

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 7, 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. rBEAST was developed to test a particular biological hypothesis
68(Ratmann *et al.* 2016), and hence was not designed for general use.

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

77 2 Description

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

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

98 Currently, **babette** does not contain all functionality in BEAUti, BEAST2
99 and their many plug-ins, because these tools themselves also change in time.
100 **babette** currently works only on DNA data, because this is the most common
101 use case. Nevertheless, **babette** provides the majority of default tree priors and
102 supports the most important command-line arguments of BEAST2, provides the
103 core Tracer analysis options, and has the most basic subset of plotting options of

104 DensiTree. Up till now, the `babette` features implemented are those requested
105 by users. Further extension of `babette` will be based on future user requests.

106 3 Usage

107 `babette` can be installed easily from CRAN **NOTE: This is not true yet:**
108 **we are still in the process of submitting to CRAN:**

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

110 For the most up-to-date version, one can download and install the package from
111 `babette`'s GitHub repository:

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

113 To start using `babette`, load its functions in the global namespace first:

```
114 library(babette)
```

115 Because `babette` calls BEAST2, BEAST2 must be installed. This can be done
116 from R, using:

```
117 install_beast2()
```

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

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

126 This code will create a (temporary) BEAST2 configuration file, from the FASTA
127 file with name `anthus_aco.fas` (which is supplied with the package, from

(Van Els & Norambuena 2018)), using the same default settings as BEAUti, which are, among others, a Jukes-Cantor site model, a strict clock, and a Yule birth tree prior. `babette` will then execute BEAST2 using that file, and parses the output. The returned data structure, named `out`, is a list of parameter estimates (called `estimates`), posterior phylogenies (called `anthus_aco_trees`, named after the alignment's name) and MCMC operator performance (`operators`). An example of using a different site model, clock model and tree prior is:

```
135 out <- bbt_run(
136   fasta_filenames = "anthus_aco.fas",
137   site_models = create_hky_site_model(),
138   clock_models = create_rln_clock_model(),
139   tree_priors = create_bd_tree_prior()
140 )
```

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:

```
149 out <- bbt_run(
150   fasta_filenames = c(
151     "anthus_aco.fas",
152     "anthus_nd2.fas"
153   ),
154   site_models = list(
155     create_tn93_site_model(),
```

```

156         create_gtr_site_model()
157     )
158 )

```

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

```

165 out <- bbt_run(
166     fasta_filenames = "anthus_aco.fas",
167     tree_priors = create_yule_tree_prior(
168         birth_rate_distr = create_exp_distr()
169     )
170 )

```

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

```

176 out <- bbt_run(
177     fasta_filenames = "anthus_aco.fas",
178     tree_priors = create_yule_tree_prior(
179         birth_rate_distr = create_exp_distr(
180             mean = create_mean_param(
181                 value = 1.0,
182                 estimate = FALSE
183             )

```

```

184     )
185   )
186 )

```

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

```

191 out <- bbt_run(
192   fasta_filenames = "anthus_aco.fas",
193   mrca_priors = create_mrca_prior(
194     taxa_names = sample(get_taxa_names("anthus_aco.fas"),
195       size = 2),
196     alignment_id = get_alignment_id("anthus_aco.fas"),
197     is_monophyletic = TRUE,
198     mrca_distr = create_normal_distr(
199       mean = create_mean_param(value = 15.0, estimate =
200         FALSE),
201       sigma = create_sigma_param(value = 0.025, estimate =
202         FALSE)
203   )
204 )
205 )

```

206 Instead of dating the ancestor of two random taxa, any subset of taxa can
207 be selected, and multiple sets are allowed. **babette** allows for the same core
208 functionality as Tracer to show the values of the parameter estimates sampled
209 in the BEAST2 run. This is called the "trace" (hence the name). The start
210 of the trace, called the "burn-in", is usually discarded, as an MCMC algorithm
211 (such as used by BEAST2) first has to converge to its equilibrium and hence

212 the parameter estimates are not representative. By default, Tracer discards the
213 first 10% of all the parameter estimates. To remove a 20% burn-in from all
214 parameter estimates in **babette**, the following code can be used:

```
215 traces <- remove_burn_ins(  
216   traces = out$estimates,  
217   burn_in_fraction = 0.2  
218 )
```

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

```
223 esses <- calc_esses(  
224   traces = traces,  
225   sample_interval = 1000  
226 )
```

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

```
232 sum_stats <- calc_summary_stats(  
233   traces = traces,  
234   sample_interval = 1000  
235 )
```

236 **babette** allows for the same functionality as DensiTree. DensiTree displays the
237 phylogenies in a posterior at the same time scale, drawn one over one another,
238 allowing to see the uncertainty in topology and branch lengths. The posterior

239 phylogenies are stored as `anthus_aco_trees` in the object `out`, and can be
240 plotted as follows:

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

242 Instead of running the full pipeline, `babette` also allows to only create a BEAST2
243 configuration file. To create a BEAST2 configuration file, with all settings to
244 default, use:

```
245 create_beast2_input_file(  
246   input_filenames = babette::get_babette_path("anthus_aco.  
247     fas"),  
248   output_filename = "beast2.xml"  
249 )
```

250 This file can then be loaded and edited by BEAUti, run by BEAST2, or run by
251 `babette`:

```
252 run_beast2(  
253   input_filename = "beast2.xml",  
254   output_log_filename = "run.log",  
255   output_trees_filenames = "posterior.trees",  
256   output_state_filename = "final.xml.state"  
257 )
```

258 In this example, we specify the names of the desired BEAST2 output files.
259 These output files can then be inspected with other tools, or used to continue a
260 BEAST2 run. `bbt_run` supports specifying the folder and name of these files,
261 which defaults to a temporary folder to keep the working directory clean of
262 intermediate and temporary files.

263 4 **babette** resources

264 **babette** is free, libre and open source software available at [http://github.](http://github.com/riche1bilderbeek/babette)
265 [com/riche1bilderbeek/babette](http://github.com/riche1bilderbeek/babette) and is licensed under the GNU General Pub-
266 lic License v3.0. **babette** uses the Travis CI (<https://travis-ci.org>) con-
267 tinuous integration service, which is known to significantly increase the number
268 of bugs exposed (Vasilescu *et al.* 2015) and increases the speed at which new
269 features are added (Vasilescu *et al.* 2015). **babette** has a 100% code cover-
270 age, which correlates with code quality (Horgan *et al.* 1994; Del Frate *et al.*
271 1995). **babette** follows Hadley Wickham’s style guide (Wickham 2015), which
272 improves software quality (Fang 2001). **babette** depends on multiple packages,
273 which are **ape** (Paradis *et al.* 2004), **beautier** (Bilderbeek 2018b), **beastier**
274 (Bilderbeek 2018a), **devtools** (Wickham & Chang 2016), **geiger** (Harmon *et al.*
275 2008), **ggplot2** (Wickham 2009), **knitr** (Xie 2017), **phangorn** (Schliep 2011),
276 **rmarkdown** (Allaire *et al.* 2017), **seqinr** (Charif & Lobry 2007), **stringr** (Wick-
277 ham 2017), **testit** (Xie 2014) and **tracerer** (Bilderbeek 2018c). We tested
278 **babette** to give a clean error message for incorrect input, by calling **babette**
279 one million times with random or random sensible inputs, using the Peregrine
280 high performance computer cluster. The test scripts are supplied with **babette**.

281 **babette**’s development takes place on GitHub, <https://github.com/riche1bilderbeek/>
282 **babette**, which accommodates collaboration (Perez-Riverol *et al.* 2016) and im-
283 proves transparency (Gorgolewski & Poldrack 2016). **babette**’s GitHub facili-
284 tates feature requests and has guidelines how to do so.

285 **babette**’s documentation is extensive. All functions are documented in the
286 package’s internal documentation. For quick use, each exported function shows
287 a minimal example. For easy exploration, each exported function’s documen-
288 tation links to related functions. Additionally, **babette** has a vignette that
289 demonstrates extensively how to use it. There is documentation on the GitHub

290 to get started, with a dozen examples of BEAUti screenshots with equivalent
291 **babette** code. Finally, **babette** has tutorial videos that can be downloaded or
292 viewed on YouTube, <https://goo.gl/weKaaU>.

293 5 Citation of babette

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

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

298 6 Acknowledgements

299 Thanks to Yacine Ben Chehida and Paul van Els for supplying their BEAST2
300 use cases. Thanks again to Paul van Els for sharing his FASTA files for use
301 by this package. Thanks to Leonel Herrera-Alsina, Raphael Scherrer and Gio-
302 vanni Laudanno for their comments on this package and article. Thanks to
303 Huw Ogilvie and one anonymous reviewer for reviewing this article. Thanks to
304 rOpenSci, and especially Noam Ross and Guangchuang Yu for reviewing the
305 package's source code. We would like to thank the Center for Information Tech-
306 nology of the University of Groningen for their support and for providing access
307 to the Peregrine high performance computing cluster.

308 7 Authors' contributions

309 RJCB and RSE conceived the idea for the package. RJCB created and tested
310 the package, and wrote the first draft of the manuscript. RSE contributed
311 substantially to revisions.

