20240424\_qmra\_hpp\_pef\_v01

# **The effect of non-thermal processing on quantitative microbial risk assessment (QMRA)**

# 1. Statement of purpose

The purpose of this QMRA is to estimate the impact of replacing conventional pasteurization processing with Pulsed Electric Fields or High Pressure processing on the annual cases of illness caused by *Escherichia coli* after the consumption of fruit juice.

# 2. A case study for juice treatment with HPP - Model definition

## 2.1 Initial concentration

We will assume that the microorganism of interest is *Escherichia coli.* We will define our initial concentration as the concentration of *E. coli* in the fruit juice tank after the juice is extracted from the fruit. We assume that this concentration can be described by a uniform distribution with minimum and maximum with equal probabilities to be 2 or 3 log10 CFU/ml, respectively. It has some variability which is defined in level 0.

logN0 <- Uniform$new("logN0", # A uniform distribution  
 level = 0)$  
 map\_input("min",  
 Constant$new("logN0\_min", 2) # with a constant min. value  
)$  
 map\_input ("max",  
 Constant$new("logN0\_max", 3) # and a constant max. value  
)

## 2.2 Inactivation using HPP processing

Moving to the next step of the microbial inactivation, we need to define our inactivation model, which in our case is a log-linear primary inactivation model:

We also want to define a secondary model to also integrate the effect of pressure level on the inactivation. For that we will use our log-linear model for fruit juices and *E. coli*, as defined from our meta-analysis:

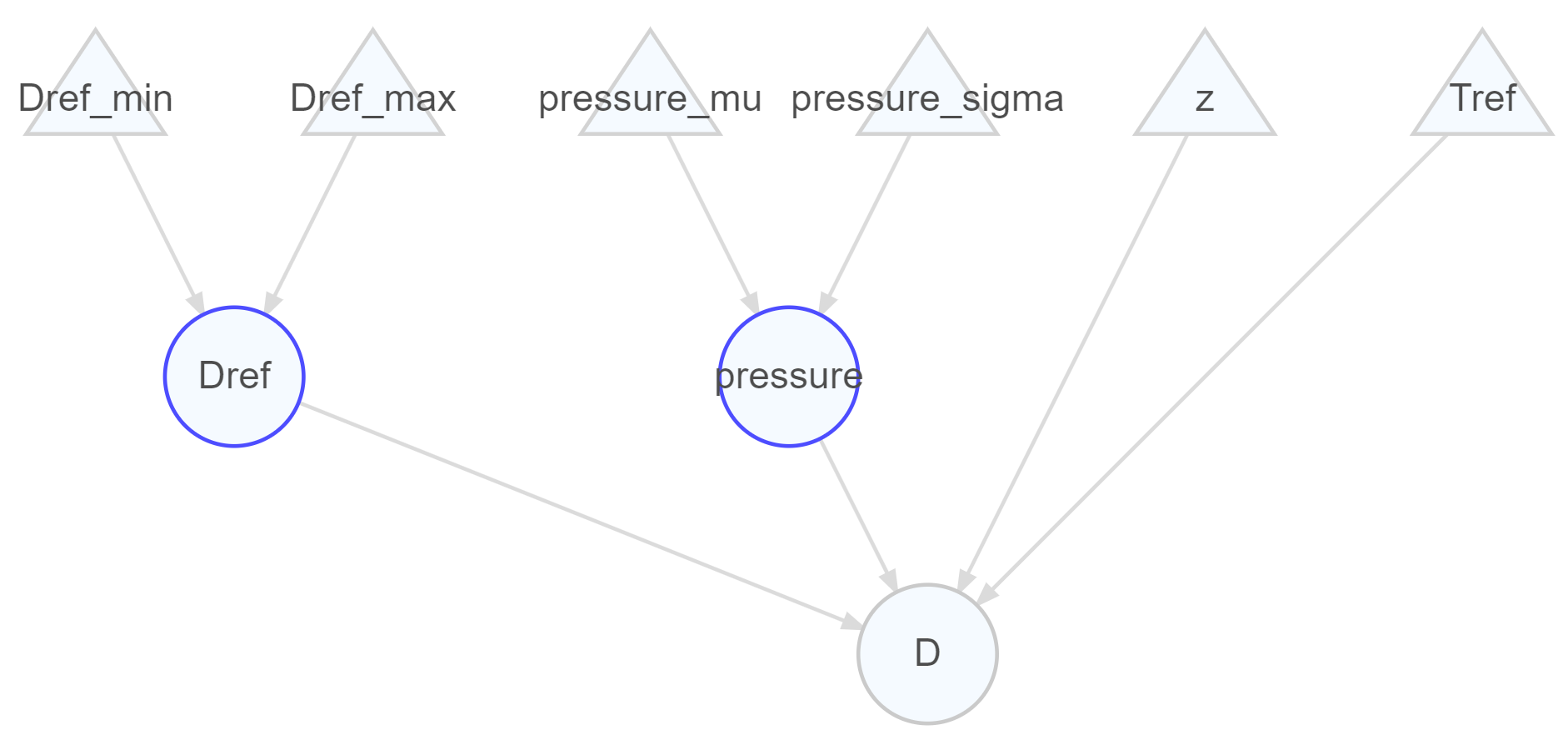
Then we need to integrate these parameters to the conventional expression of the secondary model with respect to the reference values (same equation):

The *zp* estimated from the negative inverse of the slope was 304 MPa. For the Pref we will use 400 Mpa. The treatment pressure *P* is described by a normal distribution with mean 600 MPa and standard deviation 10 MPa (variability i.e., level = 0). Dref (at the Pref) was estimated as 1.3 minutes but we assume that is described by a uniform distribution with a minimum of 1 minute and maximum of 4 minutes (variability i.e., level = 0).

Let’s implement those changes:

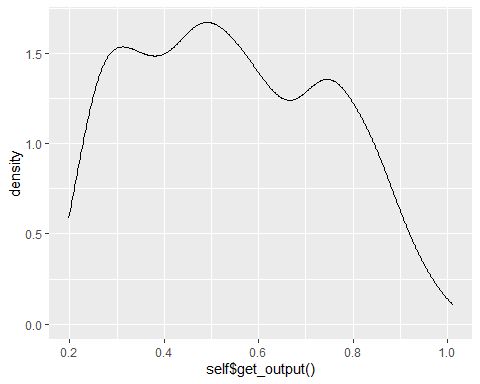
## Secondary model  
  
pressure <- Normal$new("pressure", level = 0)$ # Normal pressure level with some variability assigned  
 map\_input("mu", Constant$new("pressure\_mu", 600))$  
 map\_input("sigma", Constant$new("pressure\_sigma", 10))  
  
Dref <- Uniform$new("Dref", level = 0)$ # Variability in Dref  
 map\_input("min", Constant$new("Dref\_min", 1))$  
 map\_input("max", Constant$new("Dref\_max", 4))  
  
sec\_model <- Dz\_model$new("D")$  
 map\_input("Dref", Dref)$  
 map\_input("temperature", pressure)$  
 map\_input("z", Constant$new("z", 304))$  
 map\_input("Tref", Constant$new("Tref", 400))

The scheme of the secondary inactivation model as we defined it:



Let’s visualize what we implemented through a density plot after performing 1000 simulations:

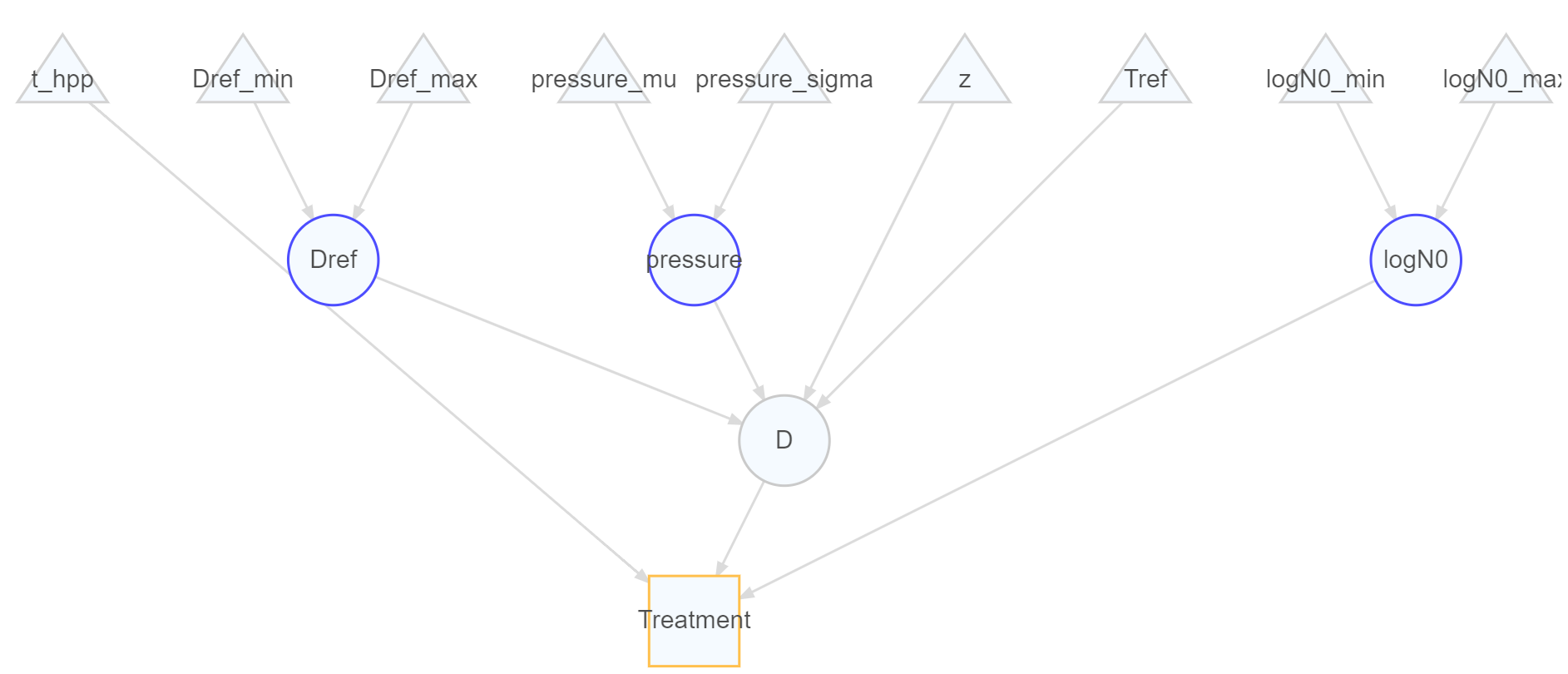
sec\_model$simulate(1000)  
sec\_model$density\_plot()



Regarding treatment, we assume that the treatment time is 5 minutes with no variation. Now let’s plug this to our primary model log-linear primary inactivation model ([Equation 1](#eq-primary)):

t\_hpp <- Constant$new("t\_hpp", 5) # the treatment time (minutes)  
  
inactivation <- LogLinInactivation$new("Treatment")$  
 map\_input("logN0", logN0)$ #We map logN0 to the initial count defined before  
 map\_input("D", sec\_model)$  
 map\_input("t", t\_hpp)

The scheme of the full inactivation model as we defined it:



## 2.3 Growth during storage

In this section we will describe the microbial growth during storage using the exponential growth model with stationary phase (bilinear model):

with the growth rate given by the Ratkowsky model:

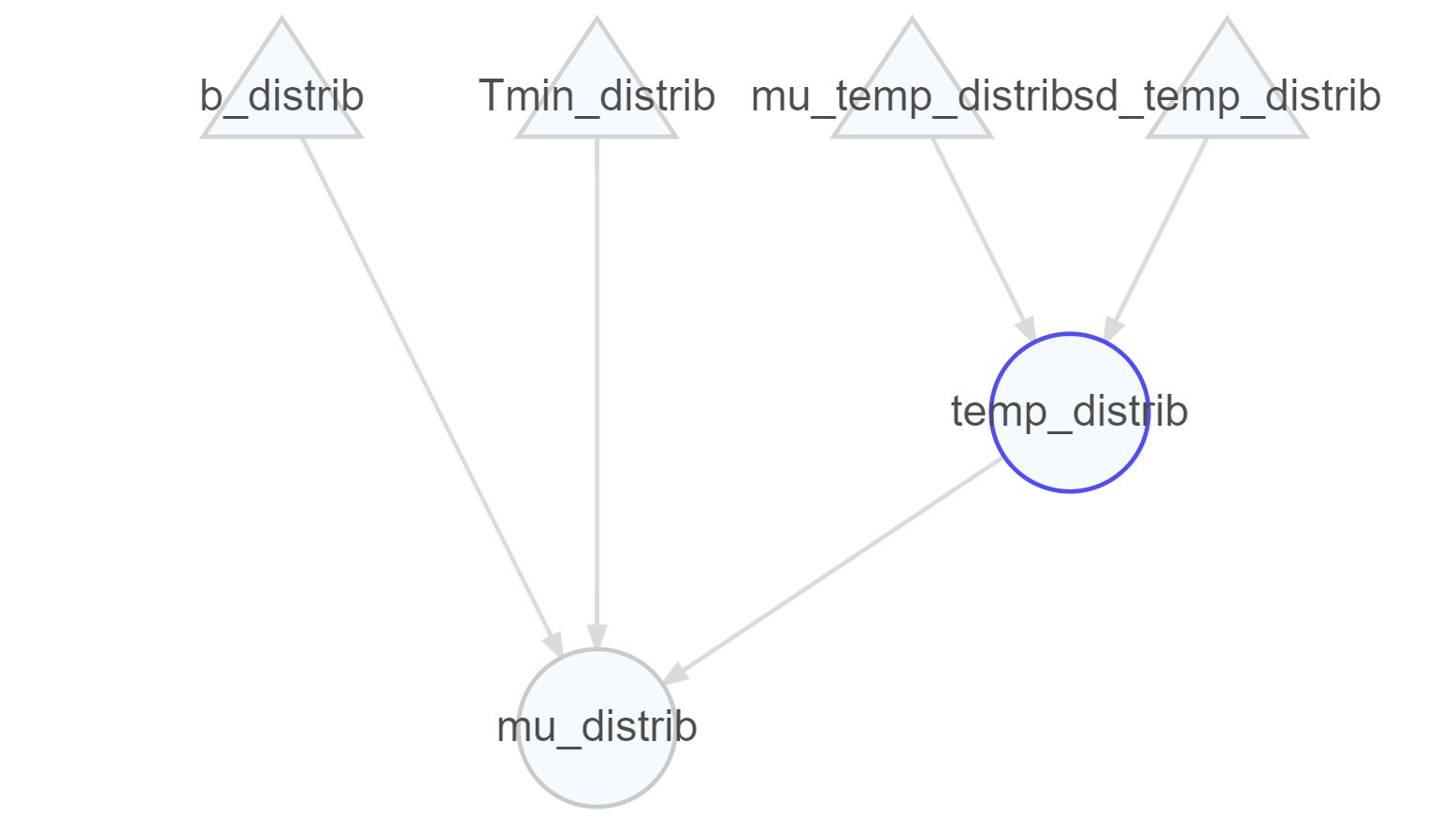
As above, we first need to define the secondary model. The Ratkowsky model has three inputs: treatment temperature, *T*min, and *b*. For the temperature, we assign a normal distribution that represents uncertainty (level = 1) with expected value (mean) of 6.35 ºC (we define the metric units) and a standard deviation of 2.83 ºC (we define the metric units) .

temp\_distrib <- Normal$new("temp\_distrib",  
 level = 1)$ # a normal distribution  
 map\_input("mu",   
 Constant$new("mu\_temp\_distrib", 6.35) # with a constant mean  
)$  
 map\_input("sigma",  
 Constant$new("sd\_temp\_distrib", 2.83) # and a constant variance  
)

Then, we define the Ratkowsky model, mapping the temperature to the element we just defined. In this case, we assume no variability or uncertainty in the model parameters, and the *b* is assumed to be 0.014 and *Tmin* to be 1.6.

mu\_distrib <- Ratkowsky\_model$new("mu\_distrib")$ #the Ratkowsky seondary model  
 map\_input("b",  
 Constant$new("b\_distrib", 0.0144) #with constant b  
)$  
 map\_input("Tmin",  
 Constant$new("Tmin\_distrib", 1.6) # and constant Tmin  
)$  
 map\_input("temperature",  
 temp\_distrib  
)

The scheme of the secondary growth model as we defined it:



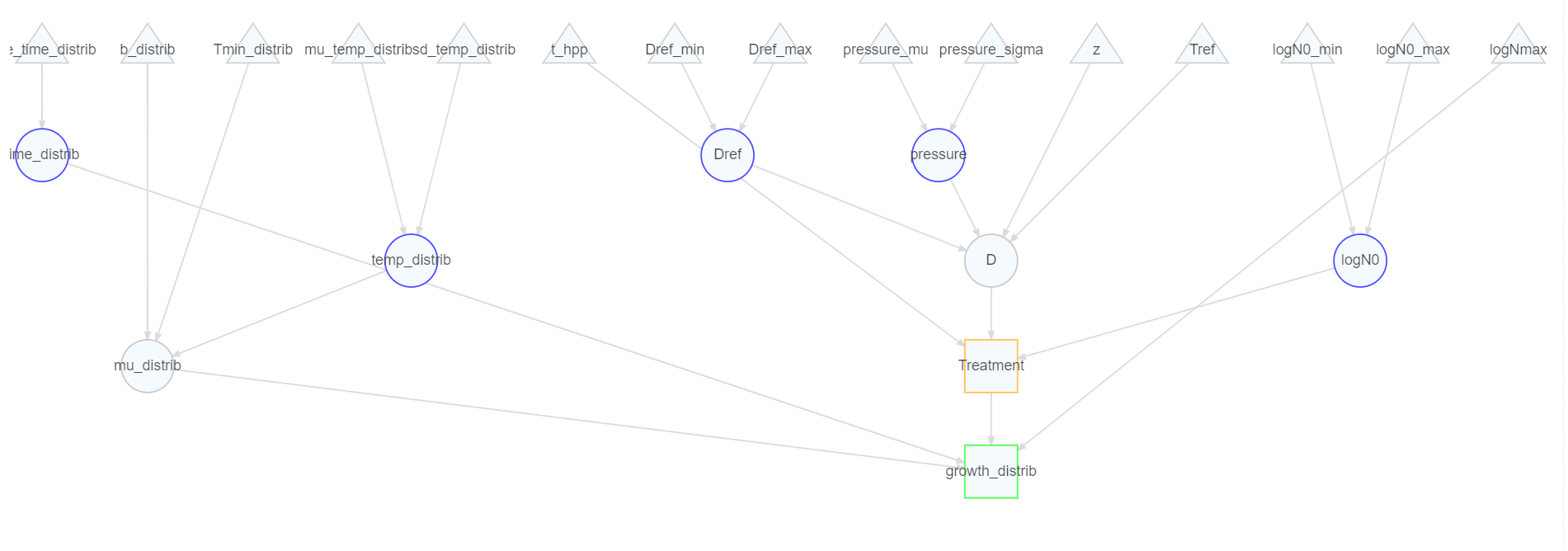
Now, we can go back to the primary growth model and integrate our secondary model there. As mentioned above, the model to be used is the **ExponentialGrowthNmax** ([Equation 4](#eq-expon_growth)). The only input left to assign is the storage time, for which we will assume an exponential distribution with rate parameter 1/29.

time\_distrib <- Exponential$new("time\_distrib")$ # an exponential distribution  
 map\_input("rate",  
 Constant$new("rate\_time\_distrib", 1/29)# with constant rate parameter  
)

Now, we have everything and we will implement the **ExponentialGrowthNmax** as our primary model. For the treatment time, we will use the storage time that we just defined. The growth rate is already mapped to the output of the Ratkowsky model. Then, the initial concentration at storage needs to be mapped to the output of the inactivation model (to be our reference logN0 for this stage). Finally, we will define a constant *Nmax* of 8 log CFU/g.

growth\_distrib <- ExponentialGrowthNmax$new("growth\_distrib")$  
 map\_input("t",  
 time\_distrib # we assign the element from above to define the time  
)$  
 map\_input("mu",  
 mu\_distrib # we assign the element from above to define the growth rate  
)$  
 map\_input("logN0",  
 inactivation # and logN0 to the output of the hpp now  
)$  
 map\_input("logNmax",  
 Constant$new("logNmax", 8) # A constant logNmax  
 )

The scheme of the initial concentration, inactivation with HPP and the growth during distribution as we defined it:



## 2.4 Consumer phase

The next step is to convert the microbial concentration (in log CFU/g) to microbial dose consumed. For this, we will use the **Concentration2Dose** element. This element considers the fact that the dose is a sampling process (i.e., the output is a discrete number of cells).

We have two inputs for the dose response model i.e., the microbial exposure at exposure and the serving size. For the latter, we use a uniform distribution that represents variability (level = 0). For the microbial concentration, we map the output of the growth element from above.

serving\_size <- Uniform$new("size", level = 0)$  
 map\_input("min", Constant$new("min\_size", 200))$  
 map\_input("max", Constant$new("max\_size", 500))  
  
consumer\_dose <- Concentration2Dose$new("dose")$  
 map\_input("logN", growth\_distrib)$  
 map\_input("size", serving\_size)

## 2.5 Risk characterization

For this stage we need first to define a dose response model. We will use the exponential dose response model:

with , assuming constant pathogen-host survival probability.

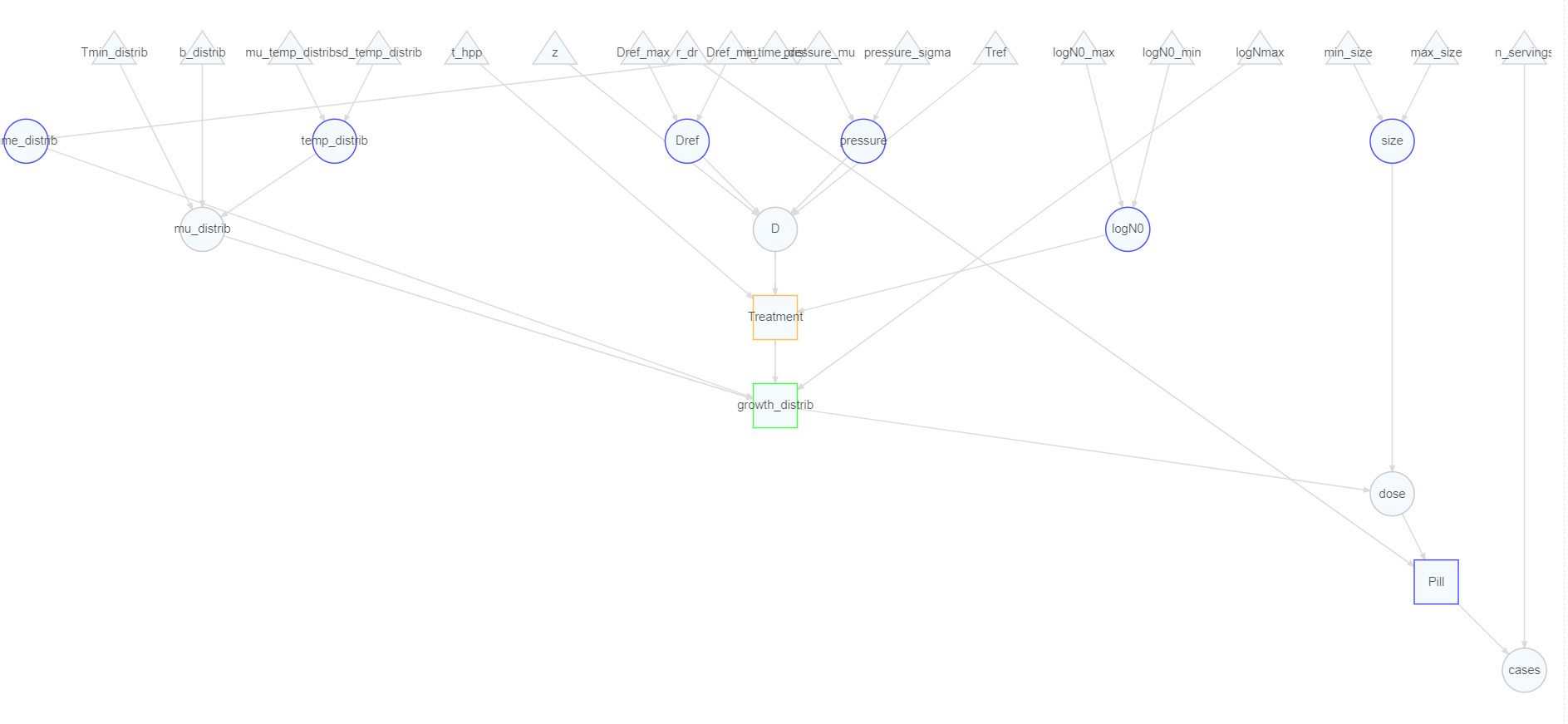
Pill <- DoseResponse\_Exponential$new("Pill")$  
 map\_input("r", Constant$new("r\_dr", 1e-12))$  
 map\_input("dose", consumer\_dose)

Then, we can estimate the number of cases. For that, **biorisk** includes the **Pill2Cases\_N** element to convert from probability of illness to number of cases, assuming that the number of cases can be described by a binomial distribution as shown below:

This element considers for each Monte Carlo iteration that the **nservings** have the same probability of illness. On the other hand, the element **Pill2Cases\_1** considers a single serving per Pill. In this case, we will make the calculations per servings.

