Program Branch

Graphical Design and Analyses of Release-Recapture Branching Models

Developed by:
Trish Lockhart, John R. Skalski, and Jim Lady
Columbia Basin Research
School of Aquatic & Fishery Sciences
University of Washington
1325 Fourth Avenue, Suite 1515
Seattle, Washington 98101-2540

Prepared for:
U.S. Department of Energy
Bonneville Power Administration
Division of Fish and Wildlife
P.O. Box 3621
Portland, Oregon 97208-3621
Project No. 1989-107-00
Contract No. 59002

December 2015

Acknowledgments

This project is funded by the Bonneville Power Administration (BPA), US Department of Energy, under Contract No. 67518, Project No. 1989-107-00, as part of the BPA's program to protect, mitigate, and enhance fish and wildlife affected by the development and operation of hydroelectric facilities on the Columbia River and its tributaries.

Table of Contents

Acknowledgments i
Chapter 1: Introduction 1
Chapter 2: Using Program Branch 3
2.1 View the Log File 4
Chapter 3: Diagram Model 5
3.1 Creating the Diagram 8
3.2 Using Connectors 13
3.3 Connecting Components 15
3.4 Selecting and Renaming Components 17
3.5 Other Diagram Editing and Viewing Commands 19
3.6 Path Probabilities 20
Chapter 4: Load Data 23
4.1 Detection Histories File 24
4.2 Bin Counts File 26
4.3 Manually Entered Bin Counts 27
4.4 Sufficient Statistics 28
Chapter 5: Estimate Parameters 29
5.1 Calculate Model Parameters 30
5.2 Define Pathways for Estimation 34
5.2.1 Defining an Unbalanced Pathway 38

Chapter 1: Introduction

With the advances in tagging technology, it is now possible to create much more complex mark-recapture models than in the past. Tagging studies are being conducted in which populations diverge on different paths and sometimes merge again to the same path. Estimating survival-related parameters for these studies requires much more complex models than before.

Program Branch was written to allow users to define custom models graphically and to estimate survival-related parameters without having to manually code the likelihood equation. The diagram, along with the corresponding observed data, is then used to estimate the parameters of interest.

A sample model with data and estimates is included in Program Branch to help a new user explore the capabilities of the program. Simply click the Help menu and then click on" View Sample" to load the sample model. Note that the "View Sample" button is only available if there is not currently an active model.

Chapter 2: Using Program Branch

Figure 1 shows Program Branch at startup; the detail on the left shows the tabs for the three main working areas of Program Branch.

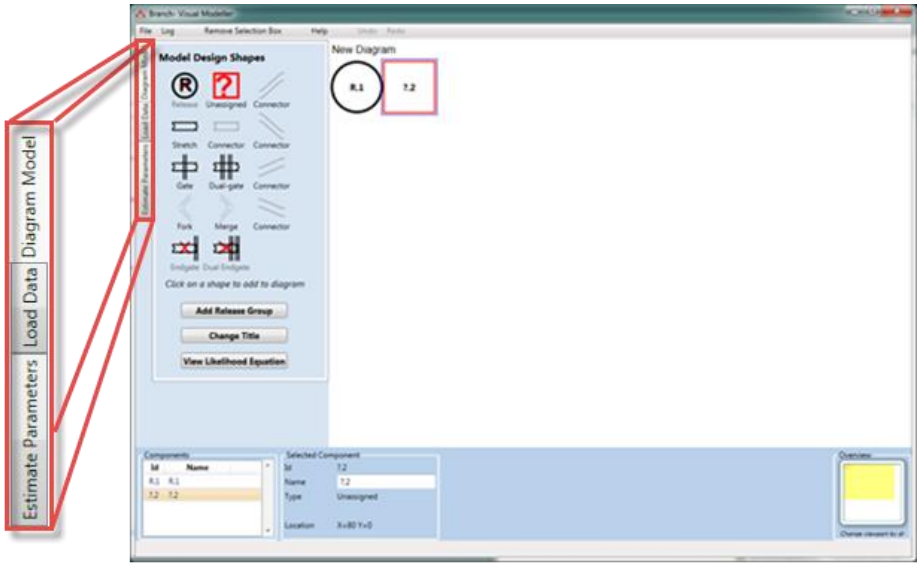


Figure 1. Program Branch at startup with the three selectable tabs shown in detail

They are:

1. **Diagram Model**, where the model is graphically created by the user. This is the tab that is active when Program Branch is started.
2. **Load Data** is used to load data for estimating the parameters. This is done after the model has been created in the Diagram Model tab.
3. **Estimate Parameters** provides the tools for estimating the model parameters after the model has been defined (Diagram Model tab) and the data has been loaded (Load Data tab).

2.1 View the Log File

Program Branch outputs important system information and error messages to a log file, indicating the severity level of the errors as well. The user may view the log file at any time by selecting “View Log” from the “Log” menu.

Chapter 3: Diagram Model

The Diagram Model tab, active when Program Branch is started, allows the user to define the model. There are two main components of the Diagram Model Tab: The canvas on the right that shows a graphic representation of the model, and the Model Design Shapes area on the left that contains the components for designing a model as shown in Figure 2.

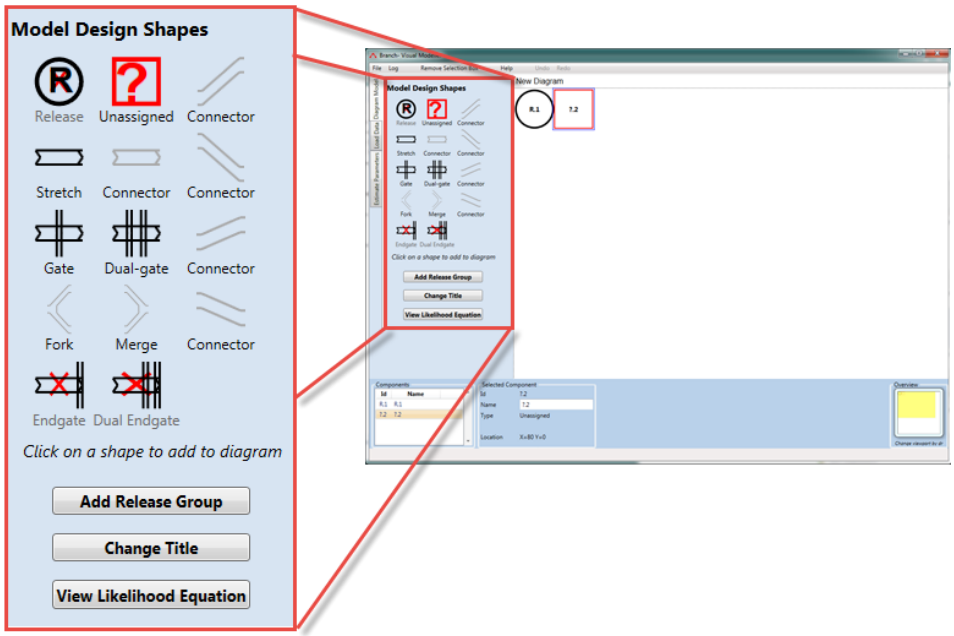


Figure 2. Program Branch at start up with the model design shapes shown in detail

Notice that some of the shapes have a red "X" through them. This indicates that the shape is unavailable in the current context. The shapes that are currently "X"-ed out will become accessible in the appropriate context.

The shapes available for building a model are as follows:

- **Release** defines a release point. This is a place where individuals are introduced into the model. Releases are names "R.n" by default, where "n" is a counting number.
- **Stretch** designates a length of time or space for which survival will be estimated. Stretches are named "S.n" by default.
- **Fork** is a place where individuals can go in one of two paths (in time or space).
- **Merge** is used to bring two different paths together.
- **Gate** indicates a detection point where the probability of detection will be estimated. Gates are named "G.n" by default.
- **Dual-gate** indicates a detection point consisting of two arrays, allowing for independent estimation of the detection probability, named "D.n" by default.
- **Endgate** is a gate at the end of a path, named "E.n" by default.
- **Dual Endgate** is a dual gate at the end of the path, named "H.n" by default.
- **Unassigned**, named "?.n", is a placeholder for a place where the user has yet to assign a model shape. This is the one component that appears when the user creates a new model.
- The connectors do not introduce any parameters into the model, but simply allow the user to add spacing to keep the diagram uncluttered and connected.

For a model to be estimable, all paths must end with either an endgate or dual endgate.

Below the Model Design Shapes are three buttons:

- **Add Release Group** allows the user to create another release group and a path that is parallel to the initial one.
- **Change Title** is used to change the name of the current model that appears at the top of the diagram. The default at startup is "New Diagram."
- **View Likelihood Equation** is used to view the likelihood equation text that is generated by the current diagram. Program Branch uses the computational engine of Program USER to perform the

actual parameter estimation. The format of the likelihood equation is described in Program USER user's manual found at http://www.cbr.washington.edu/analysis/apps/user.

Figure 3 shows a hypothetical study that we will model to illustrate how to create a model in Program Branch. The study consists of a single release of individuals (R) that travel a common path, then diverge into two paths, and then merge into one path again.

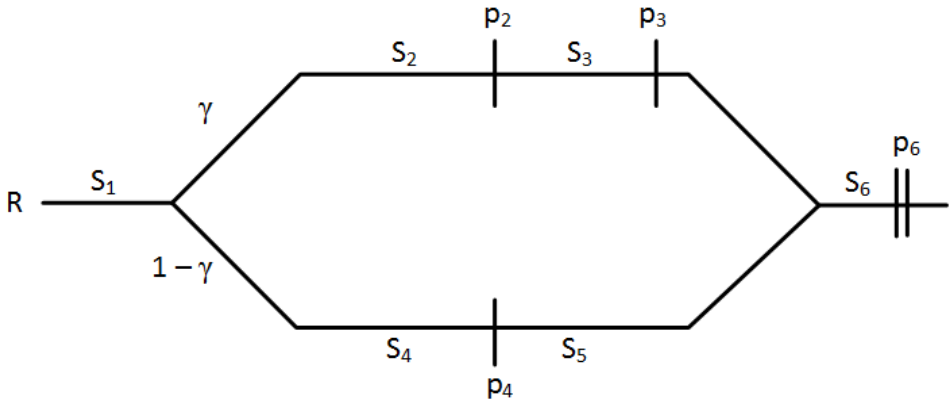


Figure 3. Hypothetical study design to illustrate modeling in Program Branch

- The S_i symbols represent a survival probability for an interval of time or space. These stretches will be modeled using the Stretch shape
- The vertical bars represent detection arrays, and the P_i's are the detection probabilities for the arrays. These will be modeled using the Gate shape.
- The final double-bar represents a double-array—two arrays close enough together that there is essentially no mortality between them. This allows an independent estimate of detection probability without the need for subsequent detections. The double-array will be modeled using the Dual Endgate shape. Note that there is also an Endgate shape that will not be used in this example that represents a double-array that is not the final detection point in a path.
- The probability of taking the upper path of the fork is represented by gamma (γ); hence the probability of taking the lower path is $1-\gamma$.

Note that these parameters are not separately estimable; only combinations of the parameters are estimable, as will be explained in Chapter 5.

3.1 Creating the Diagram

Figure 4 shows the model diagram at startup and when a new model is created. Every new model diagram starts with a Release ("R.1") followed by an Unassigned ("?.2"). The Unassigned is surrounded by a blue box, indicating that it is the currently selected shape. When the user selects a shape from the table of available shapes on the left side, it replaces the currently selected shape.

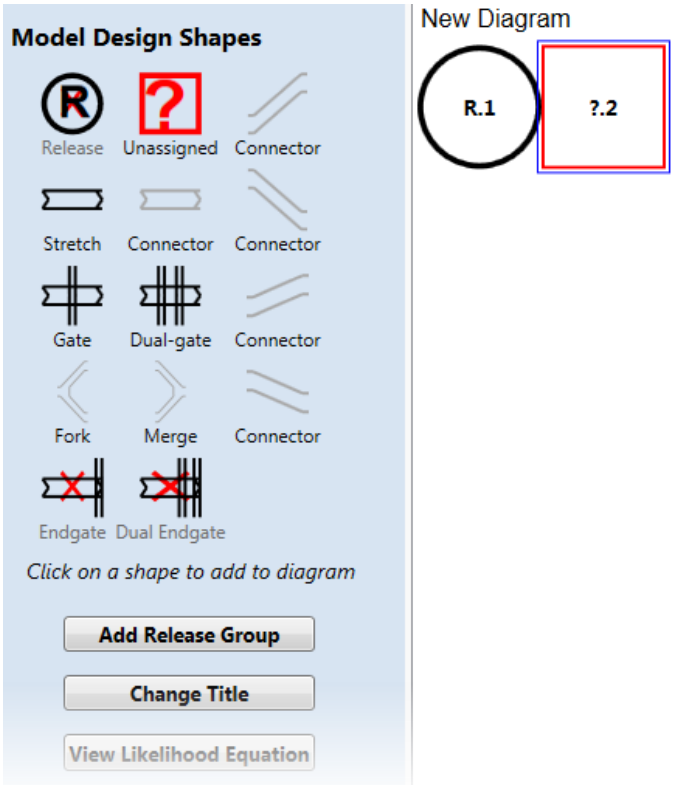


Figure 4. The diagram for starting a new model

The user clicks on the Stretch shape to represent the initial interval (S_1 in Figure 3). The diagram now appears as in Figure 5—the added Stretch shape is labeled S.2, and a new Unassigned shape ("?.3") is added and is currently selected.

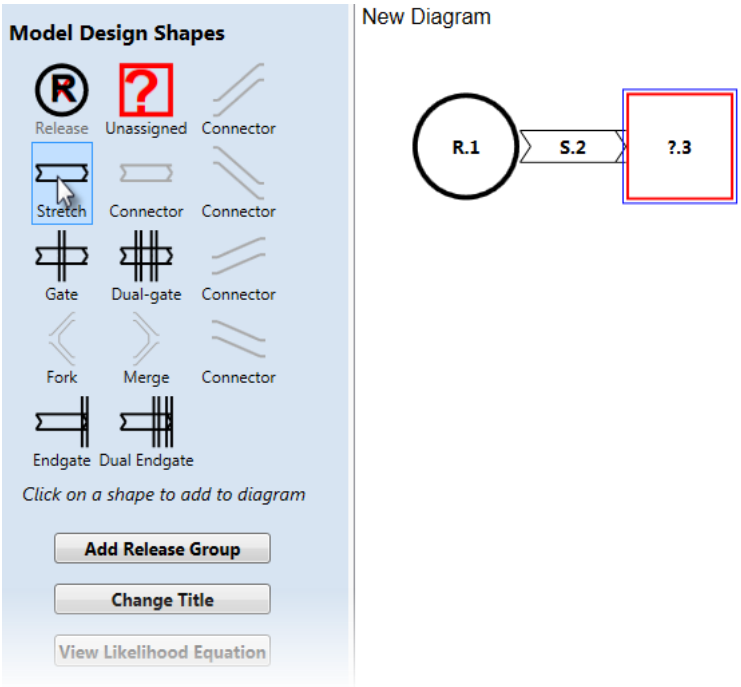


Figure 5. The diagram after adding a Stretch Shape

Figure 6 shows the diagram after a Fork has been added. There are now two unassigned shapes, one at the end of each path after the Fork. The bottom Unassigned shape is the one currently selected (as indicated by the blue box), and the user can now add shapes to the bottom fork. To add shapes to the top fork the user must click on "?.4" to select it, and then begin adding shapes from there.

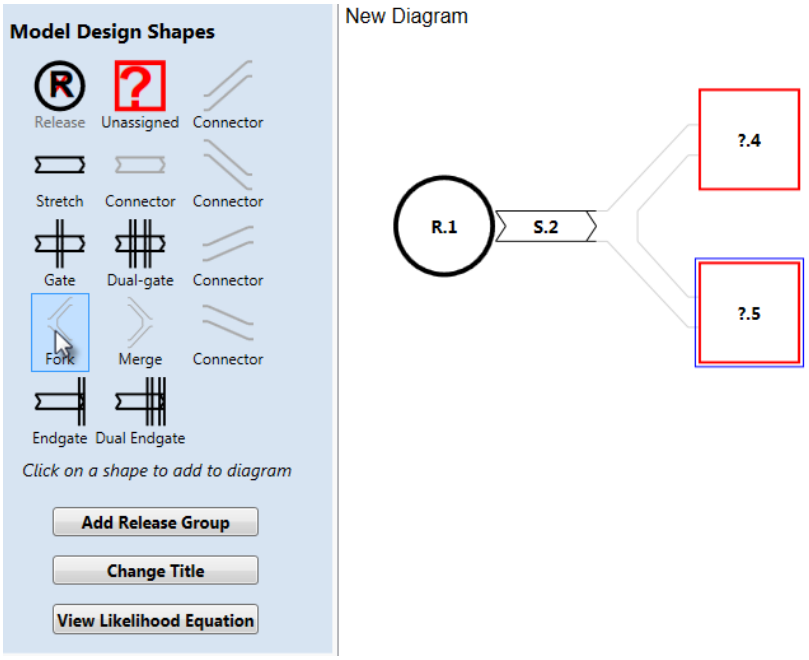


Figure 6. The diagram after a Fork has been added

Figure 7 shows the diagram after both the upper path and lower path have been completed. The two detection arrays on the upper path from Figure 3 are represented by the gates labeled G.9 and G.11. Similarly, the one detection array on the lower path is represented by the gate labeled G.6.

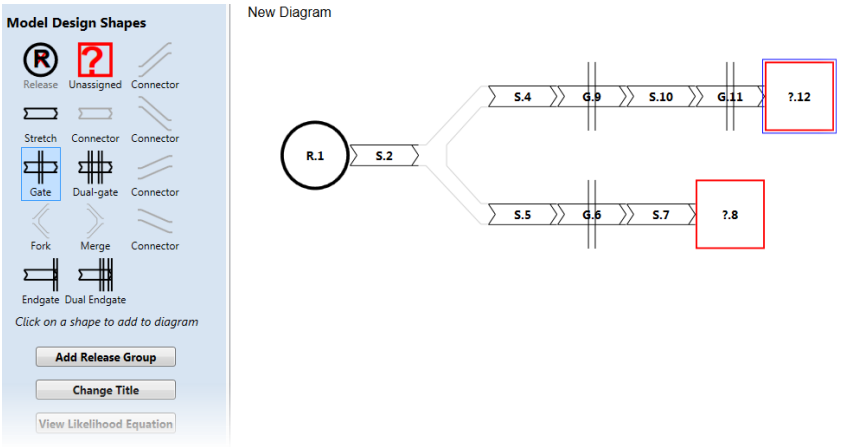


Figure 7. Diagram after completing both forks corresponding to Figure 3

The user now wants to merge the two paths into one by adding a Merge shape. For this to work, however, they need to be of the same length. This can be done by adding a straight connector to the bottom path as shown in Figure 8.

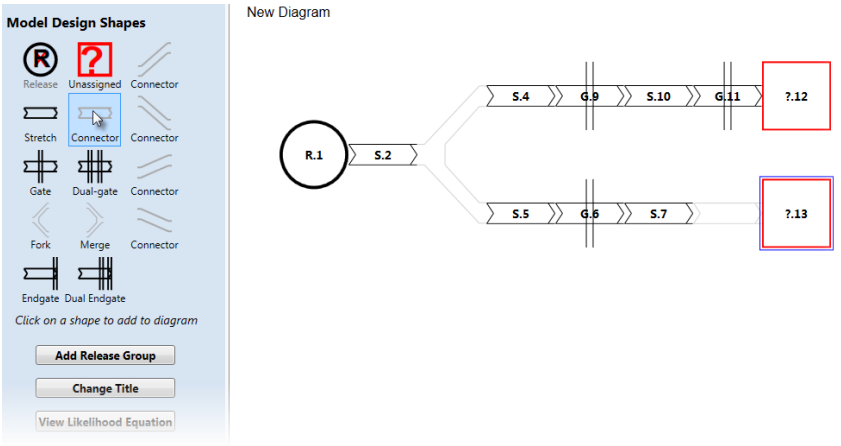


Figure 8. Diagram after adding straight connector to make both paths the same length

The user can now select the unassigned shape "?.12" and click on the Merge shape to bring the two paths to one as shown in Figure 9.

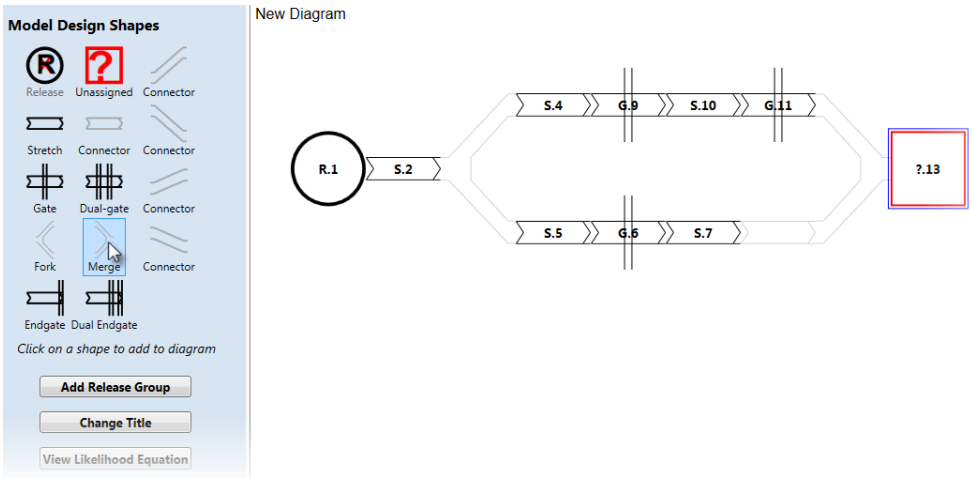


Figure 9. Diagram after adding the Merge shape

The user now completes the diagram by clicking on a Stretch and then the "Dual Endgate" to complete the diagram as shown in Figure 10.

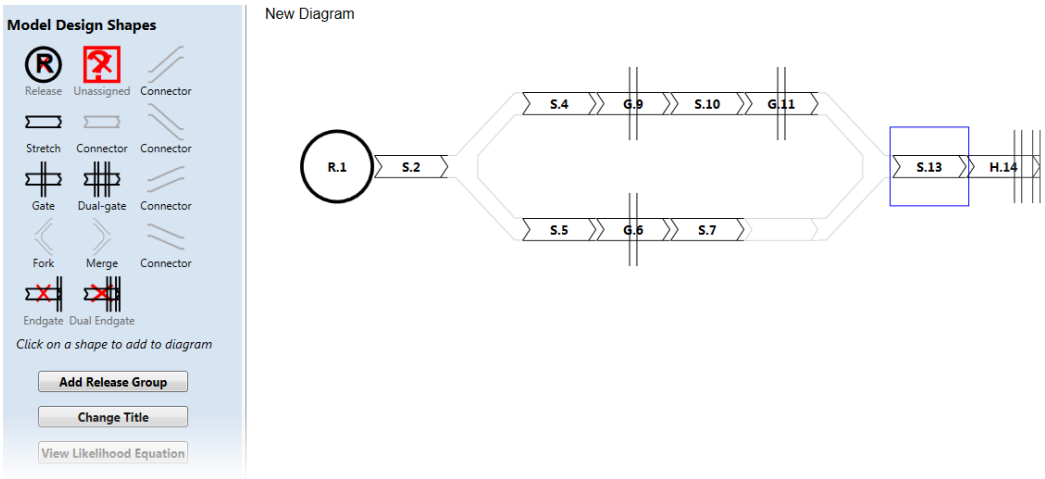


Figure 10. Completed diagram representing the model from Figure 3

3.2 Using Connectors

The available shapes in Program Branch include five that are labeled "Connector." These do not add any parameters to the model but are included to aid in creating the model diagram. There is one straight connector and four that are used to move the path up or down to varying degrees. The previous section shows how to use a straight connector to add length to a path.

