Supplemental Material for Multivariate Behavioral Research
User's guide to the BHOUM software

Zita Oravecz
Human Development and Family Studies, Pennsylvania State University

Francis Tuerlinckx
Psychology, University of Leuven, Belgium

Joachim Vandekerckhove
Cognitive Sciences, University of California, Irvine

Abstract

In this paper we present the Bayesian hierarchical Ornstein-Uhlenbeck Modeling (BHOUM) MATLAB toolbox, which can be used to simultaneously analyze continuous longitudinal measurements of two linked variables, based on the theory of the two-dimensional (bivariate) Ornstein-Uhlenbeck (OU) process. Our proposed method represents a process model approach to modeling change in two longitudinal measures here, which means that we assume that changes over time in the two manifest longitudinal variables are governed by an underlying latent process. Since we also assume that changes in the two longitudinal variable are interlinked, this process model approach can be visualized as a latent process evolving over time in two dimensions, these two dimensions being defined by our two longitudinal variables of interest. The longitudinal measurements we collect are samples from the current position of this latent process, perturbed by measurement error. Data for each person can be described by a unique OU process. Within person, this process captures meaningful properties of bivariate change over time by three dynamical parameters: a bivariate attractor point, intra-individual variability and a centralizing tendency. Cross-effects tying the changes in the two longitudinal variables together can also be included. The BHOUM toolbox applies a hierarchical extension of the two-dimensional OU process, allowing for inter-individual variability in all OU parameters. BHOUM is freely available as a standalone program featuring a user-friendly graphical user interface or as MATLAB package. BHOUM offers graphical tools for data conversion, model specification, result summaries and model checking.

Keywords: HOU model, MATLAB toolbox, hierarchical model, cross-correlation, autoregressive

Introduction

While statistical inference in the Bayesian framework for hierarchical OU (HOU) models was introduced by Oravecz, Tuerlinckx, and Vandekerckhove (2009, 2011); Oravecz and Tuerlinckx (2011), a practical guide for executing the data-analysis did not exist. We fill in the gap here by first addressing data formatting issues, then issues such as posterior predictive model checking, and finally demonstrate the diverse results that can be derived. In this paper we introduce the Bayesian hierarchical Ornstein-Uhlenbeck Modeling (BHOUM) MATLAB toolbox that is a user-friendly parameter estimation engine with graphical user interface. The software can be downloaded from the first author's website, `www.zitaoravecz.net`. Details of statistical inference derived in the Bayesian framework can be found in Oravecz et al. (2011); notational conventions in the toolbox correspond to that paper. BHOUM is primarily intended as a standalone software program (no MATLAB licence is required), controlled through a graphical user interface (GUI). [1] Although BHOUM requires no coding from the user, all MATLAB scripts are available for download.

A concise description of the hierarchical Bayesian Ornstein-Uhlenbeck model

Since this document is only a supplemental material for the paper from Oravecz, Tuerlinckx, and Vandekerchove (2016), which contains a detailed description of the hierarchical OU model, here we only provide a brief technical description of the model to remind the reader to the notation etc. introduced in the main paper.

A typical structure for an intensive longitudinal dataset would be the following: two longitudinal variables for a person p ($p = 1, \ldots, P$) are measured at n_p time points: $t_{p1}, t_{p2}, \ldots, t_{ps}, \ldots, t_{p,n_p}$ and denoted as $\boldsymbol{Y}(t_{ps}) = (Y_1(t_{ps}), Y_2(t_{ps}))^{\mathrm{T}}$ at time point t_{ps}. The index s denotes the s^{th} measurement occasion of that individual. In the HOU model we assume that these observations are functions of a latent underlying state denoted as $\boldsymbol{\Theta}(t_{ps}) = (\Theta_1(t_{ps}), \Theta_2(t_{ps}))^{\mathrm{T}}$ and some measurement error. The underlying latent state is assumed to be governed by a two-dimensional, or bivariate, OU process. For simplicity, we use only the indices p and s when denoting parameters or data which are related to the specific observation at t_{ps}. Then an HOU model for a single person p can be written as:

$$\boldsymbol{Y}_{ps} = \boldsymbol{\Theta}_{ps} + \boldsymbol{\epsilon}_{ps}, \tag{1}$$

where $\boldsymbol{Y}_{ps}$ stands for the observed random vector, $\boldsymbol{\Theta}_{ps}$ for the latent state (or true score) and $\boldsymbol{\epsilon}_{ps}$ for the measurement error with the distributional assumption: $\boldsymbol{\epsilon}_{ps} \overset{\text{iid}}{\sim} N_2(\mathbf{0}, \boldsymbol{\Sigma}_{\epsilon})$. Based on the

[1] The standalone BHOUM version with the accompanying free MATLAB Compiler Runtime (MCR) has been tested for Windows 32bit and 64bit. If the user does not want to install MCR because they have a MATLAB licence already, that MATLAB should be a 32bit version.

