

# Subflows

Complex simulations can involve many calculations that can make the simulation difficult to read. Furthermore, many simulations have components that are re-used over and over to run the same process. For this reason, calculations are often broken down into *subflows*.

## What Is A Subflow?

Subflows are “parts” of simulations that can be reused, much like a function or subroutine in a programming language, or a formula in Microsoft Excel. They can include sources, calculations, and sinks just like any other workflow, but they are “embedded” into what can be thought of as a calculation. By [encapsulating](#) subflows in this manner, large simulations can be built from easy-to-read components. These components work the same way that calculations do, and can be added to workflows as if they were calculations themselves.

### 📌 Note

Subflows run on the same node as their parent workflow. See [How Scaling Works](#).

## Why Would I Use A Subflow?

Imagine if you created a simulation that involved a diallele cross followed by selection that you had tuned to be just right. Now you want to use that same crossing and selection scheme again in a different workflow. By saving the scheme as a subflow, you can use it in as many workflows as you want, and if you change the subflow, all of the workflows that contain that particular subflow will be updated automatically.

## Working With Subflows

Creating a subflow can be accomplished in three steps:

1. Create a workflow component (like the diallele crossing and selection example above) that you will want to re-use across different workflows.
2. Create a subflow from that workflow in the Mendel UI or with the Mendel Python Client.
3. “Embed” a subflow into a larger workflow.

## Note

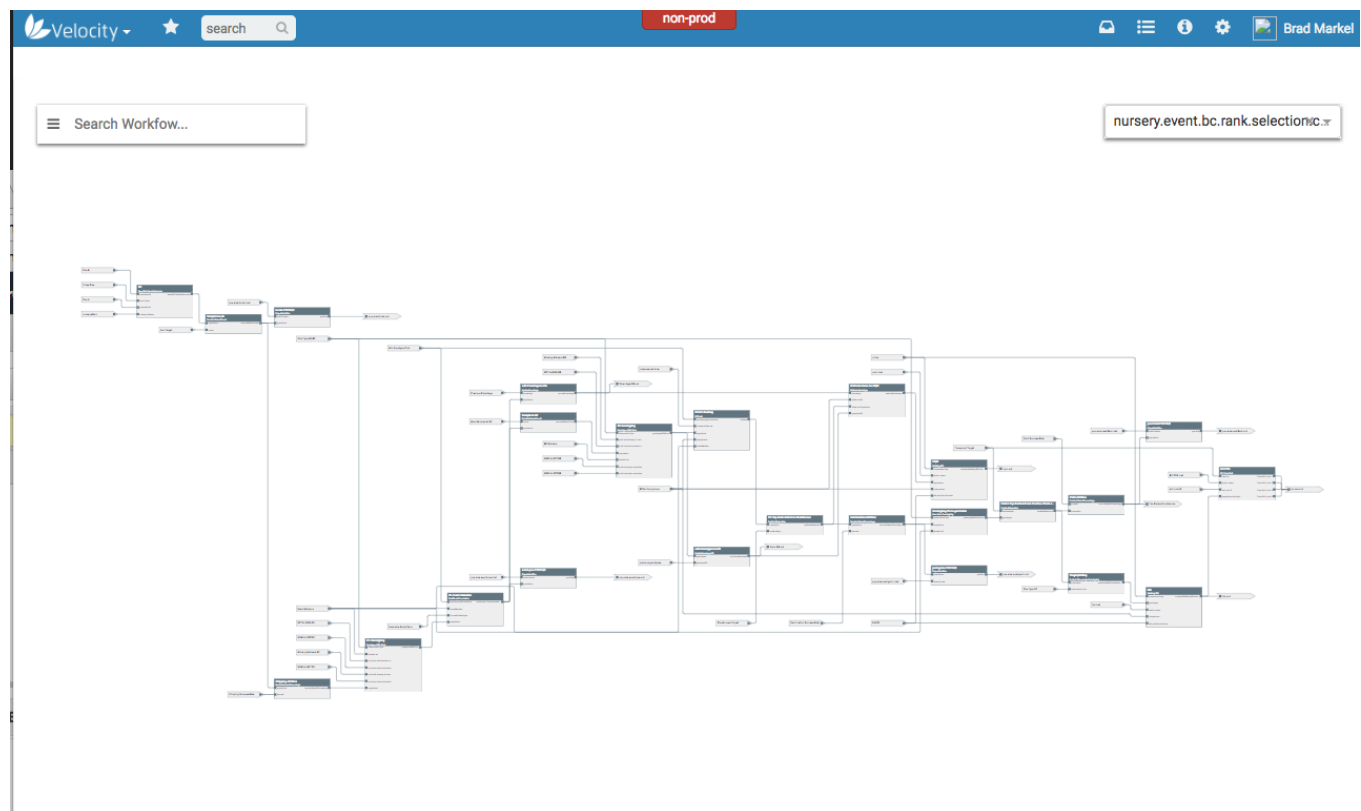
Although Mendel will let you nest subflows within subflows (sub-sub-flows), this is a bad idea and is not supported.

