find\_variants\_ben

# Purpose:

Ben has been studying a child with a trifunctional protein deficiency and would like to know if there are any other variants near the HADHB allele.

* Search all variants found in family trio
* For genes of interest in the region
* Homozygous or compound heterozygous in the newborn
* Check for Mammalian Inheritance Errors (MIE)
* Add Kaviar allele frequency to distinguish rare variants

Sex chromosomes were ignored.

# Results/Discussion

I would like to walk through the logic of each step with Brady and if we are going to be using this query again in the future, turn the entire script into a function that can be run easily based on genes of interest. I feel like something may be missing from the comp het logic regarding genotype of parents. And I would also like to identify other types of MIE’s that I should search for.

Overall, if the logic is correct, than not much was found.

# Query

This query was run to find variants from ensembl in the regions of interest, add a column to identify trio member, and return results that are potential rare variants that are either not in kaviar, or in kaviar with a frequency of less than 10%:

WITH ens AS (  
 SELECT DISTINCT chromosome as chr, start, stop, gene\_name  
 FROM public\_hg19.ensembl\_genes  
 WHERE (gene\_name IN ( 'HADH', 'HADHA', 'HADHB', 'ACAA1',  
 'ACAA2', 'EHHADH', 'ECHS1')  
 OR gene\_name LIKE 'HSD17B%')  
 AND chromosome NOT LIKE "H%"  
)  
SELECT p.sample\_id, p.qual, p.filter, k.id as rsID, (k.alle\_freq \* 100) as kav\_pct, k.alle\_cnt as kav\_count,  
gene\_name, p.chr, p.pos, p.ref, p.alt, p.gt,   
 (CASE   
 WHEN SUBSTRING(p.sample\_id, -2) = '01'  
 THEN 'M'  
 WHEN SUBSTRING(p.sample\_id, -2) = '02'  
 THEN 'F'  
 WHEN SUBSTRING(p.sample\_id, -2) = '03'  
 THEN 'NB'  
 END) as member, CONCAT(gene\_name, ":", p.chr, ":", CAST(p.pos AS STRING)) as variant\_id,   
 CONCAT(p.chr, ":", CAST(p.pos AS STRING), ":", p.alt) as alt\_id   
 FROM  
(SELECT DISTINCT p.sample\_id, p.qual, p.filter, ens.gene\_name, p.chr, p.pos, p.ref, p.alt, p.gt  
 FROM ens, p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = ens.chr  
 AND (p.pos >= ens.start AND p.pos <= ens.stop)  
 AND p.gt IS NOT NULL  
) AS p  
LEFT JOIN /\* +SHUFFLE \*/ public\_hg19.kaviar k  
 ON p.chr = k.chromosome  
 AND p.pos = k.pos  
 AND p.ref = k.ref  
 AND p.alt = k.alt  
WHERE (k.alle\_freq < .10 OR k.alle\_freq IS NULL)

## Testing results against impala

The results of the query shown above were stored as a table on impala and used to:  
- Compare counts with data subset in R  
- Examine variants at positions reported in the R analysis to verify results

#########################################  
## read in and structure query results ##  
#########################################  
suppressMessages(library(readr))  
  
#load library  
library(RODBC)  
#connect using the DSN name you created on your machine  
conn <- odbcConnect("Impala DSN")  
  
#read in query results  
query = sqlFetch(conn, "users\_selasady.ben\_query")  
  
  
#quick check on data structure  
head(query)

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 1 102-01197-03 0 PASS <NA> NA NA HSD17B1 17  
## 2 102-01197-03 0 PASS <NA> NA NA HADHB 2  
## 3 102-01197-02 454 PASS rs2066479 7.16571 10302 HSD17B3 9  
## 4 102-01197-01 0 PASS <NA> NA NA ACAA2 18  
## 5 102-01197-02 0 PASS <NA> NA NA HSD17B12 11  
## 6 102-01197-01 0 PASS <NA> NA NA HSD17B2 16  
## pos ref alt gt member variant\_id  
## 1 40702217 A <NA> 0/0 NB HSD17B1:17:40702217  
## 2 26468301 T <NA> 0/0 NB HADHB:2:26468301  
## 3 98997810 C T 1/1 F HSD17B3:9:98997810  
## 4 47312649 G <NA> 0/0 M ACAA2:18:47312649  
## 5 43874537 T <NA> 0/0 F HSD17B12:11:43874537  
## 6 82132134 T <NA> 0/0 M HSD17B2:16:82132134

#dimensions  
dim(query)

## [1] 20541 14

#split apart data frame by member id and examine data  
mom = query[which(query$member == "M"),]  
dim(mom)

## [1] 7632 14

table(mom$gt)

##   
## . 0 0/0 0/1 1/1   
## 3 4 7603 19 3

dad = query[which(query$member == "F"),]  
dim(dad)

## [1] 6635 14

table(dad$gt)

##   
## . 0 0/0 0/1 1/1   
## 1 82 6535 10 7

nb = query[which(query$member == "NB"),]  
dim(nb) #6274 newborn variants

## [1] 6274 14

table(nb$gt)

