

# Release and dispersal of *Colletotrichum gloeosporioides* conidia infecting *Dioscorea alata* yam leaves under simulated rain drops

Sébastien Guyader

04 November 2016

## Intro / argumentaire :

- Connaître les échelles de dispersion des spores (temps et espace) est une étape clé pour la modélisation.
- De nombreux travaux sur la dispersion ont été réalisés pour de multiples pathosystèmes.
- Quelques travaux ont été menés sur *Colletotrichum gloeosporioides* mais sur d'autres hôtes, rien sur l'antracnose de l'igname.

## Matériels et méthodes :

Deux types de manip ont été effectuées à partir de feuilles inoculées (variété Kabusah ou Plimbite, même sensibilité aux souches utilisées) :

### Cinétique d'épuisement de la source :

- Inoculation par goutte de suspension de spores, obtention de nécroses focalisées après incubation
- Les nécroses sont découpées à l'aide d'un emporte pièce
- Les nécroses sont placées sous un générateur de gouttes (mélange eau + bleu coton), les gouttes tombant de 7m de haut avec faible vélocité initiale, essentiellement sous l'influence de la gravité
- Les gouttelettes splashées sont récupérées sur des lames situées de part et d'autre de la nécrose, couvrant une zone entre 2.5 et 10cm (longueur de lame = 7.5 cm)
- Dans un premier temps, on fait tomber les gouttes une à une et on récupère les lames après chaque goutte pour estimer le nombre minimum de gouttes incidentes nécessaires pour mobiliser les premières spores
- Dans un second temps, on fait tomber des séries de 10 gouttes et on récupère les lames ayant reçu le splashing cumulé de 10 gouttes, pour estimer la quantité de spores splashées par chaque lot de 10 gouttes, jusqu'à 100 gouttes
- On réalise 5 runs avec une nouvelle source à chaque fois
- Pour ce dernier cas, afin de comparer les runs entre eux, on calcul la proportion suivante pour chaque run : *nombre de spores comptées dans la série de 10 gouttes / nombre total de spores comptées à la fin du run*
- On analyse les données « proportion(nb de spores) » = f(nombre de gouttes incidentes) par ajustement de courbes, avec les modèles de Fréchet et Weibull
- S'agissant d'une proportion, on utilise la fonction *gnls* pour modéliser la variance en fonction de la valeur prise par variable dépendante (Cf Wedderburn 1974)

### Dipersion horizontale :

- Inoculation de feuilles entières par aspersion d'une suspension de spores
- Les feuilles sont découpées en petits morceaux puis placées dans une boîte de Pétri de petit diamètre (6 cm)
- Juste avant passage sous simulateur de pluie, on ajoute X mL de mélange eau+bleu coton réparti uniformément ; on obtient une concentration finale en spores de 2 à  $6.56 \times 10^6$  spores/mL

- On place la boîte de pétri sous le simulateur de pluie placé à 4 m de haut (à vérifier)
- Autour de la boîte, on place des lames de microscope à 10, 20, 30 et 40 cm de la source (lames perpendiculaires) en deux séries de lames perpendiculaires (direction nord-sud, et direction est-ouest)
- On protège les lames des gouttes incidentes directes à l'aide d'un dispositif type « tente »
- On soumet la source de spores à une pluie simulée de 30 secondes (calculer l'équivalent en mm), en réalisant 10 runs (avec chaque fois une nouvelle source)
- On récupère les lames, et on compte la densité de spores (nombre de spores / cm<sup>2</sup> sur l'intégralité de chaque lame)
- On ajuste les données « densité de spores » = f(distance) au modèle « Power » ou « Négative exponential »