# Naming Subflows

By convention, subflows are named with the first character being an underscore (`\_`) to distinguish them from normal workflows. This is a subjective standard. Subflows can be normal workflows too and Mendel does not care if you choose to name it differently. However, naming the workflow that you intend to turn into a subflow with an underscore makes it clear to everyone that the workflow is a subflow and is not intended to be used alone.

# Creating Subflows

Subflows start out just like normal workflows. You create a workflow, add sources, calculations, and sinks, and then save the workflow. The subflow “inputs” are sources. The subflow “outputs” are sinks.



*This is an example of a subflow that you might wish to reuse. It is built just like a normal workflow.*

After saving the workflow, you can then create the subflow from the UI with the “Create subflow” button.

The screenshot shows the Mendel Workflow Designer interface. At the top, there's a blue header bar with the Velocity logo, a search bar, and a 'non-prod' status indicator. Below the header, the 'Workflow Designer' title is on the left, and a 'MANAGE REFERENCE DATA' button is on the right. The main area is divided into three panels: 'Designer', 'Subflows', and 'Run Summary'. The 'Designer' panel has a 'Workflow name' field and a '+ CREATE' button. It contains a list of workflow names: 'KM\_Soy\_SC1\_GEBV\_VAR\_Estimation', 'Corn-1 Donor-4NM-3MK-2ZY', 'kcpaul-bug-test', 'Corn-1 Donor-7NM-1ZY', and '\_Canola\_SYNParents'. The 'Subflows' panel has a '+ Create Subflow' button circled in red. It contains a list of subflow names: 'RP.DP.1donor.subflow', 'Nursery.MABC.BCrank.chip', 'Nursery.MABC.STACK.CQM.chip.1', 'Nursery.1ZYG.chip', and 'Nursery.1ZYG.chip.1'. The 'Run Summary' panel has a '+ Create Run' button and a text area containing 'Huashi's favorite'. At the bottom right, there's a 'Debug Logging' section with buttons for 'INFO', 'DEBUG', and 'TRACE'.

*You can create a subflow from an existing workflow by using the “Create Subflow” button on the dashboard of Mendel.*

Like normal workflows, subflows will have default inputs and outputs. Subflows allow these inputs and outputs to be turned on and off so that they are visible or invisible to the user of a subflow. If a subflow input is turned off, it must have a default.

