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_uniform_0_0/0…      26        4
     1  datasets_medium_network_norandom_uniform_0_0/0…      26        4
     2  datasets_medium_network_norandom_uniform_0_0/0…      26        4
     3  datasets_medium_network_norandom_uniform_0_0/0…      26        4
     4  datasets_medium_network_norandom_uniform_0_1/0…      22        2

        n_reticulations  msa_size     sampling_type simulation_type  \
     0                2       200  PERFECT_SAMPLING          CELINE
     1                2       200  PERFECT_SAMPLING          CELINE
     2                2       400  PERFECT_SAMPLING          CELINE
     3                2       400  PERFECT_SAMPLING          CELINE
     4                1       100  PERFECT_SAMPLING          CELINE

                                               celine_params  \
     0  {'to': 0.14860997437546947| 'lambda': 21.63797…
     1  {'to': 0.14860997437546947| 'lambda': 21.63797…
     2  {'to': 0.14860997437546947| 'lambda': 21.63797…
     3  {'to': 0.14860997437546947| 'lambda': 21.63797…
     4  {'to': 0.12766349262110696| 'lambda': 24.51074…

                           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                         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_medium_network_norandom_uniform_0_0/0…
     1  datasets_medium_network_norandom_uniform_0_0/0…
     2  datasets_medium_network_norandom_uniform_0_0/0…
     3  datasets_medium_network_norandom_uniform_0_0/0…
     4  datasets_medium_network_norandom_uniform_0_1/0…

                                       inferred_network_path likelihood_type  \
     0  datasets_medium_network_norandom_uniform_0_0/0…         AVERAGE
     1  datasets_medium_network_norandom_uniform_0_0/0…            BEST
     2  datasets_medium_network_norandom_uniform_0_0/0…         AVERAGE
     3  datasets_medium_network_norandom_uniform_0_0/0…            BEST
```

```
4  datasets_medium_network_norandom_uniform_0_1/0…         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          18670.640                       1.0
1                           0           1029.402                       0.0
2                           0           1217.179                       0.0
3                           0            709.402                       0.0
4                           0           2056.276                       0.0

      bic_true    logl_true  bic_inferred  logl_inferred    bic_raxml  \
0  3913.384732 -1554.540913   3963.151211   -1596.536980  3984.670570
1  3913.384533 -1554.540813   3984.670570   -1624.409487  3984.670570
2  7306.336325 -3218.438792   7490.636097   -3347.586922  7490.636097
3  7306.336325 -3218.438792   7490.636097   -3347.586922  7490.636097
4  1611.070345  -559.256368   1592.876235    -565.551738  1592.875978

    logl_raxml  rf_absolute_raxml  rf_relative_raxml  rf_absolute_inferred  \
0 -1624.409487                 -1                 -1                    -1
1 -1624.409487                 -1                 -1                    -1
2 -3347.586922                 -1                 -1                    -1
3 -3347.586922                 -1                 -1                    -1
4  -565.551610                 -1                 -1                    -1

   rf_relative_inferred  hardwired_cluster_distance  \
0                    -1                        12.0
1                    -1                         9.0
2                    -1                        10.0
3                    -1                        10.0
4                    -1                         7.0

   softwired_cluster_distance  displayed_trees_distance  \
0                        14.5                       2.5
1                        11.5                       2.5
2                         9.5                       2.5
3                         9.5                       2.5
4                         7.5                       1.5

   tripartition_distance  nested_labels_distance  path_multiplicity_distance  \
0                   14.5                    17.0                        10.5
1                   14.0                    16.0                        11.0
```

```
2                  13.0                    15.0                       12.0
3                  13.0                    15.0                       12.0
4                   8.5                    12.0                        9.0

   bic_diff  logl_diff
0 -0.012717  -0.027015
1 -0.018216  -0.044945
2 -0.025225  -0.040128
3 -0.025225  -0.040128
4  0.011293  -0.011257
```

[5]:

```
df.columns
```

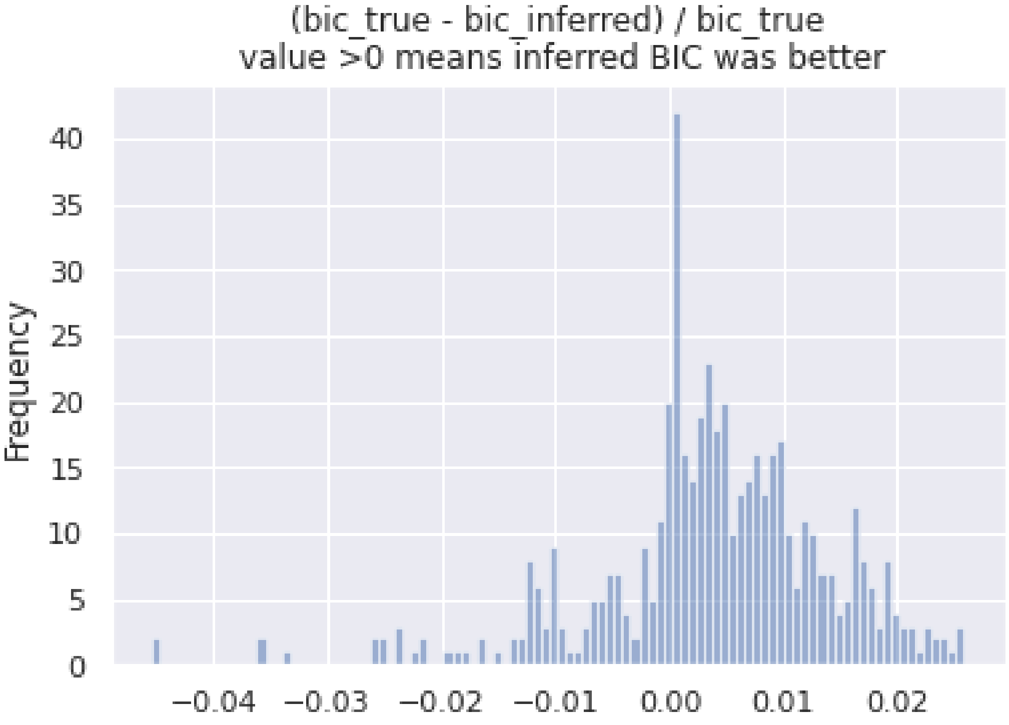
[5]:

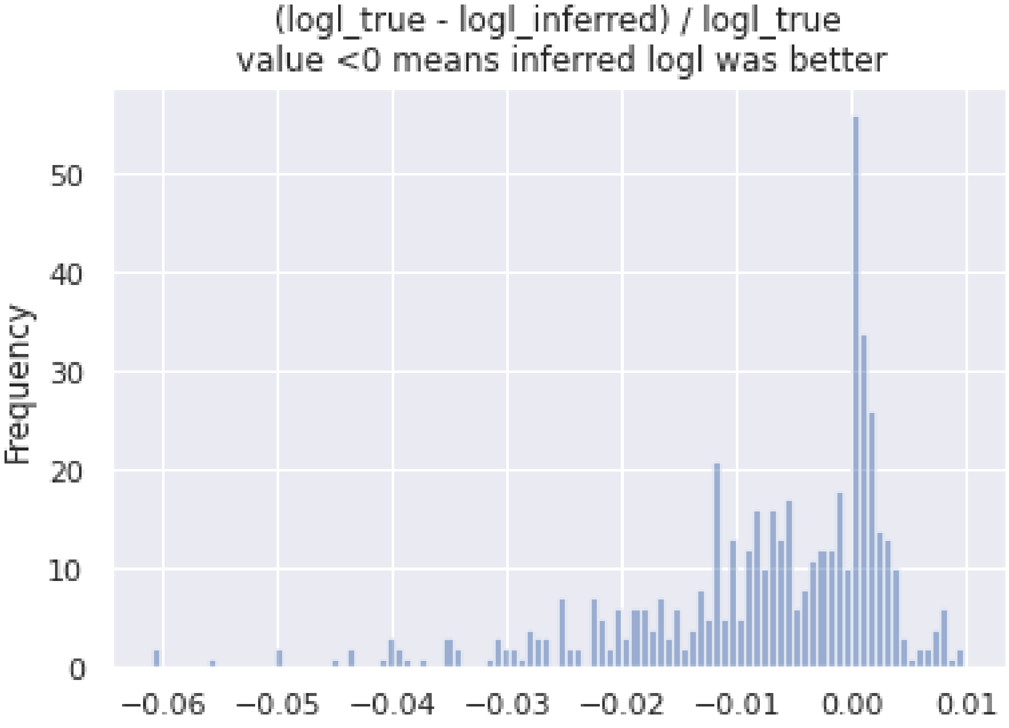
```
Index(['name', 'n_taxa', 'n_trees', 'n_reticulations', 'msa_size',
       'sampling_type', 'simulation_type', 'celine_params', 'seqgen_params',
       'near_zero_branches_raxml', 'n_equal_tree_pairs',
       'true_network_weirdness', 'true_network_path', 'inferred_network_path',
       'likelihood_type', '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', 'bic_diff', 'logl_diff'],
      dtype='object')
```

[6]:

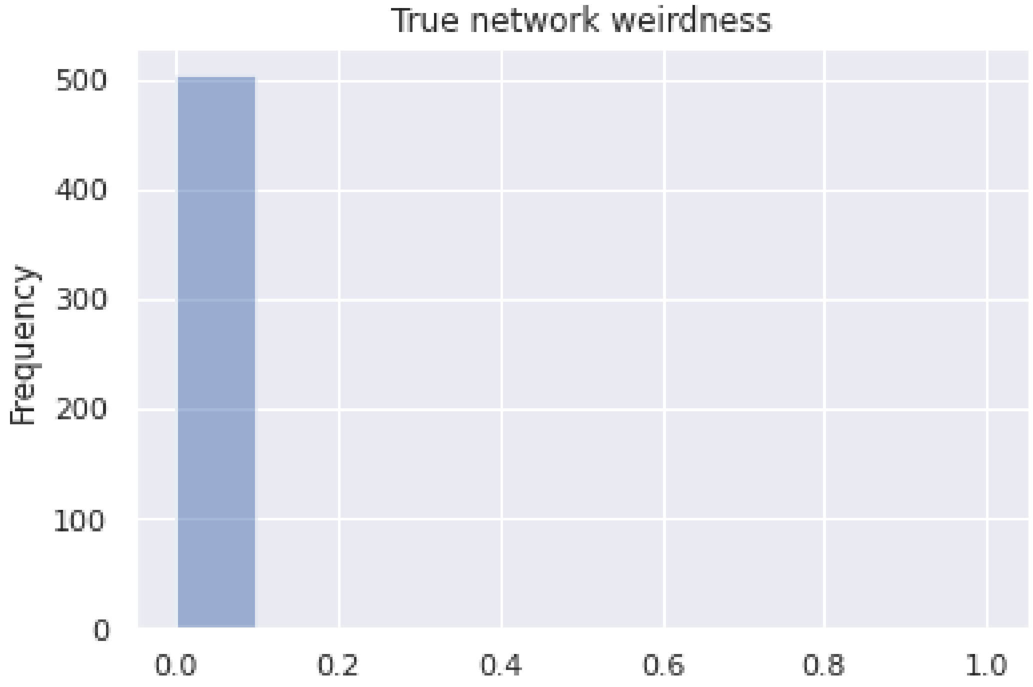
```
build_stats(df)
```

```
Inferred BIC better or equal: 382
Inferred BIC worse: 122

Inferred loglh better or equal: 168
Inferred loglh worse: 336

Inferred n_reticulations less: 415
Inferred n_reticulations equal: 89
Inferred n_reticulations more: 0
```

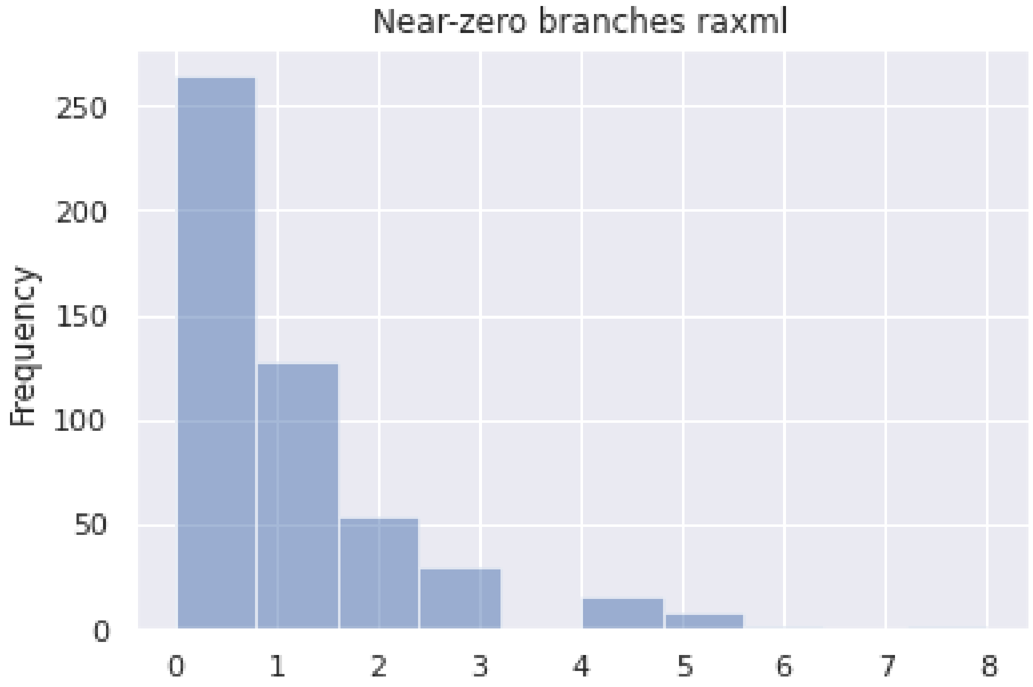

(bic_true - bic_inferred) / bic_true
value >0 means inferred BIC was better
Frequency
0
5
10
15
20
25
30
35
40
−0.04
−0.03
−0.02
−0.01
0.00
0.01
0.02


(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0
10
20
30
40
50
−0.06
−0.05
−0.04
−0.03
−0.02
−0.01
0.00
0.01

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

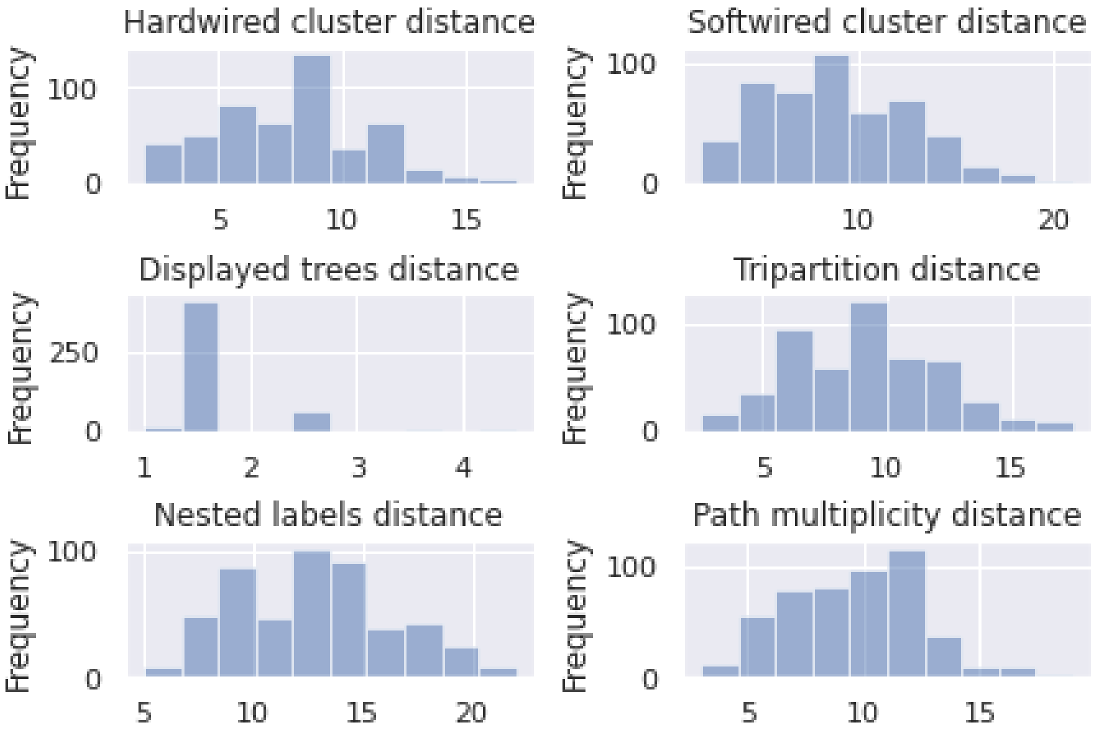


True network weirdness



Near-zero branches raxml

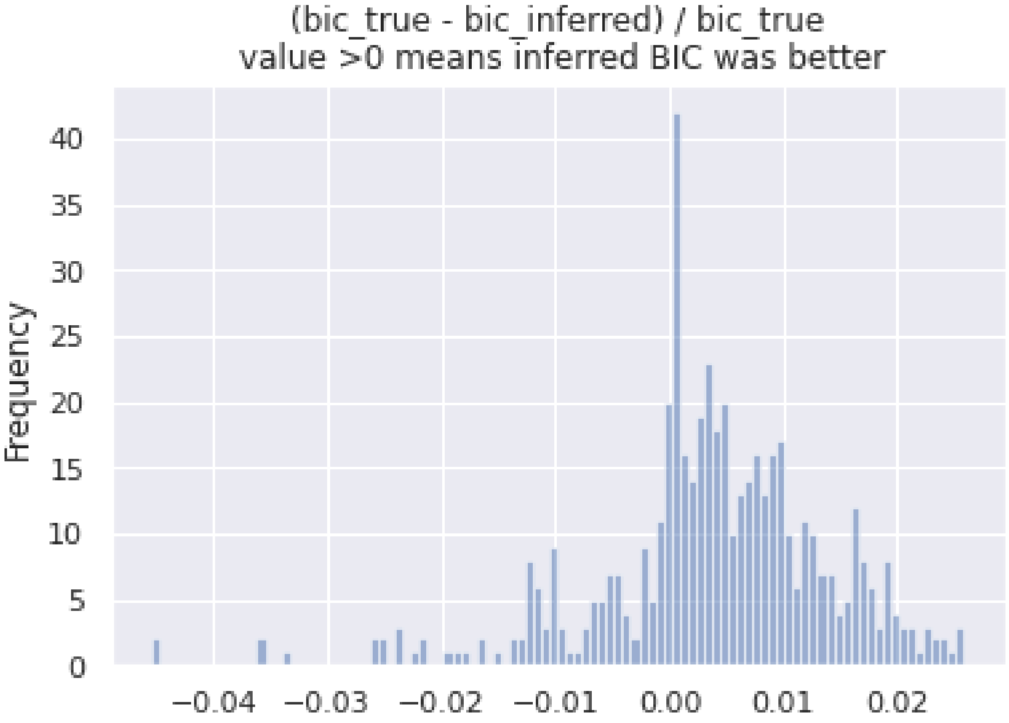
```
<Figure size 432x288 with 0 Axes>
```

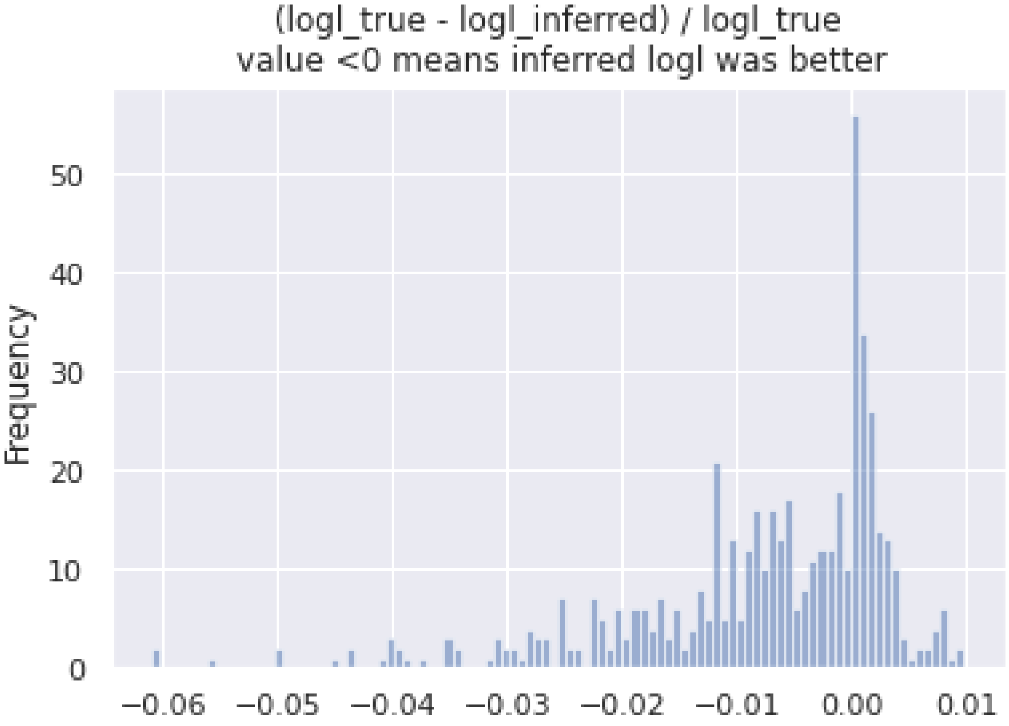


1 Plots for starting with raxml-ng best tree only

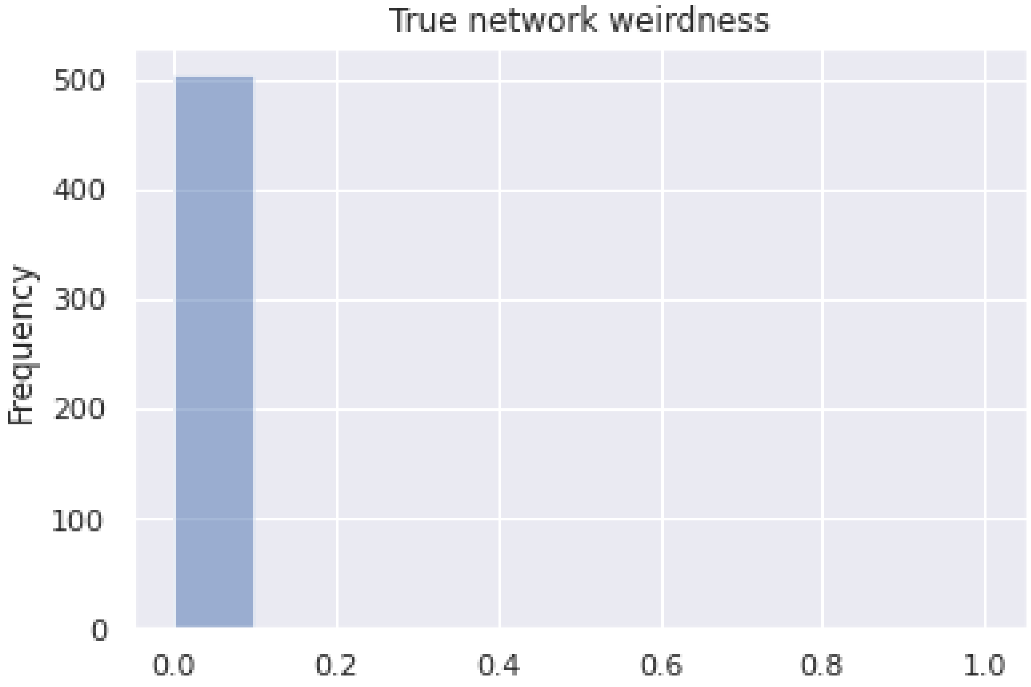
[7]:
```
df_raxml_only = df.query('start_type == "FROM_RAXML"')
build_stats(df_raxml_only)
```

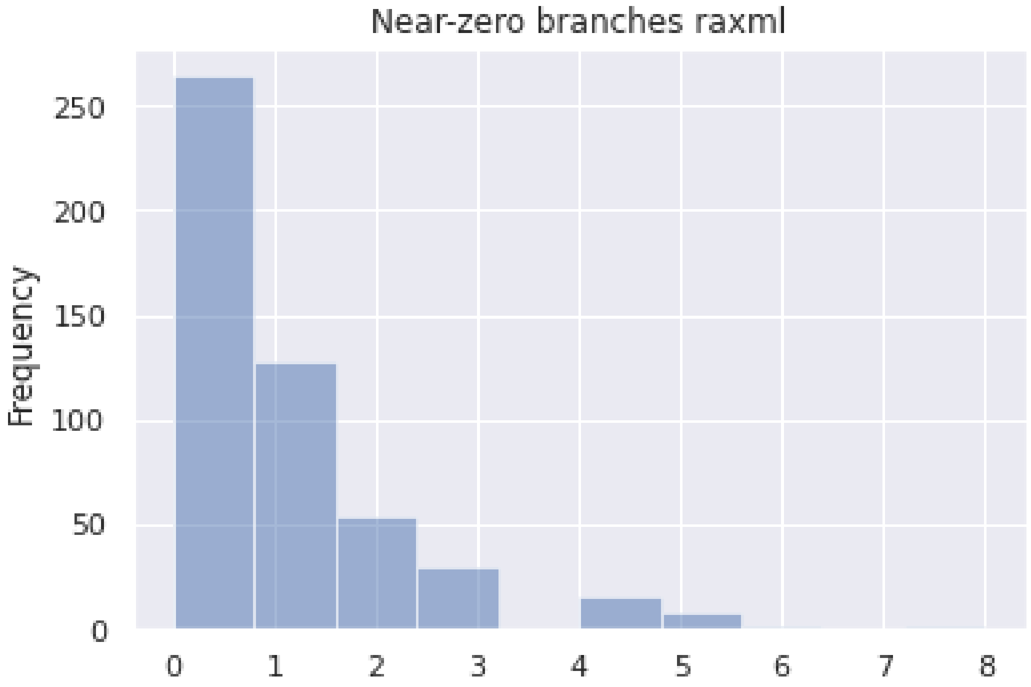
```
Inferred BIC better or equal: 382
Inferred BIC worse: 122

