

1 babette: BEAUti 2, BEAST2 and Tracer for R

2 Richèl J.C. Bilderbeek¹ and Rampal S. Etienne¹

3 ¹Groningen Institute for Evolutionary Life Sciences, University of
4 Groningen, Groningen, The Netherlands

5 May 2, 2019

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 Samenvatting

- 23 **1.** In de fylogenetica is BEAST2 een van de meest gebruikte hulppro-
24 gramma's. Het is gebundeld met de grafische gebruikersinterface BEAUti
25 2, DensiTree en Tracer, om BEAST2-configuratiebestanden te maken en
26 om BEAST2-outputbestanden te interpreteren. Echter, als veel verschil-
27 lende aligneringen of modelopzetten nodig zijn, is een werkvolgorde van
28 meerdere grafische gebruikersinterfaces onhandig.
- 29 **2.** Hier presenteren we een gratis, vrij en open-source package, **babette**:
30 'BEAUti 2, BEAST2 en Tracer voor R', voor de programmeertaal R.
31 **babette** schrijft BEAST2-configuratiebestanden, start BEAST2 and ver-
32 werkt de resultaten, alles met een enkele R functie-aanroep.
- 33 **3.** We beschrijven hoe **babette** te gebruiken is en de nieuwe mogelijkhe-
34 den die het biedt vergeleken met de originele programma's, aan de hand
35 van enkele voorbeelden.
- 36 **4.** Omdat **babette** ontworpen is voor uitbreidbaarheid en hoge kwaliteit,
37 sluiten we af met het beschrijven van de verdere ontwikkeling van dit pack-
38 age.

39
40 **Keywords:** computational biology, evolution, phylogenetics, BEAST2, R

41 **1 Introduction**

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

50 (see figure 1 for an overview of the workflow).

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

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

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

77 ence. DensiTree visualizes the phylogenies of a BEAST2 posterior, with many
78 options to improve the simultaneous display of many phylogenies.

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

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

95 2 Description

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

103 model, clock model and tree prior can be chosen. Multiple alignments can be
104 used, each with its own (unlinked) site model, clock model and tree prior.

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

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

124 3 Usage

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

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

127 For the most up-to-date version, one can download and install the package from

128 **babette**'s GitHub repository:

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

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

```
131 library(babette)
```

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

```
134 install_beast2()
```

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

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

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

```
152 out <- bbt_run(  
153   fasta_filenames = "anthus_aco.fas",
```

```

154   site_models = create_hky_site_model(),
155   clock_models = create_rln_clock_model(),
156   tree_priors = create_bd_tree_prior()
157 )

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

208 out <- bbt_run(
209   fasta_filenames = "anthus_aco.fas",

```



```

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

```

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

```

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

```

236 Tracer shows the ESSes of each posterior's variables. These ESSes are important
 237 to determine the strength of the inference. As a rule of thumb, an ESS of 200 is

238 acceptable for any parameter estimate. To calculate the effective sample sizes
239 (of all estimated variables) in **babette**:

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

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

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

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

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

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

```
262 create_beast2_input_file(  
263   input_filenames = babette::get_babette_path("anthus_aco.  
264   fas"),
```

```

265     output_filename = "beast2.xml"
266 )

```

267 This file can then be loaded and edited by BEAUti, run by BEAST2, or run by
268 **babette**:

```

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

```

275 **run_beast2** is a function that only runs BEAST2, and does not parse the output
276 files (unlike **bbt_run**). In the example above, we specify the names of the desired
277 BEAST2 output files explicitly, and these will be created in the R working
278 directory, after which they can be inspected with other tools, or used to continue
279 a BEAST2 run. When the names of these files are not specified, both **bbt_run**
280 and **run_beast2** put these files in the default temporary folder (as obtained
281 from **temp.dir()**) to keep the working directory clean of intermediate files.

282 4 **babette** resources

283 **babette** is free, libre and open source software available at [http://github.](http://github.com/richelbilderbeek/babette)
284 [com/richelbilderbeek/babette](http://github.com/richelbilderbeek/babette) and is licensed under the GNU General Pub-
285 lic License v3.0. **babette** uses the Travis CI (<https://travis-ci.org>) con-
286 tinuous integration service, which is known to significantly increase the number
287 of bugs exposed (Vasilescu *et al.* 2015) and increases the speed at which new
288 features are added (Vasilescu *et al.* 2015). **babette** has a 100% code cover-
289 age, which correlates with code quality (Horgan *et al.* 1994; Del Frate *et al.*
290 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, use:

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

317 6 Acknowledgements

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

329 7 Data Accessibility

330 All code is archived at http://github.com/richelbilderbeek/babette_article,
331 with DOI <https://doi.org/10.5281/zenodo.1251203>.

