

# 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 14, 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 Summary

1. In het onderzoeksveld van de fylogenetica is BEAST2 een van de meest gebruikte hulpprogramma's. Het is gebundeld met de grafische gebruikers interface BEAUti 2, DensiTree en Tracer, om BEAST2 configuratiebestanden te maken en om BEAST2 output bestanden te interpreteren. Echter, als veel verschillende aligneringen of modelopzetten nodig zijn, is een werkvolgorde van meerdere grafische gebruikssinterfases onhandig.
2. Hier presenteren wij een gratis, vrij en open-source package, **babette**: 'BEAUti 2, BEAST2 en Tracer voor R', voor de programmeertaal R. **babette** schrijft BEAST2 configuratiebestanden, start BEAST2 and verwerkt de resultaten, alles met een enkele R functie-aanroep.
3. Wij beschrijven hoe **babette** te gebruiken is en de nieuwe mogelijkheden die het biedt vergeleken met de originele programma's, begeleidt met enkele voorbeelden.
4. Omdat **babette** ontworpen is om van hoge kwaliteit en uitbreidbaar te zijn, sluiten wij af met het beschrijven van de verdere ontwikkeling van dit package.

**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 can also estimate relevant parameters such as extinction and speciation rates from them. 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 phylogenies

50 and model parameters, from one or more DNA, RNA or amino acid alignments  
51 (see figure 1 for an overview of the workflow).

52 BEAST2 has a graphical and a command-line interface, that both need a  
53 configuration file containing alignments and model parameters. BEAST2 is bun-  
54 dled with BEAUti 2 (Drummond *et al.* 2012) ('BEAUti' from now on), a desktop  
55 application to create a BEAST2 configuration file. BEAUti has a user-friendly  
56 graphical user interface, with helpful default settings. As such, BEAUti is an at-  
57 tractive alternative to manual and error-prone editing of BEAST2 configuration  
58 files.

59 However, BEAUti cannot be called from a command-line script. This implies  
60 that when the user wants to explore the consequences of various settings, this  
61 must be done manually. This is the manageable workflow when using a few align-  
62 ments and doing a superficial analysis of sensitivity of the reconstructed tree to  
63 model settings. For exploring many trees (for instance from simulations), when  
64 doing a sliding-window analysis, or for a more thorough sensitivity analysis,  
65 one would like to loop through multiple (simulated or shortened) alignments,  
66 nucleotide substitution models, clock models and tree priors. One such tool to  
67 replace BEAUti is **BEASTmaster** (Matzke 2015), which focuses on morphologi-  
68 cal traits and tip-dating, but also supports DNA data. **BEASTmaster**, however,  
69 requires hundreds of lines of R code to setup the BEAST2 model configuration  
70 and a Microsoft Excel file to specify alignment files.

71 BEAST2 is also associated with Tracer (Rambaut & Drummond 2007) and  
72 DensiTree (Bouckaert & Heled 2014). Both are desktop applications to an-  
73 alyze the output of BEAST2, each with a user-friendly graphical user inter-  
74 face. Tracer's purpose is to analyze the parameter estimates generated from  
75 a (BEAST1 and) BEAST2 run. It shows, among others, the effective sample  
76 size (ESS) and time series ('the trace', hence the name) of each variable in the

77 MCMC run. Both ESS and trace are needed to assess the strength of the infer-  
78 ence. DensiTree visualizes the phylogenies of a BEAST2 posterior, with many  
79 options to improve the simultaneous display of many phylogenies.

80 However, for exploring the output of many BEAST2 runs, one would like a  
81 script to collect all parameters' ESSes, parameter traces and posterior phyloge-  
82 nies. There is no single package that offers a complete solution, but examples  
83 of R packages that offer a partial solution are rBEAST (Ratmann 2015) and  
84 RBeast (Faria & Suchard 2015). RBeast provides some plotting options and  
85 parsing of BEAST2 output files, but the plotting functions are too specific for  
86 general use. rBEAST was developed to test a particular biological hypothesis  
87 (Ratmann *et al.* 2016), and hence was not designed for general use.

