

Annexes

Annex 1 : Pameseb network

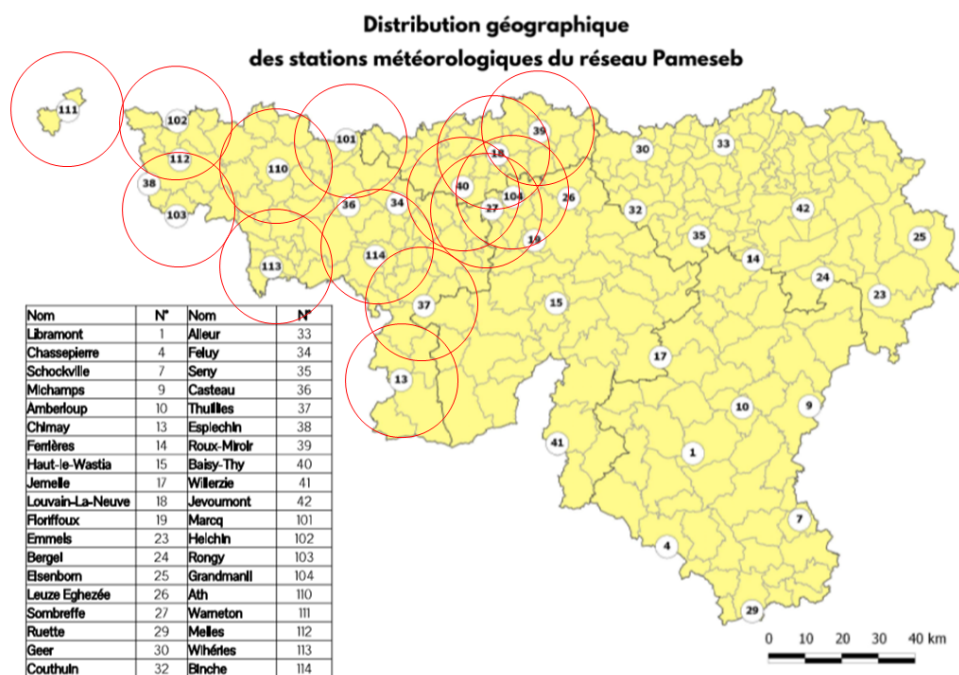


Figure 5.7: Map of the Pameseb meteorological stations network. Source = CRA-W/Pameseb network. Each station in Hainaut and Brabant-Wallon province (main Walloon potato producing regions) is surrounded by a 15 km radius to visualize their range. It can be noted that almost the entire territory of these two Provinces is covered by the radius of a station. Source: CRA-W/Pameseb.

Annex 2 : R code of the nearby-station sensitivity analysis of the Guntz-Divoux model

```

1 ##### R CODE IMPLEMENTING THE SENSITIVITY ANALYSIS OF THE GUNTZ-DIVOUX MODEL WHEN IT IS SUPPLIED BY A NEAR STATION
2 #####
3 #Author: Timothee Clement, Master 2 student in agronomy (UCLouvain) 2018 - 2019
4
5 #Load output from GD model (Run on the Pameseb website, the output is an itx file with infection risk value at each
6   hour)
7 #first station
8 station1 <- read.csv(file='/Users/timoc/Documents/BIR/BIR22/Memoire/donnees/Analyse variabilite modele GD/26_010315_
9   300915_hora_mildiou.itx',header=TRUE,sep=' ',dec='.')
10
11 station1$Datum <- as.POSIXct(station1$Datum, origin = "1903-12-31 23:00") # local
12 head(station1)
13
14 #second station
15 station2 <- read.csv(file='/Users/timoc/Documents/BIR/BIR22/Memoire/donnees/Analyse variabilite modele GD/32_010315_
16   300915_hora_mildiou.itx',header=TRUE,sep=' ',dec='.')
17
18 station2$Datum <- as.POSIXct(station2$Datum, origin = "1903-12-31 23:00") # local
19 head(station2)
20
21 #take the max risk level for each day (between 0 and 4)
22 df_station1 <- data.frame(time=station1$Datum[seq(1,length(station1$Datum),24)])
23 df_station1$ir <- 0
24 for (i in 1:length(df_station1$time)) {
25   df_station1$ir[i] <- max(station1$Infection[(i*24-23):(i*24)])
26 }
27
28 df_station2 <- data.frame(time=station2$Datum[seq(1,length(station2$Datum),24)])
29 df_station2$ir <- 0
30 for (i in 1:length(df_station2$time)) {
31   df_station2$ir[i] <- max(station2$Infection[(i*24-23):(i*24)])
32 }
33
34 diff <- data.frame(time=df_station1$time,ir_station1=df_station1$ir,ir_station2=df_station2$ir) #dataframe that compute
35   the differences between the 2 stations
36 diff$same_IR_value=TRUE
37
38 diff$true_pos <- 0
39 diff$true_neg <- 0
40 diff$false_pos <- 0
41 diff$false_neg <- 0
42
43 for (i in 1:length(df_station1$time))
44 { if (diff$ir_station2[i]!=diff$ir_station1[i]) {diff$same_IR_value[i]=FALSE}
45
46   if (df_station2$ir[i]>0 && df_station1$ir[i]==0) {diff$false_neg[i]=1}
47   if (df_station2$ir[i]==0 && df_station1$ir[i]>0) {diff$false_pos[i]=1}
48   if (df_station2$ir[i]==0 && df_station1$ir[i]==0) {diff$true_neg[i]=1}
49   if (df_station2$ir[i]>0 && df_station1$ir[i]>0) {diff$true_pos[i]=1}
50 }
51
52 print('Mean days predicted with the same maximal infection risk value(%):')
53 round(sum(diff$same_IR_value)/214*100,digits=1)
54
55 print('Mean days predicted with the same decision (presence/absence of ir) (%):')
56 round((sum(diff$true_pos)+sum(diff$true_neg))/214*100,digits=1)
57
58 print('%age of critical days truely predicted:')
59 round(sum(diff$true_pos)/(214-sum(diff$true_neg))*100,digits=1)