Correspondence concerning this article may be addressed to: Zita Oravecz, PSU, Department of Human Development and Family Studies, 216 HHD Bldg., State College, PA 16801; phone: +18148651546 ; email: zita@psu.edu

theory of the OU process (Uhlenbeck & Ornstein, 1930; Dunn & Gipson, 1977), the conditional distribution of $\mathbf{\Theta}_{ps}$ given $\mathbf{\Theta}_{p,s-1}$ is normally distributed as follows (for $s > 1$):

$$\mathbf{\Theta}_{ps}|\mathbf{\Theta}_{p,s-1} \sim N_2\left(\boldsymbol{\mu}_{ps} + e^{-\mathbf{B}_p(t_{ps}-t_{p,s-1})}(\mathbf{\Theta}_{p,s-1} - \boldsymbol{\mu}_{ps}), \mathbf{\Gamma}_p - e^{-\mathbf{B}_p(t_{ps}-t_{p,s-1})}\mathbf{\Gamma}_p e^{-\mathbf{B}_p^{\mathrm{T}}(t_{ps}-t_{p,s-1})}\right). \quad (2)$$

Parameter $\boldsymbol{\mu}_{ps}$ is the person-specific bivariate baseline position, that can change across time as a function of time-varying predictors. Since this is the point towards which regulation (see later) occurs, we also call it (bivariate) attractor. Variation around this attractor is modeled through $\mathbf{\Gamma}_p$, which is a person-specific intra-individual covariance matrix. The model assumes that there is some level of regulation towards this attractor over time and the level of this is modeled through the matrix $\mathbf{B}_p$. Note that the presence of indices p in Equation 2 reflects that all driving parameters of the OU-process are allowed to be person-specific (and can be regressed on time-invariant predictors, see later). Finally, for the the first observation, $\mathbf{\Theta}_{p1}$ it is assumed that $\mathbf{\Theta}_{p1} \sim N_2(\boldsymbol{\mu}_{ps}, \mathbf{\Gamma}_p)$.

The latent attractor parameter (also called *home base*) $\boldsymbol{\mu}_{ps}$ can be made function of person-specific time-varying and a person-specific time-invariant aspect. For including time-invariant predictors, it is assumed that k time-invariant predictors are measured with x_{jp} denoting the score of person p on predictor j ($j = 1, \ldots, k$). This model also assumes a time-invariant intercept, regardless of whether other time-invariant predictors are included. These predictors are collected into a vector of length $k + 1$, denoted as $\boldsymbol{x}_p = (x_{p0}, x_{p1}, x_{p2}, \ldots, x_{pk})^{\mathrm{T}}$, with $x_{p0} = 1$. Regarding time-varying predictors, suppose we measure predictor z for person p, and $z = 1, \ldots, E$, then the vector $\boldsymbol{z}_{ps} = (z_{ps1}, \ldots, z_{psE})^{\mathrm{T}}$ collects all these values. No intercept is introduced in the vector $\boldsymbol{z}_{ps}$. The index s indicates that values may change from one observation point to another.

Hence, the population distribution of $\boldsymbol{\mu}_{ps}$, as regressed on the time-invariant and time-varying covariates and allowing for person-specific random deviation can be written as follows:

$$\boldsymbol{\mu}_{ps} \sim N_2\left(\mathbf{\Delta}_{p\mu}\boldsymbol{z}_{ps} + \boldsymbol{A}_{\mu}\boldsymbol{x}_p, \mathbf{\Sigma}_{\boldsymbol{\mu}}\right), \quad (3)$$

where the covariance matrix $\mathbf{\Sigma}_{\boldsymbol{\mu}}$ is defined as follows:

$$\mathbf{\Sigma}_{\boldsymbol{\mu}} = \begin{pmatrix} \sigma^2_{\mu_1} & \sigma_{\mu_1\mu_2} \\ \sigma_{\mu_1\mu_2} & \sigma^2_{\mu_2} \end{pmatrix}. \quad (4)$$

The matrices $\mathbf{\Delta}_{p\mu}$ and $\boldsymbol{A}_{\mu}$ are parameter matrices of dimension $2E \times P$ and $2 \times (k+1)$, respectively, containing the regression weights for the time-varying and the time-invariant predictors respectively.

The matrix $\mathbf{\Gamma}_p$ represents the stochastic or intra-individual variability. Its diagonal elements (i.e., γ_{1p} and γ_{2p}) determine the intra-individual variances in the two dimensions, and the off-diagonals can be decomposed into $\rho_{\gamma_p}\sqrt{\gamma_{1p}\gamma_{2p}}$, where ρ_{γ_p} is the cross-correlation of the observations. Since the diagonals are constrained to be positive, their population distributions are modeled through a lognormal distribution, or equivalently their log-transformed values are sampled from normal distributions. For γ_{1p} that is:

$$\log(\gamma_{1p}) \sim N(\boldsymbol{x}_p^{\mathrm{T}}\boldsymbol{\alpha}_{\gamma_1}, \sigma^2_{\gamma_1}),$$

with $\boldsymbol{x}_p^{\mathrm{T}}$ the vector of covariates with $k + 1$ components (of which the first one is the constant 1). The vector $\boldsymbol{\alpha}_{\gamma_1}$ contains the (fixed) regression coefficients for the covariates. The parameter $\sigma^2_{\gamma_1}$ is the residual variance in the first dimension (related to the first longitudinal variable that we are modeling), after having taken the predictors into account. Please note that the level-1 (population) distribution of γ is modelled on the log-scale, therefore the variance parameter is also on the log-scale. If only the intercept is present in the model, $\sigma^2_{\gamma_1}$ reflects the total amount of variance present in the population in the log-variance of the first dimension. Similar logic applies to the modeling of γ_{p2}.

