The dataset

The germination of rapeseed (*Brassica napus* L. var. *oleifera*, cv. Excalibur) was tested at thirteen different water potentials (-0.03, -0.15, -0.3, -0.4, -0.5, -0.6, -0.7, -0.8, -0.9, -1, -1.1, -1.2, -1.5 MPa), which were created by using a polyethylene glycol solution (PEG 6000). For each water potential level, three replicated Petri dishes with 50 seeds were incubated at 20°C. Germinated seeds were counted and removed every 2-3 days for 14 days.

The dataset was published by Pace et al. (2012). It is available as `rape` in the `drcSeedGerm` package, which needs to be installed from github (see below). The following code loads the necessary packages, loads the dataset `rape` and shows the first six lines.

```
# library(devtools)
# install_github("OnofriAndreaPG/drcSeedGerm")
library(drc)
library(drcSeedGerm)
library(lmtest)
library(sandwich)
data(rape)
head(rape)
##   Psi Dish timeBef timeAf nSeeds nCum propCum
## 1   0    1       0      3     49   49    0.98
## 2   0    1       3      4      0   49    0.98
## 3   0    1       4      5      0   49    0.98
## 4   0    1       5      7      0   49    0.98
## 5   0    1       7     10      0   49    0.98
## 6   0    1      10     14      0   49    0.98
```

We can see that the data are grouped by assessment interval: ‘timeAf’ represents the moment when germinated seeds were counted, while ’timeBef’ represents the previous inspection time (or the beginning of the assay). The column ’nSeeds’ is the number of seeds that germinated during the time interval between ‘timeBef’ and ‘timeAf. The ’propCum’ column contains the cumulative proportions of germinated seeds and it is not necessary for time-to-event models. The ‘drcSeedGerm’ package contains some service functions which might help prepare the dataset in this form (see the documentation for the functions ‘makeDrm()’ and ‘makeDrm2()’).

Building hydro-time models

Models based on the distribution of germination time

How can we rework Equation 1 to predict the proportion of germinated seeds, as a function of time and water potential? One line of attack follows the proposal we made in a relatively recent paper (Onofri at al., 2018). We started from the idea that the time course of the proportion of germinated seeds (P) is expected to increase over time, according to a S-shaped curve, such as the usual log-logistic cumulative probability function (other cumulative distribution functions can be used; see our original paper):

$$P(t) = \frac{ P_{MAX} }{1 + \exp \left\{ b \left[\log(t) – \log(t_{50}) \right] \right\} } \quad \quad \quad (3)$$

where t_{50} is the median germination time, b is the slope at the inflection point and P_{MAX} is the maximum germinated proportion. Considering that the germination rate is the inverse of germination time, we can write:

$$P(t) = \frac{ P_{MAX} }{1 + \exp \left\{ b \left[\log(t) – \log(1 / GR_{50}) \right] \right\} } \quad \quad \quad (4)$$

where GR_{50} is the median germination rate in the population.

We can now express \(GR_{50}\), \(b\) and \(P_{MAX}\) as linear/nonlinear functions of \(\Psi\) (temperature and other environmental variables can be included as well. See our original paper). In our paper, for \(GR_{50}\), we used the Equation 1 above. For \(P_{MAX}\), we used a shifted exponential distribution, which implies that germination capability is fully determined by the distribution of base water potential within the population and no germinations occur at \(\Psi \leq \Psi_b\):

\[P_{MAX} = h_1(\Psi) = \textrm{min} \left\{ G \, \left[1 – \exp \left(\frac{ \Psi – \Psi_b }{\sigma_{\Psi_b}} \right) \right]; 0 \right\} \quad \quad \quad (5)\]

In the above equation, \(\sigma_{\Psi_b}\) represents the variability of \(\Psi_b\) within the population, which determines the steepness of the increase in \(P_{MAX}\) as \(\Psi\) increases. \(G\) is the germinable fraction, accounting for the fact that \(P_{MAX}\) may not reach 1, regardless of time and water potential.