```

Annex 3 : R code of the AGROMET uncertainty sensitivity analysis of the Guntz-Divoux model

```

1 ##### R CODE IMPLEMENTING THE SENSITIVITY ANALYSIS OF THE GUNTZ-DIVOUX MODEL WHEN IT IS SUPPLIED BY AGROMET-LIKE DATA (
  WITH UNCERTAINTIES) #####
2
3 #Author: Timothee Clement, Master 2 student in agronomy (UCLouvain) 2018 - 2019
4
5
6 #Clear environment
7 rm(list=ls())
8 #install.packages('gdata') #to be done 1 time
9 library(gdata)
10 perl <- "/Strawberry/perl/bin/perl5.28.0.exe" #path to the perl previously downloaded
11
12 ##### DATA EXTRACTION & PREPROCESSING #####
13
14 #Download txt weather data file
15 df1 <- read.csv(file='/Users/timoc/Documents/BIR/BIR22/Memoire/donnees/Analyse variabilite climatique/34_010315_300915_
  hora.txt',header=TRUE,sep=',',dec='.')
16
17 class(df1$timestamp)
18 head(levels(df1$timestamp))
19 typeof(df1$timestamp[1])
20
21 #transform time data type (from strings into time data)
22 df1$timestamp <- strptime(df1$timestamp,"%d/%m/%Y-%H:%M:%S")
23 class(df1$timestamp)
24 levels(df1$timestamp)
25 typeof(df1$timestamp[1])
26
27 #Run Guntz-Divoux model on real data#
28 ##### GUNTZ-DIVOUX MODEL #####
29
30 #set "wet period"
31 df1$wp <- 0
32 for (i in 5:length(df1$timestamp)) {
33   if (df1$hra[i]>=90) {
34     df1$wp[i] <- df1$wp[i-1]+1
35   }
36   else if (df1$plu[i]>=0.1) {
37     df1$wp[i] <- df1$wp[i-1]+1
38   }
39   else if (df1$hra[i]>=85 && (df1$hra[i-1]>=90 || df1$hra[i-2]>=90 || df1$hra[i-3]>=90)) {
40     df1$wp[i-3] <- df1$wp[i-4]+1
41     df1$wp[i-2] <- df1$wp[i-3]+1
42     df1$wp[i-1] <- df1$wp[i-2]+1
43   }
44 }
45
46
47 #set mean temperature for the wet period
48 df1$mt <- df1$tsa
49 j <- 0
50 for (i in 1:length(df1$timestamp)) {
51   if (df1$wp[i]>0) {
52     j <- j+1
53     df1$mt[i] <- sum(df1$tsa[seq(i-j+1,i)])/j
54   } else {
55     j <- 0
56   }
57 }
58
59 #function that compute the infection risk (between 0 and 4) from Guntz-Divoux regression equation (from table from ppt)
60 f_GDreg2 <- function(wp,t) {
61   if (wp<=10) {
62     ir <- 0
63   }
64   else {
65     ir <- -2.21 + 0.155*wp -0.211*t + 0.013*wp*t + 0.00093*wp^2 + 0.0094*t^2
66   }
67   if (ir<0) {
68     ir <- 0
69   }
70   if (ir>4) {
71     ir <- 4
72   }
73   ir=round(ir)
74   return(ir)
75 }
76
77 #function that compute the hourly incubation unit
78 f_iu <- function(t) {
79   if (t< 7.9 | t>30.1) {
80     iu <- 0
81   }
82   if (t>=7.9 & t<12) {
83     iu <- 0.031
84   }
85   if (t>=12 & t<16.5) {
86     iu <- 0.042
87   }
88   if (t>=16.5 & t<20) {
89     iu <- 0.083
90   }

```

```

91   if (t>=20 & t<=30.1) {
92     iu <- 0.042
93   }
94   return(iu)
95 }
96
97 #Compute the infection risk at each hour
98 df1$ir <- 0
99 for (i in 1:length(df1$timestamp)) {
100   df1$ir[i] <- f_GDreg2(df1$wp[i],df1$mt[i])
101 }
102
103 #take the max risk level for each day (between 0 and 4)
104 df2 <- data.frame(time=df1$timestamp[seq(1,length(df1$timestamp),24)])
105 df2$ir <- 0
106 for (i in 1:length(df2$time)) {
107   df2$ir[i] <- max(df1$ir[(i*24-23):(i*24)])
108 }
109
110 it=100 #iteration number
111 result <- data.frame (iteration = c(1:it))
112
113 for (l in 1:it) {
114
115   #Add random deviation on weather data (representing weather data uncertainty)
116   dev_temp <- rnorm(1,mean=0,sd=0.33) #temperature deviation = +-1 degreeC
117   result$dev_temp [l] <- dev_temp
118
119   dev_rh <- rnorm(1,mean=0,sd=1.67) #relative humidity deviation = +- 5%
120   result$dev_rh [l] <- dev_rh
121
122   dev_rain_POD <- rnorm(1,0.87,sd=0.027) #rainfall probabability deviation
123   dev_rain_FAR <- rnorm(1,0.13,sd=0.013)
124
125   #plot(df1$plu,type="s",col="blue",main="real rain")
126
127   curve(dnorm(x,mean=0.87,sd=0.027), from = 0.5, to = 1)
128
129   dev_df1 <- df1
130   dev_df1$tsta <- df1$tsta + dev_temp #apply temperature deviation
131   dev_df1$hra <- df1$hra + dev_rh #apply RH deviation
132
133   for (n in 1:length(df1$plu)){ #apply rain deviation: % of hours are not correctly predicted
134     binom_false_neg <- rbinom(1,1,1-dev_rain_POD)
135     binom_false_pos <- rbinom(1,1,dev_rain_FAR)
136
137     if (df1$plu[n]>0){if (binom_false_neg==1){dev_df1$plu[n]<-0}}
138     if (df1$plu[n]==0){if (binom_false_pos==1){dev_df1$plu[n] <- 1}}
139     else {paste('error',n)}
140   }
141   #plot(dev_df1$plu[1:500],type="s",col="blue",main="deviated rain")
142
143   #impossible to have relative humidity >100% or rainfall <0mm
144   for (i in 1:length(dev_df1$hra)) {
145     if (dev_df1$hra[i] >100){dev_df1$hra[i]=100}
146     if (dev_df1$plu[i]<0){dev_df1$plu[i]=0}
147
148     #plot (df1$timestamp[1:50], df1$hra[1:50], col='blue', type='l')
149     #lines(df1$timestamp[1:50], dev_df1$hra[1:50], col='red')
150
151     #Run Guntz-Divoux model on modified weather data#
152     ### GUNTZ-DIVOUX MODEL ###
153
154     #set "wet period"
155     dev_df1$wp <- 0
156     for (i in 5:length(dev_df1$timestamp)) {
157       if (dev_df1$hra[i]>=90) {
158         dev_df1$wp[i] <- dev_df1$wp[i-1]+1
159       }
160       else if (dev_df1$plu[i]>=0.1) {
161         dev_df1$wp[i] <- dev_df1$wp[i-1]+1
162       }
163       else if (dev_df1$hra[i]>=85 && (dev_df1$hra[i-1]>=90 || dev_df1$hra[i-2]>=90 || dev_df1$hra[i-3]>=90)) {
164         dev_df1$wp[i-3] <- dev_df1$wp[i-4]+1
165         dev_df1$wp[i-2] <- dev_df1$wp[i-3]+1
166         dev_df1$wp[i-1] <- dev_df1$wp[i-2]+1
167       }
168     }
169
170
171     #set mean temperature for the wet period
172     dev_df1$mt <- dev_df1$tsta
173     j <- 0
174     for (i in 1:length(dev_df1$timestamp)) {
175       if (dev_df1$wp[i]>0) {
176         j <- j+1
177         dev_df1$mt[i] <- sum(dev_df1$tsta[seq(i-j+1,i)])/j
178       } else {
179         j <- 0
180       }
181     }
182
183
184     #Compute the infection risk at each hour
185     dev_df1$ir <- 0
186     for (i in 1:length(dev_df1$timestamp)) {
187       dev_df1$ir[i] <- f_GDreg2(dev_df1$wp[i],dev_df1$mt[i])
188     }
189

