

Last update: July 2nd, 2013

The purpose of this tutorial is to outline the application of a group of two-stage Markov models that have been used to quantify recollective and nonrecollective retrieval processes (Brainerd, Aydin, & Reyna, 2012; Brainerd & Reyna, 2010; Brainerd, Reyna, & Howe, 2009; Gomes, Brainerd, & Stein, 2013). The tutorial provides a step-by-step guide on how to compute the relevant statistics to obtain parameter estimates and goodness-of-fit statistics. Because the models measure retrieval operations that can be broadly separated into recollective (direct access, D) and nonrecollective ones (reconstruction, R , and familiarity judgment, J), we also refer to them as dual-retrieval models. For specific information about the models and the theory underlying them, please see Brainerd et al. (2009).

Section I: Dual Retrieval Models

In addition to the original dual-retrieval model with 11 free parameters (Brainerd & Reyna, 2010; Brainerd et al., 2009), we have developed four other reduced models with 6 free parameters. Although the five models measure both recollective and nonrecollective processes, the original model can only fit data from experiments with at least 4 trials (T_1, T_2, T_3, T_4), whereas the reduced models can fit data from experiments with only 3 fixed trials (T_1, T_2, T_3). However, the procedures described here apply to all five models. Below is a short description of each model, separated by study (S) and test (T) design, and their respective starting vector (\mathbf{W}) and transition matrix (\mathbf{M}):

- For a $S_1T_1T_2$, S_2T_3 , S_3T_4 (noncanonical) design: The original dual-retrieval model with 11 free parameters can be applied to such data. The model has 4 direct access parameters (D_1 , D_2 , D_{3C} , D_{3E}), 2 reconstruction parameters (R_1 , R_2), 4 familiarity judgment parameters (J_1 , J_2 , J_{3C} , J_{3EC}), and 1 forgetting parameter (f).

Original model:

$$\mathbf{W} = [L(1), PE(1), PC(1), U(1)] = [D_1, (1-D_1)R_1(1-J_1), (1-D_1)R_1J_1, (1-D_1)(1-R_1)]$$

$$\mathbf{M}_1 = \begin{array}{c} \begin{array}{c} L(1) \\ PE(1) \\ PC(1) \\ U(1) \end{array} \end{array} \begin{array}{c} \begin{array}{c} L(2) \\ PE(2) \\ PC(2) \\ U(2) \end{array} \end{array} \begin{array}{|c|c|c|c|} \hline 1 & 0 & 0 & 0 \\ \hline 0 & (1-f)(1-J_{3E}) & (1-f)J_{3E} & f \\ \hline 0 & (1-f)(1-J_{3C}) & (1-f)J_{3C} & f \\ \hline 0 & 0 & 0 & 1 \\ \hline \end{array}$$

$$\mathbf{M}_2 = \begin{array}{c} \begin{array}{c} L(n+1) \\ PE(n+1) \\ PC(n+1) \\ U(n+1) \end{array} \end{array} \begin{array}{c} \begin{array}{c} L(n+2) \\ PE(n+2) \\ PC(n+2) \\ U(n+2) \end{array} \end{array} \begin{array}{|c|c|c|c|} \hline 1 & 0 & 0 & 0 \\ \hline D_{3E} & (1-D_{3E})(1-J_{3E}) & (1-D_{3E})J_{3E} & 0 \\ \hline D_{3C} & (1-D_{3C})(1-J_{3C}) & (1-D_{3C})J_{3C} & 0 \\ \hline D_2 & (1-D_2)R_2(1-J_2) & (1-D_2)R_2J_2 & (1-D_2)(1-R_2) \\ \hline \end{array}$$

Last update: July 2nd, 2013

- For a S_1T_1, S_2T_2, S_3T_3 (canonical) design: Four dual-retrieval models with 6 free parameters can be applied to such data. We refer to such models as the learning on errors model ("error model" for short), learning on successes (success model), and learning on both errors and successes model (both model). Such reduced models have 2 direct access parameters (D_1, D_2), 2 reconstruction parameters (R_1, R_2), and 2 familiarity parameters (J_1, J_2). The difference between them lies on the State L late learning entry parameter D_2 , namely whether entry in State L after trial 1 depends on prior incorrect recall (error model), prior correct recall (success model), or does not depend on prior correct or incorrect recall (both model). Finally, the fourth reduced model is an alternative version of the error model in which there is 1 reconstruction parameter (R) and 3 familiarity judgment parameters (J_1, J_2, J_3) in addition to the 2 direct access parameters.

Error model:

$$\mathbf{W} = [\underline{L}(1), \underline{PE}(1), \underline{PC}(1), \underline{U}(1)] = [D_1, (1-D_1)R_1(1-J_1), (1-D_1)R_1J_1, (1-D_1)(1-R_1)]$$

$$\mathbf{M} = \begin{matrix} & \underline{L}(n+1) & \underline{PE}(n+1) & \underline{PC}(n+1) & \underline{U}(n+1) \\ \begin{matrix} \underline{L}(n) \\ \underline{PE}(n) \\ \underline{PC}(n) \\ \underline{U}(n) \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ D_2 & (1-D_2)(1-J_2) & (1-D_2)J_2 & 0 \\ 0 & (1-J_2) & J_2 & 0 \\ D_2 & (1-D_2)R_2(1-J_2) & (1-D_2)R_2J_2 & (1-D_2)(1-R_2) \end{bmatrix} \end{matrix}$$

Alternative Error model:

$$\mathbf{W} = [\underline{L}(1), \underline{PE}(1), \underline{PC}(1), \underline{U}(1)] = [D_1, (1-D_1)R(1-J_1), (1-D_1)RJ_1, (1-D_1)(1-R)]$$

$$\mathbf{M}_1 = \begin{matrix} & \underline{L}(2) & \underline{PE}(2) & \underline{PC}(2) & \underline{U}(2) \\ \begin{matrix} \underline{L}(1) \\ \underline{PE}(1) \\ \underline{PC}(1) \\ \underline{U}(1) \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ D_2 & (1-D_2)(1-J_2) & (1-D_2)J_2 & 0 \\ 0 & (1-J_2) & J_2 & 0 \\ D_2 & (1-D_2)R(1-J_2) & (1-D_2)RJ_2 & (1-D_2)(1-R) \end{bmatrix} \end{matrix}$$

$$\mathbf{M}_2 = \begin{matrix} & \underline{L}(3) & \underline{PE}(3) & \underline{PC}(3) & \underline{U}(3) \\ \begin{matrix} \underline{L}(2) \\ \underline{PE}(2) \\ \underline{PC}(2) \\ \underline{U}(2) \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ D_2 & (1-D_2)(1-J_3) & (1-D_2)J_3 & 0 \\ 0 & (1-J_3) & J_3 & 0 \\ D_2 & (1-D_2)R(1-J_3) & (1-D_2)RJ_3 & (1-D_2)(1-R) \end{bmatrix} \end{matrix}$$

Last update: July 2nd, 2013Success model:

$$W = [L(1), P_E(1), P_C(1), U(1)] = [D_1, (1-D_1)R_1(1-J_1), (1-D_1)R_1J_1, (1-D_1)(1-R_1)]$$

	$L(n+1)$	$P_E(n+1)$	$P_C(n+1)$	$U(n+1)$
$L(n)$	1	0	0	0
$P_E(n)$	0	$(1-J_2)$	J_2	0
$P_C(n)$	D_2	$(1-D_2)(1-J_2)$	$(1-D_2)J_2$	0
$U(n)$	0	$R_2(1-J_2)$	R_2J_2	$(1-R_2)$

Both model:

$$W = [L(1), P_E(1), P_C(1), U(1)] = [D_1, (1-D_1)R_1(1-J_1), (1-D_1)R_1J_1, (1-D_1)(1-R_1)]$$

	$L(n+1)$	$P_E(n+1)$	$P_C(n+1)$	$U(n+1)$
$L(n)$	1	0	0	0
$P_E(n)$	D_2	$(1-D_2)(1-J_2)$	$(1-D_2)J_2$	0
$P_C(n)$	D_2	$(1-D_2)(1-J_2)$	$(1-D_2)J_2$	0
$U(n)$	D_2	$(1-D_2)R_2(1-J_2)$	$(1-D_2)R_2J_2$	$(1-D_2)(1-R_2)$

Section II: Example

This example used a database composed of 15 subjects that learned to recall a list of 40 words using the $S_1T_1T_2$, S_2T_3 , S_3T_4 noncanonical design mentioned in Section 1. [The database is available on the website \(Database I\)](#). The model applied to the data was the original dual retrieval model.

1. Structure of the database

The database must follow a specific format in order to compute the frequencies of correct (C) and incorrect (E) recall across trials using the data count program, namely

- The first row contains the names of variables, with the data range beginning in the second row.
- The first column contains the subject ID number.
- In the first row, each column after the first one will contain both the name of the to-be-recalled items and the trial number. Specifically, the variable names must have the following format: [name of the item][lowercase letter t][trial number]. For instance, column #2 can be called SPIDERT1, meaning the item SPIDER on trial 1. The program will look for this structure in order to compute items' history of recall across trials, so make sure that each item has the same name across trials (e.g., SPIDERT1, SPIDERT2, SPIDERT3, SPIDERT4). **NOTE 1:** The word list cannot contain different set of words on different trials. The program will not compute the frequency of C-E patterns across trials if this is the case. **NOTE 2:** Only words used in the recall test can have the above format. Thus, it is possible to have other variables (e.g., age), provided that these variables are not written in the format above.

Last update: July 2nd, 2013

- Use binary coding to assign correct recall (1) and incorrect recall (0).
- Ensure that there are no empty cells within the data range.
- The worksheet in which the database is stored must be called DATA.

Here is a screen shot of the sample database:

	A	B	C	D	E	F
1	ID	EXAMEt1	COBRAt1	BARAt1	ZANGAt1	CH0
2	1	0	0	0	0	
3	2	0	1	0	0	
4	3	0	0	1	1	
5	4	0	0	0	0	
6	5	0	0	0	1	
7	6	0	0	0	0	
8	7	0	0	0	1	
9	8	1	1	0	1	
10	9	0	0	0	0	
11	10	0	1	0	0	
12	11	1	0	0	0	
13	12	0	0	0	0	
14	13	1	1	0	0	
15	14	1	0	0	0	
16	15	1	0	0	1	
17						
18						
19						

2. Computing frequencies of C-E patterns across trials

The data count program (see Programs and Model Files) is an Excel VBA macro that reads the database and computes C-E frequencies. Because an item is either recalled (1 = C) or not (0 = E) on each test, there are a total of 2^n patterns of C-E responses across trials, in which n is the total number of trials. In our example, the database is composed of 4 separate recall tests and, therefore, there are $2^4 = 16$ patterns of C-E responses (CCCC, CCCE, ..., EEEE). For instance, if a given word is recalled successfully across all 4 trials, that would be counted as CCCC = 1; if a given word is recalled successfully on the first 3 trials but unsuccessfully on the last trial, that would be counted as CCCE = 1. This is done for all items and subjects in the DATA worksheet of the data count Excel file, and the final frequencies are the ones aggregated across both.

1. Open the data count program and the database.
2. Transfer the responses from the database to the DATA worksheet in the data count program.
3. Enable the use of macros in Excel (for security reasons, the default setting is not to allow their use). Go to Tools > Options > Security > Macro Security and then reduce the security level to Low.

Last update: July 2nd, 2013

4. Make sure that the structure is the same as noted earlier in Section 2.1.
5. Press Ctrl+U (or the button) to run the macro.
6. The results will be pasted on the RESULTS worksheet of the data count program.

For our database, the frequencies were the following:

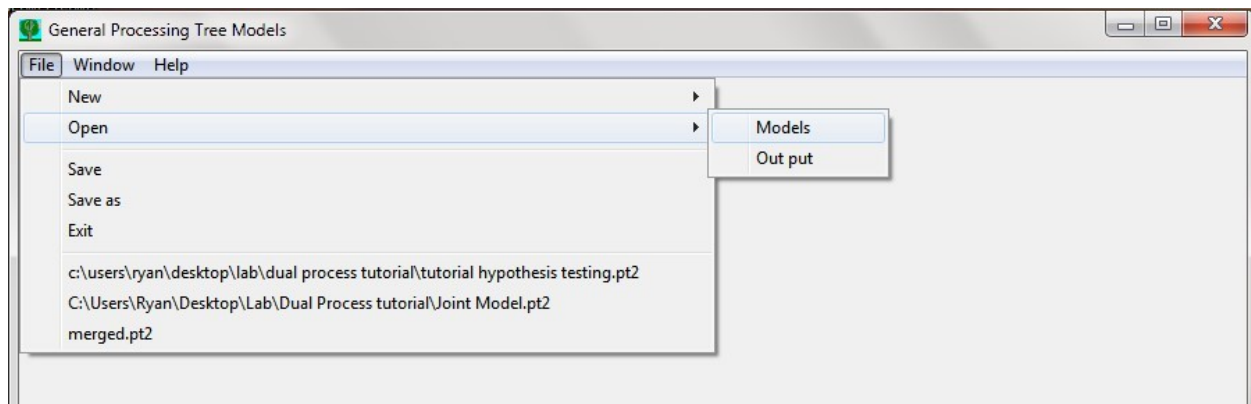
CCCC: 167 CCCE: 7 CCEC: 9 CCEE: 2 CECC: 12 CECE: 4 CEEC: 5 CEEE: 3 ECCC: 24 ECCE: 2
ECEC: 1 ECEE: 1 EECC: 123 EECE: 12 EEEC: 103 EEEE: 125

In our database, there were 167 cases in which an item was recalled on all tests (CCCC), and 125 in which an item was never recalled (EEEE). Of course, the sum of all frequencies should be equal to the product between (sample size) and (number of items), which is $15 \times 40 = 600$ in this case, and the proportion of items recalled on the i th test is simply the sum of C-E patters in which there is a C on the i th test divided by the sum of all frequencies. For instance, the proportion of items recalled on the 3rd test was $= (CCCC + CCCE + CECC + CECE + ECCC + ECCE + EECC + EECE) / [(sample\ size) \times (number\ of\ items)] = .59$.

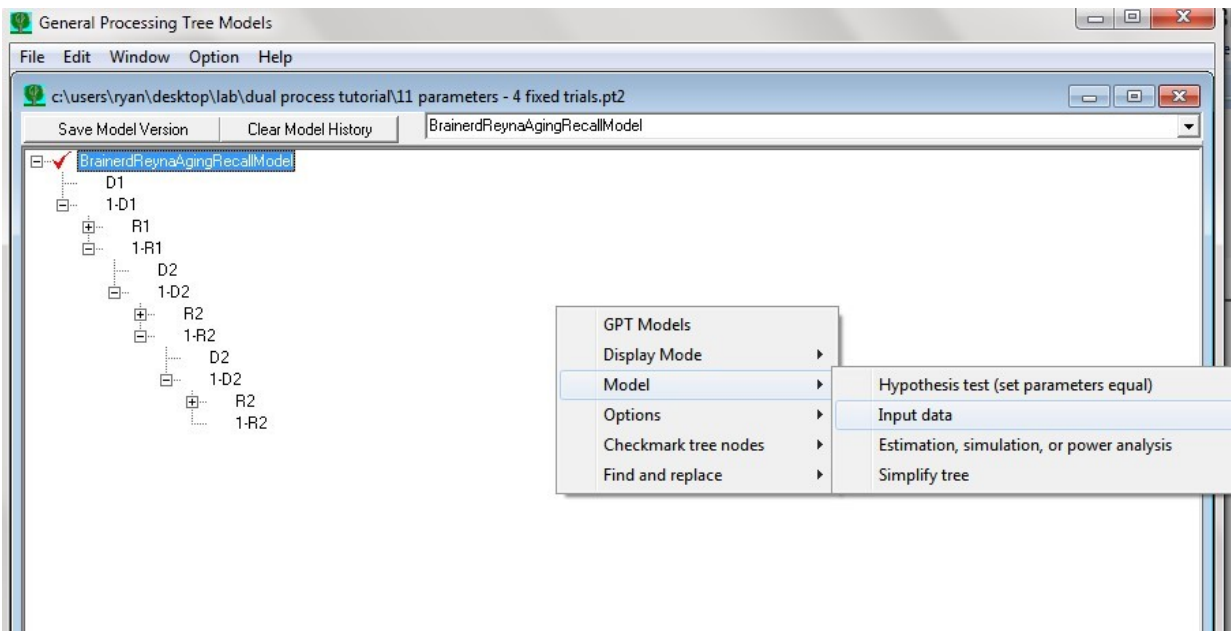
3. Obtaining parameter estimates and goodness-of-fit statistics using GPT

Once the frequencies of C-E patterns across trials have been computed, you are ready to enter them into the GPT.

