

Notes on RFIT

A Program for Fitting
Compartmental Models to
Region-of-Interest
Dynamic Emission Tomography Data

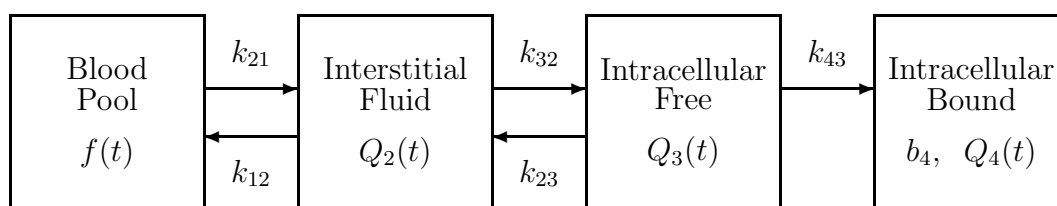
R. H. Huesman
B. L. Knittel
B. M. Mazoyer
P. G. Coxson
E. M. Salmeron
G. J. Klein
B. W. Reutter
T. F. Budinger

Center for Functional Imaging
E. O. Lawrence Berkeley National Laboratory
University of California

This work was supported in part by the Director, Office of Energy Research, Office of Health and Environmental Research, Medical Applications and Biophysical Research Division of the U. S. Department of Energy under Contract No. DE-AC03-SF00098 and in part by the U. S. Department of Health and Human Services Grant No. HL25840.

Introduction

RFIT is a package of Fortran-77 code to simulate compartmental models and estimate their parameters based on kinetic region-of-interest (ROI) data from a positron emission tomograph (PET). A compartmental model proposes that the injected isotope exists in the body in a fixed number of physical or chemical states. Each compartment represents the isotope in one state. By convention in the nuclear medicine literature, the first compartment is the blood pool and its measured values serve as inputs to the differential equation for the model. In the example below, the isotope is distributed between the blood pool, the interstitial fluid and the intracellular space. Inside the cells it is either free or bound (to a receptor or protein, e.g.).



Each compartmental model has an associated differential equation, as described below. By a simulation we mean a calculation of the perfect (noise-free) data that would result from solving the equations for a particular model with a given blood input function and a given set of parameters. Parameters of the model are the transfer rates between compartments and, in some cases, additional parameters reflecting measurement constraints (e.g. time delays between blood and tissue data).

At the present time the software described here runs on Sun/UNIX, Vax/VMS, and Cray/UNICOS systems. It is available, via *anonymous ftp*, along with illustrative examples as well as these notes from CFI.LBL.Gov. Please direct any correspondence to RHHuesman@LBL.Gov, GJKlein@LBL.Gov, or BWReutter@LBL.Gov.

The Command Line

RFIT is invoked with the command

```
rfit bloodfile -sR -i uptakefile -u# <infile >outfile ?errfile
```

where **R** is the (optional) scaling factor which adjusts for the different units and counting efficiencies of the blood and uptake data, and **#** is the number of the region of interest in `uptakefile`. The default scaling factor is 1. `bloodfile` and `uptakefile` are required on the command line. If the blood measurement was calculated as a PET region of interest, the blood and uptake data can be read from the same file:

```
rfit uptakefile -i# -u# <infile >outfile ?errfile
```

In this case, no scaling factor is necessary.

The last three files: `infile`, `outfile`, and `errfile`, are optional. `infile`, provides a list of commands which direct RFIT to carry out desired simulations, parameter fits, etc. Principal commands are summarized in the sections below. If no input file is included in the command line, RFIT will wait for commands to be entered interactively. `outfile` and `errfile` are, respectively, depositories for the results and the error messages generated by RFIT. If they are not included in the command line, then the information will be written to the display terminal.

