

NetRAX Experiment Evaluation

February 10, 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']]))))
    df['bic_diff'].plot.hist(bins=100, alpha=0.5, title='(bic_true -
    ↳bic_inferred) / bic_true\n value >0 means inferred BIC was better')

def logl_stats(df):
    print("Inferred loglh better or equal: " + str(len(df[df['logl_inferred']
    ↳>= df['logl_true']]))))
    print("Inferred loglh worse: " + str(len(df[df['logl_inferred'] <
    ↳df['logl_true']]))))
    df['logl_diff'].plot.hist(bins=100, alpha=0.5, title='(logl_true -
    ↳logl_inferred) / logl_true\n value <0 means inferred logl was better')

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)
    logl_stats(df)
    print("")
    plt.figure(2)
    reticulation_stats(df)
    print("")
    plt.figure(3)
    weirdness_stats(df)
    print("")
    plt.figure(4)
    zero_branches_stats(df)
    print("")
    plt.figure(5)
    distances(df)

```

Load the result CSV:

```

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

df['bic_diff'] = (df['bic_true'] - df['bic_inferred']) / df['bic_true']
df['logl_diff'] = (df['logl_true'] - df['logl_inferred']) / df['logl_true']

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

```

[4]:

```

                                name  n_taxa  n_trees  \
0  datasets_medium_network_norandom_0_0/0_22_taxa...      22      4
1  datasets_medium_network_norandom_0_0/0_22_taxa...      22      4
2  datasets_medium_network_norandom_0_0/0_22_taxa...      22      4
3  datasets_medium_network_norandom_0_0/0_22_taxa...      22      4
4  datasets_medium_network_norandom_0_1/0_17_taxa...      17      2

n_reticulations  msa_size      sampling_type  simulation_type  \
0                2        202  PERFECT_SAMPLING      CELINE
1                2        202  PERFECT_SAMPLING      CELINE
2                2        402  PERFECT_SAMPLING      CELINE
3                2        402  PERFECT_SAMPLING      CELINE
4                1        101  PERFECT_SAMPLING      CELINE

                                celine_params  \
0  {'to': 0.16223186561955155| 'lambda': 21.49262...
1  {'to': 0.16223186561955155| 'lambda': 21.49262...
2  {'to': 0.16223186561955155| 'lambda': 21.49262...
3  {'to': 0.16223186561955155| 'lambda': 21.49262...
4  {'to': 0.10860342742632532| 'lambda': 21.35962...

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

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_medium_network_norandom_0_0/0_22_taxa...
1  datasets_medium_network_norandom_0_0/0_22_taxa...
2  datasets_medium_network_norandom_0_0/0_22_taxa...
3  datasets_medium_network_norandom_0_0/0_22_taxa...
4  datasets_medium_network_norandom_0_1/0_17_taxa...

                                inferred_network_path  likelihood_type  \
0  datasets_medium_network_norandom_0_0/0_22_taxa...      AVERAGE
1  datasets_medium_network_norandom_0_0/0_22_taxa...      BEST
2  datasets_medium_network_norandom_0_0/0_22_taxa...      AVERAGE
3  datasets_medium_network_norandom_0_0/0_22_taxa...      BEST
```

```

4 datasets_medium_network_norandom_0_1/0_17_taxa... AVERAGE

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

n_parsimony_start_networks runtime_inference n_reticulations_inferred \
0 0 6334.250 0
1 0 1150.248 0
2 0 3626.096 0
3 0 869.273 0
4 0 4660.722 1

bic_true logl_true bic_inferred logl_inferred bic_raxml \
0 3357.579938 -1317.619632 3291.448158 -1318.150983 3291.448158
1 3358.780892 -1318.220110 3291.448158 -1318.150983 3291.448158
2 6259.305638 -2738.890554 6238.391973 -2764.783699 6238.401624
3 6260.156175 -2739.315822 6238.391973 -2764.783699 6238.401624
4 1171.233082 -384.511527 1162.979859 -380.384915 1165.501221

logl_raxml hardwired_cluster_distance softwired_cluster_distance \
0 -1318.150983 7.0 5.0
1 -1318.150983 7.0 5.0
2 -2764.788525 7.0 6.0
3 -2764.788525 7.0 6.0
4 -396.542264 12.0 14.5

displayed_trees_distance tripartition_distance nested_labels_distance \
0 2.5 9.0 14.0
1 2.5 9.0 14.0
2 2.5 9.0 12.0
3 2.5 9.0 12.0
4 1.5 11.0 14.0

path_multiplicity_distance bic_diff logl_diff
0 11.5 0.019696 -0.000403
1 11.5 0.020047 0.000052
2 9.5 0.003341 -0.009454
3 9.5 0.003477 -0.009297
4 11.5 0.007047 0.010732

```

```
[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',
          'hardwired_cluster_distance', 'softwired_cluster_distance',
          'displayed_trees_distance', 'tripartition_distance',
          'nested_labels_distance', 'path_multiplicity_distance', 'bic_diff',
          'logl_diff'],
          dtype='object')
```

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

```
Inferred BIC better or equal: 415
```

```
Inferred BIC worse: 45
```

```
Inferred loglh better or equal: 195
```

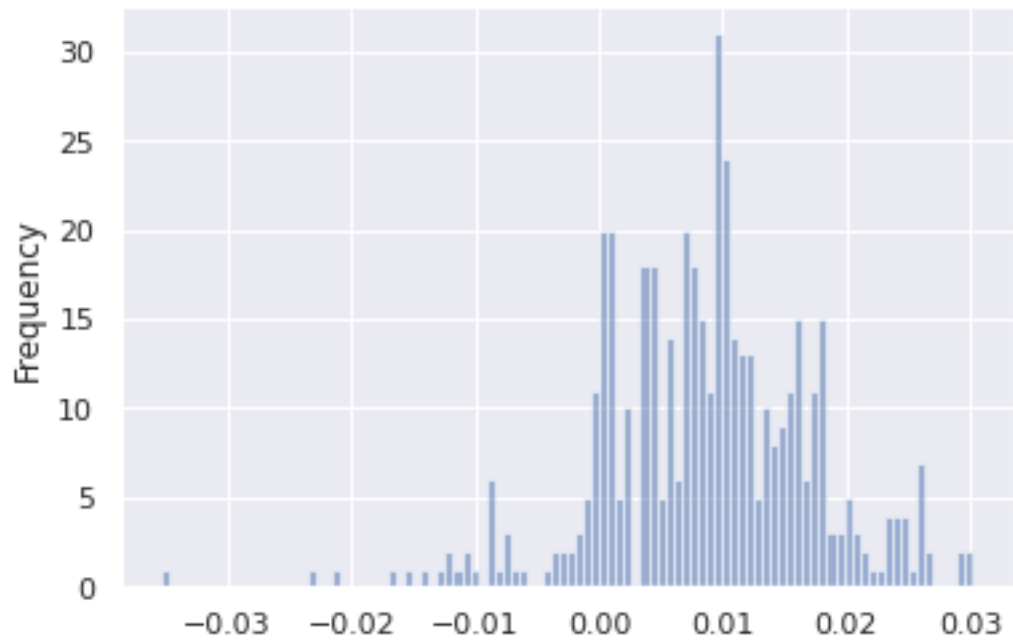
```
Inferred loglh worse: 265
```

```
Inferred n_reticulations less: 413
```

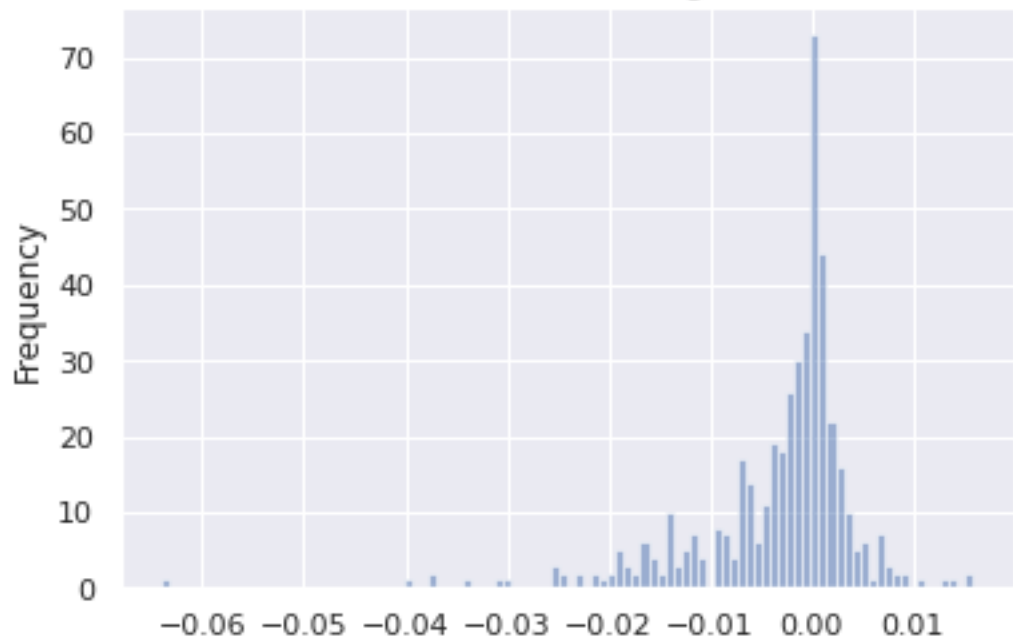
```
Inferred n_reticulations equal: 47
```

