makefile suitable for the system. Then run

```
make
```

This will produce an executable file called `QMBCFN`.

1.1.4 Compiling on OS X

From the directory with the QMBCFN source code, run

```
qmake -spec macx-g++
```

(the `qmake` program is part of Qt). This generates a Makefile suitable for the system. Then run

```
make
```

This will produce an application folder called `QMBCFN.app`.

1.2 Running QMBCFN

QMBCFN takes as optional command-line argument the name of a `mbcfn` file to be opened (see Sec. 4.5.1 for information on QMBCFN session files):

1.2.1 Running on Linux

```
./QMBCFN [inputfile.mbcfn]
```

1.2.2 Running on OS X

```
open QMBCFN.app [inputfile.mbcfn]
```

Alternatively, after `.mbcfn` files have been associated with QMBCFN, use

```
open inputfile.mbcfn
```

2 Fit Algorithm

The task solved by QMBCFN is the following: we are given M real-valued data points $\overline{y}_i$ $(i = 1...M)$, and their symmetric, positive-definite covariance matrix C_{ij}. [1] For each data point we are given a smooth, real-valued model function $f_i(\mathbf{a})$, depending on P real-valued parameters $a_1, a_2, ..., a_P$. The aim is to find those values of the parameters that minimise the following χ^2 function,

$$\begin{aligned} \chi^2(\mathbf{a}) &= \sum_{i=1}^{M}\sum_{j=1}^{M} W_{ij}\left[\overline{y}_i - f_i(\mathbf{a})\right]\left[\overline{y}_j - f_j(\mathbf{a})\right] \\ &\quad + \sum_{p=1}^{P}\frac{(a_p - A_p)^2}{\sigma_{A_p}^2}, \end{aligned} \tag{1}$$

where, depending on the chosen settings, W is either equal to C^{-1} (full inverse of data covariance matrix C), or a truncated inverse of C (see Sec. 2.1). The term in the second line of (1) is optional and provides the Gaussian priors used for Bayesian fits. In addition to the terms shown in this section, QMBCFN also allows an arbitrary additional function of the paramaters to be added to the definition of χ^2 (see Sec. 4.4).

QMBCFN uses the Levenberg-Marquardt method for minimization. It requires the derivatives of χ^2 with respect to the parameters:

$$\frac{\partial \chi^2}{\partial a_p} = -2\,\beta_p \tag{2}$$

$$\frac{\partial^2 \chi^2}{\partial a_p \partial a_q} = 2\,\alpha_{p\,q} \tag{3}$$

[1] The bar indicates that $\overline{y}_i$ may originate from an average over a set of N measurements $\{y_i^n\}$; in that case, $\overline{y_i} = \frac{1}{N}\sum_{n=1}^{N} y_i^n$ and $C_{ij} = \frac{1}{N(N-1)}\sum_{n=1}^{N}(y_i^n - \overline{y_i})(y_j^n - \overline{y_j})$. QMBCFN directly uses $\overline{y}_i$ and C_{ij} as its input, irrespective of their origin.

with

$$\begin{aligned}\beta_p(\mathbf{a}) &= \sum_{i=1}^{M}\sum_{j=1}^{M} W_{ij}\frac{\partial f_i(\mathbf{a})}{\partial a_p}\big[\overline{y}_j - f_j(\mathbf{a})\big] \;-\; \frac{a_p - A_p}{\sigma^2_{A_p}}, &(4)\\ \alpha_{p\,q}(\mathbf{a}) &= \sum_{i=1}^{M}\sum_{j=1}^{M} W_{ij}\frac{\partial f_i(\mathbf{a})}{\partial a_p}\frac{\partial f_j(\mathbf{a})}{\partial a_q} \;+\; \frac{\delta_{p\,q}}{\sigma^2_{A_p}}\\ &\quad -\sum_{i=1}^{M}\sum_{j=1}^{M} W_{ij}\frac{\partial^2 f_i(\mathbf{a})}{\partial a_p \partial a_q}\big[\overline{y}_j - f_j(\mathbf{a})\big]. &(5)\end{aligned}$$

For the Levenberg-Marquardt iteration, a new matrix $\widetilde{\alpha}$ is defined with some $\lambda > 0$:

$$\begin{aligned}\widetilde{\alpha}_{p\,p} &= (1+\lambda)\alpha_{p\,p} &(6)\\ \widetilde{\alpha}_{p\,q} &= \alpha_{p\,q} \quad (p \neq q). &(7)\end{aligned}$$

