Journal of Statistical Software September 24, 2017
Universität Innsbruck
Institut für Statistik
Universitätsstraße 15
6020 Innsbruck, Austria

Dear Sir / Madam,

We are glad that our manuscript entitled "*Dynamic Modeling, Parameter Estimation and Uncertainty Analysis in R*" is considered for publication in the Journal of Statistical Software. We thank the reviewers for their effort and time they have spent reading our manuscript in great detailed. We are happy about having received so many constructive comments and suggestions.

Enclosed in this document you find our point-by-point response to the reviewers' comments. We think that with the changes introduced, the manuscript has substantially improved. We hope that with the updated manuscript submitted today we meet all necessary requirements for publication in JSS.

Sincerely yours,

Daniel Kaschek

Dr. Daniel Kaschek
Hermann-Herder-Str. 3 – 79104 Freiburg, Germany
daniel.kaschek@gmail.com • *sysbio.uni-freiburg.de/kaschek*

Reviewer 1

The paper presents an R package 'dMod' focusing on ODE-based dynamic modeling and parameter estimation. The topic is relevant for a broad range of applications in science and engineering. Given the small set of existing R packages for inverse modeling, there appears to be room for another package and publication.
The main advantage of dMod over competitor packages (e.g. FME) appears to be the consequent use of derivative information obtained from symbolic expressions. Furthermore, the authors propose the trust-region approach to optimization which is dMod's default method. Finally, dMod provides functionality for parameter transformation and for linking observations with simulated states. dMod introduces a new syntax for concatenating and 'stacking' functions.
The paper is well structured (1. theory, 2. implementation, 3. example) and it is written in proper English.

Major issues

- In the introduction, the authors mention existing packages providing similar functionality as dMod. However, I would like to see a detailed but compact summary (e.g. as a table) pointing out when dMod is to be preferred over competitor packages.
  We have changed the list of key features of dMod into a table format, comparing dMod with the other packages mentioned in the introduction.
- The manuscript is quite long (31 pages) in ready-to-print format. The presented example covers 17 pages. I propose that the authors shorten the paper by porting part of the example to the package's vignette. The same is true for section 5 presenting extensions to dMod.
  We agree with the reviewer's concern that the paper is quite long. However, we feel that Section 4 is more than just an example. Section 4 illustrates many relevant cases encountered (and covered by dMod) when modeling dynamic systems and fitting those models to data. It illustrates both, the math introduced in Section 2 and the syntax of dMod of Section 3.
- The last author is also a co-author in half of the cited papers (10/20). This leaves room for doubts on whether similar achievements (or alternative attempts) of other work groups were properly acknowledged / taken into account.
  The dMod package has grown over the past years. It reflects the development within our research group. Over all these years we have considered numerous mathematical methods and algorithms, developed by other groups or by our own group, and have compared different approaches leading to the cited papers. These papers are cited because they document the essence of what is implemented in dMod. We tried to make this point more clear by the following new paragraph in the manuscript: *"The package has grown over the past years and reflects many developments and lessons learned from our research projects with time-resolved experimental data from Systems Biology. They form the basis of the development of dMod and are found as references in the following paragraphs."*

Detailed comments

- Different symbols in Eq. 1 and 2 are hardly distinguishable.
  We changed v to w.
- 2nd sentence of section 2.3: Replace 'for our application' and 'play a major role' by something more specific. Are you referring to parameter estimation for ODE models in general?
  We have replaced the corresponding text by: *"The dMod package supports derivative-based methods, and in particular the Newton method"*.