```
Inferred n_reticulations more: 0
```

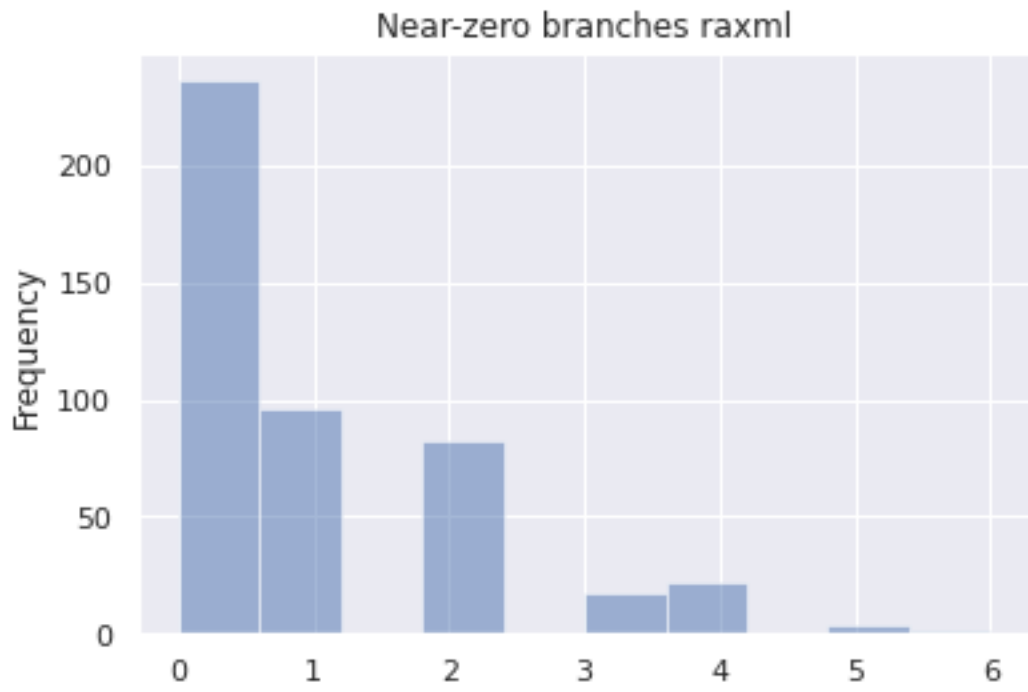
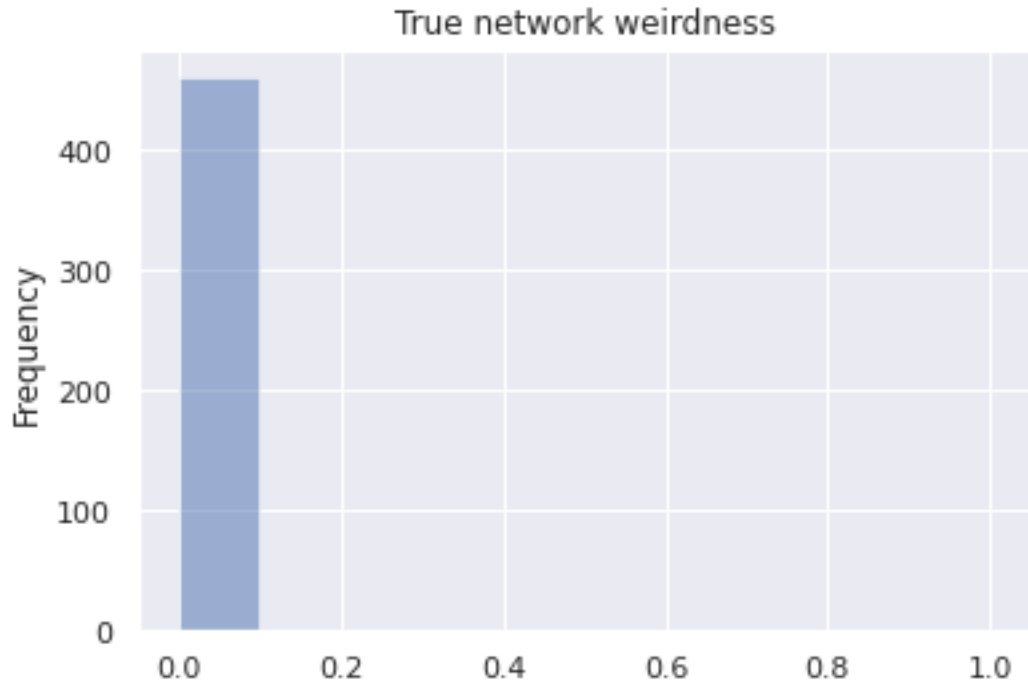
$(bic_true - bic_inferred) / bic_true$
value >0 means inferred BIC was better



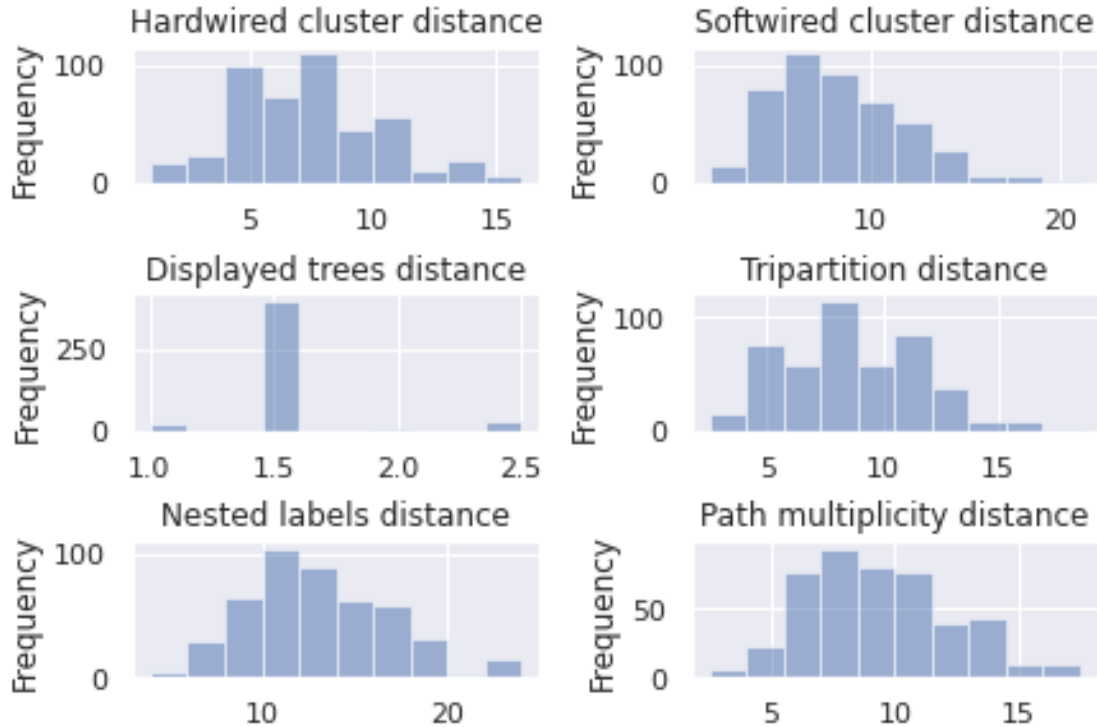
$(logl_true - logl_inferred) / logl_true$
value <0 means inferred logl was better



<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: 415

Inferred BIC worse: 45

Inferred loglh better or equal: 195

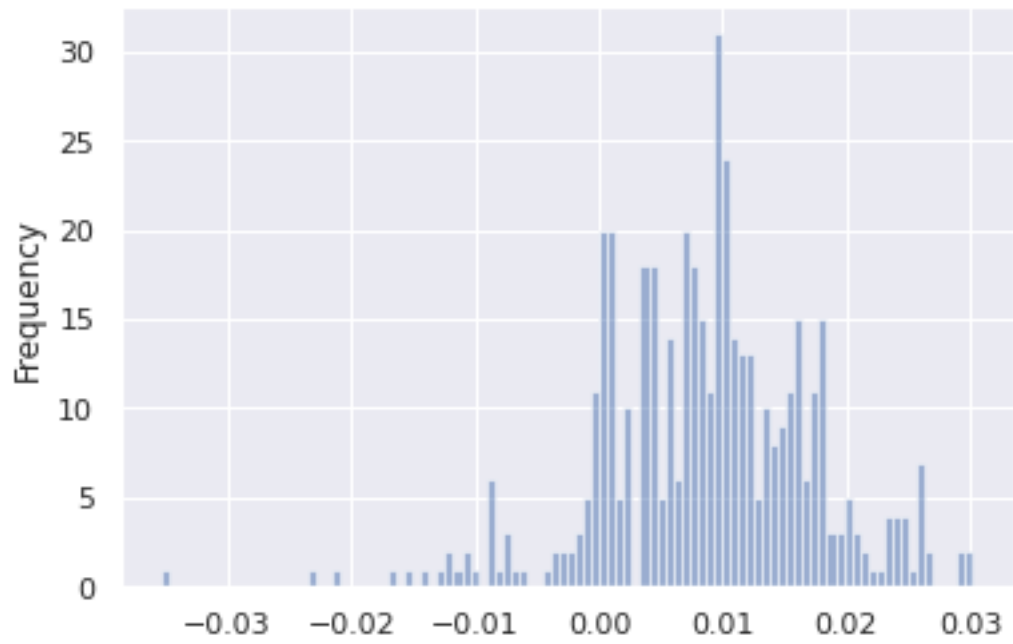
Inferred loglh worse: 265

Inferred n_reticulations less: 413

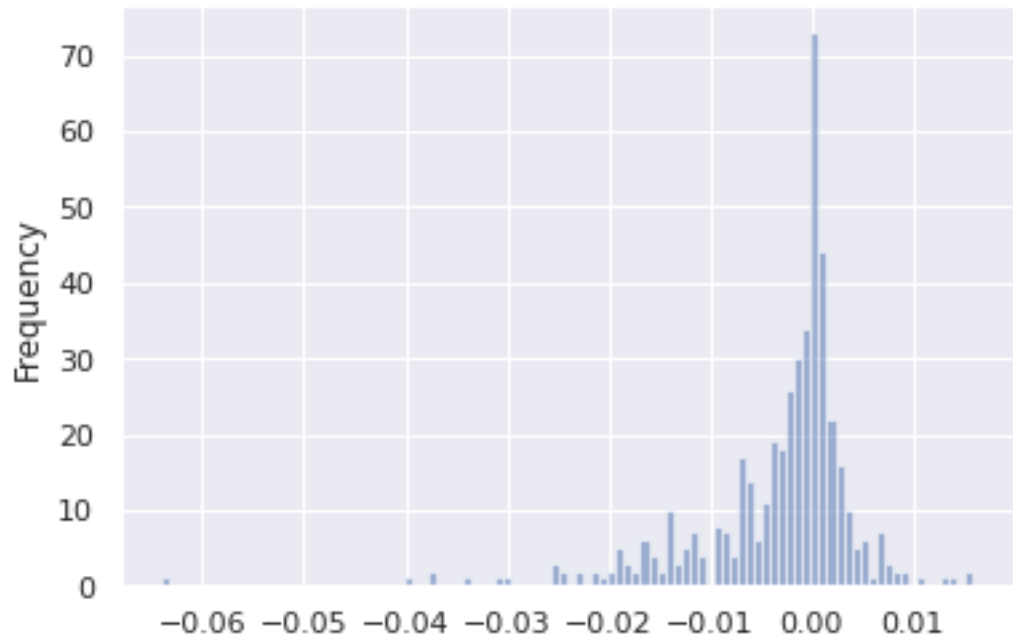
Inferred n_reticulations equal: 47

Inferred n_reticulations more: 0

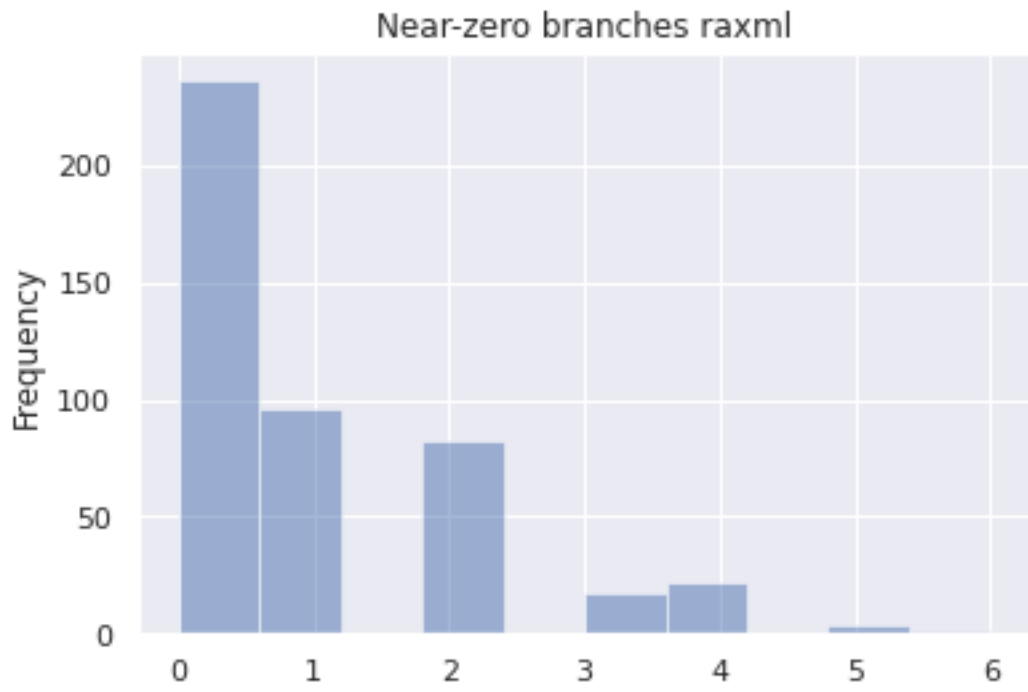
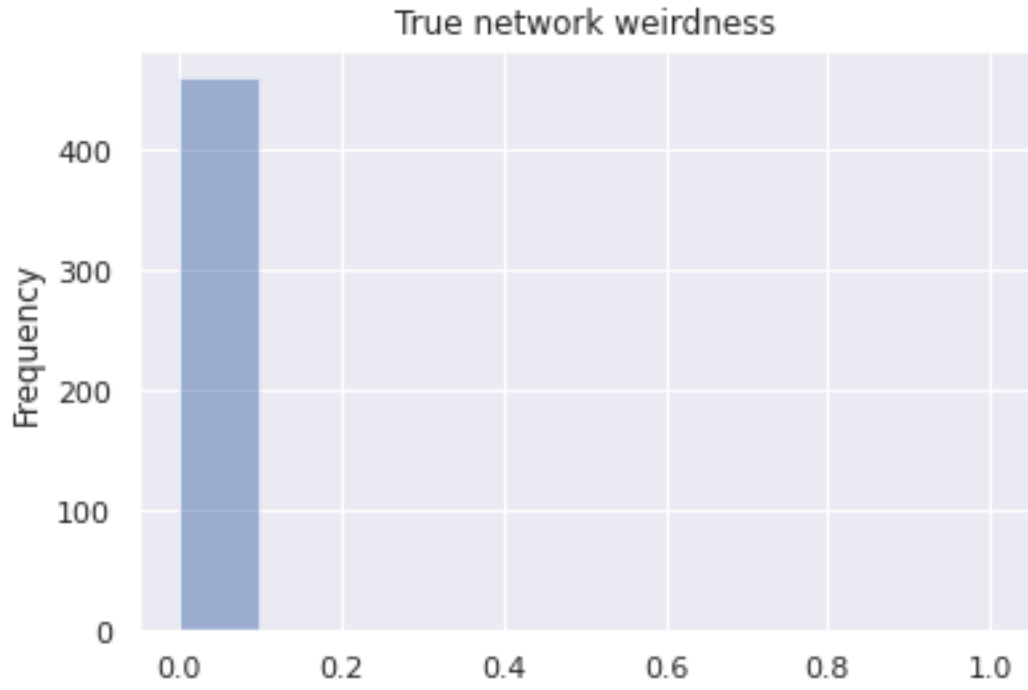
$(bic_true - bic_inferred) / bic_true$
value >0 means inferred BIC was better



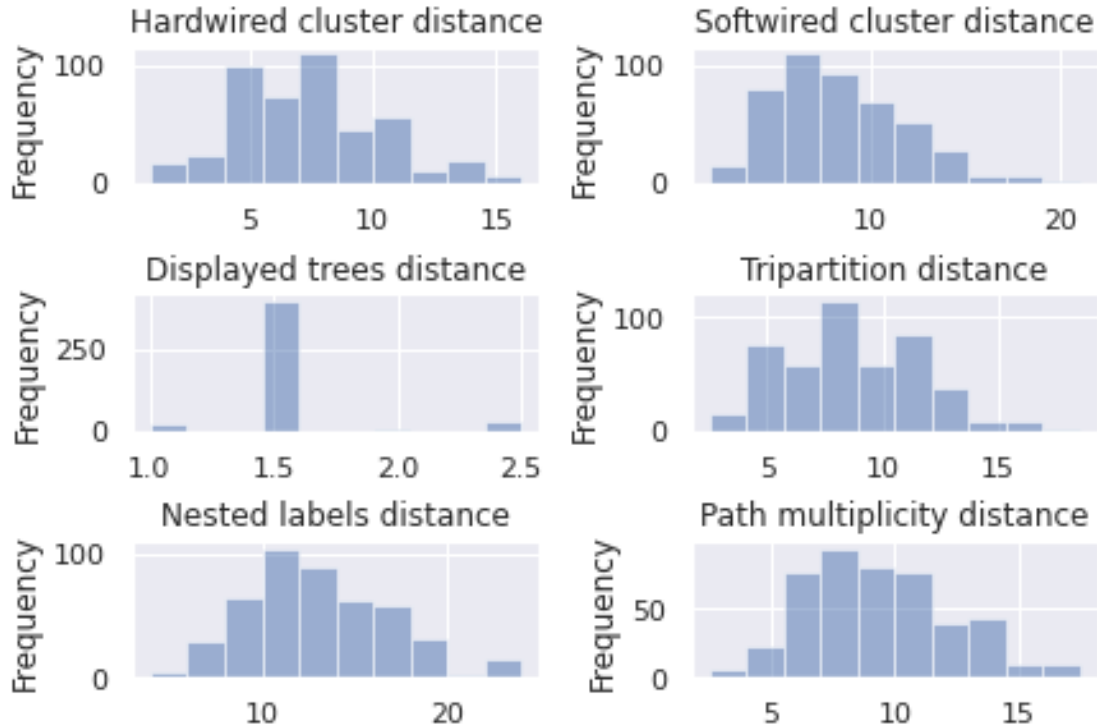
$(logl_true - logl_inferred) / logl_true$
value <0 means inferred logl was better



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



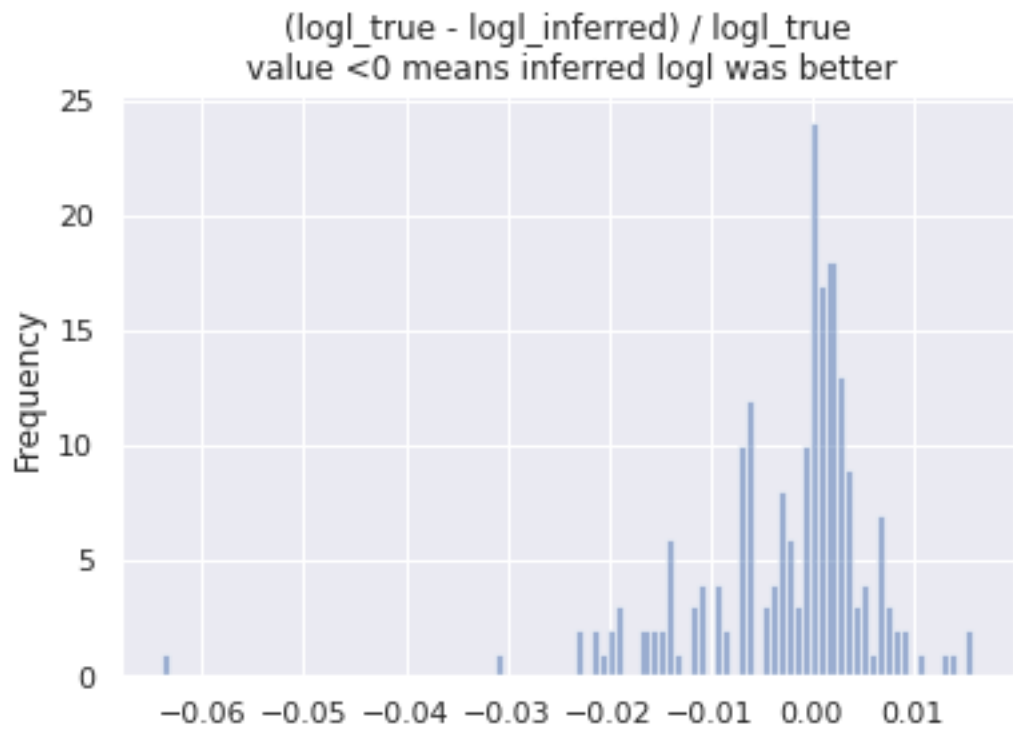
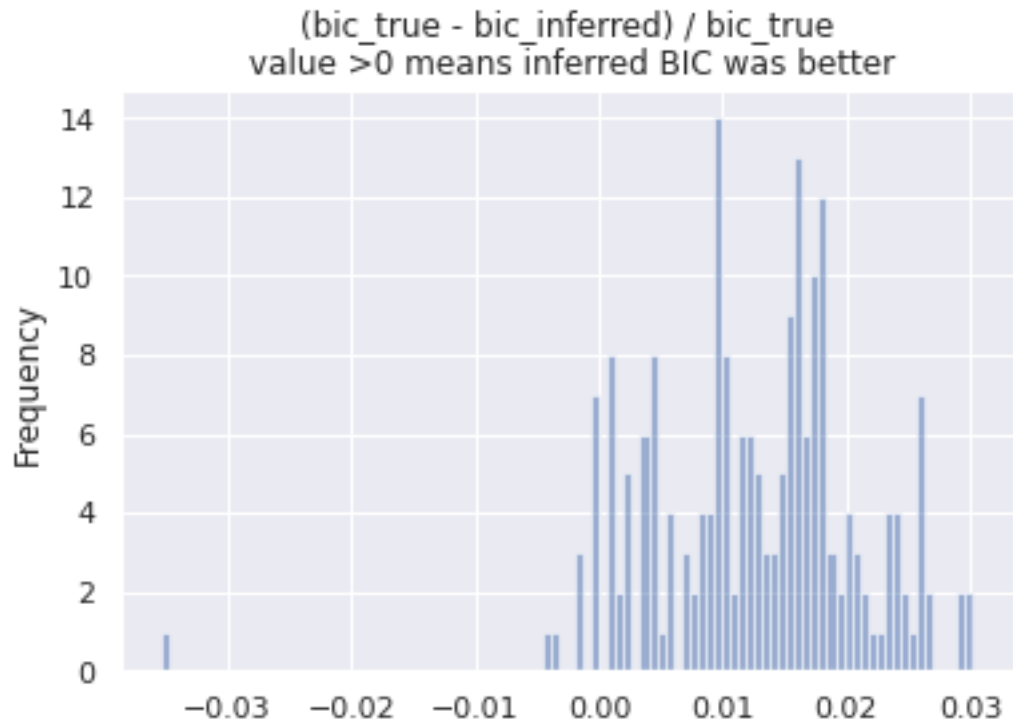
1.1 Plots for MSA_size ~ 100*n_trees

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

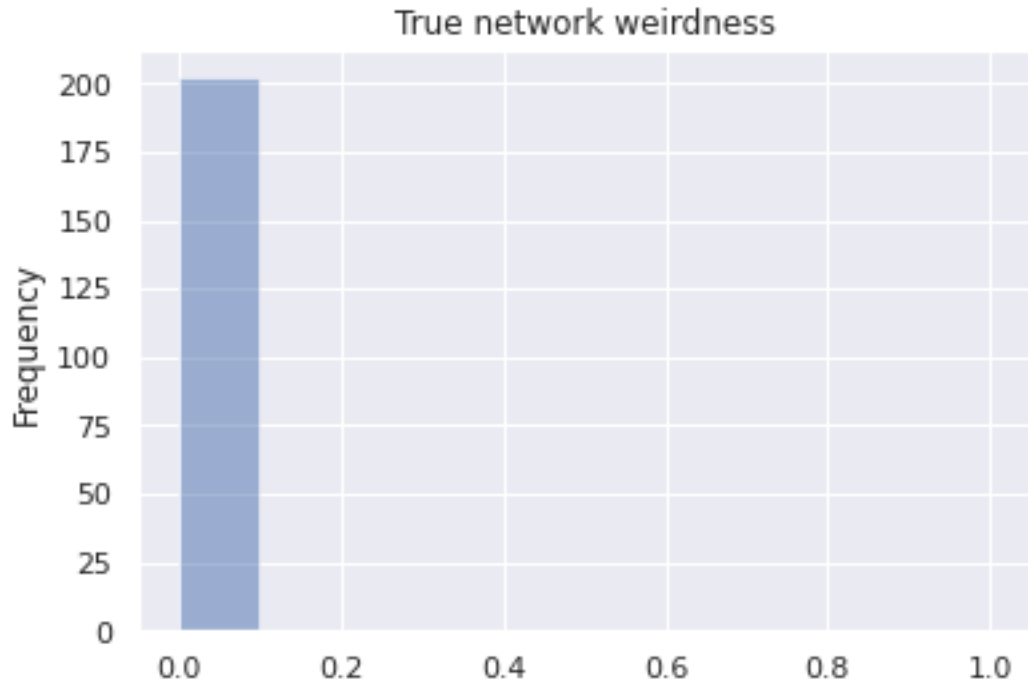
