Analysis of OCT decays

Pascal PERNOT

2018-12-07

Contents

1 Introduction

Analysis of OCT signals by a mono-exponential decay reveals two features which condition the proposed data analysis method: an heterogeneous random noise (Poisson-like; see Fig. 1), and medium-scale oscillations around the exponential decay (model inadequacy; see Fig. 2). To be able to partition unambiguously the model residuals between these two components, we proceed in two steps:

1. estimation and modeling of the random noise component, to be injected in
2. the estimation of the parameters of the decay model:
    a. test of a simple monoexponential model
    b. if the latter fails, use of a modulated decay model

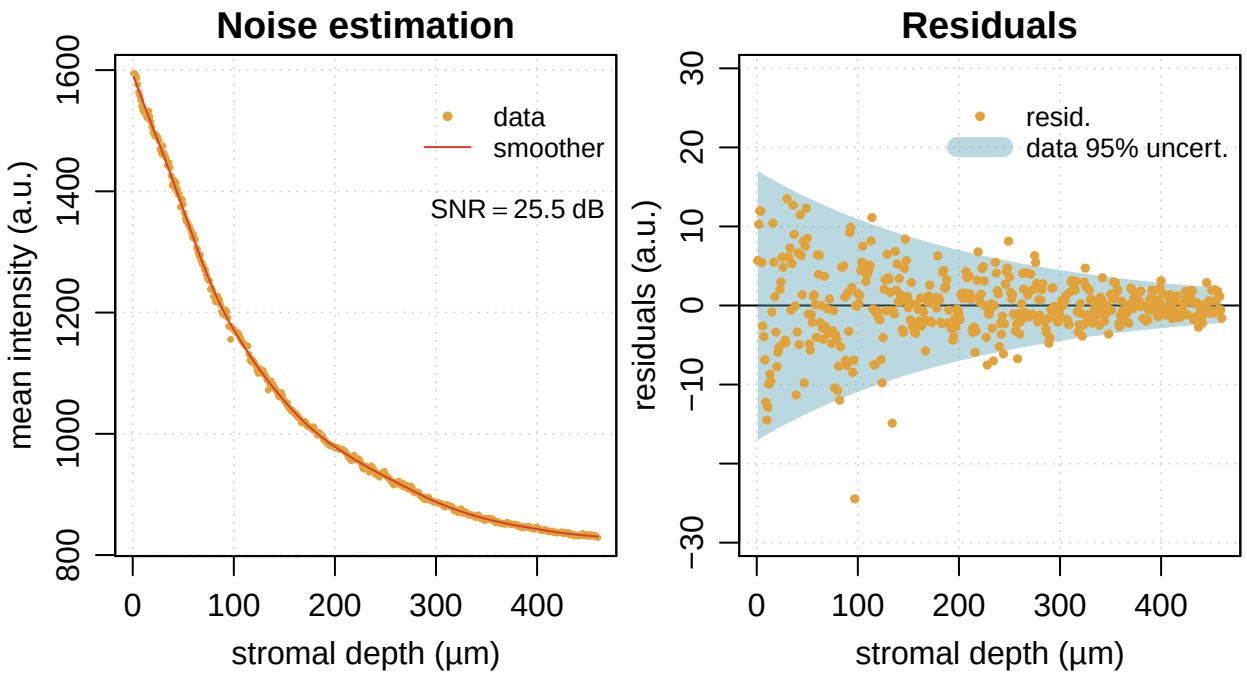

Figure 1: Splines smoothing and noise estimation

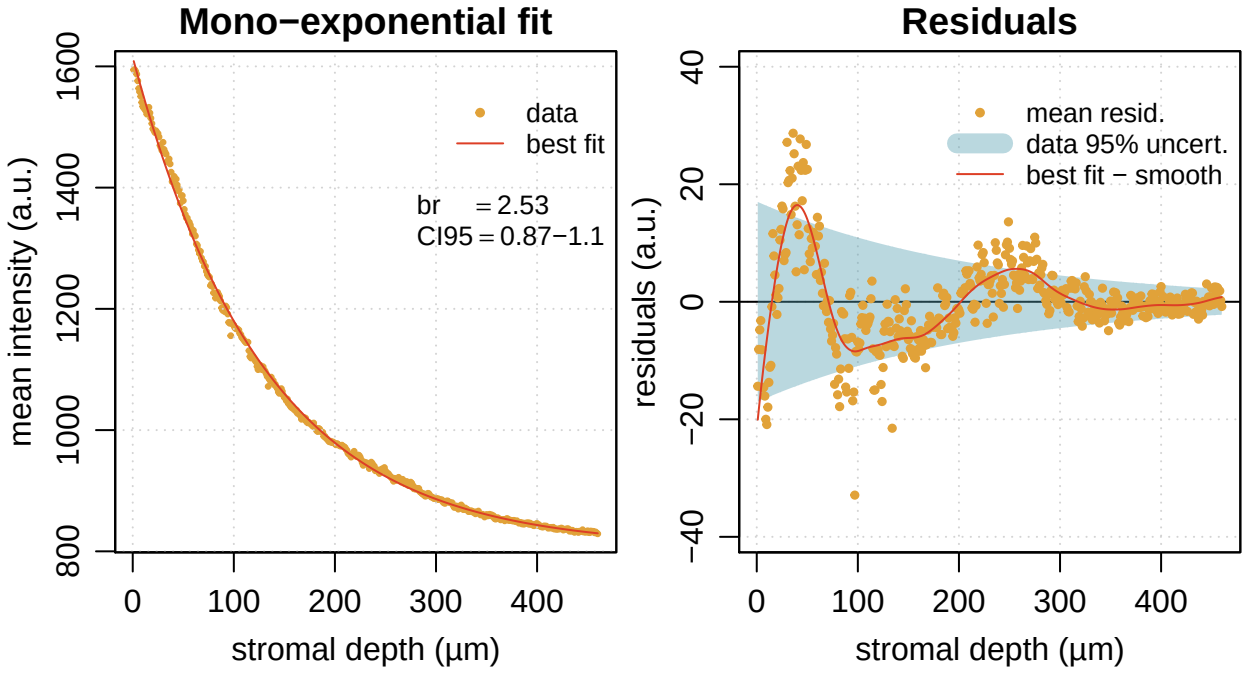

Figure 2: Mono-exponential fit and residuals.

2 Methods

Considering a set of N measured data points $\boldsymbol{D} = \{z_i, y_i\}_{i=1}^{N}$, one considers a measurement model with heterogeneous additive noise

$$y_i = f(z_i) + \epsilon_i \tag{1}$$

where $f(.)$ is a model function to be defined, and

$$\epsilon_i \sim Norm(0, \sigma_i) \tag{2}$$

represents an heterogeneous measurement noise with a normal distribution of standard deviation σ_i.

2.1 Estimation and modeling of the random noise

A cubic smoothing spline function is used to estimate the random part of the signal. The residuals $\boldsymbol{R} = \{R_i\}_{i=1}^{N}$ of the smoothing function are assigned to random noise ϵ.