```

```

190 | #take the max risk level for each day (between 0 and 4)
191 | dev_df2 <- data.frame(time=dev_df1$timestamp[seq(1,length(dev_df1$timestamp),24)])
192 | dev_df2$ir <- 0
193 | for (i in 1:length(dev_df2$time)) {
194 |   dev_df2$ir[i] <- max(dev_df1$ir[(i*24-23):(i*24)])
195 | }
196 |
197 | diff <- data.frame(time=dev_df2$time,ir_real=df2$ir,ir_dev=dev_df2$ir)
198 | diff$same_IR_value <- TRUE
199 | diff$false_pos <- 0
200 | diff$false_neg <- 0
201 | diff$true_neg <- 0
202 | diff$true_pos <- 0
203 |
204 | for (i in 1:length(dev_df2$time))
205 | { if (diff$ir_real[i]!=diff$ir_dev[i]) {diff$same_IR_value[i]=FALSE}
206 |
207 |   if (diff$ir_real[i]>0 && diff$ir_dev[i]==0) {diff$false_neg[i]=1}
208 |   if (diff$ir_real[i]==0 && diff$ir_dev[i]>0) {diff$false_pos[i]=1}
209 |   if (diff$ir_real[i]==0 && diff$ir_dev[i]==0) {diff$true_neg[i]=1}
210 |   if (diff$ir_real[i]>0 && diff$ir_dev[i]>0) {diff$true_pos[i]=1}
211 | }
212 |
213 | #Result = dataframe who stores the % of days of each metric for the iteration 'l'
214 | result$same_IR_value[l] <- sum(diff$same_IR_value)/214
215 |
216 | result$same_IR_occurrence[l] <- (sum(diff$true_pos)+sum(diff$true_neg))/214
217 |
218 | result$CSI[l] <- sum(diff$true_pos)/(214-sum(diff$true_neg))
219 |
220 | print(l)
221 | }
222 |
223 | print('Mean days predicted with the same maximal infection risk value(%):')
224 | round(sum(result$same_IR_value)/it, digits=3)
225 |
226 | print('Mean days predicted with the same decision (infection (treat) or not):')
227 | round(sum(result$same_IR_occurrence)/it,digits=3)
228 |
229 |
230 | print('%age of critical days truely predicted:')
231 | round(sum(result$CSI)/it,digits=3)

```

Annex 4 : R code of the Guntz-Divoux model

```

1 ##### R CODE IMPLEMENTING THE GUNTZ-DIVOUX FORECAST MODEL FOR THE POTATO LATE BLIGHT #####
2
3 # Author: Timothee Clement, Master 2 student in agronomy (UCLouvain) 2018 - 2019
4
5 #Clear environment
6 rm(list=ls())
7 #install.packages('gdata') #to be done 1 time
8 library(gdata)
9 perl <- "/Strawberry/perl/bin/perl5.28.0.exe" #path to the perl previously downloaded
10
11 #### DATA EXTRACTION & PREPROCESSING ####
12
13 #Download xls weather data file
14 df1 <- read.xls('/Users/timoc/Documents/BIR/BIR22/Memoire
15 /donnees/LLN_010518_121118_hora.xls',sheet=1,perl=perl)
16 head(df1)
17
18 #Download txt weather data file
19 df1 <- read.csv(file='/Users/timoc/Documents/BIR/BIR22/Memoire
20 /donnees/valid_model/01_010417_300917_hora.txt',header=TRUE,sep=',',dec='.')
21
22 class(df1$timestamp)
23 head(levels(df1$timestamp))
24 typeof(df1$timestamp[1])
25
26 #transform time data type (from strings into time data)
27 df1$timestamp <- strptime(df1$timestamp,"%d/%m/%Y-%H:%M:%S")
28 class(df1$timestamp)
29 levels(df1$timestamp)
30 typeof(df1$timestamp[1])
31
32 #### GUNTZ-DIVOUX MODEL ####
33
34 #set "wet period"
35 df1$wp <- 0
36 for (i in 5:length(df1$timestamp)) {
37   if (df1$hra[i]>=90) {df1$wp[i] <- df1$wp[i-1]+1}
38   else if (df1$plu[i]>=0.1) {df1$wp[i] <- df1$wp[i-1]+1}
39   else if (df1$hra[i]>=85 && (df1$hra[i-1]>=90 || df1$hra[i-2]>=90 || df1$hra[i-3]>=90))
40   {
41     df1$wp[i-3] <- df1$wp[i-4]+1
42     df1$wp[i-2] <- df1$wp[i-3]+1
43     df1$wp[i-1] <- df1$wp[i-2]+1
44   }
45 }
46
47 #set mean temperature for the wet period
48 df1$mt <- df1$tsa
49 j <- 0
50 for (i in 1:length(df1$timestamp)) {
51   if (df1$wp[i]>0) {
52     j <- j+1
53     df1$mt[i] <- sum(df1$tsa[seq(i-j+1,i)])/j
54   } else {
55     j <- 0
56   }
57 }
58
59 #function that compute the infection risk (between 0 and 4) from Guntz-Divoux regression equation (from table from
60 Pameseb ppt)
61 f_GDreg2 <- function(wp,t) {
62   if (wp<=10) {
63     ir <- 0
64   }
65   else {
66     ir <- -2.21 + 0.155*wp -0.211*t + 0.013*wp*t + 0.00093*wp^2 + 0.0094*t^2
67   }
68   if (ir<0) {
69     ir <- 0
70   }
71   if (ir>4) {
72     ir <- 4
73   }
74   ir=round(ir)
75   return(ir)
76 }
77
78 #function that compute the hourly incubation unit
79 f_iu <- function(t) {
80   if (t< 7.9 | t>30.1) {
81     iu <- 0
82   }
83   if (t>=7.9 & t<12) {
84     iu <- 0.031
85   }
86   if (t>=12 & t<16.5) {
87     iu <- 0.042
88   }
89   if (t>=16.5 & t<20) {
90     iu <- 0.083
91   }
92   if (t>=20 & t<=30.1) {
93     iu <- 0.042
94   }
95   return(iu)
96 }