Inferred BIC better or equal: 193
Inferred BIC worse: 9

Inferred loglh better or equal: 107
Inferred loglh worse: 95

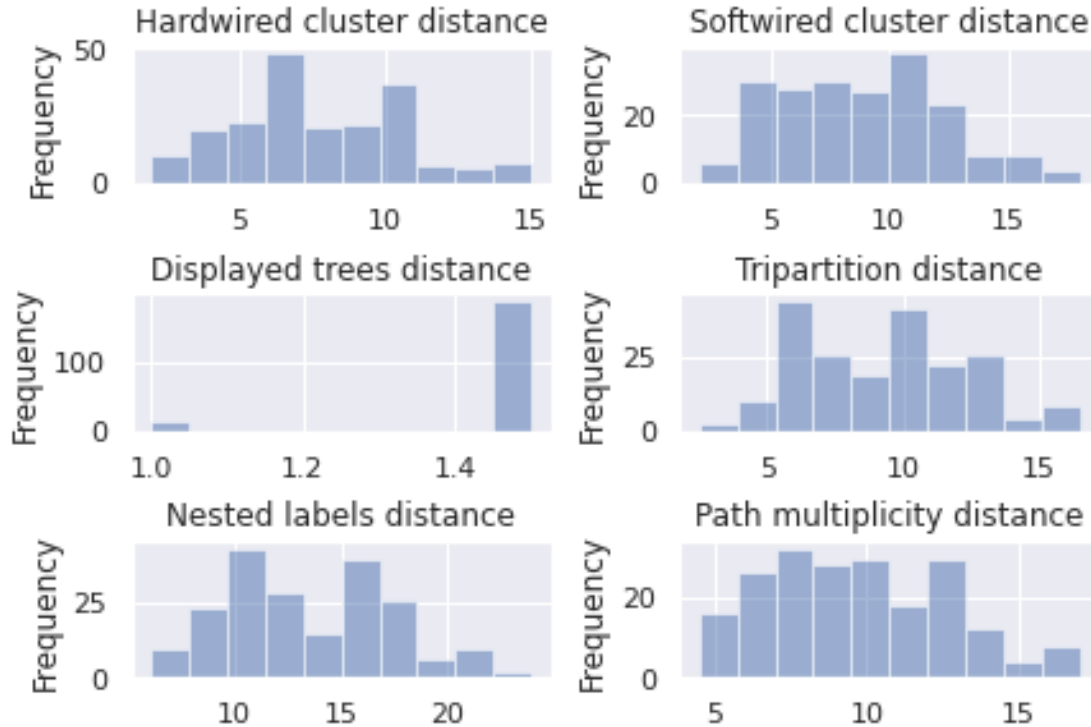
Inferred n_reticulations less: 182
Inferred n_reticulations equal: 20
Inferred n_reticulations more: 0



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



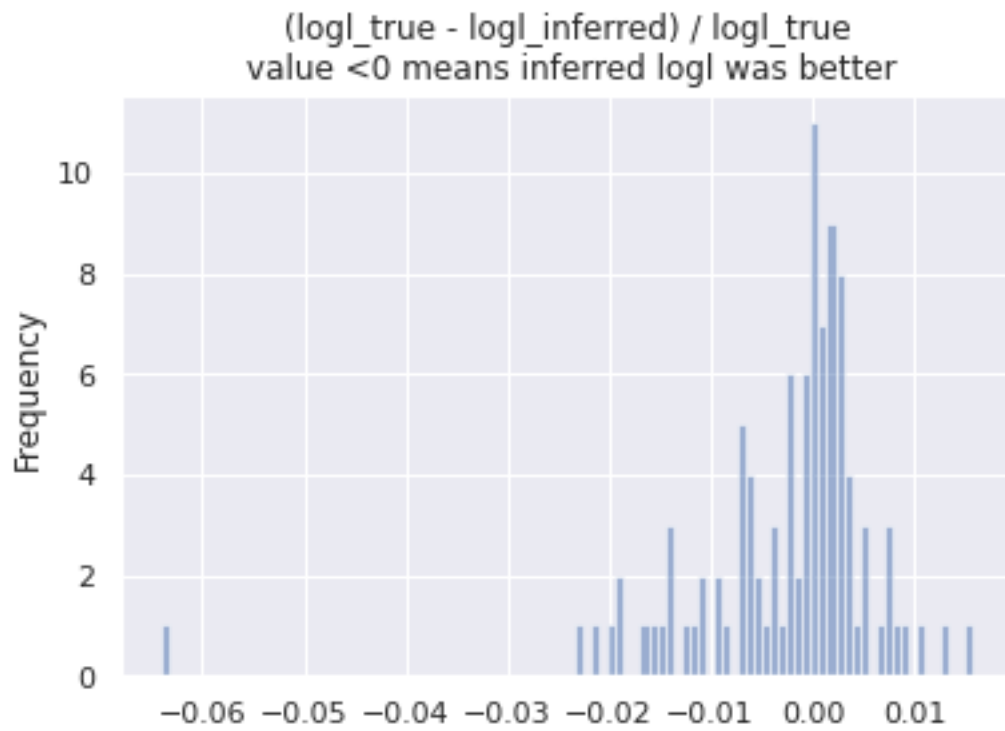
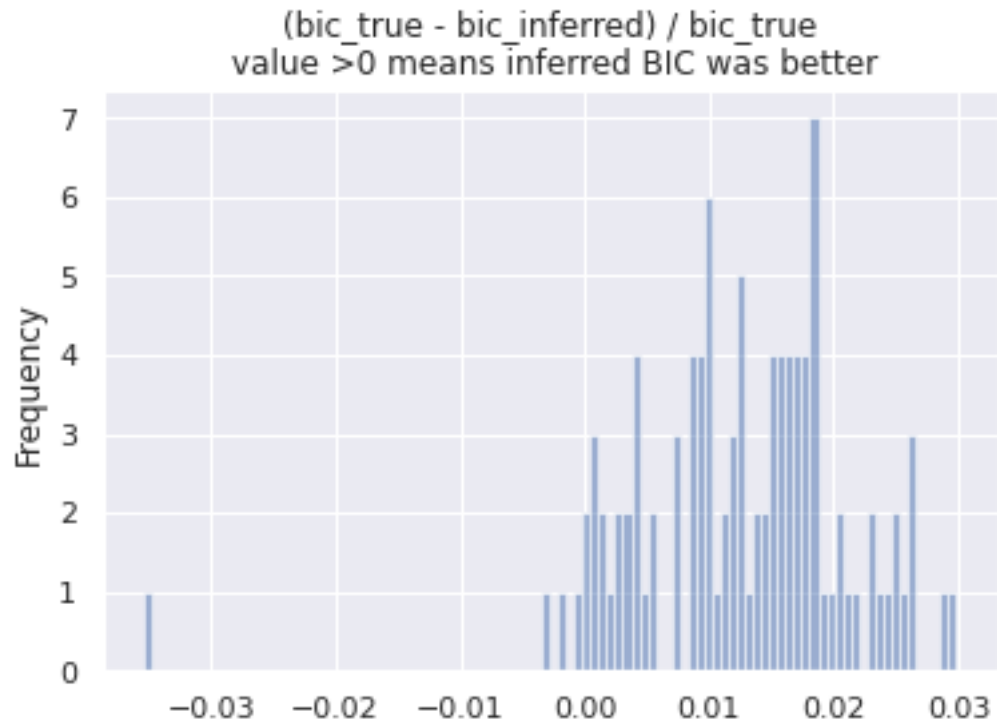
1.1.1 Plots for LikelihoodType.AVERAGE

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

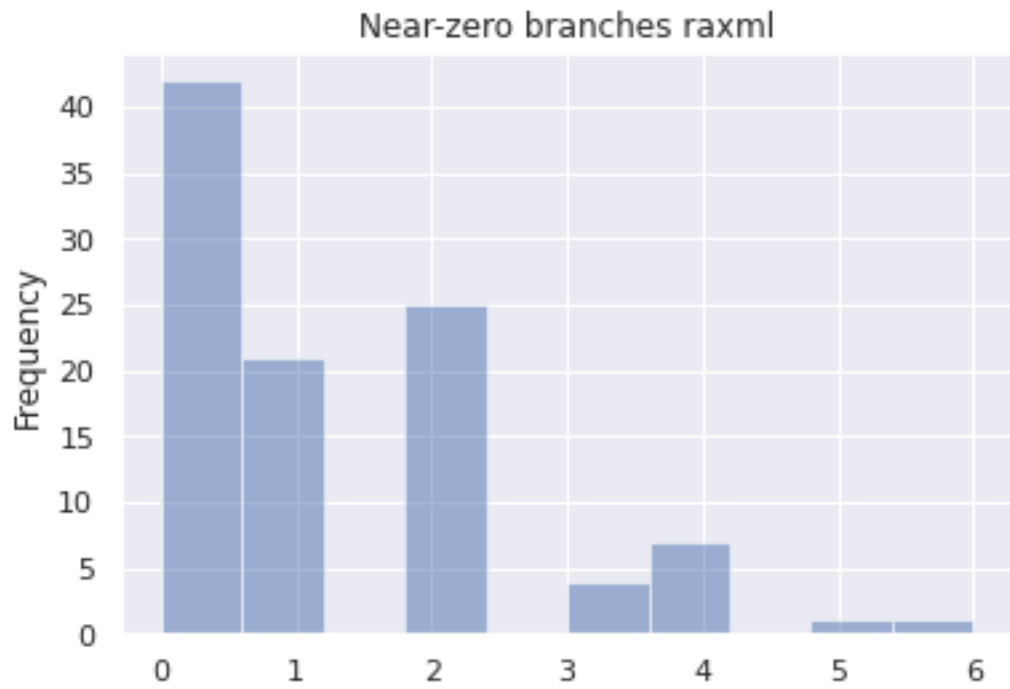
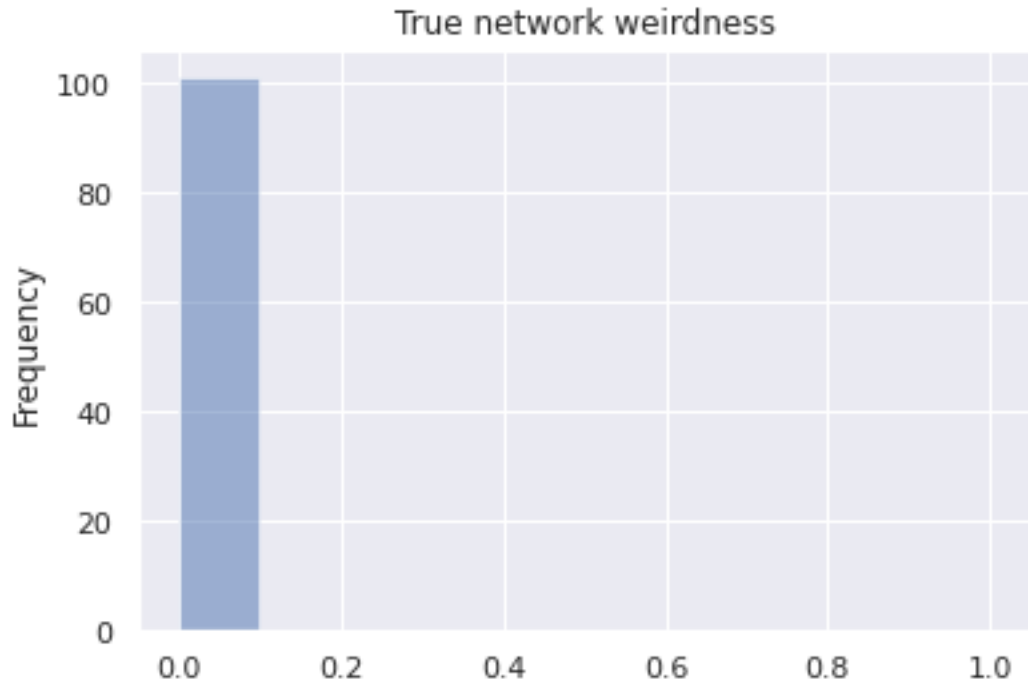
Inferred BIC better or equal: 97
Inferred BIC worse: 4

Inferred loglh better or equal: 52
Inferred loglh worse: 49

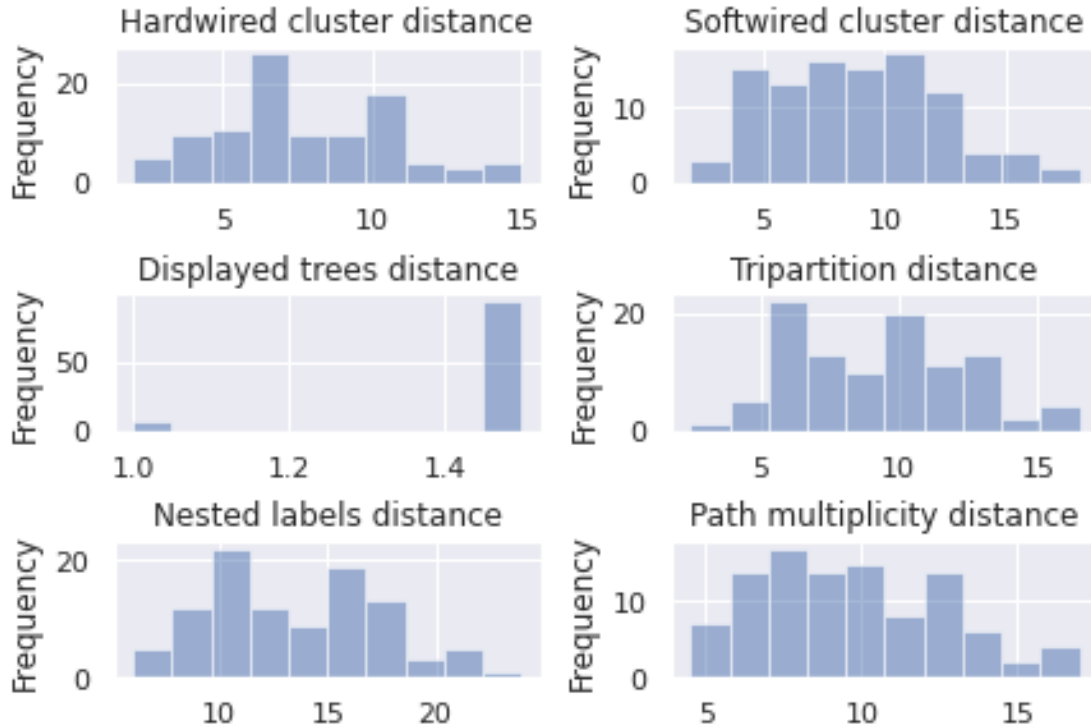
Inferred n_reticulations less: 91
Inferred n_reticulations equal: 10
Inferred n_reticulations more: 0



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



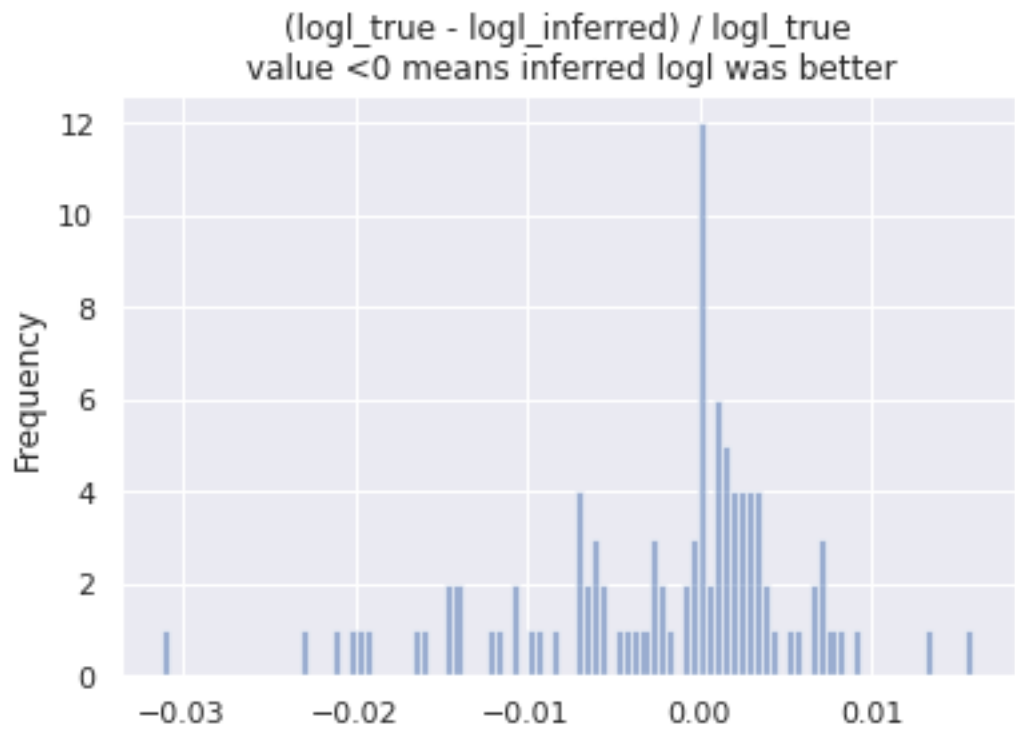
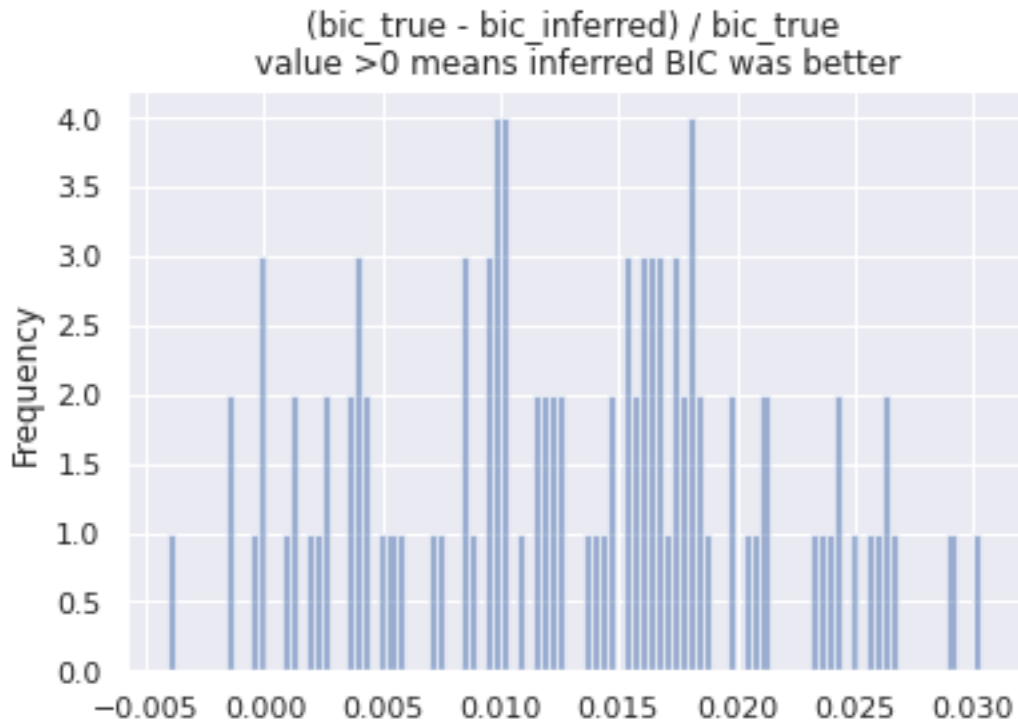
1.1.2 Plots for LikelihoodType.BEST

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

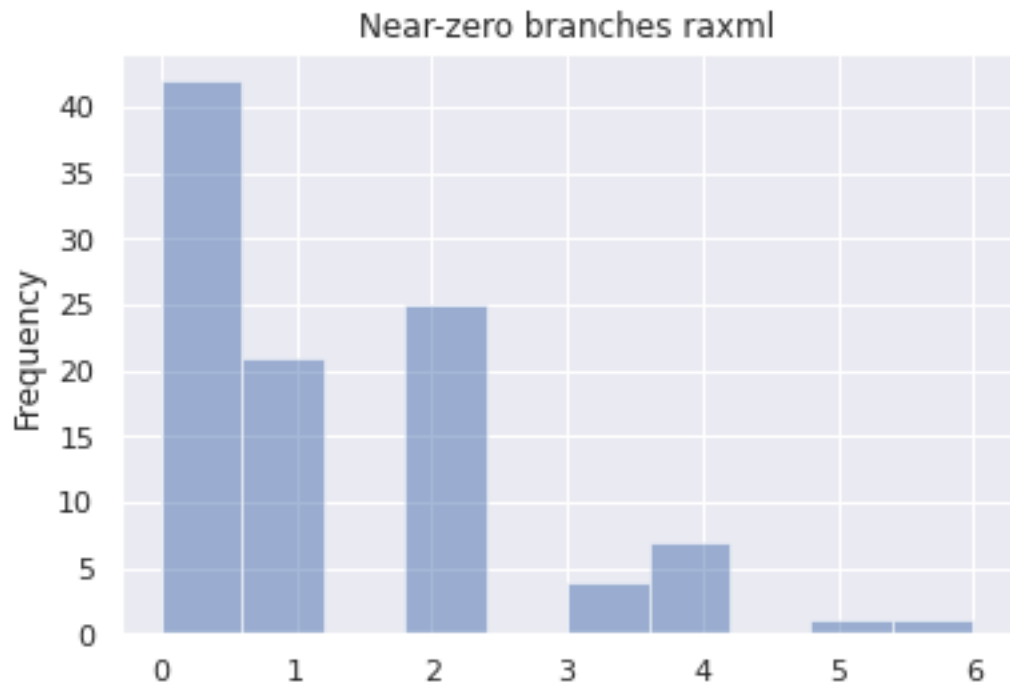
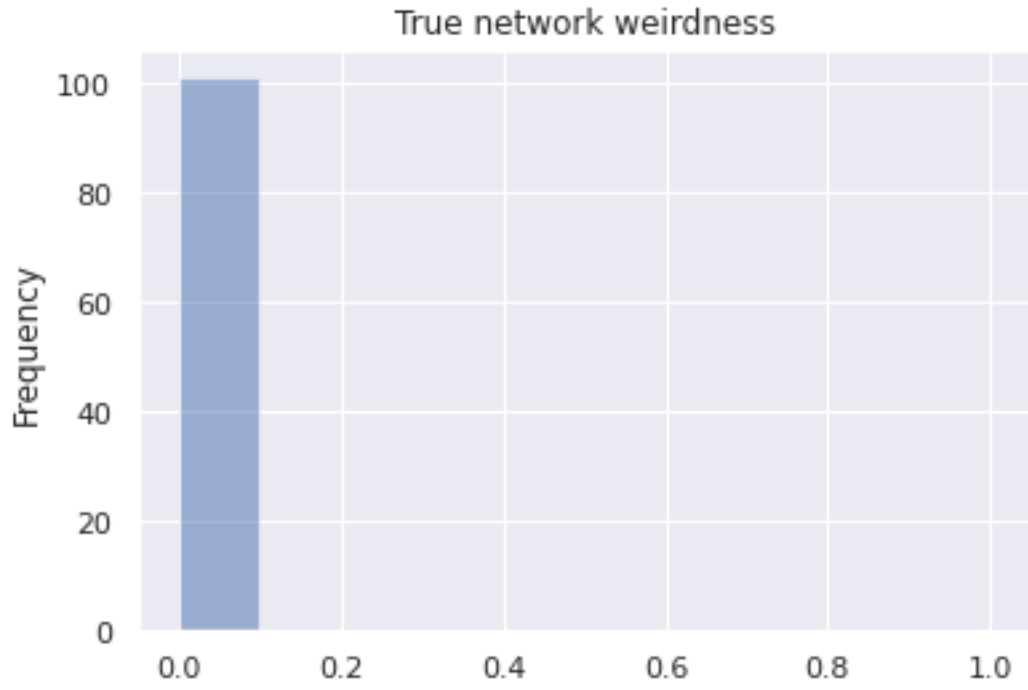
Inferred BIC better or equal: 96
Inferred BIC worse: 5