Considering that the OCT decays result from photon counting experiments, one can expect that the noise obbeys a Poisson law. In consequence, the standard deviation of the noise is modeled by an exponential decay

$$\sigma_i = a_1 * \exp\left(-\frac{2 * z_i}{a_2}\right) \tag{3}$$

This shape enables also to account for cases of *in vivo* measurements with nearly uniform noise, by letting $a_2 >> \max(\boldsymbol{z})$.

The parameters are obtained by Bayesian inference [1] with likelihood

$$\boldsymbol{R}|a_1, a_2 \sim \prod_{i=1}^{N} \mathrm{N}(0, \sigma_i) \tag{4}$$

and uniform priors for a_1 and a_2 in the range $]0, a_{max}]$, the upper value being chosen to accommodate a quasi-uniform noise model. The point estimates of the parameters are used to define the measurement uncertainty

$$u_{yi} = \hat{a_1} * \exp\left(-\frac{2 * z_i}{\hat{a_2}}\right) \tag{5}$$

to be used in the next steps, for which one has now a data set augmented with measurement uncertainties $\boldsymbol{u}_y$, *i.e.* $\boldsymbol{D} = \{z_i, y_i, u_{yi}\}_{i=1}^{N}$.

2.2 Calibration of a decay model

2.2.1 Mono-exponential decay

The mono-exponential decay curve with parameters $\boldsymbol{\vartheta} = \{a, b, l_0\}$

$$f(z; \boldsymbol{\vartheta}) = a + b * \exp\left(-\frac{2 * z}{l_0}\right) \tag{6}$$

is fitted to the data by maximization of the posterior pdf (MAP). The likelihood is

$$\boldsymbol{y}|\boldsymbol{\vartheta} \sim \prod_{i=1}^{N} \mathrm{N}(f(z_i; \boldsymbol{\vartheta}), u_{yi}) \tag{7}$$

The parameters have uniform priors on $[0, \infty[$.