Les données sont analysées avec R 3.3.0

## Résultats

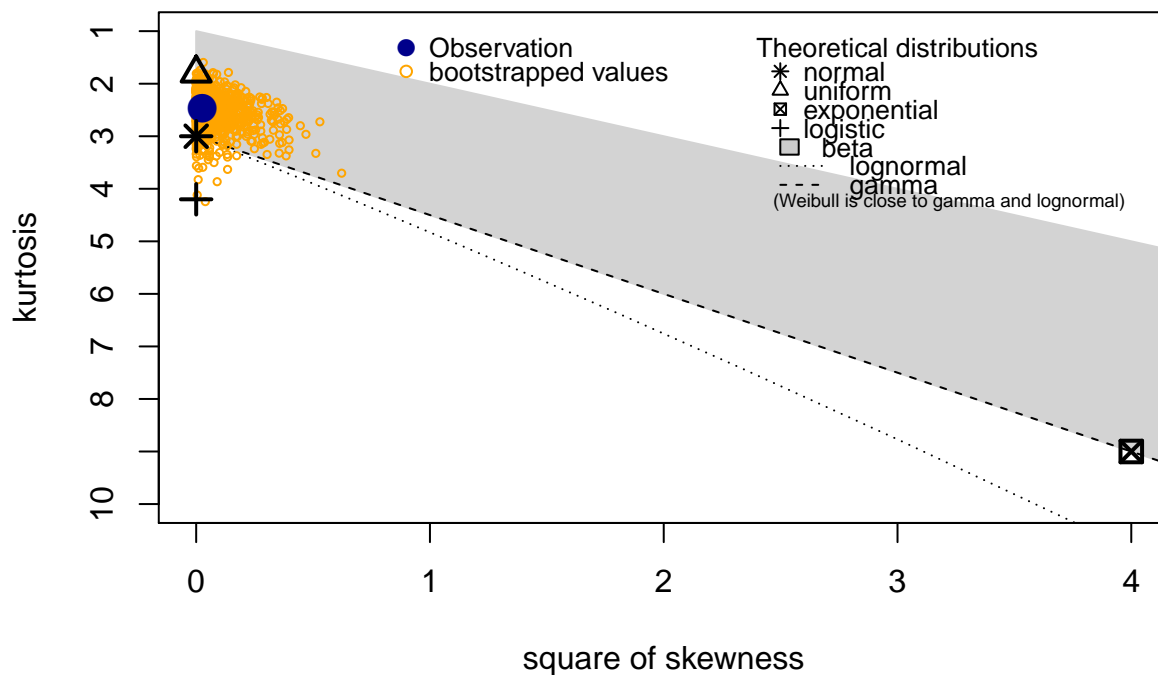
### Epuisement de la source

```
spores <- read.table("./dispersion-cinetique/spores.txt",
                     header=T, sep="\t",
                     col.names=c("cat", "rep", "spores", "total", "ratio"))
spores <- transform(spores, cumul = ave(ratio, rep, FUN = cumsum))
spores$log.ratio <- log(spores$ratio)
```

Analyse sur données  $\ln(\text{ratio})$ , avec  $\text{ratio} = (\text{nb spores observées dans une série de 10 gouttes}) / (\text{nb total de spores comptées sur l'ensemble des séries})$

```
require(fitdistrplus)
descdist(spores$log.ratio, boot=1001)
```

### Cullen and Frey graph



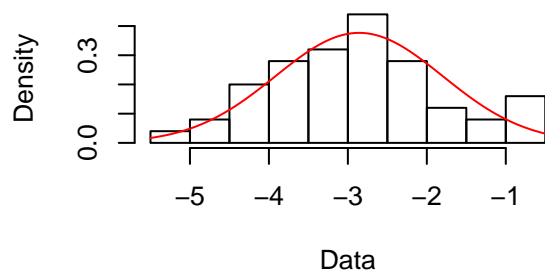
```
## summary statistics
## -----
## min: -5.06043   max: -0.7408301
## median: -2.839652
## mean: -2.85764
## estimated sd: 1.06998
## estimated skewness: 0.1605616
## estimated kurtosis: 2.469896

mle.fit <- fitdist(spores$log.ratio, "norm", method="mle")
summary(mle.fit)

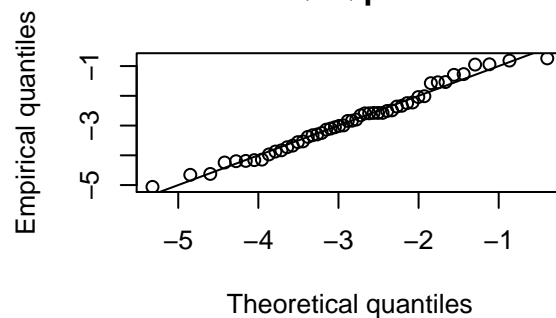
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## mean -2.857640  0.1497973
## sd    1.059227  0.1059222
## Loglikelihood: -73.82387   AIC: 151.6477   BIC: 155.4718
## Correlation matrix:
##      mean sd
## mean   1  0
## sd     0  1

plot(mle.fit)
```