Inferred loglh better or equal: 168
Inferred loglh worse: 336

Inferred n_reticulations less: 415
Inferred n_reticulations equal: 89
Inferred n_reticulations more: 0
```


(bic_true - bic_inferred) / bic_true
value >0 means inferred BIC was better
Frequency
0
5
10
15
20
25
30
35
40
−0.04
−0.03
−0.02
−0.01
0.00
0.01
0.02

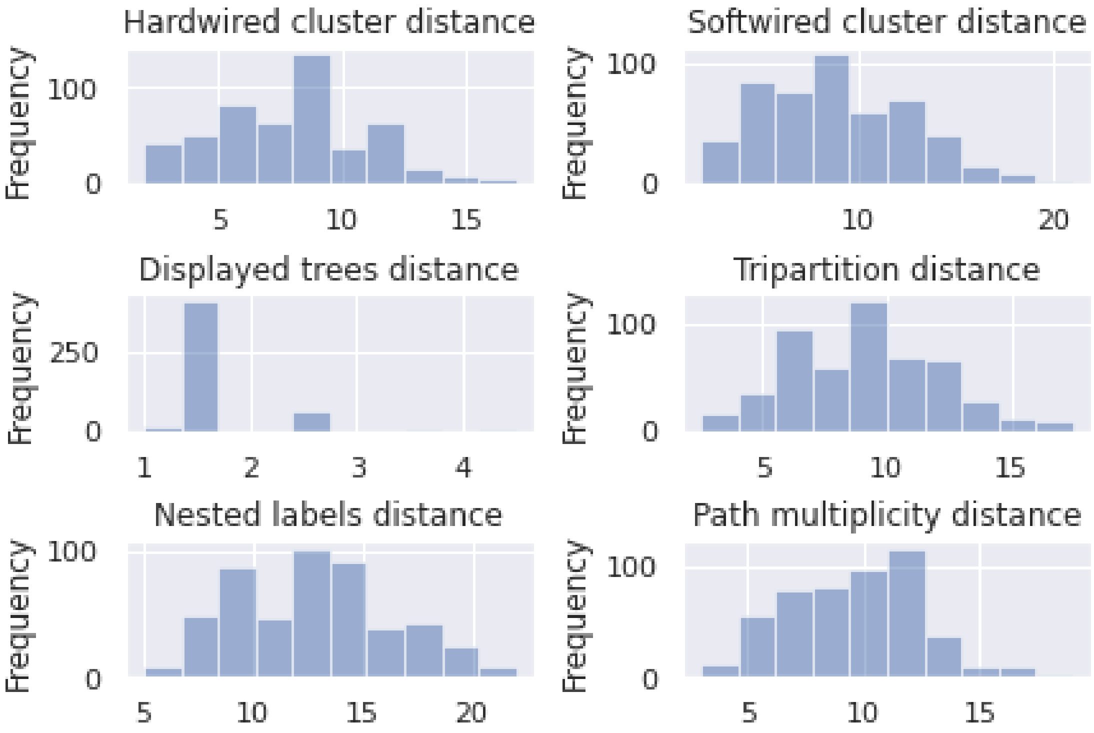

(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0
10
20
30
40
50
−0.06
−0.05
−0.04
−0.03
−0.02
−0.01
0.00
0.01

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


True network weirdness


Near-zero branches raxml

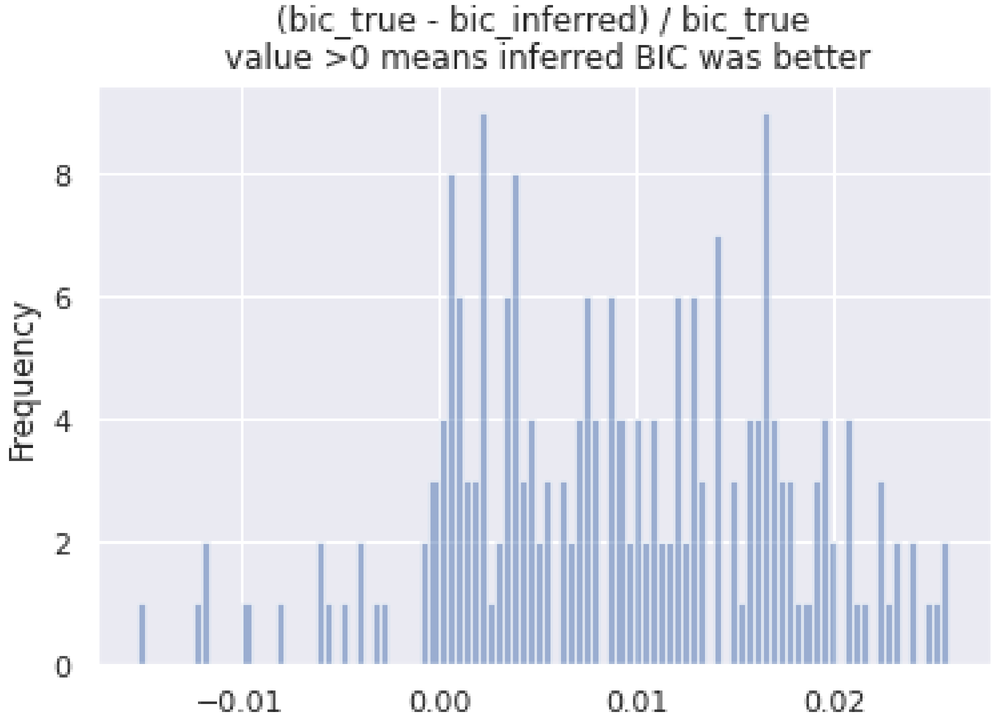
```
<Figure size 432x288 with 0 Axes>
```

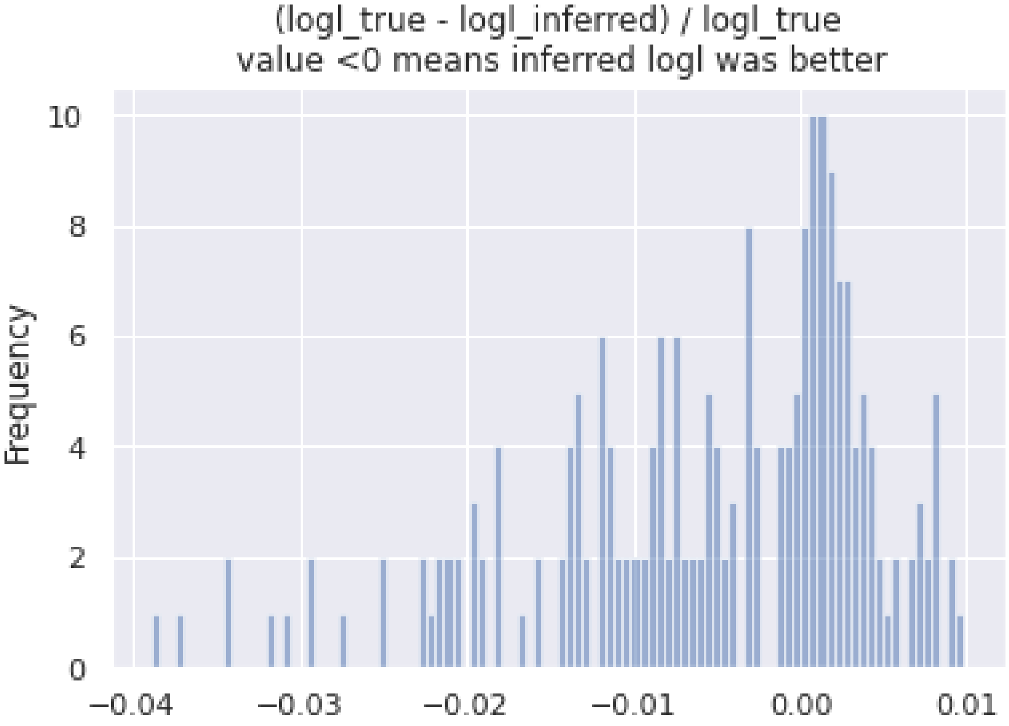


1.1 Plots for MSA_size ~ 100*n_trees

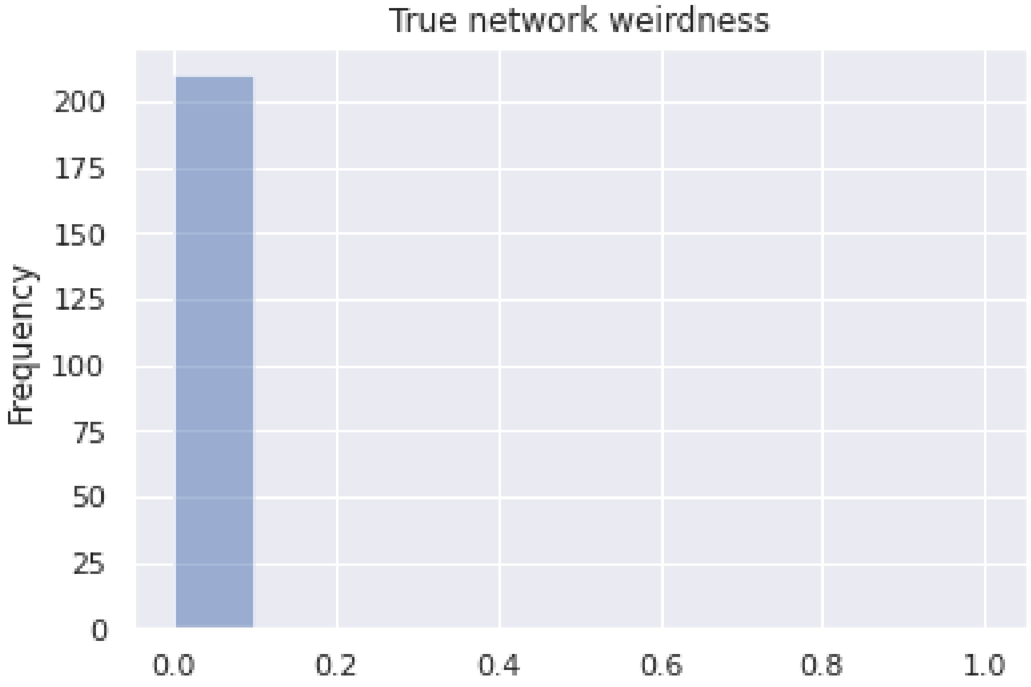
[8]:
```
df_raxml_only_msasize_100 = df_raxml_only.query('msa_size == 100')
build_stats(df_raxml_only_msasize_100)
```

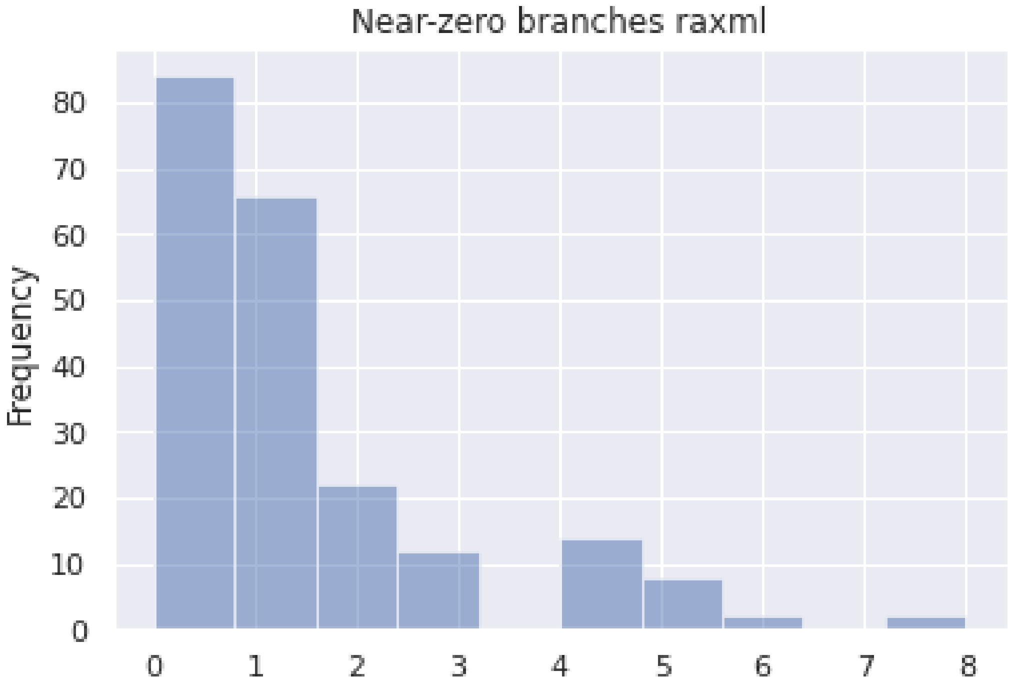
```
Inferred BIC better or equal: 190
Inferred BIC worse: 20

Inferred loglh better or equal: 87
Inferred loglh worse: 123

Inferred n_reticulations less: 179
Inferred n_reticulations equal: 31
Inferred n_reticulations more: 0
```


(bic_true - bic_inferred) / bic_true
value >0 means inferred BIC was better
Frequency
0
2
4
6
8
−0.01
0.00
0.01
0.02

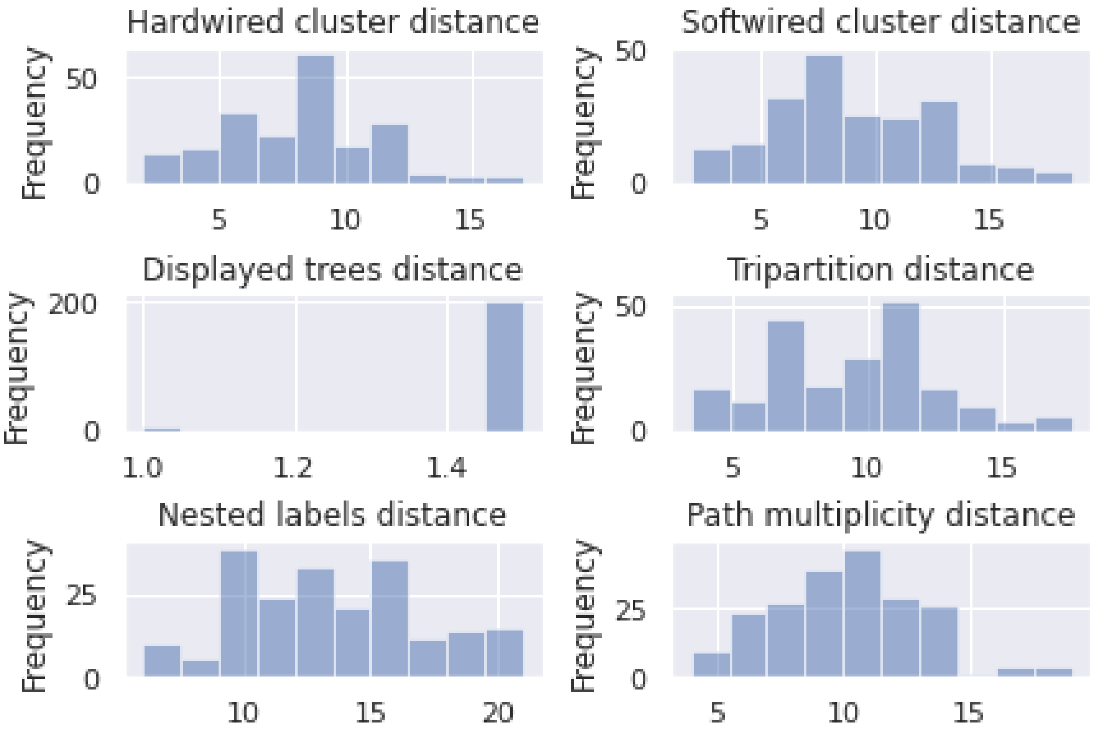

(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0
2
4
6
8
10
−0.04
−0.03
−0.02
−0.01
0.00
0.01

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


True network weirdness
Frequency
0
25
50
75
100
125
150
175
200
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
0
10
20
30
40
50
60
70
80
0
1
2
3
4
5
6
7
8

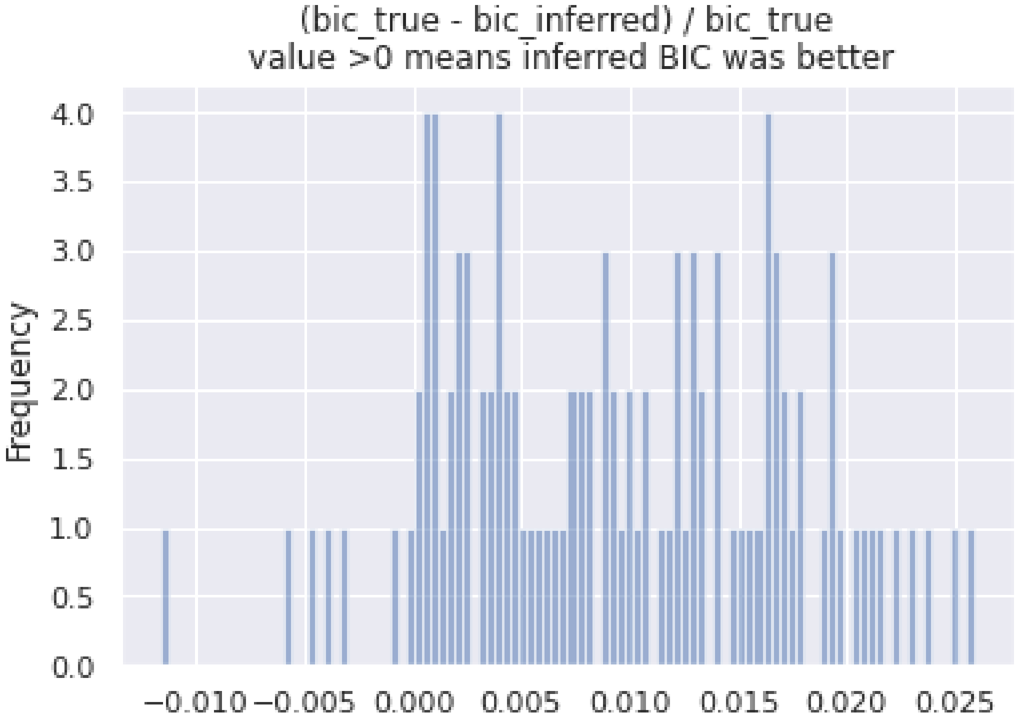
```
<Figure size 432x288 with 0 Axes>
```

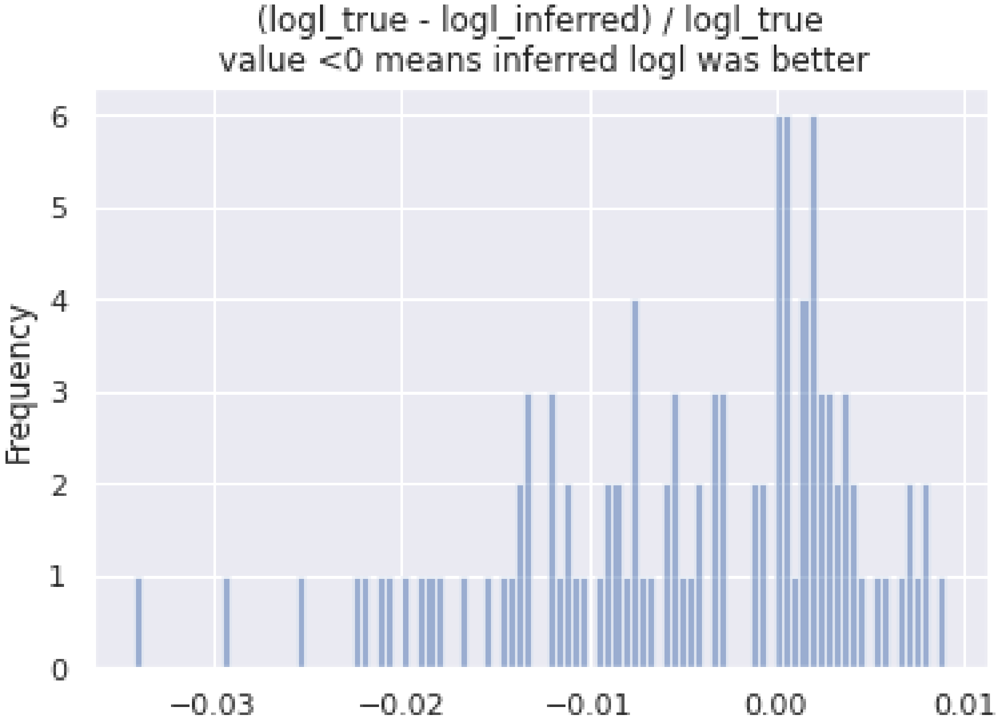


1.1.1 Plots for LikelihodType.AVERAGE

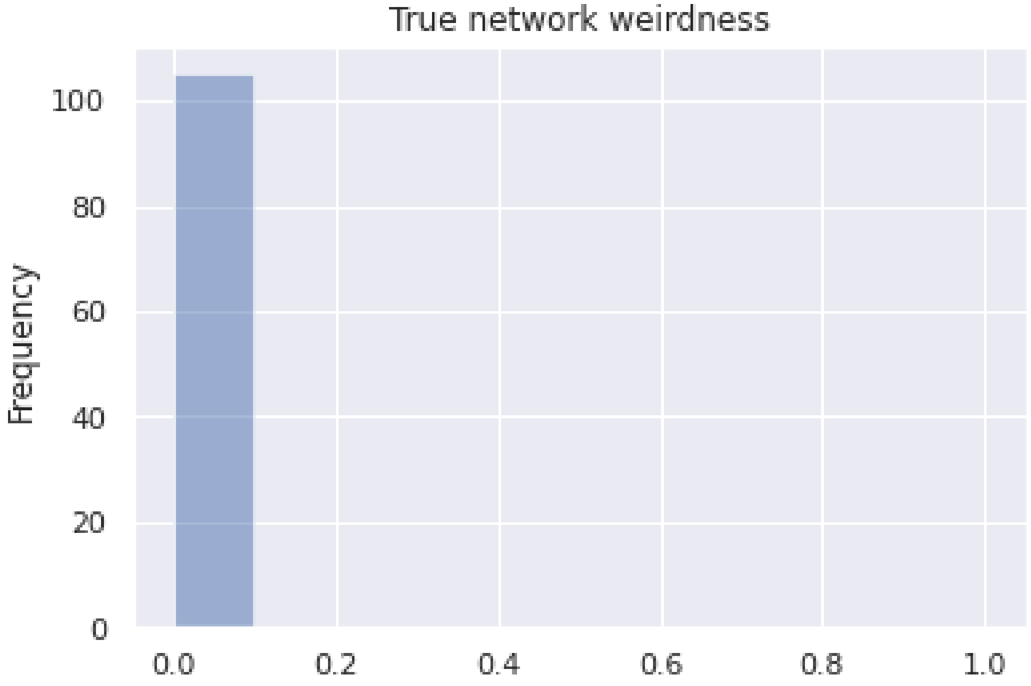
```
[9]: df_raxml_only_msasize_100_average = df_raxml_only_msasize_100.
      ↪query('likelihood_type == "AVERAGE"')
     build_stats(df_raxml_only_msasize_100_average)
```

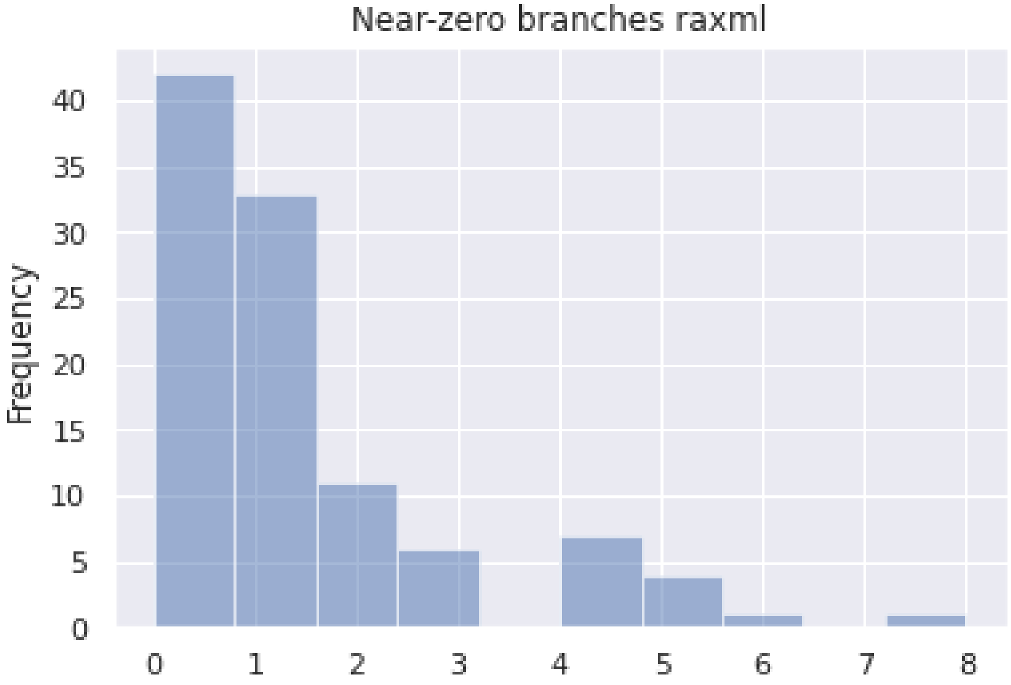
```
Inferred BIC better or equal: 98
Inferred BIC worse: 7

