

Algorithm Configuration Survival Guide

Part 2

SPECIES Summer School 2023

Leslie Pérez Cáceres

leslie.perez@pucv.cl

What are we going to use?

Don't want to install anything?

- <http://www.replit.com>
- <http://rstudio.cloud>

If you want to work in your computer:

- R <https://cran.r-project.org>
- Rstudio <https://www.rstudio.com>

What are we going to use?

- The irace package <https://iridia.ulb.ac.be/irace/>
- ACOTSP <https://iridia.ulb.ac.be/dorigo/ACO/aco-code/public-software.html>
- Repository https://github.com/leslieperez/configuration_lab

Using replit

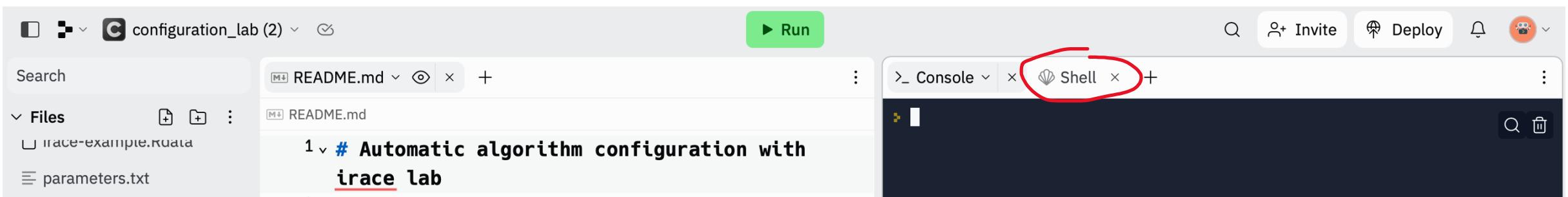
<http://www.replit.com>

The screenshot shows the Replit web interface. On the left, a sidebar lists various options: Home, Templates, My Repls, My Cycles (0), Bounties (\$61K+ open), Community, Learn, Themes, Extensions, and Teams. At the top, there's a search bar and a 'Create Repl' button, which is circled in red. Below the search bar, there are links for 'Upgrade' and 'Replit AI'. The main area is titled 'Home' and features a 'Leslie Angelica, you can now chat with' message. A central modal window is open, titled 'Import from GitHub', with the sub-tutorial 'Import from GitHub' underlined in red. The modal contains fields for 'GitHub URL' (set to 'leslieperez/configuration_lab') and 'Owner' ('LeslieAngelicaA'). It also includes a 'Language' dropdown set to 'C', which is highlighted with a yellow box. Other visible buttons include 'Create from Template', 'Manage the GitHub repos Replit can import.', 'Upgrade to make private', and a large blue 'Import from GitHub' button.

Prepare ACOTSP

Go to the shell:

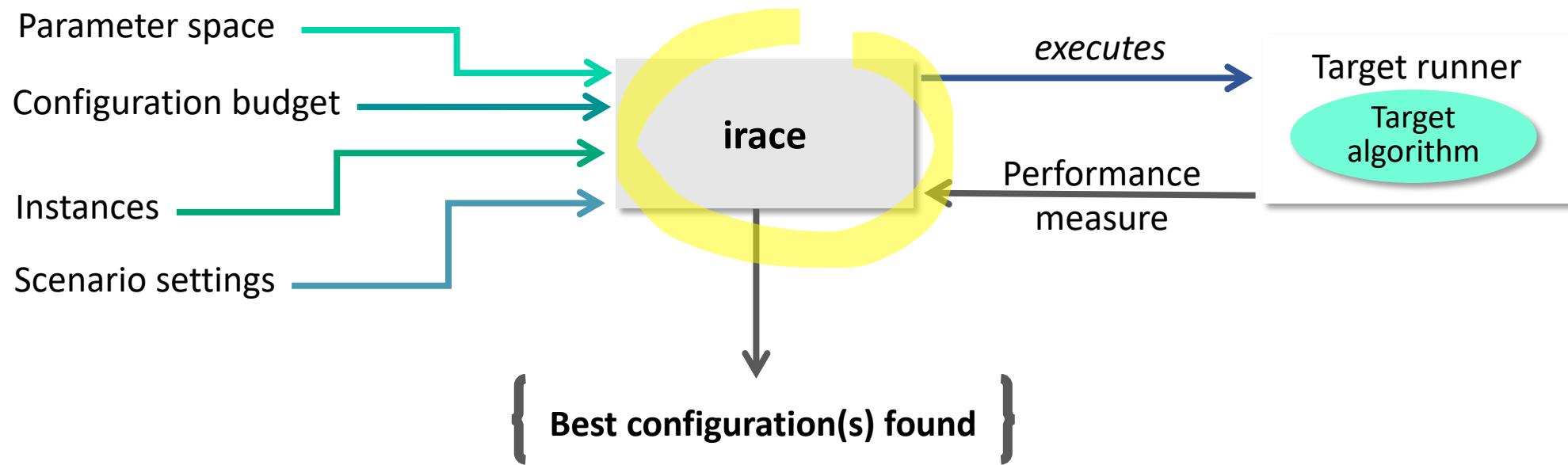
```
$ cd ACOTSP-1.03  
$ make
```



Lets create the configuration setup

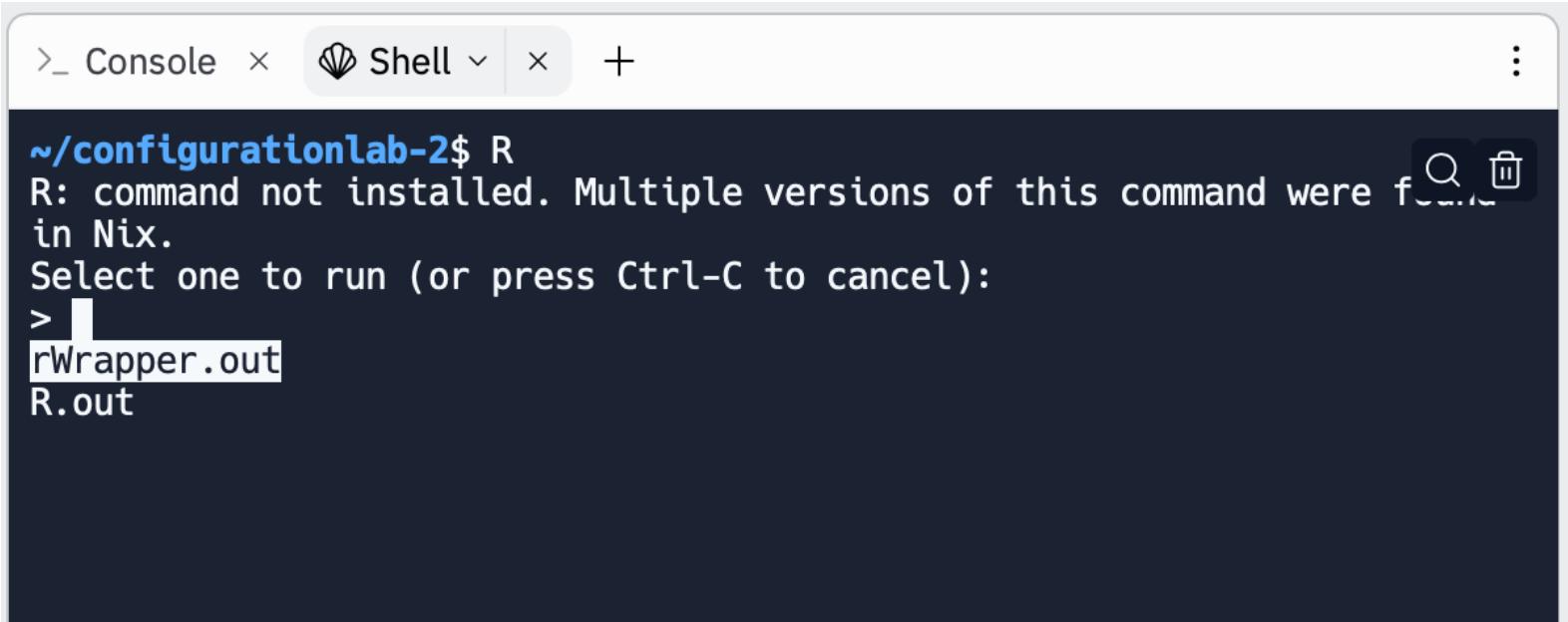
Configuration scenario setup

1



Open the R console in the shell

```
$ R
```



The screenshot shows a terminal window with a light gray header bar. In the header, there are tabs for "Console" (with an "x" icon), "Shell" (with a gear icon), and a "+" button. To the right of the tabs is a three-dot menu icon. The main area of the terminal is dark gray and contains the following text:

```
~/configurationlab-2$ R
R: command not installed. Multiple versions of this command were found in Nix.
Select one to run (or press Ctrl-C to cancel):
> rWrapper.out
R.out
```