Specification of the Model

At present, RFIT can fit parameters of linear or bilinear compartmental models with as many as nine distinct compartments. The time evolution of the distribution of an injected radioactive isotope (often attached as a label to a complex molecule) is described by a differential equation:

$$\frac{d\vec{Q}(t)}{dt} = A\vec{Q}(t) + \left\{ Q_2(t) [B\vec{Q}(t)]_2, \dots, Q_n(t) [B\vec{Q}(t)]_n \right\}^T + \vec{F}f(t) \quad (1)$$

where:

- $f(t)$ is the amount of isotope in compartment 1, the blood input function,
- $\vec{Q}(t)$ is the vector $[Q_2(t), Q_3(t), \dots, Q_n(t)]^T$ containing the amount of isotope in compartments 2 to n ,
- A is the matrix of linear transfer rates between tissue compartments,
- B is the matrix of coefficients of the bilinear transfer terms,
- \vec{F} is a vector of transfer rates from the input to each compartment.

The structure of the model is specified in RFIT with the user command `upmod`. For example,

```
upmod 12 21 23 32 r43
```

corresponds to the compartmental model shown in the introduction. The `32` indicates that there is a linear transfer rate, with transfer coefficient denoted k_{32} , from compartment 2 to compartment 3. In the nuclear medicine convention, the first compartment represents the blood pool which is the input $f(t)$. The `21` indicates that compartment 2 is fed with a linear rate $k_{21}f(t)$ from the blood pool and therefore the first entry of \vec{F} is nonzero. The `r` in front of `43` means that compartment 4 represents a saturable receptor and the rate from compartment 3 to compartment 4 is bilinear of the form $[b_4 - Q_4(t)]k_{43}Q_3(t)$. This leads

to nonzero entries in A (the linear coefficient b_4k_{43}) and in B (the bilinear coefficient k_{43}). Note that a receptor compartment introduces an additional parameter b which represents the maximum possible amount of isotope in the compartment. The associated system of differential equations (see equation 1) is:

$$\frac{dQ_2}{dt}(t) = -(k_{12} + k_{32})Q_2(t) + k_{23}Q_3(t) + k_{21}f(t) \tag{2}$$

$$\frac{dQ_3}{dt}(t) = k_{32}Q_2(t) - \{k_{23} + [b_4 - Q_4(t)]k_{43}\}Q_3(t) \tag{3}$$

$$\frac{dQ_4}{dt}(t) = [b_4 - Q_4(t)]k_{43}Q_3(t) \tag{4}$$

Each positive term in the equation for Q_i corresponds to an arrow into compartment i and each negative term to an arrow out.

Simulation of PET Measurements

One of the principal functions of RFIT is to simulate PET measurements for a given model. Each PET measurement represents an accumulation of counts of radioactive emissions over an interval of time $[t_l, t_{l+1}]$. It is assumed that some fraction (f_v) of the counts from a particular region of interest originate in the blood pool rather than the tissue. The integral nature of the data and the necessity of the vascular fraction have been previously discussed in [1]. Thus the measurement function for the system is given by:

$$y_l = (1 - f_v)[1, 1, \dots, 1] \int_{t_l}^{t_{l+1}} \vec{Q}(t)dt + f_v \int_{t_l}^{t_{l+1}} f(t)dt \tag{5}$$

The entries of A , B_i , and \vec{F} can be set to nonzero values only when they correspond to the transfer rates specified with the `upmod` command (see above). For the model (12 21 23 32 r43) the values would be assigned by the following commands:

12	=	.3
21	=	1.
23	=	.2
32	=	.2
43	=	.4
b4	=	.1
fv	=	.15

In RFIT commands, ij is shorthand for k_{ij} . Here the values of k_{12} , k_{32} , k_{23} , and b_4k_{43} will form entries of A , k_{21} will be the first entry of \vec{F} , and the value of k_{43} will be placed in B .

Specifically, A , B , and \vec{F} will be:

$$A = \begin{pmatrix} -(k_{12} + k_{32}) & k_{23} & 0 \\ k_{32} & -(k_{23} + b_4 k_{43}) & 0 \\ 0 & b_4 k_{43} & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & k_{43} \\ 0 & -k_{43} & 0 \end{pmatrix}, \quad \vec{F} = \begin{pmatrix} k_{21} \\ 0 \\ 0 \end{pmatrix}. \quad (6)$$

Note that RFIT manages the mapping between the transfer coefficients and the entries of the matrices in the differential equation.

