Chi Squared Tests for Heterogeneity

Rachael 'Rocky' Aikens, Voight Lab

June 20, 2017

Set Up

This analysis requires data on the counts of each 3mer, 5mer, and 7mer polymorphism type private to each of the 1,000 genomes nonadmixed continental groups (AFR, EUR, EAS, and SAS). All of this data has been preprocessed using the process_chrom_counts function in "code/data_wrangling/process_chrom_counts.R". I call these "count dataframes." An example is shown here:

```
##    Context  X3mer X1mer Count         Rate context_in_genome chr1 chr2
## 1 AAAAA->C AAA->C  A->C 14794 1.884656e-09          16034992 1110 1220
## 2 AAAAA->G AAA->G  A->G 24224 3.085974e-09          16034992 1824 1966
## 3 AAAAA->T AAA->T  A->T  9734 1.240046e-09          16034992  723  762
## 4 AAAAC->C AAA->C  A->C  3858 1.519035e-09           5188115  277  294
## 5 AAAAC->G AAA->G  A->G  4750 1.870248e-09           5188115  317  426
```

Pairwise Chi Squared Tests

This section details how to perform the pairwise chi squared tests from Harris 2015. These steps were mostly used for replication.

Methodology

Two R functions that I use in this analysis:

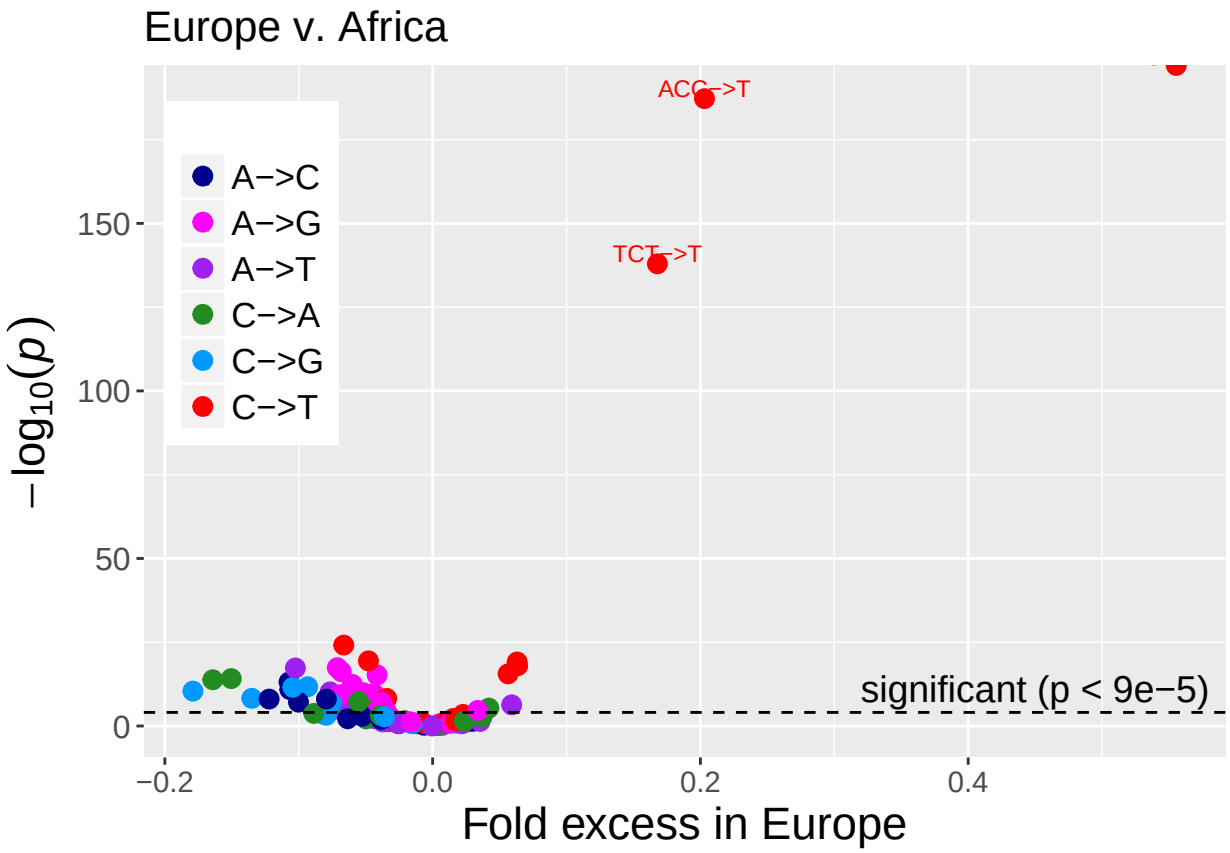
- **pairwise.chi** Given two count dataframes, output a dataframe of chi-squared test results for each context. The arguement 'filter' (set by default to be true), will output "NA" as the p-value for any test for which the chi squared assumptions may not be correct.
- **volcano.plot** Given two count dataframes, plus the output from pairwise.chi, construct a volcano plot as in Harris 2015. Also takes the arguement lab.lim, which determines the lower p-value limit for which polymorphisms types should be labeled.

The code used to definte pairwise.chi is shown below:

```
# calculates homogeneity test p values for pairwise comparisons of two dfs of counts
pairwise.chi <- function(counts.1, counts.2, filter = T){
  n.contexts = length(counts.1$Context)
  result <- data.frame(matrix(ncol=2,nrow=n.contexts))
  colnames(result) <- c("Context", "p")

  result$Context <- counts.1$Context
  sum.1 <- sum(counts.1$Count)
  sum.2 <- sum(counts.2$Count)

  for (i in 1:n.contexts){
    c.a <- c(counts.1$Count[i], counts.2$Count[i])
    c.b <- c(sum.1, sum.2) - c.a
    data <- cbind(c.a, c.b)
    warning <- is(tryCatch(chisq.test(data), warning = function(w) w), "warning")
    if (filter == T & warning){
      result$p[i] <- NA
    }
    else result$p[i] <- chisq.test(data)$p.value
  }

  return(result)
}
```

Examples

As previously mentioned, these methods are not central to my analysis and mostly important for replication. It's worthwhile to note that these functions and data replicate volcano plots like the ones from Kelley Harris's leading figure in PNAS 2015:



Fourway Tests for Homogeneity

In order to lighten the multiple hypothesis testing burden of running $\binom{4}{2} = 6$ pairwise comparisons for each possible polymorphism type, we switched to a homogeneity testing framework, which helps us rank polymorphism types based on how much they vary between populations. This is the dominant analysis technique we use to identify polymorphisms which are heterogeneous across continental groups.

Methodology

This section defines one function for calculation and two for visualization. Again, I'm hiding the code for the plotting functions in the compiled report because it's not essential to understanding.

- **fourway.chi** Given four count dataframes, output a dataframe of chi-squared test results for each context.
- **hom.test.plot** Given the output from fourway.chi, construct a volcano plot as in Harris 2015. Also takes the arguement lab.lim, which determines the lower p-value limit for which polymorphisms types should be labeled, and the boolean, NoTCC, which, when True, leaves out labels for any polymorphism with the 3mer subcontext TCC->T.
- **sigs.plot** Given the same arguments as hom.test.plot, make a plot of just the significant results.