Inferred loglh better or equal: 45
Inferred loglh worse: 60

Inferred n_reticulations less: 87
Inferred n_reticulations equal: 18
Inferred n_reticulations more: 0
```


(bic_true - bic_inferred) / bic_true
value >0 means inferred BIC was better
Frequency
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.010
−0.005
0.000
0.005
0.010
0.015
0.020
0.025

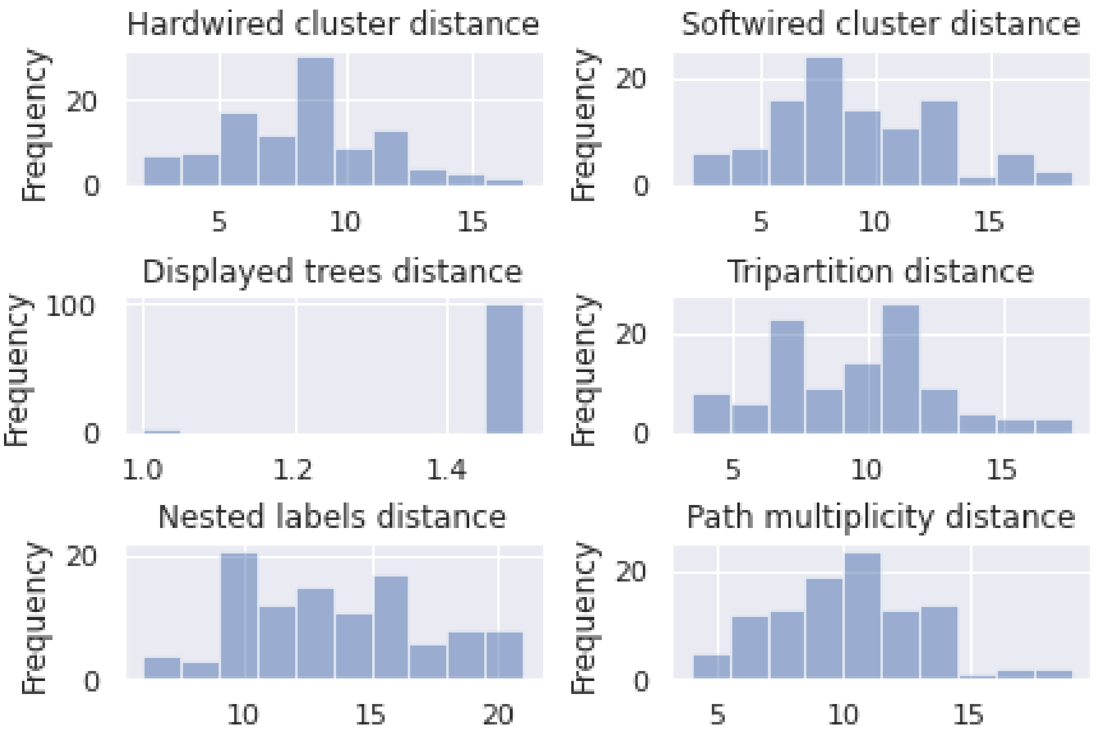

(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
6
5
4
3
2
1
0
−0.03
−0.02
−0.01
0.00
0.01

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


True network weirdness
Frequency
100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
40
35
30
25
20
15
10
5
0
0
1
2
3
4
5
6
7
8

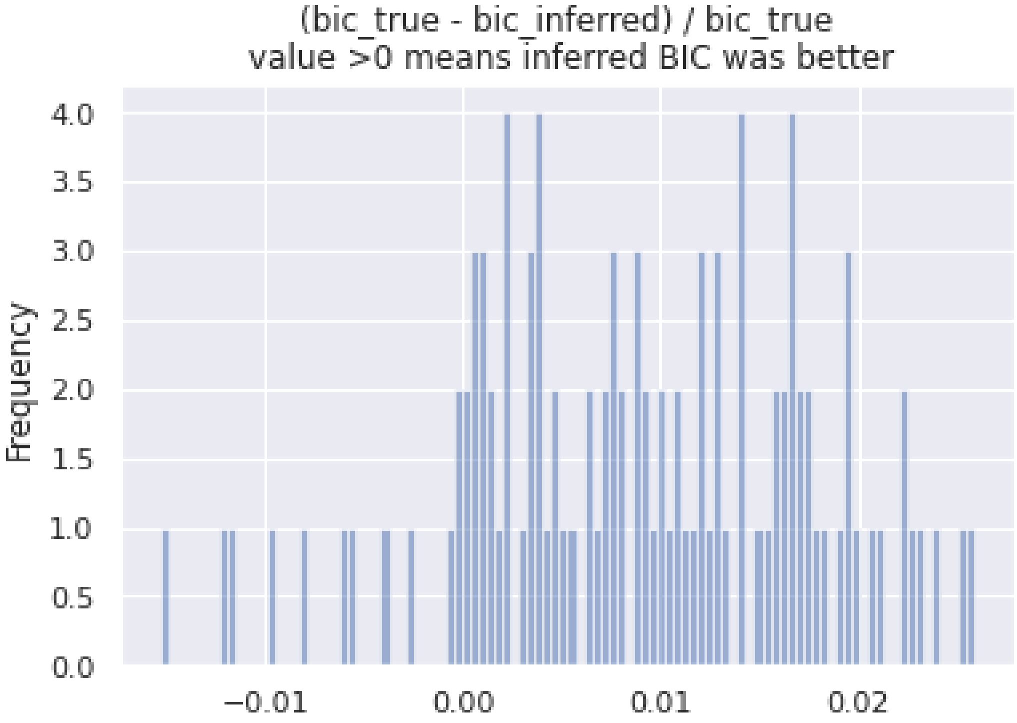
```
<Figure size 432x288 with 0 Axes>
```

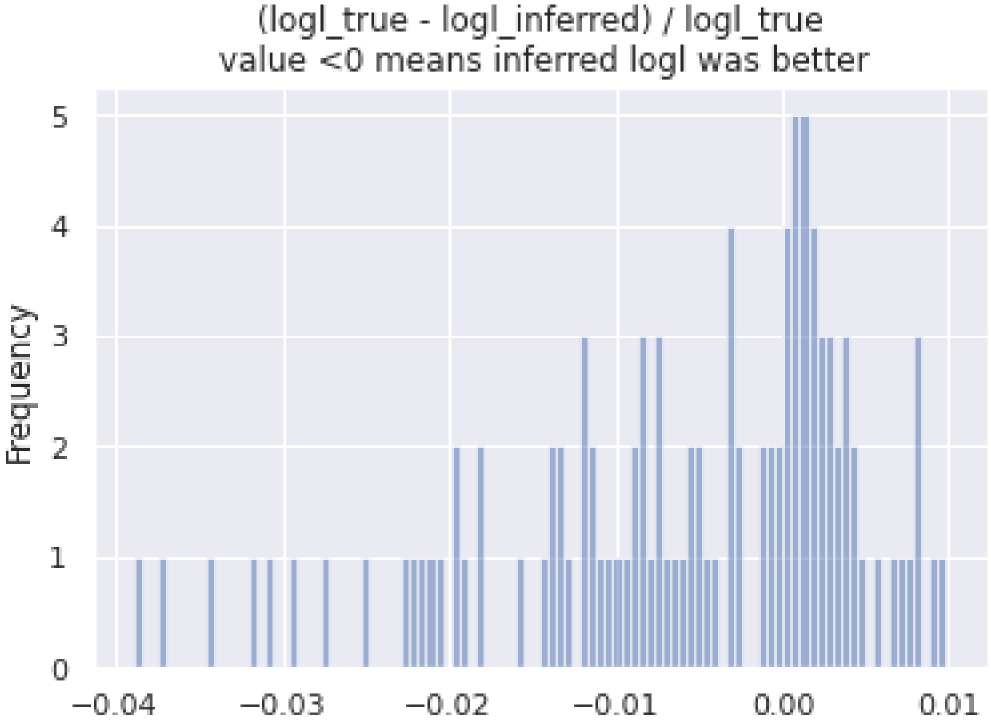


1.1.2 Plots for LikelihoodType.BEST

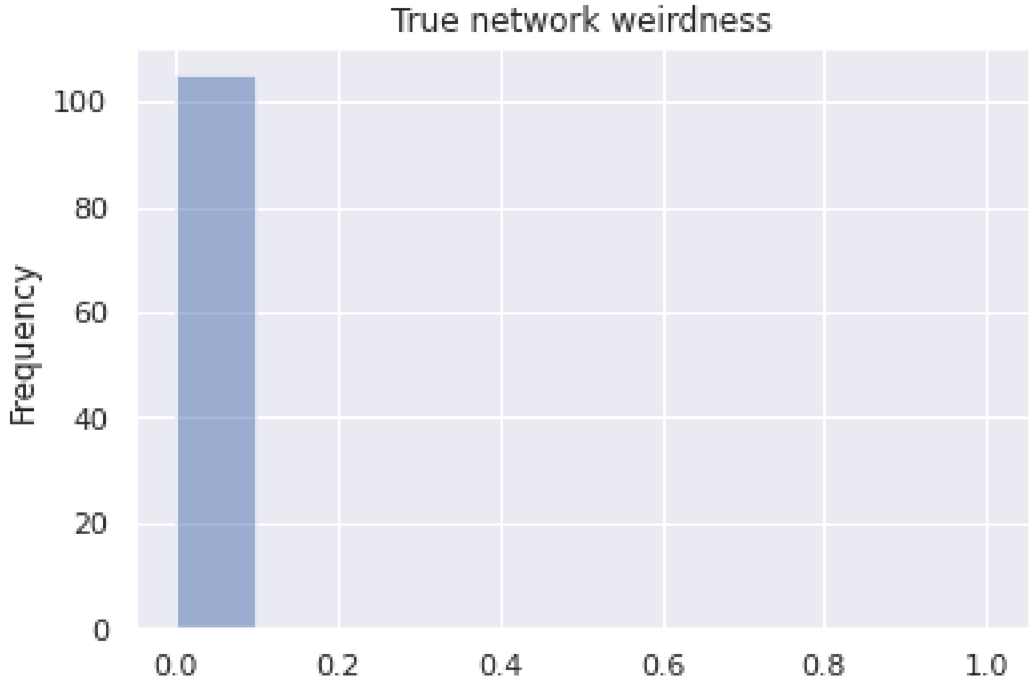
[10]:
```
df_raxml_only_msasize_100_best = df_raxml_only_msasize_100.
 ↪query('likelihood_type == "BEST"')
build_stats(df_raxml_only_msasize_100_best)
```

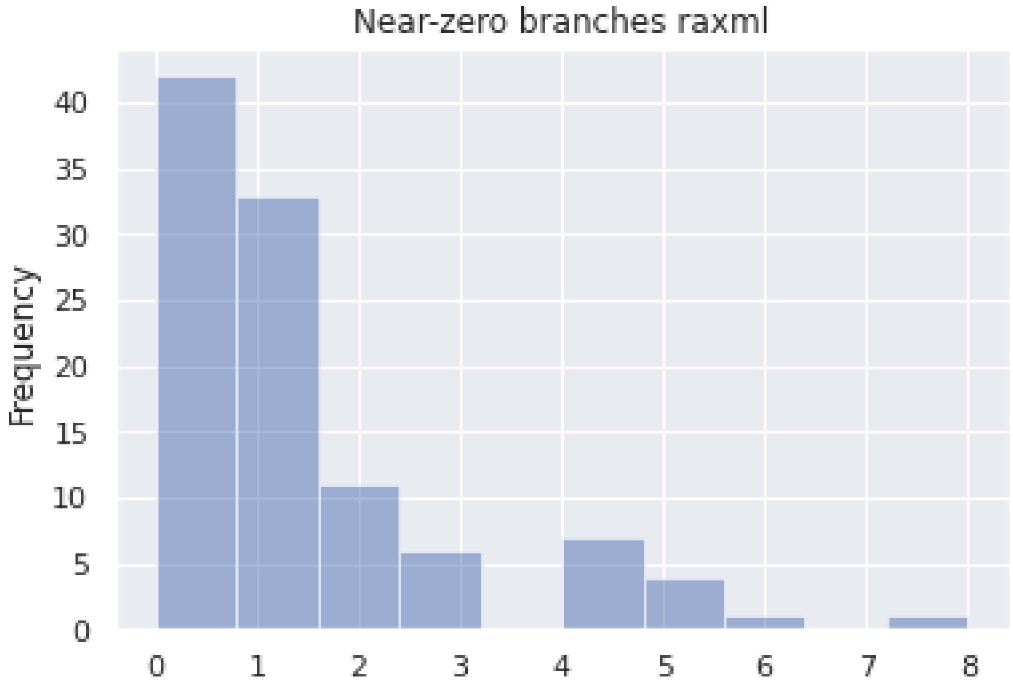
```
Inferred BIC better or equal: 92
Inferred BIC worse: 13