The parameter \(b\) was assumed to be constant and independent on \(\Psi\). In the end, our hydro-time model is composed by four sub-models:

1. a cumulative probability function (log-logistic, in our example), based on the three parameters \(P_{MAX}\), \(b\) and \(GR50\);
2. a sub-model expressing \(P_{MAX}\) as a function of \(\Psi\);
3. a sub-model expressing \(GR50\) as a function of \(\Psi\);
4. a sub-model expressing \(b\) as a function of \(\Psi\), although, this was indeed a simple identity model \(b(\Psi) = b\).

The equation is:

\[P(t) = \frac{ h_1(\Psi) }{1 + \exp \left\{ b \left[\log(t) – \log(1 / \left[GR_{50}(\Psi) \right]) \right] \right\} } \quad \quad \quad (6)\]

This hydro-time model was implemented in R as the `HTE1()` function, and it is available within the `drcSeedGerm` package, together with the appropriate self-starting routine. It can be fitting by using the `drm()` function in the `drc` package. Please, note that the argument `type` has to be set to “event”.

```
modHTE <- drm(nSeeds ~ timeBef + timeAf + Psi,
                data = rape, fct = HTE1(), type = "event")
summary(modHTE)
##
## Model fitted: Hydro-time model with shifted exponential for Pmax and linear
model for GR50
##
## Parameter estimates:
##
##                          Estimate Std. Error  t-value   p-value
## G:(Intercept)           0.9577943  0.0063663  150.448 < 2.2e-16 ***
## Psib:(Intercept)       -1.0397178  0.0047014 -221.152 < 2.2e-16 ***
## sigmaPsib:(Intercept)  0.1108836  0.0087593   12.659 < 2.2e-16 ***
## thetaH:(Intercept)     0.9060853  0.0301585   30.044 < 2.2e-16 ***
## b:(Intercept)          4.0272972  0.1960877   20.538 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As seeds are clustered in Petri dishes, in order not to violate the independence assumption, it is preferable to get cluster robust standard errors. One possibility is to use the grouped version of the sandwich estimator, as available in the `sandwich package` (Berger, 2017). The function `coeftest` is available in the `lmtest` package (Zeileis, 2002):

```
coeftest(modHTE, vcov = vcovCL, cluster = rape$Dish)
##
## t test of coefficients:
##
##                          Estimate Std. Error   t value  Pr(>|t|)
## G:(Intercept)           0.9577943  0.0080918  118.3661 < 2.2e-16 ***
## Psib:(Intercept)       -1.0397178  0.0047067 -220.9003 < 2.2e-16 ***
## sigmaPsib:(Intercept)  0.1108836  0.0121872    9.0983 < 2.2e-16 ***
## thetaH:(Intercept)     0.9060853  0.0410450   22.0754 < 2.2e-16 ***
## b:(Intercept)          4.0272972  0.1934579   20.8174 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

An alternative way to obtain cluster robust standard errors is to use the delete-a-group jackknife technique, which I described in one of my previous papers (Onofri et al., 2014). This is available in the `jackGroupSE()` function in the `drcSeedGerm` package. It takes quite a bit of computing time, so you may need to be patient, especially if you have a lot of Petri dishes.

```
jack <- jackGroupSE(modHTE, data = rape, cluster = rape$Dish)
```

Once the model is fitted, we may be interested in using the fitted to curve to retrieve some biologically relevant information. For example, it may be interesting to retrieve the germination rates for some selected percentiles (e.g., the 30th, 20th and 10th percentiles). This is possible using the `GRate()` function, that is a wrapper for the original `ED()` function in the package `drc`. It reverses the behavior of the `ED()` function, in the sense that it considers, by default, the percentiles for the whole population, including the ungerminated fraction, which is, in our opinion, the most widespread interpretation of germination rates in seed science. The `GRate()` function works very much like the `ED()` function, although additional variables, such as the selected \(\Psi\) level can be specified by using the argument `x2`.

```
#Naive standard errors
GRate(modHTE, x2 = -1, respLev = c(30, 20, 10))
##           Estimate         SE
## GR:1:30 0.00000000 0.00000000
## GR:1:20 0.03578644 0.01515519
## GR:1:10 0.05129734 0.01546193
```

In this example, we see that the \(GR_{30}\) cannot be calculated, as the germination capacity did not reach 30% at the selected water potential level (\(-1 \,\, MPa\)).

