ABC coverage vignette

Dennis Prangle

April 16, 2013

This document illustrates the use of coverage diagnostics from the *abctools* package in an analysis of the *human* dataset provided with the *abc* package. Parameter inference and model choice analyses are presented. These are similar to the analyses done in the Prangle et al. (2013) paper on other datasets, but with less demands on time and computational resources. One of the aims of this document is to illustrate how to produce plots from package output, so the full details of R commands are included.

1 Background

There follow some excerpts from the *abc* package help files describing the *human* dataset. The full *abc* package help files and vignettes contain more details, such as of the summary statistics used.

"Data is provided to estimate the posterior probabilities of classical demographic scenarios in three human populations: Hausa, Italian, and Chinese. These three populations represent the three continents: Africa, Europe, Asia, respectively. `par.italy.sim` may then [be] used to estimate the ancestral population size of the European population assuming a bottleneck model."

"The observed statistics were taken from Voight et al. (2005) (Table 1.). Also, the same input parameters were used as in Voight et al. (2005) to simulate data under the three demographic models. Simulations were performed using the software `ms` and the summary statistics were calculated using `sample_stats`" [n.b. `ms` is described in Hudson (2002) and `sample_stats` is provided with `ms` and described in its manual]

"`data(human)` loads in four R objects: `stat.voight` is a data frame with 3 rows and 3 columns and contains the observed summary statistics for three human populations, `stat.3pops.sim` is also a data frame with 150,000 rows and 3 columns and contains the simulated summary statistics, `models` is a vector of character strings of length 150,000 and contains the model indices, `par.italy.sim` is a data frame with 50,000 rows and 4 columns and contains the parameter values that were used to simulate data under a population bottleneck model. The corresponding summary statistics can be subsetted from the `stat.3pops.sim` object as `subset(stat.3pops.sim, subset=models=="bott")`."

2 Parameter inference

We focus on parameter inference under the bottleneck model using the `stat.voight["italian",]` observations. First we create a dataframe containing the summary statistics simulated under this model.

```
> library(abctools)
> data(human)
> stat.italy.sim <- subset(stat.3pops.sim, subset=models=="bott")
```

Next we perform an initial ABC rejection sampling analysis.

```
> abc.out <- abc(target = stat.voight["italian", ], param = par.italy.sim,
+                sumstat = stat.italy.sim, tol = 0.05, method = "rejection")
```

Amongst other information, the `abc` function returns details of the distance between each simulated dataset and the observations. This information is important to the coverage diagnostic analysis. Firstly it allows calculation of the ABC threshold ϵ equivalent to the 5% tolerance level used above:

```
> nacc <- nrow(par.italy.sim)*0.05
> sort(abc.out$dist, partial=nacc)[nacc]

[1] 0.7074183
```

Secondly, distance information can be used to find an interesting range of ϵ values. Here, the ϵ values to be investigated cover the full range of distances which can be achieved from these simulations (Equally spaced values on a log scale are used). This is in order to illustrate some properties of the coverage diagnostics. In practice the lower end of this range would usually be of most interest, as these produce the most accurate ABC analyses.

```
> summary(abc.out$dist)

    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
 0.06088  1.31100  1.84900  1.99800  2.48600 10.07000

> epsvec <- seq(log(0.06), log(10), length.out=15)
> epsvec <- exp(epsvec)
```

The coverage diagnostics are calculated by performing ABC analyses on many *test sets* i.e. datasets simulated from known parameter values (also sometimes referred to as *pseudo-observed datasets*.) These are selected from amongst those already simulated in `stat.italy.sim`. Prangle et al. (2013) recommend using the rows with minimal distance to the observed data as test data sets. Our initial ABC analysis output can be used to determine these. As a comparison we will also look at the performance of choosing test data sets by a simple random sample.

```
> set.seed(1)
> testsets.anywhere <- sample(1:50000, 200)
> testsets.neardata <- order(abc.out$dist)[1:200]
```

Now the *abctools* package code is used. The `cov.pi` command performs ABC analyses using each of the test sets as the observerations and several choices of ϵ. Each analysis uses the same simulated parameter values and data sets (after removing the row currently being used as observations). The results are used to compute diagnostic statistics. The required ABC analyses are run in parallel using the *parallel* package (This can be disabled by setting `multicore=FALSE`.)

```
> cabc.out.anywhere <- cov.pi(param=par.italy.sim, sumstat=stat.italy.sim,
+                             testsets=testsets.anywhere, eps=epsvec,
+                             diagnostics=c("KS", "CGR"), multicore=TRUE, cores=4)
> cabc.out.neardata <- cov.pi(param=par.italy.sim, sumstat=stat.italy.sim,
+                             testsets=testsets.neardata, eps=epsvec,
+                             diagnostics=c("KS", "CGR"), multicore=TRUE, cores=4)
```

The coverage property is a desirable property of a method producing approximations to a Bayesian posterior. It focuses on the case of a scalar parameter. For multivariate parameters, each can be examined separately. Roughly speaking, the coverage property asserts that credible intervals have the claimed coverage levels. For example 95% of 95% credible intervals should contain the true parameters values. An equivalent condition is that the cdf values of the true parameter values under the approximate posteriors (refered to p_0 values) are uniformly distributed on $[0, 1]$. This is what the `cov.pi` command tests.