cases <- Pill2Cases\_N$new("cases")$  
 map\_input("Pill", Pill)$  
 map\_input("servings", Constant$new("n\_servings", 1e12))

The scheme of the full QMRA:



# 3. Simulation and visualization

We can start with the simplest approach to see what would be the approximate, discrete prediction of the cases per servings.

cases$point\_estimate()

[1] 0.0002220446

## 3.1 Simulation as a 1D Monte Carlo

Then, we continue with our 1D Monte Carlo:

cases$simulate(100000, seed = 241)

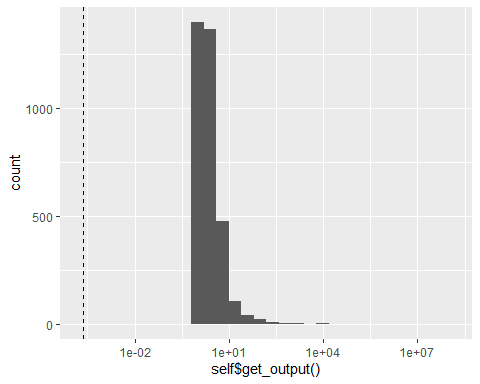
We can now visualize the number of cases per servings, as a histogram. We will use the log transform due to the heavy tail, but this removes also the 0s i.e., the simulations that lead to 0 cases. Therefore we also need to know how many are the 0s in our simulations (it’s in the warning created). We will also add a discrete line with the approximate as estimated before. (Note that this is quite biased with respect to the histogram. The reasons for this is the use of asymmetric distributions and the nonlinear models.)

cases$histogram(add\_discrete = TRUE) + scale\_x\_log10()

Warning in scale\_x\_log10(): log-10 transformation introduced infinite values.

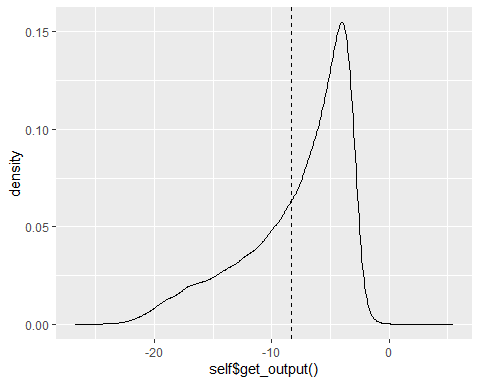
`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 96569 rows containing non-finite outside the scale range  
(`stat\_bin()`).



Then, we can also visualize other elements such as the microbial concentration at the end of storsage as a density plot (again with a discrete dashed line, “TRUE”).

growth\_distrib$density\_plot(TRUE)



Now we will take a look at the quantiles

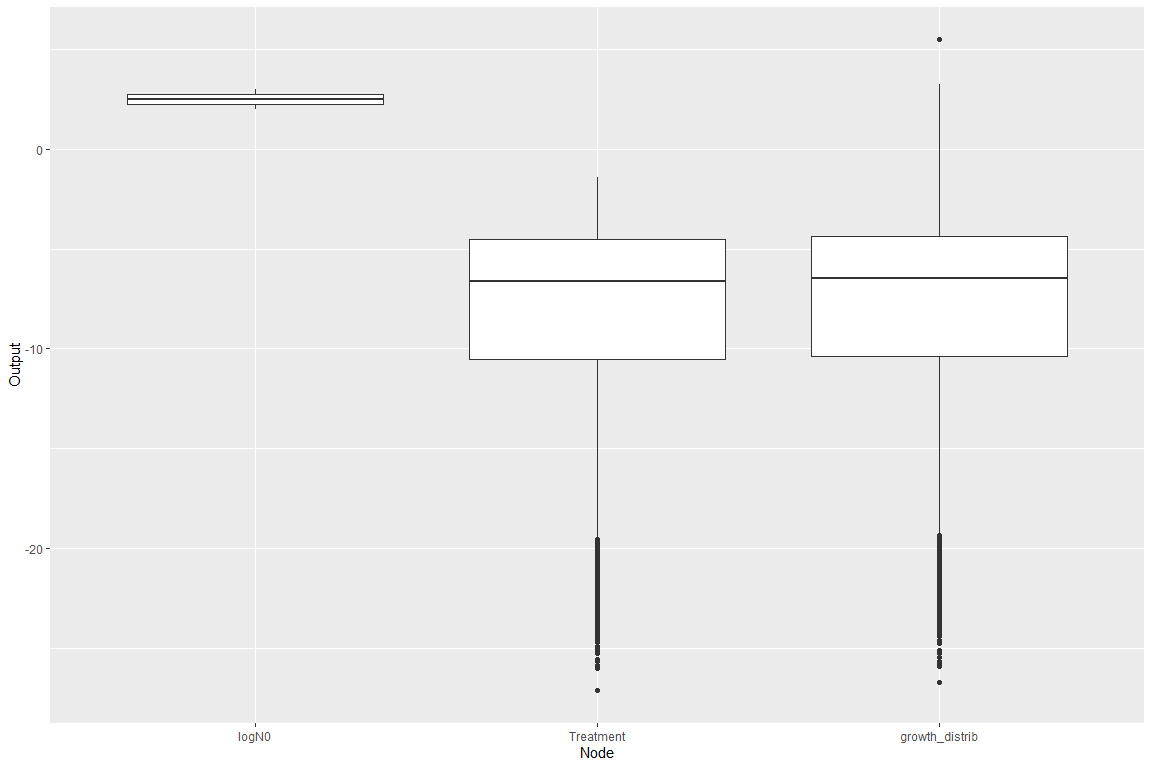
quantile\_table(cases, chosen = c("cases", "Pill", "dose", "growth\_distrib"),  
 probs = c(0.50, 0.90, 0.99))

# A tibble: 4 × 4  
 node `50%` `90%` `99%`  
 <chr> <dbl> <dbl> <dbl>  
1 growth\_distrib -6.46 -3.41 -2.34e+ 0  
2 dose 0 0 2 e+ 0  
3 Pill 0 0 2.00e-12  
4 cases 0 0 3 e+ 0

It would also be nice to monitor the variation of the microbial concentration on each step. We will do this with three ways namely, box plot, violin plot and density plot.

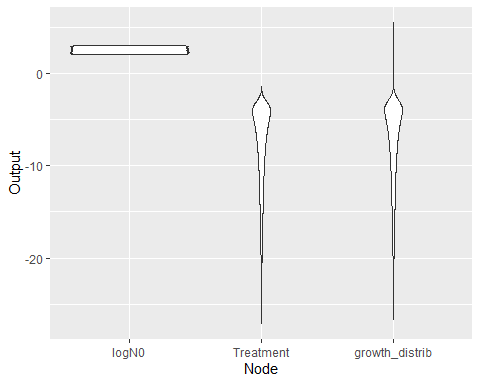
#### Box plot

plot\_outputs(cases,  
 chosen = c("logN0", "Treatment", "growth\_distrib"))



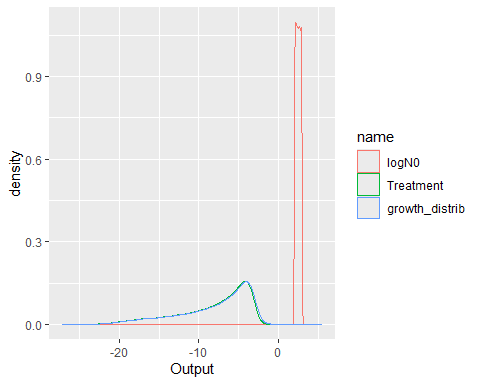
#### Violin plot

plot\_outputs(cases,  
 chosen = c("logN0", "Treatment", "growth\_distrib"),  
 type = "violin")



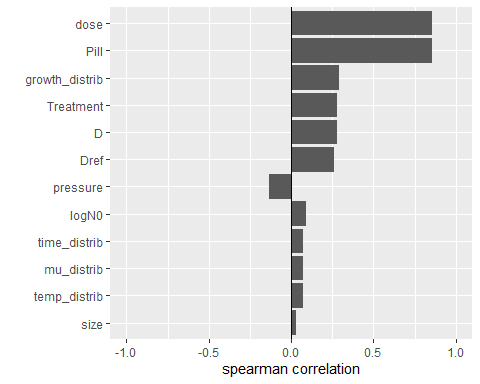
#### Density plot

plot\_outputs(cases,  
 chosen = c("logN0", "Treatment", "growth\_distrib"),  
 type = "density")



Let’s perform also a sensitiviy analysis for the overall model (**cases**):

tornado\_plot(cases)