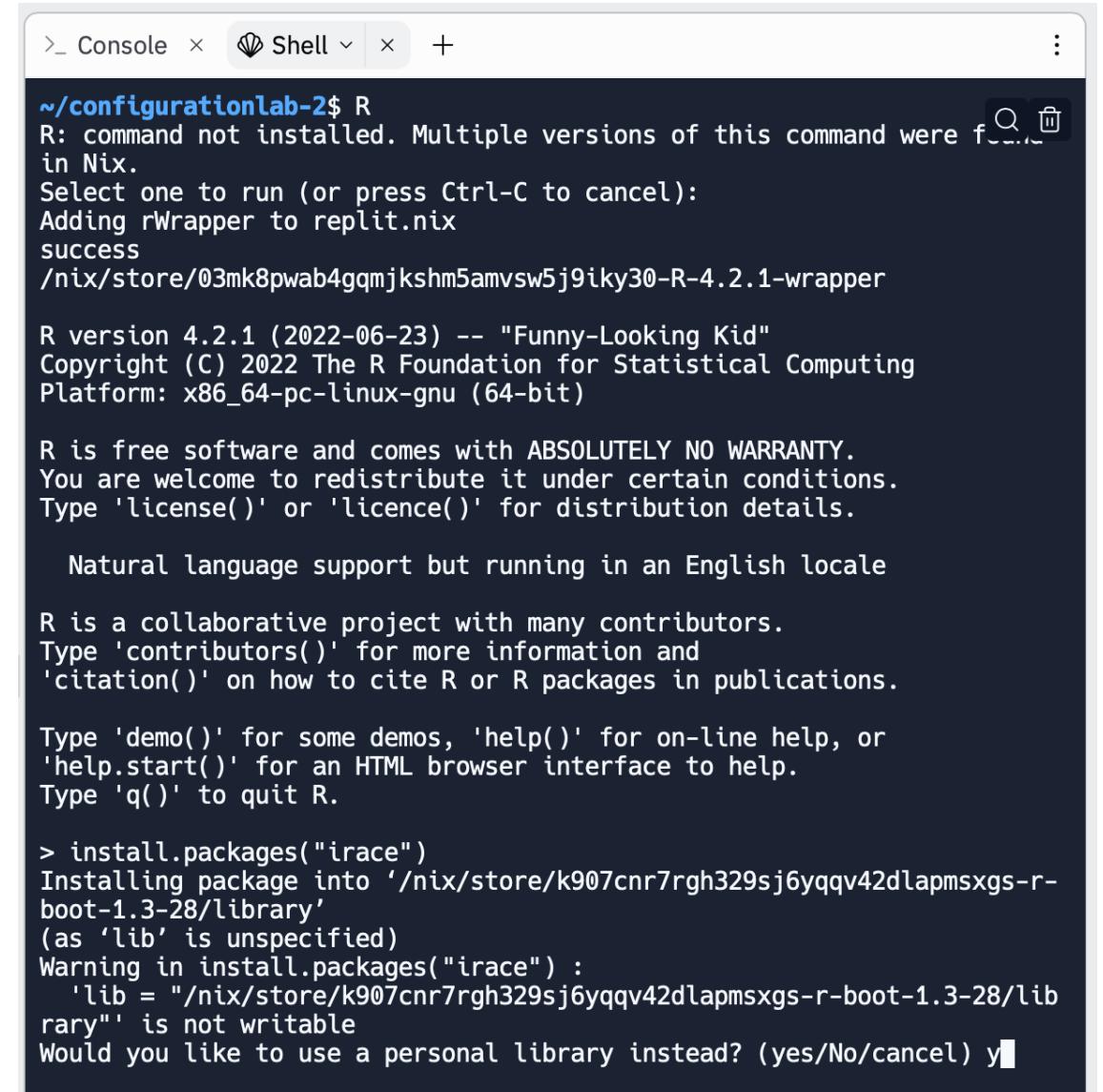
The text is white on a dark background. The "rWrapper.out" option is currently selected, indicated by a small white square at its start. There are also small icons for search and trash in the top right corner of the terminal window.

Installing irace

Install irace

```
> install.packages("irace")
```

Install irace in the local library



The screenshot shows a terminal window titled 'Console' with the shell icon selected. The command `R` is entered, resulting in an error message: 'R: command not installed. Multiple versions of this command were found in Nix.' A prompt asks to select one to run or cancel. The user selects the 'rWrapper' option from the list. The output shows the successful download of the R wrapper from the Nix store, followed by the standard R startup message, including the version (4.2.1), copyright information, and license details. It also mentions natural language support and the R collaborative project. Finally, it prompts the user to install the 'irace' package into the local library.

```
~/configurationlab-2$ R
R: command not installed. Multiple versions of this command were found in Nix.
Select one to run (or press Ctrl-C to cancel):
Adding rWrapper to replit.nix
success
/nix/store/03mk8pwab4gqmjkshm5amvsw5j9iky30-R-4.2.1-wrapper

R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("irace")
Installing package into '/nix/store/k907cnr7rgh329sj6yqqv42dlapmsxgs-r-boot-1.3-28/library'
(as 'lib' is unspecified)
Warning in install.packages("irace") :
  'lib = "/nix/store/k907cnr7rgh329sj6yqqv42dlapmsxgs-r-boot-1.3-28/library"' is not writable
Would you like to use a personal library instead? (yes/No/cancel) y
```

Installing irace

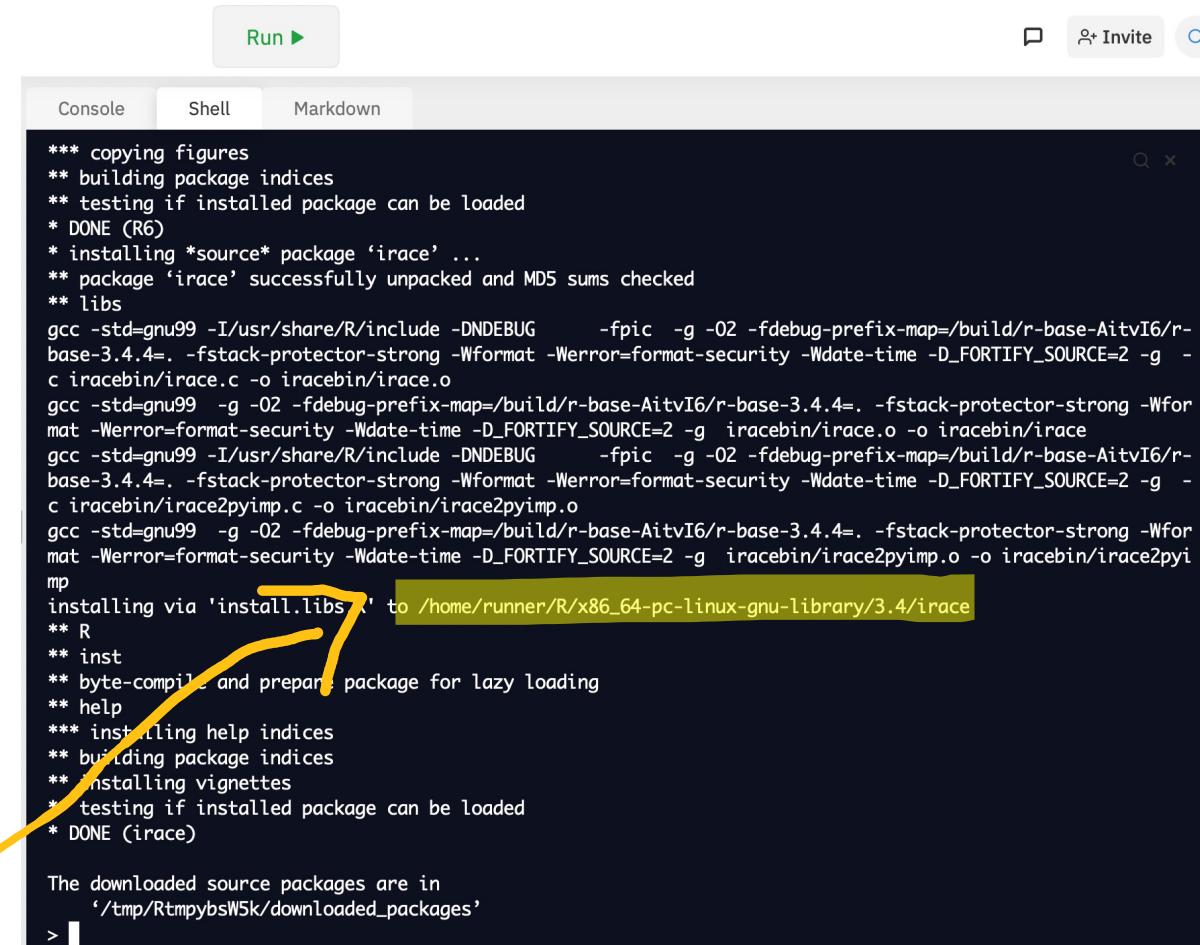
You can use irace from the **R console** or the **shell**.

- In the **R console** just load irace

```
> R  
> library("irace")
```

- In the **shell** use the executable

```
$ {irace_folder}/bin/irace
```



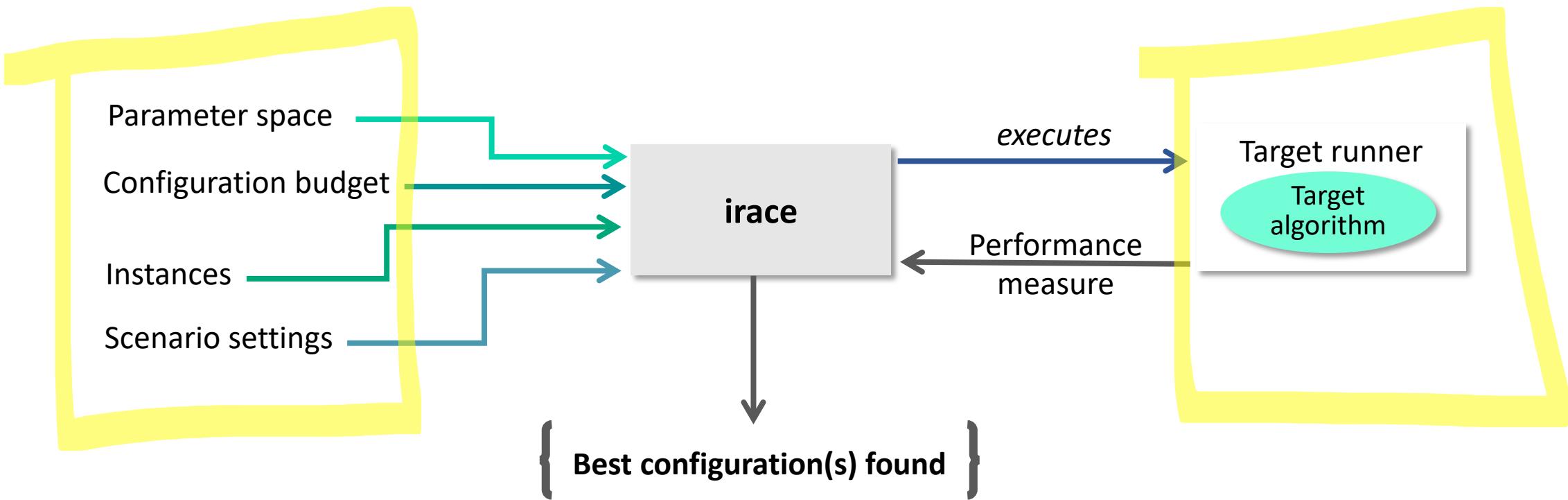
```
Run ▶  
Console Shell Markdown  
*** copying figures  
** building package indices  
** testing if installed package can be loaded  
* DONE (R6)  
* installing *source* package 'irace' ...  
** package 'irace' successfully unpacked and MD5 sums checked  
** libs  
gcc -std=gnu99 -I/usr/share/R/include -DNDEBUG -fpic -g -O2 -fdebug-prefix-map=/build/r-base-AitvI6/r-base-3.4.4=. -fstack-protector-strong -Wformat -Werror=format-security -Wdate-time -D_FORTIFY_SOURCE=2 -g -c iracebin/irace.c -o iracebin/irace.o  
gcc -std=gnu99 -g -O2 -fdebug-prefix-map=/build/r-base-AitvI6/r-base-3.4.4=. -fstack-protector-strong -Wformat -Werror=format-security -Wdate-time -D_FORTIFY_SOURCE=2 -g iracebin/irace.o -o iracebin/irace  
gcc -std=gnu99 -I/usr/share/R/include -DNDEBUG -fpic -g -O2 -fdebug-prefix-map=/build/r-base-AitvI6/r-base-3.4.4=. -fstack-protector-strong -Wformat -Werror=format-security -Wdate-time -D_FORTIFY_SOURCE=2 -g -c iracebin/irace2pyimp.c -o iracebin/irace2pyimp.o  
gcc -std=gnu99 -g -O2 -fdebug-prefix-map=/build/r-base-AitvI6/r-base-3.4.4=. -fstack-protector-strong -Wformat -Werror=format-security -Wdate-time -D_FORTIFY_SOURCE=2 -g iracebin/irace2pyimp.o -o iracebin/irace2pyimp  
installing via 'install.libs' to /home/runner/R/x86_64-pc-linux-gnu-library/3.4/irace  
** R  
** inst  
** byte-compile and prepare package for lazy loading  
** help  
*** installing help indices  
** building package indices  
** installing vignettes  
* testing if installed package can be loaded  
* DONE (irace)  
  
The downloaded source packages are in  
'/tmp/RtmpybsW5k/downloaded_packages'  
> |
```

