Review of manuscript gchron-2021-39 by Dietze et al.

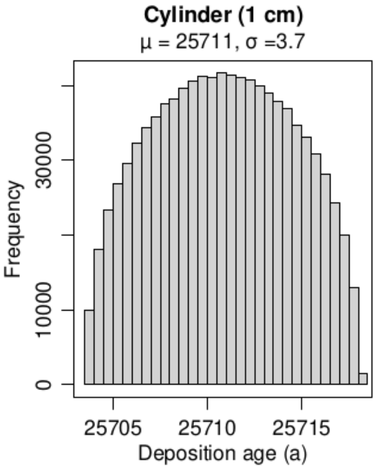
1 Strengths

The lead author is one of the most accomplished `R` programmers in the Earth Sciences. The `sandbox` package is logically implemented and well documented. It follows `R`'s list-based programming paradigm and uses the language's strengths in data visualisation whilst avoiding some of its weaknesses. For example, the package uses parallel processing for its calculation, thereby overcoming `R`'s computational limitations. All this is achieved with minimal dependencies. Unlike many other packages, sandbox does not require `ggplot2`, which makes the package light and nimble.

2 Major comments

2.1 Usefulness

I am not sure how useful `sandbox` is, and I doubt that it will ever become very popular. The three applications provided in Section 3.5 are not very convincing. Take, for example, the first case study, which investigates the effect of sample geometry on the single grain OSL age distribution. It shows that a cylindrical container results in a circular age distribution:



It is easy to derive this distribution analytically:

```
x <- seq(from=-1,to=1,length.out=100)
y <- sqrt(1-x^2)
sedrate <- 1/8   # cm/yr, sedimentation rate
thickness <- 1   # cm, diameter of the container
t0 <- 25711      # age in centre of the cylinder
age <- x*thickness/sedrate + t0
freq <- y/sum(y) # normalise
plot(age,freq,type='l')
```

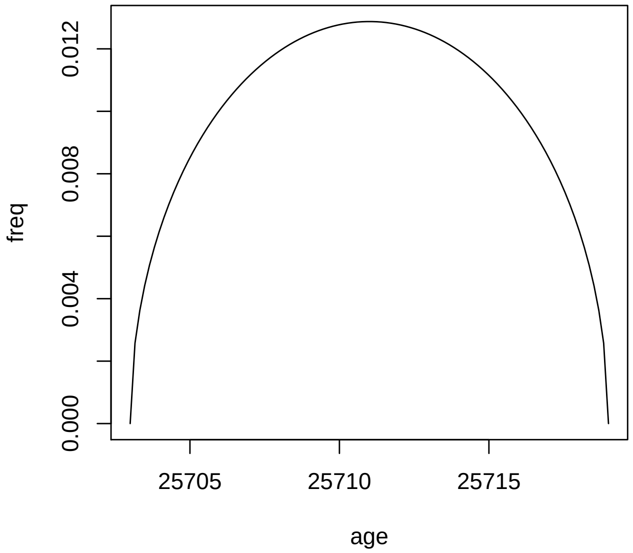


Figure 1: Expected age distribution for a cylindrical container.

The second application is similar to the first, whereas the third example did not make sense to me. I did not understand why sieving would cause ‘age inversions’ in the depth profiles. Do the coarse grains contain more inheritence than the small ones? If this is so, then I must have missed where this was specified. This should be explained better.

2.2 Crudeness

The virtual sediment sections defined by `sandbox` ‘rule books’ are not process-based. The algorithms are purely statistical and do not include any physics. This severely reduces their degree of realism, and limits their usefulness, as I will explain next.

Each location in the virtual sediment section is assumed to contain a discrete number of subpopulations. Each of these distributions is assumed to follow a lognormal grain size distribution with corresponding normally distributed mineral densities and grain packing densities. To generate virtual samples, random numbers are selected from these distributions. The grain size, mineral density and packing density are chosen independently, assuming zero covariance between their respective (log)normal populations. I think that this is an unjustifiable oversimplification.

In real sedimentary sections, grain size, density and packing density are strongly correlated with each other. For example, Stokes’ Law dictates that small zircon grains ($\rho = 4.65$ g/cm^3) are ‘hydraulically equivalent’ with larger quartz grains ($\rho = 2.65$ g/cm^3). Therefore, well sorted sediment exhibits a *size shift* between quartz and zircon. In a sand that contains both quartz and zircon, the zircon will tend to fill the gaps between the sand grains, thereby increasing the packing density. Although I am not an expert in OSL, I do think that this is important because zircon tends to be rich in actinides, and so the relationship between zircon and quartz affects the dose rate. Wouldn’t this be a more important problem to simulate than the geometry of the sample container?