Inferred loglh better or equal: 55
Inferred loglh worse: 46

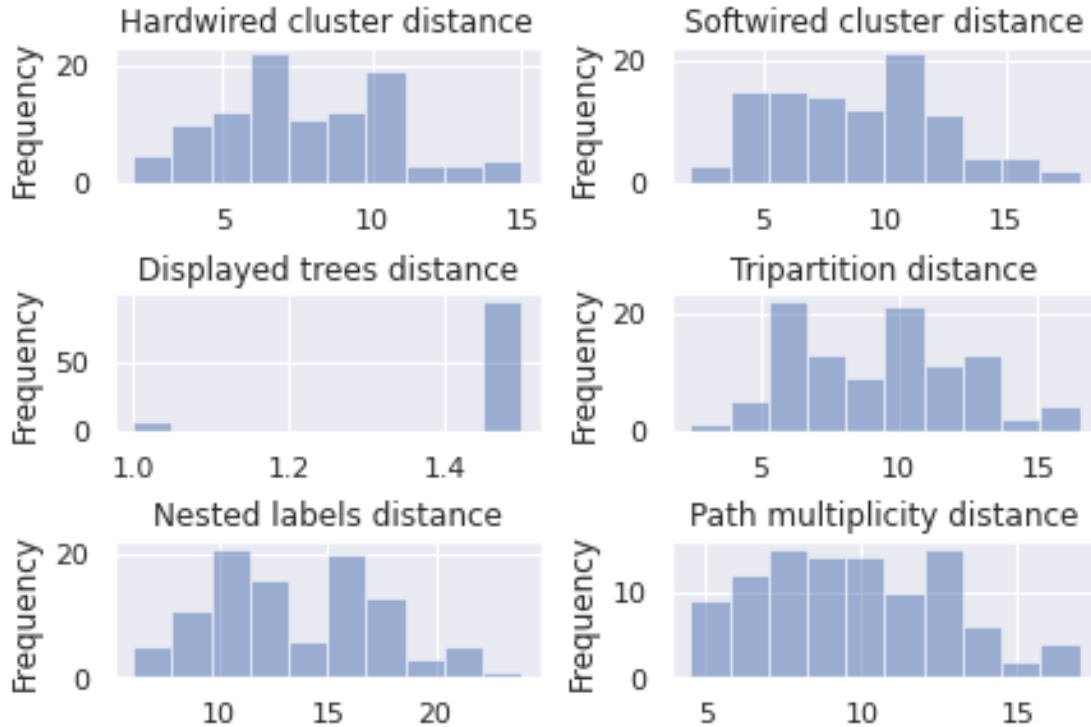
Inferred n_reticulations less: 91
Inferred n_reticulations equal: 10
Inferred n_reticulations more: 0



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



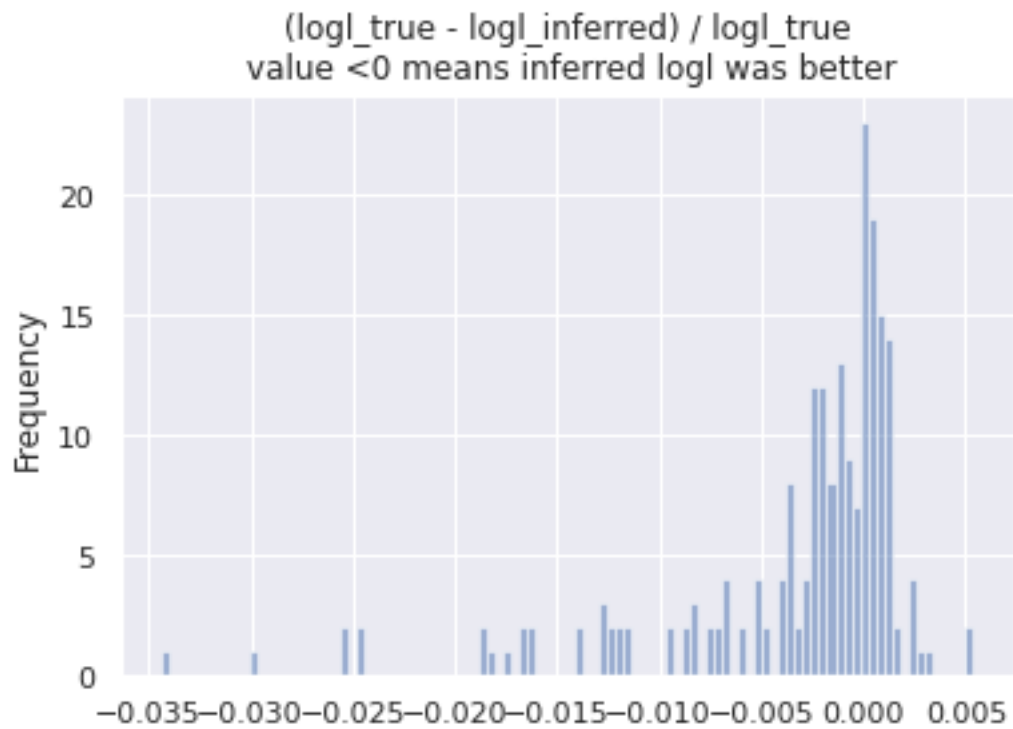
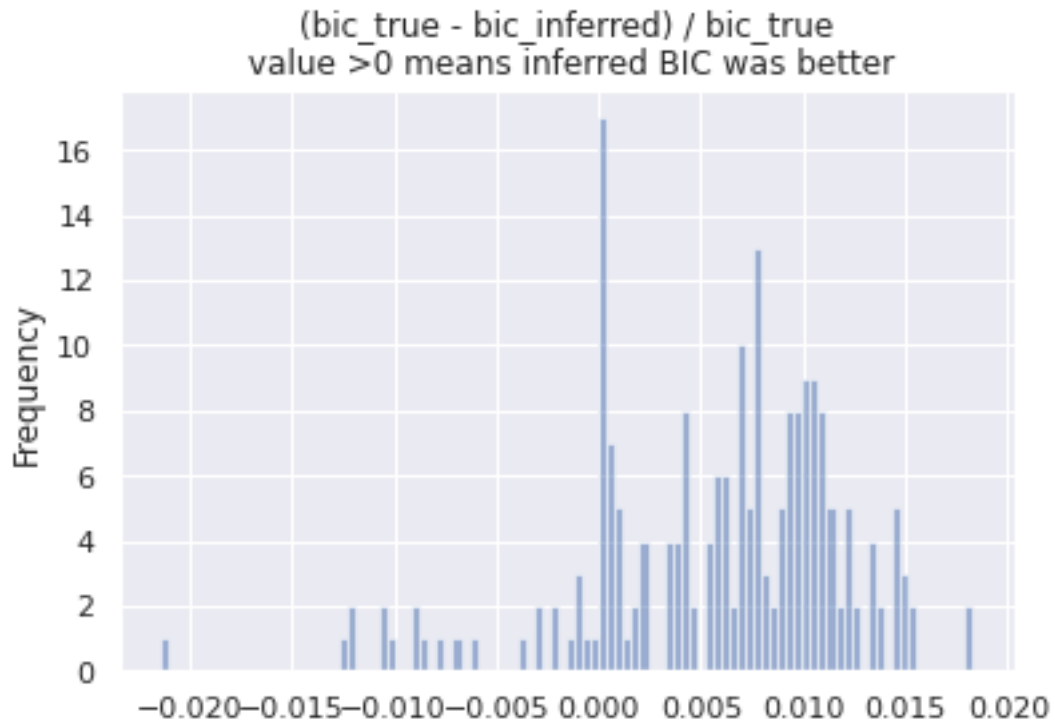
1.2 Plots for MSA_size ~ 200*n_trees

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

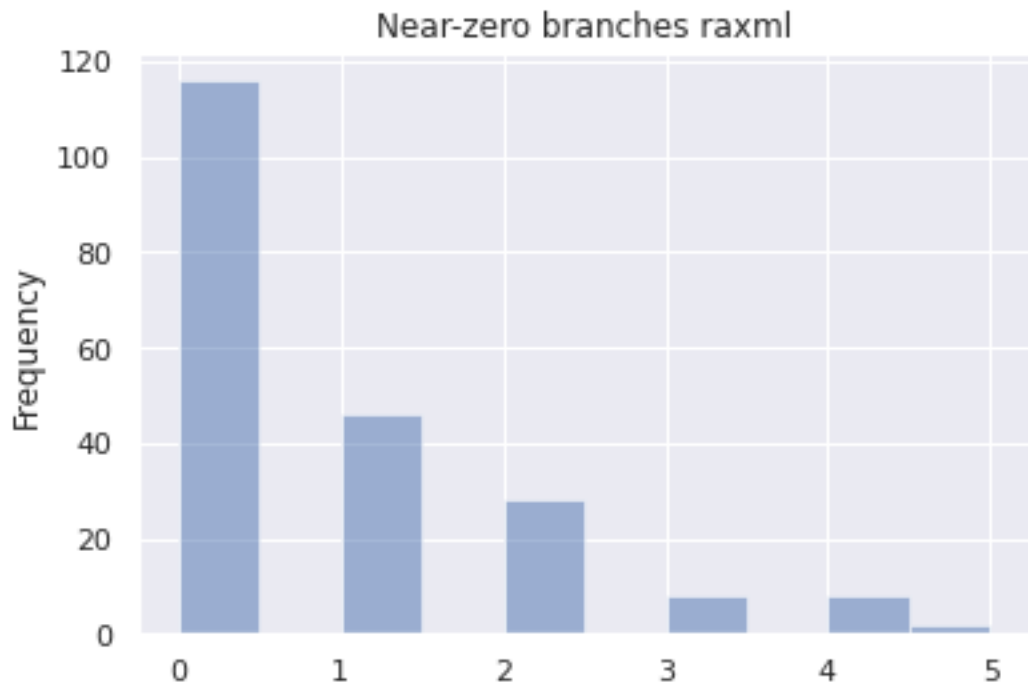
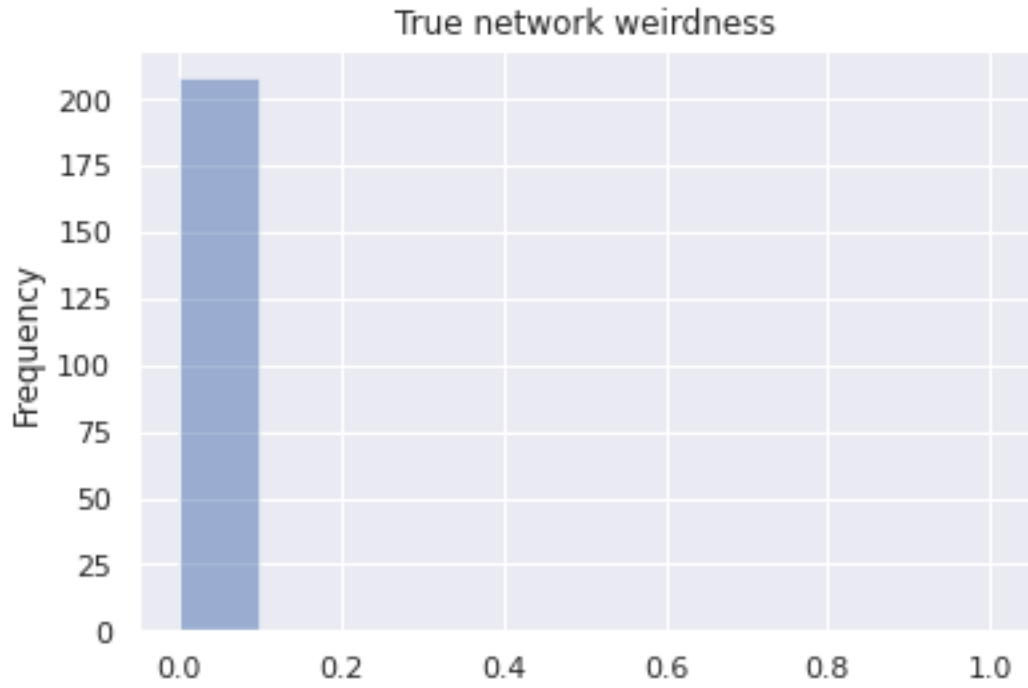
Inferred BIC better or equal: 185
Inferred BIC worse: 23

Inferred loglh better or equal: 78
Inferred loglh worse: 130

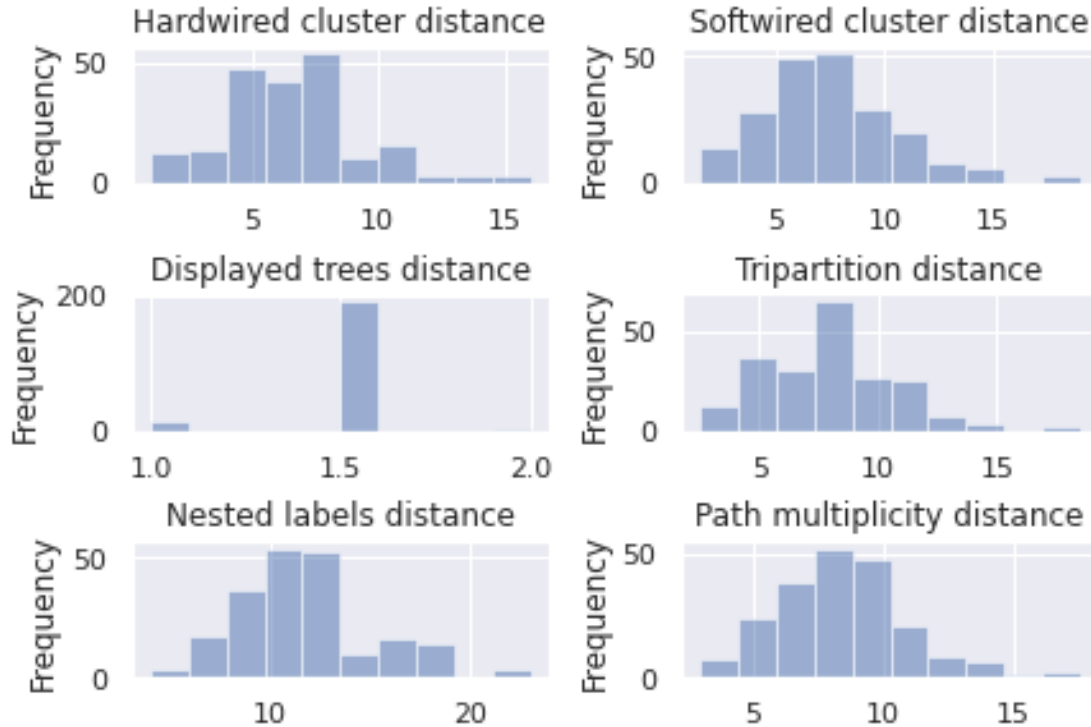
Inferred n_reticulations less: 182
Inferred n_reticulations equal: 26
Inferred n_reticulations more: 0



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



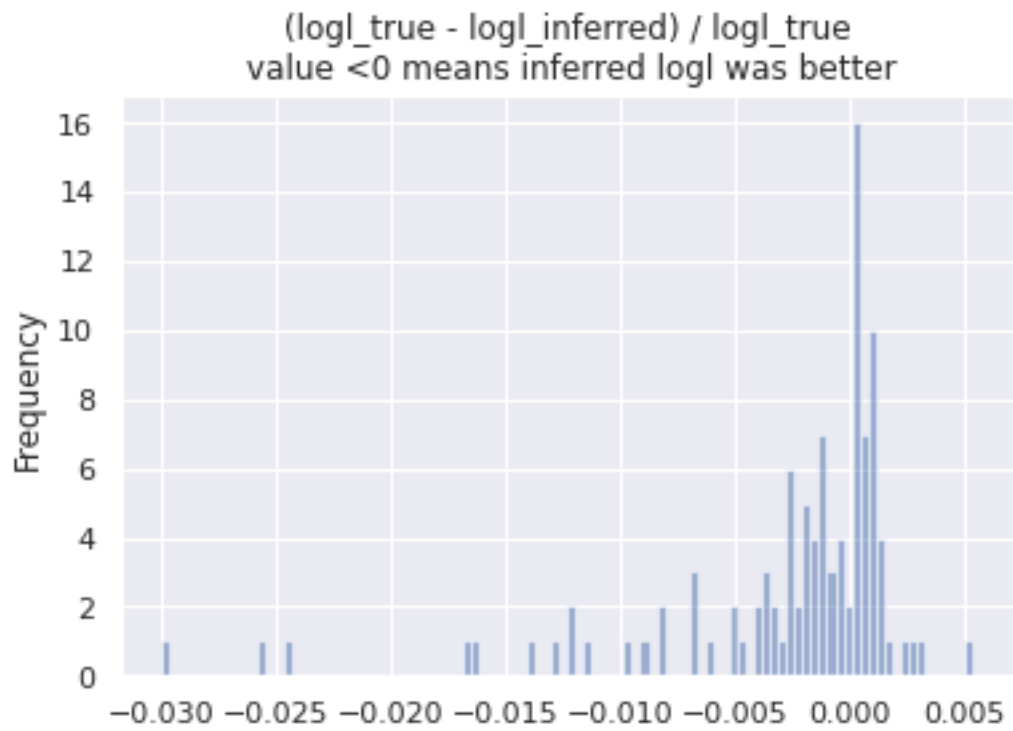
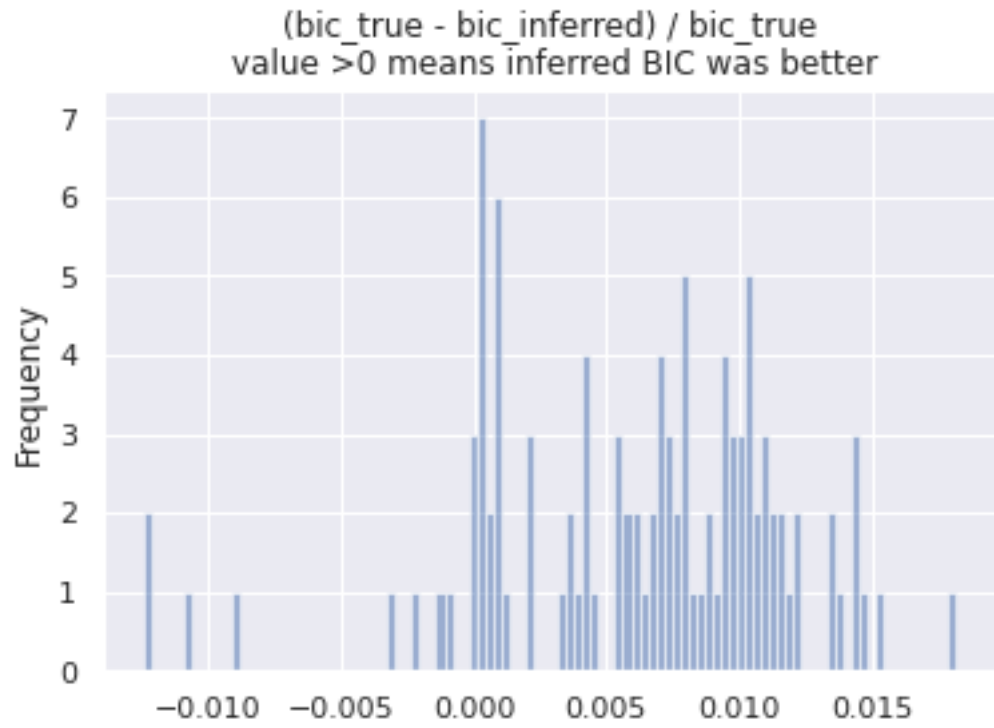
1.2.1 Plots for LikelihoodType.AVERAGE

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

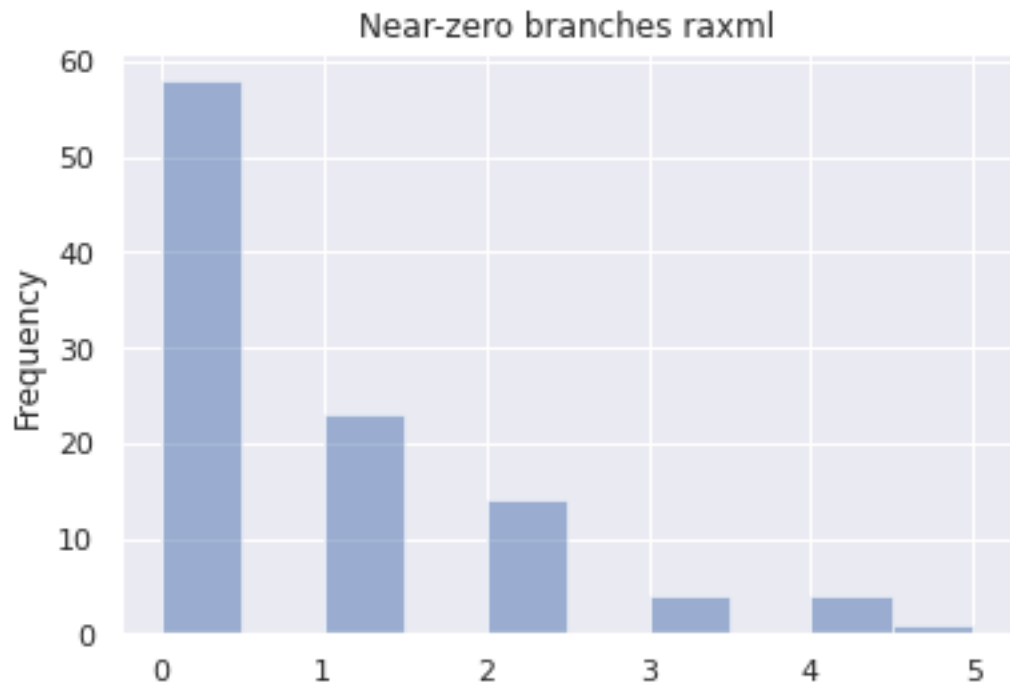
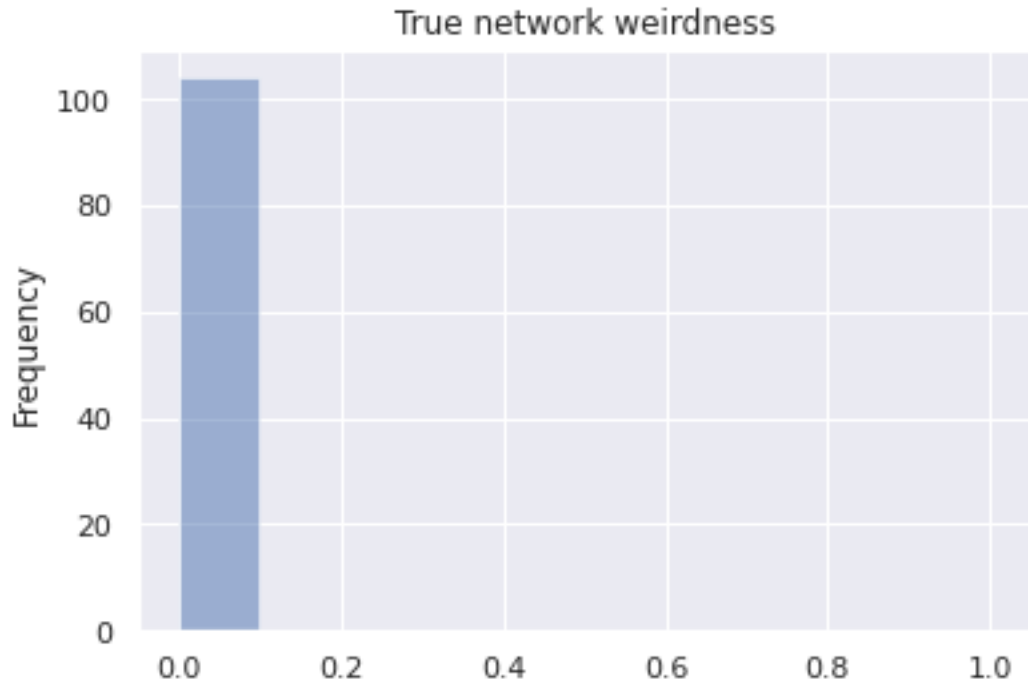
Inferred BIC better or equal: 96
Inferred BIC worse: 8

Inferred loglh better or equal: 42
Inferred loglh worse: 62

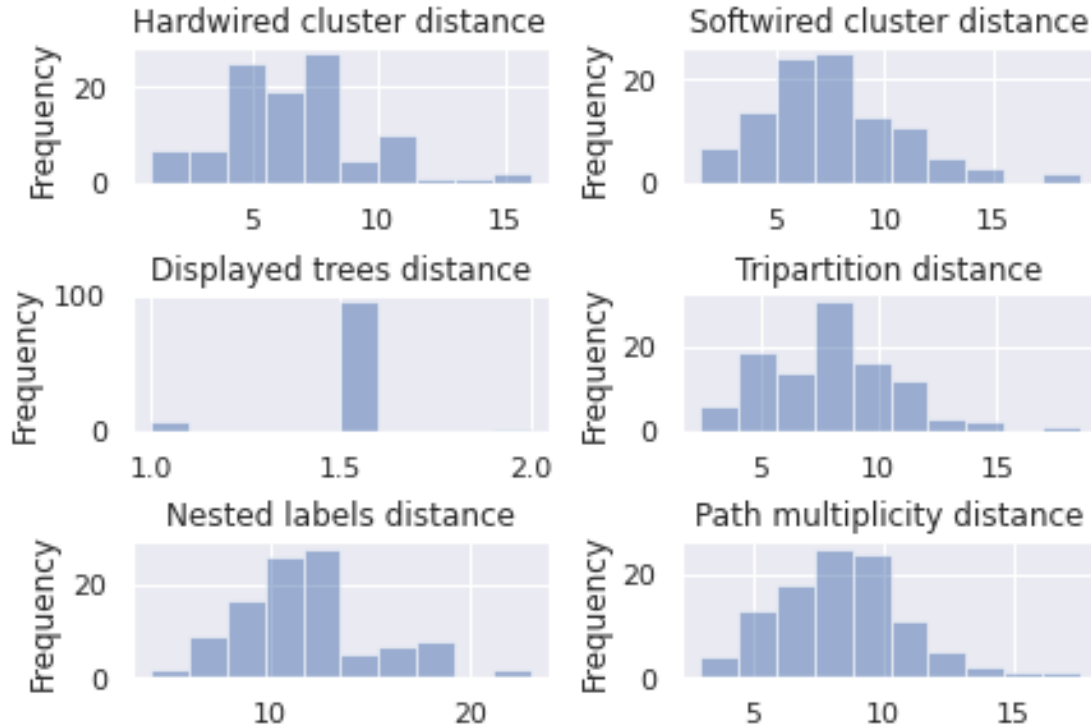