You can create a symbolic link:

```
$ ln -s {irace_folder}/bin/irace .
```

Configuration scenario setup

2



Options of irace

Options can be provided by:

- scenario.txt
- Arguments (shell)
- Direct setting (in R)

Options have a **name** and a **flag**

Ej.

parameterFile (--parameter-file)

```
scenario.txt
#####
## mode: r -- Scenario setup for Iterated Race (iRace).
#####
## To use the default value of a parameter of iRace, simply do not set
## the parameter (comment it out in this file, and do not give any
## value on the command line).

## File that contains the description of the parameters.
parameterFile = "parameters.txt"
configurationsFile = "default.txt"
forbiddenFile = "forbidden.txt"

## Directory where the programs will be run.
execDir = "."

## Directory where tuning instances are located, either absolute path or
## relative to current directory.
trainInstancesDir = "../instances/small/training/"

## Set testing data
testInstancesDir = "../instances/small/testing/"
testNumElites = 5

## The maximum number of runs (invocations of targetRunner) that will be performed. It
## determines the (maximum) budget of experiments for the tuning.
maxExperiments = 300

## Number of CPU cores to use
parallel = 3

## A value of 0 silences all debug messages. Higher values provide
## more verbose debug messages.
# debugLevel = 0
```

Options of irace

The irace package user guide:

<https://cran.r-project.org/web/packages/irace/vignettes/irace-package.pdf>

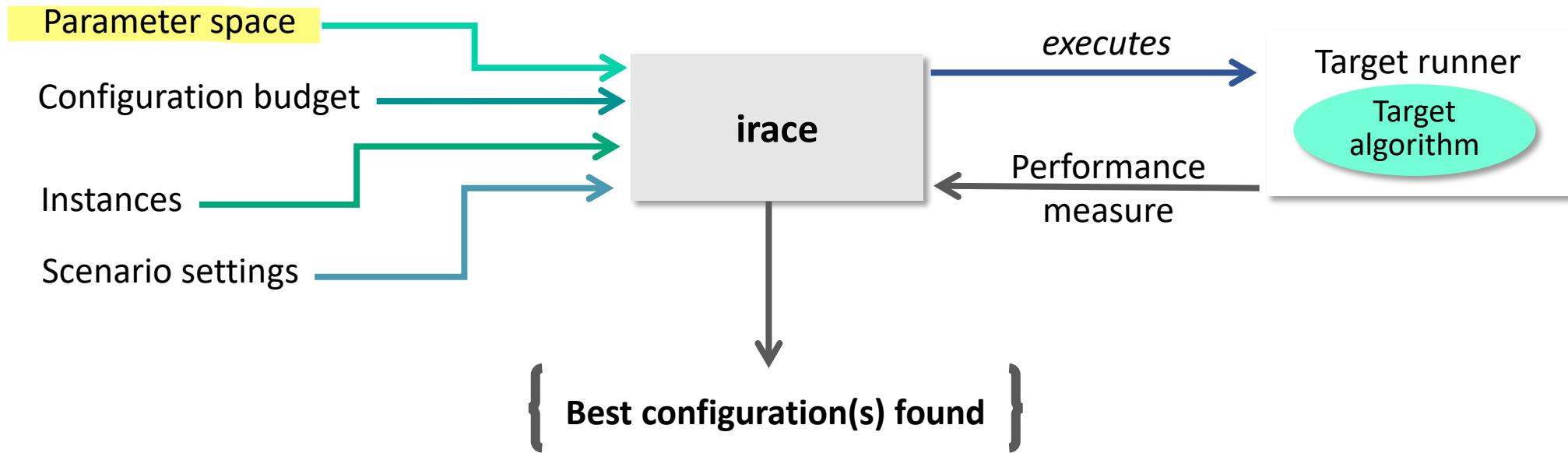
The irace package help:

```
> library("irace")
> irace.usage()
```

Or

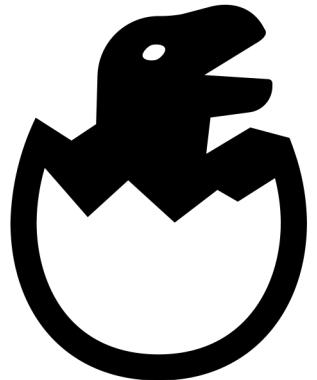
```
$ ./irace --help
```

Configuration scenario setup



Parameters space

1. List all parameters involved in your algorithm
 - Include also hidden ones (magical constants)
2. Identify conditional parameters
3. Define an appropriate type:
 1. Categorical
 2. Ordinal
 3. Numerical (real, integer, real log, integer log)
4. Define a **reasonable** domain for each parameter



components

behaviour

Parameters space

5. Check for configurations already known

- We can provide them to irace

6. Check for “issues” already known

- Parameter values combinations that are not desired
- Parameter values combinations that could cause problems



Segmentation fault?



Parameter space example

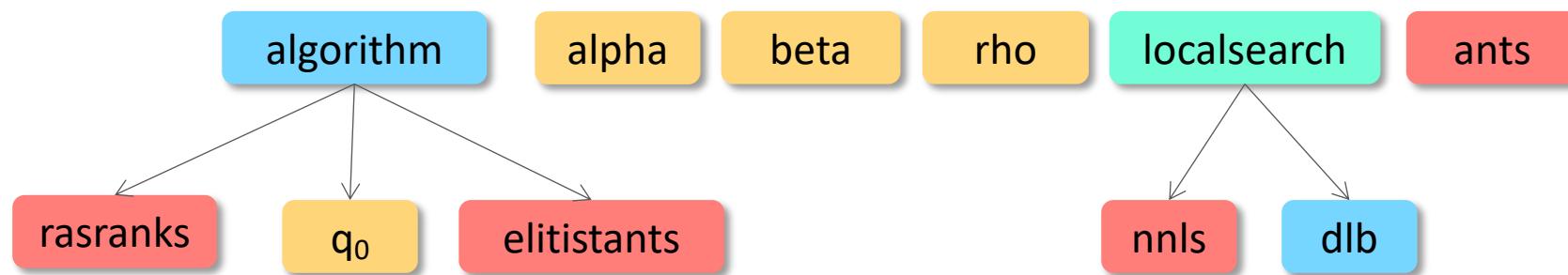
ACOTSP: Framework of Ant Colony Optimization algorithms for TSP (Thomas Stutzle)

<http://www.aco-metaheuristic.org/aco-code/public-software.html>

Which algorithms?

- Ant System
- Elitist Ant System
- Max-min Ant System
- Rank based Ant System
- Best-worst Ant System
- Ant Colony System

Parameter space example



Parameter space example

`parameters.txt`: describes the parameters to be configured

#	name	switch	type	values	[conditions (using R syntax)]
	algorithm	--"	c	(as,mmas,eas,ras,acs)	
	localsearch	--localsearch "	c	(0, 1, 2, 3)	
	alpha	--alpha "	r	(0.00, 5.00)	
	beta	--beta "	r	(0.00, 10.00)	
	rho	--rho "	r	(0.01, 1.00)	
	ants	--ants "	i	(5, 100)	
	nlls	--nlls "	i	(5, 50)	localsearch %in% c(1, 2, 3)
	q0	--q0 "	r	(0.0, 1.0)	algorithm %in% c("acs")
	dlb	--dlb "	c	(0, 1)	localsearch %in% c(1,2,3)
	rasrank	--rasranks "	i	(1, 100)	algorithm %in% c("ras")
	elitistsants	--elitistsants "	i	(1, 750)	algorithm %in% c("eas")

Parameter space example

default.txt: default parameter values or known configurations

```
algorithm localsearch alpha beta rho ants nnls dlb q0 rasrank elitists
as      0          1.0   1.0  0.95 10    NA   0    NA   NA     NA
```

Values must comply with conditionality conditions

Parameter space example

forbidden.txt: forbidden combinations of parameter values

```
## Examples of valid logical operators are: == != >= <= > < & | ! %in%
(alpha > 2.0) && (beta > 3.0)
```

Warning

You might be suspecting that irace could be used like a...