For each ABC analysis it performs, `cov.pi` records a vector of estimated p_0 values of the correct parameters. A Bayesian estimator is used which is similar to the empirical cdf, but avoids the extreme values of 0 and 1, as some later calculations cannot cope with these. These estimators are recorded in one component of `cov.pi` output, *raw*.

```
> head(cabc.out.anywhere$raw)
```

```
  testset  eps nacc        Ne         a  duration     start
1   13276 0.06    1 0.6666667 0.6666667 0.3333333 0.3333333
2   18606 0.06    3 0.8000000 0.4000000 0.8000000 0.2000000
3   28642 0.06    1 0.6666667 0.3333333 0.6666667 0.6666667
4   45408 0.06    2 0.5000000 0.5000000 0.2500000 0.7500000
5   10084 0.06    0 0.5000000 0.5000000 0.5000000 0.5000000
6   44915 0.06    3 0.2000000 0.4000000 0.6000000 0.6000000
```

Each row corresponds to one combination of test set and ϵ. These values are recorded as the first two columns. The third column gives the number of acceptances in the ABC analysis, and the remaining columns give p_0 estimates for each parameter (column titles giving the parameter names).

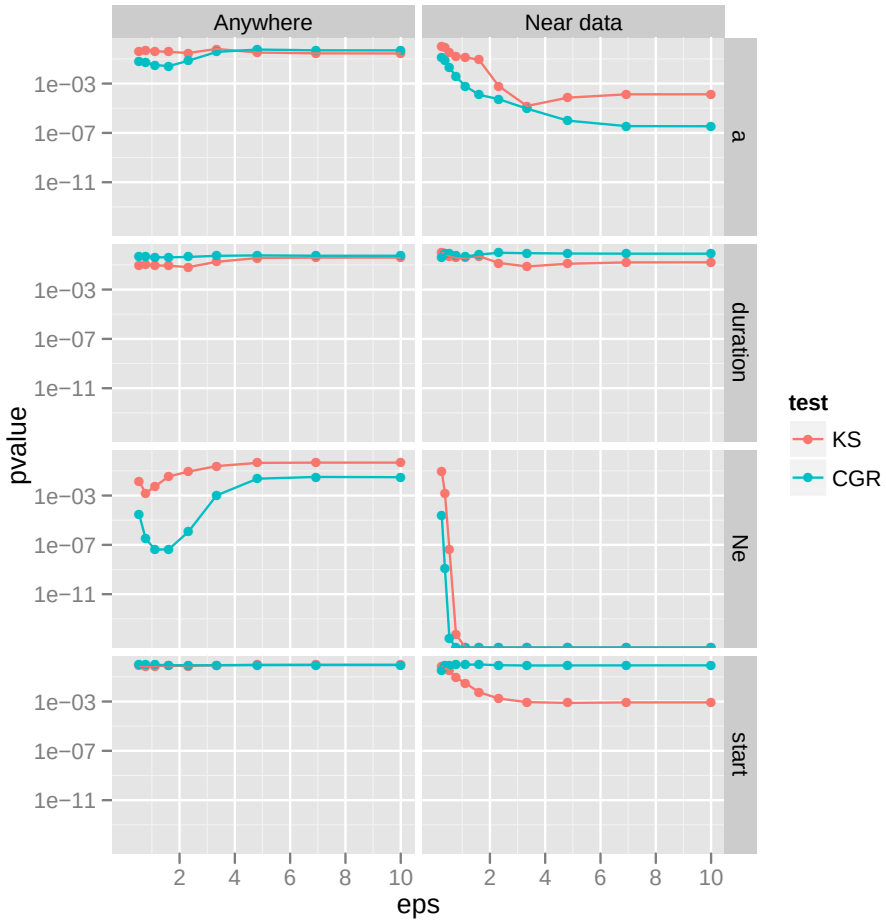
The `cov.pi` commands above have also computed several diagnostic statistics to test whether the p_0 estimates are roughly uniform on $[0, 1]$. For each combination of parameter, ϵ and test statistic, a p-value of the statistic is given under the assumption that the coverage property holds. These are recorded in another component of the output, *diag*.

```
> tail(cabc.out.anywhere$diag)
```

```
          eps parameter     pvalue test
115  6.938994  duration 0.55279077  CGR
116  6.938994     start 0.90431836  CGR
117 10.000000        Ne 0.03070583  CGR
118 10.000000         a 0.49117877  CGR
119 10.000000  duration 0.54812453  CGR
120 10.000000     start 0.90639422  CGR
```