- Last sentence of sect. 2.3: Statement requires a citation.
  Equation (8) shows that the Newton step is not defined for rank-deficient Hessian since the inverse of the Hessian is not defined. Numerically, the Hessian can be close to being rank-deficient inducing a diverging Newton step. By definition of the trust-region radius, as the manuscript states, the step proposed by the trust-region optimizer can never be longer than this radius. We have changed the last sentence into *"This makes trust-region optimization the method of choice for dMod"*.
- Section 2.4: I'd like to see a clear distinction between (a) uncertainty in estimated parameter values and (b) predictive uncertainty. These two are linked but, as far as I can see, in this section you are referring to the former only. Modify section heading?
  We thank the reviewer for the comment. We changed the heading to *Parameter uncertainty analysis*.
- Section 2.4: I miss references regarding the profile likelihood approach. New references to Venzon and Moolgavkar, 1988, and Murphy and Van der Vaart, 2000, have been introduced.
- Section 3.1: Missing reference for the 'cOde' package. Note: The same is true for other R packages mentioned in the text as well (e.g. 'trust', 'parallel', 'rootSolve'). Need to add proper citations.
  Missing citations for all R packages mentioned in the text have been added, i.e., inline, trust, rootSolve, cOde, rPython and parallel.
- I didn't fully get why the * operator is redefined for function concatenation. Is this strictly necessary in the context of dMod or just a convenience feature?
  Mathematically, the formulations $(x \circ p)(t, \theta) = x(t, p(\theta))$ are equal. However, from the implementation point of view, `p(theta)` returns a list of parameters, one list entry per experimental condition. Therefore, the expression `x(times, p(theta))` does not make sense. The parameter argument is expected to be a vector and not a list of vectors. We define the "*" operator to provide the analogous to $x \circ p$ and handle experimental conditions to feed each parameter vector in the list `p(theta)` into the corresponding prediction function `x`.
- 3.5, first sentence: "captures"
  Has been replaced.
- The statement at the end of Sect. 3.7 is not easily understood since the meaning of 'constraintL2' wasn't introduced.
  We have changed the second last sentence of the subsection into *"They allow to define quadratic parameter priors or treat data points as parameters, respectively, as shown in Section 4."* to make clear that the explanation refers to the aforementioned functions `constraintL2()` and `datapointL2()`.
- Fig. 1: Mixing of math style and code style is confusing.
  We changed everything into code style.
- page 15, 1st paragraph: "experiments" or "starts"
  New sentence: *"The experiment starts ..."*
- page 15, 3rd paragraph: "thas"
  Has been replaced by *"this"*.
- section 4.7: very long section with lot of code; seems less appropriate for a journal paper
  We agree with the reviewer. The decision whether to present short code snippets or a complete, self-contained code was not easy. With Section 4 we aim at presenting a minimal problem which both illustrates typical issues encountered in dynamic modeling and ways to detect and solve these issues with dMod. In this way, Section 4 concretizes both, Section 2, the theoretical background, and Section 3, the implementation part.
- I didn't follow the text from page 26, last paragraph to the end of section 4.
  We reformulated the paragraph providing more details.

Dr. Daniel Kaschek
Hermann-Herder-Str. 3 – 79104 Freiburg, Germany
daniel.kaschek@gmail.com • *sysbio.uni-freiburg.de/kaschek*

- citations are sometimes parenthesized where this seems inappropriate, e.g. on page 28 it should probably read 'Merkt et al. (2015)'
  We thank the reviewer for pointing this out. `\cite` and `\citep` are now used appropriately.
- section 5.2: first R statement should cause an error due to missing quotes; Why is it necessary to export the output object to a file? Also, I see no corresponding call to 'readRDS'.
  The missing quotes have been corrected. The default usage of the function `\steadyStates()` is to run the command on the model and put the `\readRDS()` line into the model definition file to avoid that steady states are computed each time the model is sourced. We refer to `\readRDS()` in the paragraph below the output of the steady-state tool.

Reviewer 2

General

This is a well written paper and the authors are to be commended for addressing a very important area in technical computing; namely, the inverse problem associated with dynamic systems. This paper describes the new software package, dMod, which is written in the R language and that can be freely downloaded and used without financial cost. The package brings together a variety of existing methods and techniques that, together, result in a unique and comprehensive system.
The software is currently aimed at the identification of chemical system dynamics, but could clearly be used for identifying other system dynamics. A key element to this paper is the provision of a complete software example that demonstrates application to the identification of parameters of a reasonably complex 'bile acid transport' system. This example demonstrates usage of a good selection of dMod features. In addition to this detailed example, vignettes are provide by way of simpler examples, when the package is downloaded.
With the correction of minor typo's listed below, this paper is recommended for publication in JSS.

Running the Software

After some initial teething problems when attempting to run on a Windows platform, the software ran as described in the paper and reproduced the data and plots. The initial problem related to specifying the number of CPU cores to be used in the simulation, and this was solved by contacting the Editor and making minor changes to the code. The vignettes were also run without problem.

