New names in RSiena

Tom A.B. Snijders

University of Groningen
University of Oxford

Overview

1. Purposes of new function names

- Better integration with *StocNet* suite.
- Understandable function names.
- Clearer design of package architecture.
- Nicer for users.

1. Purposes of new function names

- Better integration with *StocNet* suite.
- Understandable function names.
- Clearer design of package architecture.
- Nicer for users.

Restriction:

Compatibility with existing scripts —

retain existing function names; but not for multiSiena!

2. Design issues

$\Rightarrow$ Use S3 methods for which parallels could be defined also in other *StocNet* packages.

Interlude: what are S3 methods?

The best examples of S3 methods are print and plot.

S3 methods can be defined for *classes* of objects
and adapted to the kind of objects in this class.

For example in RSiena,
print methods are defined for dependent variables
(*print.sienaDependent*)
but not for covariates;
for the latter, print.default is used,
which gives the content of the object and all attributes.

Design issues (continued)

⇒ Use recognizable structures for names:

- as_..._rsiena for defining RSiena data objects from raw data;
- but set_..._saom for defining RSiena control objects for SAOM estimation;
- set_... for adding effects and interactions;
- make_..._rsiena for functions defining RSiena objects from RSiena objects;
- test_.. for testing, including GOF;
- interpret_.. for interpreting results ('postprocessing');

Design issues (continued 2)

- ⇒ Combine functions with a similar purpose (construct covariates, conduct tests).
- ⇒ No camelCase, use "_" instead of "." .
- ⇒ Separate distinct aspects of control.
- ⇒ Keep class names the same (e.g., sienaFit, sienaGOF); these are mainly used internally in R, and changing them would lead to proble of compatibility with objects created by earlier versions of RSiena;
- ⇒ But give siena data sets, as created by sienaDataCreate (now make_data_rsiena), class names sienadata as well as siena (because 'siena' is too general).

3. In particular: combine functions

- as_covariate_rsiena

  Replaces coCovar, varCovar, coDyadCovar, varDyadCovar.

  Distinguished based on class of input and keyword `type` in `"monadic", "oneMode" or "bipartite"`.

- set_effect

  Replaces setEffect and includeEffects.

- test_parameter

  Method for sienaFit objects; replaces Wald.RSiena, Multipar.RSiena, testSame.RSiena, score.Test.

  Distinguished based on class of input and keyword `method` in `"all", "same" or "score"`.

4. Separate control aspects

In the current RSiena, function sienaAlgorithmCreate contains a host of control settings of diverse nature, and is necessary even for default estimations, although only `projname` is relevant.
The function call of siena07 itself also contains some control settings: (e.g., `thetaValues, returnThetas`).

Defining control settings for estimation now is divided among three functions:

- set_model_saom for defining the model family (e.g., `modelType, maxDegree`);
- set_algorithm_saom for defining the algorithm (e.g., `maxlike, n3`);
- set_output_saom for defining the output generated (e.g., `outputName, returnThetas`);

What happened to siena07?

siena07 is replaced by siena,

an S3 method for class `sienadata` (the data set `x`):

```
siena(x, effects=NULL,
        control_model=NULL,
        control_algo=NULL,
        control_out=NULL,
        batch = FALSE, verbose = FALSE, silent=TRUE,
        initC=TRUE, nbrNodes = 1,
        clusterString=rep("localhost", nbrNodes),
        clusterType=c("PSOCK", "FORK"), cl=NULL, ...)
```

When no control objects are given (the default `NULL`),
the name of the output file
(the old `projname` of sienaAlgorithmCreate)
is the name of the data set `x` with `_out.txt` appended.

Still in the call of siena:
the specification of parallel processing,
and in ... possibly `returnDeps=TRUE` (for goodness of fit testing);

this was done, instead of putting them in `control_algo` and `control_out`, to allow straightforward coding by
modifying only the call of siena instead of
going to set_algorithm_saom and set_output_saom.

5. Changed names (also listed in the manual)

current	*new*
sienaDependent	as_dependent_rsiena
*Covar	as_covariate_rsiena
sienaCompositionChange	as_composition_rsiena
sienaCompositionChangeFromFile	as_composition_file_rsiena
sienaNodeSet	as_nodeset_rsiena
sienaDataCreate	make_data_rsiena
sienaGroupCreate	make_group_rsiena
print01Report	write_report
sienaAlgorithmCreate	set_model_saom, set_algorithm_saom, set_output_saom

model.create was dropped (sienaModelCreate retained, for the time being).

current	*new*
sienaDataConstraint	make_constraint
getEffects	make_specification
includeEffects	set_effect
setEffect	set_effect
includeInteraction	set_interaction
siena07	siena
siena08	meta_siena
siena.table ⊗	siena_table, write_result
Multipar.RSiena	test_parameter
Wald.RSiena	test_parameter
testSame	test_parameter(..., method="same", ...)
score.Test	test_parameter(..., method="score", ...)
sienaTimeTest	test_time

⊗ = old name not retained.

current	*new*
sienaGOF	test_gof
descriptives.sienaGOF ⊗	descriptives
selectionTable	interpret_selection
influenceTable	interpret_influence
sienaRI ⊗	interpret_size
sienaRIDynamics ⊗	interpret_size_dynamics
updateTheta	update_theta
updateSpecification	update_specification
and for multiSiena :	
sienaBayes ⊗	estimate_multi_saom
simulateData ⊗	simulate

⊗ = old name not retained.

For set_effect, set_interaction, includeGMOMStatistics, and includeTimeDummy,
keywords `name`, `interaction1`, and `interaction2`
were replaced by `depvar`, `covar1`, and `covar2`.

New methods: coef and vcov (cf. the statistics package).

Other S3 methods:
make_specification, make_constraint, write_report, test_parameter, write_result, test_time, test_gof, interpret_selection, interpret_influence, interpret_size, update_theta, update_specification
(some of them to be implemented also
for classes of other StOCNET packages?)

6. Function transformScript

This function transforms R scripts using the old RSiena names to scripts using the new names.
(But I also used it to transform the LaTeX code for the manual).

This function operates line by line on the input file.
It transforms the script line by line, with an exception for calls of sienaAlgorithmCreate.

Function transformScript (continued)

As function sienaAlgorithmCreate now is split into three,
no line in the new script is created for this function call;

calls of siena07 and sienaBayes are checked for the algorithms they use, and transformed into calls of siena and multi_siena respectively;

these are preceded by calls of set_model_saom, set_algorithm_saom, set_output_saom, set_algorithm_mcmc,
corresponding to the algorithm object
used by this call of siena07 or sienaBayes.

These algorithm objects constructed always have the names, respectively, `alg_model`, `alg_alg`, and `alg_out`.

Function transformScript passes three times through the old script:

1. The first time, to replace the names of functions for which the body of the call is unchanged.
2. In the second pass, calls of RSiena and multiSiena functions that are changed are transformed to calls utilizing the new names. The only RSiena functions in the script that are executed themselves here are calls of sienaAlgorithmCreate; their results are used for transforming calls of siena07 and sienaBayes. All function calls except those of sienaAlgorithmCreate are replaced with calls of the new functions.
3. In the final pass, if any old function names are left and used without being a function call — i.e., without being followed by `<- (` they are replaced with the new names.

Log file, error handling

Function transformScript also produces a log file.

If an error occurs, the log file will record what happened
until the occurrence of the error.

If in the algorithm used for siena07 or sienaBayes
(or its call) a variable occurs,
then this variable may be unknown to transformScript
which will lead to an error. For example:

```
Error in eval(parse(text = newcommand1)) : object 'nodes' not found
```

Just create the missing object and give it a value,
and run transformScript again.
In the generated script you probably will have to modify this object.