The cross-correlation ρ_{γ_p} is bounded between -1 and 1. By taking advantage of the Fisher z-transformation $F(\rho_{\gamma_p}) = \frac{1}{2}\log\frac{1+\rho_{\gamma p}}{1-\rho_{\gamma p}}$, we can transform the values to the real line:

$$F(\rho_{\gamma_p}) \sim N(\boldsymbol{x}_p^{\mathrm{T}}\boldsymbol{\alpha}_{\rho_\gamma}, \sigma^2_{\rho_\gamma}).$$

with $e_{p\gamma 2} \sim N(0, \sigma^2_{\rho_\gamma})$. Again, $\boldsymbol{\alpha}_{\rho_\gamma}$ contains $k+1$ regression weights, $\boldsymbol{x}_p^{\mathrm{T}}$ the k predictor values for person p with 1 for the intercept and $\sigma^2_{\rho_\gamma}$ represents the variation in the population in terms of cross-correlation.

The regulatory force or centralizing tendency is parameterized by the matrix $\mathbf{B}_p$, which is decomposed in the same manner as $\boldsymbol{\Gamma}_p$, henceforth it is assumed to be positive definite. The elements of the person-specific matrix $\mathbf{B}_p$ are assumed to come from population distributions that are defined in the same manner as for $\boldsymbol{\Gamma}_p$, and can be made the function of time-invariant predictors the same manner. This way it contains two centralizing tendencies, one for each dimension (i.e., β_{1p} and β_{2p}), and a standardized cross-centralizing tendency parameter (ρ_{β_p}). These parameters control the strength and the direction of the regulation towards the attractor (baseline). As can be seen above, all driving parameters of the OU process are person-specific and assumed to be drawn from joint distributions, this way making the OU model fully hierarchical. The parameter space of this model is highly multidimensional, therefore estimating parameters in the classical statistical framework would require solving a high dimensional integration problem. We overcome this by fitting the model in the Bayesian framework and relying on Markov chain Monte Carlo algorithms. These are implemented in the BHOUM toolbox.

The BHOUM toolbox implements a specific MCMC algorithm, the Metropolis-within-Gibbs sampler, to estimate the HOU model parameters in the Bayesian framework. In this algorithm, alternating conditional sampling is performed: The parameter vector is divided into subparts (a single element or a vector), and in each iteration the algorithm draws a new sample from the conditional distribution of each subpart given all the other parameters and data; these conditional distributions are called full conditional distributions. More details on the sampling algorithm and the derived full conditional distributions of each parameter, as well as the simulation studies testing the accuracy of the algorithm can be found in Oravecz and Tuerlinckx (2008) and Oravecz et al. (2011).

In the estimation algorithm, several of these Markov chains are initiated from different starting values based on the range of the data to explore the posterior distribution. The BHOUM toolbox offers the Gelman-Rubin $\hat{R}$ statistic (for more information, see Gelman, Carlin, Stern, & Rubin, 2004) as convergence check statistic.

Analyzing data with BHOUM

In this section we provide a step-by-step description of how to use the BHOUM toolbox for statistical inference with hierarchical OU models. The BHOUM toolbox contains several functions to deal with various aspects of the Bayesian statistical inference. These can be divided into three sections: (1) the *Data reader*, (2) the *Model specifier* and (3) the *Result browser*. Each section is accompanied by a user friendly graphical interface. Subsequent subsections will then demonstrate the use of each of these parts by fitting the model to emotion research data. First we will discuss the data format. To follow along with sample data, the user may consult the example data set ($example_data.csv$) that downloads with the program.

Data format

The BHOUM toolbox requires the data file to be in a specific format. Figure 1 shows an extract from the example data (described above) in a compatible format, opened in a spreadsheet program. The variables are named with strings, as displayed in the first row of the data set. As common in intensive longitudinal data analysis, the data are in wide format with the observed longitudinal variables and their measurement times listed in separate columns, one row corresponding to one observation, and a separate column for participant identification numbers. Figure 1 displays a small extract from such a data file. The first column, labeled *PP*, contains the person identifiers. The next two columns show the two intensive longitudinal variables to be modeled through a bivariate OU process are labeled PL and AC (pleasantness and activation levels). If the more general (but less informative) labels $Y1$ and $Y2$ are used as headers, BHOUM would automatically identify these columns as the OU-process dimensions (dependent variables).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	PP	PL	AC	day	hour	minute	CeHours	Hours	Z1	Z2	X1	X2	X3	X4	X5	X6	X7	X8	X9
2	1	8	4.6	17	16	58	16.97	16.97	4.97	143.87	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
3	1	7.4	1.2	17	18	25	18.42	18.42	6.42	195.17	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
4	1	7.6	1.6	17	19	25	19.42	19.42	7.42	233.01	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
5	1	7.7	3.3	17	21	22	21.37	21.37	9.37	312.53	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
6	1	-999999	4.4	18	8	44	32.73	8.73	-3.27	-67.73	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
7	1	6.3	4.9	18	9	57	33.95	9.95	-2.05	-45.00	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
8	1	7.4	3	18	11	28	35.47	11.47	-0.53	-12.52	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
9	1	7	6.2	18	12	54	36.90	12.90	0.90	22.41	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
10	1	6.4	7	18	14	39	38.65	14.65	2.65	70.62	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
11	1	6.4	6.2	18	15	24	39.40	15.40	3.40	93.16	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
12	1	7.1	4.8	18	16	28	40.47	16.47	4.47	127.15	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
13	1	4.7	2.7	18	18	34	42.57	18.57	6.57	200.72	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
14	1	7.7	1.5	18	20	8	44.13	20.13	8.13	261.35	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75
15	1	8	1.9	18	20	36	44.60	20.60	8.60	280.36	2.42	3.83	4.00	3.30	3.80	1.92	6.60	3.83	2.75

Figure 1. : Sample from a data set format readable with BHOUM.

