**Debug process 2011-10-02**

# The decomposition of *S* matrix

In previous versions of my code, I used to decompose the *S* matrix into *H* and *R* where , but we apply this trick, the generated *H* and *R* cannot fully satisfy this standard, i.e., there is a difference between and *S.* In most cases, the difference is not big (around ), but in Alison’s Warfarin data, the difference can range from -0.34 to 0.4 which is really huge. Based on this observation, I delete all the code related to *H* and *R*, (*H* and *R* are used in the calculation for the conditional expectation and variance, and to facilitating the computation for ) and replace them with the code using *S* directly. I have double checked the results when the difference between and is small, the result using *H,R* are almost identical to the result using *S* directly, but the later one is slower, since the computation for is proportional to the sample size.

BTW, I have a stupid question about the *S*. Do I have to force since the genotype of the same people is always the same? But when I do so, the rank of *S* becomes very big so that I guess I should not do so.

**The EM algorithm**

After the correction for the *S* matrix decomposition, I tried to rerun the Gain data. For previous Gain data analysis, I just ran the joint test for all possible genes combinations and the epitasis test for the genes combinations which has a significant p value for joint test. Since there are only 3 genes combinations have a significant p value, I ran the Epitasis test for 3 times. But this time I want to run the One Gene test for each gene combination so I found another problem for the EM algorithm.

In the One Gene test, if I want to test the hypothesis, I need to use the EM algorithm to estimate and first. I found out the in some cases, the estimation for can not converge, it seems that it want to converge to 0 (I have checked the estimated for each iteration for one particular case and found that drops from 1.5 to 0.004 after 6,000 iterations, then I tried another genes combination and similar things happened.) Then I wondered whether my derivation or coding is wrong, but I checked them for three times and everything is correct. Finally, I ran the simulation again and the result shows me that the EM algorithm works perfectly for the simulated data. Even I set both and to 0, the estimations will converge at around 0.02.

I think for the One Gene test, it is similar to your test, so I wonder whether you met such problem before and how you solve it. (I read the paper again and I think there may be some typo in the appendix.) Also, since the simulation study shows that if both of and are 0, the estimations are around 0.02, I can just set to zero if the estimation for is smaller than 0.01?