The purpose of the diagnostics statistics is that they can easily be plotted to judge when there is evidence to reject the hypothesis that the coverage property holds, as follows.

```
> library(ggplot2)
> diag.all <- rbind(cbind(cabc.out.anywhere$diag, V="Anywhere"),
+                   cbind(cabc.out.neardata$diag, V="Near data"))
> qplot(x=eps, y=pvalue, colour=test, facets=parameter~V, data=diag.all,
+       log="y") + geom_line()
```

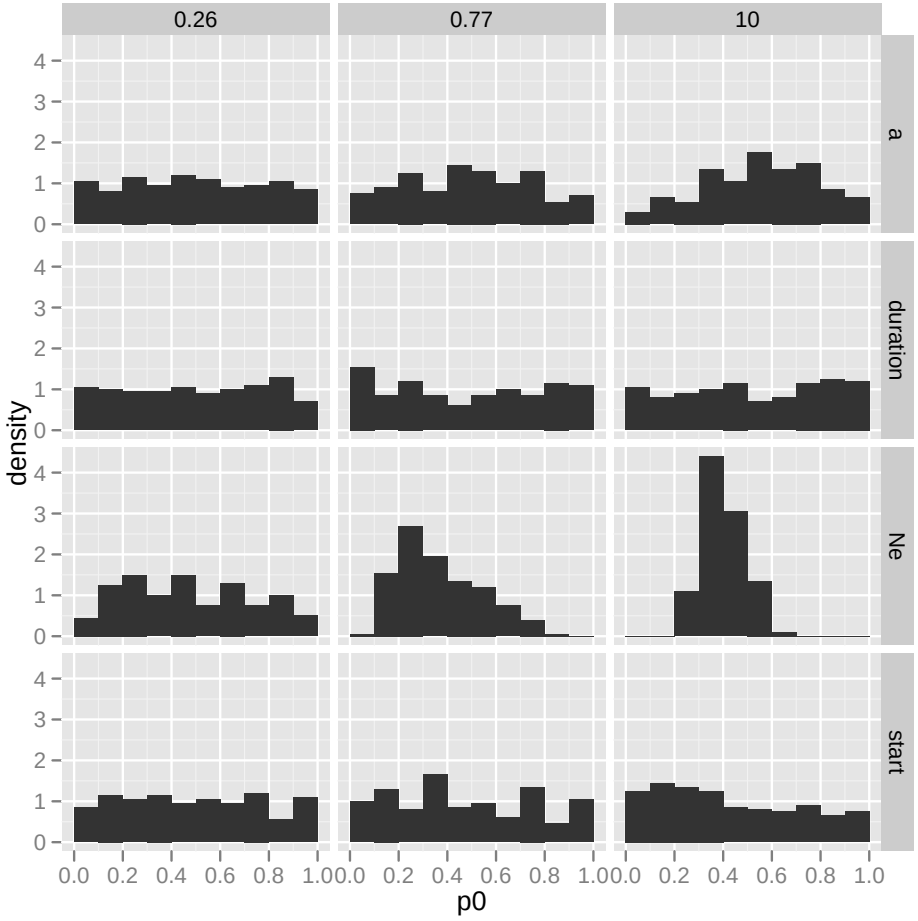


This figure illustrates several points of interest. Firstly, using a random sample of test sets does not reject the coverage hypothesis for large ϵ. In fact it can be proved (Prangle et al., 2013) that for this choice of test sets the coverage property holds under the prior distribution, which is the ABC target distribution for sufficiently large ϵ. Secondly, using a random sample of test sets cannot assess as wide a range of ϵ values as the alternative choice (i.e. fewer points are plotted in the left column.) This is because for the smallest ϵ value some of the test data sets produce too small an ABC sample to produce trustworthy estimates of p_0 values and so these diagnostics are reported as `NA` (The threshold of acceptances below which this happens can be set by the `cov.pi` argument `nacc.min`. Its default is 20.) These are the main reasons that we recommend using test sets chosen near the data.