Recommendations and Minor Typo's

- Please introduce a check in the code to identify which software platform is being used (Windows/Linux/etc.), and set parameters accordingly.
  We have introduced a check in all functions with the "cores" argument. The user is notified on Windows that the number of cores has been set to 1 automatically.
- Abstract, 8th line: Please clarify the phrase, "... investigating invariance of the prediction under change of parameter values,". It is assumed that parameters should be unique to a specific data set and, therefore, that predictions will be expected to change if parameters change.
  Usually, the prediction changes if the parameters change. However, in some cases we find simultaneous changes of parameters that in combination leave the prediction unchanged. This can be due to a bad parameterization. Another reason can be the partial observation of the system. If we assume that some parameters cause changes of states that cannot be observed, the available data will not allow to determine these parameters. By the symmetry detection and profile likelihood method we have implemented two methods to find these non-identifiabilities.
- First line, final paragraph on page 6: It may be worth considering methods other than standard Newton, such as the 'Levenberg-Marquardt' method, which for some problem's provides superior convergence.
  We thank the reviewer for mentioning Levenberg-Marquardt. Some time ago we used both, Levenberg-Marquardt and trust-region optimization. In our experience we found the global optimum more frequently with the trust-region method. However, we agree that this might well depend on the system under consideration. Implementing an interface to Levenberg-Marquardt is definitely an option for the future of dMod.
- Third line, third paragraph of page 10: Change, "... by that fact that" to "... by the fact that".

We have changed this.

- Fourth line, third paragraph of page 12: Change, " taurocholic acide" to " taurocholic acid".
  We have changed this.
- Second line, first paragraph of page 15: Change, "The experiment start with" to "The experiment starts with".
  We have changed this.
- Second line, fifth paragraph of page 15: Change, "in thas case" to "in this case".
  We have changed this.
- Third line, second paragraph, section 4.4 of page 17: Clarify, "... a general quadratic prior can be entailed on all parameters ...".
  Adding a weak prior to the parameters turns out to be very helpful when fitting with a trust region optimizer without fixed bounds for the parameter values. However, this concept has never been published and relies more on our experience.
  We introduced the function constraintL2() now with the general concept of prior knowledge on parameters that most communities should be familiar with and afterwards suggest weak priors as being helpful in the case of non-identifiable parameters.
- Sixth line, second paragraph of page 29: Change: "could e.g. be" to "could, for example, be" (more fluent).
  We have changed this.
- In the longer term the authors may consider adapting this package to other technology areas. One such area could be real-time, online 'fault detection' - detecting equipment parameter changes over time. This is an active area of research.
  We thank the reviewer for the suggestion. Indeed, parameter estimation has many applications and also, dMod as ist stands is not restricted to Systems Biology applications.
- To actively promote usage of the package the authors may consider publishing more example applications, particularly those based on real measured data sets.
  Thank you for this suggestion. More examples with real data will definitely come in the near future.

Dr. Daniel Kaschek
Hermann-Herder-Str. 3 – 79104 Freiburg, Germany
daniel.kaschek@gmail.com • *sysbio.uni-freiburg.de/kaschek*

Reviewer 3

The package dMod is a very useful addition to the suite of tools available in R for fitting nonlinear dynamic models to observational data, presented to the community by a leading research group in this area. As the paper notes, there is no excess of such tools in R, especially relative to the wide use of differential equation models in the physical and biological sciences. A paper to present, explain, and illustrate the dMod package is fully justified for publication in J. Stat. Software and my comments are thus limited to the question, how can this paper be improved?