As we said, cluster robust standard errors are recommended. The `GRate()` function allows entering a user-defined variance-covariance matrix, that is obtained by using the `vcovCL()` function in the `sandwich` package. If necessary, germination times can be obtained in a similar way, by using the `GTime()` function.

```
#Cluster robust standard errors
sc <- vcovCL(modHTE, cluster = rape$Dish)
GRate(modHTE, x2 = -1, respLev = c(30, 20, 10), vcov.=sc)
##           Estimate          SE
## GR:1:30 0.00000000 0.000000000
## GR:1:20 0.03578644 0.005452517
## GR:1:10 0.05129734 0.005870701
#Germination times
GTime(modHTE, x2 = -1, respLev = c(30, 20, 10), vcov.=sc)
##         Estimate       SE
## T:1:30       Inf       NA
## T:1:20 27.94355 4.257553
## T:1:10 19.49419 2.231004
```

Last, but not least, we can predict the proportion of germinated seeds at given time and water potential level.

```
predictSG(modHTE, se.fit=T, vcov. = vcovCL,
        newdata = data.frame(Time=c(10, 10, 10),
                             Psi=c(-1.5, -0.75, 0))
        )
##       Prediction          SE
## [1,]  0.0000000 0.00000000
## [2,]  0.8794104 0.03907374
## [3,]  0.9576615 0.01493018
```

Models based on the distribution of \(\Psi_b\)

Another type of hydro-time model was proposed by Bradford (2002) and later extended by Mesgaran et al (2013). This approach starts always from Equation 1; from that equation, considering that the germination time is the inverse of the GR, we can easily get to the following equation:

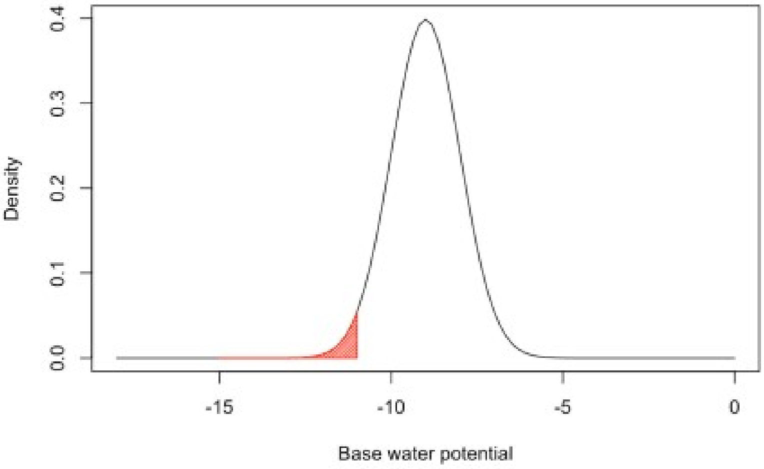
\[\Psi_b = \Psi – \frac{\theta_H}{t} \quad \quad \quad (7)\]

where \(t\) is the germination time. What does this equation tell us? Let’s assume that the hydro-time to germination is 10 \(MPa \, d\) and the environmental water potential is -1 \(MPa\). A single seed germinates in exactly one day, if its base water potential is \(-1 – 10/1 = -11\). If the base water potential is higher, germination will take more than one day; if it is lower, germination will take less than one day. But now, the following questions come: how many seeds in a population will be able to germinate in one day? And in two days? And in \(t\) days?