Another point is that there is some disagreement between the diagnostics. For example, the bottom right panel shows that for $\epsilon = 10$ the coverage property for the *start* parameter is rejected by one diagnostic only. We argue in Prangle et al. (2013) that no test statistic can be expected to detect all types of deviation from uniformity so the raw results should be investigated for interesting values of ϵ. Here, for test sets near the

observed data, we investigate $\epsilon = 0.26$ (the smallest value that produced enough acceptances), 0.77 (the closest value to that used in the analysis of the real data), and 10 (to investigate the disagreement just mentioned).

```
> temp <- subset(cabc.out.neardata$raw, eps %in% epsvec[c(5,8,15)])
> parnames <- colnames(par.italy.sim)
> temp2 <- reshape(temp, varying=list(parnames), v.names="p0",
+                  direction="long", timevar="parameter", times=parnames)
> temp2$eps <- signif(temp2$eps,2)
> ggplot(temp2, aes(x=p0)) +
+   geom_histogram(aes(y=..density..), breaks=0:10/10) +
+   facet_grid(parameter~eps)
```

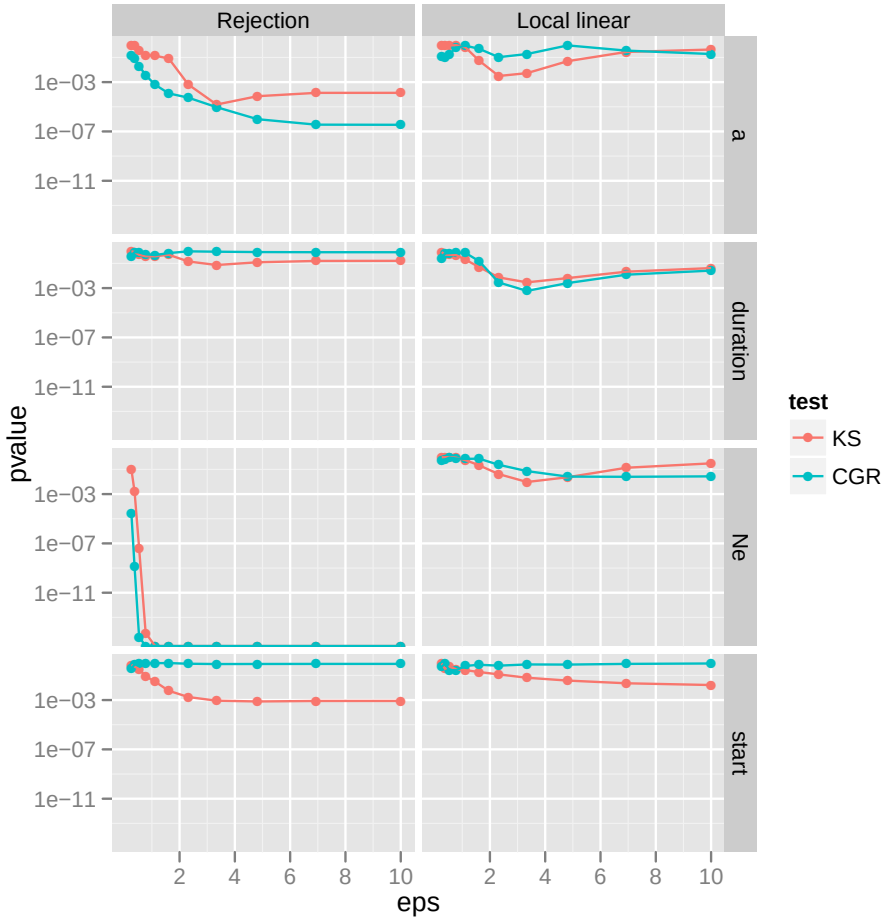


As ϵ is reduced, the histograms become closer to the uniform pdf, in line with the behaviour of the KS diagnostic in the previous figure. However even the choice of $\epsilon = 0.26$ appears too large for coverage to hold for all parameters. Note that for $\epsilon = 10$, the histogram for the *start* parameter shows a deviation from uniformity: the estimated p_0 values are biased toward smaller values. The KS diagnostic has detected this deviation while the CGR diagnostic has not.