The next three columns provide the time when the measurement was taken. Then BHOUM requires a less-common input for the 7^{th} column: the cumulative measurement time of the observation in hours, see *CeHours* in Figure 1.[2] A very important property of the measurement time is that it is cumulative within person. That is, if the measurements are taken over several days, the *CeHours* variable within an individual does not start over with each day. As can be seen in Figure 1 for example, the 5^{th} cumulative measurement time point of the person indexed as 1 is 32.73. This 5^{th} measurement is taken at 8:44 am on his second measurement day therefore $(24 + 8) + 44/60 = 32.73$.[3] The first 3 columns plus the cumulative measurement time column *CeHours* represent all the necessary data for fitting an hierarchical Ornstein-Uhlenbeck model. If the *CeHours* column were labelled as *time* (default label), the program would automatically recognize it and load it to the right field.

The columns labeled *Hours*, *Z1* and *Z2* next illustrate some time-varying covariates. Here we mention again that only the two-dimensional baseline of the OU-process can be made a function of time-varying covariates. A straightforward time-varying covariate is measurement time itself. Relating it to our example, it seems to be interesting to investigate whether the time of the day affects how pleasant and activated the participants feel on average. Therefore we add these covariates to the model nested within the day, and not cumulatively like we did for the measurement time column (*CeHours*). To reiterate, in Figure 1 column *Hours* shows the measurement time in hours within the daily cycle; while columns, labelled as *Z1* and *Z2*, are possible time-varying covariates that are functions of *Hours*: *Z1* is the measurement time in hours centered around the middle of the day, namely 12 pm (noon), and *Z2* is the squared measurement time in hours, also centered around noon. These columns allow us to model the bivariate baseline parameter as a function of linear and quadratic time effects, and our intercept will be the average attraction point, namely the attractor at 12 pm.

If the user measured any other variables (for example variables such as anger level, appraisal level, or body temperature) at the same time as the OU-process variables (column *PL* and *AC*) were measured, then those can be added to the analysis as well. The rest of the columns (11-19) show possible time-invariant covariates. As can be seen, the values of the different time-invariant covariates have to be listed in separate columns for each measurement occasion, meaning that the same value is repeated several times for all the observations of one participant. For example in Figure 1, the 11^{th} column is an example of a time-invariant covariate, a participant's level of neuroticism, that is repeated as many times as many observations are recorded.

The default missing value assigned by the program is NaN. However, the example data set in Figure 1 coded as such that-999999 stands for the missing values, as can be seen for the fifth pleasantness observation for the first person. If any other value than NaN is used to code missing values, the user needs to enter that missing value code in the second panel (see later).

[2] The program is optimized for data sets in which several measurements are taken over the course of a day. However, if measurements are taken weekly for example, days might be considered as a measurement time unit.

[3] For MS Excel users, a short piece of iterative code can be used to compute the cumulative time since the start of the first day of the study. If *PP* indicates the column where person indexes are stored, and *Day* the column where the day of the month is stored, then the line =IF(PP3<>PP2,0,IF(Day3<>Day2,A2+1,A2)) (written in field A3) will track the number of days since the start of the study for every person. Excel's auto-fill feature can compute this for every subsequent cell. A2 will need to be initialized to 0. In a new column, then, write =Hour3+Minute3/60+24*A3, where *Hour* is the column that contains the hour of the measurement, and *Minute* the column that contains the minute at which it was made. After using the auto-fill feature, this column will contain the number of hours since the start of the first day of the study, as needed.

Data reader

Figure 2 shows the *Data reader* window that opens when clicking on BHOUMtoolbox.exe (or by typing in BHOUM when the user runs the program from MATLAB). The screenshot shows *Data reader* after having browsed for a data set called $example_data.csv$. This data set is very similar to the example dataset described above. Once the dataset path is loaded, the *Load dataset* button in BHOUM can be pressed to read in the data. At this point the program reads in all default variable headers, and these automatically populate their corresponding fields. Optionally, by clicking on the *View dataset* button a visual representation of the loaded data is shown.