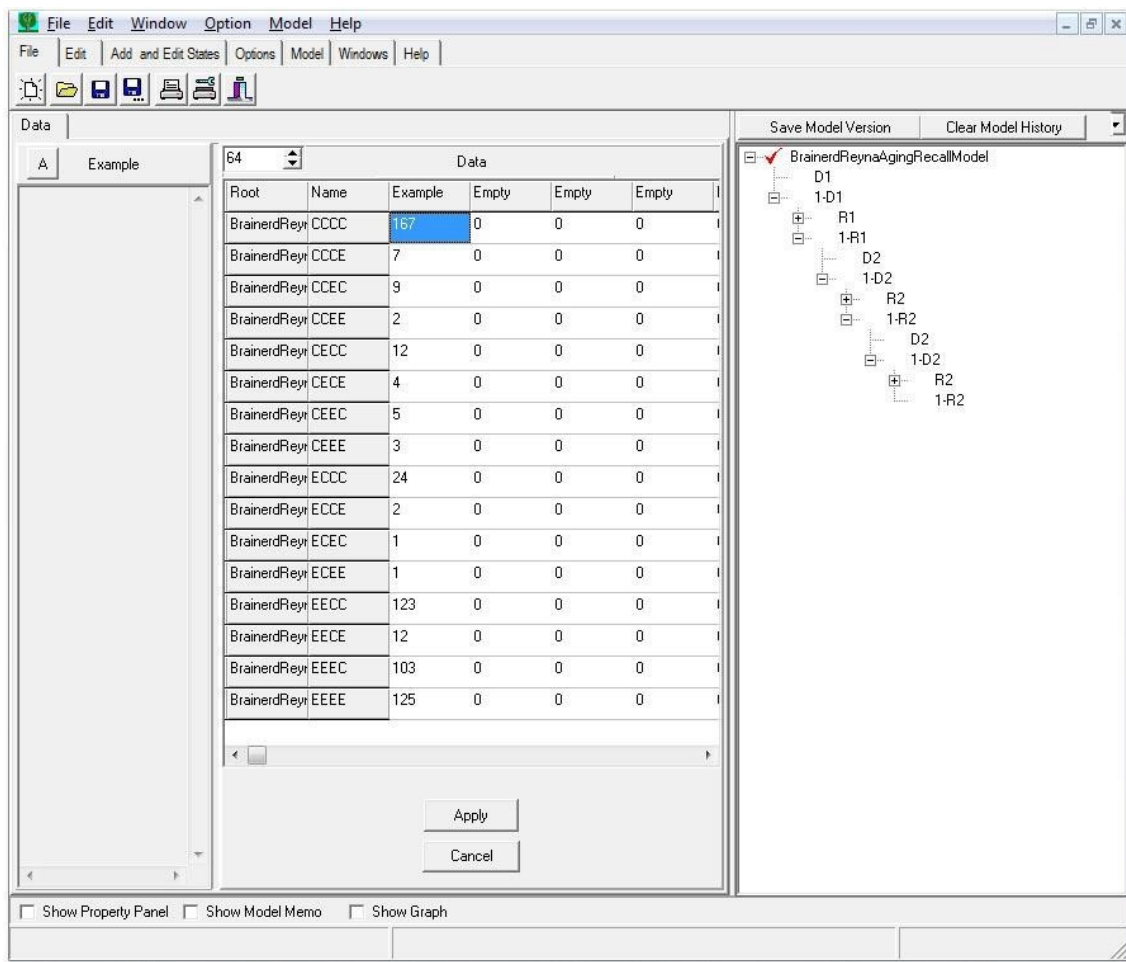
- a) Open the GPT
- b) Go to File > Open > Models and load the original dual-retrieval model .pt2 file (see Programs and Model Files)



- c) Once you have loaded the model, you can start entering the C-E frequencies: Right click on the model window > Model > Input data

Last update: July 2nd, 2013

d) Enter the frequencies in their respective cell in the first column, as follows:



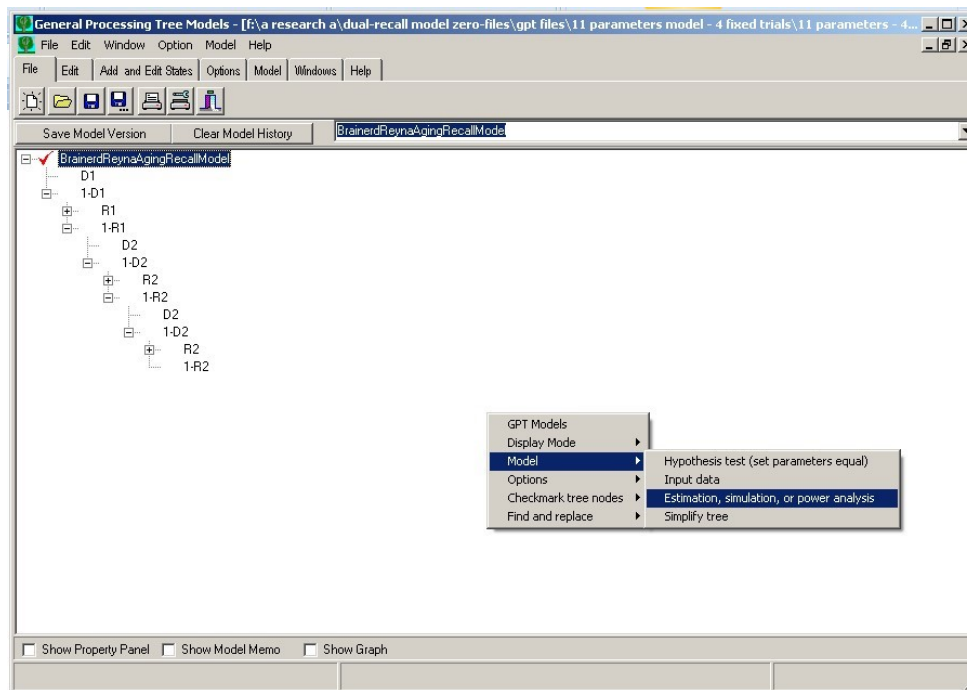
e) Add a name to the column by clicking on it with the right button, then rename.

f) Click apply and save the file

Last update: July 2nd, 2013

You are now able to compute parameter estimates and goodness-of-fit statistics.

- a) Return to the model window
- b) Right click > Model > Estimation, simulation or power analysis



- c) In the select models column, select BrainerdReynaAgingRecallModel, which is the default (unconstrained) model
- d) In the select data for estimation, select the first row or whatever row that contains the frequencies you have entered previously
- e) Then click Run
- f) Once the program stops running, the results will be presented in the **Output Tab** and will also be pasted on the **Output Window** (see Hu & Phillips, 1999, for more information). The maximum likelihood estimates of each parameter, their respective standard deviation (SD), and the goodness-of-fit statistic (χ^2) pasted on the output window should look like this:

Parameter estimates

```

D1 = 0.16064, SD = 0.04945, (initial value: 0.42594)
D2 = 0.11067, SD = 0.10306, (initial value: 0.34826)
D3C = 0.27659, SD = 0.15190, (initial value: 0.84037)
D3E = 0.00000, SD = 0.00000, (initial value: 0.24153)
F = 0.04525, SD = 0.05494, (initial value: 0.09628)
J1 = 0.74471, SD = 0.05506, (initial value: 0.04693)

```


Last update: July 2nd, 2013

```
J2 = 0.64327, SD = 0.09721, (initial value: 0.06274)
J3C = 0.82413, SD = 0.05476, (initial value: 0.80138)
J3E = 0.75965, SD = 0.09422, (initial value: 0.20643)
R1 = 0.30027, SD = 0.04601, (initial value: 0.01425)
R2 = 0.43920, SD = 0.10311, (initial value: 0.06042)
```

Test will be a Chi-square with 4 degrees of freedom Chi-square[4]=
3.75306.

Last update: July 2nd, 2013

Section III: Hypothesis Testing

Comparisons between parameter estimates can be performed via likelihood ratio tests (LRT) (e.g., Brainerd, Reyna, & Howe, 2009). LRT can be broadly separated into two types: (a) within- and (b) between-condition tests. The former type tests hypotheses regarding parameter differences within the same experimental condition, which can be either differences between parameters (e.g., $H_0: R_1 = R_2$ vs. $H_1: R_1 \neq R_2$) or differences between parameters and a constant (e.g., $H_0: f = 0$ vs. $H_1: f \neq 0$). The latter type tests hypotheses that have to do with localizing treatment effects, that is, whether parameter estimates differ across any set of $k > 1$ conditions (e.g., $H_0: J_{1, \text{Condition 1}} = J_{1, \text{Condition 2}}$ vs. $H_1: J_{1, \text{Condition 1}} \neq J_{1, \text{Condition 2}}$). Between-condition tests can be further divided into experiment-wise tests, condition-wise tests, and parameter-wise tests. Because Type I error increases as a function of the number of tests, it is often advised to perform condition-wise and parameter-wise tests only if there is global statistical evidence of treatment effects, as indicated by an experimentwise test, or adjust the family-wise alpha level using a correction such as Bonferroni ($\alpha_{\text{adjusted}} = \alpha / n$, in which n is the total number of comparisons) or Sidak ($\alpha_{\text{adjusted}} = 1 - (1 - \alpha)^{1/n}$).

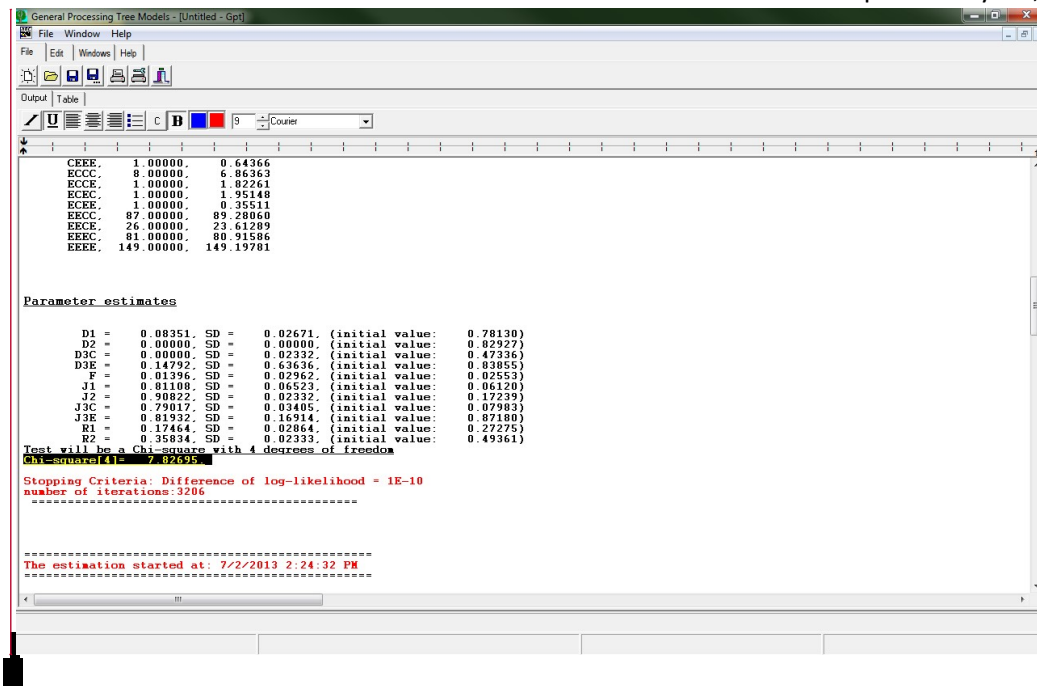
In GPT, the test statistic in all cases is computed by subtracting the fit statistic of the unconstrained model (i.e., a model in which all parameters of the model are free) from the constrained model (a model in which one imposes the restrictions of interest, e.g., $f = 0$). The resulting difference is a statistic asymptotically distributed as χ^2 , with df equal to the difference in free parameters between the two models. The critical χ^2 value of the test can therefore be calculated and compared against the observed χ^2 .

Next, we describe how to perform LRT using the GPT program. **We will be using the database provided on the lab website (Database II).** Note that the database contains three worksheets, each containing data from a different experimental condition.

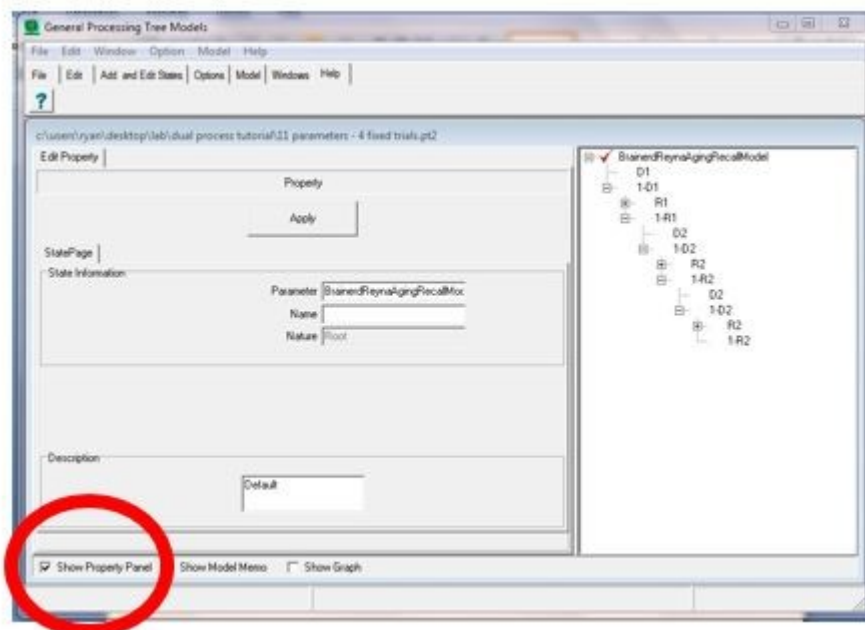
A. Within-Conditions Test: Parameter = Constant

In this section, we will perform a within-condition hypothesis test on the data from Condition 1. The null hypothesis being tested is that $D_1 = 0$.