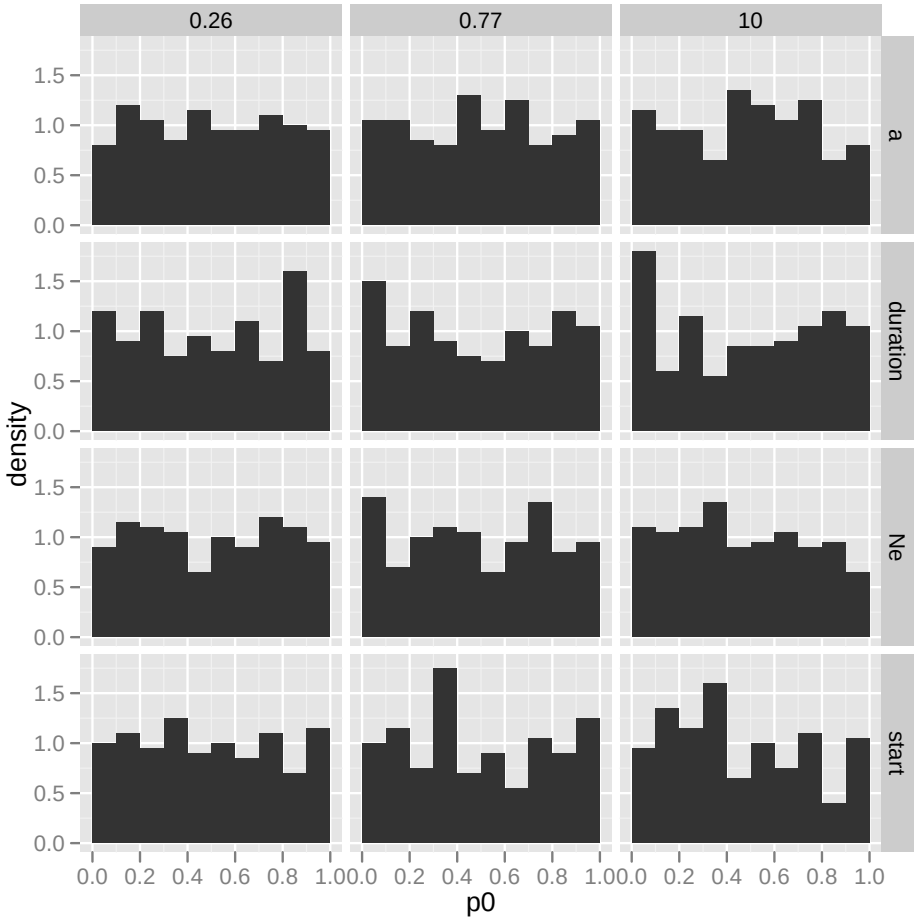
The analysis can be repeated for ABC followed by regression post-processing. The plots below show $\epsilon = 0.77$ is a roughly acceptable choice in this case.

```
> cabc.out.corr<- cov.pi(param=par.italy.sim, sumstat=stat.italy.sim,
+                        testsets=testsets.neardata, eps=epsvec,
```

```
+                              method="loclinear", diagnostics=c("KS", "CGR"),
+                              multicore=TRUE, cores=4)
> diag.all <- rbind(cbind(cabc.out.neardata$diag, method="Rejection"),
+                   cbind(cabc.out.corr$diag, method="Local linear"))
> qplot(x=eps, y=pvalue, colour=test, facets=parameter~method, data=diag.all,
+       log="y") + geom_line()
```




```
> temp <- subset(cabc.out.corr$raw, eps %in% epsvec[c(5,8,15)])
> temp2 <- reshape(temp, varying=list(parnames), v.names="p0", direction="long",
+                  timevar="parameter", times=parnames)
> temp2$eps <- signif(temp2$eps,2)
> ggplot(temp2, aes(x=p0)) + geom_histogram(aes(y=..density..), breaks=0:10/10) +
+   facet_grid(parameter~eps)
```



3 Model choice

The *abctools* package can be used similarly for model choice. Here we consider model choice between three candidate models using the same observed data as before. Again we start by performing an ABC rejection sampling analysis (This is done with the `abc` command rather than `postpr` so that distances from simulated to observed data are available as a byproduct.)

```
> mod.num <- as.numeric(factor(models))
> abc.mod <- abc(target = stat.voight["italian", ], param=mod.num,
+                sumstat=stat.3pops.sim, tol = 0.05, method = "rejection")
```

As before, the computed distances are used to find an interesting range of ϵ values and choose test data sets.

```
> set.seed(2)
> testsets.anywhere <- sample(1:1.5E5, 200)
> testsets.neardata <- order(abc.mod$dist)[1:200]
> ##Which value of epsilon was used in analysis?
> nacc <- 1.5E5*0.05
> sort(abc.mod$dist, partial=nacc)[nacc]

[1] 1.011973
```