##   
## . 0 0/0 0/1 1/1   
## 6 2 6245 16 5

###############################  
## genotype counts ##  
###############################  
#5 hom alt newborns  
hom\_alt\_nb = nb[which(nb$gt == "1/1"),]  
dim(hom\_alt\_nb)

## [1] 5 14

#15 het newborns in R query  
het\_nb = nb[(nb$gt == "0/1"),]  
dim(het\_nb)

## [1] 16 14

#6245 hom ref newborns  
hom\_ref\_nb = nb[(nb$gt == "0/0"),]  
dim(hom\_ref\_nb)

## [1] 6245 14

#19 het moms  
het\_mom = mom[which(mom$gt == "0/1"),]  
dim(het\_mom)

## [1] 19 14

#10 het moms   
het\_dad = dad[which(dad$gt == "0/1"),]  
dim(het\_dad)

## [1] 10 14

# Search for homozygous alt alleles

The data was read into R and searched for variants in the newborn marked as homozygous for the alternate allele. If the newborn alternate allele was the same as the alternate allele for both parents at this location, or if the nb allele was a no-call but both parent alleles were equal, it was marked as a homozygous alternate.

Homozygous alternate parent alleles were ignored.

#######################  
#find homozygous alt ##  
#######################  
#merge together to find genes in parents that match nb variants on each gene, by chr and position  
nb\_mom\_hom\_alt = merge(hom\_alt\_nb, het\_mom, by="variant\_id")  
dim(nb\_mom\_hom\_alt)#2/5 variants match moms variants

## [1] 2 27

#merge nb with dad  
nb\_dad\_hom\_alt= merge(nb\_mom\_hom\_alt, het\_dad, by="variant\_id")  
dim(nb\_dad\_hom\_alt)#2/5 variants match dad's variants

## [1] 1 40

#merge mom and dad's matching varaints  
hom\_alts = merge(nb\_mom\_hom\_alt, nb\_dad\_hom\_alt, by="variant\_id")  
dim(hom\_alts)#one variant matches both mom and dad

## [1] 1 66

#get rid of unnecessary columns  
hom\_alt.df = hom\_alts[,c(1:13,25:26,38:39)]  
colnames(hom\_alt.df) = c("variant\_id", "sample\_id", "qual", "filter", "rsID", "kav\_freqPct",   
 "kav\_count", "gene\_name", "chr", "pos","ref", "nb\_alt", "nb\_gt",   
 "m\_alt", "m\_gt", "f\_alt","f\_gt")  
  
#find variants where mother, father and newborn have matching alts at same chr/pos, or  
#mom and dad alts match and nb call is null  
hom\_alt.df = hom\_alt.df[which((hom\_alt.df$nb\_alt == hom\_alt.df$m\_alt) & (hom\_alt.df$nb\_alt == hom\_alt.df$f\_alt) |   
 (hom\_alt.df$m\_alt == hom\_alt.df$f\_alt & hom\_alt.df$nb\_alt == "NULL")  
 ),]  
hom\_alt.df

## variant\_id sample\_id qual filter rsID kav\_freqPct  
## 1 ACAA2:18:47339364 102-01197-03 385 PASS rs75954532 8.7829  
## kav\_count gene\_name chr pos ref nb\_alt nb\_gt m\_alt m\_gt f\_alt f\_gt  
## 1 1475 ACAA2 18 47339364 C T 1/1 T 0/1 T 1/1

## Confirm hom alt analysis

For each of the five homozygous alt newborn variants found, a query was run to manually check for all variants at that chromosome and position in the ITMI 102 puzzle table.

#Mother is 1/1 and father is 0/1  
hom\_alt\_nb[1,]

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 5144 102-01197-03 684 PASS rs72894480 7.66941 1288 HSD17B12 11  
## pos ref alt gt member variant\_id  
## 5144 43852180 A G 1/1 NB HSD17B12:11:43852180

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '16'  
 AND p.pos = 82068897  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## chr pos id ref alt gt sample\_id  
## 1 16 82068897 rs4445895 T C 0/1 102-01197-02  
## 2 16 82068897 rs4445895 T C 1/1 102-01197-01

#mother is 0/1 and father is 1/1  
hom\_alt\_nb[2,]

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 8082 102-01197-03 245 PASS rs610177 6.52023 9374 HSD17B1 17  
## pos ref alt gt member variant\_id  
## 8082 40705724 T G 1/1 NB HSD17B1:17:40705724

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '11'  
 AND p.pos = 43852180  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## chr pos id ref alt gt sample\_id  
## 1 11 43852180 rs72894480 A G 1/1 102-01197-02  
## 2 11 43852180 rs72894480 A G 0/1 102-01197-01

#both parents are 1/1  
hom\_alt\_nb[3,]

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 13217 102-01197-03 526 PASS rs4445895 9.09243 13072 HSD17B2 16  
## pos ref alt gt member variant\_id  
## 13217 82068897 T C 1/1 NB HSD17B2:16:82068897

