# ABR6xBD21 Summary

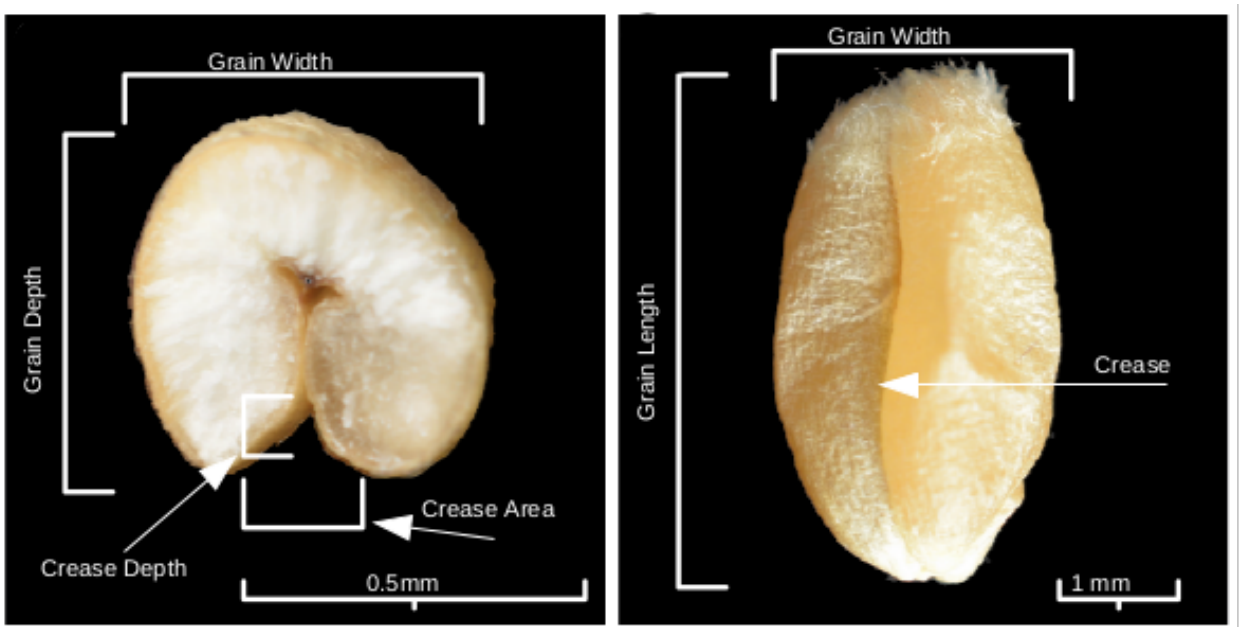
*Nathan Hughes*

*2 August 2017*

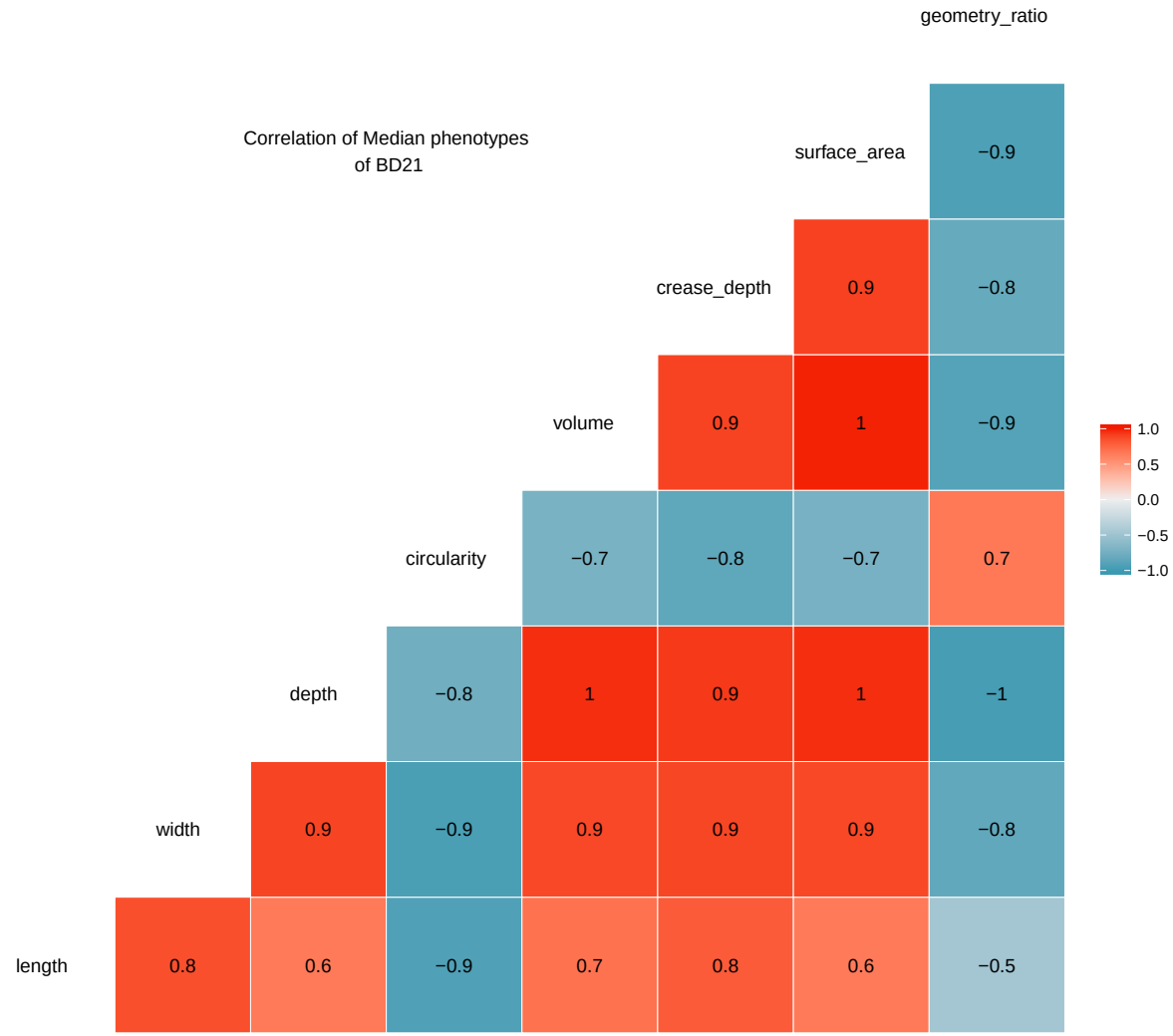
## Phenotypic Data

For clarification of Phenotypic discriptions, this is a labeled wheat grain

```
grid::grid.raster(img)
```