We know that the seeds within a population do not germinate altogether in the same moment, as they have different individual values of base water potential. If the population is big enough, we can describe the variation of \(\Psi_b\) within the population by using some density function, possibly parameterised by way of a location (\(\mu\)) and a scale (\(\sigma\)) parameter:

\[\Psi_b \sim \phi \left(\frac{\Psi_b – \mu}{\sigma} \right) \quad \quad \quad (8)\]

This is easier to understand if we make a specific example. Let’s assume that the distribution of \(\Psi_b\) values within the population is gaussian, with mean \(\mu = -9\) and standard deviation \(\sigma = 1\). Let’s also assume that the hydro-time parameter (\(\theta_H\)) is constant within the population. We have the situation depicted in the figure below.



The red left tail represents the proportion of seeds that germinate during the first day, as they have base water potentials equal to or lower than -11. By using the gaussian cumulative distribution function we can easily see that that proportion is 0.228:

```
pnorm(-1 - 10/1, mean = -9, sd = 1)
## [1] 0.02275013
```

More generally, we can write:

\[G(t, \Psi) = \Phi \left\{ \frac{\Psi – (\theta_H / t) -\mu }{\sigma} \right\} \quad \quad \quad (9)\]

where \(\Phi\) is the selected cumulative distribution function. The above model returns the proportion of germinated seeds (G), as a function of time and water potential in the substrate. According to Bradford (2002), \(\Phi\) is cumulative gaussian.

Let’s think more deeply about Equation 9 (Bradford, 2002). This function was built to represent the cumulative distribution function of base water potential within the population. However, **it can be as well taken to represent the cumulative distribution function of germination time within the population**. Obviously, while the first distribution is gaussian, the second one is not: indeed, the germination time appears at the denominator of the expression \(\theta_H / t\). It doesn’t matter: every cumulative distribution function for germination time can be fit by using time-to-event methods!

We implemented this model in R as the function `HTnorm()` that is available within the `drcSeedGerm` package and it is meant to be used with the `drm()` function, in the `drc` package.

Mesgaran et al (2013) suggested that the distribution of base water potential within the population may not be gaussian and proposed several alternatives, which we have all implemented within the package. In all, `drcSeedGerm` contains six possible distributions:

1. gaussian distribution (function `HTnorm()`)
2. logistic distribution (function `HTL()`)
3. Gumbel (function `HTG()`)
4. log-logistic (function `HTLL()`)
5. Weibull (Type I) (function `HTW1()`)
6. Weibull (Type II) (function `HTW2()`)

The equations are given at the end of this page; for gaussian, logistic and log-logistic distributions, \(\Psi_{b(50)}\) is the median base water potential within the population. For the gaussian distribution, \(\sigma_{\Psi b}\) corresponds to the standard deviation of \(\Psi_b\) within the population.

Distributions based on logarithms (the log-logistic and all other distributions thereafter) are only defined for positive amounts. On the contrary, we know that base water potential is mostly negative. Therefore, shifted distributions need to be used, by introducing a shifting parameter \(\delta\) which ‘moves’ the distribution to the left, along the x-axis, so that negative values are possible (see Mesgaran et al., 2013).

Let’s fit the above functions to the ‘rape’ dataset. But, before, let me highlight that providing starting values is not necessary, as self-starting routines are already implemented for all models.

```
library(drc)
mod1 <- drm(nSeeds ~ timeBef + timeAf + Psi, data = rape,
            fct = HTnorm(), type = "event")
mod2 <- drm(nSeeds ~ timeBef + timeAf + Psi, data = rape,
            fct = HTL(), type = "event")
mod3 <- drm(nSeeds ~ timeBef + timeAf + Psi, data = rape,
            fct = HTG(), type = "event")
mod4 <- drm(nSeeds ~ timeBef + timeAf + Psi,
            data = rape, fct = HTLL(), type = "event")
mod5 <- drm(nSeeds ~ timeBef + timeAf + Psi,
            data = rape, fct = HTW1(), type = "event")
mod6 <- drm(nSeeds ~ timeBef + timeAf + Psi,
            data = rape, fct = HTW2(), type = "event")
```

What is the best model for this dataset? Let’s use the Akaike’s Information Criterion (AIC) to decide:

```
AIC(mod1, mod2, mod3, mod4, mod5, mod6, modHTE)
##          df      AIC
## mod1    291 3516.914
## mod2    291 3300.824
## mod3    291 3097.775
## mod4    290 2886.609
## mod5    290 2889.307
## mod6    290 2998.915
## modHTE 289 2832.481
```

The first model `modHTE` considers explicitly the distribution of germination times and it is the best fitting of all. The other models consider explicitly the distribution of base water potential, while the distribution of germination times is indirectly included. Among these models, the gaussian is the worse fitting, while the log-logistic is the best one (`mod4`).