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '17'  
 AND p.pos = 40705724  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## chr pos id ref alt gt sample\_id  
## 1 17 40705724 rs610177 T G 1/1 102-01197-02  
## 2 17 40705724 rs610177 T G 1/1 102-01197-01

#father 0/1, mother 1/1, this shows up in results as expected  
hom\_alt\_nb[4,]

## sample\_id qual filter rsid kav\_pct kav\_count  
## 17627 102-01197-03 41 LowGQX;HighREFREP rs59454424 0.07145 12  
## gene\_name chr pos ref alt gt member variant\_id  
## 17627 HSD17B7 1 162762821 C CTTT 1/1 NB HSD17B7:1:162762821

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '18'  
 AND p.pos = 47339364  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## chr pos id ref alt gt sample\_id  
## 1 18 47339364 rs75954532 C T 0/1 102-01197-02  
## 2 18 47339364 rs75954532 C T 0/1 102-01197-01

#no varaint reported for mother, several variants with filter errors, low qual  
hom\_alt\_nb[5,]

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 18165 102-01197-03 385 PASS rs75954532 8.7829 1475 ACAA2 18  
## pos ref alt gt member variant\_id  
## 18165 47339364 C T 1/1 NB ACAA2:18:47339364

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '1'  
 AND p.pos = 162762821  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## chr pos id ref alt gt sample\_id  
## 1 1 162762821 <NA> C <NA> 0/0 102-01197-02  
## 2 1 162762821 <NA> C CT 1/1 102-01197-02  
## 3 1 162762821 rs59454424 C CTTT 1/1 102-01197-02

The results are consistent with manual examination of the ITMI 102 data due to the exclusion of homozygous parent variants.

## Search for homozygous ref alleles

The data was subset for variants where the newborn marked as homozygous reference. If the newborn reference allele was the same as the reference allele in both parents het at this location, or if the nb allele was a no-call but both parent alleles were equal, it was marked as a homozygous reference.

Homozygous reference parent allelels were ignored.

#############################  
#find homozygous reference ##  
#############################  
hom\_ref\_nb = nb[which(nb$gt == "0/0"),]  
dim(hom\_ref\_nb) #6245 hom\_ref variants

## [1] 6245 14

#merge hom ref nb with 19 het mom variants, matching by chr and position  
hom\_ref\_nb\_mom = merge(hom\_ref\_nb, het\_mom, by="variant\_id")  
dim(hom\_ref\_nb\_mom) #1/19 het mom variant matches with newborns

## [1] 1 27

#merge hom ref nb with het dad  
hom\_ref\_nb\_dad = merge(hom\_ref\_nb, het\_dad, by="variant\_id")  
dim(hom\_ref\_nb\_dad) #0/10 het dads match with homozygous ref newborns

## [1] 0 27

#merge mom and dad matching variants  
hom\_refs = merge(hom\_ref\_nb\_mom, hom\_ref\_nb\_dad, by= "variant\_id")  
  
#get rid of unnecessary columns  
hom\_ref.df = hom\_refs[,c(1:13,25:26,38:39)]  
colnames(hom\_ref.df) = c("variant\_id", "sample\_id", "qual", "filter", "rsID", "kav\_freqPct",  
 "kav\_count", "gene\_name", "chr", "pos", "ref", "nb\_alt", "nb\_gt",  
 "m\_alt", "m\_gt", "f\_alt","f\_gt")  
  
#find variants where mother, father and newborn have matching alts at same chr/pos, or  
#mom and dad alts match and nb call is null  
hom\_ref.df = hom\_ref.df[which((hom\_ref.df$nb\_alt == hom\_ref.df$m\_alt) & (hom\_ref.df$nb\_alt == hom\_ref.df$f\_alt) |   
 (hom\_ref.df$m\_alt == hom\_ref.df$f\_alt & hom\_ref.df$nb\_alt == "NULL")  
 ),]  
hom\_ref.df

## [1] variant\_id sample\_id qual filter rsID   
## [6] kav\_freqPct kav\_count gene\_name chr pos   
## [11] ref nb\_alt nb\_gt m\_alt m\_gt   
## [16] f\_alt f\_gt   
## <0 rows> (or 0-length row.names)

## Confirm hom ref analysis

For the 6245 homozygous ref newborns variants found, a random set of 5 variants will be selected and each reported position examined manually in impala.

#randomly sample 5 hom\_ref nb variants  
hom\_ref\_sample = hom\_ref\_nb[sample(1:nrow(hom\_ref\_nb), 5, replace=FALSE),]  
  
#no variant calls from parents  
hom\_ref\_sample[1,]

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 19306 102-01197-03 0 PASS <NA> NA NA ACAA1 3  
## pos ref alt gt member variant\_id  
## 19306 38166544 A <NA> 0/0 NB ACAA1:3:38166544

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '3'  
 AND p.pos = 38165798  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## [1] chr pos id ref alt gt sample\_id  
## <0 rows> (or 0-length row.names)

#no parent variants reported  
hom\_ref\_sample[2,]

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 19544 102-01197-03 0 PASS <NA> NA NA HADHB 2  
## pos ref alt gt member variant\_id  
## 19544 26502047 C <NA> 0/0 NB HADHB:2:26502047

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '18'  
 AND p.pos = 47340244  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## [1] chr pos id ref alt gt sample\_id  
## <0 rows> (or 0-length row.names)

#both parents 0/0 at this location  
hom\_ref\_sample[3,]

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 3615 102-01197-03 0 PASS <NA> NA NA HADHA 2  
## pos ref alt gt member variant\_id  
## 3615 26414313 A <NA> 0/0 NB HADHA:2:26414313

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '11'  
 AND p.pos = 43702298  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## chr pos id ref alt gt sample\_id  
## 1 11 43702298 <NA> C <NA> 0/0 102-01197-02  
## 2 11 43702298 <NA> C <NA> 0/0 102-01197-01

#only one parent variant reported  
hom\_ref\_sample[4,]

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 16019 102-01197-03 0 PASS <NA> NA NA HSD17B10 X  
## pos ref alt gt member variant\_id  
## 16019 53460820 A <NA> 0/0 NB HSD17B10:X:53460820

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '2'  
 AND p.pos = 26413584  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## chr pos id ref alt gt sample\_id  
## 1 2 26413584 <NA> T <NA> 0/0 102-01197-01

#no parent varaints reported   
hom\_ref\_sample[5,]

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 1144 102-01197-03 0 PASS <NA> NA NA HSD17B12 11  
## pos ref alt gt member variant\_id  
## 1144 43859922 C <NA> 0/0 NB HSD17B12:11:43859922

sqlQuery(conn, "SELECT p.chr, p.pos, p.id, p.ref, p.alt, p.gt, p.sample\_id   
 FROM p7\_ptb.itmi\_102\_puzzle p  
 WHERE p.chr = '3'  
 AND p.pos = 38164764  
 AND (p.sample\_id LIKE '%01' OR p.sample\_id LIKE '%02')", as.is=TRUE)

## [1] chr pos id ref alt gt sample\_id  
## <0 rows> (or 0-length row.names)

The results are consistent with data found in the puzzle table. We may want to add in a way of handling newborn variant calls that are missing calls from either/both parents.

## Search for compound het alleles

To locate potential compound het variants, the data was subset for het newborn variants, and the parents were subset for either heterozygous or homozygous alternate variants. The data was grouped by gene name, and a function was ran in the following order:  
- Subset variants with more than one variant position per gene  
- The mother and father variants were matched to newborn variants

If variants were found from both parents: - Check that the alt alleles in each parent were at different positions  
- Check that alt alleles in each parent are not in the same position  
- Return variants where the newborn has a diffent variant from mom and dad on the same gene

#####################  
#find compound het ##  
#####################  
#subset for possible zygosity for compound het  
het\_nb = nb[which(nb$gt == "0/1"),]  
dim(het\_nb) #16 het newborns

## [1] 16 14

#function to check for comp\_hets  
find\_comphet = function(x){  
 #check if more than one variant found per gene  
 if (length(unique(x$pos)) >1 ) {  
 #find matching variants from each parent  
 mom\_vars = mom[grep(paste(x$variant\_id, collapse="|"), mom$variant\_id),]  
 dad\_vars = dad[grep(paste(x$variant\_id, collapse="|"), dad$variant\_id),]  
 #if either parent has a variant the other does not   
 #and there is at least one variant from each, mark as comp\_het  
 if ((dim(mom\_vars[!(mom\_vars$variant\_id %in% dad\_vars$variant\_id),])[1] > 0 |   
 dim(dad\_vars[!(dad\_vars$variant\_id %in% mom\_vars$variant\_id),])[1] > 0) &  
 (dim(mom\_vars)[1] > 0 & dim(dad\_vars)[1] > 0)){  
 rbind(mom\_vars, dad\_vars)  
 } else   
 print("Variants at different positions on the gene are from the same parent.")   
 }else #(length(unique(x$pos)) >1 ) = false  
 print("Variants on gene only found in same position.")  
}#end of function  
  
##apply function to newborn comp het candidates  
comp\_hets = by(het\_nb, as.character(het\_nb$gene\_name), find\_comphet)

## [1] "Variants on gene only found in same position."  
## [1] "Variants on gene only found in same position."  
## [1] "Variants on gene only found in same position."  
## [1] "Variants at different positions on the gene are from the same parent."  
## [1] "Variants at different positions on the gene are from the same parent."

#data frame results - automate this dammit Summer  
comp\_het.df = rbind(comp\_hets$HSD17B1, comp\_hets$HSD17B13, comp\_hets$HSD17B4, comp\_hets$HSD17B7P2)

## Confirm Results of comp-het analysis

The results of the comp-het analysis will be confirmed by grouping the het variants found in the newborn by gene and manually examining all het variants found in the newborn on each gene.

Variations of the following query were used to find all het newborn variants on each of the genes of interest:

WITH ens AS (  
 SELECT DISTINCT chromosome as chr, start, stop, gene\_name  
 FROM public\_hg19.ensembl\_genes  
 WHERE gene\_name = "ACAA2"  
 AND chromosome NOT LIKE "H%"  
)  
SELECT p.sample\_id, p.qual, p.filter, k.id as rsID, (k.alle\_freq \* 100) as kav\_pct, k.alle\_cnt as kav\_count,  
gene\_name, p.chr, p.pos, p.ref, p.alt, p.gt  
 FROM  
(SELECT DISTINCT p.sample\_id, p.qual, p.filter, ens.gene\_name, p.chr, p.pos, p.ref, p.alt, p.gt  
 FROM ens, p7\_ptb.itmi\_102\_puzzle p  
 WHERE ens.chr = p.chr  
 AND p.sample\_id LIKE "%03"  
 AND (p.pos >= ens.start AND p.pos <= ens.stop)  
 AND p.gt = "0/1"  
) AS p  
LEFT JOIN /\* +SHUFFLE \*/ public\_hg19.kaviar k  
 ON p.chr = k.chromosome  
 AND p.pos = k.pos  
 AND p.ref = k.ref  
 AND p.alt = k.alt  
WHERE (k.alle\_freq < .10 OR k.alle\_freq IS NULL)

Variations on the following query were used to examine all variants found on genes with more than one het nb variant;

WITH ens AS (  
 SELECT DISTINCT chromosome as chr, start, stop, gene\_name  
 FROM public\_hg19.ensembl\_genes  
 WHERE gene\_name = "HSD17B3"  
 AND chromosome NOT LIKE "H%"  
)  
SELECT p.sample\_id, p.qual, p.filter, k.id as rsID, (k.alle\_freq \* 100) as kav\_pct, k.alle\_cnt as kav\_count,  
gene\_name, p.chr, p.pos, p.ref, p.alt, p.gt  
 FROM  
(SELECT DISTINCT p.sample\_id, p.qual, p.filter, ens.gene\_name, p.chr, p.pos, p.ref, p.alt, p.gt  
 FROM ens, p7\_ptb.itmi\_102\_puzzle p  
 WHERE ens.chr = p.chr  
 AND p.chr = "9"  
 AND (p.pos = 99003102 OR p.pos = 98997810)  
 AND (p.pos >= ens.start AND p.pos <= ens.stop)  
) AS p  
LEFT JOIN /\* +SHUFFLE \*/ public\_hg19.kaviar k  
 ON p.chr = k.chromosome  
 AND p.pos = k.pos  
 AND p.ref = k.ref  
 AND p.alt = k.alt  
WHERE (k.alle\_freq < .10 OR k.alle\_freq IS NULL)  
ORDER BY p.sample\_id, p.gt, p.chr, p.pos

#group het nb by gene and examine variants on each gene  
by\_gene = split(het\_nb, het\_nb$gene\_name)  
  
#het nb variant found at only one position on the gene  
by\_gene$ACAA2

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 840 102-01197-03 247 PASS rs149436920 0.66983 963 ACAA2 18  
## pos ref alt gt member variant\_id  
## 840 47318683 A T 0/1 NB ACAA2:18:47318683

#het nb variant found at only one position on the gene  
by\_gene$EHHADH

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 14729 102-01197-03 171 PASS rs144464757 0.09807 141 EHHADH 3  
## pos ref alt gt member variant\_id  
## 14729 184911203 G A 0/1 NB EHHADH:3:184911203

#het nb variant found at only one position on the gene  
by\_gene$HADHB

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 18440 102-01197-03 718 PASS <NA> NA NA HADHB 2  
## pos ref alt gt member variant\_id  
## 18440 26505915 CT C 0/1 NB HADHB:2:26505915

#variants reported at chr 17 40702252 and 40702760  
by\_gene$HSD17B1

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 1058 102-01197-03 85 PASS rs184482331 1.54817 260 HSD17B1 17  
## 13010 102-01197-03 62 PASS rs62075836 7.87781 1323 HSD17B1 17  
## pos ref alt gt member variant\_id  
## 1058 40702760 C T 0/1 NB HSD17B1:17:40702760  
## 13010 40702252 C T 0/1 NB HSD17B1:17:40702252

#variant at 40702760 found only in father  
#variant at 40702252 found only in mother  
  
#variants reported at chr 11 43876002 and 43877726, both from mother  
by\_gene$HSD17B12

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 8836 102-01197-03 180 PASS rs578099036 0.05955 10 HSD17B12 11  
## 14331 102-01197-03 200 PASS rs574290595 0.01786 3 HSD17B12 11  
## pos ref alt gt member variant\_id  
## 8836 43877726 A G 0/1 NB HSD17B12:11:43877726  
## 14331 43876002 T C 0/1 NB HSD17B12:11:43876002

#variants reported at chr 4 88225032 and 88238293  
by\_gene$HSD17B13

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 7071 102-01197-03 945 PASS rs201024861 0.01191 2 HSD17B13 4  
## 9194 102-01197-03 117 PASS rs201763274 0.02087 30 HSD17B13 4  
## pos ref alt gt member variant\_id  
## 7071 88225032 T TGTTA 0/1 NB HSD17B13:4:88225032  
## 9194 88238293 T C 0/1 NB HSD17B13:4:88238293

#variant at 88225032 found in both parents  
#variant at 88238293 only found in mother  
  
### check on logic using this variant ##   
find\_comphet(by\_gene$HSD17B13)

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 1231 102-01197-01 121 PASS rs201763274 0.02087 30 HSD17B13 4  
## 6703 102-01197-01 900 PASS rs201024861 0.01191 2 HSD17B13 4  
## 308 102-01197-02 1644 PASS rs201024861 0.01191 2 HSD17B13 4  
## pos ref alt gt member variant\_id  
## 1231 88238293 T C 0/1 M HSD17B13:4:88238293  
## 6703 88225032 T TGTTA 0/1 M HSD17B13:4:88225032  
## 308 88225032 T TGTTA 0/1 F HSD17B13:4:88225032

#variants reporated at chr 9 99003102 and 98997810 from father only  
by\_gene$HSD17B3

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 8552 102-01197-03 192 PASS <NA> NA NA HSD17B3 9  
## 11401 102-01197-03 115 PASS rs2066479 7.16571 10302 HSD17B3 9  
## pos ref alt gt member variant\_id  
## 8552 99003102 C T 0/1 NB HSD17B3:9:99003102  
## 11401 98997810 C T 0/1 NB HSD17B3:9:98997810

#variants reported at chr 5 118788196, 118831460 and 118970770  
by\_gene$HSD17B4

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 2978 102-01197-03 194 PASS rs26180 6.72820 9673 HSD17B4 5  
## 11931 102-01197-03 186 PASS <NA> NA NA HSD17B4 5  
## 13059 102-01197-03 209 PASS rs80063787 2.12008 3048 HSD17B4 5  
## pos ref alt gt member variant\_id  
## 2978 118788196 C G 0/1 NB HSD17B4:5:118788196  
## 11931 118970770 C T 0/1 NB HSD17B4:5:118970770  
## 13059 118831460 G A 0/1 NB HSD17B4:5:118831460

#variant at 118788196 only found in mother  
#variant at 118831460 only found in mother  
#variant at 118970770 only found in father  
  
#variants reported at chr 10 38667003 and 38667067  
by\_gene$HSD17B7P2

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 9368 102-01197-03 144 PASS rs571149853 0.02382 4 HSD17B7P2 10  
## 18422 102-01197-03 135 PASS rs2804622 3.63850 5231 HSD17B7P2 10  
## pos ref alt gt member variant\_id  
## 9368 38667067 A G 0/1 NB HSD17B7P2:10:38667067  
## 18422 38667003 A G 0/1 NB HSD17B7P2:10:38667003

#variant at 38667003 found in both parents  
#variant at 38667067 only found in mother

Variants found were consistent with manual search conducted on each gene in impala.

## Search for Mendelian Inheritance Errors

The resulting data set from the impala query was checked for congruence with laws of Mendelian inheritance, as follows.

#### Het newborn variants

For newborn variants genotyped as het (0/1):  
- If the mother is 0/0, then mother must be 0/1 and the nb alt must match the father’s alt  
- If the father is 0/0, then father must be 0/1 and the nb alt must match the mother’s alt  
- If both parents are het, nb alt and parent alts must be the same  
- If the mother is 1/1, then the father must be 0/1 or 0/0 and the nb alt must match the mother’s alt  
- If the father is 1/1, then the mother must be 0/1 or 0/0 and the nb alt must match the fathers’s alt

####################################  
## Het = 16 nb variants with 0/1 ##  
####################################  
nb\_het = nb[which(nb$gt == "0/1"),]  
  
#find matching variants from each parent  
mom\_vars = mom[grep(paste(nb\_het$variant\_id, collapse="|"), mom$variant\_id),] #10 vars  
dad\_vars = dad[grep(paste(nb\_het$variant\_id, collapse="|"), dad$variant\_id),] #9 vars  
  
#merge together to find intersection of gene:chr:pos  
mie\_het\_nb\_mom = merge(nb\_het, mom\_vars, by = "variant\_id")  
mie\_hets\_nb\_dad = merge(nb\_het, dad\_vars, by = "variant\_id")  
mie\_hets = merge(mie\_het\_nb\_mom, mie\_hets\_nb\_dad, by= "variant\_id")  
  
#get rid of unnecessary columns  
mie\_hets = mie\_hets[,c(1:13,25:26,38:39)]  
colnames(mie\_hets) = c("variant\_id", "sample\_id", "qual", "filter", "rsID", "kav\_freqPct",  
 "kav\_count", "gene\_name", "chr", "pos", "ref", "nb\_alt", "nb\_gt",  
 "m\_alt", "m\_gt", "f\_alt","f\_gt")  
  
#variants not in mie\_hets and not marked as comp\_het are missing info from one parent  
missing\_info = nb\_het[grep(paste(mie\_hets$variant\_id, collapse="|"), nb\_het$variant\_id, invert=TRUE),]  
missing\_info = missing\_info[grep(paste(comp\_het.df$variant\_id, collapse="|"), missing\_info$variant\_id, invert=TRUE),]  
missing\_info

## sample\_id qual filter rsid kav\_pct kav\_count gene\_name chr  
## 8552 102-01197-03 192 PASS <NA> NA NA HSD17B3 9  
## 8836 102-01197-03 180 PASS rs578099036 0.05955 10 HSD17B12 11  
## 11401 102-01197-03 115 PASS rs2066479 7.16571 10302 HSD17B3 9  
## 14331 102-01197-03 200 PASS rs574290595 0.01786 3 HSD17B12 11  
## 14729 102-01197-03 171 PASS rs144464757 0.09807 141 EHHADH 3  
## 18440 102-01197-03 718 PASS <NA> NA NA HADHB 2  
## pos ref alt gt member variant\_id  
## 8552 99003102 C T 0/1 NB HSD17B3:9:99003102  
## 8836 43877726 A G 0/1 NB HSD17B12:11:43877726  
## 11401 98997810 C T 0/1 NB HSD17B3:9:98997810  
## 14331 43876002 T C 0/1 NB HSD17B12:11:43876002  
## 14729 184911203 G A 0/1 NB EHHADH:3:184911203  
## 18440 26505915 CT C 0/1 NB HADHB:2:26505915

#these were verified by hand on impala as missing info from one parent  
  
#find variants that are in congruence with MI laws  
no\_mie = rbind(  
 #if the mother is 0/0 and father is 0/1, then the nb alt must match the father's alt or be null  
 mie\_hets[(mie\_hets$m\_gt == "0/0" & mie\_hets$f\_gt == "0/1" & (mie\_hets$nb\_alt == mie\_hets$f\_alt | mie\_hets$nb\_alt == "NULL")),],  
 #if the father is 0/0 and the mother is 0/1, then the nb alt must match the mother's alt or be null  
 mie\_hets[(mie\_hets$f\_gt == "0/0" & mie\_hets$m\_gt == "0/1" & (mie\_hets$nb\_alt == mie\_hets$m\_alt| mie\_hets$nb\_alt == "NULL")),],  
 #if both parents are het, nb alt and parent alts must be the same, or nb alt is null  
 mie\_hets[((mie\_hets$m\_gt == "0/1" & mie\_hets$f\_gt== "0/1") & (mie\_hets$nb\_alt == mie\_hets$f\_alt & mie\_hets$nb\_alt == mie\_hets$m\_alt)| (mie\_hets$m\_alt == mie\_hets$f\_alt & mie\_hets$nb\_alt == "NULL")),],  
 #if the mother is 1/1, then the father must be 0/1 or 0/0 and the nb alt must match the mother's alt  
 #or be null  
 mie\_hets[(mie\_hets$m\_gt == "1/1" & (mie\_hets$f\_gt == "0/1" | mie\_hets$f\_gt == "0/0") & (mie\_hets$nb\_alt == mie\_hets$m\_alt & mie\_hets$nb\_alt == mie\_hets$f\_alt | mie\_hets$m\_alt == mie\_hets$f\_alt & mie\_hets$nb\_alt == "NULL")),],  
 #if the father is 1/1, then the mother must be 0/1 or 0/0 and the nb alt must match the fathers's alt  
 mie\_hets[(mie\_hets$f\_gt == "1/1" & (mie\_hets$m\_gt == "0/1" | mie\_hets$m\_gt == "0/0") & (mie\_hets$nb\_alt == mie\_hets$f\_alt & mie\_hets$nb\_alt == mie\_hets$m\_alt | mie\_hets$f\_alt == mie\_hets$m\_alt & mie\_hets$nb\_alt == "NULL")),])  
  
#if mie\_het variants are not in no\_mie set, then they are potentially MIE  
mie\_het\_cands = mie\_hets[grep(paste(no\_mie$variant\_id, collapse="|"), mie\_hets$variant\_id, invert=TRUE),]  
mie\_het\_cands

## [1] variant\_id sample\_id qual filter rsID   
## [6] kav\_freqPct kav\_count gene\_name chr pos   
## [11] ref nb\_alt nb\_gt m\_alt m\_gt   
## [16] f\_alt f\_gt   
## <0 rows> (or 0-length row.names)

## Hom ref newborn variants

For newborn variants genotyped as homozygous ref (0/0):  
- If the mother is 0/1, then the dad must be 0/0  
- If the father is 0/1, then the mother must be 0/0  
- Or both parents are 0/0

#################################################  
## homozygous ref = 6245 nb variants with 0/0 ##  
#################################################  
nb\_hom\_ref = nb[which(nb$gt == "0/0"),]  
  
#find equivalent variants in parents  
mom\_hom\_vars = na.omit(mom[match(nb\_hom\_ref$variant\_id, mom$variant\_id),])  
dad\_hom\_vars = na.omit(dad[match(nb\_hom\_ref$variant\_id, dad$variant\_id),])  
  