Inferred loglh better or equal: 42
Inferred loglh worse: 63

Inferred n_reticulations less: 92
Inferred n_reticulations equal: 13
Inferred n_reticulations more: 0
```


(bic_true - bic_inferred) / bic_true
value >0 means inferred BIC was better
Frequency
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.01
0.00
0.01
0.02

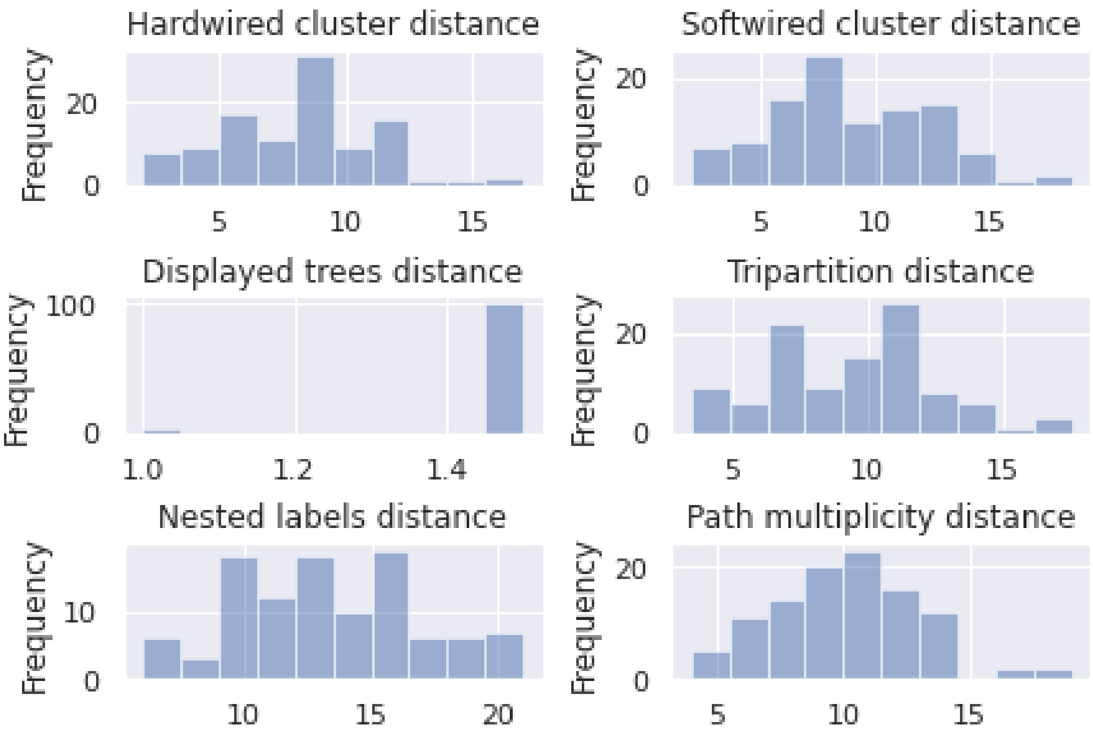

(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
5
4
3
2
1
0
−0.04
−0.03
−0.02
−0.01
0.00
0.01

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


True network weirdness
Frequency
0
20
40
60
80
100
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
0
5
10
15
20
25
30
35
40
0
1
2
3
4
5
6
7
8

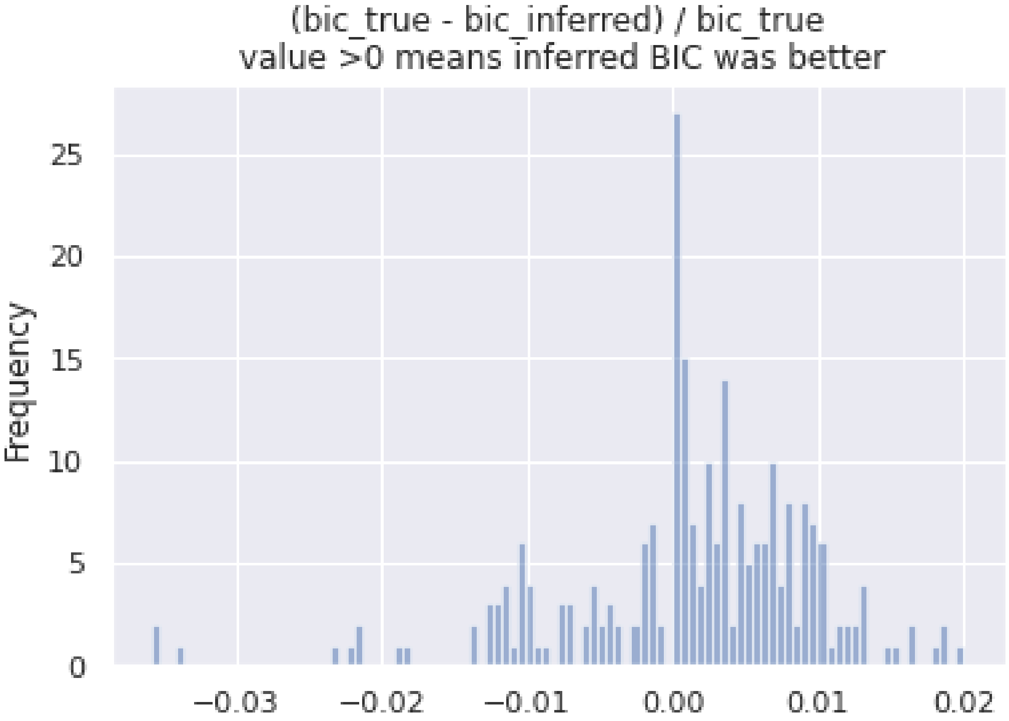
```
<Figure size 432x288 with 0 Axes>
```

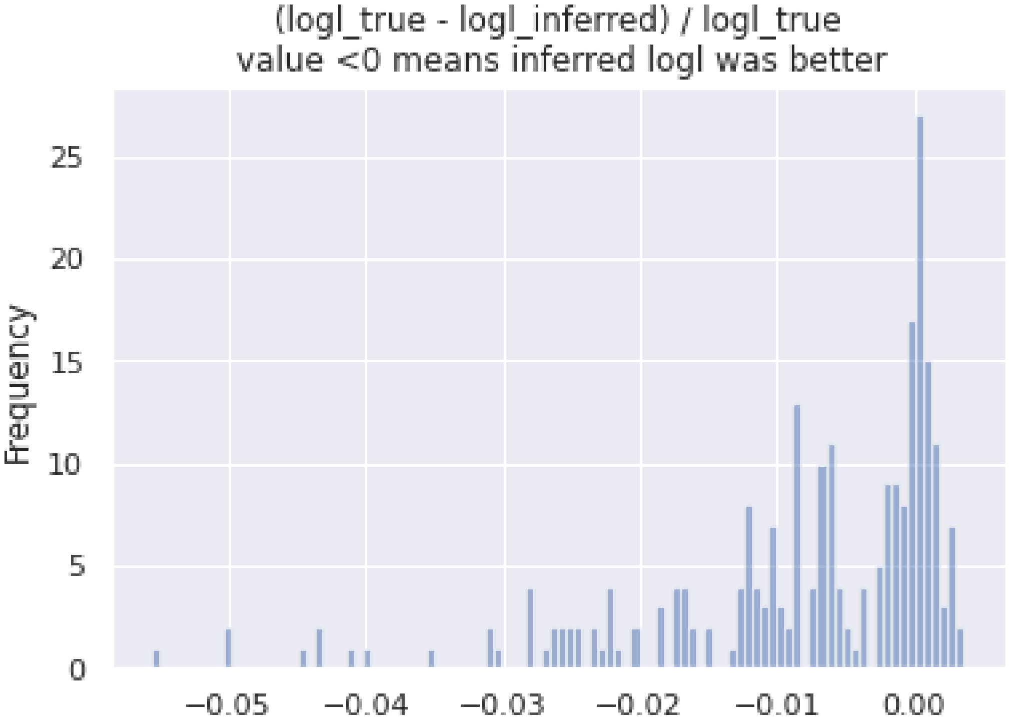


1.2 Plots for MSA_size ~ 200*n_trees

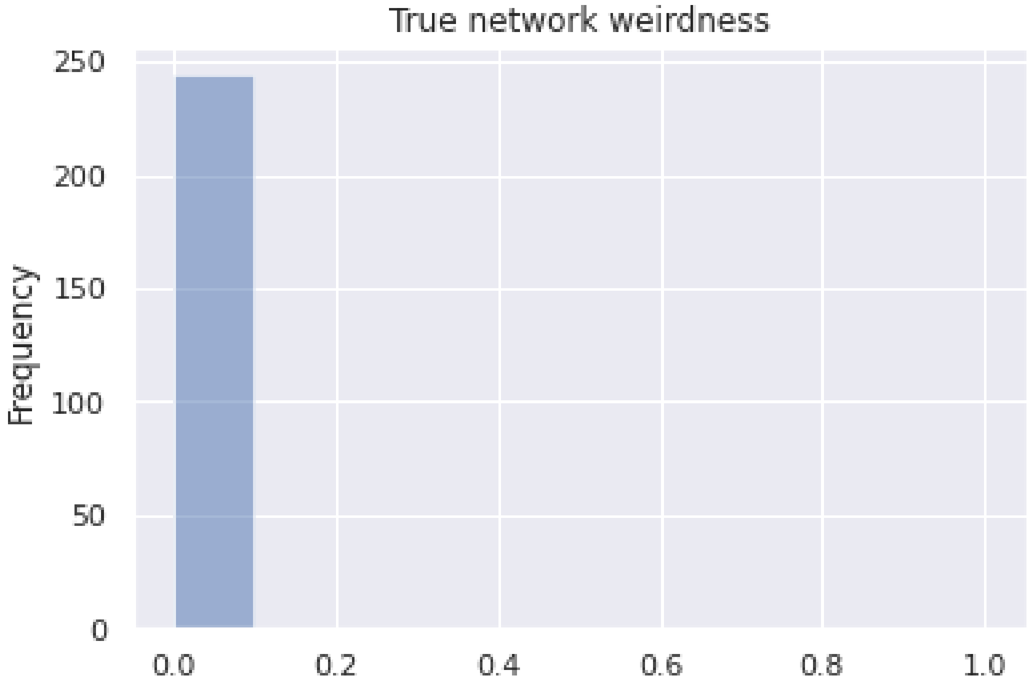
[11]:
```
df_raxml_only_msasize_200 = df_raxml_only.query('msa_size == 200')
build_stats(df_raxml_only_msasize_200)
```

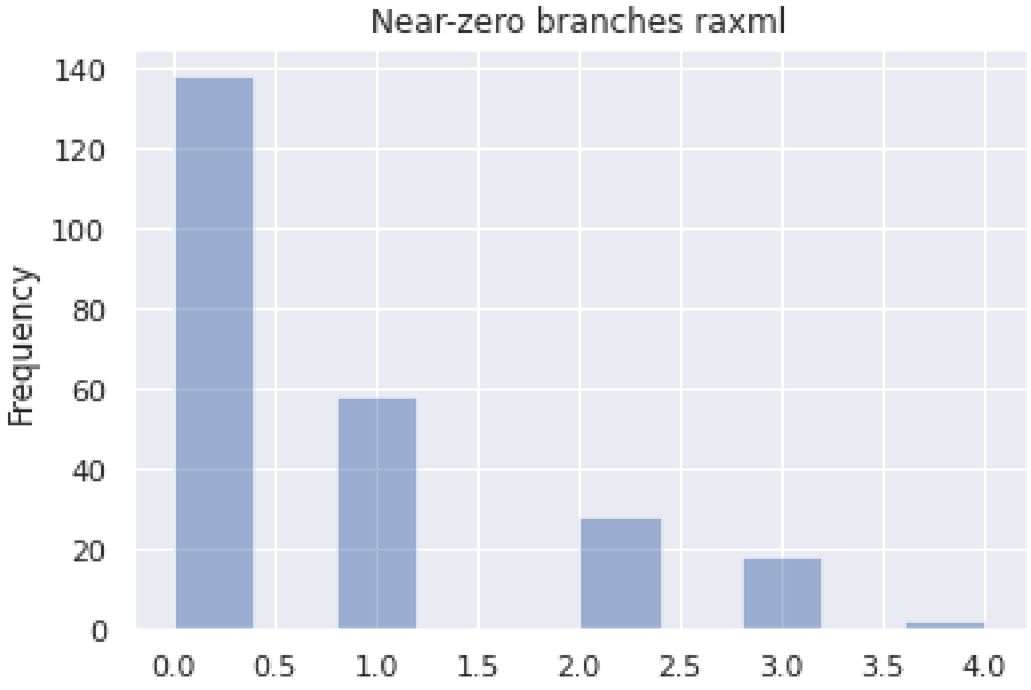
```
Inferred BIC better or equal: 170
Inferred BIC worse: 74

Inferred loglh better or equal: 75
Inferred loglh worse: 169

Inferred n_reticulations less: 187
Inferred n_reticulations equal: 57
Inferred n_reticulations more: 0
```


(bic_true - bic_inferred) / bic_true
value >0 means inferred BIC was better
Frequency
0
5
10
15
20
25
−0.03
−0.02
−0.01
0.00
0.01
0.02

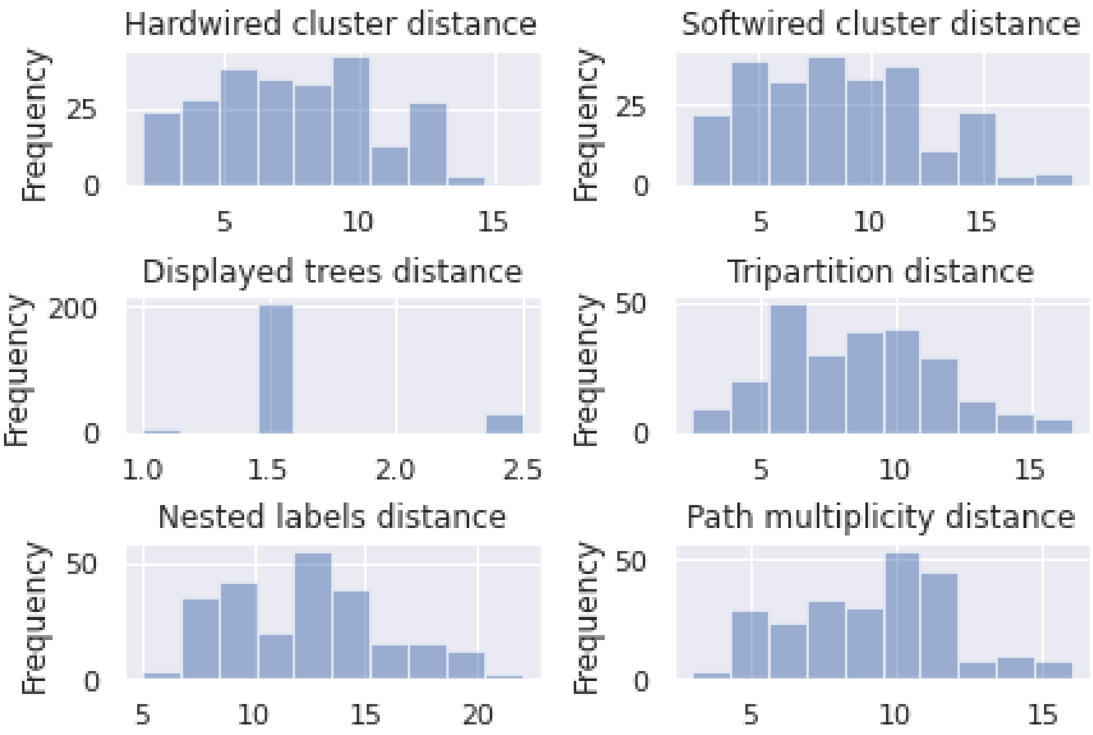

(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0
5
10
15
20
25
−0.05
−0.04
−0.03
−0.02
−0.01
0.00

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


True network weirdness
Frequency
250
200
150
100
50
0
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
140
120
100
80
60
40
20
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

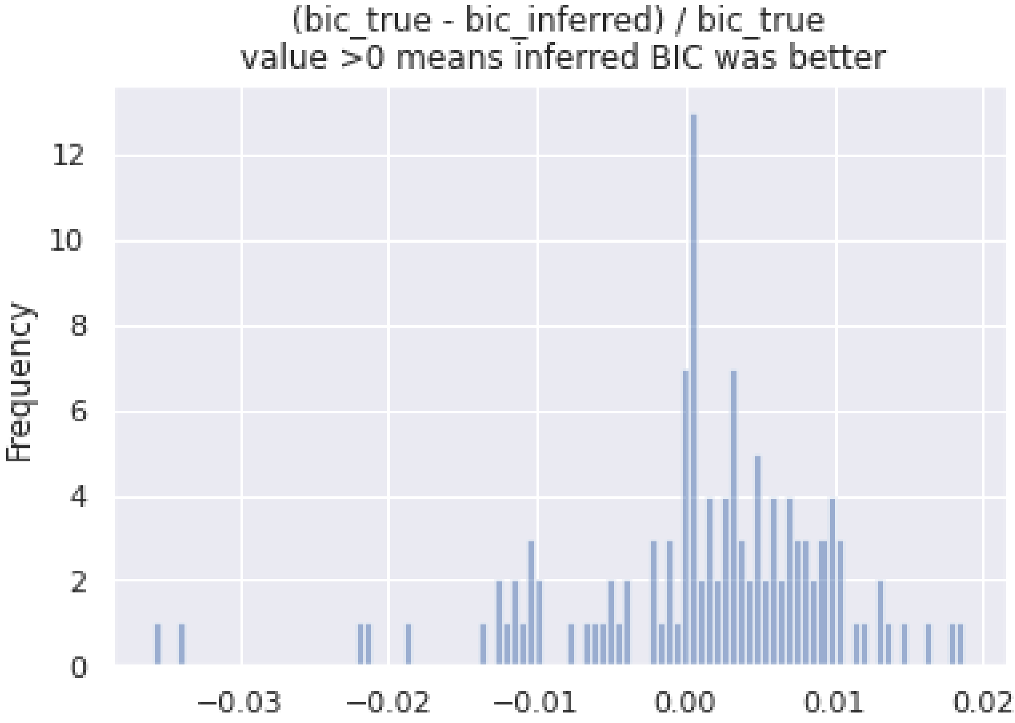
```
<Figure size 432x288 with 0 Axes>
```

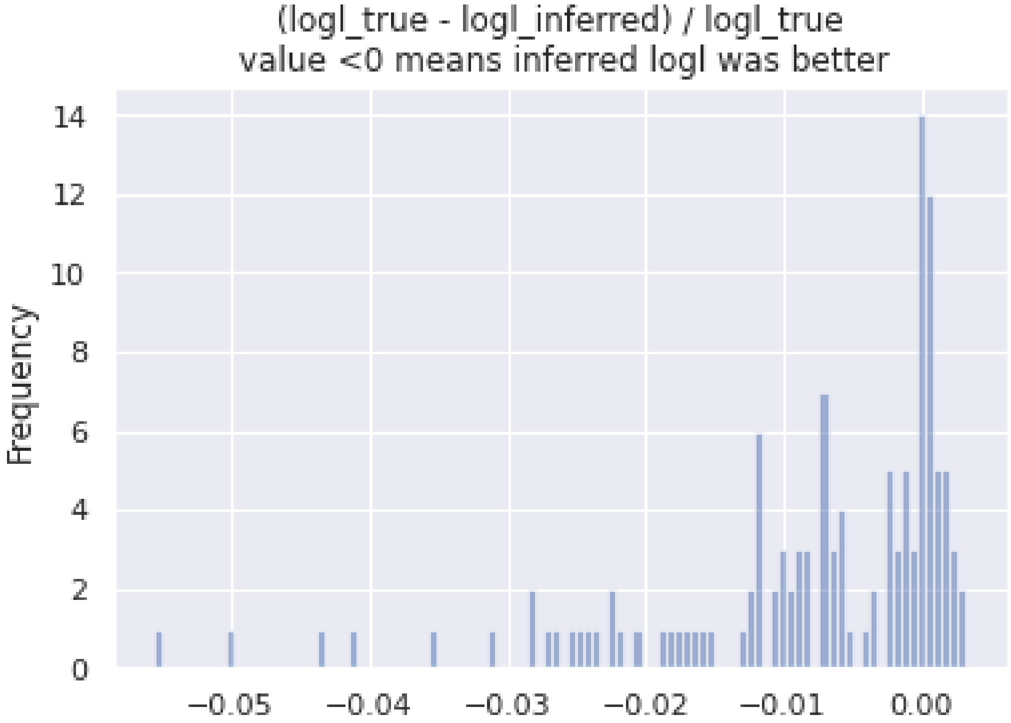


1.2.1 Plots for LikelihodType.AVERAGE

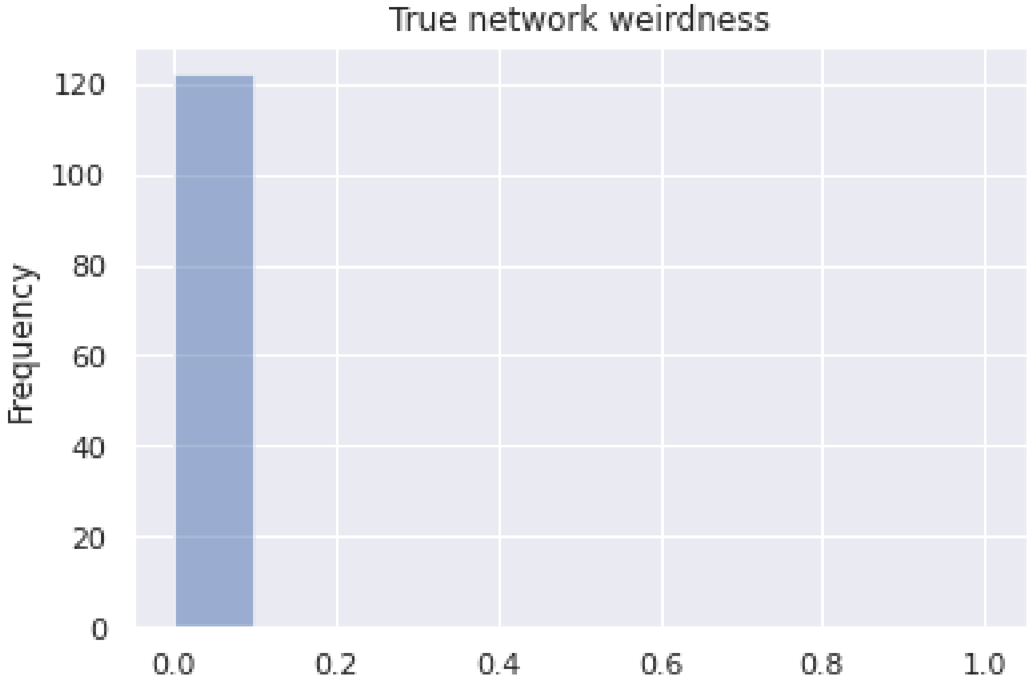
```
[12]: df_raxml_only_msasize_200_average = df_raxml_only_msasize_200.
       ↪query('likelihood_type == "AVERAGE"')
      build_stats(df_raxml_only_msasize_200_average)
```

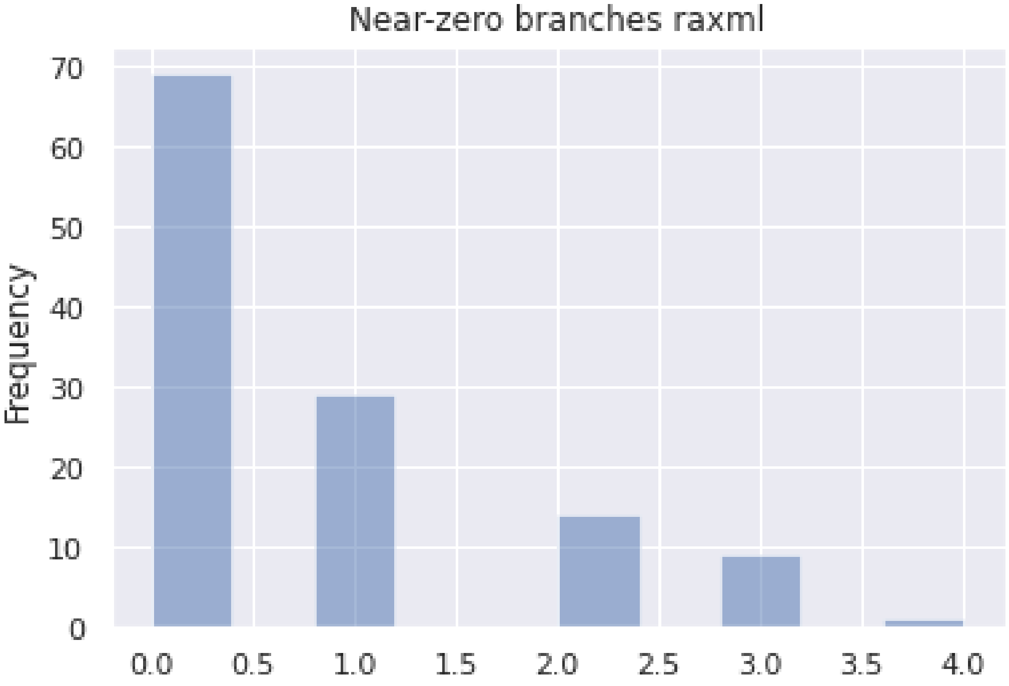
```
Inferred BIC better or equal: 85
Inferred BIC worse: 37

