

# Phenotype-Genotype Branch-Site Model

## Users Tutorial Version 1.00

By C T Jones, May 2020

[cjones2@dal.ca](mailto:cjones2@dal.ca)

random][pLasmId

# System Requirements

The PG-BSM is currently available on in Matlab code only. The code requires a licensed version of Matlab installed on your computer, and makes use of the Statistics Toolbox, the Optimization Toolbox, and the Distributed Computing Toolbox. Type `ver` at the Matlab command line to see the toolboxes installed on your computer.

Here is the full list of the toolboxes installed on the computer used to develop the PG-BSM.

>> ver

-----  
MATLAB Version: 9.7.0.1216025 (R2019b) Update 1  
-----

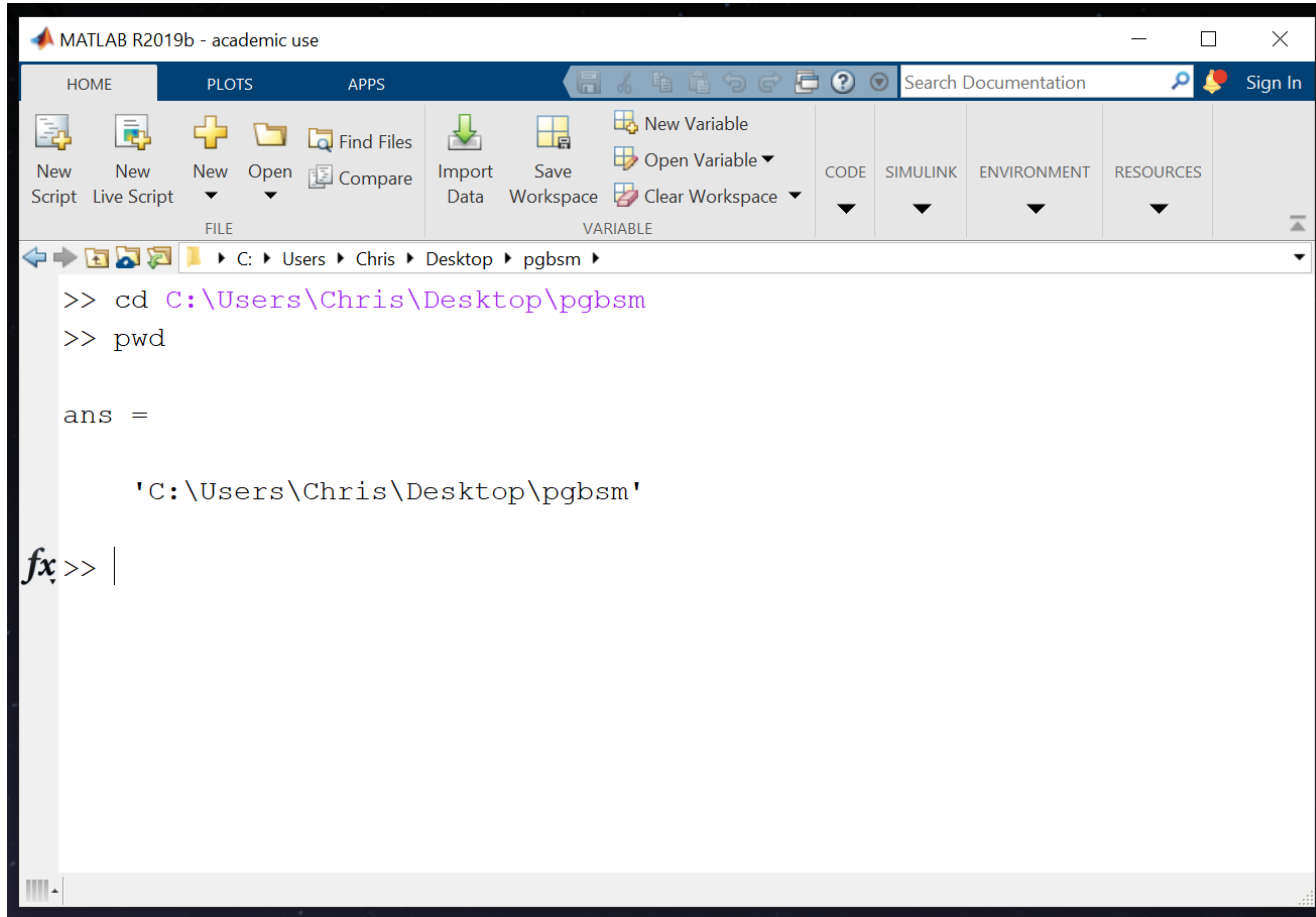
MATLAB	Version 9.7	(R2019b)
Simulink	Version 10.0	(R2019b)
Curve Fitting Toolbox	Version 3.5.10	(R2019b)
Global Optimization Toolbox	Version 4.2	(R2019b)
MATLAB Coder	Version 4.3	(R2019b)
MATLAB Compiler	Version 7.1	(R2019b)
MATLAB Compiler SDK	Version 6.7	(R2019b)
Optimization Toolbox	Version 8.4	(R2019b)
Parallel Computing Toolbox	Version 7.1	(R2019b)
Statistics and Machine Learning Toolbox	Version 11.6	(R2019b)
Symbolic Math Toolbox	Version 8.4	(R2019b)

NOTE: The PG-BSM code was updated and tested by the developer (C. T. Jones) at the time of the writing of this document (August 2020) but has not been tested by other users. Unforeseen issues may therefore arise, and if they do please contact Dr. Jones at [cjones2@dal.ca](mailto:cjones2@dal.ca). Also, we are uncertain about the limitations of the model in terms of data complexity. The PG-BSM works well when the phenotype changed on only a few internal branches of the tree but may not do so when the phenotype changed many times and/or when some changes result in ambiguities (e.g., reversions, parallel evolution etc.).

# Getting Started

Open Matlab and at the prompt change the current directory to the location of the pgbsm folder on your computer.

For example, my copy of the pgbsm folder is on my desktop:

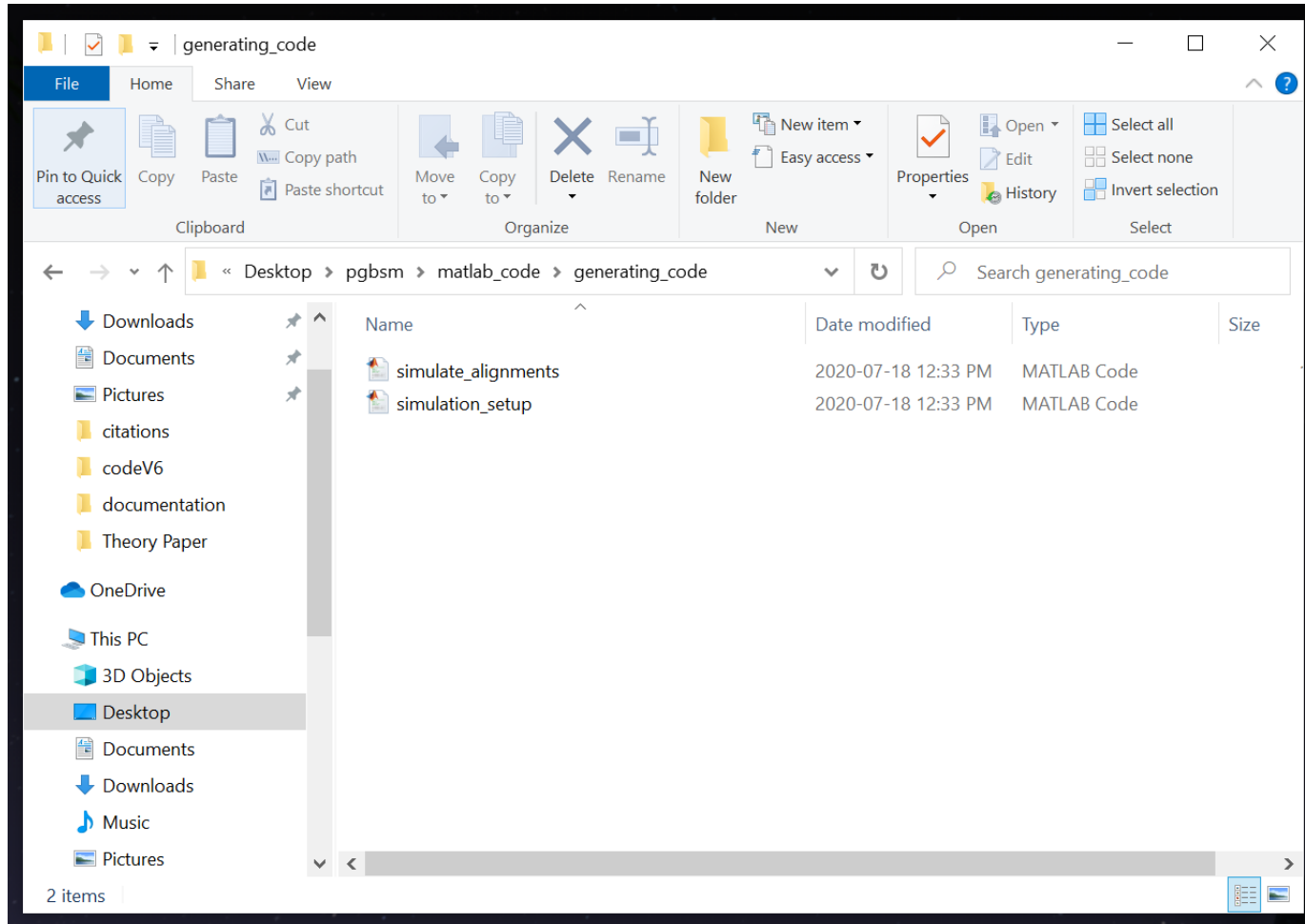


This is the first step every time you start a new Matlab session to use the PG-BSM code.

# Generating Simulated Alignments

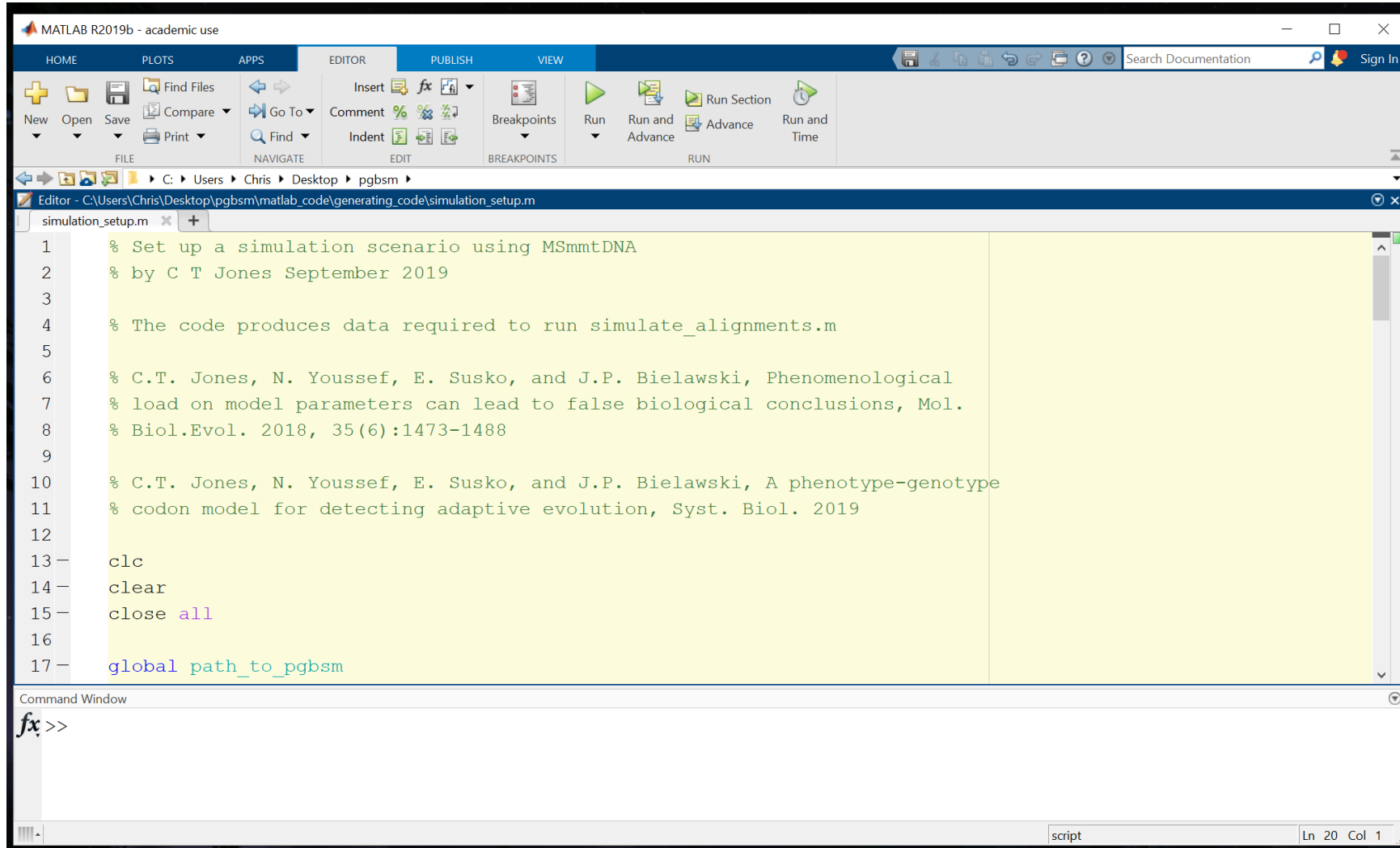
Let us start by simulating some alignments.

Go into your pgbsm folder and find the folder called generating\_code.



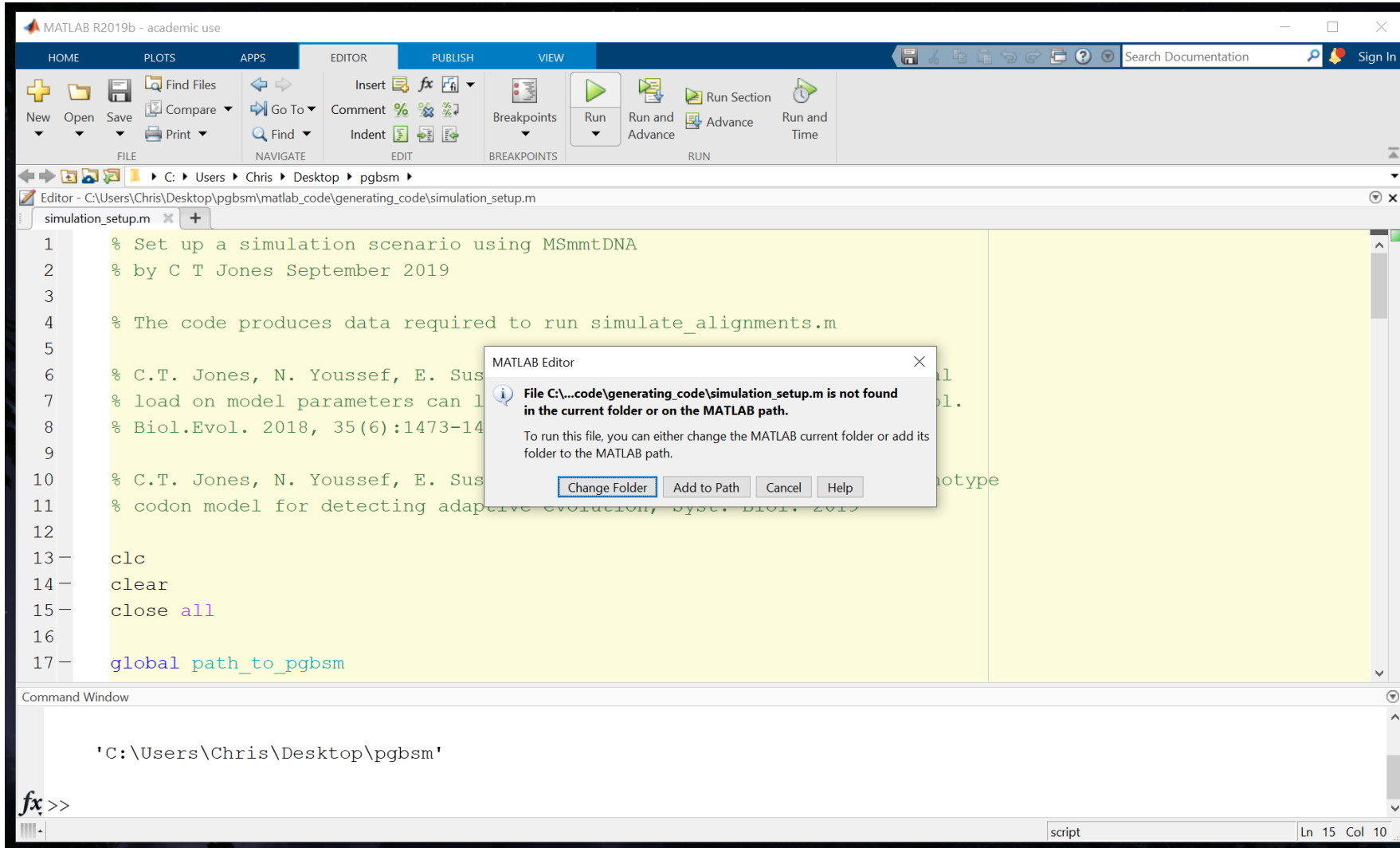
# Generating Simulated Alignments

Right click on the Matlab script `simulation_setup.m` to open it in your Matlab window.



# Generating Simulated Alignments

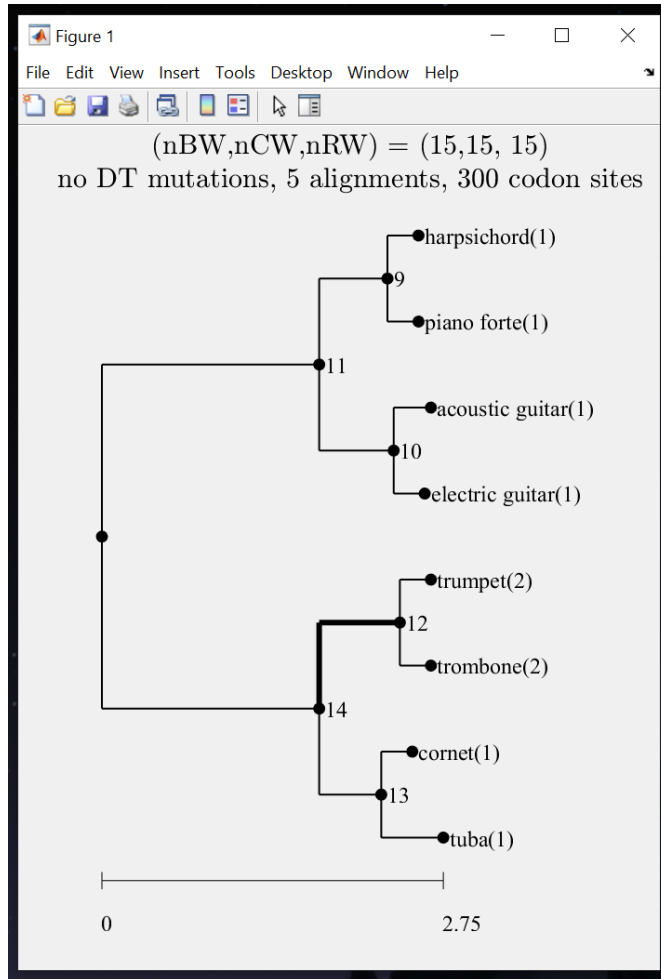
Press the big green triangle to run the script. When you do so a window should open as illustrated below. Click on “Add to Path” to run the code. This will tell Matlab where to find the simulation\_setup.m script.



Click on “Add to Path” anytime you this window opens when you attempt to run a script.

# Generating Simulated Alignments

When you run `simulation_setup.m`, a figure showing a tree will appear. Branches over which some site-specific landscapes change in concert with a change in phenotype are indicated by thicker lines. In this case changes were set to occur over branch 12 only. The phenotype associated with each taxon is indicated by a number. The number 1 corresponds to the phenotype at the root of the tree. The expected number of single nucleotide substitutions per codon site from root to the longest tip is indicated under the tree.

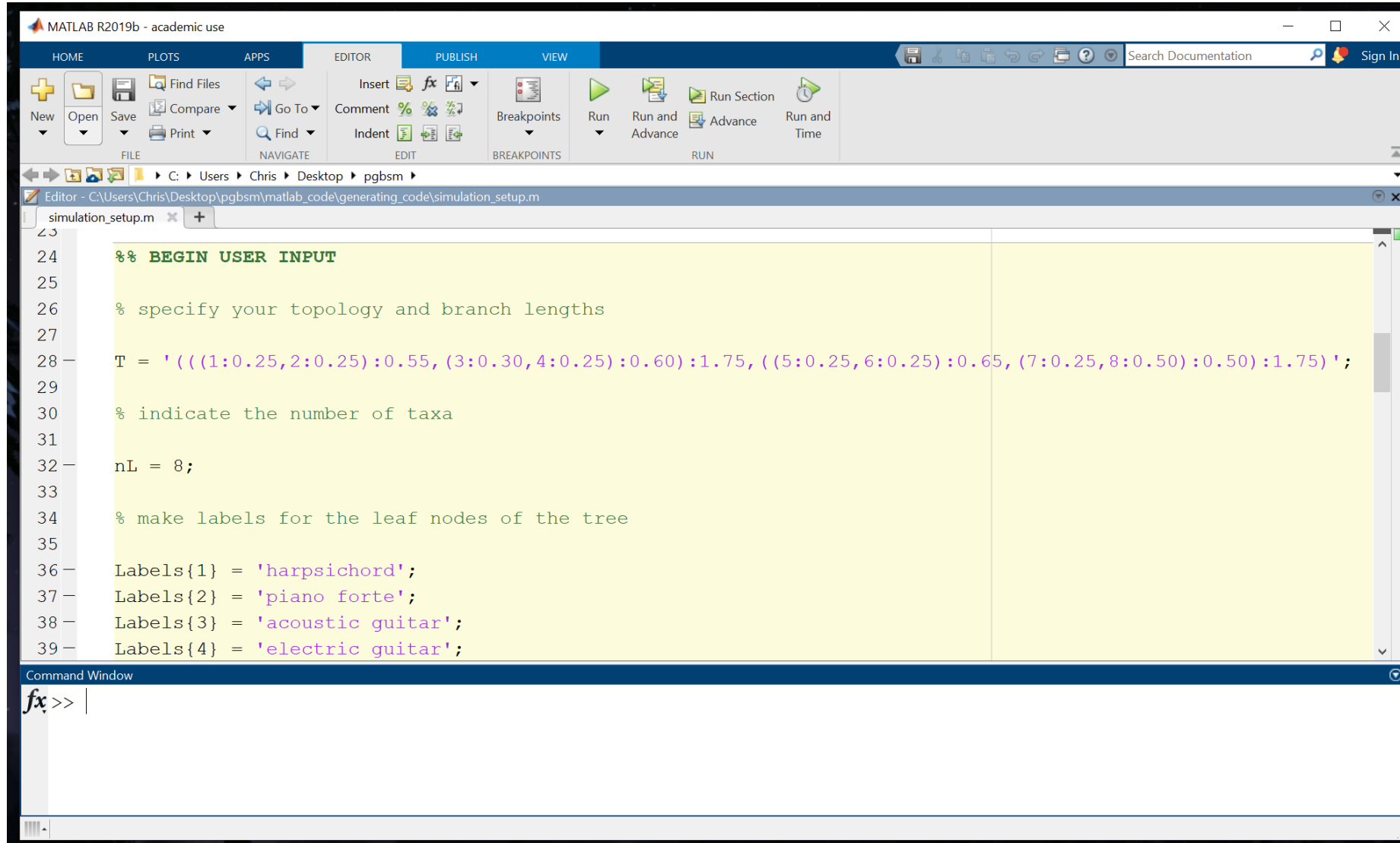


By modifying the appropriate sections of the `simulation_setup.m` script you can specify the topology and branch lengths of the binary tree, taxon names, the branches over which the phenotype will change, the number of codon sites that will evolve in a way consistent with the CW, RW or BW process (see PG-BSM Concept in the documentation folder), the proportion of double and triple mutations permitted, the number of codon sites in the alignment, and the number of alignments you want to simulate. The following slides show how to change these parameters to suit your needs.



# Generating Simulated Alignments

The binary tree is specified in standard nested format with branch lengths indicated after colons. The parameter nL that indicates the number of taxa in the tree must be consistent with the tree T.