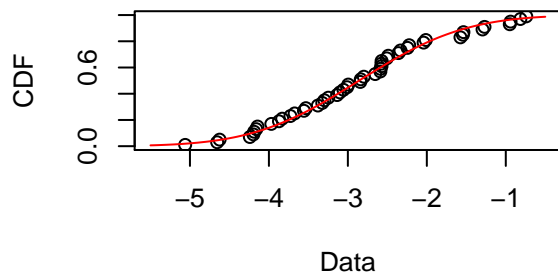
**Empirical and theoretical dens.**



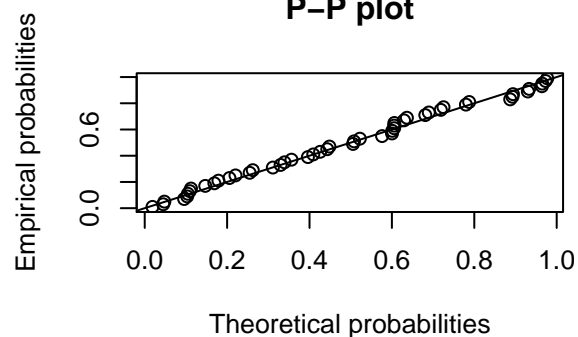
**Q-Q plot**



**Empirical and theoretical CDFs**



**P-P plot**



L'analyse montre que la transformation  $\ln(\text{ratio})$  normalise les données.

**Analyse sur donnée ratio brutes**

Modèle Weibull :

```

require(nlme)
weib.pdf <- function(cat,a,b,e) {
  a*exp(-0.5*((log(cat)-log(abs(e)))/b)^2)
}

# Nonlinear model, no variance modeling
weib <- gnls(ratio~weib.pdf(cat,a,b,e), data=spores,
            start=c(a=0.35, e=15, b=-0.52))
summary(weib)

## Generalized nonlinear least squares fit
## Model: ratio ~ weib.pdf(cat, a, b, e)
## Data: spores
##      AIC      BIC    logLik
## -115.1636 -107.5155 61.58179
##
## Coefficients:
##      Value Std.Error   t-value p-value
## a  0.351716 0.0377048   9.328157      0
## e 14.977919 0.8325888  17.989574      0
## b -0.521604 0.0645027  -8.086539      0
##
## Correlation:
##   a      e
## e 0.156
## b 0.687 0.444
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -2.2468605 -0.1249594  0.2712528  0.6572876  2.4986972
##
## Residual standard error: 0.07282932
## Degrees of freedom: 50 total; 47 residual

# Nonlinear model, with variance modeling
weib2 <- gnls(ratio~weib.pdf(cat,a,b,e), data=spores,
