CrypticIBDcheck vignette: Exploring cryptic relatedness with genome-wide data

Annick Nembot-Simo, Jinko Graham and Brad McNeney

April 23, 2012

1 Introduction

We demonstrate the use of **CrypticIBDcheck** to explore cryptic relatedness using genome-wide data from single nucleotide polymorphisms (SNPs) in HapMap Phase 3, release # 28. The data are from the LWK (Luhya in Webuye, Kenya) population and were downloaded from the HapMap website (`http://hapmap.ncbi.nlm.nih.gov/`) in March 2012. While all LWK individuals are nominally unrelated, the analysis of Pemberton *et al.* (2010) has suggested several close relationships, which we uncover here.

Our analysis illustrates that a genome-wide panel of SNPs, "thinned" to a subset of approximately independent markers, contains enough information to identify relationships up to second degree (e.g., half-siblings), and to suggest relationships up to third degree (e.g., first cousins). The steps for the analysis are as follows. First, we download the data from HapMap and read it into an object of class `IBD` suitable for input to `IBDcheck()`. Second, PLINK (Purcell *et al.*, 2007) is used to perform the thinning. Third, the thinned data produced by PLINK are passed to `IBDcheck()` to augment the `IBD` object with estimated IBD coefficients. Fourth, the plot method of the `IBD` class is used to graphically display estimated IBD coefficients and explore possible relationships. We compare the relationships that are suggested in this display to those described in Pemberton *et al.*.

The relationships among the LWK individuals inferred by Pemberton *et al.* are summarized in Table 1. The data used by these authors, reportedly

Table 1: Relationships among LWK individuals identified by Pemberton *et al.* (2010) based on data downloaded on September 9, 2009. Individuals who are not available as of March 2012 are marked with an asterisk.

First Individual	Second Individual	Relationship
NA19381	NA19382	parent-offspring
NA19432*	NA19434	parent-offspring
NA19432*	NA19444	parent-offspring
NA19470	NA19469	parent-offspring
NA19046	NA19045*	full sibling
NA19352	NA19347	full sibling
NA19374	NA19373	full sibling
NA19397	NA19396	full sibling
NA19434	NA19444	full sibling
NA19470	NA19443	full sibling
NA19027	NA19311	second degree
NA19334	NA19313	second degree
NA19380	NA19382	second degree
NA19443	NA19469	second degree

downloaded in September 2009, would be from HapMap realease #27. Not all of the individuals in the Pemberton *et al.* dataset are present in the current HapMap release #28. Excluding pairs where one member is not currently available leaves 2 parent-offspring, 5 full sibling and 4 second degree (half sibling, grandparent-grandchild or avuncular) relationships.

2 Downloading the HapMap data

We use functions from the `chopsticks` package (Leung, 2011) to download data from the HapMap website. `chopsticks` (formerly `snpMatrix`) is automatically loaded with `CrypticIBDcheck`:

```
> library(CrypticIBDcheck)
```

`chopsticks` implements the `snp.matrix` class, a data structure that compactly represents SNP genotype data, allowing storage and manipulation

of genome-wide datasets in R. A `snp.matrix` object is a matrix comprised of genotyes stored as objects of type `raw`. Genotypes are coded as 0, 1 or 2 copies of an index allele, taken to be the first in an alphabetical list of the two alleles at the SNP. Rows of the matrix correspond to subjects and columns to SNPs. The `snp.matrix` object cannot include auxiliary data on either subjects or SNPs. Such information may be stored in data frames that are separate from the `snp.matrix` object. Though there is no formal support for these auxiliary data frames, they are used frequently in the documentation and examples of the `chopsticks` package, and are given the names `subject.support` and `snp.support` for information on subjects and SNPs, respectively.

We download the genotype data for each autosome from the HapMap repository with the `read.HapMap.data` function of `chopsticks`:

```
> lwkdat <- vector(mode = "list", length = 22)
> names(lwkdat) <- paste("chr", 1:22, sep = "")
> for (i in 1:22) {
+     uu <- paste("http://hapmap.ncbi.nlm.nih.gov/downloads/genotypes/",
+         "latest_phaseIII_ncbi_b36/hapmap_format/polymorphic/genotypes_chr",
+         i, "_LWK_phase3.2_nr.b36_fwd.txt.gz", sep = "")
+     lwkdat[[i]] <- read.HapMap.data(uu)
+ }
```

All URLs listed in this vignette were valid at the time of writing (April 2012), but are subject to change. Each list element `lwkdat[[i]]`, for chromosome `i`, will itself be a list, with components `snp.data` and `snp.support`. The component `snp.data` is a `snp.matrix` object, while `snp.support` is a data frame that contains information on each SNP such as its alleles and physical map position. A `subject.support` data frame is not created by `read.HapMap.data`, but the Appendix outlines an approach to create one yourself, if necessary.