The image shows the MATLAB R2019b interface. The main window displays a script named `simulation_setup.m` with the following code:

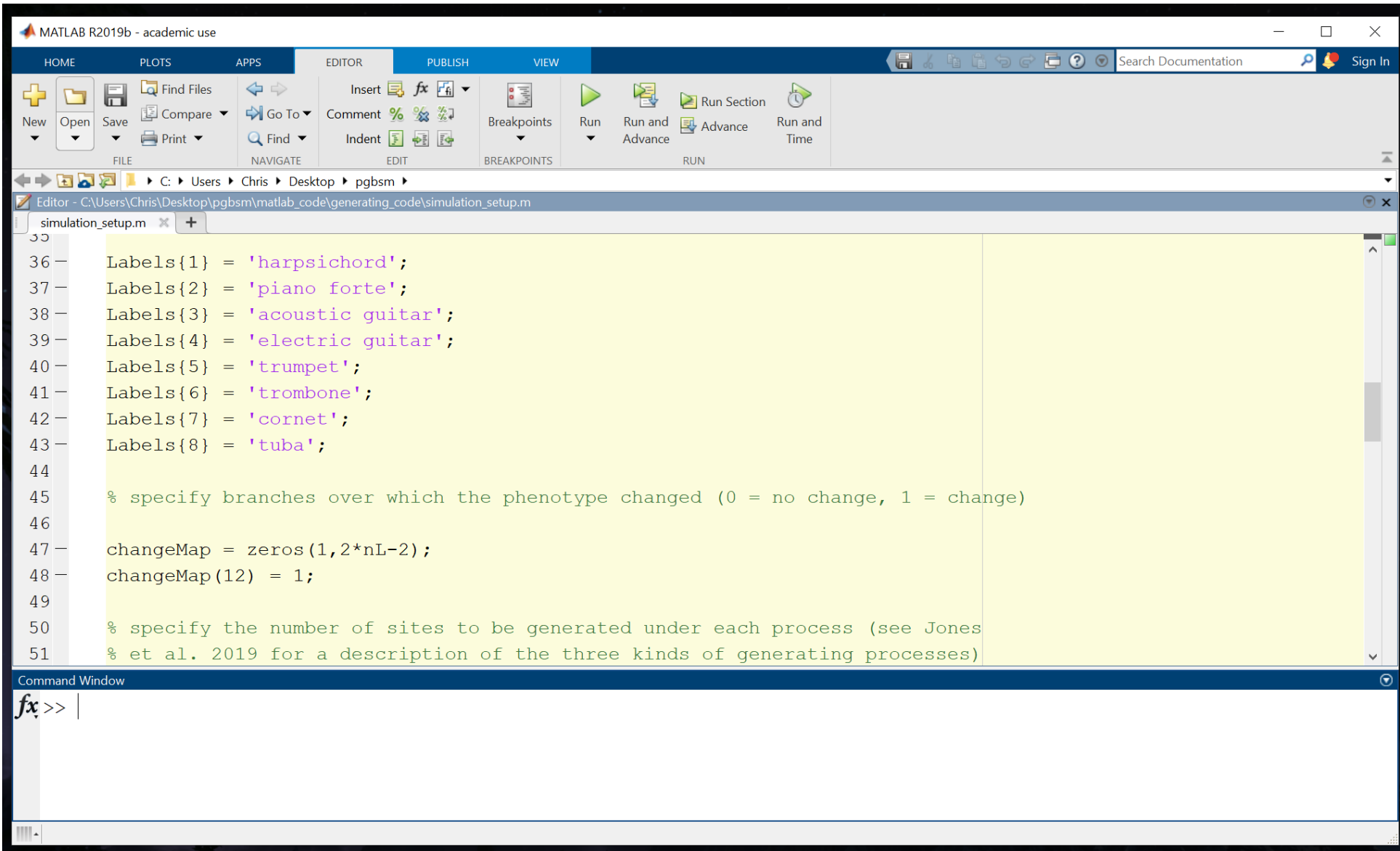
```
23  
24 %% BEGIN USER INPUT  
25  
26 % specify your topology and branch lengths  
27  
28 T = '((1:0.25,2:0.25):0.55,(3:0.30,4:0.25):0.60):1.75,((5:0.25,6:0.25):0.65,(7:0.25,8:0.50):0.50):1.75';  
29  
30 % indicate the number of taxa  
31  
32 nL = 8;  
33  
34 % make labels for the leaf nodes of the tree  
35  
36 Labels{1} = 'harpsichord';  
37 Labels{2} = 'piano forte';  
38 Labels{3} = 'acoustic guitar';  
39 Labels{4} = 'electric guitar';
```

The Command Window at the bottom shows the MATLAB prompt `fx>>`.



# Generating Simulated Alignments

Taxon names should be made to be as simple as possible and should not contain special symbols such as underscores.



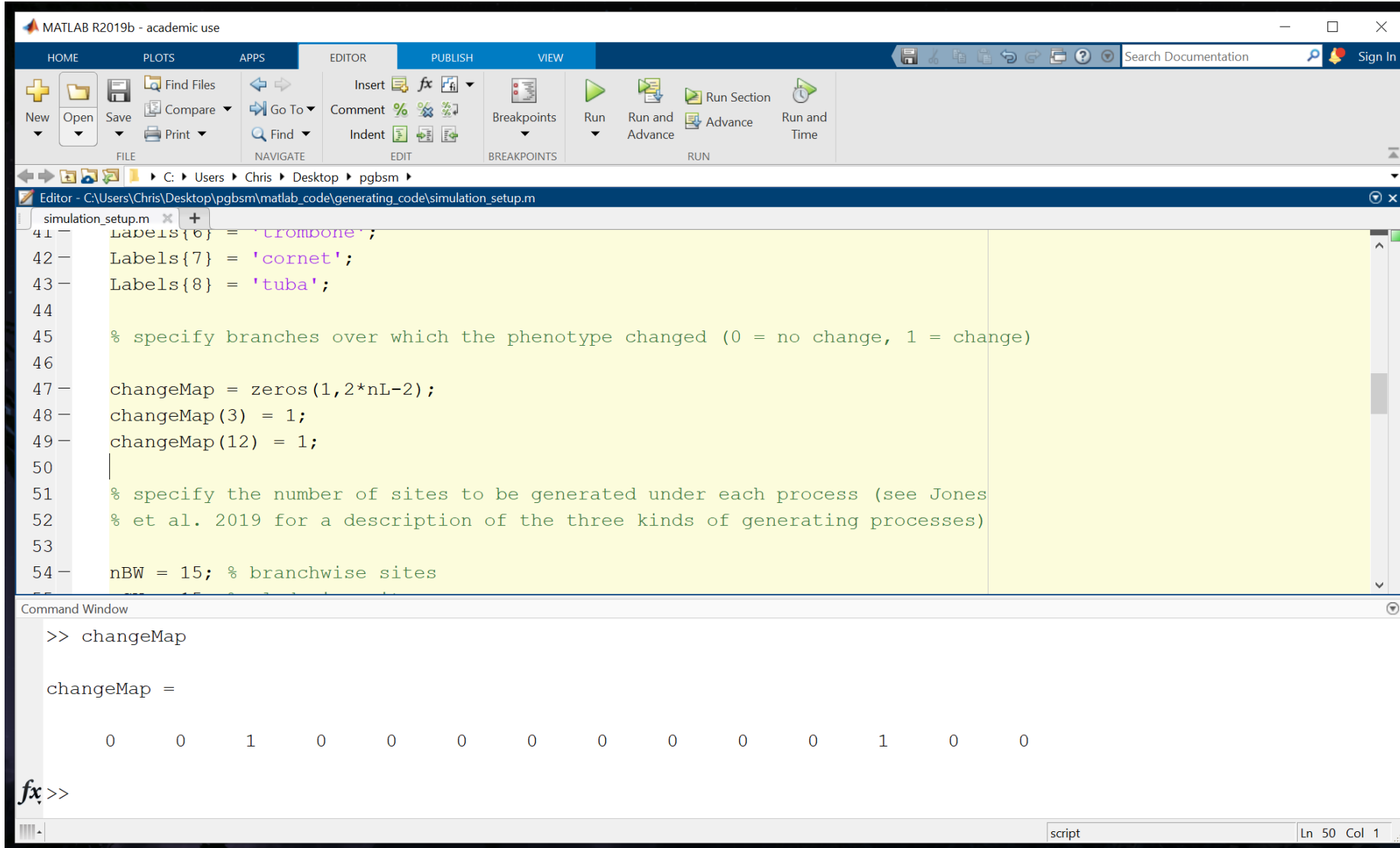
The image shows the MATLAB R2019b interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The toolbar contains icons for New, Open, Save, Find Files, Compare, Print, Go To, Find, Insert, Comment, Indent, Breakpoints, Run, Run and Advance, Run Section, and Run and Time. The main editor window displays a script named `simulation_setup.m` located at `C:\Users\Chris\Desktop\pgbsm\matlab_code\generating_code\simulation_setup.m`. The script defines a cell array `Labels` with 8 elements, each representing a taxon name. It also defines a `changeMap` matrix and a comment about the number of sites to be generated under each process.

```
35  
36 Labels{1} = 'harpsichord';  
37 Labels{2} = 'piano forte';  
38 Labels{3} = 'acoustic guitar';  
39 Labels{4} = 'electric guitar';  
40 Labels{5} = 'trumpet';  
41 Labels{6} = 'trombone';  
42 Labels{7} = 'cornet';  
43 Labels{8} = 'tuba';  
44  
45 % specify branches over which the phenotype changed (0 = no change, 1 = change)  
46  
47 changeMap = zeros(1,2*nL-2);  
48 changeMap(12) = 1;  
49  
50 % specify the number of sites to be generated under each process (see Jones  
51 % et al. 2019 for a description of the three kinds of generating processes)
```

The Command Window at the bottom shows the MATLAB prompt `fx>> |`.

# Generating Simulated Alignments

Branches over which the phenotype will change in concert with changes in site-specific landscapes are indicated by the number one in the changeMap vector. Here for example changes will occur along branches 3 and 12.



```
MATLAB R2019b - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW
New Open Save Compare Find Files Go To Find Comment Indent Breakpoints Run Run and Advance Run and Time
FILE NAVIGATE EDIT BREAKPOINTS RUN

Editor - C:\Users\Chris\Desktop\pgbsm\matlab_code\generating_code\simulation_setup.m
simulation_setup.m
41 Labels{6} = 'trombone';
42 Labels{7} = 'cornet';
43 Labels{8} = 'tuba';
44
45 % specify branches over which the phenotype changed (0 = no change, 1 = change)
46
47 changeMap = zeros(1,2*nL-2);
48 changeMap(3) = 1;
49 changeMap(12) = 1;
50
51 % specify the number of sites to be generated under each process (see Jones
52 % et al. 2019 for a description of the three kinds of generating processes)
53
54 nBW = 15; % branchwise sites

Command Window
>> changeMap

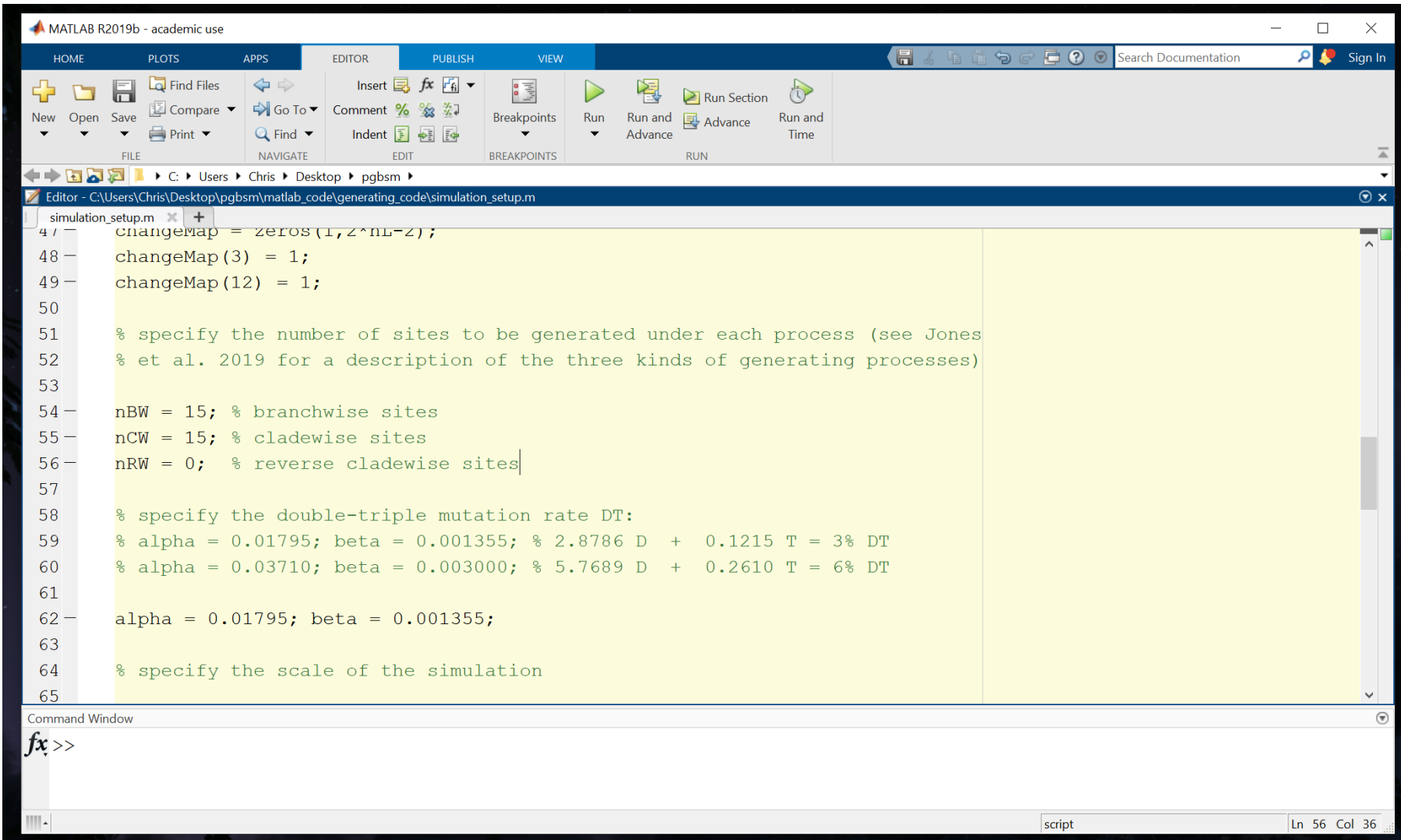
changeMap =

    0    0    1    0    0    0    0    0    0    0    0    0    1    0    0

fx>>
```

# Generating Simulated Alignments

The number of sites that will undergo a cladewise change in the stringency of selection (CW sites), a reverse cladewise change in the stringency of selection (rCW) or a branchwise change corresponding to a site-specific peak shift (BW) should typically be some small fraction of the total number of codon sites.



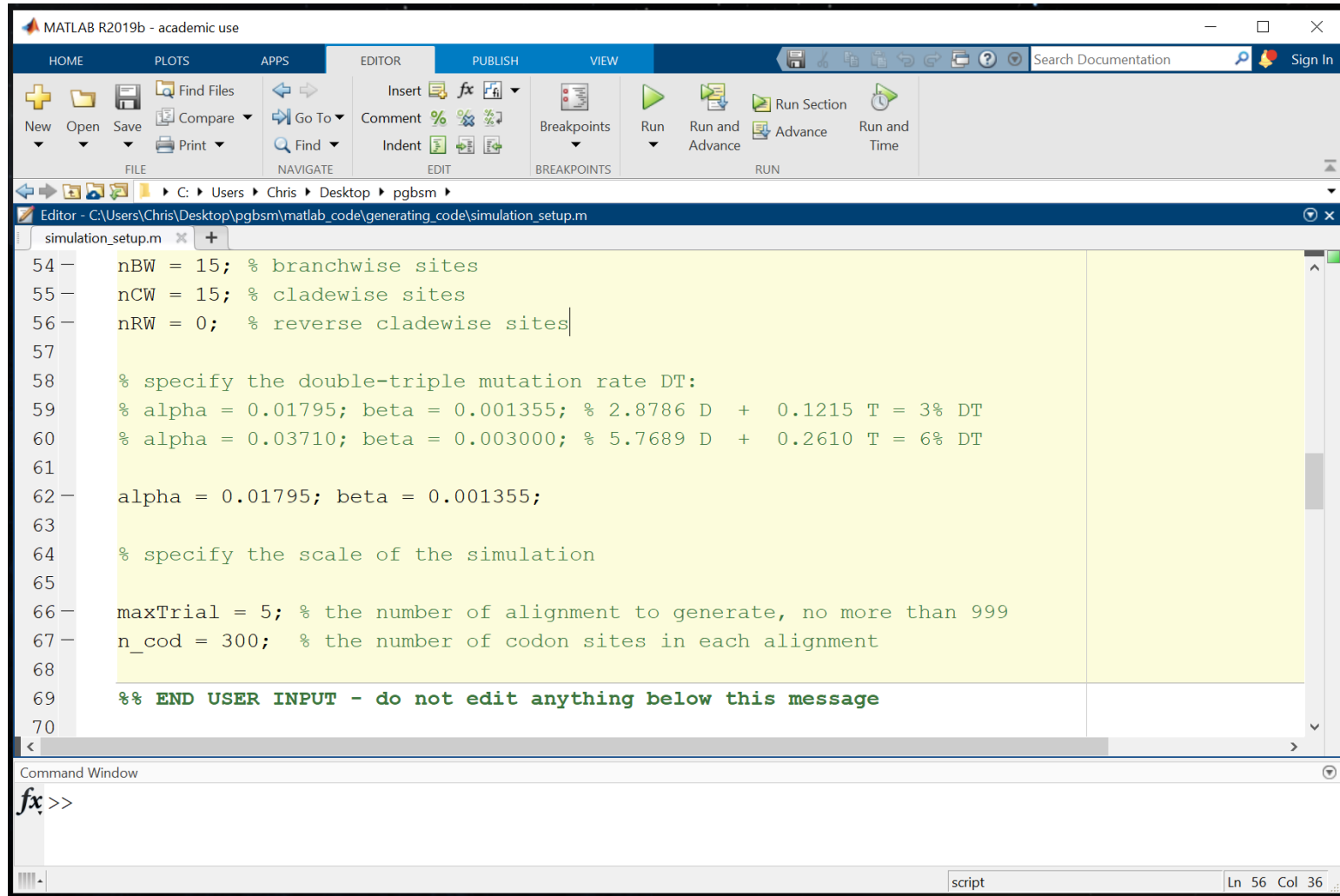
The image shows the MATLAB R2019b - academic use interface. The main window displays a script named `simulation_setup.m` in the Editor. The script defines parameters for generating simulated alignments, including the number of sites for different types of changes (branchwise, cladewise, reverse cladewise) and the double-triple mutation rate (DT).

```
4 / changeMap = zeros(1, Z^NL-Z);
48 changeMap(3) = 1;
49 changeMap(12) = 1;
50
51 % specify the number of sites to be generated under each process (see Jones
52 % et al. 2019 for a description of the three kinds of generating processes)
53
54 nBW = 15; % branchwise sites
55 nCW = 15; % cladewise sites
56 nRW = 0; % reverse cladewise sites
57
58 % specify the double-triple mutation rate DT:
59 % alpha = 0.01795; beta = 0.001355; % 2.8786 D + 0.1215 T = 3% DT
60 % alpha = 0.03710; beta = 0.003000; % 5.7689 D + 0.2610 T = 6% DT
61
62 alpha = 0.01795; beta = 0.001355;
63
64 % specify the scale of the simulation
65
```

The Command Window at the bottom shows the MATLAB prompt `>>`. The status bar at the bottom right indicates the current position in the script: `script`, `Ln 56 Col 36`.

# Generating Simulated Alignments

It has been noted that the rate fixation of double and triple mutations (DT mutations) can mislead standard codon substitution model to falsely infer adaptation. The probability of false inference is reduced under the PG-BSM because such event would have to co-occur with a change in phenotype to be detected. This can be tested by including DT mutations in the simulation.



The image shows the MATLAB R2019b interface with the Editor window open to a script named `simulation_setup.m`. The script is located at `C:\Users\Chris\Desktop\pgbsm\matlab_code\generating_code\simulation_setup.m`. The script contains the following code:

```
54 nBW = 15; % branchwise sites
55 nCW = 15; % cladewise sites
56 nRW = 0; % reverse cladewise sites
57
58 % specify the double-triple mutation rate DT:
59 % alpha = 0.01795; beta = 0.001355; % 2.8786 D + 0.1215 T = 3% DT
60 % alpha = 0.03710; beta = 0.003000; % 5.7689 D + 0.2610 T = 6% DT
61
62 alpha = 0.01795; beta = 0.001355;
63
64 % specify the scale of the simulation
65
66 maxTrial = 5; % the number of alignment to generate, no more than 999
67 n_cod = 300; % the number of codon sites in each alignment
68
69 %% END USER INPUT - do not edit anything below this message
70
```

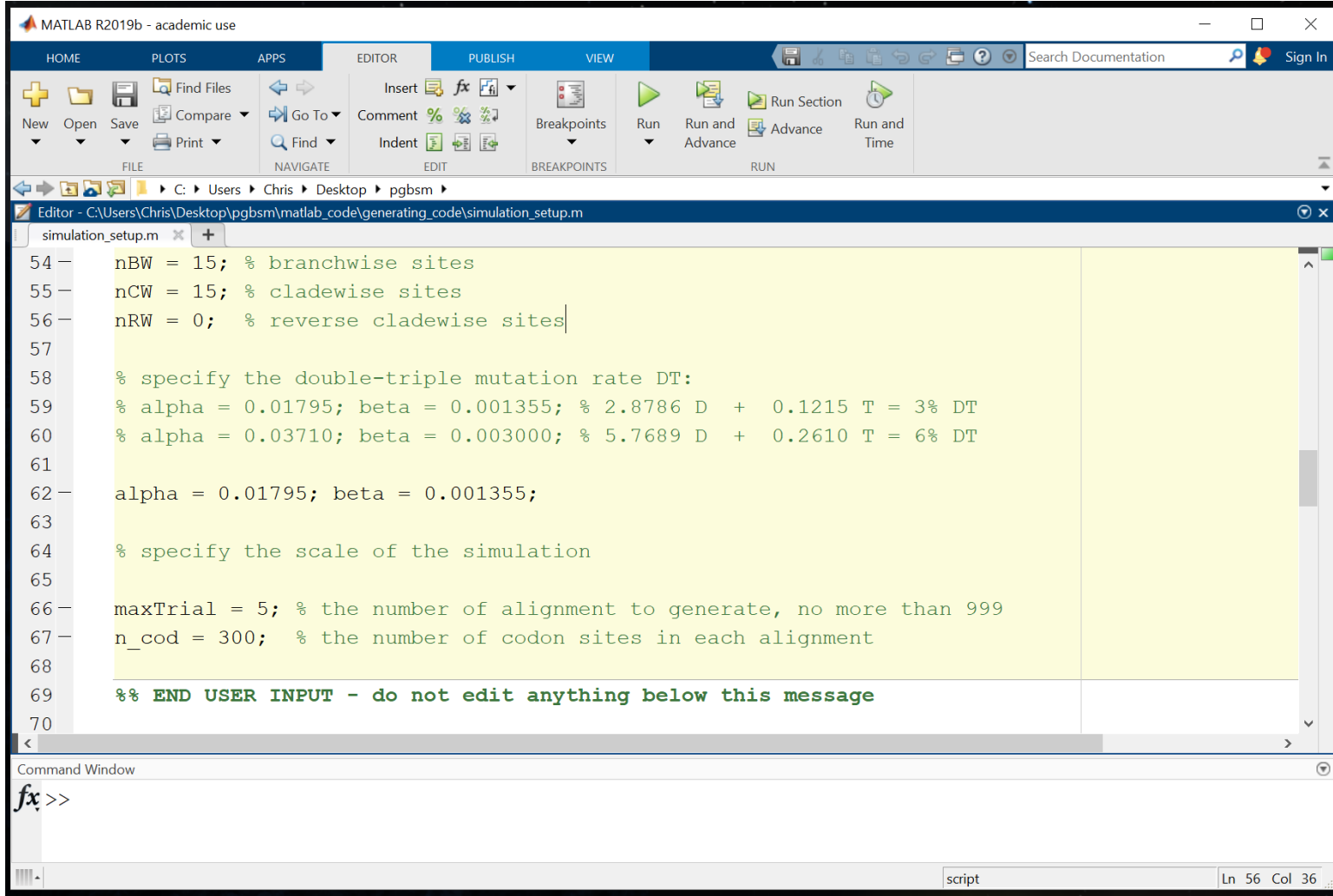
The Command Window at the bottom shows the MATLAB prompt `fx>>`. The status bar at the bottom right indicates the current position is `Ln 56 Col 36`.

The parameter  $\alpha$  determines the rate at which double mutations will arise, and  $\beta$  the rate at which triple mutations will arise.

Two settings are indicated that correspond to a total of 3% or 6% DT mutations. Current estimates of the DT mutation rate are close to 1% (see Jones et al. 2020 for citations).

# Generating Simulated Alignments

The number of alignments generated is indicated by `maxTrial` and the number of codons sites in each alignment by `n_cod`.



