Webinar EFI – Exploring SpaDES

Tati Micheletti

2024-04-12

This is a document with changes to be made during the presentation of *SpaDES: a tool to PERFICT your workflow*, on April 15th, 2024. The presentation is available on YouTube and on the ESA/EFI Statistical Methods Seminar Series.

During the presentation, we will download and run an integrated project (demo.R), and check the results and dive into the modules using the current document as a guide.

This document is available at: https://tinyurl.com/webinarEFI2/

1. Exploring the results from the simulation

Once the simulation from demo.R has ran, we can explore the results. Using `completed()` we can observe the list of completed events, their times, type, start and end clock time.

```
> SpaDES.core::completed(results)
    eventTime         moduleName            eventType eventPriority ._prevEventTimeFinish            clo
        <num>             <char>                <char>         <num>                 <POSc>
 1:      2013         checkpoint                  init             0   2024-04-12 13:46:14 2024-04-12 13
 2:      2013               save                  init             0   2024-04-12 13:46:14 2024-04-12 13
 3:      2013           progress                  init             0   2024-04-12 13:46:14 2024-04-12 13
 4:      2013               load                  init             0   2024-04-12 13:46:14 2024-04-12 13
 5:      2013   speciesAbundance                  init             1   2024-04-12 13:46:14 2024-04-12 13
 6:      2013        temperature                  init             1   2024-04-12 13:46:14 2024-04-12 13
 7:      2013 speciesAbundTempLM                  init             1   2024-04-12 13:46:14 2024-04-12 13
 8:      2013   speciesAbundance       tableToRasters             5   2024-04-12 13:46:15 2024-04-12 13
 9:      2013   speciesAbundance                 plot             5   2024-04-12 13:46:15 2024-04-12 13
10:      2013        temperature         dataToRaster             5   2024-04-12 13:46:15 2024-04-12 13
11:      2013        temperature             plotting             5   2024-04-12 13:46:15 2024-04-12 13
12:      2013 speciesAbundTempLM        tableBuilding             5   2024-04-12 13:46:15 2024-04-12 13
13:      2014   speciesAbundance       tableToRasters             5   2024-04-12 13:46:15 2024-04-12 13
14:      2014        temperature         dataToRaster             5   2024-04-12 13:46:15 2024-04-12 13
15:      2014 speciesAbundTempLM        tableBuilding             5   2024-04-12 13:46:15 2024-04-12 13
16:      2015   speciesAbundance       tableToRasters             5   2024-04-12 13:46:15 2024-04-12 13
17:      2015        temperature         dataToRaster             5   2024-04-12 13:46:15 2024-04-12 13
18:      2015 speciesAbundTempLM        tableBuilding             5   2024-04-12 13:46:15 2024-04-12 13
19:      2016   speciesAbundance       tableToRasters             5   2024-04-12 13:46:15 2024-04-12 13
20:      2016        temperature         dataToRaster             5   2024-04-12 13:46:15 2024-04-12 13
21:      2016 speciesAbundTempLM        tableBuilding             5   2024-04-12 13:46:15 2024-04-12 13
22:      2017   speciesAbundance       tableToRasters             5   2024-04-12 13:46:15 2024-04-12 13
23:      2017        temperature         dataToRaster             5   2024-04-12 13:46:15 2024-04-12 13
24:      2017 speciesAbundTempLM        tableBuilding             5   2024-04-12 13:46:15 2024-04-12 13
25:      2018   speciesAbundance                 plot             5   2024-04-12 13:46:15 2024-04-12 13
```

