NetRAX Experiment Evaluation

February 8, 2021

[1]:
```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="darkgrid")
```

[2]:
```
def bic_stats(df):
    print("Inferred BIC better or equal: " + str(len(df[df['bic_inferred'] <=␣
 ↪df['bic_true']])))
    print("Inferred BIC worse: " + str(len(df[df['bic_inferred'] >␣
 ↪df['bic_true']])))

def reticulation_stats(df):
    print("Inferred n_reticulations less: " +␣
 ↪str(len(df[df['n_reticulations_inferred'] < df['n_reticulations']])))
    print("Inferred n_reticulations equal: " +␣
 ↪str(len(df[df['n_reticulations_inferred'] == df['n_reticulations']])))
    print("Inferred n_reticulations more: " +␣
 ↪str(len(df[df['n_reticulations_inferred'] > df['n_reticulations']])))

def weirdness_stats(df):
    df['true_network_weirdness'].plot.hist(bins=10, alpha=0.5, range=(0,1),␣
 ↪title='True network weirdness')

def zero_branches_stats(df):
    df['near_zero_branches_raxml'].plot.hist(bins=10, alpha=0.5,␣
 ↪title='Near-zero branches raxml')

def distances(df):
    fig, axes = plt.subplots(3, 2, constrained_layout=True)
    df['hardwired_cluster_distance'].plot.hist(bins=10, alpha=0.5,␣
 ↪title='Hardwired cluster distance', ax=axes[0,0])
    df['softwired_cluster_distance'].plot.hist(bins=10, alpha=0.5,␣
 ↪title='Softwired cluster distance', ax=axes[0,1])
    df['displayed_trees_distance'].plot.hist(bins=10, alpha=0.5,␣
 ↪title='Displayed trees distance', ax=axes[1,0])
```

```
    df['tripartition_distance'].plot.hist(bins=10, alpha=0.5,␣
↪title='Tripartition distance', ax=axes[1,1])
    df['nested_labels_distance'].plot.hist(bins=10, alpha=0.5, title='Nested␣
↪labels distance', ax=axes[2,0])
    df['path_multiplicity_distance'].plot.hist(bins=10, alpha=0.5, title='Path␣
↪multiplicity distance', ax=axes[2,1])

def build_stats(df):
    plt.figure(0)
    bic_stats(df)
    print("")
    plt.figure(1)
    reticulation_stats(df)
    print("")
    plt.figure(2)
    weirdness_stats(df)
    print("")
    plt.figure(3)
    zero_branches_stats(df)
    print("")
    plt.figure(4)
    distances(df)
```

Load the result CSV:

[3]:
```
df = pd.read_csv('small_network_results.csv')
#df = pd.read_csv('medium_network_norandom_results.csv')
```

[4]:
```
pd.set_option('display.max_columns', None)
df.head()
```

[4]:
```
                                                name  n_taxa  n_trees  \
0  datasets_small_network_0_0/0_9_taxa_1_reticula…       9        2
1  datasets_small_network_0_0/0_9_taxa_1_reticula…       9        2
2  datasets_small_network_0_0/0_9_taxa_1_reticula…       9        2
3  datasets_small_network_0_0/0_9_taxa_1_reticula…       9        2
4  datasets_small_network_0_0/0_9_taxa_1_reticula…       9        2

   n_reticulations  msa_size     sampling_type simulation_type  \
0                1       101  PERFECT_SAMPLING          CELINE
1                1       101  PERFECT_SAMPLING          CELINE
2                1       101  PERFECT_SAMPLING          CELINE
3                1       101  PERFECT_SAMPLING          CELINE
4                1       201  PERFECT_SAMPLING          CELINE

                                      celine_params  \
0  {'to': 0.29257489911596035| 'lambda': 13.92596…
```

```
1  {'to': 0.29257489911596035| 'lambda': 13.92596…
2  {'to': 0.29257489911596035| 'lambda': 13.92596…
3  {'to': 0.29257489911596035| 'lambda': 13.92596…
4  {'to': 0.29257489911596035| 'lambda': 13.92596…

                  seqgen_params  near_zero_branches_raxml  \
0  -mHKY -t3.0 -f0.3|0.2|0.2|0.3                         1
1  -mHKY -t3.0 -f0.3|0.2|0.2|0.3                         1
2  -mHKY -t3.0 -f0.3|0.2|0.2|0.3                         1
3  -mHKY -t3.0 -f0.3|0.2|0.2|0.3                         1
4  -mHKY -t3.0 -f0.3|0.2|0.2|0.3                         0

   n_equal_tree_pairs  true_network_weirdness  \
0                   0                       0
1                   0                       0
2                   0                       0
3                   0                       0
4                   0                       0

                                 true_network_path  \
0  datasets_small_network_0_0/0_9_taxa_1_reticula…
1  datasets_small_network_0_0/0_9_taxa_1_reticula…
2  datasets_small_network_0_0/0_9_taxa_1_reticula…
3  datasets_small_network_0_0/0_9_taxa_1_reticula…
4  datasets_small_network_0_0/0_9_taxa_1_reticula…

                            inferred_network_path likelihood_type  \
0  datasets_small_network_0_0/0_9_taxa_1_reticula…         AVERAGE
1  datasets_small_network_0_0/0_9_taxa_1_reticula…         AVERAGE
2  datasets_small_network_0_0/0_9_taxa_1_reticula…            BEST
3  datasets_small_network_0_0/0_9_taxa_1_reticula…            BEST
4  datasets_small_network_0_0/0_9_taxa_1_reticula…         AVERAGE

  brlen_linkage_type  start_type  timeout  n_random_start_networks  \
0             LINKED  FROM_RAXML        0                        0
1             LINKED      RANDOM        0                        5
2             LINKED  FROM_RAXML        0                        0
3             LINKED      RANDOM        0                        5
4             LINKED  FROM_RAXML        0                        0

   n_parsimony_start_networks  runtime_inference  n_reticulations_inferred  \
0                           0           403.473                         0
1                           5          4479.135                         0
2                           0           112.167                         0
3                           5          1381.504                         0
4                           0           191.855                         0
```

```
      bic_true   logl_true  bic_inferred  logl_inferred    bic_raxml  \
0  1161.790768 -451.460827   1139.928369    -454.154318  1139.928108
1  1161.790768 -451.460827   1139.927152    -454.153709  1139.928108
2  1161.778885 -451.454886   1139.928369    -454.154318  1139.928108
3  1161.778885 -451.454886   1139.941146    -454.160706  1139.928108
4  2046.490494 -880.735187   2035.762447    -890.372222  2035.762098

   logl_raxml  rf_absolute_raxml  rf_relative_raxml  rf_absolute_inferred  \
0 -454.154187                 -1                 -1                    -1
1 -454.154187                 -1                 -1                    -1
2 -454.154187                 -1                 -1                    -1
3 -454.154187                 -1                 -1                    -1
4 -890.372048                 -1                 -1                    -1

   rf_relative_inferred  hardwired_cluster_distance  \
0                    -1                         4.0
1                    -1                         4.0
2                    -1                         4.0
3                    -1                         4.0
4                    -1                         4.0

   softwired_cluster_distance  displayed_trees_distance  \
0                         5.5                       1.5
1                         5.5                       1.5
2                         5.5                       1.5
3                         4.5                       1.5
4                         6.5                       1.5

   tripartition_distance  nested_labels_distance  path_multiplicity_distance
0                    5.5                     7.0                         6.0
1                    5.5                     7.0                         6.0
2                    5.5                     7.0                         6.0
3                    5.5                     7.0                         6.0
4                    5.5                     7.0                         6.0
```

[5]:
```
df.columns
```

[5]:
```
Index(['name', 'n_taxa', 'n_trees', 'n_reticulations', 'msa_size',
       'sampling_type', 'simulation_type', 'celine_params', 'seqgen_params',
       'near_zero_branches_raxml', 'n_equal_tree_pairs',
       'true_network_weirdness', 'true_network_path', 'inferred_network_path',
       'likelihood_type', 'brlen_linkage_type', 'start_type', 'timeout',
       'n_random_start_networks', 'n_parsimony_start_networks',
       'runtime_inference', 'n_reticulations_inferred', 'bic_true',
       'logl_true', 'bic_inferred', 'logl_inferred', 'bic_raxml', 'logl_raxml',
       'rf_absolute_raxml', 'rf_relative_raxml', 'rf_absolute_inferred',
       'rf_relative_inferred', 'hardwired_cluster_distance',
```

```
       'softwired_cluster_distance', 'displayed_trees_distance',
       'tripartition_distance', 'nested_labels_distance',
       'path_multiplicity_distance'],
      dtype='object')
```

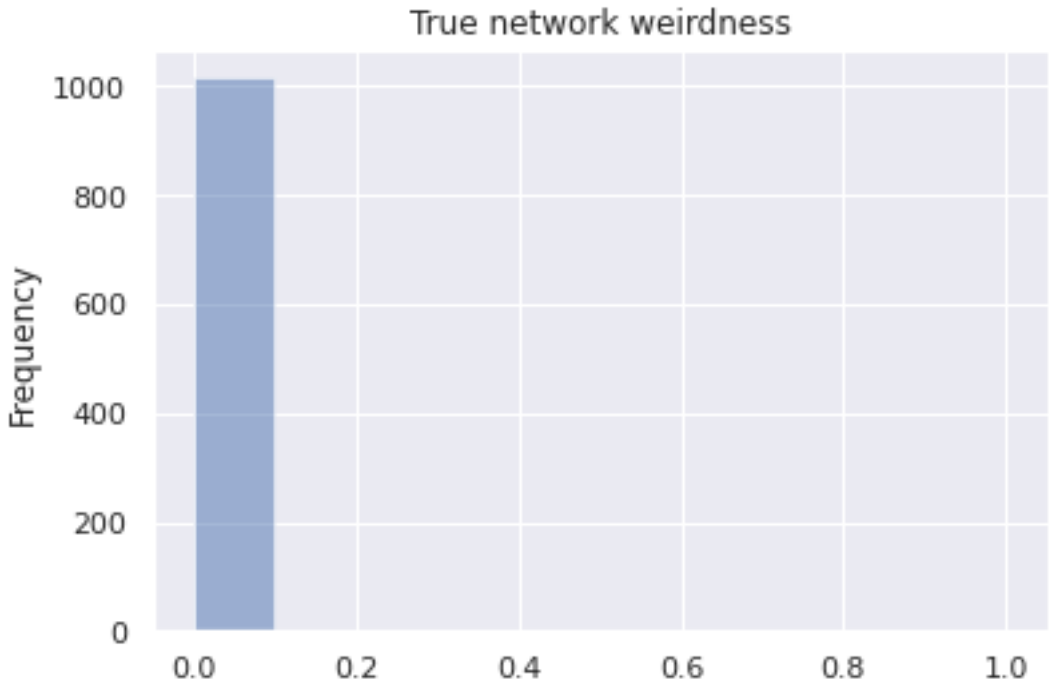
[6]:
```
build_stats(df)
```

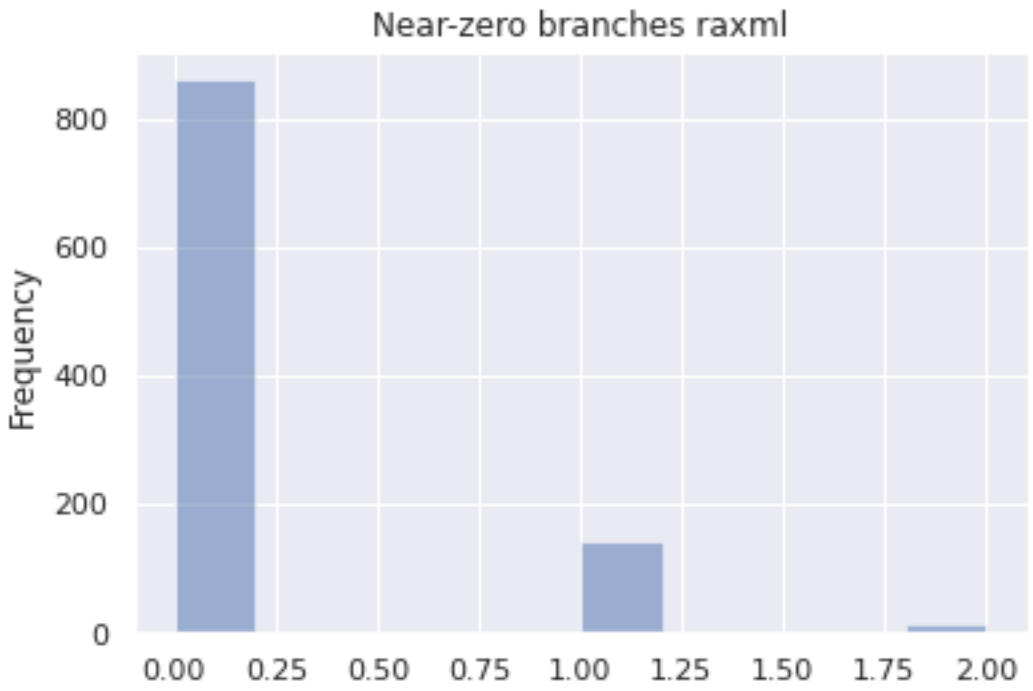
```
Inferred BIC better or equal: 977
Inferred BIC worse: 39

Inferred n_reticulations less: 912
Inferred n_reticulations equal: 104
Inferred n_reticulations more: 0
```