```

```

96 |
97 | #Compute the infection risk at each hour
98 | df1$ir <- 0
99 | for (i in 1:length(df1$timestamp)) {df1$ir[i] <- f_GDreg2(df1$wp[i],df1$mt[i])}
100 |
101 | #take the max risk level for each day (between 0 and 4)
102 | df2 <- data.frame(time=df1$timestamp[seq(1,length(df1$timestamp),24)])
103 | df2$ir <- 0
104 | for (i in 1:length(df2$time)) {df2$ir[i] <- max(df1$ir[(i*24-23):(i*24)])}
105 |
106 | #Add incubation periods
107 | j=0
108 | df1[[paste("iu",j,sep="")] <- 1
109 | for (i in 25:length(df1$timestamp)) {
110 |   if (df1$ir[i]>=1 &&
111 |       all(df1[[paste("iu",j,sep="")]][(i-12):(i-1)]!=0))
112 |   { #new incubation cycle: if we have not already launch an incubation cycle in the previous 24 hours
113 |     j <- j+1
114 |     df1[[paste("iu",j,sep="")] <- 0
115 |     df2[[paste("iu",j,sep="")] <- 0
116 |   }
117 |   for (k in 1:j){
118 |     df1[[paste("iu",k,sep="")][i] <- df1[[paste("iu",k,sep="")][i-1] + f_iu(df1$tsa[i])
119 |   }
120 | }
121 |
122 | # Set incubation units to 0 if incubation cycle is over (>7)
123 | l=0
124 | for (i in 1:length(df1$time)) {
125 |   for (k in 1:j) {
126 |     if (df1[[paste("iu",k,sep="")][i]>7) {
127 |       df1[[paste("iu",k,sep="")][i:length(df1[[paste("iu",k,sep="")])]] <- 0
128 |       if (k>1) {
129 |         l=k
130 |         print(paste(l,"eme cycle d'incubation > 7 unites d'incub", "le", df1$time[i]))
131 |       }
132 |     }
133 |   }
134 |   if (df1$ir[i]>0) {
135 |     print(paste("risque d'infection de niveau", df1$ir[i], "le", df1$time[i]))
136 |   }
137 | }
138 |
139 | #### RESULTS VISUALIZATION ####
140 |
141 | #Visualize the infection risk (per day)
142 | plot(df2$time, df2$ir, ylim=c(1,7.5), type="s", col='orange', main="Potato late blight infection risk in Roux-Mirroir in
2017", xlab='date', ylab="risque d'infection")
143 |
144 | #Visualize the infection risk (per hour)
145 | plot(df1$time, df1$ir, ylim=c(0.4,7.5), type="s", col='orange', main="Potato late blight infection risk in LLN in 2017", xlab
='date', ylab='', xaxt='n')
146 | axis.POSIXct(side=1, at=cut(df1$time, "days"), format="%d/%m")
147 |
148 | #Visualize the incubation periods
149 | k=0
150 | for (k in 1:j) {
151 |   lines(df1$time, df1[[paste("iu",k,sep="")]], type='l', col='red', ylab="unit?s d'incubation")
152 | }
153 | legend("topleft", inset=0, legend=c("Infection risk", "Incubation cycles"), col=c("orange", "red"), lty=1.3, cex=1, bty='n
')
```

Annex 5 : R code of the Milsol model

```

1 ##### R CODE IMPLEMENTING THE MILSOL FORECAST MODEL FOR THE POTATO LATE BLIGHT #####
2
3 # Timothee Clement, Master 2 student in agronomy (UCLouvain) 2018 - 2019
4
5
6 #Clear environment and console
7 rm(list=ls())
8 cat("\014")
9 #install.packages('gdata') #to be done 1 time
10 library(gdata)
11 perl <- "/Strawberry/perl/bin/perl5.28.0.exe" #path to the perl previously downloaded
12
13 ### DATA EXTRACTION & PREPROCESSING ###
14
15 ### WEATHER INPUTS ###
16 #Download txt weather data file
17 stockmet <- list()
18 stockmet[[1]] <- read.csv(file='/Users/timoc/Documents/BIR/BIR22/Memoire/donnees/valid_model/01_010410_300910_hora.txt',
19   ,header=TRUE,sep=',',dec='.')
20 stockmet[[2]] <- read.csv(file='/Users/timoc/Documents/BIR/BIR22/Memoire/donnees/valid_model/01_010410_300910_hora_12h.
21   txt',header=TRUE,sep=',',dec='.')
22
23 dfoutput <- data.frame(time=strptime(stockmet[[1]]$timestamp[seq(from=1,to=length(stockmet[[1]]$timestamp),by=24)],"%d/
24   %m/%Y-%H:%M:%S"),logsporulmax=c(0),logspospomax=c(0),gen=c(0),plu=c(0))
25
26 for (it in 1:2) #loop for the 2 time series (because a potential late blight is launched every 12 hours)
27 {
28   print(paste('it=',it))
29   dfmet <- stockmet[[it]]
30   dfoutput[[paste('sporul',it,sep='')] <- c(0)
31   dfoutput[[paste('spospo',it,sep='')] <- c(0)
32   dfoutput[[paste('gen',it,sep='')] <- c(0)
33   gen=0
34
35   dfmet$tsastate=NULL
36   dfmet$plustate=NULL
37   dfmet$hrastate=NULL
38
39   #Initiate variables at time "zero"
40   time0 <- data.frame(timestamp='1/1/0000-00:00:00',tsa=0,hra=0,plu=0)
41   dfmet <- rbind(time0,dfmet)
42
43   class(dfmet$timestamp)
44   head(levels(dfmet$timestamp))
45   typeof(dfmet$timestamp[1])
46
47   #transform time data type (from strings into time data)
48   dfmet$timestamp <- strptime(dfmet$timestamp,"%d/%m/%Y-%H:%M:%S")
49   class(dfmet$timestamp)
50   levels(dfmet$timestamp)
51   typeof(dfmet$timestamp[1])
52
53   ### PARAMETERIZATION ###
54   p1=0.5 #spore development unit coefficient when T<=18 degreeC [/h/degreeC]
55   p2=1 #spore development unit coefficient when T<=18degreeC [/h]
56   p3=10 #spore development unit coefficient when T>18degreeC [/h]
57   p4=0.05 #incubation unit coefficient [/h/degreeC]
58   p5=0.025 #incubation unit coefficient [/h/degreeC^1.5]
59   p6=1.5 #incubation unit coefficient []
60   Topt=18 #Optimal temperature for mycelium growth (incubation) [degreeC]
61   p7=75 #incubation unit coefficient []
62   p8=150 #incubation unit coefficient []
63   p9=225 #incubation unit coefficient []
64   p10=0.00033 #development delay coefficient [/h/degreeC^1.5]
65   p11=0.009 # sporulation development units coefficient [/degreeC^2]
66   p12=2 #sporulation development units coefficient[]
67   p13=1 #sporulation development units coefficient[/h]
68   p14=0.037 #sporulation development units coefficient[/h/degreeC]
69   D1=150 #degree of spore development from which ALL spores have germinated
70   D0=100 #degree of spore development from which germination occurs
71   Dc=100 #degree of development from which the hourly survival rate spore is null if Hr < 90%
72   IO=100 #primary inoculum=number of spores from the first infection (nominal value: 10, uncertainty between 0.0001 and
73     100)
74   Tgel=-2 #temperature at which spores dies because of freeze
75   CUM0=6 #CUMSPO value from which sporulation activity begin []
76   CUM1=10 #CUMSPO from which sporulation activity is at maximum []
77   Tmin=3 #minimum temperature for sporulation [degreeC]
78   FACT=2000 #Max sporulation potential
79
80   stock <- list() #create a list in which we will store the informations (variables) of each cohort
81
82   ### MILSOL MODEL ###
83   beg=1 #beginning of the potential infection
84   count_AFF=0
85
86   for (k in seq(from=beg,to=length(dfmet$timestamp)-23,by=24)) #cohort loop on the whole time serie by day
87   {
88     print(paste('calcul jour',dfmet$timestamp[k+1]))
89
90     stock[[k-1]/24+1]] <- dfmet #create new cohort for this day
91
92     stock[[k-1]/24+1]]$GRAVI=0 #Set all variables to zero for the new cohort
93     stock[[k-1]/24+1]]$TUDESPO=0
94     stock[[k-1]/24+1]]$SURVIE=1
95     stock[[k-1]/24+1]]$CUMDDS=0