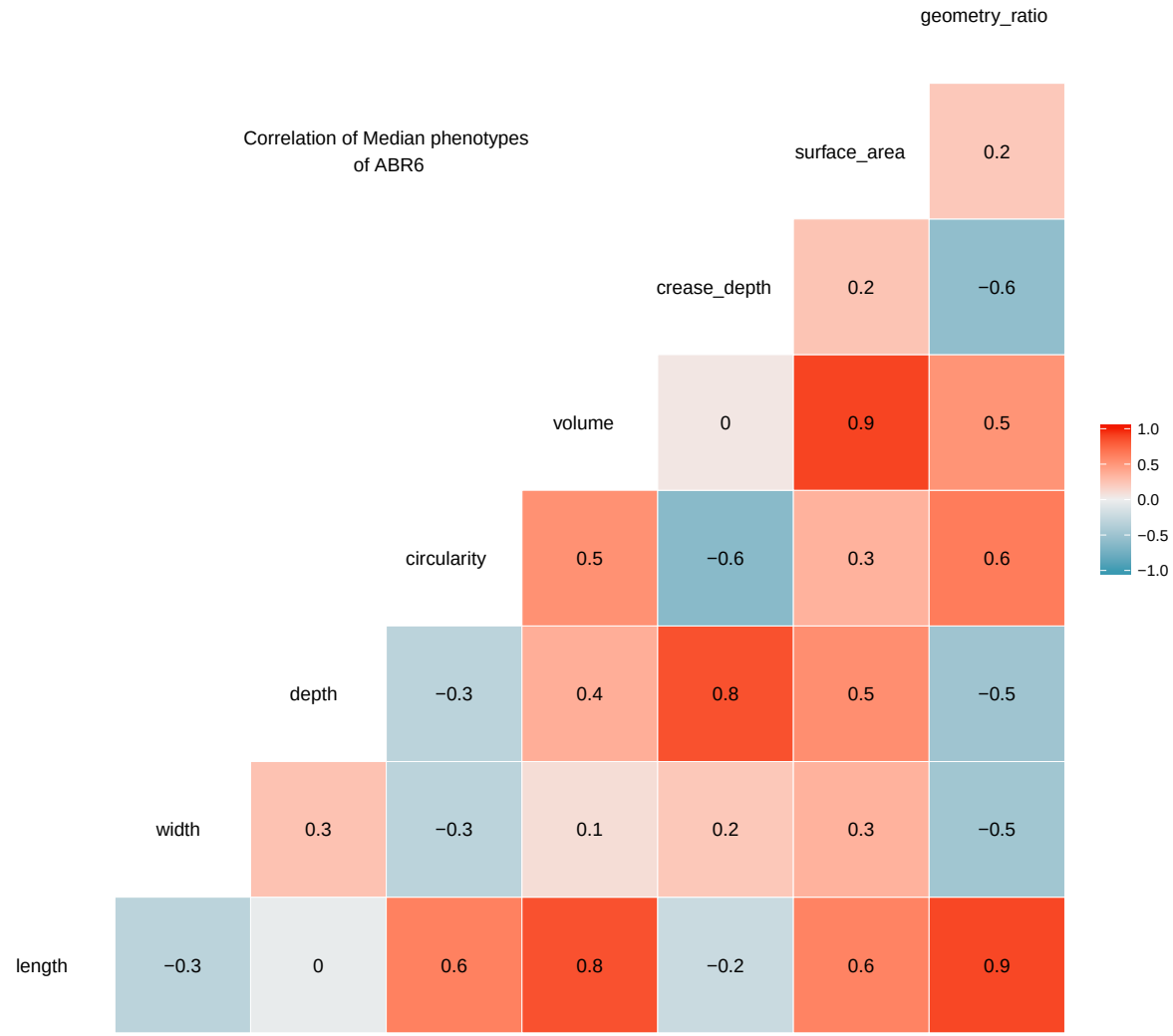
```
ggcorr(bd21[, c('length','width','depth','circularity', 'volume','crease_depth',
                'surface_area', 'geometry_ratio')],
       hjust = 0.6, label = TRUE, label_size = 4, size = 4 ) +
  annotate("Text", x = 3.5, y = 7, label = "Correlation of Median phenotypes\n of BD21")
```




```
ggsave('bd21.png', plot = last_plot())
```

```
## Saving 10 x 10 in image
```

