Full Model 1 [1]

Theory:

- Describes all possible behaviors of a FRAP recovery for a single binding reaction in the presence of diffusion.
- Can be used to fit any FRAP recovery that involves a single binding reaction
- Full model from [2]
- Equations describing the model:

$$\mathrm{frap}(t) = 1 - \phi + \phi\left[L^{-1}\left(\overline{\mathrm{frap}(p)}\right)\right] \qquad (1)$$

$$\overline{\mathrm{frap}(p)} = \left(\frac{1 - C_{\mathrm{eq}}}{p}\right)\left(1 + \frac{k_{\mathrm{on}}^{*}}{p + k_{\mathrm{off}}}\right)(2K_1(qw)I_1(qw))$$

Data preparation:

- Data must be put in a file named *frap.txt*, which must be placed in the same directory, where all MatLab scripts (extension .m) are placed.
- Data must be put in such order that the first column are values of time and in the second column the corresponding FRAP recovery.
- One can put more FRAP data into *frap.txt* in pairs of two columns in such way that the first column are time data and the second column are corresponding FRAP recovery.

Fitting:

- To perform a fitting of FRAP data one should call MatLab script in this form

  fullmodel1(Omega, OmegaErr, m, L, DeIn)

  where input parameters are:

  - *Omega* is value of the diameter of ROI (bleached area) in micrometers
  - *OmegaErr* is estimated error of omega (diameter of ROI) in micrometers
  - *m* is order number of FRAP data pair of columns in frap.txt file which one wants to be fitted *(ex. if one put 5 sets FRAP values, i.e. 10 columns, into frap.txt then m can bear values from 1 to 5 which correspond to each dataset)*
  - *L* is logical parameter and determents function of *DeIn* input parameter as:
    - If *L=1* then *DeIn* is value of diffusion constant *Df* used in fitting
    - If *L=0* then *DeIn* is number of data pairs in *frapPD.txt* file (see *Pure-diffusion dominant* model)
    - If *L=2* then value of diffusion constant *Df* will be fitted as well and *DeIn* is an initial guess of *Df* value

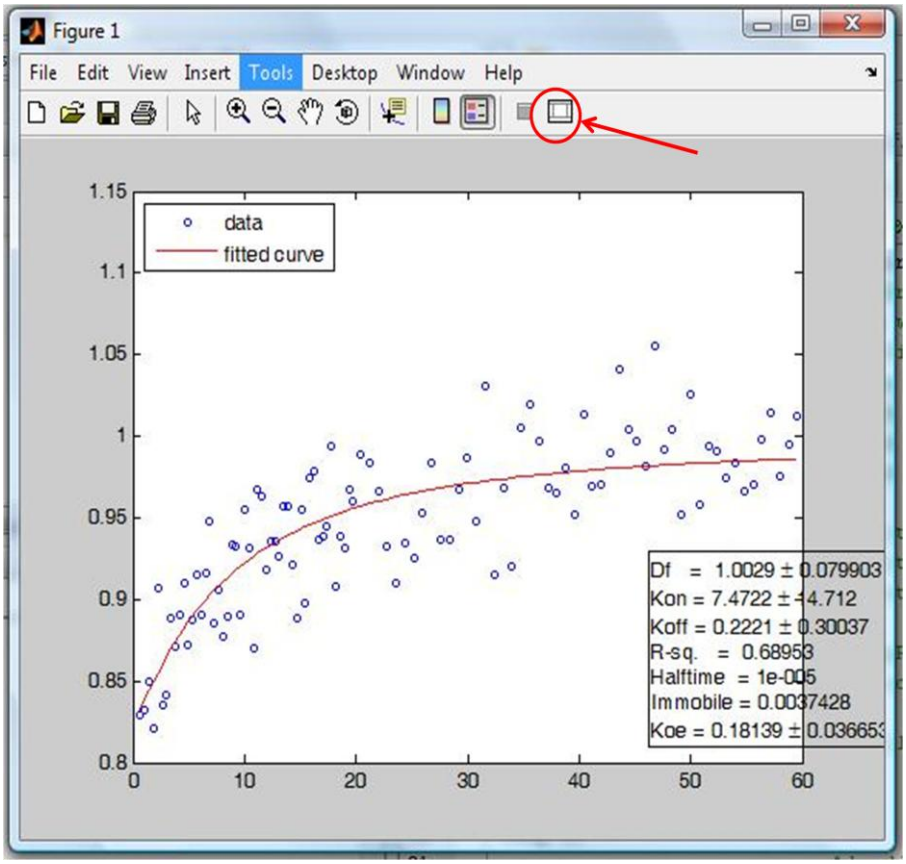
Note that if *L* = *0*, value of *Df* will be calculated with *Pure-diffusion dominant* model averaged over *DeIn* calculated *Df* values, which are later used in fitting process.

