Convert from partition probabilities to n-relatedness

$F_{n \to \mathbf{n}}$ is the probability that a group of size n has family partition structure $\mathbf{n} \dashv n$, where

$$\mathbf{n} = (n_1, \ldots n_k) \quad \text{such that} \quad n_i, k \in \mathbb{N}, \quad n_1 + \ldots + n_k = n.$$

For example, if $\mathbf{n} = (1, 1, 2, 4)$, then the group of 8 individuals has 2 families with 1 individual each (i.e., 1 common ancestor), one family with 2 individuals, and 1 family with 4 individuals. The $F_{n \to \mathbf{n}}$ probabilities are determined by the group-formation model.

$\theta_{l \to m}$ is the probability that, if we draw l individuals without replacement from the group, they will have m common ancestors. These n-relatedness parameters are needed to determine the evolutionary dynamics of the population.

The matrix M is used to convert the partition probabilities $F_{n->\mathbf{n}}$ to the n-relatednesses parameters $\theta_{l->m}$. For example, in Appendix C of [Ohtsuki (2014) (https://royalsocietypublishing.org/doi/full/10.1098/rstb.2013.0359)](https://royalsocietypublishing.org/doi/full/10.1098/rstb.2013.0359), a 10×5 matrix is used to do the conversion for $n = 4$.

$$\begin{pmatrix} \theta_{1 \to 1} \\ \theta_{2 \to 1} \\ \theta_{2 \to 2} \\ \theta_{3 \to 1} \\ \theta_{3 \to 2} \\ \theta_{3 \to 3} \\ \theta_{4 \to 1} \\ \theta_{4 \to 2} \\ \theta_{4 \to 3} \\ \theta_{4 \to 4} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1/6 & 1/2 & 1/3 & 1 \\ 1 & 5/6 & 1/2 & 2/3 & 0 \\ 0 & 0 & 1/4 & 0 & 1 \\ 0 & 1/2 & 3/4 & 1 & 0 \\ 1 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} F_{n \to (1,1,1,1)} \\ F_{n \to (1,1,2)} \\ F_{n \to (1,3)} \\ F_{n \to (2,2)} \\ F_{n \to (4)} \end{pmatrix}$$

The script in `/scripts/matrix_M/save_matrix_Ms.py` shows how the numerator and denominator of each element of M can be calculated. The results are stored in `/results/matrix_M/` and can be conveniently read using the `read_matrix_M()` function.

In [1]:

```
import sys
sys.path.append('../functions/')
from my_functions import read_matrix_M
```

In [2]:

```
lm, nV, M_num, M_den = read_matrix_M('../results/matrix_M/matrix_M4.csv')
```

`lm` is a list of the possible θ subscripts (i.e., the $l \to m$)

In [3]:

```
lm
```

Out[3]:

```
[(1, 1),
 (2, 1),
 (2, 2),
 (3, 1),
 (3, 2),
 (3, 3),
 (4, 1),
 (4, 2),
 (4, 3),
 (4, 4)]
```

`nV` is a list of the possible partitions of size $n = 4$.

In [5]:

```
nV
```

Out[5]:

```
[[1, 1, 1, 1], [1, 1, 2], [1, 3], [2, 2], [4]]
```

The order of `lm` and `nV` gives the order of the rows and columns of the matrix M.

`M_num` and `M_den` give the numerator and denominator of each entry in the matrix M.

In [6]:

```
M_num
```

Out[6]:

```
array([[1, 1, 1, 1, 1],
       [0, 1, 3, 2, 6],
       [6, 5, 3, 4, 0],
       [0, 0, 1, 0, 4],
       [0, 2, 3, 4, 0],
       [4, 2, 0, 0, 0],
       [0, 0, 0, 0, 1],
       [0, 0, 1, 1, 0],
       [0, 1, 0, 0, 0],
       [1, 0, 0, 0, 0]])
```

In [7]:

```
M_den
```

Out[7]:

```
array([[1, 1, 1, 1, 1],
       [6, 6, 6, 6, 6],
       [6, 6, 6, 6, 6],
       [4, 4, 4, 4, 4],
       [4, 4, 4, 4, 4],
       [4, 4, 4, 4, 4],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]])
```

The matrix M can be reconstructed by an element-wise divide:

In [9]:

```
M = M_num / M_den
M
```

Out[9]:

```
array([[1.        , 1.        , 1.        , 1.        , 1.        ],
       [0.        , 0.16666667, 0.5       , 0.33333333, 1.        ],
       [1.        , 0.83333333, 0.5       , 0.66666667, 0.        ],
       [0.        , 0.        , 0.25      , 0.        , 1.        ],
       [0.        , 0.5       , 0.75      , 1.        , 0.        ],
       [1.        , 0.5       , 0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 1.        ],
       [0.        , 0.        , 1.        , 1.        , 0.        ],
       [0.        , 1.        , 0.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        , 0.        , 0.        
]])
```

In [ ]:

```

```