In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
```

EM Algorithm

In [2]:

```
NITER = 10
```

In [3]:

```
#starting value N: 25, 50, 100, 150
#N: Na, Nb, Nab, No
#Starting p: 1/3, 1/3, 1/3
#p: pa, pb, po
```

In [4]:

```
npheno = np.zeros(4)
pls = [[1.0/3,1.0/3,  1.0/3], [0.25, .25, .50], [0.8, 0.1, 0.1], [0.2, 0.3, .5]]
p = np.zeros(3) + 1.0/3
npheno[0], npheno[1], npheno[2], npheno[3] =  25, 50, 100, 150
```

In [5]:

```
print(p, npheno.sum())
```

```
[0.33333333 0.33333333 0.33333333] 325.0
```

In [6]:

```
def em_algo(npheno, p):
    #Expectation step
    Naa = npheno[0]*(p[0]**2/(p[0]**2 + 2*p[0]*p[2]))
    Nao = npheno[0]*(2*p[0]*p[2]/(p[0]**2 + 2*p[0]*p[2]))
    Nbb = npheno[1]*(p[1]**2/(p[1]**2 + 2*p[1]*p[2]))
    Nbo = npheno[1]*(2*p[1]*p[2]/(p[1]**2 + 2*p[1]*p[2]))
    Nab = npheno[2]
    No = npheno[3]

    #Maximization
    p[0] = (2*Naa + Nao + Nab)/(2*npheno.sum())
    p[1] = (2*Nbb + Nbo + Nab)/(2*npheno.sum())
    p[2] = (Nao + Nbo + No)/(2*npheno.sum())

    return p
```

In [8]:

```
for p in pls:
    print(0, ' & ', str(p[0])[:7] , ' & ' , str(p[1])[:7] , ' & ' , str(p[2])[:7] )
    for itera in range(NITER):
        p = em_algo(npheno, p)
        print(itera + 1, ' & ', str(p[0])[:7] , ' & ' , str(p[1])[:7] , ' & ' , str
    print("------------------------------")
```

```
0  &  0.20163  &  0.25280  &  0.31478
1  &  0.20163  &  0.25280  &  0.31478
2  &  0.20163  &  0.25280  &  0.31478
3  &  0.20163  &  0.25280  &  0.31478
4  &  0.20163  &  0.25280  &  0.31478
5  &  0.20163  &  0.25280  &  0.31478
6  &  0.20163  &  0.25280  &  0.31478
7  &  0.20163  &  0.25280  &  0.31478
8  &  0.20163  &  0.25280  &  0.31478
9  &  0.20163  &  0.25280  &  0.31478
10  &  0.20163  &  0.25280  &  0.31478
------------------------------
0  &  0.20163  &  0.25280  &  0.31478
1  &  0.20163  &  0.25280  &  0.31478
2  &  0.20163  &  0.25280  &  0.31478
3  &  0.20163  &  0.25280  &  0.31478
4  &  0.20163  &  0.25280  &  0.31478
5  &  0.20163  &  0.25280  &  0.31478
6  &  0.20163  &  0.25280  &  0.31478
7  &  0.20163  &  0.25280  &  0.31478
8  &  0.20163  &  0.25280  &  0.31478
9  &  0.20163  &  0.25280  &  0.31478
10  &  0.20163  &  0.25280  &  0.31478
------------------------------
0  &  0.20163  &  0.25280  &  0.31478
1  &  0.20163  &  0.25280  &  0.31478
2  &  0.20163  &  0.25280  &  0.31478
3  &  0.20163  &  0.25280  &  0.31478
4  &  0.20163  &  0.25280  &  0.31478
5  &  0.20163  &  0.25280  &  0.31478
6  &  0.20163  &  0.25280  &  0.31478
7  &  0.20163  &  0.25280  &  0.31478
8  &  0.20163  &  0.25280  &  0.31478
9  &  0.20163  &  0.25280  &  0.31478
10  &  0.20163  &  0.25280  &  0.31478
------------------------------
0  &  0.20163  &  0.25280  &  0.31478
1  &  0.20163  &  0.25280  &  0.31478
2  &  0.20163  &  0.25280  &  0.31478
3  &  0.20163  &  0.25280  &  0.31478
4  &  0.20163  &  0.25280  &  0.31478
5  &  0.20163  &  0.25280  &  0.31478
6  &  0.20163  &  0.25280  &  0.31478
7  &  0.20163  &  0.25280  &  0.31478
8  &  0.20163  &  0.25280  &  0.31478
9  &  0.20163  &  0.25280  &  0.31478
10  &  0.20163  &  0.25280  &  0.31478
------------------------------
```