```

```

93 stock[(k-1)/24+1]]$SPORES=0
94 stock[(k-1)/24+1]]$SPORES[k]=IO
95 stock[(k-1)/24+1]]$POIDS=0
96 stock[(k-1)/24+1]]$RET=0
97 stock[(k-1)/24+1]]$TINCUB=0
98 stock[(k-1)/24+1]]$AGE=0
99 stock[(k-1)/24+1]]$CUMDDS=0
100 stock[(k-1)/24+1]]$KASPOTHEO=0
101 stock[(k-1)/24+1]]$TRED=0
102 stock[(k-1)/24+1]]$AFFAIB=0
103 stock[(k-1)/24+1]]$KASPOREEL=0
104 stock[(k-1)/24+1]]$POSPO=0
105 stock[(k-1)/24+1]]$SPOSPO=0
106 stock[(k-1)/24+1]]$TUSPORU=0
107 stock[(k-1)/24+1]]$CUMSPO=0
108 stock[(k-1)/24+1]]$ACTISPO=0
109 stock[(k-1)/24+1]]$SPORUL=0
110 stock[(k-1)/24+1]]$SURFMIL=0
111 stock[(k-1)/24+1]]$GEN=0
112
113
114 for (c in 1:length(stock)) #loop on all existing cohorts
115 {
116   for (i in (k+1):(k+24)) #loop on the 24 hours of the day
117   {
118     if (stock[[c]]$CUMDDS[i-1]<D1)
119     {
120       ### COMPONENT1: SPORE SURVIVAL & INFECTION ###
121
122       #UDESPO=development of spores in 1h00 at 21 degreeC (optimal conditions)=variable representing germination
123       #TUDESPO=spores development rate
124       if (stock[[c]]$tsa[i]<18) {stock[[c]]$TUDESPO[i]=p1*stock[[c]]$tsa[i]+p2}
125       else {stock[[c]]$TUDESPO[i]=p3}
126
127       #CUMDDS=stage of spores development
128       stock[[c]]$CUMDDS[i]=sum(stock[[c]]$TUDESPO[1:i])
129
130       #GRAVI=contamination rate=proportion of spores that have germinated
131       if (stock[[c]]$hra[i]>=90) {stock[[c]]$GRAVI[i]=(stock[[c]]$CUMDDS[i]-D0)/(D1-D0)}
132       else {stock[[c]]$GRAVI[i]=0}
133
134
135       #SURVIE=survival rate
136       if (stock[[c]]$hra[i]<=90 && stock[[c]]$CUMDDS[i]>Dc) {stock[[c]]$SURVIE[i]=0}
137       else if (stock[[c]]$hra[i]>90) {stock[[c]]$SURVIE[i]=1}
138       else if (stock[[c]]$hra[i]<=90 && stock[[c]]$CUMDDS[i]<=Dc) {stock[[c]]$SURVIE[i]=1-(stock[[c]]$CUMDDS[i]/Dc)}
139       else{print('ERROR!!')}
140
141       #SPORES=number of potential spores that have survived
142       #SPORES is updated at noon each day
143       if (stock[[c]]$tsa[i]<=Tgel) {stock[[c]]$SPORES[i]=0}
144       else {stock[[c]]$SPORES[i]=stock[[c]]$SPORES[i-1]*stock[[c]]$SURVIE[i]}
145
146       #POIDS=number of contaminating spores
147       stock[[c]]$POIDS[i]=stock[[c]]$GRAVI[i]*stock[[c]]$SPORES[i]
148     }
149     else if (stock[[c]]$CUMDDS[i-1]>=D1)
150     {
151       stock[[c]]$POIDS[i] <- stock[[c]]$POIDS[i-1]
152       stock[[c]]$CUMDDS[i]=stock[[c]]$CUMDDS[i-1]
153       stock[[c]]$SPORES[i]=stock[[c]]$SPORES[i-1]
154       ### COMPONENT 2: INCUBATION & POTENTIAL SPORULATION ###
155
156       #RET= corrective term corresponding to growth (incubation) retardation
157       if (stock[[c]]$tsa[i]<=Topt) {stock[[c]]$RET[i]=0}
158       else {stock[[c]]$RET[i]=p5*(stock[[c]]$tsa[i]-Topt)^p6}
159
160       #TINCUB=Unit for the hourly development of mycelia to cause lesions
161       if (stock[[c]]$tsa[i]<=Topt) {stock[[c]]$TINCUB[i]=p4*stock[[c]]$tsa[i]}
162       else {stock[[c]]$TINCUB[i]=p4*stock[[c]]$tsa[i] - stock[[c]]$RET[i]}
163
164       #AGE=value over time of mycelia development
165       stock[[c]]$AGE[i] <- sum(stock[[c]]$TINCUB[1:i])
166
167       if (stock[[c]]$AGE[i]>p7 && stock[[c]]$AGE[i]<p9)
168       {
169         #store current generation
170         if (stock[[c]]$POIDS[i]>0)
171         {if (stock[[c]]$GEN[i-1]==0) {gen <- gen+1}
172         stock[[c]]$GEN[i:length(stock[[c]]$timestamp)] <- gen}
173       }
174
175       #KASPOTHEO=Sporulation potential of a spore-induced injury as a function of AGE
176       if (stock[[c]]$AGE[i]<p8) {stock[[c]]$KASPOTHEO[i] <- FACT*(stock[[c]]$AGE[i]-p7)/p7}
177       else if (stock[[c]]$AGE[i]<p9) {stock[[c]]$KASPOTHEO[i] <- FACT*(p9-stock[[c]]$AGE[i])/p9}
178       else {stock[[c]]$KASPOTHEO[i] <- 0}
179
180       #TRED=Reducing effect on sporulation potential
181       if (stock[[c]]$tsa[i]<=18) {stock[[c]]$TRED[i]=0}
182       else {stock[[c]]$TRED[i]=FACT*p10*((stock[[c]]$tsa[i]-Topt)^p6)}
183
184       #AFFAIB=Reduction rate of the sporulation potential
185       count_AFF=count_AFF+1
186       if (count_AFF%12!=1)
187       {
188         stock[[c]]$AFFAIB[i]=stock[[c]]$AFFAIB[i-1]+stock[[c]]$TRED[i]
189       }
190     }
191   }
192 }

```

```

190     else
191     {
192         #each 12 hours, AFFAIB is set back to 0
193         stock[[c]]$AFFAIB[i]=stock[[c]]$TRED[i]
194     }
195     #KASPOREEL=real sporulation potential
196     stock[[c]]$KASPOREEL[i]=stock[[c]]$KASPOTHEO[i]-stock[[c]]$AFFAIB[i]
197
198     #POSPD=potential sporulation of lesions caused by all contaminating spores
199     stock[[c]]$POSPD[i]=stock[[c]]$POIDS[i]*stock[[c]]$KASPOREEL[i]
200
201     #SPOSPD=potential for sporulation of lesions (caused by all contaminating spores, as long as conditions are
        favourable)
202     #until they are no longer able to sporulate (lesion necrosis)
203     stock[[c]]$SPOSPD[i]=sum(stock[[c]]$POSPD[1:i])
204
205     ### COMPONENT 3: REAL SPORULATION & DAMAGES ###
206     #TUSPORU=Unit for developing sporulation as a function of hourly temperature
207     if (stock[[c]]$hra[i]<=90) {stock[[c]]$TUSPORU[i]=0}
208     else {stock[[c]]$TUSPORU[i]=(p11*(stock[[c]]$tsa[i]-Tmin)^p12)*(p13-p14*(stock[[c]]$tsa[i]-Tmin))}
209
210     #CUMSPD=Degree of development of sporulation over time
211     stock[[c]]$CUMSPD[i]=sum(stock[[c]]$TUSPORU[1:i])
212
213     #ACTISPD=proportion of spores actually produced in relation to potential sporulation
214     if (stock[[c]]$CUMSPD[i] < CUM0) {stock[[c]]$ACTISPD[i] = 0}
215     else if (stock[[c]]$CUMSPD[i] < CUM1) {stock[[c]]$ACTISPD[i] = (stock[[c]]$CUMSPD[i] - CUM0) / (CUM1 - CUM0)}
216     else {stock[[c]]$ACTISPD[i] = 1.0}
217
218     #SPORUL=number of spores actually produced and ready to be dispersed
219     stock[[c]]$SPORUL[i] <- stock[[c]]$ACTISPD[i]*stock[[c]]$SPOSPD[i]
220
221     #SURFMIL=Conversion of the number of spores into a lesion size. It corresponds to the mildewed area
222     stock[[c]]$SURFMIL[i] <- stock[[c]]$SPORUL[i]/17700
223 }
224 else if (stock[[c]]$AGE[i]>=p9 && stock[[c]]$AGE[i-1]<p9)
225 {
226 }
227 } #end condition else if CUMDDS[i-1]>=D1
228
229 } #end loop on the 24 hours of the day
230 if ((k-1)%24==0)
231 {
232     dfoutput[[paste('sporul',it,sep='')]][[(k-1)/24+1]] <- dfoutput[[paste('sporul',it,sep='')]][[(k-1)/24+1]]+sum(
        stock[[c]]$SPORUL[(i-23):i]) #store number of spores produced on that day
233     dfoutput[[paste('spospo',it,sep='')]][[(k-1)/24+1]] <- dfoutput[[paste('spospo',it,sep='')]][[(k-1)/24+1]]+sum(
        stock[[c]]$SPOSPD[(i-23):i])
234     if (stock[[c]]$GEN[i]>dfoutput[[paste('gen',it,sep='')]][[(k-1)/24+1]])
235     {dfoutput[[paste('gen',it,sep='')]][[(k-1)/24+1]] <- stock[[c]]$GEN[i]} #quelle est la generation la plus
        recente en cours
236 }
237 } #end loop on all existing cohorts
238
239 if (log10(dfoutput[[paste('sporul',it,sep='')]][[(k-1)/24+1]])>=2)
240 {
241     print(paste("seuil de spores drpasse le",dfoutput$time[(k-1)/24+1],'. Log_10(spores produites)=',log10(dfoutput[[
        paste('sporul',it,sep='')]][[(k-1)/24+1]])))
242 }
243
244 } #end loop on the whole time serie
245 } #end loop on the 2 iterations (for the 2 time series)
246
247 #take log of outputs and take max risk of the 2 time series:
248 for (i in 1:length(dfoutput$time))
249 {
250     dfoutput$sporulmax[i] <- max(dfoutput$sporul1[i],dfoutput$sporul2[i])
251     x <- c(dfoutput$gen1[i],dfoutput$gen2[i])
252     ind <- which(x==max(x))
253     dfoutput$gen[i] <- dfoutput[[paste('gen',ind[1],sep='')]][i]
254     dfoutput$spospomax[i] <- max(dfoutput$spospo1[i],dfoutput$spospo2[i],dfoutput$spospo3[i],dfoutput$spospo4[i])
255     dfoutput$plu[i] <- sum(stockmet[[1]]$plu[(i*24)-23):(i*24))
256     if (dfoutput$sporulmax[i]!=0) {dfoutput$logsporulmax[i] <- log10(dfoutput$sporulmax[i])}
257     if (dfoutput$spospomax[i]!=0) {dfoutput$logspospomax[i] <- log10(dfoutput$spospomax[i])}
258 }
259
260
261 ### PLOT RESULTS ###
262
263 plot(dfoutput$time,dfoutput$logsporulmax,xlab='time', ylab='log (Produced spores)',type='h',col='green',title='Resultat
        Milsol en 2010 a Libramont',cex.axis=0.8,ylim=c(0,14),xaxt='n')
264 title(main = 'Milsol output for Libramont, 2016')
265 axis.POSIXct(side=1, at=cut(dfoutput$time, "days"), format="%d/%m")
266 lines(dfoutput$time,dfoutput$logspospomax,col='orange',type='l')
267 abline(h=2,col='red',lty='dotted')
268 legend("topleft",inset=0, legend=c("Produced spores", "Sporulation potential"), col=c("green", "orange"), lty=1.3,cex
        =1,bty='n')

```

Annex 6 : Milsol outputs varying incubation rate ("p4" parameter)

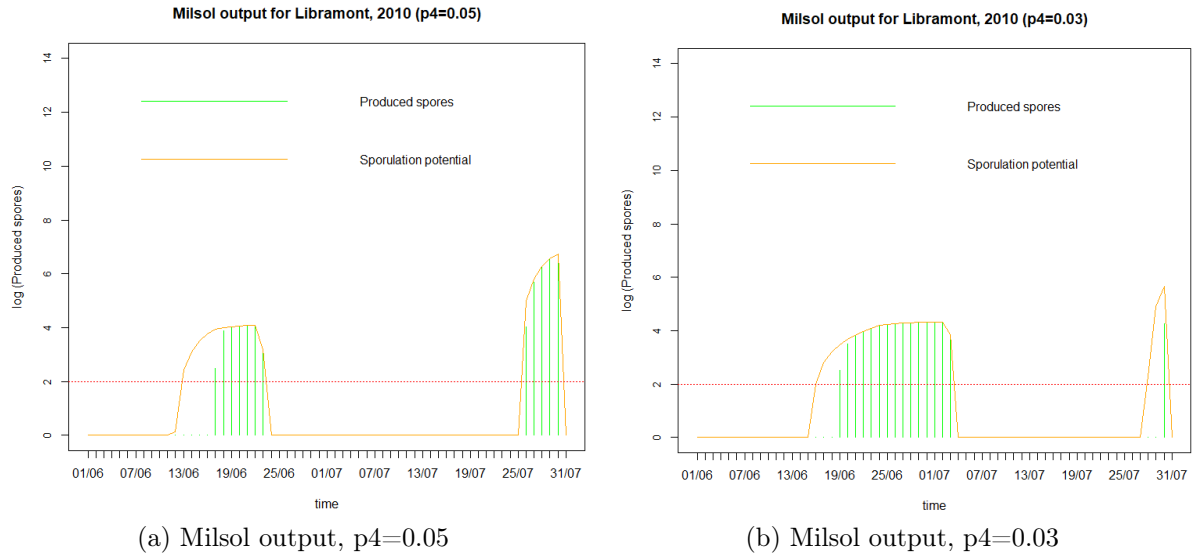


Figure 5.8: Milsol outputs for the months of June and July 2010 in Libramont, for two different values of parameter "p4" (incubation rate). It can be seen that the decrease in incubation rate (parameter p4) delays the beginning of the sporulation period, but also extends it, which is probably not representative of the process related to varietal resistance.