Evaluation of the measurement integral is carried out by two different methods, depending on whether the model has any nontrivial bilinear terms. If bilinear terms are present, y_l is evaluated numerically with a Runge Kutta integration algorithm. [2] If there are only linear terms (i.e. no \mathbf{r} transfers) then the integral y_l is evaluated in closed form — up to determination of the eigenvalues of A , which must be done numerically. The closed form solution is:

$$y_l = (1 - f_v)[1, 1, \dots, 1] \int_{t_i}^{t_{i+1}} \left[(e^{At} \vec{F}) * f(t) + e^{At} \vec{Q}(0) \right] dt + f_v \int_{t_i}^{t_{i+1}} f(t) dt \quad (7)$$

There are two substantial numerical hurdles involved in evaluating the closed form solution for y_l : 1) evaluation of the matrix exponential, and 2) evaluation of the integral of the convolution. We use triangular and Jordan form transformations to express $e^{At} \vec{F}$ and $e^{At} \vec{Q}(0)$ as $S \vec{E}(t)$ and $S_0 \vec{E}(t)$, respectively, where S and S_0 are $n \times n$ constant matrices and $\vec{E}(t)$ is an n -dimensional vector of scalar functions, $t^k e^{\lambda t}$. The triangular and Jordan form transformations of A are determined by an algorithm adapted from ACM560. [3, 4] The convolution integral is evaluated by an algorithm developed in house. [1, 5]

The RFIT command which produces simulated PET measurements is:

`write up` or `write up >simfile`

In the first case, the result of the simulation will be printed to the terminal or the `outfile`, if one was included in the command line which invoked RFIT. In the second case, the results are stored in a new file, `simfile` (and `>>` instead of `>` would append them to an already existing file).

Parameter Estimation

The second function of RFIT is to find a set of parameters that will produce simulated outputs y_l which are as close as possible to a given set of actual PET measurements, y_l^* . The criterion for goodness of fit is the weighted least squares function:

$$(\vec{Y}^* - \vec{Y})^T \Phi^{-1} (\vec{Y}^* - \vec{Y}), \quad (8)$$

where $Y = [y_1, \dots, y_N]^T$, $Y^* = [y_1^*, \dots, y_N^*]^T$, and Φ is the covariance matrix of the residuals, $\vec{Y}^* - \vec{Y}$. The ROI data files are expected to contain specification of the expected statistical

uncertainties of the data, although unweighted fits can also be performed. Details of a procedure for the estimation of statistical uncertainty of ROI values from emission tomography are given in [6]. The `noisy` option is used to specify that the blood input function contains statistical uncertainty and possibly statistical correlations with the measured tissue function. This is used for a tomographically measured noisy input function. [7]

The method for iteratively moving from one set of parameters to a better set is based on a modified Newton-Raphson [8, 9] optimization algorithm. The RFIT user specifies an initial set of parameters and the fitting routine iteratively replaces each set with a better set until progress is stalled (no better set can be found) or until a fixed maximum number of iterations has passed. No fitting algorithm is foolproof. If the criterion function has multiple local minima, the iterative search may end up at any one of these. Since no constraints are imposed on the parameters, the global minimum of the criterion could correspond to a physically unrealizable set of parameters (negative rate constants, vascular fractions greater than 1, etc.). Usually these difficulties can be overcome by making a careful choice for the initial parameter values.

The command for requesting a fit of a subset of parameters is `fit list`, where `list` contains the parameters to fit. For example, if it is desired to fit k_{21} , k_{43} , b_4 , and f_v , keeping all other parameters fixed, then the appropriate command would be

```
fit 21 43 b4 fv
```

The result of this command is a new set of parameter values, with a standard deviation estimated for each fitted parameter. A correlation matrix is estimated by inverting a numerically determined second derivative of the criterion function. The initial and final criterion values (the chi-square) are shown along with a confidence for the fitted parameters. The confidence term can be interpreted as the likelihood that a system with the fitted parameters could have produced the measured data at hand.