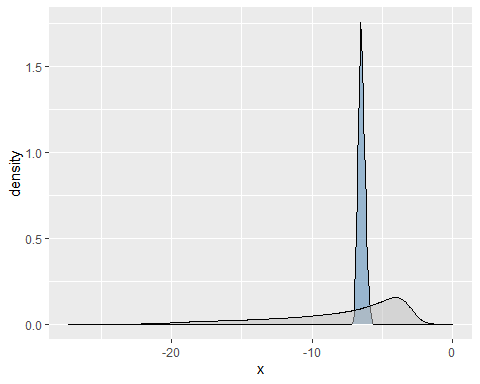


plot\_model(cases)

## 3.2 Simulation as a 2D Monte Carlo

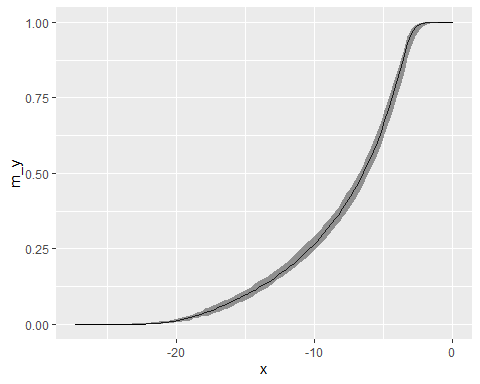
For the 2D-MC simulation we need two inputs, the number of simulations for level 0 (variability) and the number of simulations for level 1 (uncertainty). We will use 1000 iterations for variability and 100 for uncertainty. We will visualize the results for growth distribution with a density plot that compares the distribution from the variability (blue) with the distribution including all sources of variation (variability & uncertainty)(grey).

cases$simulate\_2D(1000, 100, seed = 792)  
growth\_distrib$density\_plot\_2D()



Also, we will visualize the cumulative distribution where the line represents the level 0 (variability) and the ribbon the additional variation due to the uncertainty level.

growth\_distrib$cummulative\_plot\_2D()



And our quantiles, as we did before. In this case, the quantiles are calculated both under level 0 (variability) and over the complete model (variability & uncertainty).

quantile\_table\_2D(cases, chosen = c("cases", "Pill", "dose", "growth\_distrib"),  
 probs = c(0.50, 0.90, 0.99))

# A tibble: 4 × 7  
 node `0.5\_level1` `0.9\_level1` `0.99\_level1` `0.5\_level0` `0.9\_level0`  
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 growth\_dist… -6.46 -3.44 -2.43e+ 0 -6.49 -6.16  
2 dose 0 0 2 e+ 0 0 0   
3 Pill 0 0 2.00e-12 0 0   
4 cases 0 0 2 e+ 0 0 0   
# ℹ 1 more variable: `0.99\_level0` <dbl>

# 4. The same case study with only changing the HPP inactivation to PEF

## 4.1 Initial concentration

We will assume that the microorganism of interest is *Escherichia coli.* We will define our initial concentration as the concentration of *E. coli* in the fruit juice tank after the juice is extracted from the fruit. We assume that this concentration can be described by a uniform distribution with minimum and maximum with equal probabilities to be 2 or 3 log10 CFU/ml, respectively. It has some variability which is defined in level 0.

logN0\_2 <- Uniform$new("logN0", # A uniform distribution  
 level = 0)$  
 map\_input("min",  
 Constant$new("logN0\_min", 2) # with a constant min. value  
)$  
 map\_input ("max",  
 Constant$new("logN0\_max", 3) # and a constant max. value  
)

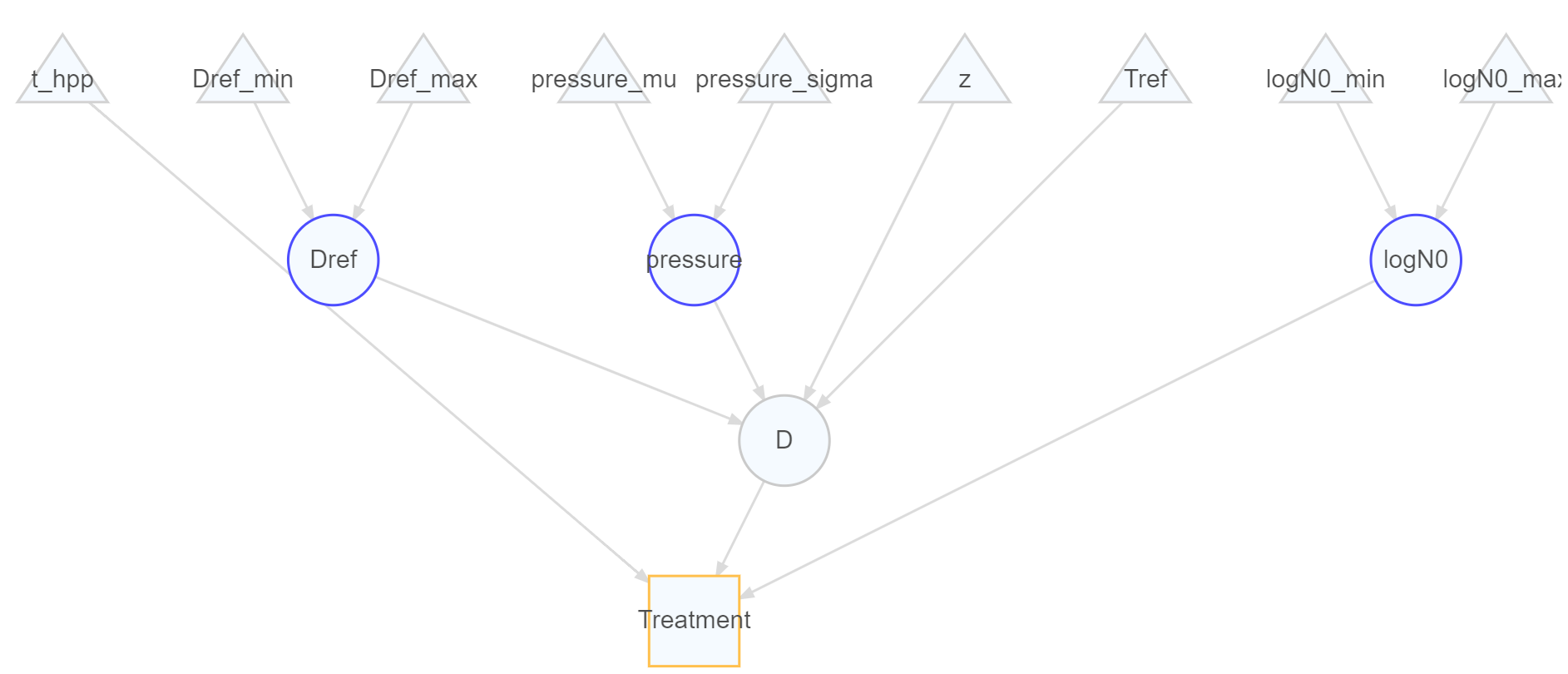
## 4.2 Inactivation using PEF processing

Moving to the next step of the microbial inactivation, we need to define our inactivation model, which in our case is a log-linear primary inactivation model, using the energy input ( in J/ml):

Here we will not define a secondary model since the effect of the energy input level is already integrated on the inactivation, replacing the time parameter. So, we move directly to the primary model, and we will use the D value, as determined from the log-linear fitting, using the model fitted for fruit juices and *E. coli*. The *D* was estimated as 82.5 J/ml. For our scenario we will use an energy input of 200 J/ml.

D\_pef <- Uniform$new("D\_pef", level = 0)$ # Variability in D\_pef  
 map\_input("min", Constant$new("D\_pef\_min", 70))$  
 map\_input("max", Constant$new("D\_pef\_max", 90))  
  
energy\_pef <- Normal$new("energy\_pef", level = 0)$ # Normal energy input level with some variability assigned  
 map\_input("mu", Constant$new("energy\_pef\_mu", 200))$  
 map\_input("sigma", Constant$new("energy\_pef\_sigma", 20))  
  
inactivation\_2 <- LogLinInactivation$new("Treatment")$  
 map\_input("logN0", logN0\_2)$ #We map logN0 to the initial count defined before  
 map\_input("D", D\_pef)$  
 map\_input("t", energy\_pef)

The scheme of the full inactivation model as we defined it:



## 4.3 Growth during storage

In this section we will describe the microbial growth during storage using the exponential growth model with stationary phase (bilinear model):

with the growth rate given by the Ratkowsky model:

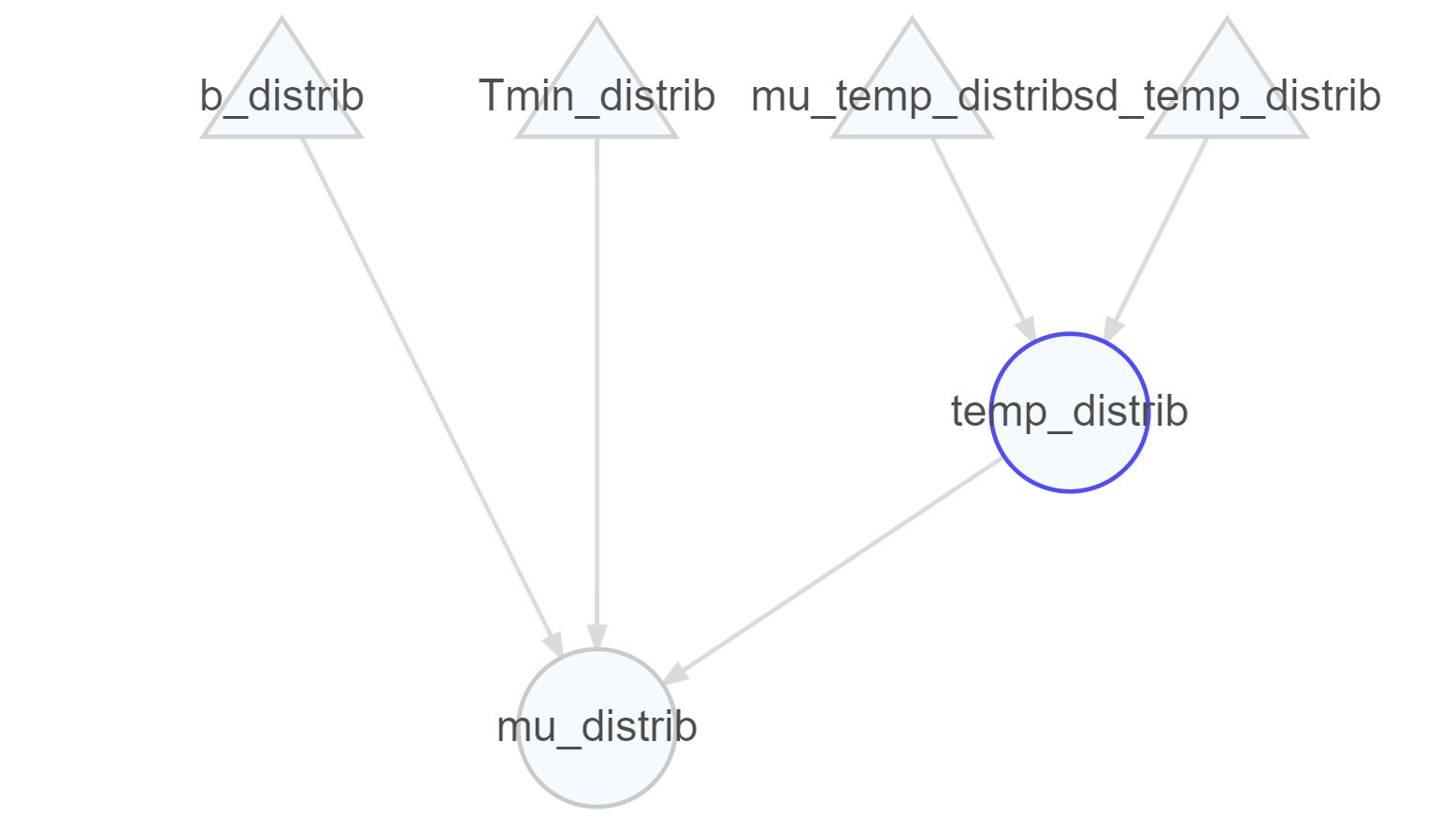
As above, we first need to define the secondary model. The Ratkowsky model has three inputs: treatment temperature, *T*min, and *b*. For the temperature, we assign a normal distribution that represents uncertainty (level = 1) with expected value (mean) of 6.35 ºC (we define the metric units) and a standard deviation of 2.83 ºC (we define the metric units) .

temp\_distrib\_2 <- Normal$new("temp\_distrib",  
 level = 1)$ # a normal distribution  
 map\_input("mu",   
 Constant$new("mu\_temp\_distrib", 6.35) # with a constant mean  
)$  
 map\_input("sigma",  
 Constant$new("sd\_temp\_distrib", 2.83) # and a constant variance  
)

Then, we define the Ratkowsky model, mapping the temperature to the element we just defined. In this case, we assume no variability or uncertainty in the model parameters, and the *b* is assumed to be 0.014 and *Tmin* to be 1.6.

mu\_distrib\_2 <- Ratkowsky\_model$new("mu\_distrib")$ #the Ratkowsky seondary model  
 map\_input("b",  
 Constant$new("b\_distrib", 0.0144) #with constant b  
)$  
 map\_input("Tmin",  
 Constant$new("Tmin\_distrib", 1.6) # and constant Tmin  
)$  
 map\_input("temperature",  
 temp\_distrib\_2  
)

The scheme of the secondary growth model as we defined it:



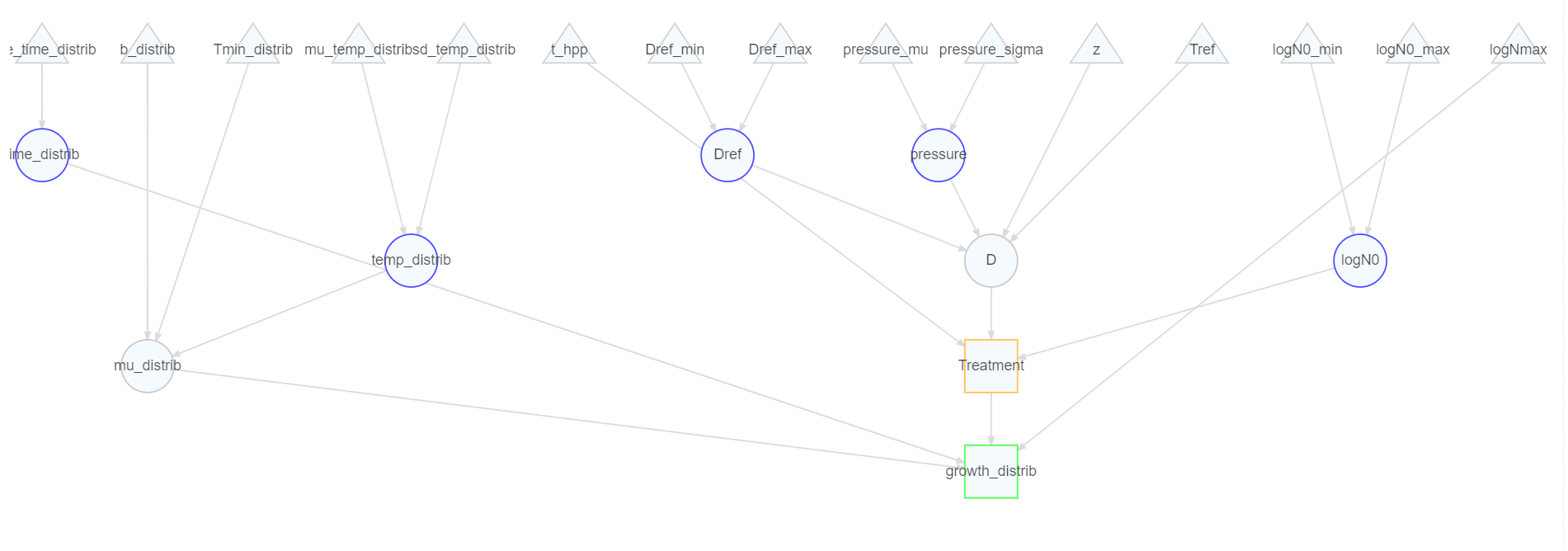
Now, we can go back to the primary growth model and integrate our secondary model there. As mentioned above, the model to be used is the **ExponentialGrowthNmax** ([Equation 4](#eq-expon_growth)). The only input left to assign is the storage time, for which we will assume an exponential distribution with rate parameter 1/29.

time\_distrib\_2 <- Exponential$new("time\_distrib")$ # an exponential distribution  
 map\_input("rate",  
 Constant$new("rate\_time\_distrib", 1/29)# with constant rate parameter  
)

Now, we have everything and we will implement the **ExponentialGrowthNmax** as our primary model. For the treatment time, we will use the storage time that we just defined. The growth rate is already mapped to the output of the Ratkowsky model. Then, the initial concentration at storage needs to be mapped to the output of the inactivation model (to be our reference logN0 for this stage). Finally, we will define a constant *Nmax* of 8 log CFU/g.

growth\_distrib\_2 <- ExponentialGrowthNmax$new("growth\_distrib")$  
 map\_input("t",  
 time\_distrib\_2 # we assign the element from above to define the time  
)$  
 map\_input("mu",  
 mu\_distrib\_2 # we assign the element from above to define the growth rate  
)$  
 map\_input("logN0",  
 inactivation\_2 # and logN0 to the output of the hpp now  
)$  
 map\_input("logNmax",  
 Constant$new("logNmax", 8) # A constant logNmax  
 )

The scheme of the initial concentration, inactivation with HPP and the growth during distribution as we defined it:



## 4.4 Consumer phase

The next step is to convert the microbial concentration (in log CFU/g) to microbial dose consumed. For this, we will use the **Concentration2Dose** element. This element considers the fact that the dose is a sampling process (i.e., the output is a discrete number of cells).