The r code used to define fourway.chi is shown below:

```
# calculates homogeneity test p values for Fourway comparisons of counts dfs
fourway.chi <- function(AFR, EUR, EAS, SAS, filter = T){
  n.contexts = length(AFR$Context)

  # make dataframe for results
  result <- data.frame(matrix(ncol=9,nrow=n.contexts))
  colnames(result) <- c("Context", "X5mer","X3mer", "X1mer",
                        "AFR.Count", "EUR.Count", "EAS.Count", "SAS.Count", "p")
  result$Context <- AFR$Context
  result$X5mer <- AFR$X5mer # for smaller contexts, X3mer and X5mer columns do not exist,
  result$X3mer <- AFR$X3mer # and will disappear at this step
  result$X1mer <- AFR$X1mer
  result$AFR.Count <- AFR$Count; result$EUR.Count <- EUR$Count
  result$EAS.Count <- EAS$Count; result$SAS.Count <- SAS$Count

  # start setting up tables
  sums <- c(sum(AFR$Count), sum(EUR$Count), sum(EAS$Count), sum(SAS$Count))

  # set up table and run test for each context
  for (i in 1:n.contexts){
    c.a <- c(AFR$Count[i], EUR$Count[i], EAS$Count[i], SAS$Count[i])
    c.b <- sums - c.a
    data <- cbind(c.a, c.b)
    warning <- is(tryCatch(chisq.test(data), warning = function(w) w), "warning")
    if (filter == T & warning){
      result$p[i] <- NA}
    else result$p[i] <- chisq.test(data)$p.value
    }
  return(result)
}
```

3mers

Now using these functions, we can run the following tests for 3mer polymorphism types which are heterogeneous across ancestral groups. We can begin with the 3mer context paradigm, which is most commonly used in the literature in this area. The table below shows all 3mer polymorphism types, ranked according to their p value in a fourway.chi test for heterogeneity across populations.

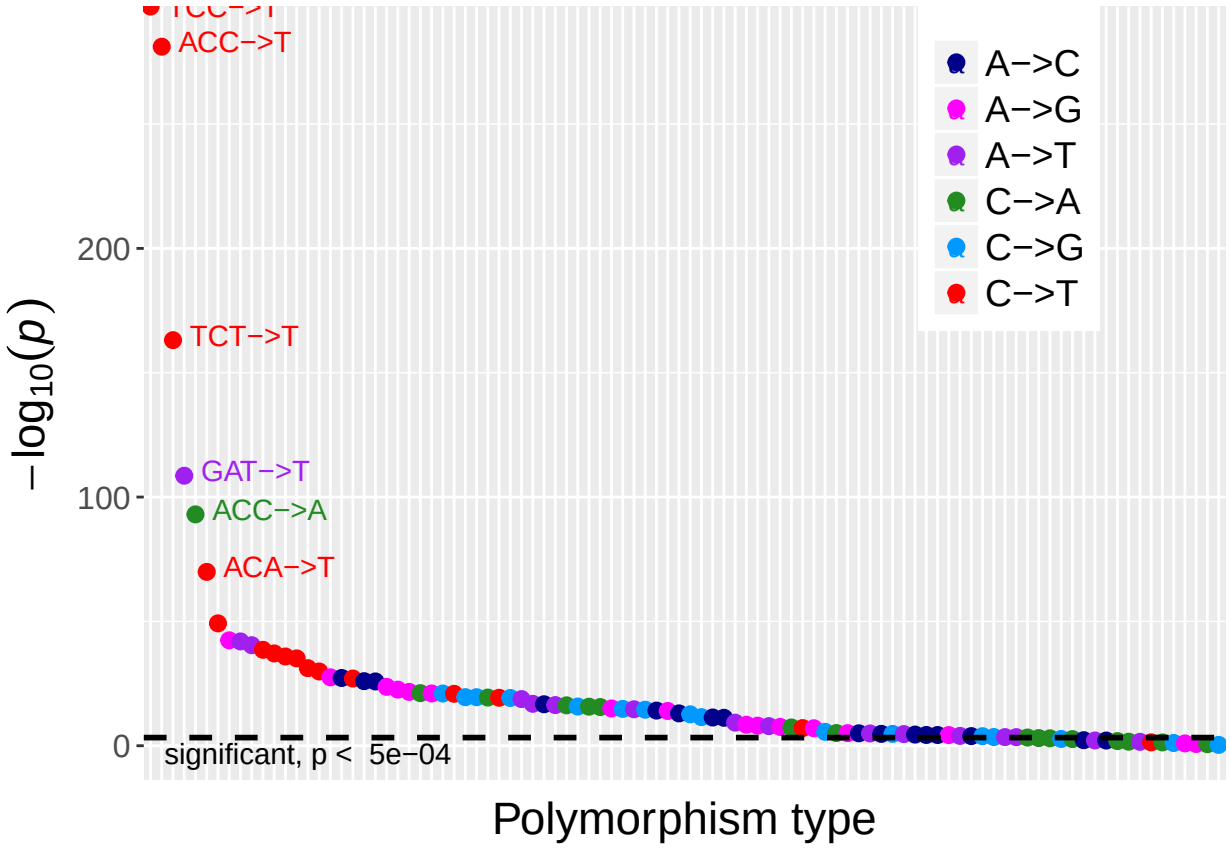


Table 1: 10 most significant 3mer polymorphisms

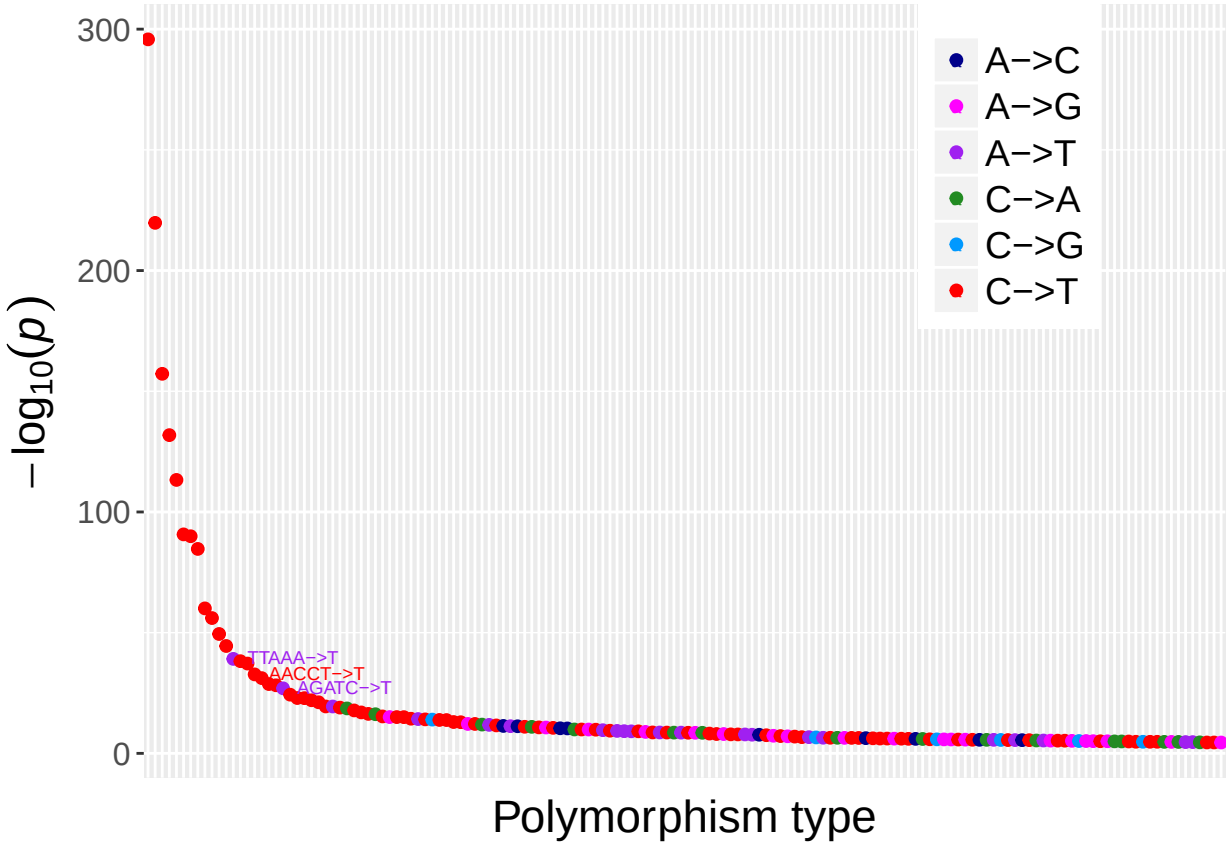
Context	X1mer	AFR.Count	EUR.Count	EAS.Count	SAS.Count	p
TCC->T	C->T	129676	37083	36252	43951	0.000000e+00
ACC->T	C->T	135088	29880	35002	40890	7.833611e-282
TCT->T	C->T	144254	30975	39522	43187	8.047424e-164
GAT->T	A->T	47152	9181	15883	15054	2.555990e-109
ACC->A	C->A	76646	14684	24481	23819	9.129322e-94
ACA->T	C->T	227803	39876	58806	62287	1.221705e-70
CCC->T	C->T	133143	26025	35129	38511	6.106054e-50
CAC->G	A->G	93424	15954	24869	24129	3.967717e-43
TAA->T	A->T	44996	7425	11763	11167	1.123517e-42
GAC->T	A->T	21353	4012	7135	6569	3.712928e-41