In summary, hydraulic sorting and selective entrainment impose a strong covariance structure on the physical properties of sediments, which `sandbox` currently does not capture. In principle, it is possible to embed this covariance structure into the `sandbox` package using multivariate (log)normal distributions. However, in practice, this would not be so easy to implement, because it would dramatically increase the number of parameters that need to be set in the rule books. A process-based algorithm would fix this, but it would require a complete redesign of the package. Unfortunately, I can’t think of a third solution.

3 Other comments

1. It is not clear from the title why this paper was submitted to Geochronology. It is only in Section 3.5 (line 335) that the geochronological relevance of the sandbox package becomes apparent. This section uses the method to create some virtual OSL samples. This should be changed, especially because I am doubtful that sandbox will ever be used for any other applications. Suggestion: change the title of the paper to:

   "*sandbox – Creating and Analysing Synthetic OSL samples with R.*"

2. All the random samples in the `sandbox` package are drawn from (log)normal distributions, which are extracted from real datasets using Dietze and Dietze's `EMMAgeo` package. An alternative and more flexible approach would be to draw random numbers from any cumulative distribution. For example:

```
set.seed(1)
phi <- c(1,3,3.5,4,5,5.25,6,8)
cdf <- c(0,0.1,0.3,0.5,0.6,0.8,0.9,1)
fun <- splinefun(x=cdf,y=phi,method="monoH.FC")
nr <- 20
r <- runif(nr)
rgs <- fun(r) # random grain sizes
par(mfrow=c(1,2))
plot(x=phi,y=cdf,type='p')
y <- seq(from=0,to=1,length.out=50)
lines(x=fun(y),y=y)
matlines(x=rbind(rep(0,nr),rgs),y=rbind(r,r),lty=3,col='black')
matlines(x=rbind(rgs,rgs),y=rbind(rep(0,nr),r),lty=3,col='black')
plot(density(rgs),main='')
rug(rgs)
```

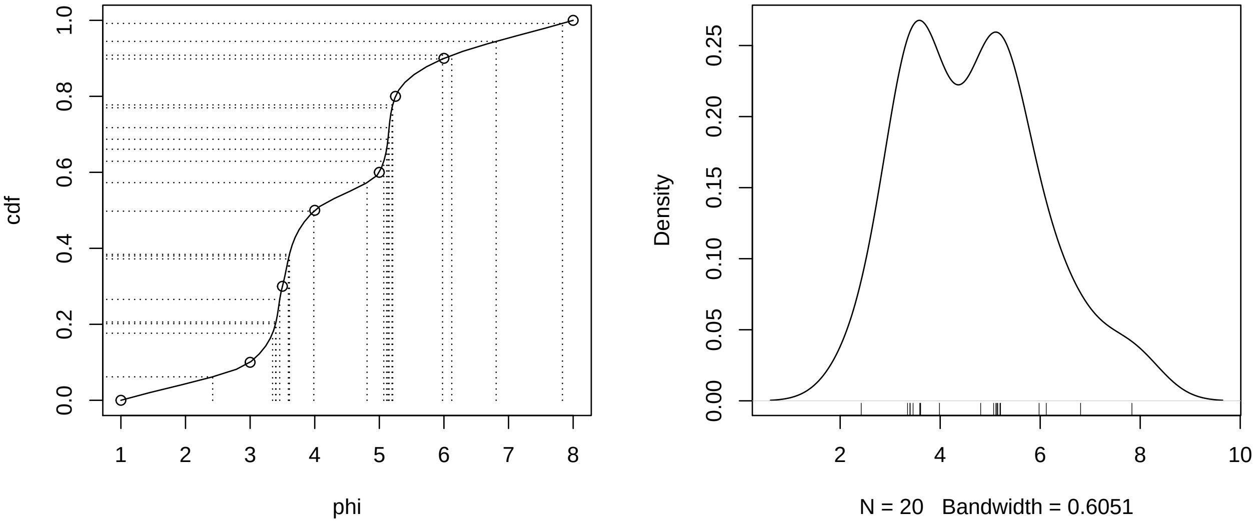


Figure 2: Left: drawing 20 random samples from an arbitrary cumulative grain size distribution (circles mark the anchor points). Right: kernel density estimate and rug plot of the 20 random grain sizes.

3. The paper is too long. `sandbox` is based on an inherently simple idea that I am confident could be explained in a paper half the length of the current manuscript. I enjoyed reading the example code and the supplementary information item. However, I must confess that I found the main text a bit tedious to get through.