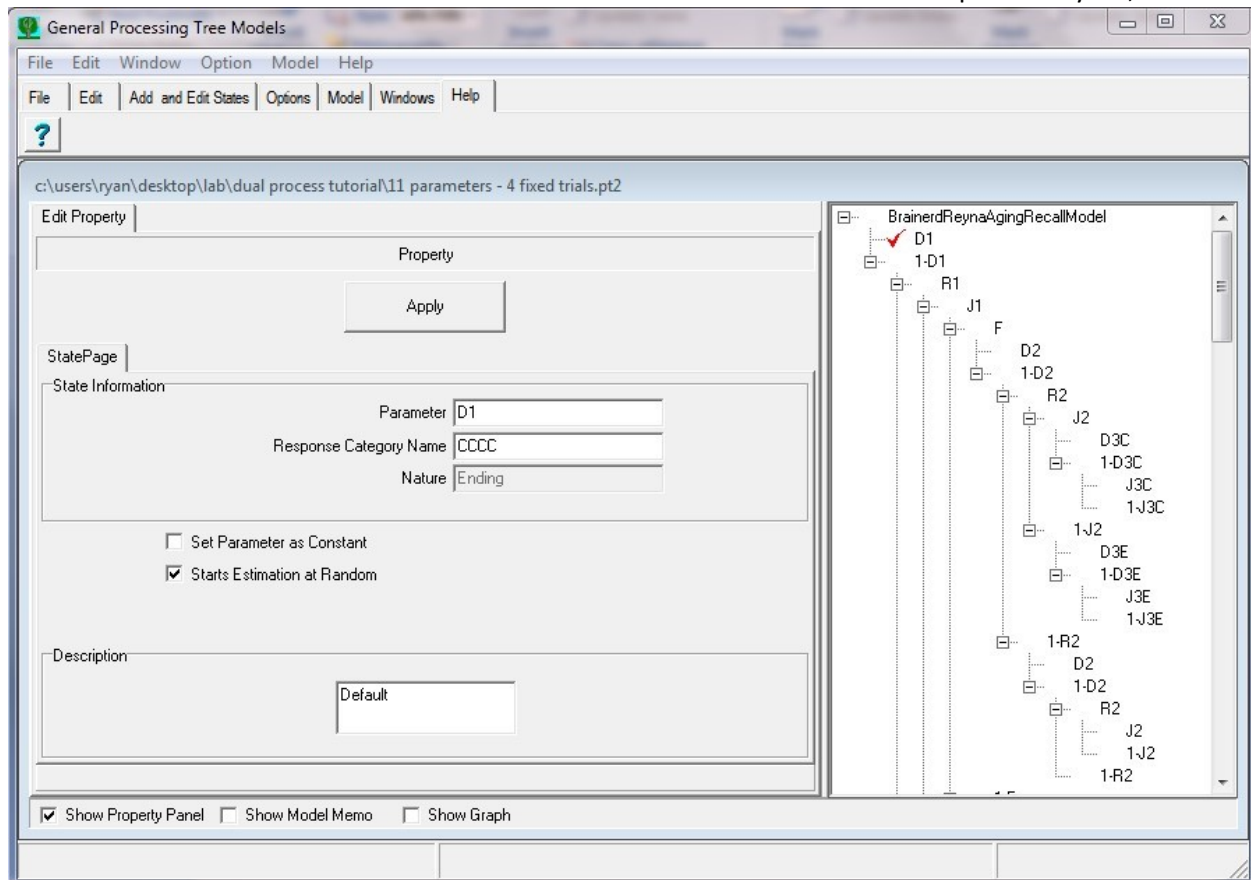
1. Compute the goodness of fit statistic for the unconstrained model by following the instructions in Section 2. It was equal to 7.83.

Last update: July 2nd, 2013

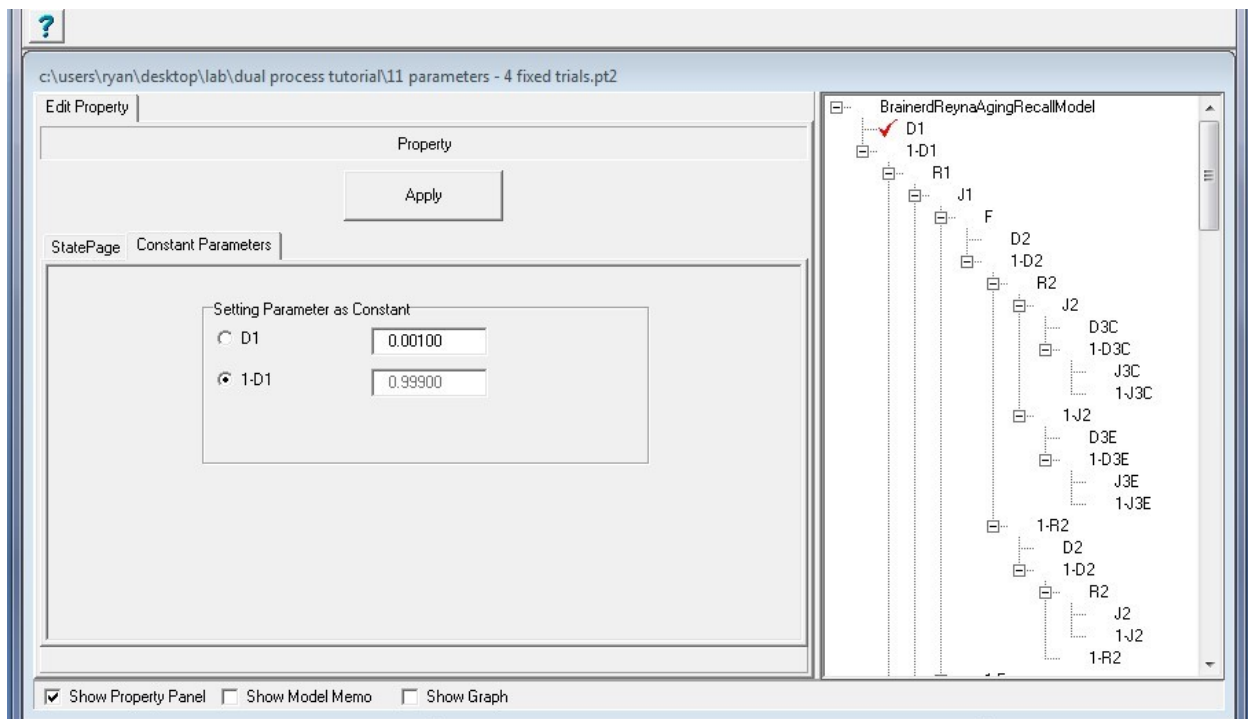
- Next, we will compute the goodness of fit statistic for the constrained model.
- Click on the show property panel in the model window, and the following screen should appear:



- Select the parameter of interest. In this case, click on D_1 in the model window.

Last update: July 2nd, 2013

5. Click “Set Parameter as Constant”, and specify the value. In our case, $D1 = 0.001$:

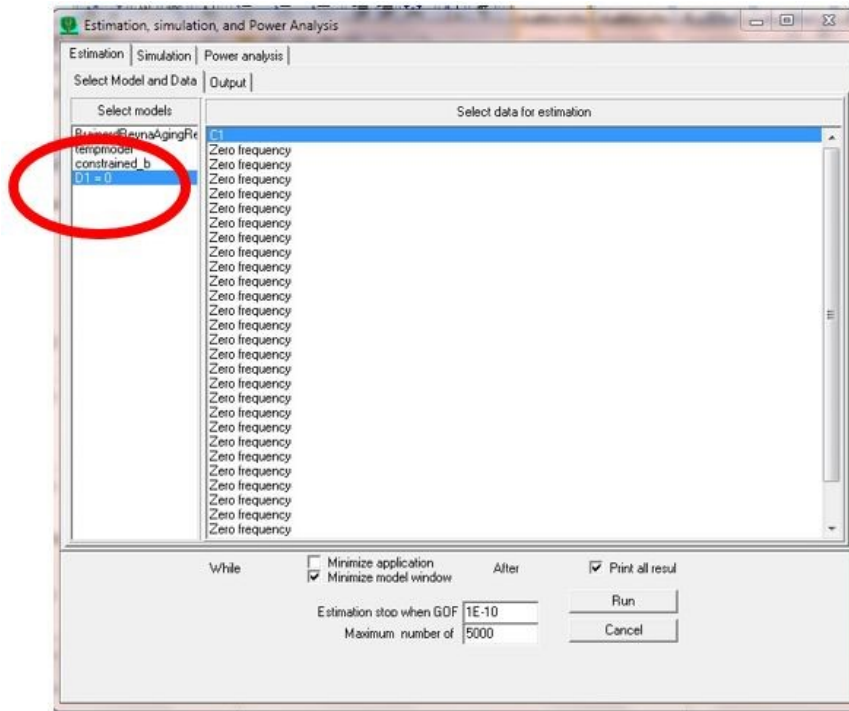


6. Click Apply

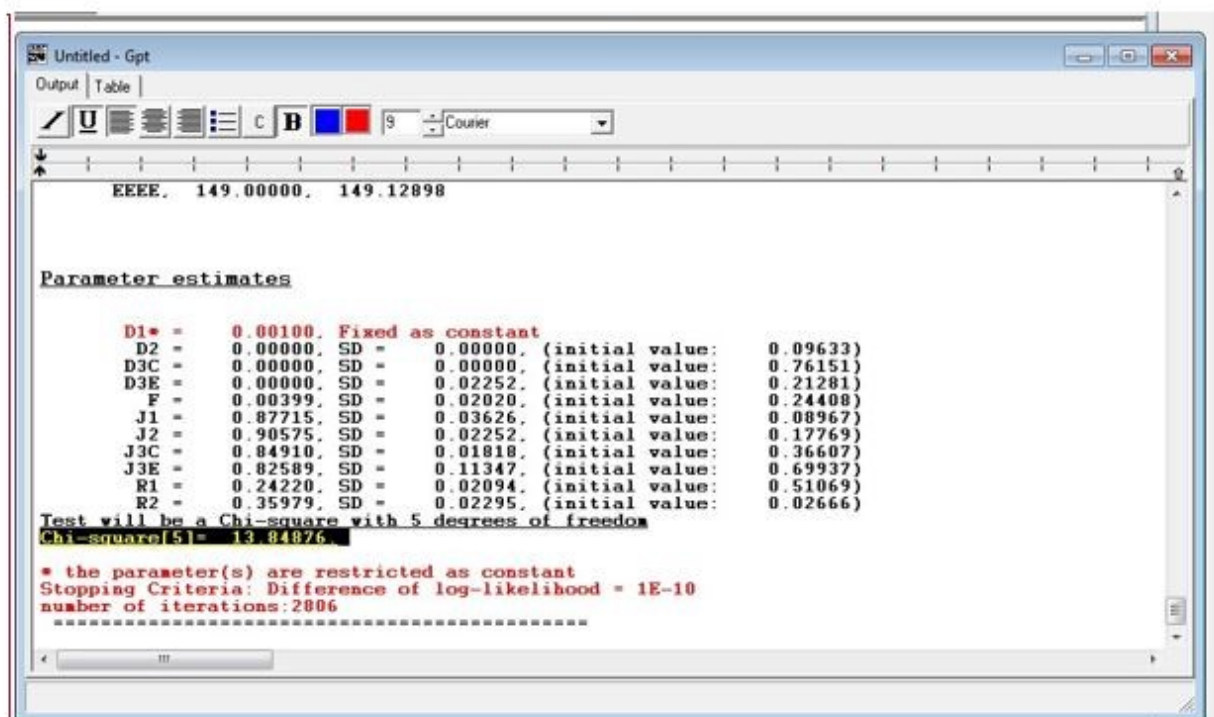
7. Deselect the “show property panel” and return to the main window.

Last update: July 2nd, 2013

8. Click Save the Model Version, and name the model. In this case, we named the constrained model, $D1 = 0$.
9. Select the constrained model and compute the goodness of fit.



10. The goodness of fit statistic was equal to 13.85:



Last update: July 2nd, 2013

11. One can now perform the hypothesis test by comparing the two goodness of fit statistics:

Constrained = 13.85**Unconstrained = 7.83**

12. Calculate the difference between the two statistics to obtain the test statistic:

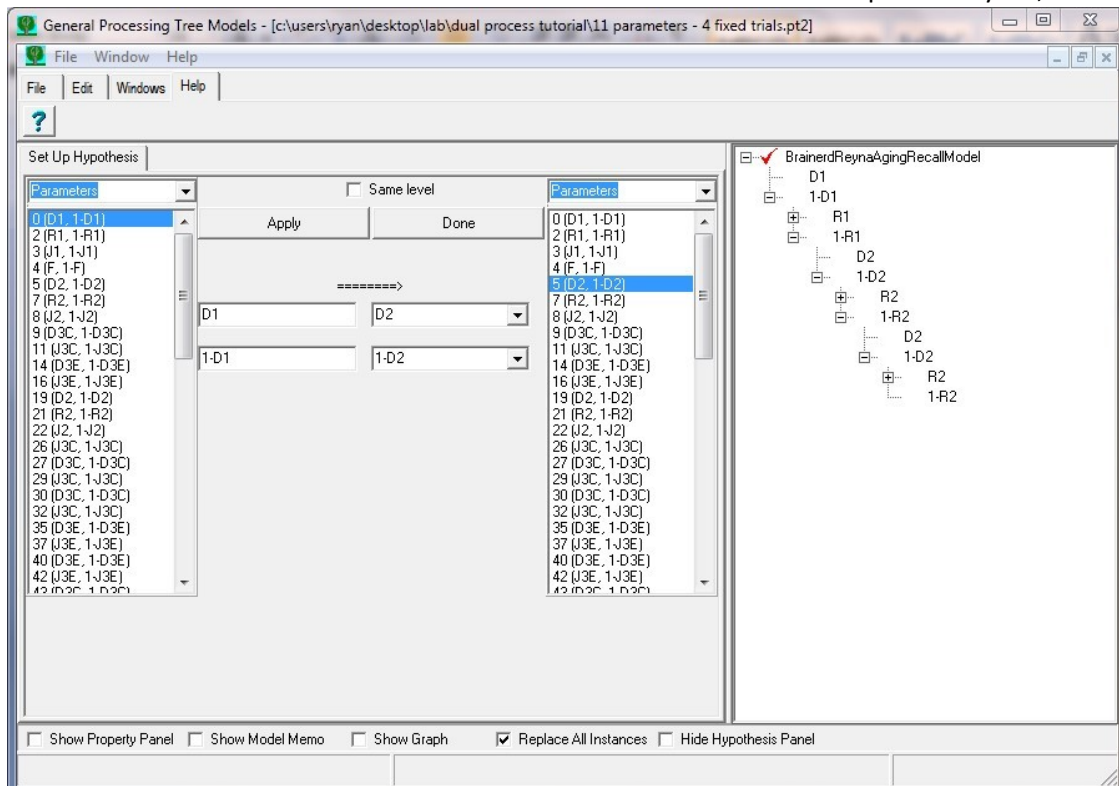
Test statistic = $13.85 - 7.83 = 6.02$

13. Compare the test statistic to the test's critical value. For instance, at a significance criterion of .05, the critical value of the test statistic is $\chi^2(1) = 3.84$. Because the test statistic is higher than the critical value, we reject the null hypothesis of the test.

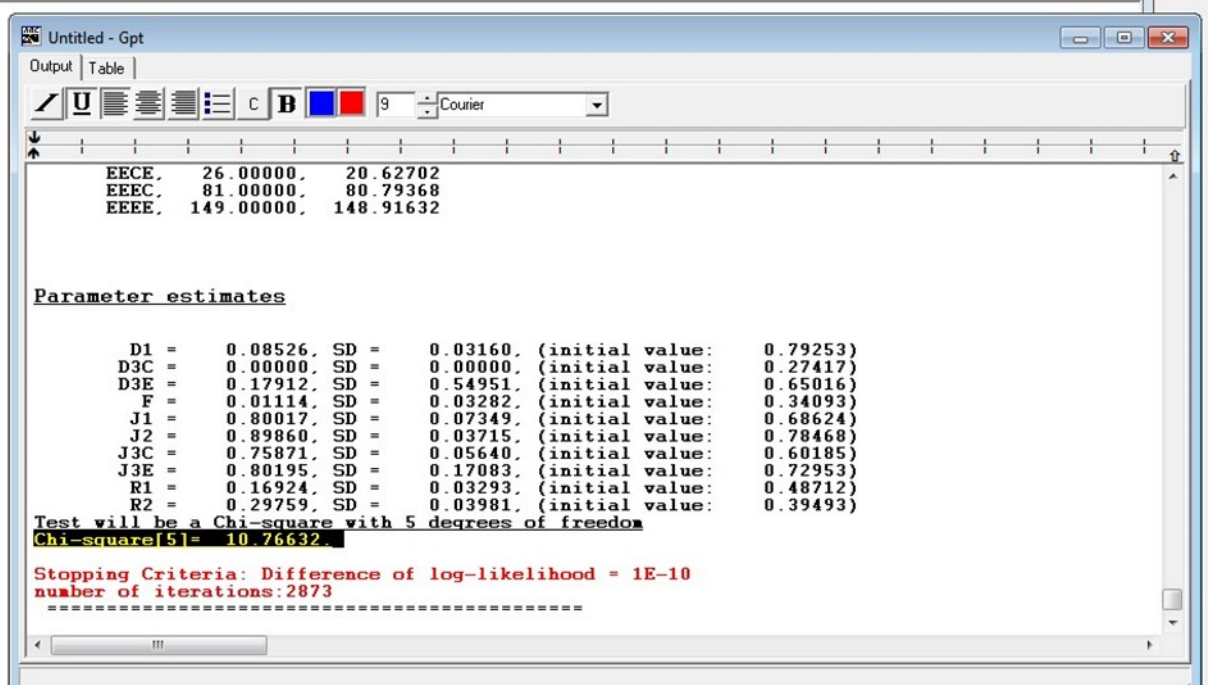
B. Within-Condition Test: Parameter_i = Parameter_j, for $i \neq j$

In this section, we will perform a within-condition hypothesis test on the data from Condition 1. The null hypothesis being tested is that $D_1 = D_2$.