332 8 Authors’ contributions

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

336 References

337 Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wick-
338 ham, H., Cheng, J. & Chang, W. (2017) *rmarkdown: Dynamic Documents*

339 *for R*. R package version 1.8.

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

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

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

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

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

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

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

361 Drummond, A.J., Suchard, M.A., Xie, D. & Rambaut, A. (2012) Bayesian phy-
362 logenetics with beauti and the beast 1.7. *Molecular biology and evolution*, **29**,
363 1969–1973.

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

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

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

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

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

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

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

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

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

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

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

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

393 Schliep, K. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**,
394 592–593.

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

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

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

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

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

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

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

411 Xie, Y. (2017) *knitr: A General-Purpose Package for Dynamic Report Genera-*
412 *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

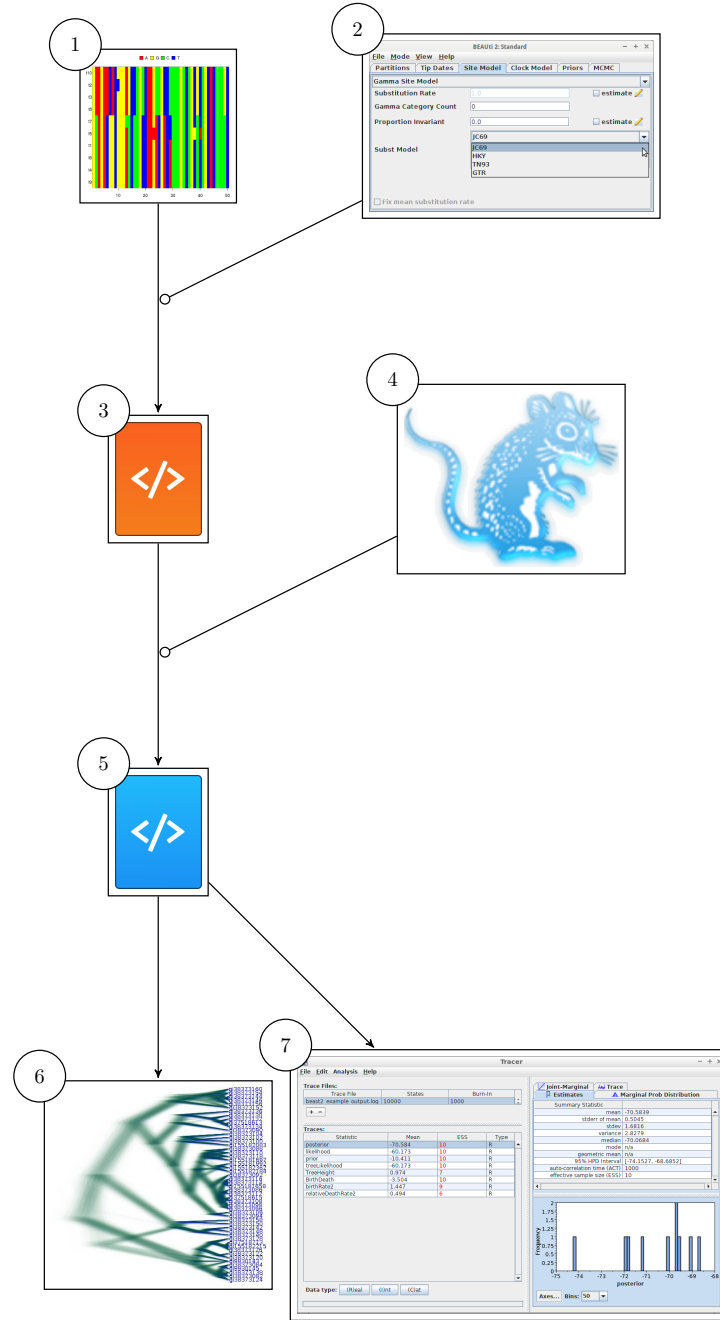


Figure 1: Workflow using GUI tools. From an alignment (1) and BEAUti (2), a BEAST2 configuration file (3) is created. BEAST2 (4) uses that file to infer a posterior, storing it in multiple files (5). These results are visualized using DensiTree (6) and Tracer (7). **babette** allows for the same workflow,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