With the starting parameters $\mathbf{a}$ and an initial value for λ, say $\lambda = 0.001$, the iteration proceeds as follows:

1. Solve
$$\sum_{q=1}^{P} \widetilde{\alpha}_{p\,q}(\mathbf{a})\,\delta a_q = \beta_p(\mathbf{a}) \tag{8}$$
for $\delta\mathbf{a}$

2. If $\chi^2(\mathbf{a}+\delta\mathbf{a}) \geq \chi^2(\mathbf{a})$, set $\lambda = b\lambda$ with $b \sim 10$, and go back to 1, otherwise set $\lambda = \lambda/b$, set $\mathbf{a} = \mathbf{a} + \delta\mathbf{a}$ and go back to 1.

The procedure is stopped on the first occasion of χ^2 *decreasing* by less than some very small amount, say 10^{-3}. Then, the parameter covariance matrix

$$[\alpha^{-1}(\mathbf{a})]_{p\,q}. \tag{9}$$

is computed. The standard error estimate for parameter a_p is then given by $\sqrt{[\alpha^{-1}(\mathbf{a})]_{p\,p}}$.

Note that $\alpha_{p\,q}(\mathbf{a})$ has a contribution with second derivatives of the model functions f_i, shown in (5), which is typically small for a good fit because of the factor $\big[\overline{y}_j - f_j(\mathbf{a})\big]$. QMBCFN allows the user to choose whether this term will be included by the fitter or replaced by zero (see Sec. 4.3). This choice can be made separately for the minimization process and for the final computation of the parameter covariance matrix.

2.1 Truncated inverse of data covariance matrix

2.1.1 Singular value decomposition

If the user chooses one of the "SVD" settings (see Sec. 4.3), the singular value decomposition of C is calculated,

$$C = U\,S\,V^T, \qquad S = \mathrm{diag}(\sigma_1, ..., \sigma_D) \qquad \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_D \geq 0, \qquad UU^T = VV^T = \mathbf{1}, \tag{10}$$

and the matrix W in (1) is defined as follows, with some $M' < M$:

$$W = (U\,S'\,V^T)^{-1} = V\,S'^{-1}\,U^T, \qquad S'^{-1} = \mathrm{diag}(1/\sigma_1, ..., 1/\sigma_{M'}, \underbrace{0, ..., 0}_{M-M'}). \tag{11}$$

The number of singular values $M - M'$ which are eliminated is referred to as the "SVD cut". It is either specified directly by the user, or chosen based on the size of the singular values (see Sec. 4.3).

2.1.2 Diagonal approximation (uncorrelated fit)

QMBCFN also provides an option to ignore the off-diagonal entries of the data covariance matrix C. If the user chooses this setting, the matrix W used in (1) is taken to be

$$W = \mathrm{diag}(1/C_{11},\ 1/C_{22},\ ...,\ 1/C_{MM}). \tag{12}$$

This corresponds to an uncorrelated fit.

3 Data File Format

3.1 "Single fit" file format

In QMBCFN, the index i labelling the M functions f_i and data points $\overline{y}_i$ is actually string-valued. That is, every function has its unique name i, which is an arbitrary string (but without whitespaces). The data file format is plain text, with the following entries: the first line of the file contains the number of data points M (a positive integer). This is followed by M lines, where each line contains a name i and the corresponding floating-point value $\overline{y}_i$. The order does not matter, but every name i must appear exactly once. Then, a second block follows, which contains the *non-zero* elements of the data covariance matrix C_{ij}: each line of the second block consists of two names i and j, and the floating-point value C_{ij}. The order of the lines within the block does not matter. The matrix C_{ij} is automatically symmetrized while reading the data file, that is, whenever an element C_{ij} is read from the file, the program also assigns $C_{ji} = C_{ij}$. It does not matter whether C_{ji} also appears in the file or not. An example "Single fit" data file is shown below:

```
4
f1    1.3423
f2   -3.1234
g1    0.3442
g2    0.7363
f1 f1    0.034
f1 f2   -0.0024
f2 f2    0.0083
f1 g1   -0.001
g1 g1    0.0156
g1 g2    0.0091
g2 g2    0.0241
```

In this case, $M = 4$, and $i = (\texttt{f1},\ \texttt{f2},\ \texttt{g1},\ \texttt{g2})$. The covariance matrix is

$$C = \begin{pmatrix} 0.034 & -0.0024 & -0.001 & 0 \\ -0.0024 & 0.0083 & 0 & 0 \\ -0.001 & 0 & 0.0156 & 0.0091 \\ 0 & 0 & 0.0091 & 0.0241 \end{pmatrix}.$$