... debugging tool

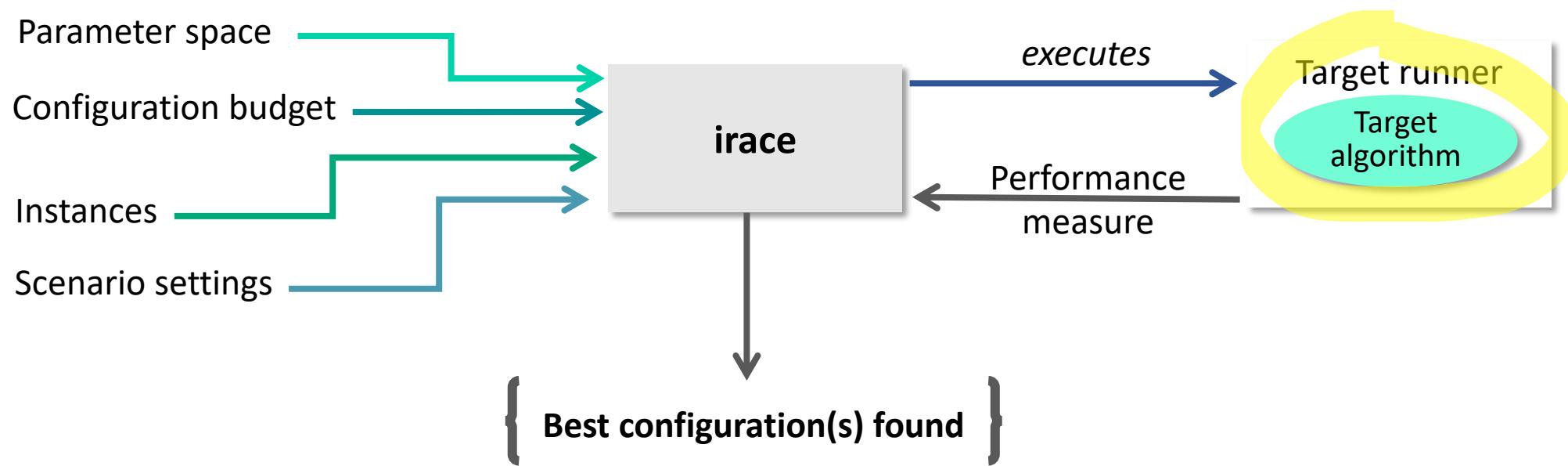


Yes!

For better or worse

Configuration scenario setup

4



Target algorithm execution

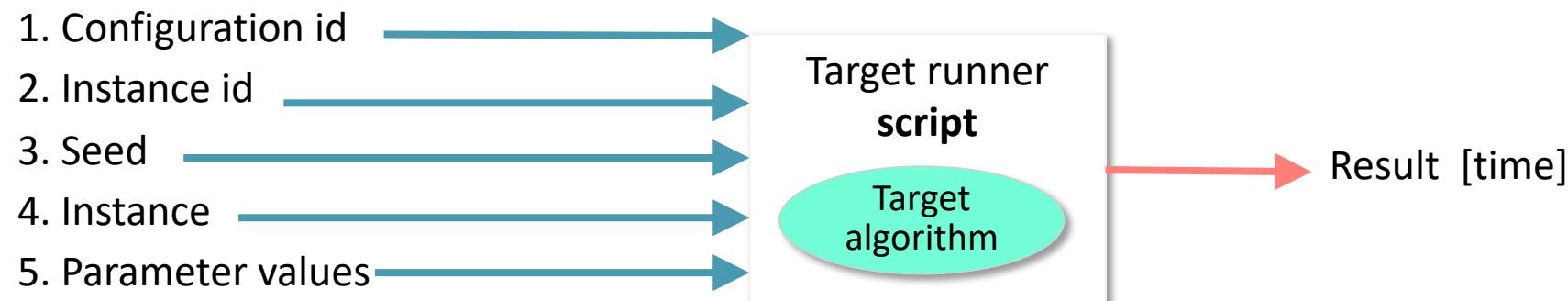
The target runner is a script that allows irace to **execute** the target algorithm

Options:

- Script file
 - Use file targetRunner in the folder
- R function
 - Use function provided in scenario.txt

Target algorithm execution: script file

The script can be provided in any programming language



Target algorithm: script file

```
targetRunner (--target-runner)
```

```
#!/bin/bash

# Path to the ACOTSP software
EXE=ACOTSP-1.03/acotsp

# Read experiment information
CONFIG_ID="$1"
INSTANCE_ID="$2"
SEED="$3"
INSTANCE="$4"

shift 4 || error "Not enough parameters"
CONFIG_PARAMS=$*

# Fixed parameters that should be always passed to ACOTSP
FIXED_PARAMS=" --tries 1 --time 5 --quiet "
```

Target algorithm: script file

```
targetRunner (--target-runner)
```

```
# File to write the execution output
STDOUT=c${CONFIG_ID}-${INSTANCE_ID}-${SEED}.stdout
STDERR=c${CONFIG_ID}-${INSTANCE_ID}-${SEED}.stderr

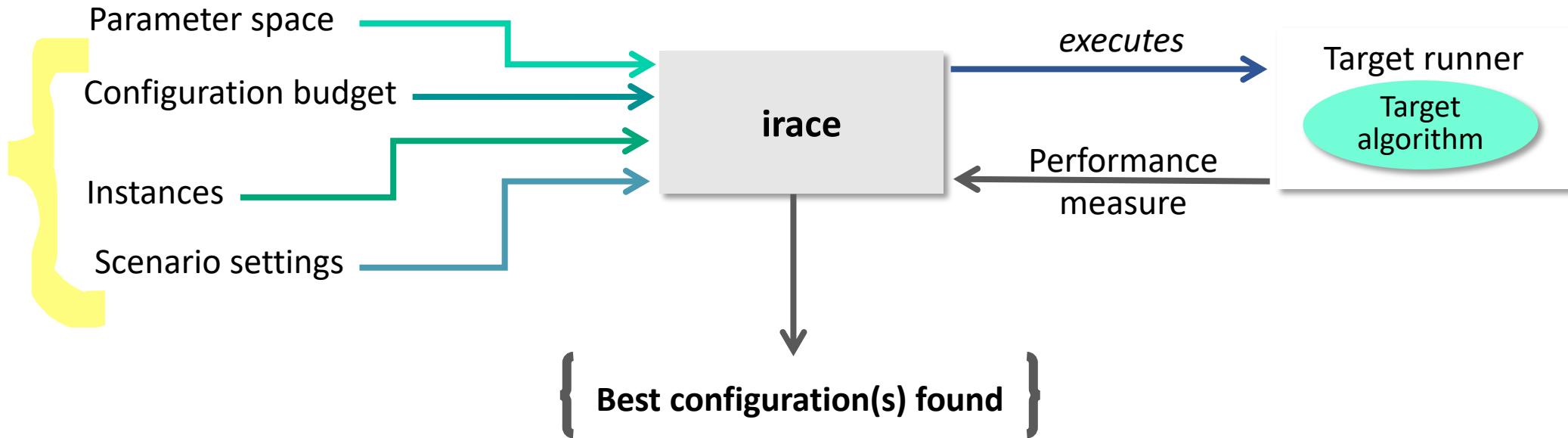
# Now we can call ACOTSP
$EXE ${FIXED_PARAMS} -i $INSTANCE --seed $SEED \
${CONFIG_PARAMS} 1> $STDOUT 2> $STDERR

# The following line is to extract the result
COST=$(cat ${STDOUT} | grep -o -E 'Best [+-0-9.e]+' | cut -d ' '
-f2)

# Print it!
echo "$COST"
```

Configuration scenario setup

5



Check the `scenario.txt` file where the options are set

Configuration budget

Configuration budget can be specified as:

- Number of evaluations: `maxEvaluations` (`--max-evaluations`)
 - Number of target runner executions allowed
- Time: `maxTime` (`--max-time`)
 - Max time allowed (in seconds)
 - You must provide the time as output in target runner!

Instance set

Instances can be provided in:

- **Folder:** `trainInstancesDir` (`--train-instances-dir`)
 - Files inside the folder will be used as instances
- **File:** `trainInstancesFile` (`--train-instances-file`)
 - Lines in the file will be used as instances
- Test instances can be provided in same way

`testInstancesDir` (`--test-instances-dir`)

`testInstancesFile` (`--test-instances-file`)

Scenarios settings

There are many options inside irace!

- `capping` (`--capping`): enables adaptive capping for runtime configuration
- `testType` (`--test-type`): selects the statistical test used for elimination
- `deterministic` (`--deterministic`): set the target algorithm as deterministic
- `parallel` (`--parallel`): set the number of cores to be used in the evaluation

Executing irace

From the R console:

```
> parameters = readParameters("parameters.txt")
> scenario = readScenario("scenario.txt")
> irace(scenario = scenario, parameters=parameters)
```

From the shell:

```
$ irace --scenario scenario.txt
```

or

```
$ Rscript run-irace.R
```

Executing irace

Run ▶

Console Shell Markdown

```
# Read 1 configuration(s) from file '/home/runner/configurationlab-1/automatic/default.txt' x
# 2021-07-05 23:29:22 UTC: Initialization
# Elitist race
# Elitist new instances: 1
# Elitist limit: 2
# nbIterations: 5
# minNbSurvival: 5
# nbParameters: 8
# seed: 1770898493
# confidence level: 0.95
# budget: 300
# mu: 5
# deterministic: FALSE

# 2021-07-05 23:29:22 UTC: Iteration 1 of 5
# experimentsUsedSoFar: 0
# remainingBudget: 300
# currentBudget: 60
# nbConfigurations: 10
# Markers:
    x No test is performed.
    c Configurations are discarded only due to capping.
    - The test is performed and some configurations are discarded.
    = The test is performed but no configuration is discarded.
    ! The test is performed and configurations could be discarded but elite configurations are preserved.
    . All alive configurations are elite and nothing is discarded

+---+-----+-----+-----+-----+-----+-----+
| | Instance | Alive | Best | Mean best | Exp so far | W timel | rhoKenWI | Qvarl |
+---+-----+-----+-----+-----+-----+-----+-----+
```