We can now combine data from the different chromosomes:

```
> snp.data <- lwkdat[[1]]$snp.data
> snp.support <- lwkdat[[1]]$snp.support[, c("Chromosome", "Position")]
> for (i in 2:22) {
+     snp.data <- cbind(snp.data, lwkdat[[i]]$snp.data)
+     snp.support <- rbind(snp.support, lwkdat[[i]]$snp.support[,
+         c("Chromosome", "Position")])
+ }
```

and remove SNPs with multiple map positions:

```
> dd <- duplicated(snp.support)
> snp.support <- snp.support[!dd, ]
> snp.data <- snp.data[, !dd]
```

Finally, we may use the function `new.IBD()` to create an object of class `IBD`. We consider all members of the sample to be randomly sampled from the population, so that they will all be used by `IBDcheck()` to estimate conditional IBS probabilities.

```
> dat <- new.IBD(snp.data, Chromosome = snp.support$Chromosome,
+     Position = snp.support$Position, popsam = rep(TRUE, nrow(snp.data)))
```

3 Using PLINK to thin the marker set

We use PLINK's facilities for linkage-disequilibrium-based SNP pruning to thin the marker set to one in which all SNPs are approximately independent of each other. In what follows we assume that PLINK is available on the user's system and is part of their `path`. To verify that PLINK is available, type the following from R:

```
> system("plink --no-web --help")
```

You should see a summary of the program's help options. **CrypticIBDcheck** does not include any formal interface with PLINK. Instead, we have written a convenience function called `thin` that can be used to call PLINK and perform the thinning. The source code for `thin` is contained in the `scripts` directory of the package, and can be `source()`'d into an R session with:

```
> source(file.path(system.file(package = "CrypticIBDcheck"), "scripts",
+     "thin.R"))
```

The first argument to `thin` is an `IBD` object. The remaining arguments, `win`, `shift` and `r2thresh`, are passed to PLINK to control how the thinning is done. PLINK's algorithm for selecting SNPs to be removed is a moving window approach comprised of the following steps:

1. Fix a window of width `win`.

2. Calculate pairwise squared allelic correlations r^2 for all SNPs in the window.

3. For each pair with allelic correlation greater than the threshold `r2thresh`, discard one member of the pair. (There is some ambiguity in the PLINK documentation about the how this step is implemented.)

4. Move the window by `shift` SNPs and repeat steps 1-3.

In the PLINK documentation, Section 10, there is an example that suggests values `win=100`, `shift=25` and `r2thresh=0.2`. In gene-drop simulations, we have found that a much stricter `r2thresh` of between 0.005 and 0.01 is required to reduce dependence between markers for inferring cryptic relatedness with genome-wide SNP data. The `IBD` object `dat` can be thinned with an `r2thresh` value of 0.005 as follows:

```
> t.dat <- thin(dat, win = 100, shift = 25, r2thresh = 0.005)
```

Each call to `thin()` will create, and subsequently delete, the following files in the user's working directory: mydata.ped, mydata.map, plink.log, plink.prune.in, and plink.prune.out.

4 Using IBDcheck() to estimate IBD coefficients

We use `IBDcheck()` to estimate IBD coefficients for pairs of study subjects and for pairs of simulated subjects. The simulated relationships considered in this example are: MZ twins/duplicates, parent-offspring, full siblings, half siblings, and first cousins. In addition, pairs of unrelated subjects are simulated. The arguments to `IBDcheck()` are: (i) an `IBD` object; (ii) a list of parameters that controls QC filtering, created by the `filter.control()` function; and (iii) a list of parameters that controls the simulations, created by the `sim.control()` function. The last two arguments are optional, and if not specified are given default values described in the help files of `filter.control()` and `sim.control()`. We leave the QC filtering options at their default values. We specify that an LD model need not be fit, and specify the relationships to simulate as follows:

```
> ss <- sim.control(simulate = TRUE, fitLD = FALSE, rships = c("unrelated",
+     "MZtwins", "parent-offspring", "full-sibs", "half-sibs",
+     "cousins"), nsim = rep(200, 6))
> cibd <- IBDcheck(t.dat, simparams = ss)
```

On Unix-like systems, the call to `IBDcheck()` will print the following warning for each chromosome of data:

```
Warning: parameter file has no LD model appended.
Assuming linkage equilirbiurm and given allele frequencies.
```

These warnings are to be expected and can be ignored.