Inferred loglh better or equal: 37
Inferred loglh worse: 85

Inferred n_reticulations less: 93
Inferred n_reticulations equal: 29
Inferred n_reticulations more: 0
```


(bic_true - bic_inferred) / bic_true
value >0 means inferred BIC was better
Frequency
0
2
4
6
8
10
12
−0.03
−0.02
−0.01
0.00
0.01
0.02

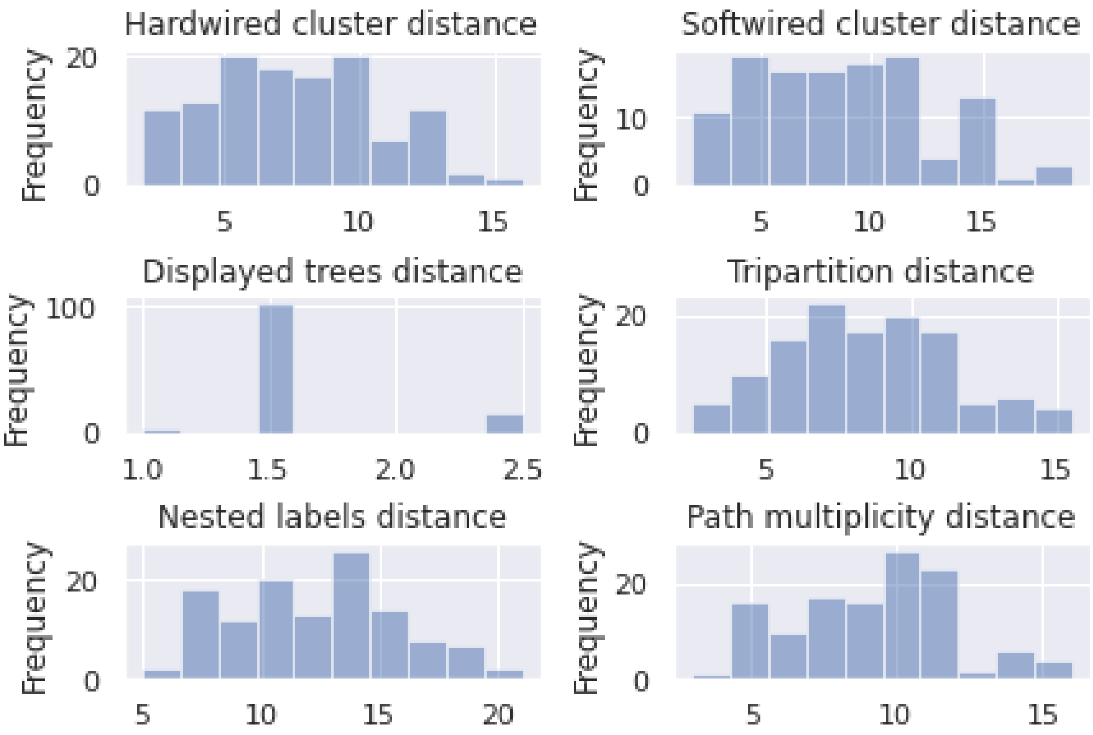

(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0
2
4
6
8
10
12
14
−0.05
−0.04
−0.03
−0.02
−0.01
0.00

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


True network weirdness
Frequency
0
20
40
60
80
100
120
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
0
10
20
30
40
50
60
70
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

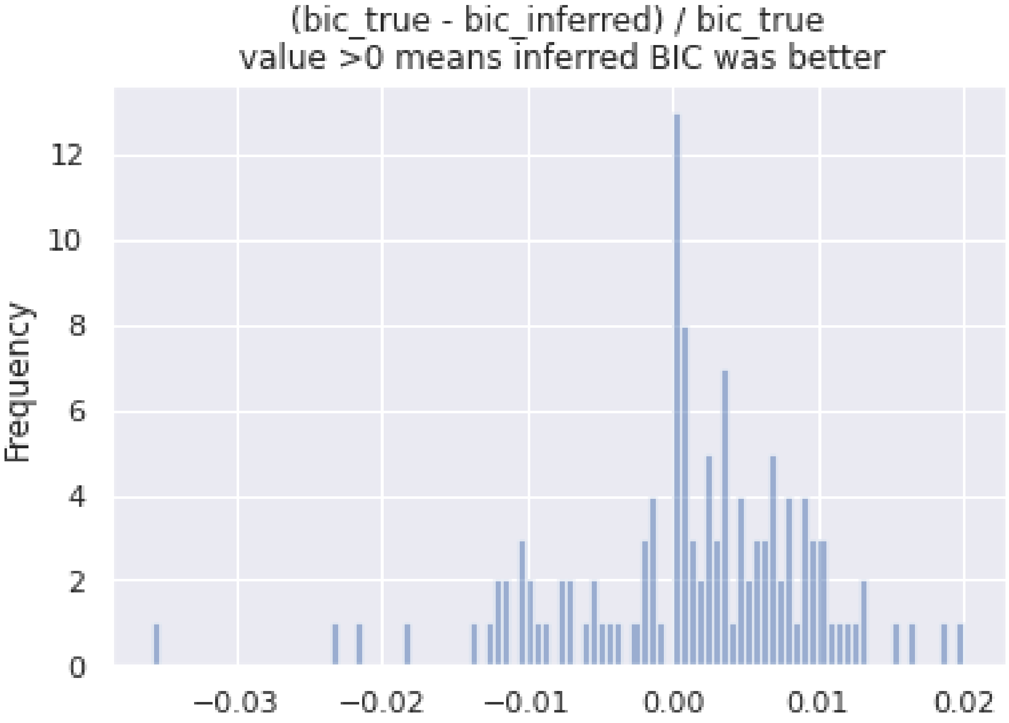
```
<Figure size 432x288 with 0 Axes>
```

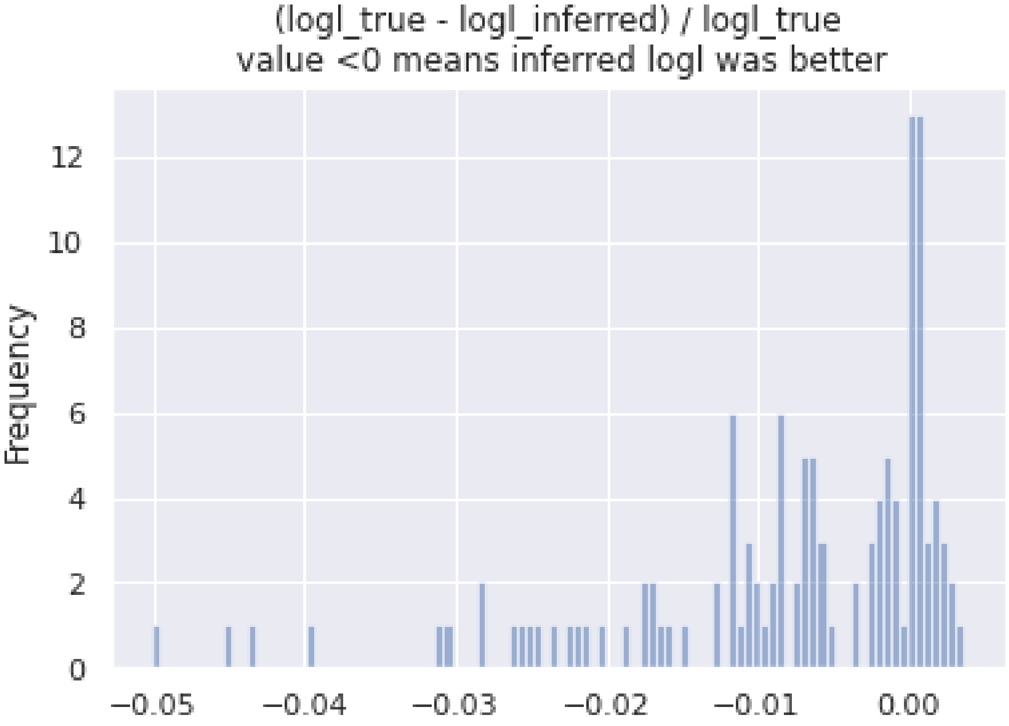


1.2.2 Plots for LikelihoodType.BEST

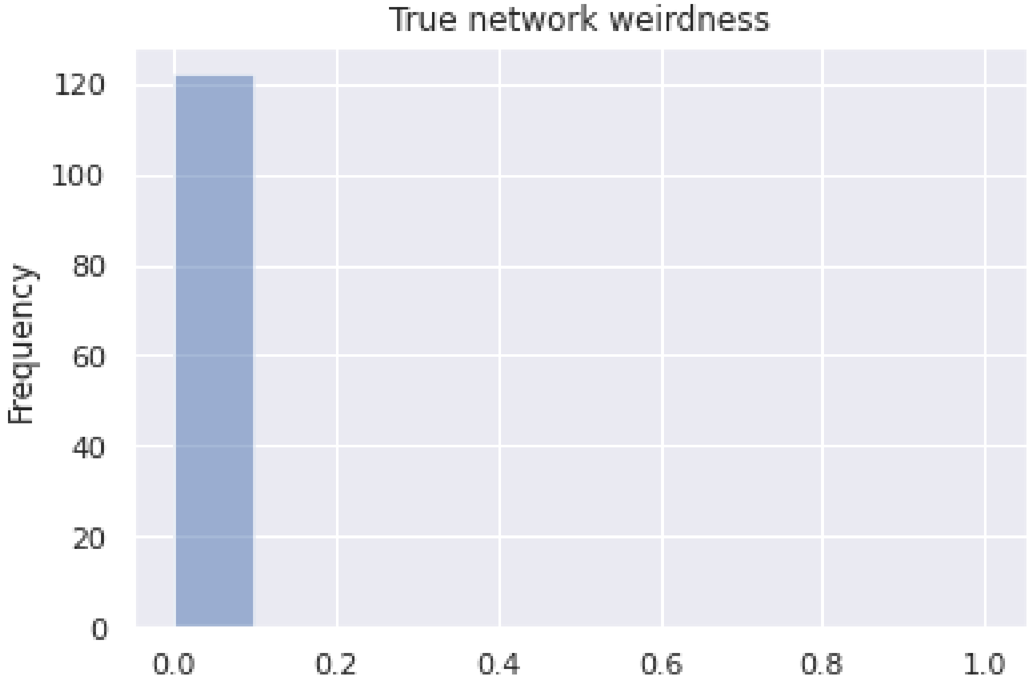
[13]:
```
df_raxml_only_msasize_200_best = df_raxml_only_msasize_200.
 ↪query('likelihood_type == "BEST"')
build_stats(df_raxml_only_msasize_200_best)
```

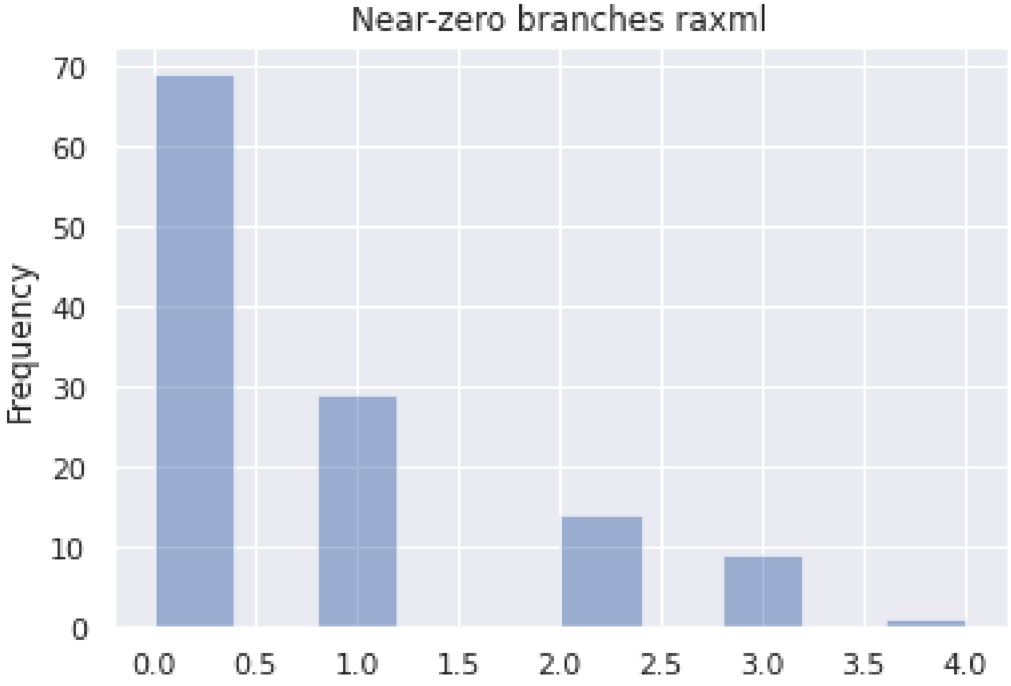
```
Inferred BIC better or equal: 85
Inferred BIC worse: 37

Inferred loglh better or equal: 38
Inferred loglh worse: 84

Inferred n_reticulations less: 94
Inferred n_reticulations equal: 28
Inferred n_reticulations more: 0
```


(bic_true - bic_inferred) / bic_true
value >0 means inferred BIC was better
Frequency
0
2
4
6
8
10
12
−0.03
−0.02
−0.01
0.00
0.01
0.02

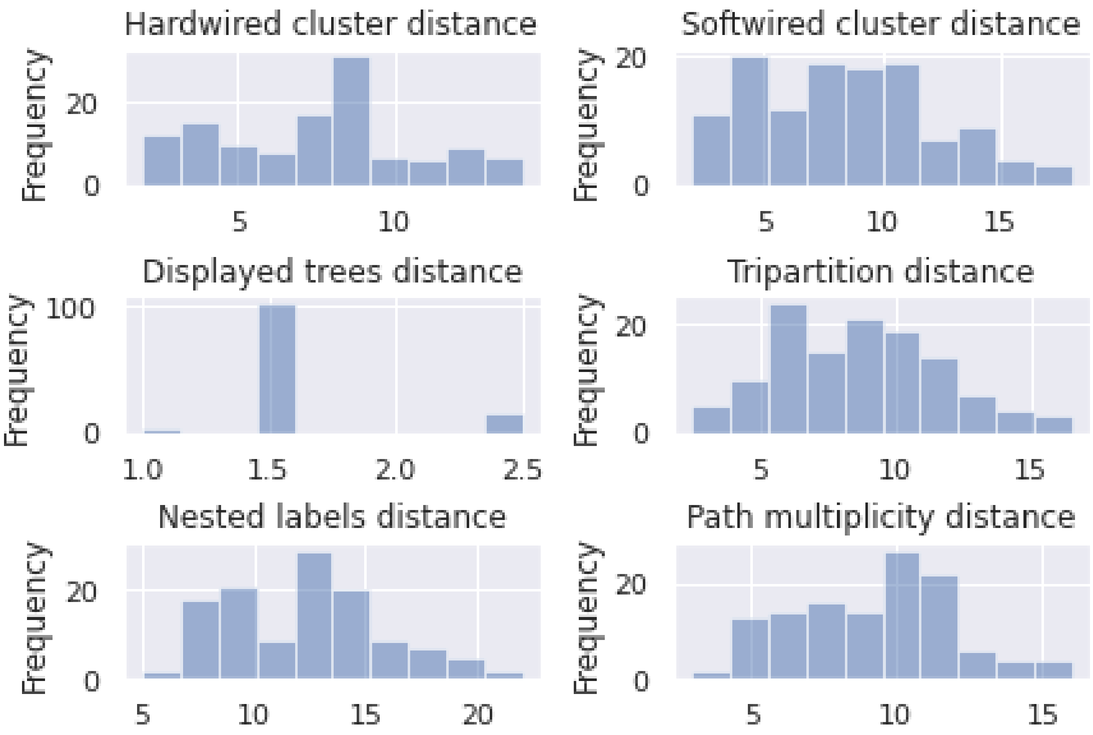

(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0
2
4
6
8
10
12
−0.05
−0.04
−0.03
−0.02
−0.01
0.00

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


True network weirdness
Frequency
0
20
40
60
80
100
120
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
0
10
20
30
40
50
60
70
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

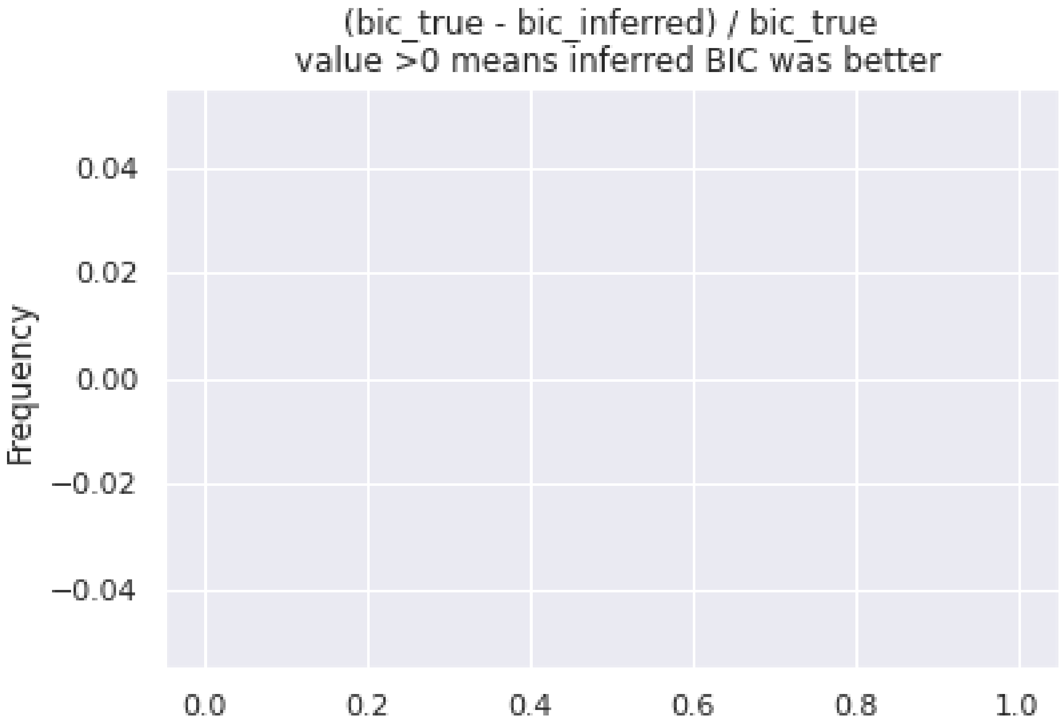
```
<Figure size 432x288 with 0 Axes>
```

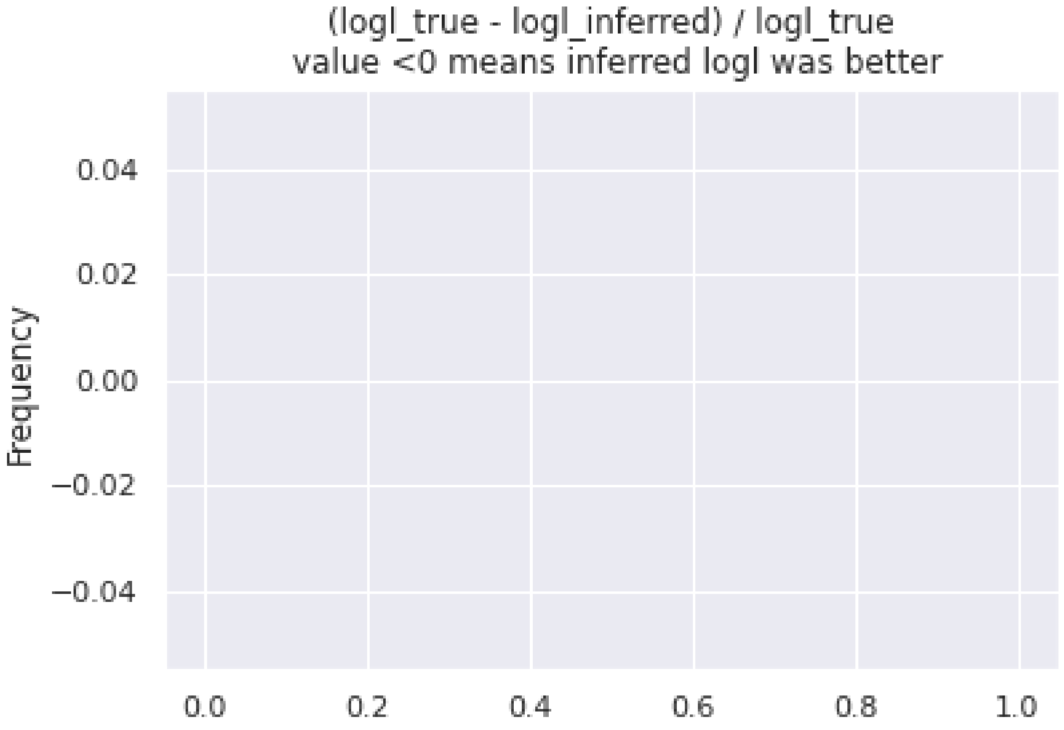


2 Plots for starting with 5 random, 5 parsimony trees

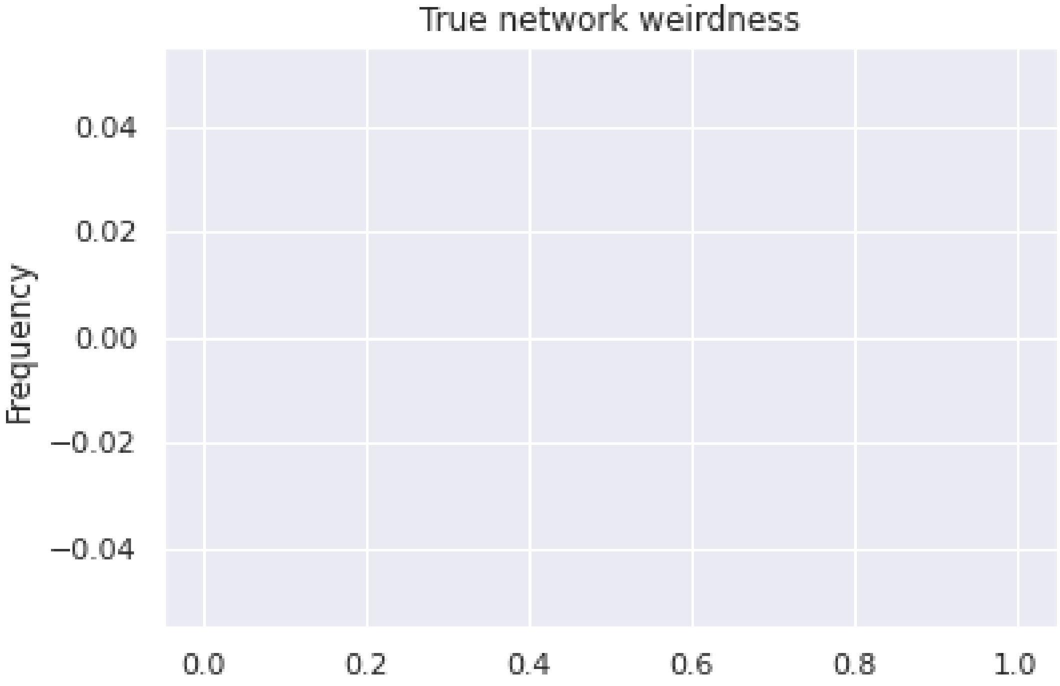
```
[14]: df_random = df.query('start_type == "RANDOM"')
      build_stats(df_random)
```

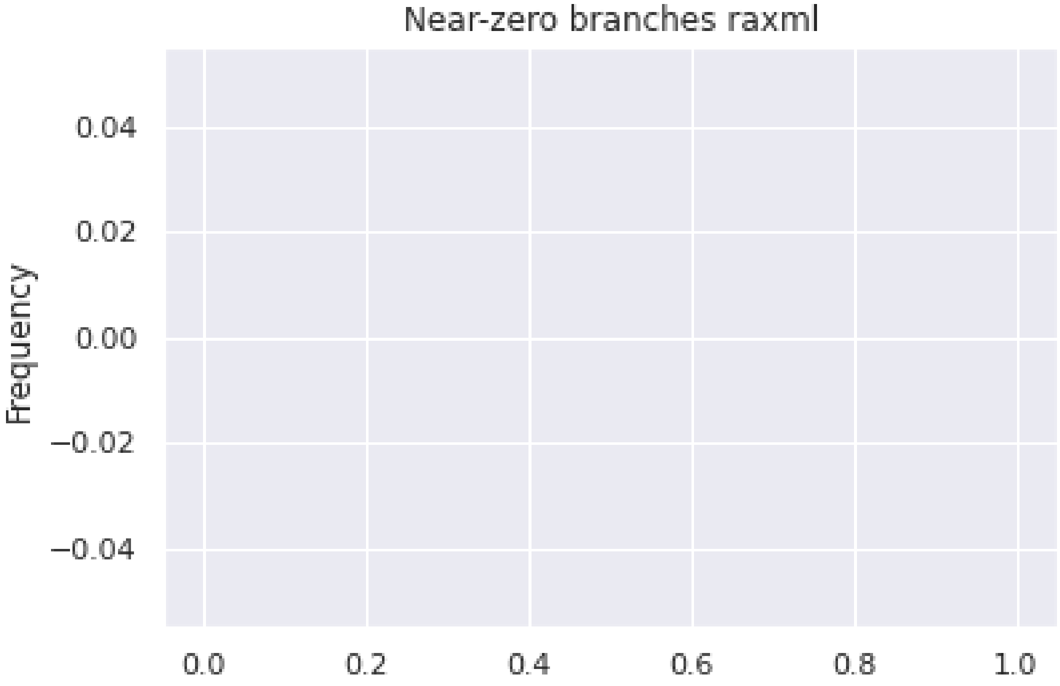
```
Inferred BIC better or equal: 0
Inferred BIC worse: 0

