

# 1 babette: BEAUti 2, BEAST2 and Tracer for R

2 Richèl J.C. Bilderbeek<sup>1</sup> and Rampal S. Etienne<sup>1</sup>

3 <sup>1</sup>Groningen Institute for Evolutionary Life Sciences, University of  
4 Groningen, Groningen, The Netherlands

5 May 2, 2018

## 6 Summary

7 **1.** In the field of phylogenetics, BEAST2 is one of the most widely  
8 used software tools. It comes with the graphical user interfaces BEAUti  
9 2, DensiTree and Tracer, to create BEAST2 configuration files and to in-  
10 terpret BEAST2's output files. However, when many different alignments  
11 or model setups are required, a workflow of graphical user interfaces is  
12 cumbersome.

13 **2.** Here, we present a free, libre and open-source package, **babette**:  
14 'BEAUti 2, BEAST2 and Tracer for R', for the R programming language.  
15 **babette** creates BEAST2 input files, runs BEAST2 and parses its results,  
16 all from an R function call.

17 **3.** We describe **babette**'s usage and the novel functionality it provides  
18 compared to the original tools and we give some examples.

19 **4.** As **babette** is designed to be of high quality and extendable, we  
20 conclude by describing the further development of the package.

21  
22 **Keywords:** computational biology, evolution, phylogenetics, BEAST2, R

# 1 Introduction

Phylogenies are commonly used to explore evolutionary hypotheses. Not only can phylogenies show us how species (or other evolutionary units) are related to each other, but we also estimate relevant parameters such as extinction and speciation rates. There are many phylogenetics tools available to obtain an estimate of the phylogeny of a given set of species. BEAST2 (Bouckaert *et al.* 2014) is one of the most widely used ones. It uses a Bayesian statistical framework to estimate the joint posterior distribution of estimated phylogenies and model parameters, from one or more DNA, RNA or amino acid alignments (see figure 1 for an overview of the workflow).

It has a graphical and a command-line interface, that both need a configuration file containing alignments and model parameters. BEAST2 is bundled with BEAUti 2 (Drummond *et al.* 2012) ('BEAUti' from now on), a desktop application to create a BEAST2 configuration file. BEAUti has a user-friendly graphical user interface, with helpful default settings. As such, BEAUti is an attractive alternative to manual and error-prone editing of BEAST2 configuration files.

However, BEAUti cannot be called from a command-line script. This implies that when the user wants to explore the consequences of various settings, this must be done manually. This is the managable workflow when using a few alignments and doing a superficial analysis of sensitivity of the reconstructed tree to model settings. For exploring many trees (for instance from simulations) and for more thorough sensitivity analysis, one would like to loop through multiple (simulated) alignments, nucleotide substitution models, clock models and tree priors. One such tool to replace BEAUti is **BEASTmasterR** (Matzke 2015), which focuses on morphological traits and tip-dating, but also supports DNA data. **BEASTmasterR**, however, requires hundreds of lines of R code to setup the

50 BEAST2 model configuration and a Microsoft Excel file to specify alignment  
51 files.

52 BEAST2 is also associated with Tracer (Rambaut & Drummond 2007) and  
53 DensiTree (Bouckaert & Heled 2014). Both are desktop applications to an-  
54alyze the output of BEAST2, each with a user-friendly graphical user inter-  
55face. Tracer’s purpose is to analyze the parameter estimates generated from a  
56BEAST2 run. It shows, among others, the effective sample size (ESS) and time  
57series (‘the trace’, hence the name) of each variable in the MCMC run. Both  
58ESS and trace are needed to assess the strength of the inference. DensiTree vi-  
59sualizes the phylogenies of a BEAST2 posterior, with many options to improve  
60the simultaneous display of many phylogenies.

