**Ticktets für Release-Versionen**

* **beta**: Für Veröffentlichung im kleinen Kreis (vor allem Christoph und Sebastian evt. auch Daniel und Maggi). Und für Datenauswertung und Diagramme für das Paper
* **release**: Wenn das Paper accepted ist. Für offiziellen release auf CRAN und luminescence.de
* **post-release**: Weitere Funktionen für später

|  |  |  |  |
| --- | --- | --- | --- |
| **Ticket** | **Zeit-aufwand** | **Fertig für …** | **Prio** |
| Plot\_OSLcurve() neu programmieren   * Neu Strukturieren (3 h) * Journal-Kompatible Diagramme (15 h) * Write whole component term (n\*e^-lambda\*t) into table instead of lambda, n, ... etc. (8 h) * Residual-Kurve unter Plot (3 h) * Sum\_OSLcurve() plot einbauen (5 h) * ~~pseudoLM-OSL Anzeige (3 h)~~ (Zeitaufwand stimmte) | ~~30 h~~  35 h | Beta | Hoch |
| Fit\_OSLcurve() Algrorithmus tauschen   * ~~DEoptim für Startparameter, siehe RLum-Fkt (8 h)~~ (Zeitwand mit LM-Fitting und Refaktorisierung war eher 15 h) * ~~korrekte Chi²-Berechnung mit nls() (5 h)~~ (getestet (2 h) und vorerst verworfen) * Step 1 Sim neu durchführen (7 h) (2h + … | 20 h | Beta | Hoch |
| Fit\_OSLcurve(): Alternativen für F-Test einprogrammieren | 20 h | Post-Release | Mittel |
| Roxygen-Dokumentationen schreiben | 20 h | Release | Hoch |
| Alle Funktionen für Verarbeitung von Data.Frame Listen anpassen | 15 h | Post-Release | Mittel |
| Calc\_OSLintervals() Algrorithmus tauschen   * DEoptim für Minimierungsproblem nutzen (und testen!) (15 h) * In „optimise\_OSLintervals()“ umbenennen (3 h) | 17 h | Release | Mittel |
| Kleinigkeiten   * Check spelling of the step 2 report, especially the Method section (2 h) * Remove "offset" in RLum.OSL\_correction and enable this: background = Raw\_data[11] (2 h) * Entferne alle library(XXXX) Zeilen und änder zu XXXX:: (2 h) | 6 h | Release | Mittel |

**Done:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Checklist Release** | All important features? | Clean code? | Library(XXXX) replaced? | Completed Roxygen? | Working example? |
| fit\_OSLcurve | **x** | **x** | **x** |  | **x** |
| decompose\_OSLcurve |  |  |  |  |  |
| calc\_OSLintervals |  |  |  |  |  |
| simulate\_OSLcurve |  |  |  |  |  |
| sum\_OSLcurves |  |  |  |  |  |
| plot\_OSLcurve |  |  |  |  |  |
| plot\_PhotoCrosssections |  |  |  |  |  |
| RLum.OSL\_correction |  |  |  |  |  |
| RLum.OSL\_global\_fitting | **x** | **x** | **x** |  |  |
| RLum.OSL\_decomposition |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Clean code? | Images and formulas ok? | Spelling checked? |
| report\_Step1 |  |  |  |
| report\_Step1 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | All example data and scripts are there and tested? | Devtools::check() gives zero warnings? | Successfully submitted to CRAN? | Introduction for luminescence.de uploaded? |
| Release <https://cran.r-project.org/submit.html> |  |  |  |  |

Progress bar:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x** | **x** | **x** | **x** | **x** | **X** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Deadline: 2020/11/01**

**Global data format:**

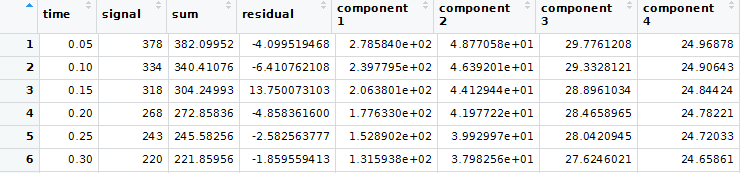
* All functions use one data.frame format and add (or overwrite) columns

test.components <- data.frame(name = c("fast","medium","slow"), lambda = c(1.5,0.5,0.1), n = c(1000,1000,10000))

Data.frame: **Components**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **$name** | **$decay** | **$decay.dev** | **$n** | **$n.error** |  |  |
| 1 | fast | 2 |  | 1000 |  |  |  |
| 2 | medium | 0.5 |  | 1000 |  |  |  |
| 3 | slow | 0.001 |  | 1000 |  |  |  |
| … | background | 0 |  | 10 |  |  |  |

Data.frame **Curve**



**Example 1:**

>

Comp4.A <- data.frame(name = c("component 1","component 2","component 3","component 4"), lambda = c(3,1,0.3,0.05), n = c(2000,1000,2000,10000))

>

Curv4.A <- simulate\_OSLcurve(Comp4.A, channel.width = 0.05, channel.number = 400, simulate.curve = TRUE, add.gaussian.noise = 3, add.background = 5)

>

Comp4.A <- calc\_OSLintervals(Comp4.A, Curv4.A)

Maximum determinant = 0.025196 with interval breaking channels [9, 43, 156] found after 960 iterations