2.2.1.1 Validation

One defines the a weighted chi-square function as

$$\chi^2_w(\boldsymbol{y}; \boldsymbol{z}, \boldsymbol{u}_y, \boldsymbol{\vartheta}) = \sum_{i=1}^{N} \frac{[y_i - f(z_i; \boldsymbol{\vartheta})]^2}{u^2_{yi}} \tag{8}$$

The value of the reduced chi-square $\chi^2_r = \chi^2_w/(N-3)$ should be close to 1 ($\chi^2_r \in IQ_{95}$), based on the quantiles of the reduced chi-square distribution with $N-3$ degrees of freedom. Moreover, the residuals should not present serial correlation. If these conditions are not met, one has to use the more elaborate model, described below.

2.2.2 Modulated decay model

The mono-exponential decay model is improved with a z-dependent optical depth $l(z; l_0, \boldsymbol{\kappa})$

$$f(z; \boldsymbol{\vartheta}, \boldsymbol{\kappa}) = a + b * \exp\left(-\frac{2 * z}{l(z; l_0, \boldsymbol{\kappa})}\right) \tag{9}$$

where the shape of the optical depth is defined as

$$l(z; l_0, \boldsymbol{\kappa}) = l_0 * (1 + \delta l(z; \boldsymbol{\kappa})) \tag{10}$$

$\delta l(.)$, *the modulation function*, is a Gaussian Process (GP) of mean 0, conditioned on M control values $\boldsymbol{\kappa} = \{\kappa_i\}_{i=1}^M$ at predefined locations $\hat{\boldsymbol{z}} = \{\hat{z}_i\}_{i=1}^M$. The mean value of the GP is used here as an interpolator between the control points, and we choose a Gaussian kernel for

its smoothness properties

$$C(z, z') = \alpha^2 * \exp\left(-\frac{(z - z')^2}{\rho^2}\right) \tag{11}$$

The α and ρ parameters of the GP are fixed *a priori*.

Considering the set of M control values $\boldsymbol{\kappa}$ for the OD modulation at locations $\hat{\boldsymbol{z}}$, δl can be obtained at any depth as the mean value of the GP:

$$\delta l(z; \boldsymbol{\kappa}) = \boldsymbol{\Omega}^T * \boldsymbol{K}^{-1} * \boldsymbol{\kappa} \tag{12}$$

where $\boldsymbol{K}$ is a $M \times M$ covariance matrix with elements $K_{ij} = C(\hat{z}_i, \hat{z}_j)$ and $\boldsymbol{\Omega}$ is a M-vector with elements $\Omega_i = C(\hat{z}_i, z)$.

The control points positions, $\hat{\boldsymbol{z}}$, are chosen *a priori* on a regular grid spanning the experimental depth range. As one does not expect short scale modulations, a small number of points is used, typically $M \simeq 10$. Similarly, for the smoothness of the interpolation, one picks the correlation length of the kernel *a priori*, at a value large enough to avoid undue oscillations between the control points and small enough to avoid excessive rigidity of the model. In the present configuration, a good compromise has been found to be $\rho = 1/M^{th}$ of the total depth range. In the same spirit, the variance parameter of the GP is taken as a small fraction of the standard deviation of the control values $\alpha = 0.1 * sd(\boldsymbol{\kappa})$. This choice of α and ρ has been found to provide a well behaved interpolator for test simulated signals. Besides, small changes around these values do not affect significantly the mean prediction of the GP.

2.2.2.1 Prior pdfs

$\boldsymbol{\kappa}$. The definition of the modulation function is a source of indetermination between l_0 and $\boldsymbol{\kappa}$; for instance, setting all values of $\boldsymbol{\kappa}$ to 1 would be exactly compensated by halving l_0. One therefore constrains $\boldsymbol{\kappa}$ to be close to zero with a Bayesian Lasso-type prior [2], in a version based on a hierarchical prior adapted from Ref.[3]:

$$\begin{aligned} \kappa_i | u_i &\sim Normal(0, s_i); i = 1, M \\ s_i | \lambda &\sim Gamma(2, \lambda); i = 1, M \\ \lambda &\sim Gamma(2, \lambda_r) \end{aligned} \tag{13}$$

where λ_r is chosen *a priori*; it defines the scale of expected deviations from zero of $\boldsymbol{\kappa}$ (typically $\lambda_r = 0.1$). An example is shown in Fig. 3.