61 However, for exploring the output of many BEAST2 runs, one would like a  
62script to collect all parameters’ ESSes, parameter traces and posterior phyloge-  
63nies. There is no single package that offers a complete solution, but examples  
64of R packages that offer a partial solution are rBEAST (Ratmann 2015) and  
65RBeast (Faria & Suchard 2015). RBeast provides some plotting options and  
66parsing of BEAST2 output files, but the plotting functions are too specific for  
67general use, whilst the functions for parsing call those of **babette**. rBEAST was  
68developed to test a particular biological hypothesis (Ratmann *et al.* 2016), and  
69hence was not designed for general use.

70 Here, we present **babette**: BEAUti 2, BEAST2 and Tracer for R, which  
71creates BEAST2 (v.2.4.7) configuration files, runs BEAST2, and analyzes its  
72results, all from an R function call. This will save time, tedious mouse clicking  
73and reduces the chances of errors in such repetitive actions. The interface of  
74**babette** mimics the tools it is based on. This familiarity helps both beginner  
75and experienced BEAST2 users to make the step from those tools to **babette**.  
76**babette** enables the creation of a single-script pipeline from sequence alignments

77 to posterior analysis in R.

## 78 2 Description

79 **babette** is written in the R programming language (R Core Team 2013) and  
80 enables the full BEAST2 workflow from an R function call, in a similar way  
81 to what BEAUti, DensiTree and Tracer do. **babette**'s main function is **run**,  
82 which configures BEAST2, runs it and parses its output. **run** needs at least the  
83 name of a FASTA file containing a DNA alignment. The default settings for  
84 the other arguments of **run** are identical to BEAUti's and BEAST2's default  
85 settings. Per alignment, a site model, clock model and tree prior can be chosen.  
86 Multiple alignments can be used, each with its own (unlinked) site model, clock  
87 model and tree prior.

88 **babette** currently has 108 exported functions to set up a BEAST2 config-  
89 uration file. **babette** can currently handle a majority of BEAUti use cases.  
90 Because of BEAUti's high number of plugins, **babette** uses a software architec-  
91 ture that is designed to be extended. Furthermore, **babette** has 13 exported  
92 functions to run and help run BEAST2. One function is used to run BEAST2,  
93 another one installs BEAST2 to a default location. Finally, **babette** has 21  
94 exported function to parse the BEAST2 output files and analyze the created  
95 posterior. **babette** gives the same ESSes and summary statistics as Tracer.  
96 The data is formatted such that it can easily be visualized using **ggplot2** (for  
97 a trace, similar to Tracer) or **phangorn** (Schliep 2011) (for the phylogenies in a  
98 posterior, similar to DensiTree).

99 Currently, **babette** does not replace all functionality in BEAUti, BEAST2  
100 and their many plug-ins. This is reasonable, as these themselves also change  
101 in time. **babette** provides the majority of default tree priors and works only  
102 on DNA data. **babette** supports the most important command-line arguments

103 of BEAST2, provides the core Tracer analysis options, and has the most basic  
104 subset of plotting options of DensiTree. Up until now, the **babette** features  
105 implemented are those requested by users. Further extension of **babette** will  
106 be based on future user requests.

### 107 3 Usage

108 **babette** can be installed easily with the **devtools** package (Wickham & Chang  
109 2016):

```
110 library(devtools)
111 install_github("richelbilderbeek/babette")
```

112 To start using **babette**, load its functions in the global namespace first:

```
113 library(babette)
```

114 For most convenience, let **babette** install BEAST2 to a default location:

```
115 install_beast2()
```

116 BEAUti, and likewise **babette**, needs at least a FASTA filename to produce a  
117 BEAST2 configuration file. In BEAUti, this is achieved by loading a FASTA file,  
118 then saving an output file using a common save file dialog. After this, BEAST2  
119 needs to be applied to the created configuration file. It creates multiple files  
120 storing the posterior. These output files must be parsed by either Tracer or  
121 DensiTree. In **babette**, all this is achieved by:

```
122 out <- run(fasta_filenames = "anthus_aco.fas")
```