During the execution

```
# 2021-07-05 22:36:50 -03: Iteration 1 of 5
# experimentsUsedSoFar: 0
# remainingBudget: 300
# currentBudget: 60
# nbConfigurations: 10
# Markers:
  x No test is performed.
  c Configurations are discarded only due to capping.
  - The test is performed and some configurations are discarded.
  = The test is performed but no configuration is discarded.
  ! The test is performed and configurations could be discarded but elite configurations are preserved.
  . All alive configurations are elite and nothing is discarded

[+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| | Instance| Alive| Best| Mean best| Exp so far| W time| rho|KenW| Qvar|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|x|     1|    10|    10| 2773.00000|      10|00:00:23|   NA|   NA|   NA|
|x|     2|    10|    10| 26539.00000|      20|00:00:24|+0.98|0.99|0.0109|
|x|     3|    10|    10| 36006.00000|      30|00:00:21|+0.98|0.98|0.0169|
|x|     4|    10|    10| 34559.00000|      40|00:00:20|+0.95|0.96|0.0219|
|-|     5|     1|    10| 36432.00000|      50|00:00:20|   NA|   NA|   NA|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Best-so-far configuration:          10      mean value: 36432.00000
Description of the best-so-far configuration:
  .ID. algorithm localsearch alpha beta rho ants q0 rasrank elitistsants
10  10      mmas          0 1.3508 2.2476 0.6061   58 NA       NA       NA
  .PARENT.
10      NA

# 2021-07-05 22:38:40 -03: Elite configurations (first number is the configuration ID; listed from best to worst according to the sum of ranks):
  algorithm localsearch alpha beta rho ants q0 rasrank elitistsants
10      mmas          0 1.3508 2.2476 0.6061   58 NA       NA       NA
```



Current best configuration

Once the configuration budget is over...

```
# 2021-07-05 23:48:48 -03: Elite configurations (first number is the configuration ID; listed from best to worst according to the sum of ranks):
# 2021-07-05 23:48:48 -03: Elite configurations (first number is the configuration ID; listed from best to worst according to the sum of ranks):
algorithm localsearch alpha beta rho ants q0 rasrank elitists
43      acs      0 1.7807 6.9714 0.6217  29 0.8073     NA     NA
36      acs      0 1.8288 6.5449 0.6078  34 0.7417     NA     NA
45      acs      0 1.3133 5.7576 0.5361   8 0.5153     NA     NA
35      acs      0 1.5458 7.5555 0.1707  24 0.4302     NA     NA
26      acs      0 1.7294 8.1875 0.4038  18 0.4167     NA     NA
# 2021-07-05 23:48:48 -03: Stopped because budget is exhausted
# Iteration: 7
# nbIterations: 6
# experimentsUsedSoFar: 300
# timeUsed: 0
# remainingBudget: 0
# currentBudget: 29
# number of elites: 5
# nbConfigurations: 7
.ID. algorithm localsearch alpha beta rho ants q0 rasrank
43  43      acs      0 1.7807 6.9714 0.6217  29 0.8073     NA
36  36      acs      0 1.8288 6.5449 0.6078  34 0.7417     NA
45  45      acs      0 1.3133 5.7576 0.5361   8 0.5153     NA
35  35      acs      0 1.5458 7.5555 0.1707  24 0.4302     NA
26  26      acs      0 1.7294 8.1875 0.4038  18 0.4167     NA
elitists .PARENT. .ALIVE. .RANK. .WEIGHT.
43      NA      31    TRUE    37 0.33333333
36      NA      31    TRUE    37 0.26666667
45      NA      35    TRUE    48 0.20000000
35      NA      14    TRUE    49 0.13333333
26      NA       2    TRUE    50 0.06666667
```

Final elite configurations

We also perform the testing

```
# 2021-07-06 00:00:27 -03: Testing configurations (in no particular order): 43 36 45 35 26
  algorithm localsearch alpha   beta   rho ants   q0 rasrank elitists
43      acs        0 1.7807 6.9714 0.6217  29 0.8073     NA     NA
36      acs        0 1.8288 6.5449 0.6078  34 0.7417     NA     NA
45      acs        0 1.3133 5.7576 0.5361   8 0.5153     NA     NA
35      acs        0 1.5458 7.5555 0.1707  24 0.4302     NA     NA
26      acs        0 1.7294 8.1875 0.4038  18 0.4167     NA     NA
# Testing of elite configurations: 5
# Testing iteration configurations: FALSE
# 2021-07-06 00:01:57 -03: Testing results (column number is configuration ID in no particular order):
        43      36      45      35      26
1t  12496  12352  12325  12359  12362
2t  313329 307856 313761 325329 312230
3t  49915  51216  51107  50300  51959
4t   9089   9148   9246   9257   9120
5t  39907  39134  39521  39349  41394
# 2021-07-06 00:01:57 -03: Finished testing
```



Testing results

Configuration data

- The configuration process produces a Rdata file (`irace.Rdata`)
 - It contains all performance data gathered by irace
- The data can be loaded in the R console

```
$ R  
> load("irace-example.Rdata")  
> iraceResults$parameters  
> iraceResults$experiments  
> iraceResults$allConfigurations
```

What can we do now?

You may have some good questions that configuration data can answer

Presenting the R package `iraceplot`:

<https://auto-optimization.github.io/iraceplot/>

Configuration data with iraceplot

In R

```
> R  
> install.packages("iraceplot")
```

Let's use the package

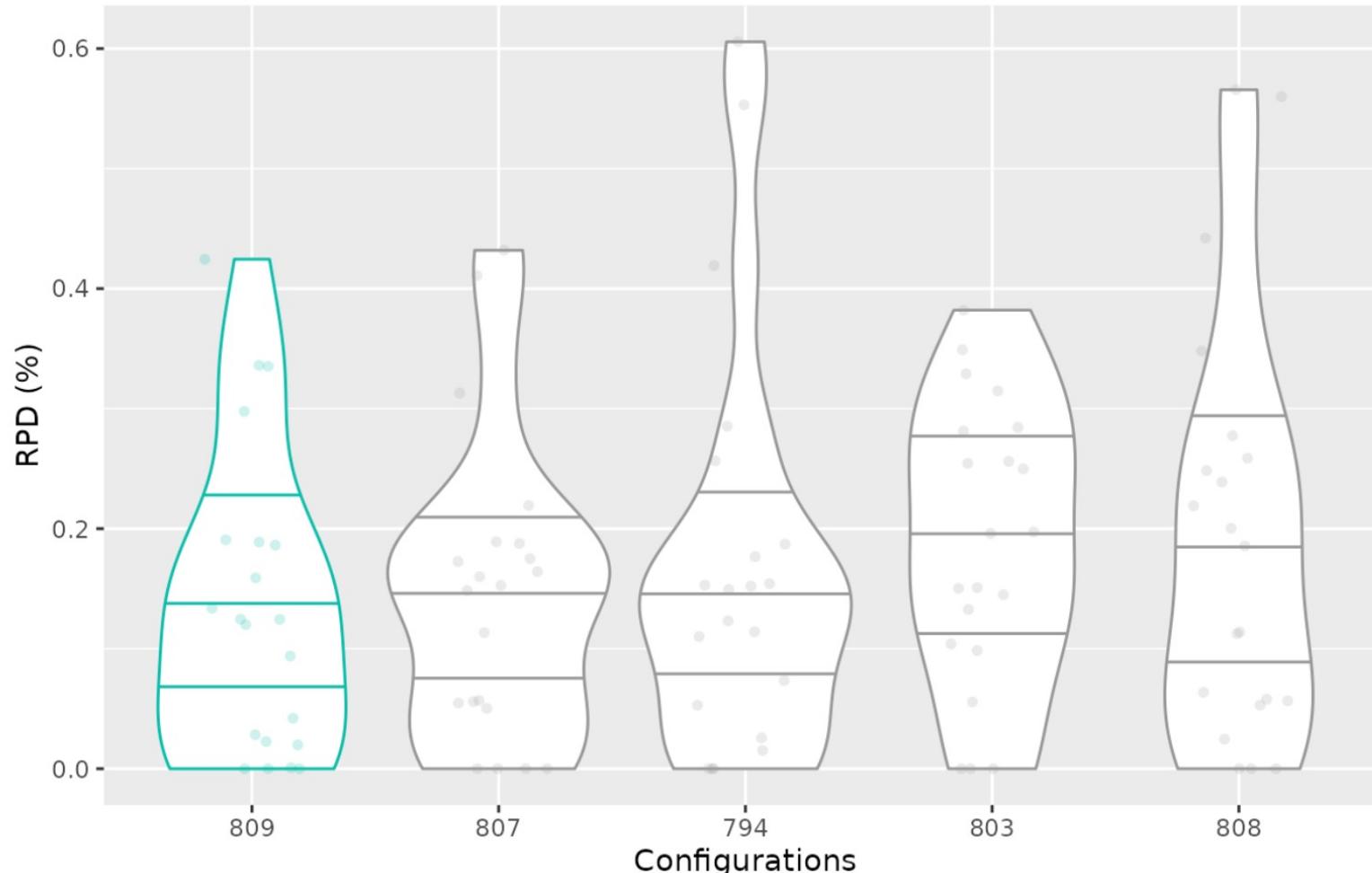
```
> library("iraceplot")  
> load("irace-example.R")  
> report(iraceResults, filename="report")
```

Testing / training performance

How the **training** performance of the elite configurations looks like?

```
> boxplot_training(iraceResults)
```

Testing / training performance



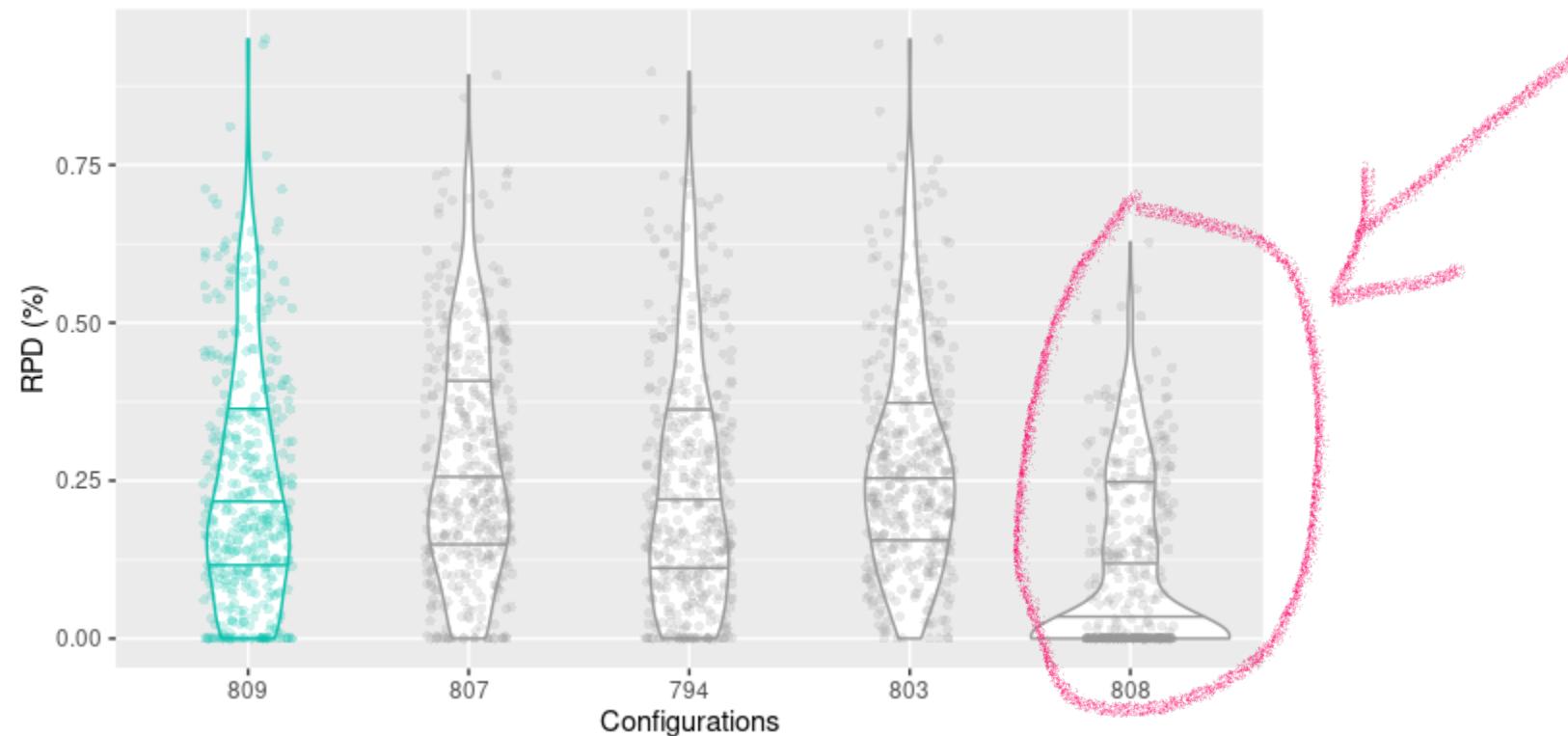
Testing / training performance

How the test performance of the elite configurations looks like?

```
> boxplot_test(iraceResults, type="best")
```

Testing / training performance

How the performance of the elite configurations looks like?



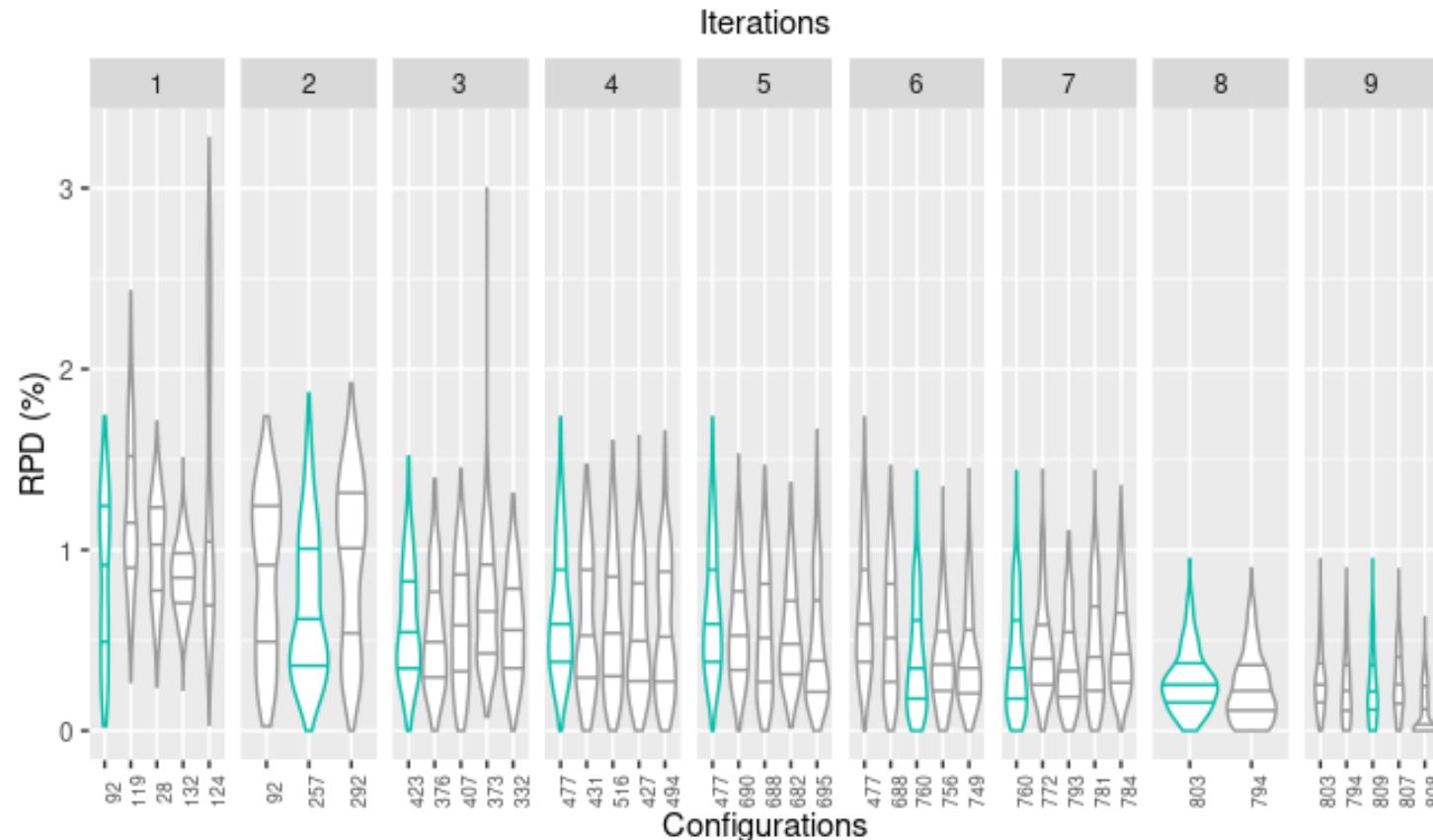
Testing / training performance

How the performance of all elite configurations looks like?

```
> boxplot_test(iraceResults, type="all", show_points=FALSE)
```

Testing / training performance

How the performance of all elite configurations looks like?



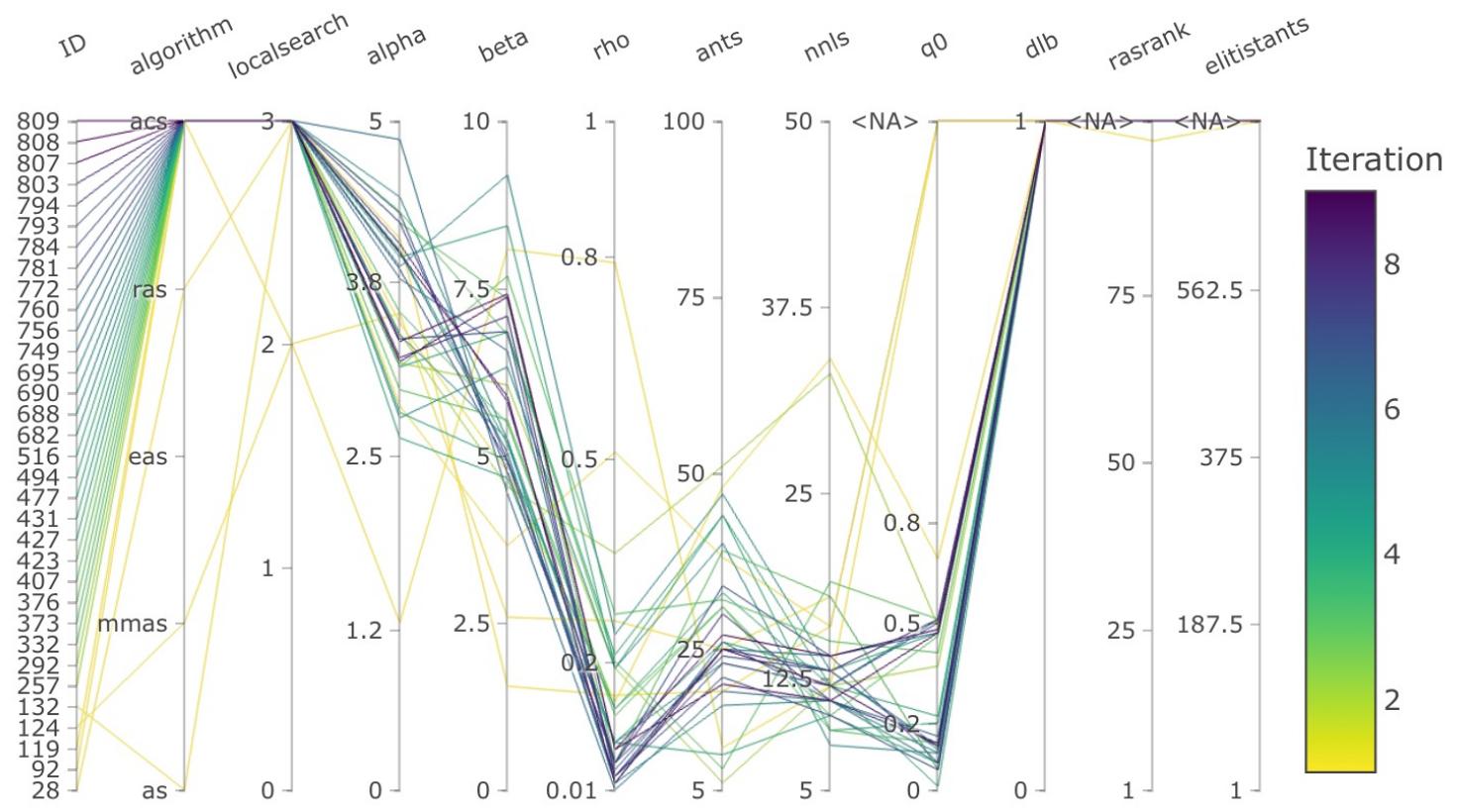
Configuration data

How elite configurations changed during the tuning?

```
> parallel_coord(iraceResults,  
+                   iterations=1:iraceResults$state$nbIterations,  
+                   color_by_instances = FALSE)
```

Configuration data

How elite configurations changed during the tuning?



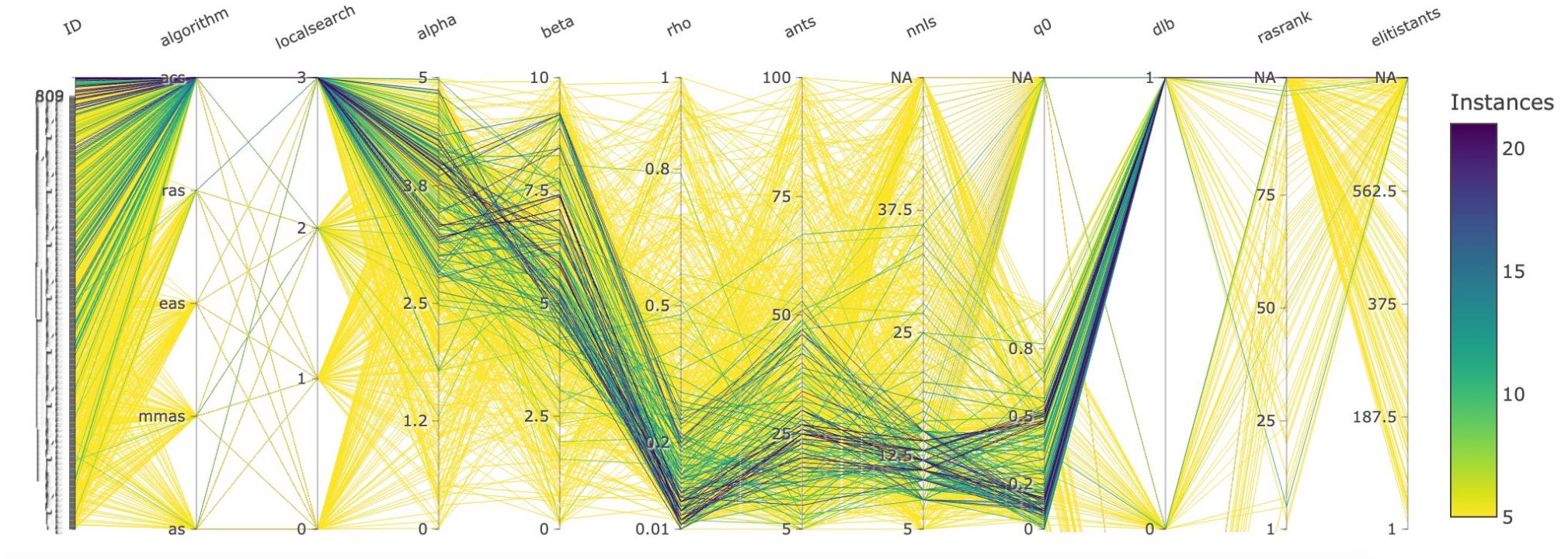
Configuration data with iraceplot

What about all configurations?

```
> parallel_coord(iraceResults,  
+                   iterations=1:iraceResults$state$nbIterations,  
+                   color_by_instances = TRUE,  
+                   only_elite = FALSE)
```

Configuration data with iraceplot

What about all configurations?

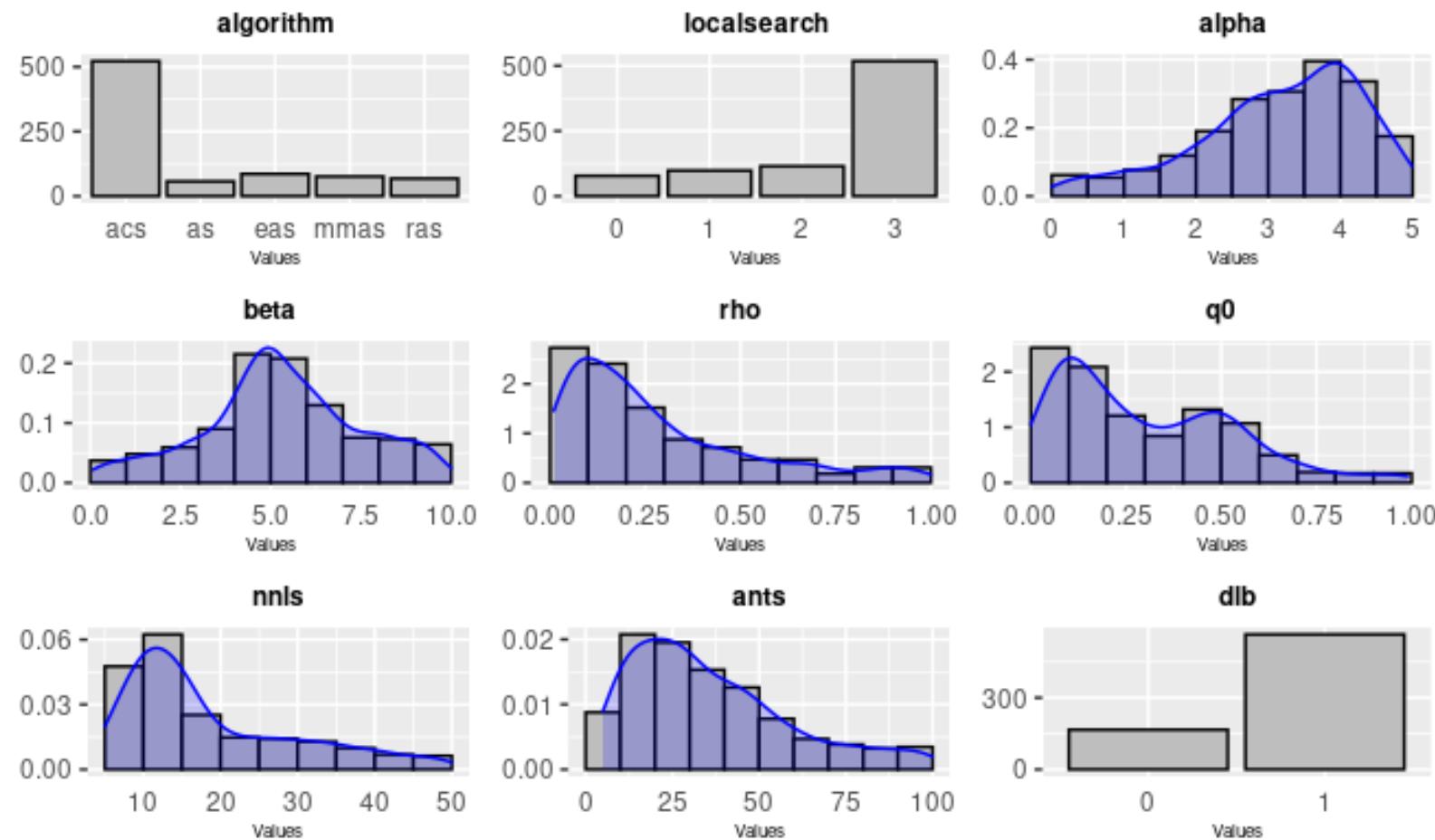


Parameter values

What parameter values irace focused the search on?

```
> sampling_frequency(iraceResults,  
                      param_names = c("algorithm", "localsearch",  
                                      "alpha", "beta", "rho",  
                                      "q0", "nnls", "ants",  
                                      "dlb"))
```