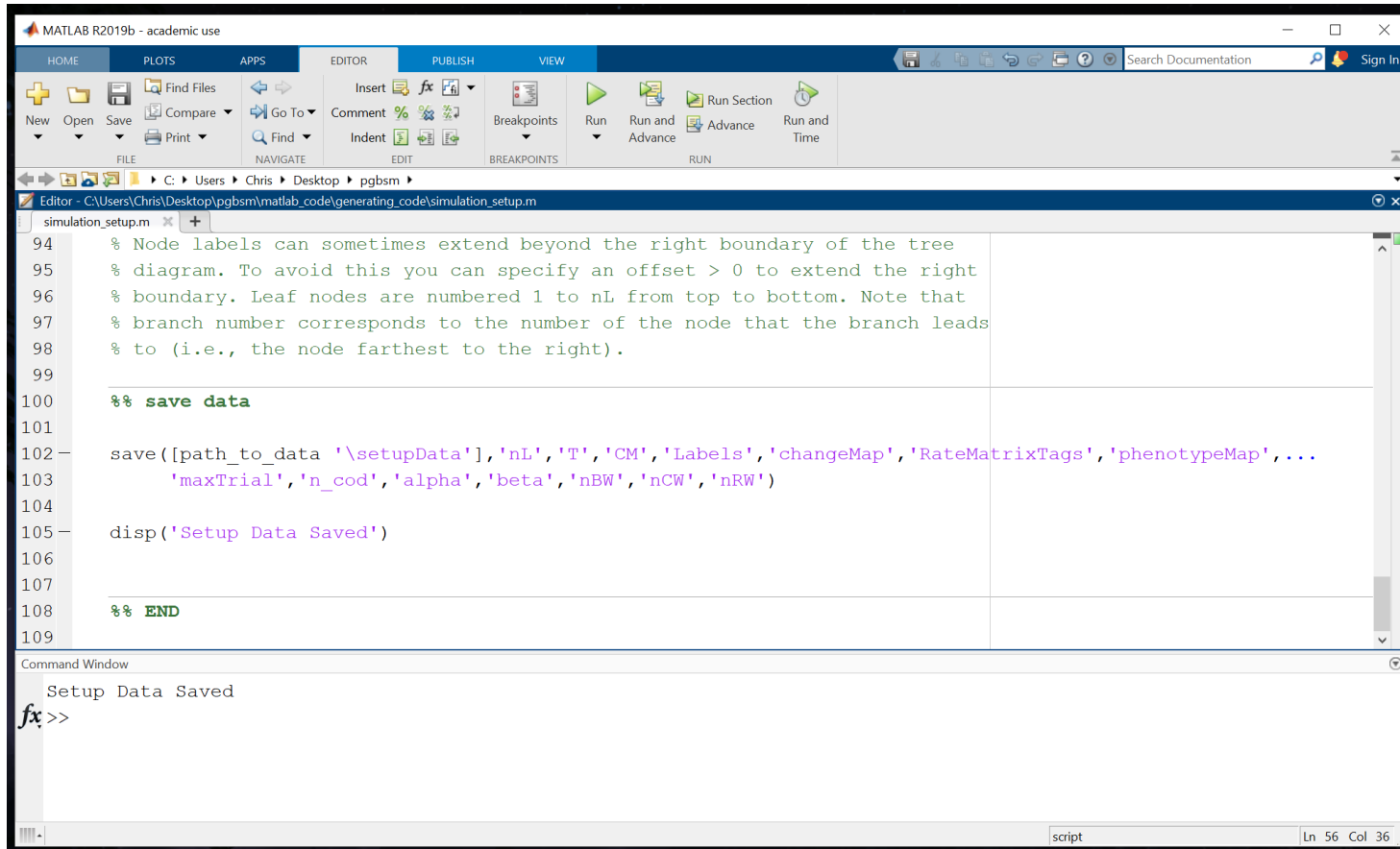
The image shows the MATLAB R2019b interface with the following components:

- Toolbar:** Includes icons for File (New, Open, Save, Print), Navigate (Go To, Find), Edit (Insert, Comment, Indent), Breakpoints, Run (Run, Run and Advance, Run Section, Advance, Run and Time), and a Search Documentation field.
- Editor:** Displays the script `simulation_setup.m` with the following code:

```
54 - nBW = 15; % branchwise sites
55 - nCW = 15; % cladewise sites
56 - nRW = 0; % reverse cladewise sites
57
58 % specify the double-triple mutation rate DT:
59 % alpha = 0.01795; beta = 0.001355; % 2.8786 D + 0.1215 T = 3% DT
60 % alpha = 0.03710; beta = 0.003000; % 5.7689 D + 0.2610 T = 6% DT
61
62 - alpha = 0.01795; beta = 0.001355;
63
64 % specify the scale of the simulation
65
66 - maxTrial = 5; % the number of alignment to generate, no more than 999
67 - n_cod = 300; % the number of codon sites in each alignment
68
69 %% END USER INPUT - do not edit anything below this message
70
```
- Command Window:** Shows the MATLAB prompt `>>`.
- Status Bar:** Indicates the current position is at line 56, column 36.

# Generating Simulated Alignments

Running `simulation_setup.m` automatically generates and saves all the information required for your simulation and produces a figure of your tree. The saved information is used when you run the script `simulate_alignments.m`.

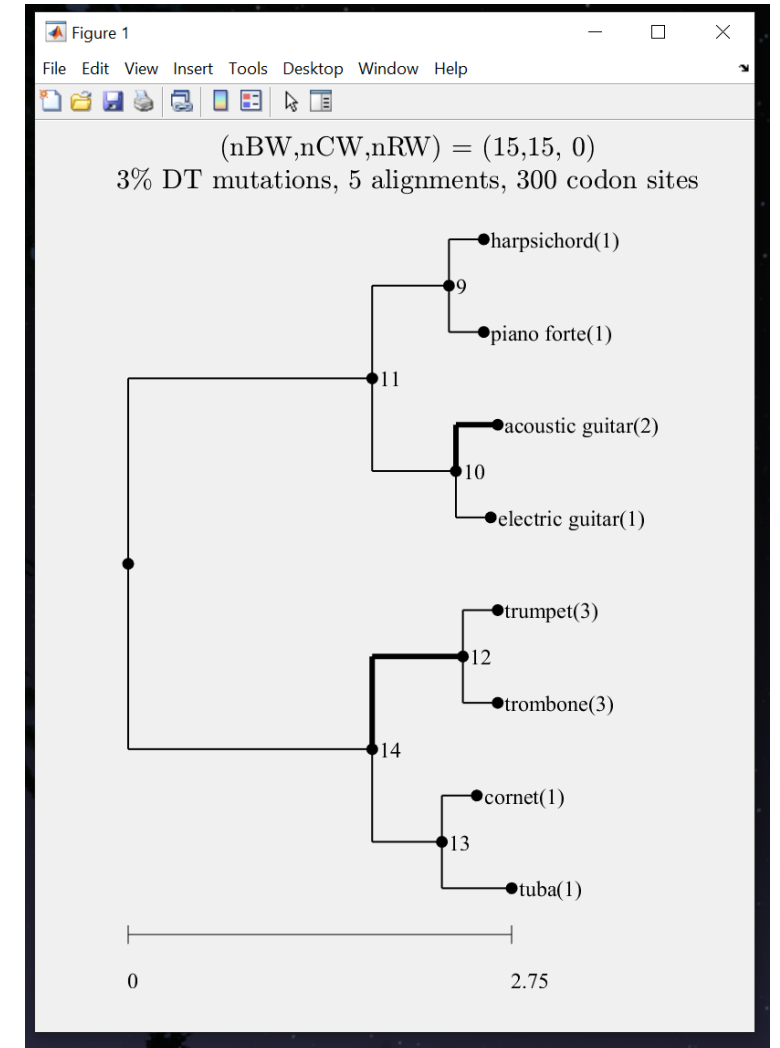


The image shows the MATLAB R2019b interface. The Editor window displays the `simulation_setup.m` script. The Command Window shows the output "Setup Data Saved" and the prompt `fx>>`.

```
94 % Node labels can sometimes extend beyond the right boundary of the tree
95 % diagram. To avoid this you can specify an offset > 0 to extend the right
96 % boundary. Leaf nodes are numbered 1 to nL from top to bottom. Note that
97 % branch number corresponds to the number of the node that the branch leads
98 % to (i.e., the node farthest to the right).
99
100 %% save data
101
102 save([path_to_data 'setupData'], 'nL', 'T', 'CM', 'Labels', 'changeMap', 'RateMatrixTags', 'phenotypeMap', ...
103      'maxTrial', 'n_cod', 'alpha', 'beta', 'nBW', 'nCW', 'nRW')
104
105 disp('Setup Data Saved')
106
107
108 %% END
109
```

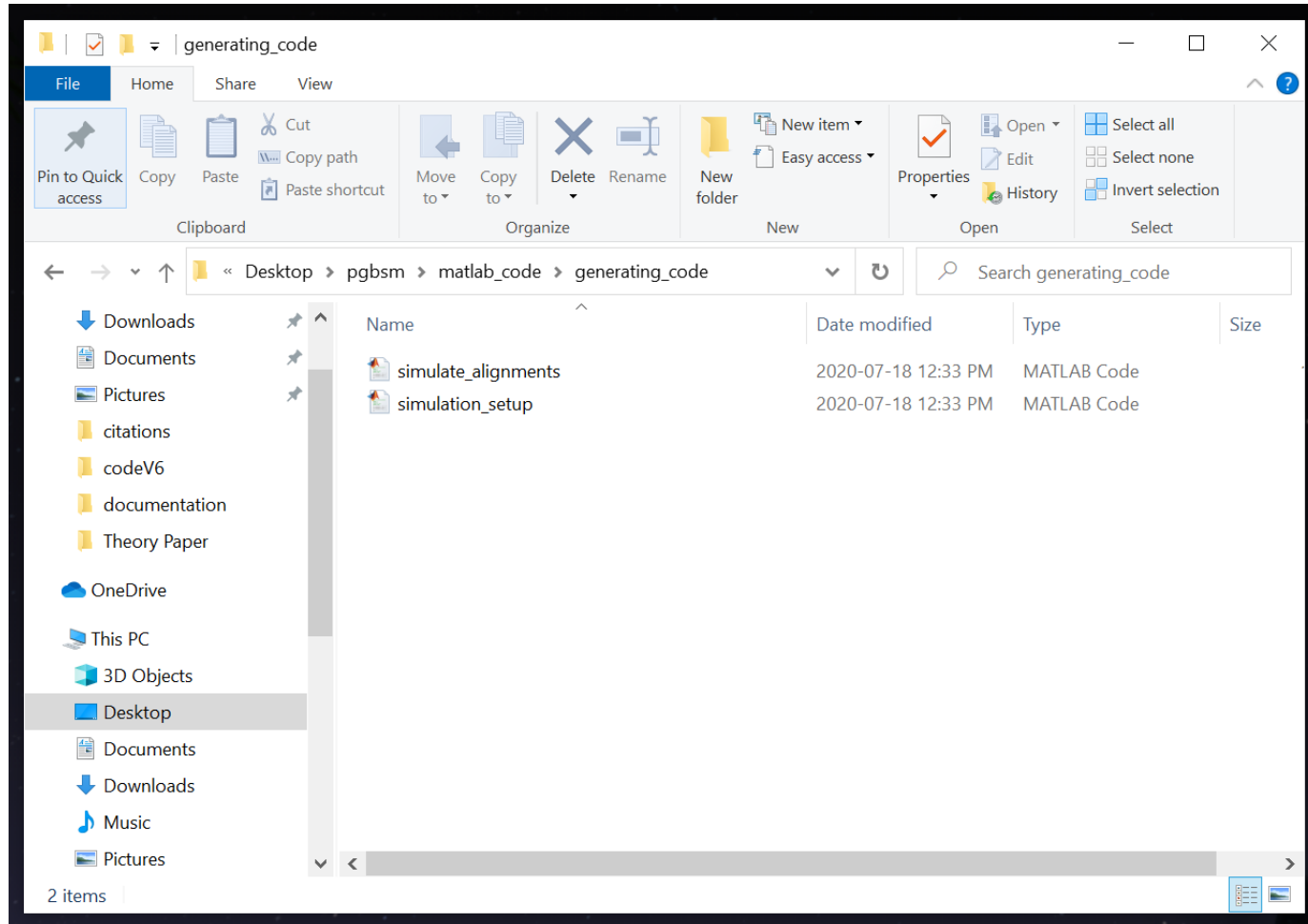
Command Window

```
Setup Data Saved
fx>>
```



# Generating Simulated Alignments

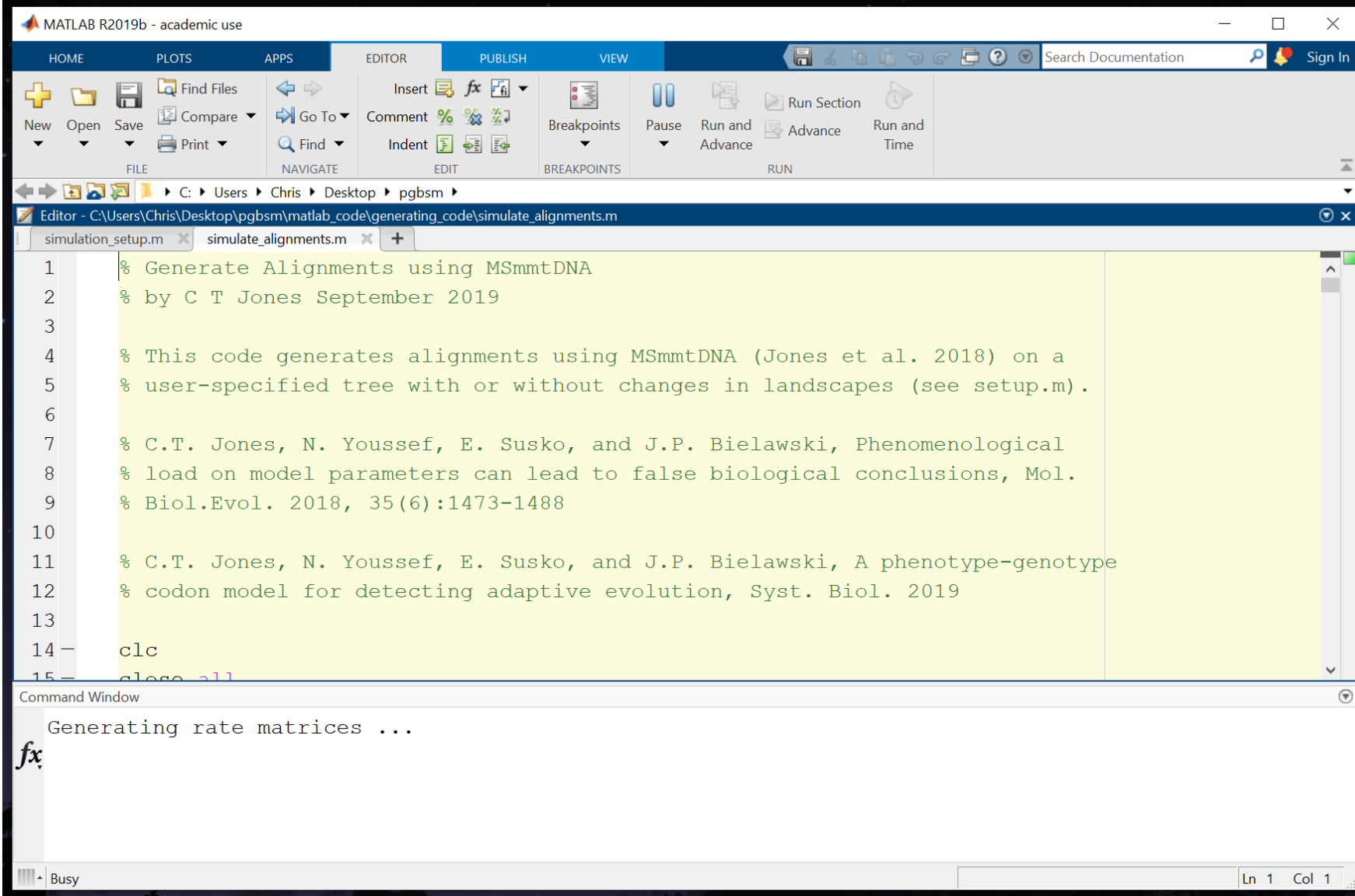
Now to generate your alignments open the script `simulate_alignments.m`.





# Generating Simulated Alignments

Press the big green triangle to run the script. The code will take some time to run so be patient.



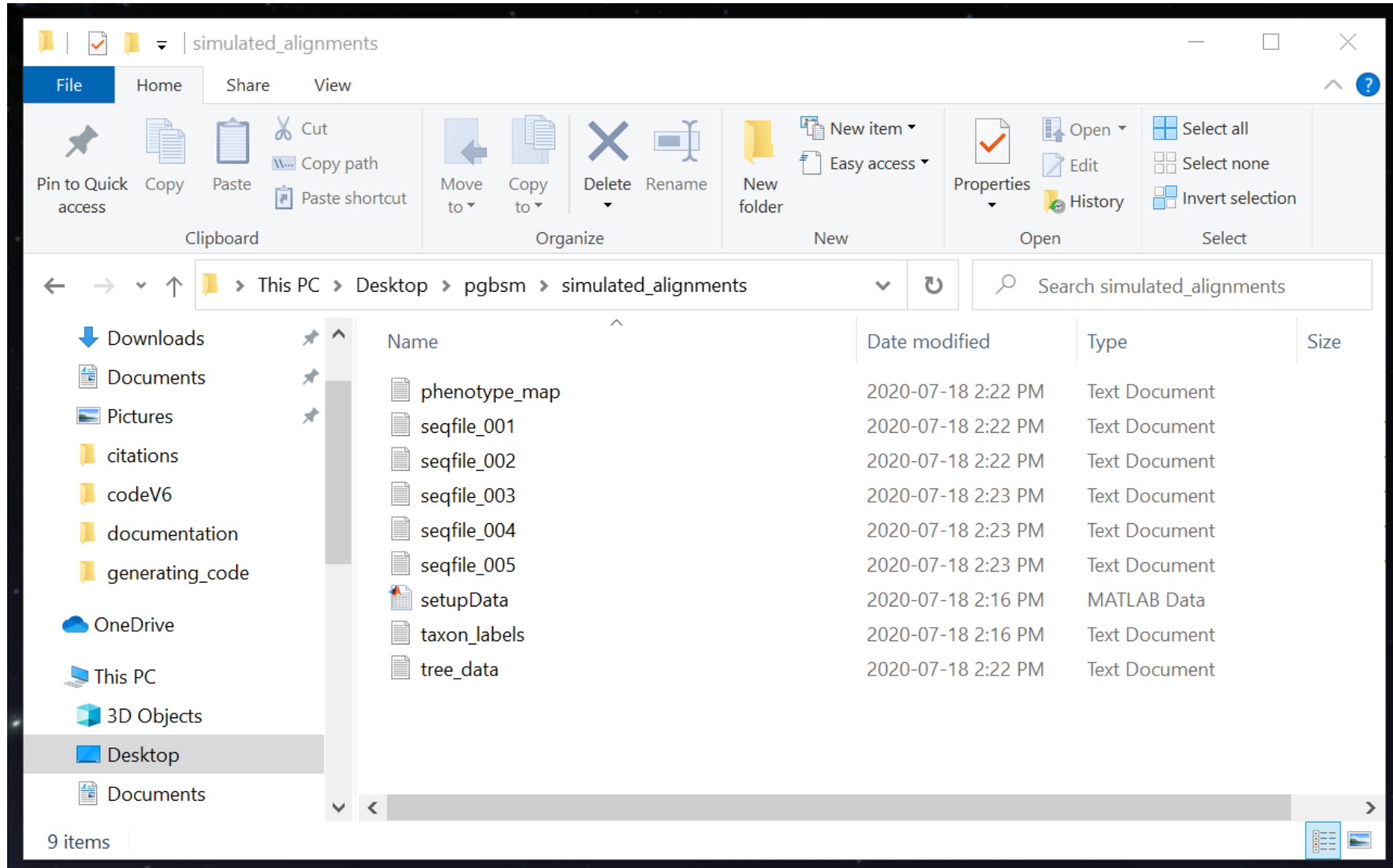
The image shows the MATLAB R2019b interface. The title bar reads "MATLAB R2019b - academic use". The ribbon includes tabs for HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The EDITOR tab is active, showing a toolbar with icons for file operations (New, Open, Save, Find Files, Compare, Print), navigation (Go To, Find), editing (Insert, Comment, Indent), breakpoints, and running (Pause, Run and Advance, Run Section, Advance, Run and Time). The current file is "simulate\_alignments.m" located at "C:\Users\Chris\Desktop\pgbsm\matlab\_code\generating\_code\simulate\_alignments.m". The script content is as follows:

```
1 % Generate Alignments using MSmmtDNA
2 % by C T Jones September 2019
3
4 % This code generates alignments using MSmmtDNA (Jones et al. 2018) on a
5 % user-specified tree with or without changes in landscapes (see setup.m).
6
7 % C.T. Jones, N. Youssef, E. Susko, and J.P. Bielawski, Phenomenological
8 % load on model parameters can lead to false biological conclusions, Mol.
9 % Biol.Evol. 2018, 35(6):1473-1488
10
11 % C.T. Jones, N. Youssef, E. Susko, and J.P. Bielawski, A phenotype-genotype
12 % codon model for detecting adaptive evolution, Syst. Biol. 2019
13
14 clc
15 close all
```

The Command Window at the bottom shows the text "Generating rate matrices ..." and a small "fx" icon. The status bar at the bottom indicates "Busy" and shows the cursor position as "Ln 1 Col 1".

# Generating Simulated Alignments

Once its done look in the simulated\_alignments folder. There you will find your sequences plus text files that are required for the script that fits the PG-BSM to the data. These include the phenotype\_map, taxon\_labels, and tree\_data.



# Generating Simulated Alignments

The text files should look like these:

```
seqfile_001 - Notepad
File Edit Format View Help
8 900

harpischord  GGC ACA TTC AAC GAC CTT GGC TAC AAC GAT ATC TTA ATC TCG TTA TTA CAT TCA ATA TCA CTC TAC ATT CAA TTA TTA CTT TCC
TTC GCA ACC CTC TTC ATT ATC ATA TTA ATC CAA GCG CTA TTC GCA ACA ACG ATA CTA CTC TCC ATA ACC ATA TAC GCC CTA GCA ACA GAC AAA

piano forte  GGC ACG TTC AAC GAC CTC GGC TAC AAC GAT ATC CTA ATC TCA TTA TTA CAT TCA ATA TCA CTC TAC ATC CAA CTA TTA CTT TCC
TTC GCA ACC CTA TTC ATT ATT ATG TTA ATC CAA GCC TTA TTC GCA ACA ACC ATA CTA TTA TCA ATA ACC ATA TAC GCA CTC CTC ATA GAC AAA

acoustic guitar  GAA ATT CGC CTC GAA TGA TGC CAC AAC ATA CTT ATA CTC TCT CTA CTA CGA CCA GTA CTC GTA TAT CTT TAC CTA CTT GAC
ATA TTC GCC ACA CTT CTT ATC ATT ATA GTT ATT CAA GCA CTA TTC GCC ACA ACA ATA CTA AAA TCA ATA ACG ATA TAC GCC TTG ATA ATA GAC

electric guitar  GGA ACG TTC AAT GAC CTA GGT TAC AAC GAC ATT TTA ATT TCT CTA CTA CAC TCA ATG TCA TTA TAC ATC CAA TTA CTC CTC
ATA TTC GCA ACA CTT TTC ATC ATT ATA CTA ATT CAA GCA CTA TTC GCC ACA ACC ATA TTA AAC TCA ATA ACA ATA TAC GCC TTA ATA ATA GAT

trumpet  ATC CTA ATC CTC GCC TCC GCA TCC AGC ACT TCC CAA ACA TCA CTA TTG TAC CCA GTA GTC CTA GTT GCA AAT TTC GTA CTT CCT ACC
GCC CTA TTA TTC ATC ATC ATA CAC ATC CAG GCT CTT CTC TCT ACA ACA ATA CTC TTA TCC ATT ACG ATA TAC GCG CTC ATA ATA GAC AAA ATA

trombone  ATC CCA ATC CTA GCC TCC ATC TCC AGC ACT TCC CAA ACC TCA CTA TCA TAC CCA GTA ATC CTA TTT GCA AGT CGC GTA TCT CCC CT
T GCC CTA TTA TTC ATC ATC ATA CAA ATT CAG GCT CTT TTC GCT ACA ACA ATA CTC TTA TCC ATA ACG ATA TAC GCA CTT ATA ATA GAT CAA AT

cornet  GGT ACC TTC AAT GAT CTC GGC TAT AAT GAC ATC CTG ATC TCC CTA CTA CAC TCA ATA TCA TTA TAC ATT CAA CTC TTA CTT TCC TTC
GCC CTA CTA TTT CTA ATC ATA CTT CTC CAA GCT CTA TTC GCC ACA ACA ATA CTC CTC TCA ATA ACA ATA TAC GCC CTC CTC ATA GAT AAC ATA

tuba  GGT ACC TTC AAC GAT CTC GGA TAT AAT GAC ATC CTG ATC TCC CTA CTA CAC TCA ATA TCA TTA TAC ATT CAA CTC TTA CTA TCC TTC TT
C CTG CTC TTT ATC ATT ATA CTT ATC CAA GCA CTA TTC TCT ACA ACA ATA CTC CTA TCA ATC ACA ATA TAT GCT CTC CTA ATG GAC AAA ATA TA
```

```
phenotype_map - Notepad
File Edit Format View Help
phenotypeMap = 1 1 2 1 3 3 1 1
```

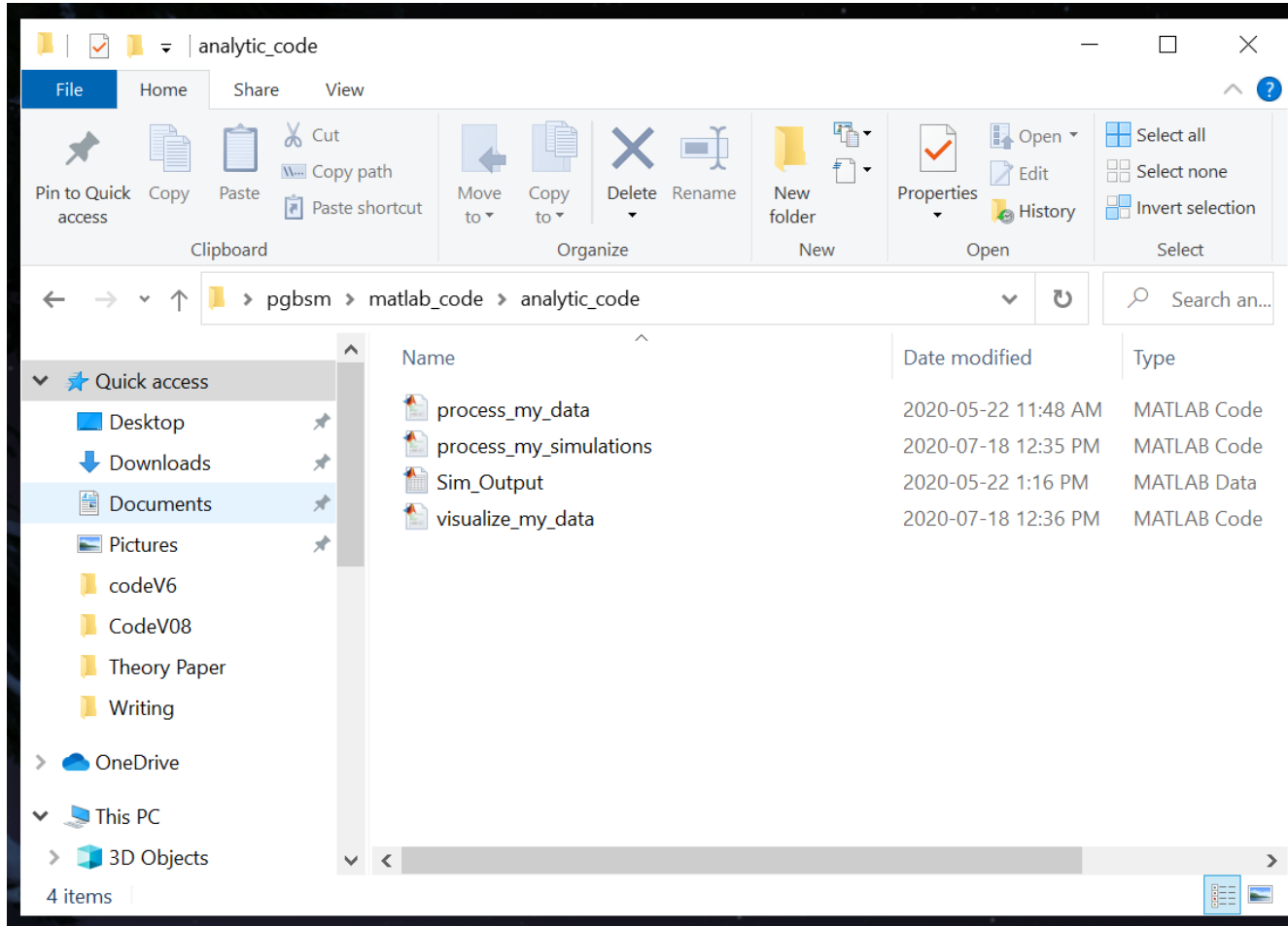
```
taxon_labels - Notepad
File Edit Format View Help
harpischord
piano forte
acoustic guitar
electric guitar
trumpet
trombone
cornet
tuba
```

```
tree_data - Notepad
File Edit Format View Help
|(((1:0.25,2:0.25):0.55,(3:0.30,4:0.25):0.60):1.75,((5:0.25,6:0.25):0.65,(7:0.25,8:0.50):0.50):1.75);
```

# Processing Simulated Alignments

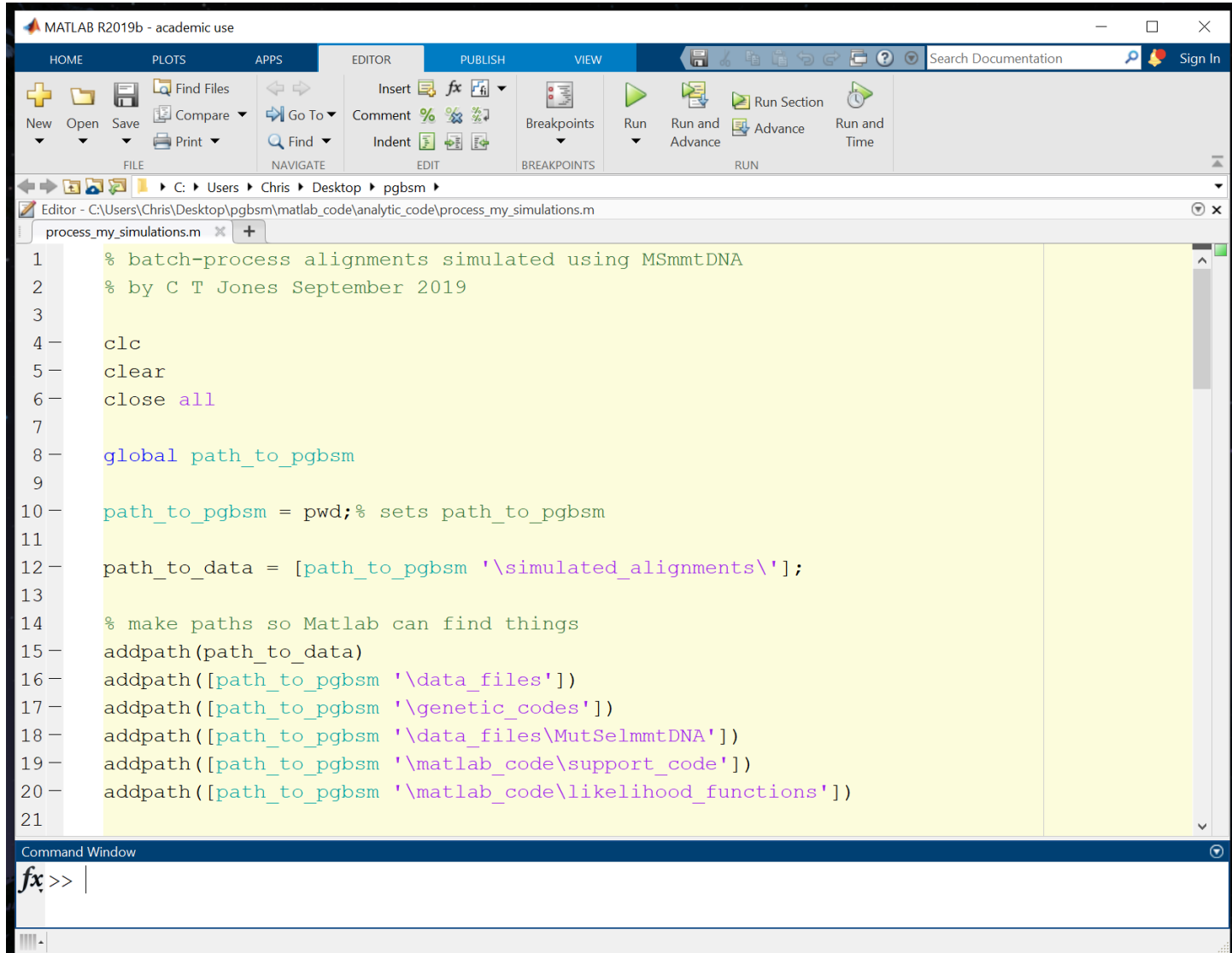
Now let us fit the simulated alignments to the models.

Go into your pgbsm folder and find the folder called analytic\_code.



# Processing Simulated Alignments

Right click on the Matlab script process\_my\_simulations.m to open it in your Matlab window. Press Run to get the code started. Note it can take a while for the code to complete, especially for larger alignments.

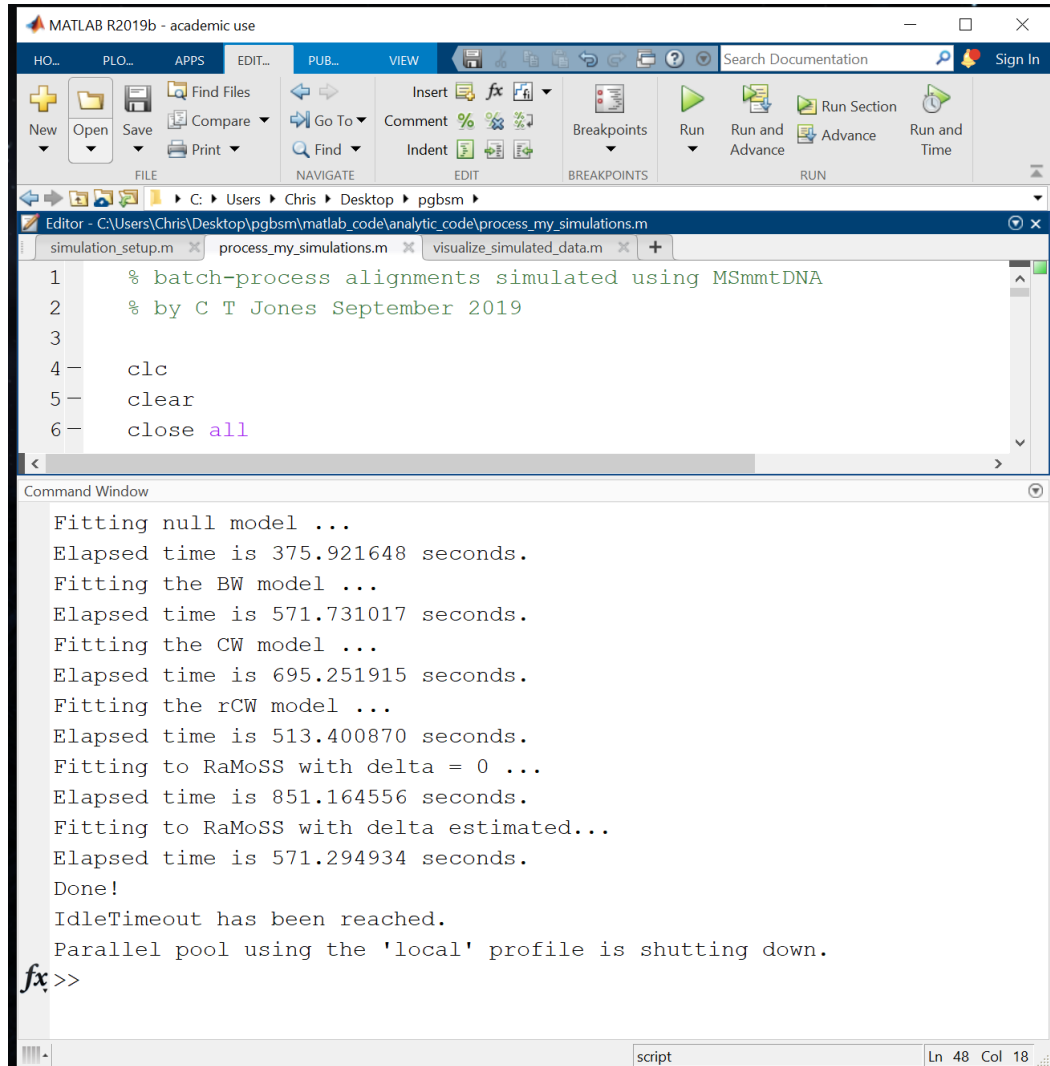


The image shows the MATLAB R2019b interface. The title bar reads "MATLAB R2019b - academic use". The ribbon includes tabs for HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The EDITOR tab is active, showing a toolbar with icons for file operations (New, Open, Save, Find Files, Compare, Print), navigation (Go To, Find), editing (Insert, Comment, Indent), breakpoints, and running (Run, Run and Advance, Run Section, Run and Time). The Command Window at the bottom shows the MATLAB prompt "fx >> |".

```
1  % batch-process alignments simulated using MSmmtdNA
2  % by C T Jones September 2019
3
4  clc
5  clear
6  close all
7
8  global path_to_pgbsm
9
10 path_to_pgbsm = pwd;% sets path_to_pgbsm
11
12 path_to_data = [path_to_pgbsm '\simulated_alignments\'];
13
14 % make paths so Matlab can find things
15 addpath(path_to_data)
16 addpath([path_to_pgbsm '\data_files'])
17 addpath([path_to_pgbsm '\genetic_codes'])
18 addpath([path_to_pgbsm '\data_files\MutSelmmtdNA'])
19 addpath([path_to_pgbsm '\matlab_code\support_code'])
20 addpath([path_to_pgbsm '\matlab_code\likelihood_functions'])
21
```

# Visualizing Your Results

Once the code has run its time to look at the results.



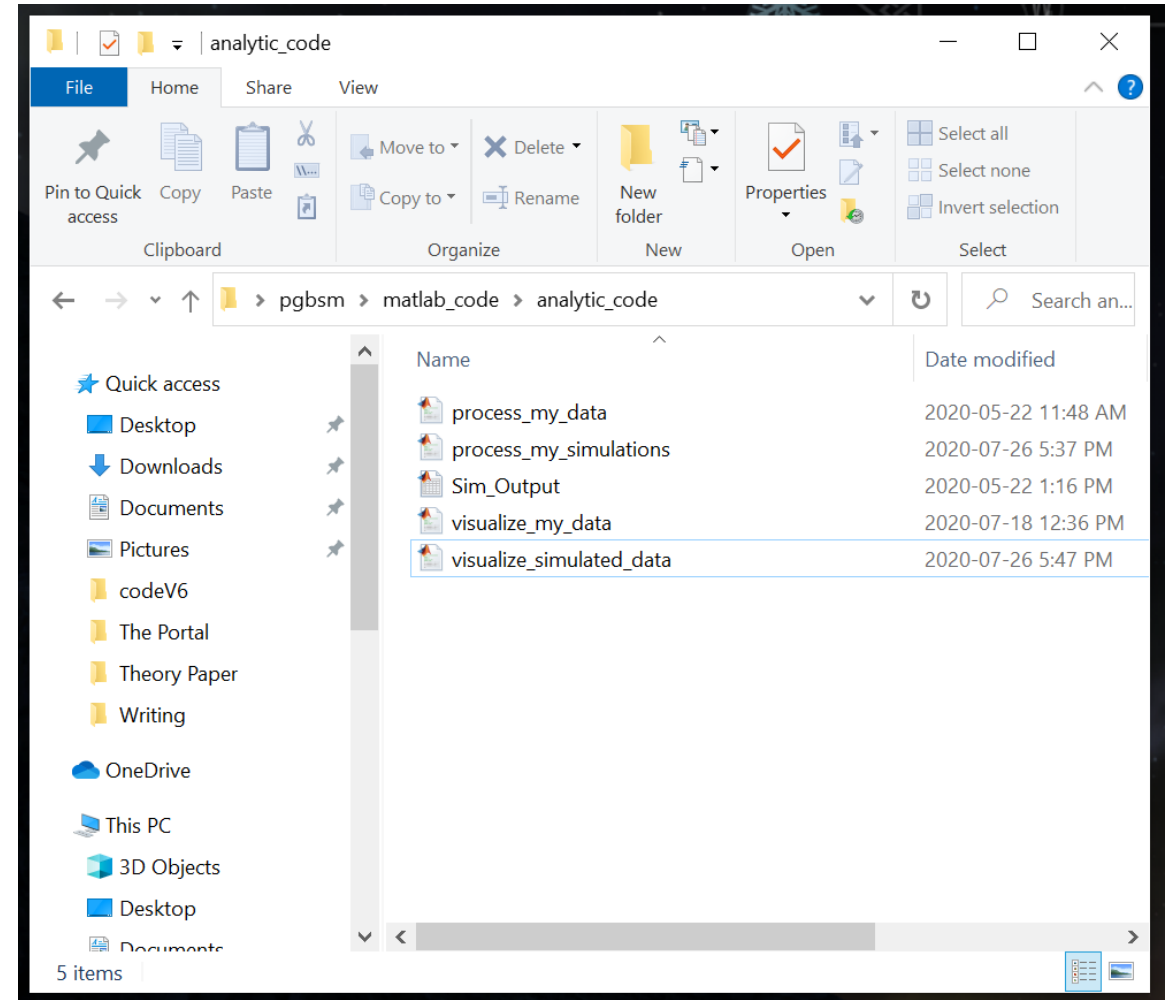
The MATLAB R2019b interface is shown with the Command Window displaying the following output:

```
Fitting null model ...  
Elapsed time is 375.921648 seconds.  
Fitting the BW model ...  
Elapsed time is 571.731017 seconds.  
Fitting the CW model ...  
Elapsed time is 695.251915 seconds.  
Fitting the rCW model ...  
Elapsed time is 513.400870 seconds.  
Fitting to RaMoSS with delta = 0 ...  
Elapsed time is 851.164556 seconds.  
Fitting to RaMoSS with delta estimated...  
Elapsed time is 571.294934 seconds.  
Done!  
IdleTimeout has been reached.  
Parallel pool using the 'local' profile is shutting down.  
fx>>
```

The Editor window shows the script `visualize_simulated_data.m` with the following code:

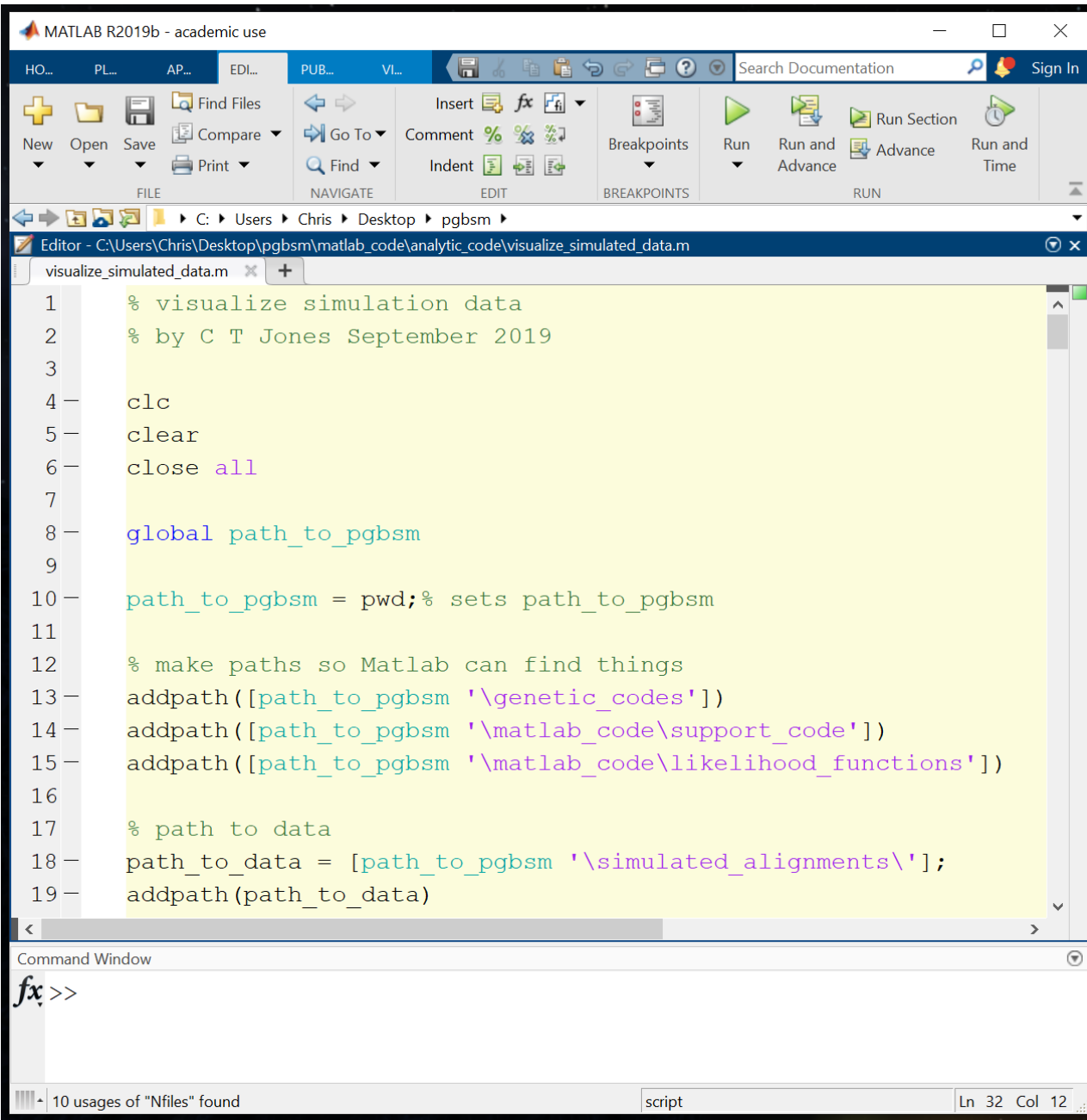
```
1 % batch-process alignments simulated using MSmmtDNA  
2 % by C T Jones September 2019  
3  
4 clc  
5 clear  
6 close all
```

Right click on the Matlab script `visualize_simulated_data.m` to open it in your Matlab window.



# Visualizing Your Results

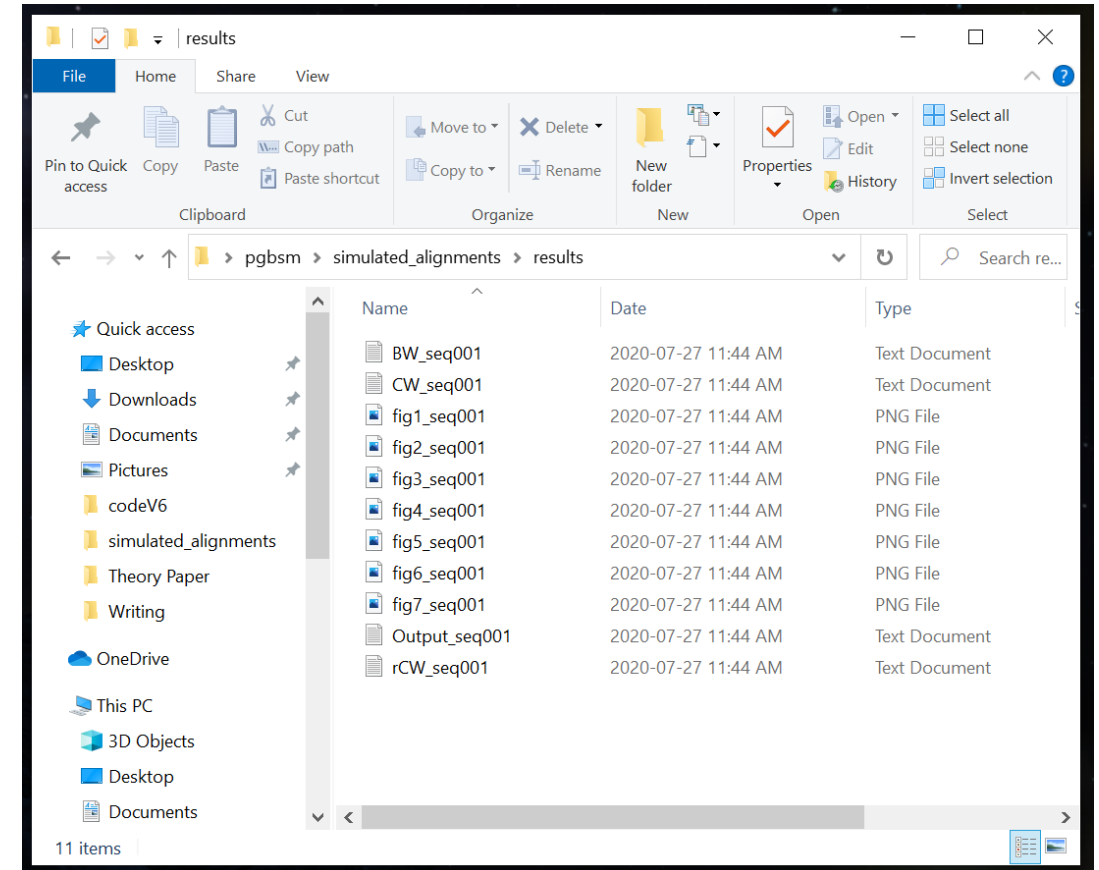
Click on run. The code will generate seven figures and four text files that will all appear in the results folder.