A variety of error and warning messages are possible, such as `Too many iterations` (maximum number of iterations was reached) or `Matrix invert after convergence` (the second derivative of the criterion appears to be singular). If the calculated confidence is extremely low, a warning will be printed in the same line (`Is this acceptable?`). This last feature should not be used to substitute for a careful assessment of the suitability of the fit.

Input Function Fitting

RFIT has a limited ability to fit parametric forms to input function data. This long undocumented feature of the program was intended to be used to simultaneously fit both input and uptake, but the link was never implemented. The general form of the functions available to

fit to the input is given by

$$f(t) = \sum_i^{N_I} a_i (t - t_I)^{c_i} e^{-m_i(t-t_I)}, \quad (9)$$

and the model is specified with the command `in = N_I` . If it is set nonzero an input function fit will be assumed, otherwise the uptake function is fit.

Command Summary

A list of commands to RFIT is given below. The first of these is a generic operation to set the values of various parameters and variables described in the following section. Parameters and variables may be referenced by the first two characters of their name in commands. Similarly, commands shown below may often be referenced by less than their full extent (usually four characters).

<code>var = #</code>	Set a variable or parameter value to <code>#</code> . Only the first two letters of the variable name need be entered. Variable and parameter names which can be set by the user are listed in the following section.
<code>convolve</code>	Use the triangular and Jordan form transformations for matrix exponentiation and subsequent explicit convolution.
<code>debug off</code>	Turn off debug output.
<code>debug crider</code>	Turn on debug output for the routine <code>crider</code> .
<code>debug critup</code>	Turn on debug output for the routine <code>critup</code> .
<code>debug hnoisy</code>	Turn on debug output for the routine <code>hnoisy</code> .
<code>debug newton</code>	Turn on debug output for the routine <code>newton</code> .
<code>debug odeint</code>	Turn on debug output for the routine <code>odeint</code> .
<code>debug setlam</code>	Turn on debug output for the routine <code>setlam</code> .
<code>do filename</code>	Take command from the input file <code>filename</code> .
<code>dump</code>	Print out data, variable and parameter values.
<code>fit list</code>	Attempt to minimize the criterion by finding new values for the parameters in <code>list</code> .
<code>nodebug</code>	Turn off debug output.

<code>omfv on</code>	Turn on flag to simulate uptake multiplied by the factor $(1 - f_v)$.
<code>omfv off</code>	Turn off flag to simulate uptake multiplied by the factor $(1 - f_v)$.
<code>noisy</code>	Fit using a tomographically measured noisy input function.
<code>pcoef #</code>	Fit a partition coefficient for compartment <code>#</code> instead of a rate.
<code>pcoef off</code>	Turn off fitting of partition coefficients.
<code>quit</code>	Close files and exit the program.
<code>show var</code>	Show the value of <code>var</code> .
<code>show all</code>	Show the values of all variables and parameters.
<code>show all in</code>	Show the values of all input parameters.
<code>show all fit</code>	Show the values of all fitting variables.
<code>show all up</code>	Show the values of all uptake parameters.
<code>solve</code>	Use the Runge Kutta ODE solver.
<code>solve sum</code>	Use on-the-fly summation for <code>solve</code> .
<code>solve nosum</code>	Do not use on-the-fly summation for <code>solve</code> .
<code>upmod model</code>	Enter an uptake model denoted by <code>model</code> . The model is made up of comma or blank separated fields indicating the presence of a transfer channel between compartments. The fields contain decimal digits, the first of which indicates the compartment number of the destination of the transfer and the rest of which indicate the compartment numbers of the sources. If the first character of a field is the letter “r”, the destination for that field is a receptor compartment.
<code>vmeas on</code>	Turn on flag to simulate venous measurements.
<code>vmeas off</code>	Turn off flag to simulate venous measurements.
<code>write cp >file</code>	Write compartments of the uptake function and the input function to <code>file</code> .
<code>write in >file</code>	Write input function data and fit to <code>file</code> .
<code>write up >file</code>	Write uptake function, input function and fit to <code>file</code> .

