Prediction and expression profiling of novel miRNAs in Turkeys

Tom Kono (konox006@umn.edu)

2023-02-13

Contents

1 Input Data

1.1 Sequence Reads

The input RNA for this experiment were small RNAs isolated from turkey muscle cells from two genotypes. The extracts were prepared into libraries with the Takara smRNA kit and sequenced on a NovaSeq 6000 with a 2x50bp output mode. A description of the input samples is given below:

Sample Name	Genotype	N. Fragments
NCT_48_38_33_101_smRNA	NCT	21,728,024
NCT_48_38_33_102_smRNA	NCT	21,552,098
NCT_48_38_38_61_smRNA	NCT	20,274,703
NCT_48_38_38_62_smRNA	NCT	20,879,610
NCT_48_38_43_51_smRNA	NCT	30,452,556
NCT_48_38_43_52_smRNA	NCT	16,756,472
NCT_72_33_32_smRNA	NCT	16,185,368
NCT_72_33_53_smRNA	NCT	18,663,090
NCT_72_38_71_smRNA	NCT	18,513,867
NCT_72_38_73_smRNA	NCT	15,089,576
NCT_72_43_21_smRNA	NCT	19,911,181
NCT_72_43_22_smRNA	NCT	20,772,409
RBC2_48_38_33_101_smRNA	RBC2	16,405,230
RBC2_48_38_33_102_smRNA	RBC2	23,417,907
RBC2_48_38_38_12_smRNA	RBC2	16,314,326
RBC2_48_38_38_14_smRNA	RBC2	22,244,407
RBC2_48_38_43_22_smRNA	RBC2	21,406,067
RBC2_48_38_43_24_smRNA	RBC2	15,893,508
RBC2_72_38_11_smRNA	RBC2	15,555,080
RBC2_72_38_12_smRNA	RBC2	18,487,011
RBC2_72_43_41_smRNA	RBC2	19,835,488
RBC2_72_43_42_smRNA	RBC2	19,131,708

1.2 External Data

1.2.1 Known miRNAs

Known miRNAs were downloaded from miRBase release 22.1. Mature miRNA sequences and miRNA hairpin sequences from chicken (*Gallus gallus*, miRBase species code gga) were extracted from the miRNA sets and used as input for the novel miRNA prediction (see below).

1.2.2 Reference Genome

The reference genome used for this analysis was downloaded from NCBI (accession code GCA_943295565.1).

2 Methods

2.1 Novel miRNA Prediction

2.1.1 Read Cleaning

The "forward" (R1) reads of each library were cleaned of adaptor contaminants following the instructions from the library preparation kit vendor. The cleaning was performed with `cutadapt` 4.2 with the following options

```
-m 15 -u 3 -a AAAAAAAAAA
```

to discard reads that are shorter than 15nt after trimming, remove the first three nt (which are non-biological), and trim the poly-A adaptor sequence. The trimming is implemented in the `Scripts/Cutadapt.sh` script.

The reads were then screened for rRNA contamination based on K-mer matches to a set of reference rRNA sequences. The reference rRNA sequences were the LSU and SSU sequence sets downloaded from the SILVA database release 132. The K-mer matching was performed with `BBDuk` version 39.01 with the following options

```
k=15 minlength=15 editdistance=0
```

to remove reads with exact matches of at least 15nt to a reference rRNA sequence. The rRNA removal is implemented in the `Scripts/rRNA_Depletion.sh` script.

For novel miRNA prediction, the cleaned and rRNA-depleted reads were combined into a single file. This is implemented in the `Scripts/Combine_miRNA_Reads.sh` script.

2.1.2 miRDeep2

2.1.2.1 Installation

miRDeep2 and its dependencies were installed into a Conda environment with the following command:

```
conda create \
    -p ./miRDeep2_env \
    --copy -y \
    -c bioconda -c conda-forge -c defaults \
    mirdeep2 bowtie viennarna squid randfold perl-pdf-api2
```

2.1.2.2 Preparing Reference Genome and miRBase miRNAs

miRDeep2 does not allow whitespace in the sequence names for the reference genome nor the miRBase reference miRNAs. The whitespace was removed with the `awk` program:

```
awk '{print $1}' genome.fa > genome_miRDeep2_Names.fa
awk '{print $1}' mature_gga.fa > mature_gga_miRDeep2_Names.fa
awk '{print $1}' hairpin_gga.fa > hairpin_gga_miRDeep2_Names.fa
```