1. Compute the goodness of fit statistic for the unconstrained model by following the instructions in Section 2. It was equal 7.83.
2. Next, we will compute the goodness of fit statistic for the constrained model.
3. Right Click ---> Model ---> Hypothesis Test
4. Deselect "Same Level"
5. Select D1 in the left hand column and then select D2 in the right hand column.

Last update: July 2nd, 2013

6. Click Apply and Done
7. Click "Save Model Version", and name the constrained model appropriately. Here, we named it D1 = D2.
8. Compute the goodness of fit statistic for the constrained model.



Last update: July 2nd, 2013

9. One can now perform the hypothesis test by comparing the two goodness of fit statistics.

Constrained= 7.83

Unconstrained: 10.77

10. Calculate the difference between the two chi-square statistics to obtain the test statistic.

Test statistic: $10.77 - 7.83 = 2.94$

11. Compare the test statistic to the test's critical value. For instance, at a significance criterion of .05, the critical value of the test statistic is $\chi^2(1) = 3.84$. Because the test statistic is lower than the critical value, we do not reject the null hypothesis of the test.

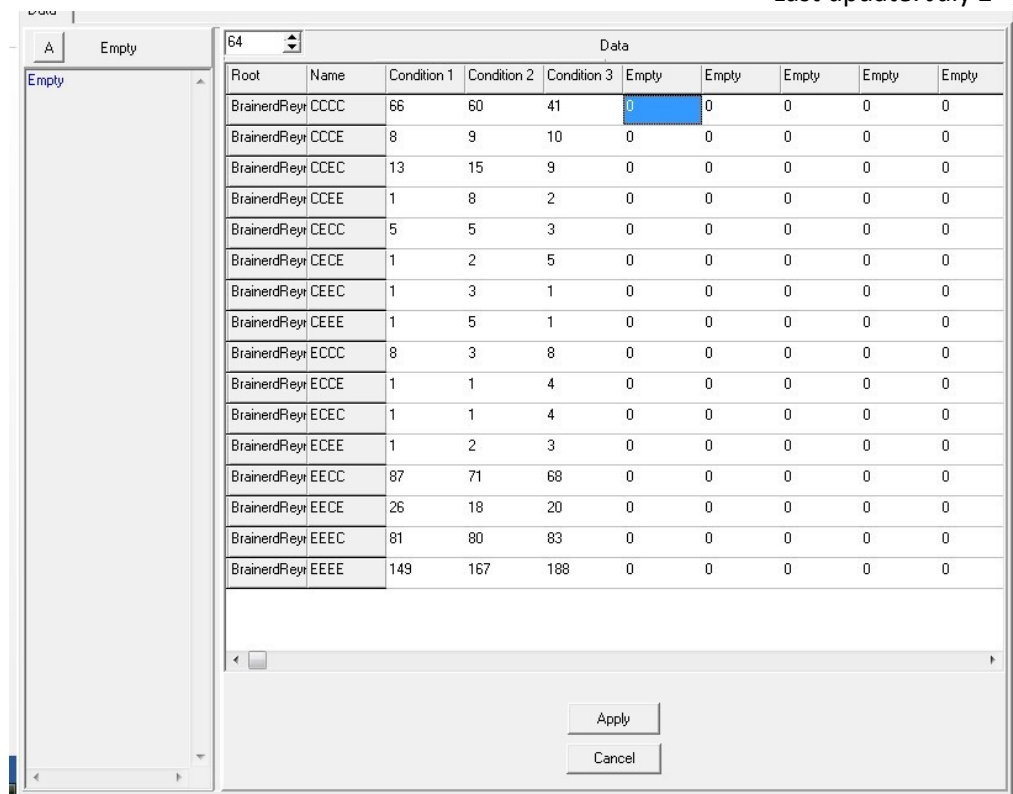
C. Between-Condition Tests: Experiment-wise test

Between-condition tests answer questions regarding treatment effects (e.g., was direct access higher for short lists relative to long lists?). Running between-condition tests consist of three steps: (a) running an experiment-wise test in which the null hypothesis holds that all parameters are *the same across all experimental conditions*; (b) running condition-wise tests if the null hypothesis of the experiment-wise test was rejected, in which the null hypotheses hold that all parameters are *the same between two conditions*; and (c) running parameter-wise tests if the null hypothesis of the relevant condition-wise test was rejected, in which the null hypotheses hold that *a parameter of the model is equal between two conditions* (e.g., $H_0: J_{1, \text{Condition 1}} = J_{1, \text{Condition 2}}$).

This section will begin with the instructions for an experiment-wise test. In our example, there are three experimental conditions, the experiment-wise test and condition-wise test are the same.

The unconstrained model in between-conditions tests is a joint conditions model. Follow the instructions below to create the joint model.

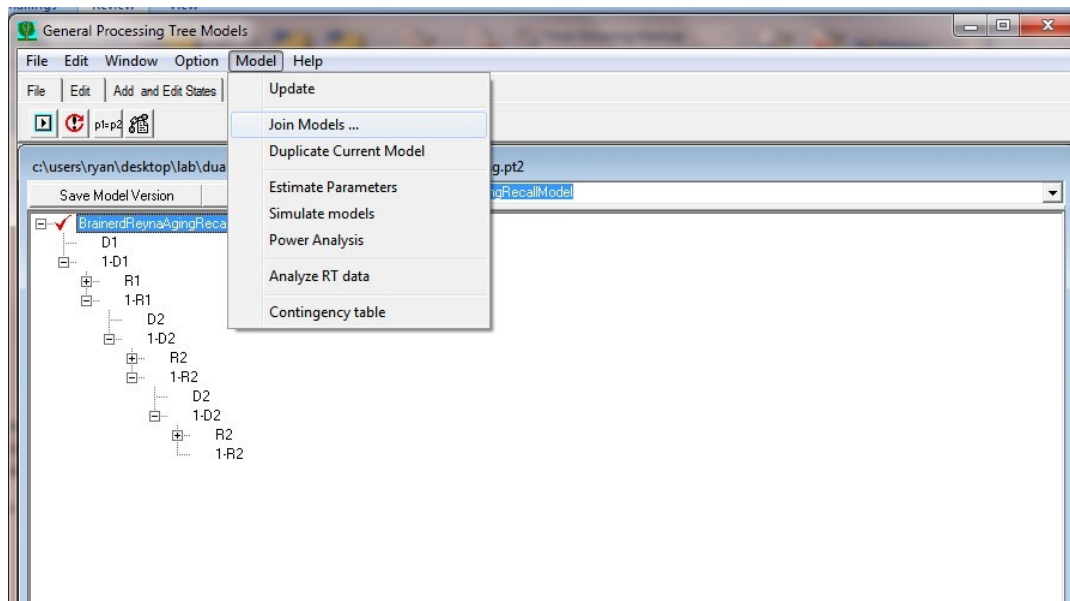
1. Enter the error-success frequencies for both conditions.

Last update: July 2nd, 2013


Root	Name	Condition 1	Condition 2	Condition 3	Empty	Empty	Empty	Empty	Empty
BrainerdReyn	CCCC	66	60	41	0	0	0	0	0
BrainerdReyn	CCCE	8	9	10	0	0	0	0	0
BrainerdReyn	CCEC	13	15	9	0	0	0	0	0
BrainerdReyn	CCEE	1	8	2	0	0	0	0	0
BrainerdReyn	CECC	5	5	3	0	0	0	0	0
BrainerdReyn	CECE	1	2	5	0	0	0	0	0
BrainerdReyn	CEEC	1	3	1	0	0	0	0	0
BrainerdReyn	CEEE	1	5	1	0	0	0	0	0
BrainerdReyn	ECCC	8	3	8	0	0	0	0	0
BrainerdReyn	ECCE	1	1	4	0	0	0	0	0
BrainerdReyn	ECCE	1	1	4	0	0	0	0	0
BrainerdReyn	ECEE	1	2	3	0	0	0	0	0
BrainerdReyn	EECC	87	71	68	0	0	0	0	0
BrainerdReyn	EECE	26	18	20	0	0	0	0	0
BrainerdReyn	EEEC	81	80	83	0	0	0	0	0
BrainerdReyn	EEEE	149	167	188	0	0	0	0	0

2. Now create a joint conditions model as follows:

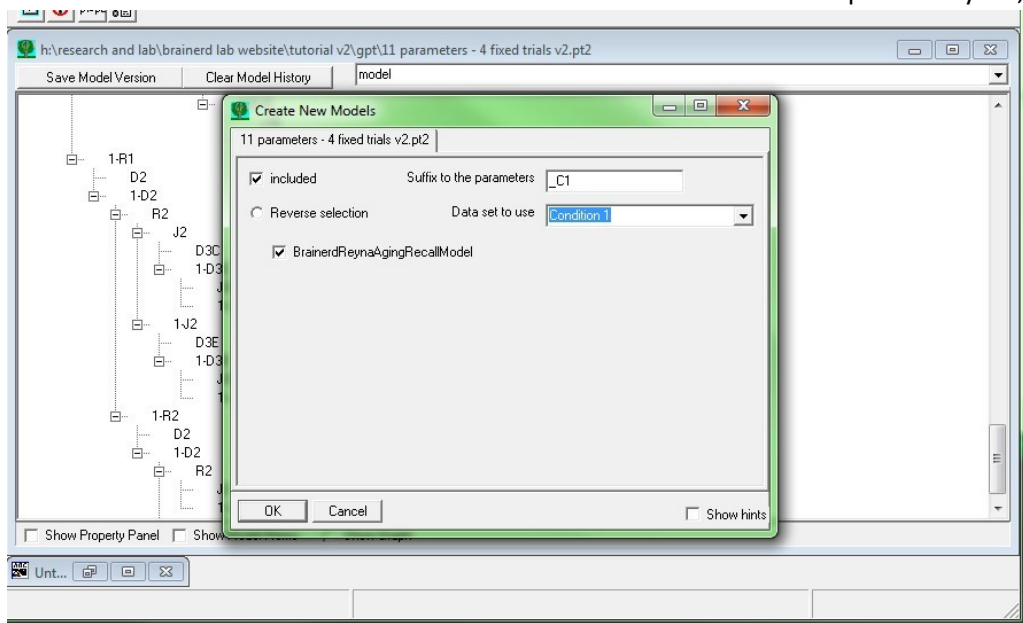
a. Click Model ---> Join Models



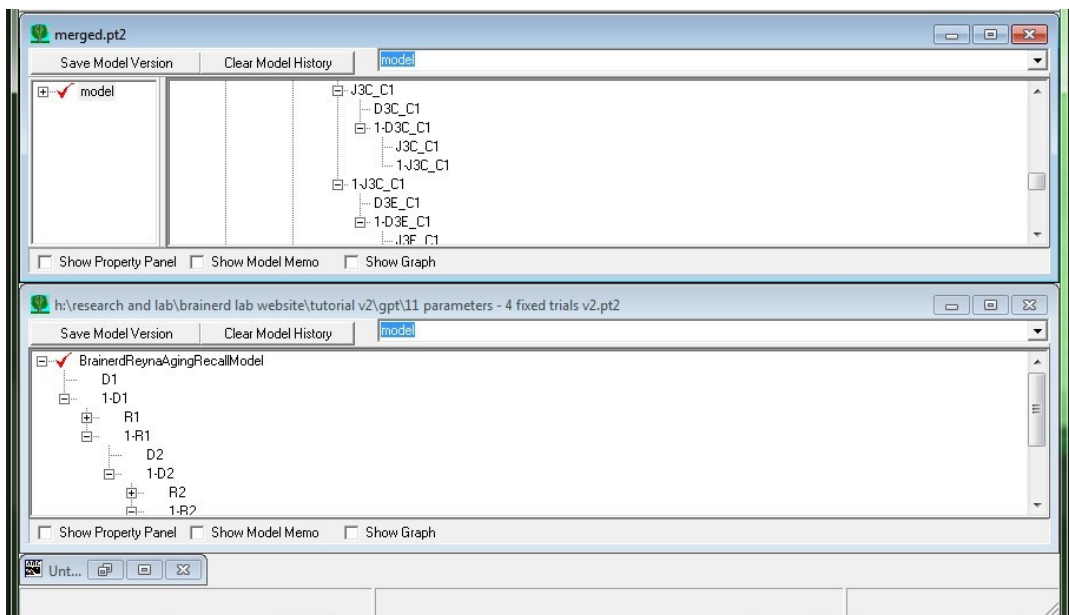
b. Under “suffix to the parameters”, enter “C1”, to denote data from Condition 1.

c. Under “Data set to use”, select “Condition 1”

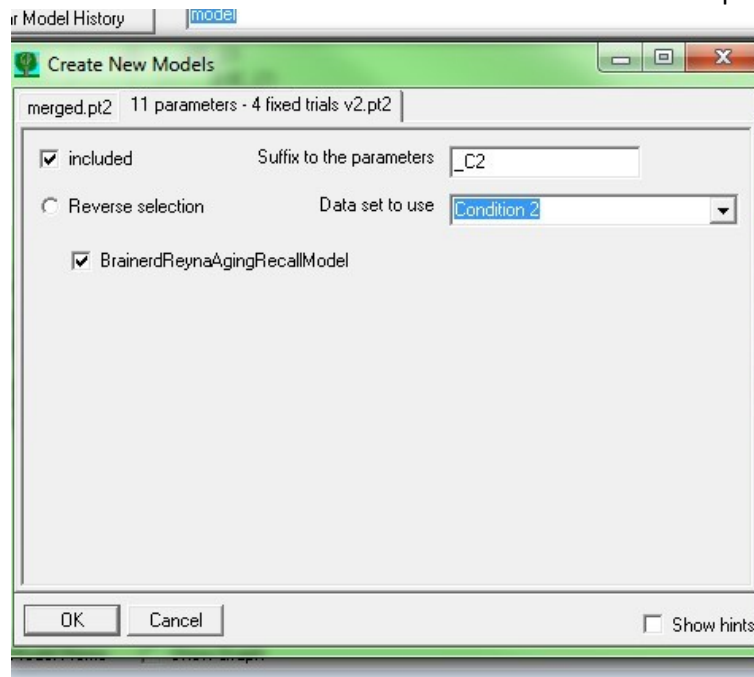
d. Click “Ok”

Last update: July 2nd, 2013

- e. You will now see two models in the main screen. One will have the name, “merge.pt2”. This is the model you just created.

