In [1]:
```
import os
os.chdir('/Users/gc3045/scmail_v1/LAML/')
```

In [2]:
```
from laml_libs.IO_handler.sequence_lib import *
from laml_libs.IO_handler.DLT_parser import *
```

Test character matrix (.csv) as input

In [3]:
```
charMtrxFile = "examples/example1/character_matrix.csv"
delimiter = ","
missing_char="?"
outputFile = "laml_io_tests/example1_test.json"
```

In [4]:
```
charMtrx, site_names = read_sequences(charMtrxFile, filetype="charMtr
# returns a dictionary of cell name to list of states
```

In [5]:
```
tmp = DLT_parser(charMtrxFile)
```

In [6]:
```
# testing conversion of character matrix to json
tmp.charMtrx_to_json(delimiter=delimiter, outputfile=outputFile, miss
```

Out[6]:
```
True
```

In [7]:
```
tmp.datafile
```

Out[7]:
```
'laml_io_tests/example1_test.json'
```

In [8]:
```
data_struct = tmp.parse_json()
```

In [9]:
```
tmp.K, tmp.J, len(tmp.data)
```

Out[9]:
```
(30, 1, 250)
```

In [10]:
```
data = tmp.data
```

In [11]:
```
len([x["cassette_idx"] for x in data[0]["cassettes"]])
```

Out[11]:
```
30
```

In [12]:
```
len(data[0]['cassettes'][0]['cassette_state'])
```

Out[12]:
```
1
```

In [13]:
```
data[0]
```

Out[13]:
```
{'cell_name': '1424',
 'cassettes': [{'cassette_idx': 0, 'cassette_state': [0]},
  {'cassette_idx': 1, 'cassette_state': [3]},
  {'cassette_idx': 2, 'cassette_state': [1]},
  {'cassette_idx': 3, 'cassette_state': [1]},
  {'cassette_idx': 4, 'cassette_state': [0]},
  {'cassette_idx': 5, 'cassette_state': [15]},
  {'cassette_idx': 6, 'cassette_state': [0]},
  {'cassette_idx': 7, 'cassette_state': [10]},
  {'cassette_idx': 8, 'cassette_state': [3]},
  {'cassette_idx': 9, 'cassette_state': []},
  {'cassette_idx': 10, 'cassette_state': []},
  {'cassette_idx': 11, 'cassette_state': [3]},
  {'cassette_idx': 12, 'cassette_state': []},
  {'cassette_idx': 13, 'cassette_state': []},
  {'cassette_idx': 14, 'cassette_state': [2]},
  {'cassette_idx': 15, 'cassette_state': [12]},
  {'cassette_idx': 16, 'cassette_state': [0]},
  {'cassette_idx': 17, 'cassette_state': [0]},
```

In [14]:
```
a = tmp.set_alphabet()
a
```

Out[14]:
```
[[(), (0,), (1,), (3,)],
 [(), (0,), (1,), (2,), (3,), (4,)],
 [(), (0,), (1,)],
 [(), (0,), (1,), (4,), (6,)],
 [(),
  (0,),
  (1,),
  (3,),
  (11,),
  (13,),
  (17,),
  (18,),
  (36,),
  (37,),
  (40,),
  (41,),
  (44,),
  (65,)],
 [(), (0,), (1,), (2,), (3,), (5,), (6,), (7,), (8,), (10,), (1
```

Test character matrix (.json) as input

In [15]:
```
charMtrxFile =  "laml_io_tests/example1_test.json"
tmp = DLT_parser(charMtrxFile)
tmp.parse_json()
```

Out[15]:
```
{'1424': {0: (0,),
  1: (3,),
  2: (1,),
  3: (1,),
  4: (0,),
  5: (15,),
  6: (0,),
  7: (10,),
  8: (3,),
  9: (),
  10: (),
  11: (3,),
  12: (),
  13: (),
  14: (2,),
  15: (12,),
  16: (0,),
  17: (0,),
  18: (7,),
```

In [16]:
```
a = tmp.set_alphabet()
a
```

Out[16]:
```
[[(), (0,), (1,), (3,)],
 [(), (0,), (1,), (2,), (3,), (4,)],
 [(), (0,), (1,)],
 [(), (0,), (1,), (4,), (6,)],
 [(),
  (0,),
  (1,),
  (3,),
  (11,),
  (13,),
  (17,),
  (18,),
  (36,),
  (37,),
  (40,),
  (41,),
  (44,),
  (65,)],
 [(), (0,), (1,), (2,), (3,), (5,), (6,), (7,), (8,), (10,), (1
```

Test allele table (.json) as input

In [17]:
```
alleleTableFile =  "laml_io_tests/TLS_Bar8_formatted.json"
tmp = DLT_parser(alleleTableFile)
tmp.parse_json()
```