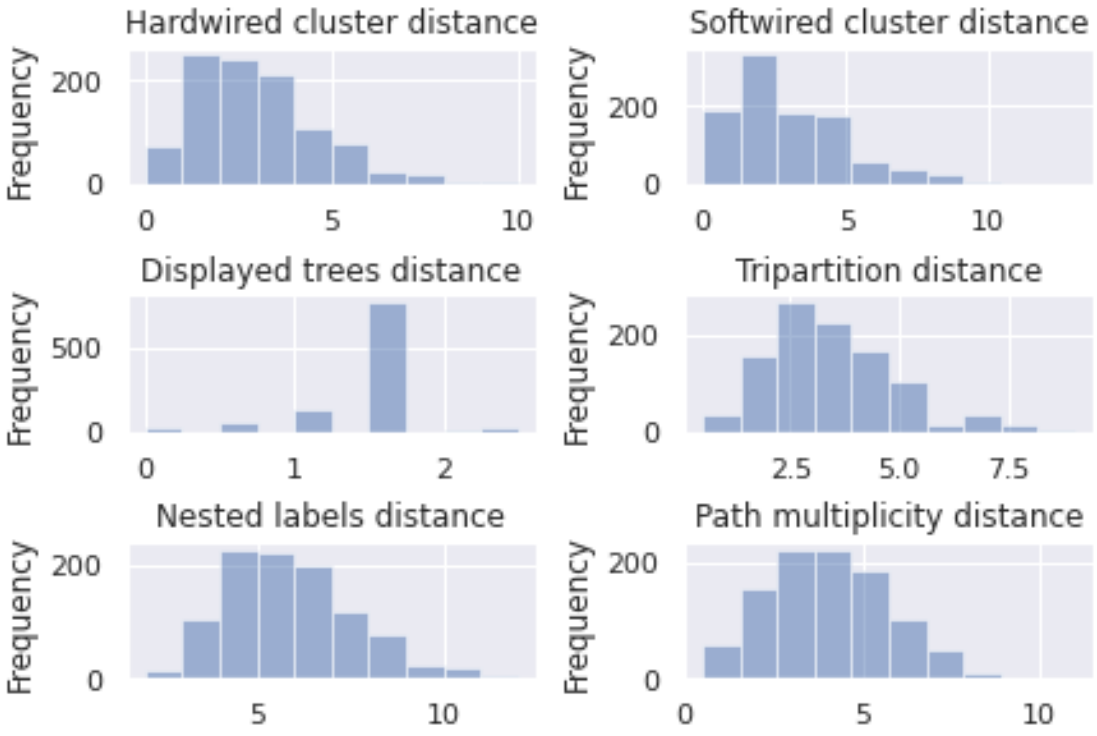
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```





<Figure size 432x288 with 0 Axes>



1 Plots for starting with raxml-ng best tree only

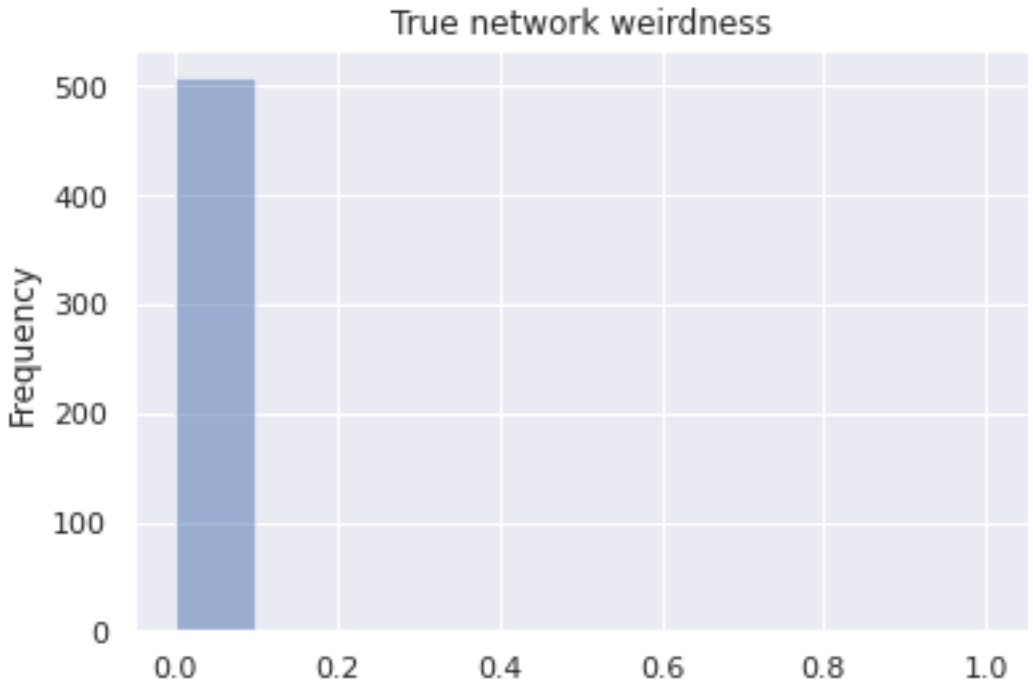
[7]:
```
df_raxml_only = df.query('start_type == "FROM_RAXML"')
build_stats(df_raxml_only)
```

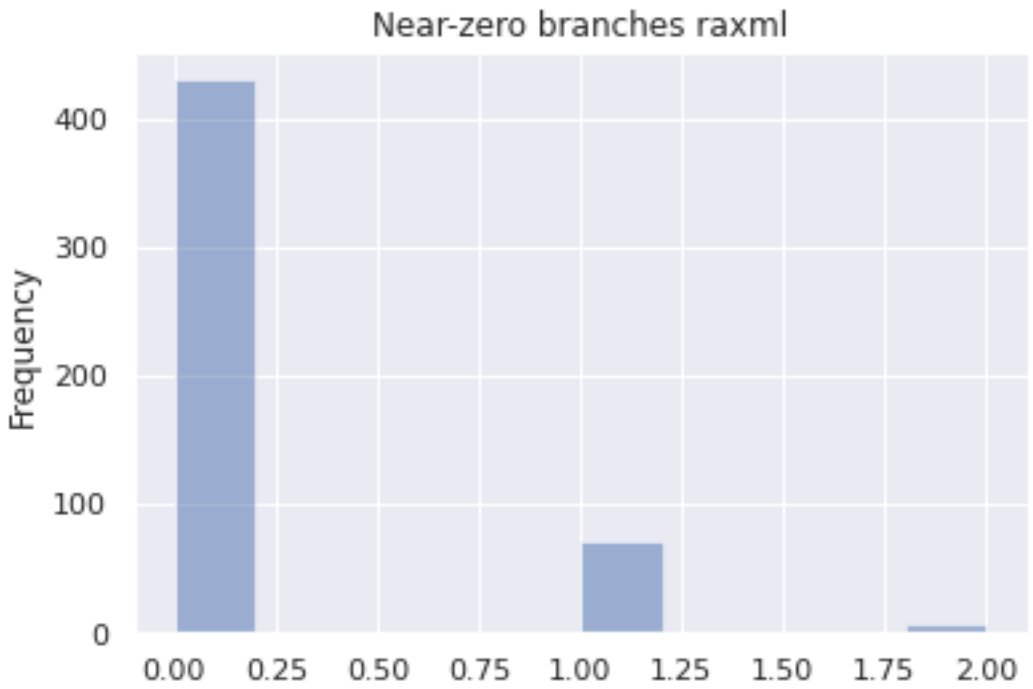
```
Inferred BIC better or equal: 474
Inferred BIC worse: 34

Inferred n_reticulations less: 462
Inferred n_reticulations equal: 46
Inferred n_reticulations more: 0
```

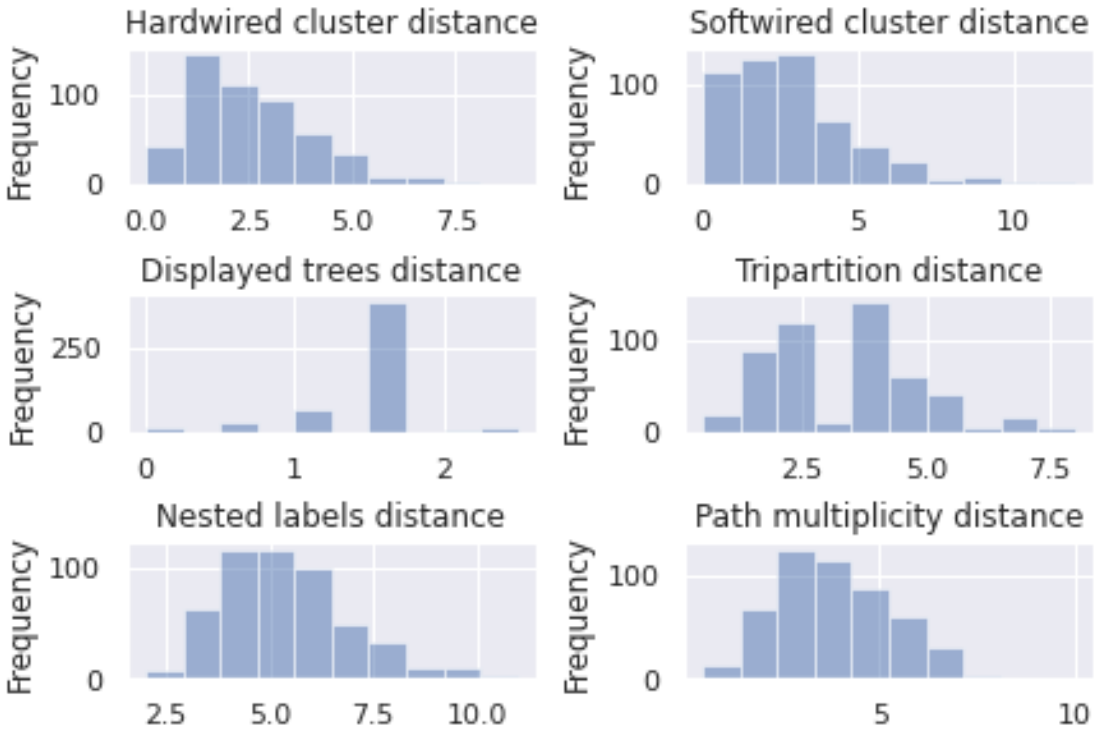
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```






```
<Figure size 432x288 with 0 Axes>
```



1.1 Plots for MSA_size ~ 100*n_trees

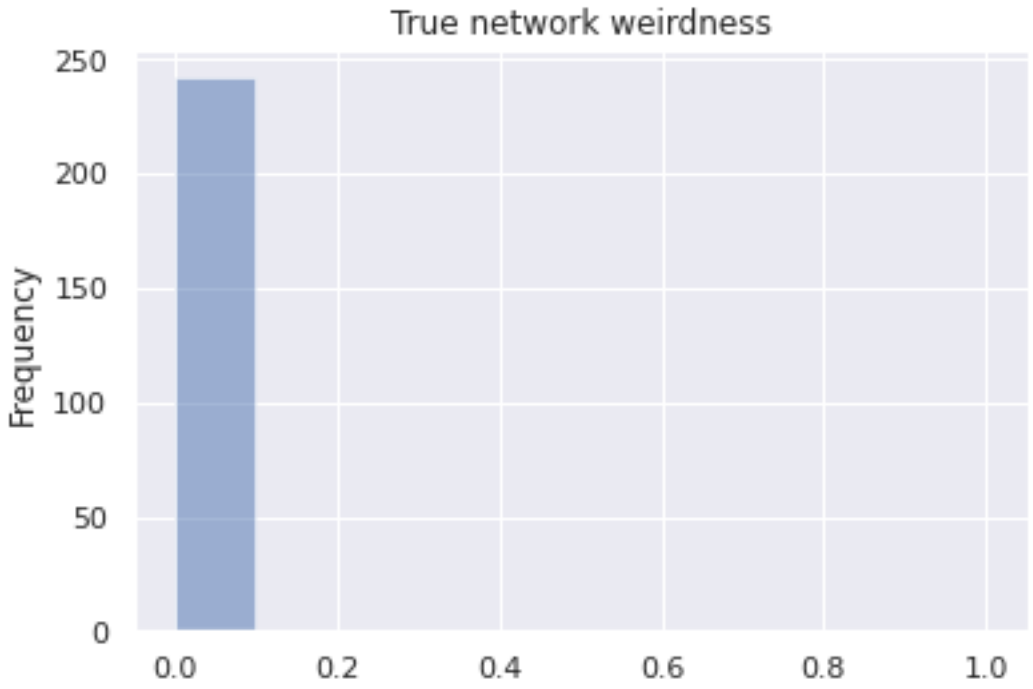
[8]:
```
df_raxml_only_msasize_100 = df_raxml_only.query('msa_size == 101')
build_stats(df_raxml_only_msasize_100)
```

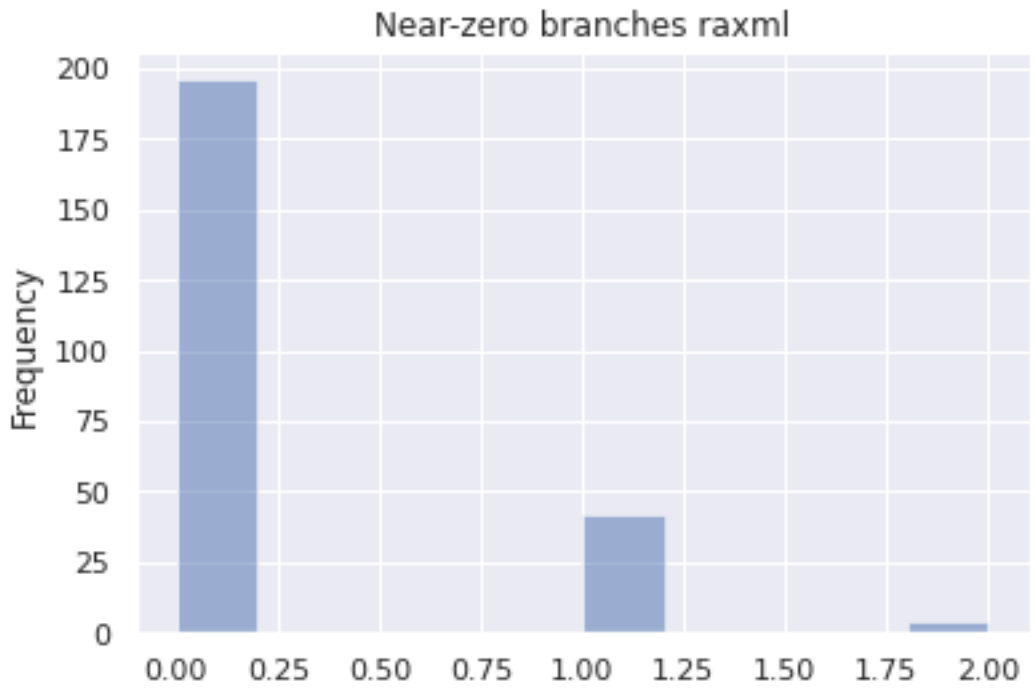
```
Inferred BIC better or equal: 234
Inferred BIC worse: 8

Inferred n_reticulations less: 230
Inferred n_reticulations equal: 12
Inferred n_reticulations more: 0
```

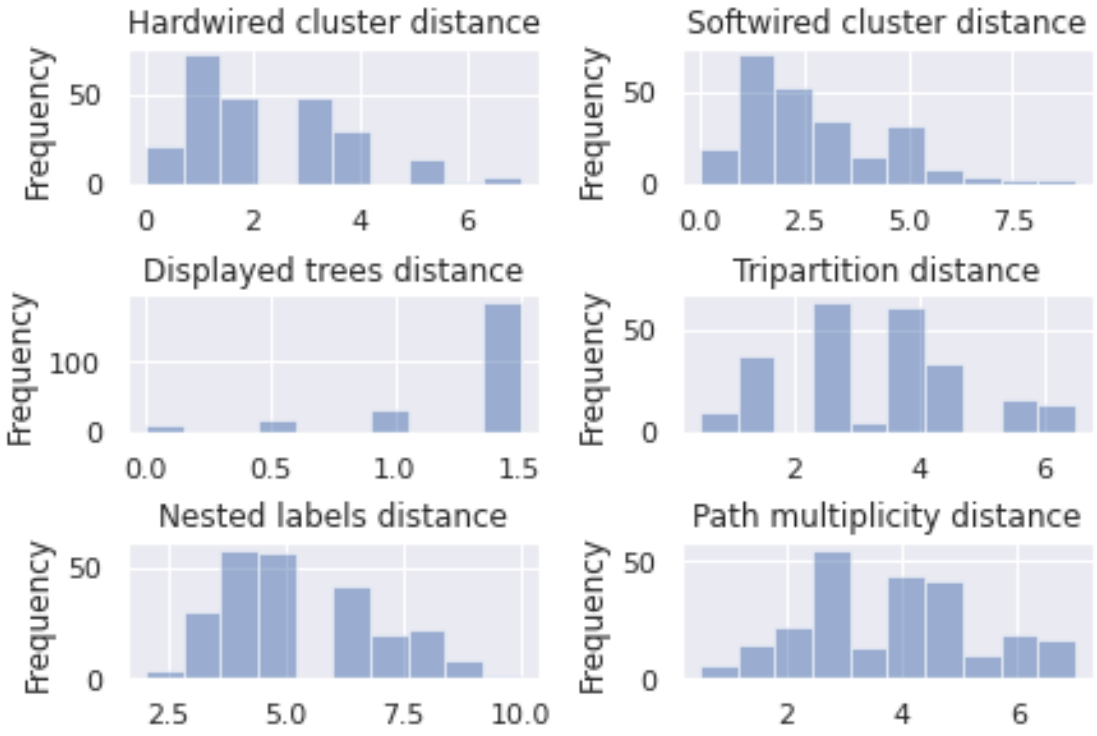
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```





<Figure size 432x288 with 0 Axes>



1.1.1 Plots for LikelihodType.AVERAGE

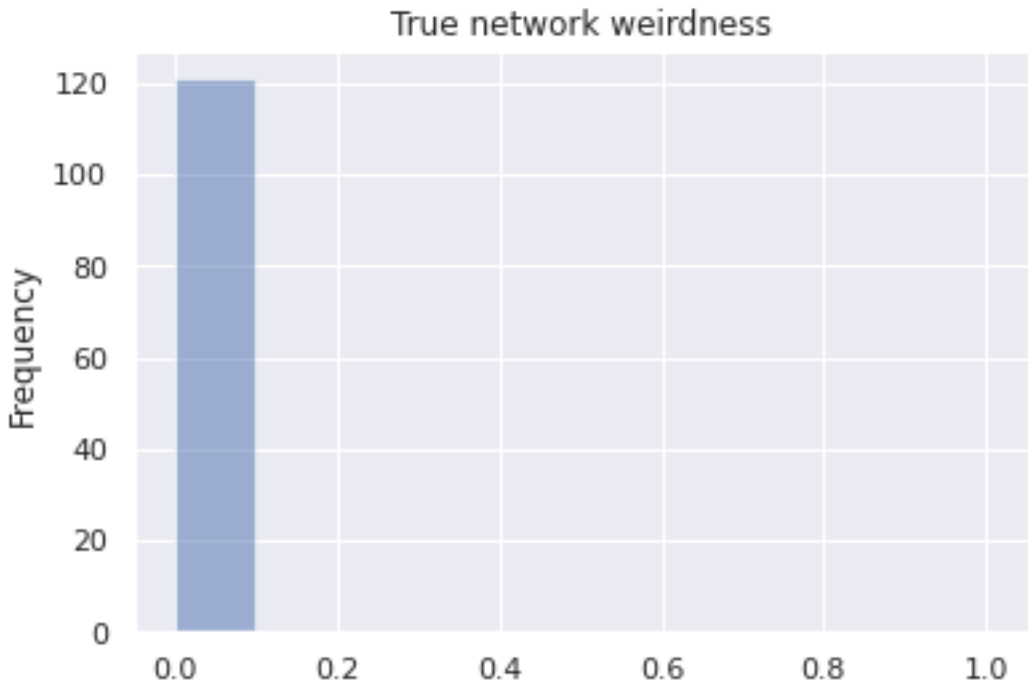
[9]:
```
df_raxml_only_msasize_100_average = df_raxml_only_msasize_100.
 ↪query('likelihood_type == "AVERAGE"')
build_stats(df_raxml_only_msasize_100_average)
```

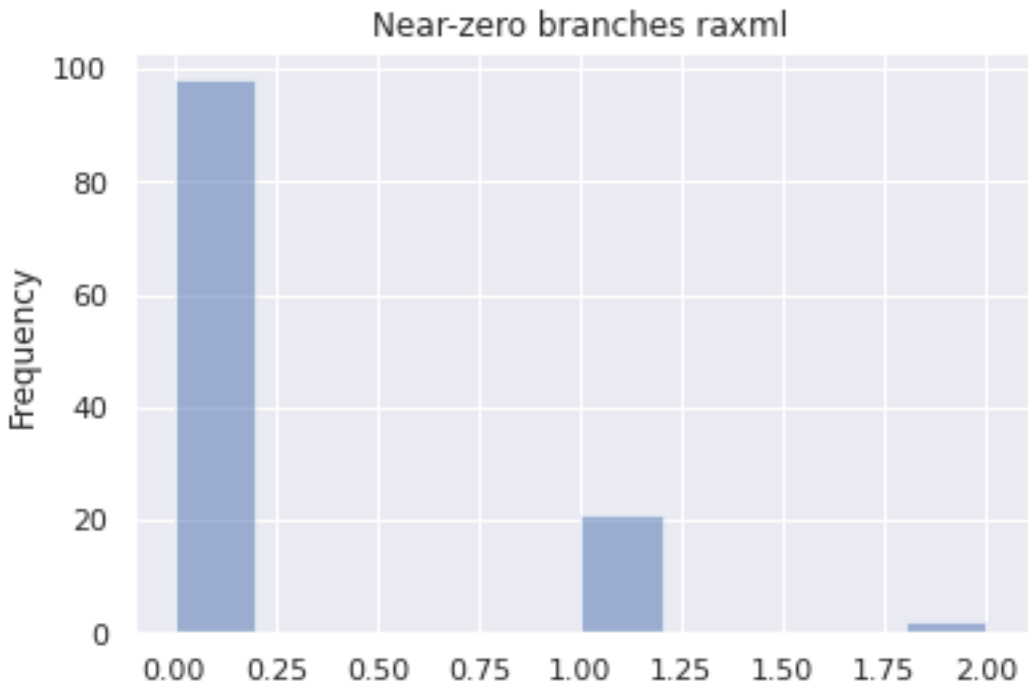
```
Inferred BIC better or equal: 117
Inferred BIC worse: 4

Inferred n_reticulations less: 115
Inferred n_reticulations equal: 6
Inferred n_reticulations more: 0
```

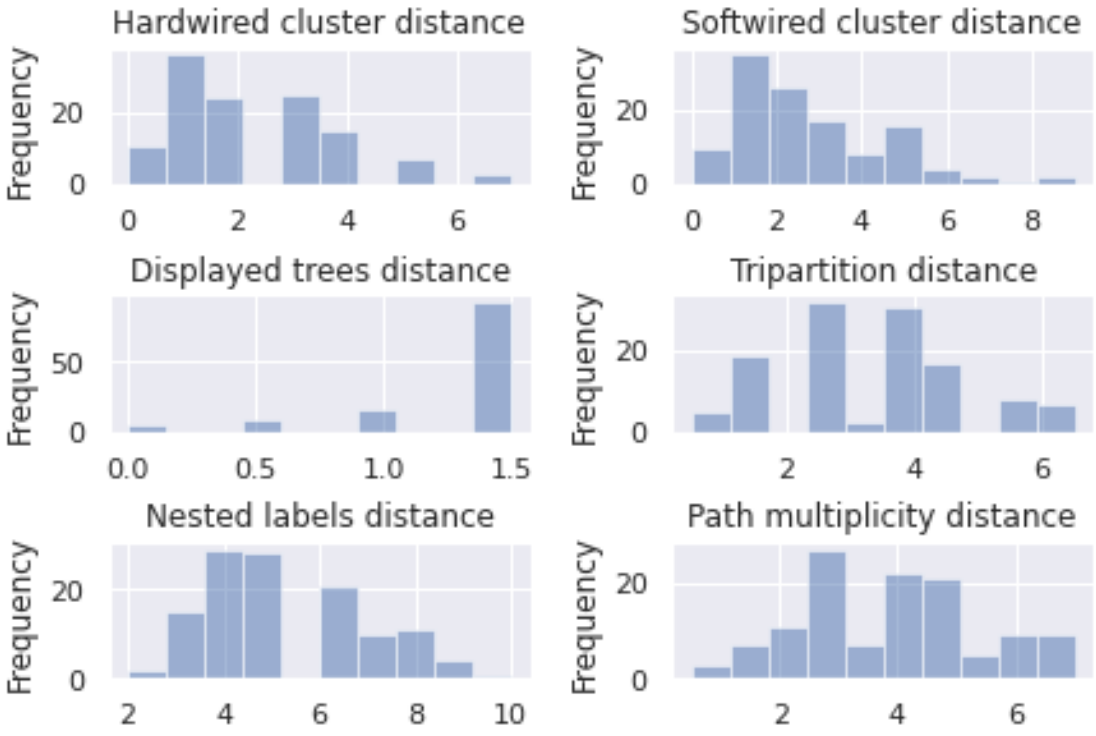
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```