Inferred n_reticulations less: 88
Inferred n_reticulations equal: 16
Inferred n_reticulations more: 0



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



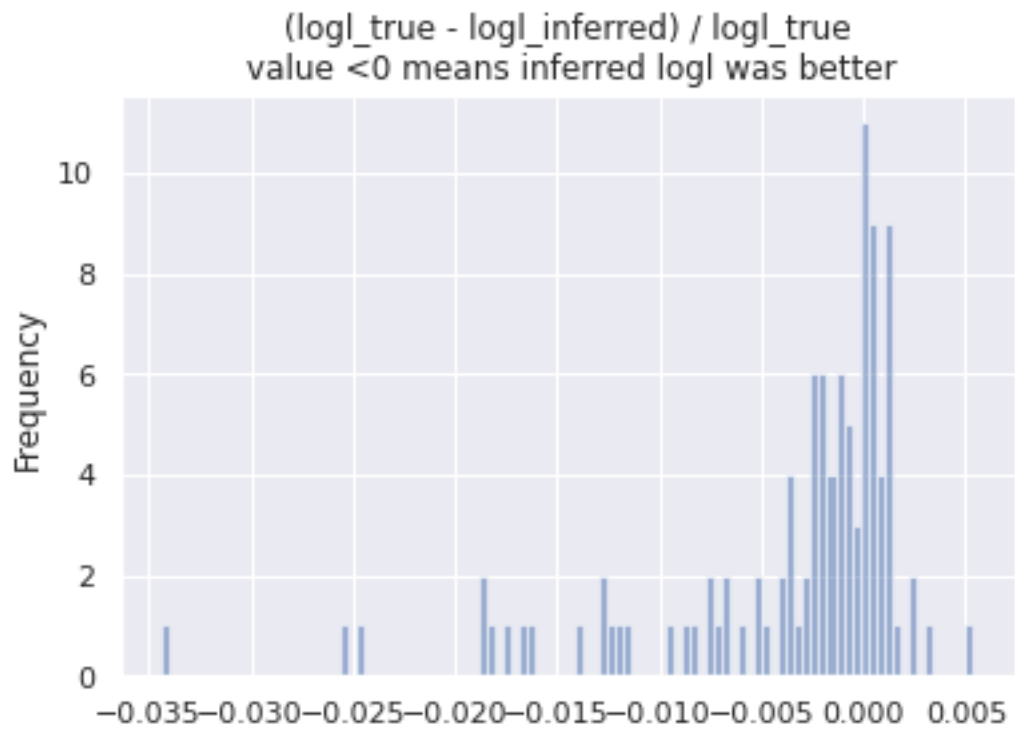
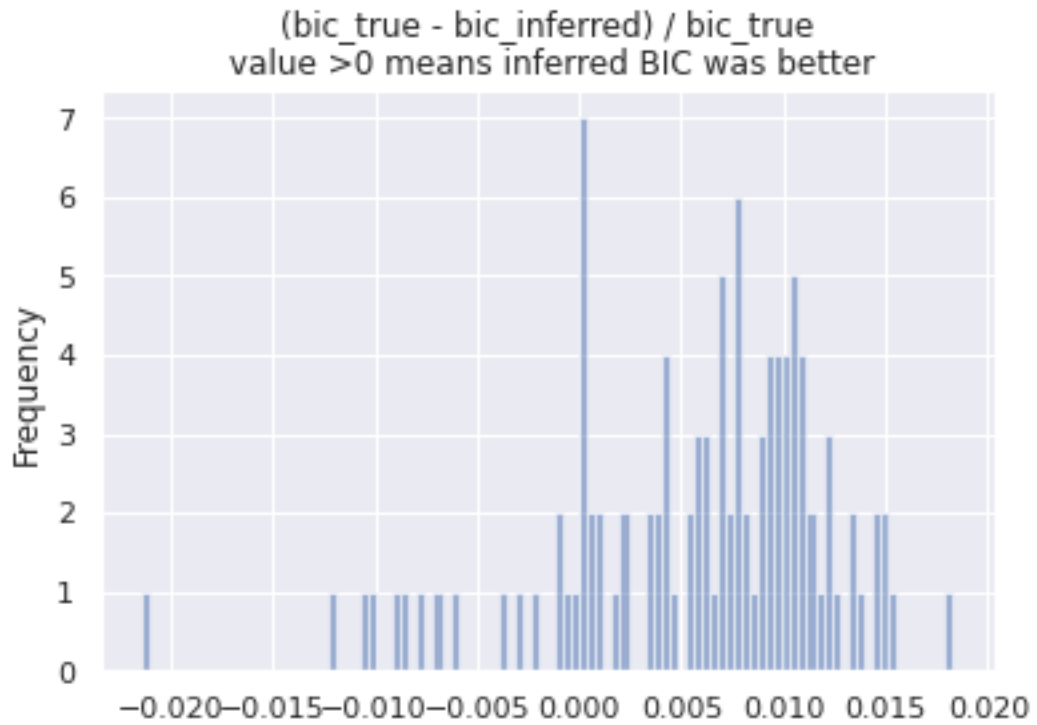
1.2.2 Plots for LikelihoodType.BEST

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

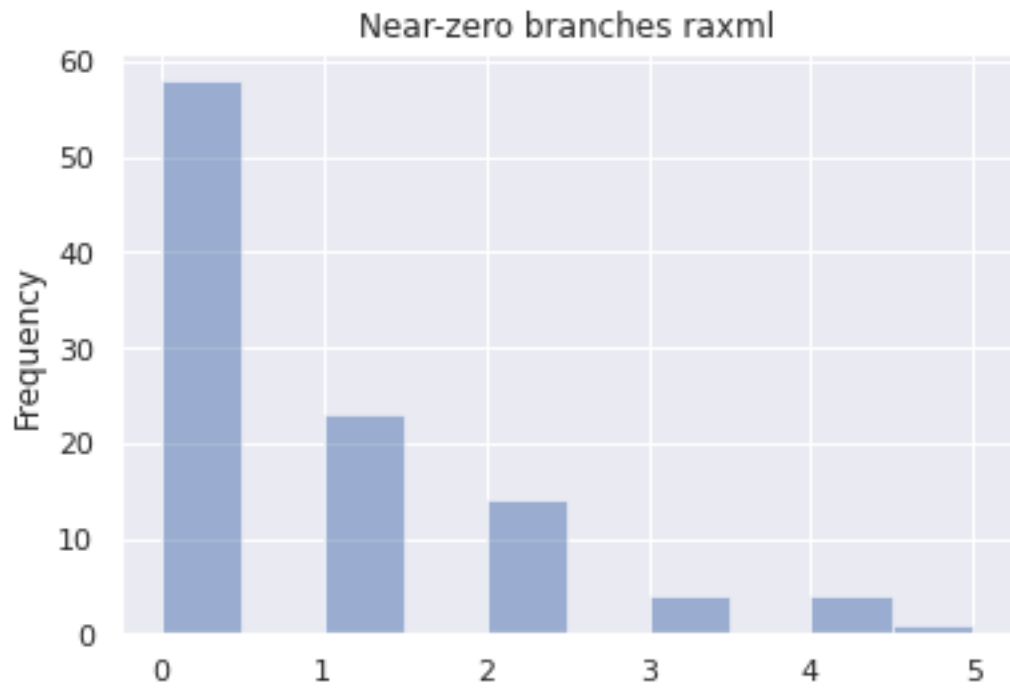
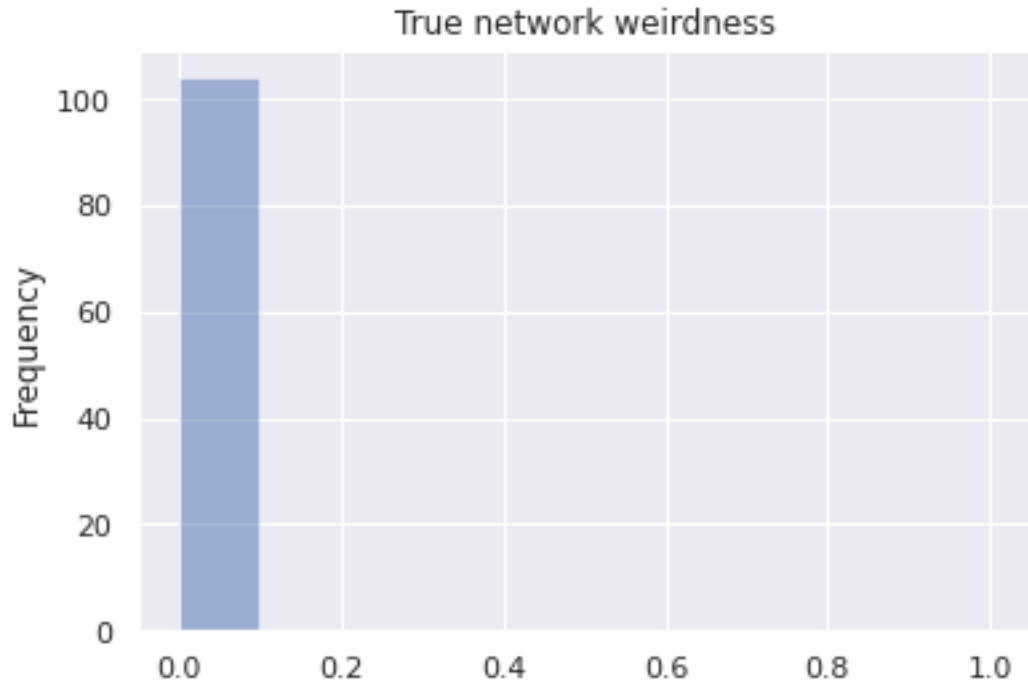
Inferred BIC better or equal: 89
Inferred BIC worse: 15

Inferred loglh better or equal: 36
Inferred loglh worse: 68

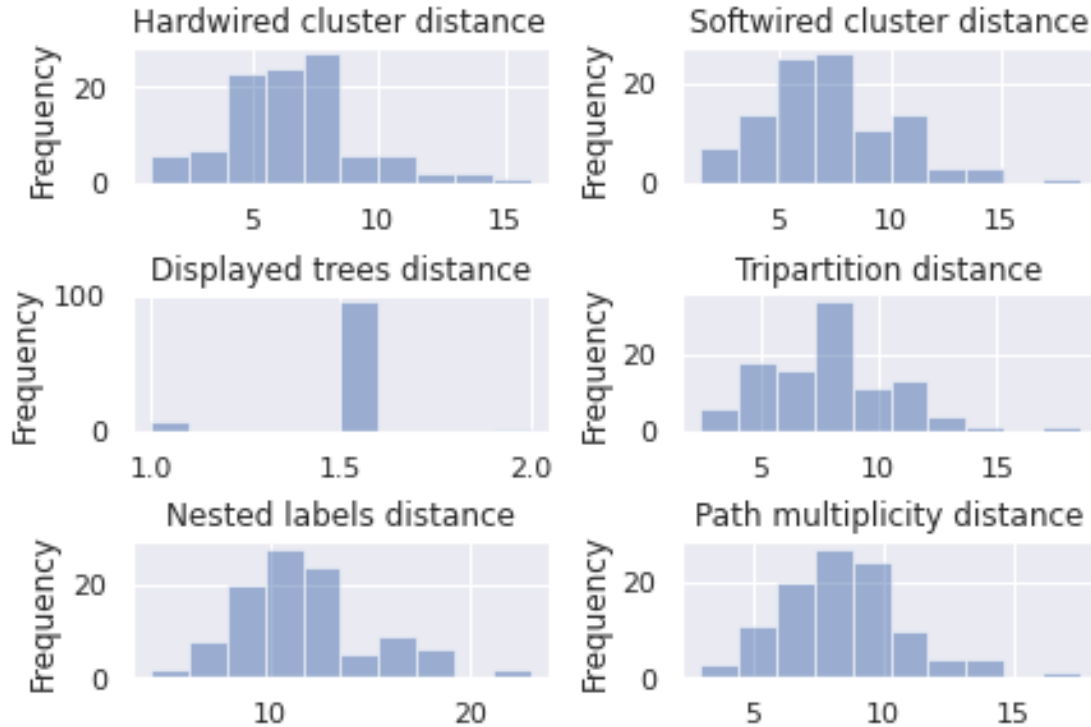
Inferred n_reticulations less: 94
Inferred n_reticulations equal: 10
Inferred n_reticulations more: 0



<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: 0

Inferred BIC worse: 0

Inferred loglh better or equal: 0

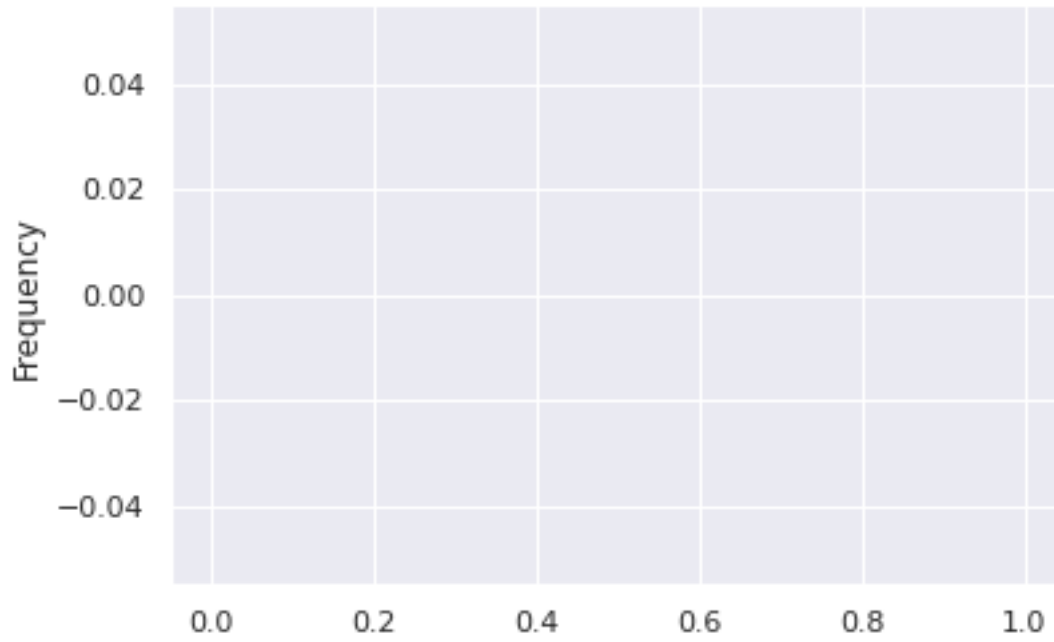
Inferred loglh worse: 0

Inferred n_reticulations less: 0

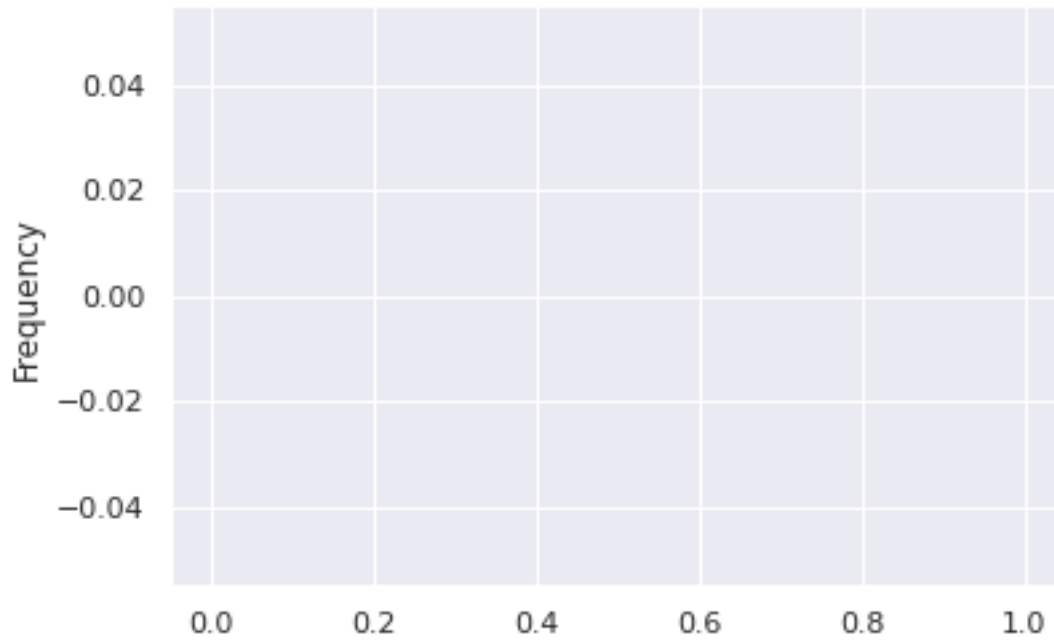
Inferred n_reticulations equal: 0

Inferred n_reticulations more: 0

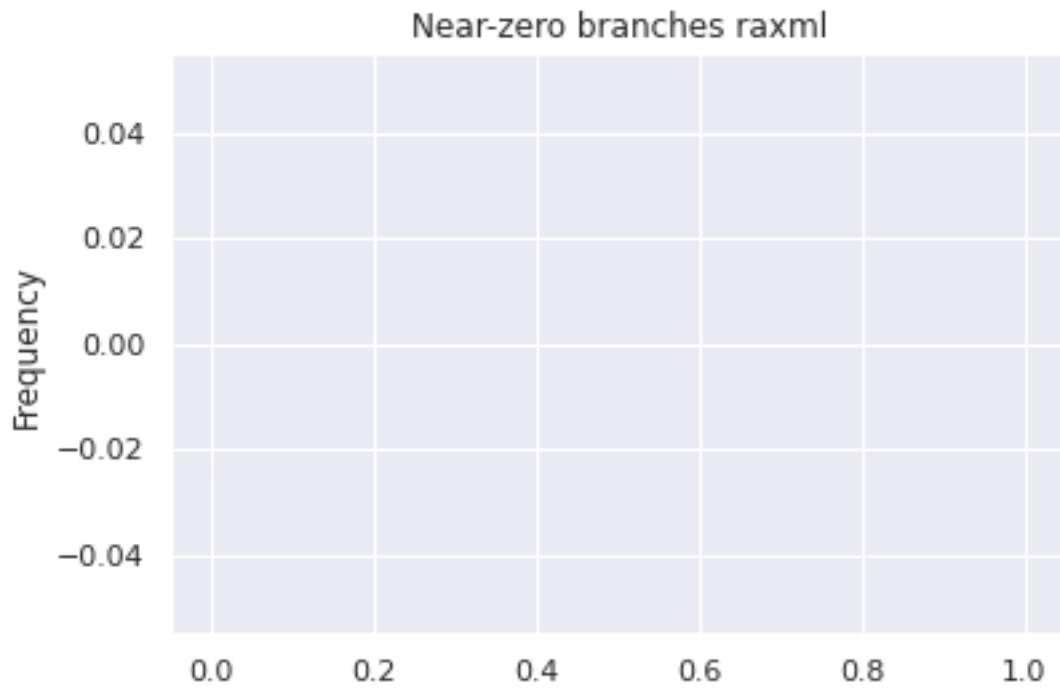
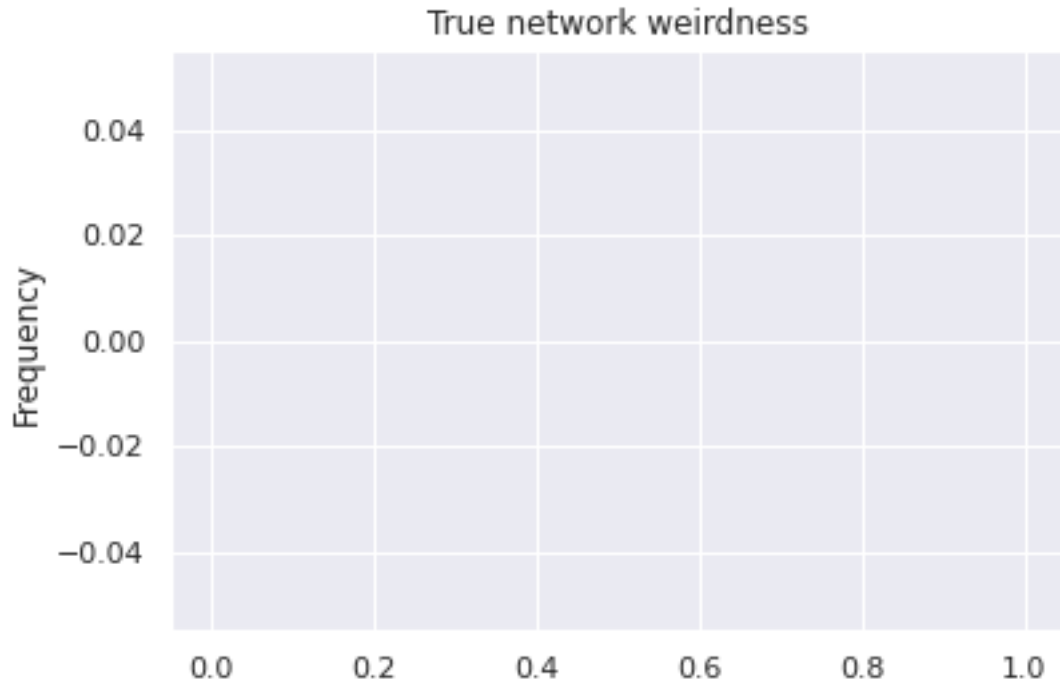
$(bic_true - bic_inferred) / bic_true$
value >0 means inferred BIC was better



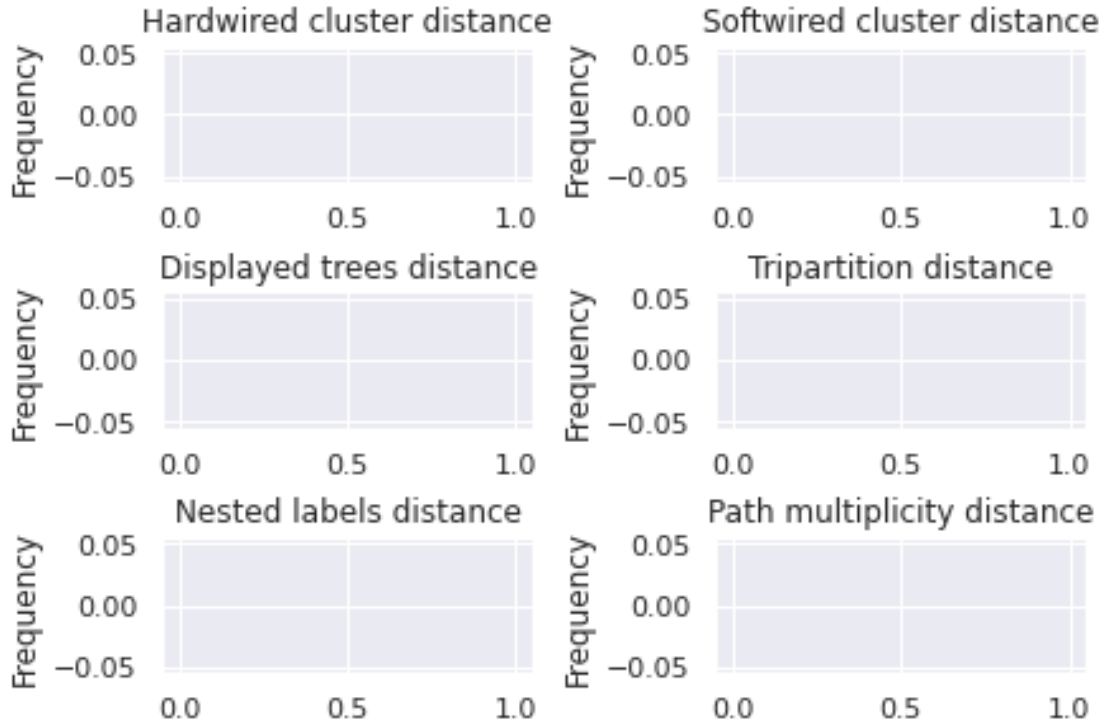