123 This code will create a (temporary) BEAST2 configuration file, from the FASTA  
124 file with name **anthus\_aco.fas** (which is supplied with the package, from  
125 (Van Els & Norambuena 2018)), using the same default settings as BEAUti,  
126 which are, among others, a Jukes-Cantor site model, a strict clock, and a Yule

127 birth tree prior. `babette` will then execute BEAST2 using that file, and parses  
 128 the output. The returned data structure, named `out`, is a list of parameter  
 129 estimates (called `estimates`), posterior phylogenies (called `anthus_aco_trees`,  
 130 named after the alignment's name) and MCMC operator performance (`operators`).  
 131 An example of using a different site model, clock model and tree prior is:

```
132 out <- run(  
133   fasta_filenames = "anthus_aco.fas",  
134   site_models = create_hky_site_model(),  
135   clock_models = create_rln_clock_model(),  
136   tree_priors = create_bd_tree_prior()  
137 )
```

138 This code uses an HKY site model, a relaxed log-normal clock model and a birth-  
 139 death tree prior, each with their default settings in BEAUti. Table 1 shows an  
 140 overview of all functions to create site models, clock models and tree priors. Note  
 141 that the arguments' names `site_models`, `clock_models` and `tree_priors` are  
 142 plural, as each of these can be (a list of) one or more elements. Each of these  
 143 arguments must have the same number of elements, so that each alignment has  
 144 its own site model, clock model and tree prior. An example of two alignments,  
 145 each with its own site model, is:

```
146 out <- run(  
147   fasta_filenames = c(  
148     "anthus_aco.fas",  
149     "anthus_nd2.fas"  
150   ),  
151   site_models = list(  
152     create_tn93_site_model(),  
153     create_gtr_site_model()  
154   )
```

155 )

156 **babette** also uses the same default prior distributions as BEAUti for each of  
157 the site models, clock models and tree priors. For example, by default, a Yule  
158 tree prior assumes that the birth rate follows a uniform distribution, from minus  
159 infinity to plus infinity. One may prefer a different distribution instead. Here  
160 is an example how to specify an exponential distribution for the birth rate in a  
161 Yule tree prior in **babette**:

```
162 out <- run(  
163   fasta_filenames = "anthus_aco.fas",  
164   tree_priors = create_yule_tree_prior(  
165     birth_rate_distr = create_exp_distr()  
166   )  
167 )
```

168 In this same example, one may specify the initial shape parameters of the expo-  
169 nential distribution. In BEAST2's implementation, an exponential distribution  
170 has one shape parameter: its mean, which can be set to any value with BEAUti.  
171 Within **babette**, to set the mean value of the exponential distribution to a fixed  
172 (non-estimated) value, do:

```
173 out <- run(  
174   fasta_filenames = "anthus_aco.fas",  
175   tree_priors = create_yule_tree_prior(  
176     birth_rate_distr = create_exp_distr(  
177       mean = create_mean_param(  
178         value = 1.0,  
179         estimate = FALSE  
180       )  
181     )  
182   )
```

183 )

184 **NOTE: suggest to remove this paragraph until node dating** Our initial  
185 motivation to create **babette** was that we wanted to fix the crown age (the  
186 time of first divergence) of a phylogeny. BEAUti assumes that a phylogeny has  
187 a crown age that needs to be jointly estimated with the phylogeny and other  
188 parameters. It does not allow for fixing the crown age. Without **babette**,  
189 one needs to manually edit the BEAST2 configuration file, to remove all tree  
190 operators that change the height of the root node, which is tedious and prone  
191 to errors. Fixing the crown ages is useful when it is unnecessary to accurately  
192 model the uncertainty of absolute divergence times. This is how to specify a  
193 fixed crown age with **babette**:

```
194 out <- run(  
195   fasta_filenames = "anthus_aco.fas",  
196   posterior_crown_age = 15  
197 )
```