Then, the reference genome was prepared for mapping with bowtie:

```
bowtie-build -p 12 -f genome.fa ./bowtie_idx/bt_idx
```

2.1.3 Running the miRNA Prediction Algorithm

The miRNA prediction algorithm for miRDeep2 was run in two steps. The first step involves mapping the sequence reads against the genome with bowtie and producing a collapsed reads file and an alignment file. The second step involves the core miRDeep2 algorithm, which includes comparing known miRNA sequences to genomic regions covered by the sequence reads, identifying secondary structure, and assigning confidence scores to each potential miRNA.

These steps are implemented in the `Scripts/Run_miRDeep2.sh` script.

2.2 miRNA Expression Profiling

miRNA expression profiling was performed with the reads that were cleaned of adaptor sequences and rRNA contaminants, as described above. However, the expression profiling was performed with the reads from each sample maintained as separate FASTQ files, rather than the pooled file that was used for novel miRNA prediction.

2.2.1 Read Mapping

Cleaned and rRNA-depleted reads from each sample were mapped to the reference genome with bowtie, using the same parameters as used for miRNA discovery. The script that implements the mapping and conversion of SAM to BAM is available as `Scripts/Bowtie_Mapping.sh`.

2.2.2 Expression Quantification

The expression of miRNAs was quantified with the `featureCounts` program from the `subread` package. `featureCounts` requires an annotation file, either in "SAF" or "GTF" format to quantify expression. The BED file of mature miRNA loci on the genome assembly was converted to SAF with a custom Python script:

```
python Scripts/miRNA_BED_to_SAF.py result_20_12_2022_t_11_07_40.bed \
    | sort -k2V -k3,4n \
    > Turkey_miRNAs.saf
```

Next, the final BAM files were symlinked into a separate directory to ease the parsing of the `featureCounts` output file. The symlinking is implemented in the `Scripts/Link_BAM.sh` script.

Then, the matrix of read counts was generated with `featureCounts`. Only reads with a MAPQ value of 10 or greater were used for miRNA expression quantification. The script to run `featureCounts` is available as `Scripts/featureCounts_miRNA_NewGenome.sh`.

2.2.3 Sample Metadata

Samples were analyzed with the following metadata labels:

Sample Name	Genotype	Temperature	Time
NCT_48_38_33_101_smRNA	NCT	33	48
NCT_48_38_33_102_smRNA	NCT	33	48
NCT_48_38_38_61_smRNA	NCT	38	48
NCT_48_38_38_62_smRNA	NCT	38	48
NCT_48_38_43_51_smRNA	NCT	43	48
NCT_48_38_43_52_smRNA	NCT	43	48
NCT_72_33_32_smRNA	NCT	33	72
NCT_72_33_53_smRNA	NCT	33	72
NCT_72_38_71_smRNA	NCT	38	72
NCT_72_38_73_smRNA	NCT	38	72
NCT_72_43_21_smRNA	NCT	43	72
NCT_72_43_22_smRNA	NCT	43	72
RBC2_48_38_33_101_smRNA	RBC2	33	48
RBC2_48_38_33_102_smRNA	RBC2	33	48
RBC2_48_38_38_12_smRNA	RBC2	38	48
RBC2_48_38_38_14_smRNA	RBC2	38	48
RBC2_48_38_43_22_smRNA	RBC2	43	48
RBC2_48_38_43_24_smRNA	RBC2	43	48
RBC2_72_38_11_smRNA	RBC2	38	72
RBC2_72_38_12_smRNA	RBC2	38	72
RBC2_72_43_41_smRNA	RBC2	43	72
RBC2_72_43_42_smRNA	RBC2	43	72

For the expression analyses, the `Time=48` samples and the `Time=72` samples were separated and analyzed as two different experiments.

2.2.4 Global Expression Analysis

The following analyses are implemented in two scripts, `Scripts/miRNA_edgeR.R` and `Scripts/miRNA_VarPart.R`. Snippets of the R code are reproduced here for illustrative purposes only.