Parameter values

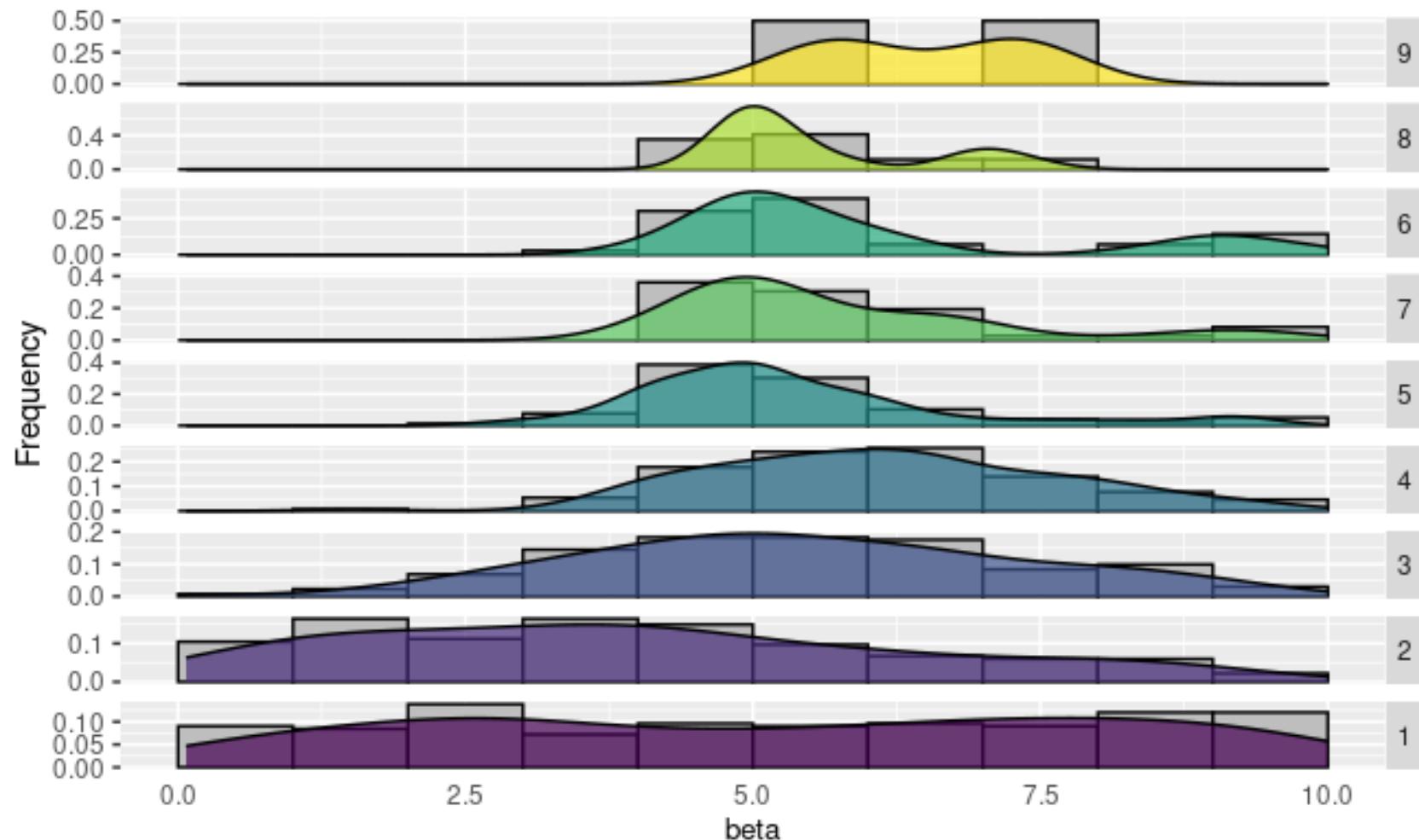


Parameter values

How a sampled value changed over the iterations?

```
> sampling_frequency_iteration(iraceResults,  
                                param_name = "beta")
```

Parameter values



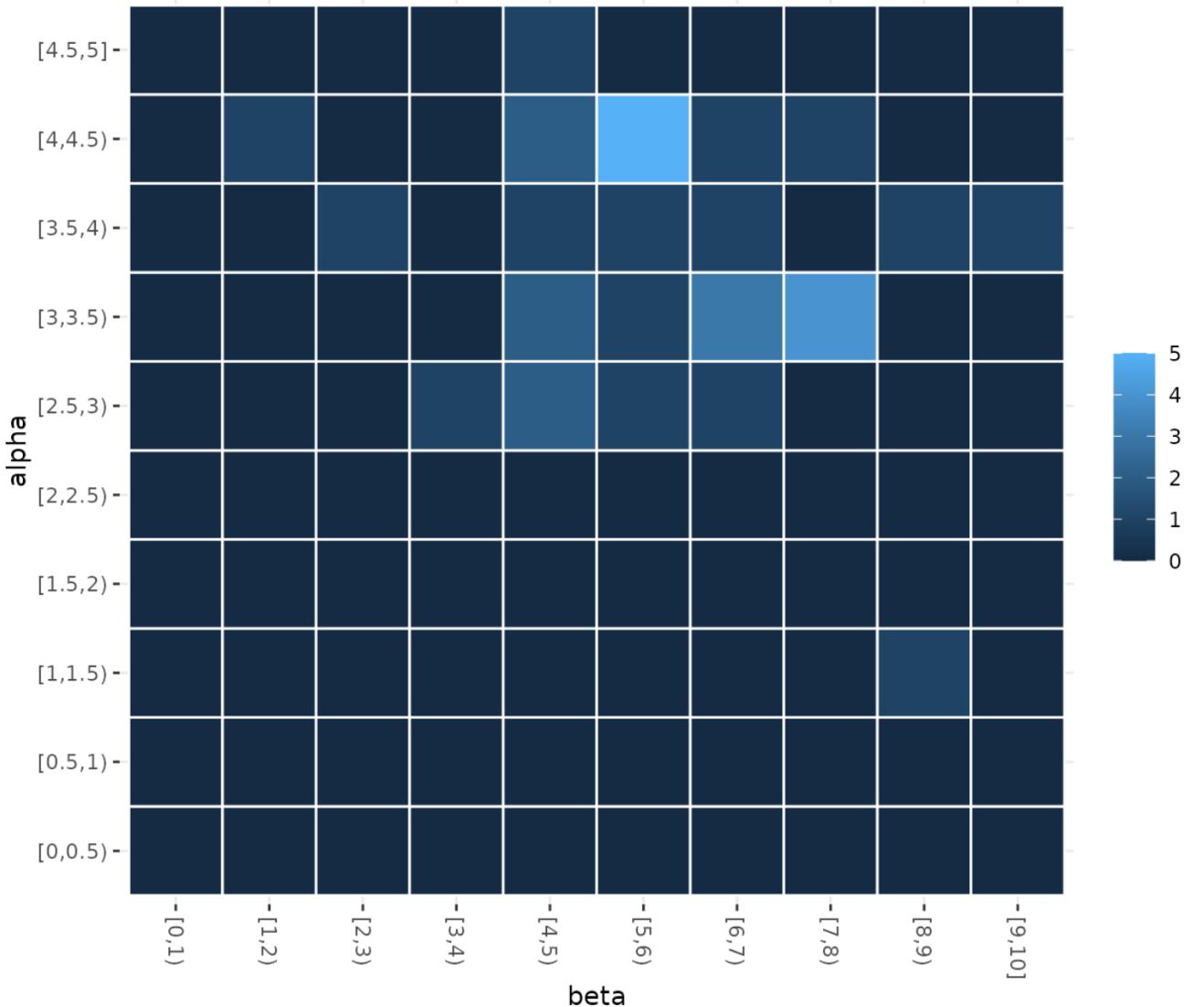
Parameter values

How visualize interacciones?

```
> sampling_heatmap(iraceResults,  
                    param_names = c("alpha","beta"))
```

Parameter values

How visualizar interacciones?



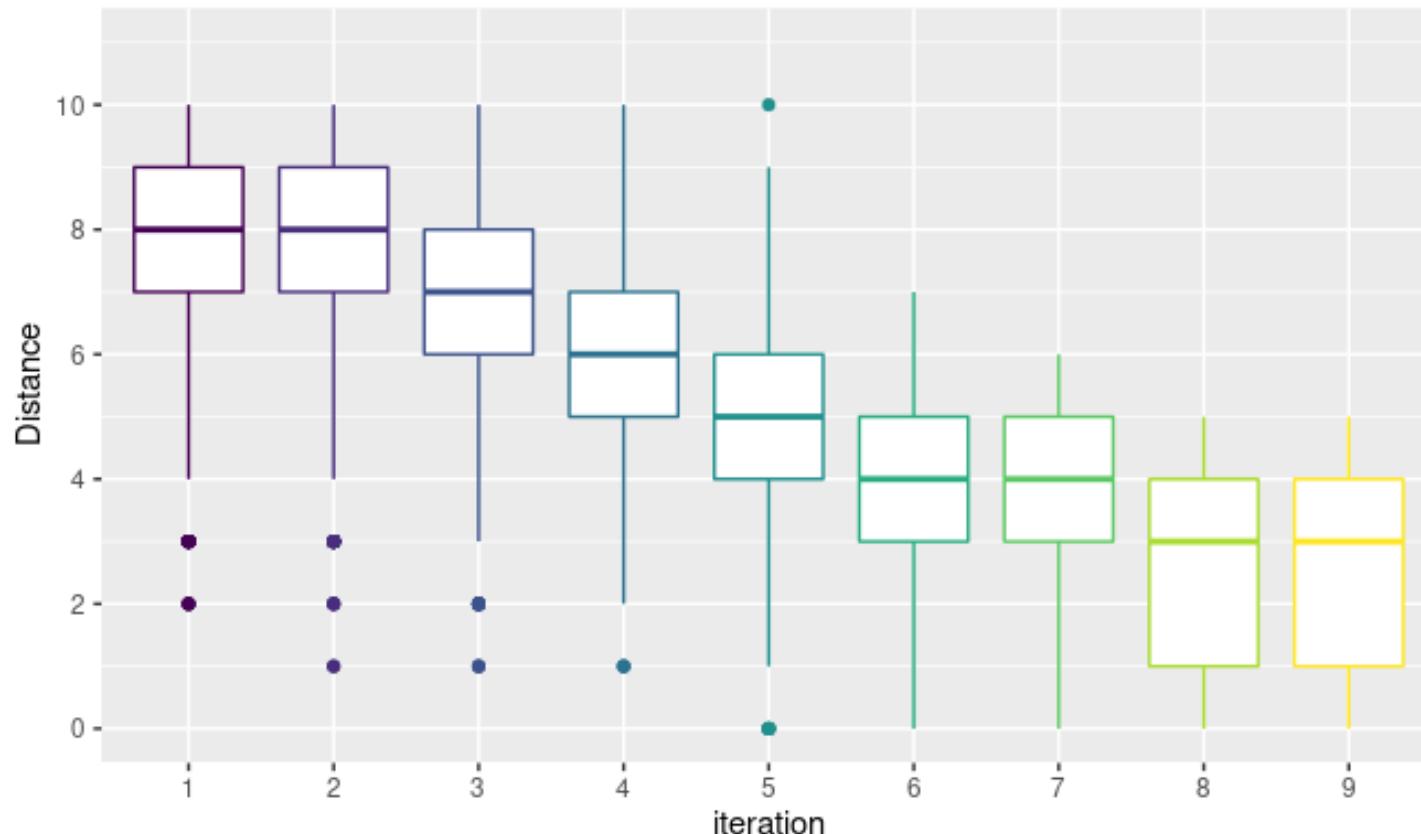
Parameter values

Sampling distance?

```
> sampling_distance(iraceResults, t=0.05)
```

Parameter values

Sampling distance?



How to contact us & keep updated

- The irace package website

<https://iridia.ulb.ac.be/irace/>

- The irace github repository

<https://github.com/MLopez-Ibanez/irace>

- The support group

<https://groups.google.com/g/irace-package?pli=1>

Now... A challenge

- Search the literature for an optimization algorithm
 - Experimental setup should be clear
 - Code should be provided
 - Instances should be available
- Configure it and compare your results with default performance!

Thanks for your attention !
Questions, comments?