## Create Subflow

### Name

Nursery.MABC.BCrank.chip

Friendly Subflow Name

### Workflow

nursery.event.bc.rank.selection

Workflow that encapsulates

### Inputs

<input checked="" type="checkbox"/>	Pop.A	Pop.A
<input checked="" type="checkbox"/>	Pop.B	Pop.B
<input checked="" type="checkbox"/>	Cross.Size	Cross.Size
<input checked="" type="checkbox"/>	Lab.Target	Lab.Target
<input checked="" type="checkbox"/>	Chipping.Success.Rate	Chipping.Success.Rate
<input checked="" type="checkbox"/>	Event.Markers	Event.Markers
<input checked="" type="checkbox"/>	BK.Markers	BK.Markers
<input checked="" type="checkbox"/>	RP.for.Comparison	RP.for.Comparison
<input checked="" type="checkbox"/>	Shadehouse.Target	Shadehouse.Target
<input checked="" type="checkbox"/>	Germination.Success.Rate	Germination.Success.Rate
<input checked="" type="checkbox"/>	cmap	cmap
<input checked="" type="checkbox"/>	BC.POS.Number.of.Seeds	BC.POS.Number.of.Seeds
<input checked="" type="checkbox"/>	Previous.Parentage	Previous.Parentage
<input checked="" type="checkbox"/>	previous.genotypes	previous.genotypes
<input type="checkbox"/>	pop.size.to.lab.met	pop.size.to.lab.met
<input type="checkbox"/>	Missing.Markers.EV	Missing.Markers.EV
<input type="checkbox"/>	Obs.Type.MABC	Obs.Type.MABC
<input type="checkbox"/>	HET.to.HOM.EV	HET.to.HOM.EV
<input type="checkbox"/>	Favorable.Event.Geno	Favorable.Event.Geno

Friendly name for input/output

Whether subflow inputs are visible or invisible when using the subflow in the designer

*Creating a subflow with optional renamed inputs. Subflows with inputs turned off must have defaults in the workflow design.*

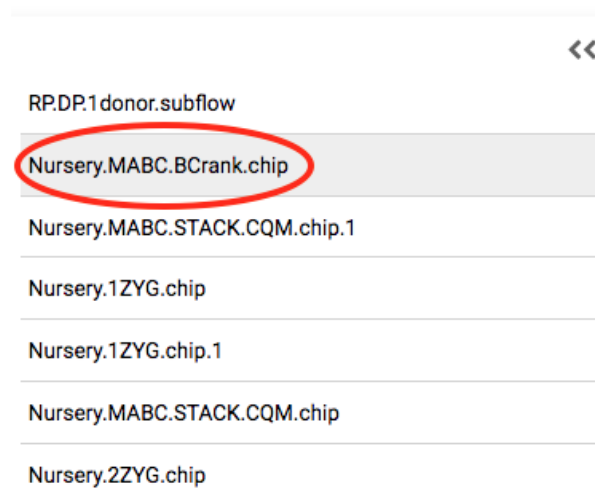
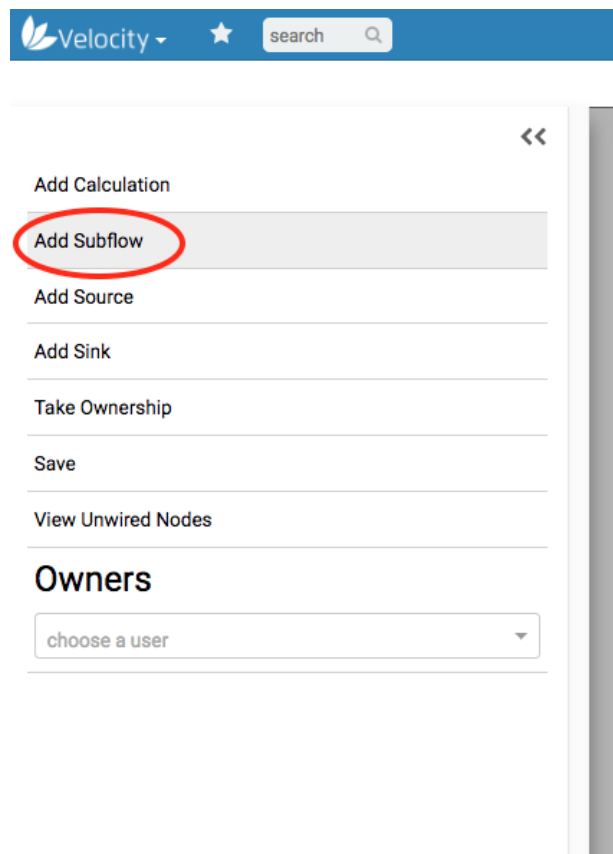
If possible, you should always add defaults to your subflows to make it easier to understand if something goes wrong. You can also give subflows a friendly name that is less cryptic than starting it with an underscore.