The image shows the MATLAB R2019b editor window. The title bar reads "MATLAB R2019b - academic use". The menu bar includes "HO...", "PL...", "AP...", "EDI...", "PUB...", and "VI...". The toolbar contains icons for "New", "Open", "Save", "Find Files", "Compare", "Print", "Go To", "Find", "Insert", "Comment", "Indent", "Breakpoints", "Run", "Run and Advance", "Run Section", "Advance", and "Run and Time". The current file is "visualize\_simulated\_data.m" located at "C:\Users\Chris\Desktop\pgbsm\matlab\_code\analytic\_code\visualize\_simulated\_data.m". The script content is as follows:

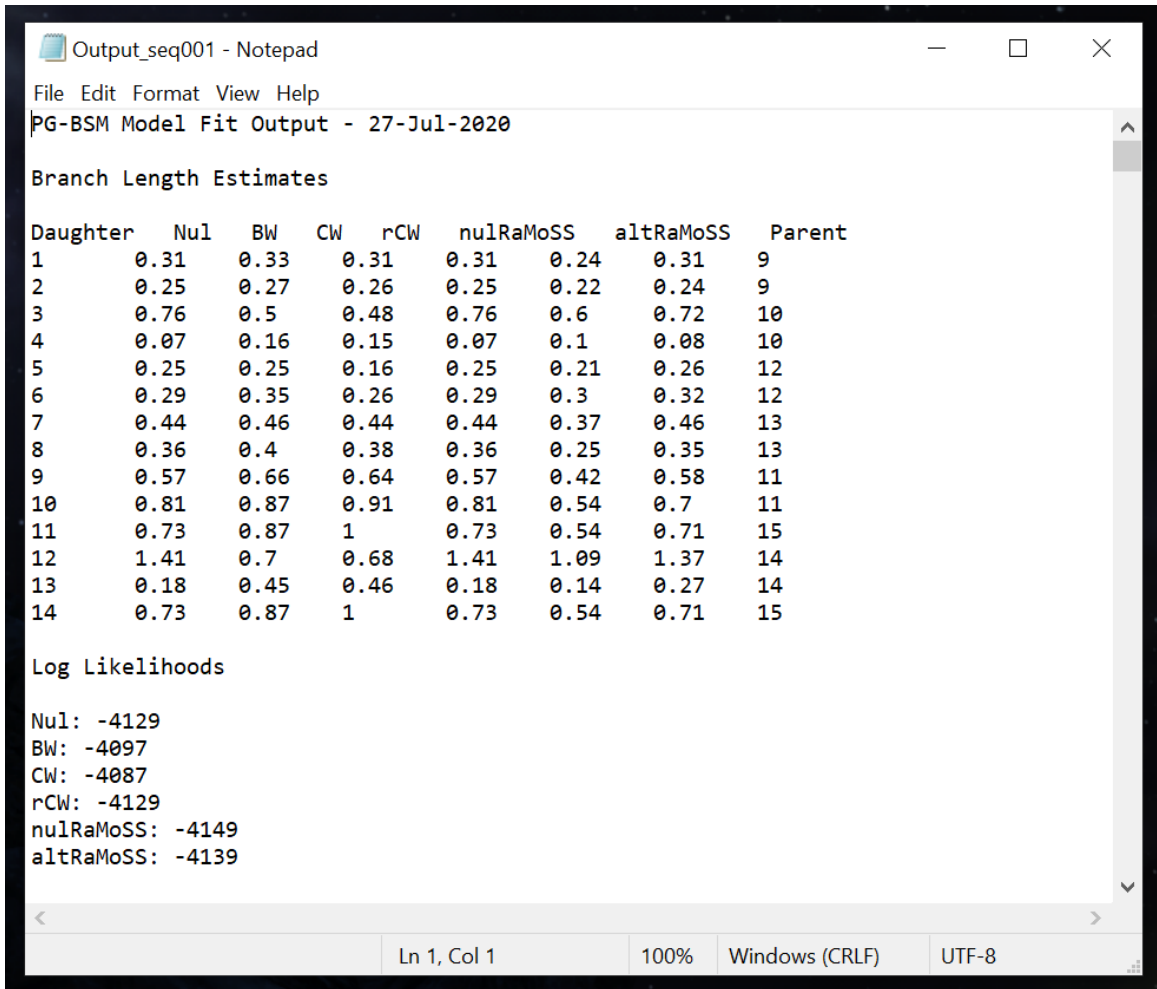
```
1 % visualize simulation data
2 % by C T Jones September 2019
3
4 clc
5 clear
6 close all
7
8 global path_to_pgbsm
9
10 path_to_pgbsm = pwd;% sets path_to_pgbsm
11
12 % make paths so Matlab can find things
13 addpath([path_to_pgbsm '\genetic_codes'])
14 addpath([path_to_pgbsm '\matlab_code\support_code'])
15 addpath([path_to_pgbsm '\matlab_code\likelihood_functions'])
16
17 % path to data
18 path_to_data = [path_to_pgbsm '\simulated_alignments\'];
19 addpath(path_to_data)
```

The Command Window at the bottom shows the prompt "fx >>". The status bar at the bottom indicates "10 usages of 'Nfiles' found" and "script" with line 32 and column 12.





# Interpreting Your Results



Output\_seq001 - Notepad

File Edit Format View Help

PG-BSM Model Fit Output - 27-Jul-2020

Branch Length Estimates

Daughter	Nul	BW	CW	rCW	nulRaMoSS	altRaMoSS	Parent
1	0.31	0.33	0.31	0.31	0.24	0.31	9
2	0.25	0.27	0.26	0.25	0.22	0.24	9
3	0.76	0.5	0.48	0.76	0.6	0.72	10
4	0.07	0.16	0.15	0.07	0.1	0.08	10
5	0.25	0.25	0.16	0.25	0.21	0.26	12
6	0.29	0.35	0.26	0.29	0.3	0.32	12
7	0.44	0.46	0.44	0.44	0.37	0.46	13
8	0.36	0.4	0.38	0.36	0.25	0.35	13
9	0.57	0.66	0.64	0.57	0.42	0.58	11
10	0.81	0.87	0.91	0.81	0.54	0.7	11
11	0.73	0.87	1	0.73	0.54	0.71	15
12	1.41	0.7	0.68	1.41	1.09	1.37	14
13	0.18	0.45	0.46	0.18	0.14	0.27	14
14	0.73	0.87	1	0.73	0.54	0.71	15

Log Likelihoods

Nul: -4129  
BW: -4097  
CW: -4087  
rCW: -4129  
nulRaMoSS: -4149  
altRaMoSS: -4139

Ln 1, Col 1 100% Windows (CRLF) UTF-8

The processing code fits each alignment to size model:

- 1) Nul = the null PG-BSM that assumes there are no phenotype-genotype associations
- 2) BW = the alternate PG-BSM testing for branch-wise sites
- 3) CW = the alternate PG-BSM testing for clade-wise sites
- 4) rCW = the alternate PG-BSM testing for reverse clade-wise sites
- 5) nulRaMoSS = assumes no heterotachy,  $\delta = 0$
- 6) altRaMoSS = allows for heterotachy,  $\delta > 0$  is estimated

RaMoSS tests for sites that evolve under a constant dN/dS ratio and sites that exhibit heterotachy (a random mixture of static and switching sites, see Jones et al. 2018 for details). The key parameter is the switching rate  $\delta$  which is set to zero under the null model (no heterotachy) and is estimated under the alternate model. Rejection of the null model means that heterotachy was detected.

# Interpreting Your Results

```
Output_seq001 - Notepad
File Edit Format View Help

Nu1: -4129
BW: -4097
CW: -4087
rCW: -4129
nulRaMoSS: -4149
altRaMoSS: -4139

Nu1 PG-BSM MLEs
pi0      w1      w2      p1      delta    kappa    lambda
0.57     0.08    2.78    0.82    0.24     3.29     0.52

BW PG-BSM MLEs
pi0      w1      w2      p1      delta    kappa    lambda    piBW
0.54     0.05    2.52    0.8     0.19     3.4      0.52     0.09

CW PG-BSM MLEs
pi0      w1      w2      p1      delta    kappa    lambda    piCW
0.48     0.03    1.66    0.79    0.11     3.35     0.51     0.11

rCW PG-BSM MLEs
pi0      w1      w2      p1      delta    kappa    lambda    pirCW
0.57     0.08    2.78    0.82    0.24     3.29     0.5      0

nulRaMoSS MLEs
piCL     w1M3    w2M3    p1M3    w1CL    w2CL    p1CL    delta    kappa
0.72     0.1     1.28    0.89    0       0.41    0.77    0       2.7

altRaMoSS MLEs
piCL     w1M3    w2M3    p1M3    w1CL    w2CL    p1CL    delta    kappa
0.77     0.18    0.85    0.73    0       5.19    0.96    0.06    3.25

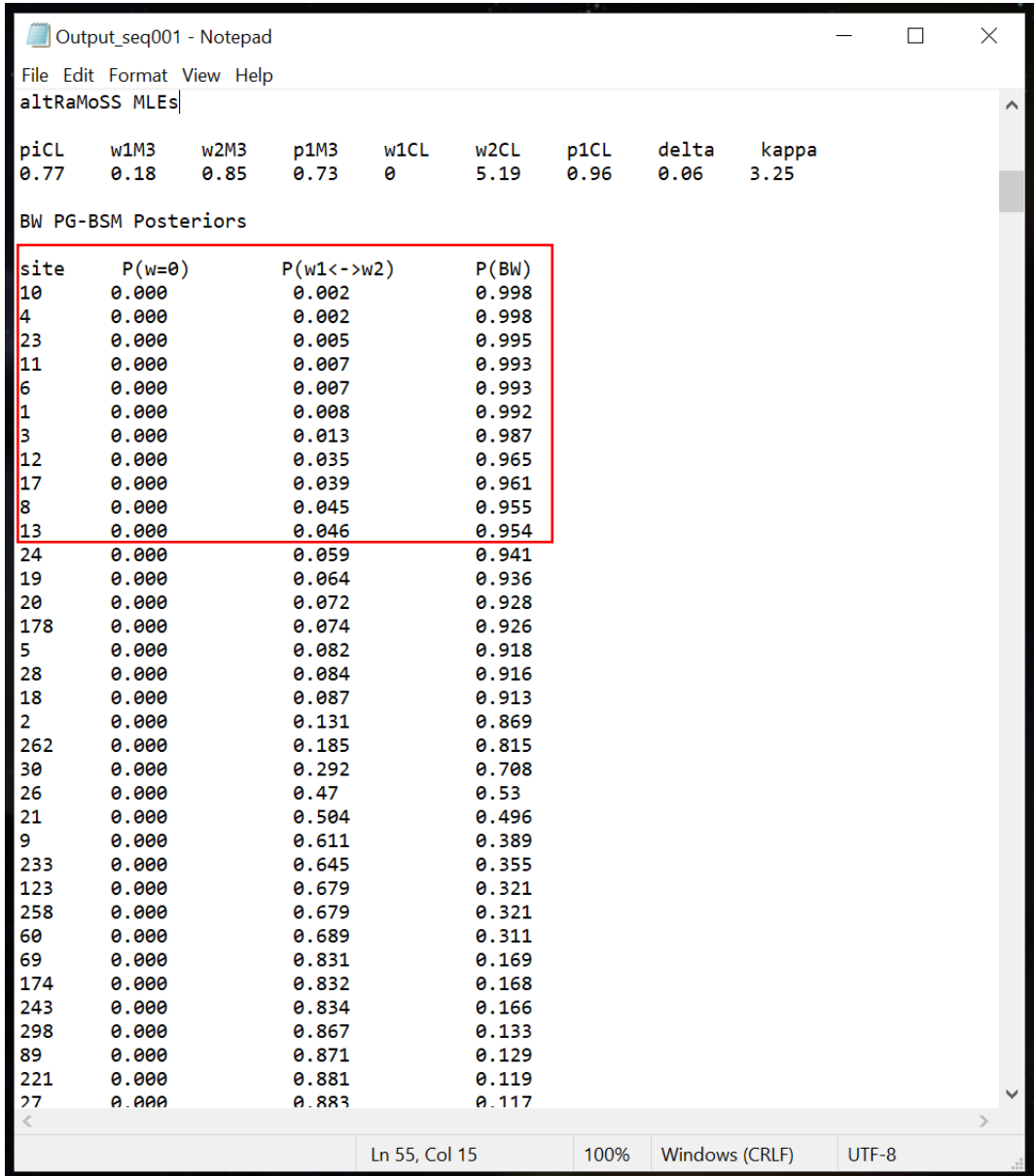
Ln 1, Col 1    100%    Windows (CRLF)    UTF-8
```

The alternate RaMoSS model fits better than its counterpart null model, indicating the detection of heterotachous sites. The estimated proportion of such sites is 77%.

Most of these sites are not associated with phenotype, meaning that their site patterns are not consistent with changes in dN/dS that correspond to changes in phenotype.

The PG-BSM indicates that 11% and 9% of sites exhibit heterotachy consistent with the CW and BW process, respectively. This illustrates how the PG-BSM can be used to identify from a set of site patterns that exhibit heterotachy-by-any-cause those with specific patterns of heterotachy consistent with changes in site-specific landscapes that are inferred to have co-occurred with changes in phenotype.

# Interpreting Your Results



Output\_seq001 - Notepad

File Edit Format View Help

altRaMoSS MLEs

piCL	w1M3	w2M3	p1M3	w1CL	w2CL	p1CL	delta	kappa
0.77	0.18	0.85	0.73	0	5.19	0.96	0.06	3.25

BW PG-BSM Posteriors

site	P(w=0)	P(w1<->w2)	P(BW)
10	0.000	0.002	0.998
4	0.000	0.002	0.998
23	0.000	0.005	0.995
11	0.000	0.007	0.993
6	0.000	0.007	0.993
1	0.000	0.008	0.992
3	0.000	0.013	0.987
12	0.000	0.035	0.965
17	0.000	0.039	0.961
8	0.000	0.045	0.955
13	0.000	0.046	0.954
24	0.000	0.059	0.941
19	0.000	0.064	0.936
20	0.000	0.072	0.928
178	0.000	0.074	0.926
5	0.000	0.082	0.918
28	0.000	0.084	0.916
18	0.000	0.087	0.913
2	0.000	0.131	0.869
262	0.000	0.185	0.815
30	0.000	0.292	0.708
26	0.000	0.47	0.53
21	0.000	0.504	0.496
9	0.000	0.611	0.389
233	0.000	0.645	0.355
123	0.000	0.679	0.321
258	0.000	0.679	0.321
60	0.000	0.689	0.311
69	0.000	0.831	0.169
174	0.000	0.832	0.168
243	0.000	0.834	0.166
298	0.000	0.867	0.133
89	0.000	0.871	0.129
221	0.000	0.881	0.119
27	0.000	0.883	0.117

Ln 55, Col 15 100% Windows (CRLF) UTF-8

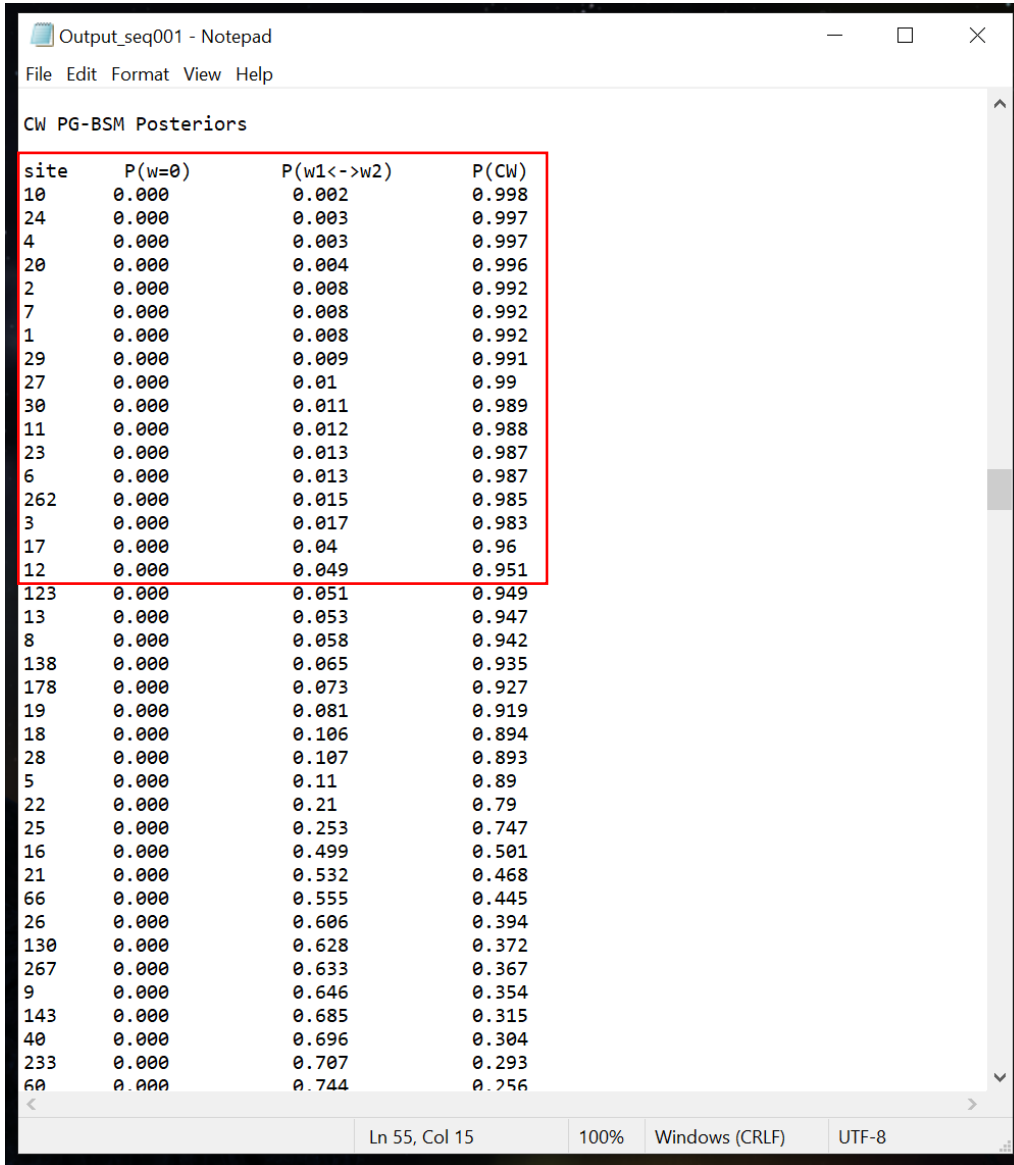
$P(w=0)$  is the posterior probability that the site is fixed.

$P(w1 \leftrightarrow w2)$  is the posterior probability that the site is heterotachous but not consistent with the BW process.

$P(BW)$  is the posterior that the site is consistent with the BW process.

In this case the data was generated with sites 1 to 15 under the BW process and 16 to 30 under the CW process. There are 11 sites with  $P(BW) > 0.95$ . All of these are among the first 30 sites and 9 are among the first 15. Hence, 9 BW sites were correctly detected assuming a posterior threshold of 0.95, with 2 false positives that were in fact CW sites.

# Interpreting Your Results



Output\_seq001 - Notepad

File Edit Format View Help

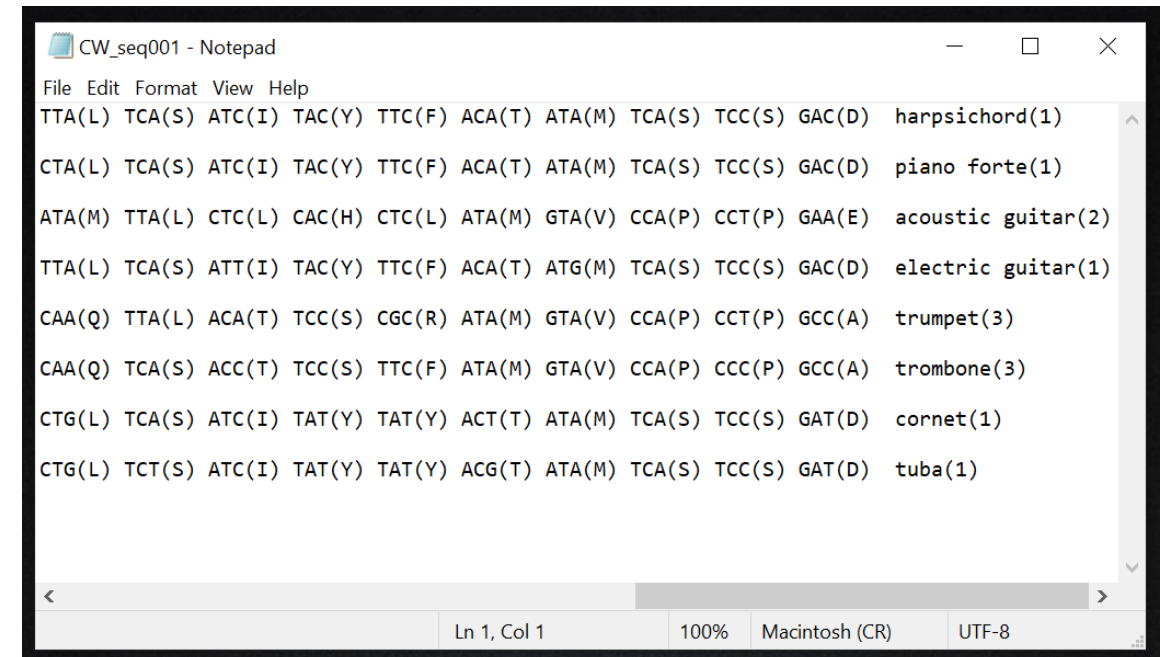
CW PG-BSM Posteriors

site	P(w=0)	P(w1<->w2)	P(CW)
10	0.000	0.002	0.998
24	0.000	0.003	0.997
4	0.000	0.003	0.997
20	0.000	0.004	0.996
2	0.000	0.008	0.992
7	0.000	0.008	0.992
1	0.000	0.008	0.992
29	0.000	0.009	0.991
27	0.000	0.01	0.99
30	0.000	0.011	0.989
11	0.000	0.012	0.988
23	0.000	0.013	0.987
6	0.000	0.013	0.987
262	0.000	0.015	0.985
3	0.000	0.017	0.983
17	0.000	0.04	0.96
12	0.000	0.049	0.951
123	0.000	0.051	0.949
13	0.000	0.053	0.947
8	0.000	0.058	0.942
138	0.000	0.065	0.935
178	0.000	0.073	0.927
19	0.000	0.081	0.919
18	0.000	0.106	0.894
28	0.000	0.107	0.893
5	0.000	0.11	0.89
22	0.000	0.21	0.79
25	0.000	0.253	0.747
16	0.000	0.499	0.501
21	0.000	0.532	0.468
66	0.000	0.555	0.445
26	0.000	0.606	0.394
130	0.000	0.628	0.372
267	0.000	0.633	0.367
9	0.000	0.646	0.354
143	0.000	0.685	0.315
40	0.000	0.696	0.304
233	0.000	0.707	0.293
60	0.000	0.744	0.256

Ln 55, Col 15 100% Windows (CRLF) UTF-8

Here there are 17 sites with  $P(CW) > 0.95$ , all of which are among the first 30 sites apart from site 262. Assuming a posterior threshold of 0.95, the model correctly identified 8 CW sites (sites from 16 to 30 with  $P(CW) > 0.95$ ) with 9 false positives, all but one of which is among the first 30 site patterns.

The site patterns identified by the method used to control the false discover rate (Jones et al.2020) are listed in the files labeled BW\_seqxxx, CW\_seqxxx, and rCW\_seqxxx.



CW\_seq001 - Notepad

File Edit Format View Help

TTA(L) TCA(S) ATC(I) TAC(Y) TTC(F) ACA(T) ATA(M) TCA(S) TCC(S) GAC(D) harpsichord(1)

CTA(L) TCA(S) ATC(I) TAC(Y) TTC(F) ACA(T) ATA(M) TCA(S) TCC(S) GAC(D) piano forte(1)

ATA(M) TTA(L) CTC(L) CAC(H) CTC(L) ATA(M) GTA(V) CCA(P) CCT(P) GAA(E) acoustic guitar(2)

TTA(L) TCA(S) ATT(I) TAC(Y) TTC(F) ACA(T) ATG(M) TCA(S) TCC(S) GAC(D) electric guitar(1)

CAA(Q) TTA(L) ACA(T) TCC(S) CGC(R) ATA(M) GTA(V) CCA(P) CCT(P) GCC(A) trumpet(3)

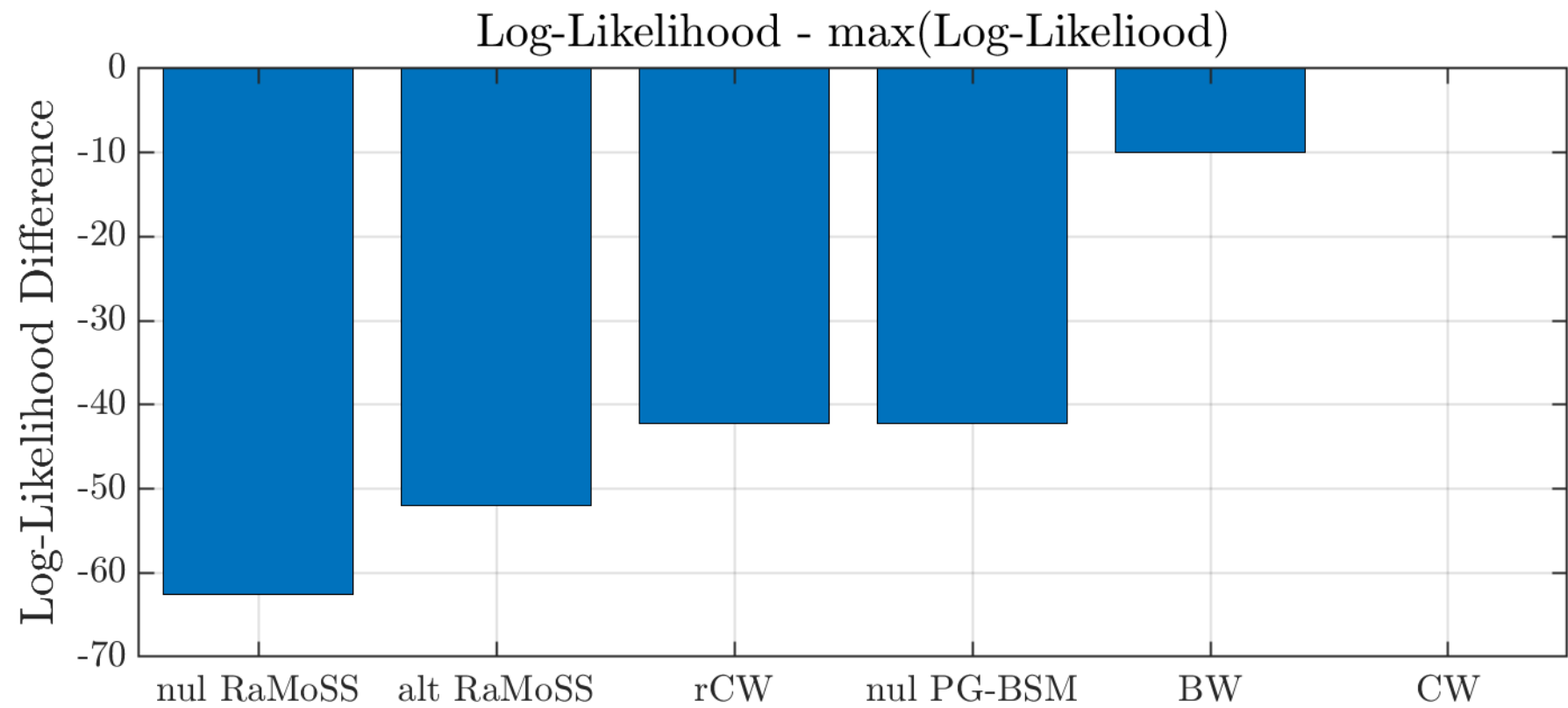
CAA(Q) TCA(S) ACC(T) TCC(S) TTC(F) ATA(M) GTA(V) CCA(P) CCC(P) GCC(A) trombone(3)