In the example in Figure 11, the user tried to add a Fork at the currently selected unassigned shape "?.9." Since there was no room in the diagram for a fork at the current location, the user got an error message saying "Change was not successful probably due to positioning. Try a different location."

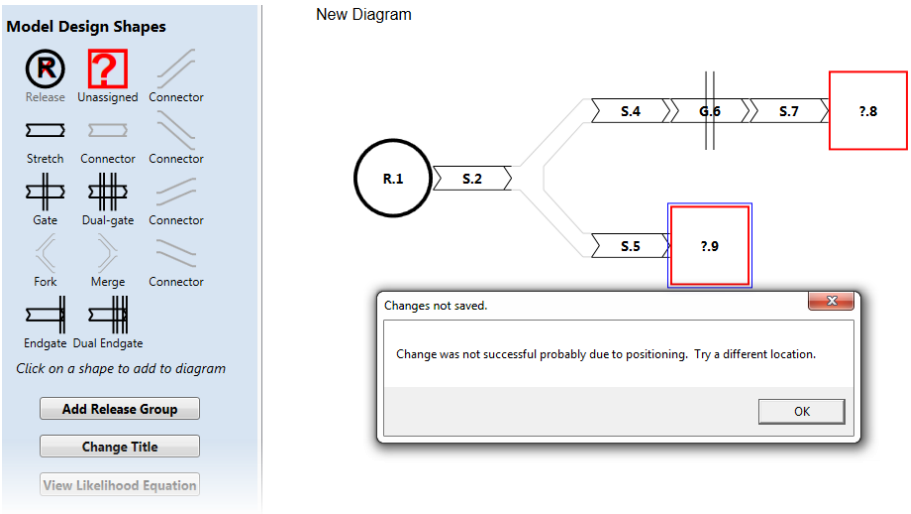


Figure 11. Model Diagram after attempting to add a Fork where there is insufficient room

To remedy this situation, the user adds a downward connector as shown in Figure 12. The user can then add the desired fork as shown in Figure 13.

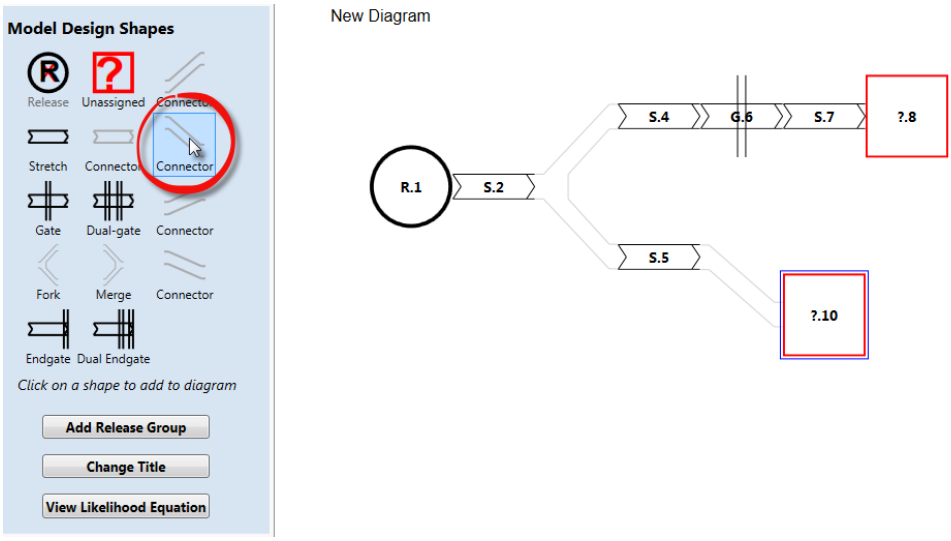


Figure 12. Model Diagram after adding a downward connector

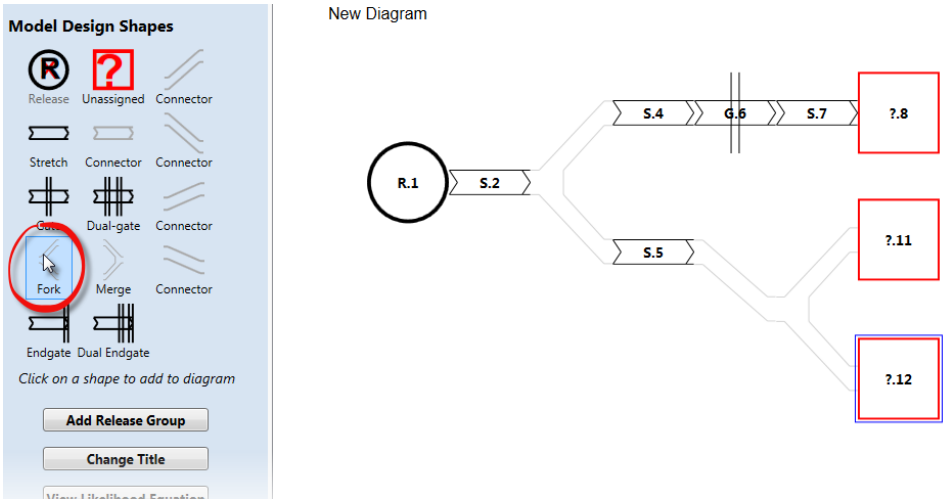


Figure 13. Model Diagram after adding Fork

3.3 Connecting Components

There may be cases where there is a need to connect one component to another to connect two parts of a path, but it's not possible to use the usual connectors due to the layout of the diagram. Figure 14 shows a somewhat contrived example where the user wants to join the path form "?.21" to "?.9." To do so, the lower path from "S.2" needs to be crossed.

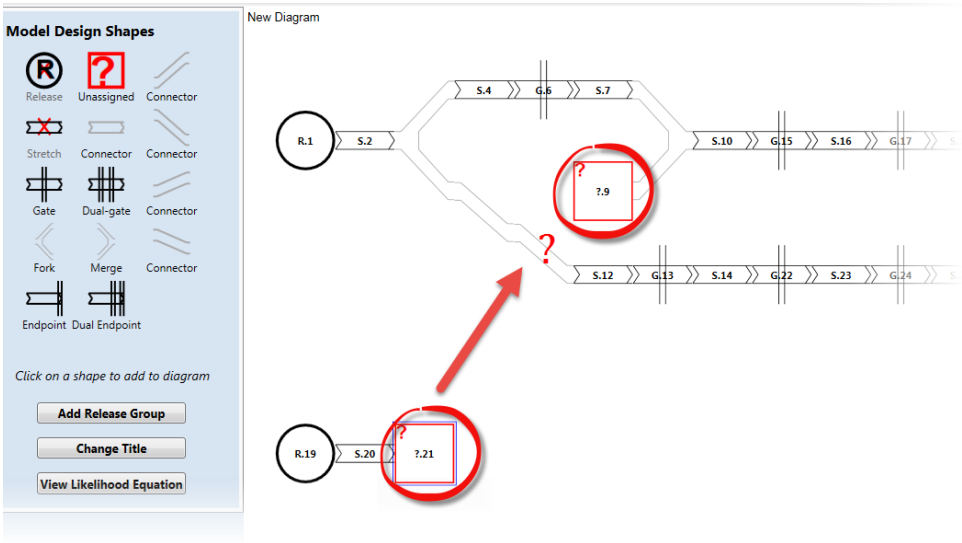


Figure 14. Example where a connector is needed

In order to make the connection, right-click on "?.21, and Select "Connect Component…" and the "Connect to ?.9" as shown in Figure 15. If there were more than one available pathway, they would all be shown in the menu. After making the connection, the diagram appears as in Figure 16.

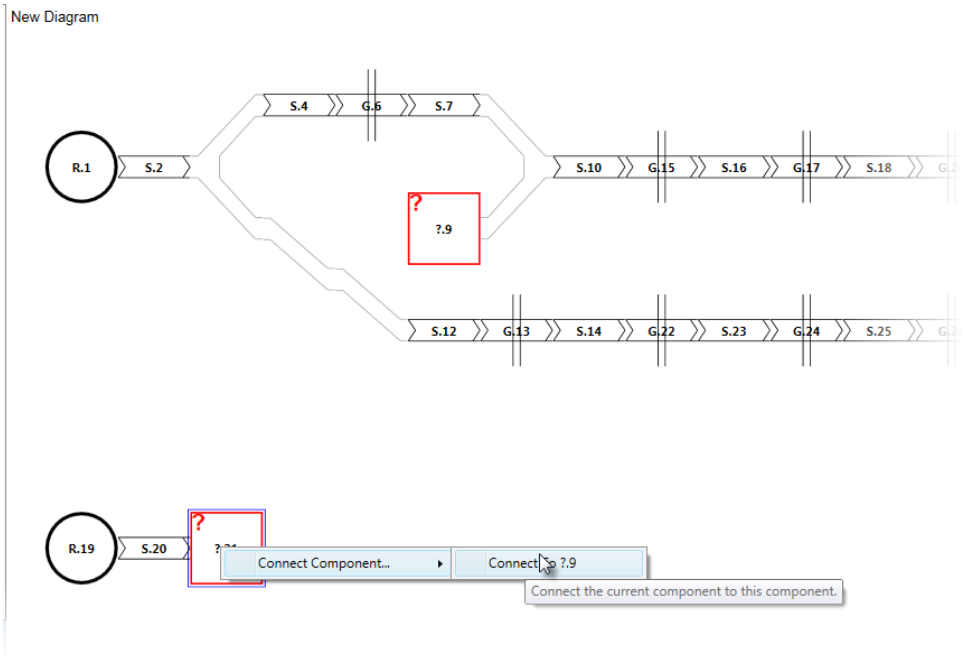


Figure 15. Making a connection

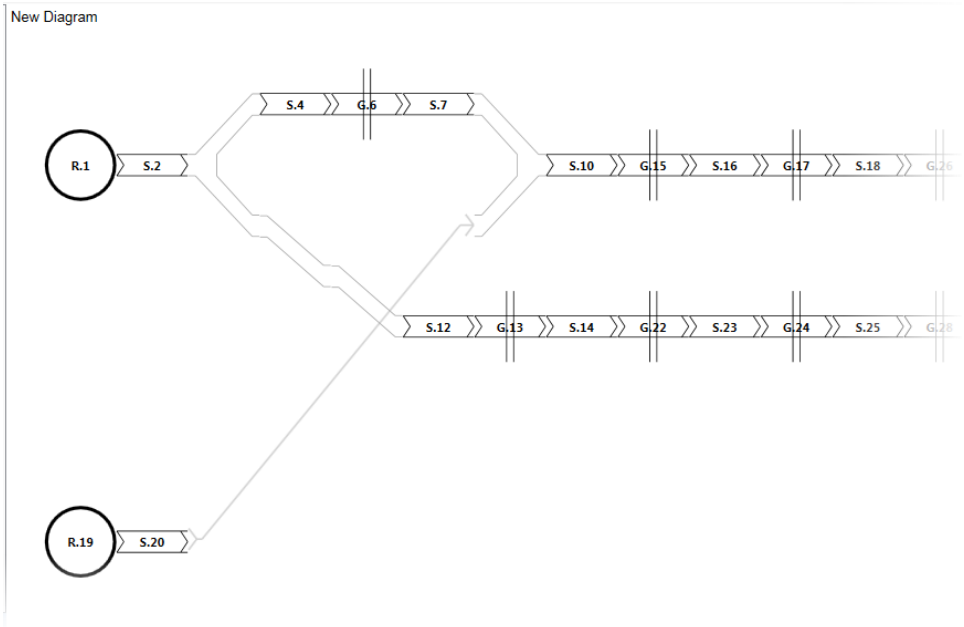


Figure 16. Connection made to another component

Note that this type of connection can only be made between an unassigned upstream component and an unassigned downstream component.