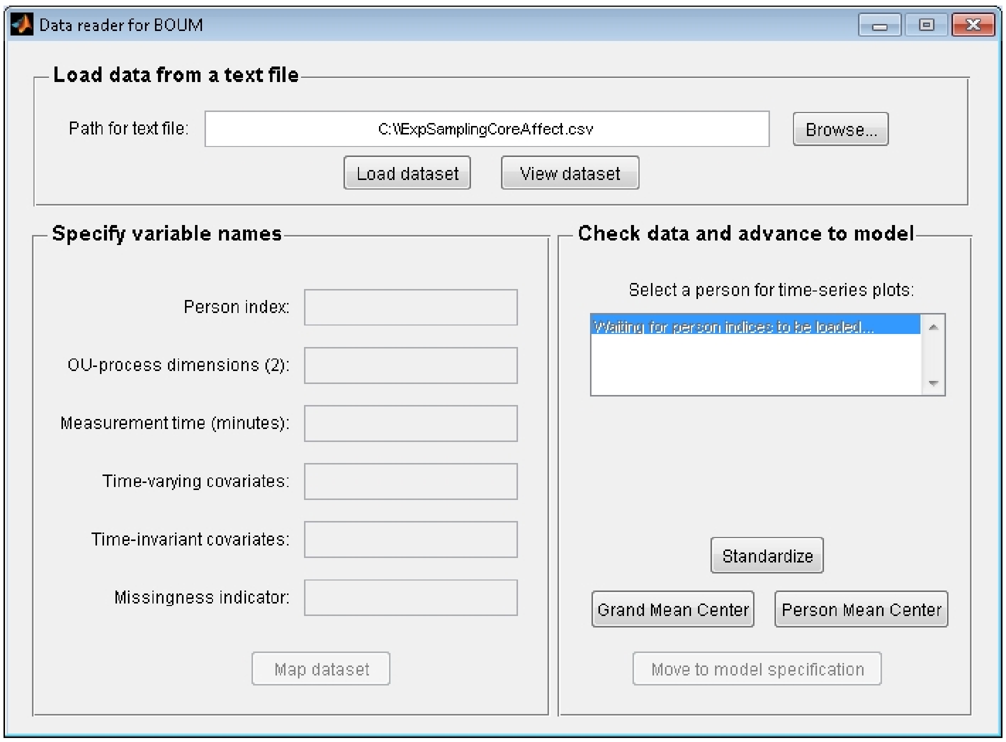

Figure 2. : Screenshot of the very first window of BHOUM. This window, called *Data reader* , opens when the program starts.

The *Data reader* window is divided into three panels: the already described top panel for loading the data, the bottom left panel to specify the variables and the bottom right panel to check these specifications before proceeding to the *Model specifier* window. Having loaded the dataset the second panel becomes active. As shown in the sample dataset (Figure 1), many variable headers in the first row are set to the default string values which are automatically recognized by the program. These are the person index *PP*, and the covariate variables (Z-s and X-s). While these variables headers populate the appropriate fields in the bottom left panel, specific variable names for your dataset must be entered manually, namely *PL* and *AC* for the OU-process dimensions, and *CeHours* for measurement times. The default header for these variables are *Y1*, *Y2* and *time*.

Finally, the missingness field can be left empty if the missingness indicator was NaN, but if it was set to -999999 like in Figure 1, -999999 would have to be entered.

Figure 3 displays the completed state of the *Data reader*. In the bottom left panel we see that both time-invariant covariates (ranging from $X1$ to $X10$) and time-varying covariates ($Z1$ and $Z2$) were read in successfully. When loading the data, BHOUM automatically populated these covariate fields with default variable headers. If the user does not wish to use those variables, or if there are others without default variable headers, they must be removed or added manually through the BHOUM interface.

The bottom right panel of Figure 3 provides the user with a graphical illustration of the observations of selected participants (one at a time), offering further verification that the data were mapped correctly. The lowermost button takes users to the *Model specifier* window, for specifying the type of HOU model they wish to estimate.

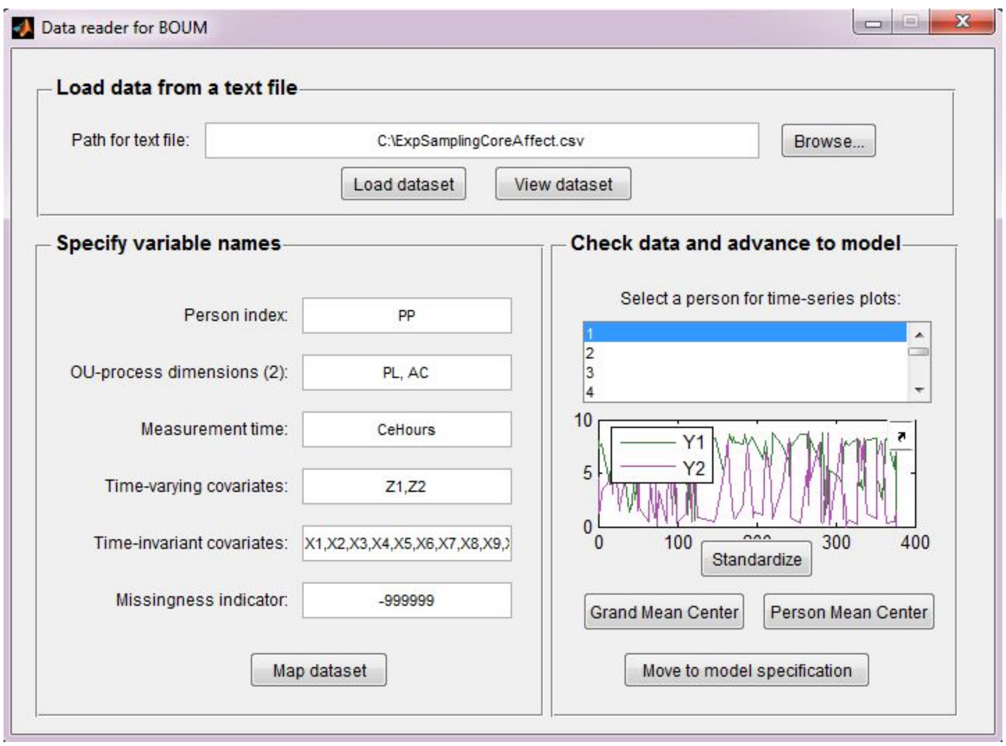

Figure 3. : Screenshot of the first window of BHOUM: A ready *Data reader*

As soon as all fields in the left panel are set to the required values (covariate information is not necessary however, see later), the button 'Map dataset' can be clicked. This will convert the user's data into the necessary format for estimating the HOU model. Executing the mapping results in pop-up window with an overview of the imported data. An example can be seen in Figure 4. This output allows the user to check whether their variables were specified and read correctly.