198 **babette** also supports node dating. Like BEAUti, one can specify Most Recent  
199 Common Ancestor ('MRCA') priors. An MRCA prior allows to specify taxa  
200 having a common ancestor, including a distribution for the date of that ancestor.  
201 An application of such a prior is to specify a (close to) fixed crown age, by  
202 including all taxa and assuming a tight normal distribution around the desired  
203 crown age. Within **babette**, this is achieved as such:

```
204 posterior <- run(  
205   fasta_filenames = "anthus_aco.fas",  
206   mrca_priors = create_mrca_prior(  
207     taxa_names = get_taxa_names("anthus_aco.fas"),  
208     alignment_id = get_alignment_id("anthus_aco.fas"),  
209     mrca_distr = create_normal_distr(  

```



```

210     mean = create_mean_param(value = 15.0, estimate =
211         FALSE),
212     sigma = create_sigma_param(value = 0.01, estimate =
213         FALSE)
214 )
215 )
216 )

```

217 **babette** allows for the same core functionality as Tracer, in which it works on  
218 the values of the parameter estimates sampled in the BEAST2 run. This is  
219 called the "trace" (hence the name). The start of the trace is usually discarded,  
220 as an MCMC algorithm (such as used by BEAST2) first has to converge to  
221 its equilibrium. The start of the trace, called the "burn-in", will be removed,  
222 because its parameter estimates are not representative. By default, Tracer dis-  
223 cards the first 10% of all the parameter estimates. To remove a 20% burn-in  
224 from all parameter estimates in **babette**, the following code can be used:

```

225 traces <- remove_burn_ins(
226     traces = out$estimates,
227     burn_in_fraction = 0.2
228 )

```

229 Tracer shows the ESSes of each posterior's variables. These ESSes are important  
230 to determine the strength of the inference. As a rule of thumb, an ESS of 200 is  
231 acceptable for any parameter estimate. To calculate the effective sample sizes  
232 (of all estimated variables) in **babette**:

```

233 esses <- calc_esses(
234     traces = traces,
235     sample_interval = 1000
236 )

```

237 Tracer displays multiple summary statistics for each estimated variable: the  
238 mean and its standard error, standard deviation, variance, median, mode, geo-  
239 metric mean, 95% highest posterior density interval, auto-correlation time and  
240 effective sample size. It displays these statistics per variable. In **babette**, these  
241 summary statistics are collected for all estimated parameters at once:

```
242 sum_stats <- calc_summary_stats(  
243   traces = traces,  
244   sample_interval = 1000  
245 )
```

246 **babette** allows for the same functionality as DensiTree. DensiTree displays the  
247 phylogenies in a posterior at the same time scale, drawn one over one another,  
248 allowing to see the uncertainty in topology and branch lengths. Within the  
249 object **out**, the posterior phylogenies are stored as **anthus\_aco\_trees**, and can  
250 be plotted as such:

```
251 plot_densitree(phylos = out$anthus_aco_trees)
```

252 Instead of running the full pipeline, **babette** also allows to only create a BEAST2  
253 configuration file. To create a BEAST2 configuration file, with all settings to  
254 default, use:

```
255 create_beast2_input_file(  
256   input_filenames = babette::get_babette_path("anthus_aco.  
257     fas"),  
258   output_filename = "beast2.xml"  
259 )
```

260 This file can then be loaded and edited by BEAUti, or run by BEAST2. **babette**  
261 also allows to only run BEAST2 from a BEAST2 configuration file:

```
262 run_beast2(  
263   input_filename = "beast2.xml",
```

```

264     output_log_filename = "run.log",
265     output_trees_filenames = "posterior.trees",
266     output_state_filename = "final.xml.state"
267 )

```

268 In this example, we specify the names of the desired BEAST2 output files. These  
 269 output files can then be inspected with other tools, or to be used to continue a  
 270 BEAST2 run. Also `run` supports to specify these filenames, and these files are  
 271 put in a temporary folder by default.

