SticsRPacks: R packages for STICS, where are we?

Samuel Buis[1], *Patrice Lecharpentier*[2], *Rémi Vezy*[3], *Michel Giner*[4], *Cyrille Midingoyi*[4]

[1] INRAE, UMR EMMAH, Avignon, France, samuel.buis@inrae.fr

[2] INRAE, US Agroclim, Avignon, France, patrice.lecharpentier@inrae.fr

[3] CIRAD, UMR AMAP, Montpellier, France

[4] CIRAD, UPR AIDA, Montpellier, France

XIIIth Seminar of developers and users of the soil-plant model STICS
Aerocampus Aquitaine Bordeaux Nov. 13-16, 2023

Overview

SticsRPacks, a collection of R packages:

- **SticsOnR** and **SticsRFiles:** packages for managing STICS from R
  - Finding parameters/variables names (description)
  - Setting parameters values and output variables
  - Handling input and output files
  - Running simulations (parallel, forcing parameters, etc.)

- **CroptimizR** and **CroPlotR: generic packages** for coupling crop models with mathematical methods
  - Parameter estimation (Bayesian / frequentist, choice of criterion, consideration of constraints, AgMIP protocols, etc.)
  - Plots and statistical criteria

Overview

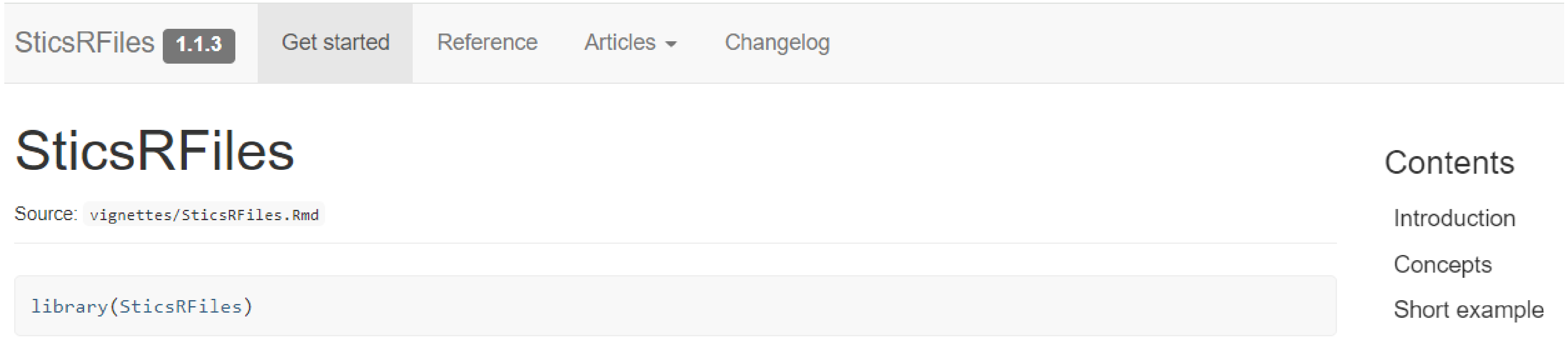
Availability:

- Online git repositories on GitHub platform
- SticsRFiles on the CRAN
- All in one installation using SticsRPacks package
- L-GPL License

Compatibility with STICS versions: version 8.5.0 and following

Website for each package e.g. https://sticsrpacks.github.io/SticsRFiles/articles/SticsRFiles.html

Online tutorial



SticsRFiles

Source: `vignettes/SticsRFiles.Rmd`

```
library(SticsRFiles)
```

UseCase I: Multi-simulation workflow

Generating usms data files (SticsRFiles)

existing xml workspace (JavaStics)

Xml workspace generation from tables of parameters (using xml templates or not)

gen_*_xml(table_*)

gen_xml2txt(workspace)

STICS input files (individual directories)

run_stics(stics_exe, workspace, usms),
sim <- **stics_wrapper**(model_options, parameters, usms, variables, dates)

Running multi-simulations (SticsOnR)

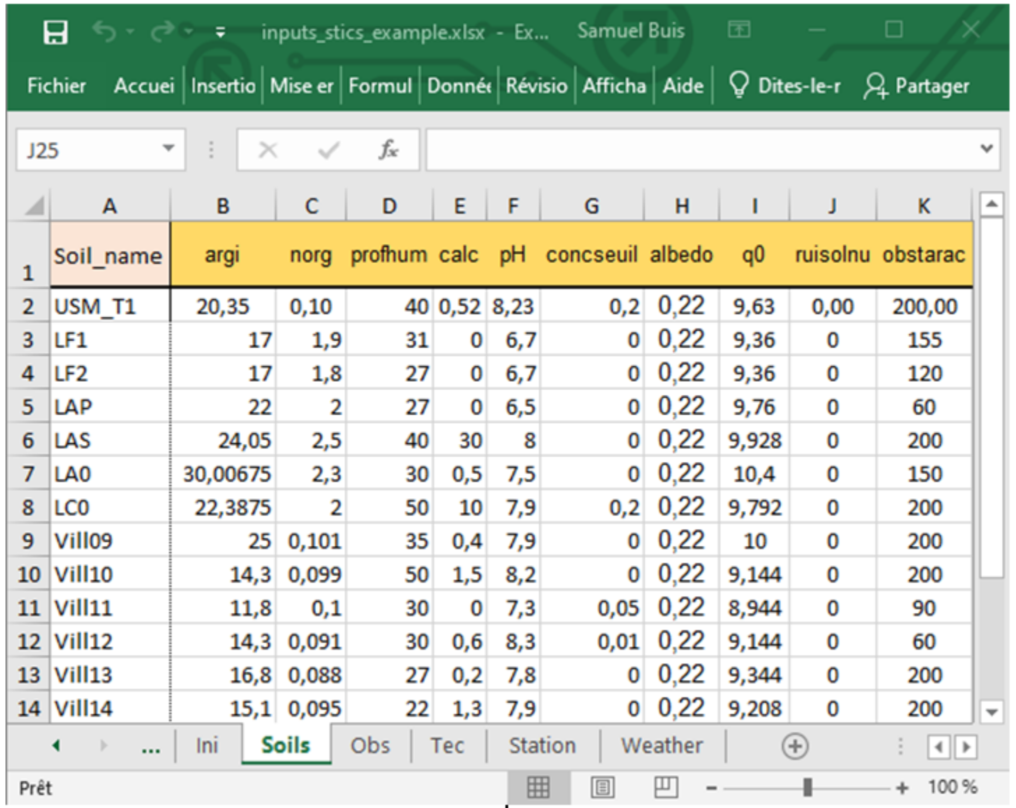
- independent, chained
- for single crops or inter-crops

plot(sim[, obs])
summary(sim[, obs])

Plotting / analysing outputs (CroPlotR)

- dynamic plots
- prediction evaluations / experimental data
- statistical criteria

Xml workspace generation from tables of parameters


inputs_stics_example.xlsx - Ex... Samuel Buis
Fichier Accuei Insertio Mise er Formul Donné Révisio Afficha Aide Dites-le-r Partager
J25
Ini Soils Obs Tec Station Weather
Prêt
100 %

	A	B	C	D	E	F	G	H	I	J	K
1	Soil_name	argi	norg	profhum	calc	pH	concseuil	albedo	q0	ruisolnu	obstarac
2	USM_T1	20,35	0,10	40	0,52	8,23	0,2	0,22	9,63	0,00	200,00
3	LF1	17	1,9	31	0	6,7	0	0,22	9,36	0	155
4	LF2	17	1,8	27	0	6,7	0	0,22	9,36	0	120
5	LAP	22	2	27	0	6,5	0	0,22	9,76	0	60
6	LAS	24,05	2,5	40	30	8	0	0,22	9,928	0	200
7	LA0	30,00675	2,3	30	0,5	7,5	0	0,22	10,4	0	150
8	LC0	22,3875	2	50	10	7,9	0,2	0,22	9,792	0	200
9	Vill09	25	0,101	35	0,4	7,9	0	0,22	10	0	200
10	Vill10	14,3	0,099	50	1,5	8,2	0	0,22	9,144	0	200
11	Vill11	11,8	0,1	30	0	7,3	0,05	0,22	8,944	0	90
12	Vill12	14,3	0,091	30	0,6	8,3	0,01	0,22	9,144	0	60
13	Vill13	16,8	0,088	27	0,2	7,8	0	0,22	9,344	0	200
14	Vill14	15,1	0,095	22	1,3	7,9	0	0,22	9,208	0	200