<Figure size 432x288 with 0 Axes>



1.1.2 Plots for LikelihoodType.BEST

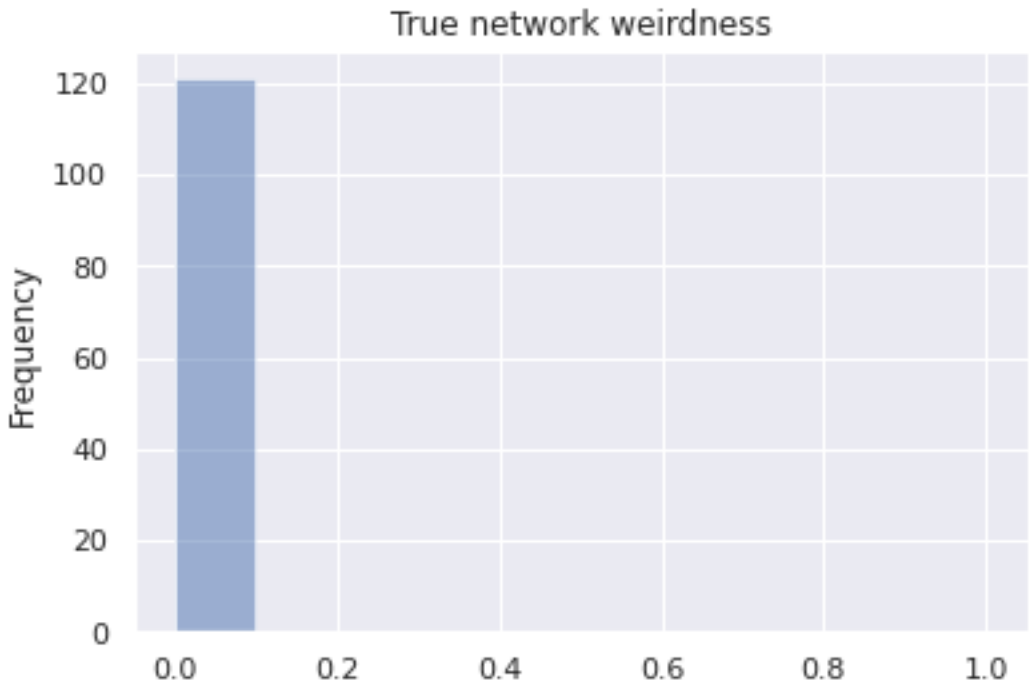
[10]:
```
df_raxml_only_msasize_100_best = df_raxml_only_msasize_100.
 ↪query('likelihood_type == "BEST"')
build_stats(df_raxml_only_msasize_100_best)
```

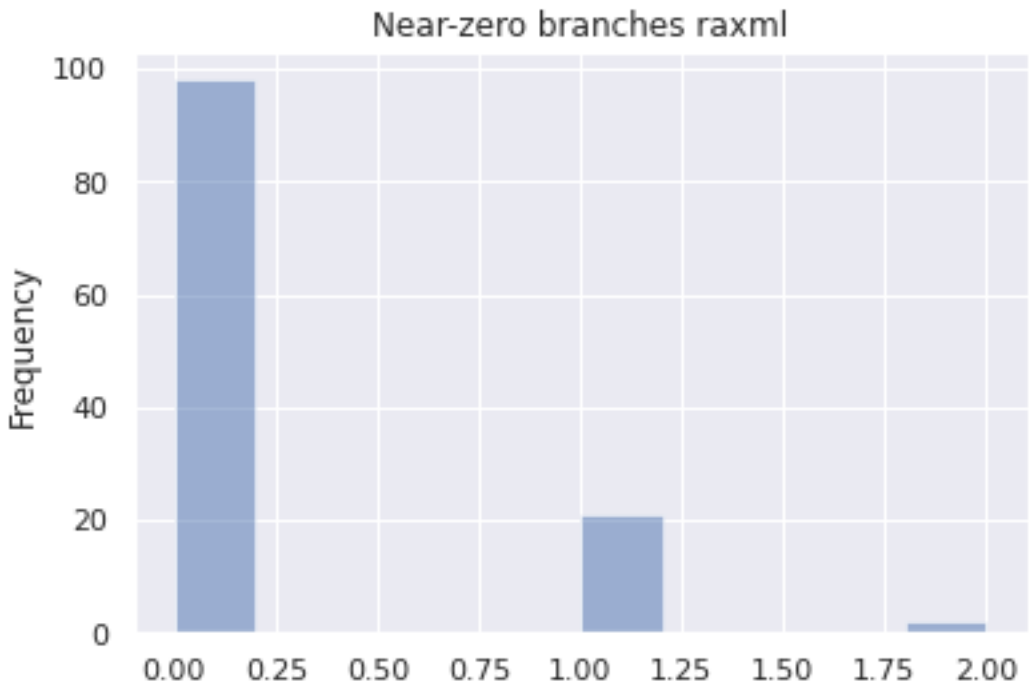
```
Inferred BIC better or equal: 117
Inferred BIC worse: 4

Inferred n_reticulations less: 115
Inferred n_reticulations equal: 6
Inferred n_reticulations more: 0
```

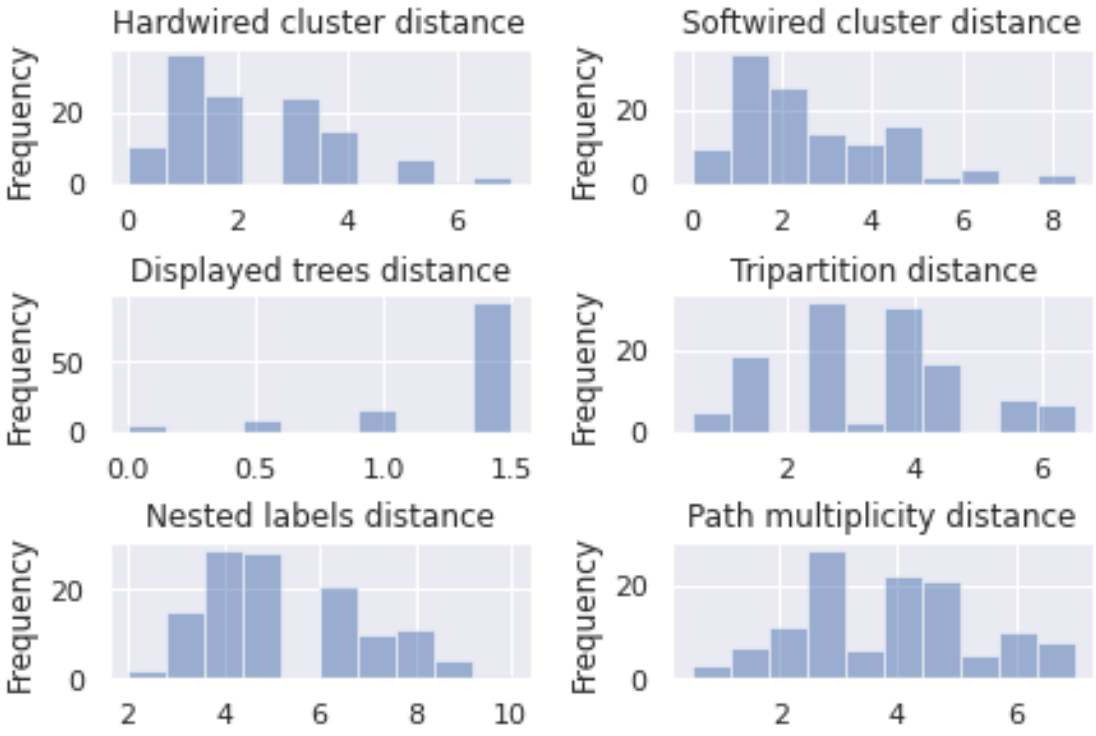
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```





<Figure size 432x288 with 0 Axes>



1.2 Plots for MSA_size ~ 200*n_trees

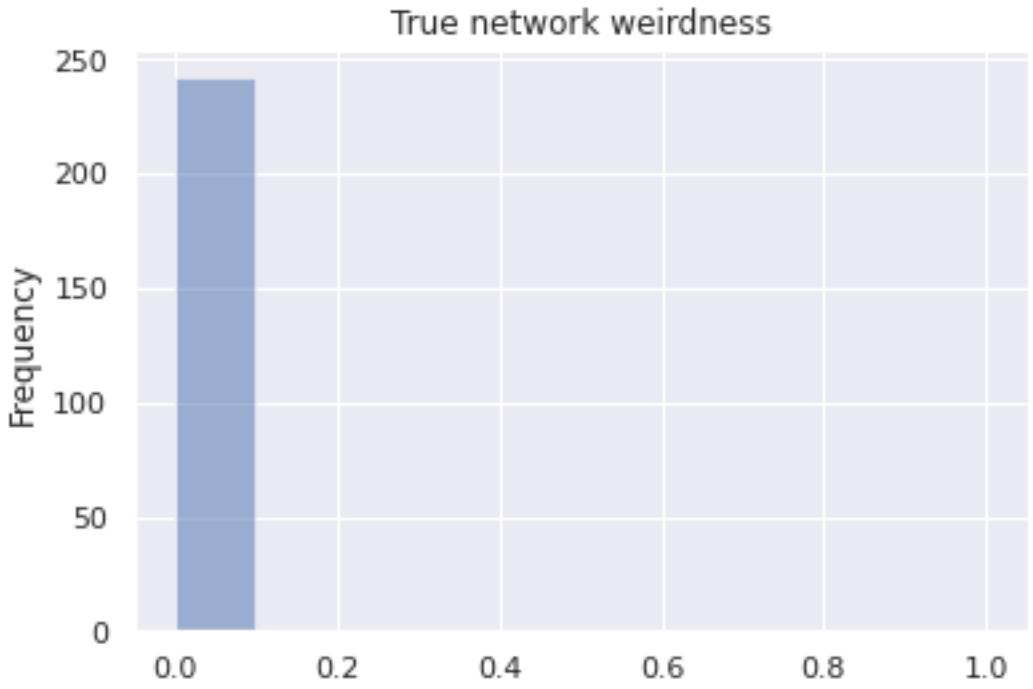
[11]:
```
df_raxml_only_msasize_200 = df_raxml_only.query('msa_size == 201')
build_stats(df_raxml_only_msasize_200)
```

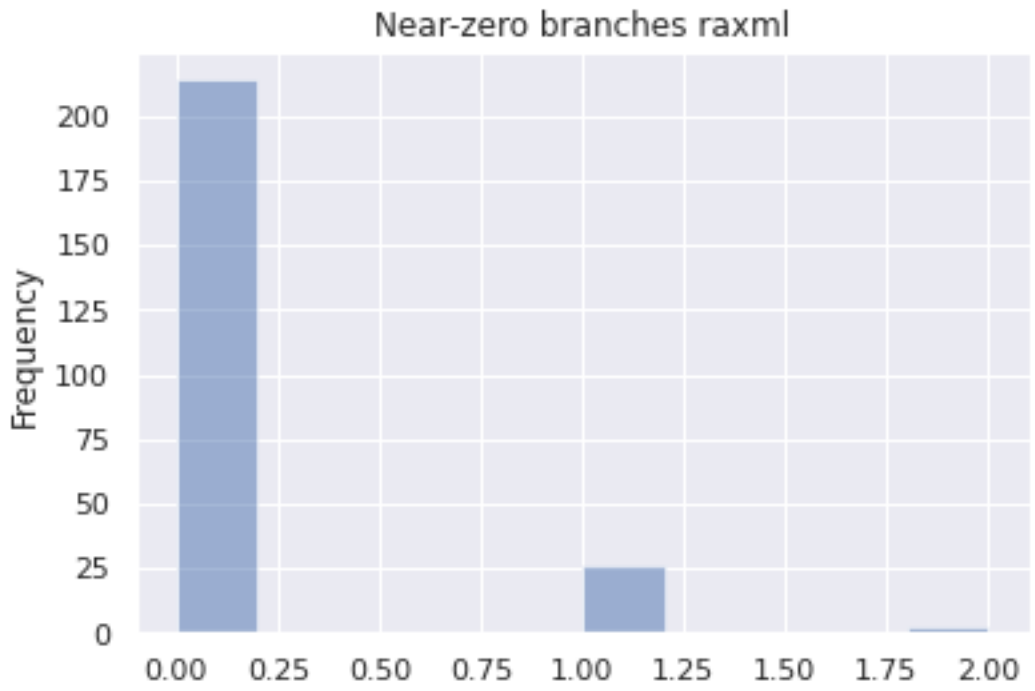
```
Inferred BIC better or equal: 216
Inferred BIC worse: 26

Inferred n_reticulations less: 208
Inferred n_reticulations equal: 34
Inferred n_reticulations more: 0
```