For this latter model, we take a look at the value of estimated parameters. Cluster robust standard errors can be obtained as before, by way of the sandwich estimator or a fully iterated delete-a-group jackknife estimator.

```
sand <- coeftest(mod4, vcov = vcovCL, cluster = rape$Dish)
# jack <- jackGroupSE(mod4, data = rape, cluster = rape$Dish)
```

```
sand
## 
## t test of coefficients:
## 
##                      Estimate Std. Error  t value Pr(>|t|)    
## thetaH:(Intercept)  0.677472   0.072902   9.2930  < 2e-16 ***
## delta:(Intercept)   1.136963   0.102440  11.0988  < 2e-16 ***
## Psib50:(Intercept) -0.948101   0.020341 -46.6097  < 2e-16 ***
## sigma:(Intercept)   0.372172   0.173360   2.1468  0.03264 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# jack
```

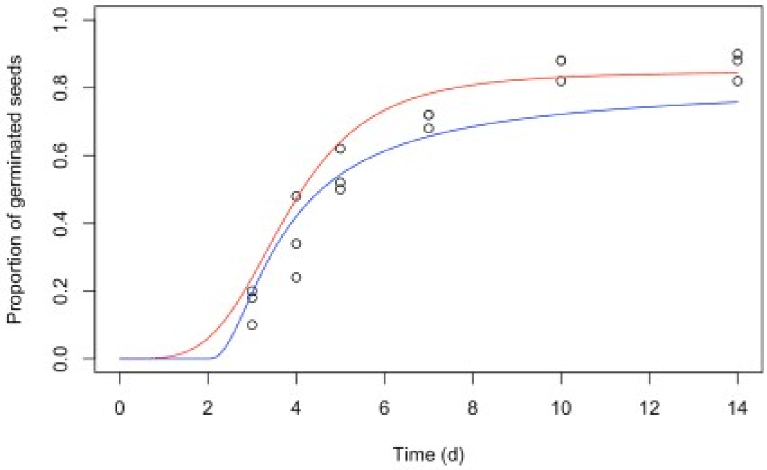
Germination rates and times for a certain percentile (e.g. GR50, GR30), can be obtained by using the `GRate()` and `GTime()` function in `drcSeedGerm`. Again, the use of cluster-robust standard errors is highly recommended.

```
GRate(mod4, respLev=c(30, 50, 70), x2 = 0, vcov. = vcovCL)
##         Estimate        SE
## GR:1:30 1.474866 0.1796681
## GR:1:50 1.399469 0.1648390
## GR:1:70 1.296120 0.1561949
GTime(mod4, respLev=c(30, 50, 70), x2 = 0, vcov. = vcovCL)
##         Estimate         SE
## T:1:30 0.6780277 0.08259731
## T:1:50 0.7145569 0.08416538
## T:1:70 0.7715337 0.09297725
```

We can also make predictions about the germinated proportion for a certain time and water potential level. The code below returns the maximum germinated proportions at -1.5, -0.75, and 0 MPa.

```
predictSG(mod4, se.fit=T, vcov. = vcovCL,
        newdata = data.frame(Time=c(10, 10, 10),
                             Psi=c(-1.5, -0.75, 0))
        )
##        Prediction           SE
## [1,] 6.658526e-15 1.124604e-13
## [2,] 8.038034e-01 5.594629e-02
## [3,] 9.906059e-01 9.465305e-03
```

Let’s use the `predictSG()` function to plot the ‘modHTE’ and ‘mod4’ objects together in the same graph.



Thanks for reading!