3.2 "Multiple fits, fixed covariance matrix" file format

This file format provides multiple samples of data, but with a fixed covariance matrix. It is similar to the "Single fit" file format (Sec. 3.1), except that there are N_S samples for the set of values $\overline{y}_i$ (rather than just 1). As before, the first line of the file contains the number of data points M, but the second line now contains the number of samples N_S. This is followed by N_S blocks, each containing M values

$\overline{y}_i$. In addition to the index i, the entries are labelled by a predecing sample index that runs from 1 to N_S. At the end of the file, the (fixed) covariance matrix follows, in exactly the same way as in 3.1. Here is an example with $M = 4$ and $N_S = 3$:

```
4
3
1  f1   1.3423
1  f2  -3.1234
1  g1   0.3442
1  g2   0.7363
2  f1   1.6671
2  f2   0.1234
2  g1   0.6442
2  g2   0.4363
3  f1   1.5426
3  f2  -1.1574
3  g1   0.623
3  g2  -0.1756
f1 f1   0.034
f1 f2  -0.0024
f2 f2   0.0083
f1 g1  -0.001
g1 g1   0.0156
g1 g2   0.0091
g2 g2   0.0241
```

The ordering of the index i in each sample block must match that of the first block (QMBCFN will give an error message if that is not the case).

When using the "Multiple fits, fixed covariance matrix" file format, QMBCFN will perform N_S fits, and write the resulting stream of fit parameters to output files.

3.3 "Multiple fits, varying covariance matrix" file format

This file format is similar to the format described in Sec. 3.2, except that each sample gets its own values for the covariance matrix. The N_S blocks for $\overline{y}_i$ have the same format as in Sec. 3.2. However, the entries of the covariance matrices are now additionally labelled by the sample index. An example is shown below:

```
4
3
1  f1   1.3423
1  f2  -3.1234
1  g1   0.3442
1  g2   0.7363
2  f1   1.6671
2  f2   0.1234
2  g1   0.6442
2  g2   0.4363
3  f1   1.5426
3  f2  -1.1574
3  g1   0.623
3  g2  -0.1756
1  f1 f1   0.034
```

```
1  f1 f2  -0.0024
1  f2 f2   0.0083
1  f1 g1  -0.001
1  g1 g1   0.0156
1  g1 g2   0.0091
1  g2 g2   0.0241
2  f1 f1   0.034
2  f2 f2   0.0043
2  g1 g1   0.0196
2  g1 g2   0.0041
2  g2 g2   0.0211
3  f1 g1  -0.003
3  g1 g1   0.026
3  g1 g2   0.00491
3  g2 g2   0.3241
3  f1 f1   0.086
3  f2 f2   0.0073
```

Note that unlike the blocks for $\overline{y}_i$, the ordering of the covariance matrix entries is arbitrary, and different samples can have different numbers of non-zero entries. QMBCFN will simply continue reading covariance matrix entries till the end of the file.

As in Sec. 3.2, QMBCFN will perform N_S fits, and write the resulting stream of fit parameters to output files.

4 Using QMBCFN

4.1 Defining the fit functions, parameters, and constants

In the "Model Functions" Tab of QMBCFN, select the number M of functions f_i / data points $\overline{y}_i$. Then, for each data point, enter the name i of the function, and the function $f_i(\mathbf{a})$ itself. The number of functions and their names must match the data file (in the above example, $M = 4$ and the function names are `f1`, `f2`, `g1`, `g2`). The functions can depend on an arbitrary number of parameters a_p and constants. The model functions are constructed using the built-in operations

```
+, -, *, /,
(, ),
exp(...), log(...),
sin(...), cos(...), tan(...),
sinh(...), cosh(...), tanh(...),
arcsin(...), arccos(...), arctan(...),
sqr(...), sqrt(...).
```

The names of the parameters and constants are alpha-numeric strings, must not contain spaces, and must not begin with a number character (the underscore "_" may be used).

The total number of fit parameters (P) is selected in the "Parameters/Derivatives" Tab of QMBCFN. For each parameter, the parameter name must be entered. If numerical differentiation for the first-order derivatives is disabled (see Sec. 4.3), additionally the first-order derivatives of all functions with respect to that parameter must be entered, expressed using the same available built-in operations as above (second-order derivatives, if needed, will always be calculated numerically). Also selected in the

"Parameters/Derivatives" Tab is the number $P' \leq P$ of parameters to be subtracted from the number of degrees of freedom (M) when calculating the χ^2 per effective degrees of freedom[2],

$$\frac{\chi^2}{\mathrm{dof}} = \frac{\chi^2}{M - P'}.$$

For non-Bayesian fits, the default value is $P' = P$, while for Bayesian fits, the default value is $P' = 0$.

The number of constants is selected in the "Constants" Tab of QMBCFN. For each constant, the name and its numerical value must be entered.

4.2 Selecting the active functions

(Introduced in QMBCFN 1.30) In the "Model Functions" Tab of QMBCFN, each row has a checkbox controlling whether the function/data point i will be included in the fit. Note that the data file must always contain *all* data points i listed in the Model Functions Tab (and only those), irrespective of whether some points are deactivated. However, deactivated data points are not used in the fit. This means that the corresponding rows and columns in the data covariance matrix C are removed (before the inversion), and all sums over i and j in the calculation of χ^2 etc. exclude the deactivated values of the indices.

4.3 Fit Settings Tab

The Tab "Fit Settings" contains input fields that control the fit algorithm. The user can activate or deactivate the Gaussian priors by clicking on "Activate Bayesian constraints". When "Use numerical differentiation for first derivatives" is activated, the first-order derivatives in Eqs. (4) and (5) are approximated using a symmetric difference operation, using

$$\frac{\partial f_i(a_1, ..., a_P)}{\partial a_p} \approx \frac{f_i(..., a_p + h, ...) - f_i(..., a_p - h, ...)}{2h}. \tag{13}$$

The step size h used in (13) can also be specified by the user in the input field "Numerical differentiation step size". Note that making the steps size too large may give a poor approximation of the derivative, while making it too small may cause significant round-off errors. Because QMBCFN uses double-precision floating-point arithmetic, the default value $h = 10^{-8}$ is thought to be a good compromise between the two possible causes of error. To be safe, the user should check that the fit results are independent of the step size to the desired precision (or compare the results to the case where numerical differentiation is disabled and the symbolic derivatived are entered by hand in the "Parameters/Derivatives Tab").

The options "Use second derivatives for parameter covariance matrix" and "Use second derivatives for minimization" control whether the term with the second derivatives of the model functions f_i, shown in (5), is included in the calculation of $\alpha_{p\,q}$ or is replaced by zero. The first option only affects the final calculation of $\alpha_{p\,q}$ after the minimum of χ^2 was found. Hence, it affects the parameter covariance matrix $[\alpha^{-1}]_{p\,q}$, but not the central values of the fitted parameters. It is recommended that this option be enabled in order to get the most accurate estimate of the parameter covariance matrix. Conversely, the second option only affects the intermediate calculations of $\widetilde{\alpha}_{p\,q}$ during the search for the minimum of χ^2. It is usually recommended that this option be disabled, because it may destabilize the iterations.

[2] If an SVD cut is used, M is replaced by M', as defined in Sec. 2.1.1.

The second derivatives of the model functions, if enabled, will always be computed numerically. If the user has supplied the first-order derivatives symbolically and the option "Use numerical differentiation for first derivatives" is *disabled*, the second derivatives will be calculated using

$$\frac{\partial^2 f_i(a_1, ..., a_P)}{\partial a_p\, \partial a_q} \approx \frac{\partial_{a_q} f_i(..., a_p + h, ...) - \partial_{a_q} f_i(..., a_p - h, ...)}{2h}, \tag{14}$$

where $\partial_{a_q} f_i$ are the symbolic first-order derivatives.

On the other hand, if the option "Use numerical differentiation for first derivatives" is *enabled*, the second derivatives will be calculated using

$$\begin{aligned} \frac{\partial^2 f_i(a_1, ..., a_P)}{\partial a_p\, \partial a_q} \approx \frac{1}{4h'^2} \Big[& f_i(..., a_p + h', ..., a_q + h', ...) - f_i(..., a_p + h', ..., a_q - h', ...) \\ & - f_i(..., a_p - h', ..., a_q + h', ...) + f_i(..., a_p - h', ..., a_q - h', ...) \Big], \end{aligned} \tag{15}$$

where the new step size $h' = \sqrt{h}$ is used to minimize round-off errors. Note that (14) is numerically more precise than (15).

The Fit Settings tab also contains input fields for the starting value for λ, the lambda factor b, the tolerance for decreases in χ^2 at which the iteration stops, and the maximum number of fitting iterations (see Sec. 2 for the meaning of these parameters).