## 272 4 **babette** resources

273 **babette** is free, libre and open source software available at <http://github.com/richelbilderbeek/babette> and is licensed under the GNU General Public License v3.0. **babette** uses the Travis CI (<https://travis-ci.org>) continuous integration service, which is known to significantly increase the number  
 276 of bugs exposed (Vasilescu *et al.* 2015) and increases the speed at which new  
 277 features are added (Vasilescu *et al.* 2015). **babette** has a 100% code coverage,  
 278 which correlates with code quality (Horgan *et al.* 1994; Del Frate *et al.*  
 279 1995). **babette** follows Hadley Wickham’s style guide (Wickham 2015), which  
 280 improves software quality (Fang 2001). **babette** depends on multiple packages,  
 281 which are **ape** (Paradis *et al.* 2004), **beautier** (Bilderbeek 2018b), **beastier**  
 282 (Bilderbeek 2018a), **devtools** (Wickham & Chang 2016), **geiger** (Harmon *et al.*  
 283 2008), **ggplot2** (Wickham 2009), **knitr** (Xie 2017), **phangorn** (Schliep 2011),  
 284 **rmarkdown** (Allaire *et al.* 2017), **seqinr** (Charif & Lobry 2007), **stringr** (Wickham  
 285 2017), **testit** (Xie 2014) and **tracerer** (Bilderbeek 2018c). We tested  
 286 **babette** to give a clean error message for incorrect input, by calling **babette**  
 287 one million times with random or random sensible inputs, using the Peregrine  
 288 high performance computer cluster. The scripts to do so are supplied with  
 289

290 **babette**.

291 **babette**'s development takes place on GitHub, <https://github.com/richelbilderbeek/>  
292 **babette**, which accommodates collaboration (Perez-Riverol *et al.* 2016) and im-  
293 proves transparency (Gorgolewski & Poldrack 2016). **babette**'s GitHub facili-  
294 tates feature requests and has guidelines how to do so.

295 **babette**'s documentation is extensive. All functions are documented in the  
296 package's internal documentation. For quick use, each exported function shows  
297 a minimal example. For easy exploration, each exported function's documen-  
298 tation links to related functions. Additionally, **babette** has a vignette that  
299 demonstrates extensively how to use it. There is documentation on the GitHub  
300 to get started, with a dozen examples of BEAUti screenshots with equivalent  
301 **babette** code. Finally, **babette** has tutorial videos that can be downloaded or  
302 viewed on YouTube, <https://goo.gl/weKaaU>.

## 303 5 Citation of babette

304 Scientists using **babette** in a published paper can cite this article, and/or cite  
305 the **babette** package directly. To obtain this citation from within an R script,  
306 use:

```
307 > citation("babette")
```

## 308 6 Acknowledgements

309 Thanks to Yacine Ben Chehida and Paul van Els for supplying their BEAST2  
310 use cases. Thanks again to Paul van Els for sharing his FASTA files for use  
311 by this package. Thanks to Leonel Herrera-Alsina, Raphael Scherrer and Gio-  
312 vanni Laudanno for their comments on this package and article. Thanks to  
313 Huw Ogilvie and one anonymous reviewer for reviewing this article. Thanks

314 to rOpenSci, and especially Noam Ross, Guangchuang Yu, David Winter and  
 315 Joëlle Barido-Sottani for reviewing the package’s source code. We would like  
 316 to thank the Center for Information Technology of the University of Groningen  
 317 for their support and for providing access to the Peregrine high performance  
 318 computing cluster.

## 319 7 Authors’ contributions

320 RJCB and RSE conceived the idea for the package. RJCB created and tested  
 321 the package, and wrote the first draft of the manuscript. RSE contributed  
 322 substantially to revisions.