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
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0


(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

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


True network weirdness
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

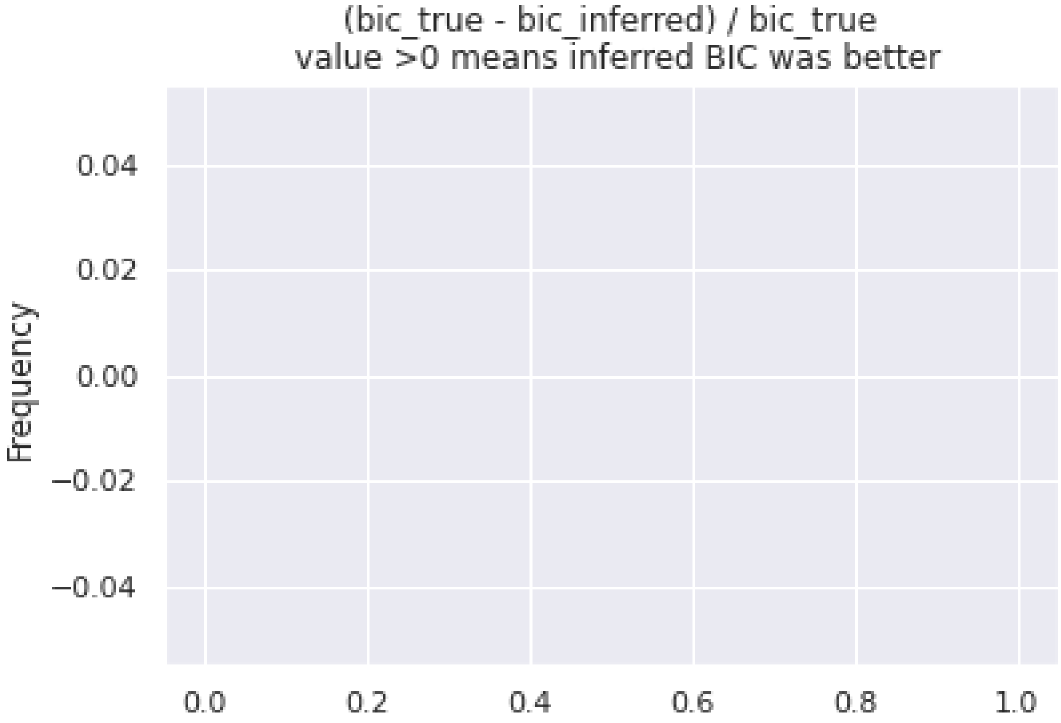

Near-zero branches raxml
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

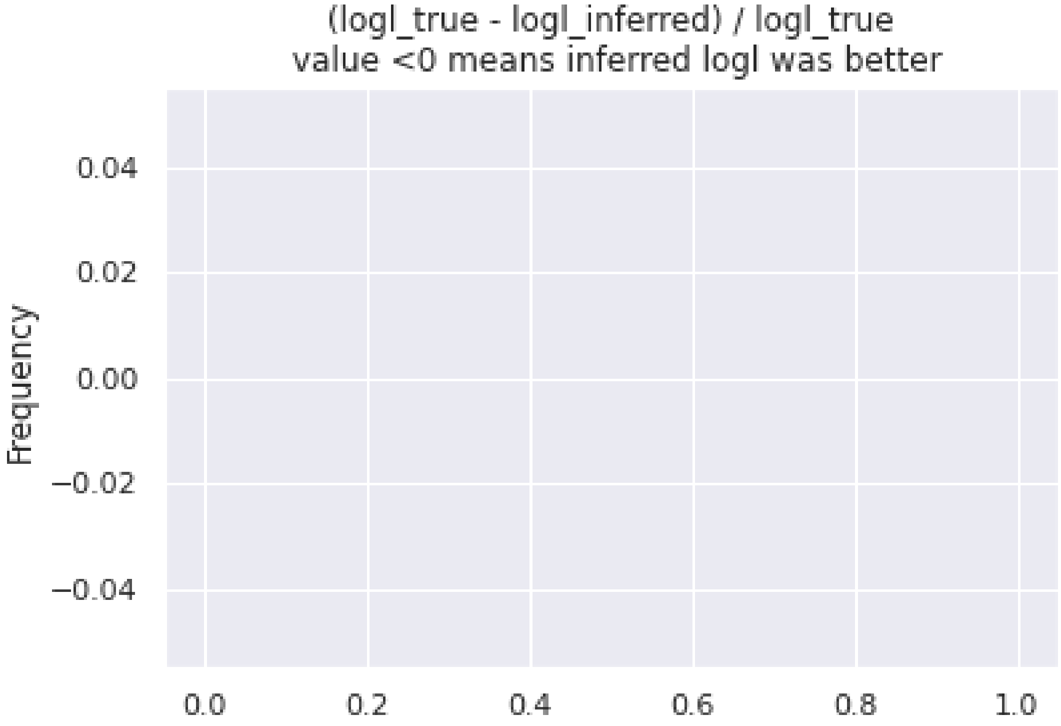
```
<Figure size 432x288 with 0 Axes>
```

2.1 Plots for MSA_size ~ 100*n_trees

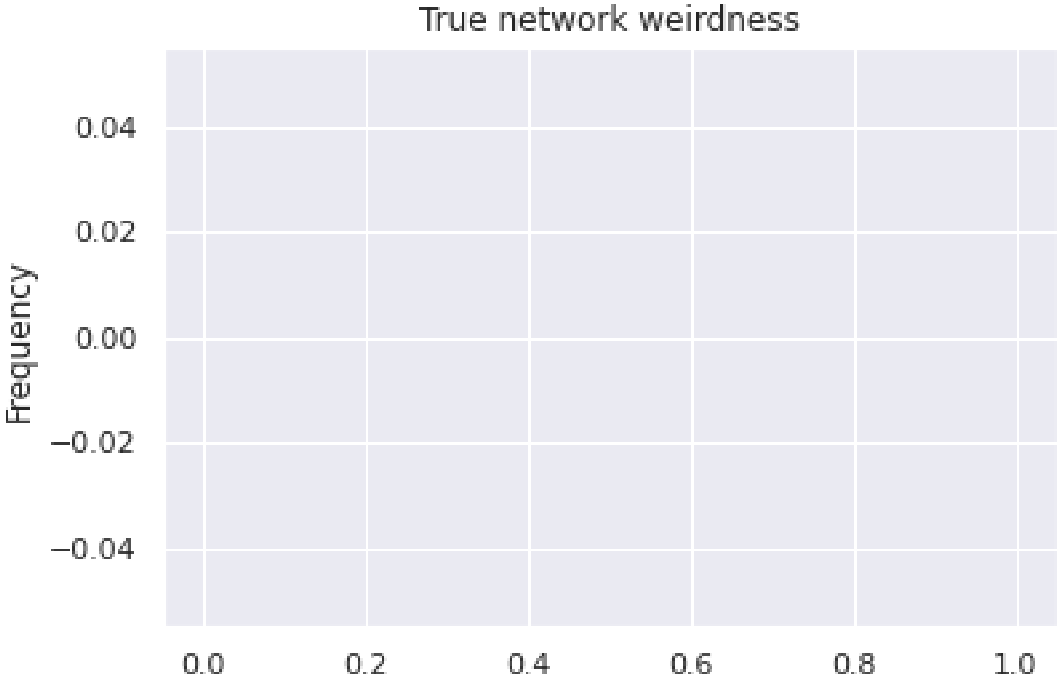
[15]:
```
df_random_msasize_100 = df_random.query('msa_size == 100')
build_stats(df_random_msasize_100)
```

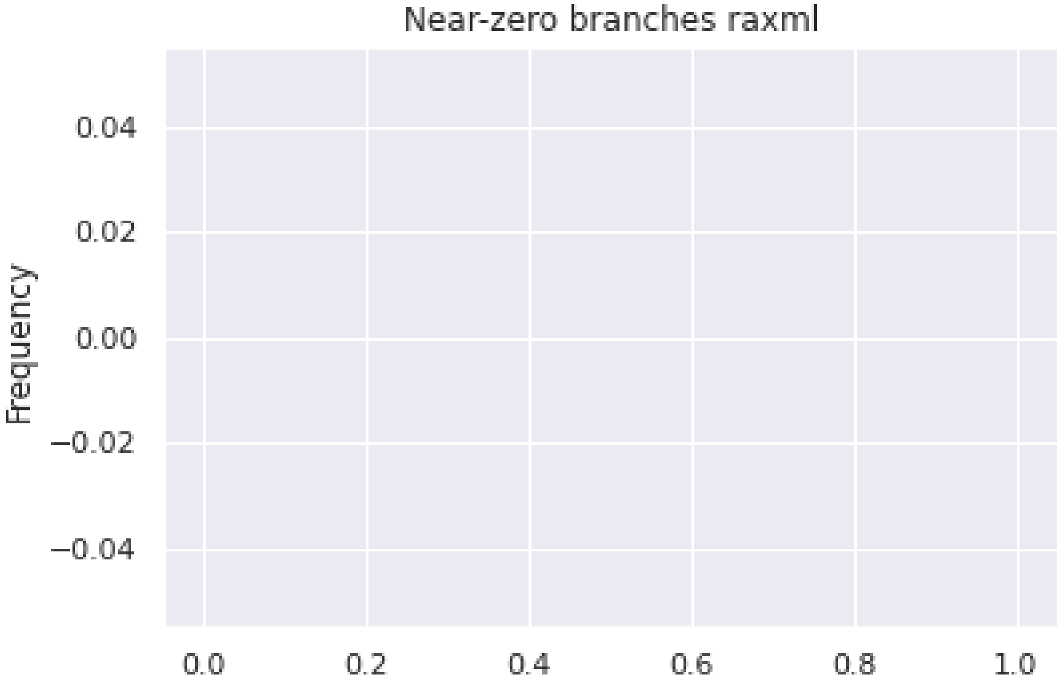
```
Inferred BIC better or equal: 0
Inferred BIC worse: 0

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
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0


(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

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


True network weirdness
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

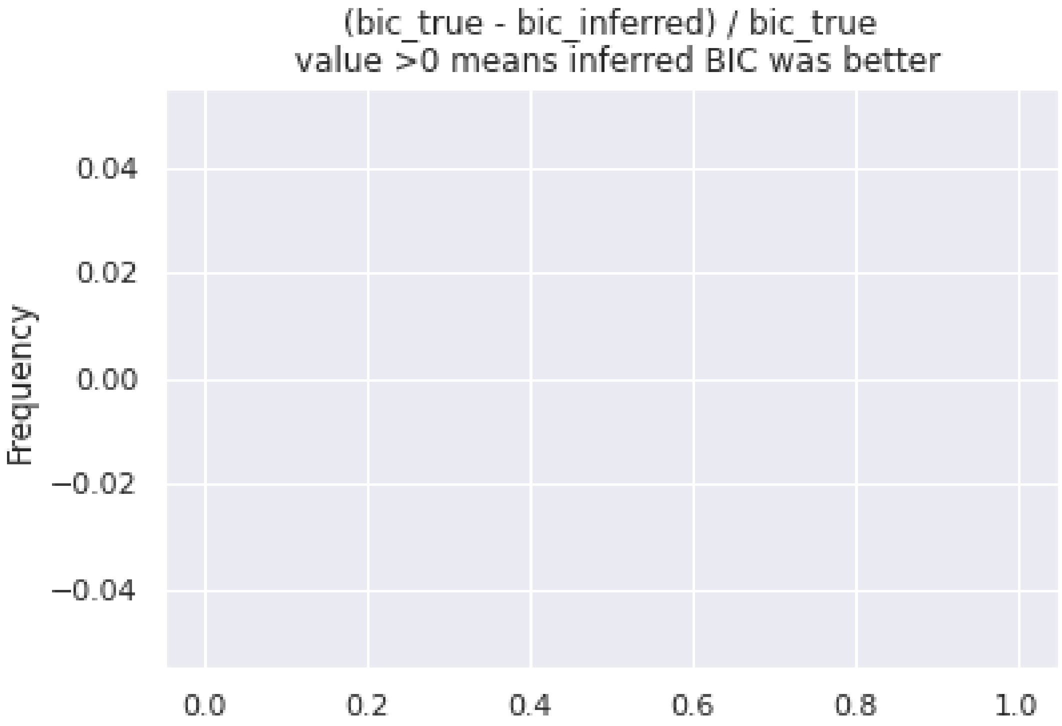

Near-zero branches raxml
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

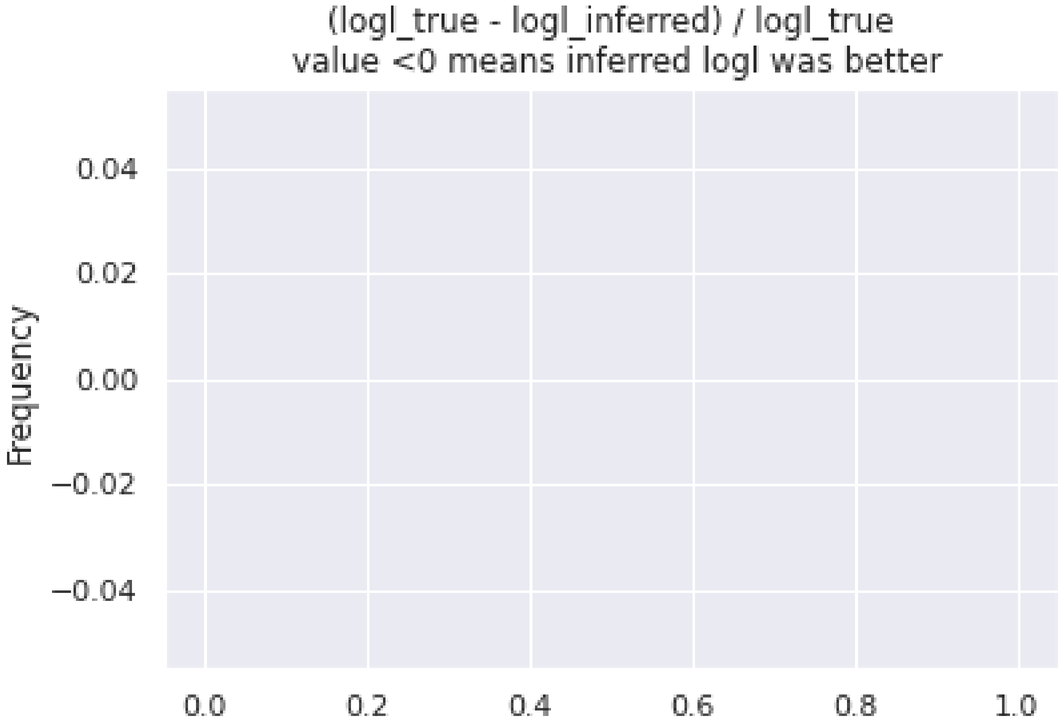
```
<Figure size 432x288 with 0 Axes>
```

2.1.1 Plots for LikelihodType.AVERAGE

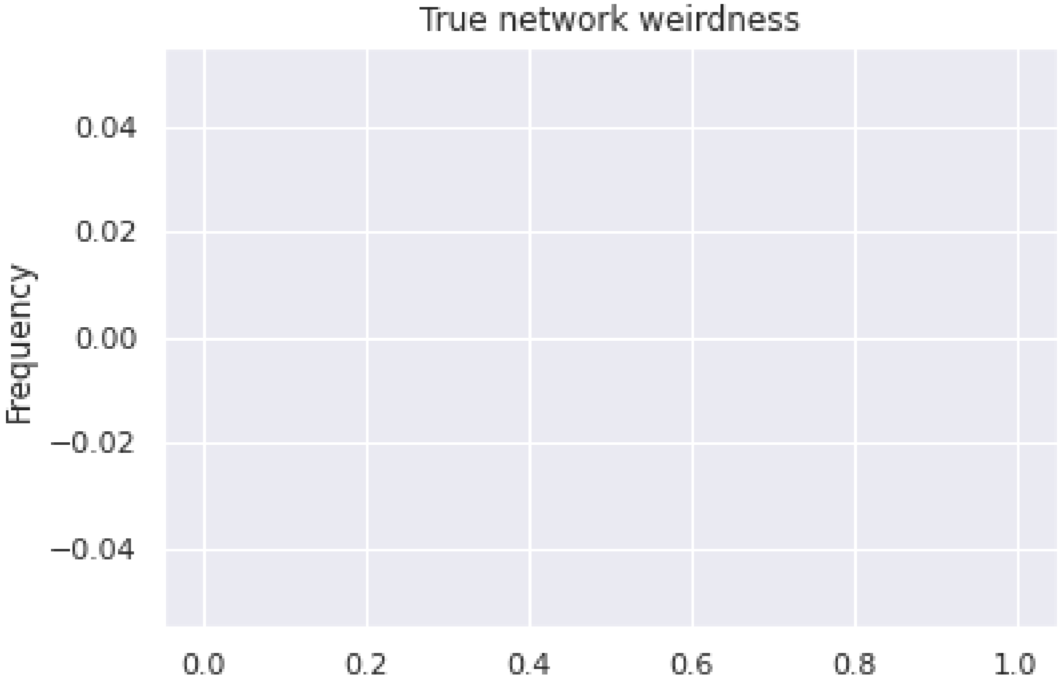
[16]:
```
df_random_msasize_100_average = df_random_msasize_100.query('likelihood_type ==␣
↪"AVERAGE"')
build_stats(df_random_msasize_100_average)
```

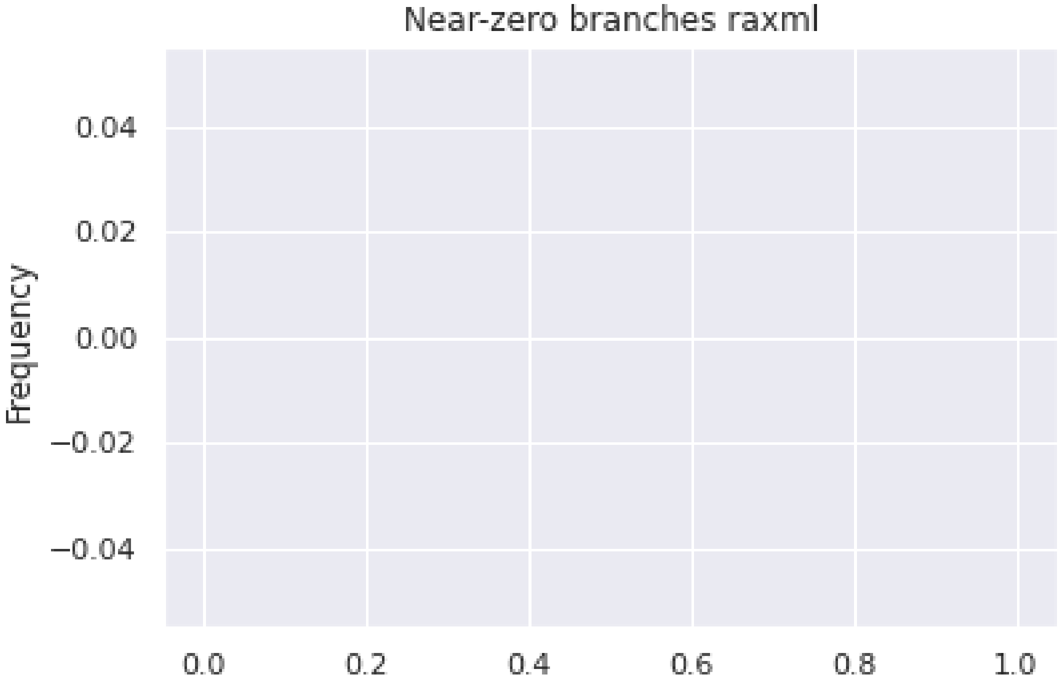
```
Inferred BIC better or equal: 0
Inferred BIC worse: 0