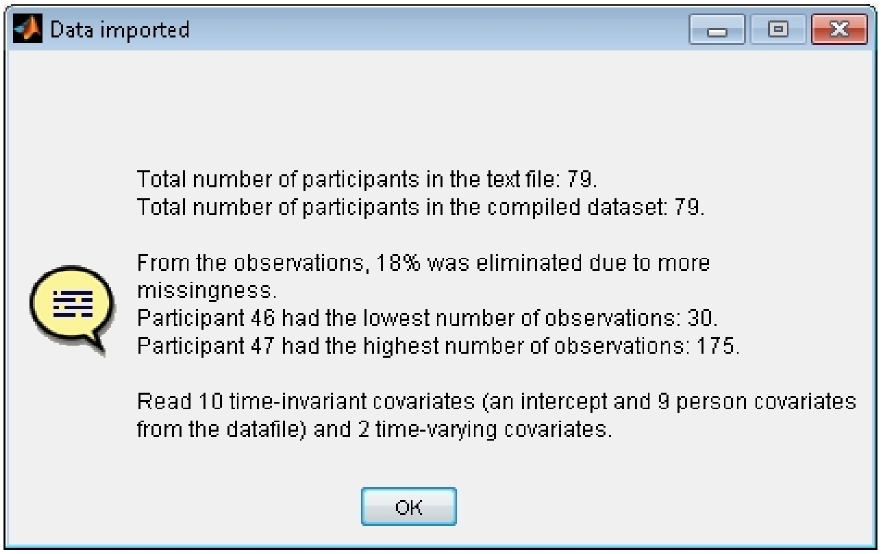

Figure 4. : Screenshot from BHOUM: the *Data reader* window after the data set has been loaded and the map data set button has been pushed.

Model specifier

Figure 5 shows the *Model specifier* window in which the parameter estimation has already been started. This window contains four panels: one to select the model type, one to set the MCMC specifications, one to save or load such specifications and one to actually start estimating the model parameters. In the top panel the user can choose whether they want to fit the full Bayesian Hierarchical OU model (*default*) or an alternative one. The default model is what was briefly described in the previous section, namely one in which we can capture inter-individual differences in every OU parameter. This way the means of the two-dimensions (μ_{1p} and μ_{2p}), the corresponding stochastic variances (γ_{1p} and γ_{2p}) and the cross-correlation (ρ_{γ_p}), as well as the two centralizing tendency (or autocorrelation, namely β_{1p} and β_{2p}) parameters and their cross-effect (ρ_{β_p}) are allowed to be person-specific. In case the user specified time-invariant covariates in the *Data reader*, these person-specific parameters are regressed on those covariates. If the user specified time-varying covariates as well, the baseline will change as a function of these covariates (similar to what one would expect in a growth curve model). According to the current settings displayed in the *Data Reader* in Figure 3, all OU-process dynamical parameters listed above are regressed on 9 covariates, labelled from $X1$-$X9$. Moreover, the two-dimensional home base is made function of $Z1$ and $Z2$.

The user can also ask for alternative models which are simplified versions of the default HOU model. Choosing the first option eliminates the measurement error from the model, that is $\boldsymbol{\Theta}_{ps} = \boldsymbol{Y}_{ps}$. The second option does not let the $\boldsymbol{\Gamma}$ to be person-specific and sets its off-diagonals (the cross-correlation ρ_{γ_p} parameter) to 0. The third option provides the same setting for $\mathbf{B}$. The fourth option eliminates the cross-effects ($\rho_{\gamma_p} = 0$, $\rho_{\beta_p} = 0$ and $\sigma_{\mu_1\mu_2} = 0$), so the model is estimated as

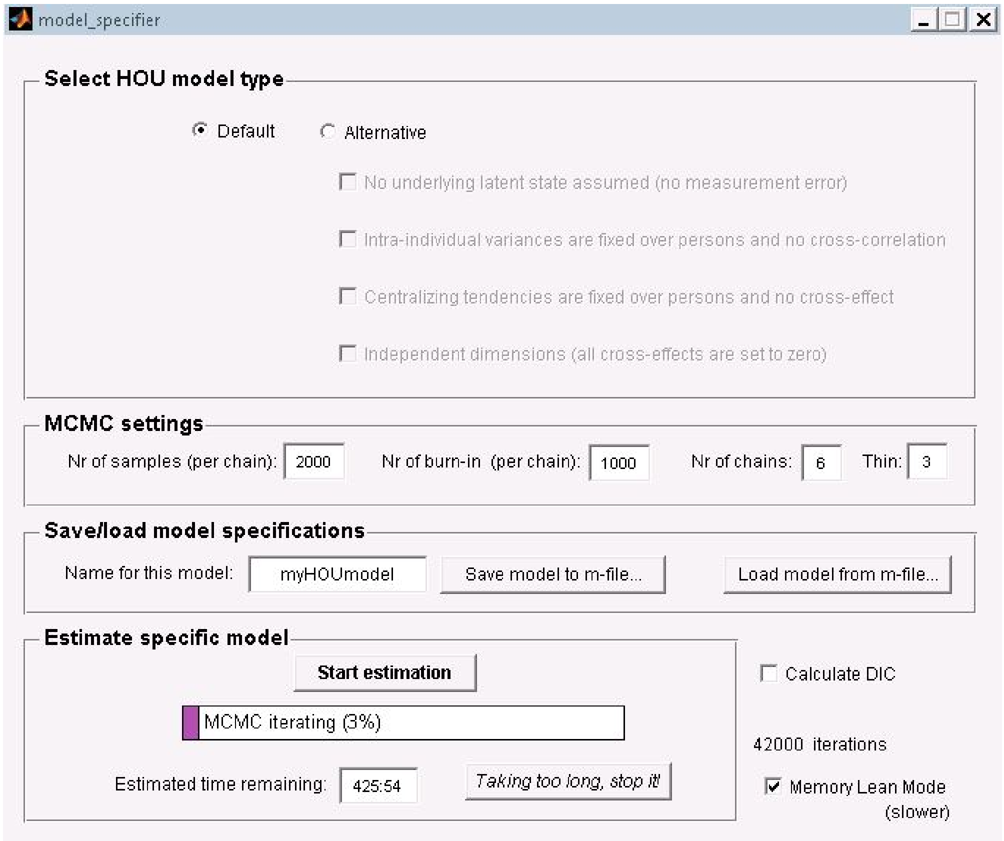

Figure 5. : A screenshot of the *Model specifier* window in which the parameter estimation has already been started.