This plot highlights the top six contexts, which are significant at p<1e-60. They include TCC->T, ACC->T, and TCT->T, which have been previously reported as part of a European signal of C->T elevation. The next three contexts have not been noted by any previous analyses of mutation rate heterogeneity. There are 79 significant polymorphisms falling out from this analysis after Bonferroni correction.

5mers

Now we move to higher levels of sequence context, which may capture more detail in how mutation rates vary. In this section, we run the same analysis as above for 5mers, identifying variable polymorphism types which may not have been highlighted at the 3mer level.

The plot below shows the homogeneity test p values for just the 152 5mers which are significant after bonferroni correction.



It is clear from this plot (and the one for 7mers) that many of the signficant polymorphisms at the 5mer and 7mer level are a part of the signal of C->T elevation that we observe at the 3mer level. This begs the question: how many significant 5mer signals are there outside of the 3mer subcontexts we have already idenified? To answer this question, I removed from the significant 5mer set all mutations whose 3mer subcontexts correspond to the European C->T elevation (TCC->T, ACC->T, TCT->T, and CCC->T), or otherwise highlighted by the heatmap signals 1-3 we identify in another analysis (see heatmaps report) identified in the previous section. This leaves a total of 80 new significant polymorphisms. The following table shows the most highly significant 5mers *outside* of these 3mer signals that have already been noted:

Table 2: 10 most significant 5mers not noted on a 3mer level

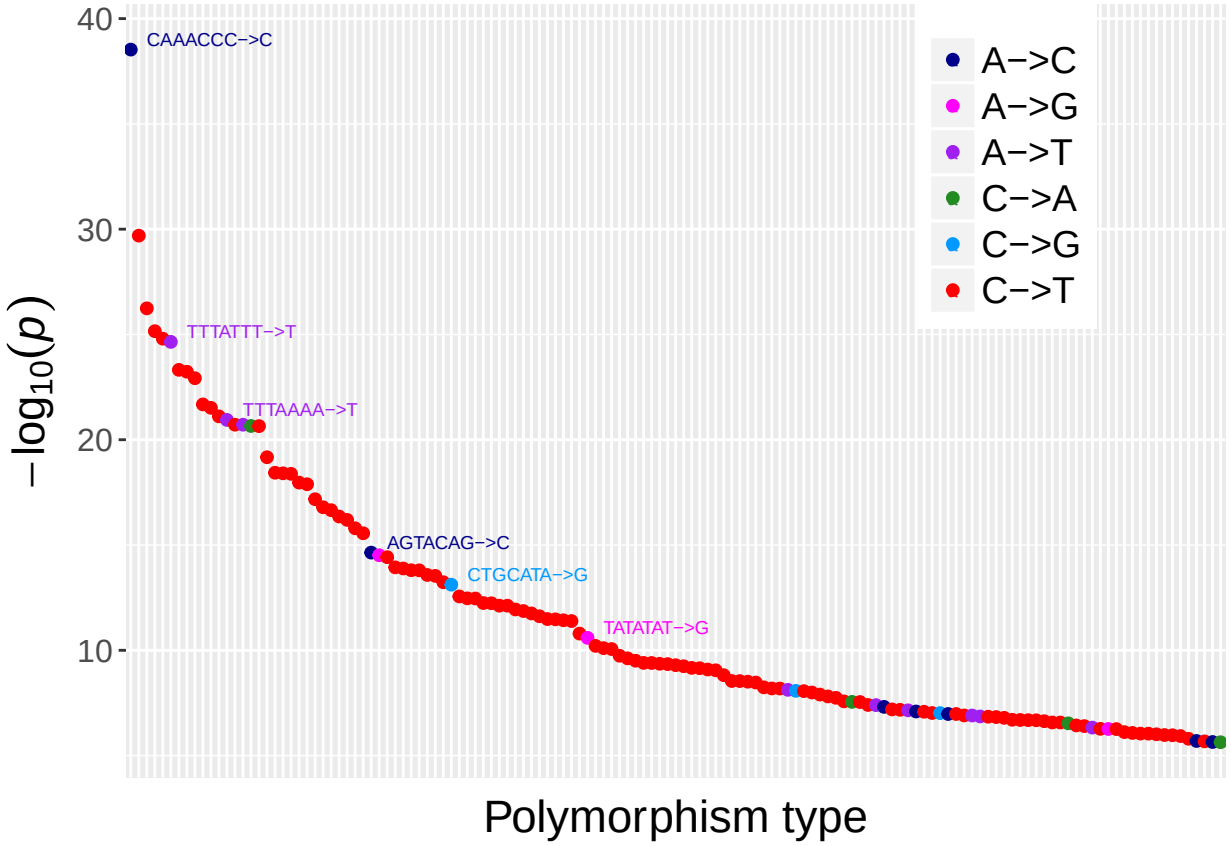
Context	AFR.Count	EUR.Count	EAS.Count	SAS.Count	p
TTAAA->T	20166	3211	4964	4742	7.300355e-40
GGCTT->T	9825	1789	3370	2834	4.790367e-25
TTATT->T	8644	1462	2549	1979	3.875805e-20
GGCCC->T	7208	1395	2470	2228	1.086694e-17
GGCTG->T	10307	1838	3390	3039	4.460545e-17
ACACC->G	5383	846	1232	1244	9.132616e-16

Context	AFR.Count	EUR.Count	EAS.Count	SAS.Count	p
TACAT->T	20674	3582	5097	5433	9.728627e-16
GGCTC->T	9323	1727	3078	2791	4.435647e-15
CCCCT->G	5389	1097	1826	1766	1.041212e-14
CCACC->G	7322	1175	1739	1807	6.428937e-13

Note that the most highly significant new 5mer is TTAAA->T, which corresponds to the 8th most significant 3mer, TAA->T. As we will see, the 7mer TTTAAAA->T is also one of the top significantly variable 7mers.

7mers

Now we move to the same analysis at the 7mer level, beginning with a homogeneity test plot:



The plot above shows heterogeneity test p values for the 137 7mers signifcant after bonferroni correction. We can ask the same question about these results as we did with the 5mers: which of these 7mers are results that we have not previously picked out from our 3mer analysis? Filtering these signals leaves 29 significant results, the top ten of which are shown below:

Table 3: 10 most significant 7mers not noted on a 3mer level

Context	AFR.Count	EUR.Count	EAS.Count	SAS.Count	p
CAAACCC->C	127	22	128	12	2.967288e-39
TTTATTT->T	2796	431	808	478	2.254142e-25
TTTAAAA->T	12011	1961	2939	2846	1.147367e-21
ATTAAAA->T	3773	496	857	808	1.938961e-21
AAACAAA->A	3108	446	766	578	2.220689e-21
AGTACAG->C	51	14	55	9	2.297518e-15
ACTAAAA->G	2187	513	833	705	3.060487e-15
CTGCATA->G	72	19	63	12	7.674004e-14
TATATAT->G	7093	1181	1710	1724	2.568108e-11
AGGCTTT->T	1174	177	442	339	3.396660e-09

Summary

The following table summarizes the numbers of significant results from this section.

Context Model	Number Significant	Number New
3mer	79	–
5mer	152	80
7mer	137	29

False Discovery Rate Corrections

All of the tests in the above section use the Bonferroni Correction, which is conservative even when hypothesis tests are positively correlated (as is most-likely the case here.) However, the Bonferroni correction is often criticized as being *too* conservative. For these reasons, it may be useful to apply other significance thresholds which account for the multiple testing burden.

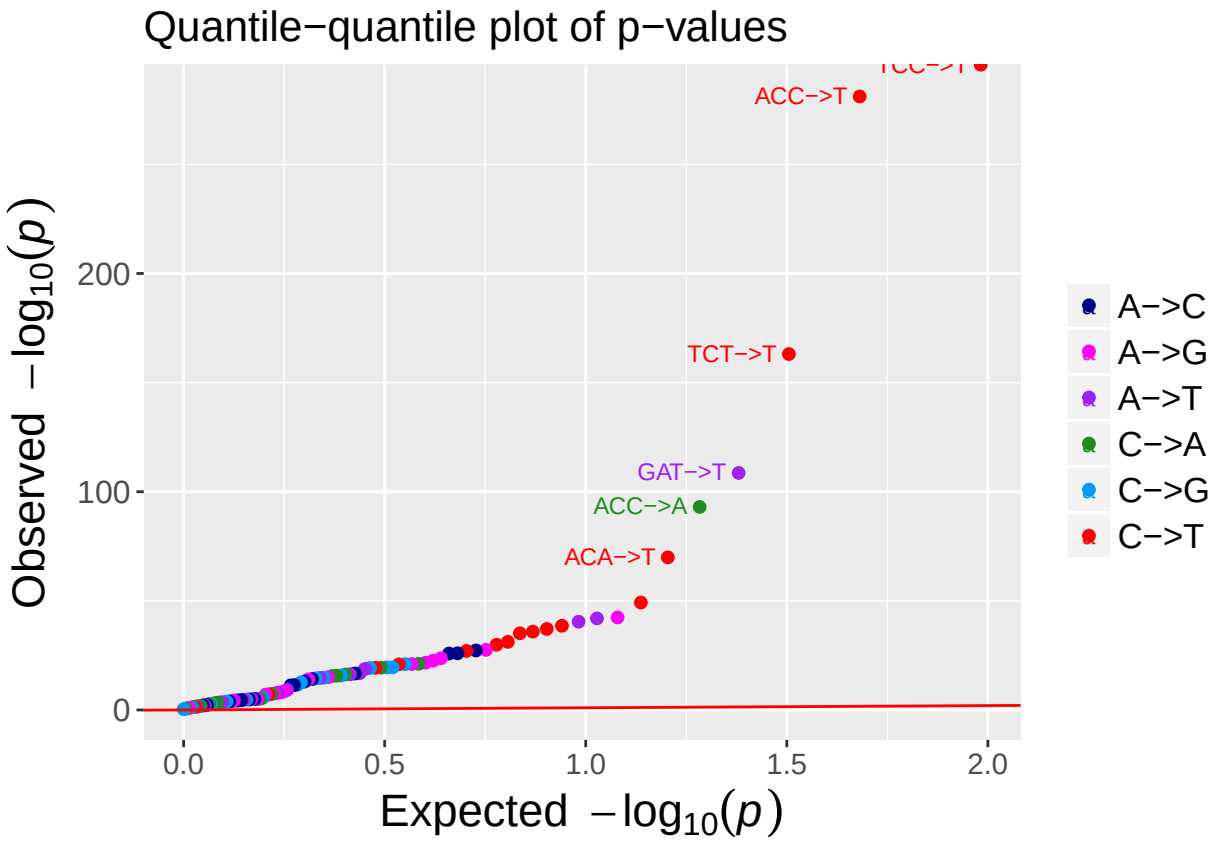
Methodology

Initially, I tried to use the qvalue package to perform false discovery rate analysis. However, this package proved difficult to use, since our p-values from our homogeneity tests don't follow a uniform [0,1] distribution (they range from 0-0.45). Instead, I decided to use the built-in R function, p.adjust(), which uses Benjamini-Hoochberg-Yekutieli. These methods should be acceptable even when the p-values are positively correlated. The following function, **fdr**, performs simple fdr analysis on an output dataframe from a chi-squared function.

```
fdr <- function(p.data){
  p.data <- p.data[complete.cases(p.data),]

  # This uses Benjamini-Hochberg-Yekutieli fdr
  p.data$fdr <- p.adjust(p.data$p, method = "fdr")

  # multiple hypothesis correction by holm
  p.data$holm <- p.adjust(p.data$p, method = "holm")

  alpha = 0.05/length(p.data$p)
  p.data <- p.data[complete.cases(p.data), ]

  n.sig <- c(length(p.data$p), sum(p.data$p < alpha), sum(p.data$holm< 0.05),
             sum(p.data$fdr< 0.1), sum(p.data$fdr< 0.05),
             sum(p.data$fdr< 0.01), sum(p.data$fdr< 0.001))

  names(n.sig) <- c("Total tests","Bonferroni", "Holm",
                    "FDR<0.1", "FDR<0.05",
                    "FDR<0.01", "FDR<0.001")

  return(list(n.sig, p.data))
}
```

I am additionally defining the funtion **qq.labels**, which takes in a p-value dataframe, a lab.lim, a title (default = "Quantile-quantile plot of p-values"), and the NoTCC arguement and returns a qq plot of all contexts, color-coded and labeled. In the following section, I will construct qq plots and run fdr analysis for each of the 3mer, 5mer, and 7mer models.

3mers

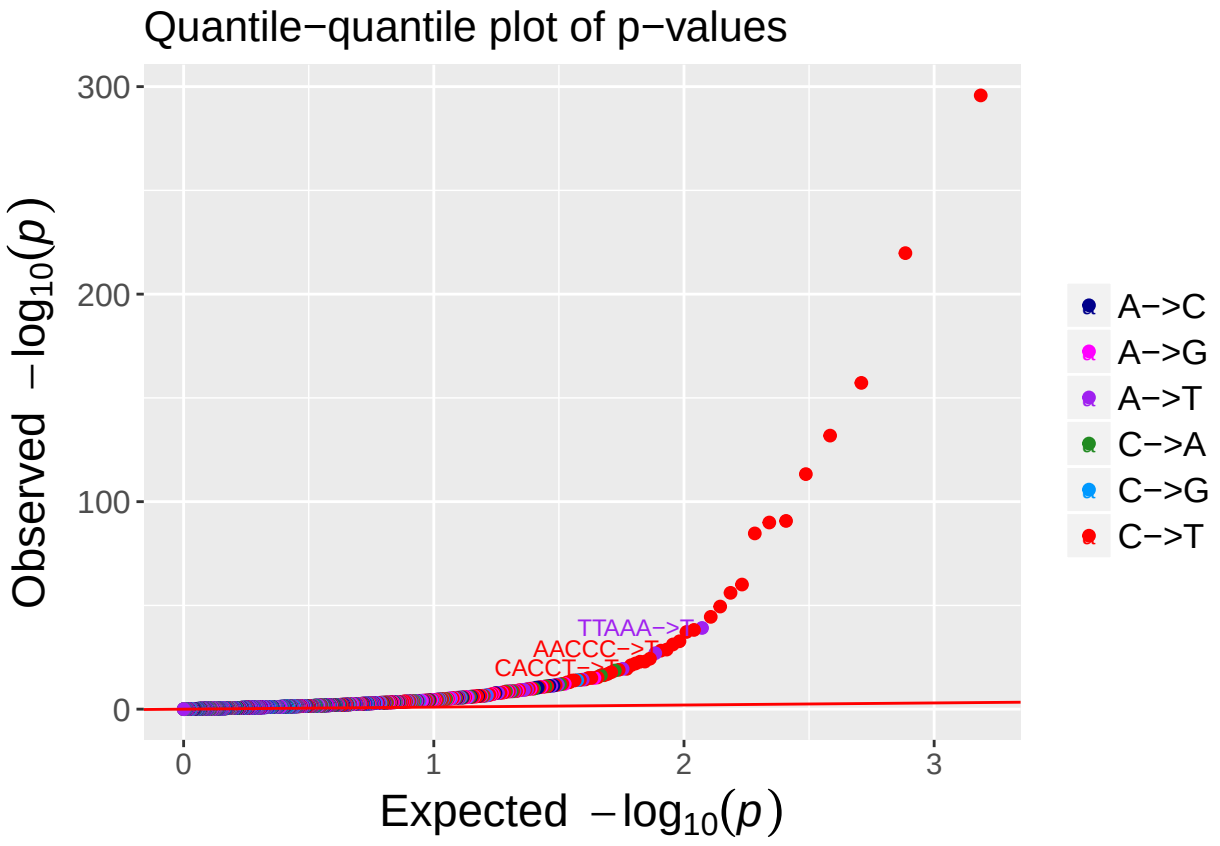



```
## [[1]]
## Total tests  Bonferroni        Holm     FDR<0.1    FDR<0.05    FDR<0.01
##          96          79          83          92          91          86
##   FDR<0.001
##          80
```

The qq plots shown above display relatively the same information as the p-value plots by context. However, it is worth noting that the observed p values, even at the lower end, are above expected p-value quantiles. This may suggest that in fact, every context is significant so that the null distribution of p values does not hold. More realistically, this appears to be an artifact of the fact that hypothesis tests set up as above are actually positively correlated (that is, a small p-value in one test probably increases the likelihood of a small p-value in another test).

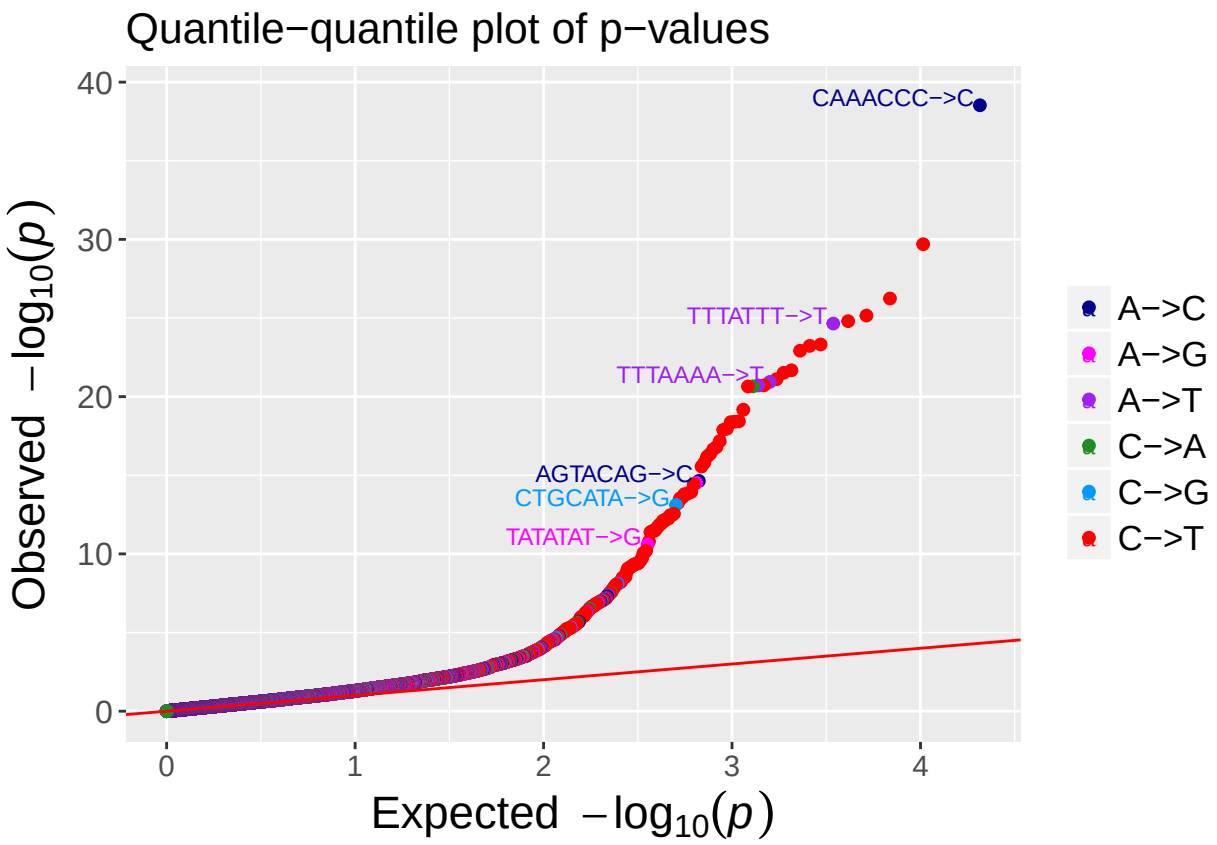
One possible solution to this problem would be to simulate null-distributed datasets to approximate an emprical distribution for expected p-value.

5mers




```
## [[1]]
## Total tests  Bonferroni        Holm     FDR<0.1    FDR<0.05    FDR<0.01
##        1535         152         154         512         405         278
##   FDR<0.001
##         179
```

7mers




```
## [[1]]
## Total tests  Bonferroni        Holm     FDR<0.1    FDR<0.05    FDR<0.01
##       20668         137         137         431         331         215
##   FDR<0.001
##         158
```

Ordered p-values

Methodology

The following function, **ordered.p**, returns a p-value dataframe with p calculated based on the methods from Harris and Pritchard, 2017. This method is proven to give less-significant results, but helps partially combat the problem of positive correlation between p values using our original methods.

```
ordered.p <- function(pdata){
  #preprocess data to order and remove nas
  pdata <- pdata[complete.cases(pdata$p),]
  myorder <- order(pdata$p)
  n.muts <- length(pdata$p)

  p.ordered <- rep(0, n.muts)

  #set largest p-value
  j <- myorder[n.muts]
  p.ordered[j] <- pdata$p[j]

  #initialize not mutated counts based on this lowest p-value mutation
  not.mut <- c(pdata$AFR.Count[j], pdata$EUR.Count[j],
               pdata$EAS.Count[j], pdata$SAS.Count[j])

  for (i in n.muts:1){
    j <- myorder[i]
    mut <-c(pdata$AFR.Count[j], pdata$EUR.Count[j],
            pdata$EAS.Count[j], pdata$SAS.Count[j])
    data <- cbind(mut, not.mut)
    p.ordered[j] <- chisq.test(data)$p.value

    #add these mutations to the not.mutated counts for future tests
    not.mut <- not.mut + mut
  }
  pdata$p <- p.ordered
  return(pdata)
}
```

In the following sections, I will repeat all of the above analyses for fourway homogeneity test in terms of ordered p value.

3mers

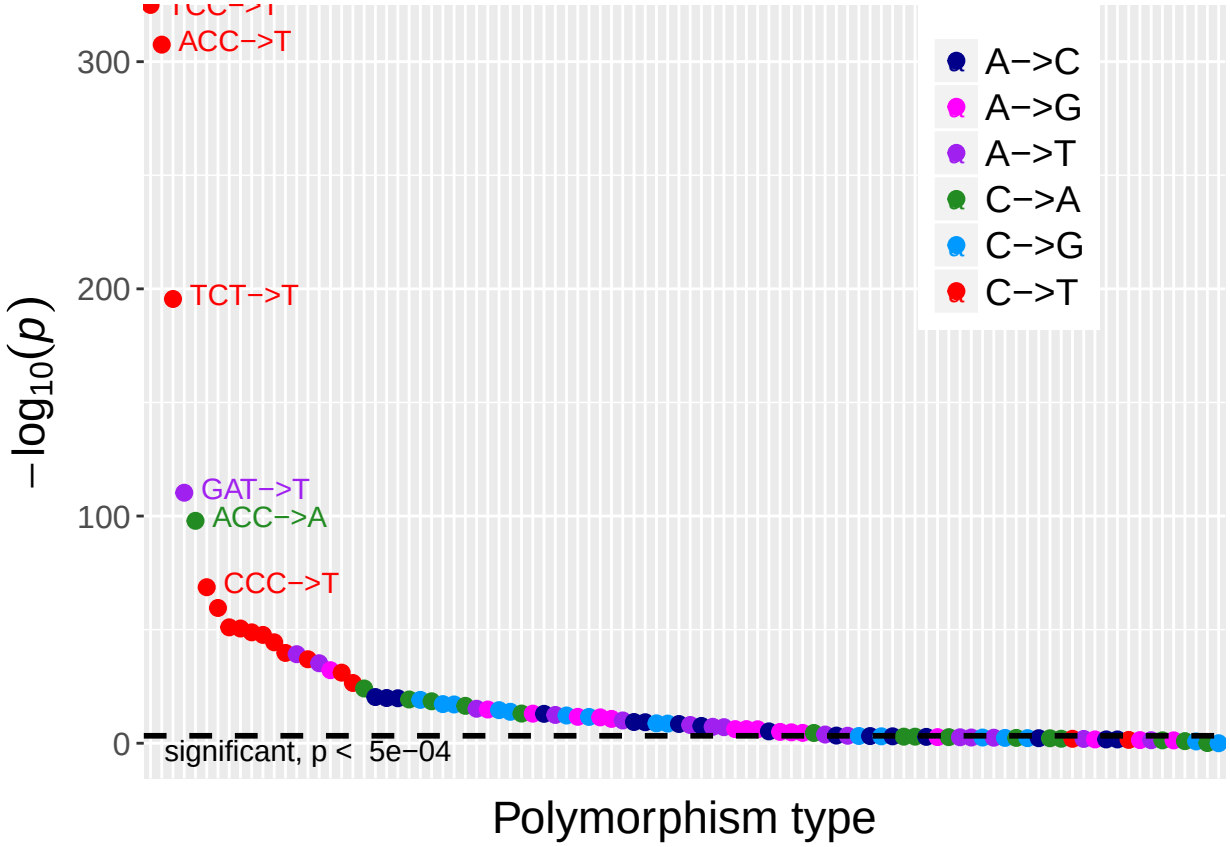
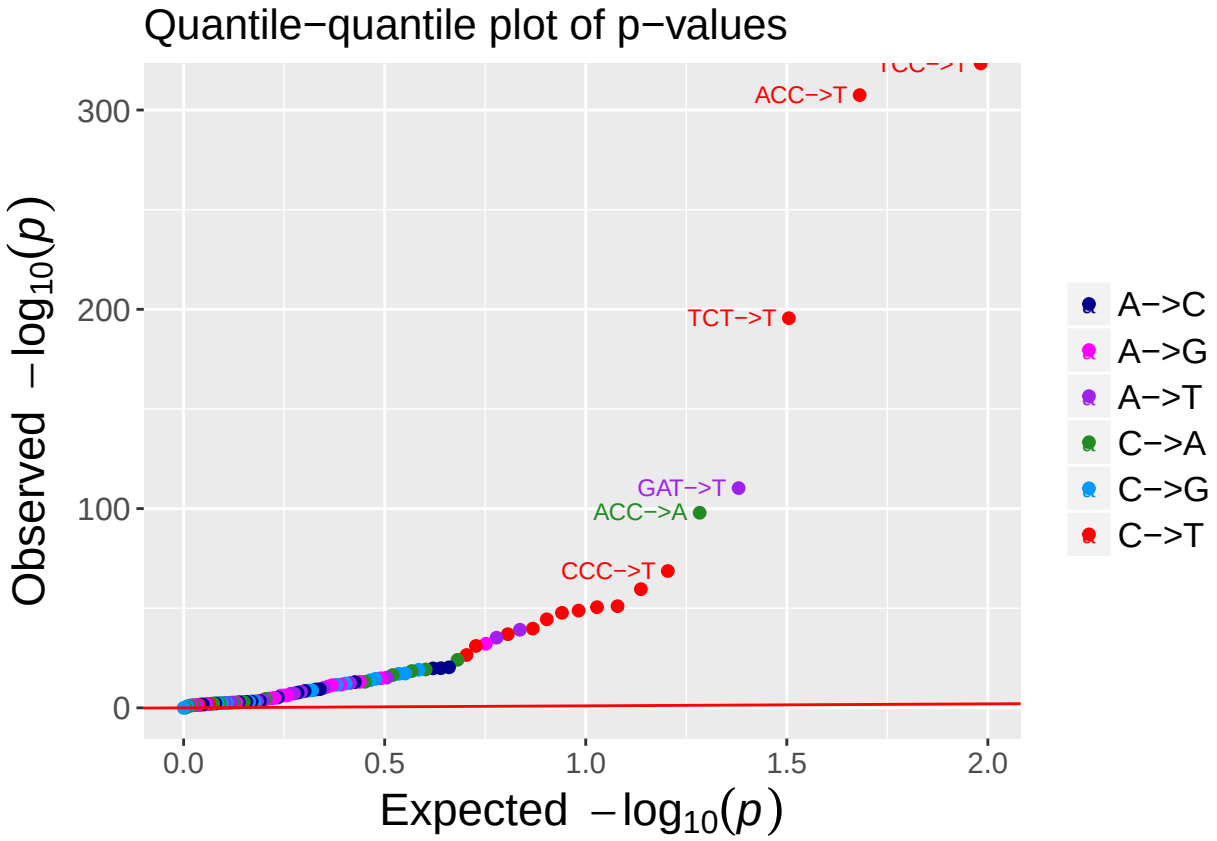


Table 5: 10 most significant 3mers using ordered p value correction

Context	X1mer	AFR.Count	EUR.Count	EAS.Count	SAS.Count	p
TCC->T	C->T	129676	37083	36252	43951	0.000000e+00
ACC->T	C->T	135088	29880	35002	40890	3.000000e-308
TCT->T	C->T	144254	30975	39522	43187	2.868119e-196
GAT->T	A->T	47152	9181	15883	15054	5.361344e-111
ACC->A	C->A	76646	14684	24481	23819	1.248600e-98
CCC->T	C->T	133143	26025	35129	38511	2.154370e-69
ACA->T	C->T	227803	39876	58806	62287	2.904922e-60
TCA->T	C->T	134514	26126	36472	38964	9.743309e-52
ACT->T	C->T	157475	29610	41237	44484	3.142140e-51
TCG->T	C->T	202752	37854	58472	60351	1.556530e-49

Notice that, as before, GAT->T and ACC->A are the 4th and 5th most significant results. Meanwhile, ACA->T, the third highly significant signal from earlier, is moved from the 6th to the 9th place in terms of significance, and no longer sticks out from the remaining mutation types as it once did. Certain mutations (for example TAA->T) have dropped in significance notably (from 8th to 17th), while C->T mutations seem to be featured much more prominently among the most significant polymorphism types.




```
## [[1]]
## Total tests  Bonferroni        Holm     FDR<0.1    FDR<0.05    FDR<0.01
##          96          63          70          92          89          81
##   FDR<0.001
##          64
```

5mers

```
## [1] 156
## [1] 78
```

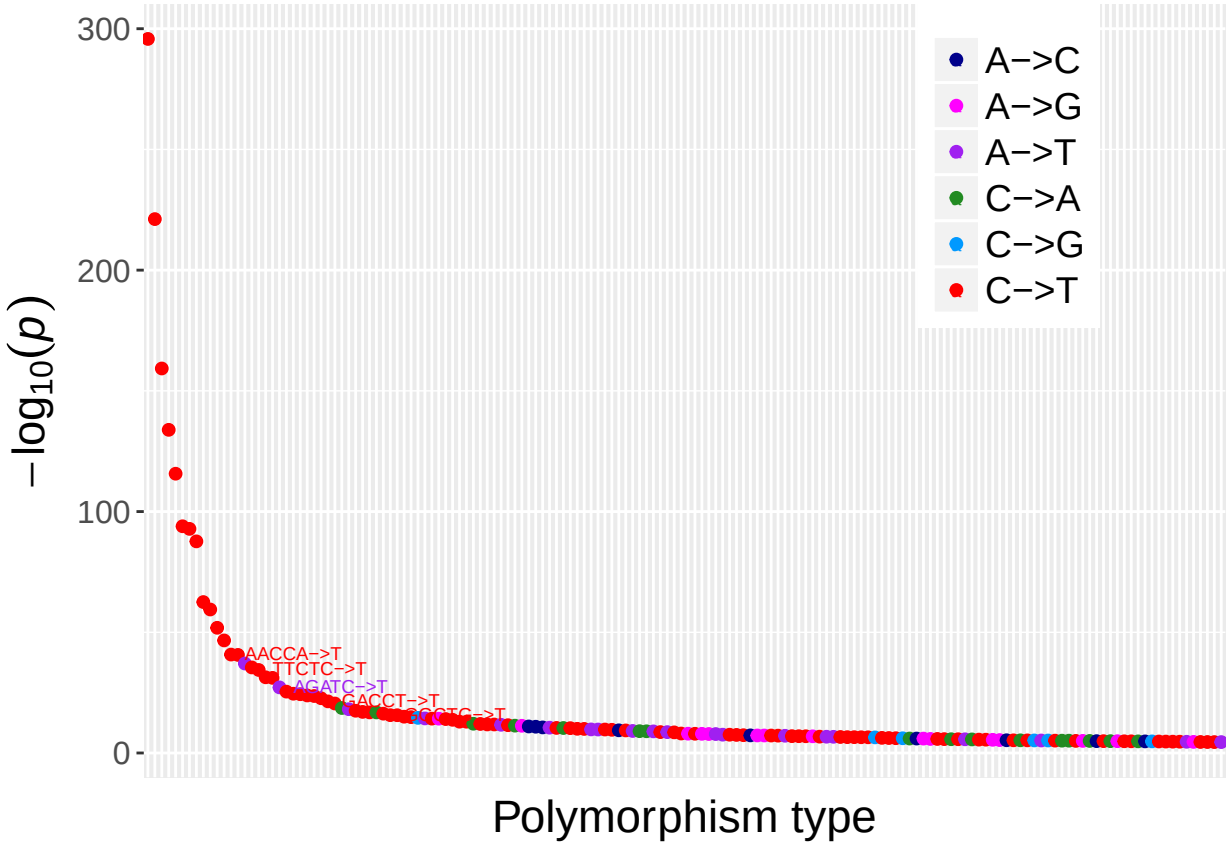
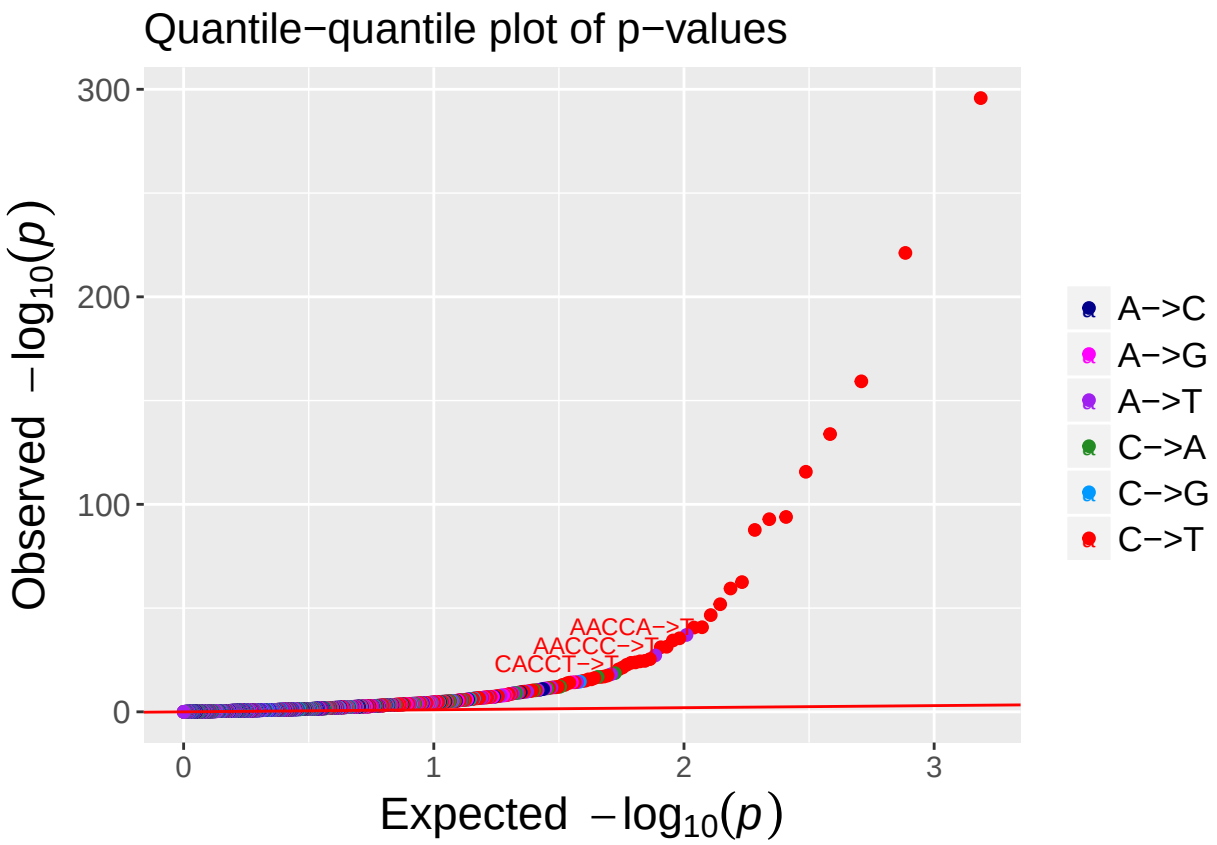


Table 6: 10 most significant new 5mers using ordered p value correction

Context	AFR.Count	EUR.Count	EAS.Count	SAS.Count	p
TTAAA->T	20166	3211	4964	4742	8.738995e-38
GGCTT->T	9825	1789	3370	2834	2.470290e-24
TTATT->T	8644	1462	2549	1979	7.283480e-19
GGCCC->T	7208	1395	2470	2228	9.483102e-18
GGCTG->T	10307	1838	3390	3039	2.150870e-16
CCCCT->G	5389	1097	1826	1766	2.823631e-15
TACAT->T	20674	3582	5097	5433	5.861060e-15
ACACC->G	5383	846	1232	1244	6.023723e-15
GGCTC->T	9323	1727	3078	2791	8.716711e-15
GGCTA->T	5989	1099	2026	1747	1.489862e-12

Notice that, as before, TTAAA->T is the most signifcant new 5mer, with a p-value several orders of magnitude smaller than the other new significant 5mers. Again, many of the top results are present in a different order than in the original test.




```
## [[1]]
## Total tests  Bonferroni        Holm     FDR<0.1    FDR<0.05    FDR<0.01
##        1535         156         158         457         395         272
##   FDR<0.001
##         182
```