Example of calling the script with a ROI diameter of 5 µm, estimated error 0,1 µm, fourth data set from *frap.txt* (columns 8 and 9) to fit, with fixed value of *Df*=0.5:

fullmodel1(5, 0.1, 4, 1, 0.5)

Output:

- *fullmodel1* script creates a file *Fullfit1.txt* in the same directory
- First two columns in *Fullfit1.txt* are values of time and FRAP recovery used in fitting and in the third columns are corresponding values of fitting curve
- The script simultaneously also writes the values of
    - *Df* – value of fitted diffusion constant or used value (depending on input parameter *L*)
    - *ErrDf* – error of *Df*
    - *Rsq* - R-square value of the fitting
    - *Kon* – calculated value of the kinetical rate k_{on}
    - *ErrKon* – error of *Kon*
    - *Koff* – calculated value of the kinetical rate k_{off}
    - *ErrKoff* – error of *Koff*
    - *Koe* – value of *Φ* from Eq. (1)
    - *ErrKoe* – error of *Koe*
    - *Halftime* – value of halftime of the fitted recovery
    - *Immobile* – immobile fraction
- The script also plots FRAP data with fitted curve and writes some important values in a box in bottom-right corner. To change appearance or adjust the plot click on an icon, just as it is shown on Picture 1.



Picture 1

Full Model 2 [3]

Theory:

- Full Model that accounts boundary effects
- It is appropriate to use for splicing on the model gene array
- Situation with different conditions inside and outside of the bleach zone (different k_{on} and k_{off} rates at the localized binding site and at the nonlocalized binding site)
- Radial Binding Model – Full Model from [3]
- Equations describing are much more complex, but nicely illustrated in [3]

Data preparation:

- Same as for *Full Model 1.* Data must be put in a file named *frap.txt*, which must be placed in the same directory, where all MatLab scripts (extension .m) are placed.

Fitting:

- To perform a fitting of FRAP data one should call MatLab script in this form

fullmodel2(Omega, OmegaErr, Kon1, Koff1, Rb, m, L, DeIn, Rto)

where input parameters are:

- *Omega* is value of the diameter of ROI (bleached area) in micrometers
- *OmegaErr* is estimated error of omega (diameter of ROI) in micrometers
- *Kon1* is on rate at the nonlocalized binding site
- *Koff1* is off rate at the nonlocalized binding site
- *Rb* is radius of the cell nucleus in micrometers
- *m* is order number of FRAP data pair of columns in frap.txt file which one wants to be fitted *(ex. if one put 5 sets FRAP values, i.e. 10 columns, into frap.txt then m can bear values from 1 to 5 which correspond to each dataset)*
- *L* is logical parameter and determents function of *DeIn* input parameter as:
  - If *L=1* then *DeIn* is value of diffusion constant *Df* used in fitting
  - If *L=0* then *DeIn* is number of data pairs in *frapPD.txt* file (see *Pure-diffusion dominant* model)
- *Rto* is logical parameter:
  - if *Rto=1* then the ratio of *Kon1* and *Koff1* will be fitted, where values given to the input will serve as an initial guess for fitting
  - if *Rto=0* then will of *Kon1* and *Koff1* will be fixed

Note that if *L = 0*, value of *Df* will be calculated with *Pure-diffusion dominant* model averaged over *DeIn* calculated *Df* values, which are later used in fitting process.