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

## 96 2 Description

97 **babette** is written in the R programming language (R Core Team 2013) and  
98 enables the full BEAST2 workflow from a single R function call, in a similar  
99 way to what subsequent usage of BEAUti, DensiTree and Tracer would produce.  
100 **babette**'s main function is **bbt\_run**, which configures BEAST2, runs it and  
101 parses its output. **bbt\_run** needs at least the name of a FASTA file containing  
102 a DNA alignment. The default settings for the other arguments of **bbt\_run**

are identical to BEAUti's and BEAST2's default settings. Per alignment, a site model, clock model and tree prior can be chosen. Multiple alignments can be used, each with its own (unlinked) site model, clock model and tree prior.

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

Currently, **babette** does not contain all functionality in BEAUti, BEAST2 and their many plug-ins, because these tools themselves also change in time. **babette** currently works only on DNA data, because this is the most common use case. Nevertheless, **babette** provides the majority of default tree priors and supports the most important command-line arguments of BEAST2, provides the core Tracer analysis options, and has the most basic subset of plotting options of DensiTree. Up till now, the **babette** features implemented are those requested by users. Further extension of **babette** will be based on future user requests.

### 3 Usage

**babette** can be installed easily from CRAN:

```
install.packages("babette")
```

128 For the most up-to-date version, one can download and install the package from  
129 **babette**'s GitHub repository:

```
130 devtools::install_github("richelbilderbeek/babette")
```

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

```
132 library(babette)
```

133 Because **babette** calls BEAST2, BEAST2 must be installed. This can be done  
134 from R, using:

```
135 install_beast2()
```

136 This will install BEAST2 to the default user data folder, but a different path  
137 can be specified as well. BEAUti, and likewise **babette**, needs at least a FASTA  
138 filename to produce a BEAST2 configuration file. In BEAUti, this is achieved  
139 by loading a FASTA file, then saving an output file using a common save file  
140 dialog. After this, BEAST2 needs to be applied to the created configuration  
141 file. It creates multiple files storing the posterior. These output files must be  
142 parsed by either Tracer or DensiTree. In **babette**, all this is achieved by:

```
143 out <- bbt_run(fasta_filenames = "anthus_aco.fas")
```

144 This code will create a (temporary) BEAST2 configuration file, from the FASTA  
145 file with name **anthus\_aco.fas** (which is supplied with the package, from  
146 Van Els & Norambuena 2018), using the same default settings as BEAUti, which  
147 are, among others, a Jukes-Cantor site model, a strict clock, and a Yule birth  
148 tree prior. **babette** will then execute BEAST2 using that file, and parses the  
149 output. The returned data structure, named **out**, is a list of parameter estimates  
150 (called **estimates**), posterior phylogenies (called **anthus\_aco\_trees**, named af-  
151 ter the alignment's name) and MCMC operator performance (**operators**). An  
152 example of using a different site model, clock model and tree prior is:

```
153 out <- bbt_run(
```

```

154   fasta_filenames = "anthus_aco.fas",
155   site_models = create_hky_site_model(),
156   clock_models = create_rln_clock_model(),
157   tree_priors = create_bd_tree_prior()
158 )

```

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

```

167 out <- bbt_run(
168   fasta_filenames = c(
169     "anthus_aco.fas",
170     "anthus_nd2.fas"
171   ),
172   site_models = list(
173     create_tn93_site_model(),
174     create_gtr_site_model()
175   )
176 )

```

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

182 Yule tree prior in **babette**:

```
183 out <- bbt_run(  
184   fasta_filenames = "anthus_aco.fas",  
185   tree_priors = create_yule_tree_prior(  
186     birth_rate_distr = create_exp_distr()  
187   )  
188 )
```

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

```
194 out <- bbt_run(  
195   fasta_filenames = "anthus_aco.fas",  
196   tree_priors = create_yule_tree_prior(  
197     birth_rate_distr = create_exp_distr(  
198       mean = create_mean_param(  
199         value = 1.0,  
200         estimate = FALSE  
201       )  
202     )  
203   )  
204 )
```