5 Plotting the IBD object

We can now plot the IBD object `cibd` as follows:

```
> ibdpairs <- plot(cibd)
```

In this example, the plotting function produces six plots, shown in Figures 1–3, and an output data frame `ibdpairs` that contains information on study pairs flagged by the last four plots in Figures 2 and 3:

```
   member1 member2         pz0       pz1     relationship
1  NA19381 NA19382 0.004458855 1.0024774 parent-offspring
2  NA19470 NA19469 0.007283001 1.0123109 parent-offspring
3  NA19470 NA19443 0.249534110 0.4864767        full sibs
4  NA19397 NA19396 0.236143770 0.5230856        full sibs
5  NA19352 NA19347 0.229077961 0.5164333        full sibs
6  NA19434 NA19444 0.265322306 0.5164379        full sibs
7  NA19374 NA19373 0.228584986 0.5114019        full sibs
8  NA19027 NA19311 0.484353454 0.5088007        half sibs
9  NA19334 NA19313 0.500990846 0.5092976        half sibs
10 NA19443 NA19469 0.541079762 0.4624135        half sibs
11 NA19380 NA19382 0.444633963 0.5613230        half sibs
12 NA19380 NA19381 0.660470086 0.3343943          cousins
13 NA19397 NA19350 0.846029547 0.1581139          cousins
14 NA19028 NA19385 0.860153761 0.1434600          cousins
15 NA19359 NA19309 0.681516041 0.3286831          cousins
16 NA19452 NA19451 0.765855213 0.2496486          cousins
```

The first plot to appear (Figure 1, left panel) is non-clickable and shows the estimated IBD coefficients for all pairs of study subjects, along with the prediction ellipse for unrelated, simulated pairs. Subsequent plots (Figure 1, right panel and all of Figures 2 and 3) are clickable and correspond to each relationship requested in the call to `IBDcheck()`. These relationship-specific plots are for identifying pairs of study subjects which could have the relationship. The plotting regions are restricted to the neighborhood of the prediction ellipse for the simulated pairs of that relationship, which is also drawn. If, however, the plotting region overlaps with the prediction ellipse for simulated unrelated pairs, the ellipse for simulated unrelated pairs is drawn as well. Points falling within the prediction ellipse for the relationship and outside the prediction ellipse for unrelated pairs are automatically flagged. In addition, users may click on points of study pairs that appear to be related but are not automatically flagged, such as the apparent parent-offspring pair `NA19470:NA19469` that appears just outside the prediction ellipse for simulated parent-offspring pairs. The data frame `ibdpairs` is comprised of information on pairs that have been flagged on the different plots, either automatically or interactively by the user through clicking the mouse.

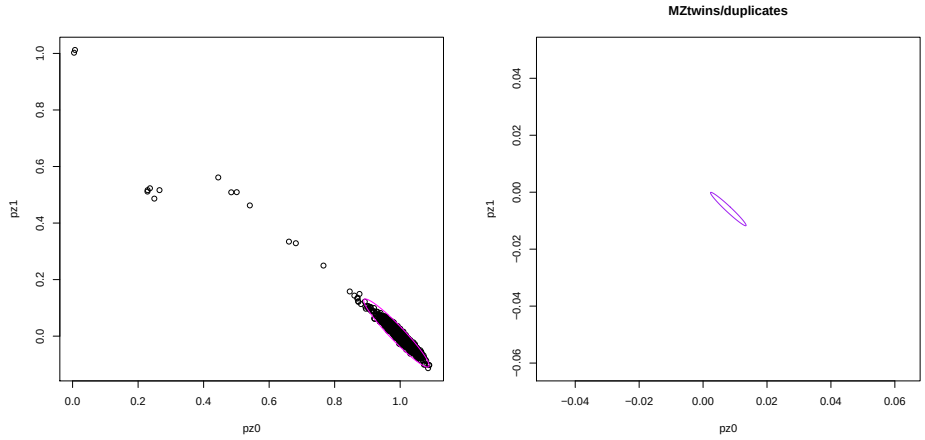

Figure 1: All observed pairs with the prediction ellipse for unrelated pairs (left panel) superposed, and the prediction ellipse for MZ twins/duplicates (right panel). There are no estimated IBD coefficients in the vicinity of the prediction ellipse for MZ twins/duplicates.