	eventTime	moduleName	eventType	eventPriority	._prevEventTimeFinish	clo
26:	2018	speciesAbundance	tableToRasters	5	2024-04-12 13:46:15	2024-04-12 13
27:	2018	temperature	dataToRaster	5	2024-04-12 13:46:15	2024-04-12 13
28:	2018	speciesAbundTempLM	tableBuilding	5	2024-04-12 13:46:15	2024-04-12 13
29:	2019	speciesAbundance	tableToRasters	5	2024-04-12 13:46:15	2024-04-12 13
30:	2019	temperature	dataToRaster	5	2024-04-12 13:46:15	2024-04-12 13
31:	2019	speciesAbundTempLM	tableBuilding	5	2024-04-12 13:46:15	2024-04-12 13
32:	2020	speciesAbundance	tableToRasters	5	2024-04-12 13:46:15	2024-04-12 13
33:	2020	temperature	dataToRaster	5	2024-04-12 13:46:16	2024-04-12 13
34:	2020	speciesAbundTempLM	tableBuilding	5	2024-04-12 13:46:16	2024-04-12 13
35:	2021	speciesAbundance	tableToRasters	5	2024-04-12 13:46:16	2024-04-12 13
36:	2021	temperature	dataToRaster	5	2024-04-12 13:46:16	2024-04-12 13
37:	2021	speciesAbundTempLM	tableBuilding	5	2024-04-12 13:46:16	2024-04-12 13
38:	2022	speciesAbundance	tableToRasters	5	2024-04-12 13:46:16	2024-04-12 13
39:	2022	temperature	dataToRaster	5	2024-04-12 13:46:16	2024-04-12 13
40:	2022	speciesAbundTempLM	tableBuilding	5	2024-04-12 13:46:16	2024-04-12 13
41:	2023	temperature	plotting	5	2024-04-12 13:46:16	2024-04-12 13
42:	2023	temperature	dataToRaster	5	2024-04-12 13:46:16	2024-04-12 13
43:	2023	speciesAbundTempLM	tableBuilding	5	2024-04-12 13:46:16	2024-04-12 13
44:	2023	speciesAbundTempLM	modelBuilding	5	2024-04-12 13:46:16	2024-04-12 13
45:	2023	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:16	2024-04-12 13
46:	2023	speciesAbundTempLM	plot	5	2024-04-12 13:46:16	2024-04-12 13
47:	2024	temperature	dataToRaster	5	2024-04-12 13:46:16	2024-04-12 13
48:	2024	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:16	2024-04-12 13
49:	2024	speciesAbundTempLM	plot	5	2024-04-12 13:46:17	2024-04-12 13
50:	2025	temperature	dataToRaster	5	2024-04-12 13:46:17	2024-04-12 13
51:	2025	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:17	2024-04-12 13
52:	2025	speciesAbundTempLM	plot	5	2024-04-12 13:46:17	2024-04-12 13
53:	2026	temperature	dataToRaster	5	2024-04-12 13:46:17	2024-04-12 13
54:	2026	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:17	2024-04-12 13
55:	2026	speciesAbundTempLM	plot	5	2024-04-12 13:46:17	2024-04-12 13
56:	2027	temperature	dataToRaster	5	2024-04-12 13:46:17	2024-04-12 13
57:	2027	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:17	2024-04-12 13
58:	2027	speciesAbundTempLM	plot	5	2024-04-12 13:46:17	2024-04-12 13
59:	2028	temperature	dataToRaster	5	2024-04-12 13:46:17	2024-04-12 13
60:	2028	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:17	2024-04-12 13
61:	2028	speciesAbundTempLM	plot	5	2024-04-12 13:46:17	2024-04-12 13
62:	2029	temperature	dataToRaster	5	2024-04-12 13:46:17	2024-04-12 13
63:	2029	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:17	2024-04-12 13
64:	2029	speciesAbundTempLM	plot	5	2024-04-12 13:46:18	2024-04-12 13
65:	2030	temperature	dataToRaster	5	2024-04-12 13:46:18	2024-04-12 13
66:	2030	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:18	2024-04-12 13
67:	2030	speciesAbundTempLM	plot	5	2024-04-12 13:46:18	2024-04-12 13
68:	2031	temperature	dataToRaster	5	2024-04-12 13:46:18	2024-04-12 13
69:	2031	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:18	2024-04-12 13
70:	2031	speciesAbundTempLM	plot	5	2024-04-12 13:46:18	2024-04-12 13
71:	2032	temperature	dataToRaster	5	2024-04-12 13:46:18	2024-04-12 13
72:	2032	speciesAbundTempLM	abundanceForecasting	5	2024-04-12 13:46:18	2024-04-12 13
73:	2032	speciesAbundTempLM	plot	5	2024-04-12 13:46:18	2024-04-12 13

We can also see how modules and objects interact using the functions `objectDiagram()` and `moduleDiagram()`.

```
SpaDES.core::moduleDiagram(results)
```