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>
```


True network weirdness
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

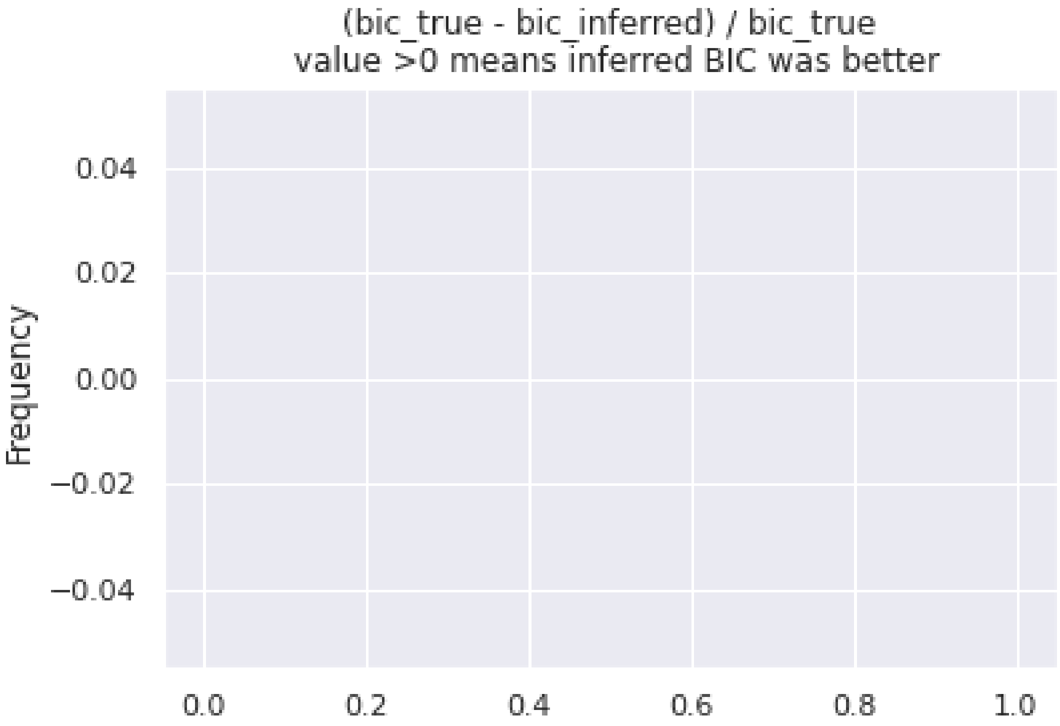

Near-zero branches raxml
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

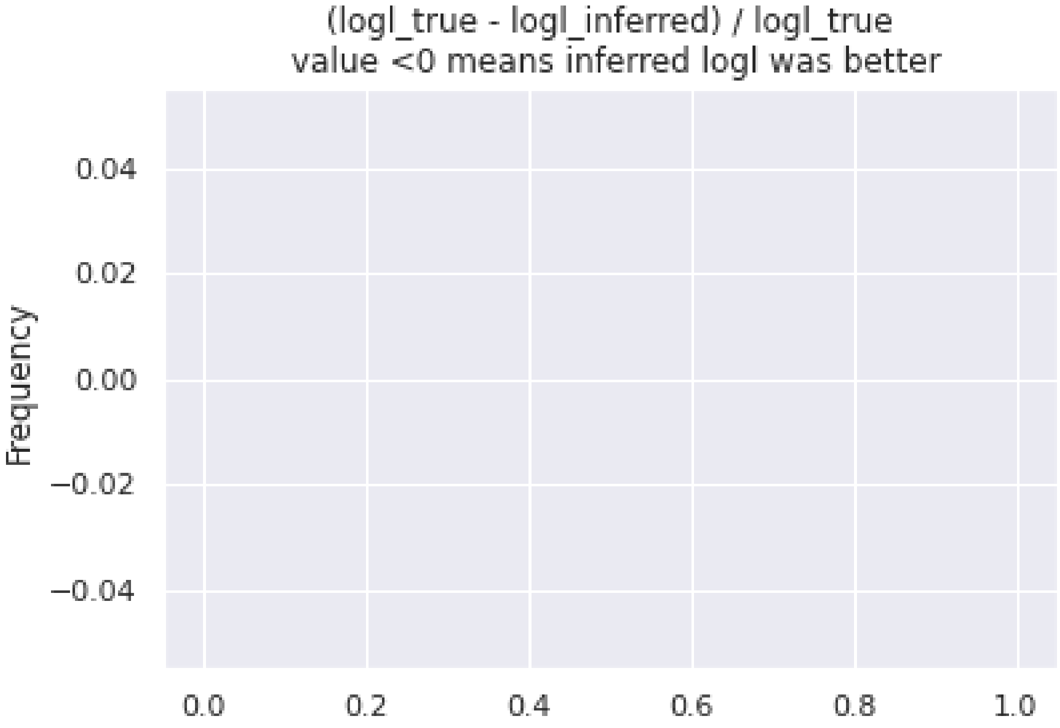
```
<Figure size 432x288 with 0 Axes>
```

2.1.2 Plots for LikelihoodType.BEST

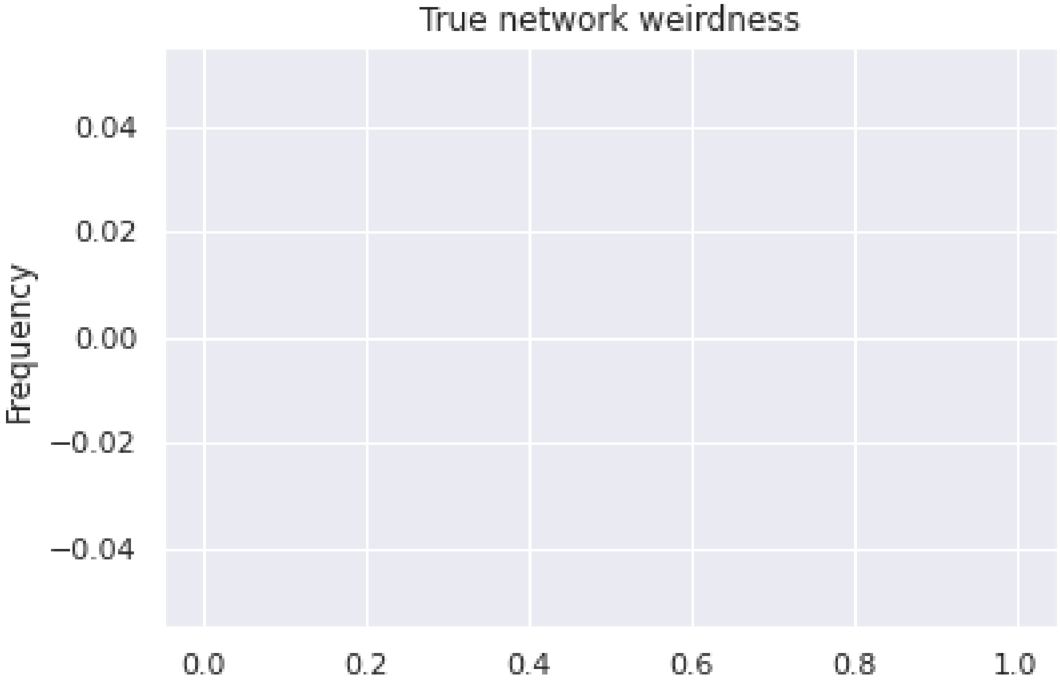
```
[17]: df_random_msasize_100_best = df_random_msasize_100.query('likelihood_type ==␣
      ↪"BEST"')
      build_stats(df_random_msasize_100_best)
```

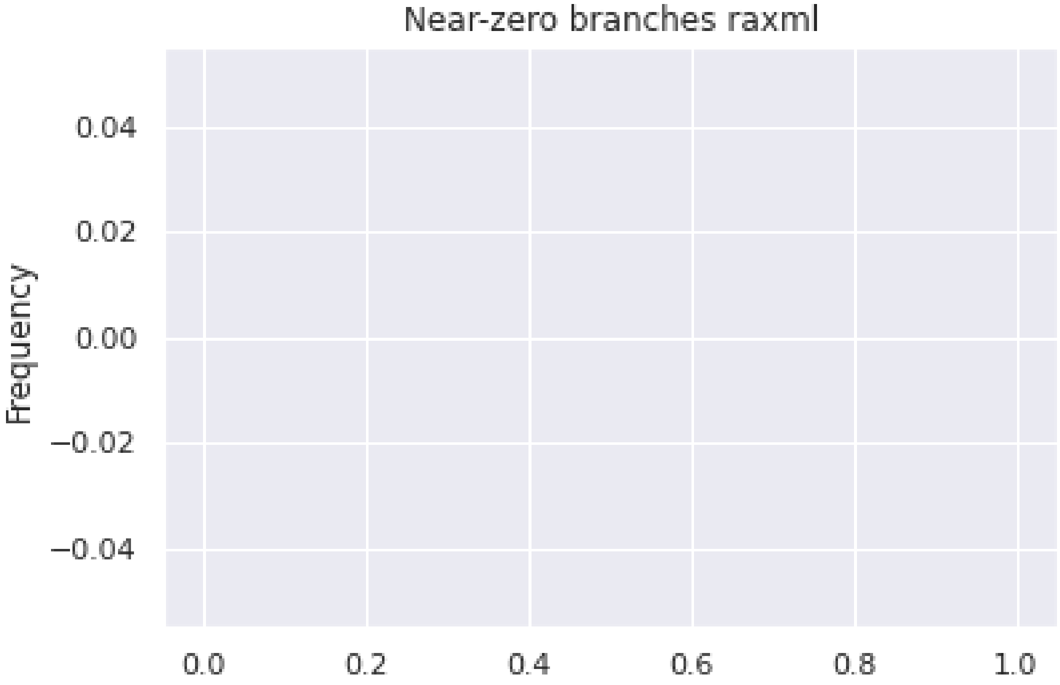
```
Inferred BIC better or equal: 0
Inferred BIC worse: 0

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
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

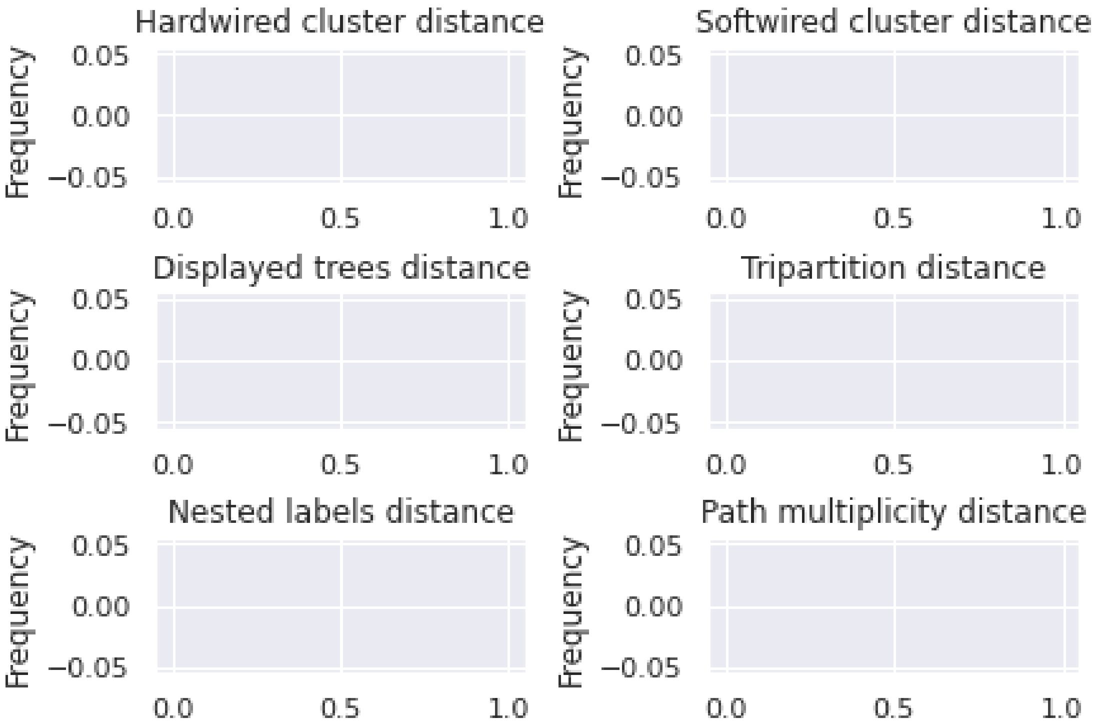

(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

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


True network weirdness
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

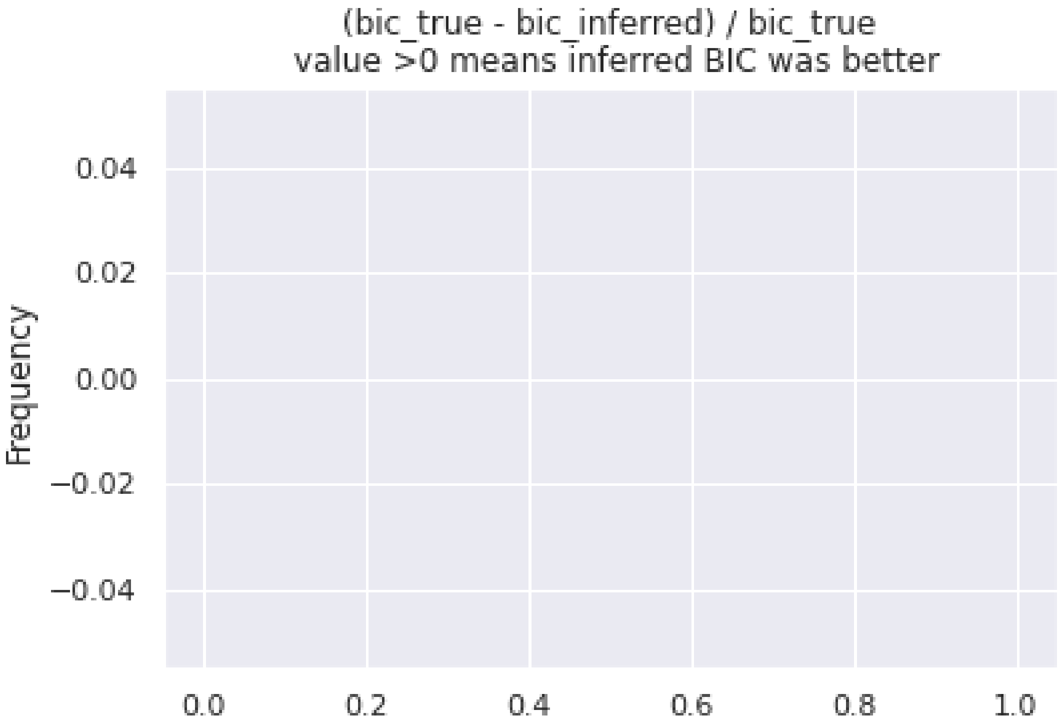
```
<Figure size 432x288 with 0 Axes>
```

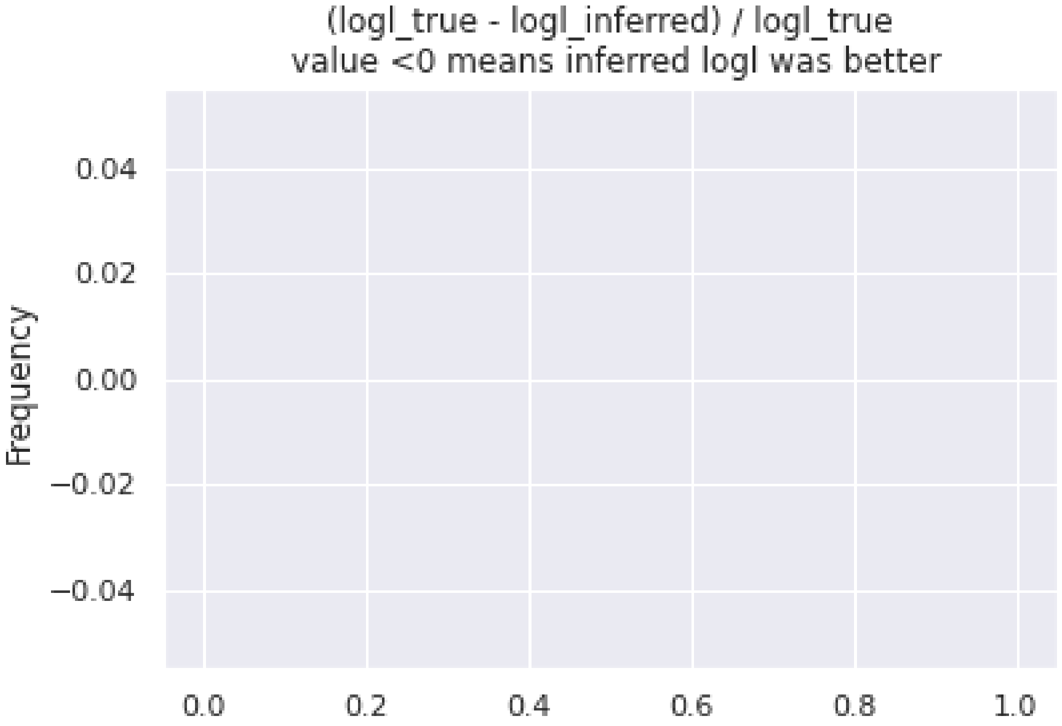


2.2 Plots for MSA_size ~ 200*n_trees

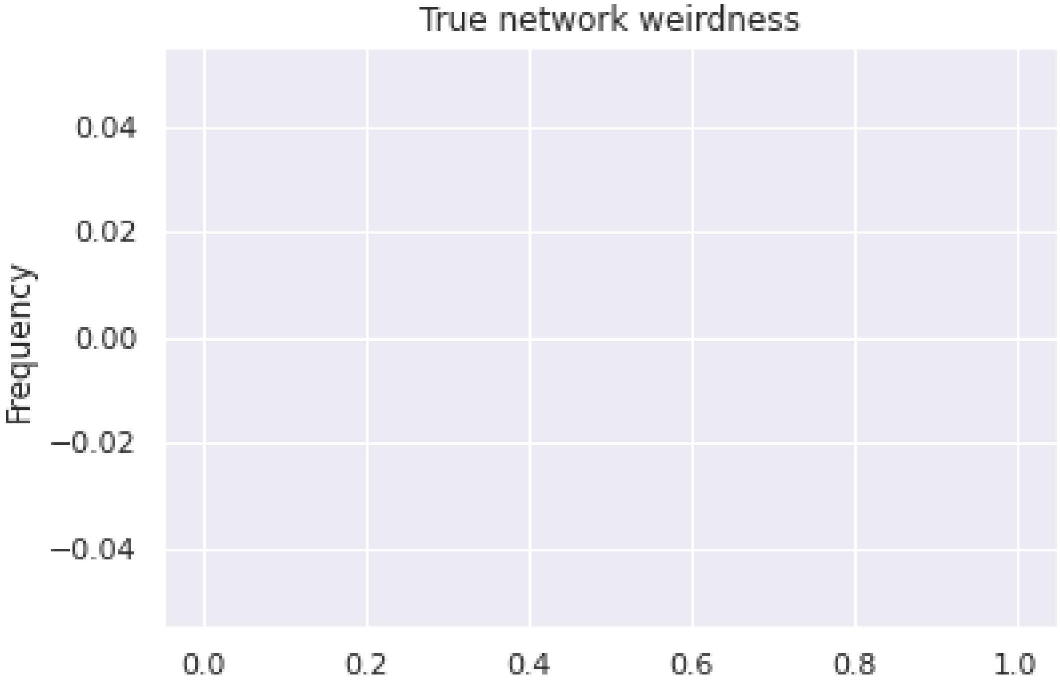
```
[18]: df_random_msasize_200 = df_random.query('msa_size == 200')
      build_stats(df_random_msasize_200)
```

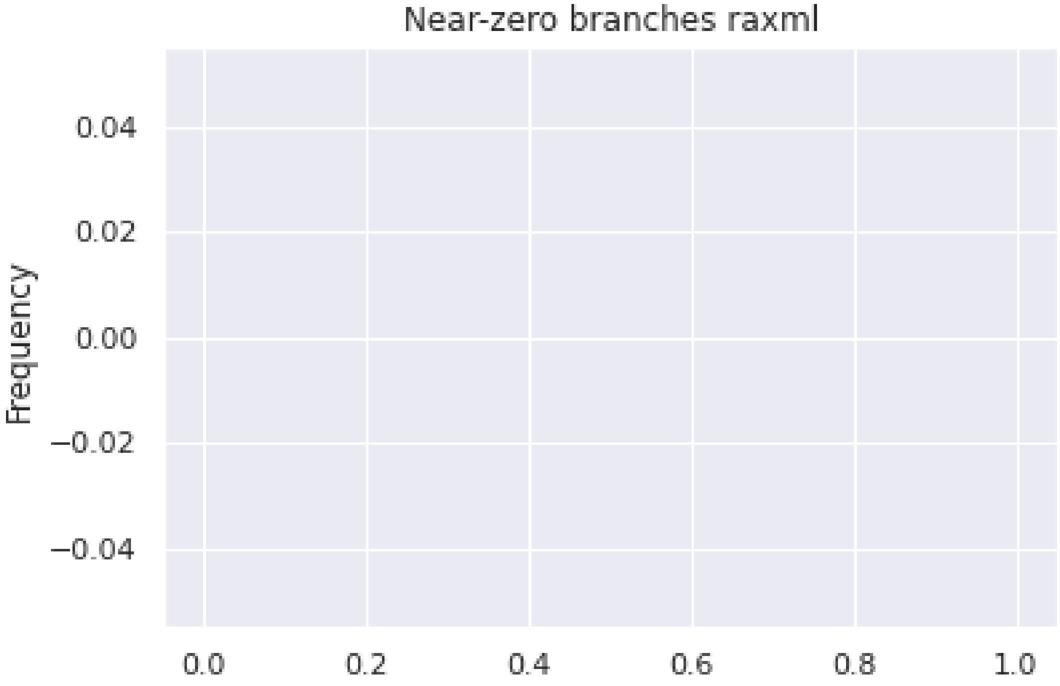
```
Inferred BIC better or equal: 0
Inferred BIC worse: 0

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
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0


(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

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


True network weirdness
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

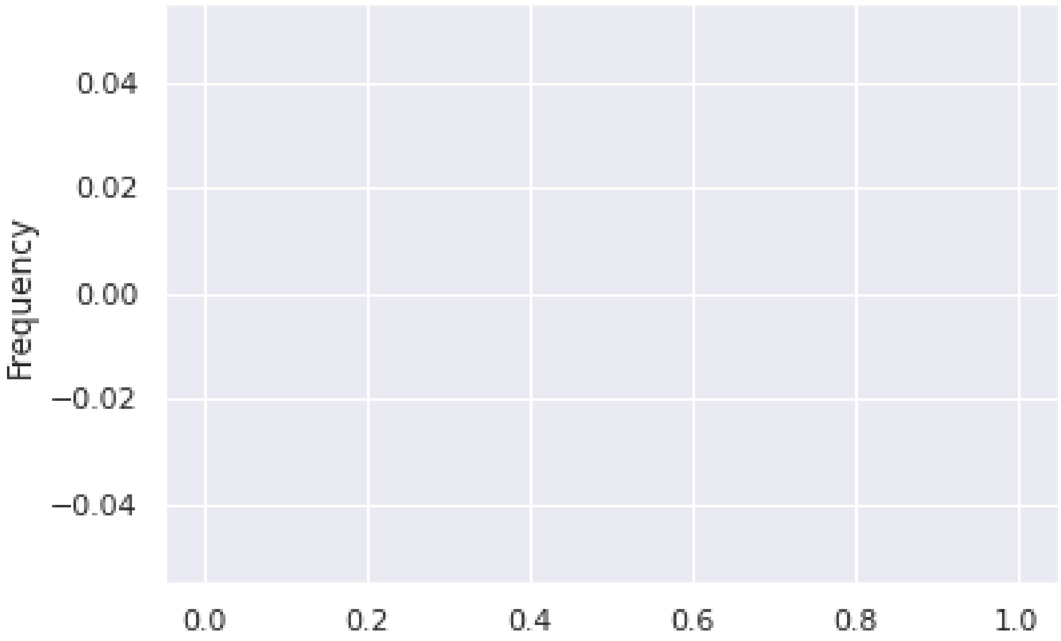
```
<Figure size 432x288 with 0 Axes>
```

2.2.1 Plots for LikelihodType.AVERAGE

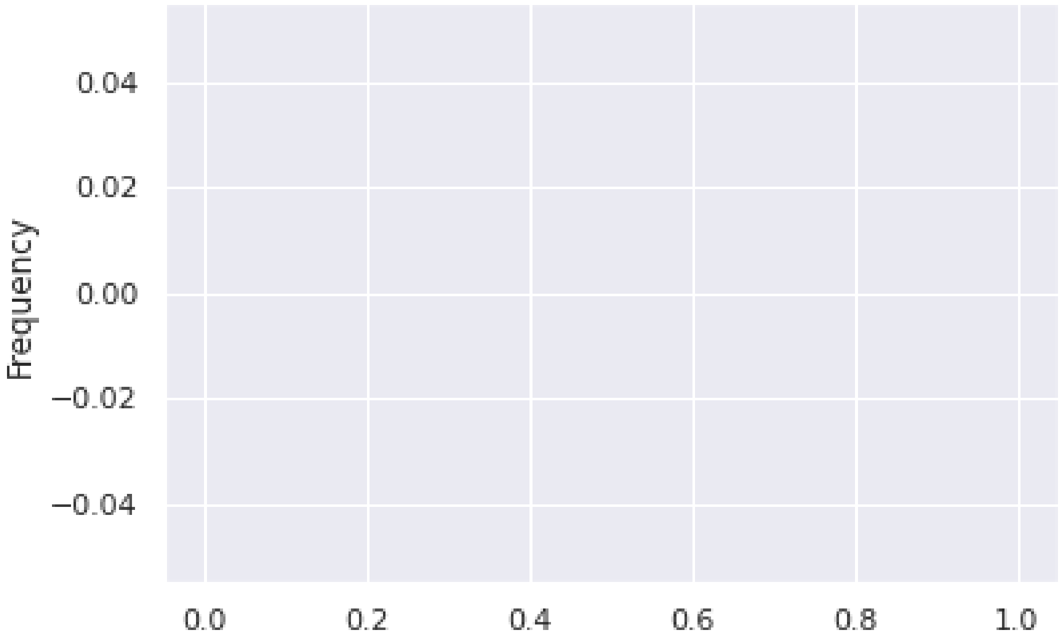
[19]:
```
df_random_msasize_200_average = df_random_msasize_200.query('likelihood_type ==␣
 ↪"AVERAGE"')
build_stats(df_random_msasize_200_average)
```