CTG(L) TCA(S) ATC(I) TAT(Y) TAT(Y) ACT(T) ATA(M) TCA(S) TCC(S) GAT(D) cornet(1)

CTG(L) TCT(S) ATC(I) TAT(Y) TAT(Y) ACG(T) ATA(M) TCA(S) TCC(S) GAT(D) tuba(1)

Ln 1, Col 1 100% Macintosh (CR) UTF-8

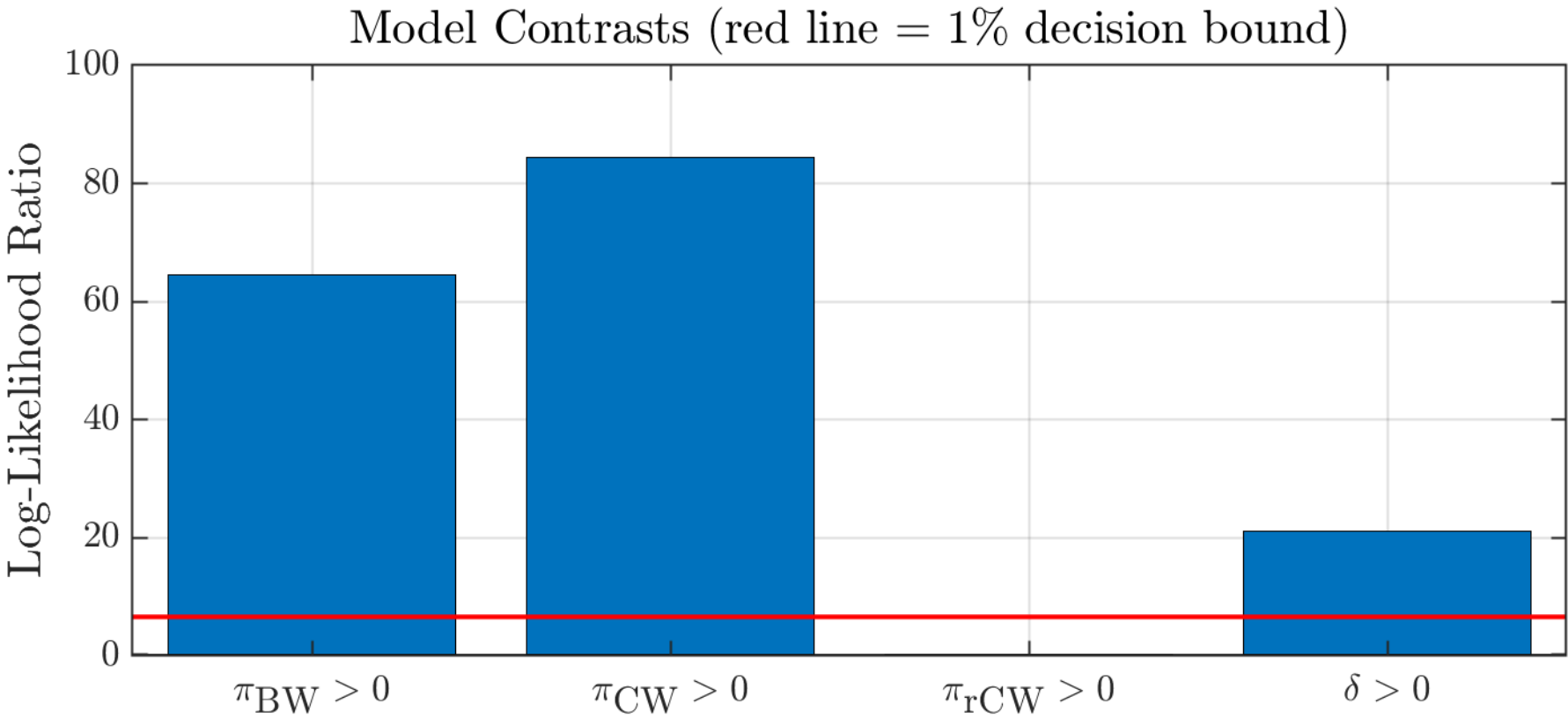
# Interpreting Your Results



- BW\_seq001
- CW\_seq001
- fig1\_seq001
- fig2\_seq001
- fig3\_seq001
- fig4\_seq001
- fig5\_seq001
- fig6\_seq001
- fig7\_seq001
- Output\_seq001
- rCW\_seq001

Results includes 7 figures. Figure 1 illustrates the difference in log-likelihood for each model compared to the best fitting model. These are shown in order from worst to best fit. Here the null RaMoSS model provided the worst fit and the PG-BSM with the CW process provided the best fit.

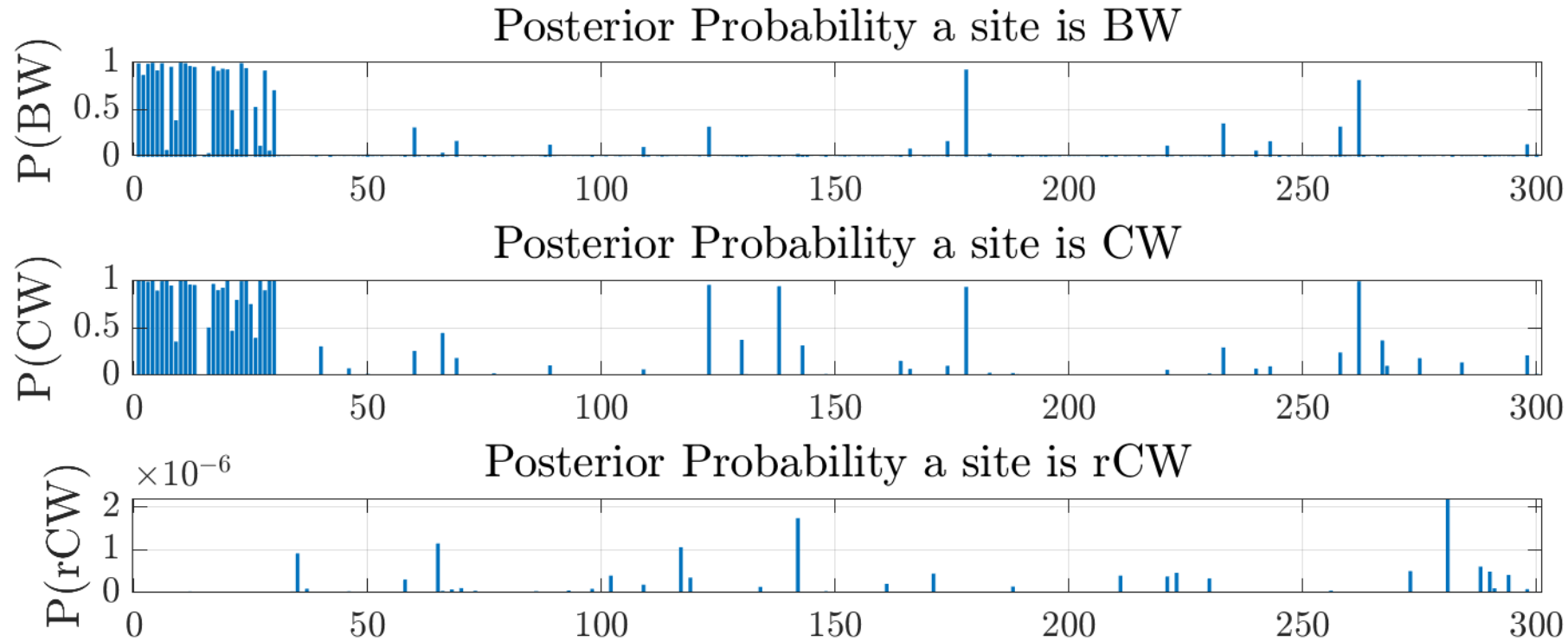
# Interpreting Your Results



- BW\_seq001
- CW\_seq001
- fig1\_seq001
- fig2\_seq001**
- fig3\_seq001
- fig4\_seq001
- fig5\_seq001
- fig6\_seq001
- fig7\_seq001
- Output\_seq001
- rCW\_seq001

Figure 2 shows the log-likelihood for the contrast between the null PG-BSM and the PG-BSM with the BW, CW and rCW processes and between the null RaMoSS and alternate RaMoSS. Values above the 1% decision bound indicate that the null should be rejected.

# Interpreting Your Results

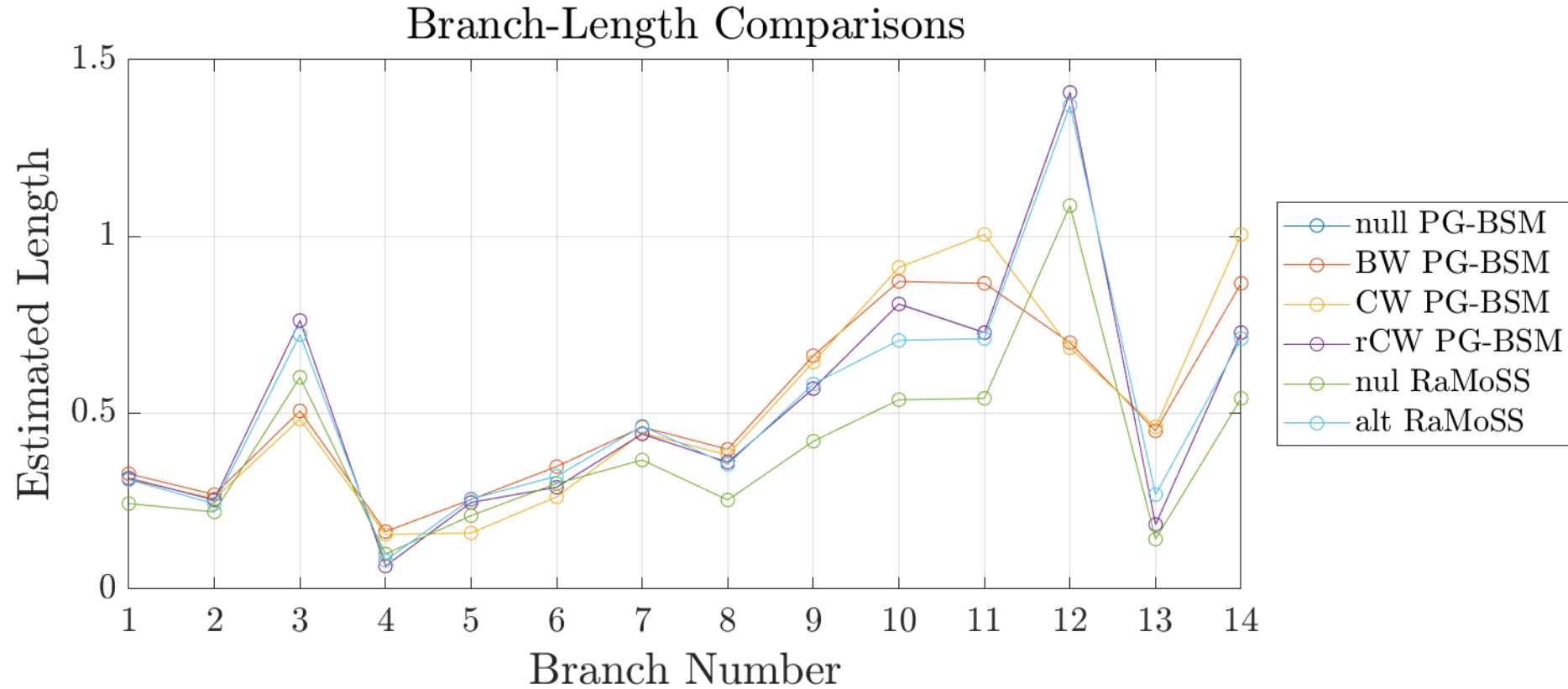


- BW\_seq001
- CW\_seq001
- fig1\_seq001
- fig2\_seq001
- fig3\_seq001**
- fig4\_seq001
- fig5\_seq001
- fig6\_seq001
- fig7\_seq001
- Output\_seq001
- rCW\_seq001

Figure 3 shows site-specific posterior probabilities. Probabilities closer to one indicates sites whose patterns are more consistent with the respective process, BW, CW, or rCW. Note that the simulation code places all such sites at on left site of the sequence. Here for example sites 1 to 15 were evolved under the BW process and sites 16 to 30 under the CW process. In this case the results suggest that the model cannot easily distinguish between the two.



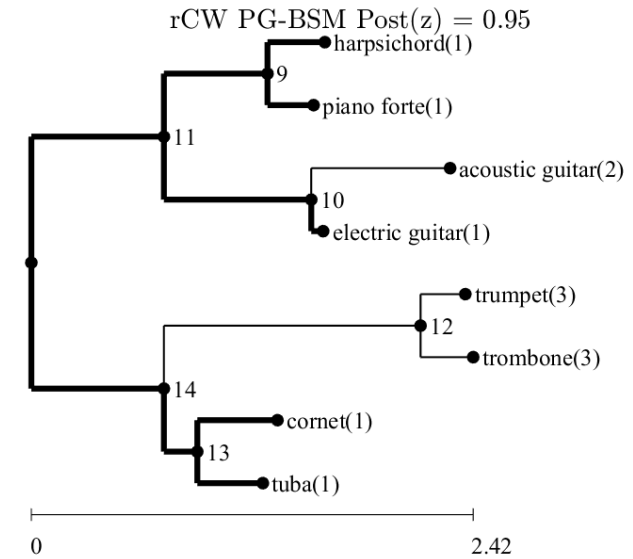
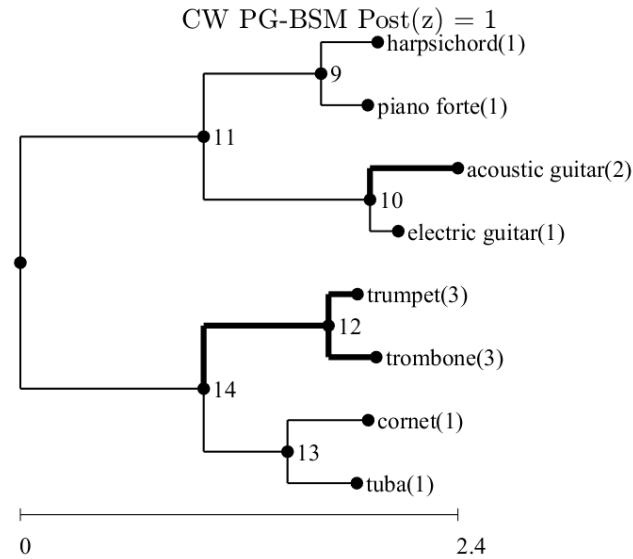
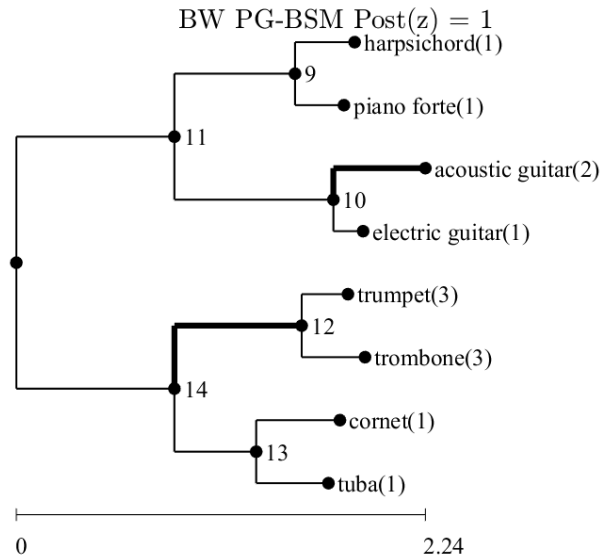
# Interpreting Your Results



- BW\_seq001
- CW\_seq001
- fig1\_seq001
- fig2\_seq001
- fig3\_seq001
- fig4\_seq001**
- fig5\_seq001
- fig6\_seq001
- fig7\_seq001
- Output\_seq001
- rCW\_seq001

Figure 4 compares branch-length estimates. Non-stationarity (i.e., changes in some site-specific landscapes) is indicated when the estimate of a branch length is substantially smaller under one of the alternate PG-BSM models than it is under the other models (e.g., branch 12).

# Interpreting Your Results



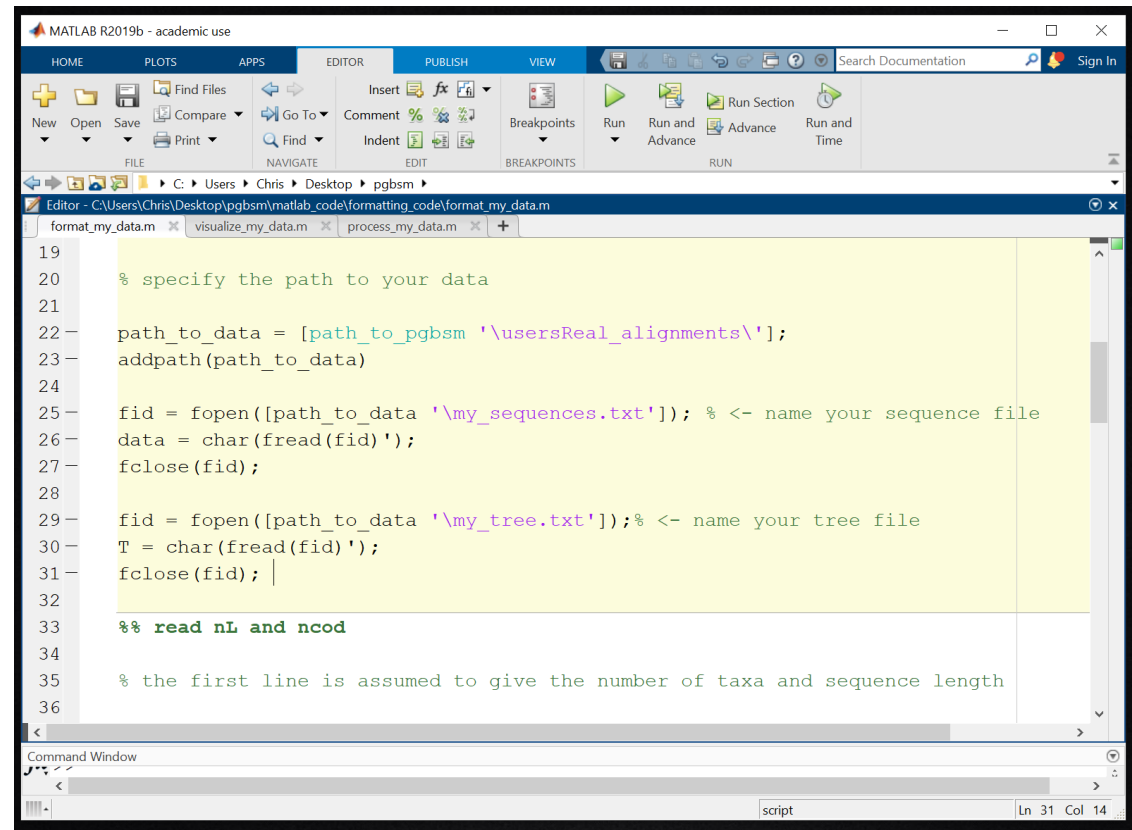
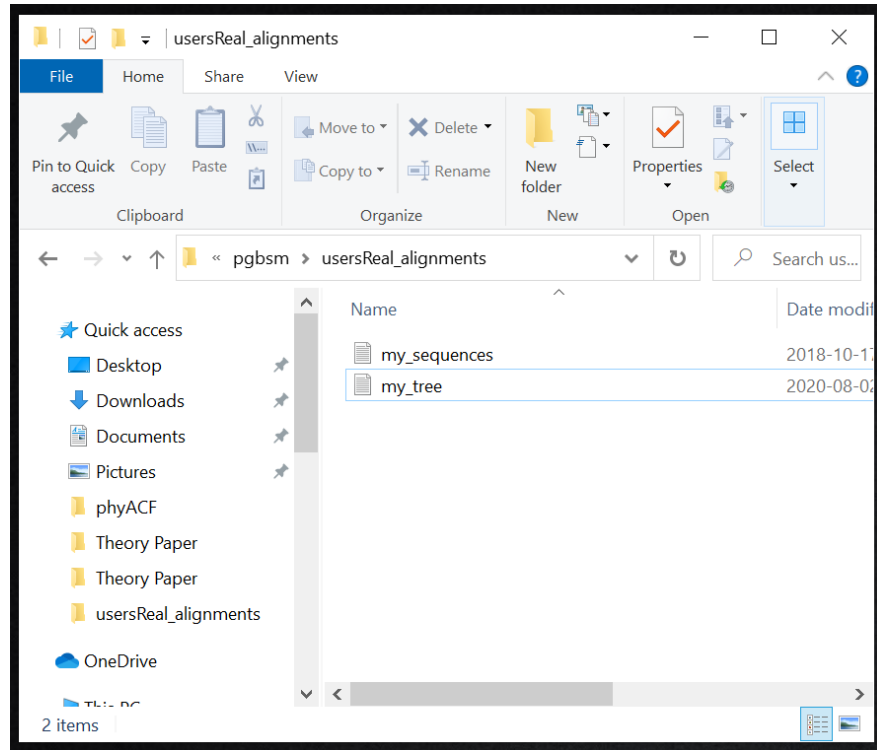
- BW\_seq001
- CW\_seq001
- fig1\_seq001
- fig2\_seq001
- fig3\_seq001
- fig4\_seq001
- fig5\_seq001
- fig6\_seq001
- fig7\_seq001
- Output\_seq001
- rCW\_seq001

Figures 5 to 7 illustrate the branches over which the phenotype is most likely to have changed (i.e., the change map  $z$ ). The posterior probability that  $z$  is the correct pattern of change is given by  $\text{Post}(z)$  (see Jones et al. 2020 for details). In this case all three versions of the alternate PG-BSM correctly identified branches 3 and 12 as those over which the phenotype changed in coordination with changes in some site-specific landscapes.

# Processing your own Data

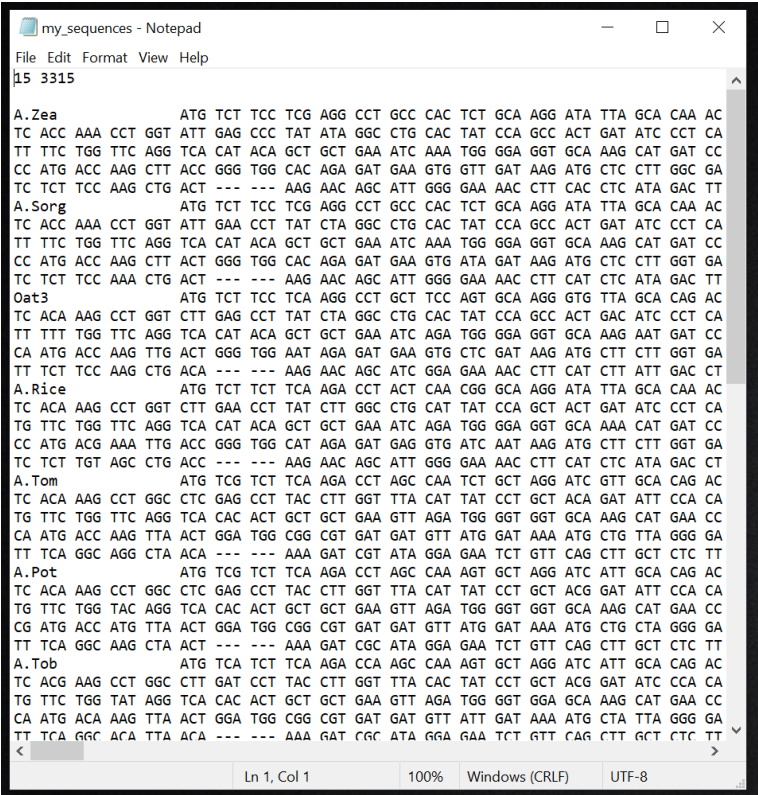
This section shows how to adjust your own data to conform to the format assumed by the PG-BSM code.

First place the data file containing your sequences and a file containing your tree into the director usersReal\_sequences. In this example the sequence file is called my\_sequences.txt and the tree file my\_tree.txt. You can name these anything you like, but the same names must appear in the Matlab script format\_my\_data.m as shown on below on the right.



# Processing your own Data

In this case the alignment consists of 15 angiosperm phytochrome sequences. (Yang and Nielsen, 2002; Zhang et al., 2005). Note that the code will only work with a rooted tree. The tree must also include branch lengths that serve as initial estimates for the optimization. If none are available, then insert dummy branch lengths as shown below.



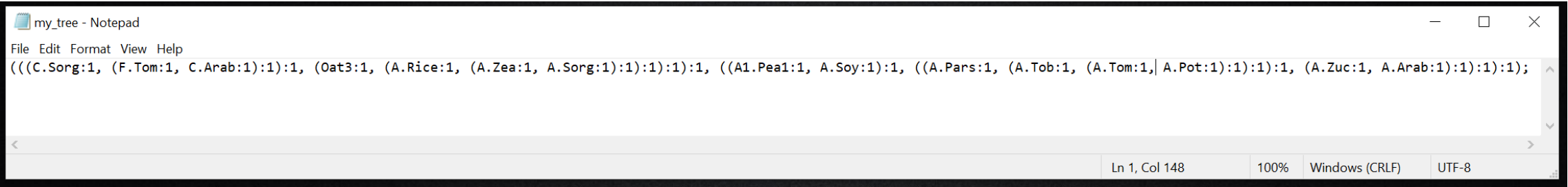
```
my_sequences - Notepad
File Edit Format View Help
Ln 1, Col 1 100% Windows (CRLF) UTF-8

A.Zea      ATG TCT TCC TCG AGG CCT GCC CAC TCT GCA AGG ATA TTA GCA CAA AC
TC ACC AAA CCT GGT ATT GAG CCC TAT ATA GGC CTG CAC TAT CCA GCC ACT GAT ATC CCT CA
TT TTC TGG TTC AGG TCA CAT ACA GCT GCT GAA ATC AAA TGG GGA GGT GCA AAG CAT GAT CC
CC ATG ACC AAG CTT ACC GGG TGG CAC AGA GAT GAA GTG GTT GAT AAG ATG CTC CTT GGC GA
TC TCT TCC AAG CTG ACT --- --- AAG AAC AGC ATT GGG GAA AAC CTT CAC CTC ATA GAC TT
A.Song     ATG TCT TCC TCG AGG CCT GCC CAC TCT GCA AGG ATA TTA GCA CAA AC
TC ACC AAA CCT GGT ATT GAA CCT TAT CTA GGC CTG CAC TAT CCA GCC ACT GAT ATC CCT CA
TT TTC TGG TTC AGG TCA CAT ACA GCT GCT GAA ATC AAA TGG GGA GGT GCA AAG CAT GAT CC
CC ATG ACC AAG CTT ACT GGG TGG CAC AGA GAT GAA GTG ATA GAT AAG ATG CTC CTT GGT GA
TC TCT TCC AAG CTG ACT --- --- AAG AAC AGC ATT GGG GAA AAC CTT CAT CTC ATA GAC TT
Oat3       ATG TCT TCC TCA AGG CCT GCT TCC AGT GCA AGG GTG TTA GCA CAG AC
TC ACA AAG CCT GGT CTT GAG CCT TAT CTA GGC CTG CAC TAT CCA GCC ACT GAC ATC CCT CA
TT TTT TGG TTC AGG TCA CAT ACA GCT GCT GAA ATC AGA TGG GGA GGT GCA AAG AAT GAT CC
CA ATG ACC AAG TTG ACT GGG TGG AAT AGA GAT GAA GTG CTC GAT AAG ATG CTT CTT GGT GA
TT TCT TCC AAG CTG ACA --- --- AAG AAC AGC ATC GGA GAA AAC CTT CAT CTT ATT GAC CT
A.Rice     ATG TCT TCT TCA AGA CCT ACT CAA CGG GCA AGG ATA TTA GCA CAA AC
TC ACA AAG CCT GGT CTT GAA CCT TAT CTT GGC CTG CAT TAT CCA GCT ACT GAT ATC CCT CA
TG TTC TGG TTC AGG TCA CAT ACA GCT GCT GAA ATC AGA TGG GGA GGT GCA AAA CAT GAT CC
CC ATG ACG AAA TTG ACC GGG TGG CAT AGA GAT GAG GTG ATC AAT AAG ATG CTT CTT GGT GA
TC TCT TGT AGC CTG ACC --- --- AAG AAC AGC ATT GGG GAA AAC CTT CAT CTC ATA GAC CT
A.Tom      ATG TCG TCT TCA AGA CCT AGC CAA TCT GCT AGG ATC GTT GCA CAG AC
TC ACA AAG CCT GGC CTC GAG CCT TAC CTT GGT TTA CAT TAT CCT GCT ACA GAT ATT CCA CA
TG TTC TGG TTC AGG TCA CAC ACT GCT GCT GAA GTT AGA TGG GGT GGT GCA AAG CAT GAA CC
CA ATG ACC AAG TTA ACT GGA TGG CGG CGT GAT GAT GTT ATG GAT AAA ATG CTG TTA GGG GA
TT TCA GGC AGG CTA ACA --- --- AAA GAT CGT ATA GGA GAA TCT GTT CAG CTT GCT CTC TT
A.Pot      ATG TCG TCT TCA AGA CCT AGC CAA AGT GCT AGG ATC ATT GCA CAG AC
TC ACA AAG CCT GGC CTC GAG CCT TAC CTT GGT TTA CAT TAT CCT GCT ACG GAT ATT CCA CA
TG TTC TGG TAC AGG TCA CAC ACT GCT GCT GAA GTT AGA TGG GGT GGT GCA AAG CAT GAA CC
CG ATG ACC ATG TTA ACT GGA TGG CGG CGT GAT GAT GTT ATG GAT AAA ATG CTG CTA GGG GA
TT TCA GGC AAG CTA ACT --- --- AAA GAT CGC ATA GGA GAA TCT GTT CAG CTT GCT CTC TT
A.Tob      ATG TCA TCT TCA AGA CCA AGC CAA AGT GCT AGG ATC ATT GCA CAG AC
TC ACG AAG CCT GGC CTT GAT CCT TAC CTT GGT TTA CAC TAT CCT GCT ACG GAT ATC CCA CA
TG TTC TGG TAT AGG TCA CAC ACT GCT GCT GAA GTT AGA TGG GGT GGA GCA AAG CAT GAA CC
CA ATG ACA AAG TTA ACT GGA TGG CGG CGT GAT GAT GTT ATT GAT AAA ATG CTA TTA GGG GA
TT TCA GGC ACA TTA ACA --- --- AAA GAT CGC ATA GGA GAA TCT GTT CAG CTT GCT CTC TT
```

Yang, Z. H. and Nielsen, R. 2002. Codon-substitution models for detecting molecular adaptation at individual sites along specific lineages. *Mol. Biol. Evol.*, 19: 908–917.

Zhang, J., Nielsen, R., and Yang, Z. H. 2005. Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. *Mol. Biol. Evol.*, 22: 2472–2479.

Note that missing codons must be indicated by “---” as shown here on the left. When the data is processed, any site pattern with at least one missing codon will be removed from the alignment.



```
my_tree - Notepad
File Edit Format View Help
Ln 1, Col 148 100% Windows (CRLF) UTF-8

(((C.Song:1, (F.Tom:1, C.Arab:1):1):1, (Oat3:1, (A.Rice:1, (A.Zea:1, A.Song:1):1):1):1, ((A1.Pea:1, A.Soy:1):1, ((A.Pars:1, (A.Tob:1, (A.Tom:1, A.Pot:1):1):1):1, (A.Zuc:1, A.Arab:1):1):1):1);
```

# Processing your own Data

The PG-BSM code assumes that sequences are listed in the same order that they appear in the tree. In this case the first sequence A.Zea does not match the first taxon C.Sorg in the tree file. If the sequence file and tree file are correctly formatted (i.e., they look like what is shown here), then you can run `format_my_data.m` to arrange the sequences in proper order.

my\_sequences - Notepad

File Edit Format View Help

15 3315

A.Zea

ATG TCT TCC TCG AGG CCT GCC CAC TCT GCA AGG ATA TTA GCA CAA AC

TC ACC AAA CCT GGT ATT GAG CCC TAT ATA GGC CTG CAC TAT CCA GCC ACT GAT ATC CCT CA

TT TTC TGG TTC AGG TCA CAT ACA GCT GCT GAA ATC AAA TGG GGA GGT GCA AAG CAT GAT CC

CC ATG ACC AAG CTT ACC GGG TGG CAC AGA GAT GAA GTG GTT GAT AAG ATG CTC CTT GGC GA

TC TCT TCC AAG CTG ACT --- --- AAG AAC AGC ATT GGG GAA AAC CTT CAC CTC ATA GAC TT

A.Sorg

ATG TCT TCC TCG AGG CCT GCC CAC TCT GCA AGG ATA TTA GCA CAA AC

TC ACC AAA CCT GGT ATT GAA CCT TAT CTA GGC CTG CAC TAT CCA GCC ACT GAT ATC CCT CA

TT TTC TGG TTC AGG TCA CAT ACA GCT GCT GAA ATC AAA TGG GGA GGT GCA AAG CAT GAT CC

CC ATG ACC AAG CTT ACT GGG TGG CAC AGA GAT GAA GTG ATA GAT AAG ATG CTC CTT GGT GA

TC TCT TCC AAA CTG ACT --- --- AAG AAC AGC ATT GGG GAA AAC CTT CAT CTC ATA GAC TT

Oat3

ATG TCT TCC TCA AGG CCT GCT TCC AGT GCA AGG GTG TTA GCA CAG AC

TC ACA AAG

TT TTT TGG

CA ATG ACC

TT TCT TCC

A.Rice

TC ACA AAG

TG TTC TGG

CC ATG ACG

TC TCT TGT

A.Tom

TC ACA AAG

TG TTC TGG

CA ATG ACC

TT TCA GGC

A.Pot

ATG TCG TCT TCA AGA CCT AGC CAA AGT GCT AGG ATC ATT GCA CAG AC

TC ACA AAG

TG TTC TGG

CG ATG ACC

TT TCA GGC

A.Tob

ATG TCA TCT TCA AGA CCA AGC CAA AGT GCT AGG ATC ATT GCA CAG AC

TC ACG AAG

TG TTC TGG

CA ATG ACG

TT TCA GGC

my\_tree - Notepad

File Edit Format View Help

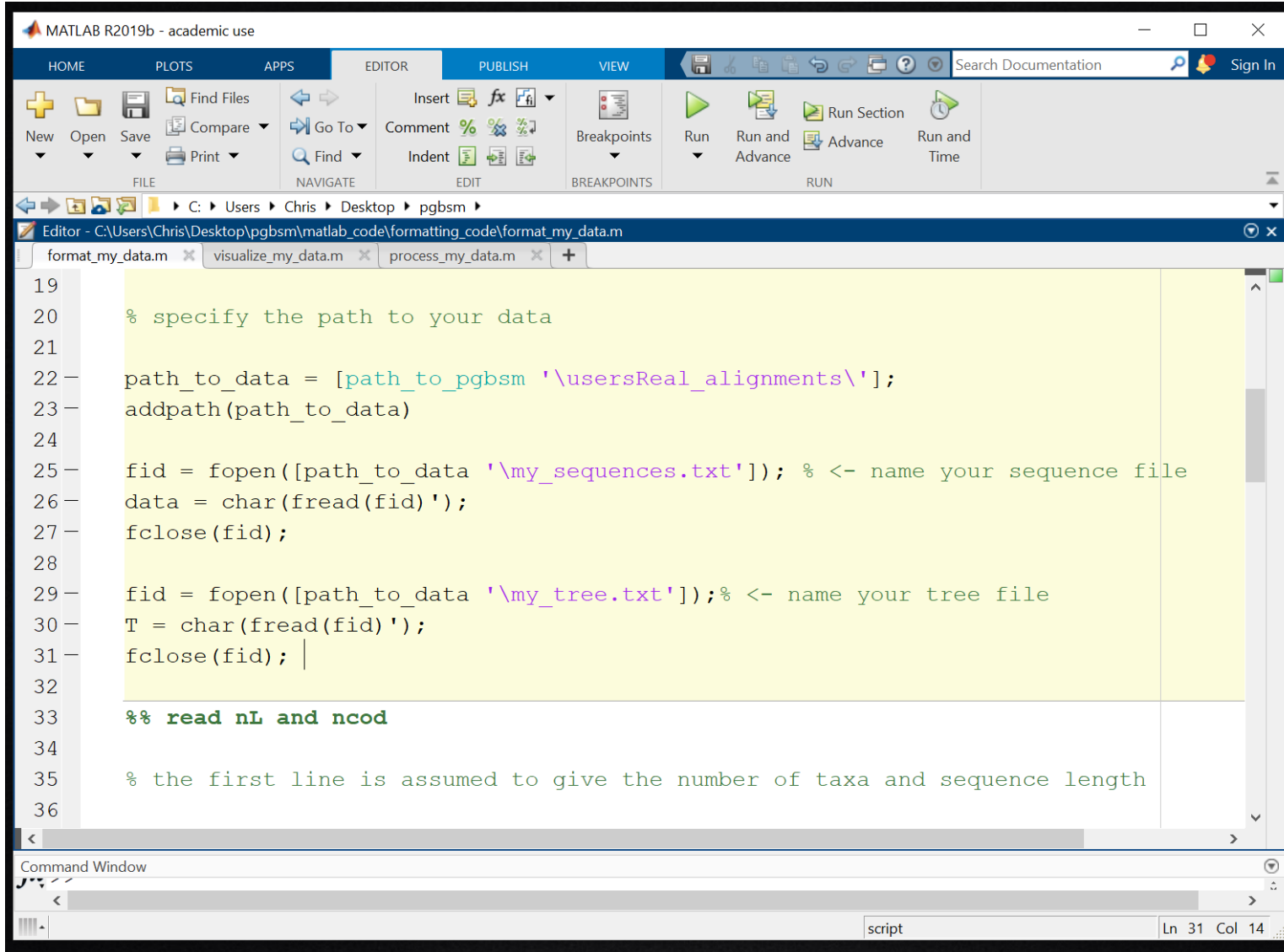
((C.Sorg:1, (F.Tom:1, C.Arab:1):1):1, (Oat3:1, (A.Rice:1, A.Zea:1, A.Sorg:1):1):1, ((A1.Pea1:1, A.Soy:1):1, ((A.Pars:1, (A.Tob:1, (A.Tom:1, A.Pot:1):1):1):1, (A.Zuc:1, A.Arab:1):1):1);

Ln 1, Col 148 100% Windows (CRLF) UTF-8

Ln 1, Col 1 100% Windows (CRLF) UTF-8

# Processing your own Data

The script `format_my_data.m` will arrange the sequences in proper order and replace taxon names with numbers.

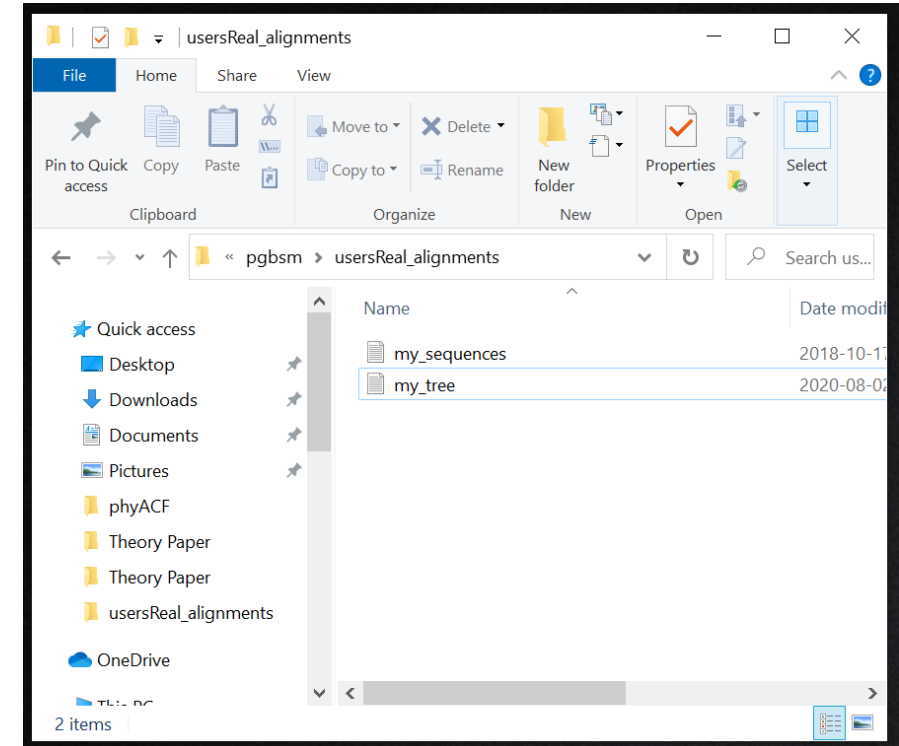


The image shows the MATLAB R2019b interface. The main window displays the script `format_my_data.m` in the Editor. The script is as follows:

```
19
20 % specify the path to your data
21
22 path_to_data = [path_to_pgbasm '\usersReal_alignments\'];
23 addpath(path_to_data)
24
25 fid = fopen([path_to_data '\my_sequences.txt']); % <- name your sequence file
26 data = char(fread(fid)');
27 fclose(fid);
28
29 fid = fopen([path_to_data '\my_tree.txt']); % <- name your tree file
30 T = char(fread(fid)');
31 fclose(fid);
32
33 %% read nL and ncod
34
35 % the first line is assumed to give the number of taxa and sequence length
36
```

The Command Window at the bottom shows the status bar with "Ln 31 Col 14".

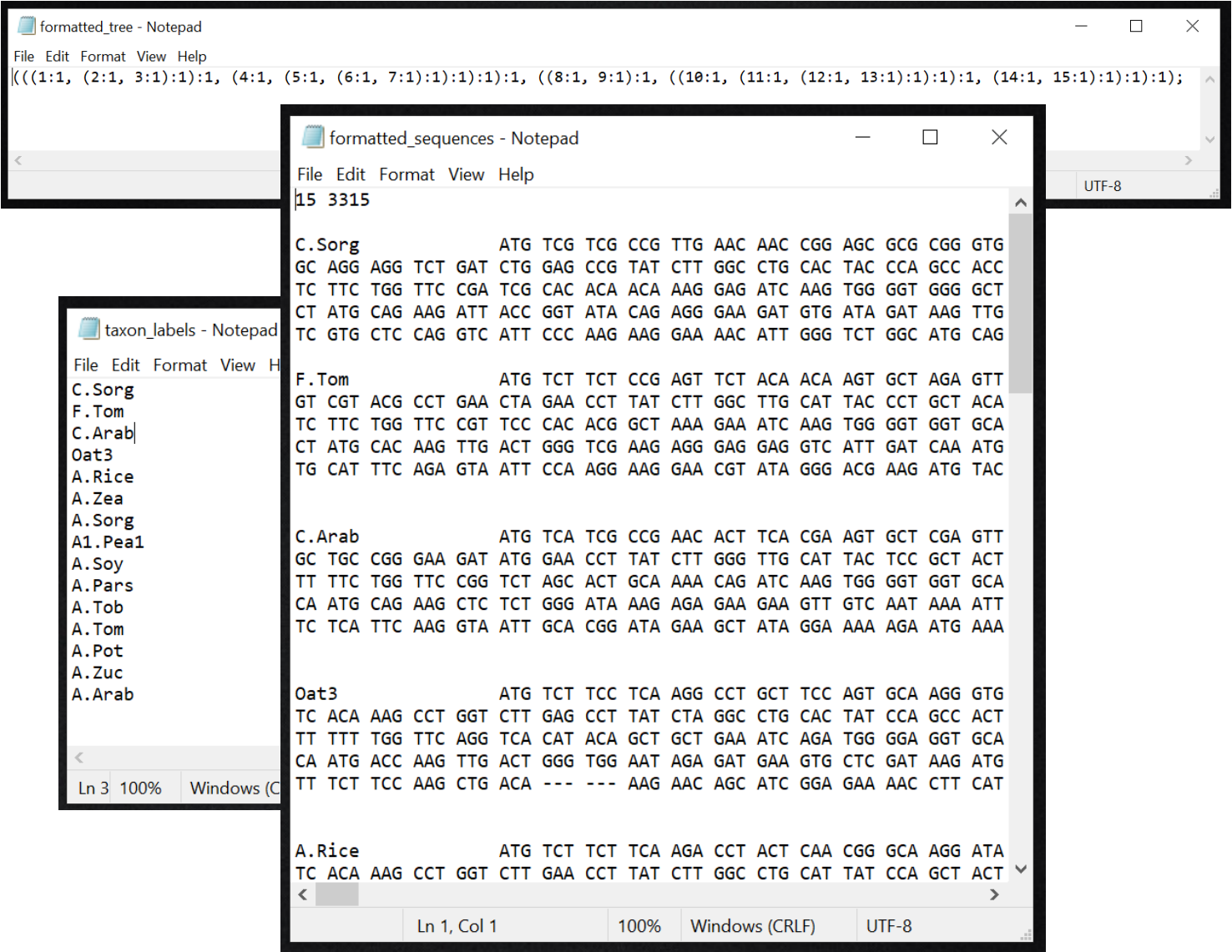
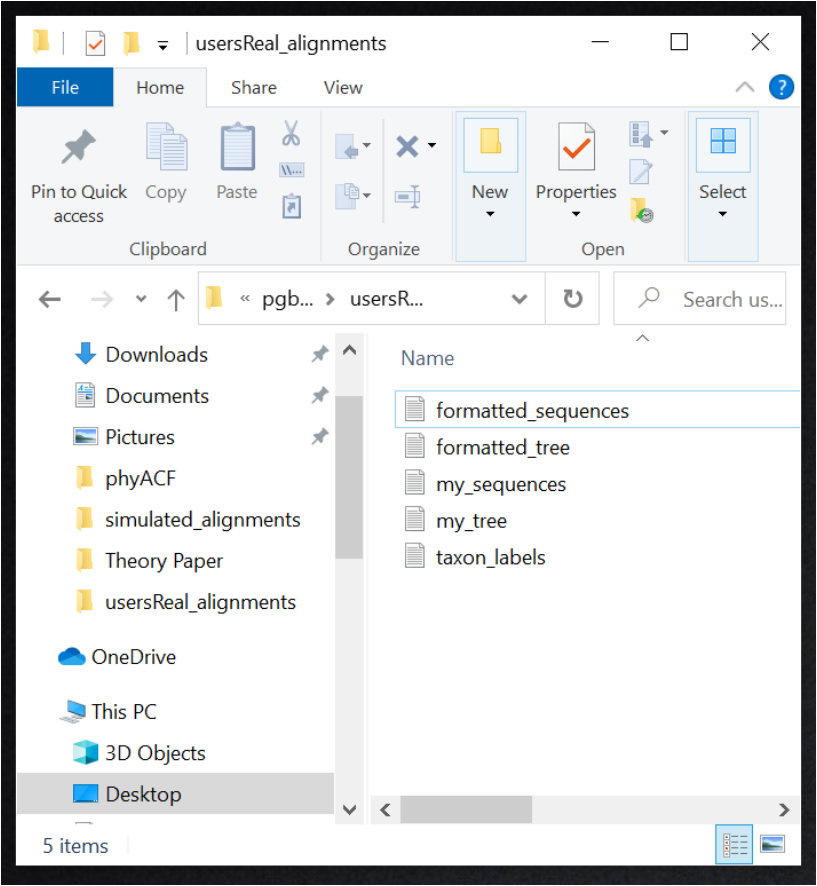
Before running the `format_my_data.m` script make sure the names of your sequence file and tree file in `usersReal_alignments` match what is in the Matlab script.





# Processing your own Data


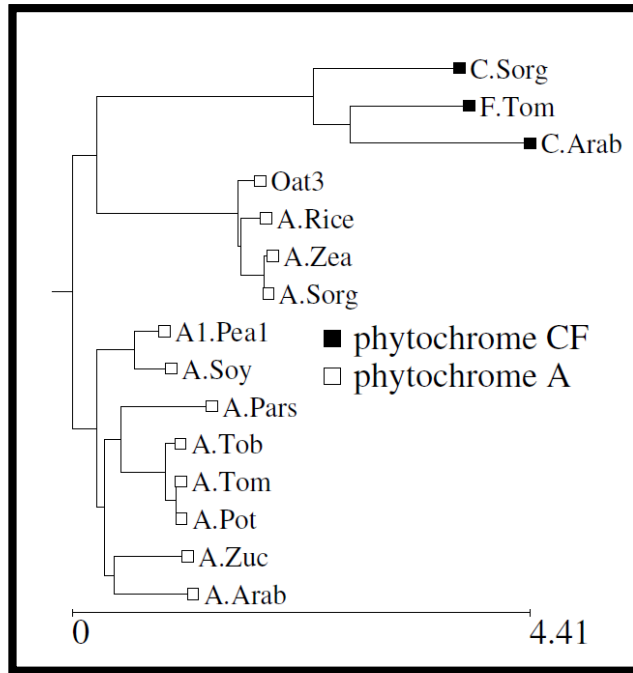
Running `format_my_data.m` creates the three files shown below. The processing code will make use of these files to fit the models to your data.





# Processing your own Data

The last step before processing you will need to make a phenotype\_map.txt file to indicate the phenotype of each taxon using integer indicators starting from zero. In this case the sequences are partitioned into two phenotypes, phyA and phyC + phyF as shown in the tree below.