2.2.4.1 Filtering

The matrix of raw read counts was analyzed with the `edgeR` package in the R statistical computing environment. First, the miRNAs with low expression were removed from the analysis. To be retained, a miRNA had to have a log2(CPM) value of at least C in at least N samples, where C is the log2(CPM) that corresponds to 3 reads in the smallest library, and N is the size of the smallest experimental group. In R code:

```
min_cts <- 3
smallest_grp <- min(table(exp_meta_48$Group))
min_cpm <- log2((1 + min_cts) / min(dge_dat_48$samples$lib.size) * 1e6)
# Then if a gene has at least N samples with CPM greater than or equal to this
# CPM value, we keep it. N is the size of the smallest group.
cpms <- cpm(dge_dat_48, log=TRUE, prior.count=1)
keep <- apply(cpms, 1, function(x) {
    cvals <- as.numeric(x)
    pass_flt <- sum(cvals ≥ min_cpm)
    if(pass_flt ≥ smallest_grp) {
        return(TRUE)
```

```
    } else {
        return(FALSE)
    }
})
```

Library sizes and normalization factors were then recalculated after filtering miRNAs with low expression. The biological coefficient of variation (BCV) was assessed with the figures generated from the `plotBCV()` function.

2.2.4.2 Principal Components Analysis

Normalized log2(CPM) values from the filtered miRNA expression matrix were used for a principal components analysis with the `prcomp()` function in R. The values were centered and scaled, and the first two principal components were plotted to visualize global patterns of variation among the samples.

2.2.4.3 Variance Partitioning

The matrix of raw counts was used to estimate the variance in miRNA expression that could be explained by the effects of `Genotype` and `Temperature` within each of the `Time=48` and `Time=72` experiments. This analysis used the `variancePartition` package for R. Normalized expression values and the mean-variance relationship of normalized expression were calculated with the `voom()` function from the `voom` package. Then, a linear model was used to estimate the variance explained by the experimental factors. In R code:

```
exp_meta_48$Incubation.Temp <- factor(
    exp_meta_48$Incubation.Temp,
    levels=c("33", "38", "43"))
modmat_48 <- model.matrix(
    ~Incubation.Temp + Genotype + Incubation.Temp:Genotype,
    data=exp_meta_48)
voom_48 <- voom(dge_dat_48, modmat_48)
vp_mod_48 <- ~(1|Incubation.Temp) + (1|Genotype) + (1|Incubation.Temp:Genotype)
vp_fit_48 <- fitExtractVarPartModel(voom_48, vp_mod_48, exp_meta_48)
```

2.2.5 Differential Expression Analysis

The matrix of raw counts was used to identify differentially expressed miRNAs in both the `Time=48h` and `Time=72h` experiments. miRNAs with low expression were filtered as described above. Differential expression was tested across incubation temperatures, accounting for the variation due to genotype using the quasi-likelihood F test (`glmQLFTest()`) in `edgeR`. In R code:

```
modmat_48 <- model.matrix(
    ~0 + as.factor(Incubation.Temp) + Genotype,
    data=exp_meta_48)
colnames(modmat_48) <- c("Temp33", "Temp38", "Temp43", "GenotypeRBC2")
dge_dat_48 <- estimateDisp(dge_dat_48, design=modmat_48)
dge_48_fit <- glmQLFit(dge_dat_48, design=modmat_48)
contrasts_48 <- makeContrasts(
    Temp38_vs_Temp33=Temp38-Temp33,
    Temp43_vs_Temp38=Temp43-Temp38,
    Temp43_vs_Temp33=Temp43-Temp33,
    levels=modmat_48)
dge_48_38vs33 <- glmQLFTest(
    dge_48_fit, contrast=contrasts_48[,"Temp38_vs_Temp33"])
dge_48_43vs38 <- glmQLFTest(
```

```
    dge_48_fit, contrast=contrasts_48[,"Temp43_vs_Temp38"])
dge_48_43vs33 <- glmQLFTest(
    dge_48_fit, contrast=contrasts_48[,"Temp43_vs_Temp33"])
```

Specific comparisons were also performed within-genotype and between incubation temperatures. These tests were performed with the same functions, but with a subset of the expression matrix that only included the samples being compared.

False discovery rate (FDR) control was performed with the Benjamini-Hochberg procedure.

2.3 Software Environment

2.3.1 HPC Environment

The following software modules were loaded on MSI:

```
module load python3/3.8.3_anaconda2020.07_mamba
module load java/openjdk-17.0.2
module load samtools/1.14
module load parallel/20210822
```

2.3.1.1 Conda Environments

These are the Conda environments that were used for some of the software used in this project. Specification files for reproducing these environments are available in the GitHub repository.

