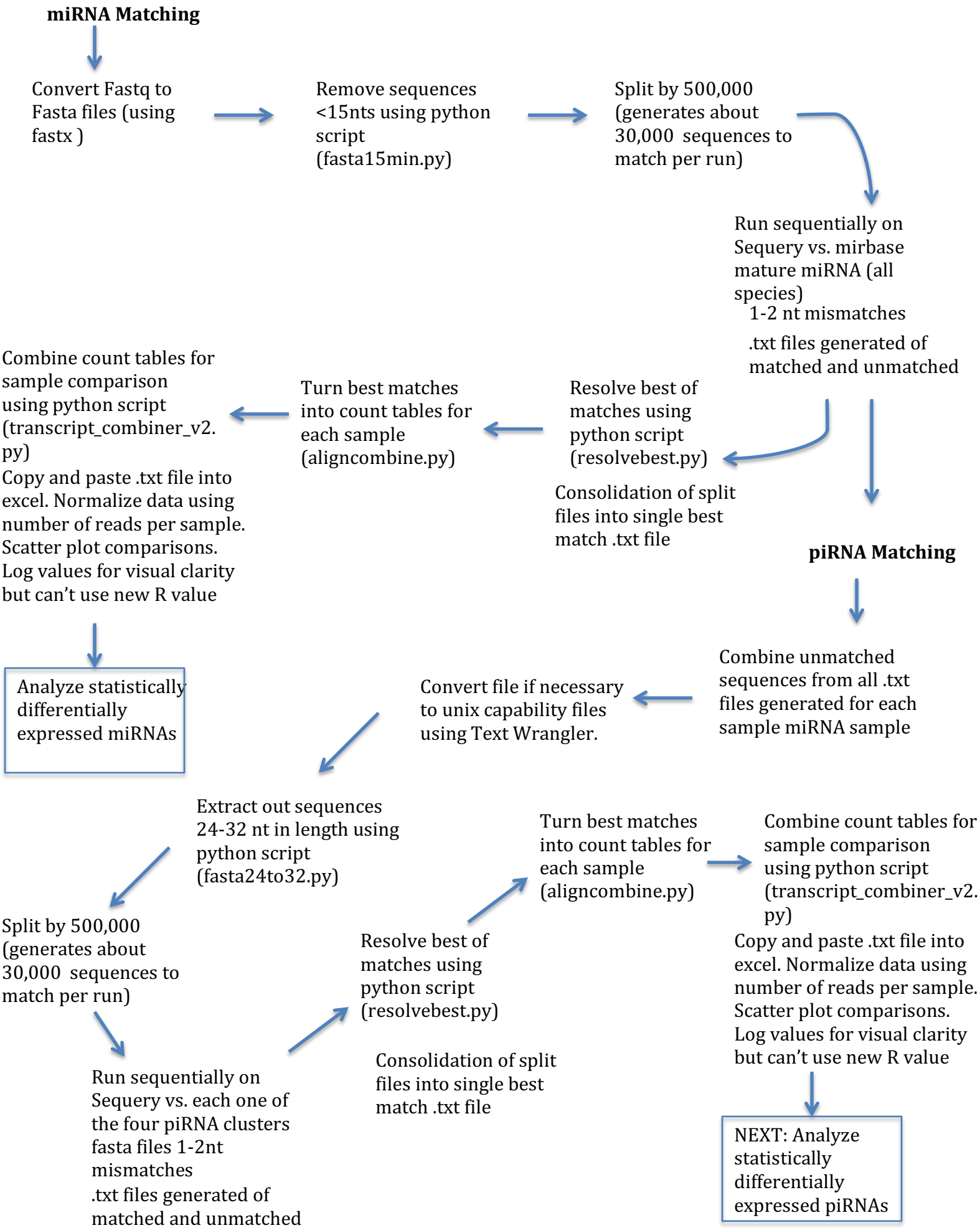
miRNA Matching
Convert Fastq to Fasta files (using fastx)
Remove sequences <15nts using python script (fasta15min.py)
Split by 500,000 (generates about 30,000 sequences to match per run)
Run sequentially on Sequery vs. mirbase mature miRNA (all species)
1-2 nt mismatches
.txt files generated of matched and unmatched
Resolve best of matches using python script (resolvebest.py)
Consolidation of split files into single best match .txt file
Turn best matches into count tables for each sample (aligncombine.py)
Combine count tables for sample comparison using python script (transcript_combiner_v2.py)
Copy and paste .txt file into excel. Normalize data using number of reads per sample.
Scatter plot comparisons.
Log values for visual clarity but can't use new R value
Analyze statistically differentially expressed miRNAs
piRNA Matching
Combine unmatched sequences from all .txt files generated for each sample miRNA sample
Convert file if necessary to unix capability files using Text Wrangler.
Extract out sequences 24-32 nt in length using python script (fasta24to32.py)
Split by 500,000 (generates about 30,000 sequences to match per run)
Run sequentially on Sequery vs. each one of the four piRNA clusters fasta files 1-2nt mismatches
.txt files generated of matched and unmatched
Resolve best of matches using python script (resolvebest.py)
Consolidation of split files into single best match .txt file
Turn best matches into count tables for each sample (aligncombine.py)
Combine count tables for sample comparison using python script (transcript_combiner_v2.py)
Copy and paste .txt file into excel. Normalize data using number of reads per sample.
Scatter plot comparisons.
Log values for visual clarity but can't use new R value
NEXT: Analyze statistically differentially expressed piRNAs