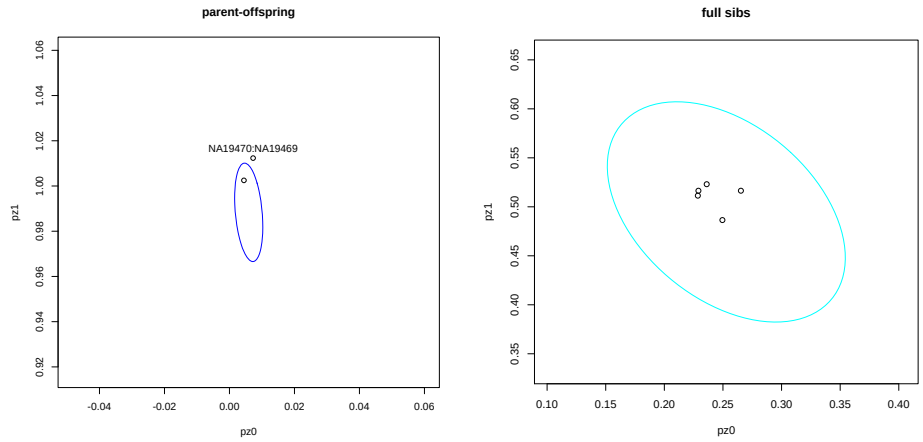

Figure 2: Observed pairs with prediction ellipses for parent-offspring pairs (left panel) and full siblings (right panel) superposed.

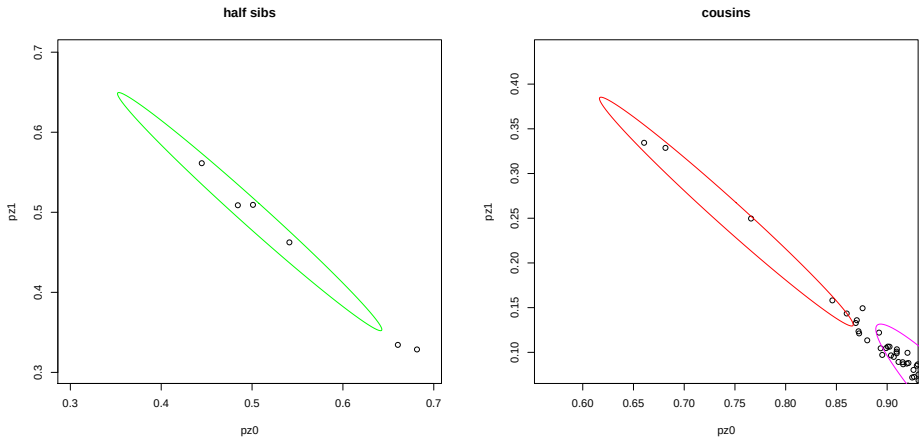

Figure 3: Observed pairs with prediction ellipses for second degree relative pairs such as half sibling (left panel) and third degree relative pairs such as first cousins (right panel) superposed. In the right panel, the prediction ellipse based on pairs of unrelated pairs of subjects (magenta line) appears in the bottom-right of the plot.

The pairs of subjects identified by plotting the `IBD` object `cibd` include all parent-offspring, full sibling and second order relationships in the currently-

available LWK sample that were identified by Pemberton *et al.*. These authors did not attempt to identify first cousins, because the likelihood method they used is not considered to be reliable for inference of cousin relationships (Boehnke and Cox, 1997; Epstein *et al.*, 2000). The graphical approach of **CrypticIBDcheck** is exploratory rather than inferential, and allows the user to informally explore possible first-cousin relationships. The following pairs were identified as potential first cousins (rearranged from the original output for convenience):

```
member1 member2         pz0       pz1     relationship
NA19380 NA19381 0.660470086 0.3343943          cousins
NA19359 NA19309 0.681516041 0.3286831          cousins
NA19452 NA19451 0.765855213 0.2496486          cousins
NA19397 NA19350 0.846029547 0.1581139          cousins
NA19028 NA19385 0.860153761 0.1434600          cousins
```

It seems plausible that the first three pairs are relatives, as their estimated IBD coefficients are clearly separated from the magenta prediction ellipse for unrelated pairs that appears in the bottom-right of the display in the right panel of Figure 3. However, the last two pairs in this list are not clearly separated from the cloud of points in and around the prediction ellipse for unrelated pairs, and may be unrelated pairs whose estimated IBD coefficients fall in the tail of that distribution.

