# 1 Tutorial on Using `R` Code

This tutorial explains how to use the `R` code presented in the paper:

Skelly DA, Johansson M, Madeoy J, Wakefield J, Akey JM. (2011). A powerful and flexible statistical framework for testing hypotheses of allele-specific gene expression from RNA-Seq data.

## 1.1 `R` Scripts

For the analysis you will need two main `R` scripts and one "accessory" `R` script.

1. `DNAmodel.R` - implements the model for read counts derived from genomic DNA, which is used to estimate parameters for the RNA read count model.

2. `RNAmodel.R` - implements the model for read counts derived from cDNA, which is used to identify allele-specific expression.

3. `readGzippedMcmcOutput.R` - script that can be `source`d from within an `R` session and is used to read the saved output of scripts 1 and 2.

## 1.2 Input format

The read counts that must be fed into `DNAmodel.R` and `RNAmodel.R` should consist of read counts at each SNP, not aggregated by gene. The file must be in the following tab-delimited format:

| | | | |
|---|---|---|---|
| # comments | | | |
| header (ignored) | | | |
| geneNameColumn | snpIndexColumn | alleleOneCount | alleleTwoCount |
| ⋮ | ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ | ⋮ |

where `geneNameColumn` names the first column of genes, `snpIndexColumn` is a column used to give a unique name/numerical value to each SNP (not currently used in the scripts), and `alleleOneCount` and `alleleTwoCount` are just the read counts themselves.

## 1.3 Basic Usage

The scripts are designed to be used from the command-line in a unix-like environment, but could easily be adapted for Windows usage. `DNAmodel.R` and `RNAmodel.R` have command-line flags that give the basic details for usage. The arguments for `DNAmodel.R` are:

1. name of infile - genomic DNA read counts
2. name of outfile - the file in which to store results of MCMC. File will be gzipped.
3. number of iterations of MCMC to run
4. value of "thin" - 1 out of every thin iterations, the parameter values will be written to your outfile
5. number of scaling iterations - the proposal distributions can be scaled to achieve an acceptance rate of (by default) approximately 30%. The tuning will happen after blocks of MCMC of this number of iterations. This number does not count towards the number of MCMC iterations specified above, and the parameter values during the scaling iterations are not saved.

For details on the arguments to `RNAmodel.R`, run the program with no arguments (`./RNAmodel.R`). Your options are similar to those for `DNAmodel.R`, but can be specified in a slighly more flexible manner. The general look of a command to run this program is

```
./RNAmodel.R [options] dataset1 [dataset2 dataset3 ...]  outfile
```

where the number of input datasets can vary but must be at least one. As stated above, the script `readGzippedMcmcOutput.R` is designed to be `source`d from within an `R` session and is used to read the saved output of scripts 1 and 2.

## 1.4 Example Usage

Let's say we have collected one lane of genomic DNA read counts and four lanes of RNA-Seq data using the Illumina sequencing platform, and have calculated allele-specific read counts for each sample and collected them in the files `DNA.txt`, `RNA1.txt`, `RNA2.txt`, `RNA3.txt`, and `RNA4.txt`. A simple analysis of this data might be:

1. First, run an analysis of your genomic DNA data:
   ```
   ./DNAmodel.R DNA.txt DNAresults.gz 200000 40 2000
   ```

2. Next, analyze the results of step 1 and obtain estimates of $\hat{a}$ and $\hat{d}$. *Note the warning below about checking for convergence.* Inside an `R` session, you could execute the commands:
   ```
   > source(‘readGzippedMcmcOutput.R’)
   > result <- read.mcmc(‘DNAresults.gz’)
   > n.iter <- result$n.iter/result$thin
   > burnin <- 0.1*n.iter
   > a.hat <- median(exp(result$mcmc$logA[burnin:n.iter]))
   > d.hat <- median(exp(result$mcmc$logD[burnin:n.iter]))
   ```

3. Run an analysis on your RNA data:
   ```
   ./RNAmodel.R --n.iter=500000 --thin=100 --n.scaling.iter=2000
   --a.hat=2000 --d.hat=500 --max.rounds.of.scaling=8 RNA1.txt
   RNA2.txt RNA3.txt RNA4.txt RNAresults.gz
   ```
   *Note the warning below about checking for convergence.*

4. Examine the results of the RNA run in `R`:
   ```
   > source(‘readGzippedMcmcOutput.R’)
   > result <- read.mcmc(‘RNAresults.gz’)
   ```

## 1.5 Warning!

MCMC can be dangerous! Make sure you understand something about MCMC before attempting to run these scripts. You should always run multiple chains from different starting parameters, and verify convergence by examining time series plots or by other more formal measures such as the convergence diagnostics proposed by Gelman and Rubin (1992), Geweke (1992), or Heidelberger and Welch (1983). These scripts do nothing to check that you have completed any of these tasks.