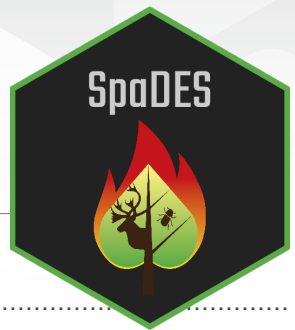


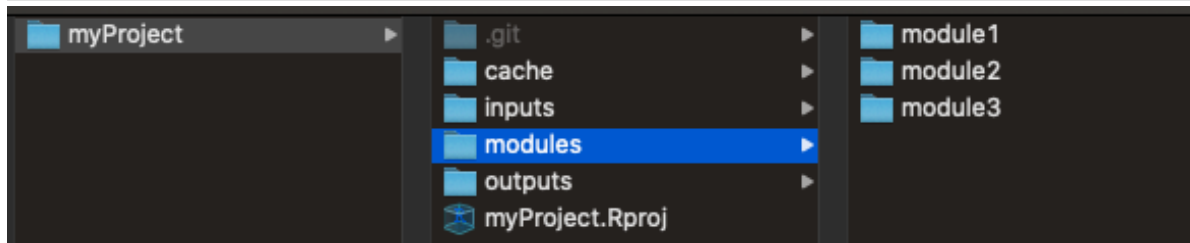
Simulation modelling in SpaDES : : CHEAT SHEET



Project setup

1. PROJECT DIRECTORY STRUCTURE

```
newProject("projName", "projPath")
```



2. GET OR CREATE MODULES

```
downloadModule("module4", path = "modules")
copyModule("module4", "module5", "modules", open = TRUE)
newModule("module6", path = "modules")
```

3. WORKSPACE SETUP

```
setPaths(
  cachePath = "cache", inputPath = "inputs",
  modulePath = "modules", outputPath = "outputs",
  rasterPath = file.path(dirname(tempdir()), "myProj")
)
```

use relative paths

set a scratch directory for intermediate raster objects

Running simulations

```
mySim <- simInit(...)
mySimOut <- spades(Copy(mySim), ...)
myExpt <- experiment(mySim, ...)
restartSpades()
```

initialize a simulation

run single simulation

multiple simulations (in parallel)

if something goes wrong

Accessing the simList

Accessor	Description
globals	global (i.e., non-module-specific) parameters
params, P	module-specific parameters
inputs, outputs	module input and output objects
ls.objects, ls.str, obj	list objects stored in the simList environment
paths, cachePath, modulePath, inputPath, outputPath, dataPath	simulation paths
times, end, start, time	simulation times
events, current, completed	simulation events
modules	modules in use
packages	simulation and module package dependencies
depends	simulation module dependencies (advanced)
envir	the simList environment (advanced)

Module Examples

```
system.file("sampleModules", package = "SpaDES.core")
getOption("spades.modulesRepo")
```

Module Development

1. METADATA

ensure metadata always up to date.
metadata is for humans too!

2. DEFINE AND SCHEDULE EVENTS

```
scheduleEvent(sim, time(sim) + 10, "module5",
  "myEvent", .normal())
scheduleConditionalEvent(sim, ...)
```

3. DEFAULT OBJECTS CREATED IN .inputObjects

PLOTTING (MODULE-LEVEL)

```
Plot()
clearPlot()
rePlot()
newPlot()
dev()
```

fast, modular plotting

change plot device



PLOTTING (SIMULATION-LEVEL)

```
moduleDiagram(mySim, ...)
objectDiagram(mySim, ...)
eventDiagram(mySim, ...)
```

Simulation & event caching

```
mySimOut <- Cache(spades(mySim), ...)
showCache(...)
keepCache(...)
```



Package options

```
?SpaDES.core
```