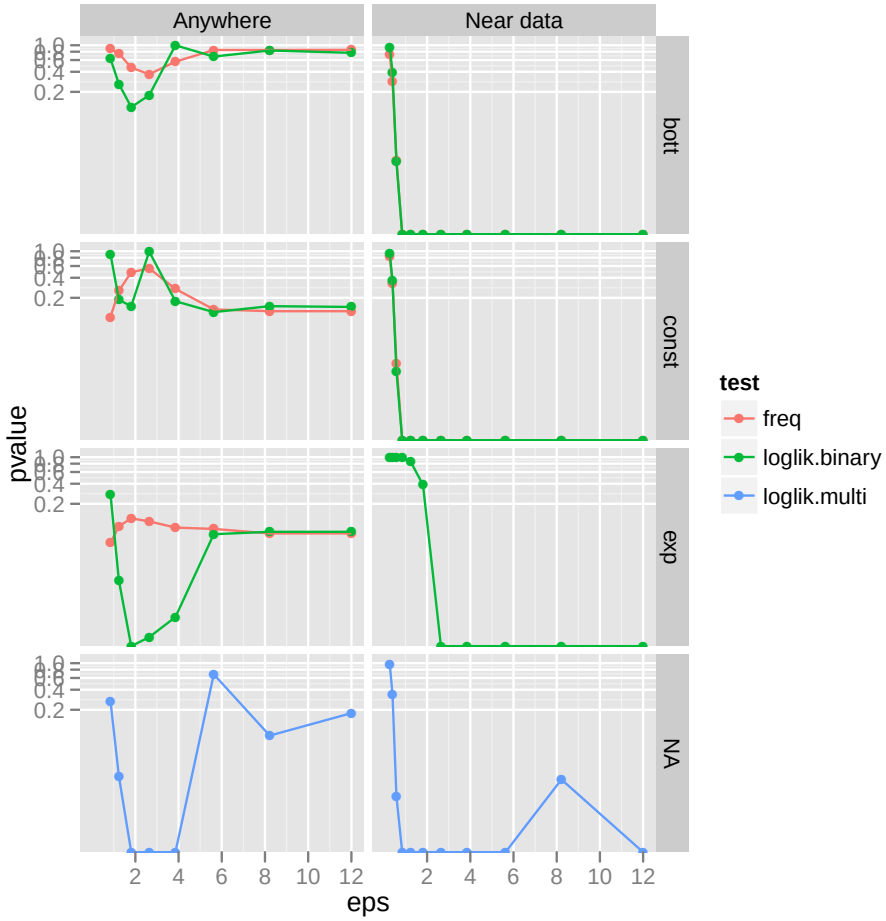
```
> ##Find interesting range of epsilon values
> summary(abc.mod$dist)

    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
 0.05862  1.95100  2.79100  2.76200  3.50800 11.50000

> epsvec <- seq(log(0.06), log(12), length.out=15)
> epsvec <- exp(epsvec)
```

Similar syntax to the parameter inference case is used to perform the ABC analyses and the calculation and plotting of diagnostics. A rough definition of the coverage property for model choice is given below: see Prangle et al. (2013) for more discussion.

```
> cabc.mod.anywhere <- cov.mc(index=models, sumstat=stat.3pops.sim,
+                             testsets=testsets.anywhere, eps=epsvec,
+                             diagnostics=c("freq", "loglik.binary",
+                             "loglik.multi"),
+                             multicore=TRUE, cores=4)
> cabc.mod.neardata <- cov.mc(index=models, sumstat=stat.3pops.sim,
+                             testsets=testsets.neardata, eps=epsvec,
+                             diagnostics=c("freq", "loglik.binary",
+                             "loglik.multi"),
+                             multicore=TRUE, cores=4)
> diag.all <- rbind(cbind(cabc.mod.anywhere$diag, V="Anywhere"),
+                   cbind(cabc.mod.neardata$diag, V="Near data"))
> qplot(x=eps, y=pvalue, colour=test, facets=parameter~V, data=diag.all,
+       log="y") + geom_line()
```



The *freq*, *loglik.binary* and *loglik.multi* diagnostics correspond to the test statistics U, V and W in Prangle et al. (2013). The first two of these test coverage for one model in particular, so we show three rows considering each model in turn. On the other hand, *loglik.multi* tests coverage for all models, so it is shown in a separate row. As for the parameter inference case, using test sets sampled from the prior does not reject coverage for large ϵ values, and choosing test sets close to the data is recommended. The plots suggest $\epsilon = 0.40$ is sufficient for coverage to approximately hold.