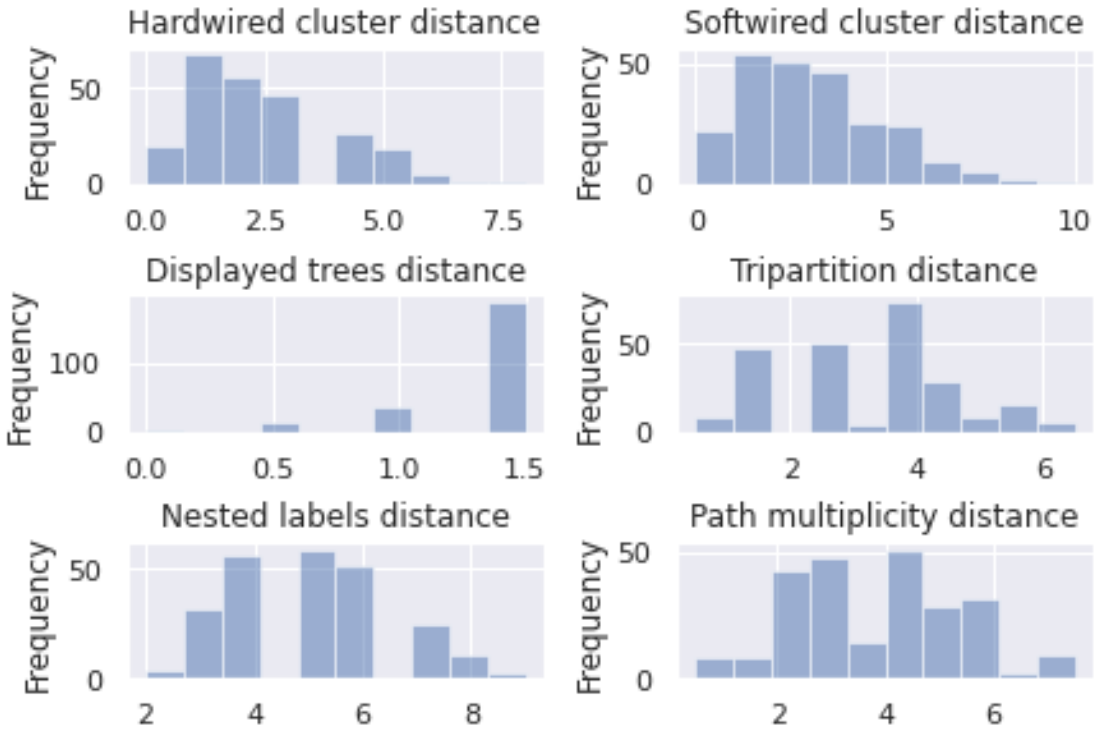
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```





<Figure size 432x288 with 0 Axes>



1.2.1 Plots for LikelihodType.AVERAGE

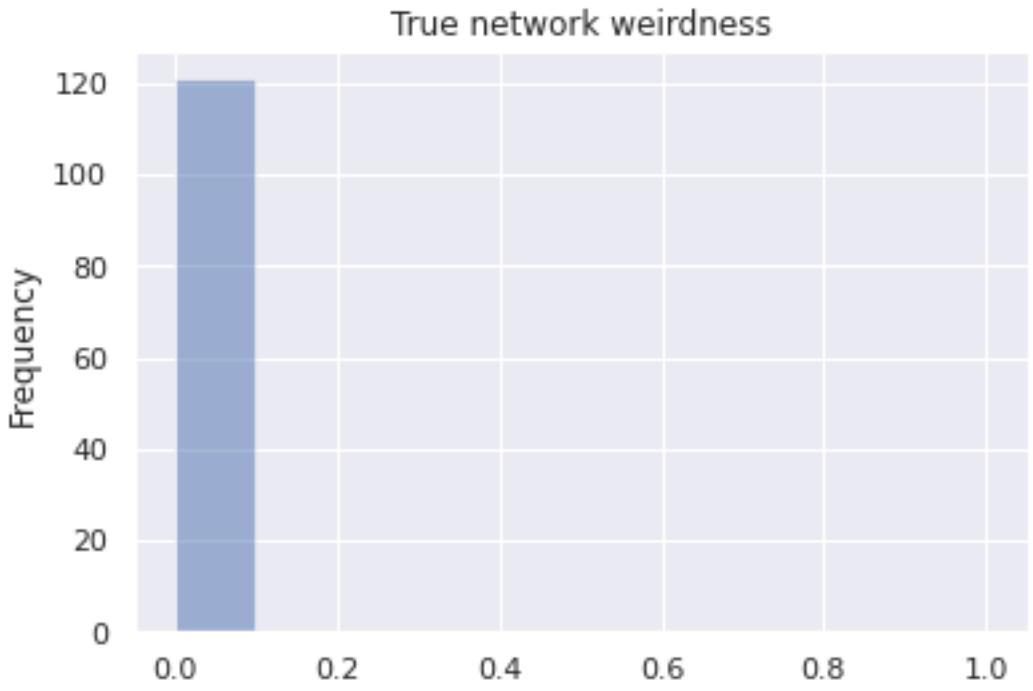
[12]:
```
df_raxml_only_msasize_200_average = df_raxml_only_msasize_200.
 ↪query('likelihood_type == "AVERAGE"')
build_stats(df_raxml_only_msasize_200_average)
```

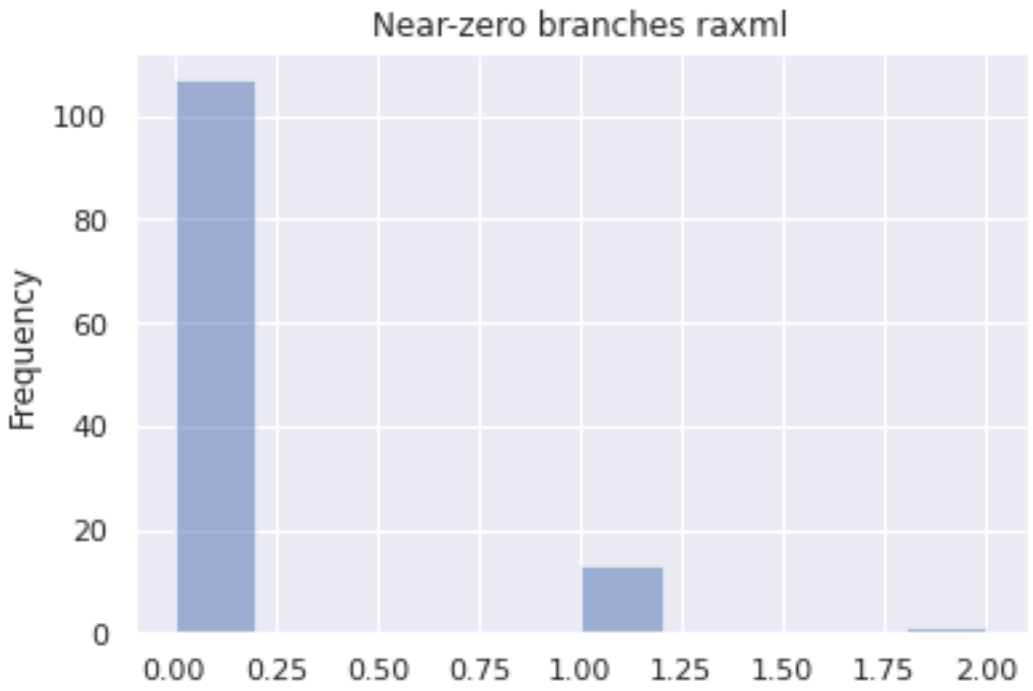
```
Inferred BIC better or equal: 109
Inferred BIC worse: 12

Inferred n_reticulations less: 104
Inferred n_reticulations equal: 17
Inferred n_reticulations more: 0
```

```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```






```
<Figure size 432x288 with 0 Axes>
```

1.2.2 Plots for LikelihoodType.BEST

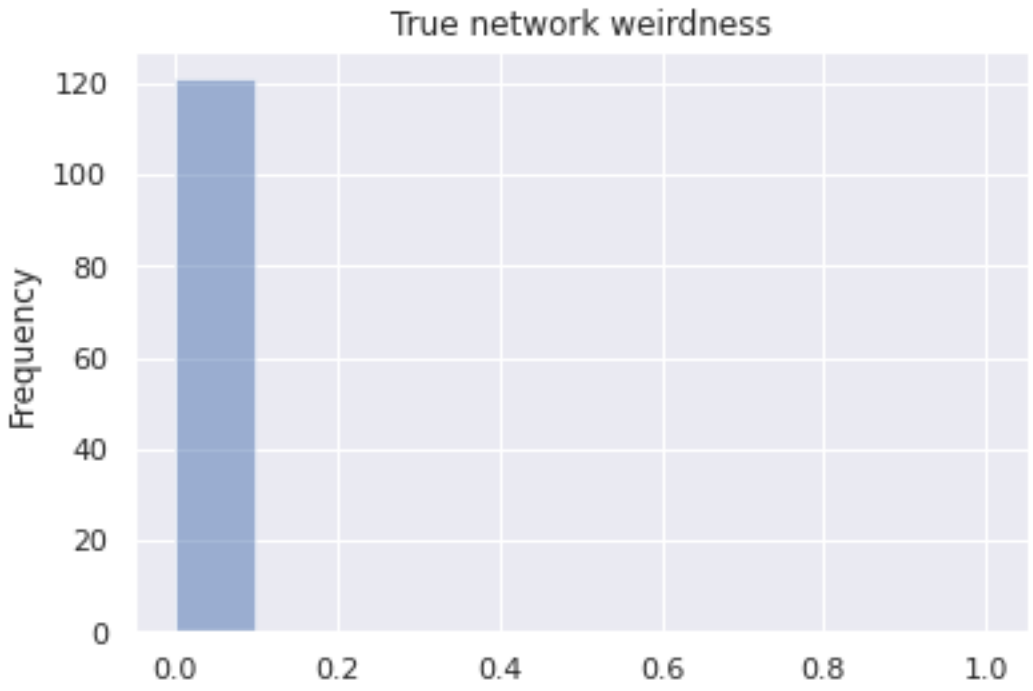
[13]:
```
df_raxml_only_msasize_200_best = df_raxml_only_msasize_200.
 ↪query('likelihood_type == "BEST"')
build_stats(df_raxml_only_msasize_200_best)
```

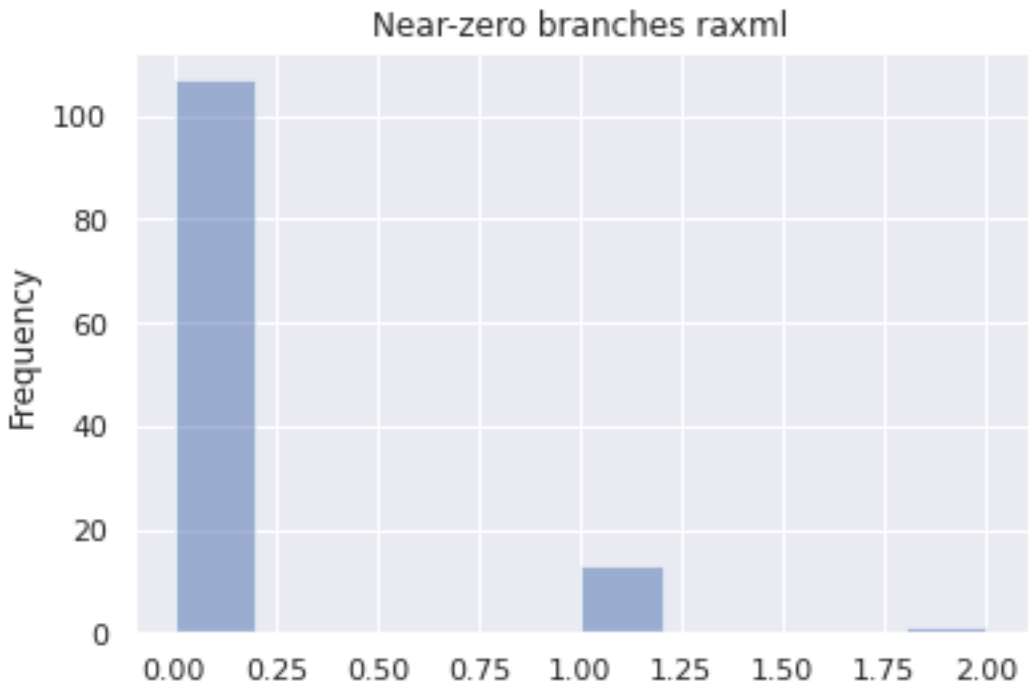
```
Inferred BIC better or equal: 107
Inferred BIC worse: 14

Inferred n_reticulations less: 104
Inferred n_reticulations equal: 17
Inferred n_reticulations more: 0
```

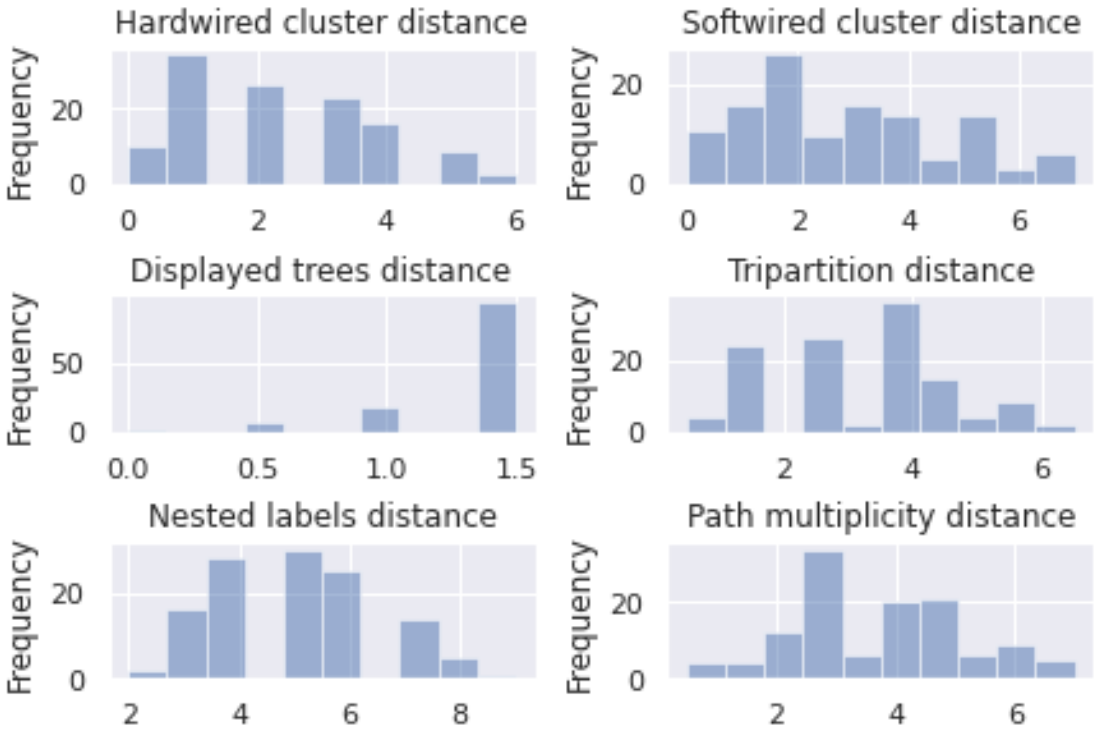
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```






```
<Figure size 432x288 with 0 Axes>
```



2 Plots for starting with 5 random, 5 parsimony trees

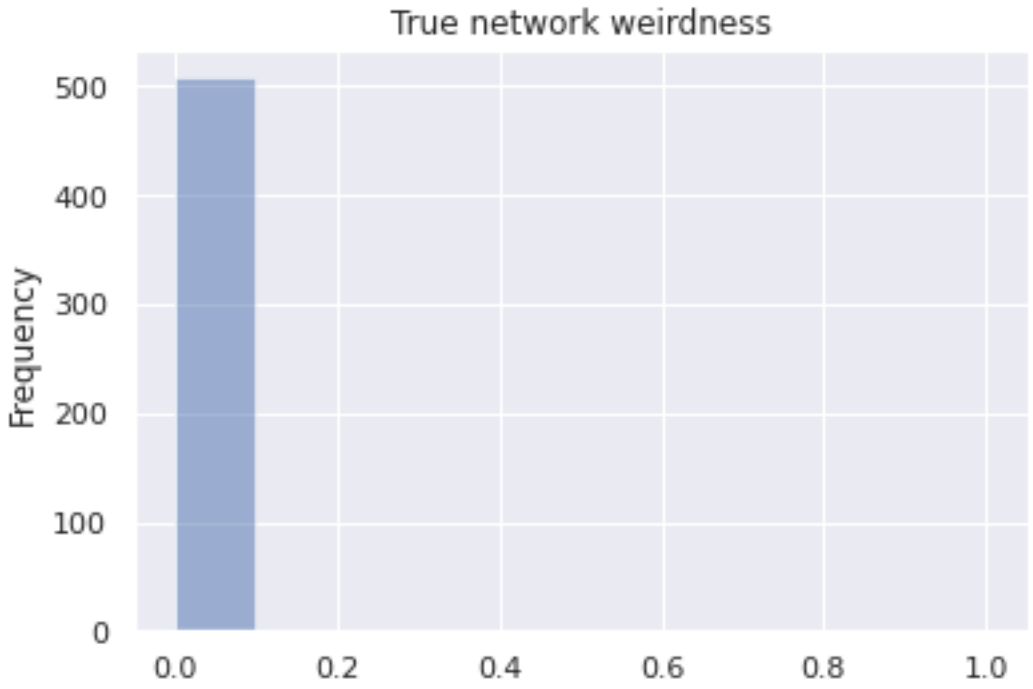
[14]:
```
df_random = df.query('start_type == "RANDOM"')
build_stats(df_random)
```

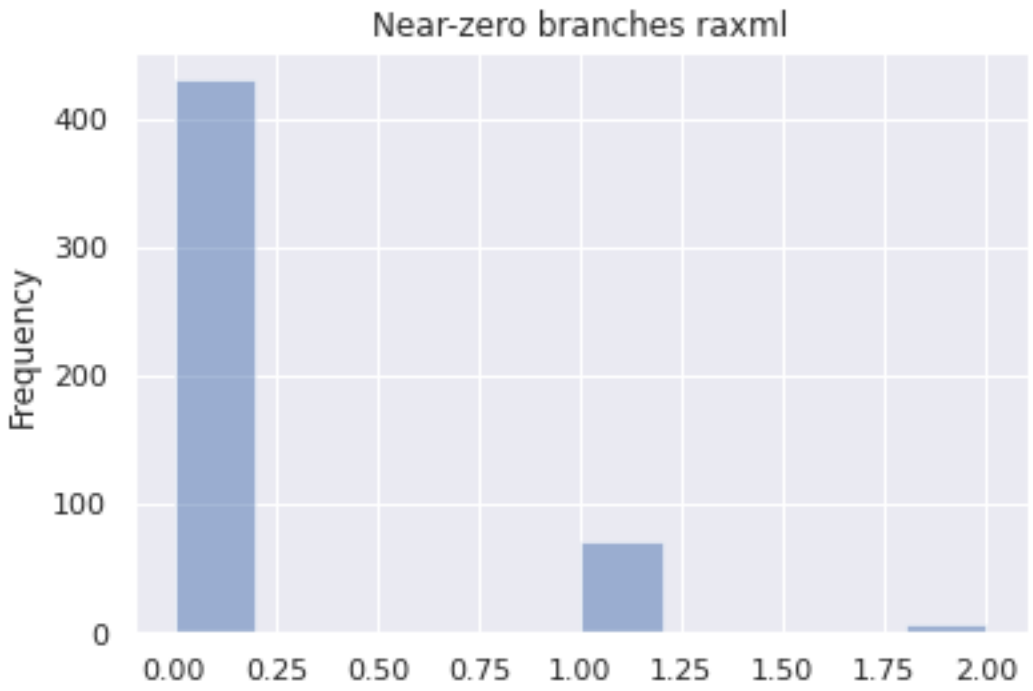
```
Inferred BIC better or equal: 503
Inferred BIC worse: 5

Inferred n_reticulations less: 450
Inferred n_reticulations equal: 58
Inferred n_reticulations more: 0
```

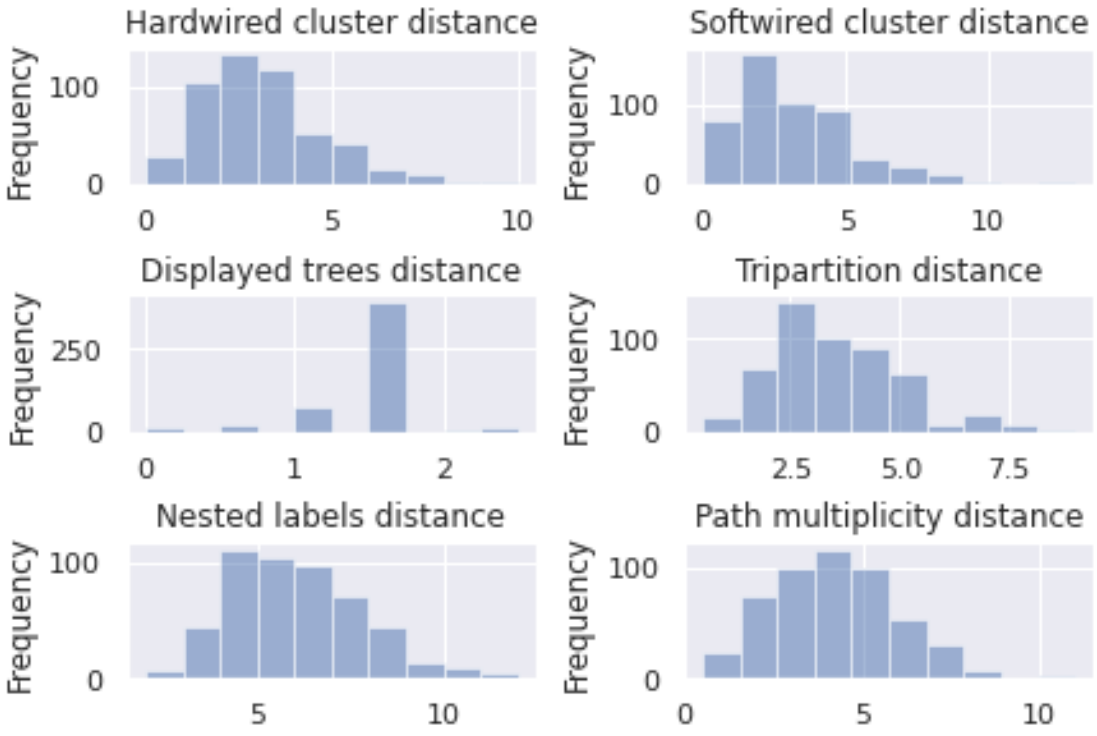
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```






```
<Figure size 432x288 with 0 Axes>
```



2.1 Plots for MSA_size ~ 100*n_trees

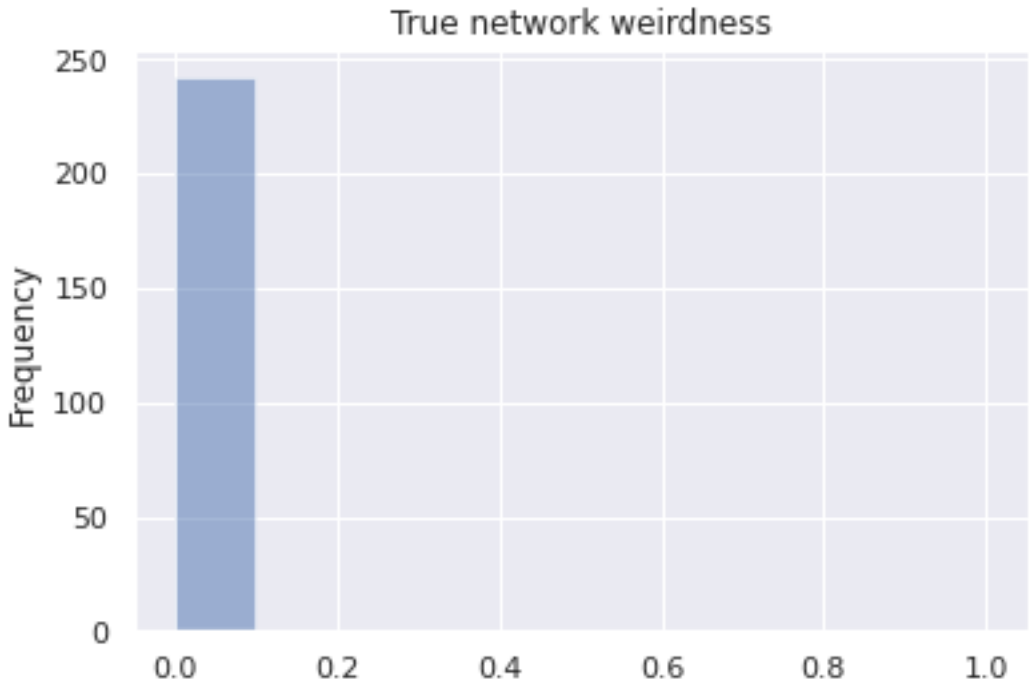
[15]:
```
df_random_msasize_100 = df_random.query('msa_size == 101')
build_stats(df_random_msasize_100)
```

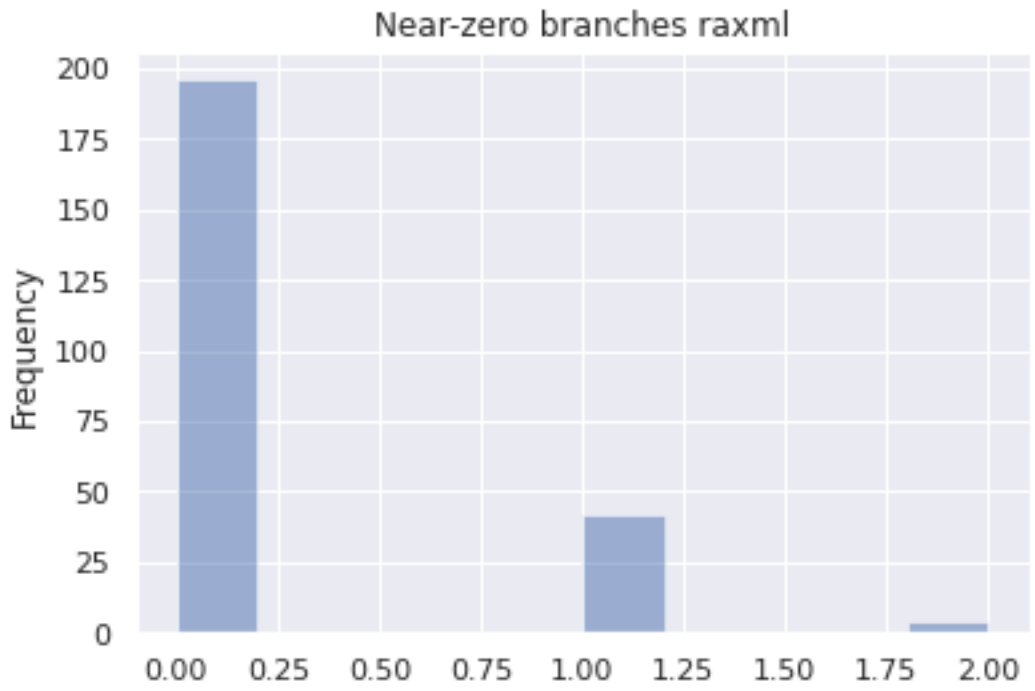
```
Inferred BIC better or equal: 240
Inferred BIC worse: 2

