

## Answers to reviewer #1:

*This looks like a very useful piece of software that has already delivered some interesting results and applications. The webpage gives a fair amount of documentation although I think potential appliers will still have questions and it is conceivable that the FAQ section will increase.*

Thank you for your positive evaluation. Our webpage is constantly evolving. We hope that you will be pleased that the developer community and coauthors of this manuscript tried to quickly implement your feedback into the online documentation as well.

*The following questions and comments occurred to me:*

*(1) Although it is stated that the software is free, it is based on MATLAB which is of course not free (in fact it is rather expensive) and not available to many researchers.*

We agree that MATLAB is non-free software, however, it is heavily used not only in industry but also in academia. Academic licenses for students are fairly cheap. Nevertheless, there are open source alternatives such as GNU Octave (<http://www.gnu.org/software/octave/>). We are confident that our code could be easily converted to an Octave compatible version. Here is a quote from the website: “The Octave language is quite similar to MATLAB so that most programs are easily portable.” This is however still beyond our current development capacities.

*(2) There seem to be a huge amount of packages that fit ODE's to data (also in R which is free) and a potential applier has a huge choice. To help the applier I think the authors should provide some reasoning why and when their proposed package is preferable.*

We believe that the main strength of our software is its ability to perform efficient parameter estimation by using derivatives calculated based on the sensitivity equations. We tried to bring our main point across in the title of the manuscript: “Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems”. We are not aware of any other readily available software package for MATLAB that provides this functionality, flexibility and performance. The limitations of alternative packages were the major reason for developing our modeling environment. In the revised version of the manuscript, we tried to be more specific by changing the sentence “We present a modeling environment **for MATLAB** that pioneers these challenges.” in the abstract and “We showed previously (Raue et al., 2013b) that this approach, **which is not implemented in any other software package**, is not only about ten times faster but also more precise than the default approach using finite differences.” in the main text.

*(3) The paper states that 'prior knowledge about parameters can be considered as well'. But this is not clear.*

*I would assume you mean in the form of prior distributions but I have not found any details about how to specify this and how the algorithm is turned into a Bayesian estimation?*

By adding prior distributions to the parameters the software performs a maximum a posteriori estimation. In fact, if no particular prior distribution is defined, the software

assumes a uniform prior distribution within the bounds for each of the parameters. Effectively this changes the objective function back to the normal likelihood within those bounds. To be more specific, we extended the sentence: “Prior knowledge about the parameters can be considered **by prior distributions. The respective penalization of the likelihood leads to maximum a posteriori estimation.**” However, true Bayesian parameter inference is possible using our software as well. Several algorithms for MCMC sampling are implemented. We tried to be more specific in the sentence: “**A variety of Markov-chain Monte Carlo approaches are implemented to perform Bayesian inference (Raue et al., 2013a; Hug et al., 2013).**” We also improved the documentation about how to set prior distributions on the webpage <https://bitbucket.org/d2d-development/d2d-software/wiki/Priors%20in%20the%20d2d%20framework> and explained how priors enter in the objective function used for parameter estimation in the subsection “Objective function, likelihood and chi-square” of the webpage <https://bitbucket.org/d2d-development/d2d-software/wiki/Objective%20function,%20likelihood%20and%20chi-square%20in%20the%20d2d%20framework>

*(4) Does the software provide any assessment of how well the model fits and the validity of normality of Likelihood?*

Yes, quality controls such as QQ-plots of the residuals and auto-correlation of the residuals are available (see function `arPlotResiduals.m`). To further clarify, we added the sentence: “**Goodness of fit analysis, Likelihood ratio test, QQ-plots and auto-correlation of the residuals are available.**” We added a new page to the documentation on the webpage about this: <https://bitbucket.org/d2d-development/d2d-software/wiki/Objective%20function,%20likelihood%20and%20chi-square%20in%20the%20d2d%20framework>

*(5) Minor: the FAQ "How do I restart the solver upon a step input?" is linking to the wrong page on the web.*

Thank you, we corrected this mistake.

## **Answers to reviewer #2:**

*Data2Dynamics is a note on a software system for modeling dynamical systems. This is an important problem in which there is growing interest and the environment this package offers will make such analytical tools available to a wider audience at the computing / biology interface. I have taken a little long to review this short manuscript because I got my student working in the area of systems biology to download and test it out. He was able to comfortably set it up and run the Lokta-Volterra model on it and obtain results similar to what he can achieve with his own code. We have not tried larger models, but the software is well worth reporting on to the community and I am happy to recommend publication of the note.*

We thank the reviewer for his encouraging comment. We would also be very happy to welcome his students in our developer community to further improve the software.

### Answers to reviewer #3:

*Global comments:*

*Modeling dynamical systems is a key issue in systems biology. The software Data2dyn presents a modeling environment in Matlab that enables the user to define model, analyze them in terms of sensitivity and identifiability, infer state and parameters and design experiments.*

*Thus, this is a very useful tool well appropriate for modelers well used to Matlab. Moreover it is a solid piece of software that proved to be efficient in several DREAM challenge (ranked second and first at DREAM6 and 7).*

We thank the reviewer for his positive comments. He is right, the software formally ranked second in DREAM6. However, in that year there were two winners and both were awarded as “best performer” in their respective sub challenges.