Ownership rules apply for subflows just like they do other workflows. Although it is possible to embed a subflow that you do not own, it cannot be modified. If you wish to modify a subflow

that you do not own, copy the workflow, then create another subflow.

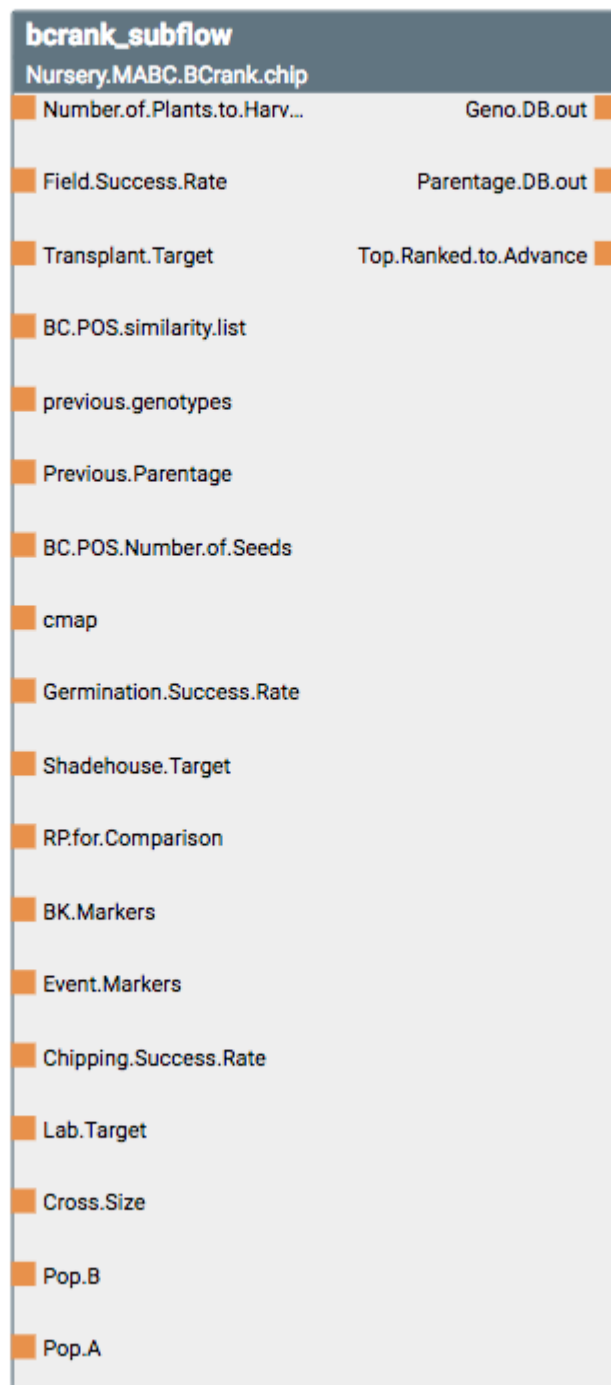
## Using Subflows in Workflows

After a subflow is created, it will appear on the left side of the Designer as a “Subflow” which you can then add to a parent workflow. If a subflow is ever changed or erased, the workflow will stop with an error. If a subflow has inputs or outputs that are renamed, then the UI will detect and warn you of the changes.



Choose the subflow you wish to add to the workflow. The name displayed is the “friendly name.”

Add a subflow to your workflow by clicking “Add Subflow” in the Designer.



*Subflows work like any other calculation. The sources correspond to inputs, the sinks correspond to outputs.*

**Note**

Should you use a calculation or a subflow? Well, in general, the answer is to use a subflow when you can, and a user-defined calculation when you have to. User-defined calculations have a number of requirements that must be followed and are harder to create and maintain (see [Developing Calculations](#))