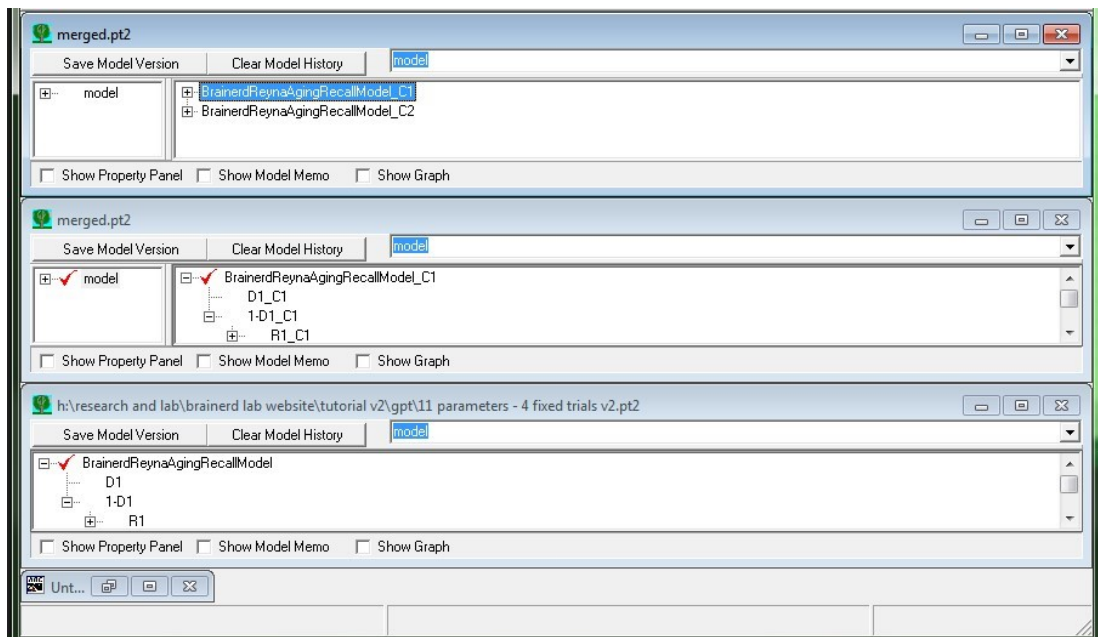


- f. Now, we have to add the dataset from condition 2. Click on the model you just created (merged.pt2)
- g. Model ---> Join Model
- h. Enter “C2” for Suffix for Parameters, and select “Condition 2” from “data set to use”

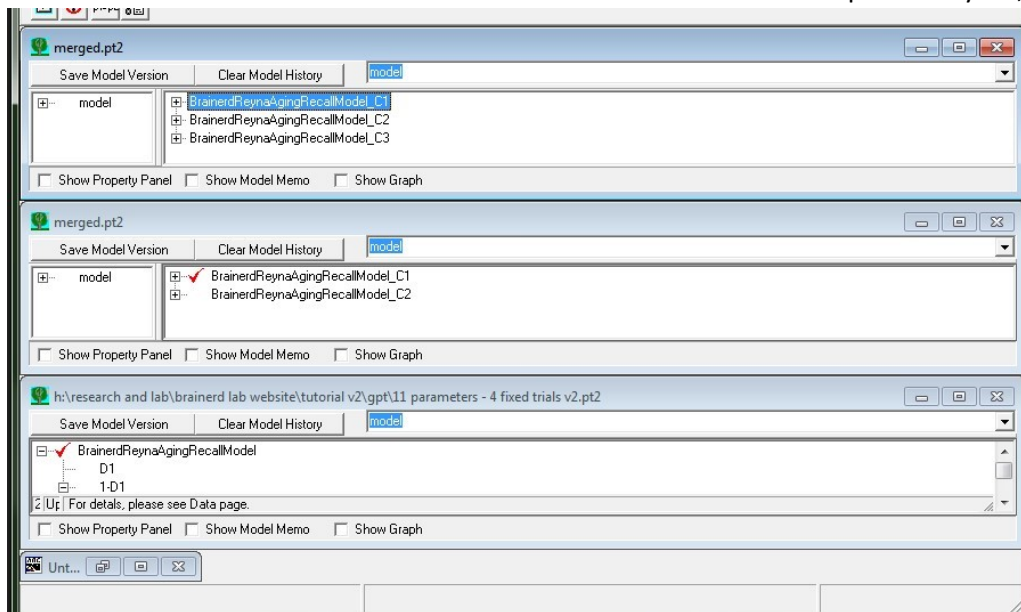
Last update: July 2nd, 2013

i. Click “Ok”

3. The new window will show all 3 models: the original model, and the two models you just created. The model that has both “Condition 1” and “Condition 2” is the joint model that we want. Select the joint model.

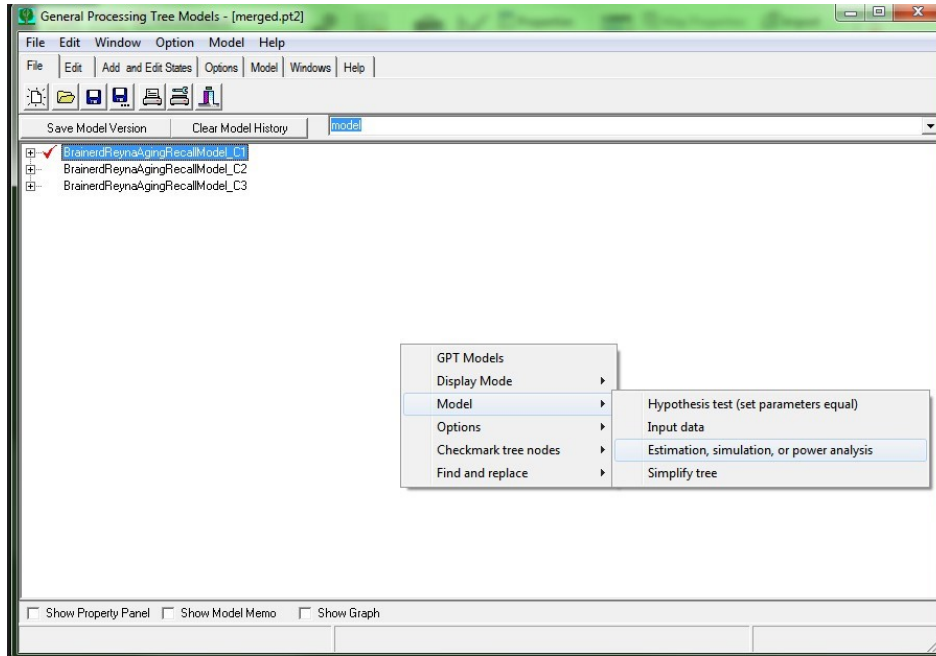


4. To include the data from Condition 3, first close the window of all models but (a) the original model and (b) the last joint model you created, then repeat the procedure described above

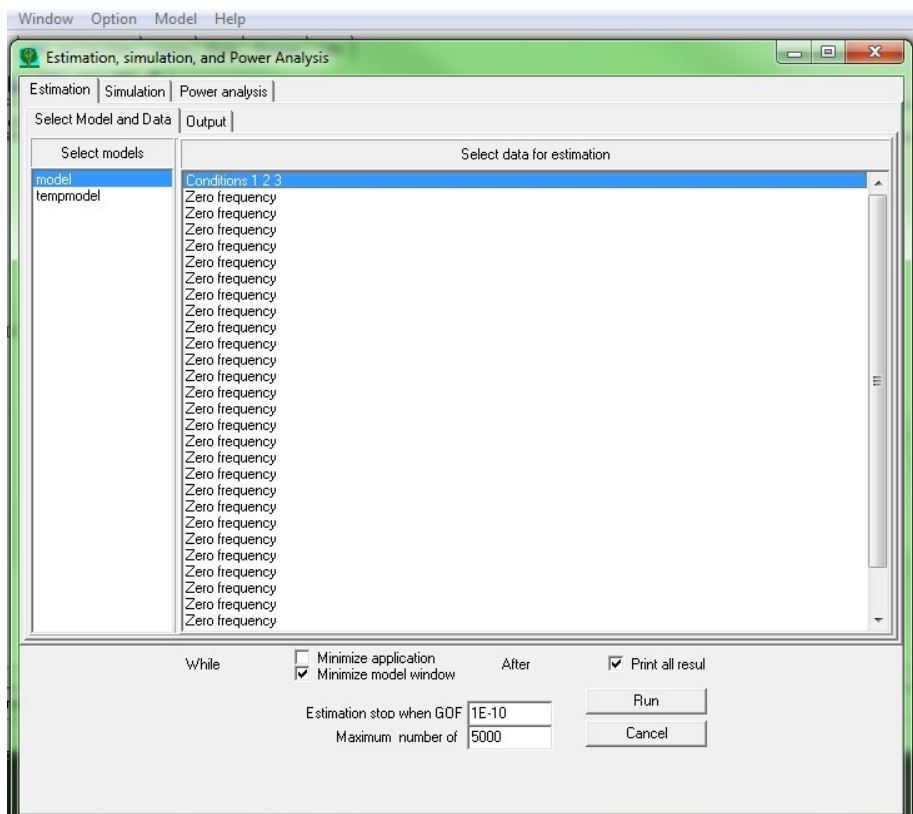
Last update: July 2nd, 2013

5. Select the model window of the Joint Model Unconstrained
6. Delete all but the first data column of the joint model and rename the first column
 - a. Go to Input Data
 - b. Delete all columns but the first one
 - c. Select the first column and rename it to reflect that the first column has the data from all conditions (e.g., Conditions 1 2 3)
 - d. Click Apply
7. Compute the goodness of fit for the unconstrained joint model you just created.
 - a. Go to the Estimation, Simulation, and Power analysis window

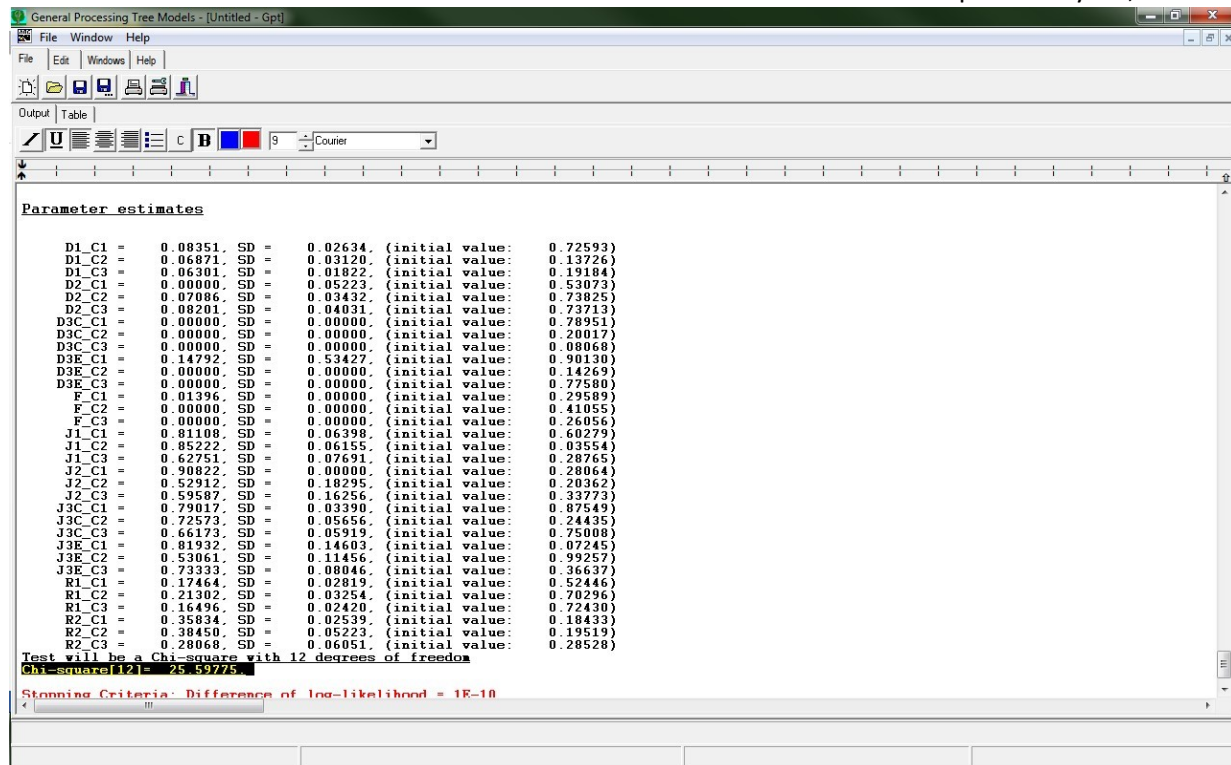
Last update: July 2nd, 2013



- b. Select the model called Joint Model Unconstrained and the joint dataset (always the first one)



- c. Click on Run

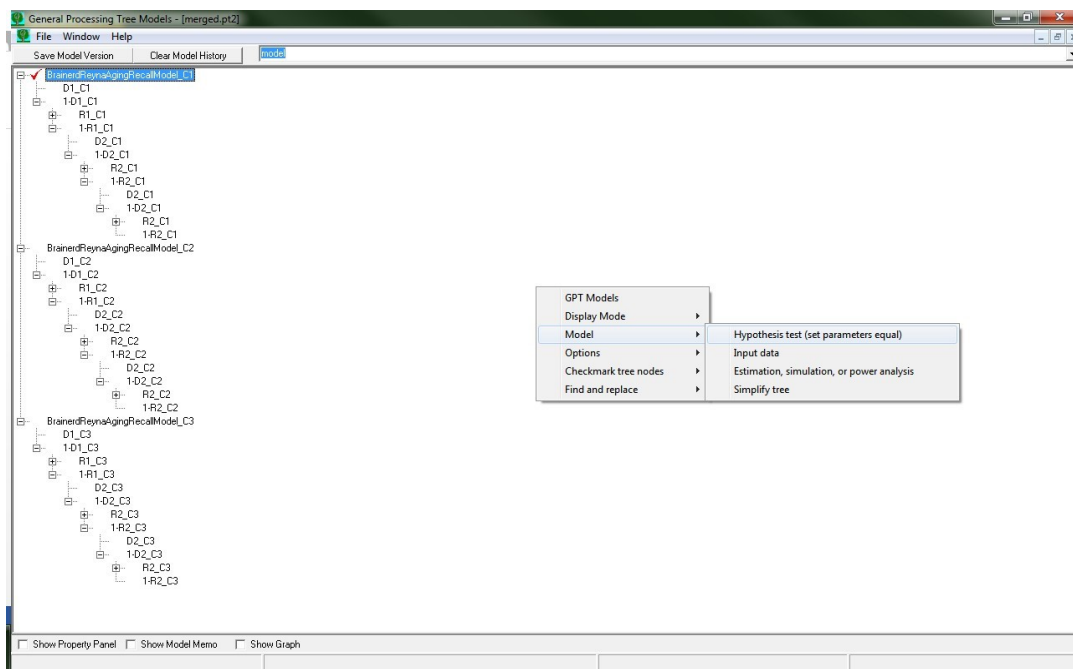
Last update: July 2nd, 2013

8. Note the goodness of fit of the unconstrained joint model.

Chi-square[12] = 25.59775.

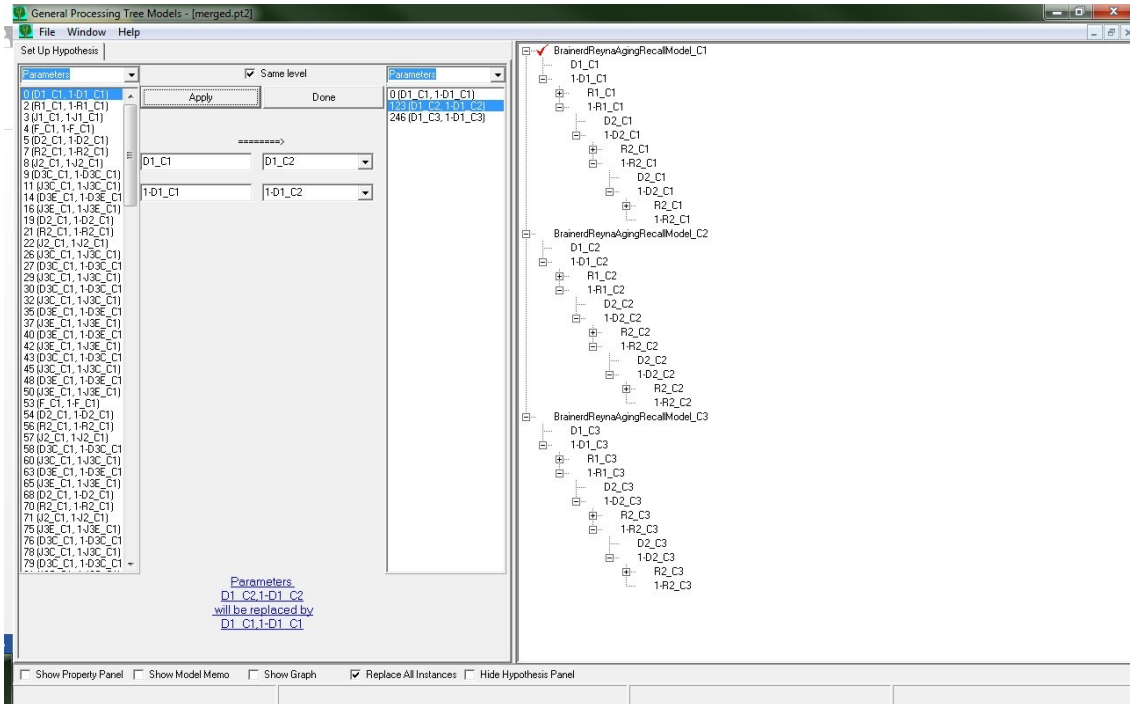
9. We are now ready to create the constrained joint model for the experiment-wise test.

a. Go to the Hypothesis Test window

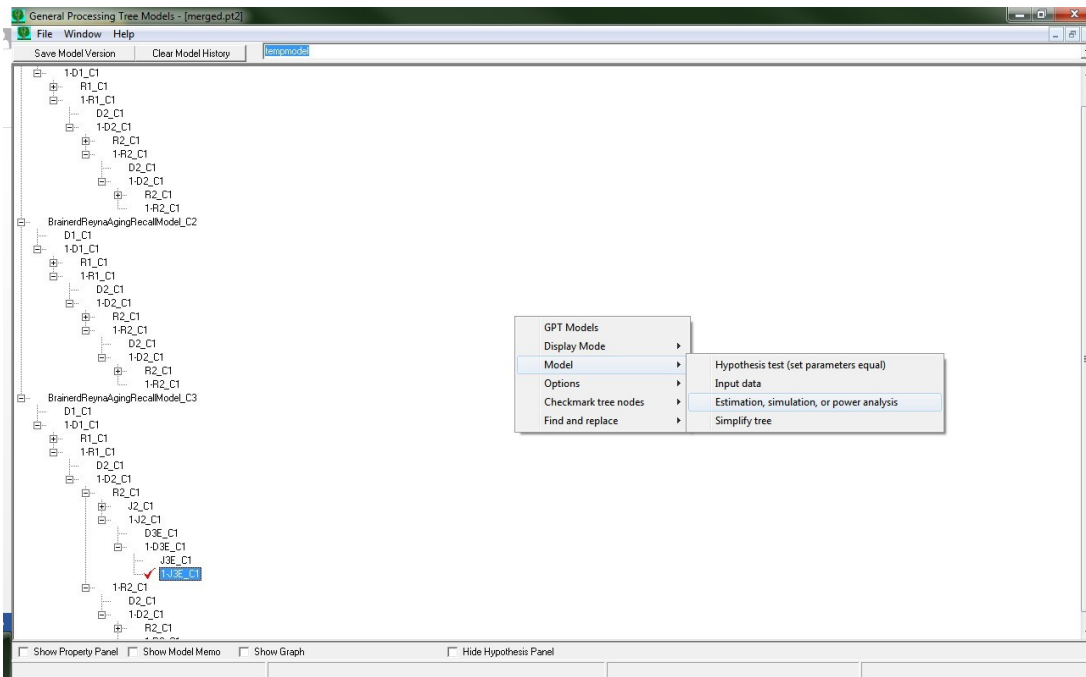


10. Set all the parameters equal to each other among all three conditions using the Apply button.
 D1C1 = D1C2 , ..., R2C2 = R2C3 (set all parameters equal among the three conditions)

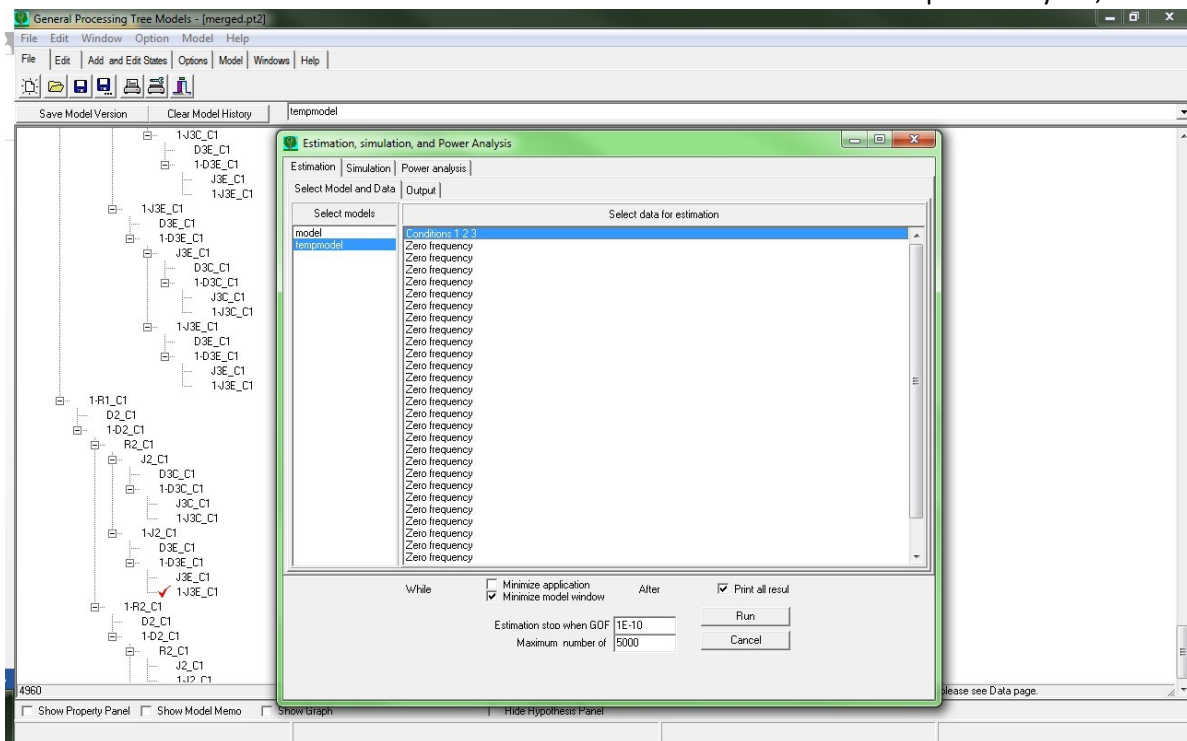
Last update: July 2nd, 2013



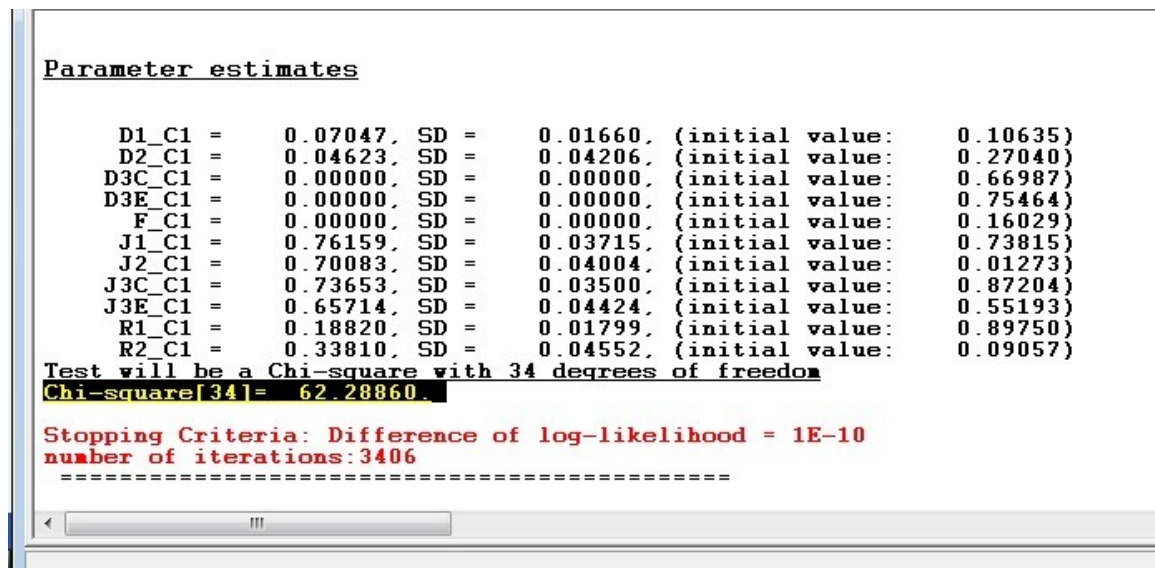
11. Click Done when you finish
12. The model you just created will be called “tempmodel”
13. Obtain the parameter estimates for the constrained model.
 - a. Go to the Estimation, Simulation, and Power analysis window



- b. Select the model called tempmodel and the joint dataset (always the first one)

Last update: July 2nd, 2013

c. Click on Run



14. One can now perform the hypothesis test by comparing the two goodness of fit statistics.

Constrained: Chi-square[34]= 62.28860
 Unconstrained: Chi-square[12]= 25.59775

15. Calculate the difference between the two chi-square statistics to obtain the test statistic.

Test statistic: 62.29 – 25.60 = 36.69

16. Compare the test statistic to the critical value. For instance, at a significance criterion of .05, the critical value of the test statistic is $\chi^2(22) = 33.92$. Because the test statistic is higher than the

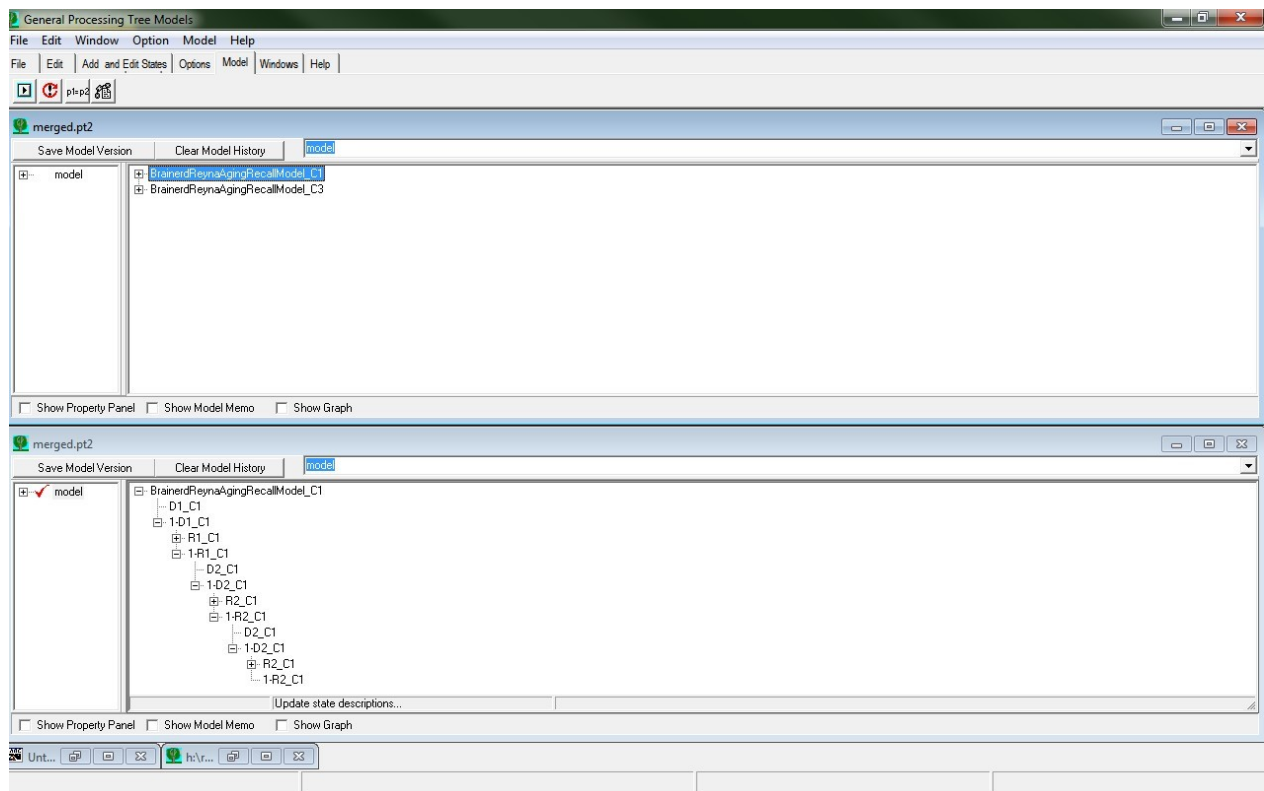
Last update: July 2nd, 2013

critical value, we reject the null hypothesis of no difference among the three experimental conditions.

D. Between-Condition Tests: Condition-wise test

There are three conditions in our example. In this section, we show how to test the null hypothesis of no difference between the parameters of any two conditions in an experiment (e.g., Parameters_{Condition 1} = Parameters_{Condition 2}). The following example illustrates a condition-wise test between Conditions 1 and 3.

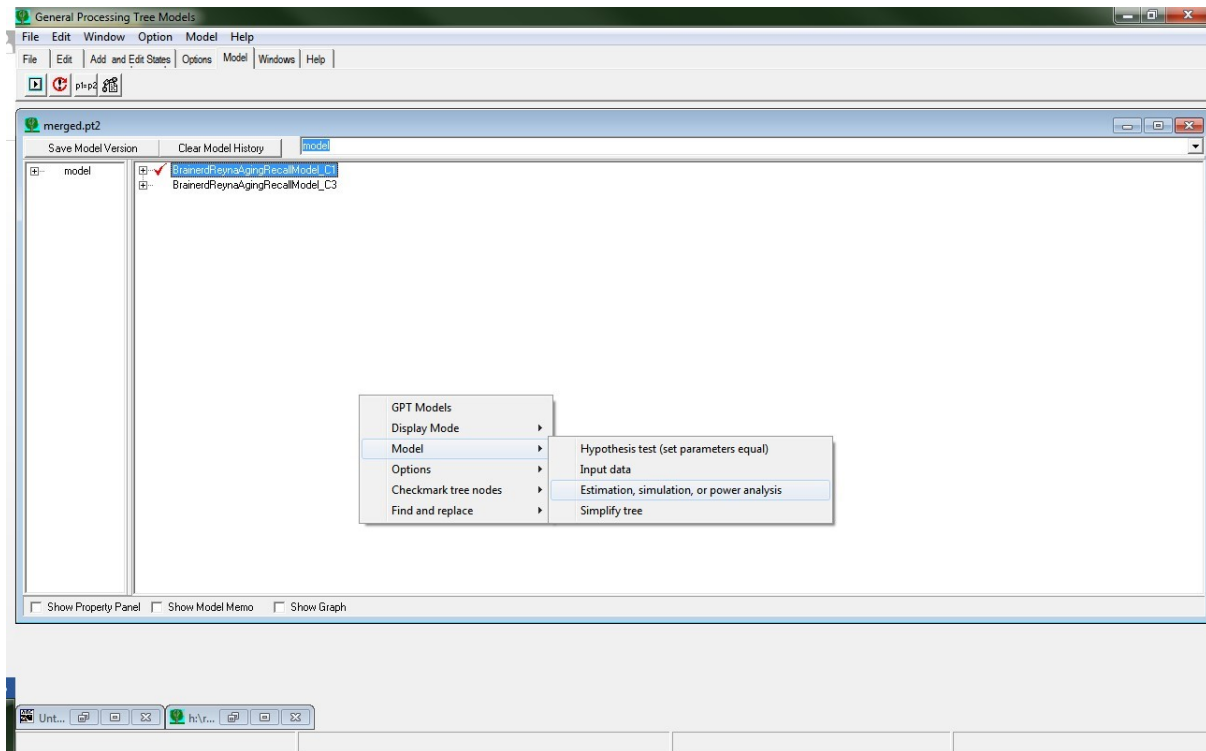
1. Create a joint model of Conditions 1 and 3 as described in the previous section.