3.4 Selecting and Renaming Components

Figure 17 highlights the area in the lower-left area of Program Branch where components can be selected and renamed. In this example the "G.5" component has been selected under "Components," causing the G.5 gate to be selected on the diagram, and its information being displayed under "Selected Component." Notice that each component has both an ID and a Name. By default, the name is the same as the ID.

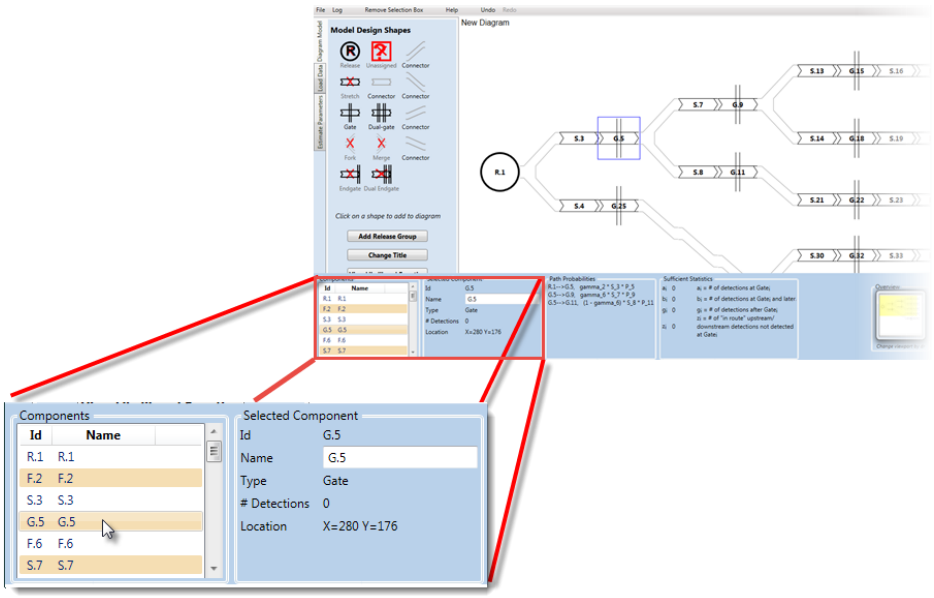


Figure 17. Component Selection and Renaming area of Program Branch

The user can change the name in order to make the diagram more readable by editing the "Name" field under "Selected Component" as shown in Figure 18. The new name now appears on the diagram as shown in Figure 19. Component names cannot contain spaces. Be aware that all path probabilities and area probabilities (described below) will use the original ID—not the user-defined name.

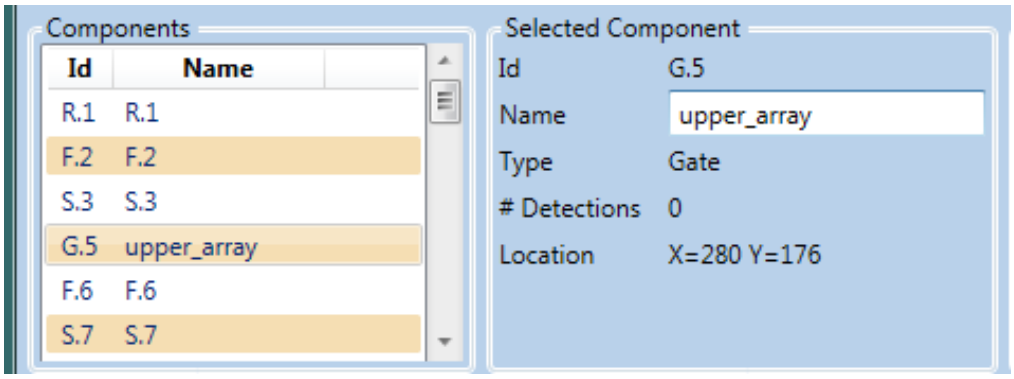


Figure 18. Changing the name of a Gate component from G.5 to "upper_array"

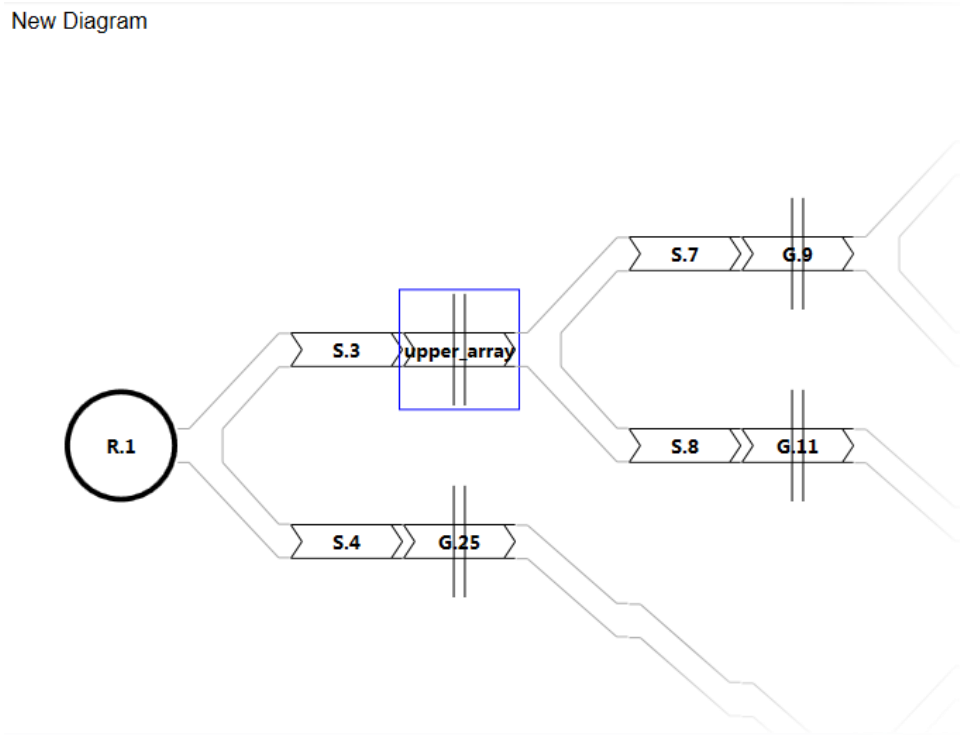


Figure 19. Diagram with the name of a component changed

3.5 Other Diagram Editing and Viewing Commands

- **Undo Command:** There is an "Undo" menu item that allows the user to undo the last command.
- **Remove Selection Box:** The "Remove Selection Box" menu item removes the blue selection box from the diagram. This is useful for screen captures when you don't want the selection box on the diagram. Once the user clicks on the diagram again, the selection box reappears.
- **Change the viewport:** For large diagrams that do not fit on the screen, the user can change the portion currently visible using the "Overview" box in the lower right. Simply use the mouse to drag the yellow area to cover the area you wish to view. Alternatively, the user may drag the diagram directly using the mouse.
- **Zoom in and out:** Use the mouse wheel to make the diagram larger and smaller.
- **Centering the diagram:** The diagram can be centered around a desired point by double-clicking on the point.

A user can move a component by right-clicking on it and selecting "Remove component" as show in Figure 20.

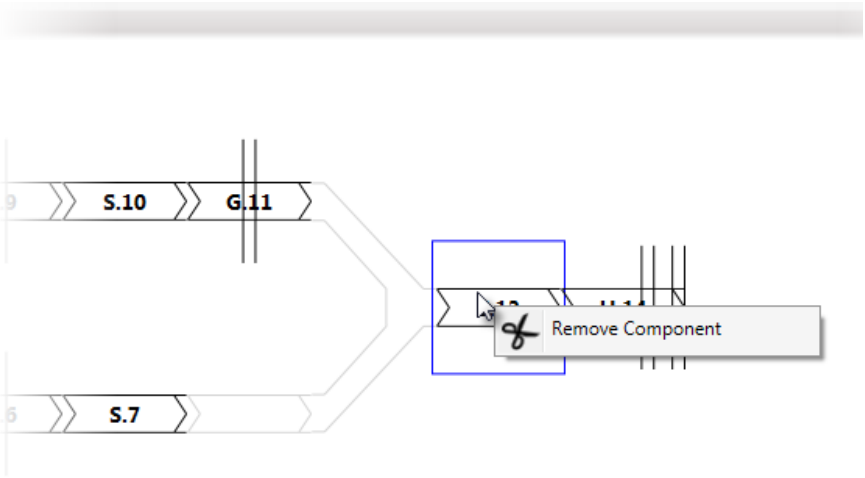


Figure 20. Removing a Component

3.6 Path Probabilities

When a user selects either a release or a gate on the model diagram, a "Path Probabilities" area appears at the bottom as shown in Figure 21. The Path Probabilities area shows the probabilities for all paths to and from consecutive releases and gates that start or end at the selected gate or release and are detected at the end gate. For gate G.15 in Figure 21, there are probabilities for three paths: (1) G.8 to G.15, (2) G.10 to G.15, and (3) G.15 to E.17.

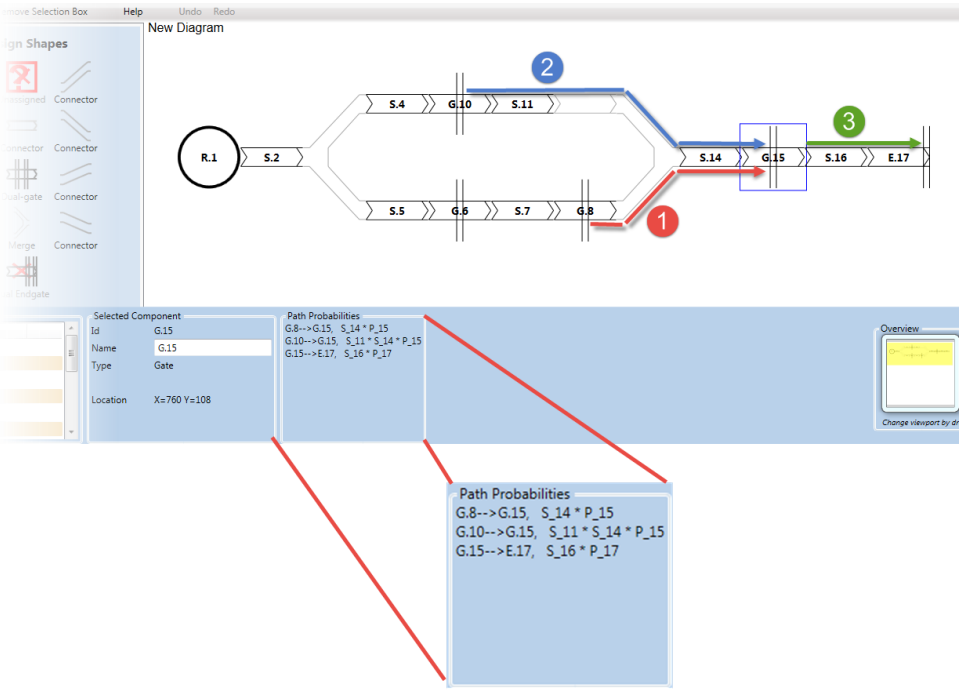


Figure 21. "Path Probabilities" area displayed for the selected gate G.15

All path probabilities are expressed in terms of the following:

- **S_n:** The probability of survival from the start of Stretch S.n to the end of the stretch,
- **P_n:** The probability of detection at gate G.n (or E.n),

- **gamma_n:** The probability of taking the top-most path of a fork. The probability of taking the corresponding bottom path is 1 – gamma_n.

