

Documentation sandbox

Michael Dietze

January 11, 2016

TO DO LIST

- add function `add_Parameter()` and `remove_Parameter()`
- add function `show_Rule(book, parameter)`, should be plot output

What **sandbox** is

So, **sandbox** is a package for the free statistical software *R*, i.e., a collection of functions along with documentation and example data. Thus, **sandbox** is a flexible tool to virtually “play” with most part of the concept of luminescence dating and beyond. All actions in **sandbox** are probabilistic and rule-based. The tool lets a user create a virtual sediment profile, or rather the definitions of such a profile. Then, these definitions can be used to create a virtual sample, which contains `n` sediment grains, each generated from prior defined probabilistic properties, modified by depth-based rules. Each sediment grain has a distinct combination of parameters. These parameters can describe dimensional characteristics (e.g., mean and standard deviation of the diameter), mineralogic/chemical/physical properties (e.g., mineral type/erodibility, electron trapping efficacy, response intensity to stimulation, anomalous fading trend) and further properties. The virtual sample can then be prepared virtually, split into a series of aliquots and measured virtually to gain virtual luminescence data. This output may later be analysed with the R-package **Luminescence**.

There are a few core concepts, assumptions and constraints that build the foundation of **sandbox**:

- *Population* - **sandbox** is population-based. A population is the most basic, coherent element in the entire model. A population is a set of sediment grains that share a very common characteristic. All grains from one population share the same (range of) properties of certain parameters, such as grain-size, depositional age, mineralogic composition and so on. Individual sediment grains are always sampled from a given population.
- *Parameter* - Parameters are used to describe populations and, hence, sediment grains drawn from these populations. Parameters can be seen as the “horizontal” or “thematic” definition of a virtual sediment deposit. There are two major groups of parameters; general and specific. General parameters describe a sediment regardless of the population of the grains that build this sediment. Examples are water content and dose rate. Specific parameters describe sediment grains with respect to the population to which a grain belongs. Hence, for each population there is another parameter definition. Examples are grain-size, mineralogy and specific density. Furthermore, there are two default superior parameters: age (i.e., the actual true age of a grain with respect to a given depth) and population (i.e., the depth-dependent likelihood for a given grain to originate from one of the provided populations).
- *Rule* - Rules are used to describe how parameters behave or change with depth. Rules can be regarded as “vertical” definition of a sediment deposit. Rules are defined as interpolation functions, based on a discrete number of known (user-defined) parameter-depth-relationships. The default interpolation function is a spline, because it is the most reasonable compromise between flexibility, usability and rationality. A spline can be easily used to describe a constant behaviour, a linear trend and further, depth-variant trends in parameters. Splines are also meaningful smoothing interpolators between “hard” parameter-depth value pairs, a characteristic that can be seen as a pro or a con. It is also possible to define a rule as constant if desired.
- *Rule book* - A rule book is the combination of parameters (“horizontal” definition) and rules (“vertical” definition) to one coherent reference book. There are several predefined and one empty rule book available in **sandbox**. The user can take one of these rule books at any time and modify parameters

(i.e., how a population/grain is defined) and rules (i.e. how the parameter behaves with depth) to then use the modified rule book to generate a virtual sediment sample.

- *Analysis functions* - Once a virtual sediment deposit is defined by a rule book, this deposit can be “harvested” and “treated”. These tasks are performed by the many analysis functions, such as `make_Sieving()` and `make_Aliquot()`. All these `make_`-functions (apart from `make_Sample()`) use information stored for each grain in the parameter list.
- *Probability* - Almost anything in `sandbox` is organised in a probabilistic way. Though this may be regarded as a not really intuitive approach it certainly is the most honest and flexible way to describe the phenomena we explore in nature. There would actually be little use in defining the parameter grain-size as one discrete, fix value since we already consider a grain-size (or equivalent dose) distribution when measuring a sample. Accordingly, the major element in defining parameters is a distribution function, in specific a normal distribution function. There are (or will be) further distribution functions available in `sandbox` but a (log-)normal distribution function is a good start for most sediment properties we encounter in nature.
- *Framework* - `sandbox` is a framework in first line. It is only a skeleton which serves a user to ask and answer questions. The *R*-package `sandbox` provides just the numeric and structural foundations to build virtual sediment deposits, with as much flexibility as possible and as much coherence as required. Currently, the focus of `sandbox` is on luminescence and partly grading into geochemical fingerprinting. But it is possible to add further parameters (and remove existing if these are not needed) to make the framework adequate for the scientific question of interest.

Which questions can be pursued

The *R*-package `sandbox` allows approaching nearly every phenomenon of luminescence analysis and beyond from a numeric, probabilistic perspective. In contrast to other numeric models (e.g., landscape evolution models or geodynamic models) most of the user’s time needs to be sacrificed to build the rule book (or modify one of the existing ones). Once this task is done, the analysis functions can be applied.

There are several overarching directions into which research questions can aim:

- *Explorative modelling* - driven by pure scientific interest. Focus on testing of hypotheses.
- *Explanatory modelling* - focused on retrospective questions in time or interpolation in space and thematic domain. Models are used as tools to explain and interpolate the measured record.
- *Forecast modelling* -
- *Inverse modelling* - finding the set of parameter combinations that best explain my (real) measurement results

A dream of a loess section

Let us start to dream and build an imaginary loess section. It shall be 10 m thick. Its deposition started exactly 25.000 years ago. In total we consider three grain populations are involved in this section: bulk loess, locally reworked material and pedogenic clay.

```
age <- c(25000, 12000, 11000, 10000, 2000, 200, 0) depth.age <- c(10, 8, 7, 4, 0.5, 0.2, 0) plot(age, -depth.age, type = "b")
```

Package contents

Adding further parameters

Available parameters

Available rule types

Available rule books

Modifying a rule book

A gentle start

```
## load library
library("sandbox")

## get empty rule book
book_1 <- get_RuleBook(book = "empty")

## set dose rate parameterisation from default to "rnorm"
book_2 <- set_Parameter(book = book_1,
                        parameter = "dose_rate",
                        type = "rnorm")

## set grainsize parameterisation from default to "constant"
book_3 <- set_Parameter(book = book_1,
                        parameter = "grainsize",
                        type = "constant")

## define true age vector
age <- list(age = c(0, 1000, 2000))

## define corresponding depth vector
depth <- c(0, 5, 10)

## update rule book
book_2 <- set_Rule(book = book_1,
                  name = "age",
                  value = age,
                  depth = depth,
                  plot = TRUE)

## use updated rule book for a simple interpolation
book_2$depth(x = seq(from = 1000, to = 1020))
```

A typical model run

To be added/bugs

- `check_RuleBook()`
- `plot_RuleBook()`
- `get_Sample()`