The raw data now take a different form. For each combination of test data set and ϵ value, the estimated probability of each model is given (as well as number of acceptances):

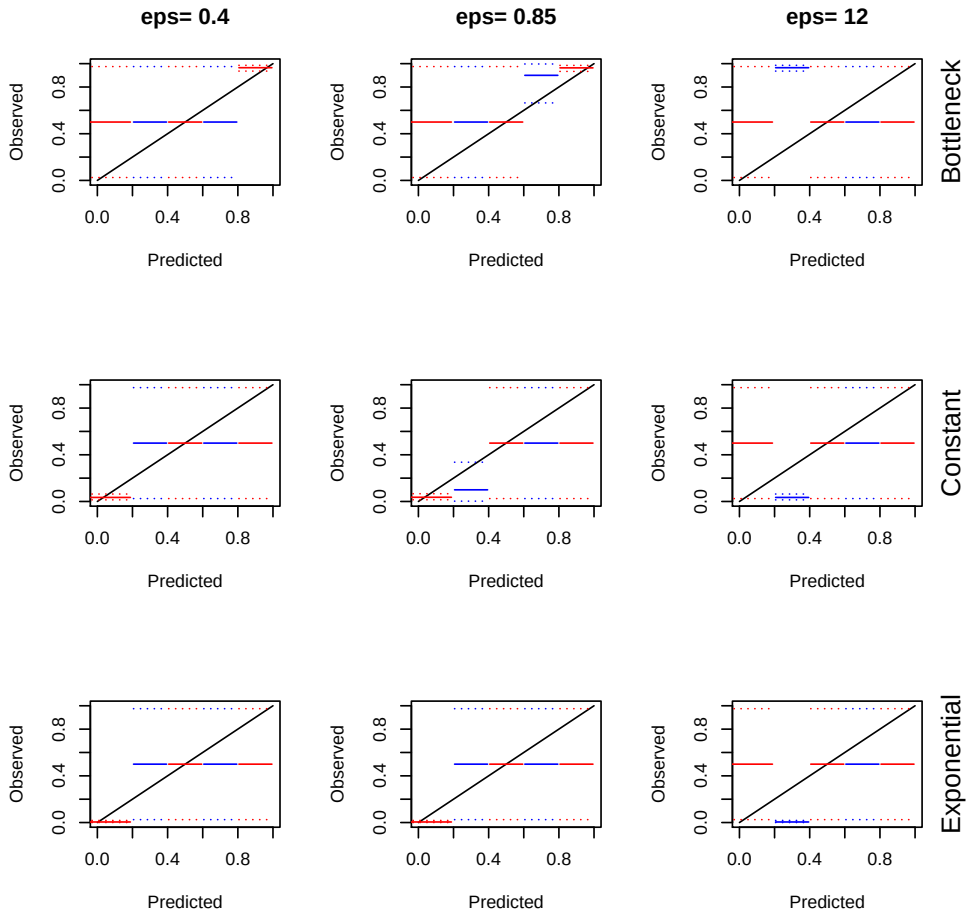
```
> tail(cabc.mod.anywhere$raw)
```

```
     testset eps   nacc      bott      const        exp
2995   75360  12 149993 0.3333333 0.3333200 0.3333467
2996   30194  12 149999 0.3333333 0.3333333 0.3333333
2997   29920  12 149993 0.3333333 0.3333200 0.3333467
2998   26686  12 149993 0.3333333 0.3333200 0.3333467
2999   41026  12 149998 0.3333289 0.3333356 0.3333356
3000   19568  12 149999 0.3333333 0.3333333 0.3333333
```

Under the coverage property, the true probability that a model occurs conditional on an estimated probability z_0 should equal z_0, and the diagnostics test this. Testing this property is harder than the continuous parameter case, and the diagnostics are intended

as a starting point for analysis that should be treated with caution. Raw plots of results for a particular ϵ value and target model can be plotted using the `mc.ci` command. This splits the range of estimated model probabilities $[0, 1]$ into intervals and estimates the conditional true probability of the model for each interval. When coverage holds, the 95% credible intervals shown should usually contain the 45 degree line. Colours are used in the plot to distinguish results for adjoining intervals.