$(logl_true - logl_inferred) / logl_true$
value <0 means inferred logl was better



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



2.1 Plots for $MSA_size \sim 100 * n_trees$

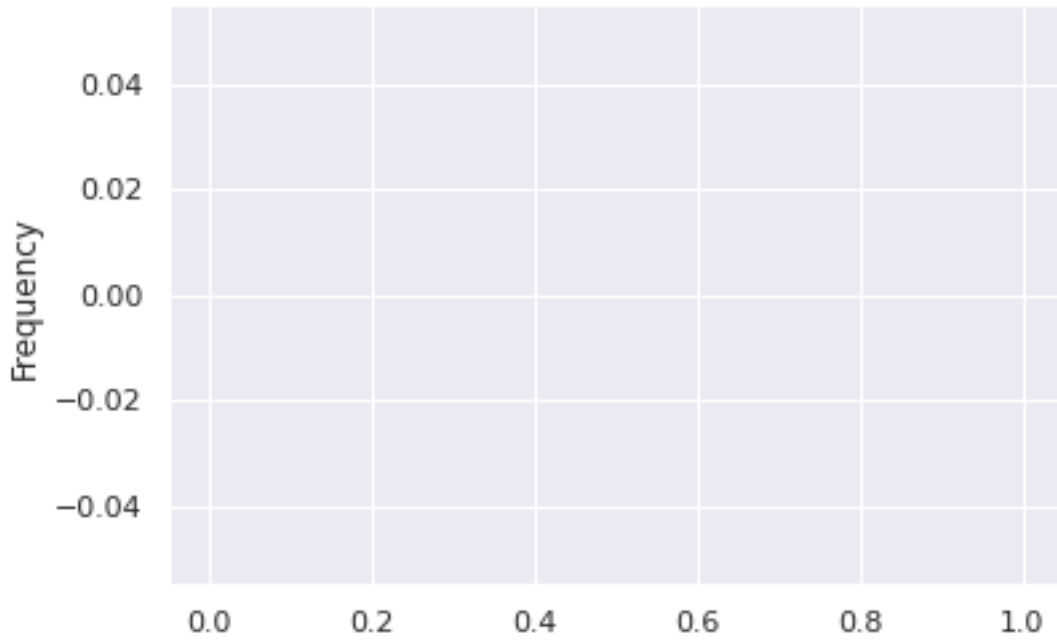
```
[15]: df_random_msasize_100 = df_random.query('msa_size == 101')
      build_stats(df_random_msasize_100)
```

Inferred BIC better or equal: 0
Inferred BIC worse: 0

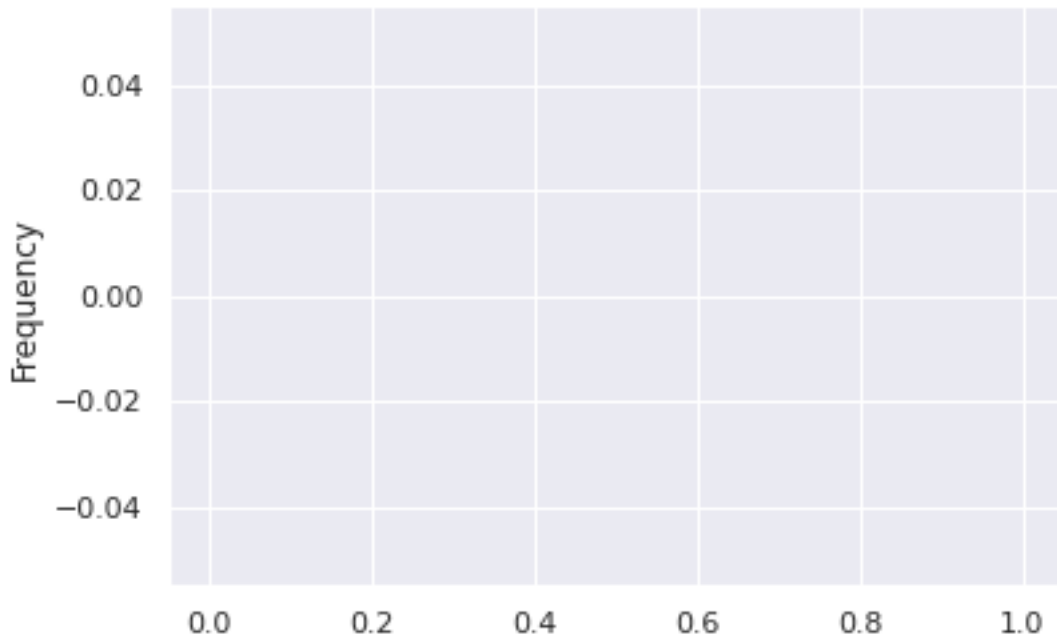
Inferred loglh better or equal: 0
Inferred loglh worse: 0

Inferred n_reticulations less: 0
Inferred n_reticulations equal: 0
Inferred n_reticulations more: 0

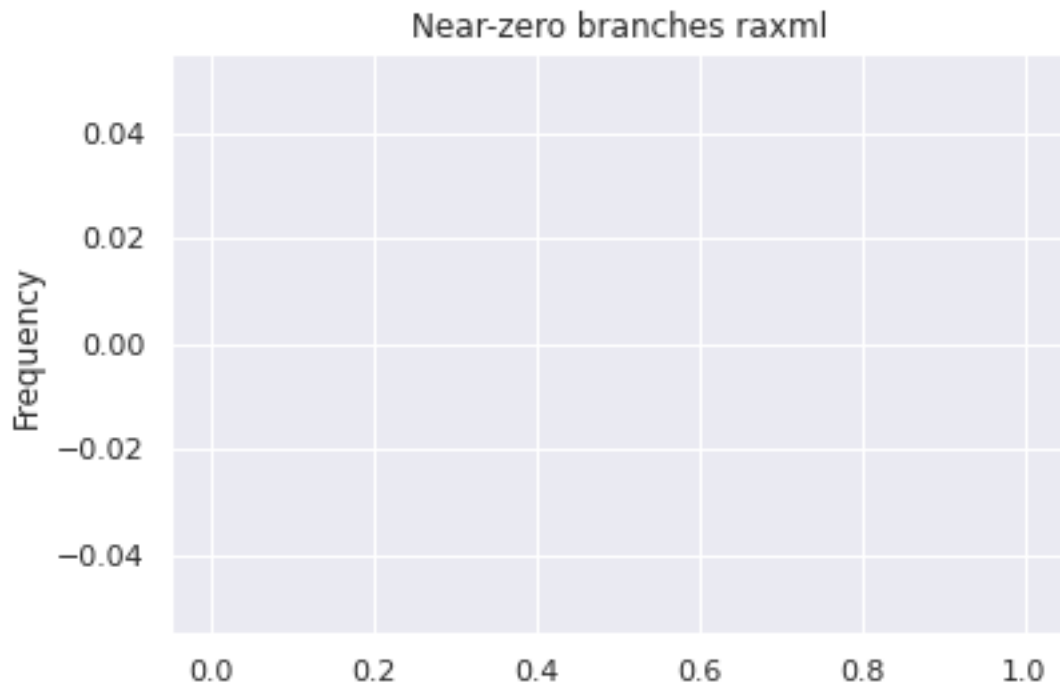
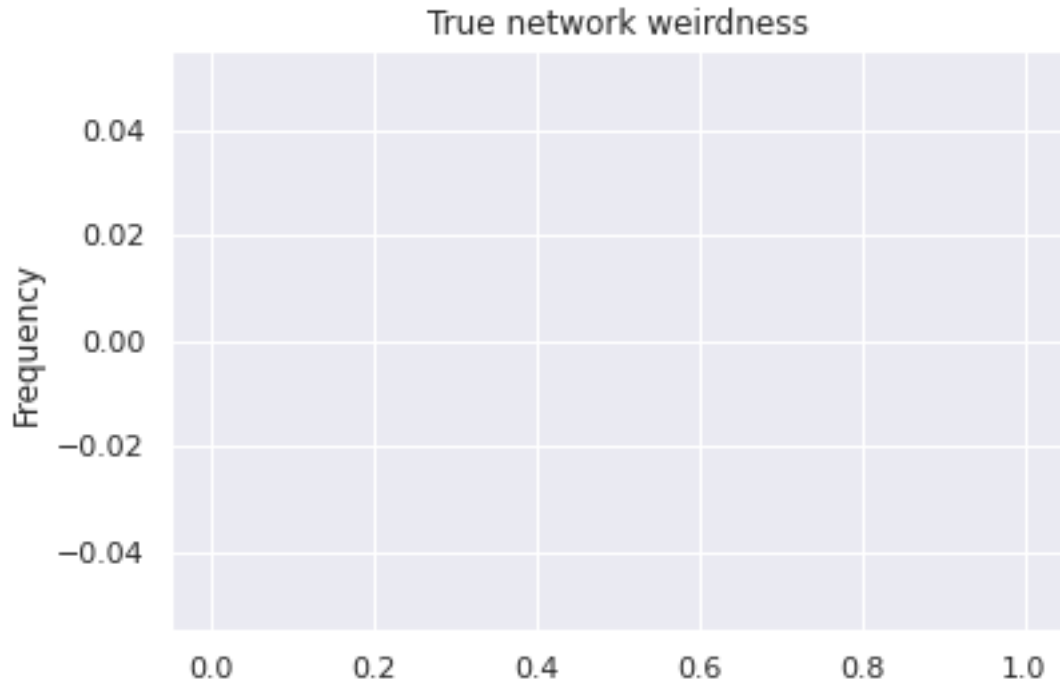
$(bic_true - bic_inferred) / bic_true$
value >0 means inferred BIC was better



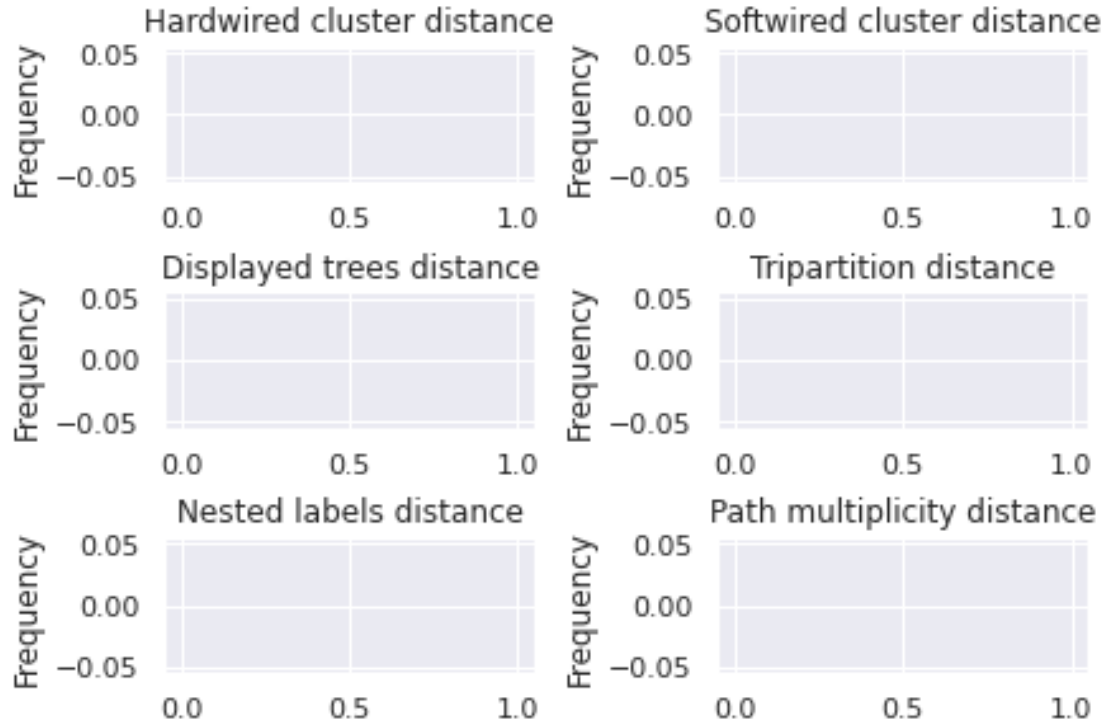
$(logl_true - logl_inferred) / logl_true$
value <0 means inferred logl was better



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



2.1.1 Plots for LikelihoodType.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: 0

Inferred BIC worse: 0

Inferred loglh better or equal: 0

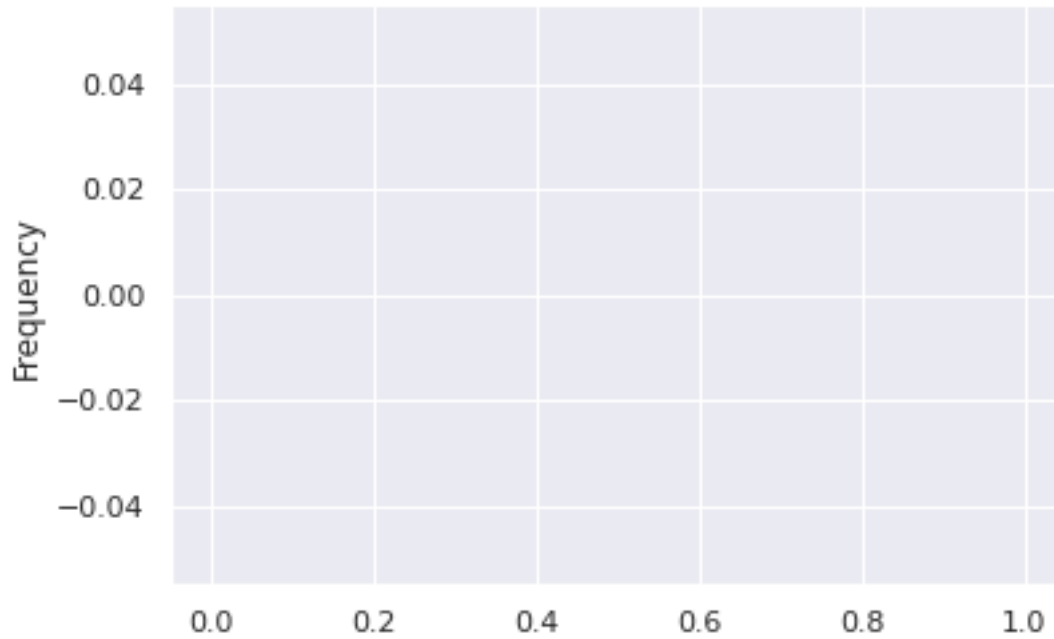
Inferred loglh worse: 0

Inferred n_reticulations less: 0

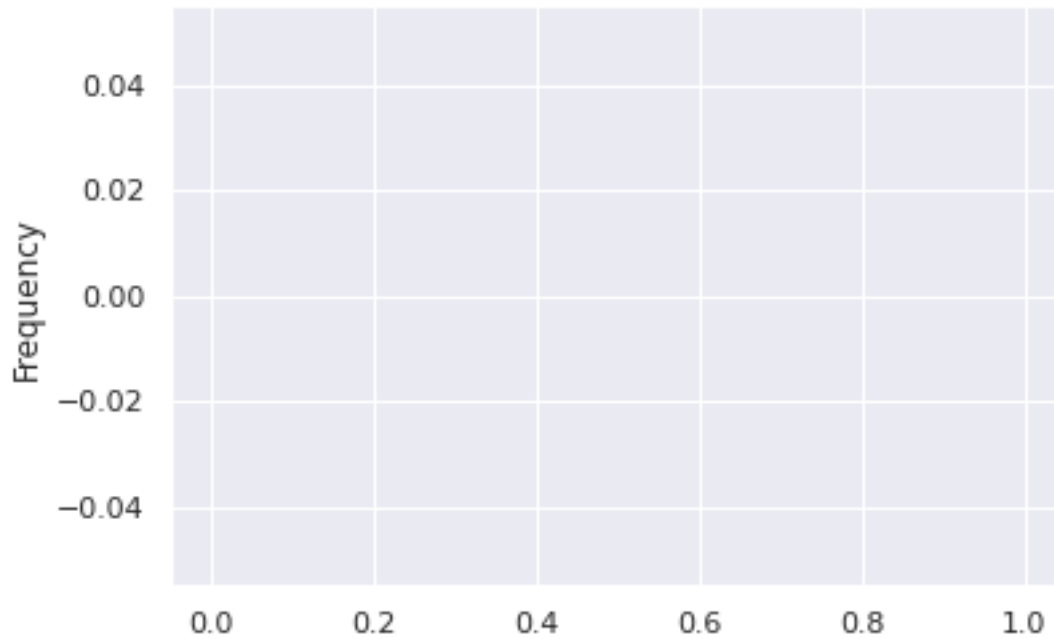