if it had two independent dimensions. Using the latter option is also useful if there is a only one longitudinal variable measured. In this case this variable has to be loaded twice and by choosing this option the program will fit two independent one-dimensional HOU models.

The second panel allows the user to set certain properties of the MCMC sampling algorithm. Namely, the following things can be specified: (1) number of posterior samples (per chain, same for each chain) that will be used for posterior inference, (2) length of the adaptive period (the so-called burn-in) preceeding these posterior samples, (3) number of chains that are run from different starting values to explore the posterior density and (4) thinning factor. The thinning option is primarily implemented for computer memory capacity considerations. Because of high within-chain autocorrelation, some parameter estimates might require running longer chains to explore the posterior density. In thinning these long chains, we store only the x^{th} value, where x equals the input of the $Thin$ field.

As a general guideline, we recommend setting six chains each consisting of a few thousand iterations, thinned by factor 3, following an adaptation period of a 2000 iterations (never thinned).

On the bottom right corner we can see the total number of iterations the program will execute. Note: with these settings, and given a data set that has several thousand entries, model estimation will take several hours on an average desktop. Also in the bottom right corner *Memory Lean Mode* (MLM) is set as default. Memory Lean Mode lowers the demands on random access memory (RAM), which is recommended on systems with lower memory capacity. Systems with large memory capacity may benefit from turning MLM off. Above this button there is the option to ask for DIC precalculation during the iterations, for use in later model comparison. However, selecting this option will add increase the total computation time by about 10 percent.

In the next panel we can save the model specifications (including the HOU model type and the MCMC settings) and/or load a previously saved specification. The default name is "my-HOUmodel"; changing this default will also change the name of the result structure eturned to MATLAB (if applicable) after estimating the model. We briefly mention here that the possibilities in specifying an OU model go beyond the specifications available in the GUI. These extra specifications concern mainly testing purposes, but users with MATLAB experience might find it instructive to inspect the HOUmore.m function and inline documentation to find out about different alternatives they can define.

Finally, in the bottom panel, the user can proceed to parameter estimation. The approximate remaining time for the iterations is shown by a time bar in percentage and by a field in terms of minutes left. In case the user considers the iteration time too long, the estimation can be stopped. Then this window should be restarted with new numbers for the iterations and/or chains.

When the iterations are finished, two new windows will pop up: the *Result browser* (see description in the next paragraph) and a non-interactive table which gives a summary of the posterior statistics of the most important parameters. This window shows the posterior means, standard deviations and percentiles of these parameters. Moreover, it provides information about convergence by displaying $\hat{R}$ statistics (see next paragraphs), effective number of samples (the number of independent samples, computed by using the total number of posterior samples and a measure of their mutual dependence where more dependent samples count as fewer, while entirely independent samples count fully) and sample sizes.

Result browser

The *Result browser*, as shown in Figure 6, contains two panels: one dealing with posterior inference and one with Bayesian model checking called Posterior predictive checks. Upon appearance of the *Result browser*, the posterior predictive check results are not displayed but they can be calculated upon request.

In the upper panel of this window the conditional posterior distributions of all model parameters provides multifaceted assessment. The program automatically checks for convergence. If all parameters passed it, the top left button displays this message. If there are non-converged parameters, this button gives a warning. Pushing this button displays the non-converged parameters. Non-convergence here is measured by the $\hat{R}$ statistic exceeding 1.1. Chains with $\hat{R}$ statistic under this value are considered converged by the program. However, visual checks based on sample chain plots are also advisable. This option is provided in the same panel. By using the list box containing all parameter names, a subgroup of parameters can easily be selected and by clicking the "Plot chain(s)" option the sampled values will be plotted. In case convergence is not reached, the user