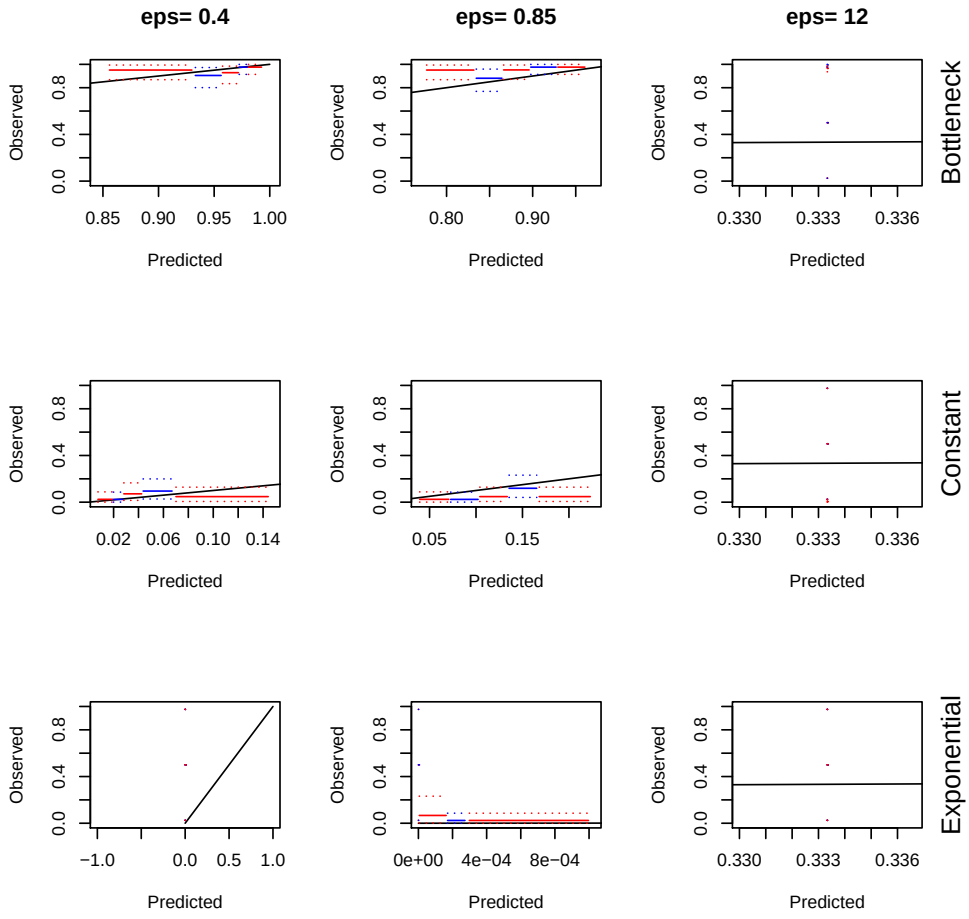
```
> par(mfcol=c(3,3))
> for (i in c(6,8,15)) {
+   mc.ci(cabc.mod.neardata$raw, eps=epsvec[i], modname="bott",
+         modtrue=models)
+   title(main=paste("eps=", signif(epsvec[i],2)))
+   if (i==15) mtext("Bottleneck", side=4, line=1)
+   mc.ci(cabc.mod.neardata$raw, eps=epsvec[i], modname="const",
+         modtrue=models)
+   if (i==15) mtext("Constant", side=4, line=1)
+   mc.ci(cabc.mod.neardata$raw, eps=epsvec[i], modname="exp",
+         modtrue=models)
+   if (i==15) mtext("Exponential", side=4, line=1)
+ }
```



The plots illustrate that coverage fails to hold for $\epsilon = 12$. Here all predicted probabilities roughly equal 1/3, the prior weight, which is incorrect. The plots are not conclusive here

for smaller ϵ and serve only as a starting point for application specific further investigation. One way to do so is to use a different partition of $[0, 1]$. `mc.ci` also supports intervals based on quantiles of predicted probabilities, which in this case are harder to interpret visually but do reveal some possible deviations from coverage for $\epsilon = 0.85$:

```
> par(mfcol=c(3,3))
> for (i in c(6,8,15)) {
+   mc.ci(cabc.mod.neardata$raw, eps=epsvec[i], modname="bott",
+         modtrue=models, bintype="quantile")
+   title(main=paste("eps=", signif(epsvec[i],2)))
+   if (i==15) mtext("Bottleneck", side=4, line=1)
+   mc.ci(cabc.mod.neardata$raw, eps=epsvec[i], modname="const",
+         modtrue=models, bintype="quantile")
+   if (i==15) mtext("Constant", side=4, line=1)
+   mc.ci(cabc.mod.neardata$raw, eps=epsvec[i], modname="exp",
+         modtrue=models, bintype="quantile")
+   if (i==15) mtext("Exponential", side=4, line=1)
+ }
```



References

Hudson, R. R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18:337–338.

Prangle, D., Blum, M. G. B., Popovic, G., and Sisson, S. A. (2013). Diagnostic tools for approximate Bayesian computation using the coverage property. *Preprint*. arxiv.org/abs/1301.3166.

Voight, B. F., Adams, A. M., Frisse, L. A., Qian, Y., Hudson, R. R., and Di Rienzo, A. (2005). Interrogating multiple aspects of variation in a full resequencing data set to infer human population size changes. *PNAS*, 102:18508–18513.