Inferred n_reticulations equal: 0

Inferred n_reticulations more: 0

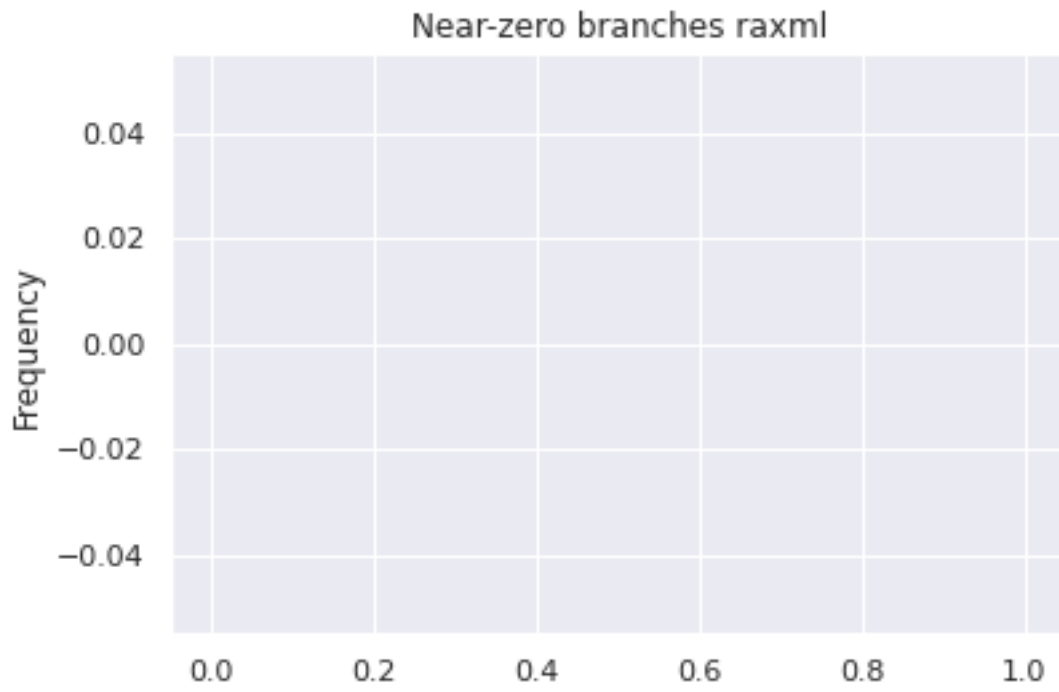
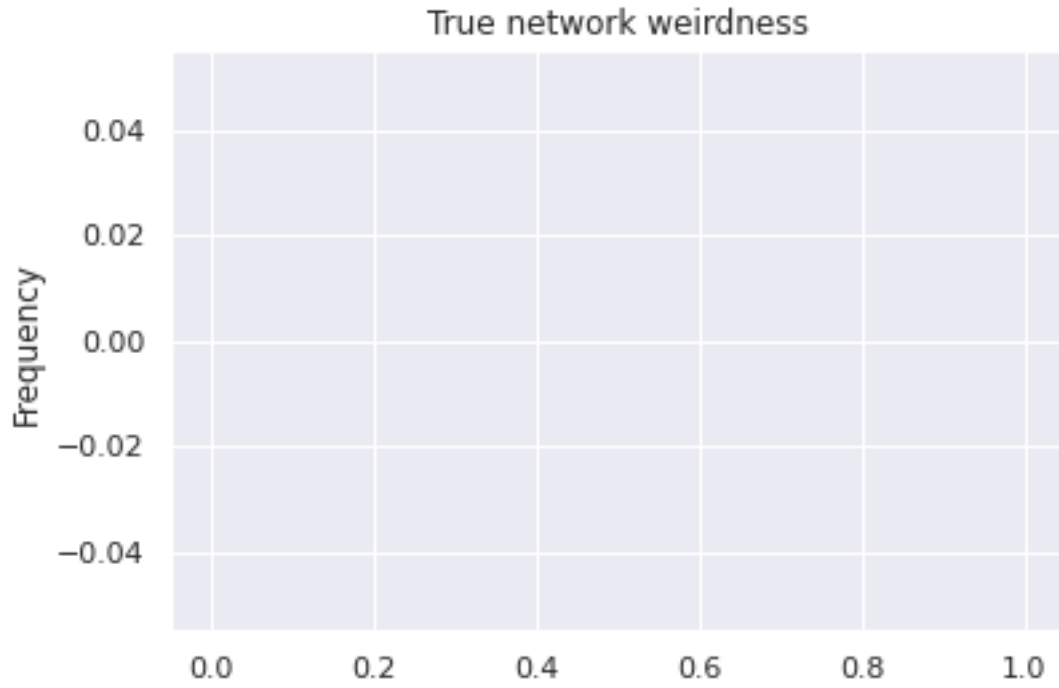
$(bic_true - bic_inferred) / bic_true$
value >0 means inferred BIC was better



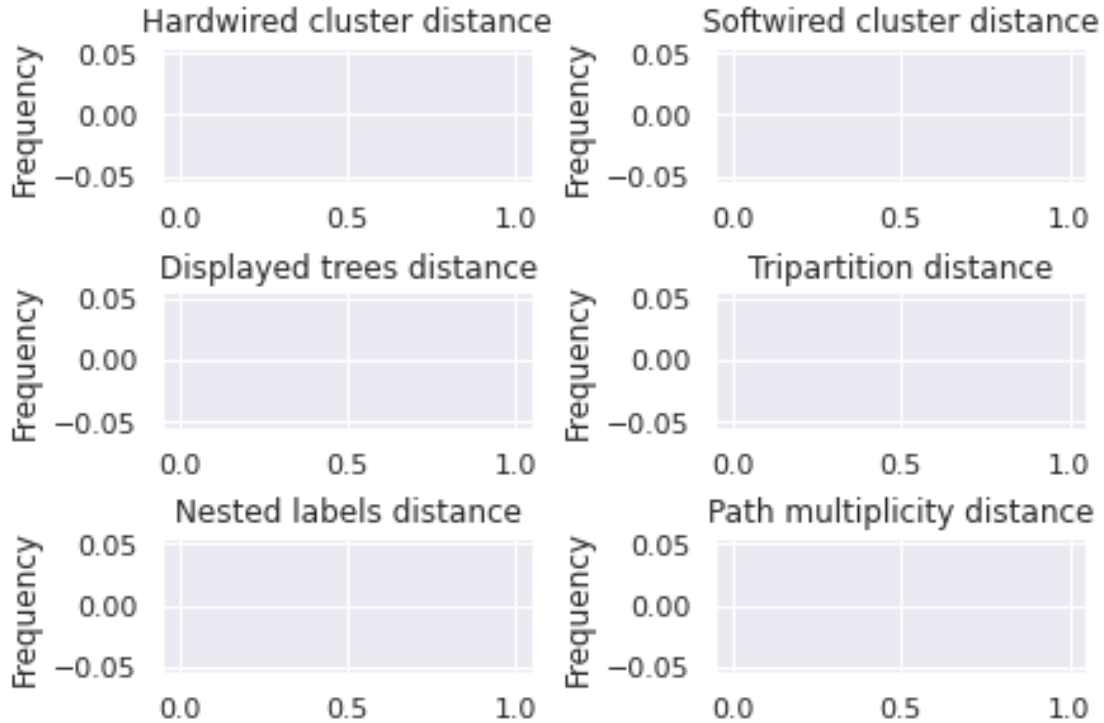
$(logl_true - logl_inferred) / logl_true$
value <0 means inferred logl was better



<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: 0

Inferred BIC worse: 0

Inferred loglh better or equal: 0

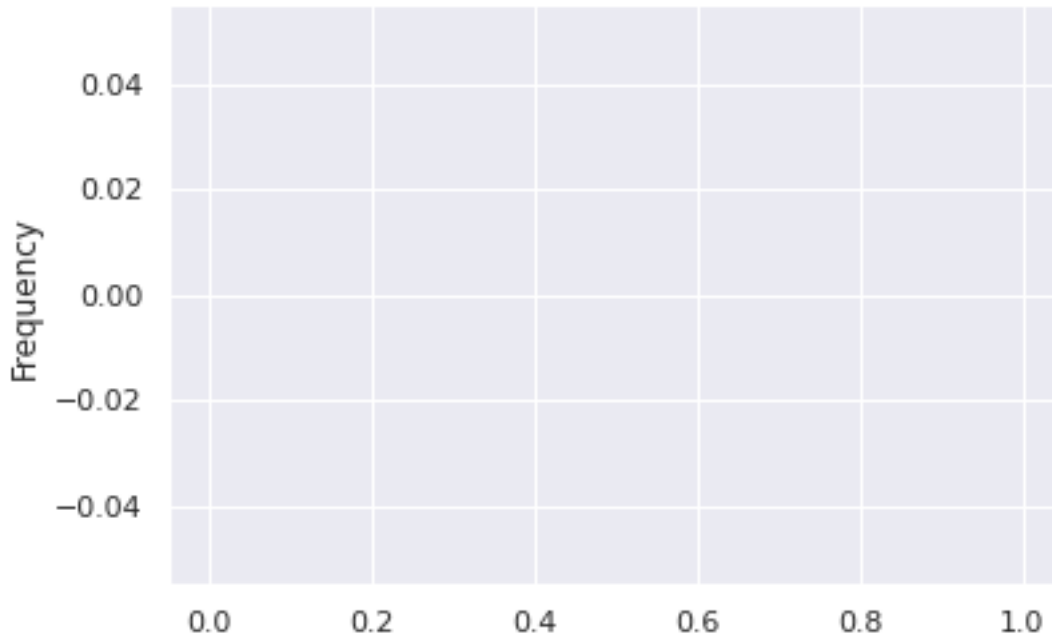
Inferred loglh worse: 0

Inferred n_reticulations less: 0

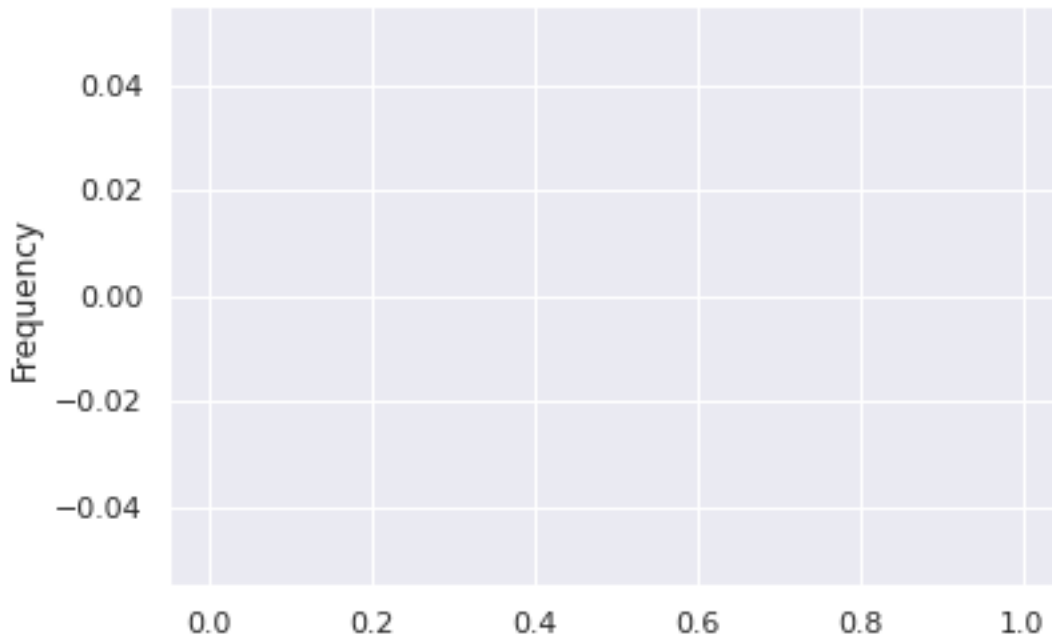
Inferred n_reticulations equal: 0

Inferred n_reticulations more: 0

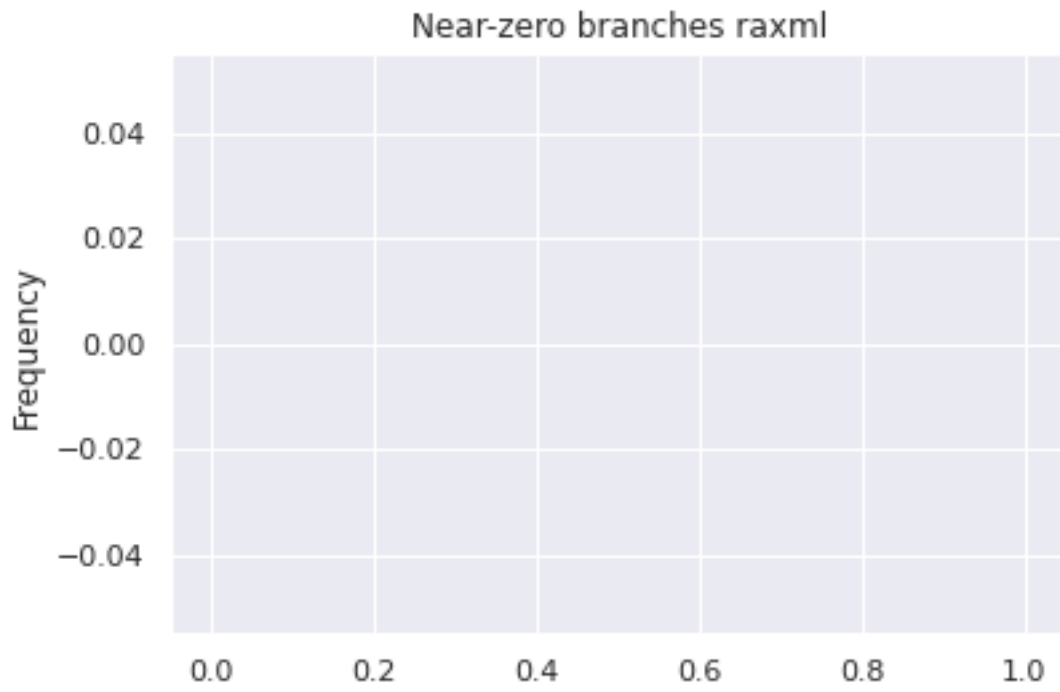
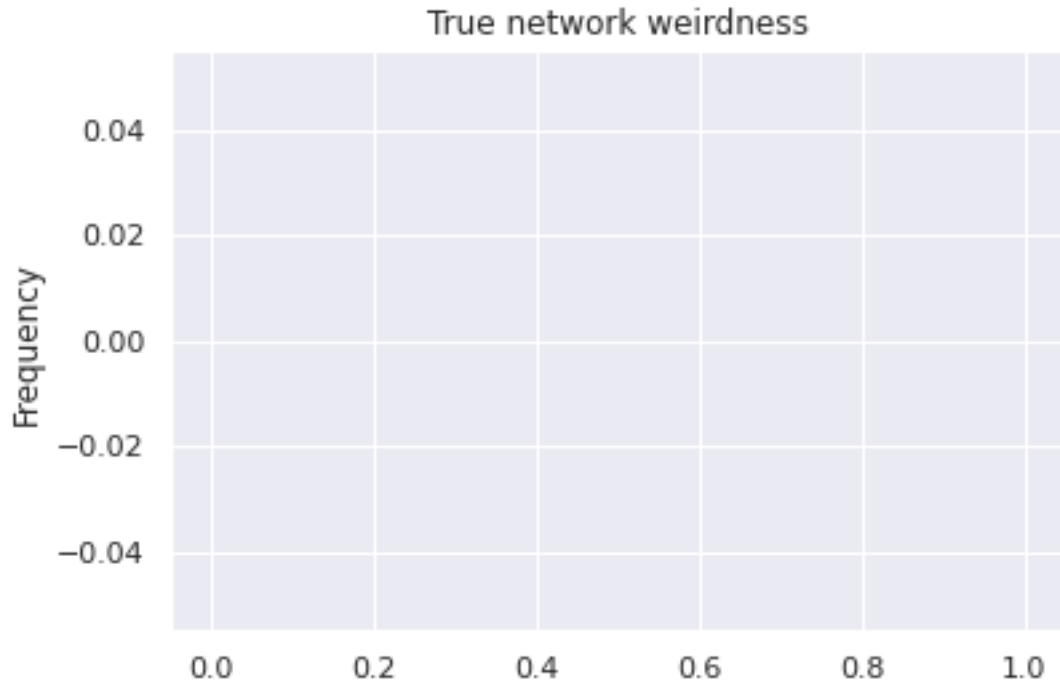
$(bic_true - bic_inferred) / bic_true$
value >0 means inferred BIC was better



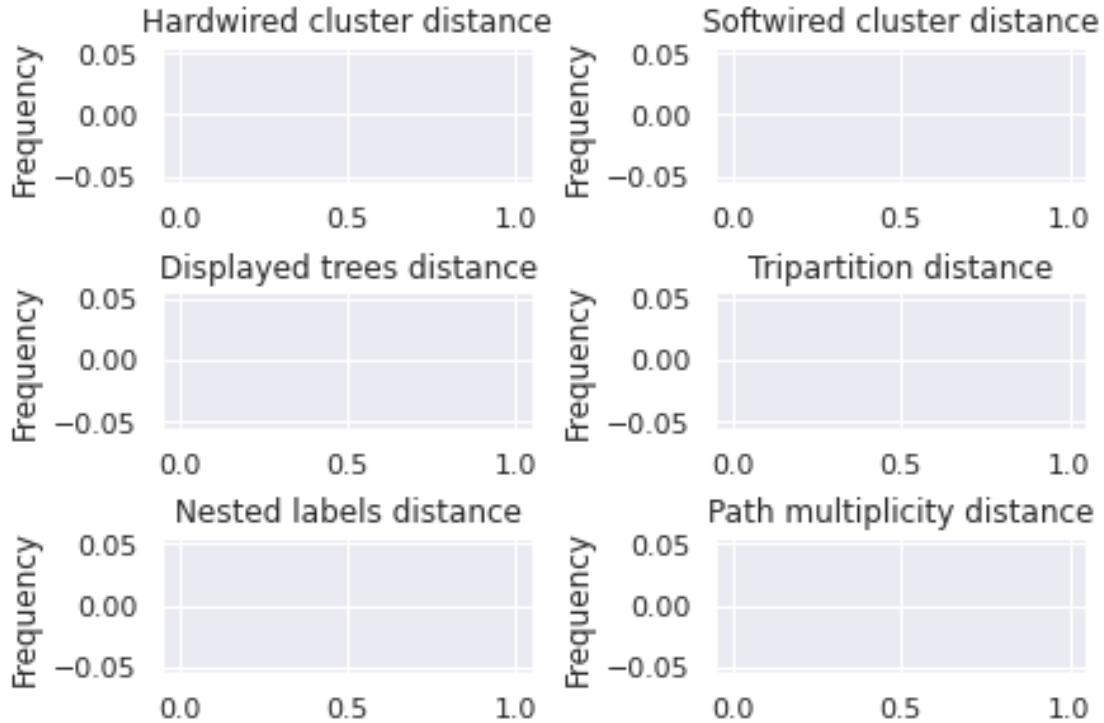
$(logl_true - logl_inferred) / logl_true$
value <0 means inferred logl was better



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



2.2 Plots for $MSA_size \sim 200 * n_trees$

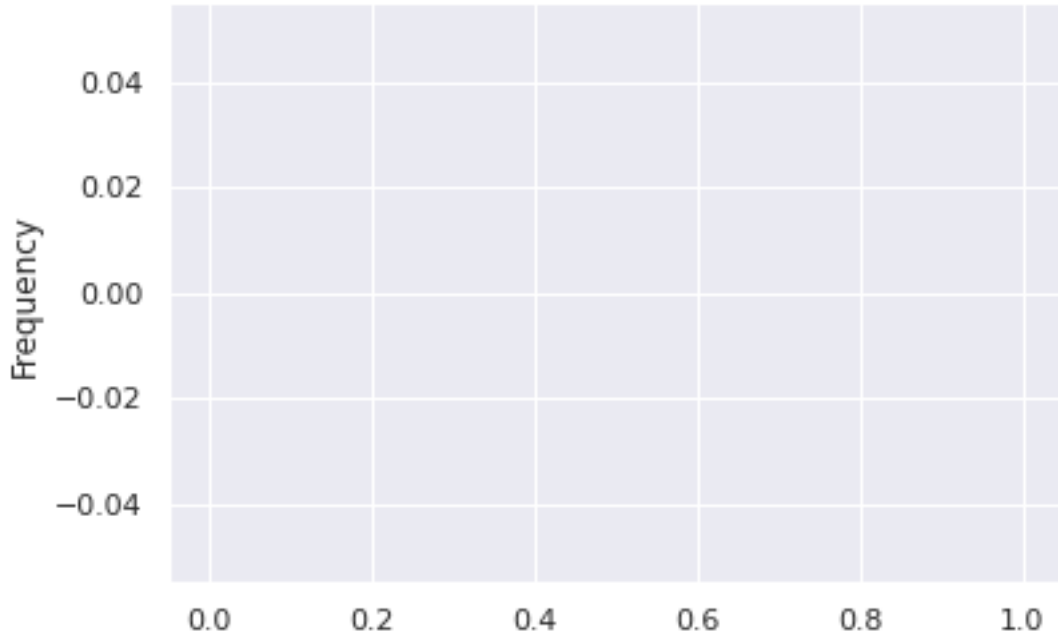
```
[18]: df_random_msasize_200 = df_random.query('msa_size == 201')
      build_stats(df_random_msasize_200)
```

