MethylGenotyper: Accurate estimation of SNP genotypes from DNA methylation data

2024-03-13

Introduction

The `MethylGenotyper` package provides functions to infer genotypes (produce a standard VCF file) for specific probes and on Illumina methylation array (EPIC or 450K). Three types of probes capable of calling genotypes were used, including SNP probes, Type I probes with color-channel-switching (CCS) SNPs at the extension bases, and Type II probes with SNPs at the extension bases. We defined `RAI` as the Ratio of Alternative allele Intensity to total intensity and calculated RAI for each probe and sample.

- **SNP probe:** There are 59 SNP probes (started with "rs") on EPIC array and 65 SNP probes on 450K array. Probes on sex chromosomes (six in EPIC and eight on 450K) were removed. We aligned each probe sequence to reference genome and calculated RAI, which is defined as the proportion of probe signals supporting alternative allele.
- **Type I probe:** We focus on Type I probes with CCS SNPs (A,T <-> C,G mutation) at the extension bases. The signals for probes with CCS SNPs are called out-of-band signals. The RAI is defined as the proportion of out-of-band signals over total signals.
- **Type II probe:** We focus on Type II probes with SNPs at the extension bases (CpG target sites). The alternative allele of SNP can be either A/T (CCS SNP) or G (SNP not switching color channel). Please refer to the manuscript for details in calculating RAI.

The RAI values are usually distributed with three peaks, representing the three genotypes (reference homozygous, heterozygous, and alternative homozygous). To call genotypes from the RAI values, we fit a mixture of three beta distributions for the three genotypes and a uniform distribution for outliers based on the Expectation–maximization (EM) algorithm. Probe-specific weights derived from allele frequencies (AFs) were used in the EM algorithm. A VCF file containing dosage genotype ($\hat{D}_{ij}$), AF ($\hat{q}_i$), and $\hat{R}_i^2$ will be produced:

- $\hat{D}_{ij} = \sum_{k=0}^{2} k\hat{P}_{ijk}$, where $\hat{P}_{ijk}$ is the genotype probability
- $\hat{q}_i = \frac{\sum_{j=1}^{n} \hat{D}_{ij}}{2n}$, where n is the sample size
- $\hat{R}_i^2 = \frac{Var(\hat{D}_{ij})}{2\hat{q}_i(1-\hat{q}_i)}$

For samples of mixed population, we provided an option to infer population structure and calculate individual-specific AFs, which can improve the accuracy of estimating kinship coefficients.

We also provided functions to estimate kinship coefficients and genetic relatedness.

Recommended workflow

Load the MethylGenotyper package

This package has the following dependencies: `minfi`, `tidyverse`, `foreach`, `doParallel`, `HardyWeinberg`, `multimode`, `rlist`, `stats4`, `ggplot2`, `ggpubr`.

```
library(MethylGenotyper)
```

Read IDAT files and perform noob and dye-bias correction

Read IDAT file list. Here is an example of processing three IDAT files from `minfiDataEPIC`. Note that this is just an exemplification of how this tool works. We strongly recommend to use a larger sample size to test the code, such as GSE112179. Your may process your own data by specifying your target file list. Required collumns: Sample_Name, Basename.

```
target <- get_target(platform="EPIC")
head(target)
#>   Sample_Name          Basename
#> 1     sample1  /path/to/sample1
#> 2     sample2  /path/to/sample2
#> 3     sample3  /path/to/sample3
```

With the following code, the IDAT files listed in `target` will be read one-by-one. For each sample, a noob background correction and dye-bias correction will be conducted. You can specify the number of CPUs to enable parallel processing. After that, a list of four elements will be returned, including corrected signals of probe A and probe B for the two color channels.

```
rgData <- correct_noob_dye(target, cpu=3)
```

Call genotypes

The genotype-calling procedure can be done for SNP probes, Type I probes, and Type II probes, separately. You should specify the correct population (One of EAS, AMR, AFR, EUR, SAS, and ALL) for your samples to get accurate genotype calls. If you have samples of mixed population, please specify the population with the largest sample size.

You can plot the distribution of the RAI values and produce a VCF file of the inferred genotypes by specifying `plotBeta=TRUE` and `vcf=TRUE`.

You can also specify cutoffs of R^2, MAF, HWE, and missing rate to filter variants. Note that for VCF output, variants beyond the cutoffs will be marked in the `FILTER` column.

