StoX User Manual

J. Martín-Herrero

v3.1 (April 2024)

Contents

1 Introduction

1.1 What?

StoX (read "S to X", like in *seed to unknown*, but also with an emphasis on its underlying στόχ-astic nature) is a software application designed to help you make sense of all those data so laboriously collected in the field to study the seed dispersal effectiveness of an assemblage of dispersers of a plant species or of an entire community.

1.2 Why?

If you are reading this manual, you probably have been, or plan to soon begin, collecting data on how many seeds your plant species produce in each of the habitats of interest, how many of those seeds are taken by each of the dispersers (animal or abiotic) interacting with those plants in each habitat, how many of those seeds end up in each of the different relevant microhabitat types (under the mother plant, under other plants of this or that type, in open field of this or that type), how many seeds are predated or otherwise destroyed in each of them, how many of the survivor seeds for each combination are able to germinate in the first or later seasons, how many of the germinated seeds survive in later life stages in each microhabitat within each habitat, and so on. All this for a number of replicates sufficient to achieve statistical significance. And most likely repeated along several seasons as well, to also incorporate the environmental variability. The idea of summarizing all that vast amount of data by using averages may have crossed your mind. But you have noticed the high variability among replicates. Renouncing to take it into account sounds like too big a loss of treasured knowledge. You may then be tempted to resort to standard deviations. But things start to get quite complicated: So many processes taking the seeds along so many branching stages. How will you combine all the standard deviations? And have you tested your data for normality? Please do. It is highly likely that you may find not nice surprises. A distribution-agnostic systematic way to squeeze till the last drop all those painstakingly acquired data would come really handy, wouldn't it? Well, look no more, because you have found it.

Yummy *Corema album* fruits waiting for an anemic *Larus michahellis* juvenile to pick them and bring their seeds to hospitable ground. But blackbirds and rabbits like them too. Who provides the best services to the plant?

1.3 What for?

StoX allows you to build a tree of successive branching *stages* connected by means of processes characterized by transition probability matrices (*castings*) directly built with your replicates. The transition probabilities are resampled by means of bootstrapping (a Monte Carlo strategy which does not make any assumption on the underlying probability distributions) to distribute an initial population of seeds among all the stages a selectable number of times. The resulting number of seeds at each stage of interest can then be analyzed by means of standard spreadsheet or statistical software. StoX produces tabular output that can be saved to a file or directly copied to a spreadsheet. StoX data can be used to compute the seed dispersal effectiveness and its quantity and quality components, and also to perform sensitivity analyses (what would happen to the recruitment if the population of this or that disperser were to decrease or increase? or the available surface of this or that microhabitat would be increased or reduced? and so on) and to evaluate the importance of different processes involved in the recruitment. Last, but very importantly, StoX outputs can be validated by statistical comparison with emergence and survival data directly measured in the field, to properly back up any conclusions derived from them. Models without validation are dangerous tools.

1.4 How?

Before you begin, you will need a collection of transition tables associated to each process in your seed dispersal model in the particular context of, in general, a given habitat, disperser and microhabitat where relevant. In a typical model, the plant species of interest can be found in a diversity of habitats, and their seeds are dispersed by a diversity of dispersers (probably differently in different habitats), which drop seeds in different proportions in a variety of microhabitats, and those seeds suffer different rates of predation in the different microhabitats, maybe also depending on what disperser brought them there, and the survivor seeds have potentially different germination rates depending on what disperser handled them (e.g. effect of gut passage) and to what microhabitat. They also may be subjected to secondary dispersal (another disperser takes them to somewhere else), or may lay dormant for several seasons during which they are repeatedly exposed to predation (and eventually secondary dispersal also). And those that germinate may have then different survival rates along later life stages, until, eventually, maturity, when they become reproducers themselves, closing the loop. StoX models can therefore be relatively simple (e.g. a single plant species, a single habitat type, a few dispersers, only a few microhabitats, no secondary dispersal nor dormancy, follow up only till seedling stage) or huge (an entire plant community, many habitat types, a large disperser assemblage, many microhabitat types, secondary dispersal, dormancy across several seasons, follow up till maturity). The usage is the same, though, because the tree structure provides a systematic way to deal with an unlimited degree of complexity. The complexity bill is paid only in the field, and StoX warrants the usability of all that field effort.