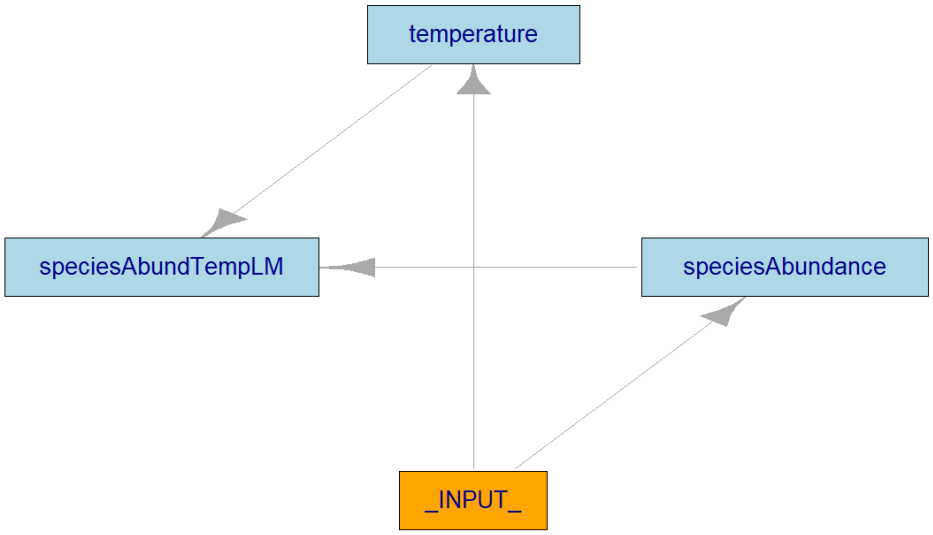

Figure 1: Figure 1. Modules' diagram showing the interactions among modules.

```
SpaDES.core::objectDiagram(results)
```

Accessing the forecasted abundances and the difference raster is also possible. This is done by calling the `object` name from the `results` list.

```
terra::plot(rast(results$forecasts))
```

```
terra::plot(results$forecastedDifferences, col = c("#49006A", "#7A0177", "#AE017E",
                                                   "#DD3497", "#F768A1", "#FA9FB5",
                                                   "#FCC5C0", "#FDE0DD", "#FFF7F3"))
```

2. Add a validation module to make it more PERFICT

To add another module to our project, for example, an evaluation one, we simply need to add the following line `"tati-micheletti/evaluateLM@main"` to the setupProject call (in the modules' list) and re-run it. The call should look like this:

```
Setup <- SpaDES.project::setupProject(

  paths = list(projectPath = "integratingSpaDESmodules",
               modulePath = "SpaDES_Modules",
```

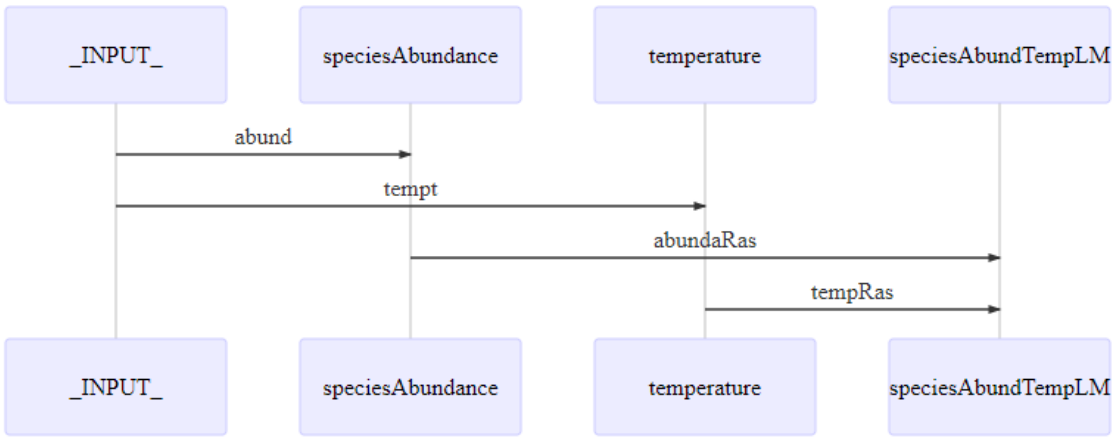

Figure 2: Figure 2. Object diagram showing the interactions among modules through objects.

```
              outputPath = "outputs"),

  modules = c("tati-micheletti/speciesAbundance@main",
              "tati-micheletti/temperature@main",
              "tati-micheletti/speciesAbundTempLM@main",
              "tati-micheletti/evaluateLM@main"), # <~~ Here we added the new module

  times = list(start = 2013,
               end = 2032),

  Restart = TRUE
)
```

After it has finished, we can observe several evaluation metrics performed by the `evaluateLM` module, such as RSE, R^2 and p value of covariates, and a cross-validation:

```
$summary

Call:  glm(formula = abundance ~ temperature, family = "poisson", data = sim$modDT)

Coefficients:
(Intercept)  temperature
    3.06226      0.04636

Degrees of Freedom: 98009 Total (i.e. Null);  98008 Residual
Null Deviance:      985100
Residual Deviance: 916100   AIC: 1496000

$crossValidation

Linear Regression

98010 samples
```

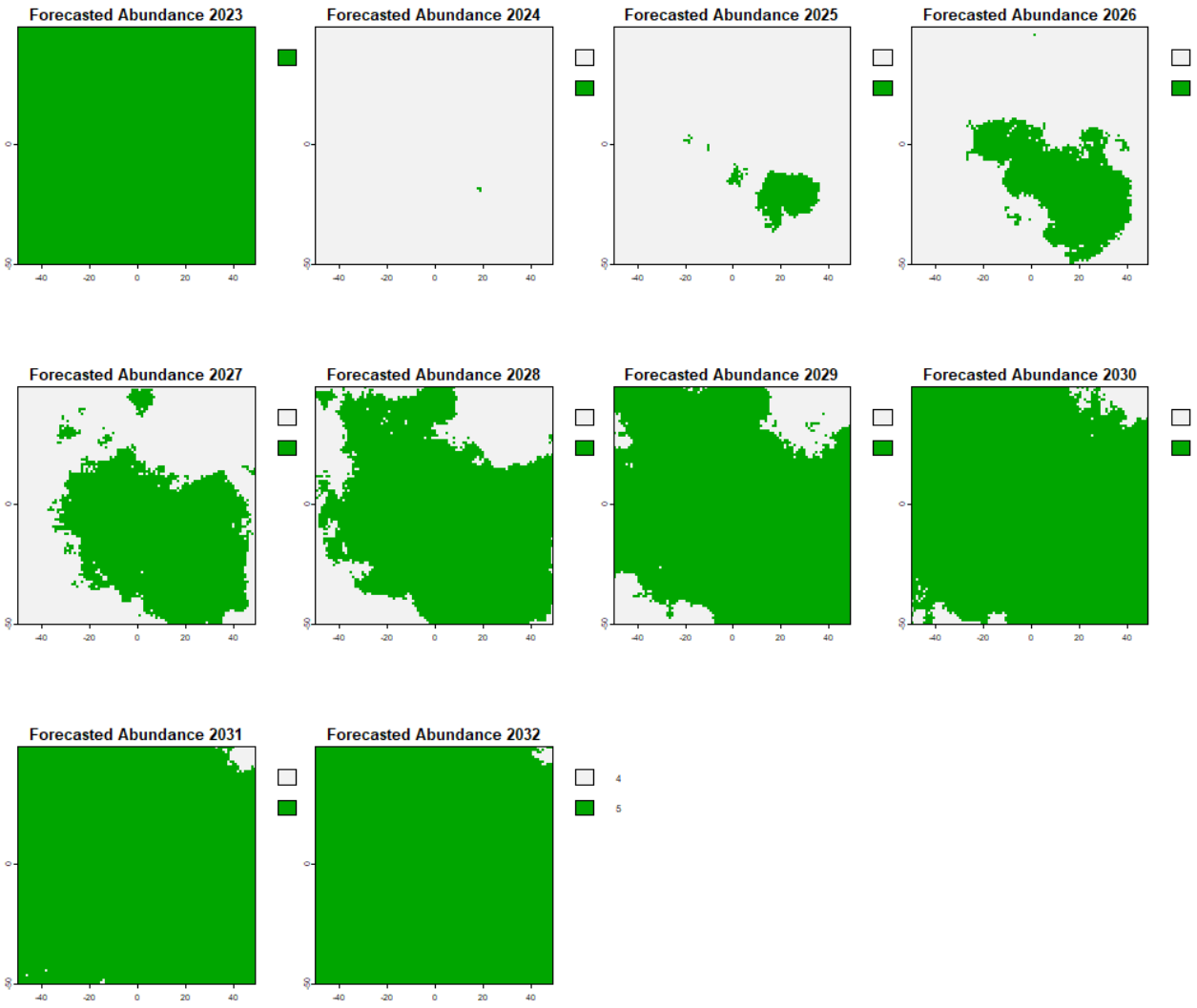

Figure 3: Figure 3. Forecasted abundances from 2023 to 2032.

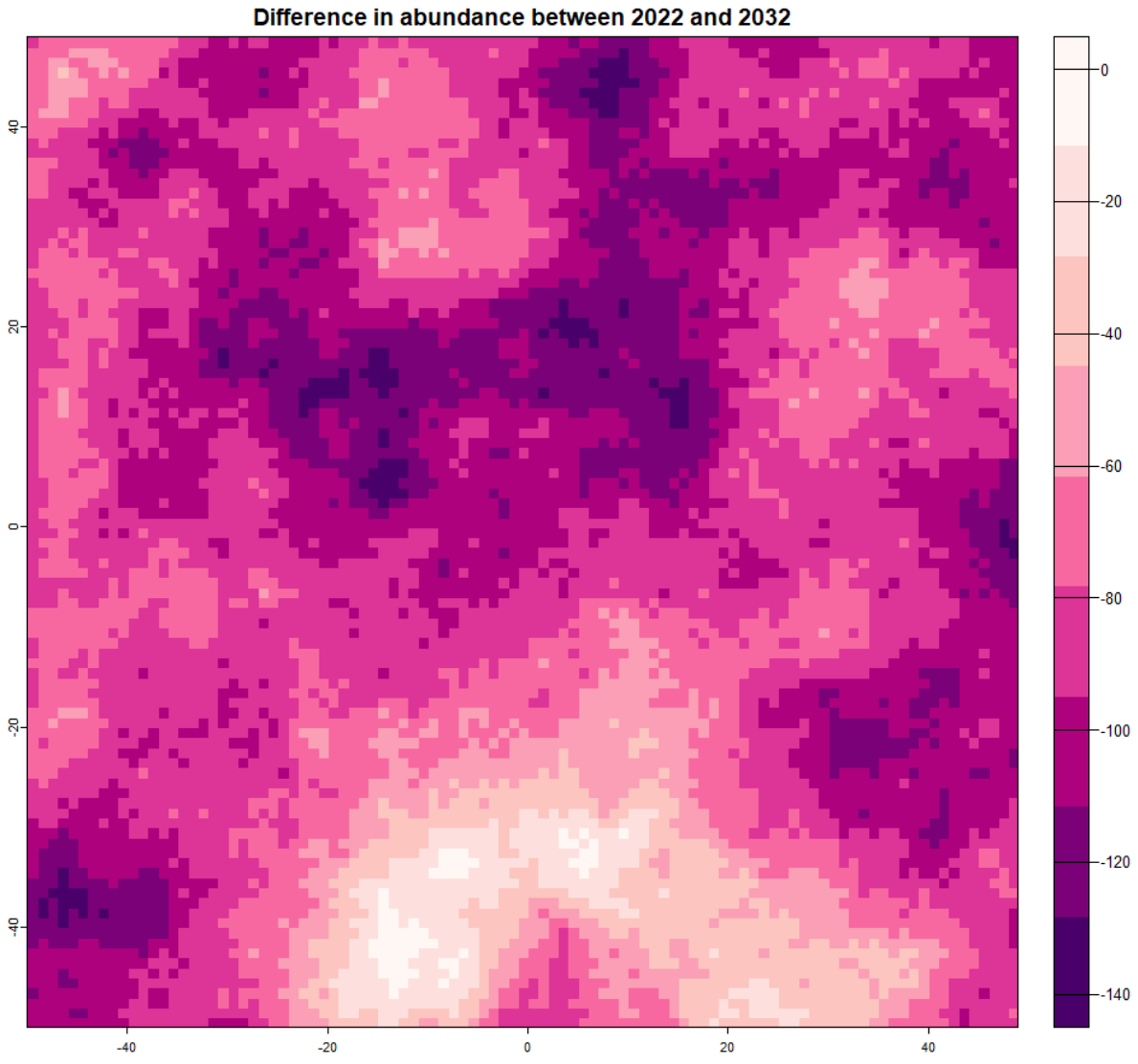

Figure 4: Figure 4. Calculated difference in abundance between 2022 and 2032.

```
    1 predictor

No pre-processing
Resampling: Cross-Validated (20 fold)
Summary of sample sizes: 93109, 93109, 93109, 93110, 93109, 93110, ...
Resampling results:

  RMSE      Rsquared    MAE
  23.93154  0.07304125  19.47062

Tuning parameter 'intercept' was held constant at a value of TRUE
```