#merge together to find matching parent varients  
mie\_hom\_nb\_mom = merge(nb\_hom\_ref, mom\_hom\_vars, by = "variant\_id")  
mie\_hom\_nb\_dad = merge(mie\_hom\_nb\_mom, dad\_hom\_vars, by = "variant\_id")  
mie\_homs = merge(mie\_hom\_nb\_mom, mie\_hom\_nb\_dad, by= "variant\_id")  
  
#clean up resuts  
mie\_homs = mie\_homs[,c(1:13,25:26,38:39)]  
colnames(mie\_homs) = c("variant\_id", "sample\_id", "qual", "filter", "rsID", "kav\_freqPct",  
 "kav\_count", "gene\_name", "chr", "pos", "ref", "nb\_alt", "nb\_gt",  
 "m\_alt", "m\_gt", "f\_alt","f\_gt")  
  
#variants that are not in mie\_homs and not marked hom ref are missing parent info  
missing\_info\_miehoms = nb\_hom\_ref[grep(paste(mie\_homs$variant\_id, collapse="|"), nb\_hom\_ref$variant\_id, invert=TRUE),]  
missing\_info\_miehoms = missing\_info[grep(paste(hom\_ref.df$variant\_id, collapse="|"), missing\_info$variant\_id, invert=TRUE),]  
missing\_info\_miehoms

## [1] sample\_id qual filter rsid kav\_pct kav\_count   
## [7] gene\_name chr pos ref alt gt   
## [13] member variant\_id  
## <0 rows> (or 0-length row.names)

#find variants that are in congruence with MI laws  
no\_mie\_homs = rbind(  
 #if the mother is 0/1, then the dad must be 0/0   
 mie\_homs[(mie\_homs$m\_gt == "0/1" & mie\_homs$f\_gt == "0/0"),],  
 #if the father is 0/1, then the mother must be 0/0  
 mie\_homs[(mie\_homs$d\_gt == "0/1" & mie\_homs$m\_gt == "0/0"),],  
 #both parents are 0/0  
 mie\_homs[(mie\_homs$m\_gt == "0/0" & mie\_homs$f\_gt== "0/0"),]  
)  
  
#if nb\_hom variants are not in no\_mie\_homs set, then they are MIE  
na.omit(nb\_hom\_ref[!match(nb\_hom\_ref$variant\_id, no\_mie\_homs$variant\_id),])

## [1] sample\_id qual filter rsid kav\_pct kav\_count   
## [7] gene\_name chr pos ref alt gt   
## [13] member variant\_id  
## <0 rows> (or 0-length row.names)

## Hom alt newborn variants

For newborn variants genotyped as het (1/1):  
- If the father is 0/1, then the mother must be 1/1  
- If the mother is 0/1, then the father must be 1/1  
- Or both parents are 1/1

##############################################  
## homozygous alt = 5 nb variants with 1/1 ##  
##############################################  
nb\_hom\_alt = nb[which(nb$gt == "1/1"),]  
  
#find equivalent variants in parents  
mom\_hom\_alt = na.omit(mom[match(nb\_hom\_alt$variant\_id, mom$variant\_id),])  
dad\_hom\_alt = na.omit(dad[match(nb\_hom\_alt$variant\_id, dad$variant\_id),])  
  
#merge together to find intersection of gene:chr:pos  
mie\_alt = merge(nb\_hom\_alt, mom\_hom\_alt, by = "variant\_id")  
mie\_alt = merge(mie\_alt, dad\_hom\_alt, by = "variant\_id")  
  
#clean up resuts  
mie\_alts = mie\_alt[,c(1:13,25:26,38:39)]  
colnames(mie\_alts) = c("variant\_id", "qual", "filter", "rsID", "kav\_pct",  
 "kav\_count", "gene\_name", "chr", "pos", "ref", "nb\_alt",  
 "nb\_gt", "m\_alt", "m\_gt", "f\_alt", "f\_gt")  
  
#find variants that are in congruence with MI laws  
mie\_hom\_alt = rbind(  
 #if the mother is 0/1, then the father must be 1/1  
 mie\_alts[(mie\_alts$m\_gt == "0/1" & mie\_alts$f\_gt == "1/1"),],  
 #if the father is 0/1, then the mother must be 1/1  
 mie\_alts[(mie\_alts$d\_gt == "0/1" & mie\_alts$m\_gt == "1/1"),],  
 #both parents are 1/1  
 mie\_alts[(mie\_alts$m\_gt == "1/1" & mie\_alts$f\_gt == "1/1"),]  
)  
  
#if nb\_hom variants are not in no\_mie\_homs set, then they are MIE  
na.omit(nb\_hom\_alt[!match(nb\_hom\_alt$variant\_id, mie\_hom\_alt$variant\_id),])

## [1] sample\_id qual filter rsid kav\_pct kav\_count   
## [7] gene\_name chr pos ref alt gt   
## [13] member variant\_id  
## <0 rows> (or 0-length row.names)