$\boldsymbol{\vartheta}$. The prior on $\boldsymbol{\vartheta}$ is a multivariate normal distribution $\pi(\boldsymbol{\vartheta}) = N(\hat{\boldsymbol{\vartheta}}_1, \boldsymbol{\Sigma}_\vartheta)$ centered on the best estimate from the mono-exponential fit, $\hat{\boldsymbol{\vartheta}}_1$, with a covariance matrix $\boldsymbol{\Sigma}_\vartheta$. Because the present model is used when the mono-exponential decay is inadequate, one cannot rely directly on

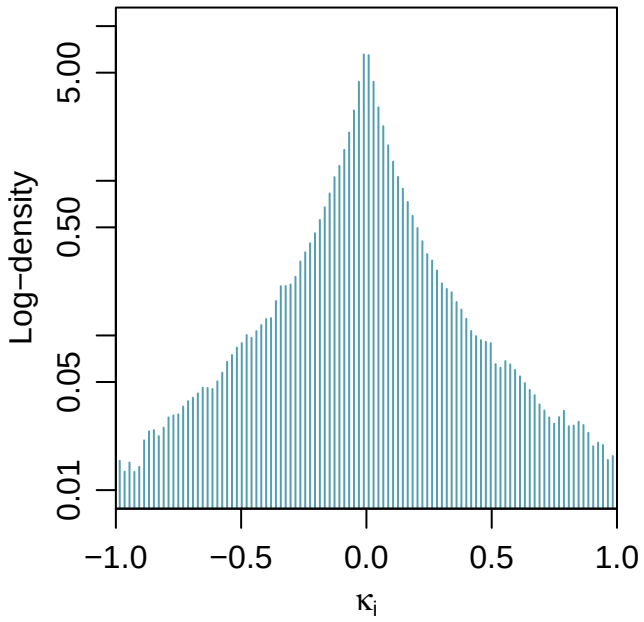

Figure 3: Log-prior pdf of a κ parameter.

the covariance matrix extracted from its calibration, $\boldsymbol{\Sigma}_{\vartheta_1}$. Instead, a covariance matrix is estimated to cover the range of variations of the mono-exponential residuals [4], following two approaches:

1. the covariance matrix is built from the correlation matrix $\boldsymbol{C}_{\vartheta_1}$ issued from the monoexponential fit, and a vector of standard deviations specified from relative uncertainties on the parameters:
$$\boldsymbol{\Sigma}_{\vartheta} = \boldsymbol{I}(\boldsymbol{u}_{\vartheta}) * \boldsymbol{C}_{\vartheta_1} * \boldsymbol{I}(\boldsymbol{u}_{\vartheta}) \tag{14}$$
where $\boldsymbol{I}(\boldsymbol{u}_{\vartheta})$ is a diagonal matrix with elements $\boldsymbol{u}_{\vartheta} = r * \hat{\boldsymbol{\vartheta}}_1$. The uncertainty factor r is typically chosen as a small percentage, *e.g.*, $r = 0.05$.

2. a *diagonal* covariance matrix $\boldsymbol{\Sigma}_{\vartheta} = \boldsymbol{I}(\boldsymbol{u}_{\vartheta}^2)$ is built by a moments-matching procedure. The standard deviations $\boldsymbol{u}_{\vartheta}$ are optimized to match two criteria:

   a. S_1, the 2-sigma prediction uncertainty of the mono-exponential model has to match Q_{95}, the 95^{th} quantile of the absolute errors of the mono-exponential model (all statistics are weighted by $\boldsymbol{u}_y$) [5];

   b. the standard deviation of the prediction uncertainty has to be as small as possible. The first criterion ensures that the mean prediction uncertainty of the mono-exponential model is in agreement with the amplitude of the model's residuals [4]. This criterion can typically be matched by an infinity of solutions, and the second one selects those parameters which provides the 'flatest' prediction band. The optimization is done by sampling the posterior pdf of the parameters $\boldsymbol{u}_{\vartheta}$ with