- The most important areas for improvement are in explaining the modeling and statistical assumptions underpinning the methods implemented in dMod, and what those imply for its applicability to a given set of data and for the interpretation of the results. From beginning to end, the underlying dynamic model is assumed to be a system of ordinary differential equations (ODE). This is stated at the start of section 2, and might be inferred from the Abstract, but a clear should come much sooner so that readers know from the outset what class of models dMod can be applied to. I suggest that it could even go into the title: "Dynamic Modeling, Parameter Estimation and Uncertainty Analysis in R for Ordinary Differential Equations." And readers should be alerted, early, that "dynamic model(ing)" always means "ordinary differential equation model(ing)" in this paper.
  The reviewer is right about our focus on ordinary differential equations. We have now emphasized at different positions in the abstract and in the introduction that dMod deals with ordinary differential equations only to avoid false expectations.
- A related key point, which should also be emphasized in the Introduction, is that the underlying dynamics are assumed to be deterministic, with stochasticity only in random measurement errors. Systems of chemical reactions are often specified by differential equations, but simulated as discrete stochastic molecular events by Gillespie's method or an approximation (such as τ -leaping or approximation by a stochastic differential equation). A crucial assumption for the dMod methods is that the system dynamics are perfectly deterministic, and appear somewhat noisy only because they are observed with error. This issue is also relevant to the discussion of available packages in the Introduction. The authors write "there are not more than four R packages published on the topic" on CRAN, but that is a statement about packages for exactly the kind of models that dMod handles. For nonlinear dynamic models with some stochasticity in the dynamics, there are two more that I know of, CollocInfer and pomp. nimble might count as well, since numerical solution of an ODE or stochastic ODE system could be embedded in a BUGS model formulation, and there might well be others. These packages should also be discussed at this point in the paper, to help clarify what problems dMod is and isn't suitable for.
  We have referenced the additional packages suggested by the reviewer. We have adapted the introduction making clear that deterministic dynamic systems are only a part of the story: *Taking a broader perspective on the topic of dynamic modeling and inference, we find more packages, e.g., dealing with discrete-time and continuous-time stochastic systems (CollocInfer) or with the statistical inference of partially observed markov processes (pomp).* In the discussion what dMod is, we have added a sentence on the limitations of dMod: *The dMod package deals with noise in the observation but not with noise in the dynamics. Therefore, the application of dMod is restricted to systems that are described or can be approximated by a deterministic set of differential equations.*
- On the same theme, the likelihood used by dMod is (if I understand correctly) a very specific likelihood, which is based on assuming that the data consisted of Gaussian errors added to noise-free solutions of the fitted ODE model. When readers are told that dMod does maximum likelihood parameter estimation (top of p. 3 in the manuscript) they should also be told what likelihood is used. The description of the likelihood (section 3.2) also needs to be clearer about exactly what function dMod optimizes, and how much control (if any) a user has in specifying the likelihood:

We agree with the reviewer that the user should know which likelihood ist optimized. We have added a paragraph in Section 3.7 referring to the equations in Section 2.2. We have further commented on the topic within the next bullet points.

- Can the error distribution be anything besides Gaussian? The normL2 function has an argument errmodel, which is perhaps how users can specify to some degree the measurement error distribution that is used to compute the likelihood, but ?normL2 only explains that errmodel is an object of class obsfn and does not give any explanation of how it can be used or give any examples of its use. A search on errmodel in the manuscript PDF did not find any mention of it in the manuscript.
  Can the errors be correlated – over time, between one state variable and another?
  Right now there is no function in dMod that produces a log-likelihood objective function from an arbitry symbolic expression of the noise distribution. We have thought about implementing the Poisson distribution which would be beneficial for count data with many small or zero values. However, this would be something for the future. The log-likelihood provided by the normL2 function assumes un-correlated, normally distributed noise. In addition, observation functions allow to take transformations of the data into account, e.g., log-transform of the data in case of log-normally distributed noise or combinations of observed data points to remove correlations amongst them.
- Are the σ_i assumed to be known, or are they estimated? The text on p. 6 talks about them being estimated, but in the example on p. 15 they are specified.
  When the σ_i are estimated, can users specify a variance-mean relationship (variance proportional to squared mean, for example, with the constant of proportionality to be estimated), or specify that there is one σ per state variable (e.g., one σ for all observations of X 1 and a second for all observations of X 2 ?)?
  By default we expect that values for σ_i are provided with the data. However, in recent versions of dMod we support error functions. The error function is formally an observation function that is defined on the observables, e.g., $\sqrt{\sigma_{abs}^2 + y_{obs}^2\sigma_{rel}^2}$, where y_{obs} is an observable (a function of the internal states) and σ_{abs} and σ_{rel} are error model parameters. Each observable can have its own error function. We have extended Section 3.7 to cover this topic.
- The questions listed above are, I think, crucial for understanding the scope of what dMod can do, so the manuscript should make the answers clear for readers. Maximimum likelihood is indeed wonderful and powerful, but only if you're maximizing the (at least approximately) right likelihood for the data.

◦ There also needs to be much more discussion about what it means statistically to add a prior to the objective function, and about what that implies for results and their interpretation (in particular, for profile likelihood plots and their interpretation via chi-square cutoff values). The case-study in the manuscript illustrates using a prior to diagnose parameter non-identifiability (Fig. 8), but what about situations where users really want to put a prior on parameters and make that a part of the likelihood? Does dMod properly deal with situations like that, and if so, how should the prior be specified to make it compatible with the rest of the likelihood function, and how should model output about uncertainties be interpreted?
  We have extended Section 4.4 discussing the usage of priors and the function `constraintL2()` in more detail. As stated in the new text, we distinguish between prior information and a general "uninformative" prior. Prior knowledge is included like a data point and modifies the log-likelihood accordingly. A general prior on the other hand is an auxiliary function that is used during model development. The complexity of the final model should reflect the information in the data such that regularization by a general prior is not necessary any more. With the `attr.name` argument in dMod we have a way to deal with the distinction of prior knowledge and general prior.