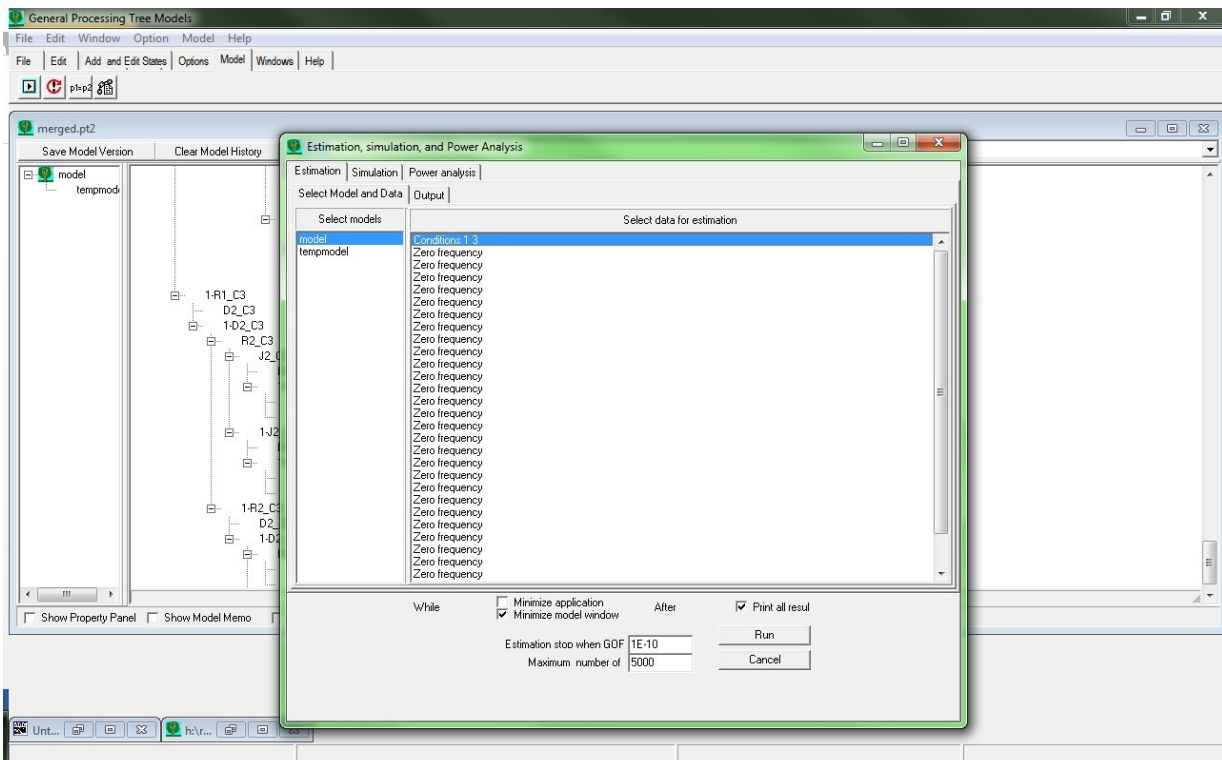
2. Select the model window of the joint model
3. Delete all but the first data column of the joint model and rename the first column a. Go to Input Data
 - a. Delete all columns but the first one
 - b. Select the first column and rename it to reflect that the first column has the data from all conditions (e.g., Conditions 1 3)

Last update: July 2nd, 2013

- d. Click Apply
4. Compute the goodness of fit for the unconstrained joint model you just created.
 - a. Go to the Estimation, Simulation, and Power analysis window

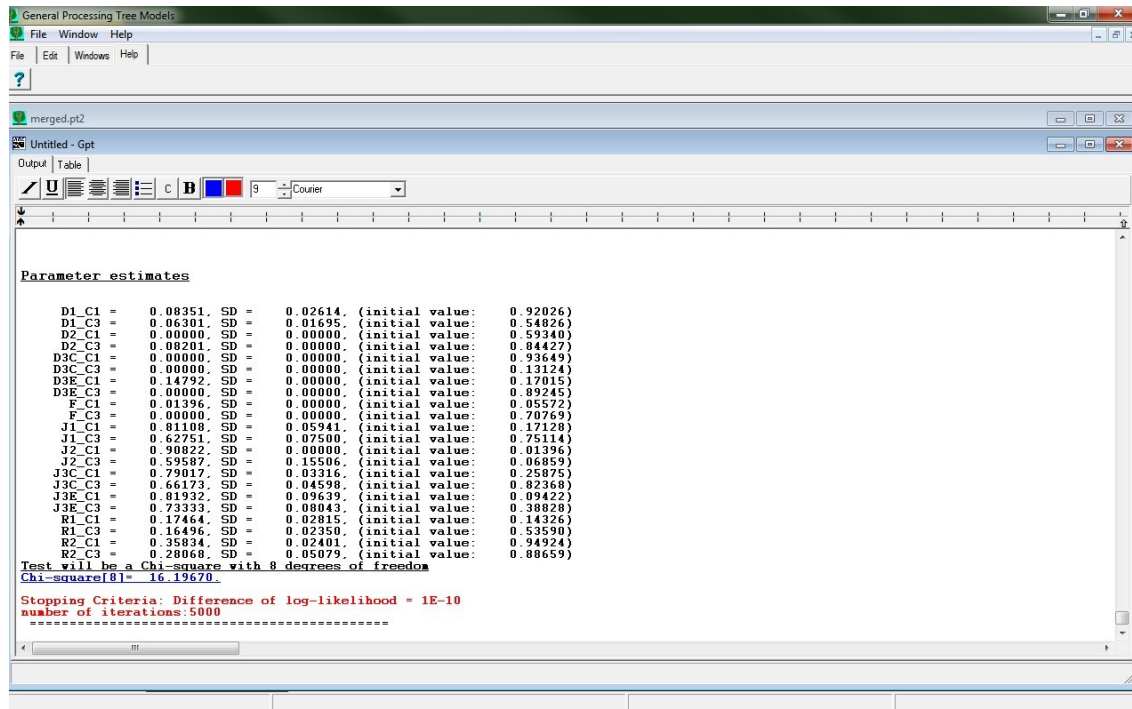


- b. Select the model called Joint Model Unconstrained and the joint dataset (always the first one)



Last update: July 2nd, 2013

c. Click on Run

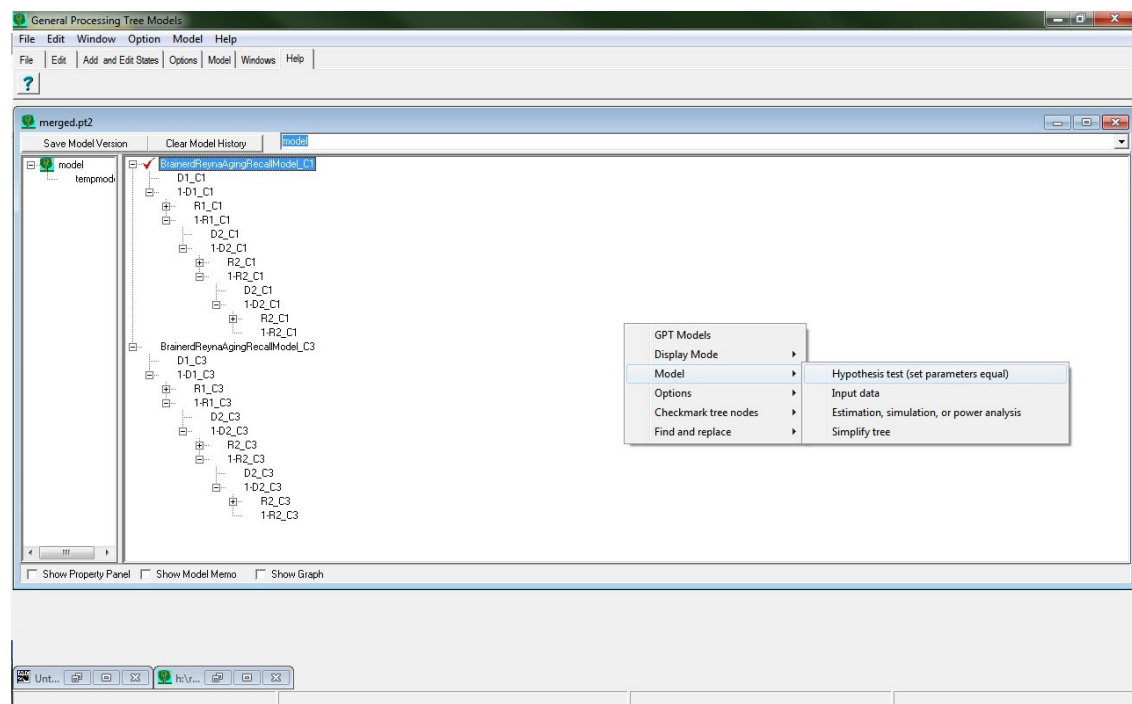


5. Note the goodness of fit of the unconstrained joint model.

Chi-square[8]= 16.19670.

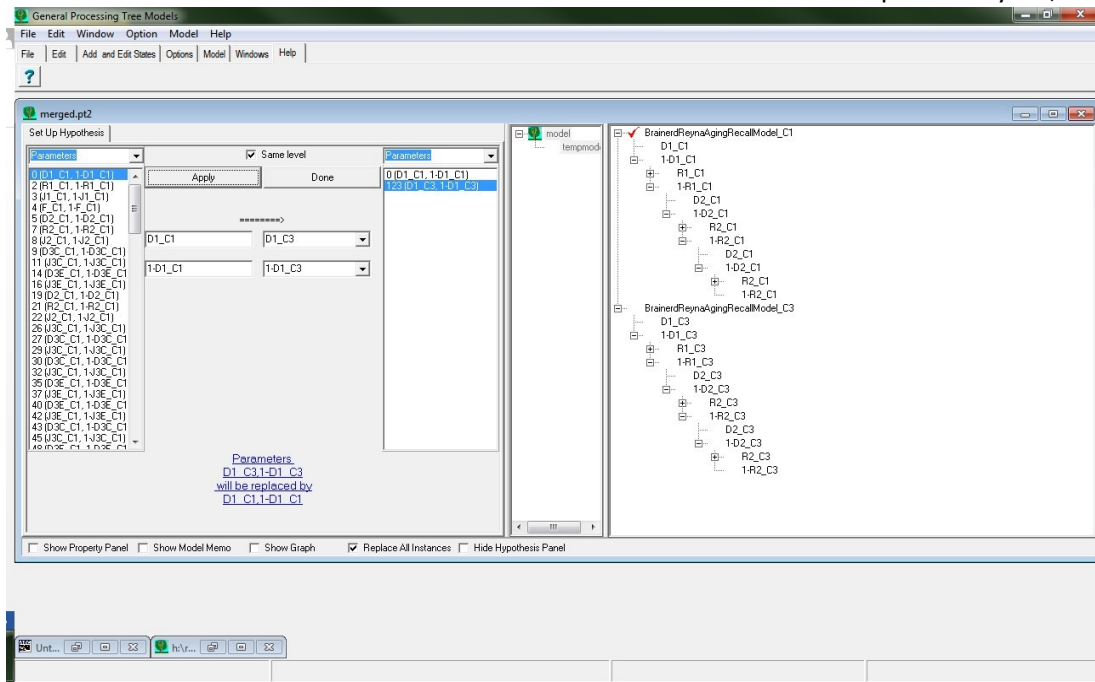
6. We are now ready to create the constrained joint model for the condition-wise test.

a. Go to the Hypothesis Test window

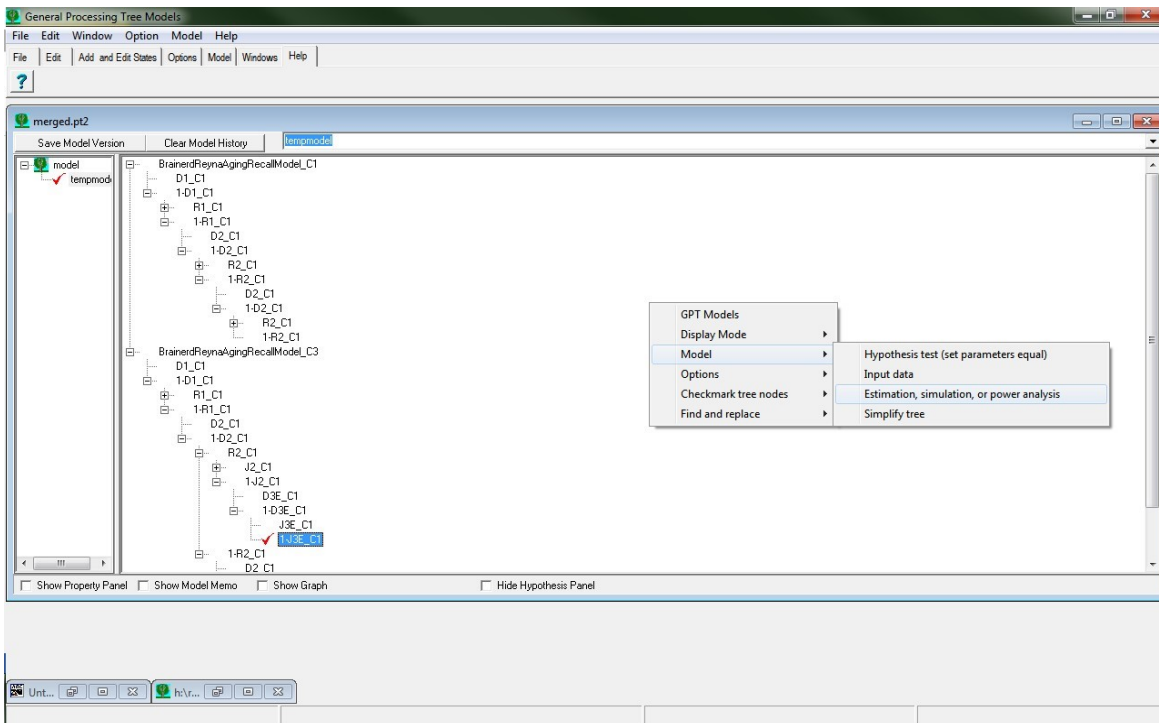


7. Set all the parameters equal to each other between the two conditions using the Apply button (i.e., D1C1 = D1C3 , ..., R2C1 = R2C3)