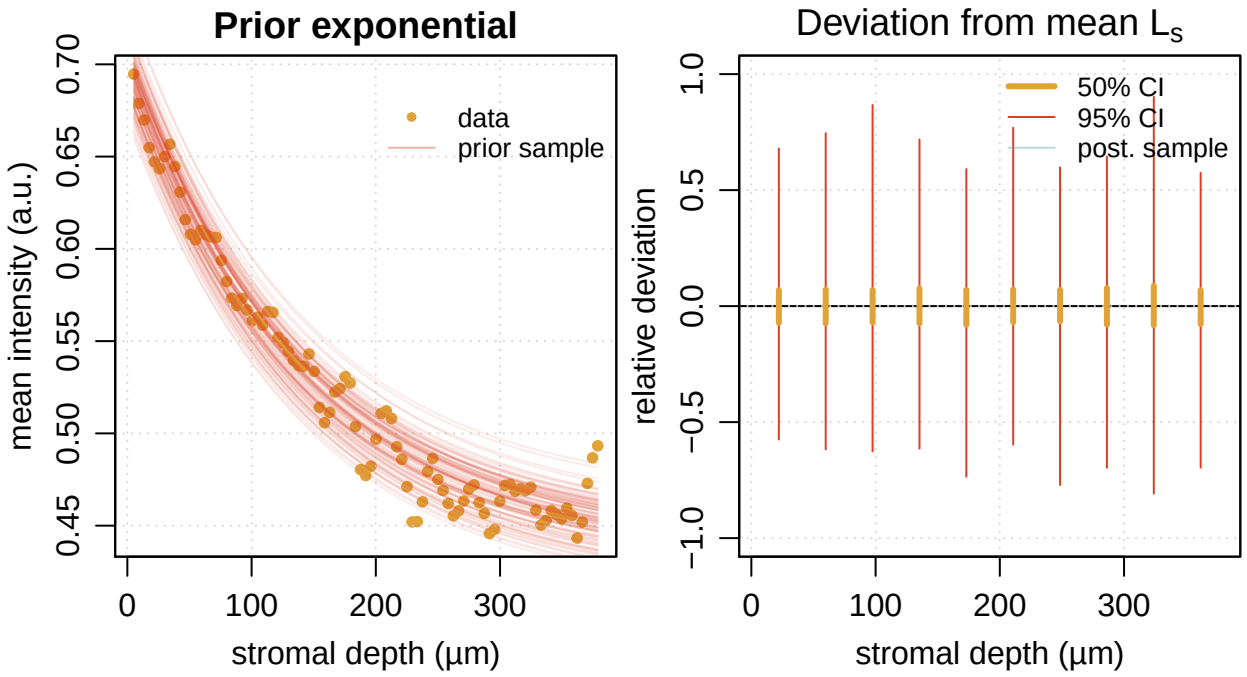

Figure 4: Samples from prior pdf for an *in vivo* signal.

uniform priors on $[0, \infty[$, and a likelihood function

$$\{Q_{95}, 0\} \,|\boldsymbol{u}_\vartheta \sim \mathrm{N}_2(\{S_1(u_\vartheta), S_2(u_\vartheta)\}\,, \varepsilon) \tag{15}$$

where $S_1 = 1.96 * \sqrt{< \boldsymbol{u}_p^2/\boldsymbol{u}_y^2 >}$, $\boldsymbol{u}_p$ is the prediction uncertainty of the mono-exponential model estimated by linear uncertainty propagation [6], Q_{95} is the the 95^{th} quantile of the absolute weighted residuals $|\,\{\boldsymbol{y} - f(\boldsymbol{z};\boldsymbol{\vartheta})\}\,/\boldsymbol{u}_y|$, $S_2 = \mathrm{sd}(\boldsymbol{u}_p/\boldsymbol{u}_y)$ and ε is a predefined precision factor ($\varepsilon = 10^{-3}$).

A sample generated with this moments-matching prior for an *in vivo* signal is shown in Fig. 4(left).

σ. To compensate for defaults in the estimation of the noise, a parameter σ is introduced as a multiplicative factor of $\boldsymbol{u}(y)$. The prior on σ is a normal distribution, centered on 1, with standard deviation 0.1.

The parameters to be sampled are therefore $\boldsymbol{\vartheta}$, $\boldsymbol{\kappa}$, λ_r and σ.