Cutadapt:

```
# packages in environment at /home/riss/konox006/conda_envs/cutadapt_env:
#
# Name                    Version                   Build  Channel
_libgcc_mutex             0.1                 conda_forge    conda-forge
_openmp_mutex             4.5                       2_gnu    conda-forge
bzip2                     1.0.8                h7f98852_4    conda-forge
ca-certificates           2022.12.7            ha878542_0    conda-forge
cffi                      1.15.1                   pypi_0    pypi
cutadapt                  4.2                      pypi_0    pypi
dnaio                     0.10.0                   pypi_0    pypi
isa-l                     2.30.0               ha770c72_4    conda-forge
isal                      1.1.0                    pypi_0    pypi
ld_impl_linux-64          2.39                 hcc3a1bd_1    conda-forge
libffi                    3.4.2                h7f98852_5    conda-forge
libgcc-ng                 12.2.0              h65d4601_19    conda-forge
libgomp                   12.2.0              h65d4601_19    conda-forge
libnsl                    2.0.0                h7f98852_0    conda-forge
libsqlite                 3.40.0               h753d276_0    conda-forge
libstdcxx-ng              12.2.0              h46fd767_19    conda-forge
libuuid                   2.32.1            h7f98852_1000    conda-forge
libzlib                   1.2.13               h166bdaf_4    conda-forge
ncurses                   6.3                  h27087fc_1    conda-forge
openssl                   3.0.7                h0b41bf4_1    conda-forge
pbzip2                    1.1.13                        0    conda-forge
pigz                      2.6                  h27826a3_0    conda-forge
pip                       22.3.1             pyhd8ed1ab_0    conda-forge
pycparser                 2.21               pyhd8ed1ab_0    conda-forge
```

```
python                    3.10.8          h4a9ceb5_0_cpython    conda-forge
python-isal               1.1.0           py310h5764c6d_1    conda-forge
python_abi                3.10                    3_cp310    conda-forge
readline                  8.1.2                h0f457ee_0    conda-forge
setuptools                65.5.1             pyhd8ed1ab_0    conda-forge
tk                        8.6.12               h27826a3_0    conda-forge
tzdata                    2022g                h191b570_0    conda-forge
wheel                     0.38.4             pyhd8ed1ab_0    conda-forge
xopen                     1.7.0                    pypi_0    pypi
xz                        5.2.6                h166bdaf_0    conda-forge
zlib                      1.2.13               h166bdaf_4    conda-forge
zstandard                 0.19.0                   pypi_0    pypi
zstd                      1.5.2                h6239696_4    conda-forge
```