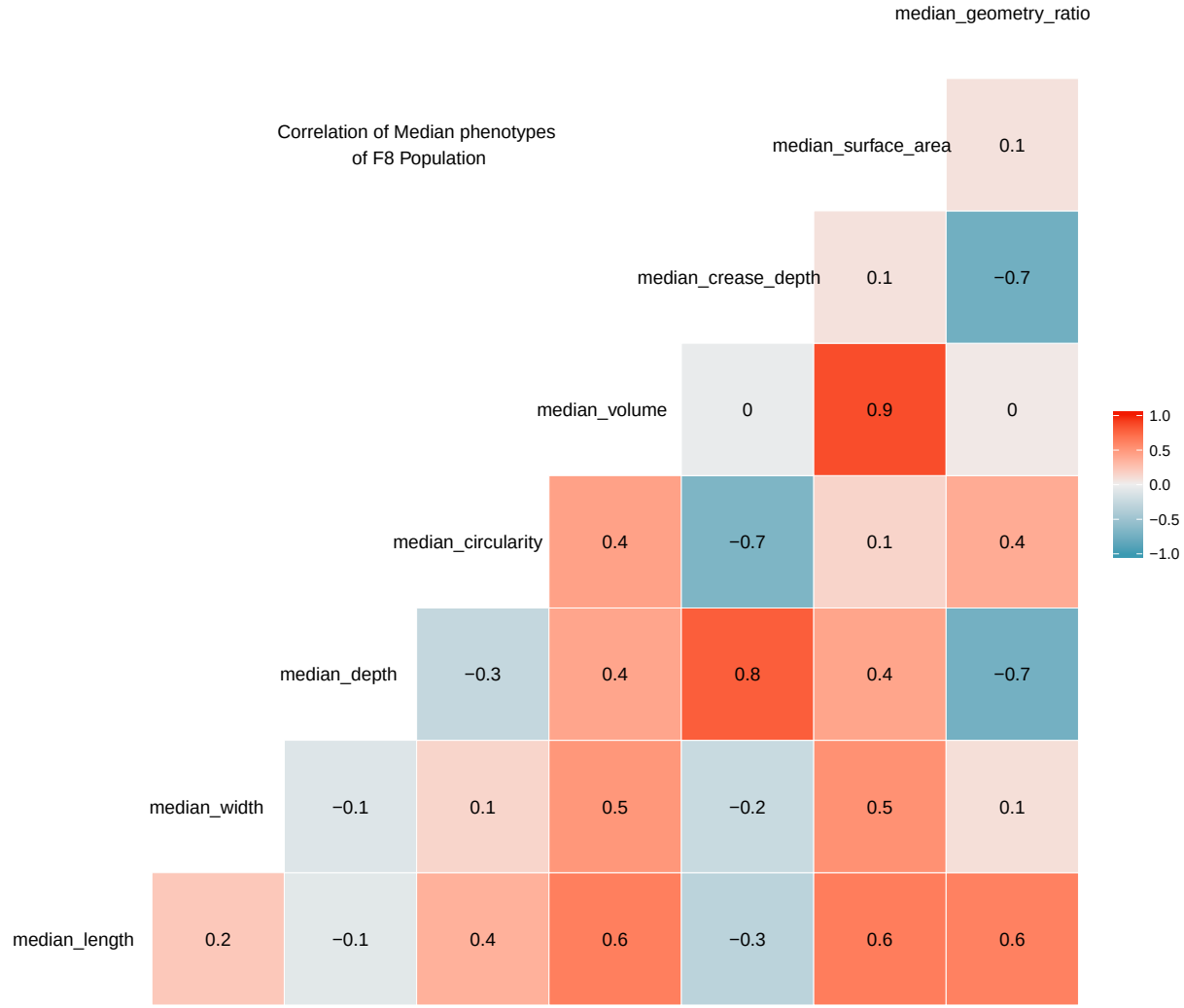
```
ggcorr(abr6[, c('length','width','depth','circularity', 'volume','crease_depth',
                'surface_area', 'geometry_ratio')],
       hjust = 0.6, label = TRUE, label_size = 4, size = 4 ) +
  annotate("Text", x = 3.5, y = 7, label = "Correlation of Median phenotypes\n of ABR6")
```




```
ggsave('abr6.png', plot = last_plot())
```

```
## Saving 10 x 10 in image
```

```
ggcorr(phendata[, 2:9], hjust = 0.6, label = TRUE, label_size = 4, size = 4 ) +
  annotate("Text", x = 3.5, y = 7,
           label = "Correlation of Median phenotypes\n of F8 Population")
```