205 **babette** also supports node dating. Like **BEAUti**, one can specify Most Recent Common Ancestor ('MRCA') priors. An MRCA prior allows to specify taxa having a common ancestor, including a distribution for the date of that ancestor. With **babette**, this is achieved as follows:

```
209 out <- bbt_run(  
210   fasta_filenames = "anthus_aco.fas",  
211   tree_priors = create_yule_tree_prior(  
212     birth_rate_distr = create_exp_distr(  
213       mean = create_mean_param(  
214         value = 1.0,  
215         estimate = FALSE  
216       )  
217     )  
218   )  
219 )
```



```

210     fasta_filenames = "anthus_aco.fas",
211     mrca_priors = create_mrca_prior(
212         taxa_names = sample(get_taxa_names("anthus_aco.fas"),
213             size = 2),
214         alignment_id = get_alignment_id("anthus_aco.fas"),
215         is_monophyletic = TRUE,
216         mrca_distr = create_normal_distr(
217             mean = create_mean_param(value = 15.0, estimate =
218                 FALSE),
219             sigma = create_sigma_param(value = 0.025, estimate =
220                 FALSE)
221         )
222     )
223 )

```

224 Instead of dating the ancestor of two random taxa, any subset of taxa can  
 225 be selected, and multiple sets are allowed. **babette** allows for the same core  
 226 functionality as Tracer to show the values of the parameter estimates sampled  
 227 in the BEAST2 run. This is called the "trace" (hence the name). The start  
 228 of the trace, called the "burn-in", is usually discarded, as an MCMC algorithm  
 229 (such as used by BEAST2) first has to converge to its equilibrium and hence  
 230 the parameter estimates are not representative. By default, Tracer discards the  
 231 first 10% of all the parameter estimates. To remove a 20% burn-in from all  
 232 parameter estimates in **babette**, the following code can be used:

```

233 traces <- remove_burn_ins(
234     traces = out$estimates,
235     burn_in_fraction = 0.2
236 )

```

237 Tracer shows the ESSes of each posterior's variables. These ESSes are important

238 to determine the strength of the inference. As a rule of thumb, an ESS of 200 is  
239 acceptable for any parameter estimate. To calculate the effective sample sizes  
240 (of all estimated variables) in **babette**:

```
241 esses <- calc_esses(  
242   traces = traces,  
243   sample_interval = 1000  
244 )
```

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

```
250 sum_stats <- calc_summary_stats(  
251   traces = traces,  
252   sample_interval = 1000  
253 )
```

254 **babette** allows for the same functionality as **DensiTree**. **DensiTree** displays the  
255 phylogenies in a posterior at the same time scale, drawn one over one another,  
256 allowing to see the uncertainty in topology and branch lengths. The posterior  
257 phylogenies are stored as **anthus\_aco\_trees** in the object **out**, and can be  
258 plotted as follows:

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

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

```
263 create_beast2_input_file(
```

```

264   input_filenames = babette::get_babette_path("anthus_aco.
265       fas"),
266   output_filename = "beast2.xml"
267 )

```

268 This file can then be loaded and edited by BEAUti, run by BEAST2, or run by

269 **babette**:

```

270 run_beast2(
271   input_filename = "beast2.xml",
272   output_log_filename = "run.log",
273   output_trees_filenames = "posterior.trees",
274   output_state_filename = "final.xml.state"
275 )

```

276 **run\_beast2** is a function that only runs BEAST2, and does not parse the output  
 277 files (unlike **bbt\_run**). In the example above, we specify the names of the desired  
 278 BEAST2 output files explicitly, and these will be created in the R working  
 279 directory, after which they can be inspected with other tools, or used to continue  
 280 a BEAST2 run. When the names of these files are not specified, both **bbt\_run**  
 281 and **run\_beast2** put these files in the default temporary folder (as obtained  
 282 from **temp.dir()**) to keep the working directory clean of intermediate files.

## 283 4 **babette** resources

284 **babette** is free, libre and open source software available at [http://github.](http://github.com/richelbilderbeek/babette)  
 285 [com/richelbilderbeek/babette](http://github.com/richelbilderbeek/babette) and is licensed under the GNU General Pub-  
 286 lic License v3.0. **babette** uses the Travis CI (<https://travis-ci.org>) con-  
 287 tinuous integration service, which is known to significantly increase the number  
 288 of bugs exposed (Vasilescu *et al.* 2015) and increases the speed at which new  
 289 features are added (Vasilescu *et al.* 2015). **babette** has a 100% code cover-