Note that these probabilities are not separately estimable; only the combinations of these parameters are estimable as will be explained below in Chapter 5.

For example, in Figure 21, the probability of the path labeled "2" is S_11*S_14*P_15, that is the probability of surviving stretches S.11 and S.14, and then being detected at G.15.

In Figure 22, the gate G.6 is selected, and there are two path probabilities: (1) R.1 to G.6, and (2) G.6 to G.8. For the path R.1 to G.6, the probability is S_2*(1-gamma_3)*S_5*P_6, which is the probability of surviving the stretch S.2 (S_2) times the probability of taking the bottom path of the fork (1 – gamma_3) times the probability of surviving the stretch S.5 (S_5) times the probability of being detected at G.6 (P_6).

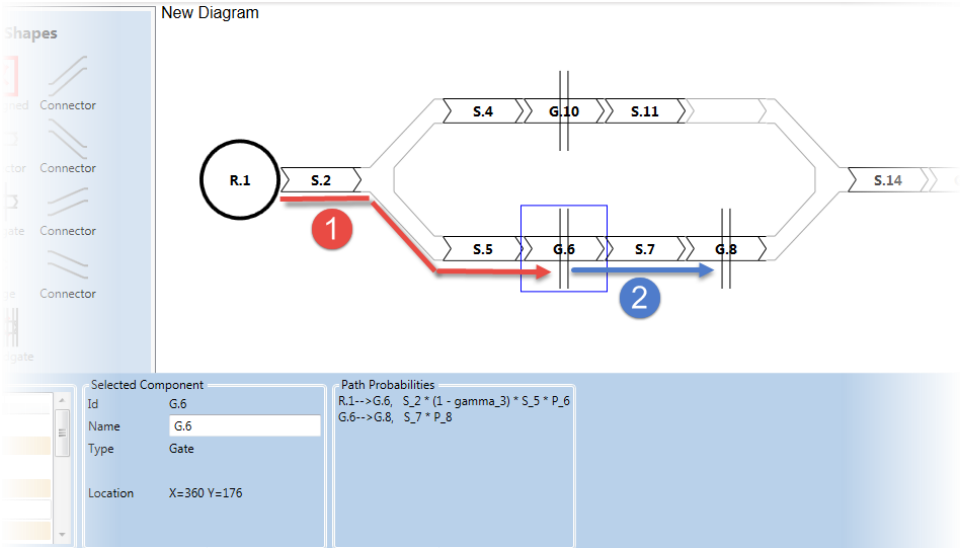


Figure 22. Path Probabilities that include the "gamma" probability at a fork

Chapter 4: Load Data

Figure 23 shows the completed diagram representing the model from Figure 3. This chapter will demonstrate loading data for the model into Program Branch.

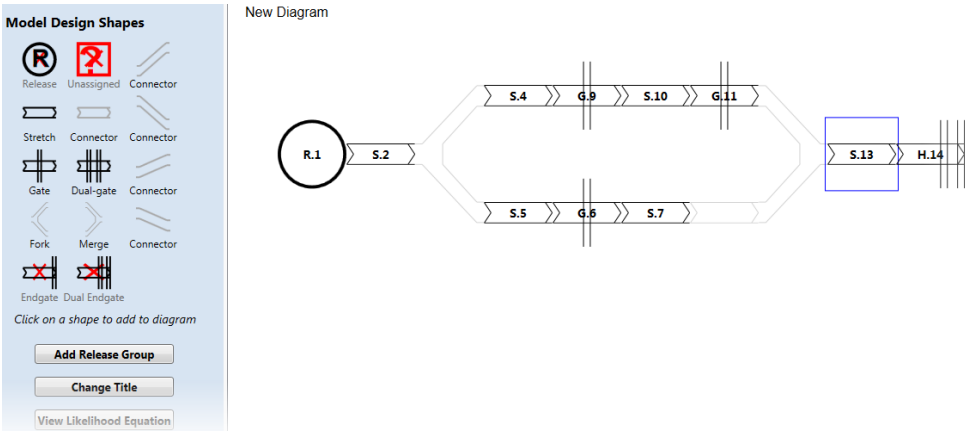


Figure 23. Example model diagram to illustrate loading data into Program Branch

The user clicks on the "Load Data" tab (Figure 24) to load data.

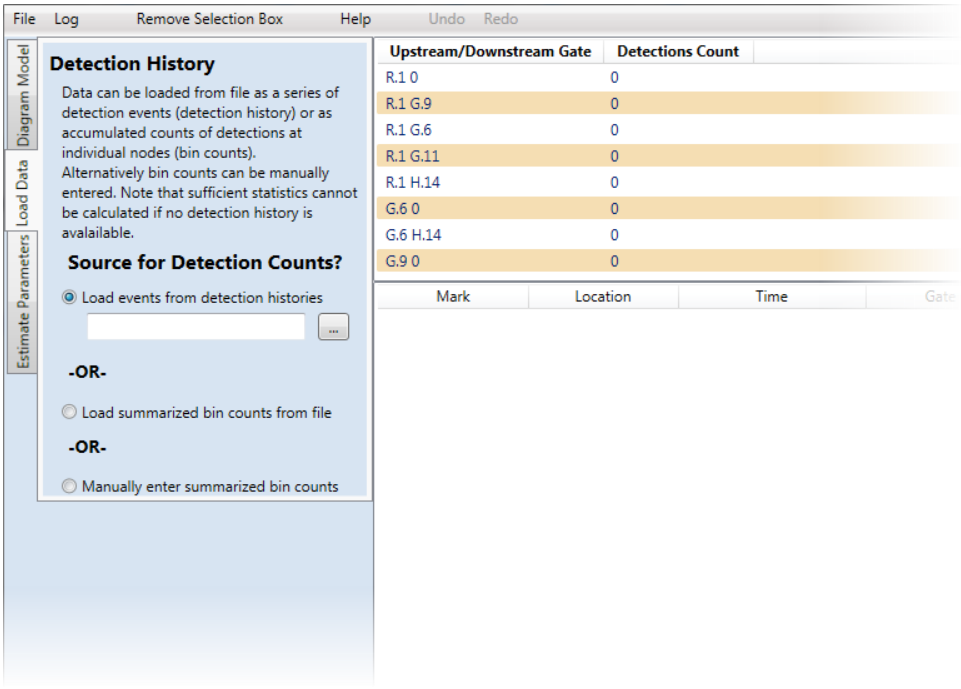


Figure 24. Left portion of the Load Data page of Program Branch

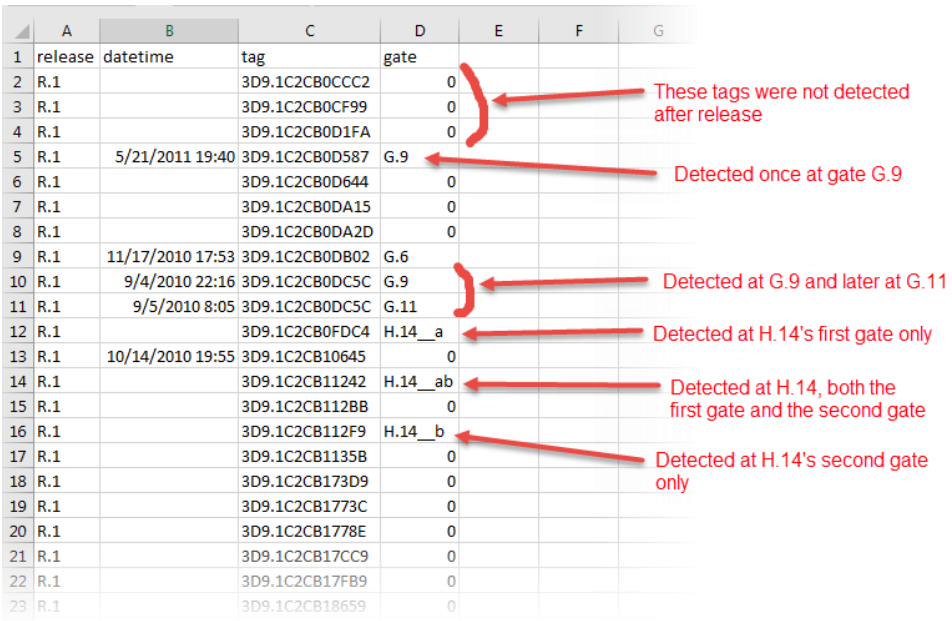
Program Branch allows data to be loaded in three ways: (1) Load events from detection histories, (2) load summarized bin counts, or (3) manually enter summarized bin counts. When the user selects either "Load events from detection histories" or "Load summarized bin counts from file," they will be prompted to select the appropriate input file. If "Manually enter summarized bin counts" is selected, the user will be presented with a form for manually entering the bin counts.

4.1 Detection Histories File

The detection histories file is a Comma-Separated-Value (CSV) file. There is record for each detection event, and four columns: (1) release location, (2) detection time, (3) tag ID, and (4) detection gate.

Figure 25 shows the top portion of a detection histories file for the example of Figure 23. The release location has to correspond to one of the releases in the model—"R1" in this case. The detection time is optional. The detection gate field must be one of the gates from the diagram, or "0" (zero) indicating no detection. In this example, the first three tags, released from "R1," were never detected again after release. The tag on line 5 was detected once at "G.9." The tag in lines 10 and 11 was detected first at "G.9" and later at "G.11."

For dual gates, a detection at the first gate but not at the second is indicated by appending "__a" (two underscores followed by "a") to the gate name. Likewise, a detection at the second gate is indicated with "__b", and a detection at both gates by "__ab". In the example in Figure 25, the tag on line 12 was detected only at the first gate of "H.14"—similarly with the tags at lines 14 and 16.



	A	B	C	D	E	F	G
1	release	datetime	tag	gate			
2	R.1		3D9.1C2CB0CCC2	0			
3	R.1		3D9.1C2CB0CF99	0			
4	R.1		3D9.1C2CB0D1FA	0			
5	R.1	5/21/2011 19:40	3D9.1C2CB0D587	G.9			
6	R.1		3D9.1C2CB0D644	0			
7	R.1		3D9.1C2CB0DA15	0			
8	R.1		3D9.1C2CB0DA2D	0			
9	R.1	11/17/2010 17:53	3D9.1C2CB0DB02	G.6			
10	R.1	9/4/2010 22:16	3D9.1C2CB0DC5C	G.9			
11	R.1	9/5/2010 8:05	3D9.1C2CB0DC5C	G.11			
12	R.1		3D9.1C2CB0FDC4	H.14__a			
13	R.1	10/14/2010 19:55	3D9.1C2CB10645	0			
14	R.1		3D9.1C2CB11242	H.14__ab			
15	R.1		3D9.1C2CB112BB	0			
16	R.1		3D9.1C2CB112F9	H.14__b			
17	R.1		3D9.1C2CB1135B	0			
18	R.1		3D9.1C2CB173D9	0			
19	R.1		3D9.1C2CB1773C	0			
20	R.1		3D9.1C2CB1778E	0			
21	R.1		3D9.1C2CB17CC9	0			
22	R.1		3D9.1C2CB17FB9	0			
23	R.1		3D9.1C2CB18659	0			

Figure 25. An example of a detection histories file shown in Microsoft Excel (2013)

4.2 Bin Counts File

A bin counts file is a text file. An example corresponding to the model in Figure 23 is shown in Figure 26. Each bin consists of a starting point (a release or gate) and a next detection point (gate). Each bin is followed by a colon (":") and then the number of counts for the bin. All possible bins (all releases and gates with all possible next detection sites) must be included in the bin counts file, even if the count is zero. The counts for a bin include all tags that were released/detected at the starting point and next detected at the end point.