Annex 7 : Milsol outputs varying incubation degree at which sporulation begins ("p7" parameter)

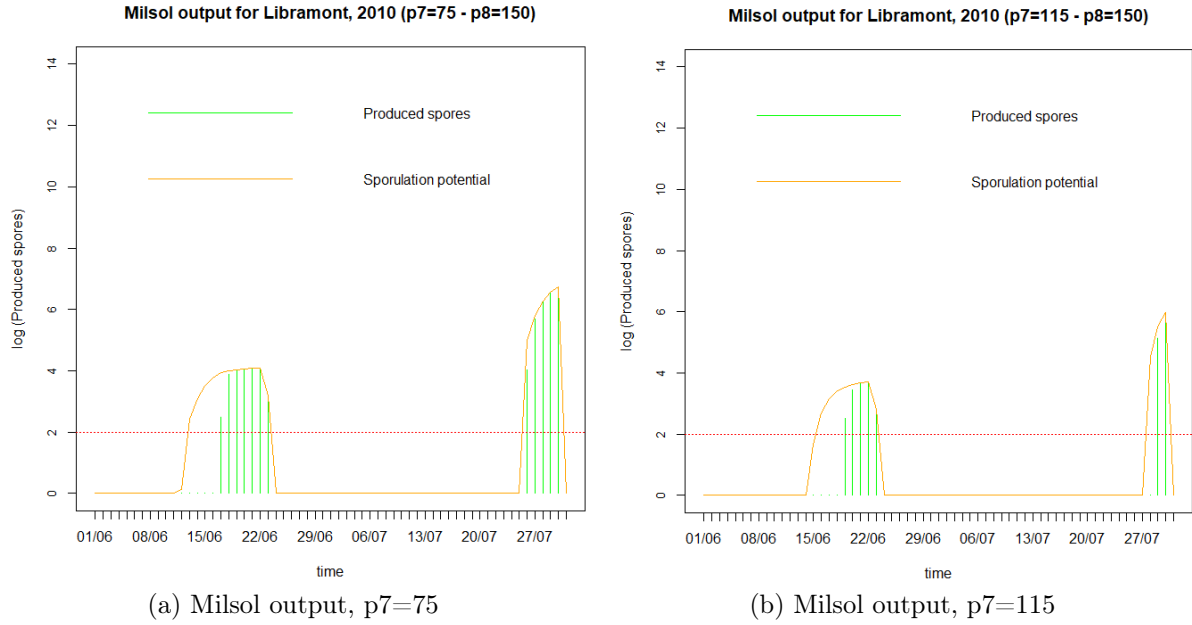


Figure 5.9: Milsol outputs for the months of June and July 2010 in Libramont, for two different values of parameter "p7" (incubation degree at which sporulation begins). It can be seen that the increase in p7 delays the beginning of the sporulation period, but does not extend it.

Annex 8 : R code of the Milsol model calibration to identify the effect of host varietal resistance

```

1 #install.packages('gdata') #to be done 1 time
2 library(gdata)
3 perl <- "/Strawberry/perl/bin/perl5.28.0.exe" #path to the perl previously downloaded
4
5
6 years=c('LIB_2010','LIB_2011','LIB_2012','LIB_2013','LIB_2014','LIB_2015','LIB_2016','LIB_2017','GBX_2016')
7
8 p4=0.05
9 p7_cal=c(75,95,115)
10 p8_cal=c(150,170,190)
11 sporul_tresh_cal=c(2,3,4,5,6,7)
12 gen_tresh_cal=c(2,3,4,5,6,7,8,9,10,11,12,13,14,15)
13
14 iteration=1
15 iteration_total=length(p7_cal)*length(p8_cal)*length(sporul_tresh_cal)*length(gen_tresh_cal) #total parameters
16   combinations
17 dfcalibr <- data.frame(p7=rep(NA,iteration_total),p8=NA,sporul_tresh=NA,gen_tresh=NA,POD_susc=NA,FAR_susc=NA,CSI_susc=
18   NA,POD_mod=NA,FAR_mod=NA,CSI_mod=NA,POD_res=NA,FAR_res=NA,CSI_res=NA)
19
20 a_susc=0
21 b_susc=0
22 c_susc=0
23 a_mod=0
24 b_mod=0
25 c_mod=0
26 a_res=0
27 b_res=0
28 c_res=0
29
30 #COMPUTE ALL MILSOL OUTPUTS (all years + all levels of the parameters p7 and p8)
31 for (p7 in p7_cal)
32 {
33   for (p8 in p8_cal)
34   {
35     for (y in years)
36     { print(paste('p7=',p7,'p8=',p8,'- year = ',y))
37       meteo_0h <- read.csv(file=paste('/Users/timoc/Documents/BIR/BIR22/Memoire/donnees/model_calib/MET_',y,'.txt',sep=
38         ''),header=TRUE,sep=',',dec='.')
39       meteo_12h <- read.csv(file=paste('/Users/timoc/Documents/BIR/BIR22/Memoire/donnees/model_calib/MET_',y,'_12h','.
40         txt',sep=''),header=TRUE,sep=',',dec='.')
41
42       #RUN MILSOL MODEL#
43       assign(paste('dfoutput_',p7,p7,'_p8',p8,'_',y,sep=''),milsol(meteo_0h,meteo_12h,p4,p7,p8))
44       write.csv(get(paste('dfoutput_',p7,p7,'_p8',p8,'_',y,sep='')),paste('dfoutput_',p7,p7,'_p8',p8,'_',y,'.csv',
45         sep=''),row.names = FALSE)
46     }
47   }
48 }
49
50 #COMPUTE, FOR EACH PARAMETERS COMBINATION, THE DIFFERENCES BETWEEN OBSERVED DATA AND MILSOL OUTPUTS
51 for (p7 in p7_cal)
52 {
53   for (p8 in p8_cal)
54   {
55     for (sporul_tresh in sporul_tresh_cal)
56     {
57       for (gen_tresh in gen_tresh_cal)
58       {
59         for (y in years)
60         { #LOAD DATA
61           print(paste('p7=',p7,' - p8=',p8,' - sporul_tresh=',sporul_tresh,' - gen_tresh=',gen_tresh,'- year = ',y))
62           dfoutput <- get(paste('dfoutput_',p7,p7,'_p8',p8,'_',y,sep=''))
63           dfobs <- read.csv(file=paste('/Users/timoc/Documents/BIR/BIR22/Memoire/donnees/model_calib/MILVAR_',y,'.csv',
64             sep=''),header=TRUE,sep=',',dec='.')
65
66           #COMPUTE CONTINGENCY SCORES FOR THE YEAR SIMULATED EACH RESISTANCE LEVEL
67           for (RL in c('susc','mod','res'))
68           {
69             for (i in 1:length(dfobs$time))
70             { if (is.na(dfobs[[paste('sporul_',RL,sep='')]][i])) {}
71               else if (dfobs[[paste('sporul_',RL,sep='')]][i]==0 && dfoutput$logsporulmax[i]>=sporul_tresh && dfoutput$
72                 gen[i]>=gen_tresh) {assign(paste('b_',RL,sep=''),get(paste0('b_',RL,sep=''))+1)} #false positive
73             }
74             else if (dfobs[[paste('sporul_',RL,sep='')]][i]==0 && (dfoutput$logsporulmax[i]<sporul_tresh || dfoutput$
75                 gen[i]<gen_tresh)) {} #true negative
76             else if (dfobs[[paste('sporul_',RL,sep='')]][i]==1 && dfoutput$logsporulmax[i]>=sporul_tresh && dfoutput$
77                 gen[i]>=gen_tresh) {assign(paste('a_',RL,sep=''),get(paste0('a_',RL,sep=''))+1)} #true positive
78             else if (dfobs[[paste('sporul_',RL,sep='')]][i]==1 && (dfoutput$logsporulmax[i]<sporul_tresh || dfoutput$
79                 gen[i]<gen_tresh)) {assign(paste('c_',RL,sep=''),get(paste0('c_',RL,sep=''))+1)} #false negative
80             else {print(paste('error at line',i))}
81           }
82         }
83       }
84     }
85   }
86 }
87
88 #STORE FINAL CONTINGENCY SCORE FOR ALL YEARS, FOR THE PARAMETERS COMBINATION AND FOR THE 3 HOST RESISTANCE
89   LEVELS
90 dfcalibr$p7[iteration] <- p7
91 dfcalibr$p8[iteration] <- p8
92 dfcalibr$sporul_tresh[iteration] <- sporul_tresh
93 dfcalibr$gen_tresh[iteration] <- gen_tresh