6 Summary

In this vignette we have shown how to use **CrypticIBDcheck** to explore cryptic relatedness with genome-wide SNP data from the HapMap LWK sample. The full panel of 1,475,584 SNPs was aggresively thinned to an approximately independent subset of size 14,289, from which IBD coefficients were estimated. The exploratory display of these estimated IBD coefficients, along with those from simulated pairs of known relationship, enabled us to identify all close relationships in the currently-available LWK data described in Pemberton *et al.* (2010). In addition, our exploratory approach was able to suggest three possible first-cousin relationships that were not identified by Pemberton *et al.*, due to limitations of the formal likelihood-based methods they used.

In our simulations, we have found that correctly specifying the underlying LD model is important for getting the reference clusters right. For example, with dense genome-wide SNPs, when pairs from parent-offspring or half-sibling (i.e., unilineal) relationships are simulated under a mis-specified model of linkage equilibrium, their estimated coefficients for two alleles IBD tend to be slightly positive, even though the true IBD coefficients are zero. On the IBD plot, this has the effect of shifting reference clusters for half-siblings down and to the left, away from the diagonal line of slope -1 where they should lie. For parent-offspring pairs, the reference clusters are shifted downwards. This shifting problem is eliminated by aggresively thinning the SNPs to an approximately independent set, as discussed in Section 3.

For genome-wide data, an alternate approach to exploring cryptic relatedness is described in Section 5.2 of the `DataCleaning` vignette in the **GWASTools** Bioconductor package (Gogarten *et al.*, 2012). The `ibdPlot()` function of **GWASTools** treats estimates of IBD coefficients as observed values and uses results from Hill and Weir (2011) on the moments of the distribution of IBD coefficients to produce reference clusters. *Ad hoc* inflations of these clusters are suggested to account for the fact that IBD coefficients must be estimated.

7 Appendix

In this vignette, additional information on subjects is not needed and so there is no need to create a `subject.support` data frame. However, for other HapMap populations comprised of mother-father-offspring trios, such as CEU (Utah residents with Northern and Western European ancestry from the CEPH collection), information on known relationships would be required to explore cryptic relatedness. If, for example, we wish to subset the CEU sample to include only the mothers and fathers, we might proceed as follows:

```
> uu <- paste("http://hapmap.ncbi.nlm.nih.gov/downloads/genotypes/",
+     "latest_phaseIII_ncbi_b36/relationships_w_pops_121708.txt",
+     sep = "")
> hapmap.info <- read.table(uu, header = TRUE, as.is = TRUE)
> subject.support <- hapmap.info[hapmap.info$population == "CEU",
+     ]
> parent <- (subject.support$mom == 0 | subject.support$dad ==
+     0)
```

```
> subject.support <- subject.support[parent, ]
> rm(hapmap.info)
```

where we have used the fact that mothers and fathers are "founders" and therefore have no mother (`mom==0`) or father (`dad==0`) in the trio. The subject information obtained by the above code snippet is for all CEU parents in the `relationships_w_pops_121708.txt` file. However, the parents with genotype data in the current release could be a subset of these. To subset `subject.support` to the subjects with genotype data in a `snp.matrix` object called `snp.data`, we could proceed as follows:

```
> id = rownames(snp.data)
> subject.support = subject.support[match(id, subject.support$IID),
+     ]
```

References

Boehnke M, Cox NJ (1997). "Accurate inference of relationships in sib-pair linkage studies." *Am. J. Hum. Genet.*, **61**(2), 423–429.

Epstein MP, Duren WL, Boehnke M (2000). "Improved inference of relationship for pairs of individuals." *Am. J. Hum. Genet.*, **67**(5), 1219–1231.

Gogarten SM, Laurie C, Bhangale T, Conomos MP, Laurie C, McHugh C, Painter I, Zheng X, Shen J, Swarnkar R (2012). *GWASTools: Tools for Genome Wide Association Studies*. R package version 1.2.0.

Hill WG, Weir BS (2011). "Variation in actual relationship as a consequence of Mendelian sampling and linkage." *Genet Res (Camb)*, **93**(1), 47–64.

Leung HT (2011). *chopsticks: The snp.matrix and X.snp.matrix classes*. R package version 1.18.3, URL `http://outmodedbonsai.sourceforge.net/`.

Pemberton TJ, Wang C, Li JZ, Rosenberg NA (2010). "Inference of unexpected genetic relatedness among individuals in HapMap Phase III." *Am. J. Hum. Genet.*, **87**, 457–464.

Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira M, Bender D, Maller J, Sklar P, de Bakker P, Daly M, Sham P (2007). "PLINK: a tool set

for whole-genome association and population-based linkage analyses." *The American Journal of Human Genetics*, **81**, 559–575.