Variable and Parameter Names

The names of variable and parameters which can be changed by the user are listed below. Initialization of parameter values is necessary before a fit is started. Changes of the values of variables which influence the fitting procedure may be necessary in special cases or when other than a 64-bit floating point data type is used.

<i>ij</i>	Rate constant from compartment <i>j</i> to compartment <i>i</i> . This is a shorthand for k_{ij} , and the units are min^{-1} .
<i>bi</i>	Saturation value for receptor compartment <i>i</i> .
<i>pi</i>	Partition coefficient for compartment <i>i</i> . (k_{i1}/k_{1i})
<i>fv</i>	Vascular fraction.
<i>t0</i>	Time delay.
<i>dt</i>	Time constant for input function blurring.
<i>in</i>	Input function model. (Null for uptake function fits.)
<i>ai</i>	Amplitude for input function model fits.
<i>ci</i>	Time exponent for input function model fits.
<i>mi</i>	Decay rate for input function model fits.
<i>ti</i>	Time offset for input function model fits.
<i>cabs</i>	Absolute convergence criterion.
<i>crel</i>	Relative convergence criterion.
<i>ylam</i>	Initial relative gradient weighting to condition a singular Hessian matrix.
<i>vlam</i>	Factor by which relative gradient weighting is cut for good iterative steps.
<i>coz</i>	Cosine of an angle determined to be sufficiently close to the direction of the negative gradient for bad iterative steps.
<i>vconst</i>	Factor by which step size is cut for bad iterative steps which are near the direction of the negative gradient.
<i>eps</i>	Minimum relative gradient weighting.
<i>dabs</i>	Absolute derivative increment.

drel	Relative derivative increment.
rkeys	Maximum relative error for Runge Kutta integration.
h1	Initial stepsize for Runge Kutta integration.
hmin	Minimum stepsize for Runge Kutta integration.
ein	Tolerance parameter used in the grouping of numerical multiple eigenvalues for the Jordan Normal Form.
tol	Tolerance parameter used in the determination of block rank for the Jordan Normal Form.
nsteps	Maximum number of iteration steps.

References

- [1] Mazoyer BM, RH Huesman, TF Budinger, and BL Knittel. Dynamic PET data analysis. *J Comput Assist Tomogr*, **10**(4):645–653, 1986.
- [2] Press WH, BP Flannery, SA Teukolsky, and WT Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 1986.
- [3] Kagstrom B and A Ruhe. An algorithm for numerical computation of the Jordan normal form of a complex matrix. *ACM Transactions on Mathematical Software*, **6**(3):398–419, 1980.
- [4] Kagstrom B and A Ruhe. Algorithm 560: JNF, an algorithm for numerical computation of the Jordan normal form of a complex matrix [F2]. *ACM Transactions on Mathematical Software*, **6**(3):437–443, 1980.
- [5] Coxson PG, EM Salmeron, RH Huesman, and BM Mazoyer. Simulation of compartmental models for kinetic data from a positron emission tomograph. *Comput Methods Programs Biomed*, **37**:205–214, 1992.
- [6] Huesman RH. A new fast algorithm for the evaluation of regions of interest and statistical uncertainty in computed tomography. *Phys Med Biol*, **29**(5):543–552, 1984.
- [7] Huesman RH and BM Mazoyer. Kinetic data analysis with a noisy input function. *Phys Med Biol*, **32**(12):1569–1579, 1987.
- [8] Bates DM and DG Watts. *Nonlinear Regression Analysis and its Applications*. John Wiley & Sons, 1988.
- [9] Seber GAF and CJ Wild. *Nonlinear Regression*. John Wiley & Sons, 1989.