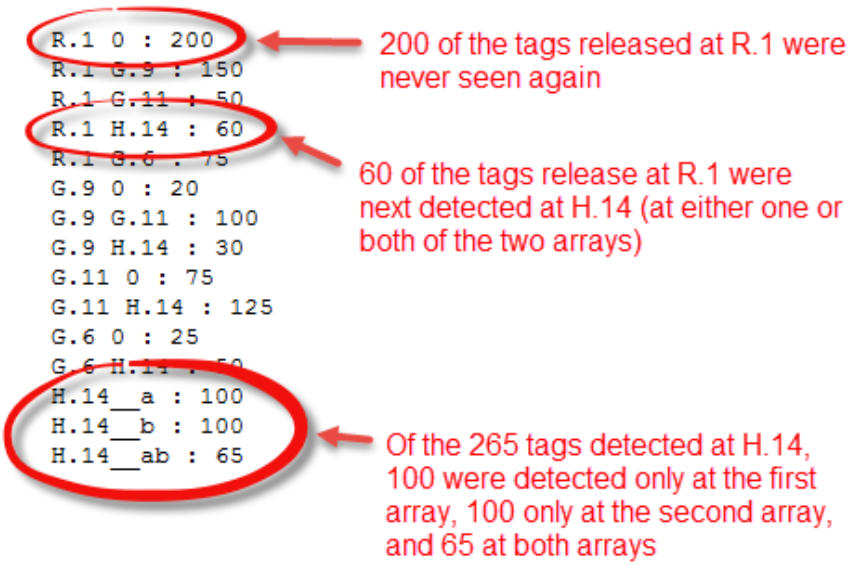


Figure 26. Bin counts file corresponding to the model in Figure 23

For a dual gate (H.14 in this example), total detections are counted irrespective of which of the dual gates. There are then separate entries indicating which detections were at the first gate only, the second gate only, or both. For example, in Figure 26, 60 of the tags released from R.1 were next detected at one or both of the arrays at H.14; at the bottom are three lines indicating at which arrays the 265 tags at H.14 were detected.

4.3 Manually Entered Bin Counts

If the user selects to manually enter the summarized bin counts, a button will appear for entering the bin counts, that when pressed, will present an entry form as shown in Figure 27. The bins are defined as described in the previous section. The user can simply tab down or click in the Detections Count column on the right to enter the counts.

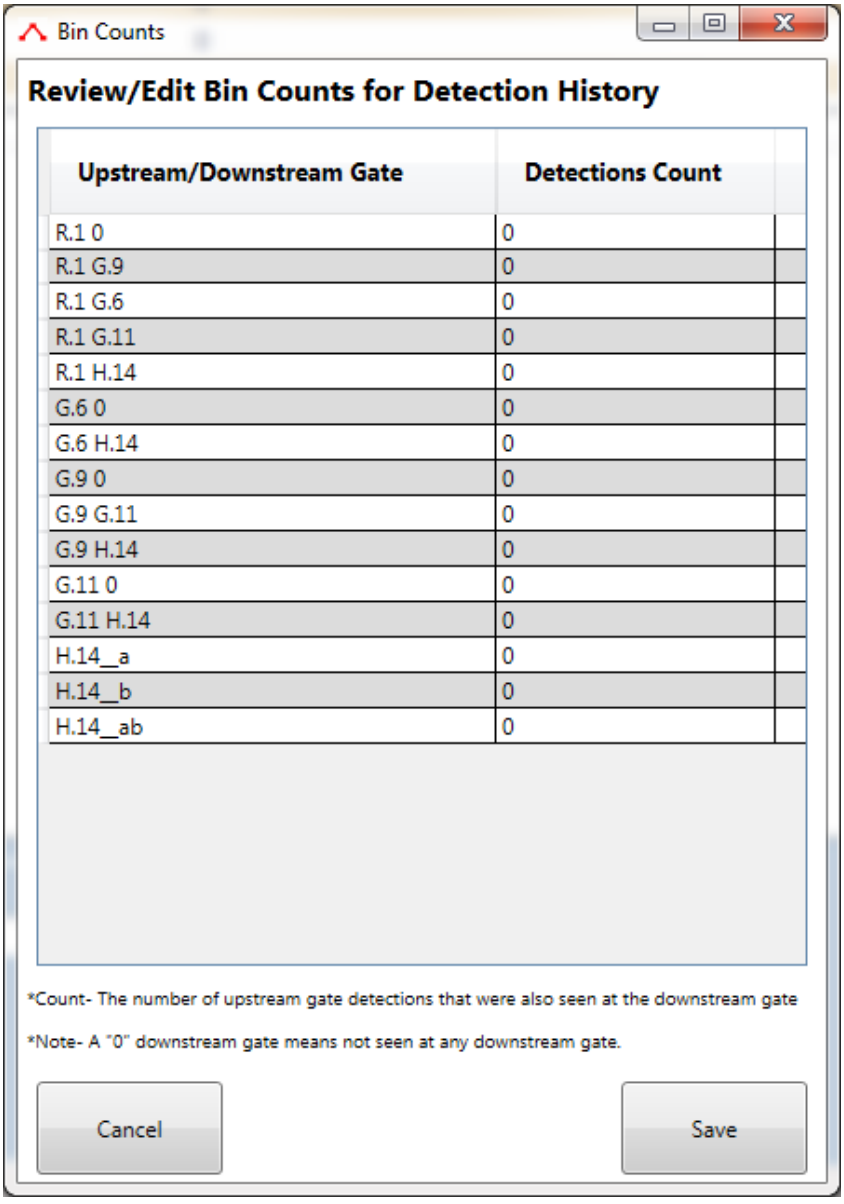


Figure 27. Form for manually entering summarized bin counts

4.4 Sufficient Statistics

After the data has been loaded into Program Branch, the user may click on any gate on the model diagram to view the sufficient statistics at the bottom, as demonstrated with the sample model in Figure 28. For a knowledgeable user, this can be an aid in determining which parameters are estimable and which are not. Note that all sufficient statistics are available only when the data are loaded from detection histories; if loaded from bin counts, only the number detected at the selected gate (a_i) is available.

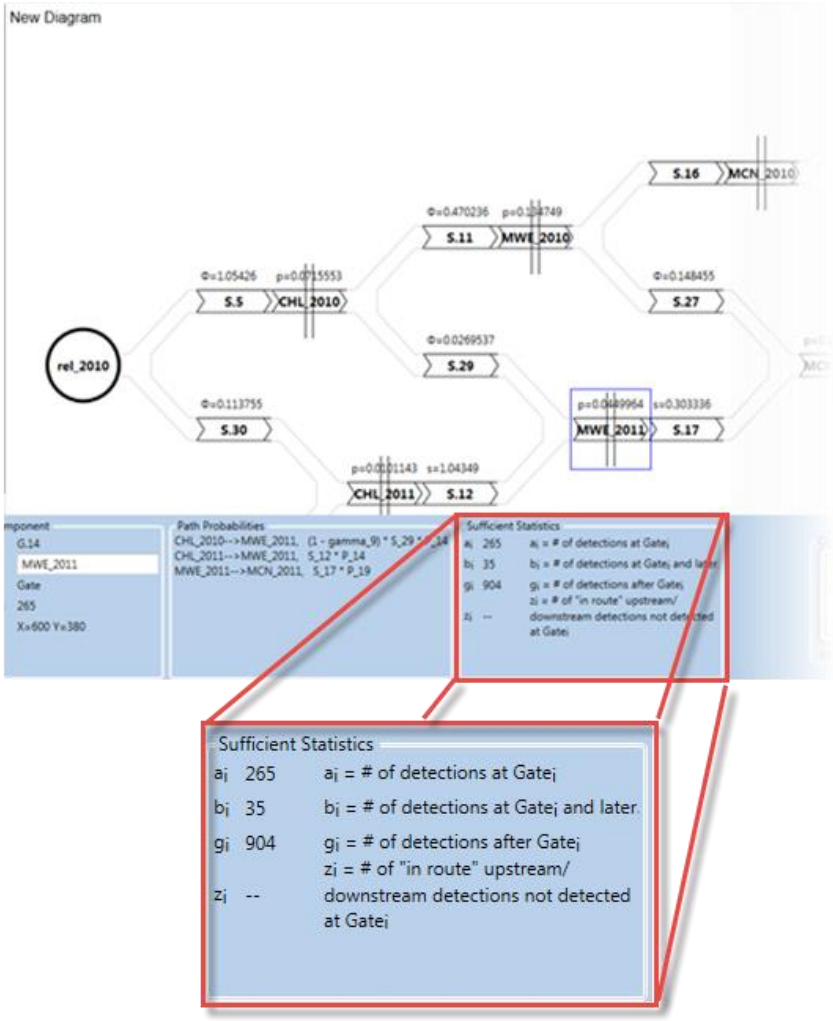


Figure 28. Sample model diagram with sufficient statistics for the selected gate highlighted

Chapter 5: Estimate Parameters

Once the model has been defined and the data entered, the user can click on the "Calculate Model Parameters" tab to estimate the model parameters. Figure 29 shows the left portion of the "Estimate Parameters" tab.

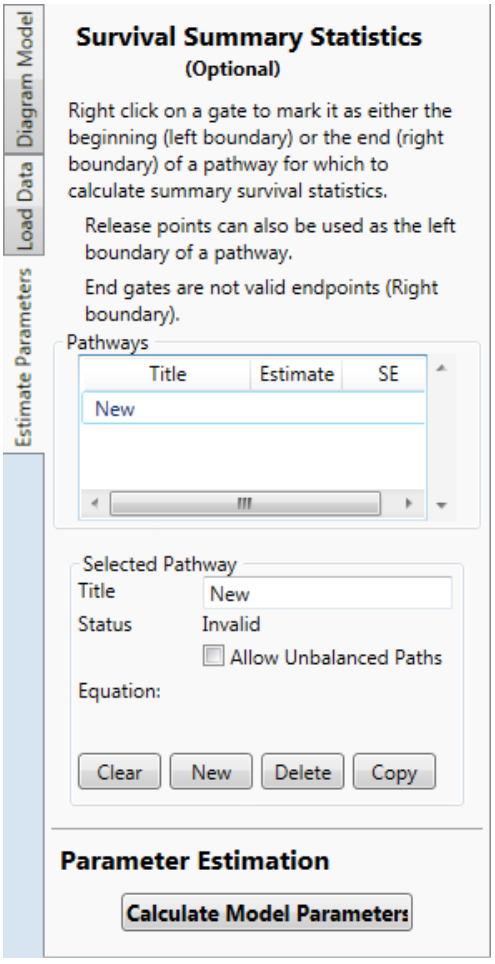


Figure 29. Left portion of the Estimate Parameters tab

5.1 Calculate Model Parameters

The Calculate Model Parameters button (shown at bottom on Figure 29) brings up the Estimation Dialog shown in Figure 30.