age, which correlates with code quality (Horgan *et al.* 1994; Del Frate *et al.* 1995). **babette** follows Hadley Wickham’s style guide (Wickham 2015), which improves software quality (Fang 2001). **babette** depends on multiple packages, which are **ape** (Paradis *et al.* 2004), **beautier** (Bilderbeek 2018b), **beastier** (Bilderbeek 2018a), **devtools** (Wickham & Chang 2016), **geiger** (Harmon *et al.* 2008), **ggplot2** (Wickham 2009), **knitr** (Xie 2017), **phangorn** (Schliep 2011), **rmarkdown** (Allaire *et al.* 2017), **seqinr** (Charif & Lobry 2007), **stringr** (Wickham 2017), **testit** (Xie 2014) and **tracerer** (Bilderbeek 2018c). We tested **babette** to give a clean error message for incorrect input, by calling **babette** one million times with random or random sensible inputs, using a high performance computer cluster. The test scripts are supplied with **babette**.

**babette**’s development takes place on GitHub, <https://github.com/richelbilderbeek/babette>, which accommodates collaboration (Perez-Riverol *et al.* 2016) and improves transparency (Gorgolewski & Poldrack 2016). **babette**’s GitHub facilitates feature requests and has guidelines how to do so.

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

## 5 Citation of babette

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

316 use:

317 `> citation("babette")`

## 318 **6 Acknowledgements**

319 Thanks to Yacine Ben Chehida and Paul van Els for supplying their BEAST2  
320 use cases. Thanks again to Paul van Els for sharing his FASTA files for use  
321 by this package. Thanks to Leonel Herrera-Alsina, Raphael Scherrer and Gio-  
322 vanni Laudanno for their comments on this package and article. Thanks to  
323 Huw Ogilvie and one anonymous reviewer for reviewing this article. Thanks to  
324 rOpenSci, and especially Noam Ross and Guangchuang Yu for reviewing the  
325 package's source code. We would like to thank the Center for Information Tech-  
326 nology of the University of Groningen for their support and for providing access  
327 to the Peregrine high performance computing cluster. We thank the Nether-  
328 lands Organization for Scientific Research (NWO) for financial support through  
329 a VICI grant awarded to RSE.

## 330 **7 Authors' contributions**

331 RJCB and RSE conceived the idea for the package. RJCB created and tested  
332 the package, and wrote the first draft of the manuscript. RSE contributed  
333 substantially to revisions.

## 334 **References**

335 Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wick-  
336 ham, H., Cheng, J. & Chang, W. (2017) *rmarkdown: Dynamic Documents*  
337 *for R*. R package version 1.8.

338 Bilderbeek, R.J. (2018a) beastier: BEAST2 from R. [`https://github.com/`](https://github.com/)  
339 `richelbilderbeek/beastier` [Accessed: 2018-03-16].

340 Bilderbeek, R.J. (2018b) beautier: BEAUti 2 from R. [`https://github.com/`](https://github.com/)  
341 `richelbilderbeek/beautier` [Accessed: 2018-03-16].

342 Bilderbeek, R.J. (2018c) tracerer: Tracer from R. [`https://github.com/`](https://github.com/)  
343 `richelbilderbeek/tracerer` [Accessed: 2018-03-16].

344 Bouckaert, R. & Heled, J. (2014) Densitree 2: Seeing trees through the forest.  
345 *bioRxiv*, p. 012401.

346 Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C.H., Xie, D., Suchard,  
347 M.A., Rambaut, A. & Drummond, A.J. (2014) Beast 2: a software platform  
348 for bayesian evolutionary analysis. *PLoS Comput Biol*, **10**, e1003537.

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

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

359 Drummond, A.J., Suchard, M.A., Xie, D. & Rambaut, A. (2012) Bayesian phy-  
360 logenetics with beauti and the beast 1.7. *Molecular biology and evolution*, **29**,  
361 1969–1973.

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

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

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

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

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

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

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

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

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

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

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

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

391 Schliep, K. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**,  
392 592–593.

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

395 Vasilescu, B., Yu, Y., Wang, H., Devanbu, P. & Filkov, V. (2015) Quality and  
396 productivity outcomes relating to continuous integration in github. *Proceed-*  
397 *ings of the 2015 10th Joint Meeting on Foundations of Software Engineering*,  
398 pp. 805–816. ACM.