We have two inputs for the dose response model i.e., the microbial exposure at exposure and the serving size. For the latter, we use a uniform distribution that represents variability (level = 0). For the microbial concentration, we map the output of the growth element from above.

serving\_size\_2 <- Uniform$new("size", level = 0)$  
 map\_input("min", Constant$new("min\_size", 200))$  
 map\_input("max", Constant$new("max\_size", 500))  
  
consumer\_dose\_2 <- Concentration2Dose$new("dose")$  
 map\_input("logN", growth\_distrib\_2)$  
 map\_input("size", serving\_size\_2)

## 4.5 Risk characterization

For this stage we need first to define a dose response model. We will use the exponential dose response model:

with , assuming constant pathogen-host survival probability.

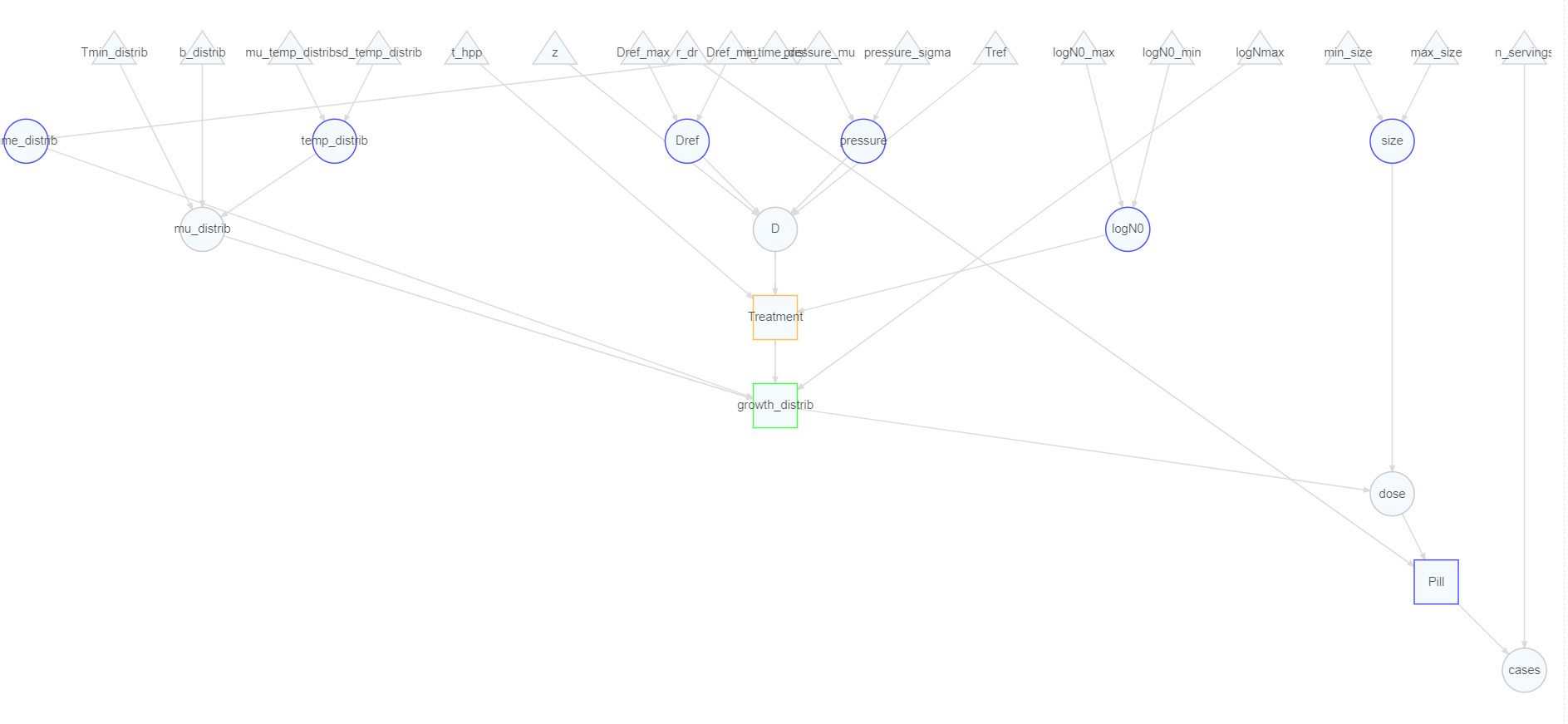
Pill\_2 <- DoseResponse\_Exponential$new("Pill")$  
 map\_input("r", Constant$new("r\_dr", 1e-12))$  
 map\_input("dose", consumer\_dose\_2)

Then, we can estimate the number of cases. For that, **biorisk** includes the **Pill2Cases\_N** element to convert from probability of illness to number of cases, assuming that the number of cases can be described by a binomial distribution as shown below:

This element considers for each Monte Carlo iteration that the **nservings** have the same probability of illness. On the other hand, the element **Pill2Cases\_1** considers a single serving per Pill. In this case, we will make the calculations per servings.

cases\_2 <- Pill2Cases\_N$new("cases")$  
 map\_input("Pill", Pill\_2)$  
 map\_input("servings", Constant$new("n\_servings", 1e12))

The scheme of the full QMRA:



# 5. Simulation and visualization

We can start with the simplest approach to see what would be the approximate, discrete prediction of the cases per servings.

cases\_2$point\_estimate()

[1] 53642781

## 5.1 Simulation as a 1D Monte Carlo

Then, we continue with our 1D Monte Carlo:

cases\_2$simulate(100000, seed = 241)

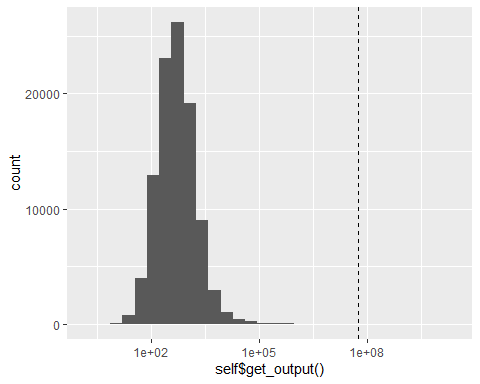
We can now visualize the number of cases per servings, as a histogram. We will use the log transform due to the heavy tail, but this removes also the 0s i.e., the simulations that lead to 0 cases. Therefore we also need to know how many are the 0s in our simulations (it’s in the warning created). We will also add a discrete line with the approximate as estimated before. (Note that this is quite biased with respect to the histogram. The reasons for this is the use of asymmetric distributions and the nonlinear models.)

cases\_2$histogram(add\_discrete = TRUE) + scale\_x\_log10()

Warning in scale\_x\_log10(): log-10 transformation introduced infinite values.

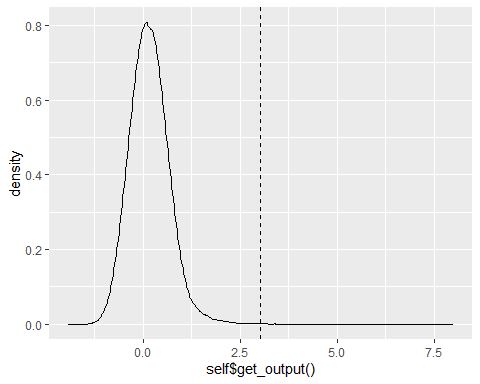
`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 2 rows containing non-finite outside the scale range  
(`stat\_bin()`).