We can also see the Q-Q Plot:

And the residuals vs fitted values plot:

3. Remove modules without breaking the workflow

We can also remove modules, and test different combinations of modules without breaking the workflow. We can test, for example, the module `speciesAbundace` alone:

```
Setup <- SpaDES.project::setupProject(

  paths = list(projectPath = "integratingSpaDESmodules",
               modulePath = "SpaDES_Modules",
               outputPath = "outputs"),

  modules = "tati-micheletti/speciesAbundance@main", # <~~ Here we run only one module

  times = list(start = 2013,
               end = 2032),

  Restart = TRUE
)
```

4. Testing, testing, testing

We can also observe the module tests located in the folder: `~/integratingSpaDESmodules/SpaDES_Modules/speciesAbundan`
all tests can be run by running the script `~/integratingSpaDESmodules/SpaDES_Modules/speciesAbundance/tests/unitT`

```
> # to test all the test files in the tests folder:
> source(file.path(getwd(), "SpaDES_Modules/speciesAbundance/tests/testthat/test-fullModule.R"))
Test passed
```

5. Modifying an existing Module

Finally, we can reuse modules by modifying their code. Here we will create a new function to add to the module `speciesAbundance`.

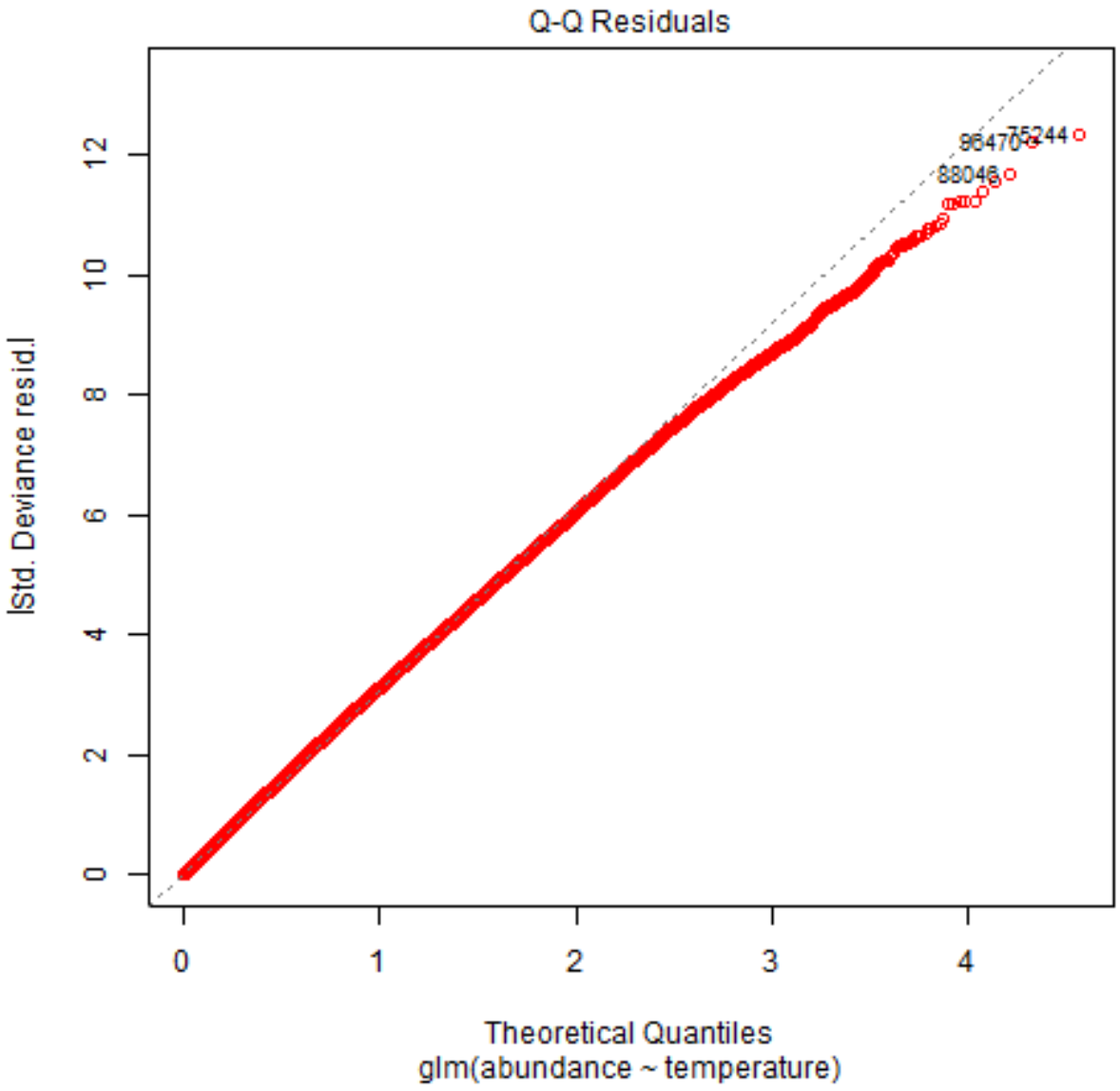

Figure 5: Figure 5. Modules' diagram showing the interactions among modules.