References

- Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J. & Chang, W. (2017) *rmarkdown: Dynamic Documents for R*. R package version 1.8.
- Bilderbeek, R.J. (2018a) beastier: BEAST2 from R. <https://github.com/ricelbilderbeek/beastier> [Accessed: 2018-03-16].
- Bilderbeek, R.J. (2018b) beautier: BEAUti 2 from R. <https://github.com/ricelbilderbeek/beautier> [Accessed: 2018-03-16].
- Bilderbeek, R.J. (2018c) tracerer: Tracer from R. <https://github.com/ricelbilderbeek/tracerer> [Accessed: 2018-03-16].
- Bouckaert, R. & Heled, J. (2014) Densitree 2: Seeing trees through the forest. *bioRxiv*, p. 012401.
- Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C.H., Xie, D., Suchard, M.A., Rambaut, A. & Drummond, A.J. (2014) Beast 2: a software platform for bayesian evolutionary analysis. *PLoS Comput Biol*, **10**, e1003537.
- Charif, D. & Lobry, J. (2007) SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. U. Bastolla, M. Porto, H. Roman & M. Vendruscolo, eds., *Structural approaches to sequence evolution: Molecules, networks, populations*, Biological and Medical Physics, Biomedical Engineering, pp. 207–232. Springer Verlag, New York. ISBN : 978-3-540-35305-8.
- Del Frate, F., Garg, P., Mathur, A.P. & Pasquini, A. (1995) On the correlation between code coverage and software reliability. *Software Reliability Engineering, 1995. Proceedings., Sixth International Symposium on*, pp. 124–132. IEEE.

337 Drummond, A.J., Suchard, M.A., Xie, D. & Rambaut, A. (2012) Bayesian phy-
338 logenetics with beauti and the beast 1.7. *Molecular biology and evolution*, **29**,
339 1969–1973.

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

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

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

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

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

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

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

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

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

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

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

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

369 Schliep, K. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**,
370 592–593.

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

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

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

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

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

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

385 Xie, Y. (2014) *testit: A Simple Package for Testing R Packages*. R package
386 version 0.4, <http://CRAN.R-project.org/package=testit>.
387 Xie, Y. (2017) *knitr: A General-Purpose Package for Dynamic Report Genera-*
388 *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

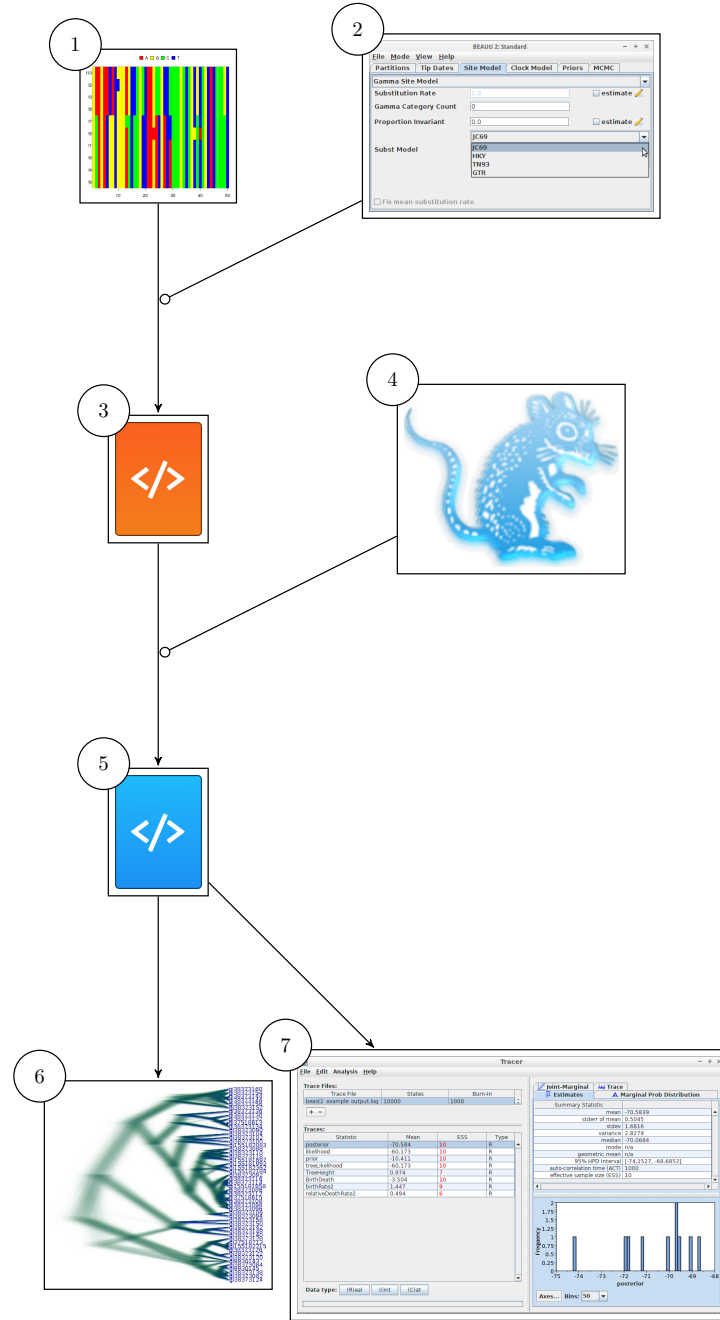


Figure 1: Workflow using GUI tools. From an alignment (1) and BEAUti (2), a BEAST2 configuration file (3) is created. BEAST2 (4) uses that file to infer a posterior, storing it in multiple files (5). These results are visualized using DensiTree (6) and Tracer (7). babette allows for the same workflow,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