```
Inferred BIC better or equal: 0
Inferred BIC worse: 0
```

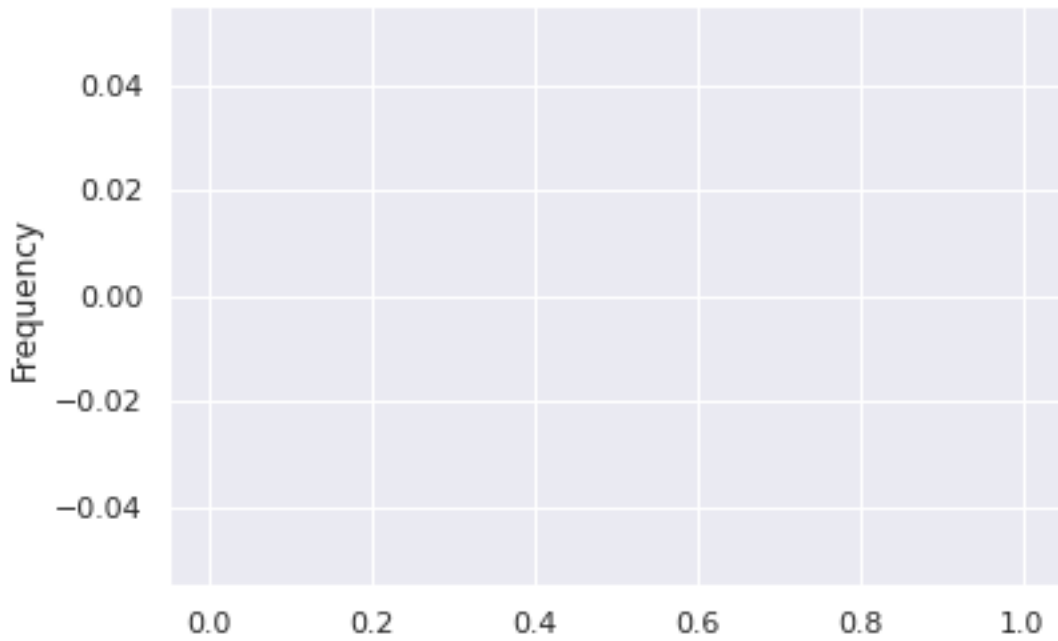
```
Inferred loglh better or equal: 0
Inferred loglh worse: 0
```

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

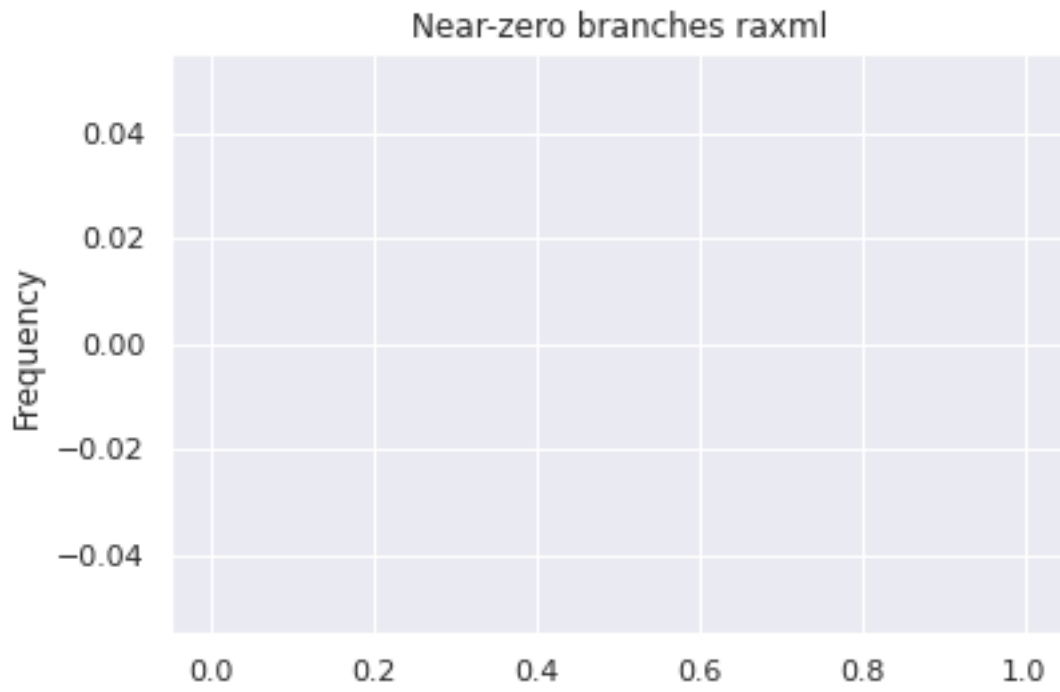
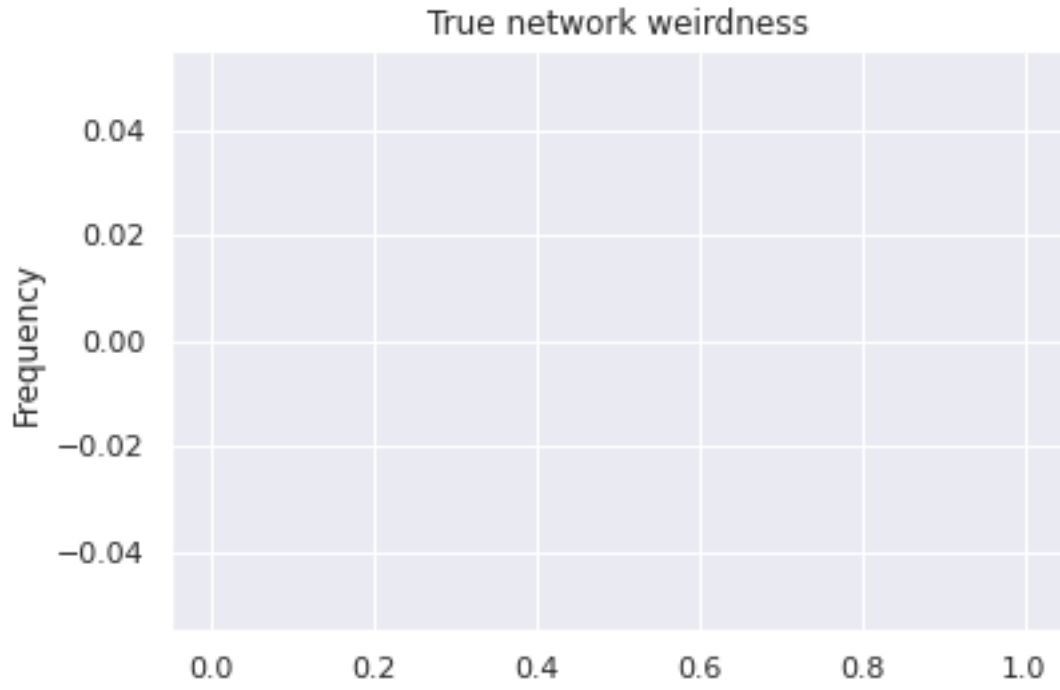
$(bic_true - bic_inferred) / bic_true$
value >0 means inferred BIC was better



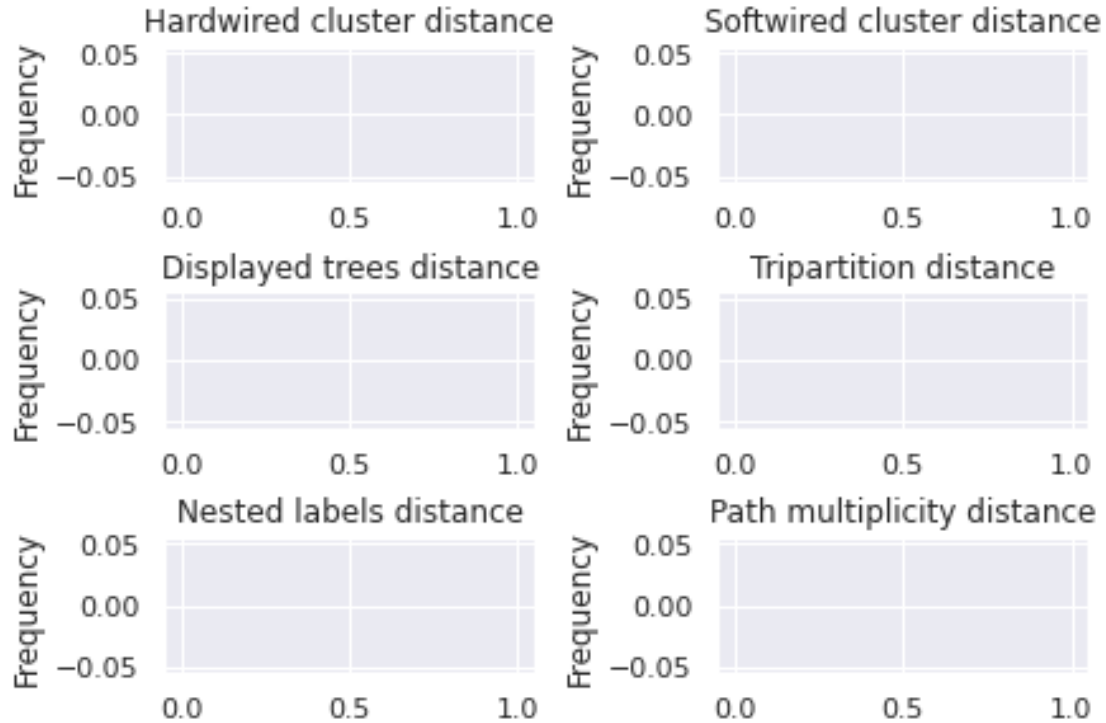
$(logl_true - logl_inferred) / logl_true$
value <0 means inferred logl was better



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



2.2.1 Plots for LikelihoodType.AVERAGE

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

Inferred BIC better or equal: 0

Inferred BIC worse: 0

Inferred loglh better or equal: 0

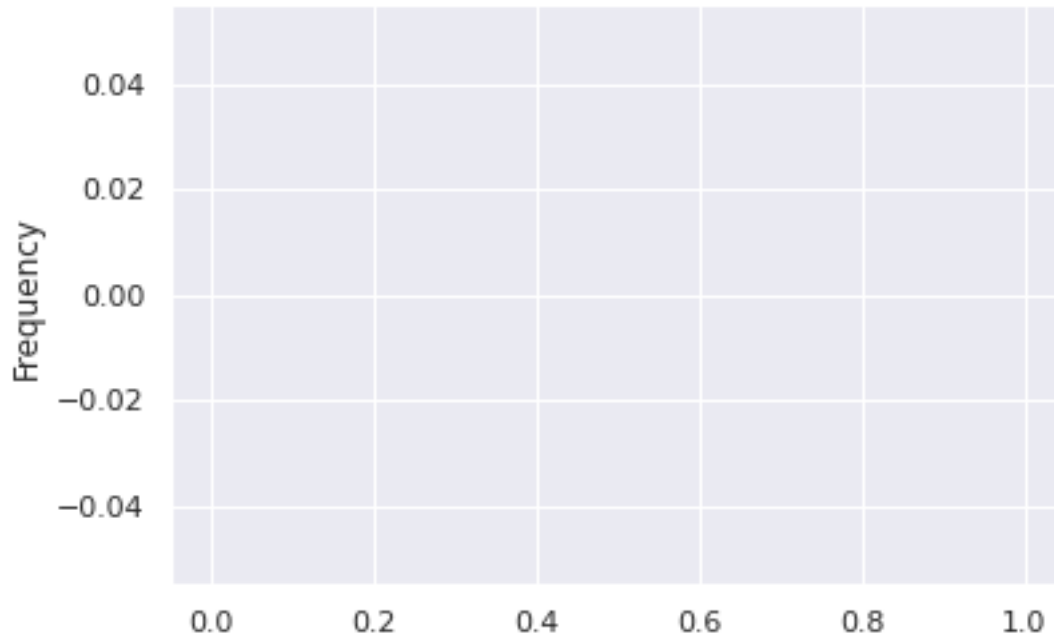
Inferred loglh worse: 0

Inferred n_reticulations less: 0

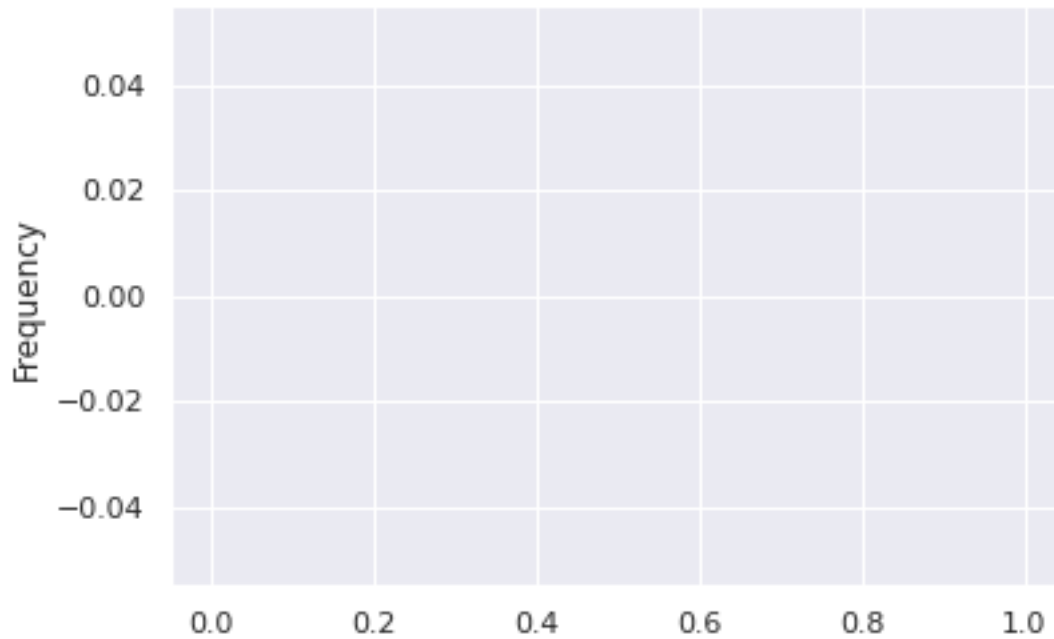
Inferred n_reticulations equal: 0

Inferred n_reticulations more: 0

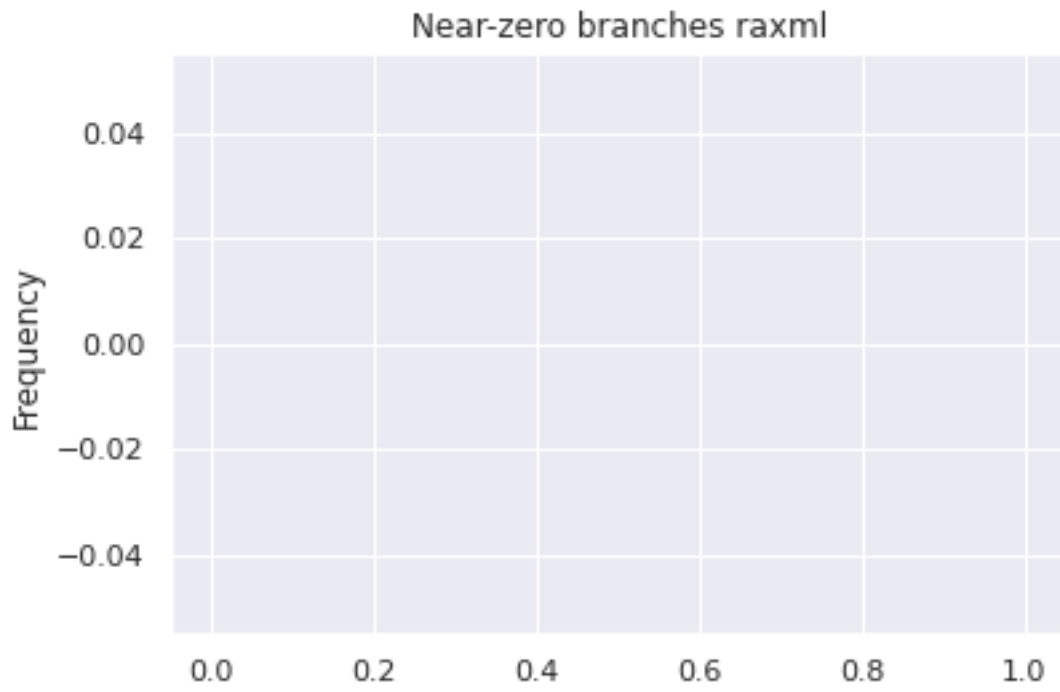
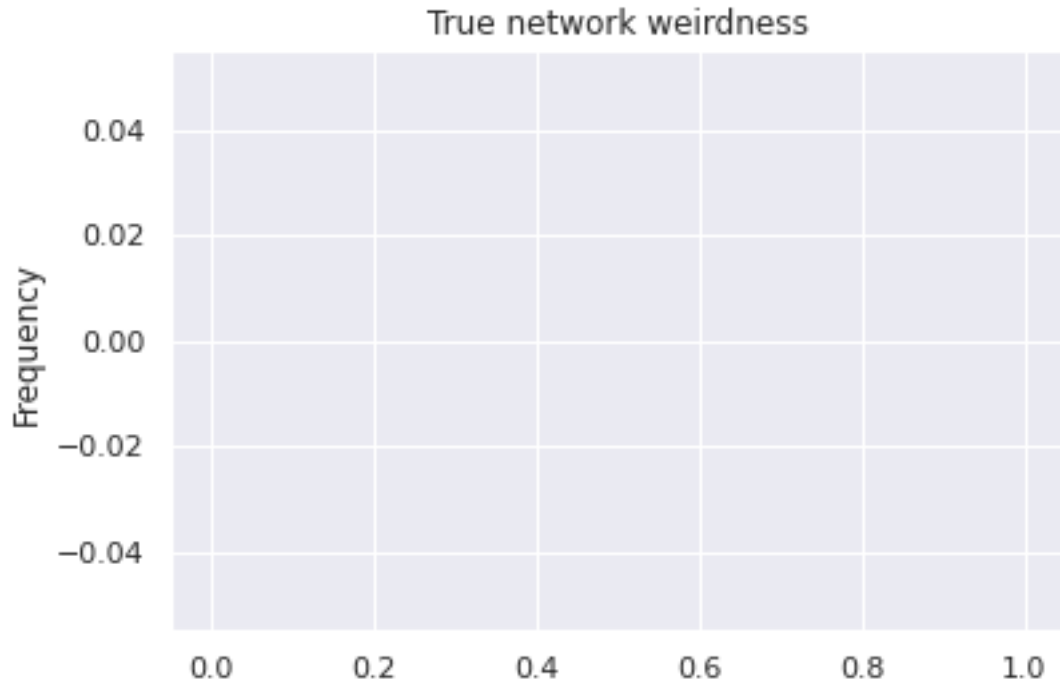
$(bic_true - bic_inferred) / bic_true$
value >0 means inferred BIC was better



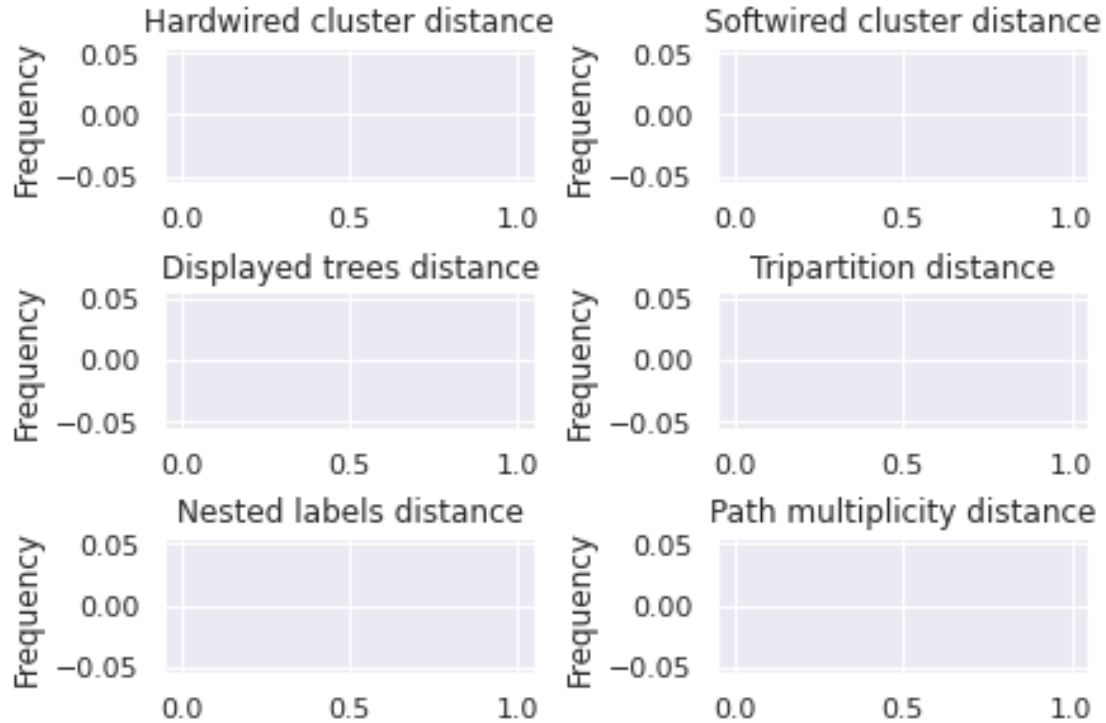
$(logl_true - logl_inferred) / logl_true$
value <0 means inferred logl was better



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



2.2.2 Plots for LikelihoodType.BEST

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

Inferred BIC better or equal: 0

Inferred BIC worse: 0

Inferred loglh better or equal: 0

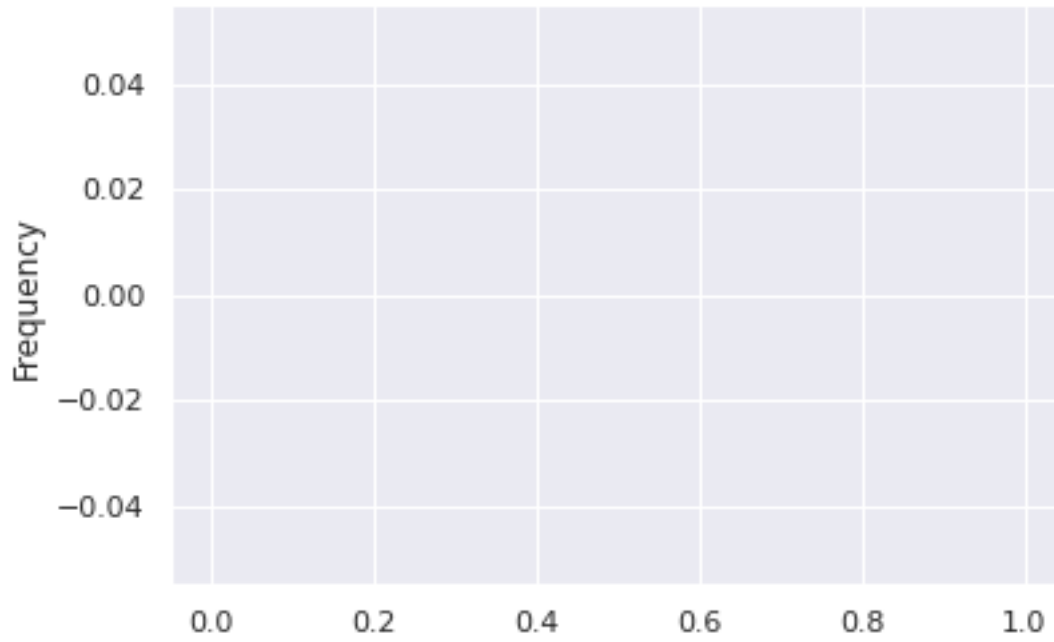
Inferred loglh worse: 0

Inferred n_reticulations less: 0

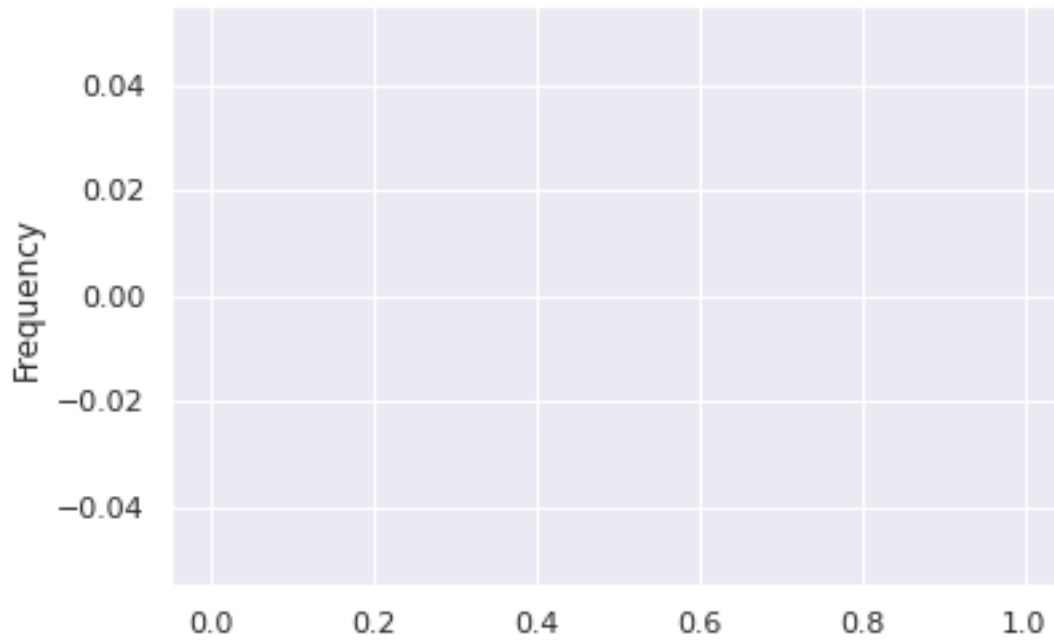
Inferred n_reticulations equal: 0

Inferred n_reticulations more: 0

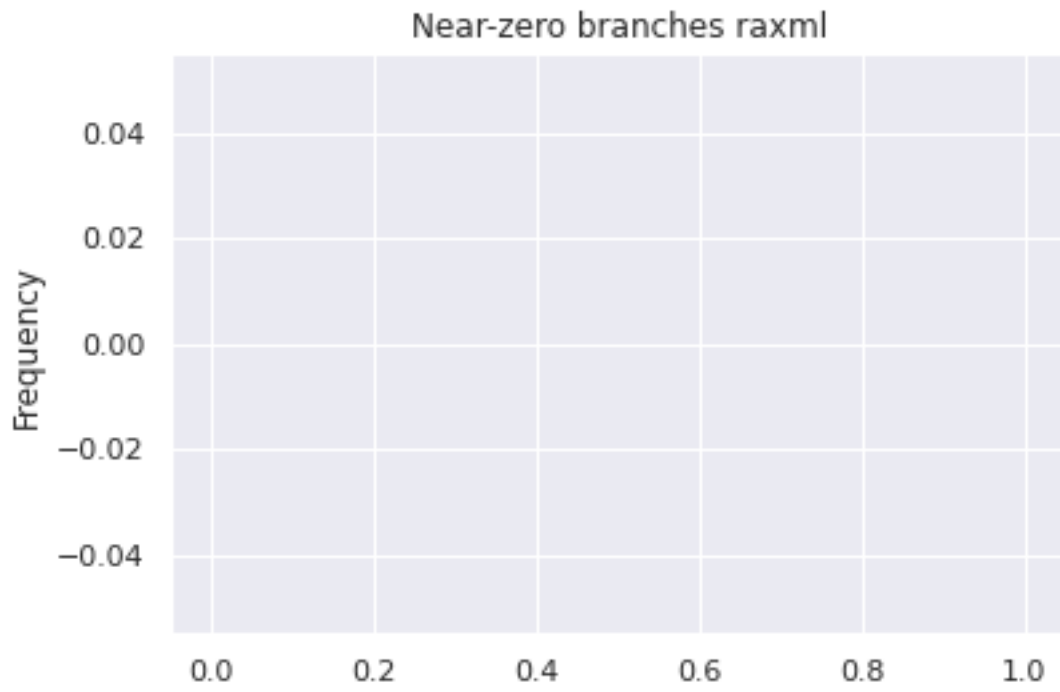
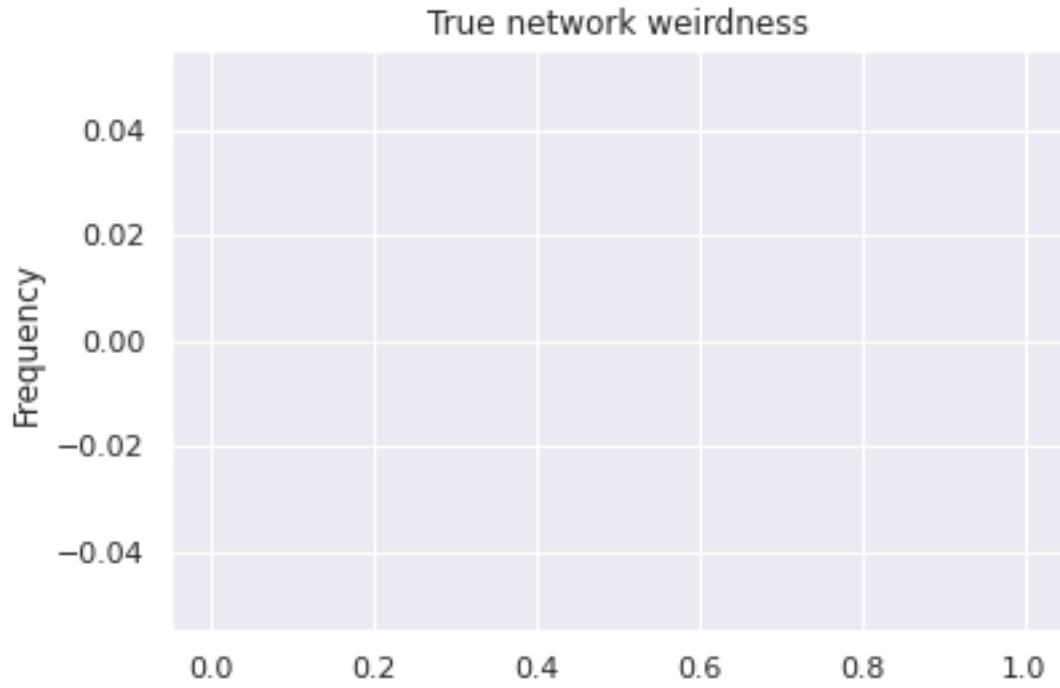
$(bic_true - bic_inferred) / bic_true$
value >0 means inferred BIC was better



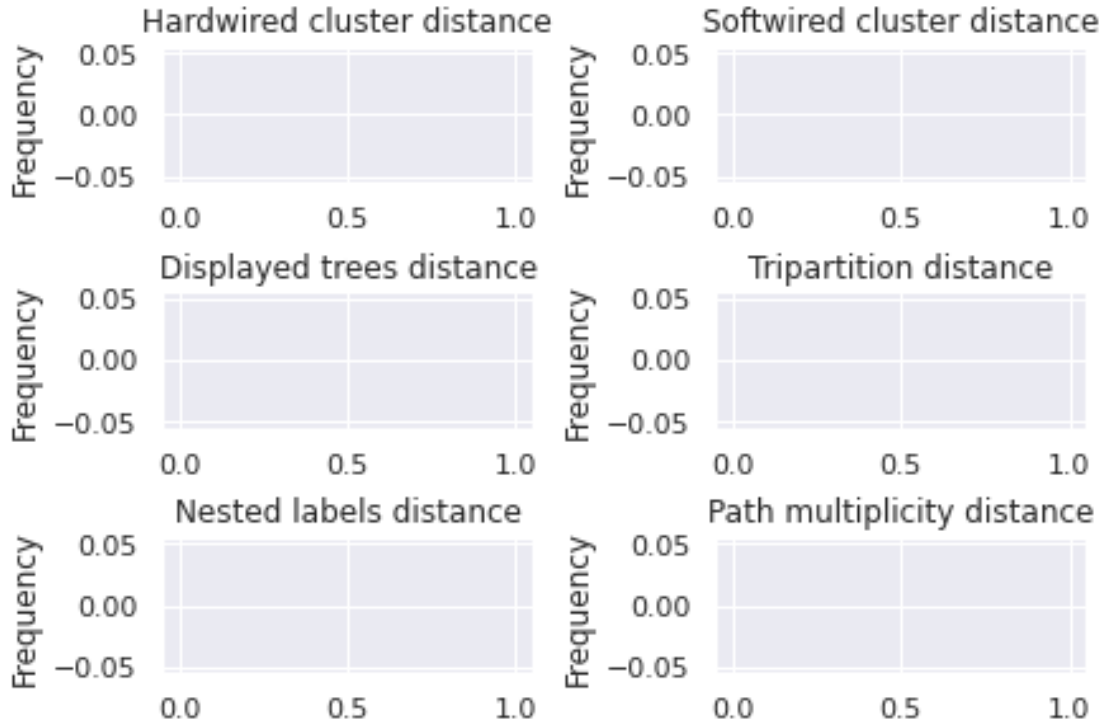
$(logl_true - logl_inferred) / logl_true$
value <0 means inferred logl was better



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



[]: