eSPC, an Online Data Analysis Platform for Molecular Biophysics

PhotoMol 2.0 User Documentation

February 2026

Table of Contents

New features in v2.0

1. Analysis
    1.1. Fitting model
    1.2. Input file
    1.3. Bin width
    1.4. Minimum observed mass
    1.5. Starting values
    1.6. Window range
    1.7. Curve fitting
2. Calibration
Contact details

Overview

PhotoMol was developed to estimate the masses of different species in a sample after a Mass Photometry experiment. More details about this technology can be found at https://www.refeyn.com/.

New features in v2.0

The current version of PhotoMol differs from v1.0 in the following ways:

1) Multiple files can be imported, both for analysis and for calibration.
2) The baseline value is no longer fixed by the user but it used as a parameter to be fitted.
3) There are more options to configure the plots.
4) PhotoMol analysis code has been moved into a standalone python package, available for download via pip (https://pypi.org/project/pyphotomol/).

1. Analysis

1.1. Fitting model

The function that we use to fit is a sum of truncated Gaussians

$$\widehat{y} = b + \sum_{i=1}^{n} g_i(x) \quad (1)$$

where *y* represents the histogram counts, *x* the masses, *n* is the number of truncated Gaussians *g(x)*, *b* is a baseline, and *g(x)* is defined as follows.

$$g(x) = amp * exp(-\frac{(x-center)^2}{2\sigma^2}) \; if \; x \geq x_{threshold} \; else \; 0 \qquad (2)$$

where $x_{threshold}$ is the minimum value of *x* (mass) that can be observed, *center* is the center of the gaussian, σ is the standard deviation, and *amp* is the amplitude.

1.2. Input file

PhotoMol accepts as input a '.h5' (Hierarchical Data Format) file. This file should have one 1D dataset called 'masses_kDa' and can be exported using the software Refeyn DiscoverMP. In the DiscoverMP version < 2.5, the file eventsFitted.h5 is saved in the folder when saving the results. In version 2.5 the events can be exported individually selecting a custom file name. For calibration, only a dataset called ‘contrasts’ is needed.

Additionally, a csv (comma-separated values) file with headers can be loaded. The column ''masses_kDa' and ‘contrasts’ are respectively required for the mass distribution data analysis and calibration.

Multiple files can be imported together.

1.3. Bin width

Integer value (kDa) used to group data and build the histogram.

1.4. Minimum observed mass

Integer value (kDa) that defines the left limit for the truncated multi gaussian.

1.5. Starting values

List of numbers separated by spaces. Each value is used to define the initial guess of the mean of a (truncated) Gaussian.

1.6. Window range

Limits (kDa) for building the histogram.

1.7. Curve fitting

The histogram defined by the bin width and window region is fitted using the Levenberg Marquardt (damped least-squares) algorithm.
The relative errors of the fitted parameters are estimated from the square root of the diagonal elements of the parameter covariance matrix returned by *scipy.curve_fit*. These errors tend to underestimate the true errors of the fitted parameters.

2. Calibration

Ratiometric contrasts can be converted to masses by loading a '.h5' file with known masses (3 different species at least), or using parameters from a previous calibration. In both cases, the calibration experiment should have been done with the same buffer, at the same temperature, and using the same instrument parameters (i.e., the field of view).

The fitting function and parameters are the same as previously described for analyzing the histogram of the observed masses, with the exception that the units are now ‘Ratiometric contrasts’ (instead of kDa).

Contact details

For further assistance, please contact us:

📧 spc@embl-hamburg.de
📌EMBL (c/o DESY), Notkestrasse 85, Build. 25a, 22607 Hamburg, Germany

Packages

PhotoMol is possible thanks to:

R language: R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

R package shiny: Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie and Jonathan McPherson (2020). shiny: Web Application Framework for R. R package version 1.4.0.2. https://CRAN.R-project.org/package=shiny

R package shinydashboard: Winston Chang and Barbara Borges Ribeiro (2018). shinydashboard: Create Dashboards with 'Shiny'. R package version 0.7.1. https://CRAN.R-project.org/package=shinydashboard

R package ggplot2: H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

R package reshape2: Hadley Wickham (2007). Reshaping Data with the reshape Package. Journal of Statistical Software, 21(12), 1-20. URL http://www.jstatsoft.org/v21/i12/.

R package tippy: John Coene (2018). tippy: Add Tooltips to 'R markdown' Documents or 'Shiny' Apps. R package version 0.0.1. https://CRAN.R-project.org/package=tippy

R package shinyalert: Pretty Popup Messages (Modals) in 'Shiny'. R package version 1.1. https://CRAN.R-project.org/package=shinyalert

R package plotly: C. Sievert. Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC Florida, 2020.

R package shinyjs: Dean Attali (2020). shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds. R package version 1.1. https://CRAN.R-project.org/package=shinyjs

R package reticulate: Kevin Ushey, JJ Allaire and Yuan Tang (2020). reticulate: Interface to 'Python'. R package version 1.16. https://CRAN.R-project.org/package=reticulate

R package shinycssloaders: Andras Sali and Dean Attali (2020). shinycssloaders: Add CSS Loading Animations to 'shiny' Outputs. R package version 0.3. https://CRAN.R-project.org/package=shinycssloaders

Python3.7 language: Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

Python package numpy: Travis E, Oliphant. A guide to NumPy, USA: Trelgol Publishing, (2006). Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37

Python package pandas: Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

Python package scipy: Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272.