Snp HeRitability Estimation Kit (SHREK) Development Log

Sam Shing Wan, Choi
`choishingwan@gmail.com`

Johnny Sheung Him, Kwan
`shkwan@hku.hk`

Henry Chi-Ming, Leung
`cmleung2@cs.hku.hk`

February 3, 2016

Contents

1 Assumption in implementation

To make life easier, I made a number of assumption when I implement the programme and they are listed as follow

1. The intervals within each individual bed files are non-overlapping

2. SNPs filtered out when reading the p-value file will not be included in the output no matter what (avoid storing unnecessary information)

3. We will remove *all* SNPs that are in perfect LD with each-other

    (a) Realistically, we can not remove *all* SNPs that are in perfect LD with each-other without first computing the LD matrix for the whole genome. However, that will incur a large speed penalty to the algorithm. To compromise, we remove SNPs that are in perfect LD with each-other *within the same window*. Take into consideration of the calculation of variance, we will need to plan forward and consider SNPs within 4 windows. That's definitely overkill but that avoid problem of crazy perfect LD interactions (to be honest, I would consider those as error, yet it is easier for me to just remove them)

4. We do not consider a SNP ambiguous unless we need to perform flipping

2 January 25, 2016

We are going to re-write SHREK starting from today. There are a number of main goal in this re-work

1. The whole parameter parsing

2. Better help messages

3. Allow filtering by imputation score (0-1)

4. Filtering of SNPs that have different reference/alternative allele with reference

5. Use Armadillo instead of EIGEN to improve speed

6. Filter *all* SNPs that are in perfect LD

7. Implement the advance variance calculation

8. Start documenting the codes

2.1 Finished today

1. Finished the error message for different modes

2. Finished the parsing for binary trait

3. Finished the parsing for quantitative trait

4. Disabled risk prediction, should focus on the heritability estimation first

I have not perform debugging, there can be error

3 January 26, 2016

Continue on where I left yesterday Assumptions we made in the programme

1. p-value file has header

2. the numbers return from Command class are index (0 based)

I really want to write a more compacted code. Will try to restructure the command class

3.1 Finished today

1. Finished refining the usage messages

2. Finished the basic parameter parsing (Still haven't finish the index based parameter though)

4 January 27, 2016

4.1 Aim

We want to at least finish the region parsing and SNP class update today

4.2 Finished today

1. Completed the update of the Command class (compiled without problem)

2. Completed the update of the Region class (compiled without problem)

3. Completed the update of the Snp class (compiled without problem)

I also *haven't* test run the programme, so there might be additional syntax / logic error that are not picked up at the current stage.

5 January 28, 2016

5.1 Finished today

1. Finished the checking of reference panel

2. Found a different algorithm for popcount, which should be platform independent

3. Installed armadillo and OpenBLAS

    (a) I have to add the OpenBLAS to the PATH and LD_LIBRARY_PATH for armadillo to detect it

4. Updated the makefile accordingly

(a) However, I have not updated the decomposition class, therefore there will be problem when we try to compile the programme

5. Updated the r^2 calculation function such that it is using the correct sample size

   (a) The sample size when calculating the r^2 is defined as the number of samples containing *both* SNPs

   (b) Because of how complicated the genotype class is, I really don't want to modify it

   (c) The main bottleneck for the computation is in the decomposition anyway, so it should be fine

6   January 29, 2016

Need to at least finish the getSNP function. Must be careful with it as this is not only complicated, but also extremely important for the whole programme.

I spent whole day in annotating the document of the class to make sure I don't overlook any problem.

7   February 1, 2016

Start to implement the getSNP function. A number of point to note

1. First get all the SNPs that passed all filtering (e.g. MAF)

2. Then for each window, calculate the LD

3. When encountered with perfect LD, we retain the *first* SNP

   (a) The summary statistics will be the mean of all SNPs in perfect LD

   (b) All SNPs except the first are *removed*

   (c) So they will not appear in subsequent analysis

8 February 2, 2016

Try to perform extensive check on codes that has been written

8.1 Finished today

1. Finished debugging, everything up until the initialization of genotype-FileHandler is ok

9 February 3, 2016

Sat in a meeting with Pak today, now we have a few things to-do.

9.1 To Do

1. Compare the condition number of the LD matrix before and after the LD correction

2. Compare the performance of case control estimation with different sample size

3. For LD correction, might also worth testing how the density and total number of SNPs affect the estimation

4. For effect size estimation, don't use the complicated method

9.2 Finished today

We have switched to use list instead of deque. The benefit of list is that it is a double linked list, therefore it is much easier for one to remove the middle part. Such property is ideal for the handling of perfect LD. However, it also means that we cannot access the elements of the list by index. The only way for us to access the elements is to use the iterators and the advance function from the iterator library. Although this makes the implementation more difficult, it should be able to help us to speed up our code even if we want to perform perfect LD removal.

1. Extensive testing of the previous classes

2. Now use iterators for the boundary, this is more dynamic and can work with the perfect LD removal without inducing much complication into the algorithm (however, the implementation itself can be complicated)

9.3   Problem identified today

So assuming that the SNPs A B C are occurring in the genome by alphabetical order, when we calculate the LD of A with B C, we do not observe any perfect LD. This will lead us to continue with the calculation of SNP B and C. However, if SNP B and SNP C is in perfect LD, we will need to go back to the LDs of SNP A to remove the column of SNP C. Not only is this procedure complicates the implementation, it will also significantly slow down the computation because of the constant resizing of the matrix.

To conclude, it might be better for us to use the original "sausage" approach, where we should also locate all the perfect LD location. Then we can remove the perfect LD afterwards.