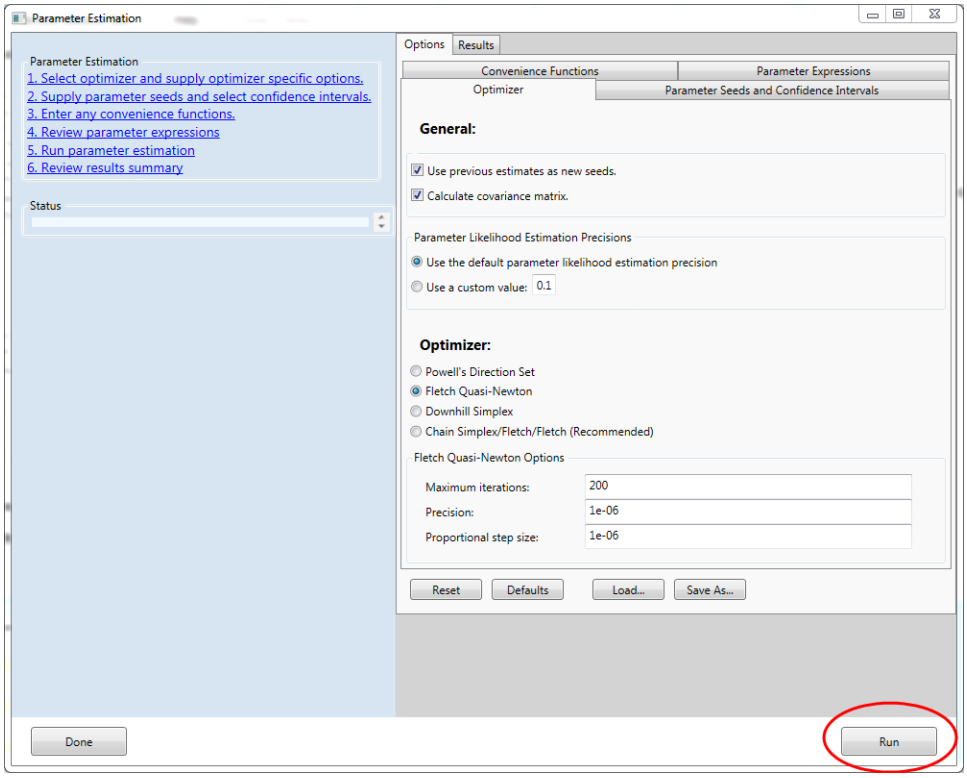


Figure 30. Estimation Dialog

The optimization options available on this dialog are explained in the Program USER manual, found at http://www.cbr.washington.edu/analysis/apps/user. For most cases, the user can accept the defaults and press the "Run" button on the lower right.

If the estimation completes successfully, the results tab is shown with two tabs beneath it: a "Summary" tab and a "Details" tab.

Figure 31 shows the Summary tab with a USER-style Estimation Summary Report.

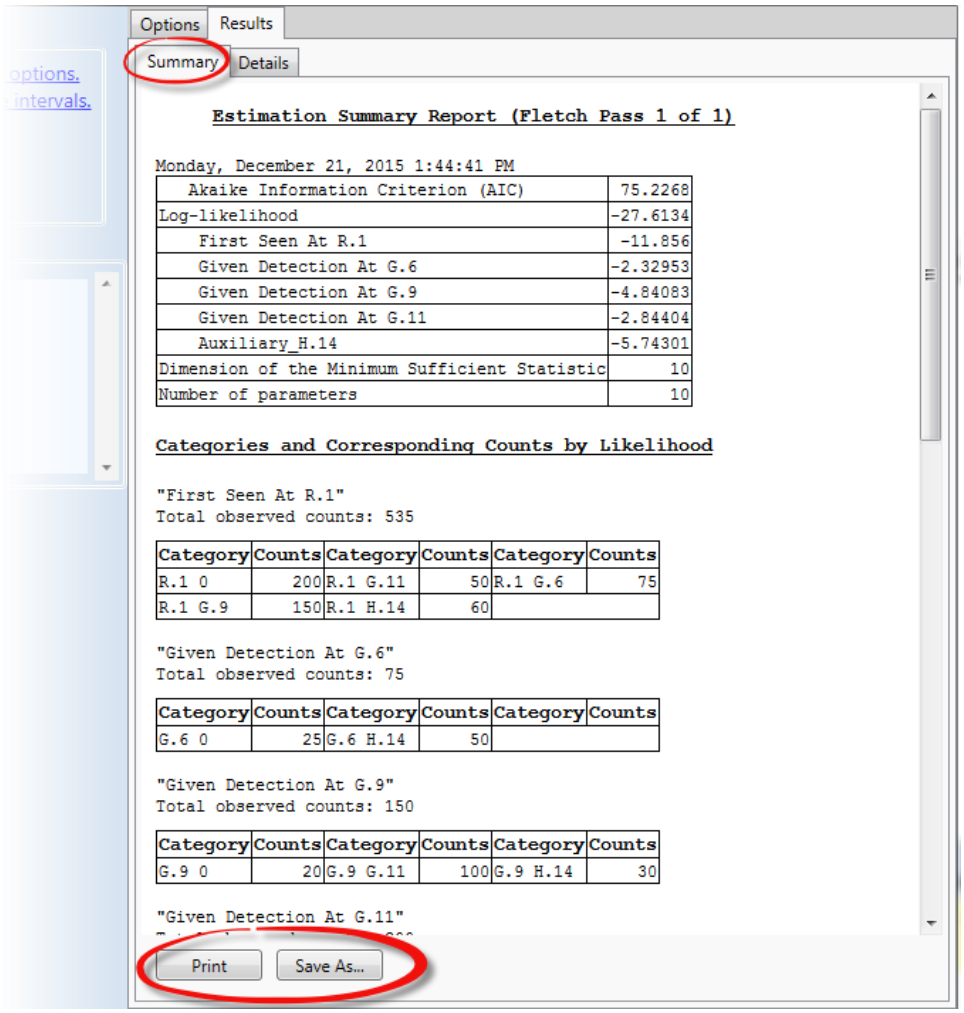


Figure 31. Estimation Summary Report

This report can be printed using the "Print" button or saved at an HTML report using the "Save As…" button (circled at bottom). Figure 32 shows the Details tab which gives the parameter estimates. These may be copied to the system clipboard ("Copy to Clipboard" button) or saved a CSV file ("Save to CSV" button).

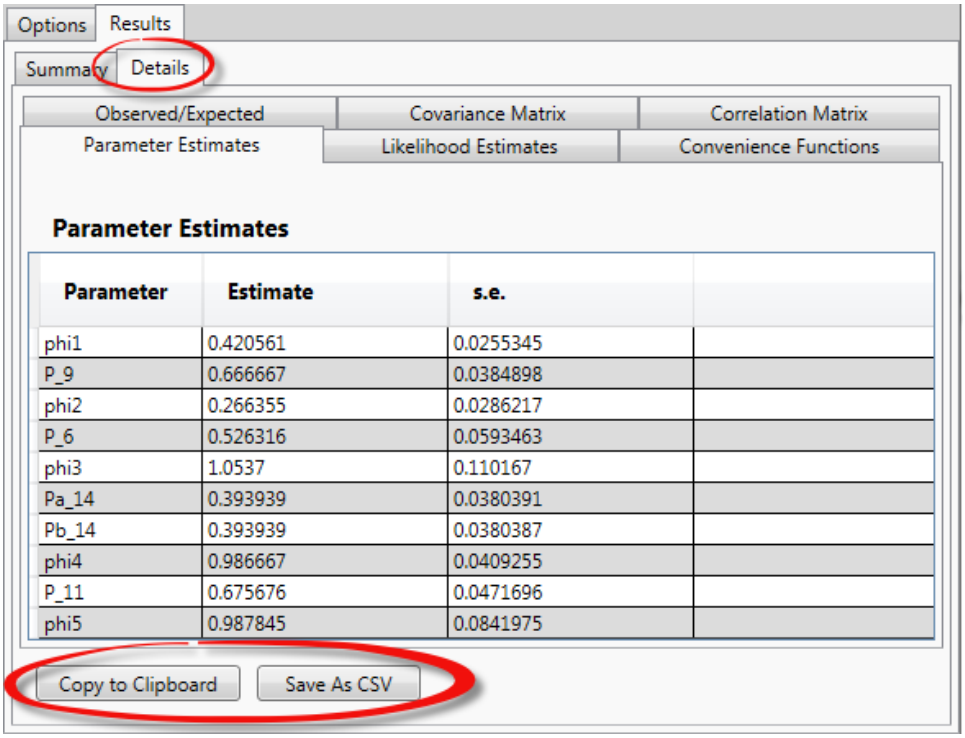


Figure 32. Estimation Details Report

As stated previously, the individual survival, detection and branching probabilities are not separately estimable. Instead, Program Branch estimates composite parameters as follows:

- The Greek letter phi (ϕ) represents the probability of selecting a particular branch in the pathway and surviving one or more stretches and branching.
- "s" represents the probability of surviving a stretch (with no branching involved),
- "p" represents the probability of being detected at a gate,

- The Greek letter lambda (λ) represents the probability of surviving to and being detected at an end gate.

Once the user exits the Estimation Dialog by pressing "Done," the parameter estimates are displayed on the model diagram, as shown in Figure 33 (note that the λ estimates are not shown). In this example, the probability of surviving S.2, taking the upper branch, surviving S.4, and being detected at S.4 is 0.4206 (circled in red); the probability of being detected at G.6 is 0.5263 (circled in blue).

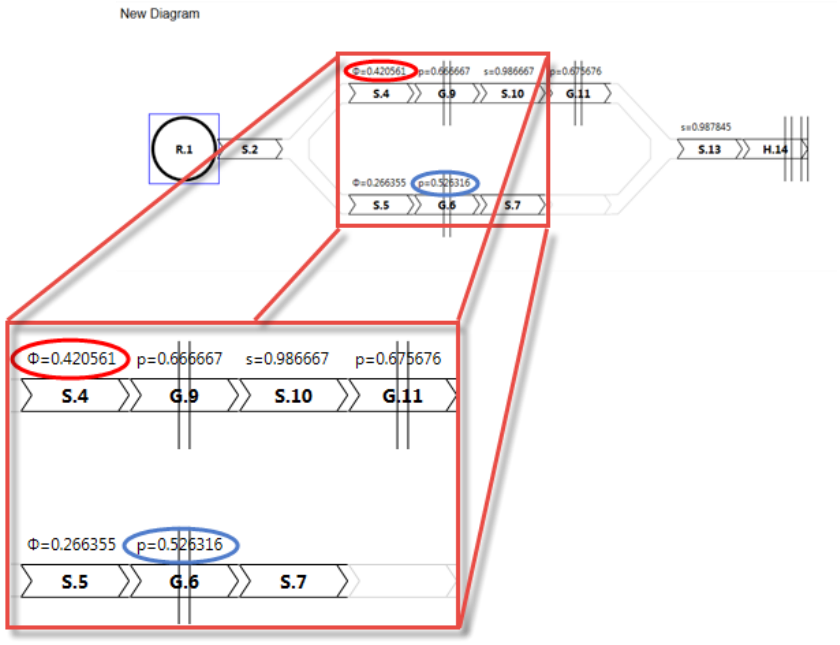


Figure 33. Model Diagram with parameter estimates shown (close-up view in box)

The standard errors can be observed by holding the mouse over an estimate as shown in Figure 34.