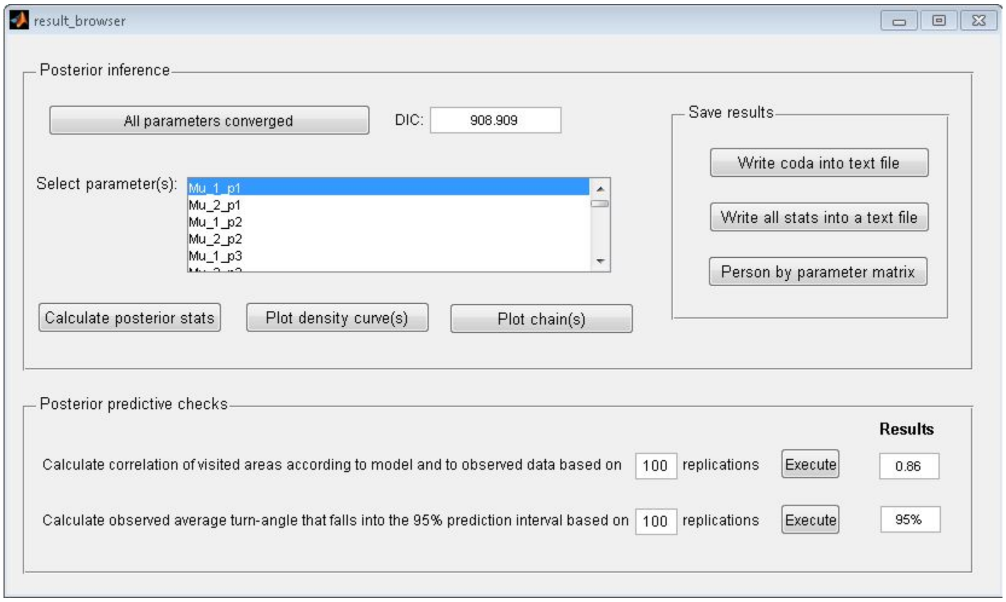

Figure 6. : The Result browser

can go back to the *Model specifier* window and add more iterations. These will be added to the already sampled ones, unless the user closes the *Model specifier* window and restarts the process.

The "Plot density curve(s)" button provides the user with smoothed posterior density plot(s). Posterior statistics may also be calculated for selected parameters. Moreover, in the subpanel on the right the user may choose to write all iterated values into text files and/or write all posterior statistics into a text file (the latter provides the $\hat{R}$ statistic for all the parameters). Finally this upper panel also displays the DIC value if the user asked for it in the *Model specifier* window. While DIC provides the user with a relative goodness-of-fit measure, the lower subsection offers posterior predictive checks to give an idea of the absolute goodness of fit of the model.

All posterior predictive checks in the lower panel are based on generating new data sets from the full conditional distribution of the parameters, and comparing certain properties of the observed and the generated data sets. While trajectories in the two-dimensional space corresponding to the two longitudinal variables are easily drawn from the observed as well as replicated data sets, checking whether they actually correspond well is a non-trivial task in stochastic models where the inherent randomness is part of the model.

Here we present three strategies to evaluate absolute goodness-of-fit. For each strategy the user may choose the number of replicated data sets from which the measure should be based (these fields are pre-filled with the value 100, to increase accuracy the user might choose to set it to a higher value, for example 1000 generated data sets).

The first posterior check assesses the similarity between the observed and replicated trajectories: it computes the overlap between the observed and simulated trajectories by calculating the correlation between the frequencies that observed data fall in a certain area in a two-dimensional space, and the average frequency that they fall in that same area across replicated data sets. The two-dimensional space is subdivided into 9 equal areas, and the visit frequencies in these areas for every person are calculated and correlated with the corresponding mean visit frequencies (across the replicated data sets). The resulting measure is a correlation coefficient for every participant and in the GUI field the mean over persons is displayed. As we can see our data set shows an average of 0.85 correlation, which indicates good fit.

Another measure that can give us information about whether the observed and replicated trajectories look alike is their turning angles. By turning angle, we mean the clockwise angle between two line segments which connect three subsequent points in time in the two-dimensional space. We average over all turning angles person-wise, resulting in a person-specific average turning angle value. We calculate this measure for replicated data sets and based on them a 95 % prediction interval is established for every person. The GUI field returns how many percent of the observed average turning angle would fall in this interval. With respect to this measure it turns out that the model fits the data set rather well as it shows 95 % of the generated person-specific average turn-angles are within the 95 % prediction interval.

Discussion

The HOU model is a potentially important psychometric modeling tool that can be applied to various phenomena that are assumed to change dynamically over time. We showed through an example application how many various aspects of the change mechanism can be explored by this model, some of which have not yet been incorporated in any other modeling framework (i.e. regulation).

Additionally, inter-individual variability may be explained in the HOU model, when meaningful predictors are measured. By enabling us to regress model parameters on covariates in one step, the HOU model increases the accuracy of the estimated regression coefficients. Such desirable properties, together with a user-friendly parameter estimation tool, will hopefully make the model widely applied and popular among substantive researchers.

Acknowledgements

We are grateful to Marlies Houben and Peter Kuppens for their beta-testing efforts of BHOUM and suggestions, and to Chelsea Muth for her most helpful comments on the manuscript.

References

Dunn, J. E., & Gipson, P. S. (1977). Analysis of radio telemetry data in studies of home range. *Biometrics*, *33*, 85–101. doi: 10.2307/2529305

Gelman, A., Carlin, J., Stern, H., & Rubin, D. (2004). *Bayesian data analysis*. New York: Chapman & Hall.

Oravecz, Z., & Tuerlinckx, F. (2008). Bayesian estimation of an Ornstein-Uhlenbeck process based hierarchical model. In *Proceedings of the seventh international conference on social science methodology* (pp. [CD–ROM]). Napels: University of Naples.

Oravecz, Z., & Tuerlinckx, F. (2011). The linear mixed model and the hierarchical Ornstein-Uhlenbeck model: Some equivalences and differences. *British Journal of Mathematical and Statistical Psychology*, *64*, 134–160. doi: 10.1348/000711010X498621

Oravecz, Z., Tuerlinckx, F., & Vandekerckhove, J. (2009). A hierarchical Ornstein-Uhlenbeck model for continuous repeated measurement data. *Psychometrika*, *74*, 395–418.

Oravecz, Z., Tuerlinckx, F., & Vandekerckhove, J. (2011). A hierarchical latent stochastic differential equation model for affective dynamics. *Psychological Methods*, *16*, 468–490. doi: 10.1037/a0024375

Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of Brownian motion. *Physical Review*, *36*, 823–841. doi: 10.1103/PhysRev.36.823