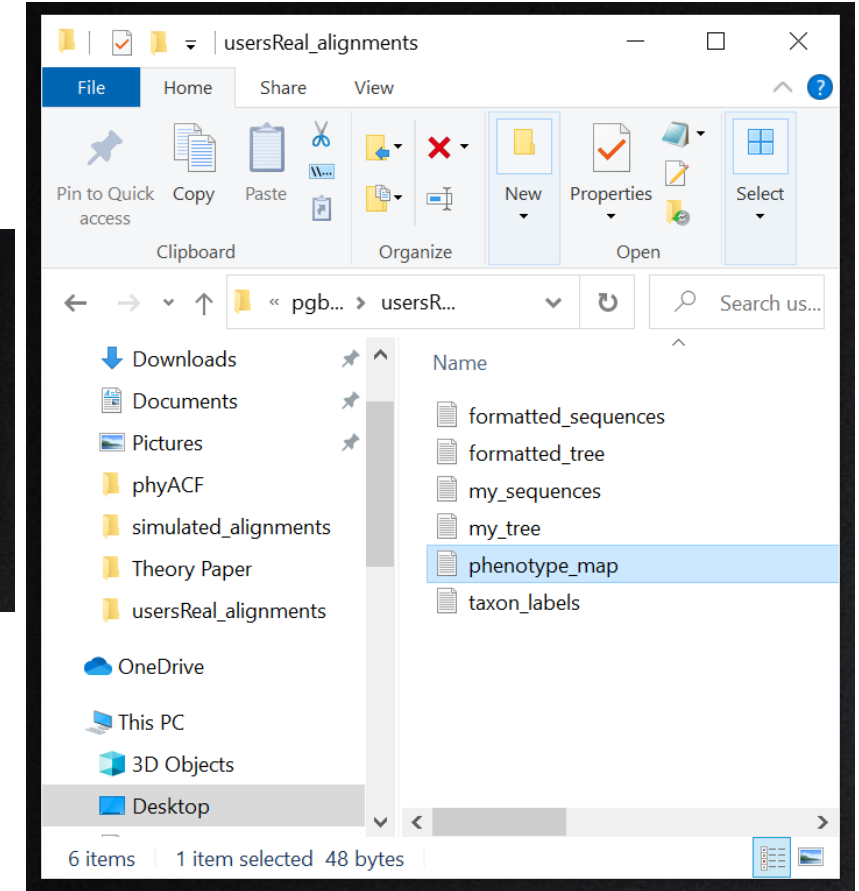


phenotype\_map - Notepad

File Edit Format View Help

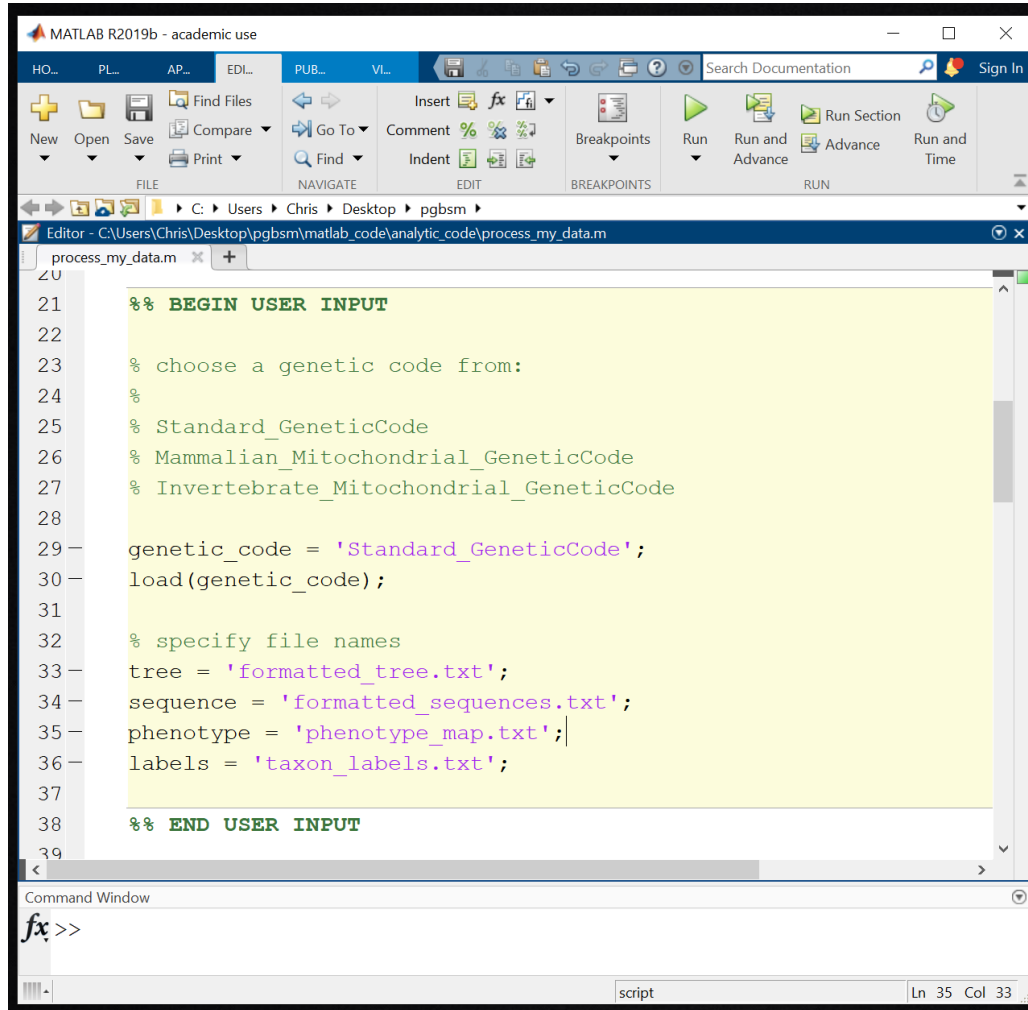
```
phenotypeMap = 0 0 0 1 1 1 1 1 1 1 1 1 1
```

Ln 4, Col 1 100% Macintosh (CR) UTF-8



# Processing your own Data

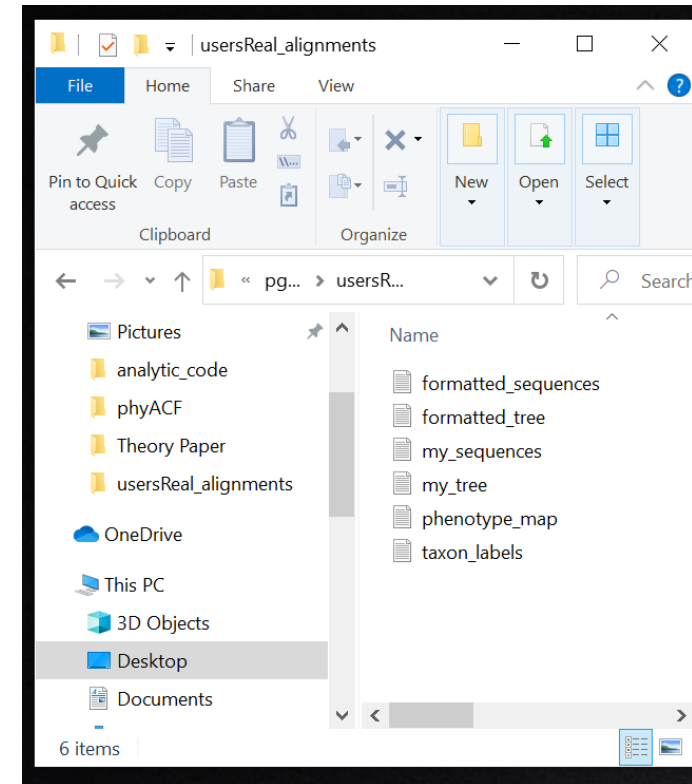
Now open `process_my_data.m` and make sure that the file names in the script are the same as the file names in `usersReal_alignments`. You will also need to select the correct genetic code for your data. Once all this is set press the green triangle to run the code. Processing can a fair amount of time (i.e., more than an hour) depending on the size of your data set.



The image shows the MATLAB R2019b editor window. The script `process_my_data.m` is open, displaying the following code:

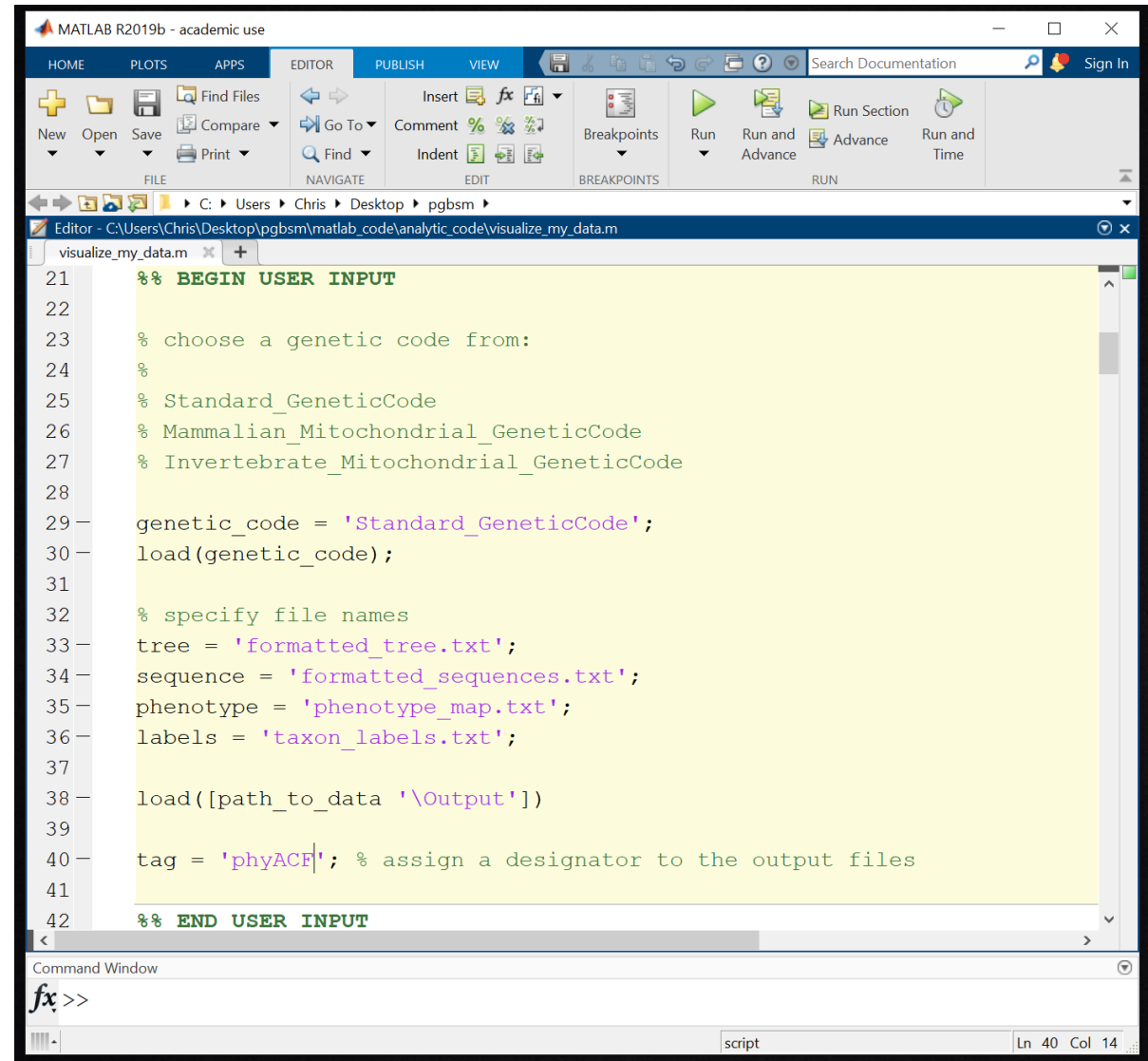
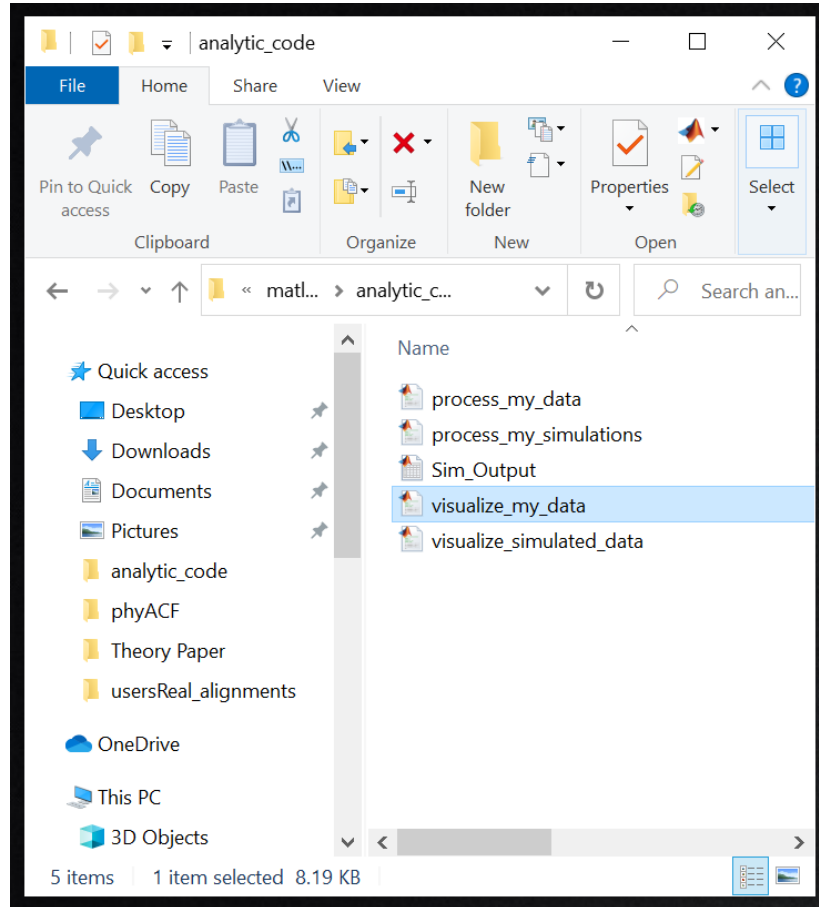
```
21 %% BEGIN USER INPUT
22
23 % choose a genetic code from:
24 %
25 % Standard_GeneticCode
26 % Mammalian_Mitochondrial_GeneticCode
27 % Invertebrate_Mitochondrial_GeneticCode
28
29 genetic_code = 'Standard_GeneticCode';
30 load(genetic_code);
31
32 % specify file names
33 tree = 'formatted_tree.txt';
34 sequence = 'formatted_sequences.txt';
35 phenotype = 'phenotype_map.txt';
36 labels = 'taxon_labels.txt';
37
38 %% END USER INPUT
39
```

The Command Window at the bottom shows the MATLAB prompt `>>`.



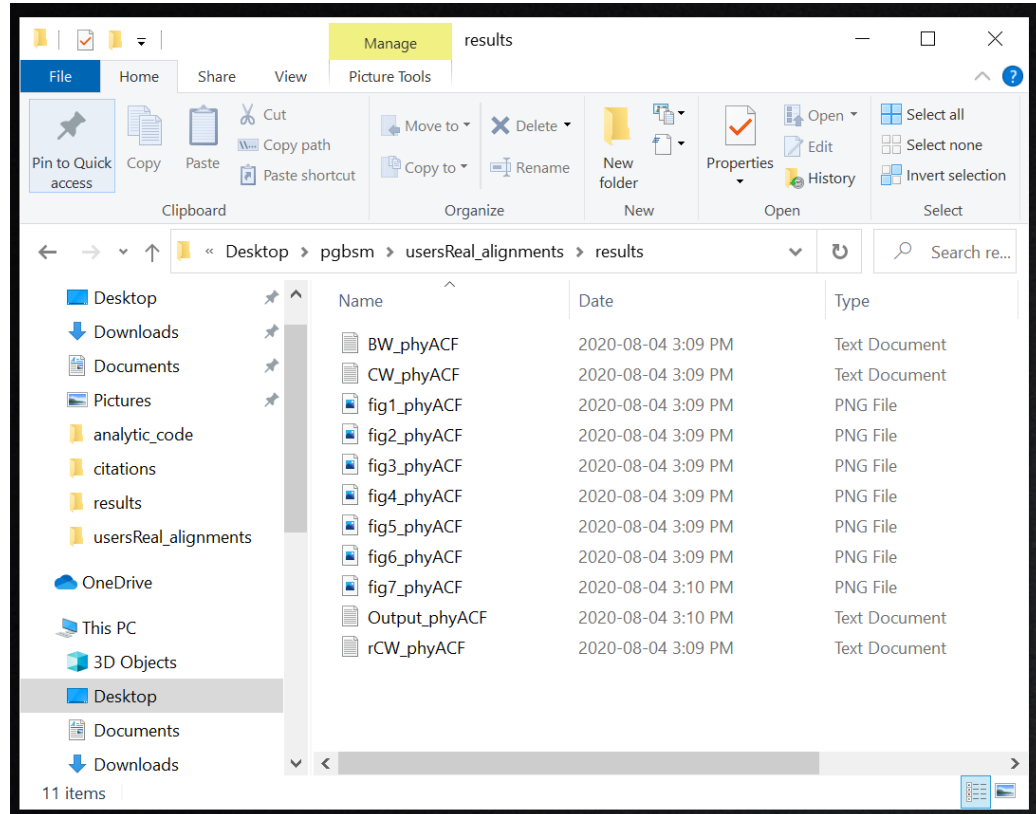
# Visualizing Your Results

Right click on the Matlab script visualize\_my\_data.m to open it in your Matlab window. Be sure all the file names are correct before running the code. You can assign a tag to the output files to help organize the outputs for multiple data sets.



# Visualizing Your Results

In this case the PG-BSM in any of its forms did not provide as good a fit to the data as the alternate RaMoSS model. Hence, whereas there is significant evidence of heterotachy at more than half of the sites in the alignment (the estimated proportion of covarion-like sites was  $\text{piCL} = 0.63$  or 63% of sites – see Jones et al. 2018 for a full description of RaMoSS), there was no evidence for the particular modes of heterotachy tested for by the PG-BSM (i.e., no BW, CW, or rCW sites).



Output\_phyACF - Notepad  
File Edit Format View Help  
PG-BSM Model Fit Output - 05-Aug-2020

Branch Length Estimates

Daughter	Nul	BW	CW	rCW	nulRaMoSS	altRaMoSS	Parent
1	1.41	1.41	1.41	1.41	1.37	1.42	17
2	1.15	1.15	1.15	1.15	1.12	1.15	16
3	1.73	1.74	1.73	1.73	1.7	1.73	16
4	0.22	0.22	0.22	0.22	0.21	0.22	20
5	0.25	0.25	0.25	0.25	0.25	0.25	19
6	0.08	0.08	0.08	0.08	0.08	0.08	18
7	0.04	0.04	0.04	0.04	0.04	0.04	18
8	0.29	0.29	0.29	0.29	0.29	0.29	22
9	0.35	0.36	0.35	0.35	0.35	0.35	22
10	0.87	0.87	0.87	0.87	0.85	0.87	25
11	0.14	0.14	0.14	0.14	0.14	0.14	24
12	0.05	0.05	0.05	0.05	0.05	0.05	23
13	0.06	0.06	0.06	0.06	0.05	0.06	23
14	0.71	0.71	0.7	0.71	0.69	0.71	26
15	0.76	0.76	0.76	0.76	0.75	0.77	26
16	0.35	0.35	0.35	0.35	0.36	0.35	17
17	2.08	2.01	2.08	2.08	2.12	2.14	21
18	0.23	0.23	0.23	0.23	0.23	0.23	19
19	0.03	0.03	0.03	0.03	0.04	0.03	20
20	1.35	1.36	1.35	1.35	1.36	1.38	21
21	0.23	0.23	0.23	0.23	0.21	0.22	29
22	0.37	0.37	0.37	0.37	0.34	0.36	28
23	0.1	0.1	0.1	0.1	0.1	0.1	24
24	0.43	0.43	0.43	0.43	0.42	0.43	25
25	0.17	0.17	0.17	0.17	0.17	0.16	27
26	0.1	0.1	0.1	0.1	0.1	0.1	27
27	0.07	0.07	0.07	0.07	0.08	0.06	28
28	0.23	0.23	0.23	0.23	0.21	0.22	29

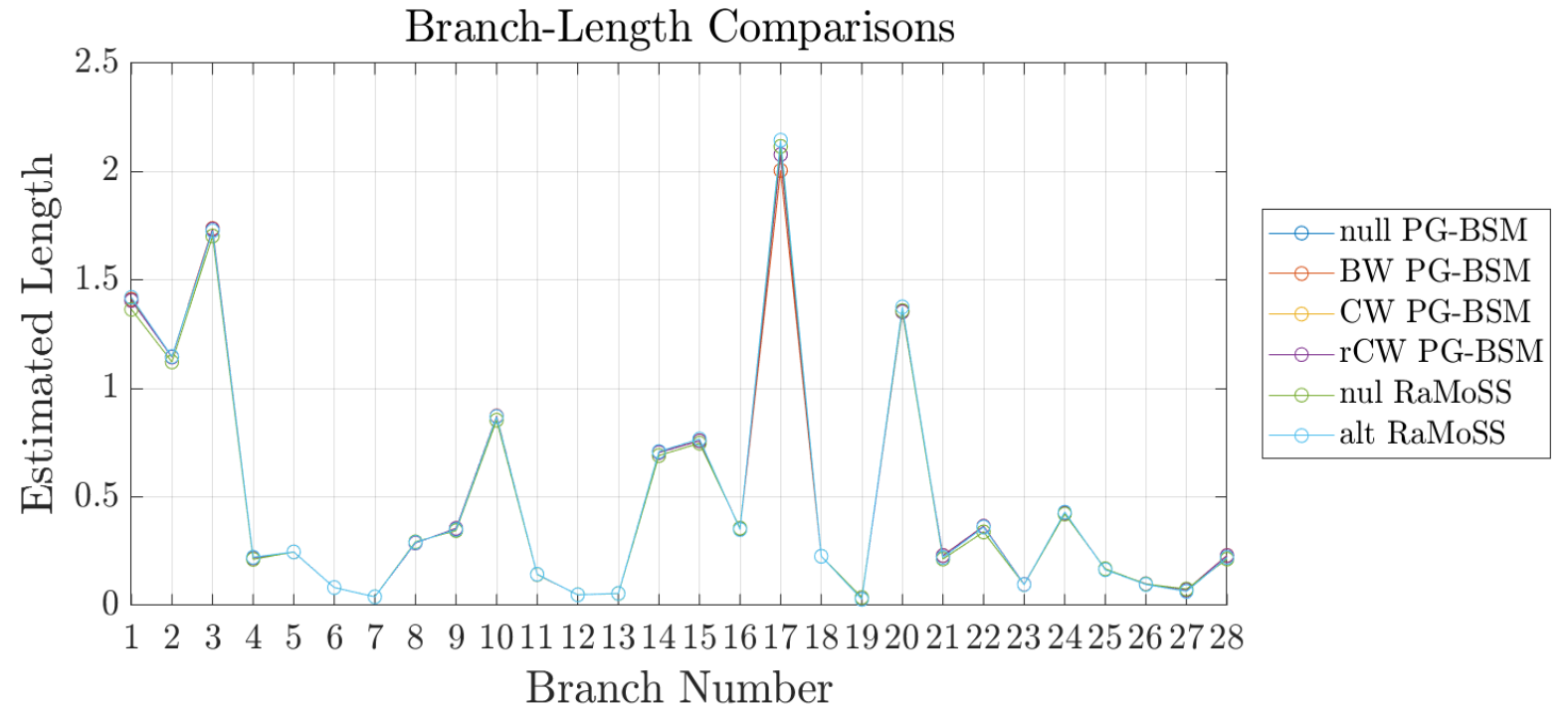
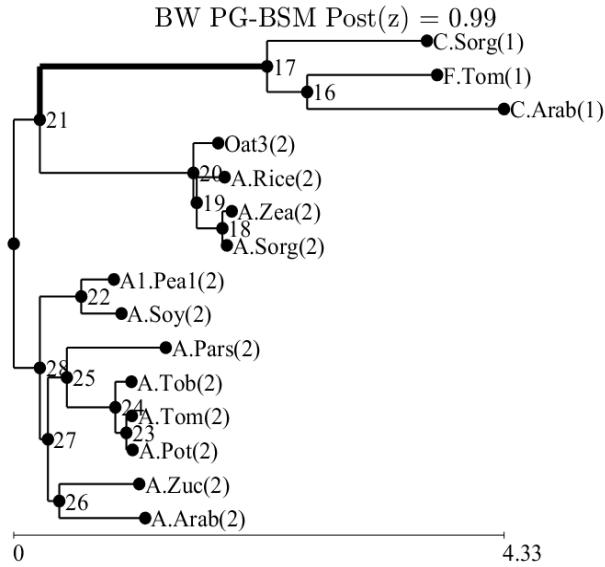
Log Likelihoods

Nul: -28715  
BW: -28715  
CW: -28715  
rCW: -28715  
nulRaMoSS: -28752  
altRaMoSS: -28690

Output\_phyACF - Notepad  
File Edit Format View Help  
altRaMoSS MLEs

piCL	w1M3	w2M3	p1M3	w1CL	w2CL	p1CL	delta	kappa
0.63	0	0.44	0.76	0	0.37	0.64	0.79	2.13

# Interpreting Your Results



Failure to reject the null PG-BSM suggests that none of the non-stationary BW, CW and rCW processes under which a subset of sites undergo changes in their site-specific landscapes along branches over which the phenotype changed (see the PG-BSM Concept slide show for details) had occurred. This is consistent with the negligible difference in branch-length estimates under the null and alternate PG-BSM as shown here (cf. slide 30).

The covarion-like (CL) component of the null PG-BSM and the alternate RaMoSS captures random changes in site-specific rate ratios (heterotachy-by-any-cause, see the PG-BSM Concept slide show for details), which is a stationary process. Rejection of the null RaMoSS in this case demonstrates heterotachy in the data, whereas failure to reject the null PG-BSM demonstrates that the heterotachy is consistent with the stationary CL process. The non-stationary BW, CW and rCW processes are not required to explain the data in this case.



Fin

# Users Tutorial Version 1.00

By C T Jones, May 2020

[cjones2@dal.ca](mailto:cjones2@dal.ca)

random][pLasmId