Inferred n_reticulations less: 226
Inferred n_reticulations equal: 16
Inferred n_reticulations more: 0
```

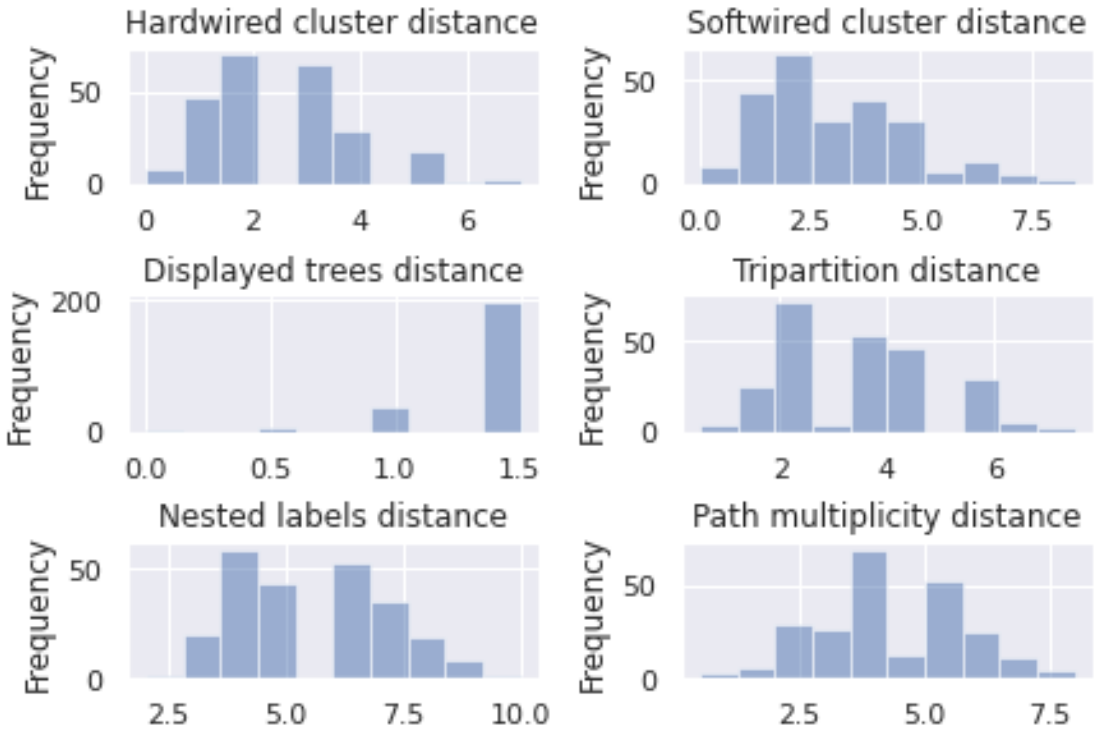
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```





<Figure size 432x288 with 0 Axes>



2.1.1 Plots for LikelihodType.AVERAGE

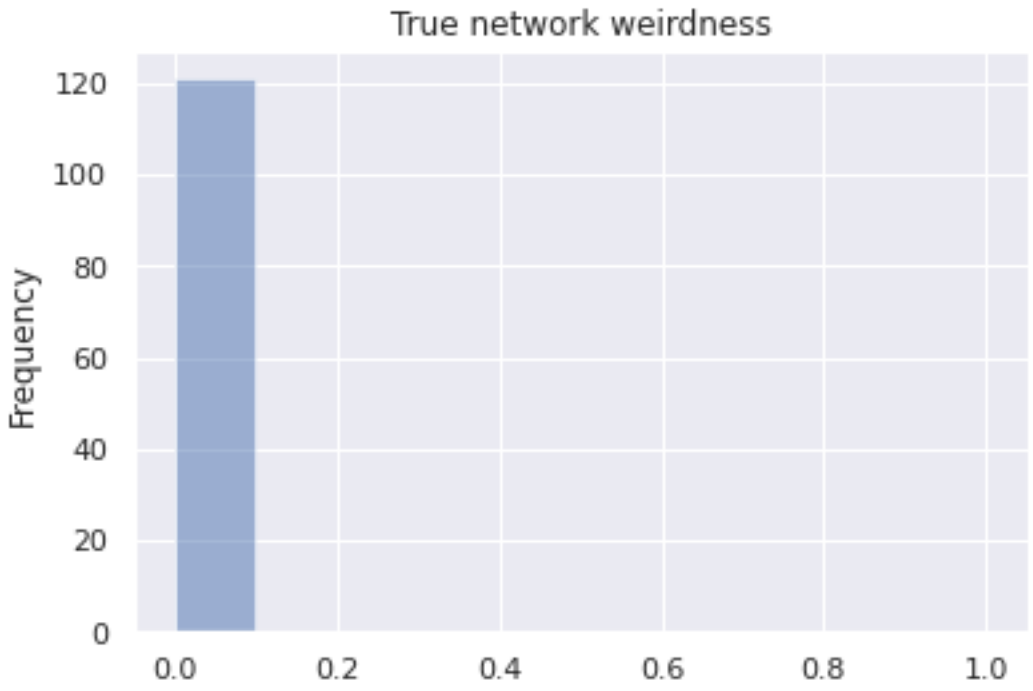
[16]:
```
df_random_msasize_100_average = df_random_msasize_100.query('likelihood_type ==␣
↪"AVERAGE"')
build_stats(df_random_msasize_100_average)
```

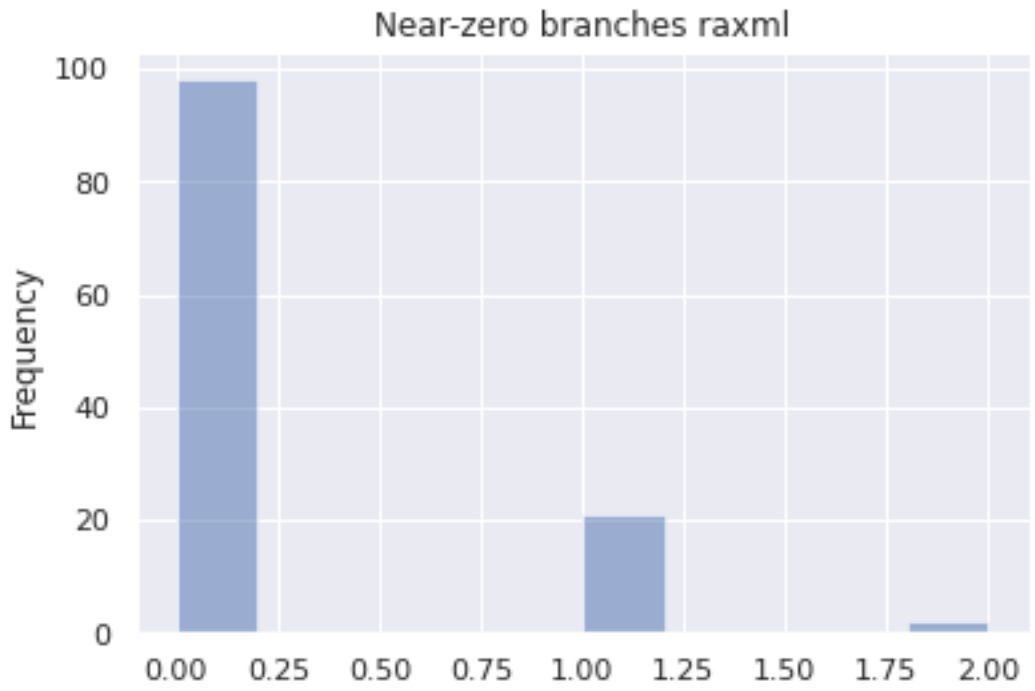
```
Inferred BIC better or equal: 120
Inferred BIC worse: 1