            start=c(a=0.35, e=15, b=-0.52), weights=varFixed(~ratio*(1-ratio)))
summary(weib2)

## Generalized nonlinear least squares fit
## Model: ratio ~ weib.pdf(cat, a, b, e)
## Data: spores
##      AIC      BIC    logLik
## -152.4332 -144.7851 80.21659
##
## Variance function:
## Structure: fixed weights
## Formula: ~ratio * (1 - ratio)
##
## Coefficients:
##      Value Std.Error   t-value p-value
## a  0.228013 0.0314400   7.252346 0e+00
## e 11.941830 2.8749752   4.153716 1e-04
## b -0.780487 0.1431042  -5.453979 0e+00

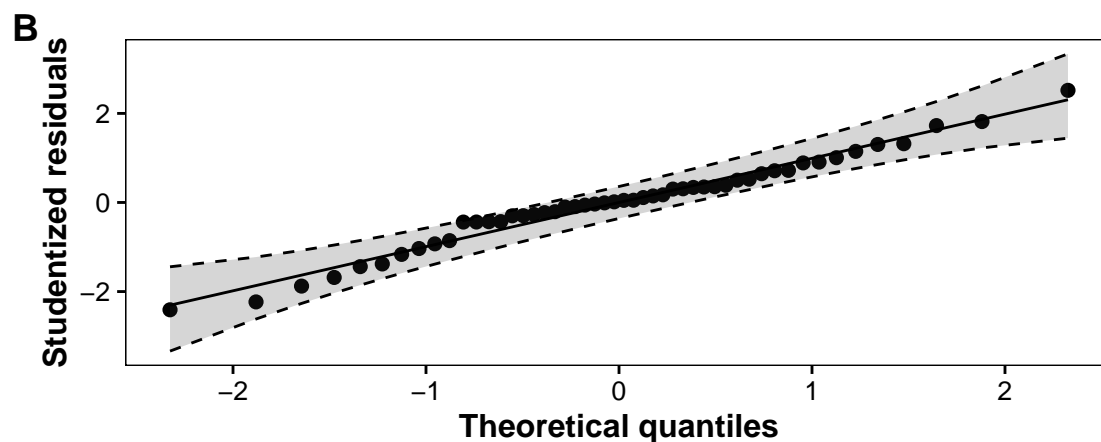
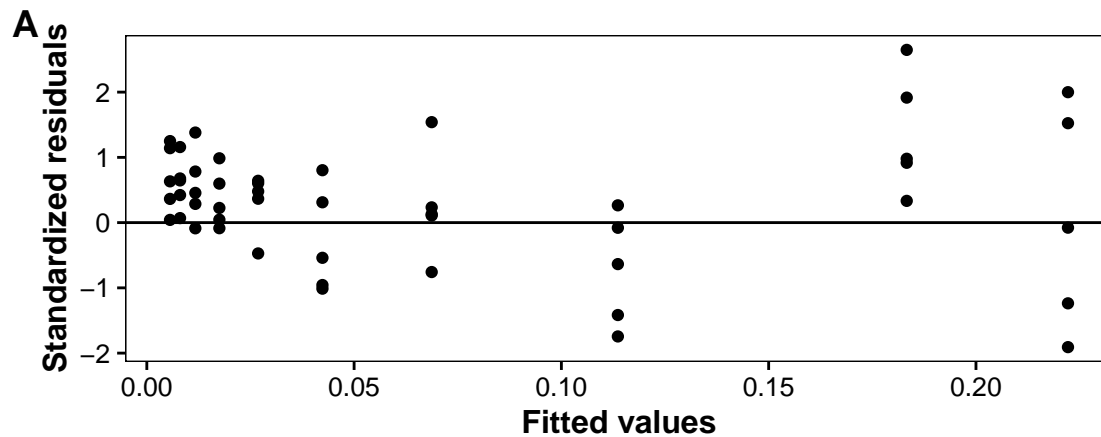
```

```
##
## Correlation:
## a e
## e -0.175
## b 0.104 0.909
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.90828616 -0.07829611 0.34883739 0.79842594 2.64803965
##
## Residual standard error: 0.2218516
## Degrees of freedom: 50 total; 47 residual

# Fitted vs Observed
weib.fit.obs <- as.data.frame(cbind(fitted(weib2), resid(weib2, type="n")), col.names=c("residuals", "fitted values"))
weib.fit.obs <- weib.fit.obs %>%
  aes(x=fitted values, y=residuals) +
  geom_point() +
  geom_hline(yintercept=0) +
  xlab("Fitted values") + ylab("Standardized residuals") +
  my_ggplot_theme()

# QQPlot
weib.qqplot <- my_qqplot(weib2)

plot_grid(weib.fit.obs, weib.qqplot, ncol=1, labels="AUTO")
```