## 323 References

- 324 Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wick-  
 325 ham, H., Cheng, J. & Chang, W. (2017) *rmarkdown: Dynamic Documents*  
 326 *for R*. R package version 1.8.
- 327 Bilderbeek, R.J. (2018a) beastier: BEAST2 from R. [https://github.com/](https://github.com/ricelbilderbeek/beastier)  
 328 [ricelbilderbeek/beastier](https://github.com/ricelbilderbeek/beastier) [Accessed: 2018-03-16].
- 329 Bilderbeek, R.J. (2018b) beautier: BEAUti 2 from R. [https://github.com/](https://github.com/ricelbilderbeek/beautier)  
 330 [ricelbilderbeek/beautier](https://github.com/ricelbilderbeek/beautier) [Accessed: 2018-03-16].
- 331 Bilderbeek, R.J. (2018c) tracerer: Tracer from R. [https://github.com/](https://github.com/ricelbilderbeek/tracerer)  
 332 [ricelbilderbeek/tracerer](https://github.com/ricelbilderbeek/tracerer) [Accessed: 2018-03-16].
- 333 Bouckaert, R. & Heled, J. (2014) Densitree 2: Seeing trees through the forest.  
 334 *bioRxiv*, p. 012401.
- 335 Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C.H., Xie, D., Suchard,

336 M.A., Rambaut, A. & Drummond, A.J. (2014) Beast 2: a software platform  
 337 for bayesian evolutionary analysis. *PLoS Comput Biol*, **10**, e1003537.

338 Charif, D. & Lobry, J. (2007) SeqinR 1.0-2: a contributed package to the R  
 339 project for statistical computing devoted to biological sequences retrieval and  
 340 analysis. U. Bastolla, M. Porto, H. Roman & M. Vendruscolo, eds., *Struc-*  
 341 *tural approaches to sequence evolution: Molecules, networks, populations*, Bi-  
 342 ological and Medical Physics, Biomedical Engineering, pp. 207–232. Springer  
 343 Verlag, New York. ISBN : 978-3-540-35305-8.

344 Del Frate, F., Garg, P., Mathur, A.P. & Pasquini, A. (1995) On the correlation  
 345 between code coverage and software reliability. *Software Reliability Engi-*  
 346 *neering, 1995. Proceedings., Sixth International Symposium on*, pp. 124–132.  
 347 IEEE.

348 Drummond, A.J., Suchard, M.A., Xie, D. & Rambaut, A. (2012) Bayesian phy-  
 349 logenetics with beauti and the beast 1.7. *Molecular biology and evolution*, **29**,  
 350 1969–1973.

351 Fang, X. (2001) Using a coding standard to improve program quality. *Quality*  
 352 *Software, 2001. Proceedings. Second Asia-Pacific Conference on*, pp. 73–78.  
 353 IEEE.

354 Faria, N. & Suchard, M.A. (2015) RBeast. [https://github.com/beast-dev/](https://github.com/beast-dev/RBeast)  
 355 **RBeast** [Accessed: 2018-03-02].

356 Gorgolewski, K.J. & Poldrack, R. (2016) A practical guide for improving trans-  
 357 parency and reproducibility in neuroimaging research. *bioRxiv*, p. 039354.

358 Harmon, L., Weir, J., Brock, C., Glor, R. & Challenger, W. (2008) Geiger:  
 359 investigating evolutionary radiations. *Bioinformatics*, **24**, 129–131.

360 Horgan, J.R., London, S. & Lyu, M.R. (1994) Achieving software quality with  
361 testing coverage measures. *Computer*, **27**, 60–69.

362 Matzke, N.J. (2015) BEASTmaster: R tools for automated conversion of  
363 NEXUS data to BEAST2 XML format, for fossil tip-dating and other uses.  
364 <https://github.com/nmatzke/BEASTmaster> [Accessed: 2018-02-28].