◦ My last substantive suggestion (and in this case I really mean “suggestion”) is to add an explicit

speed comparison for dMod versus a more widely used method such as perhaps FME. Users with limited experience in this domain may not understand that speed is absolutely crucial, because the objective functions for nonlinear ODEs frequently have many local minima and the only (semi-)reliable way to find the global optimum is through massive multi-start optimization, trying dozens (if not thousands) of initial parameter guesses. The speed that comes from C code generation combined with sensitivity equations is a very, very appealing feature of dMod and the authors should brag about it a bit. There is a pretty significant learning curve to master the new syntax and functions in dMod. Potential users are more likely to become users, I think, if they see what the payoff will be.
We thank the reviewer for this suggestion. It is not easy to provide a fair speed comparison for the different frameworks because they all have a different focus and offer different options how to implement the estimation problem. However, we decided to work out the aspect of performance improvement by the combination of compiled code and symbolic derivatives that we offer in dMod. We have added a new section at the end of the example and have tried to classify the available packages accordingly.

Minor comments (in order of appearance)

- The paper's English is perfectly understandable but not quite idiomatic (e.g. "values decide upon" in the Abstract, and many examples throughout the paper). I think it would be helpful if a native English speaker could be convinced to proofread the paper sentence by sentence.
  Since the manuscript is quite long, we have convinced somebody to proofread at least the abstract and introduction. We hope that this represents an acceptable compromise.
- Be careful about quotation marks in Tex, e.g. "open" versus “open”. There are many instances.
  We have corrected quotation marks. Quotations in code mode are still upright.
- p.4 The notation in equation (1) is totally unfamiliar to me, and the notation in equations (2) and (3) is not clear. Please use standard mathematical notation (notation that would be encountered by engineers in a first undegraduate course on differential equations, for example) or fully explain the notation. I don't know what v or v' are, I don't know how S would be obtained from a collection of $v' - v$ vectors, I don't know what "$\cdot$" means in equation (2) and (3).
  We have split equation 2 into two equations. The first one formulates the chemical reaction whereas the second equation shows the corresponding differential equation. We have added an explanation of the stoichiometric matrix and how it is constructed from the coefficients ν and ν'. We removed the product sign "$\cdot$" between the matrix S and the vector w.
- p.7 "assume Wilks' theorem". Wilks' Theorem is a large-sample result, so the key assumption is not the theorem itself. Rather, the assumption is that the sample size is large enough to justify tests based on critical points of the the asymptotic chi-square distribution.
  We have corrected the sentence into ... *we assume a sufficiently large sample size to apply Wilks' theorem* ....
- p.10 I don't think that the $\oplus$ lines help. The lines immediately below them explain what's going on much more clearly.
  The reason for the $\oplus$ lines is the analogy to the "+" operator. We have added a footnote to highlight the analogy between mathematical formulation and code, similar to $\circ$ and "*".
- p.11 I think it would be useful to start the case study with one- or two-paragraph overview. What are the (big picture) steps in the process of going from a data set on the computer and a differential equation model on a pad of paper, including for each step the decisions that the user has to make (observation equations, parameter transformations, error model, etc.) and the key dMod functions that are used.
  We thank the reviewer for the suggestion. We have added a paragraph at the beginning of

Section 4 to give an overview of the subsequent steps.

- I end this review by confessing that I have not worked through the case study in the paper to make sure that all the code is right and that the package performs as advertised. However, I think Windows users should be warned that they need to have Rtools installed (I can't say what is required on other OS's). I recently moved to a new Windows workstation at work, and it took me several days to figure out that dMod wasn't working for me because I had neglected to re-install Rtools on my new mac

  In the paper we now point out that additional information on system requirements and installation can be found on the project's github page. There we have a link to RTools and explain how to install the package from github.

Dr. Daniel Kaschek
Hermann-Herder-Str. 3 – 79104 Freiburg, Germany
daniel.kaschek@gmail.com • *sysbio.uni-freiburg.de/kaschek*