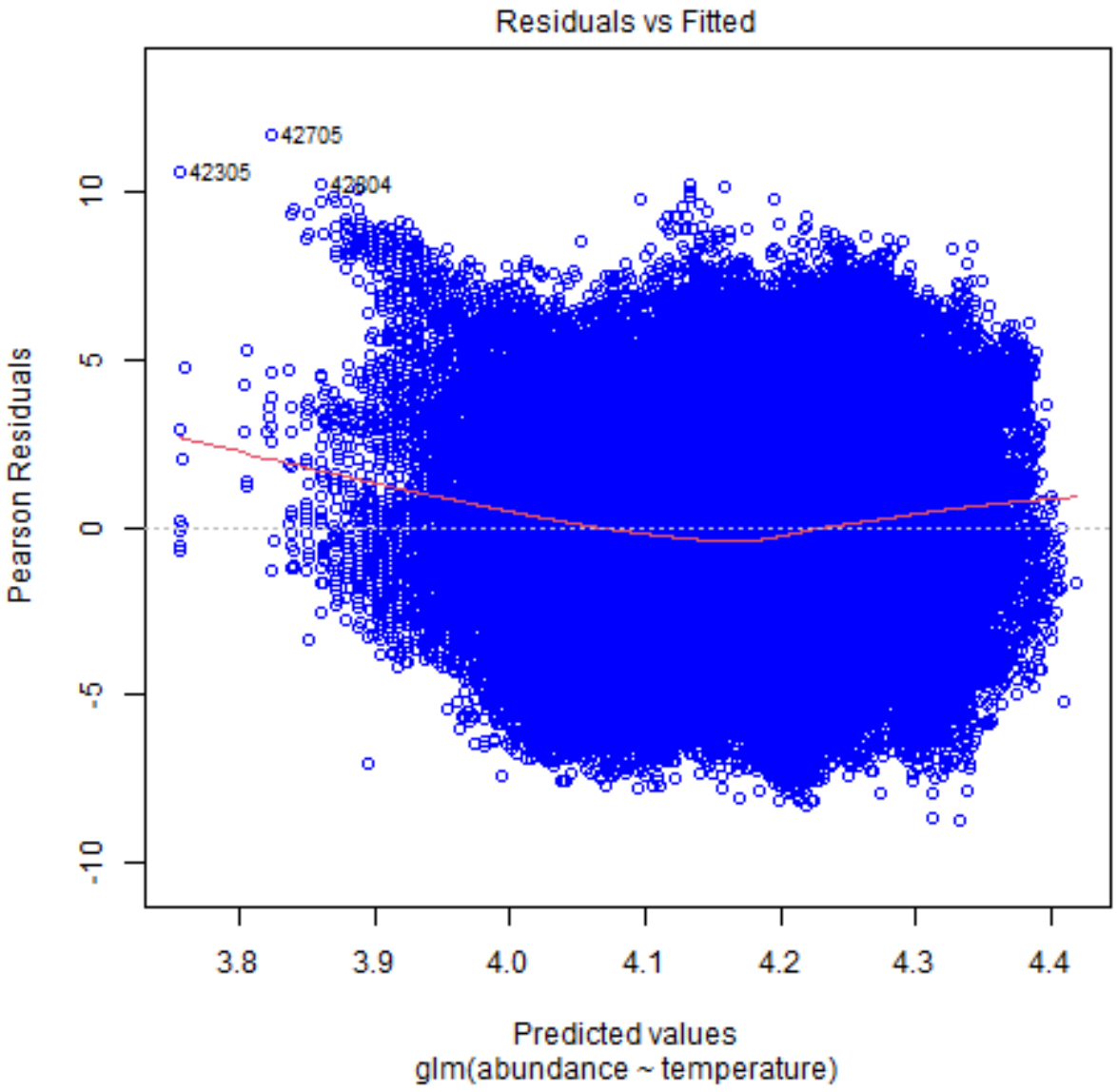

Figure 6: Figure 6. Modules' diagram showing the interactions among modules.

5.1. Create the function and add it to the module

It is important to highlight that the function may be places in the module, or in a script in the module's `R` folder.

```
modelAbundTime <- function(abundanceData){
  modAbund <- lm(formula = abundance ~ years, data = abundanceData)
  summary(modAbund)
  return(modAbund)
}
```

5.2. Add the event to the module

Now we need to add the event that will run our function to the module. For that, it helps copying an event and just changing the code.

```
  abundanceThroughTime = {

  sim$modAbund <- modelAbundTime(abundanceData = sim$abund)

  # No need to schedule further events as this one happens at the end of the
  # module's data
}
```

Note that, alternatively, we could have directly added the block of code to the event as below, without creating the function. The drawback is that once more complexity is added, the code becomes long and harder to follow.

```
  abundanceThroughTime = {

  modAbund <- lm(formula = abundance ~ years, data = sim$abund)
  sim$modAbund <- summary(modAbund)


  # No need to schedule further events as this one happens at the end of the
  # module's data
}
```

5.3. Schedule the event `abundanceThroughTime` in the `init` event

In order for our new event to run, we need to schedule it in the `init` event. An `init` event is the only mandatory in all modules. We will schedule our event to the last year of the project run.

```
sim <- scheduleEvent(sim, end(sim), "speciesAbundance", "abundanceThroughTime")
```

5.4. Declare `modAbund` as a createdOutput

```
createsOutput(objectName = "modAbund", objectClass = "lm",