Inferred n_reticulations less: 113
Inferred n_reticulations equal: 8
Inferred n_reticulations more: 0
```

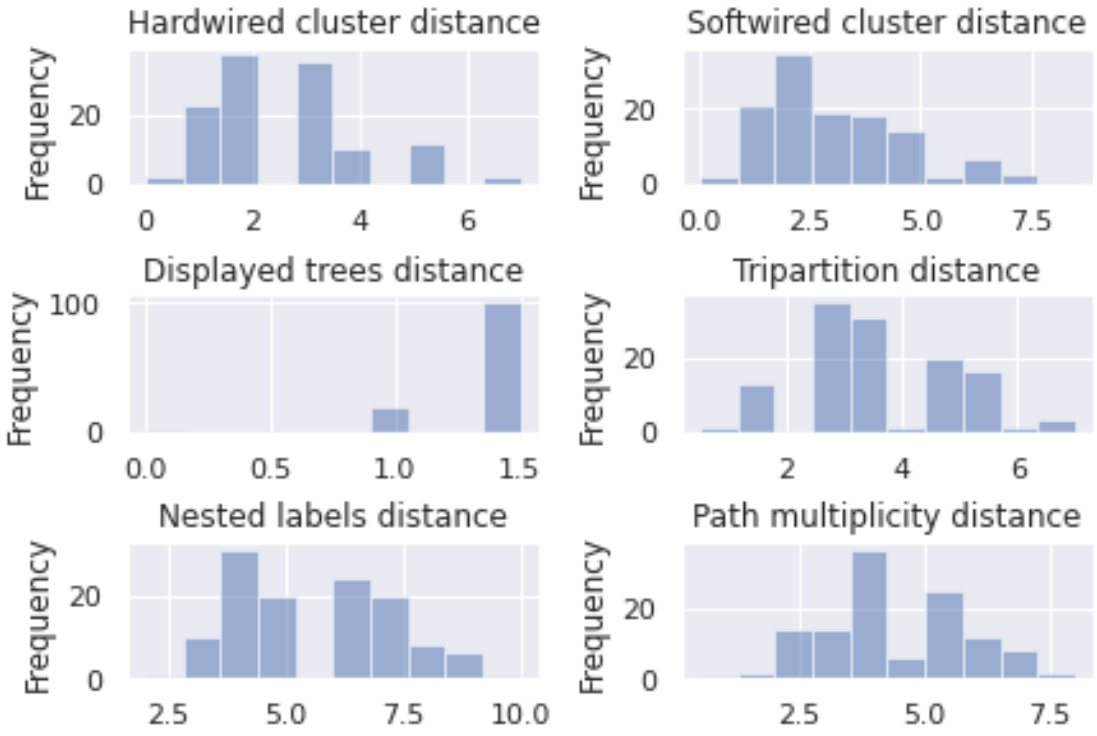
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```





<Figure size 432x288 with 0 Axes>



2.1.2 Plots for LikelihoodType.BEST

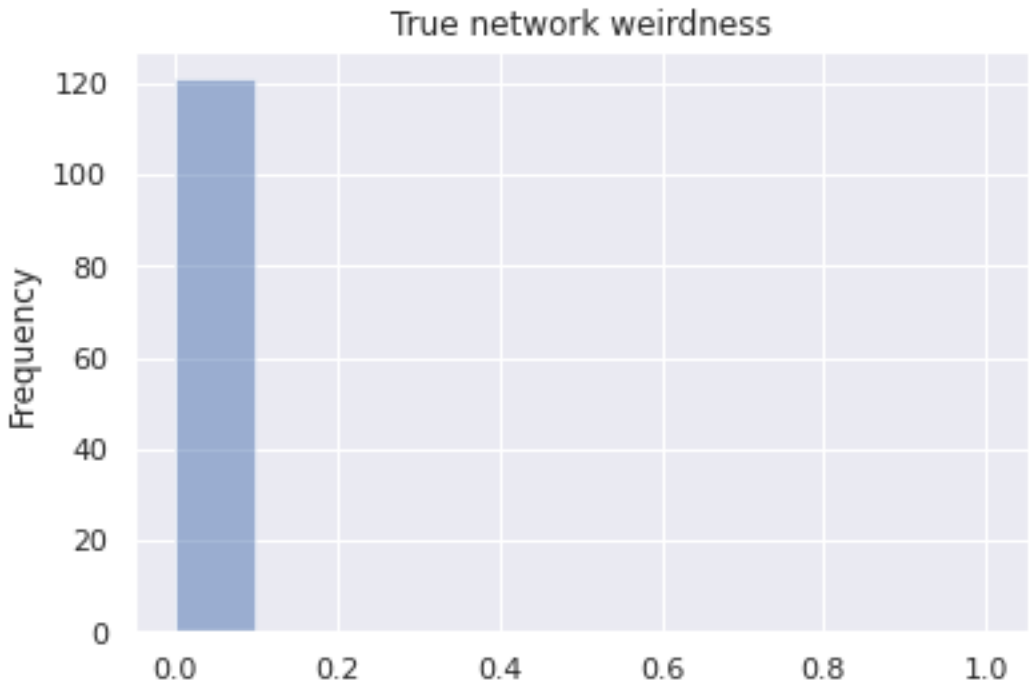
[17]:
```
df_random_msasize_100_best = df_random_msasize_100.query('likelihood_type ==␣
↪"BEST"')
build_stats(df_random_msasize_100_best)
```

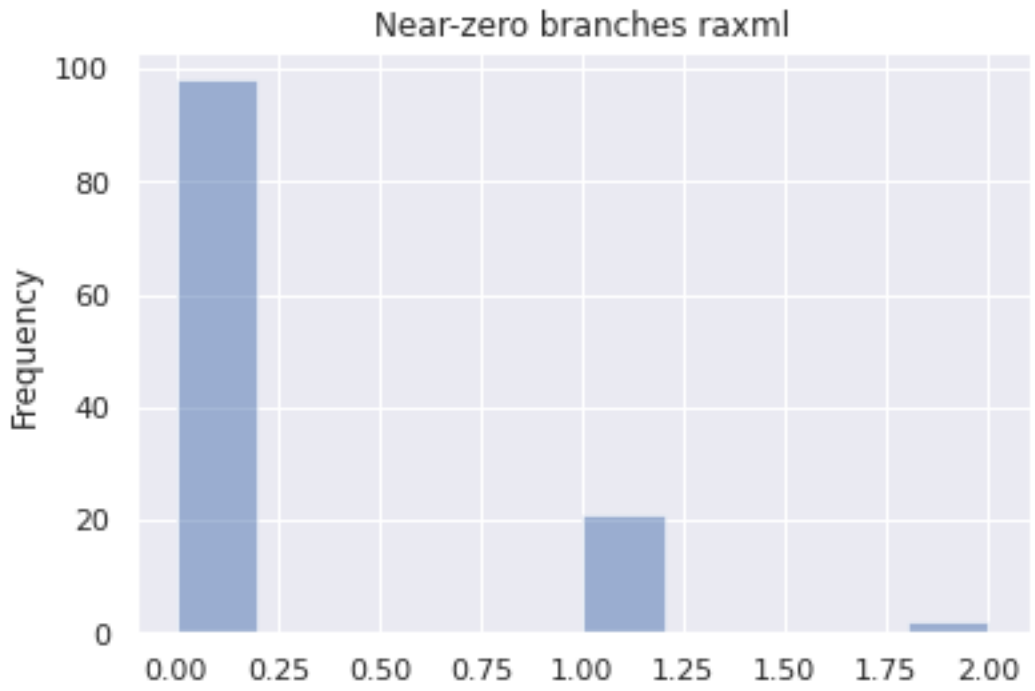
```
Inferred BIC better or equal: 120
Inferred BIC worse: 1

Inferred n_reticulations less: 113
Inferred n_reticulations equal: 8
Inferred n_reticulations more: 0
```

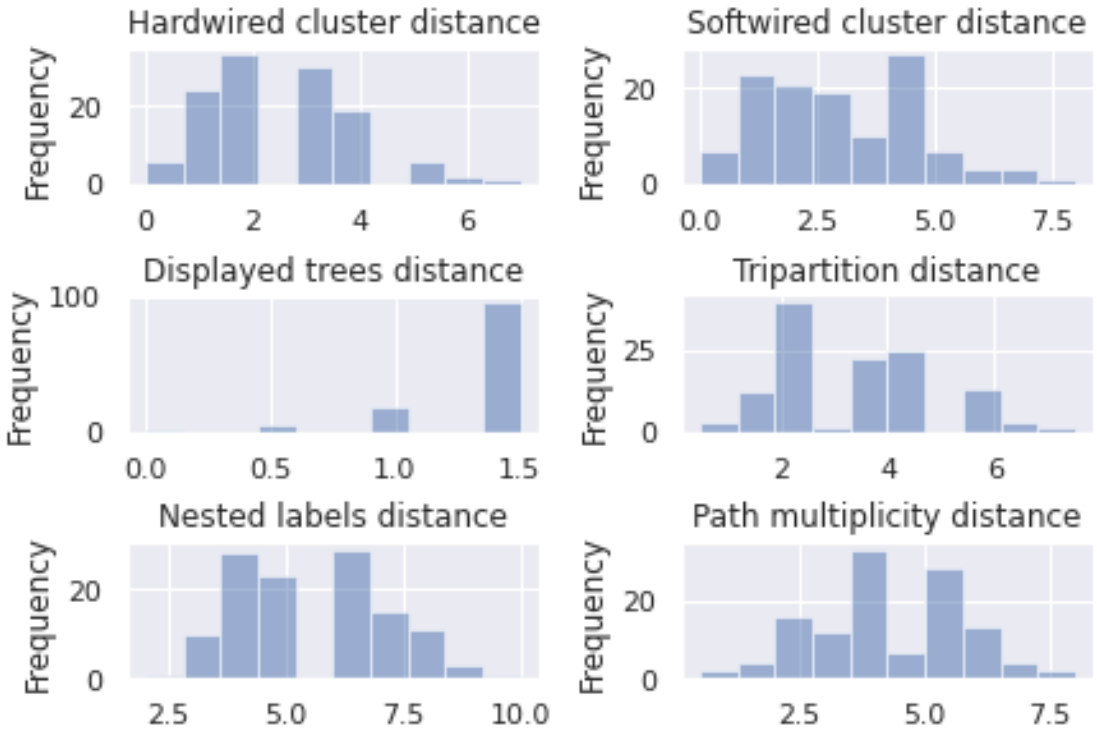
```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```





<Figure size 432x288 with 0 Axes>



2.2 Plots for MSA_size ~ 200*n_trees

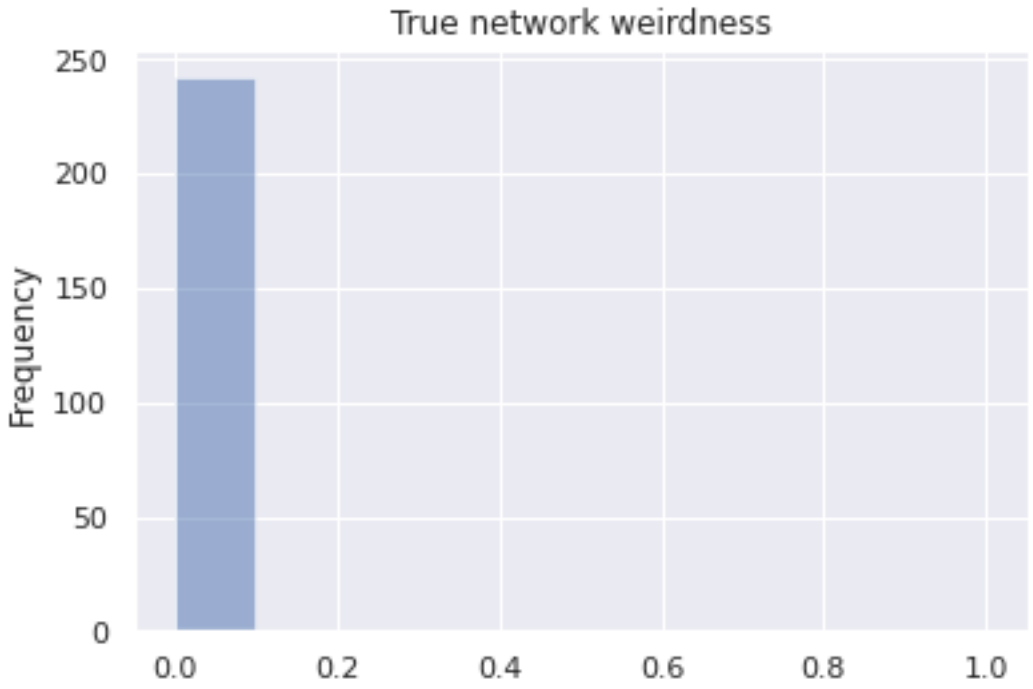
[ ]:
```
df_random_msasize_200 = df_random.query('msa_size == 201')
build_stats(df_random_msasize_200)
```

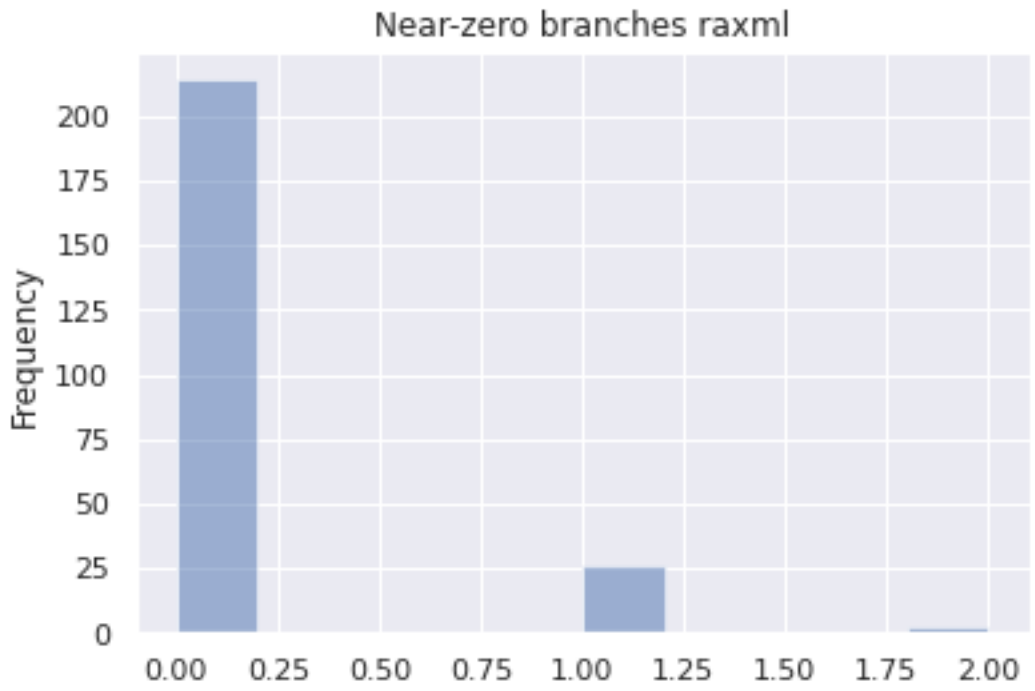
```
Inferred BIC better or equal: 239
Inferred BIC worse: 3

Inferred n_reticulations less: 200
Inferred n_reticulations equal: 42
Inferred n_reticulations more: 0
```

```
<Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```






```
<Figure size 432x288 with 0 Axes>
```

2.2.1 Plots for LikelihodType.AVERAGE

[ ]:
```
df_random_msasize_200_average = df_random_msasize_200.query('likelihood_type ==␣
↪"AVERAGE"')
build_stats(df_random_msasize_200_average)
```

2.2.2 Plots for LikelihoodType.BEST

[ ]:
```
df_random_msasize_200_best = df_random_msasize_200.query('likelihood_type ==␣
↪"BEST"')
build_stats(df_random_msasize_200_best)
```

[ ]:
```

```