We noted that in the example data, most of variants have R^2=0. This is because we only used three samples here. Again, we strongly recommend to use a larger sample size to test the code, such as GSE112179.

```
# Call genotypes for SNP probes, Type I probes, and Type II probes
genotype_snp <- callGeno_snp(rgData, input="raw", vcf=TRUE, pop="EAS")
genotype_typeI <- callGeno_typeI(rgData, vcf=TRUE, pop="EAS")
genotype_typeII <- callGeno_typeII(rgData, input="raw", vcf=TRUE, pop="EAS")

# Combine genotypes inferred from the three probe types
dosage <- rbind(genotype_snp$dosage, genotype_typeI$dosage, genotype_typeII$dosage)
```

As an alternative option, you can input a matrix of beta values or M values, with each row indicates a probe and each column indicates a sample. This option only works for SNP probes and Type II probes. Here are the examples of calling genotypes from beta values. For input of M values, please specify `input="mval"`. Remember to conduct background correction and dye-bias correction before running the following code. Also be noted that other correction should NOT be conducted, like BMIQ, as it flattens the peaks through a scale transformation.

```
# Call genotypes for SNP probes and Type II probes
genotype_snp <- callGeno_snp(beta_matrix, input="beta", vcf=TRUE, pop="EAS")
genotype_typeII <- callGeno_typeII(beta_matrix, input="beta", vcf=TRUE, pop="EAS")

# Combine genotypes inferred from the two probe types
dosage <- rbind(genotype_snp$dosage, genotype_typeII$dosage)
```

Infer population structure and individual-specific AFs for mixed population

Project the study samples to reference ancestral space: Principal Components Analyses (PCA) are conducted in 1KGP individuals (the ancestral space) and a combination of 1KGP individuals and each study sample. Projection Procrustes analyses are then conducted to project each study sample to reference ancestral space. This step was originally implemented by the TRACE software and we have adapted it in R (Wang et al. Nat Genet 2014, Wang et al. Am J Hum Genet 2015).

```
# PCA and Procrustes analysis, based on genotypes of all probes passing QC
pc <- projection(dosage, plotPCA=TRUE, cpu=3)
```

Estimate individual-specific AFs: For each SNP, we model genotypes of the reference individuals as a linear function of top four PCs ($v.$): $G \sim \beta_0 + \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3 + \beta_4 v_4$. Then, the individual AF (q) for each SNP and each sample can be obtained by: $\hat{q} = \frac{1}{2}(\hat{\beta}_0 + \hat{\beta}_1 \hat{v}_1 + \hat{\beta}_2 \hat{v}_2 + \hat{\beta}_3 \hat{v}_3 + \hat{\beta}_4 \hat{v}_4)$, where $\hat{v}.$ are top four PCs in study samples. (Dou et al. PLoS Genet 2017)

```
data(cpg2snp)
snpvec <- cpg2snp[c(
  rownames(genotype_snp$genotypes$RAI),
  rownames(genotype_typeI$genotypes$RAI),
  rownames(genotype_typeII$genotypes$RAI)
)]
indAF <- get_indAF(snpvec, pc$refPC, pc$studyPC)
```

Estimate sample relationships and sample contamination

With the inferred genotypes, you can estimate sample relationships and sample contamination using the `getKinship` function. The output of this function is a list of two elements: 1) a data frame containing kinship coefficient (ϕ) and sample relationships between each two samples; 2) a vector of inbreeding coefficients, which can be used to infer sample contamination.

Kinship coefficient is calculated according to the SEEKIN software (Dou et al. PLoS Genet 2017):

$$2\phi_{ij} = \frac{\sum_m (G_{im} - 2p_m)(G_{jm} - 2p_m)}{\sum_m 2p_m(1-p_m)(R_m^2)^2}$$

where ϕ_{ij} denotes the kinship coefficient between i-th and j-th sample. G_{im} and G_{jm} denotes genotypes of m-th SNP for i-th and j-th sample. p_m denotes allele frequency of m-th SNP. R^2 is calculated as $R^2 = \frac{Var(D)}{2q(1-q)}$, where D is the dosage genotype. We classified sample pairs as k-degree related if $2^{-k-1.5} < \phi_{ij} < 2^{-k-0.5}$ (Manichaikul et al. Bioinformatics 2010). A zero-degree related pair means monozygotic twins (MZ) or duplicates. Sample pairs more distant than 3rd degree are treated as unrelated.

Inbreeding coefficients are calculated as: $F = 2\phi_{ii} - 1$, where ϕ_{ii} is the self-kinship coefficient for sample i.

```
res <- getKinship(dosage)
kinship <- res$kinship # kinship coefficients
inbreed <- res$inbreed # inbreeding coefficients
```