Excel/CSV file(s) or R data.frames describing parameters values

Templates of JavaSTICS files

usms.xml, *_ini.xml, sols.xml, *_tec.xml, *_sta.xml, *.year

`gen_usms_xml, gen_sol_xml, gen_tec_xml, …`

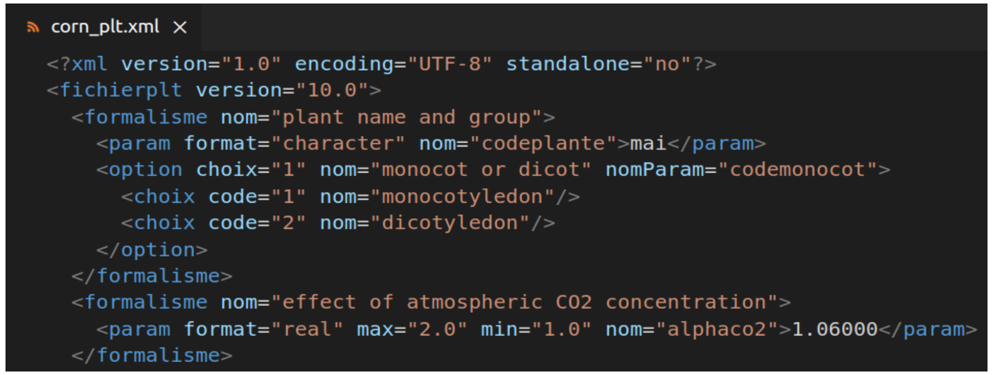
$\downarrow$

XML JavaStics input files

usms.xml, *_ini.xml, sols.xml, *_tec.xml, *_sta.xml, *.year

Generate txt files from XML without using JavaSTICS

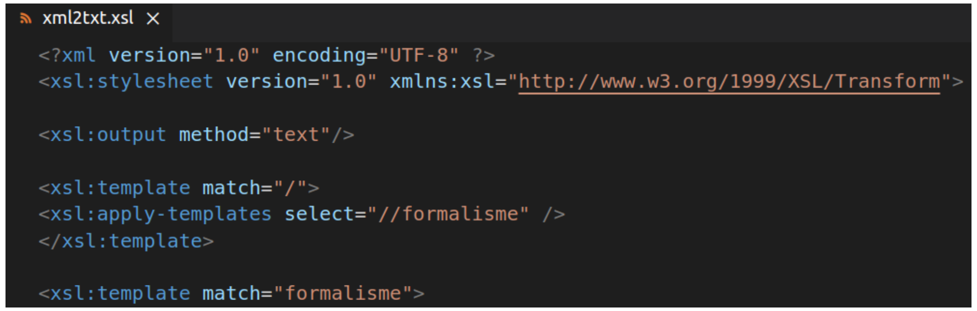
xml file



corn_plt.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<fichierplt version="10.0">
  <formalisme nom="plant name and group">
    <param format="character" nom="codeplante">mai</param>
    <option choix="1" nom="monocot or dicot" nomParam="codemonocot">
      <choix code="1" nom="monocotyledon"/>
      <choix code="2" nom="dicotyledon"/>
    </option>
  </formalisme>
  <formalisme nom="effect of atmospheric CO2 concentration">
    <param format="real" max="2.0" min="1.0" nom="alphaco2">1.06000</param>
  </formalisme>
```

xml style file (transformation rules)



xml2txt.xsl

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="text"/>

<xsl:template match="/">
<xsl:apply-templates select="//formalisme" />
</xsl:template>

<xsl:template match="formalisme">
```

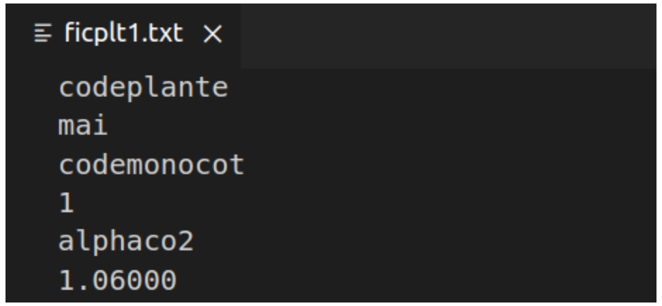
Generation of all usm input files, including xml tranformations

- At least 4 times faster than JavaStics conversion
- Available soon on CRAN, already on Github

gen_usms_xml2txt(workspace)

uses xslt R package

text file



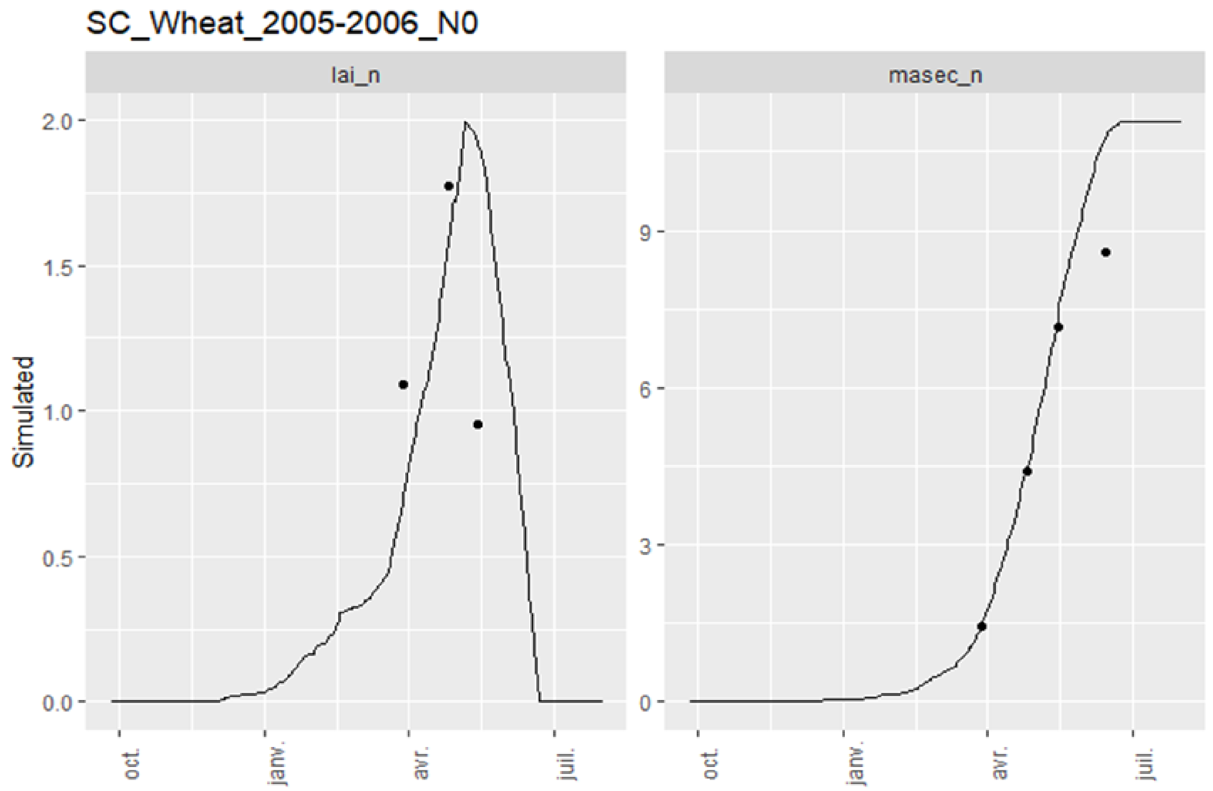
ficplt1.txt

```
codeplante
mai
codemonocot
1
alphaco2
1.06000
```

Plots and stats: new package CroPlotR

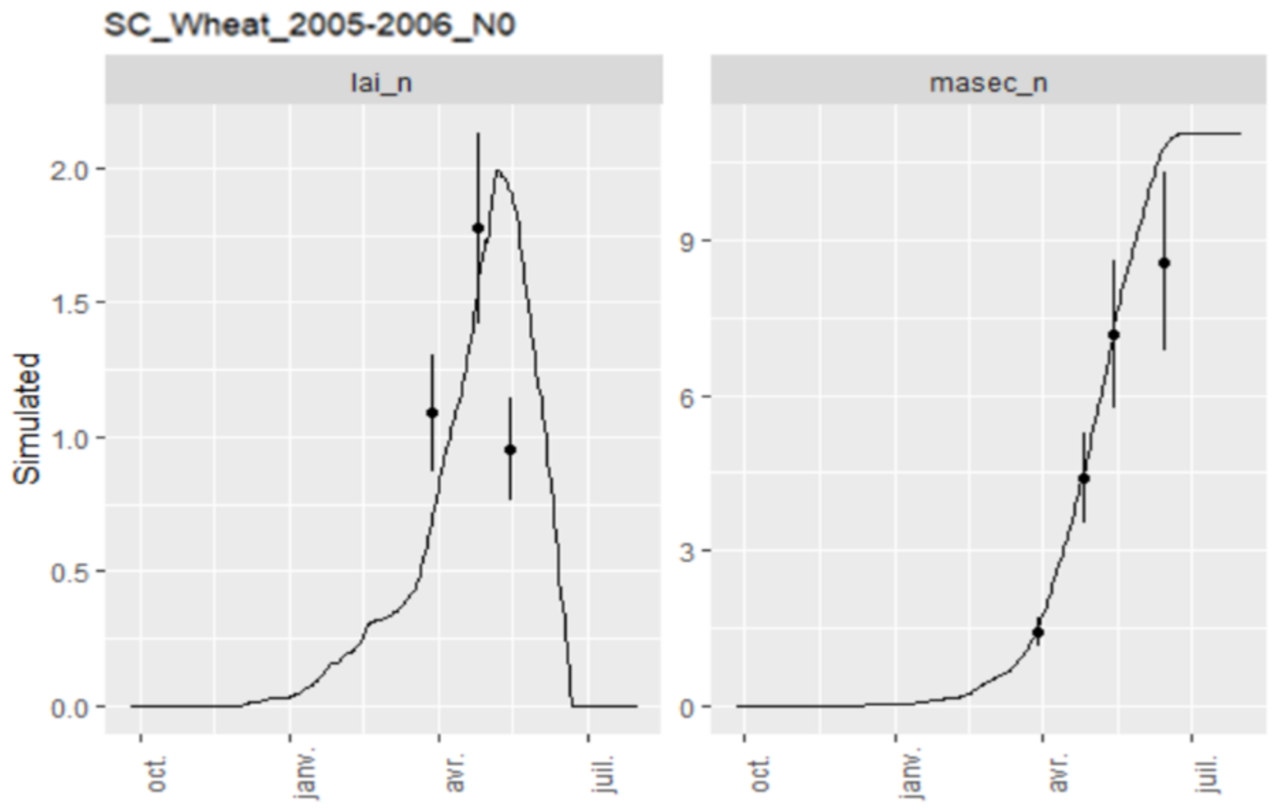
```
sim <- model_wrapper(model_options, sit_names, …)
```

```
ggplot_list <-  plot(sim, obs= obs_list)
```



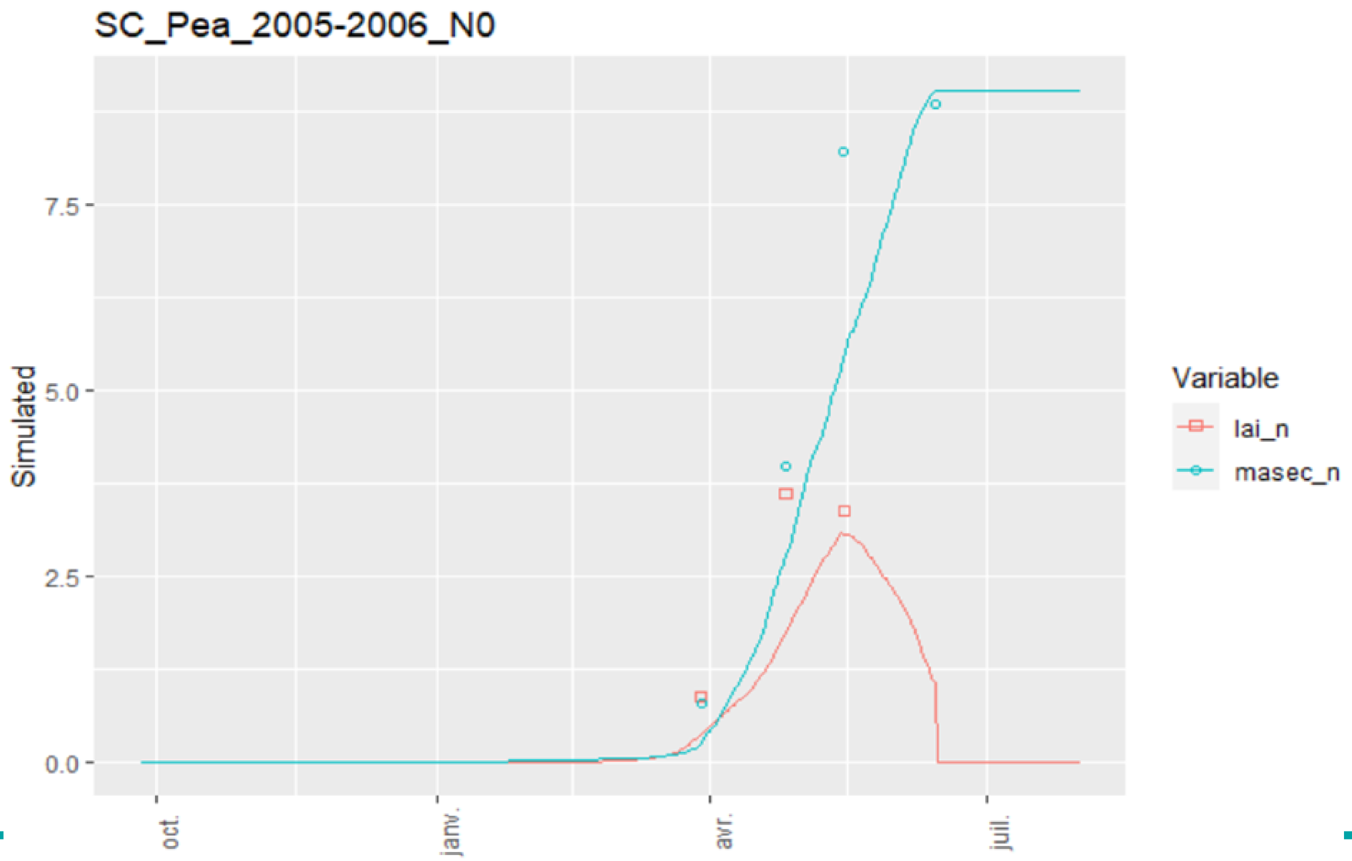
Plots and stats: new package CroPlotR

```
plot(sim, obs= obs_list, obs_sd=obs_sd)
```



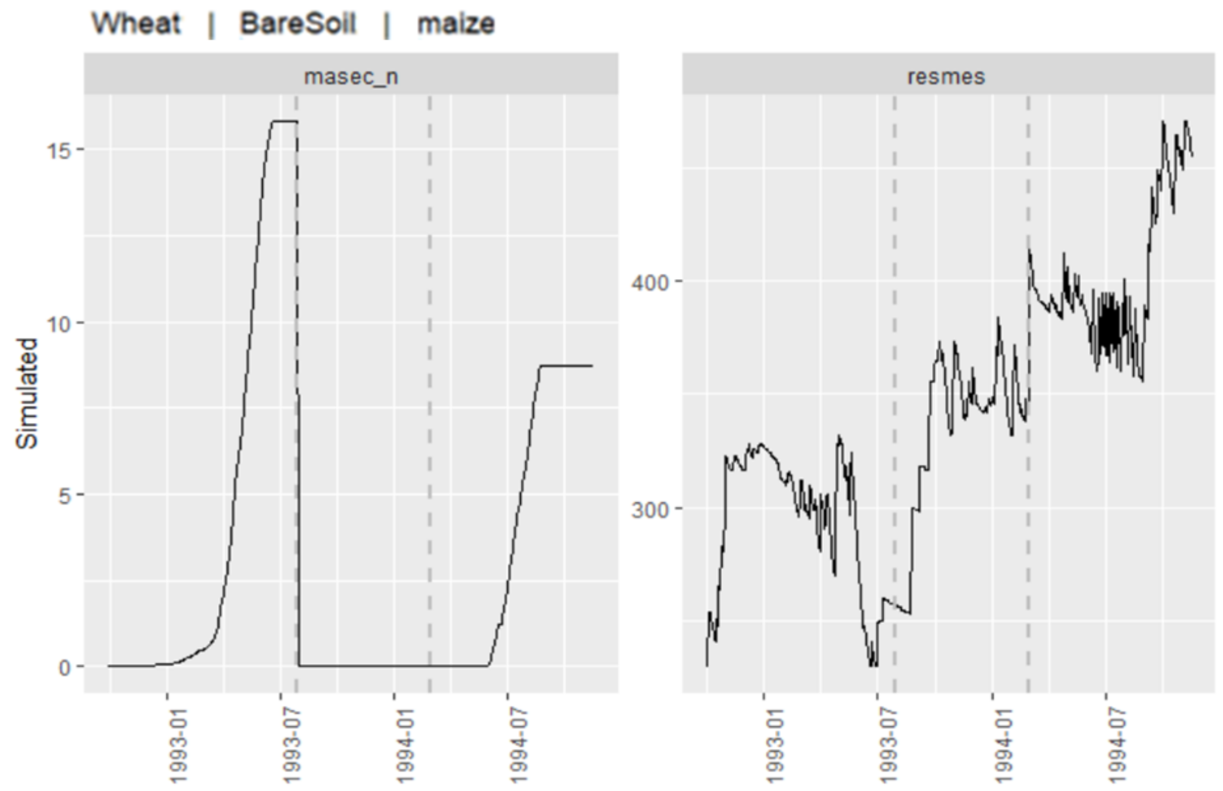
Plots and stats: new package CroPlotR

```
plot(sim, obs= obs_list, overlap = list(list("lai_n","masec_n")))
```



Plots and stats: new package CroPlotR

```
plot(sim, obs= obs_list, successive = list(list("Wheat","BareSoil","maize")))
```

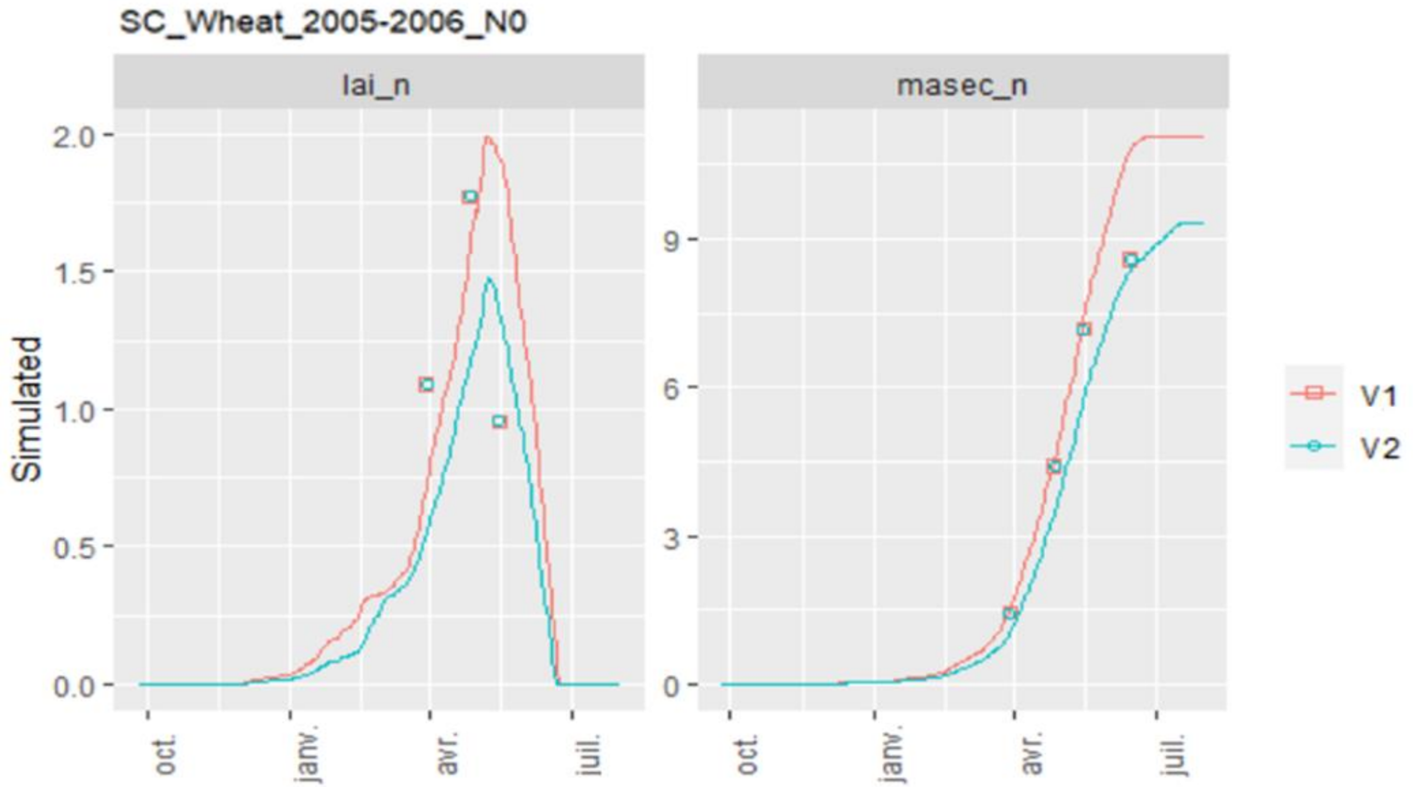


Plots and stats: new package CroPlotR

```
sim1 <- model_wrapper(param_values1, model_options, sit_names, ...)
```

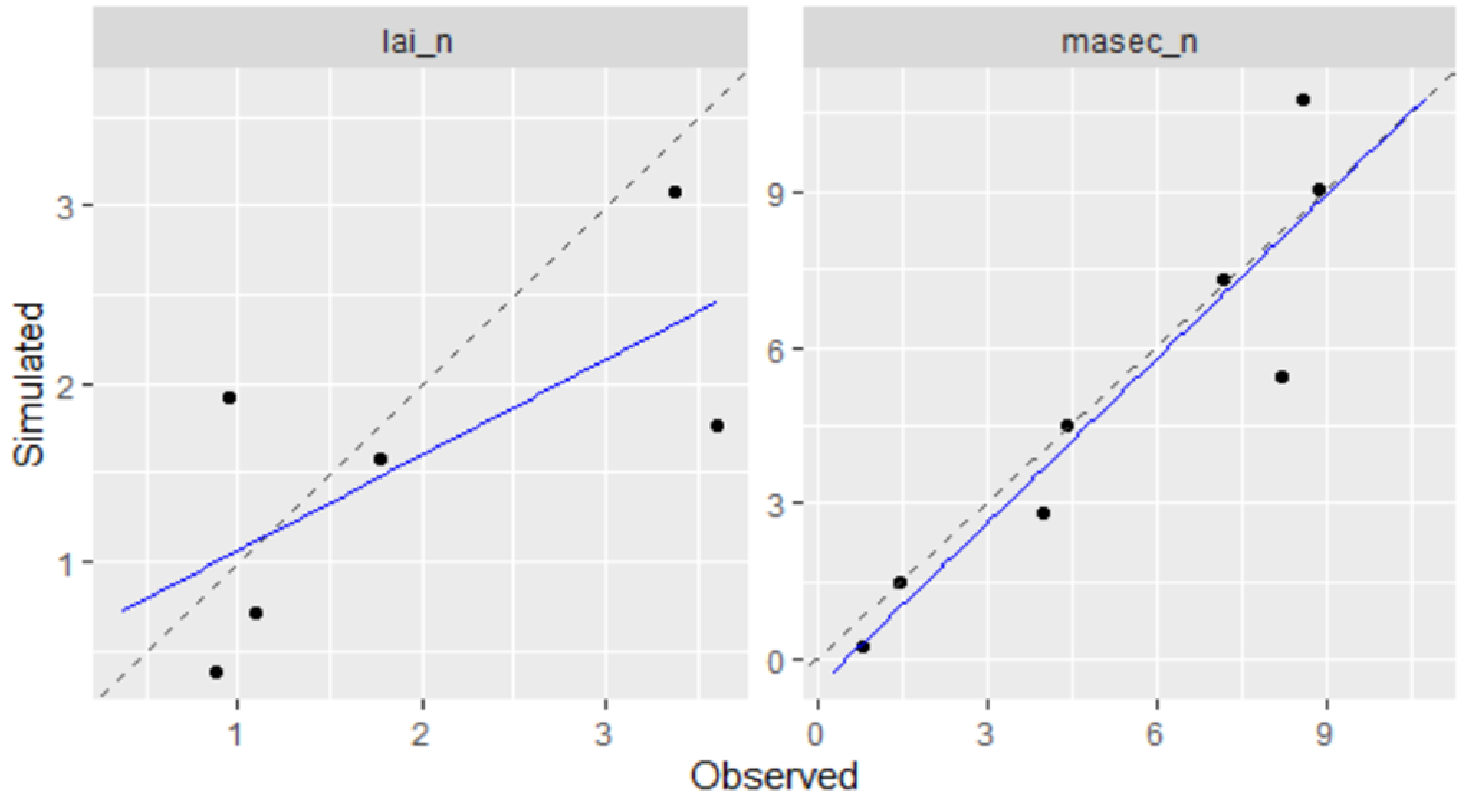
```
sim2 <- model_wrapper(param_values2, model_options, sit_names, ...)
```

```
plot(V1=sim1, V2=sim2, obs= obs_list)
```



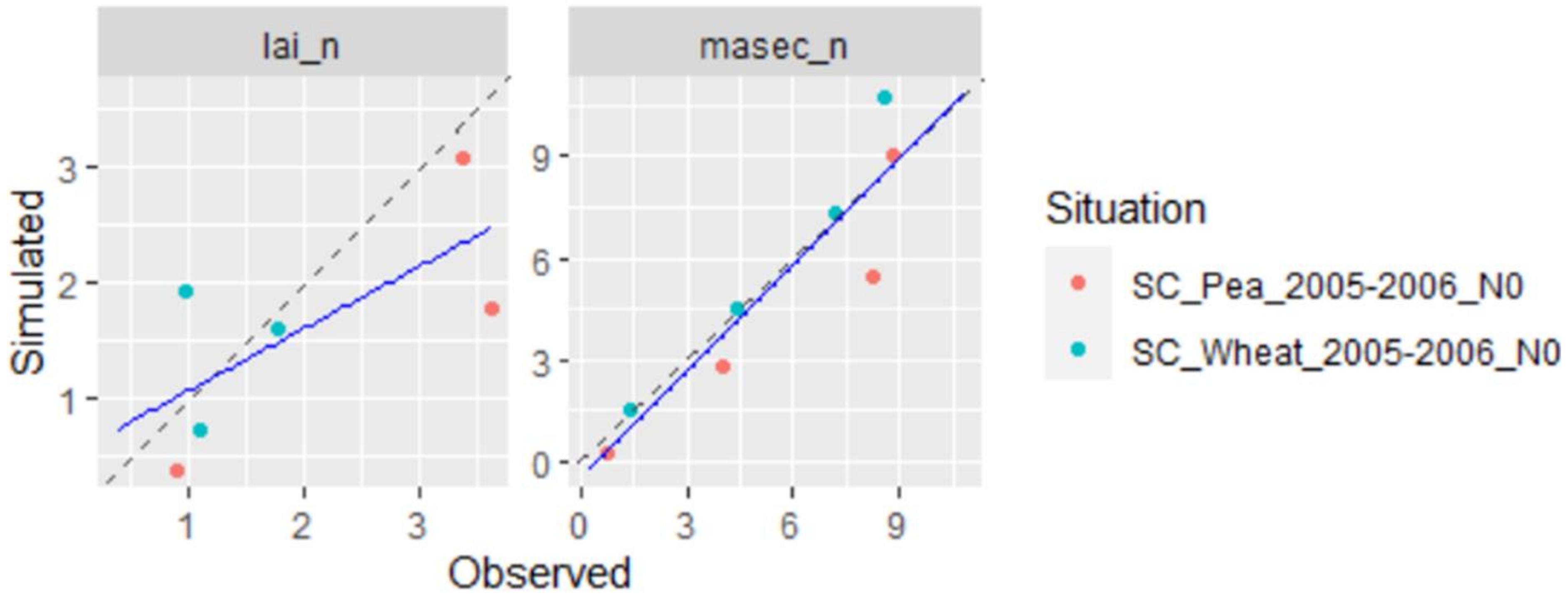
Plots and stats: new package CroPlotR

```
plot(sim, obs= obs_list, type = "scatter")
```



Plots and stats: new package CroPlotR

```
plot(sim, obs= obs_list, type = "scatter", shape_site="symbol")
```

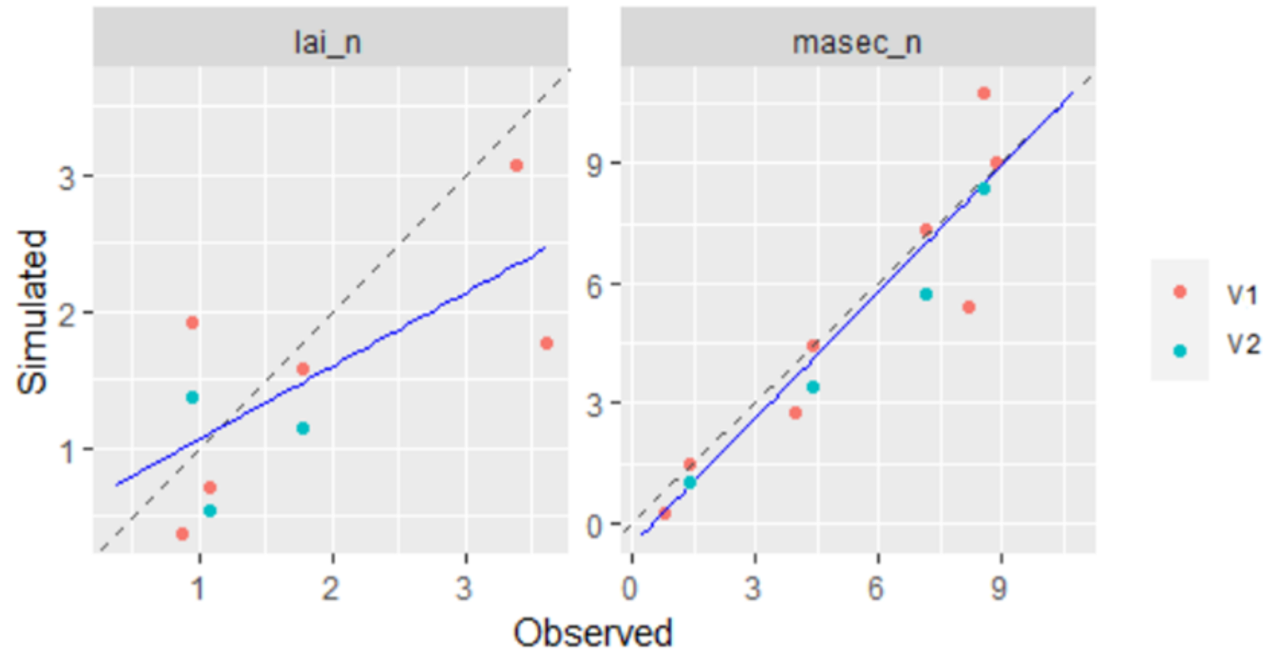


Plots and stats: new package CroPlotR

```
sim1 <- model_wrapper(param_values1, model_options, sit_names, ...)

     sim2 <- model_wrapper(param_values2, model_options, sit_names, ...)

plot(V1=sim1, V2=sim2, obs= obs_list, type = "scatter")
```



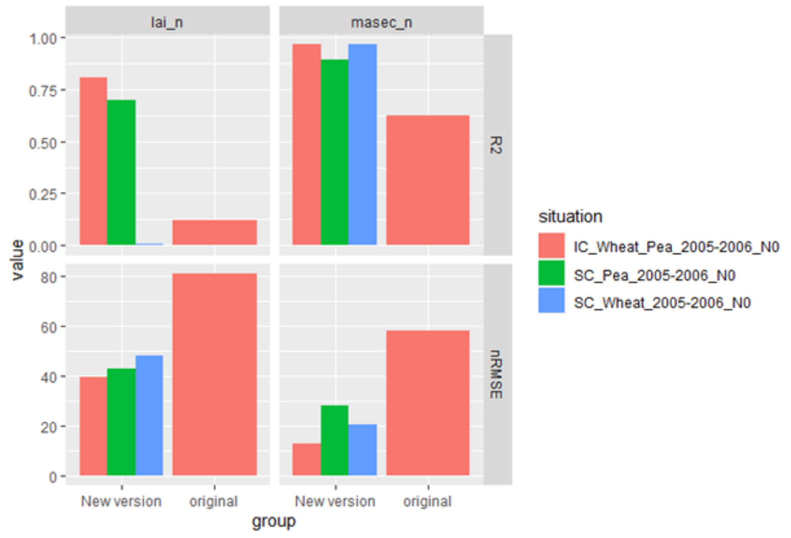
Plots and stats: new package CroPlotR

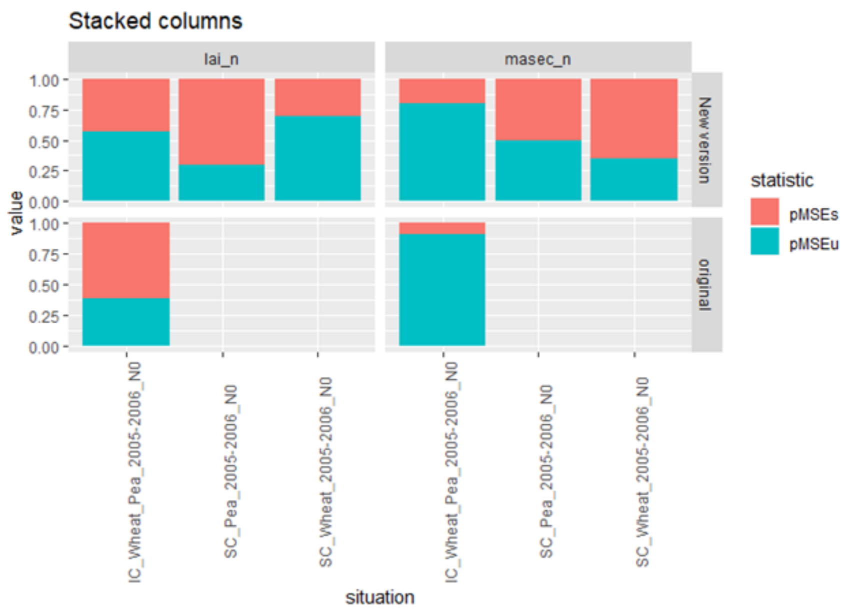
```
stats <- summary(sim, obs= obs_list)
```

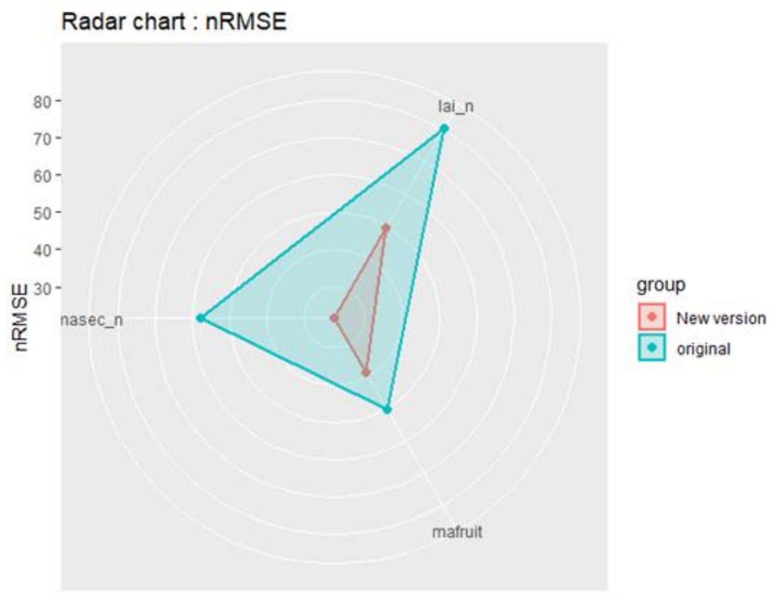
group	situation	variable	n_obs	mean_obs	mean_sim	r_means	sd_obs	sd_sim	CV_obs	CV_sim	R2	SS_res	Inter	Slope	RMSE	RMSEs	RMSEu
Version_1	all_situations	lai_n	14	1.270476	1.023683	80.57474	1.085427	0.8698547	85.43465	84.97306	0.6923882	5.57851	0.1764799	0.6668389	0.6312408	0.4270089	0.4648960
Version_1	all_situations	masec_n	18	4.326759	4.142749	95.74715	2.710052	2.9836105	62.63468	72.02007	0.8994510	16.06896	-0.3749370	1.0441269	0.9448386	0.2176378	0.9194312

…

```
plot(stats)
```

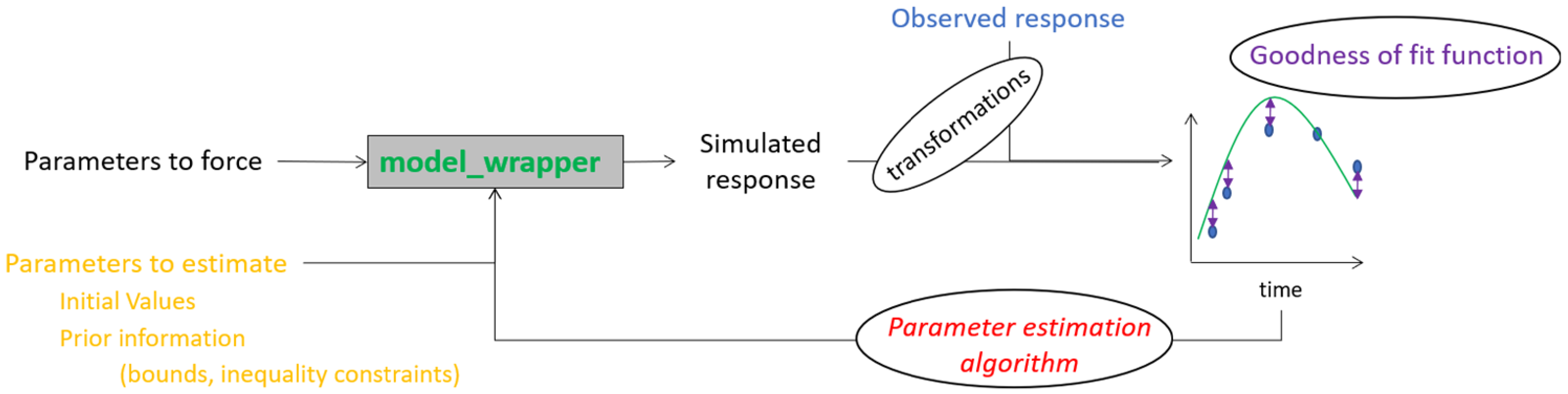






UseCase II: parameter estimation

estim_param(param_info, model_wrapper, model_options, obs_list, crit_function, optim_method, optim_options, transform_sim/obs, …)

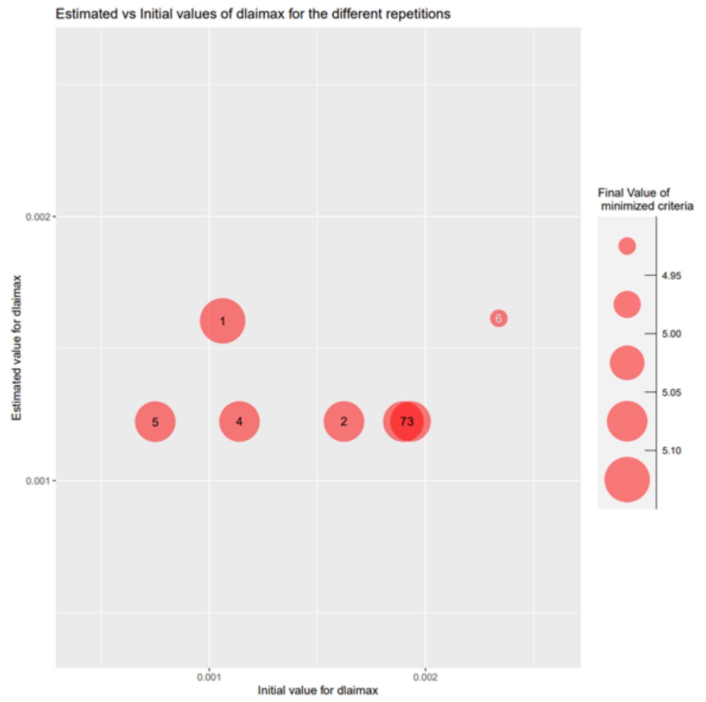


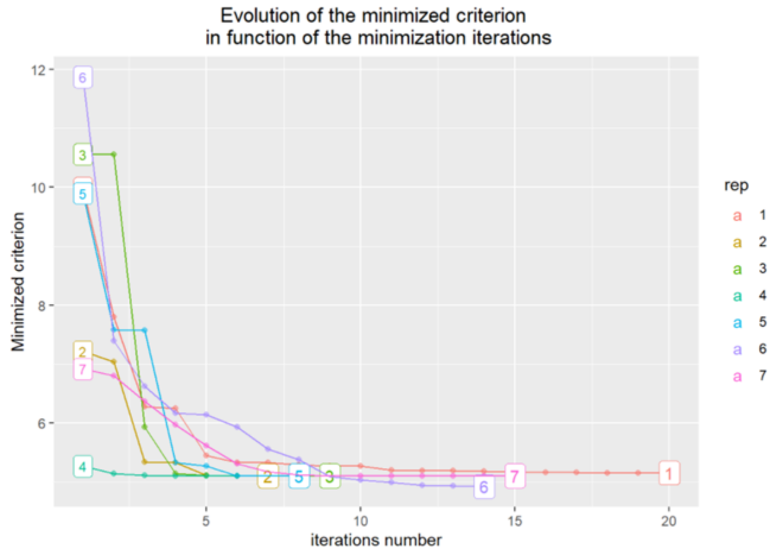
Parameter estimation: automatic generation of diagnostics

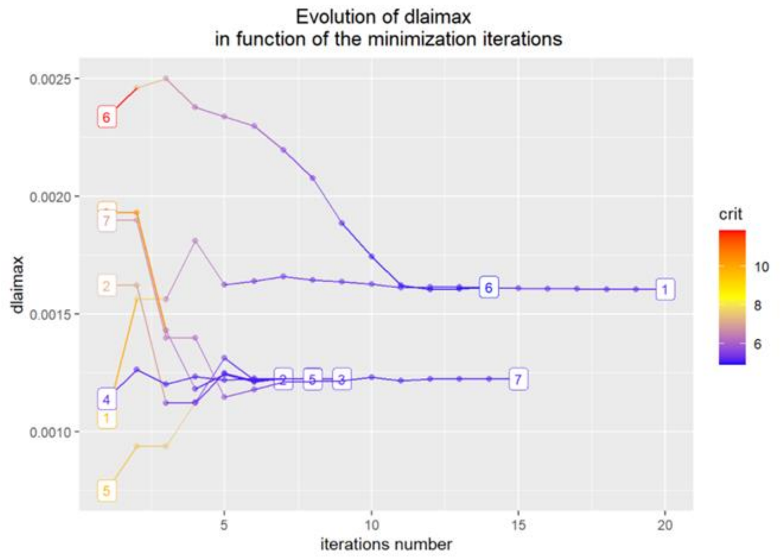
Plots, screen display and returned data structure specific for each family of method (frequentist, bayesian ...)

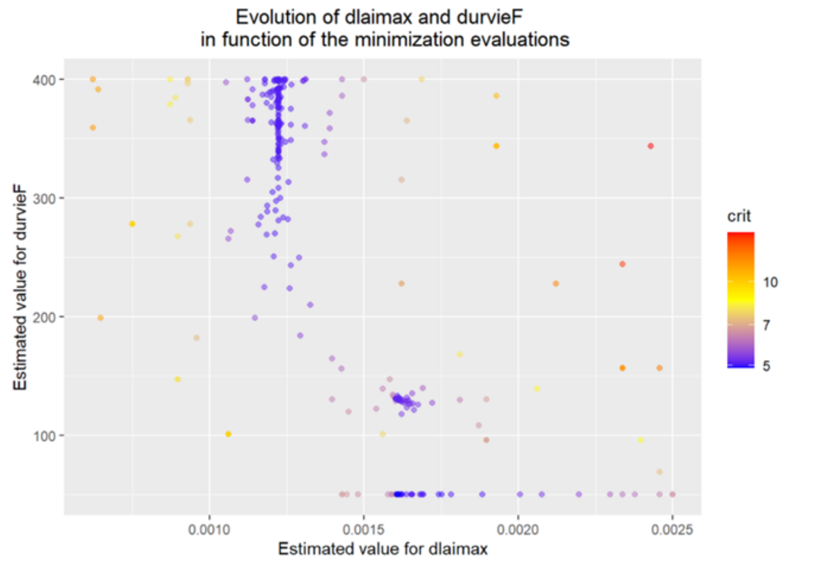
e.g. for frequentist methods:

```
⋮
Estimated value for dlaimax :  0.0016
Estimated value for durvieF :  50
Minimum value of the criterion: 4.9
Complementary graphs and results can be found in folder D:\Home\sbuis\Documents\TMP
Average time for the model to simulate all required situations:  3.89 sec elapsed
Total number of criterion evaluation: 431
Total time of model simulations:  1678.23 sec elapsed
Total time for parameter estimation process: 1682.75 sec elapsed
```







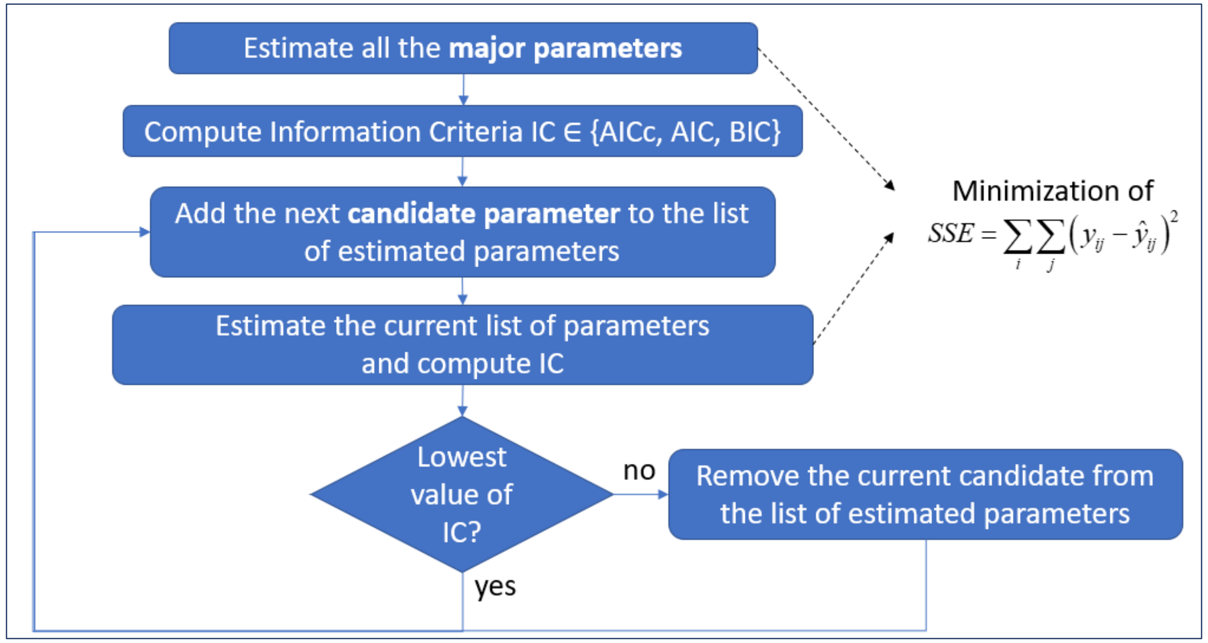


Parameter estimation: automatic selection of parameters

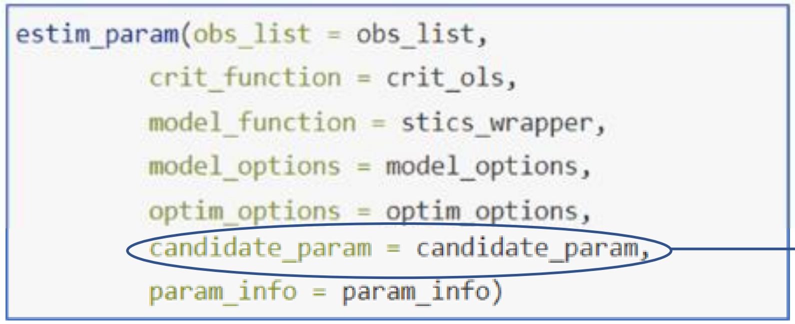
Some parameters must be estimated => **major parameters**

Some parameters may improve the predictions
=> **candidate parameters**

=> automatic selection among candidate parameters
(AgMIP-Calibration Phase III parameter selection algorithm, Wallach et al., 2023, ASD)


Estimate all the major parameters
Compute Information Criteria IC ∈ {AICc, AIC, BIC}
Add the next candidate parameter to the list of estimated parameters
Estimate the current list of parameters and compute IC
Lowest value of IC?
no
yes
Remove the current candidate from the list of estimated parameters
Minimization of

$$SSE = \sum_i \sum_j \left(y_{ij} - \hat{y}_{ij}\right)^2$$



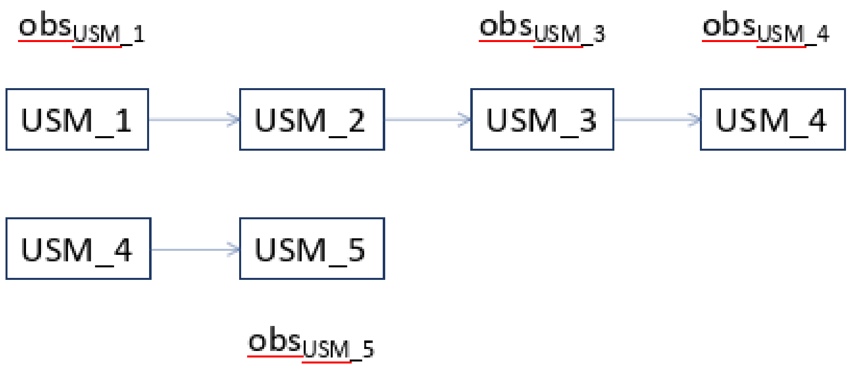
```
estim_param(obs_list = obs_list,
        crit_function = crit_ols,
        model_function = stics_wrapper,
        model_options = model_options,
        optim_options = optim_options,
        candidate_param = candidate_param,
        param_info = param_info)
```

Names of the parameters, among those defined in the argument param_info, that must only be considered as candidate for parameter estimation (see details section).
e.g. c(“tdmin”,”stressdev”,”tdmax”)

see documented example in vignette “Estimating phenology following AgMIP-calibration phase III protocol”

Parameter estimation: handling successive USMs and different parameter values for the USMs

Example of a parameter estimation for 3 parameters on 2 successions of USMS



Standard definition for the observations

```
obs_list$USM_1 <- data.frame(...)
obs_list$USM_3 <- data.frame(...)
...
```

Definition of successions in STICS model_options

```
successive <- list(c("USM_1", "USM_2", "USM_3", "USM_4"),
                   c("USM_4","USM_5"))

model_options <- stics_wrapper_options(javastics, workspace, successive)
```

Definition of groups of situations in parameter information

p1: one value for all USMs

p2: only used in USM_2

p3: one value for USM_3
and one value for USM_4

```
param_info$p1 <- list(sit_list = list(c("USM_1", "USM_2", "USM_3", "USM_4", "USM_5")),
                      lb = 0, ub = 100)

param_info$p2 <- list(sit_list = c("USM_3"),
                      lb = 0.1, ub = 0.5)

param_info$p3 <- list(sit_list = list(c("USM_3"),
                                      c("USM_4")),
                      lb = 1, ub = 10)
```

*estim_param(**param_info**, model_wrapper, **model_options**, **obs_list**, …)*

Conclusion

- Many improvements and new features since 2020
- Reached robust state
- Integrated into STICS trainings
- Widely used in research projects: e.g. AgMIP, CarSolEl, REDELAC, ANR FFAST, ...
- Used within other tools: e.g. SIWAA simulation workflow, see presentation Chabrier et al. ; IdeSTICS ; under work for STICS automatic test and evaluation system ...
- The generic packages (CroptimizR, CroPlotR) interfaced or integrated with other crop models: SQ2, APSIM nextGen, DSSAT (Linkage INRAE-UF) + many models in AgMIP Calibration project

Perspectives

- Optimize performances:
  - SticsRFiles: parallelization of XML->txt ; use of XML2 package, ...
  - SticsOnR: STICS simulations
  - CroPlotR: plot generation
- Add new features:
  - SticsRFiles: generalize the STICS input files conversion function
  - CroptimizR: integrate the AgMIP phase IV protocol
  - CroPlotR: integrate functions for diagnostics on model inputs
- Submit all the packages to the CRAN (done for SticsRFiles)