Analysis of example data

In [1]:

```
#Import PyMSQ module and relevant libraries
# from PyMSQ import msq
import msq
import numpy as np
import pandas as pd
import time
```

In [2]:

```
# Import example data
gmap = pd.read_table("chr.txt", sep=" ")
group = pd.read_table("group.txt", sep=" ")
meff = pd.read_table("effects.txt", sep=" ")
gmat = pd.read_table("phase.txt", header=None, sep=" ")
ped = pd.read_table("ped.txt", sep=" ")
# gmap, meff, gmat, group, ped = msq.example_data()
```

One of our main research interests is estimating Mendelian sampling variances for milk traits (fat, protein, and pH), as well as covariances and correlations between traits. We're also interested in creating a similarity matrix based on Mendelian sampling values for aggregate genotype assuming index weights of one.

In [3]:

```
# check main information for errors and create class object to store the information
index_wt = [1, 1, 1]
start = time.time()
data = msq.Datacheck(gmap=gmap, meff=meff, gmat=gmat, group=group,
                     indwt=index_wt, progress=True)
print('Time taken: ', round(time.time() - start, 2), 'secs')

del gmap, meff, gmat, group, index_wt
```

```
Converting phased haplotypes to genotypes
Data passed the test!
Number of individuals:   265
Number of groups:        1 :  ['F']
Number of specific maps: 1
Number of chromosomes:   29
Total no. markers:       39780
Number of trait(s):      3
Trait name(s) and Index weight(s)
fat :  1
pH :  1
protein :  1
Time taken:  2.86 secs
```

In [4]:

```
# create population covariance matrices
start = time.time()
covmatrices = msq.popcovmat(info=data, mposunit="cM", method=1)
print('Time taken: ', round(time.time() - start, 2), 'secs')
```

```
Time taken:  1.97 secs
```

In [5]:

```
# Estimate Mendelian sampling (co)variance
start = time.time()
msvmsc_g = msq.msvarcov_g(info=data, covmat=covmatrices,
                          sub_id=None, progress=True)
print('Time taken: ', round(time.time() - start, 2), 'secs')
print(msvmsc_g)
```

```
Progress: |██████████████████████████████████████████████████| 100% Complete
Time taken:  27.81 secs
        ID Group       fat    pH_fat        pH  protein_fat  protein_pH  \
0    10001     F  0.000466  0.000154  0.001805     0.000136   -0.000157
1    10002     F  0.023262  0.014908  0.013225     0.017306    0.011082
2    10003     F  0.021665  0.013760  0.010475     0.015662    0.009849
3    10004     F  0.000949  0.000317  0.004836    -0.000044   -0.000158
4    10005     F  0.021892  0.014447  0.012676     0.015576    0.010120
..     ...   ...       ...       ...       ...          ...         ...
260  10261     F  0.022112  0.012953  0.010050     0.017090    0.009524
261  10262     F  0.023853  0.016183  0.012896     0.017691    0.012189
262  10263     F  0.000604  0.000096  0.002694     0.000025    0.000126
263  10264     F  0.022809  0.015301  0.012748     0.017123    0.011174
264  10265     F  0.022491  0.016226  0.013892     0.016460    0.011973

      protein    AG_fat     AG_pH  AG_protein        AG
0    0.002321  0.000756  0.001802    0.002300  0.004858
1    0.015098  0.055475  0.039215    0.043486  0.138176
2    0.013710  0.051087  0.034084    0.039221  0.124392
3    0.001352  0.001223  0.004995    0.001149  0.007367
4    0.013318  0.051915  0.037243    0.039013  0.128172
..        ...       ...       ...         ...       ...
260  0.015024  0.052155  0.032527    0.041637  0.126319
261  0.015089  0.057727  0.041268    0.044968  0.143963
262  0.001841  0.000725  0.002916    0.001992  0.005633
263  0.014931  0.055232  0.039223    0.043227  0.137683
264  0.013582  0.055177  0.042091    0.042014  0.139282

[265 rows x 12 columns]
```

In [6]:

```
# Estimate Mendelian correlation
start = time.time()
msvmsc_gcorr = msq.msvarcov_gcorr(msvmsc=msvmsc_g)
print('Time taken: ', round(time.time() - start, 2), 'secs')
print(msvmsc_gcorr)
```

```
Time taken:  0.05 secs
        ID Group    pH_fat  protein_fat  protein_pH    AG_fat     AG_pH  \
0    10001     F  0.168243     0.130355   -0.076619  0.502305  0.608696
1    10002     F  0.849938     0.923450    0.784271  0.978501  0.917349
2    10003     F  0.913412     0.908779    0.821836  0.984097  0.944224
3    10004     F  0.148098    -0.038853   -0.061912  0.462340  0.836841
4    10005     F  0.867249     0.912187    0.778874  0.980058  0.923970
..     ...   ...       ...          ...         ...       ...       ...
260  10261     F  0.868944     0.937647    0.775095  0.986844  0.912919
261  10262     F  0.922706     0.932494    0.873801  0.985100  0.957769
262  10263     F  0.075172     0.023946    0.056342  0.393020  0.748465
263  10264     F  0.897337     0.927839    0.809875  0.985611  0.936218
264  10265     F  0.918000     0.941739    0.871634  0.985839  0.956892

     AG_protein
0      0.684896
1      0.952081
2      0.949738
3      0.364236
4      0.944280
..          ...
260    0.955789
261    0.964839
262    0.618574
263    0.953390
264    0.965978

[265 rows x 8 columns]
```

In [7]:

```
# Derive similarity matrix based on Mendelian sampling values for aggregate genotype
start = time.time()
sim_g = msq.simmat_g(info=data, covmat=covmatrices, sub_id=None,
                     chrinterest="none", save=False, stdsim=False,
                     progress=True)
print('Time taken: ', round(time.time() - start, 2), 'secs')
```

```
Progress: |██████████████████████████████████████████████████| 100% Complete
Creating similarity matrix based on aggregate genotype
Progress: |██████████████████████████████████████████████████| 100% Complete
Time taken:  6.26 secs
```

In [8]:

```
sim_g = np.array(sim_g[0][0])
sim_g.shape
```

Out[8]:

```
(265, 265)
```

In [9]:

```
# Write data to file for visualization in R
msvmsc_g.to_csv("msvmsc_g.csv", header=True, index=False)
msvmsc_gcorr.to_csv("msvmsc_gcorr.csv", header=True, index=False)
ped.to_csv("pedigree.csv", header=True, index=False)
np.save("sim_g.npy", sim_g)
```