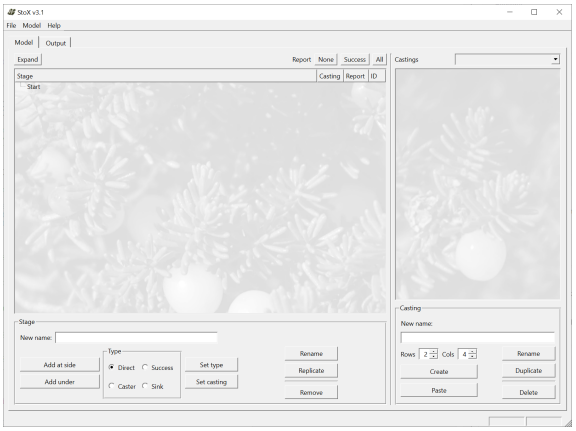
Then you build the model tree of successive stages, and associate each transition between successive stages to the corresponding casting table of transition probabilities. Terminal stages represent either lost seeds ("sink" stages) or seeds that successfully reached the end state ("success" stages; emerged seedlings, or later plant life stages, depending on the scope of the study). A transition stage can be followed by a single stage (and then it is a "direct" stage, without associated casting: all seeds go from that stage to its only child stage), or by several stages (a "caster" stage, with an associated casting table). Casting tables have as many columns as child stages, with each column representing the proportion of seeds transferred from the caster stage to each one of its child stages, and as many rows as replicates obtained in the study for that particular process. Different processes can have different number of replicates, but the number of columns must always match the number of following stages. Normally, each row in a casting should sum 1.0, with lost seeds in the process ending up in a "sink" stage, i.e. nor any seeds should be unaccounted for in the model, nor any seeds should appear from thin air. A consistency check on the model will make sure that the above constraints are met before running the model. It will also assign a unique hierarchical identifier to each stage for identification purposes in the model output. The model output lists the number of seeds that end in each of the stages selected to be reported, e.g. all success stages, any combination of stages, or all of them. The output in tabular form can be saved to a file or copied to a spreadsheet for statistical analysis.

2 Installation

Download the zip file containing the binary executable for Windows. Extract the zip file into a folder in your computer. The program does not need installation, and is ready to run. It can also be run from an external drive or pen drive. Go to the `\bin` folder and double click `Stox.exe`. You can optionally right-click the `.exe` file and create a direct link, then move the link to your desktop for quick access to the program. To uninstall, just delete the entire folder.

3 User interface

When you run the program, you will see the user interface depicted here on the right. There is a menu bar on top, with a set of options to create a new model, open a saved model, save a model to file, and exit the program; another set to check the model and run it; and finally another to show a window with licensing, citing and contact information. Below, the main screen is split in two parts: on the left the model tree view, where all revolves around stages, and on the right a table view (empty at the beginning), where castings are managed. An alternate tab lets you switch between the model screen Model and the output screen Output, where the run parameters can be set.



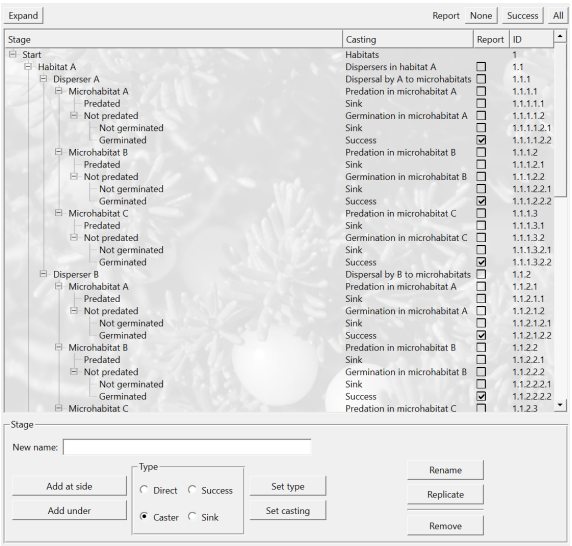
The binary distribution includes a simple generic sample model that you can load using the Open option of the File menu or by pressing Ctrl+O. To start a new model of your own, you can use Ctrl+N or the New option of the File menu. This will close the current model, if any, eventually asking first if you want to save any modifications done. You do not need to useCtrl+N nor New if you just started the program: you already have an empty model ready to begin with. You can save your model at any time by pressing Ctrl+S or with the Save or Save as options of the File menu. Consider pressing Ctrl+S from time to time when performing relevant changes to a model that you want to preserve, do not delay saving till you have finished a long session of unsaved work. If you are not sure whether you like those changes, save a copy of your model under a modified name with Save as so you will be able to revert to the original version with the original name in case you change your mind. Use informative names so you know what is what along the different versions of your model.

The status bar at the bottom of the screen will show momentary error or information messages. If something does not happen as expected, take a look there to get information on the cause of disagreement between your expectations and what really happened, or rather did not happen. For instance, if you press Rename to change the name of a model stage but you did not enter any new name in the corresponding box, a momentary message will let you know that you must first enter a new name to be able to rename a stage. The status bar also has on the right side two indicators that let you know whether the model has been modified since last saved and whether the model has been checked and is ready to run.

3.1 Model tree

The model tree view has four headings, from left to right: stage name, associated casting (or stage type, if not a caster stage), report state (whether the stage will be included in the model output report or not), and ID, a unique stage hierarchical identifier that will be automatically assigned

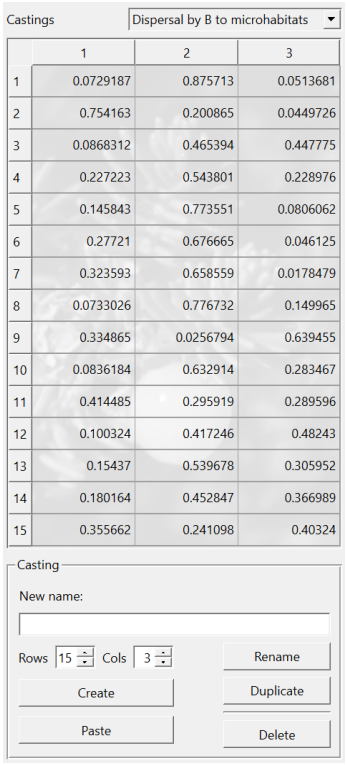
to each stage to facilitate their identification in the report. The automatic unique ID allows the use of stage names that do not need to be unique, which otherwise would be a burden in large or complex models. Above the tree window there is a button on the left to expand the whole tree [Expand] (individual tree parts can be expanded or collapsed by clicking on the little + or - symbols at the tree junctions), and on the right three buttons that will check all stages for report [All], only stages of type success [Success], or uncheck them all [None]. Individual stages can be checked or unchecked for report by clicking the individual checkboxes in the tree.



Under the tree view there is the panel to manage stages. The New name box is where you type either the name of a new stage or a new name to rename an existing stage. The Type box allows you to set the type for a new stage or to change the type of an existing stage. New stages can be added either at the same level as the currently selected stage in the tree (a sibling stage) [Add at side], or following the currently selected stage (a child stage) [Add under]. You must first write a name for the stage in the New name box. The name can be later changed with [Rename]. The stage is created of the type selected in the Type box. This can also be changed later by selecting another type in the box and then clicking [Set type]. If a casting is selected in the Castings list above the casting view and the currently selected type is Caster, the active casting is assigned to the new stage at the time of its creation. This can also be changed later by selecting another casting in the list and clicking the [Set casting]. Any stage can be deleted from the model with [Remove]. Note that deleting a stage with child stages under it will also delete all stages downstream from that stage. You will be notified first, and asked to confirm.

Last, [Replicate] replicates the subtree hanging from and including the currently selected stage in the tree. You first select the stage at the top of the subtree that you want to replicate, then click [Replicate] and finally click again on the stage you want the replicated subtree to hang from. This is a most convenient tool, taking advantage of the fact that model trees are normally made of a hierarchical combination of repeating subtrees with identical or very similar structures. After replication you normally modify some of the new stage names (but not necessarily, because stage names can be repeated) and assign different castings where needed. The replication mechanism is used to best advantage if the tree is built depth-first rather than breadth-first: you first build one branch till a terminal stage, e.g. for a single given habitat, by a single given disperser, to a given microhabitat, and then successively replicate the relevant parts of the branch to go on adding parallel structures for other microhabitats first, then other dispersers with all microhabitats already incorporated, and finally other habitats with all dispersers and microhabitats already incorporated. Structural differences among habitats, dispersers or microhabitats are then implemented by adding or removing specific stages in some of the replicas. In this way complex trees can be built quite quickly and with minimal effort.

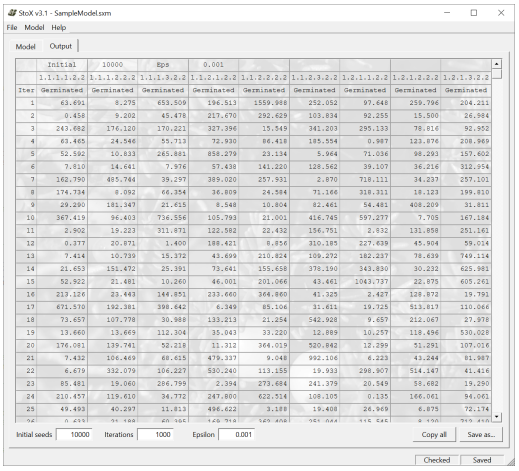
3.2 Castings



On top of the casting view on the right of the screen there is a drop-down list (initially empty) to select the current casting that will appear on the table view. The selected casting in the drop-down list is the one that will be assigned to a new caster stage, or with Set casting at any later time. Under the table view there is a panel to manage the castings, similar to the one to manage the stages. The New name box is where you type either the name for a new casting or a new name to rename the current casting by clicking Rename. In contrast to stages, names of castings must be unique, to avoid any ambiguities in the model. Two smaller boxes allow you to set the number of rows and columns for the new casting, which is then created by clicking Create. This will create an empty table of the indicated size with the given name. Then you can directly enter the individual values for each cell of the table one by one. If you use this method, you will not be able to enter cell values that cause the sum of the row to be greater than one. In that case, the difference up to 1.0 will appear in the cell instead of the value you typed in. This may come in handy to avoid having to type the precise last value of each row: you just type 1 or any other big number instead of a long precise decimal number, and the correct number will be replaced for you. Alternatively, and more efficiently, you can copy the table from a standard spreadsheet such as MS Excel (select all the rows and columns, without headers, and press Ctrl+C), and then click Paste. A casting with the proper number of rows and columns will be created with the name entered in the New name box and all the proper cell values. Note that in this way you could temporarily circumvent the row sum check, but you will encounter it later when the model is checked before running. Tables can also be copied from a simple text file, as long as columns are separated by single tabs, with one row per line of text.

3.3 Output

By clicking the Output tab above the model tree view you will access the output screen. There you can set values for the initial number of seeds, the number of iterations to run the model, and epsilon, the tiny number that is used for the tail of any distribution with infinite support (epsilon is your practical null probability). StoX will remember these parameters between sessions.



If you try to run the model without having checked it for consistency first, a check will be performed first, pointing out any inconsistencies in the model structure or the castings, such as stages of type direct with more than one child stage terminal stages (type sink or success) with child stages, or caster stages without casting or with a casting with the wrong number of columns (castings must have as many columns as child stages has the caster stage). It will also warn about casting tables with row sums different from 1.0 (transition probabilities from a stage to its child stages should always add up 1.0, otherwise seeds are either vanishing in or appearing from thin air). You also can perform an explicit check at any moment by using the Check option of the Model menu.

A consistent model can be run by using the Run option of the Model menu or by pressing Ctrl+R. The model run can be stopped at any moment during the execution by clicking Cancel. However, model execution is so efficient that it normally will be already completed before you have time to reach the button. The model output is shown in tabular form, with a row per each iteration under a heading with the unique ID's and names of the reported stages. Reported stages are those that have their corresponding checkboxes checked in the model tree. Epsilon and the initial number of seeds are also recorded in the top row. The output can be saved to a text file in `.txt` or `.html` format by clicking Save as... below on the right, or copied to the clipboard with Copy all for subsequent pasting into statistical or spreadsheet software such as MS Excel or the like. Every new model run overwrites the previous output on the screen.

References

Bascompte, J. and Jordano, P. (2007). Plant–animal mutualistic networks: the architecture of biodiversity. *Annual Review of Ecology, Evolution and Systematics*, 38:567–593.

Calviño-Cancela, M. (2011). Simplifying methods to assess site suitability for plant recruitment. *Plant Ecology*, 212(8):1375–1383.

Calviño-Cancela, M. (2012). Gulls (*Laridae*) as frugivores and seed dispersers. *Plant Ecology*, 212(7):1149–1157.

Calviño-Cancela, M. and Martín-Herrero, J. (2009). Effectiveness of a varied assemblage of seed dispersers of a fleshy-fruited plant. *Ecology*, 90(12):3505–3515.

Calviño-Cancela, M. and Rubido-Bara, M. (2012). Effects of seed passage through slugs on germination. *Plant Ecology*, 213(4):663–673.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA.

Escribano-Avila, G., Calviño-Cancela, M., Pias, B., Virgos, E., Valladares, F., and Escudero, A. (2014). Diverse guilds provide complementary dispersal services in a woodland expansion process after land abandonment. *Journal of Applied Ecology*, 51(6):1701–1711.

González-Castro, A., Calviño-Cancela, M., and Nogales, M. (2015). Comparing seed dispersal effectiveness by frugivores at the community level. *Ecology*, 96(3):808–818.

Loayza, A. P., Luna, C. A., and Calviño-Cancela, M. (2020). Predators and dispersers: Context-dependent outcomes of the interactions between rodents and a megafaunal fruit plant. *Scientific Reports*, 10(1):6106.

Martín-Herrero, J. and Calviño-Cancela, M. (2024). StoX: Stochastic multistage recruitment model for seed dispersal effectiveness. *Software Impacts*. Submitted.

Neghme, C., Santamaria, L., and Calviño-Cancela, M. (2017). Strong dependence of a pioneer shrub on seed dispersal services provided by an endemic endangered lizard in a Mediterranean island ecosystem. *PLoS One*, 12(8):e0183072.

Schupp, E. W., Jordano, P., and Gómez, J. (2010). Seed dispersal effectiveness revisited: A conceptual review. *New Phytologist*, 188:333–353.

Alphabetical Index