2.2.2.2 Likelihood function

The likelihood function is the product of univariate normal distributions

$$\boldsymbol{y}|\boldsymbol{\vartheta}, \boldsymbol{\kappa}, \sigma, \sim \prod_{i=1}^{N} \mathrm{N}(f(z_i;\boldsymbol{\vartheta},\boldsymbol{\kappa}), \sigma * u_{yi}) \tag{16}$$

2.2.2.3 Validation

The quality of the fit can be estimated by inspection of the residuals which should not present serial correlations and should conform with the random experimental noise.

The value of the reduced chi-square $\chi^2_r = \chi^2_w/(N-\nu)$, where ν is the number of effective free parameters, should be close to 1. One has $\nu = 5 + \hat{M}$, where $0 \leq \hat{M} \leq M$ is the number of control values significantly different from zero, *i.e.* those for which $0 \notin IQ_{90}(\kappa_i)$.

Posterior predictive samples are also generated and plotted with the reference data to confirm the quality of the fit.

3 Implementation

The core functions are in the package FitOCTLib.

Basic algorithm

0. Read signal `(x, y)` and apply eventual thinning and subsetting with function `selX()`
1. Noise estimation: `en <- estimateNoise(x, y)` which returns a list containing `uy`
2. Mono-exponential fit: `fit1 <- fitMonoExp(x, y, en$uy)`. Validity is checked by function `br <- printBr(fit1)`. If `is.null(br$alert)`, the model is OK, on can stop.
3. Modulated exponential fit: `fit2 <- fitExpGP(x, y, en$uy)`. If the fit is not valid (`!is.null(printBr(fit2)$alert)`)), one might increase the number of control points, *e.g.* `fit2 <- fitExpGP(x, y, en$uy, Nn = 15)`.

3.1 Smoothing

Estimation of the random errors is done with the `smooth.spline` from `R` [7]; A satisfying degree of smoothing for all examples considered here was obtained by setting the smoothing `df` parameter to 15.

3.2 Bayesian inference

The Bayesian models are implemented in `stan` [8], using the `rstan` interface package [9] for `R` [7]. `Stan` is a very flexible and efficient probabilistic programming language to implement Bayesian statistical models. The No-U-Turn sampler [10] was used for this study.

The main outputs of the stan codes are samples of the posterior pdf of the parameters, from which statistics and plots can be generated in R. Convergence of the sampling is assessed by examining the traces of parameters samples and the 'split Rhat' statistics provided by rstan. In the present application, the Markov chains converge rapidly, and all models are run with four parallel Markov Chains of 1500 iterations each, 1000 of which are used as warm-up for the No-U-Turn sampler and dispatched. The convergence criteria and parameters statistics are therefore estimated on a sample of 2000 points.

References

1. Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis* (3rd ed.). Chapman; Hall/CRC. Retrieved from https://www.crcpress.com/Bayesian-Data-Analysis-Third-Edition/Gelman-Carlin-Stern-Dunson-Vehtari-Rubin/p/book/9781439840955?source=igodigital

2. Park, T., & Casella, G. (2008). The bayesian lasso. *J. Am. Stat. Assoc.*, *103*, 681–686. doi:10.1198/016214508000000337

3. Mallick, H., & Yi, N. (2014). A new bayesian lasso. *Statistics and Its Interface*, *7*, 571–582. doi:10.4310/SII.2014.v7.n4.a12

4. Pernot, P. (2017). The parameter uncertainty inflation fallacy. *J. Chem. Phys.*, *147*(10), 104102. doi:10.1063/1.4994654

5. Pernot, P., & Savin, A. (2018). Probabilistic performance estimators for computational chemistry methods: The empirical cumulative distribution function of absolute errors. *J. Chem. Phys.*, *148*, 241707. doi:10.1063/1.5016248

6. BIPM, IEC, IFCC, ILAC, ISO, IUPAC, ... OIML. (2008). *Evaluation of measurement data - Guide to the expression of uncertainty in measurement (GUM)* (No. 100:2008). Joint Committee for Guides in Metrology, JCGM. Retrieved from http://www.bipm.org/utils/common/documents/jcgm/JCGM_100_2008_F.pdf

7. R Core Team. (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org/

8. Gelman, A., Lee, D., & Guo, J. (2015). Stan: A probabilistic programming language for Bayesian inference and optimization. *J. Educ. Behav. Stat.*, *40*, 530–543. doi:10.3102/1076998615606113

9. Stan Development Team. (2016). *RStan: The R interface to Stan*. Retrieved from http://mc-stan.org/

10. Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 1593–1623. Retrieved from http://jmlr.org/papers/v15/hoffman14a.html