Modèle Fréchet (paramétrage wikipedia) :

```
frech.pdf <- function(cat,A,alpha,beta){
  A*(alpha/beta)*(((cat-4)/beta)^(-1-alpha))*exp(-((cat-4)/beta)^-alpha)
}

# Nonlinear model, no variance modeling
frech <- gnls(ratio~frech.pdf(cat, A, alpha, beta), data=spores,
             start=c(A=9, alpha=1, beta=8.8))
summary(frech)

## Generalized nonlinear least squares fit
## Model: ratio ~ frech.pdf(cat, A, alpha, beta)
## Data: spores
##      AIC      BIC    logLik
## -121.0171 -113.369 64.50855
##
## Coefficients:
##      Value Std.Error  t-value p-value
## A      9.241076 0.9531202 9.695603      0
## alpha  1.504630 0.2337786 6.436129      0
## beta  13.633644 1.4517347 9.391278      0
##
## Correlation:
##      A      alpha
## alpha -0.651
## beta  0.724 -0.943
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -2.3146595 -0.3407387 0.1189612 0.4940661 2.7169715
##
## Residual standard error: 0.06868861
## Degrees of freedom: 50 total; 47 residual

# Nonlinear model, with variance modeling
frech2 <- gnls(ratio~frech.pdf(cat, A, alpha, beta), data=spores,
              start=c(A=9, alpha=1, beta=8.8), weights=varFixed(~ratio*(1-ratio)))
summary(frech2)

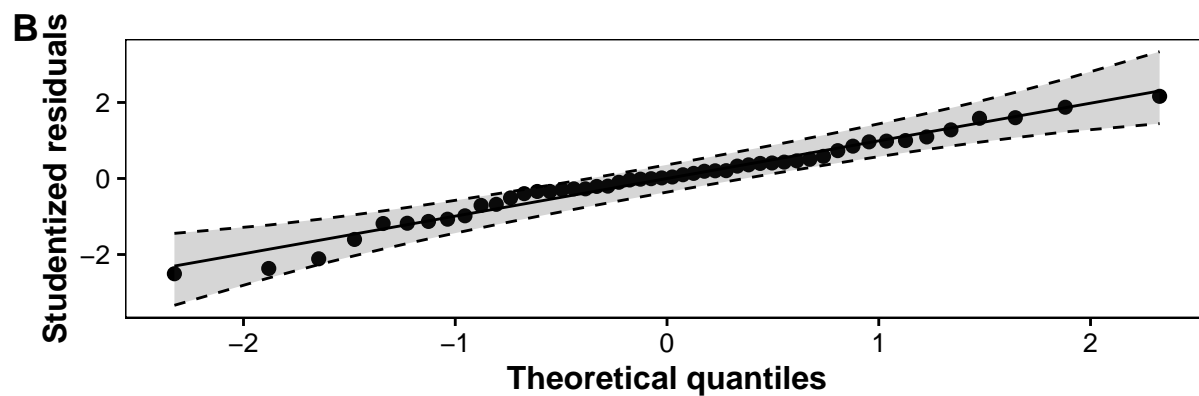
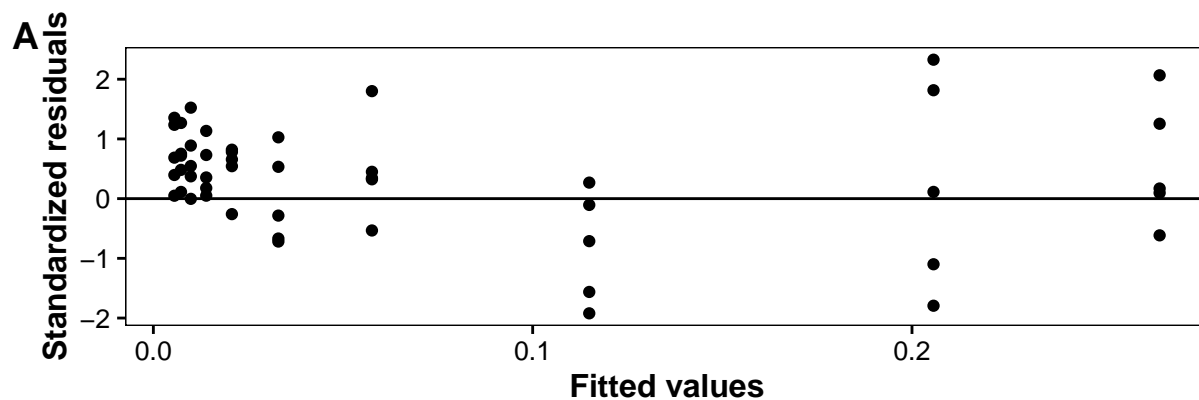
## Generalized nonlinear least squares fit
## Model: ratio ~ frech.pdf(cat, A, alpha, beta)
## Data: spores
##      AIC      BIC    logLik
## -160.3661 -152.718 84.18304
##
## Variance function:
## Structure: fixed weights
## Formula: ~ratio * (1 - ratio)
##
## Coefficients:
##      Value Std.Error  t-value p-value
## A      7.646854 0.7434403 10.285767      0
## alpha  1.559454 0.2170987 7.183155      0
## beta  13.470173 1.1368974 11.848186      0
```

```
##
## Correlation:
##      A      alpha
## alpha -0.025
## beta   0.170 -0.862
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.920703422  0.009827355  0.384969218  0.810536081  2.327212943
##
## Residual standard error: 0.2049323
## Degrees of freedom: 50 total; 47 residual

# Fitted vs Observed
frech.fit.obs <- as.data.frame(cbind(fitted(frech2), resid(frech2, type="n")), col.names=c("residuals",
  aes(x=V1, y=V2)) +
  geom_point() +
  geom_hline(yintercept=0) +
  xlab("Fitted values") + ylab("Standardized residuals") +
  my_ggplot_theme()

# QQPPlot PDF
frech.qqplot <- my_qqplot(frech2)

plot_grid(frech.fit.obs, frech.qqplot, ncol=1, labels="AUTO")
```