```

```

82   for (RL in c('susc','mod','res'))
83   {
84     a=get(paste0('a_',RL,sep=''))
85     b=get(paste0('b_',RL,sep=''))
86     c=get(paste0('c_',RL,sep=''))
87     dfcalibr[[paste('POD_',RL,sep='')]][iteration] <- a/(a+c)
88     dfcalibr[[paste('FAR_',RL,sep='')]][iteration] <- b/(a+b)
89     dfcalibr[[paste('CSI_',RL,sep='')]][iteration] <- a/(a+b+c)
90   }
91
92   a_susc=0
93   b_susc=0
94   c_susc=0
95   a_mod=0
96   b_mod=0
97   c_mod=0
98   a_res=0
99   b_res=0
100  c_res=0
101
102  iteration <- iteration+1
103
104  }
105 }
106 }#end loop on parameters
107 }
108
109 write.csv(dfcalibr, "dfcalibr.csv",row.names = FALSE) #save file
110
111 dfcalibr <- read.csv(file=paste('/Users/timoc/Documents/BIR/BIR22/Memoire/donnees/model_calib/dfcalibr_p7_p8.csv',sep=
112   ''),header=TRUE,sep=',',dec='.')
113
114 #FIND BEST SET OF PARAMETERS FOR SUSCEPTIBLE VARIETY
115 max(dfcalibr$POD_susc)
116 which(dfcalibr$POD_susc==max(dfcalibr$POD_susc))
117
118 min(dfcalibr$FAR_susc)
119 which(dfcalibr$FAR_susc==min(dfcalibr$FAR_susc))
120
121 max(dfcalibr$CSI_susc)
122 which(dfcalibr$CSI_susc==max(dfcalibr$CSI_susc))
123
124
125 #FIND BEST SET OF PARAMETERS FOR MODERATELY SUSCEPTIBLE VARIETY
126 max(dfcalibr$POD_mod)
127 which(dfcalibr$POD_mod==max(dfcalibr$POD_mod))
128
129 min(dfcalibr$FAR_mod)
130 which(dfcalibr$FAR_mod==min(dfcalibr$FAR_mod))
131
132 max(dfcalibr$CSI_mod)
133 which(dfcalibr$CSI_mod==max(dfcalibr$CSI_mod))
134
135
136 #FIND BEST SET OF PARAMETERS FOR RESISTANT VARIETY
137 max(dfcalibr$POD_res)
138 which(dfcalibr$POD_res==max(dfcalibr$POD_res))
139
140 min(dfcalibr$FAR_res)
141 which(dfcalibr$FAR_res==min(dfcalibr$FAR_res))
142
143 max(dfcalibr$CSI_res)
144 which(dfcalibr$CSI_res==max(dfcalibr$CSI_res))

```

Annex 9 : Results of MILVar trials at Libramont in 2016

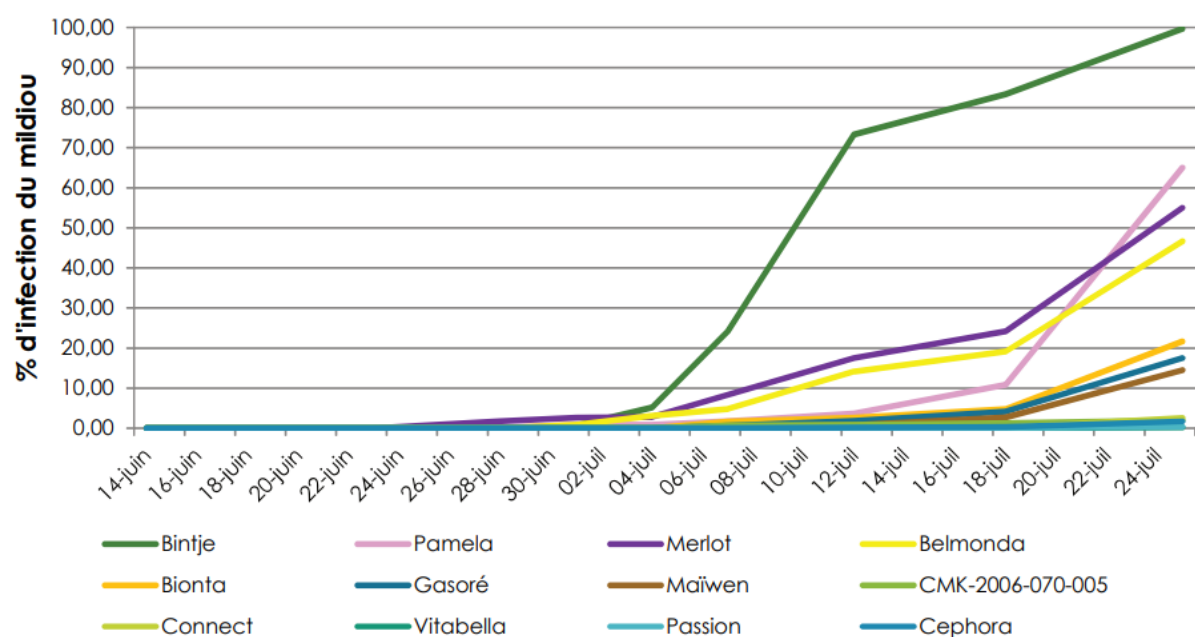


Figure 5.10: Dynamics of foliage infection by *Phytophthora infestans* (late blight) for some varieties of the MilVar trial in Libramont in 2016 [20].