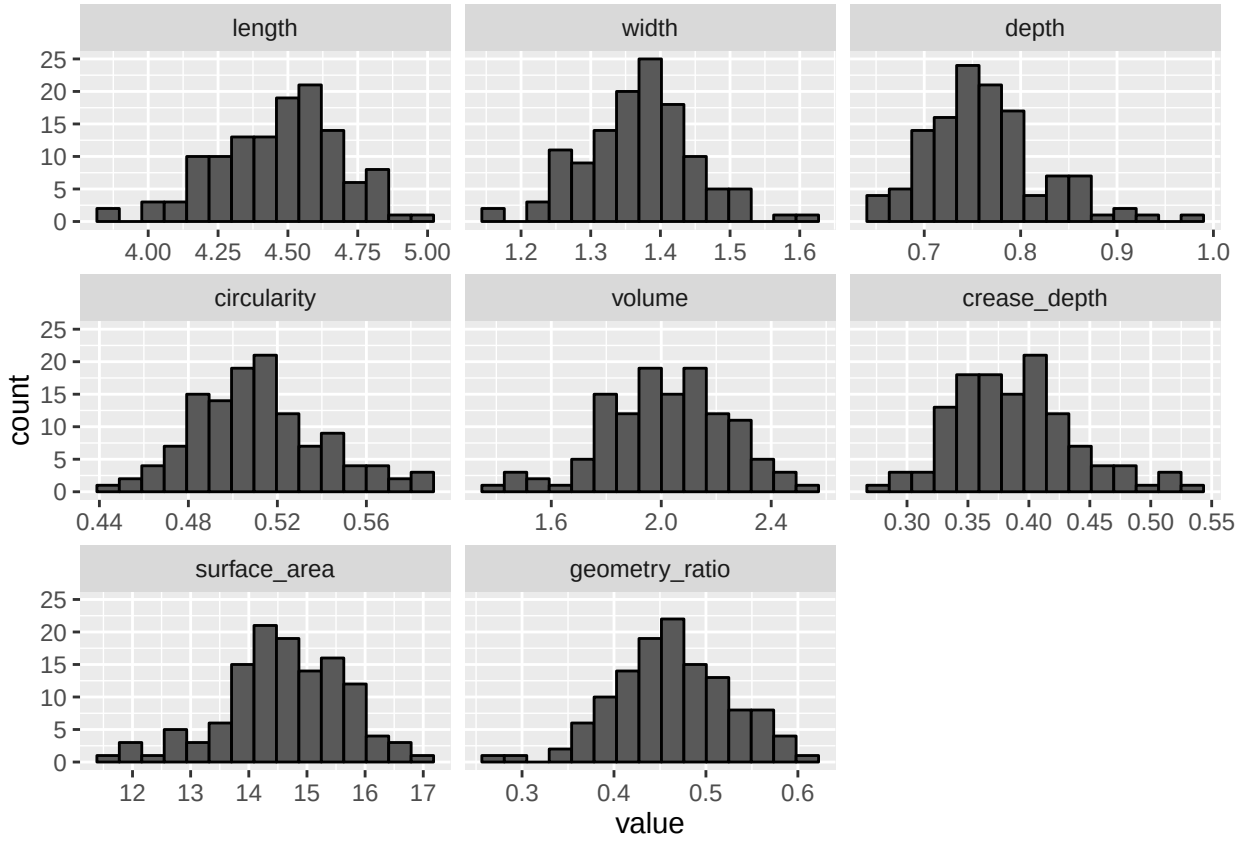



```
ggsave('cross-corr.png', plot = last_plot())
```

```
## Saving 10 x 10 in image
```

## Phenotypic Data (Cross Distributions)

```
colnames(phendata) <- substring(colnames(phendata), 8)
d <- melt(phendata[,c('length','width','depth','circularity', 'volume','crease_depth',
                      'surface_area', 'geometry_ratio')], id.vars = NULL)
ggplot(d,aes(x = value)) +
    facet_wrap(~variable,scales = "free_x") +
    geom_histogram(bins = 15, col = "black")
```



