P_fFit User Guide

Nava Moran, Arava Shatil-Cohen and Menachem Moshelion
The R.H. Robert Institute of Plant Sciences and Genetics in Agriculture
The Hebrew University of Jerusalem

TABLE OF CONTENTS

INTRODUCTION

BACKGROUND

P_fFit is a three-part program for the calculation of osmotic water permeability of cell membrane (P_f) from volume changes of a spherical, wall-less cell, evoked by a non-instantaneous change of osmolarity in the external medium. The P_f calculation takes into account the rate of the osmolarity change.

P_fFit has been written originally using MATLAB ver. 6.5, rel 13, by Nava Moran and Menachem Moshelion, with GUI written by Roy Novik, compiled and bundled for distribution by Dvorah Weisman, and was recently adapted to run under MATLAB ver. 8.2.0.701, rel R2013b (win32) and recompiled as a stand-alone program by Nava Moran. The PfFit program basic assumptions and definitions are explained in the paper by Menachem Moshelion, Nava Moran, and Francois Chaumont: “Dynamic Changes in the Osmotic Water Permeability of Protoplast Plasma Membrane”[1], including the paper’s Supplemental Material on the ASPB website. PfFit can be downloaded in a self-installing format, free of charge, from

http://departments.agri.huji.ac.il/plantscience/staff-eng/moran-pffit.html

OVERVIEW

The PfFit program consists of three parts:

A: **IndicatorFit,** for determining the parameters of the time course of concentration changes of the bath osmoticum;

B: **ModelMake**, for generating simulations of bath osmoticum changes, changes [optional] of the osmotic water permeability (P_f) and of the cell volume changes;

C: **VolumeFit**, for extracting the osmotic water permeability (P_f) from the time course of changes in cell volume and the changes of bath osmoticum.

Upon the very first run, IndicatorFit requires just two external inputs: the time-course of transmittance changes of the Indicator Dye flushed through the bath in the experimental chamber and initializing parameters; examples for both these files are supplied upon installation (named: 'IndicatorDyeFlushOut.txt' and 'IndicatorDyeParamsInitial.ind').

VolumeFit requires initially a single external input: the time-course of cell volume changes during the [hypo-] osmotic challenge; an example file is also supplied upon installation (named: 'AreasTimeCourse.txt') and, additionally, parameters describing the osmoticum concentration changes in the bath, originating from 'IndicatorFit'.

VolumeFit and IndicatorFit can be tested on simulations created by ModelMake. ModelMake can use either parameters describing the chamber concentration changes originating from IndicatorFit, or manually entered, real or invented parameter values.

IMPORTANT:

This program involves curve fitting. The ease of obtaining "results" may be deceptive!

Check the reported values of the error (FVAL), strive to make it as small as possible, re-try the fit with many various sets of initial guesses! Create simulated data, add 'noise', then try to fit - to get the feeling. Even a small improvement in error may be important! **Be cautious and skeptical!!**

[1] Moshelion et al., 2004, Plant Physiol 135: 2301-2317

INSTALLATION

A connection to the Internet is required for the duration of the installation.

Download the **PfFit installer** ('P$_f$FitInstaller_web.exe') from *http://departments.agri.huji.ac.il/plantscience/staff-eng/moran-pffit.html* .

Double-click on **P$_f$FitInstaller_web.exe** to start installation, choose the program directory according to the Installer's instructions. Click 'Next' to download the 'Matlab Compiler Runtime' (i.e., Routines Library of about 340 Mb required for the stand-alone P$_f$Fit) into the suggested folder, then accept the license terms. The download will last – depending on the system and connections – about 7-12 min. Choose to create a shortcut to the program on the desktop. Press 'Finish' to end the installation.

IMPORTANT!: So as not to clutter your desktop with all of the PfFit outputs**,** **DON'T** leave the direct shortcut to the program PfFit.exe on the desktop; instead, create a working directory (for example, C:\PfFit_workplace\), move there the three example files appearing in the same folder as P$_f$Fit.exe (Program Files\HUJI\P$_f$Fit\application\) to P$_f$Fit_workplace and make a shortcut from the desktop to this working directory. NOTE: not only will PfFit seek its input there but all of your output will go there! Path setting occurs automatically upon the installation.

UNINSTALL

Use the Programs 'change'/uninstall' feature from your computer's Control Panel. You may need to remove the program folder and your working folder manually.

INSTALLING THE IMAGEJ PLUGINS

The two ImageJ plugins, **Image_Explorer.jar** and **Protoplas_Analyzer.jar** (by Xavier Drye) described in the JoVE protocol and used for the analysis of image stacks resulting from video-filming of the Indicator Dye flush and of protoplast volume changes (Shatil et al., 2014) are included with the PfFit Installer. Assuming you have already installed the ImageJ software in your computer, all you need to do is to copy both Image_Explorer.jar and Protoplast_Analyzer.jar to ImageJ\plugins\jars, then restart ImageJ.

RUNNING P_fFit

We assume the program P_fFit has been now successfully installed in your working directory. Three examples (input data and parameters) are supplied with the P_fFit program. They might be helpful during your first attempts to familiarize yourself with the program (the various '.txt', '.ind', '.mod' and '.vol' input or output files can be opened and manipulated using a notepad).

Figures containing plots and lists of results (and initial values) accompany the output tables (text files). To improve visibility, you can stretch the figures and / or shift the legends around. They can be saved, or printed out.

→ Arrows on the left margin and yellow highlight point at our specific suggestions for the supplied examples.

Invoking 'P_fFit' (by clicking twice on the shortcut in the working directory) brings up a MATLAB window with a 'Press to Start' button; pressing it opens the main panel with three choices, each invoking, in turn, one of the three program parts: '**IndicatorFit**', '**ModelMake**' and '**VolumeFit**'.

A: 'IndicatorFit'.

Program Input. Initially, the user must supply (using '**Browse**') the name of the file containing the time course of transmittance change in the bath. These values are sampled while flushing an indicator dye through the bath. To allow initial 'hands-on' → experience, we supply such a time course of flush-out of an indicator (averaged over several repeated runs, to smooth out the noise): IndicatorDyeFlushOut.txt. NOTE that the file is a txt file with no missing or spurious values, commas, or any letters).

The transmittance (T) data are converted to absorbance (A) by Lambert's law: A = -log(T)=-log(I/Io). This generates the Dye$_t$ time course[2]*. Maximum transmittance is assumed to be 1 (A=0), when the light intensity (I) exiting the bath equals the entering light intensity (Io). In fact, because of noise (transmittance fluctuations) during sampling, the experimental maximum value of I can appear to be larger than Io (i.e., I = Io + noise, which could result in T>1). In order to avoid T larger than 1, so as to keep A always ≥0, we accept this maximum I value as the denominator.*

[2] Moshelion et al., 2004, Supplemental Material: '_Appendix_S-II: Equations Used in the Analyses.doc', Eqs 1a-1d)

Absorbance is converted to concentration (Conc.) by Beer's law:
A= Conc. molar_absorptivity * pathway_length, hence: Conc.= A * constant. As a consequence of this linear relation between A and Conc., concentration vs. time (in arbitrary units) has the same shape as absorbance.*

Next is the parameters/variables list, which can be filled out manually, by writing directly into the designated windows (sometimes double-click is required to enable changing the current value), or obtained from a parameter file created on another occasion, through '**Choose a File**'. Try our 'IndicatorDyeParamsInitial.ind', as an example. After the first run, you'll be able to use also the most recently created (and automatically saved) file, 'PreviouslyRunParams', or a file saved previously using '**Save**'. To initiate the parameters to the default values, press '**Reset**'). Take advantage of the '*balloons'* (which can be seen upon resting the mouse pointer for a while atop the various terms in the main panel) to remind yourself what these parameters are all about!

Among the 5 parameter values to be supplied, three are set by the experimenter in each experiment: '***true_C_init***' and '***true_C_end***' are the steady-state values of the initial and the end osmolarities in the bath, i.e., the osmolarities of the solutions used in the experiment (corresponding, respectively, to $Cout_{orig}$ and $Cout_{end}$; Eq. 1b, *Ibid.*), and '***t_start_wash***' is the duration of baseline sampling (with $Cout_{orig}$ in the bath). The 4th value, '***threshold_%***' is used to determine automatically the additional, intrinsic '***flush_lag***', or '*lag*' of the perfusion system (i.e., the lag between the opening of the perfusion valve and the actual entry of the new solution into the bath), by detecting the departure of the indicator absorbance from its baseline (in units of % of the absorbance difference between the end value and the baseline); it has to be higher, the higher is the noise in the Dye_t record and it is limited to 1-7 % (1-4% are usually the most effective). The 5th value, '***N_steady_st_pts***', the number of end samples of Dye_t to be averaged, is required to determine the end steady-state value of dye absorbance (i.e., Dye_{end} in Eq. 1a, *Ibid.*)[3], crucial for determining the exact correspondence between the dye absorbance and the osmoticum concentration. Note, that if an ImageJ software is used to translate the video images of the changing transmittance of the

[3] Important: It is assumed that during the preparatory experiment runs, the bath solution was sampled not only during the dye exchange but also when the new concentration reached steady state.

Indicator Dye to the transmittance time course (as described in the main text[4]), the resulting time course will have n samples for each image taken, according to the width (of n pixels) of the rectangle extracted from each image during this translation. For example, if n=10, and if the 5 last images of the video movie captured the steady-state transmittance of the bath solution, '***N_steady_st_pts***' will be 50.

Additionally, 2 parameters, '$\boldsymbol{t_{width}}$' and '$\boldsymbol{t_{half}}$', are the initial guesses - "seeds" - for two of the four parameters of the sigmoidal curve to be fitted to Dye_t ; 't_{width}' (or '*width*') is related to the duration of the transition part of the sigmoid, 't_{half}' is the midpoint of the transition. These can be initially estimated by eyeballing Dye_t vs. *t* (but then, if the solution exchange time-course begins rather fast, like a hyperbole, and not like a creeping "foot" of a sigmoid, the fit can be frequently improved by using a negative number for 't_{half}').

Program output. At the end of the fit a ***results-summary figure*** appears, which you can save (using the figure menu). Plotted in the figure are, on top: the absorbance data (*Dye vs. t*) and, on bottom: the calculated osmoticum concentration time course.

The correspondence between dye concentration and the osmoticum concentration is based on the following relationship:

$$(Dye_{end} - Dye_{baseline})/(Dye_{baseline} - Dye_{init}) = (Cout_{end} - Cout_{orig})/(Cout_{orig}) - Cout_{init}),$$

where $Dye_{baseline}$ is the initial absorbance of the Indicator Dye in the bath and Dye_{end} and $\boldsymbol{Dye_{init}}$ are the asymptotic values of the Indicator Dye absorbance at +∞ and -∞ ("plus infinity" and "minus infinity"), respectively (Eq. 1a (Moshelion et al., 2004, Appendix S-II), and, similarly, $Cout_{end}$ and $\boldsymbol{Cout_{init}}$ are the asymptotic values of osmoticum concentration at +∞ and -∞, respectively. With Dye_{end} and $Dye_{baseline}$ determined directly from the Dye vs. t data, and Dye_{init} resulting as the best-fit absorbance parameter, and $Cout_{end}$ and $Cout_{orig}$ given as input ('true_C_init' and 'true_C_end'), C_{init} can be calculated from the above relationship and subsequently serve for the calculation of the sigmoidal osmoticum time course, Cout vs t according to Eq. 1b (Ibid.). The present "translation" method is more faithful than that based on Eq.1c-1d (Ibid.).