```
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
```

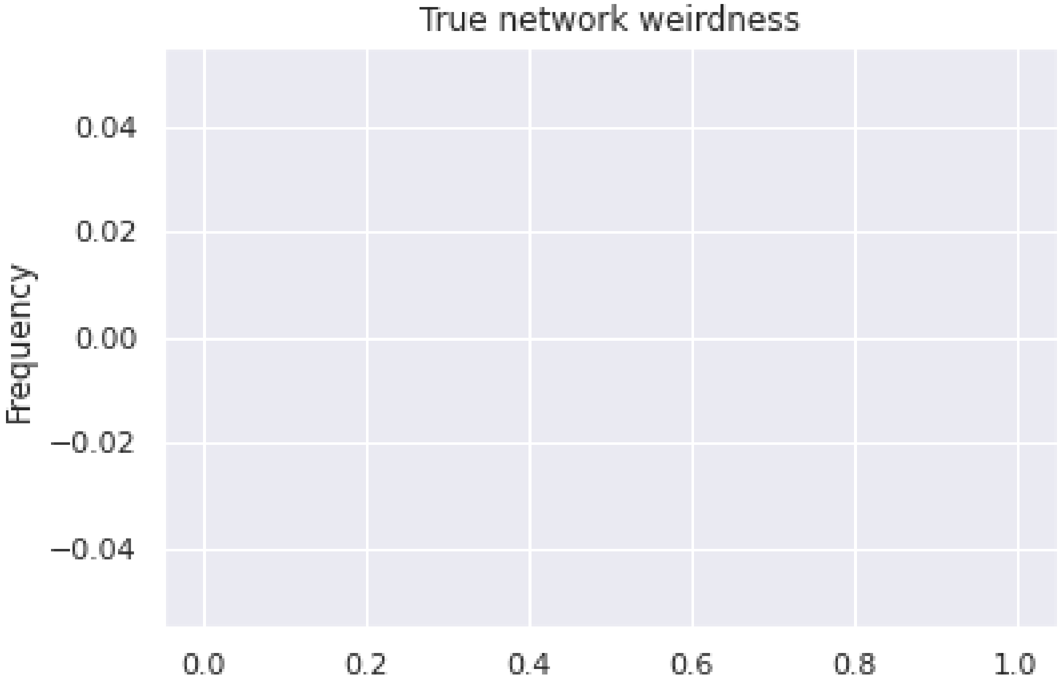
(bic_true - bic_inferred) / bic_true
value >0 means inferred BIC was better

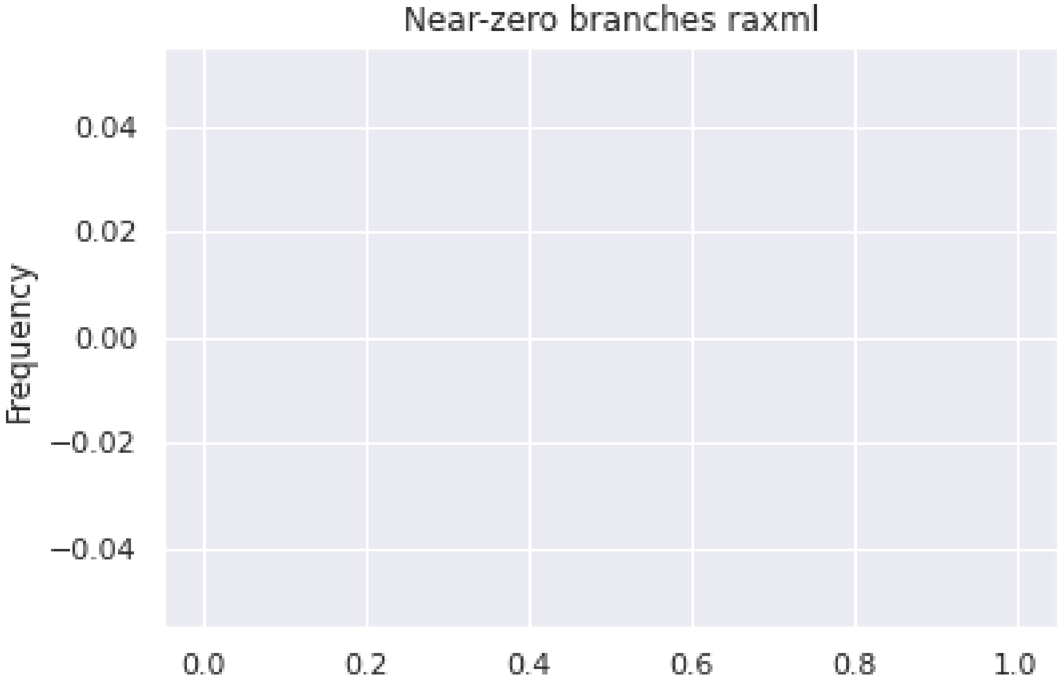


(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better

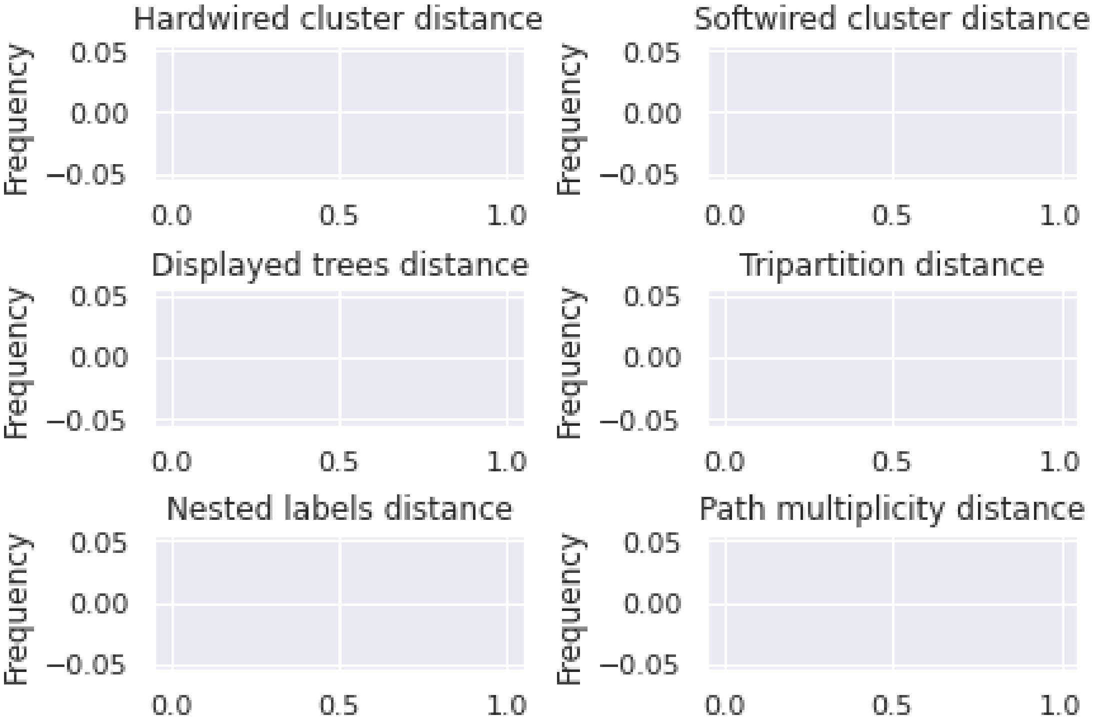


```
<Figure size 432x288 with 0 Axes>
```


True network weirdness
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

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



Hardwired cluster distance

Softwired cluster distance

Displayed trees distance

Tripartition distance

Nested labels distance

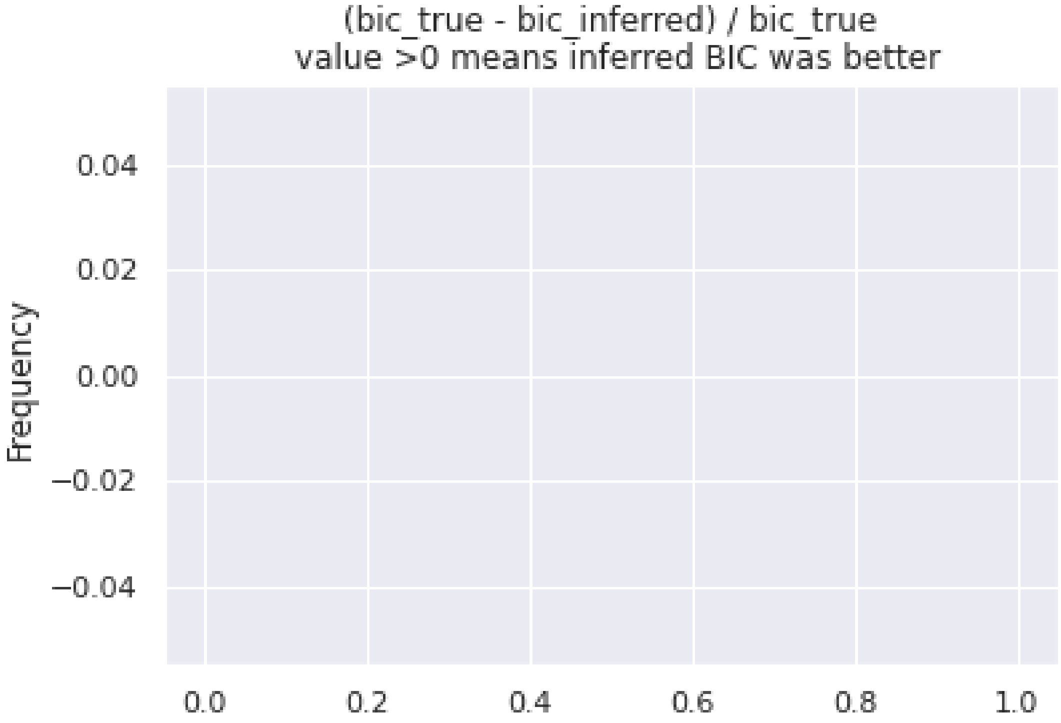
Path multiplicity distance

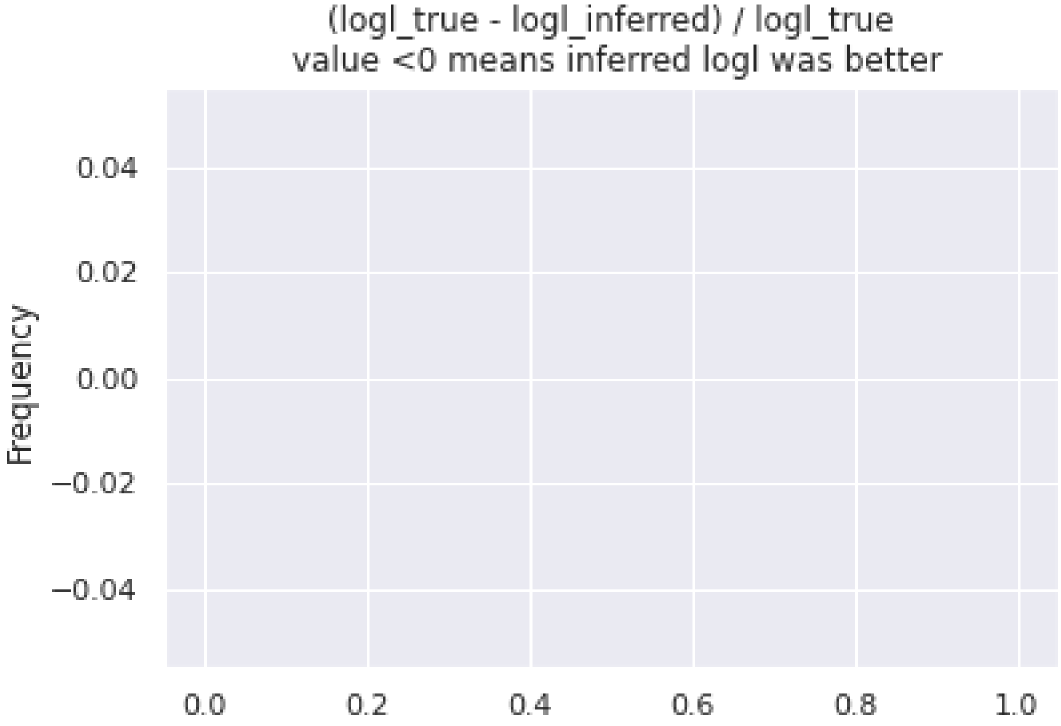
2.2.2 Plots for LikelihoodType.BEST

[20]:

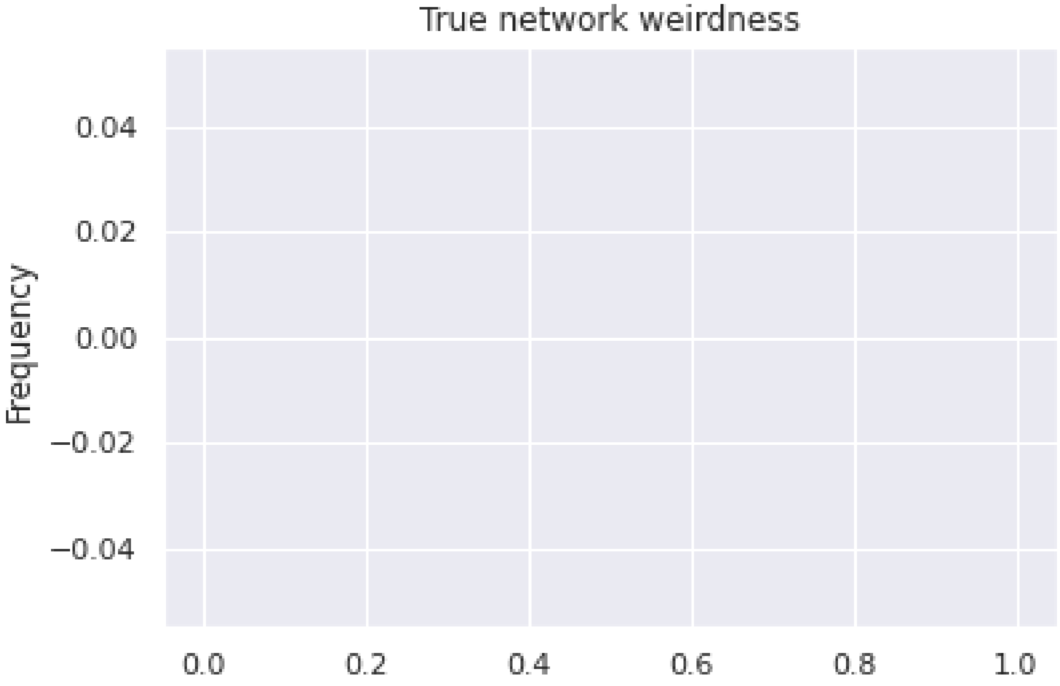
```
df_random_msasize_200_best = df_random_msasize_200.query('likelihood_type ==␣
 ↪"BEST"')
build_stats(df_random_msasize_200_best)
```

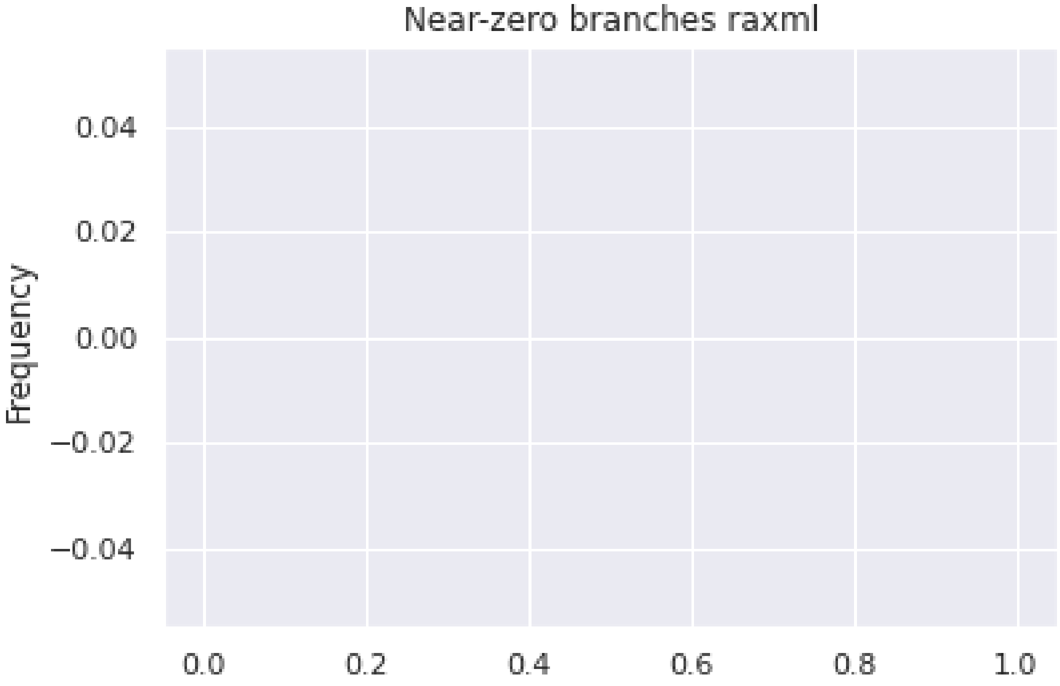
```
Inferred BIC better or equal: 0
Inferred BIC worse: 0

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

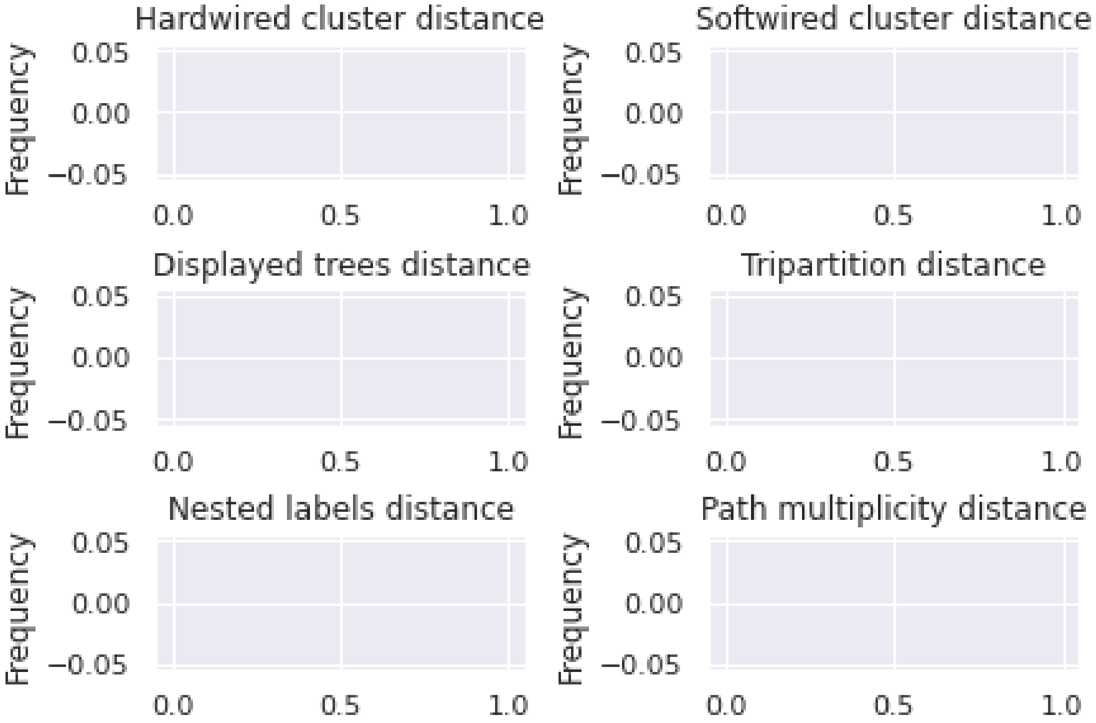

(logl_true - logl_inferred) / logl_true
value <0 means inferred logl was better

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


True network weirdness
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0


Near-zero branches raxml
Frequency
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

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


Hardwired cluster distance
Softwired cluster distance
Displayed trees distance
Tripartition distance
Nested labels distance
Path multiplicity distance
Frequency
0.05
0.00
−0.05
0.0
0.5
1.0

```
[ ]:
```