7mers

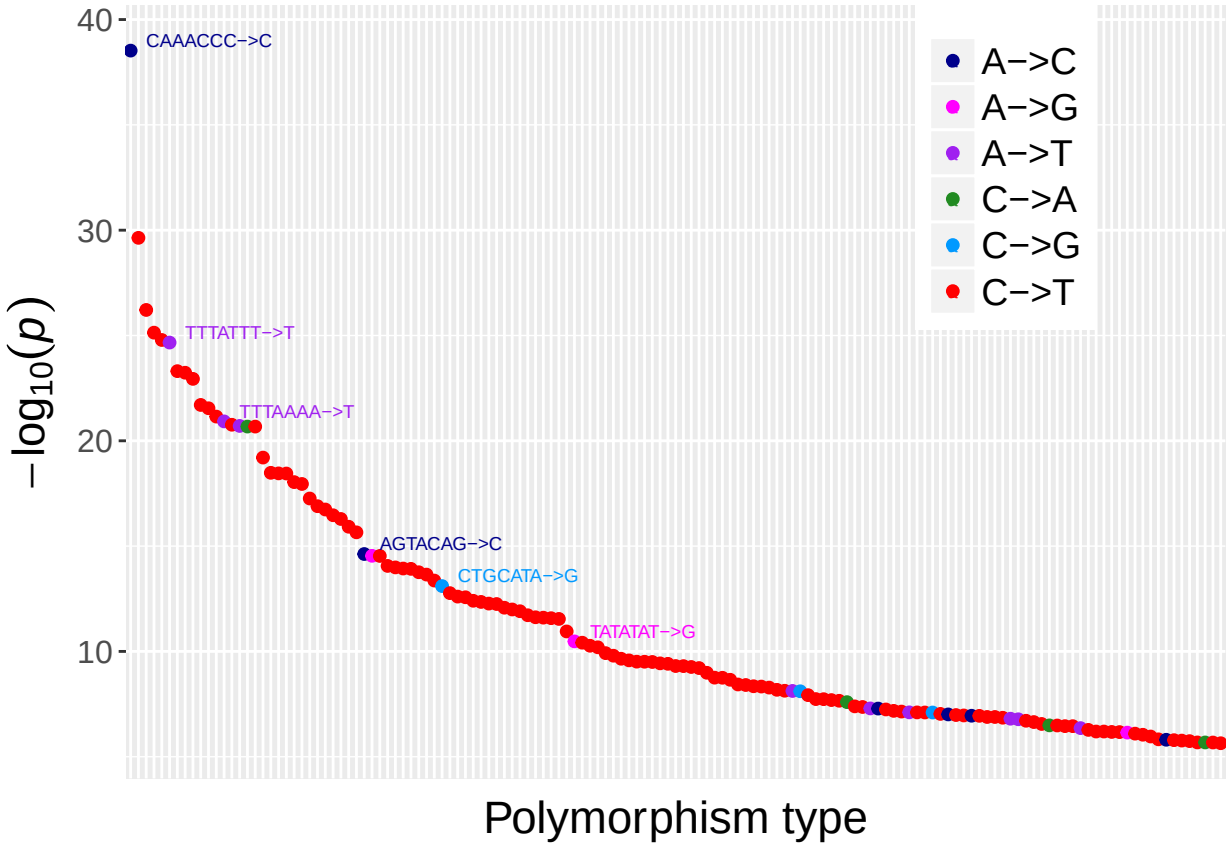
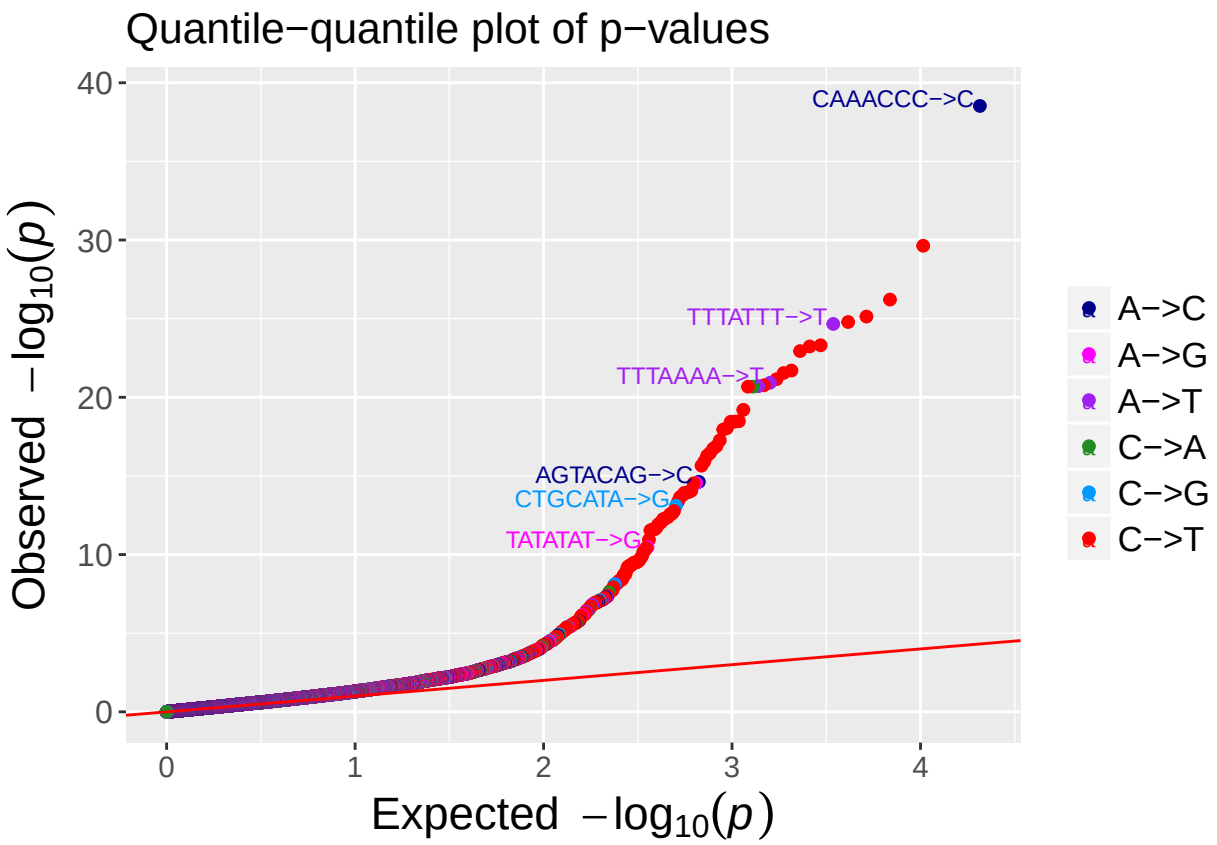


Table 7: 10 most significant new 7mers using ordered p value correction

Context	AFR.Count	EUR.Count	EAS.Count	SAS.Count	p
CAAACCC->C	127	22	128	12	2.984752e-39
TTTATTT->T	2796	431	808	478	2.166804e-25
TTTAAAA->T	12011	1961	2939	2846	1.199912e-21
ATTAAAA->T	3773	496	857	808	1.968521e-21
AAACAAA->A	3108	446	766	578	2.110224e-21
AGTACAG->C	51	14	55	9	2.375565e-15
ACTAAAA->G	2187	513	833	705	2.887438e-15
CTGCATA->G	72	19	63	12	7.903406e-14
TATATAT->G	7093	1181	1710	1724	3.338030e-11
AGGCTTT->T	1174	177	442	339	4.507439e-09

Notice that, for 7mers, the ordering of the top ten most significant results is entirely unchanged.




```
## [[1]]
## Total tests  Bonferroni        Holm     FDR<0.1    FDR<0.05    FDR<0.01
##       20668         141         141         454         343         220
##   FDR<0.001
##         164
```

Summary

The following table summarizes the numbers of significant results from this section.

Context Model	Number Significant	Number New
3mer	63	–
5mer	156	78
7mer	141	28

Notice that using the ordered p value calculation causes us to pick up far fewer significant 3mers, but slightly more 5mers and 7mers. Moreover, it seems that using p ordered on 3mers has a much greater effect than on 5mers and 7mers. This might be expected because, for the most significant 7mers or 5mers, our p-value calculations still include most of the data, however, for our highly significant 3mers, a much larger portion of the data is excluded.