miRDeep2:

```
# packages in environment at /home/riss/konox006/conda_envs/miRDeep2_env:
#
# Name                    Version                   Build  Channel
_libgcc_mutex             0.1                 conda_forge    conda-forge
_openmp_mutex             4.5                       2_gnu    conda-forge
bowtie                    1.3.1           py310h4070885_4    bioconda
bzip2                     1.0.8                h7f98852_4    conda-forge
ca-certificates           2022.12.7            ha878542_0    conda-forge
expat                     2.5.0                h27087fc_0    conda-forge
glpk                      5.0                  h445213a_0    conda-forge
gmp                       6.2.1                h58526e2_0    conda-forge
icu                       70.1                 h27087fc_0    conda-forge
ld_impl_linux-64          2.39                 hcc3a1bd_1    conda-forge
libffi                    3.4.2                h7f98852_5    conda-forge
libgcc-ng                 12.2.0              h65d4601_19    conda-forge
libgomp                   12.2.0              h65d4601_19    conda-forge
libhwloc                  2.8.0                h32351e8_1    conda-forge
libiconv                  1.17                 h166bdaf_0    conda-forge
libnsl                    2.0.0                h7f98852_0    conda-forge
libsqlite                 3.40.0               h753d276_0    conda-forge
libstdcxx-ng              12.2.0              h46fd767_19    conda-forge
libuuid                   2.32.1            h7f98852_1000    conda-forge
libxml2                   2.10.3               h7463322_0    conda-forge
libzlib                   1.2.13               h166bdaf_4    conda-forge
mirdeep2                  2.0.0.8                     0    bioconda
ncurses                   6.3                  h27087fc_1    conda-forge
openssl                   3.0.7                h0b41bf4_1    conda-forge
perl                      5.32.1         2_h7f98852_perl5    conda-forge
perl-font-ttf             1.06            pl5321hdfd78af_1    bioconda
perl-io-string            1.08            pl5321hdfd78af_4    bioconda
perl-pdf-api2             2.043           pl5321hdfd78af_0    bioconda
perl-threaded             5.32.1                hdfd78af_1    bioconda
perl-xml-parser           2.44_01     pl5321hc3e0081_1003    conda-forge
pip                       22.3.1             pyhd8ed1ab_0    conda-forge
python                    3.10.8       h4a9ceb5_0_cpython    conda-forge
python_abi                3.10                    3_cp310    conda-forge
randfold                  2.0.1                hec16e2b_4    bioconda
readline                  8.1.2                h0f457ee_0    conda-forge
setuptools                65.5.1             pyhd8ed1ab_0    conda-forge
squid                     1.5                  h82967d4_6    bioconda
tbb                       2021.7.0             h924138e_1    conda-forge
tk                        8.6.12               h27826a3_0    conda-forge
```

```
tzdata                    2022g                h191b570_0    conda-forge
viennarna                 2.5.1     py310pl5321hc8f18ef_0    bioconda
wheel                     0.38.4             pyhd8ed1ab_0    conda-forge
xz                        5.2.6                h166bdaf_0    conda-forge
zlib                      1.2.13               h166bdaf_4    conda-forge
```

2.3.2 R Environment

The following R environment was used for expression analyses. A `renv` lock file for reproducing this environment is available in the GitHub repository for this project.

```
R version 4.2.2 (2022-10-31)
Platform: aarch64-apple-darwin21.6.0 (64-bit)
Running under: macOS Ventura 13.2

Matrix products: default
BLAS:   /opt/homebrew/Cellar/openblas/0.3.21/lib/libopenblasp-r0.3.21.dylib
LAPACK: /opt/homebrew/Cellar/r/4.2.2_1/lib/R/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices datasets  utils     methods   base

other attached packages:
[1] variancePartition_1.28.1 BiocParallel_1.32.5      ggplot2_3.4.0
[4] edgeR_3.40.1             limma_3.54.0

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.9              locfit_1.5-9.7          mvtnorm_1.1-3
 [4] lattice_0.20-45         tidyr_1.2.1             prettyunits_1.1.1
 [7] gtools_3.9.4            RhpcBLASctl_0.21-247.1  foreach_1.5.2
[10] utf8_1.2.2              plyr_1.8.8              R6_2.5.1
[13] backports_1.4.1         pillar_1.8.1            Rdpack_2.4
[16] gplots_3.1.3            rlang_1.0.6             progress_1.2.2
[19] minqa_1.2.5             nloptr_2.0.3            RUnit_0.4.32
[22] Matrix_1.5-3            splines_4.2.2           lme4_1.1-31
[25] stringr_1.5.0           remaCor_0.0.11          munsell_0.5.0
[28] broom_1.0.2             compiler_4.2.2          pkgconfig_2.0.3
[31] BiocGenerics_0.44.0     tidyselect_1.2.0        tibble_3.1.8
[34] codetools_0.2-18        fansi_1.0.3             crayon_1.5.2
[37] dplyr_1.0.10            withr_2.5.0             MASS_7.3-58.1
[40] bitops_1.0-7            rbibutils_2.2.13        grid_4.2.2
[43] nlme_3.1-161            gtable_0.3.1            lifecycle_1.0.3
[46] magrittr_2.0.3          scales_1.2.1            KernSmooth_2.23-20
[49] stringi_1.7.12          cli_3.6.0               reshape2_1.4.4
[52] renv_0.16.0             doParallel_1.0.17       ellipsis_0.3.2
[55] generics_0.1.3          vctrs_0.5.1             boot_1.3-28.1
[58] aod_1.3.2               iterators_1.0.14        tools_4.2.2
[61] Biobase_2.58.0          glue_1.6.2              purrr_1.0.1
[64] hms_1.1.2               parallel_4.2.2          pbkrtest_0.5.1
[67] colorspace_2.0-3        caTools_1.18.2          clusterGeneration_1.3.7
```