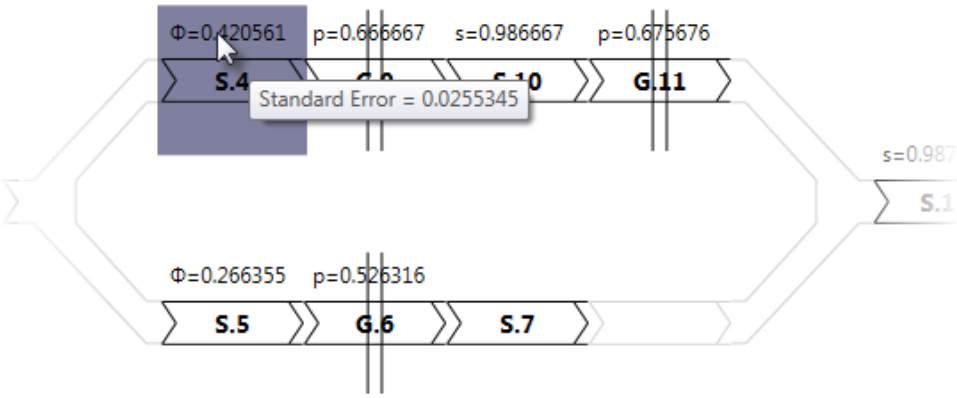


Figure 34. Parameter Estimates with standard error shown

5.2 Define Pathways for Estimation

Program Branch allows the user to define pathways for survival estimation on the Estimate Parameters tab. For example, if the user is interested in survival from release to the final detection site he or she would proceed as follows:

1. Right-click on R.1 and select "Mark Left Boundary" as shown in Figure 35. R.1 will now be highlighted in green.

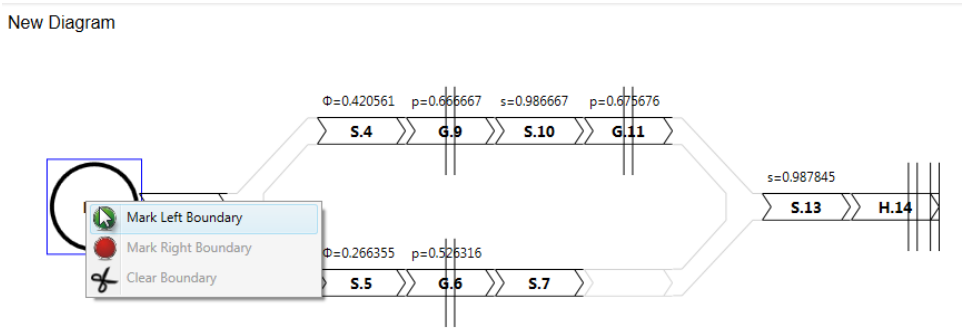


Figure 35. Mark the left boundary of a pathway of interest

2. Right-click on H.14 and select "Mark Right Boundary" as shown in Figure 36. H.14 will now be highlighted in red. All gates and stretches between the two endpoints will now be highlighted in yellow.

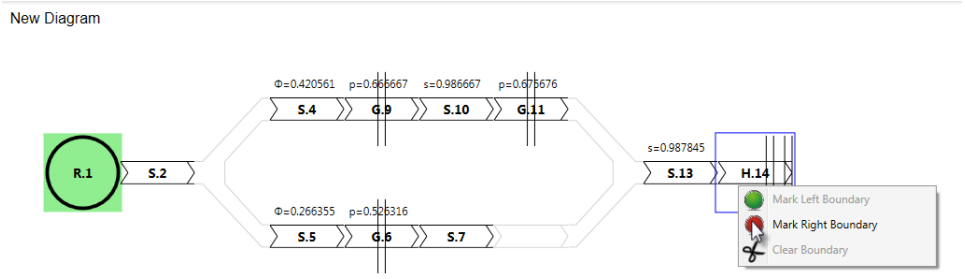


Figure 36. Mark the right boundary of a pathway of interest

3. In the left pane, under "Selected Pathway," change the name from "New" to something meaningful, such as "R1 to H.14" as shown in Figure 37.

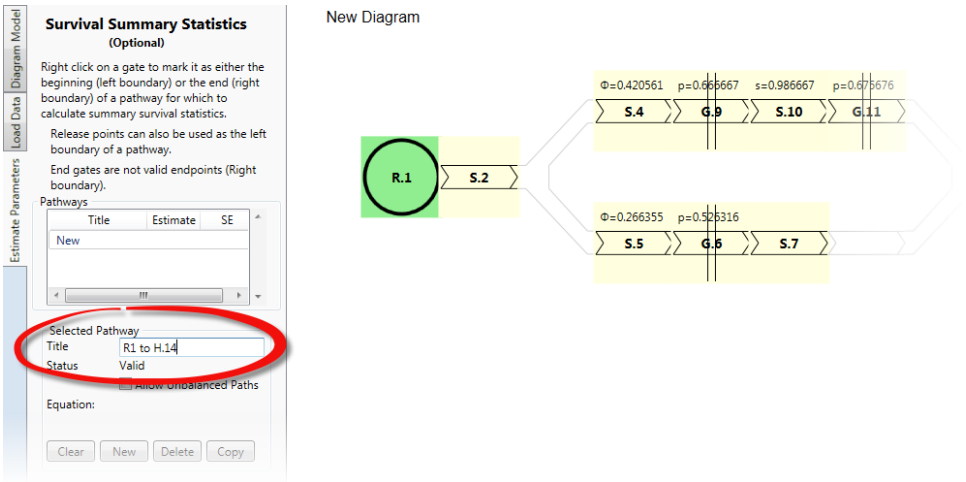


Figure 37. Changing the name of a pathway

Note that we were able to select the final gate (H.14) as a right boundary because it is a dual-gate array, allowing us to estimate survival separately from detection; if the end gate had been a single-array gate, we would only be able to estimate the product of survival and detection (λ), and Program Branch would not have allowed it to be used as a right boundary of a pathway.

The user may define multiple pathways of interest. To define a new pathway, click on the "New" button on the left and repeat the process described above. In Figure 38, the user has defined a pathway with two right boundaries, G.9 and G.6, and named the area "R1 to first gate." Note that multiple right boundaries may be specified, but only one left boundary for a pathway.

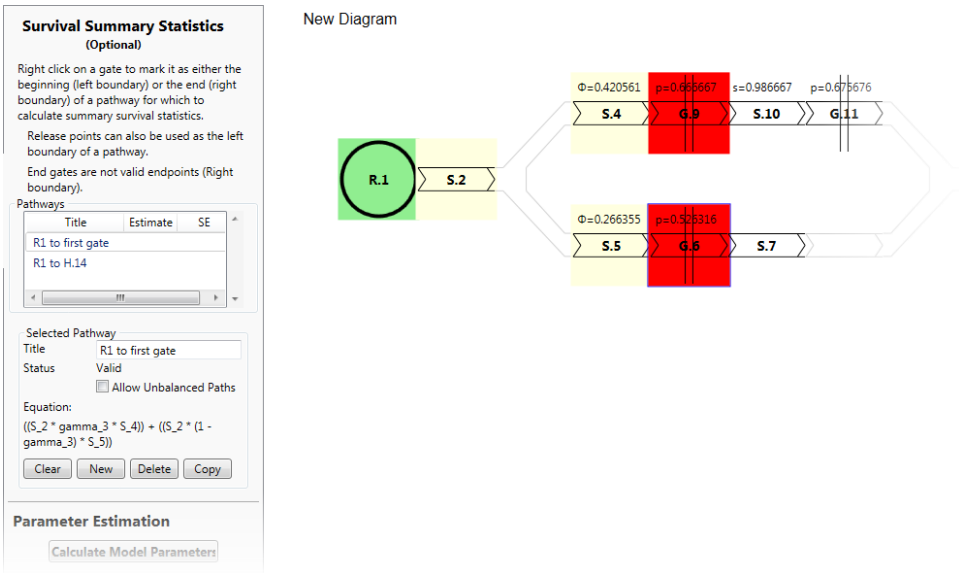


Figure 38. A pathway defined with multiple right boundaries

To estimate the probabilities for the defined pathways, repeat the estimation process by clicking on "Calculate Model Parameters" and proceed as described above. The table on the left side will now be filled in with the estimates and standard errors ("SE") filled in as shown in Figure 39. The user can then use the Copy button (circled) to copy the estimates to the clipboard to be copied into another application such as a spreadsheet.

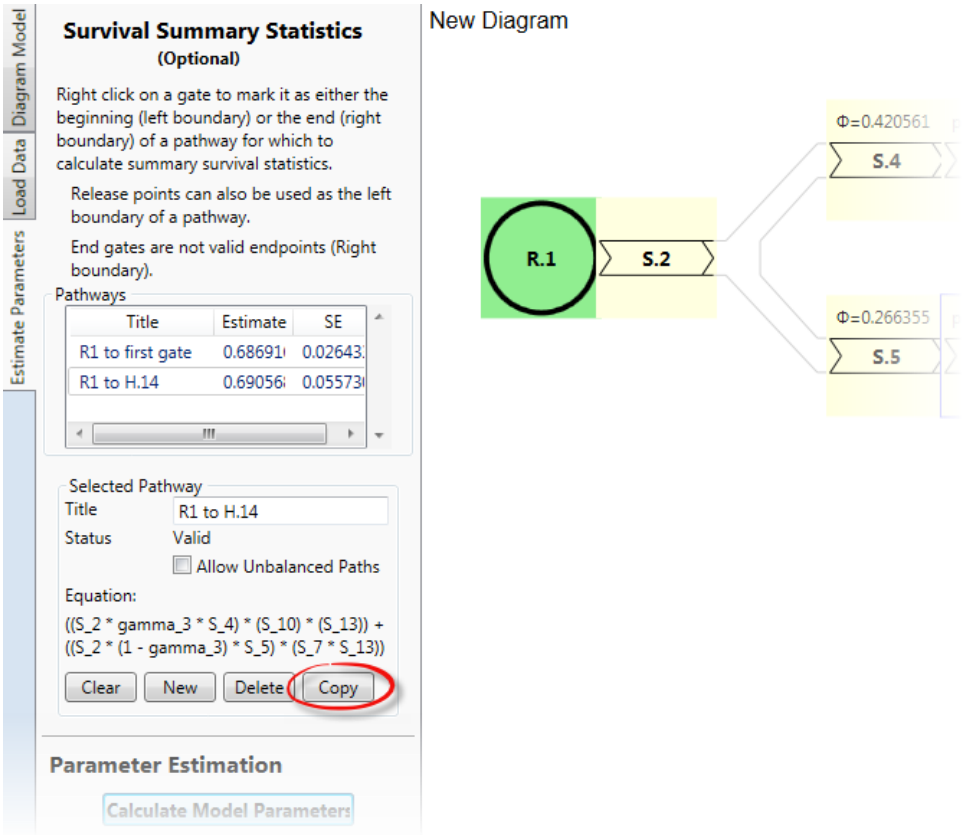


Figure 39. Estimates and standard errors for user-defined areas

5.2.1 Defining an Unbalanced Pathway

In Figure 40, notice that it is possible for an individual from R.1 to survive but not pass through the defined endpoint G.9 by following the path marked in blue. Note that the status under "Selected Pathway" is listed as "Invalid" because of this. By default, Program Branch will not estimate the survival probability in this pathway; if the user wants estimates from an unbalanced pathway, the "Allow Unbalanced Paths" check box must be checked before performing the estimation. When this is done, a warning is issued saying that this could lead to invalid survival estimates; Program Branch will estimate the joint probability of route selection and survival.

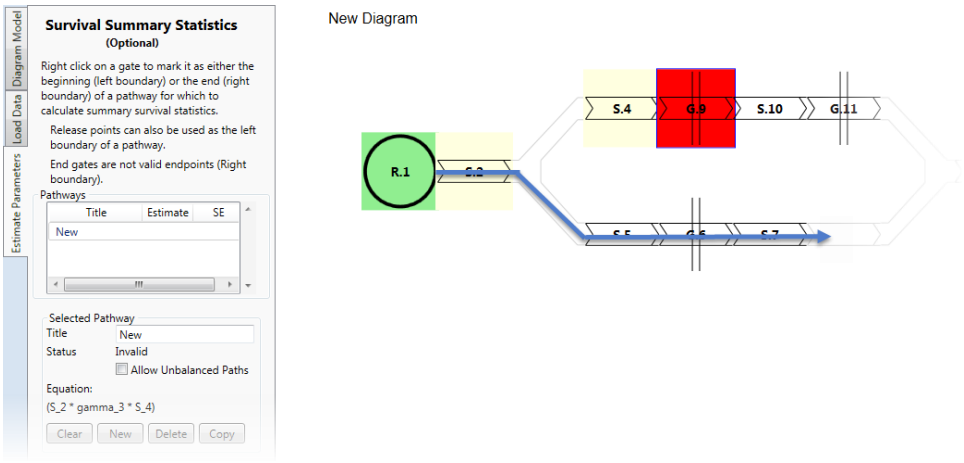


Figure 40. An "unbalanced" pathway defined