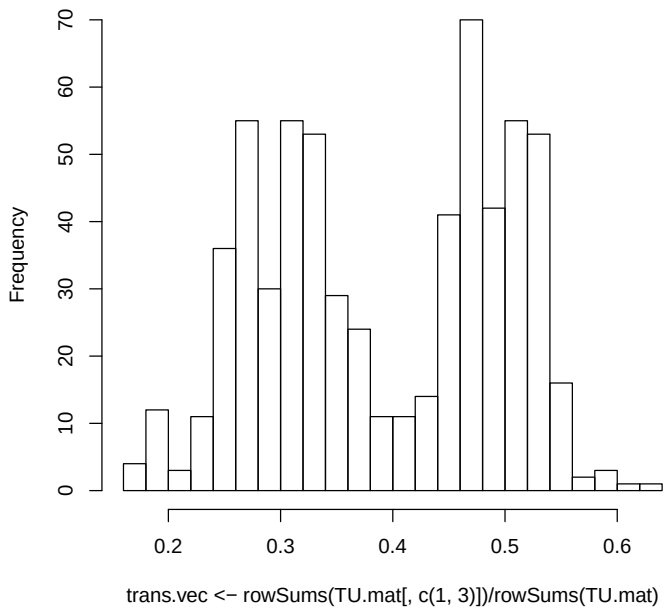
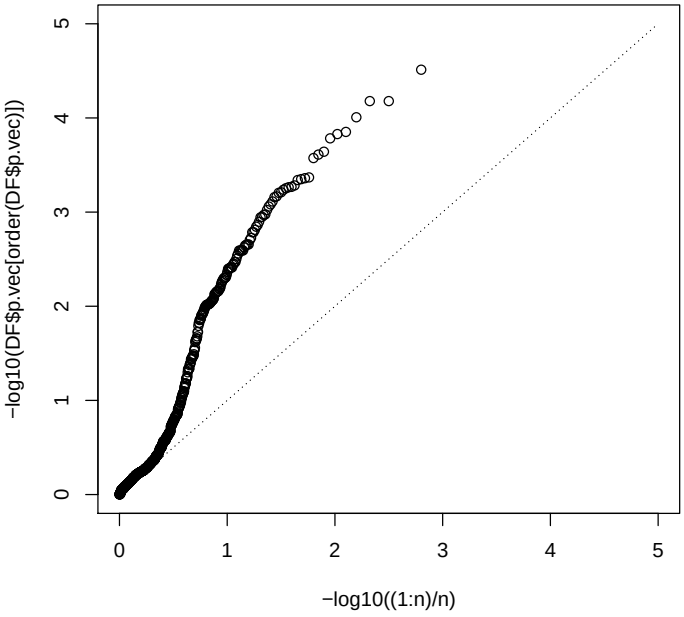


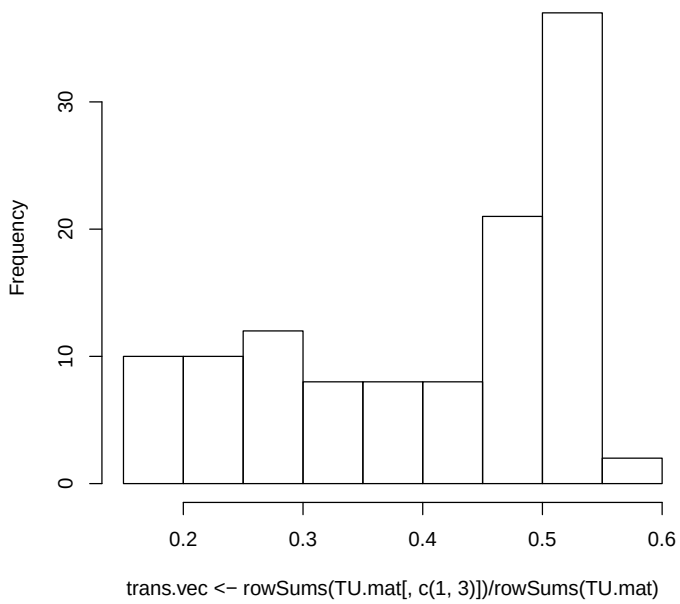

```
> TU.mat.beaty <- matrix(unlist(lapply(table.list.beaty, trioClasses:::CountTU)),
     nrow = length(table.list.beaty), ncol = 2, byrow = TRUE)
> TU.mat.pitt <- matrix(unlist(lapply(table.list.pitt, trioClasses:::CountTU)),
     nrow = length(table.list.pitt), ncol = 2, byrow = TRUE)
> TU.mat <- cbind(TU.mat.beaty, TU.mat.pitt)
> testable <- which((rowSums(TU.mat[, 1:2]) >= 25) & (rowSums(TU.mat[,
     3:4]) >= 25))
> TU.mat <- TU.mat[testable, ]
> rownames(TU.mat) <- names(table.list.beaty)[testable]
> colnames(TU.mat) <- c("T.case", "U.case", "T.con", "U.con")
> DF <- DataFrame(TU.mat, rowData(fe.beaty)[testable])
> colnames(DF) <- c(colnames(TU.mat), "grange")


> fish.list <- apply(TU.mat, 1, trioClasses:::TU.fish)
> p.vec <- unlist(lapply(fish.list, function(obj) return(obj$p.value)))
> DF <- DataFrame(DF, p.vec)

> c(length(DF$grange), length(reduce(DF$grange)))

[1] 632  45
```

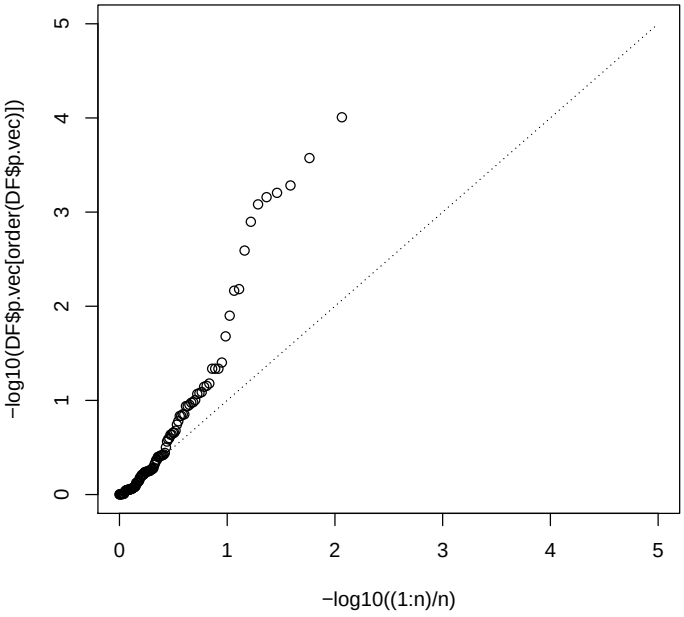

−log10(DF$p.vec[order(DF$p.vec)])
−log10((1:n)/n)
0
1
2
3
4
5




```
> badloci.gr <- reduce(rowData(fe.beaty[testable]))[which(countOverlaps(reduce(rowData(fe.be
     rowData(fe.beaty[testable])) > 10)]
> bad <- subjectHits(findOverlaps(badloci.gr, DF$grange))
> DF <- DF[-bad, ]
> TU.mat <- TU.mat[-bad, ]

> c(length(DF$grange), length(reduce(DF$grange)))

[1] 116  24
```


−log10(DF$p.vec[order(DF$p.vec)])
−log10((1:n)/n)
0
1
2
3
4
5