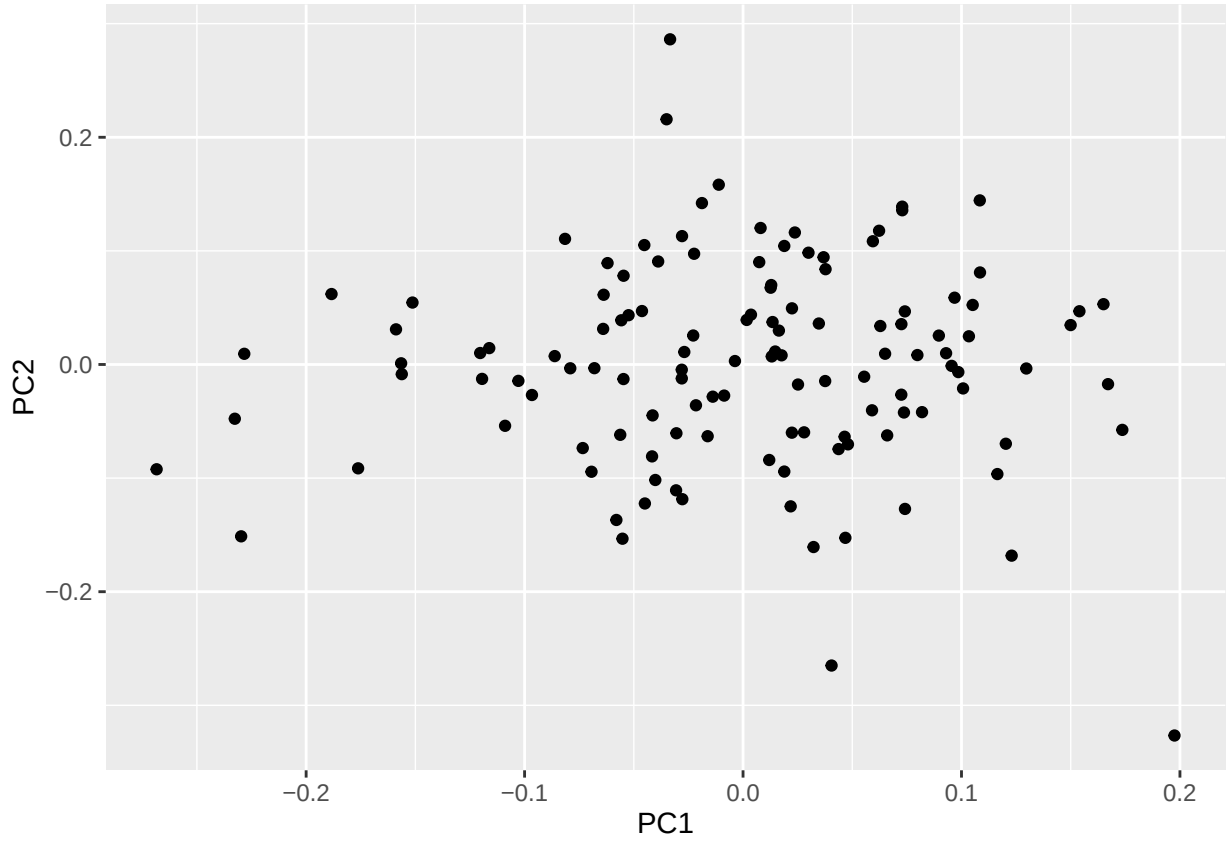

```
ggsave('hist.png', plot = last_plot())
```

```
## Saving 6.5 x 4.5 in image
```

## PCA of Phenotypes

```
d <- phendata[,c('length','width','depth','circularity', 'volume','crease_depth',
                      'surface_area')]


pcr <- prcomp(d)
autoplot(pcr)
```




```
ggsave('pca.png', plot = last_plot())
```

```
## Saving 6.5 x 4.5 in image
```

## Genotypic Data

```
png('geno-est-map.png')
plot(est.map(qtl))
dev.off()
```

```
## pdf
##   2
```

```
png('geno-image.png')
geno.image(qtl)
dev.off()
```

```
## pdf
##   2
```

- Few missing readings (white spots)
- Mostly complete and we can simulate and guess at missing data when working on more complex models

```
png('geno-rf-est.png')
plot.rf(est.rf(qtl))
dev.off()
```

```
## pdf
##   2
```

### Checking for QTLs

```
for (col in colnames(qtl$pheno)){

  s1 <- scanone(qtl, method = "hk", pheno.col = col)
  s2 <- scanone(qtl, method = "em", pheno.col = col)
  s3 <- scanone(qtl, method = "mr", pheno.col = col)
  operm.hk <- scanone(qtl, method = "hk", pheno.col = col, model = "normal",
                      n.perm = 1000, verbose = FALSE)

  lod_threshold <- summary(operm.hk)

  perms <- scanone(qtl, pheno.col = col, n.perm = 1000, method = "hk")

  summary(s1, perms=perms, pvalues=T)

    if (max(s1$lod) >= 2.5 && max(s2$lod) > 2.5 && max(s3$lod) > 2.5) {
    kable(summary(s1))

    png(sprintf('%s.png', col))
    par(mfrow = (c(2, 1)))

    plot(s1, s2, s3, main = " ", col = c('green', 'blue', 'orange'), ity = 2)

    title(sprintf("One-way scan QTL for: %s", col), outer=TRUE, line=-1)
    add.threshold(out=s1, perms = perms, col='red', ity=2, alpha=0.2)
    plot(perms)
    abline(v=quantile(perms,.95), col='red', ity=2)

   dev.off()
  }
}
```