Comparaison des deux modèles par critère d'information Bayésien :

```
BIC(weib, frech)
```

```
##      df      BIC
## weib  4 -107.5155
## frech  4 -113.3690
```

Graphique données individuelles + courbes Weibull (trait plein) et Fréchet (pointillé) :

```
weibfun <- function(x) {
  coef(weib2)[1]*exp(-0.5*((log(x)-log(coef(weib2)[2]))/(coef(weib2)[3]))^2)
}
frechfun <- function(x) {
  coef(frech2)[1]*(coef(frech2)[2]/coef(frech2)[3])*
  (((x-4)/coef(frech2)[3])^(-1-coef(frech2)[2]))*
  exp(-((x-4)/coef(frech2)[3])^-coef(frech2)[2])
}

ggplot(data=spores, aes(x=cat, y=ratio)) +
  geom_point() +
  stat_function(fun=weibfun, xlim=c(1,100)) +
  stat_function(fun=frechfun, linetype="dashed", xlim=c(1,100)) +
  xlab("Number of impacting drops") + ylab("Ratio of released spores") +
  guides(alpha=FALSE) +
  my_ggplot_theme()
```

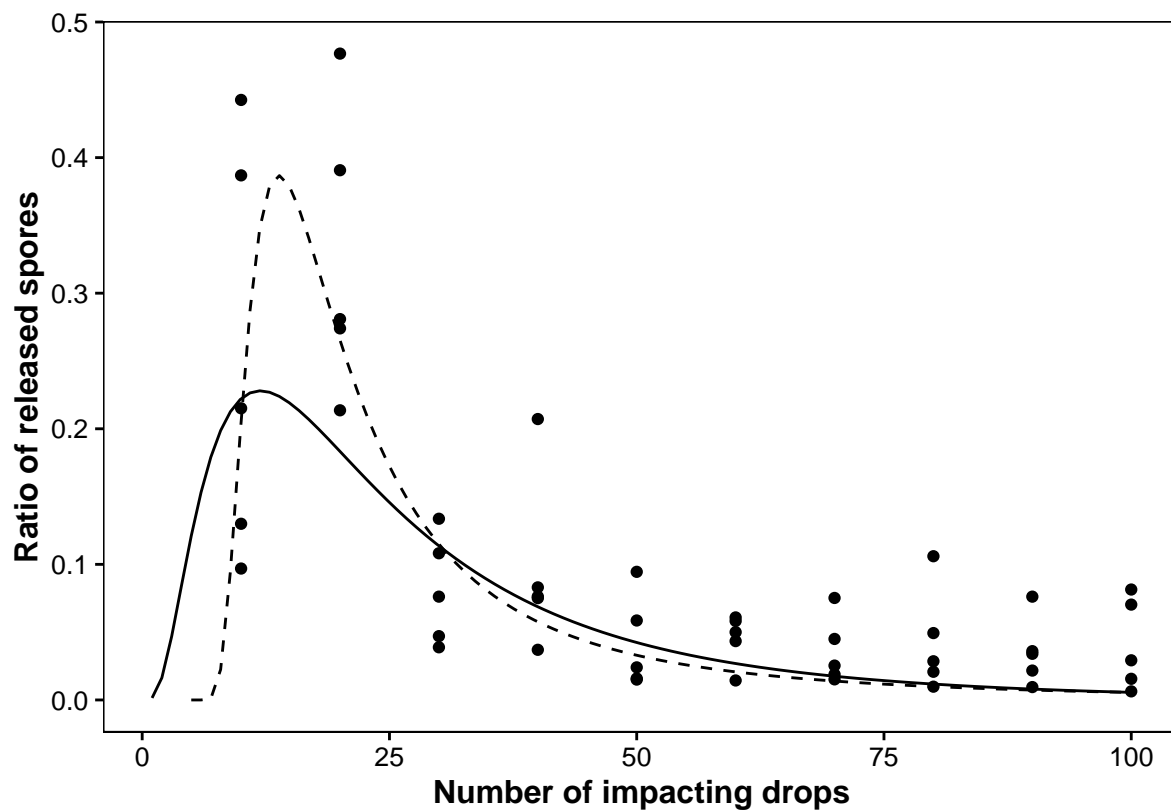


Table des moyennes :

```
moy.ratio <- spores %>% group_by(cat) %>% na.omit() %>% summarise(
  N=n(),
  mean=mean(ratio),
```

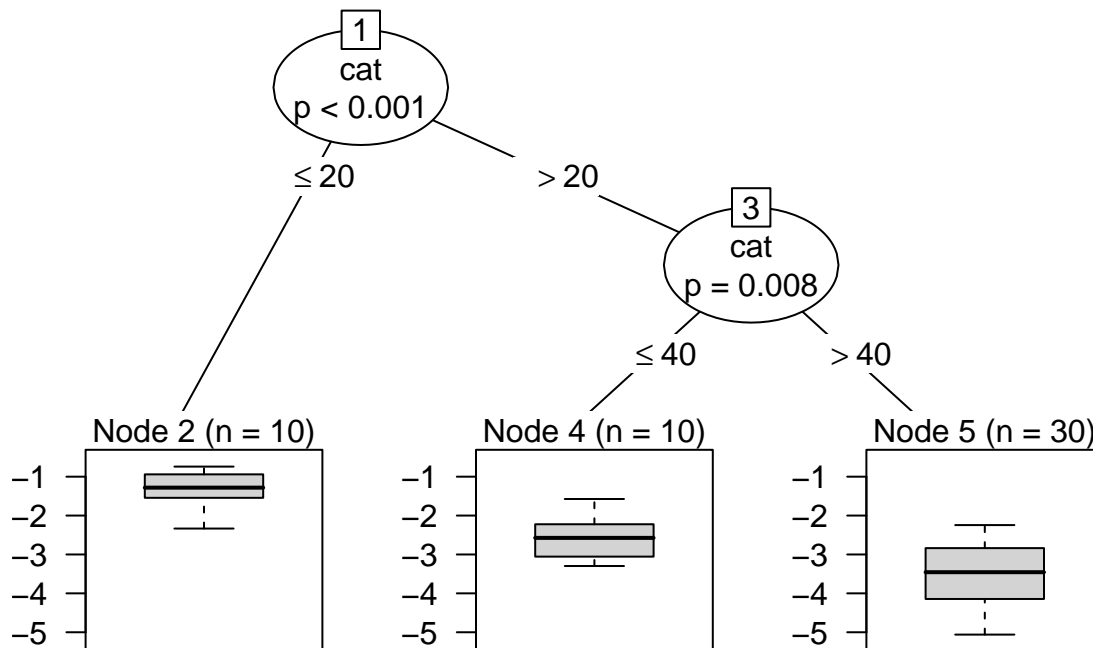


```
sd=sd(ratio),
se=sd/sqrt(N),
ci.student=qt(0.975,df=N-1)*se
)
```

Regression tree :

```
library(partykit)
party.fit <- ctree(log(ratio)~cat, data=spores)
plot(party.fit, main="Conditional Inference Tree for log(ratio)")
```

Conditional Inference Tree for log(ratio)



```
print(party.fit)
```

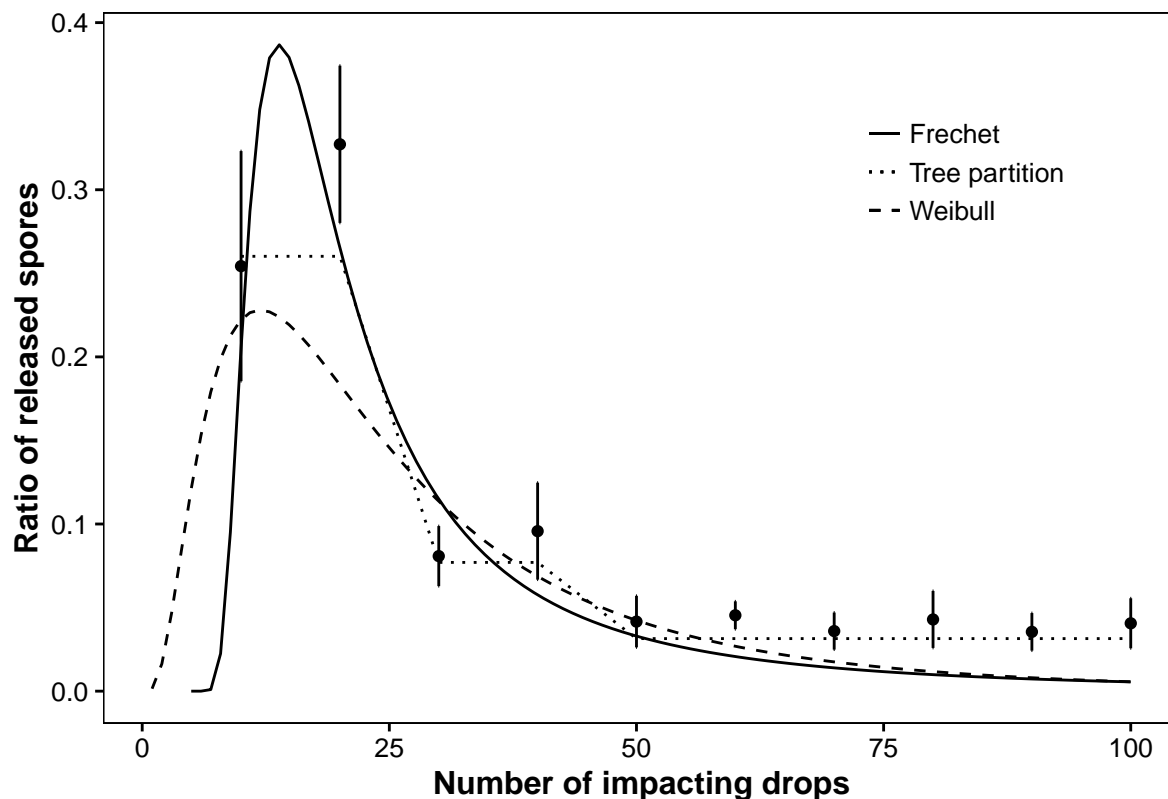
```
##
## Model formula:
## log(ratio) ~ cat
##
## Fitted party:
## [1] root
## |   [2] cat <= 20: -1.346 (n = 10, err = 2.5)
## |   [3] cat > 20
## |   |   [4] cat <= 40: -2.564 (n = 10, err = 2.7)
## |   |   [5] cat > 40: -3.459 (n = 30, err = 16.4)
##
## Number of inner nodes:    2
## Number of terminal nodes: 3
```