Example of calling the script with a ROI of diameter of 5 µm, estimated error 0,1 µm, with guess for *Kon1=1* and *Koff1=10*, cell nucleus radius of 15 micrometers, second data set from *frap.txt* (columns 4 and 5) to fit, 10 FRAP measurements in file *frapPD.txt* from which Df will be calculated and used:

Fullmodel2(5, 0.1, 1, 10 , 15, 2, 0, 10, 1)

Output:

- *fullmodel2* script creates a file *Fullfit2.txt* in the same directory
- First two columns in *Fullfit2.txt* are values of time and FRAP recovery used in fitting and in the third columns are corresponding values of fitting curve
- The script simultaneously also writes the values of
  - *Df* – value of fitted diffusion constant or used value (depending on input parameter *L*)
  - *ErrDf* – error of *Df*
  - *Rsq* - R-square value of the fitting
  - *Kon* – calculated value of the kinetical constant k_{on}
  - *ErrKon* – error of *Kon*
  - *Koff* – calculated value of the kinetical constant k_{off}
  - *ErrKoff* – error of *Koff*

- *Koe* – value of Φ from Eq. (1)
- *ErrKoe* – error of *Koe*
- *Halftime* – value of halftime of the fitted recovery
- *Immobile* – immobile fraction

- The script also plots FRAP data with fitted curve and writes some important values in a box in bottom-right corner. To change appearance or adjust the plot click on an icon, just as it is shown on Picture 1.

Pure-Diffusion Dominant [1]

Theory:

- Describes a scenatio when most of the fluorescent molecules are free, i.e. no binding (k_{on}/k_{off} <<1)
- Used in determining diffusion constant *Df*
- Model is explained in more details in [1]
- Equation describing the model:

$$frap(t) = e^{-\frac{\tau_D}{2t}}\left(I_0\left(\frac{\tau_D}{2t}\right) + I_1\left(\frac{\tau_D}{2t}\right)\right), \quad (3)$$

$$\tau_D = (w^2/D_f)$$

- We also added an addition constant (Koef) to equation (3), because some FRAP recovery data have recoveries that go over 1. This is cause by complexness of FRAP experiment dealing with live cells. It does not play a big role in Pure-diffusion model, but we have founded that this adjustment helps the fitting process.

Data preparation:

- Data must be put in a file named *frapDP.txt*, which must be placed in the same directory, where all MatLab scripts (extension .m) are placed.
- Data must be put in such order that the first column are values of time and in the second column the corresponding FRAP recovery.
- One can put more FRAP data into *frapDP.txt* in pairs of two columns in such way that the first column are time data and the second column are corresponding FRAP recovery.

Fitting:

- To perform a fitting of FRAP data one should call MatLab script in this form

  puredif(Omega, OmegaErr,NumOfData)

  where input parameters are:

  - *Omega* is value of the diameter of ROI (bleached area) in micrometers

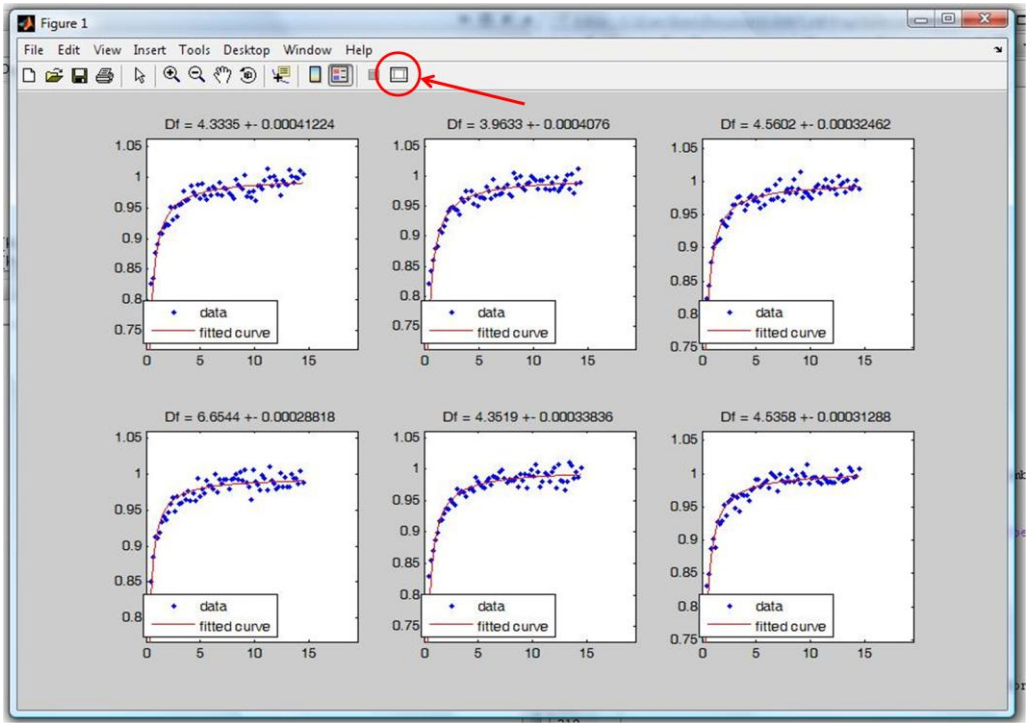
- *OmegaErr* is estimated error of omega (diameter of ROI) in micrometers
- *NumOfData* is number of FRAP data pairs column in frapDP.txt file

Example of calling the script with a diameter of 5 µm, estimated error 0,1 µm, with eight FRAP data pairs in *frapDP.txt* (16 columns):

puredif(5, 0.1, 8)

Output:

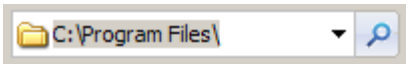
- *puredif* script creates output file for every data set from *frapDP.txt*
- Every output file is called *Purediffit#.txt,* where # is number corresponding to data set from *frapDP.txt*
- First two columns in every output file are values of time and FRAP recovery used in fitting and in the third columns are corresponding values of fitting curve
- The script simultaneously writes down the list of calculated values:
  - *Df* – value of fitted diffusion constant
  - *ErrDf* – error of *Df*
  - *Rsq* - R-square value of the fitting
  - *Koef* – additional constant in equation (3)
- The script also plots all FRAP data with fitted curves. To change appearance or adjust the individual plots click on an icon, just as it is shown on Picture 2.



Picture 2

Finding a way in MatLab

- Most of the problems can be solved by using Help (F1)
- To see a short description of any of above mentioned scripts, how it is used, what are the meaning of the values, etc, just type into the command line
  help nameofthescript (ex. help fullmodel1) and small description will appear
- To change current folder click on Address bar in Toolbar



  It is important to be in the same directory as where are all the mentioned scripts
  Best way to do it would be to extract all the scripts downloaded from this page and extract it into Matlab home directory (depends on installation; usually is in C:\Documents\MATLAB)
- For more details on changing the plot and adjusting it to yours desires follow the link
  http://www.mathworks.com/access/helpdesk/help/techdoc/creating_plots/f9-18692.html

Literature

[1] Sprague et al., *Biophys. J.*, **2004**, *Vol. 86,* 3473-3495
[2] McNally, *Methods in Cell Biology,* **2008**, *Vol. 85,* 329-351
[3] Sprague et al., *Biophys. J.*, **2006**, *Vol. 91*, 1169-1191