365 Paradis, E., Claude, J. & Strimmer, K. (2004) APE: analyses of phylogenetics  
366 and evolution in R language. *Bioinformatics*, **20**, 289–290.

367 Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Lepre-  
368 vost, F., Fufezan, C., Ternent, T., Eglen, S.J., Katz, D.S. *et al.* (2016) Ten  
369 simple rules for taking advantage of git and github. *bioRxiv*, p. 048744.

370 R Core Team (2013) *R: A Language and Environment for Statistical Computing*.  
371 R Foundation for Statistical Computing, Vienna, Austria.

372 Rambaut, A. & Drummond, A.J. (2007) *Tracer v1.4*. Available from  
373 <http://beast.bio.ed.ac.uk/Tracer>.

374 Ratmann, O. (2015) rBEAST. <https://github.com/olli0601/rBEAST> [Ac-  
375 cessed: 2018-03-02].

376 Ratmann, O., Van Sighem, A., Bezemer, D., Gavryushkina, A., Jurriaans, S.,  
377 Wensing, A., De Wolf, F., Reiss, P., Fraser, C. *et al.* (2016) Sources of hiv  
378 infection among men having sex with men and implications for prevention.  
379 *Science translational medicine*, **8**, 320ra2–320ra2.

380 Schliep, K. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**,  
381 592–593.

382 Van Els, P. & Norambuena, H.V. (2018) A revision of species limits in neotrop-  
383 ical pipits anthus based on multilocus genetic and vocal data. *Ibis*.

- 384 Vasilescu, B., Yu, Y., Wang, H., Devanbu, P. & Filkov, V. (2015) Quality and  
 385 productivity outcomes relating to continuous integration in github. *Proceed-*  
 386 *ings of the 2015 10th Joint Meeting on Foundations of Software Engineering*,  
 387 pp. 805–816. ACM.
- 388 Wickham, H. (2009) *ggplot2: elegant graphics for data analysis*. Springer New  
 389 York.
- 390 Wickham, H. (2015) *R packages: organize, test, document, and share your code*.  
 391 O’Reilly Media, Inc.
- 392 Wickham, H. (2017) *stringr: Simple, Consistent Wrappers for Common String*  
 393 *Operations*. R package version 1.2.0.
- 394 Wickham, H. & Chang, W. (2016) *devtools: Tools to Make Developing R Pack-*  
 395 *ages Easier*. R package version 1.12.0.9000.
- 396 Xie, Y. (2014) *testit: A Simple Package for Testing R Packages*. R package  
 397 version 0.4, <http://CRAN.R-project.org/package=testit>.
- 398 Xie, Y. (2017) *knitr: A General-Purpose Package for Dynamic Report Genera-*  
 399 *tion in R*. R package version 1.17.



| Name                      | Description   |
|---------------------------|---|
| run                       | Run BEAST2  |
| create_gtr_site_model     | Create a GTR site model                               |
| create_hky_site_model     | Create an HKY site model                              |
| create_jc69_site_model    | Create a Jukes-Cantor site model                      |
| create_tn93_site_model    | Create a TN93 site model                              |
| create_rln_clock_model    | Create a relaxed log-normal clock model               |
| create_strict_clock_model | Create a strict clock model                           |
| create_bd_tree_prior      | Create a birth-death tree prior                       |
| create_cbs_tree_prior     | Create a coalescent Bayesian skyline tree prior       |
| create_ccp_tree_prior     | Create a coalescent constant-population tree prior    |
| create_cep_tree_prior     | Create a coalescent exponential-population tree prior |
| create_yule_tree_prior    | Create a Yule tree prior                              |
| create_beta_distr         | Create a beta distribution                            |
| create_exp_distr          | Create an exponential distribution                    |
| create_gamma_distr        | Create a gamma distribution                           |
| create_inv_gamma_distr    | Create an inverse gamma distribution                  |
| create_laplace_distr      | Create a Laplace distribution                         |
| create_log_normal_distr   | Create a log-normal distribution                      |
| create_normal_distr       | Create a normal distribution                          |
| create_one_div_x_distr    | Create a 1/X distribution                             |
| create_poisson_distr      | Create a Poisson distribution                         |
| create_uniform_distr      | Create a uniform distribution                         |