Then, we can also visualize other elements such as the microbial concentration at the end of storsage as a density plot (again with a discrete dashed line, “TRUE”).

growth\_distrib\_2$density\_plot(TRUE)



Now we will take a look at the quantiles

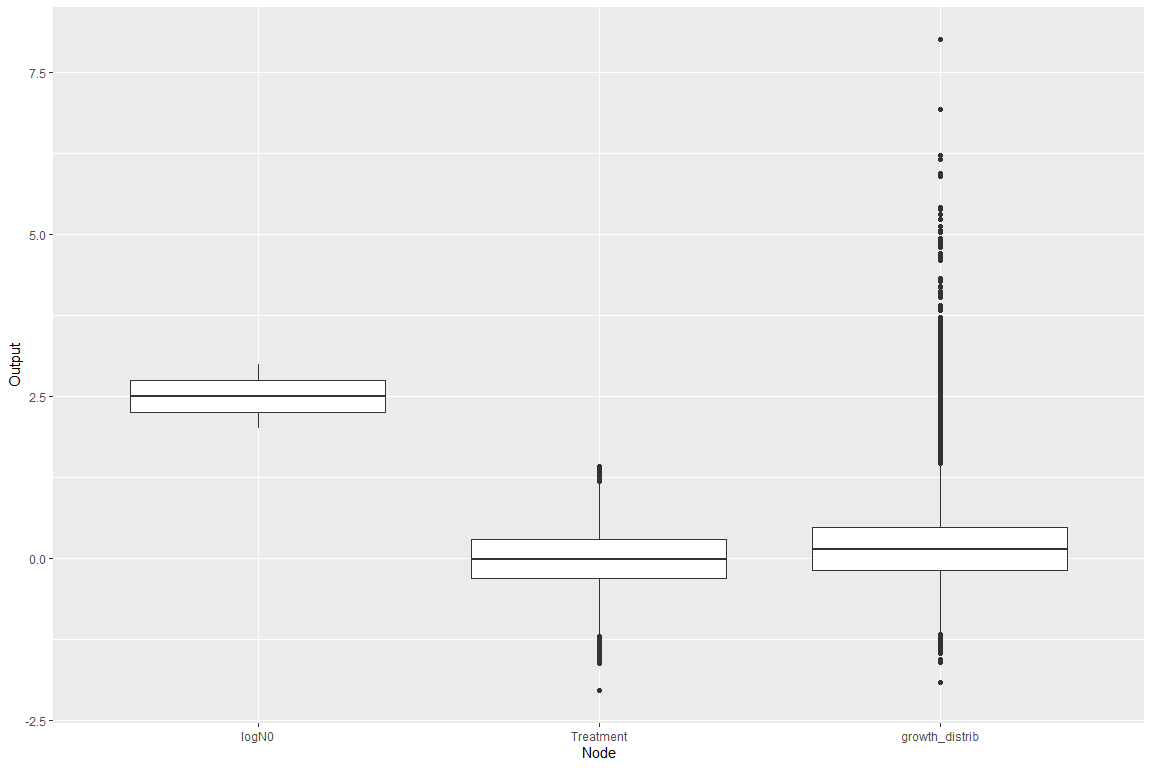
quantile\_table(cases\_2, chosen = c("cases", "Pill", "dose", "growth\_distrib"),  
 probs = c(0.50, 0.90, 0.99))

# A tibble: 4 × 4  
 node `50%` `90%` `99%`  
 <chr> <dbl> <dbl> <dbl>  
1 growth\_distrib 1.42e- 1 7.98e-1 1.66e+0  
2 dose 4.7 e+ 2 2.20e+3 1.64e+4  
3 Pill 4.70e-10 2.20e-9 1.64e-8  
4 cases 4.7 e+ 2 2.21e+3 1.64e+4

It would also be nice to monitor the variation of the microbial concentration on each step. We will do this with three ways namely, box plot, violin plot and density plot.

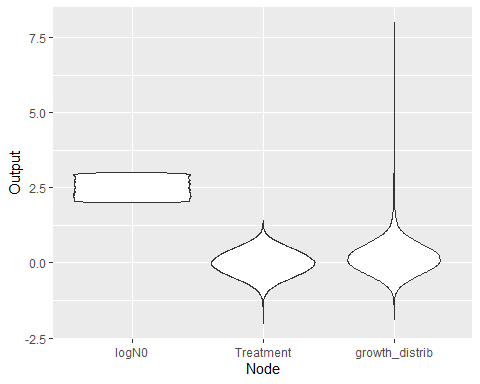
#### Box plot

plot\_outputs(cases\_2,  
 chosen = c("logN0", "Treatment", "growth\_distrib"))



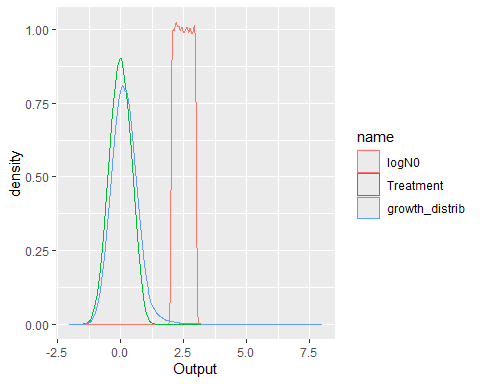
#### Violin plot

plot\_outputs(cases\_2,  
 chosen = c("logN0", "Treatment", "growth\_distrib"),  
 type = "violin")



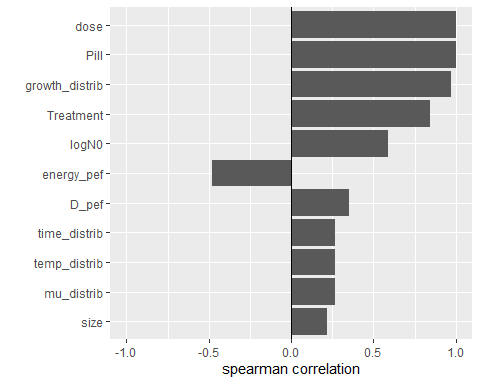
#### Density plot

plot\_outputs(cases\_2,  
 chosen = c("logN0", "Treatment", "growth\_distrib"),  
 type = "density")



Let’s perform also a sensitiviy analysis for the overall model (**cases**):

tornado\_plot(cases\_2)

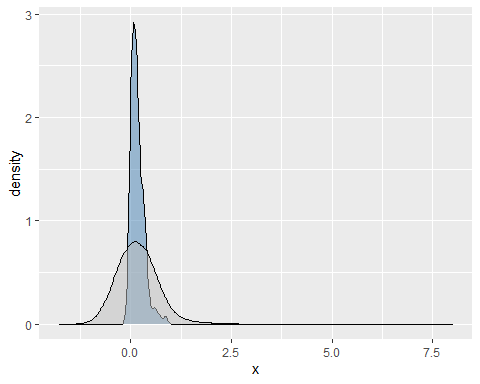


plot\_model(cases\_2)

## 5.2 Simulation as a 2D Monte Carlo

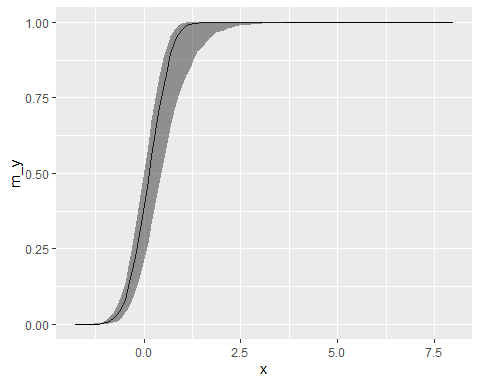
For the 2D-MC simulation we need two inputs, the number of simulations for level 0 (variability) and the number of simulations for level 1 (uncertainty). We will use 1000 iterations for variability and 100 for uncertainty. We will visualize the results for growth distribution with a density plot that compares the distribution from the variability (blue) with the distribution including all sources of variation (variability & uncertainty)(grey).

cases\_2$simulate\_2D(1000, 100, seed = 792)  
growth\_distrib\_2$density\_plot\_2D()



Also, we will visualize the cumulative distribution where the line represents the level 0 (variability) and the ribbon the additional variation due to the uncertainty level.

growth\_distrib\_2$cummulative\_plot\_2D()



And our quantiles, as we did before. In this case, the quantiles are calculated both under level 0 (variability) and over the complete model (variability & uncertainty).

quantile\_table\_2D(cases\_2, chosen = c("cases", "Pill", "dose", "growth\_distrib"),  
 probs = c(0.50, 0.90, 0.99))

# A tibble: 4 × 7  
 node `0.5\_level1` `0.9\_level1` `0.99\_level1` `0.5\_level0` `0.9\_level0`  
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 growth\_dist… 1.50e- 1 8.11e-1 1.81e+0 1.29e- 1 3.46e- 1  
2 dose 4.81e+ 2 2.30e+3 2.28e+4 4.62e+ 2 7.55e+ 2  
3 Pill 4.81e-10 2.30e-9 2.28e-8 4.62e-10 7.55e-10  
4 cases 4.82e+ 2 2.29e+3 2.29e+4 4.61e+ 2 7.49e+ 2  
# ℹ 1 more variable: `0.99\_level0` <dbl>