              desc = paste0("A fitted model (of the `lm` class) of abundance through time"))
```

Now you can re-run the project and see the results from the added event. You can access the new object created (`modAbund`) in the same way we accessed the forecasts earlier:

```
> results$modAbund

Call:
lm(formula = abundance ~ years, data = sim$abund)

Residuals:
    Min      1Q  Median      3Q     Max
-77.487 -16.566   0.277  17.042  72.513

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.830e+03  5.249e+01  -111.1   <2e-16 ***
years        2.922e+00  2.602e-02   112.3   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.4 on 98008 degrees of freedom
Multiple R-squared:  0.114, Adjusted R-squared:  0.114
F-statistic: 1.261e+04 on 1 and 98008 DF,  p-value: < 2.2e-16
```

Challenge!

Now try changing the module yourself. Add, for example, the function below to the **existing** event named `plot`, plotting the data in the first (2013) and the last (2022) years (the answer can be found here):

```
plotAbundance <- function(abundanceData, yearsToPlot){
  Sys.sleep(1.2) # To ensure we will see the results from the previous plot
  dataplot <- abundanceData[years %in% yearsToPlot,]
  abundData <- Copy(dataplot)
  abundData[, years := as.factor(years)]
  abundData[, averageYear := mean(abundance), by = "years"]
  pa <- ggplot(data = abundData, aes(x = abundance, group=years, color=years, fill = years)) +
    geom_histogram(binwidth=5) +
    facet_grid(years ~ .) +
    geom_vline(data = unique(abundData[, c("years", "averageYear")]),
               aes(xintercept = averageYear),
               linetype="dashed", color = "black") +
    theme(legend.position = "none")
  print(pa)
  return(pa)
}
```