Table 1: babette's main functions

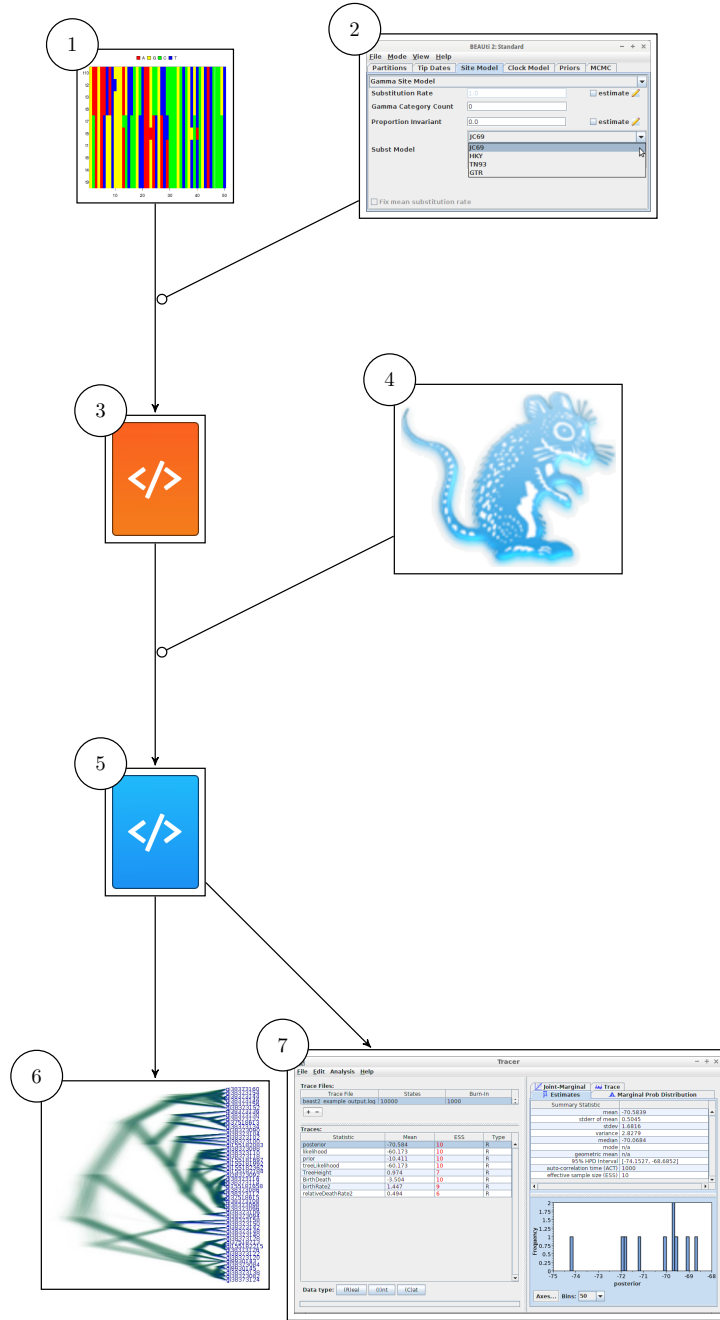


Figure 1: Workflow using GUI tools. From an alignment (1) and BEAUti (2), a BEAST2 configuration file (3) is created. BEAST2 (4) uses that file to infer a posterior, storing it in multiple files (5). These results are visualized using DensiTree (6) and Tracer (7). babette allows for the same workflow, all from an R function call.