*The parallelization at the level of the integration of several ODE systems (for instance, under different experimental conditions) seems to be very relevant for the Sensitivity analysis. On the contrary, I am not convinced that this parallelization makes sense for the joint estimation task.*

In our experience the parallelization does have a large impact on the runtime of parameter estimations. In each step of the optimizer the objective function has to be evaluated, including its derivatives in case of a deterministic optimizer. For every evaluation of the objective function each ODE system variant, and their respective sensitivity system for the derivatives, has to be solved independently. Therefore, parallelization of the individual ODE solutions offers a direct speed improvement. We would like to refer to the documentation webpage for further details:

<https://bitbucket.org/d2d-development/d2d-software/wiki/Parallelization>

*The environment is well presented in the note and well documented online.*

*However a figure showing the architecture of the code would be of great help in the note itself.*

We agree with the reviewer. However, we are very much constrained in space and number of figures in this publication format (Application Note: one figure, not more than two pages) and therefore cannot include such figure in the note itself. However, we created three new webpages that explain different aspects of the code architecture:

<https://bitbucket.org/d2d-development/d2d-software/wiki/CodeArchitecture>

<https://bitbucket.org/d2d-development/d2d-software/wiki/What%20are%20the%20most%20important%20functions%20in%20d2d%3F>

<https://bitbucket.org/d2d-development/d2d-software/wiki/What%20are%20the%20most%20important%20fields%20of%20the%20variable%20ar%3F>

*The code itself however lacks of useful comments. Weak points:*

*- Installation is a bit complex, I had trouble with Mac OS, but it was OK for one of our Matlab Windows installation as well as our computation clusters.*

We apologize for the difficulties. We tried to make the installation as easy as possible. However, the most difficult part is actually not the setup required for the software itself, but to configure MATLAB properly. Our software relies on the mex-functionality that allows compiling C code into MATLAB functions. This greatly improves computational performance. However, we have only limited options to improve this functionality for usage across platforms as it relies on MATLAB itself. Since version R2015a, MATLAB is shipped with a preinstalled compiler. This will greatly simplify this procedure from this version on. Around fifty users already installed our environment successfully. We are therefore convinced that there are no serious installation issues.

*- one cannot use all the features of the toolbox without the Symbolic toolbox. One suggestion should be to use the CV ode Wrapper of Joep Vanlier (toolbox PUA [APA]) which converts ODEs in C without needing the Matlab Symbolic Toolbox*

*Ref: van Riel, N. A., Tiemann, C. A., Vanlier, J., & Hilbers, P. A. (2013). Applications of analysis of dynamic adaptations in parameter trajectories. Interface focus, 3(2), 20120084.*

The Symbolic toolbox is indeed one of the key requirements. The functionality provided by our software goes far beyond converting the right hand side of the ODE to C code. Our software provides an entire modeling framework that appropriately links data, measured under various conditions, to a multitude of model simulations. The Symbolic toolbox is used to calculate the symbolic derivatives required to set up the sensitivity equations throughout the code. In the CVode wrapper by Vanlier *et al.* one had two options. Either use finite differences to compute the model sensitivities or compute the relevant Jacobian by means of symbolic differentiation. The latter, which is more numerically reliable, is actually analogous to the approach taken in our software and does depend on the symbolic toolbox as well. Actually, Joep Vanlier is co-author of this paper and this response is in concordance with him.

*- no a global optimization algorithm is used and it seems difficult to extend to other optimization functions.*

We do provide a variety of global optimization algorithms provided by the EvA2 toolbox. We added the sentence “**A variety of global optimization algorithms is implemented as well (Kronfeld 2010).**” and created a new webpage about this topic:

<https://bitbucket.org/d2d-development/d2d-software/wiki/Optimisation%20algorithms%20available%20in%20the%20d2d-framework>

The optimization function can be extended quite easily, see the documentation page:

<https://bitbucket.org/d2d-development/d2d-software/wiki/Objective%20function,%20likelihood%20and%20chi-square%20in%20the%20d2d%20framework>

*- the lack of strong structure (object and module) in the code makes it difficult to extend but for a user it is enough.*

We are aware of this issue and agree. Objects and modules would be a very convenient feature to create code that is easier extendible. MATLAB, however, possesses various restrictions for using object-oriented code in combination with the mex-functionality,

which leads to time-consuming memory copies when a mex-function is called. We preferred fast calculations to modular code in this case. If these restrictions are resolved in further releases of MATLAB, the code will be transformed into object-oriented language.

*In conclusion, the code as it is can be very usefull for practionners of modeling in systems biology. It would be nice to make some corrections to improve the code : installation, independence from the Symbolic toolbox, in the note, the architecture of the code for developers.*

We hope that the reviewer is satisfied with the corrections and explanations we provided.