The selector "(Pseudo-)Inversion method for data covariance matrix" controls the definition of the matrix W in Eq. (1). The options are:

- "LU decomposition": full inversion, $W = C^{-1}$
- "SVD with fixed cut": pseudo-inverse using singular value decomposition (see Sec. 2.1.1), remove given number $M - M'$ of smallest eigenvalues
- "SVD with EV ratio cut": pseudo-inverse using singular value decomposition (see Sec. 2.1.1), remove eigenvalues that are smaller than the specified fraction of the largest eigenvalue σ_1
- "SVD with EV value cut": pseudo-inverse using singular value decomposition (see Sec. 2.1.1), remove eigenvalues smaller than some given value
- "Diagonal only (uncorrelated fit)": keep only diagonal elements in data covariance matrix (see Sec. 2.1.2)

The option "Use Gaussian random priors for multiple fits" is only relevant for Bayesian fits in combination with the "Multiple fits" file formats (see Secs. 3.2 and 3.3). See the end of Sec. 4.5.5 for more details.

4.4 Additional term in chi sqr

The Tab "Additional term in chi^2" can be used to add an arbitrary function of the fit parameters to χ^2. This option is only intended for special applications, for example forcing two fit parameters to be close to each other by adding the square of their difference, divided by some width, to χ^2.

The elementary operations that can be used for defining such a function are the same as in Sec. 4.1. The function may contain any of the fit parameters defined in the Parameters/Derivatives Tab. However, the constants defined in the "Constants" tab can **not** be used here. Instead the "Additional term in chi^2" tab provides the option of defining new constants solely for use in this additional term in χ^2.

QMBCFN always computes the first and second derivatives of this additional function (as needed by the Levenberg-Marquardt algorithm) numerically. The step size used for this differentiation entered here is independent from that entered in the Fit Settings tab (Sec. 4.3).

4.5 The Main Tab

4.5.1 Opening and Saving `mbcfn` Files

Using the "File" Menu (or the toolbar), the user can open and save `.mbcfn` files, which contain all session settings of the graphical user interface, i.e. model functions, parameters, derivatives, starting values etc.

4.5.2 Selecting a Data File

A data file name including the full path can be entered in the input field labelled "Data file", or selected using a file open dialog by clicking the button next to the input field. The file will only be loaded when necessary for the fit.

4.5.3 Selecting an Output Directory

The output directory is used for writing the fitted parameters and their covariance matrix to files (see Sec. 4.5.6).

4.5.4 Setting the Start Parameters, Priors and Prior Widths

The table in the middle of the main Tab allows the user to enter starting values for the fit parameters, and, if activated (see section 4.3), the central values of the Gaussian priors A_p and their widths σ_{A_p}.

4.5.5 Fitting

To start the fit(s), click on the botton "Start fit(s)".

If the "Single fit" file format (see Sec. 3.1) is used, QMBCFN will show the values of χ^2/dof and λ are after every iteration. After the fit is finished, the fitted parameters and their uncertainties are shown in window. Additionally, the results and the complete parameter covariance matrix can be written to the selected output directory by using the menu or the toolbar (see Sec. 4.5.6).

If one of the "Multiple fits" file formats (see Secs. 3.2 and 3.3) is used, QMBCFN will perform N_S fits according to the samples in the data file. The fit results are not displayed; instead the streams of fitted parameters will be written to files in the output directory. Every fit parameter gets its own file. For example, if there are two paramaters named `A` and `B`, and the current session file is named `fit1.mbcfn`, the output files `fit1.mbcfn_multifit_A.dat` and `fit1.mbcfn_multifit_B.dat` will be generated.

For Bayesian fitting with the "Multiple fits" file formats, it is possible to resample the priors in addition to the data. If the option "Use Gaussian random priors for multiple fits" in the "Fit Settings" Tab is enabled, the priors for each fit will be choosen randomly from Gaussian distributions with the given prior widths.

4.5.6 Other Tools

The “Tools” menu (and the corresponding buttons in the toolbar) provides some additional useful functionality:

- set the central values of the parameter priors equal to the start values
- set the central values of the parameter priors equal to the previous fit results
- set the parameter start values equal to the previous fit results
- compute the χ^2/dof for the current parameter start values
- reload the data file (use if the data file was modified by an external program while QMBCFN is running)
- write the fit results and the complete parameter covariance matrix to files in the output directory