Intégration des prédictions du modèle party dans la table de données :

```
spores$log.ratio.party.predict <- predict(party.fit)
spores$ratio.party.predict <- exp(spores$log.ratio.party.predict)
```

Graphe données moyennes +/- SE + courbes :

```
ggplot(data=moy.ratio, aes(x=cat, y=mean)) +
  geom_point() +
  stat_function(fun=weibfun, xlim=c(1,100), aes(colour="Weibull", linetype="Weibull")) +
  stat_function(fun=frechfun, xlim=c(1,100), aes(colour="Frechet", linetype="Frechet")) +
  geom_line(data=spores, aes(x=cat, y=ratio.party.predict, colour="Tree partition", linetype="Tree partit
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se), width=.1) +
  xlab("Number of impacting drops") + ylab("Ratio of released spores") +
  #guides(alpha=FALSE) +
  scale_linetype_manual("", values=c("Weibull"="dashed", "Frechet"="solid", "Tree partition"="dotted")) +
  scale_colour_manual("", values = c("Weibull"="black", "Frechet"="black", "Tree partition"="black"),
    guide=guide_legend(override_aes=list(
      linetype=c("Weibull"="dashed", "Frechet"="solid", "Tree partition"="dotted")))
  ) +
  my_ggplot_theme() +
  theme(legend.position=c(0.8, 0.8))
```



On constate qu'il faut au minimum 3 à 5 gouttes avant d'observer les premières spores splashées. Les 20 gouttes suffisent pour expulser plus de 50% du contenu de la nécrose. Entre 50 et 100 gouttes, le nombre de spores expulsées se stabilise. Il reste donc des spores encore mobilisables au sein de la nécrose au-delà de 100 gouttes. L'équation de Fréchet (trait plein) semble s'ajuster mieux à ces données que l'équation de Weibull (trait pointillé)

## Gradient de dispersion

```
newdata <-
ggplot(data=moy.ratio, aes(x=cat, y=mean, coulour="Data")) +
  geom_point() + geom_line(aes(linetype = "Data"), alpha = 0) +
  stat_function(fun=weibfun, xlim=c(1,100), aes(colour="Weibull", linetype="Weibull")) +
```

```

stat_function(fun=frechfun, xlim=c(1,100), aes(colour="Frechet", linetype="Frechet")) +
geom_line(data=spores, aes(x=cat, y=ratio.party.predict, colour="Tree partition")) +
geom_errorbar(aes(ymin=mean-se, ymax=mean+se), width=.1) +
xlab("Number of impacting drops") + ylab("Ratio of released spores") +
#guides(alpha=FALSE) +
scale_linetype_manual(values=c("Data"="blank", "Weibull"="solid", "Frechet"="dashed", "Tree partition"="dashed")) +
scale_colour_manual("", values = c("Data"="black", "Weibull"="black", "Frechet"="black", "Tree partition"="black")) +
my_ggplot_theme() +
theme(legend.position=c(0.8, 0.8))

```