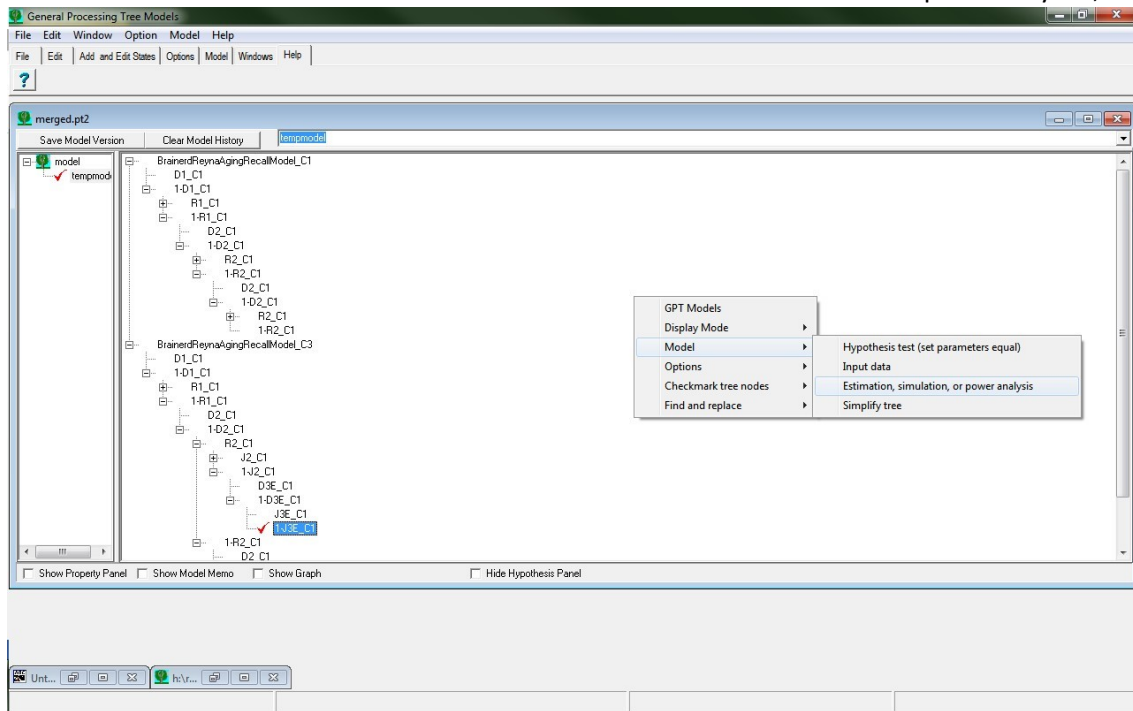
Last update: July 2nd, 2013



8. Click Done when you finish
9. The model you created will be called “tempmodel”
10. Obtain the parameter estimates for the constrained model.
 - a. Go to the Estimation, Simulation, and Power analysis window

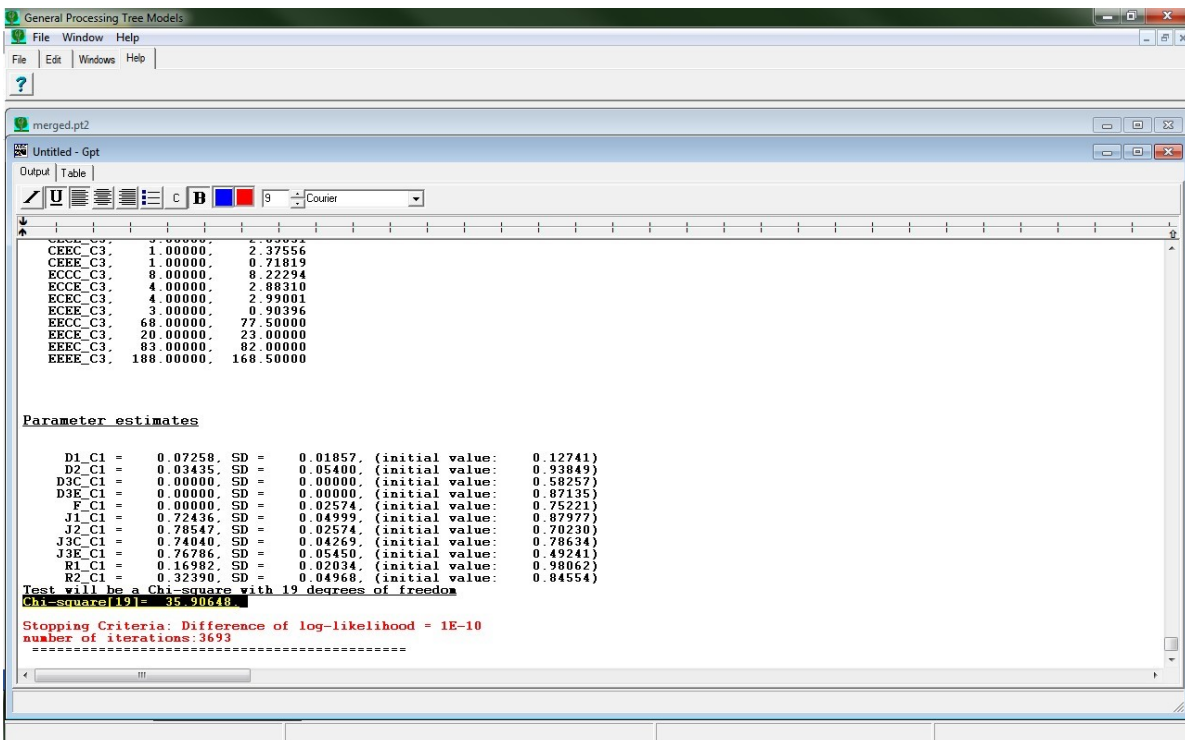


- b. Select the model called Joint Model Constrained Omnibus and the joint dataset (always the first one)

Last update: July 2nd, 2013

C.

Click on Run



- One can now perform the hypothesis test by comparing the two goodness of fit statistics.

Constrained: Chi-square[19] = 35.90648
 Unconstrained: Chi-square[8] = 16.19670

- Calculate the difference between the two chi-square statistics to obtain the test statistic.

Last update: July 2nd, 2013**Test statistic:** $35.91 - 16.20 = 19.71$

13. Compare the test statistic to the critical value. For instance, at a significance criterion of .05, the critical value of the test statistic is $\chi^2(11) = 19.68$. Because the test statistic is higher than the critical value, we reject the null hypothesis of no difference between Conditions 1 and 3.

E. Between-Condition Tests: Parameter-wise test

In this section, we will perform a parameter-wise test using the data from Conditions 1 and 3, namely the null hypothesis that $R_2, \text{Condition 1} = R_2, \text{Condition 3}$.

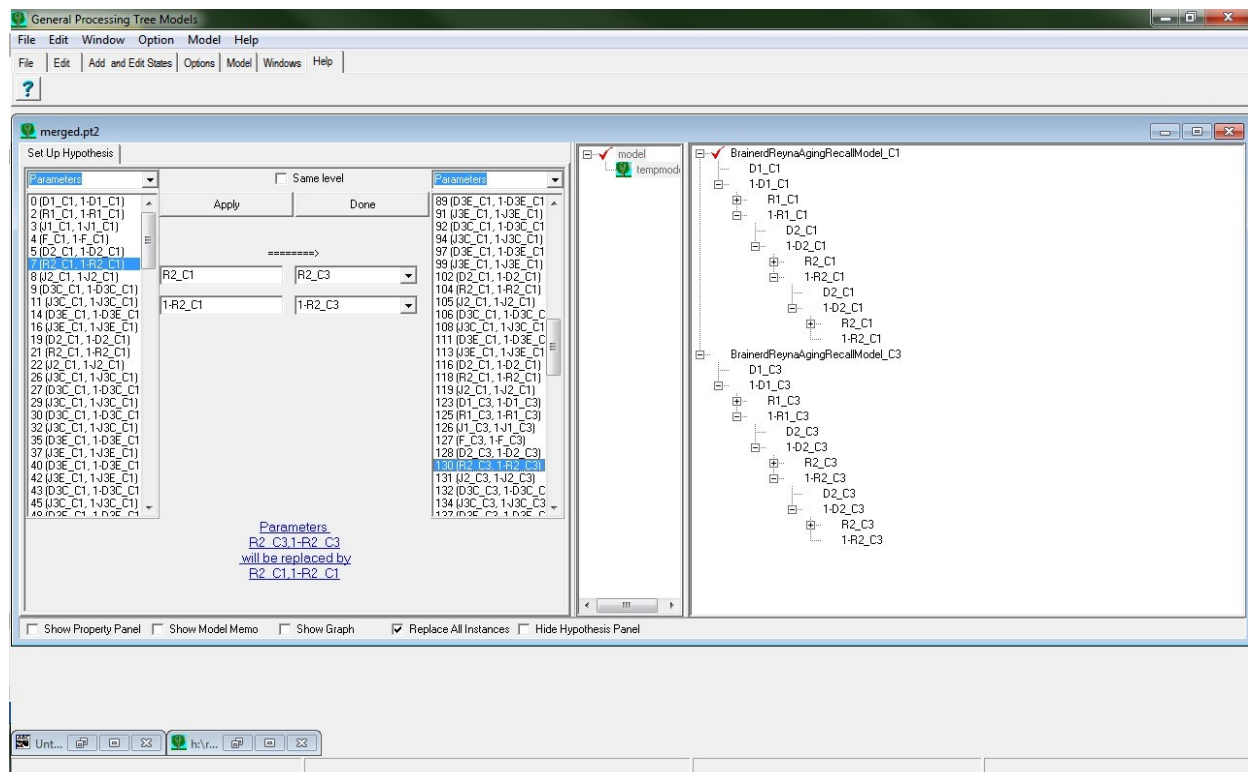
1. Construct an unconstrained joint model between the two conditions.
2. Record the goodness of fit of the unconstrained model.

Chi-square[8]= 16.19670

3. We are now ready to construct the constrained model.

Right Click ---> Model ---> Hypothesis Test

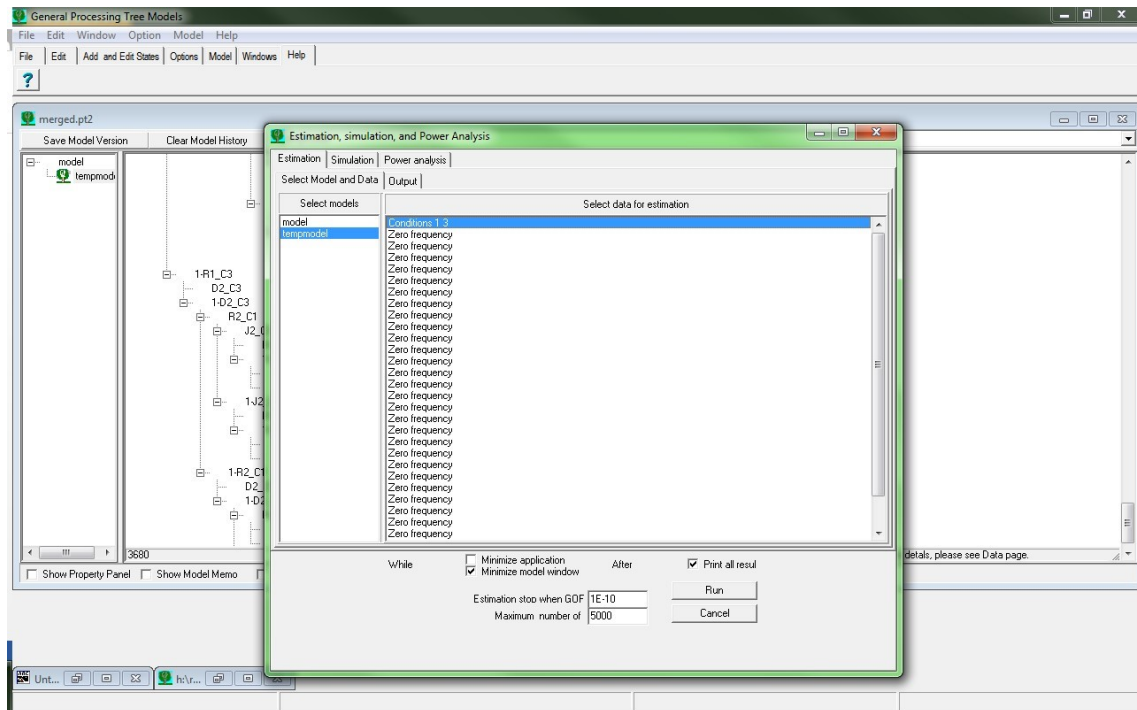
4. Set equal the parameter "R2_C1" to the parameter "R2_C3"



5. Click Apply, and then Click Done

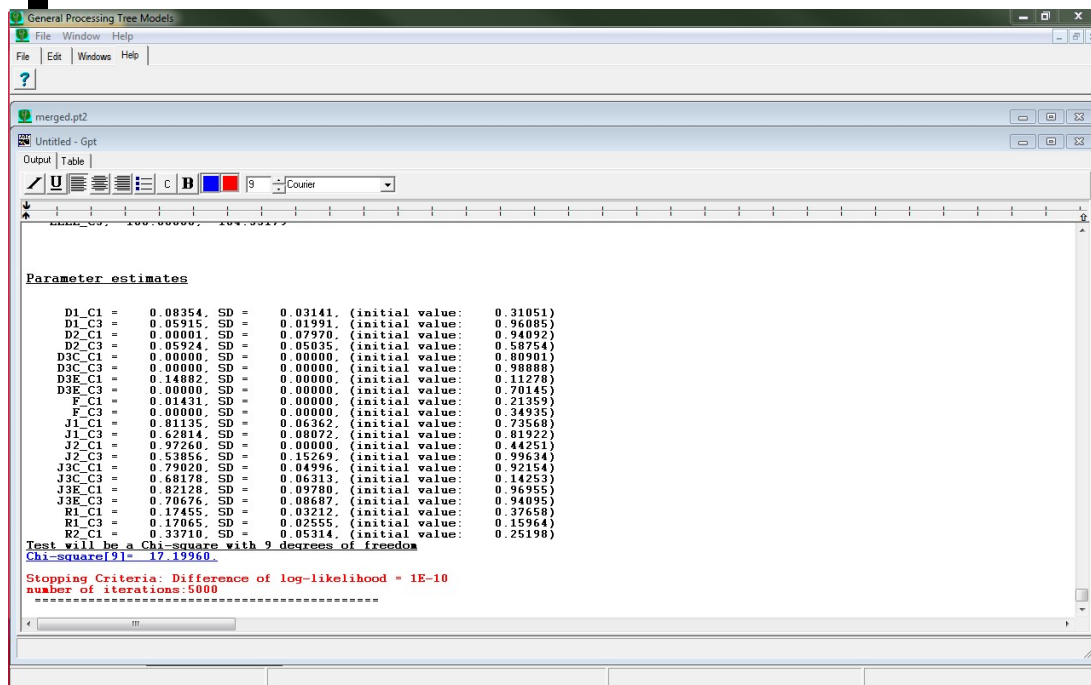
Last update: July 2nd, 2013

6. As before, the model you created will be called “tempmodel”
7. Compute the goodness of fit for the constrained Joint Model just created. Remember to click on the appropriate model when running the parameter estimates.



8. Record the goodness of fit statistic for the joint constrained model.

Chi-square[9] = 17.1996



12. One can now perform the hypothesis test by comparing the two goodness of fit statistics.

Last update: July 2nd, 2013**Constrained = 17.19960****Unconstrained = 16.19670**

13. Calculate the difference between the two chi-square statistics to obtain the test statistic.

$$\text{Test statistic} = 17.2 - 16.2 = 1$$

14. Compare the test statistic to the critical value. For instance, at a significance criterion of .05, the critical value of the test statistic is $\chi^2(1) = 3.84$. Because the test statistic is lower than the critical value, we do not reject the null hypothesis of the test.

References

- Brainerd, C. J., Aydin, C., & Reyna, V. F. (2012). Development of dual-retrieval processes in recall: Learning, forgetting, and reminiscence. *Journal of Memory and Language*, 66, 763-788. doi:10.1016/j.jml.2011.12.002
- Brainerd, C. J., & Reyna, V. F. (2010). Recollective and nonrecollective recall. *Journal of Memory and Language*, 63, 425-445. doi:10.1016/j.jml.2010.05.002
- Brainerd, C. J., Reyna, V. F., & Howe, M. L. (2009). Trichotomous processes in early memory development, aging, and cognitive impairment: A unified theory. *Psychological Review*, 116, 783-832. doi:10.1037/a0016963
- Gomes, C. F. A., Brainerd, C. J., & Stein, L. M. (2013). Effects of emotional valence and arousal on recollective and nonrecollective recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 39, 663-677. doi:10.1037/a0028578
- Hu, X., & Phillips, G. A. (1999). GPT.EXE: A powerful tool for the visualization and analysis of general processing tree models. *Behavior Research Methods, Instruments, & Computers*, 31, 220-234. doi:10.3758/BF03207714