miRNA Analysis

Prior to analysis, the following programs need to be download and installed on computer:

Sequery (Yan Lab website)
GCC4.2.1
Python 2.7.5
FASTX toolkit (Hannon Lab)
Text Wrangler

File preparation for miRNA Sequery matching

1. Convert Fastq files to Fasta files.

Open terminal:

```
fastq_to_fasta -Q 33 -i desktop/DZ1_28_14-1606/miRNA01.fastq
-o miRNA01.fa
```

2. Search for new .fa file and drag into the folder
3. Change directory (into the folder that has the script and files) to eliminate sequences under 15nt

```
cd desktop/DZ1_28_14-1606file name/

python fasta15min.py miRNA01.fa miRNA01_15min.fa
```

4. Split the files into 250,000 read files due to Sequery limitations

```
split -l 1000000 miRNA01_15min.fa
```

5. Files will begin to appear in the folder that being with “xaa”, “xab”, “xac” etc. Rename these files into subsets for each sample (e.g. miRNA01_15min1.fa, miRNA01_15min2.fa, etc)

miRNA Sequery matching

Each subset .fa file should be matched to the mirbase20_mature.fa

1. open sequence analyzer execution window (Sequery should automatically appear)
2. R click to open folders and L click to select the sequence file
3. Once the sequence file has been uploaded on the Sequery window, R click to open options window, R click on Load/Save and R click on Load File
4. L click/select the mirbase20_mature.fa file, this should bring you back to the Sequery screen

5. On the top of the screen, L click on the red "T" for the mirbase20_mature.fa file to make it a reference file (becomes a red "R")
6. R click on the screen, R click on compare, L click on second option (3M T-R 1st Sub1+=2)
7. In the terminal window, you should the program running and matching the sequences (this will take 30min-1hr)
8. When the program is finished, R click on Sequery screen, R click on Output, L click on first option (3M Tests in Ref + U)
9. This will bring you back to the gray screen with the cursor blinking on the file name.
10. Rename file and hit enter. This will save your file to the desktop. You will see the tests in the terminal window (this will take 1-2hrs).
11. Once the tests are finished, you can just exit out of the program.

Post-match .txt files analysis

1. Resolve the best of matches and combine subsets of each sample. Make sure python scripts are in the folder with your files

```
cd desktop/folder name

python resolvebest.py -i ALL your files for each sample
miRNA01_15min1.txt miRNA01_15min2.txt miRNA01_15min3.txt
miRNA01_15min4.txt miRNA01_15min5.txt miRNA01_15min6.txt -o
miRNA01_best.txt
```

2. Create a transcript count table for each sequence sample

```
python aligncombine.py -i1 miRNA01_best.txt –o
miRNA01_count.txt
```

3. Combine and compare transcript count tables from each replicate or sample

```
python transcript_combiner_v2.py miRNA01_count.txt
miRNA02_count.txt etc. miRNAtotalcount.txt
```

4. Open comp.txt, select all, and copy and paste into excel. Normalize each specific count number with the correct number of total reads for each sample. (e.g. (2/3,476,397)*1000000).

5. Generate scatter plot, add trendline, r-value, and equation.

piRNA Analysis

File preparation for piRNA Sequery matching

1. Retrieve unaligned sequences from miRNA Sequery .txt files
    - Open .txt file
    - "Find" >
    - Scroll all the way down to the end of the file, Shift, and click on the last sequence to select all from the first ">" to the last ">" (these are the unaligned sequences)
    - Copy and paste into new text edit file.
    - Repeat with the rest of the subset files
    - Save as unmatched sequences (e.g. miRNA01_unmat.txt)

2. Extract out sequences from 24-32 nts

    open terminal, go into the directory with your unmatched files and python script

    ```
    cd desktop/unmatfolder folder name

    python fasta24to32.py miRNA01_unmat.txt miRNA01_24-32.fa
    ```

3. If you generate a file that is not unix executable (i.e. a document) then use this code to change the document type

    ```
    chmod +x
    ```

    Space after the x and then drag and drop the document file into the terminal, this should provide it with the correct path.
    Hit enter, this should change the icon of your file

4. Again, break down files into smaller subsets for Sequery compatibility

    ```
    split -l 1000000 miRNA01_24-32.fa
    ```

piRNA Sequery matching

Each subset .fa file should be matched to each one of the four (4) master cluster's file

1. open sequence analyzer execution window (Sequery should automatically appear)
2. R click to open folders and L click to select the sequence file
3. Once the sequence file has been uploaded on the Sequery window, R click to open options window, R click on Load/Save and R click on Load File
4. L click/select the master cluster1, 2, 3 or 4, this should bring you back to the Sequery screen

5. On the top of the screen, L click on the red "T" for the master cluster file to make it a reference file (becomes a red "R")
6. R click on the screen, R click on compare, L click on first option (3M T-R 1st Exact=0)
7. In the terminal window, you should the program running and matching the sequences (this will take >2 hours)
8. When the program is finished, R click on Sequery screen, R click on Output, L click on first option (3M Tests in Ref + U)
9. This will bring you back to the gray screen with the cursor blinking on the file name.
10. Rename file and hit enter. This will save your file to the desktop. You will see the tests in the terminal window (this will take 1-2hrs).
11. Once the tests are finished, you can just exit out of the program.

Classic edgeR analysis of significant differentially expressed miRNA

R and edge R must both be downloaded prior to loading in the file

1. Open R and enter:

```
source("http://bioconductor.org/biocLite.R")
 biocLite("edgeR")
```

enter

```
library("edgeR")
```

to load the package

2. Follow the commands in order to determine significant differential expression: make sure you're in the correct working directory (Change by going to Misc in header)

```
x <- read.delim("fileofcounts.txt",row.names="id")
group <- factor(c(2,1,2,1…))if you're comparing H to C
d <- DGEList(counts=x,group=group)
d <- estimateCommonDisp(d)
d <- estimateTagwiseDisp(d)
et <- exactTest(d)
topTags(et, n=# for however many you want to show)
```

3. To show you a summary of how many sequences are significantly expressed use:

```
summary(de <- decideTestsDGE(et, p=0.05, adjust="BH"))
```

-1 denotes downregulated tags, 0 indicates non-differentially expressed, and

1 indicates upregulated tags

4. To generate a heatmap

```
y <- cpm(d, prior.count=2, log=TRUE)
heatmap(y)
```

General Linear Model edgeR analysis of significant differentially expressed miRNA- Family blocking

Load edgeR library

```
library(edgeR)
```

Make sure all the columns for each sample is changed so that it begins with a number (i.e. HT4F06H-> 6H)

1. Enter read counts .txt (making sure you're in the correct directory)

   ```
   rawdata<-read.delim("name.txt", check.names=FALSE,
   stringsAsFactors=FALSE)
   ```

2. Put data into a DGEList object. The numbers in the brackets dictate which column are the counts and which is the IDs

   ```
   y <- DGEList(counts=rawdata[,2:11], genes=rawdata[,1])
   ```

3. Design the matrix

   ```
   Family<-factor(c(6,6,9,9,11,11,etc.))
   Treatment<-
   factor(c("C","H","C","H","C","H","C","H","C","H"))
   data.frame(Sample=colnames(y),Family,Treatment)
   ```

   check to see if the sample IDs match with the matrix you've designed

   ```
   design<-model.matrix(~Family+Treatment)
   rownames(design)<-colnames(y)
   ```

4. Estimate the dispersion

   ```
   y<-estimateGLMTrendedDisp(y,design)
   y<-estimateGLMTagwiseDisp(y,design)
   ```

5. Differential expression

   ```
   fit<-glmFit(y,design)
   lrt<-glmLRT(fit)
   ```

```
topTags(lrt)
```

6. Summary of DE miRNAs

```
summary(de<-decideTestsDGE(lrt))
```

7. Smearplot

```
detags<-rownames(y)[as.logical(de)]
plotSmear(lrt,de.tags=detags)
abline(h=c(-1, 1),col="blue")
```

8. Genewise tests adjusting for baseline differences between families, counts-per-million in individual samples top genes:

```
o<-order(lrt$table$PValue)
cpm(y)[o[1:10],]
```

General Linear Model edgeR analysis of significant differentially expressed miRNA: Interaction between treatment and haplotype (Family is blocked)

1. Load edgeR library

```
library(edgeR)
```

2. Enter read counts .txt (making sure you're in the correct directory)

```
rawdata<-read.delim("name.txt", check.names=FALSE,
stringsAsFactors=FALSE)
```

3. Design the matrix

```
Family<-factor(c(6,6,9,9,11,11,etc.))
Treatment<factor(c("C","H","C","H","C","H","C","H","C","H"))
MitoHap<-factor(c("S","S","S","S","S","S","D","D","D","D"))
data.frame(Sample=colnames(y),Family,Treatment,MitoHap)
```

check to see if the sample IDs match with the matrix you've designed

```
design1 <- model.matrix(~Family)
design2 <- model.matrix(~MitoHap*Treatment)
design <- cbind(design1,design2[,3:4])
rownames(design)<-colnames(y)
```

4. Estimate the dispersion

```
y<-estimateGLMTrendedDisp(y,design)
y<-estimateGLMTagwiseDisp(y,design)
```

5. Differential expression

```
fit<-glmFit(y,design)
lrt<-glmLRT(fit,coef=7)
topTags(lrt)
```

6. Summary of DE miRNAs

```
summary(de<-decideTestsDGE(lrt))
```

7. Smearplot

```
detags<-rownames(y)[as.logical(de)]
plotSmear(lrt,de.tags=detags)
abline(h=c(-1, 1),col="blue")
```

8. Genewise tests adjusting for baseline differences between families, counts-per-million in individual samples top genes:

```
o<-order(lrt$table$PValue)
cpm(y)[o[1:10],]
```