[4] Shatil et al…..JoVE reference [to be added]

All the input and output parameters/variables mentioned above (except one) are also listed in the *results-summary figure*, and, in particular, three of the best fit parameters describing the time course of bath concentration change, 't_{width}' and 't_{half}', and '***lag'***, or '*flush_lag*' (see definitions above). 'Dye_{init}' and '*lag'* result from the fit without an initial guess, but 'Dye_{init}'is not listed and the calculated '$\boldsymbol{Cout_{init}}$' is listed instead.

Additionally, these values (in the following order: '*t_start_wash*', '*flush_lag*', 't_{width}', 't_{half}', '$Cout_{init}$', $Cout_{orig}$ and $Cout_{end}$) are saved automatically in an ASCII tab-delimited text file: '**IndResults.txt**'. This file will be overwritten in the next run.

The input parameters appearing in the '**IndicatorFit**' panel can be saved by '**Save**' for a later use, with the identifier '.ind' appended automatically to the file name.

To retrieve any previously saved figure: Run the program and produce any figure. The window with the figure has a menu with the word 'file', as well as icons. Use 'open' and choose the appropriate file name.

B: 'ModelMake'.

This is a cell-volume-change simulation program, designed to examine various hypotheses and the relative importance of the parameters in each of the models.

Program Input. The user needs to supply bath-flush parameters to run the program, like in the case of '**IndicatorFit**' (except no data file here). The '*t_start_wash*', '*lag*', 't_{width}', t_{half}, '*Cout_init*', '*true_C_init*' and '*true_C_end*' , described above, can be obtained via '**LastIndicatorFitting**', transferred automatically from the '**IndicatorFit**' program, or via '**PreviouslyRunParams**', from the most recent run, or through '**Choose a file**', if they have been saved previously in the '**ModelMake**' program using '**Save**'.

Additional input parameters are required as follows:

- three parameters related to the dynamics of the cell's osmotic water permeability, '***Pf***' ('P_{f_i}'), the P_f value at the very start of volume change, '***slope_Pf***'' ('$slope_{P_f}$'), the rate of P_f change and '***delay***', which has two meanings: it is either 0, meaning, the time period between the start of bath perfusion and start of volume change (the proper, "physiological" delay) equals 0, i.e., there is no physiological delay, or it is

>0, in which case it means '***delay_inclsv***'(all-inclusive delay), i.e., it includes *t_start_wash* (i.e., the baseline sampling period, till the valve opening) **and** *lag* (the period between opening of the valve and start of bath perfusion), **and** a physiological delay. Try the default values, in the first runs.

- the '***model type***' / '***Class***', defines the P_f dynamics (see detailed descriptions in Results and in Materials and Methods of Moshelion et al., 2004); generally, the models are lumped into three classes: *Class* I (equivalent to model 1); *Class* II (comprising models 2-5)[5]; *Class* III (equivalent to model 6 of Moshelion et al., 2004)[6].

Pressing '**Run**' invokes a detailed list of button-choices of the particular model.

Try, for example, class II, model 5, with non-zero values for 'P_{f_i}', '$slope_{Pf}$' and '*delay*' (use the "all-inclusive delay" value). Note: the three parameters 'P_{f_i}', '$slope_{Pf}$' and '*delay*', as well as '*Class*', may be saved along with the other parameters using '**Save**' and retrieved via '**Choose a file**', or via '**PreviouslyRunParams**'; but they will be reset to default whenever you choose the option '**LastIndicatorFitting**'.

The other parameters to be specified by the users are '***Radius***' (the cell radius), '***Num_of_points***' (the number of sample points to be created in the simulation), '***Duration***' (the total duration of the simulated record of cell volume change) and '***Noise_%***' (the SD of random noise distributed normally around a zero mean, relative to the baseline – to be added to the calculated simulated time courses of cell volume and cell cross-section area). In the beginning, use the defaults; later, add some noise (0.2-2%).

Program output. The simulations are saved in four ASCII tab-delimited text files for alternative display or tests of the analysis program (NOTE: they are overwritten in each run):

[5] Models differ based on the values of "$slope_{P_f}$" and "*delay*": in model 2 both $slope_{P_f}$ and *delay* = 0; in model 3, only *delay* = 0; in model 4, only $slope_{P_f}$=0 and model 5, both $slope_{P_f}$ and *delay* ≠ 0.

[6] Like in model 5, in model 6 all three parameters are non-zero. However, in the framework of model 5, P_f acquires a non-zero value immediately after the delay, but during the delay $P_f = 0$. In model 6, both during the delay and **immediately** thereafter P_f has the same small, non-zero value.

- (1) ‘**MODEL_T_areas.txt**’ (the simulated time course of changing cell’s cross-sectional area): a 2-column matrix, including time, *t*, and area values, without or with added random noise; you can use this file as an input for the ‘VolumeFit’ program.

- (2) ‘**MODEL_T_SIMvolumes.txt**’ (the simulated time course of changing volume): a 3-column matrix including time, *t*, the simulated volume (without or with added random noise) and the simulated volume without noise;

- (3) ’**MODEL_T_P$_f$abs_P$_f$scaled.txt**’ (the simulated time course of P_f), : a 3-column matrix including time, *t*, the simulated P_f in absolute values and the simulated P_f in relative values, scaled for optimized display on the figure;

- (4) ’**MODEL_T_Cout_Cin_delC.txt**’ (the time courses of concentrations): a 4-column matrix, comprised of time, *t*, bath concentration, C_{out}, intracellular concentration C_{in} and the difference between the last two, *delC*.

An additional output is a **figure** showing plots of the simulated data (all three concentrations *vs.* *t*, Volume *vs.* *t* and P_f *vs.* *t*), along with the listed values of all of the input parameters.

C: ‘VolumeFit’.

This is a cell-volume-change fitting program, designed to extract the P_f from the cell swelling dynamics and the osmoticum change in the bath, according to a chosen model.

Program Input. Data file name is required - the time-course of cross-sectional areas; for a first try, use the simulation from ‘**ModelMake**’: ‘**MODEL_T_areas.txt**’, or our “real-life” example: ‘**AreasTimeCourse.txt**’

Parameters of bath flush (below the box) can be input from one of three sources as described above for ‘**ModelMake**’. Note: the initial guesses of the three parameters ‘P_{f_i}’, ‘$slope_{Pf}$’, and ‘delay’, as well as ‘Class’ and the choices whether to fix or free the parameters can be saved with the other parameters by ‘**Save**’ and read in via ‘**Choose a file**’, or via ‘**PreviouslyRunParams**’; but they are reset to default whenever you choose the option ‘**LastIndicatorFitting**’.

The models are categorized, as in ‘ModelMake’, into three types, with the same basic differences with regard to the P_f dynamics. However, rather than being related to the *values* of the parameters (0 or not 0, as in '**ModelMake**'), the model numbers refer here to which parameters are being fixed and which are being fitted (i.e., freely variable and optimized during the fitting procedure).

What to Fit?

By pressing the button "***Combinations***" on the Program's panel you obtain the list of all of the presently-permitted options:

Class I: model 1 – only P_f is free to vary, $slope_{Pf}$ and *delay* can be zero or non zero, but their values remain fixed;

Class II: model 2 – same as model 1; model 3 – P_f and $slope_{Pf}$ are free to vary, *delay* is fixed; model 4 – P_f and *delay* are free to vary, $slope_{Pf}$ is fixed; model 5 – all three are allowed to vary;

Class III, model 6 – all three parameters vary; models 7 and 8, not described before, are based on model 6; model 7 – $slope_{Pf}$ and *delay* are free to vary, P_f is fixed; model 8 – $slope_{Pf}$ is fixed and P_f and *delay* vary[7].

To test the fitting of the simulated data, initially use the same *Class* and model and similar bath-flush parameters as during the simulation. Try also ClassII-model5 with the provided "real-life" examples, and '1' and '1' and '30' for P_f and $slope_{Pf}$ and *delay*, respectively.

Running: Pressing '**Run**' invokes a figure with a 1 s-spaced grid over the time-course of volumes calculated from the cross-sectional areas. Estimate the total length of the baseline from the 1st point till the start of volume change (the "all inclusive delay"). Use this as an input parameter for the "*delay*'(unless you are sure that there is no delay beyond the '*t_start_flush*' + '*lag*') . A query - "do you want to change any of the parameters?" - requires an answer: either an '***N***', to signify "No change"; or a '***Y***', allowing to stop the program, to return to the main panel (one click on the window hiding behind the figure with the grid), to adjust the parameters (in particular: the '*delay*'), to edit the 'AreasTimeCourse' text file (by deleting the points beyond about

[7] Reminder: Like in model 5, in model 6 all three parameters are non-zero. However, in the framework of model 5, P_f acquires a non-zero value immediately after the delay, but during the delay $P_f = 0$. In model 6, both during the delay and **immediately** thereafter P_f has the same small, non-zero value (footnote No. 6 repeated).

15 s after the '*inclusive* delay') and to '**run**' the program again. Upon the 2nd repeat of the query, reply '*N*'; the program should run to completion.

Program output. Upon completion, the program produces a ***results-summary figure*** showing plots of the data *vs*. *t* (on top: the osmoticum concentrations in the bath, in the cell, and the difference between the two, and on bottom: cell volume and P_f) and the fit (to *volume-t*). Also listed on the figure are the values of the various input and output parameters/ variables, including the fit error (usually below 0.5) and a flag ($\neq 1$ indicates a fault in the fitting process).

The bottom legend explains additional symbols on the figure: a blue '**+**' ('**Eo-totLag**', '*t_start_flush*' + '*flush_lag*') shows where the "physiological delay" starts; the "physiological delay" ends at a red '**x**', at which P_f may start to change (as portrayed by the pink dashed line). If the "physiological delay" is null, then both the '**+**' and the '**x**' overlap. Additionally, a few calculated results are given: the volume corresponding to 3% increase in surface area ('**Area up 3%**', the presumed limit of the "allowed" membrane stretching), the volume and the surface area increases ('**avg VOLm %**' and '**avg Area %**') attained at the end of the fitting period (data sequence).

Five ASCII tab-delimited text files allow re-plotting of the data and simulations in user-preferred formats, or tabulating various values for further analyses (NOTE: they are overwritten in each run):

- (1) '**EXPTl_T_volumes.txt**' (the experimentally determined time course of cell's volume): a 2-column matrix, including time and volume;

- (2) '**FIT_T_volumes.txt**' (the simulated best-fit time course of volume): a 2-column matrix including time and the simulated volume;

- (3) '**FIT_T_Pfabs_Pfscaled.txt**' (the simulated time course of P_f): a 3-column matrix including time, the simulated P_f in absolute values and the simulated P_f in relative values, scaled for optimized display on the figure;

- (4) '**FIT_T_Cout_Cin_delC.txt**' (the time courses of concentrations, reconstructed, like in ModelMake, based on parameters from *IndicatorFit*): a 4-column matrix, comprised of time, *t*, bath concentration, C_{out}, intracellular concentration C_{in} and of the difference between the last two, *delC*;

(5) ‘**FIT_Vol_Results.txt**‘ (the list of fit results): ‘*key’* (model number), cell *radius*, ‘ΔC’ (the concentration step between the initial and final bath concentrations, in mM, P_{f_i}, $slope_{Pf}$, *delay*, ‘*FVAL*’ (fit error, normalized to the number of fitted data points and to the baseline volume value), ‘ΔV*_%*’ (the final relative change in cell volume), ‘ΔA*_%*’ (the final relative change in cell surface area), $final_{Pf}$, ‘*V_gro_dur*’ (the duration of the post-delay period of volume change), ‘*true_C_init*’ and ‘*true_C_end*’ ($Cout_{orig}$ and $Cout_{end}$, respectively, in mM).