399 Wickham, H. (2009) *ggplot2: elegant graphics for data analysis*. Springer New  
400 York.

401 Wickham, H. (2015) *R packages: organize, test, document, and share your code*.  
402 O'Reilly Media, Inc.

403 Wickham, H. (2017) *stringr: Simple, Consistent Wrappers for Common String*  
404 *Operations*. R package version 1.2.0.

405 Wickham, H. & Chang, W. (2016) *devtools: Tools to Make Developing R Pack-*  
406 *ages Easier*. R package version 1.12.0.9000.

407 Xie, Y. (2014) *testit: A Simple Package for Testing R Packages*. R package  
408 version 0.4, <http://CRAN.R-project.org/package=testit>.



409 Xie, Y. (2017) *knitr: A General-Purpose Package for Dynamic Report Genera-*  
410 *tion in R*. R package version 1.17.

Name	Description
bbt_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

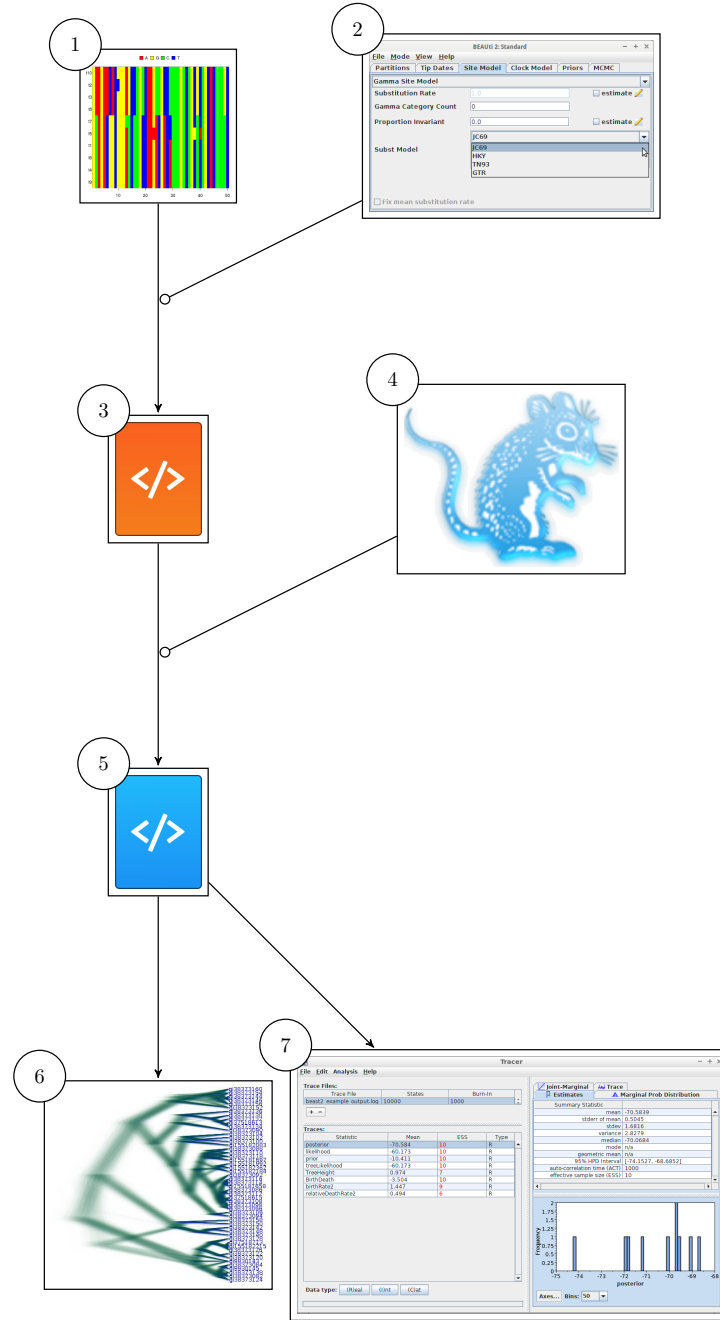


Figure 1: Workflow using GUI tools. From an alignment (1) and BEAUti (2), a BEAST2 configuration file (3) is created. BEAST2 (4) uses that file to infer a posterior, storing it in multiple files (5). These results are visualized using DensiTree (6) and Tracer (7). babette allows for the same workflow,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