Out[17]:
```
{'AACAGGGCAGCAGTAG-1': {0: {(1, 1, 0): 1,
   (2, 2, 0): 1,
   (0, 1, 0): 225,
   (0, 3, 0): 1},
  1: {(1, 1, 0): 1, (2, 0, 0): 1, (0, 0, 0): 176},
  2: {(0, 1, 0): 1},
  3: {(1, 1, 0): 1, (0, 2, 0): 2, (0, 0, 0): 2},
  4: {(1, 1, 0): 1,
   (2, 1, 0): 1,
   (3, 2, 0): 1,
   (4, 1, 0): 1,
   (5, 3, 1): 1,
   (0, 1, 0): 318,
   (0, 4, 0): 1,
   (0, 0, 0): 3},
  5: {(0, 1, 0): 132, (0, 2, 0): 2, (0, 0, 1): 1, (0, 0, 0): 2},
  6: {(1, 1, 0): 1, (2, 2, 0): 1, (0, 0, 0): 1}},
 'AAGGTAAGTCAGTCGC-1': {0: {(3, 4, 0): 2,
   (1, 0, 0): 1,
```

In [18]:
```
tmp.data_struct['AACAGGGCAGCAGTAG-1'][0]
```

Out[18]:
```
{(1, 1, 0): 1, (2, 2, 0): 1, (0, 1, 0): 225, (0, 3, 0): 1}
```

In [19]:
```
a = tmp.set_alphabet()
a
```

Out[19]:
```
[[[-1,
   0,
   1,
   2,
   3,
   4,
   5,
   6,
   7,
   8,
   9,
   10,
   11,
   12,
   13,
   14,
   15,
   16,
   17,
```

In [20]:
```
Q = uniform_priors(tmp.alphabet, tmp.datatype, tmp.K, tmp.J, 0, "?")
```

In [21]:
```
len(Q)
```

Out[21]:
```
7
```

In [22]:
```
tmp.alphabet.M
```

Out[22]:
```
[10500, 9000, 10976, 21472, 11050, 6468, 3936]
```

In [23]:
```
tmp.alphabet.K, tmp.alphabet.J
```

Out[23]:
```
(7, 3)
```

Test priorFile (standard csv)

In [24]:
```
pfile = "/Users/gc3045/scmail_v1/LAML/examples/example1/priors.csv"
```

In [25]:
```
Q = read_priors(pfile)
```

In [26]:
```
len(Q)
```

Out[26]:
```
30
```

In [27]:
```
Q[0]
```

Out[27]:
```
{1: 0.03636363636363637,
 2: 0.012121212121212123,
 3: 0.8545454545454546,
 4: 0.07272727272727274,
 5: 0.012121212121212123,
 6: 0.012121212121212123}
```

Test priorFile (K=7, J=3)

In [28]:
```
pfile = "/Users/gc3045/scmail_v1/LAML/laml_io_tests/priors_Bar8.csv"
```

In [29]:
```
Q = read_priors(pfile)
```

In [30]:
```
len(Q), len(Q[0])
```

Out[30]:
```
(21, 28)
```

In [31]:
```
alphabet_ds = get_alphabet_prior(Q, J=3)
```

In [32]:
```
alphabet_ds
```

Out[32]:
```
[[{1: 0.9571428571428572,
   2: 0.0285714285714285,
   3: 0.6142857142857143,
   4: 0.0142857142857142,
   5: 0.9428571428571428,
   6: 0.9,
   7: 0.6142857142857143,
   8: 0.1714285714285714,
   9: 0.9142857142857144,
   10: 0.0857142857142857,
   11: 0.9285714285714286,
   12: 0.4714285714285714,
   13: 0.3,
   14: 0.5428571428571428,
   15: 0.0142857142857142,
   16: 0.3857142857142857,
   17: 0.9428571428571428,
   18: 0.2,
   19: 0.1571428571428571,
   20: 0.4571428571428571
```

In [33]:
```
K = 7
J = 3
```

In [34]:
```
a = Alphabet(K,J,alphabet_ds)
a.get_cassette_alphabet(0)
```

Out[34]:
```
[(1, 1, 1),
 (1, 1, 2),
 (1, 1, 3),
 (1, 1, 4),
 (1, 1, 5),
 (1, 2, 1),
 (1, 2, 2),
 (1, 2, 3),
 (1, 2, 4),
 (1, 2, 5),
 (1, 3, 1),
 (1, 3, 2),
 (1, 3, 3),
 (1, 3, 4),
 (1, 3, 5),
 (1, 4, 1),
 (1, 4, 2),
 (1, 4, 3),
 (1, 4, 4),
```

In [35]:
```
(0, 0, 0) in a.get_cassette_alphabet(0)
```

Out[35]:
```
False
```

In [36]:
```
tmp.parse_prior(pfile,J)
```

Out[36]:
```
([[{1: 0.9571428571428572,
    2: 0.0285714285714285,
    3: 0.6142857142857143,
    4: 0.0142857142857142,
    5: 0.9428571428571428,
    6: 0.9,
    7: 0.6142857142857143,
    8: 0.1714285714285714,
    9: 0.9142857142857144,
    10: 0.0857142857142857,
    11: 0.9285714285714286,
    12: 0.4714285714285714,
    13: 0.3,
    14: 0.5428571428571428,
    15: 0.0142857142857142,
    16: 0.3857142857142857,
    17: 0.9428571428571428,
    18: 0.2,
    19: 0.1571428571428571,
```

In [ ]:

In [ ]: