easyVAF: an R package for integrated VAF analysis

Junxiao Hu[1,2]**, Vida Alami**[1]**, Yonghua Zhuang**[1,2]**, and Dexiang Gao**[1,2]

1 Biostatistics Shared Resource, University of Colorado Cancer Center, University of Colorado Anschutz Medical Campus, CO, USA **2** Department of Pediatrics, School of Medicine, University of Colorado Anschutz Medical Campus, CO, USA

DOI:

Software

- Review
- Repository
- Archive

Submitted:
Published:

Summary

Somatic sequence variants are associated with a cancer diagnosis, prognostic stratification, and treatment response. Variant allele frequency (VAF) is the percentage of sequence reads with a specific DNA variant over the read depth at that locus. Researchers often compare VAFs of targeted loci under different experimental conditions. Because this is a common analysis with burgeoning clinical applications, a set of tools to streamline the process is valuable. The R package "easyVAF" allows for parametric and non-parametric comparison of VAFs among multiple treatment groups. It is accompanied by an interactive R Shiny App for ease of use.

Statement of need

VAF is useful for assessing the unique and complex set of genomic variations. VAF analyses are widely conducted on sequencing data, particularly in cancer research, to identify and validate DNA variants. Though there is a rich ecosystem of bioinformatic software on various platforms, a library specifically targeted for downstream comparison of VAF has been missing from the R environment.

"easyVAF" is a flexible R package designed to support investigators in analyzing VAF across multiple treatment groups. A user-friendly Shiny app accompanies the package and includes interactive plots, customizable options for analysis, and a downloadable table of loci with comparison results.

When working with the package functions, we recommend the following VAF analysis workflow:

1. Start with exploratory plots for read depth, variant allele count, and VAF to check the quality of the sample (i.e., unexpected biological variability, batch effect, technical effect).
2. Conduct a statistical test to assess the variability of overall VAF distribution among the experiment samples (i.e., test if the heterogeneity of experiment samples is significant within each treatment group).
3. Perform the main comparison of VAFs, which runs as described below:
    a. For each locus, first conduct the goodness-of-fit test for binomial distribution (overdispersion).
    b. Select an appropriate method (model-based or non-parametric) according to the results from 3a for the VAF comparison among treatment groups.

The output is a table reporting the raw and adjusted p-values for each locus, as well as the estimated VAFs, the difference in VAFs, and the corresponding 95% confidence intervals (only available for two group comparisons).

Statistical Method

Quality of sample

The investigator might be concerned about the biological or experimental variability of the experiment samples before conducting the comparison analysis for the VAFs. The *QCchecking* function runs a likelihood ratio test to assess the variability of overall VAF distribution among samples using a linear mixed model, adjusting for potential non-independence of VAFs within the same chromosome. An alternative test using a linear model without random effects is provided as an alternative option if the dependence of VAFs within the same chromosome is not a concern. If sample heterogeneity is significant, we recommend increasing sample size or incorporating important biological covariates into analysis to increase statistical power and estimation accuracy.

Main comparison of VAFs among N groups

The main purpose of this software is to compare the VAFs of specific loci among multiple groups, given the variant allele counts and read depths from each sample. Without model assumptions, the data can be presented in two (1 or 0 indicates the presence or absence of a variant allele, respectively) by N (number of groups) contingency tables for each locus. In this software, the *VAFmain* function conducts Fisher's exact test (Fisher, 1970) as default to examine the significance of the difference in VAFs among the N groups. Fisher's exact test is set as the default since it is valid for all sample sizes and provides the exact p-values. The Pearson's Chi-squared test (Pearson, 1900), is also provided as an option, which generates comparable results to the exact test for a large sample size (i.e., large number of cell counts in the contingency table). Since Fisher's exact test may be computationally in-feasible for large sample sizes and the accuracy of the Chi-squared approximation increases with larger number of samples, the Chi-squared test maybe considered in this case.

Beta-binomial model for overdispersion

The main comparison using a contingency table assumes that all samples within one group have the same VAF for a specific locus. In practice, VAF can differ among samples within the same group due to biological variability. To account for this additional variability in VAF among samples, "easyVAF" implements a VAF model using a beta-binomial distribution.

The proposed beta-binomial model is formulated as follows: Let Y_{ij} denote the variant allele count for sample i in group j, and n_{ij} denote the read depth for sample i in group j, for i in $1, 2, ..., m$; and j in $1, 2, ..., N$. Then

$$Y_{ij}|P_{ij} \sim Binomial(n_{ij}, P_{ij}) \tag{1}$$

with probability mass function

$$f(y) = \binom{n_{ij}}{y} P_{ij}^{y}(1 - P_{ij})^{(n_{ij}-y)}. \tag{2}$$

And P_{ij} follows a Beta distribution

$$P_{ij} \sim Beta(a_j, b_j), \tag{3}$$

with probability density function

$$g(p) = \frac{p^{a_j-1}(1-p)^{b_j-1}}{B(a_j, b_j)}, \tag{4}$$

in which $B(a_j, b_j)$ denotes the beta function.

The probability mass function of the marginal beta-binomial distribution for Y_{ij} is then

$$h(y) = \binom{n_{ij}}{y} \frac{B(y + a_j, n_{ij} - y + b_j)}{B(a_j, b_j)}. \tag{5}$$

Let θ_j denote the mean VAF and ϕ_j denote the dispersion parameter for group j, then with reparameterization

$$\theta_j = \frac{a_j}{a_j + b_j}; \tag{6}$$

$$\phi_j = \frac{1}{a_j + b_j + 1}. \tag{7}$$

The likelihood ratio test is conducted to test the hypothesis

H_0: $\theta_j = \theta$ for all j,

H_1: θ_js are not all equal.

using the marginal beta-binomial likelihood.

Test for Binomial overdispersion

The package includes a *Tarone.test* function (Tarone, 1979) to assist with the choice of test for the main comparison. For each group, the Tarone test evaluates the goodness of fit of the binomial distribution, which is the optimal test against the beta-binomial alternative (Tarone, 1979). For group j, the test statistic Z_j is defined by the equation

$$Z_j = \frac{S - \sum_i n_{ij}}{\sqrt{(2 \sum_i n_{ij}(n_{ij} - 1))}}, \tag{8}$$

in which,

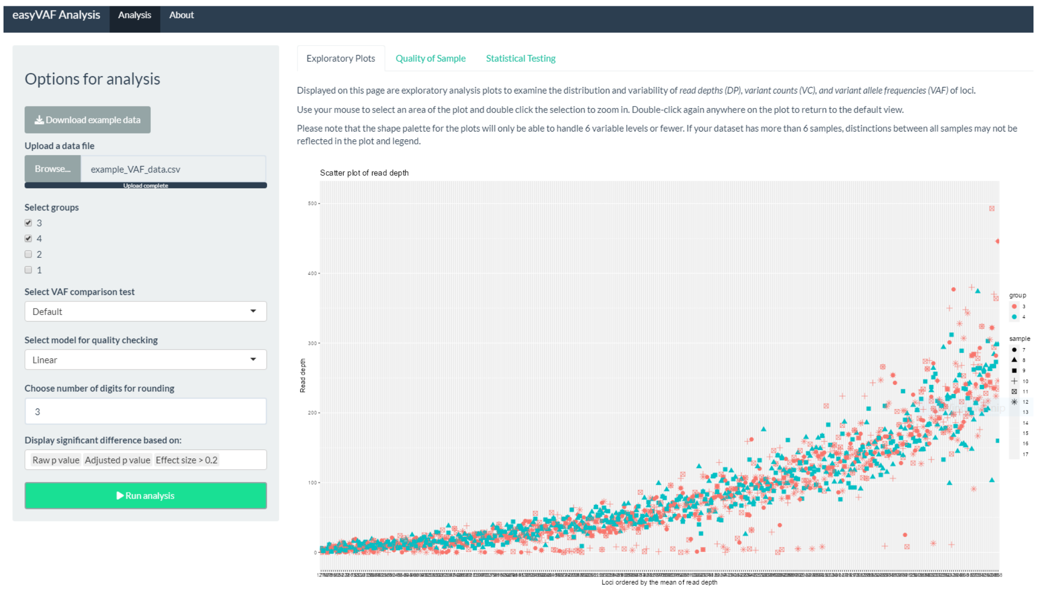
$$S = \sum_i \frac{(y_{ij} - n_{ij}\hat{p}_j)^2}{\hat{p}_j(1 - \hat{p}_j)}, \tag{9}$$

$$\hat{p}_j = \frac{\sum_i y_{ij}}{\sum_i n_{ij}}. \tag{10}$$

And Z_j follows an asymptotic standard normal distribution under the null hypothesis with zero dispersion.

Multiple comparison

VAF analysis is typically performed for multiple loci. Multiple comparison adjustment is implemented to control the false discover rate using Benjamini & Hochberg's method (Benjamini & Hochberg, 1995).



Upload a data file: example_VAF_data.csv

Select groups: [✓ 3] [✓ 4] 2 1

Select VAF comparison test: Default

Select model for quality checking: Linear

Choose number of digits for rounding: 3

Display significant difference based on: Raw p value Adjusted p value Effect size > 0.2

Displayed on this page are exploratory analysis plots to examine the distribution and variability of *read depths (DP), variant counts (VC), and variant allele frequencies (VAF)* of loci.

Use your mouse to select an area of the plot and double click the selection to zoom in. Double-click again anywhere on the plot to return to the default view.

Please note that the shape palette for the plots will only be able to handle 6 variable levels or fewer. If your dataset has more than 6 samples, distinctions between all samples may not be reflected in the plot and legend.

Figure 1: An example of the read depth exploratory plot and layout of the easyVAFanalysis Shiny app.

The R package

The R package was built under R version 4.1.0 (2021-05-18) (R Core Team, 2021). The package has the following dependencies: lme4 (Bates et al., 2015), aod (Lesnoff et al., 2012), dplyr (Wickham et al., 2021), tibble (Müller & Wickham, 2021), and ggplot2 (Wickham, 2016).

The Shiny app

The accompanying Shiny app can be found at https://vapps-uccc-bs.shinyapps.io/easyVAFanalysis/.

In addition to the "easyVAF" package dependencies, the app was built with: shiny (Chang et al., 2017), shinythemes (Chang et al., 2018), DT (Xie et al., 2022), ggiraph (Gohel & Skintzos, 2016), and cowplot (Wilke, 2020).

Exploratory plots

In addition to deploying the functions from the package, the app includes exploratory plots that allow the user to quickly view the distribution of their data. These are scatter plots of the loci, ordered on the x-axis by the mean of read depth, for read depth, variant count, and VAF (computed from the previous two values). Sections of the plots can be enlarged by drawing a box on a target area and double-clicking the selection.

Example

Data requirement

For illustrative purposes, an example data set, which is derived by resampling (with replacement) from a real DNA sequencing application with pseudo locus ID and chrome ID, is included in the package. The *VAFmain* function requires a data frame input with the columns named exactly as below,

```
data(VAF)
names(VAF)
```

[1] "Locus" "vc" "dp" "chrom" "sample" "group"

```
print(kable(head(VAF), row.names=F,
            caption="VAF data example"))
```

Table 1: VAF data example

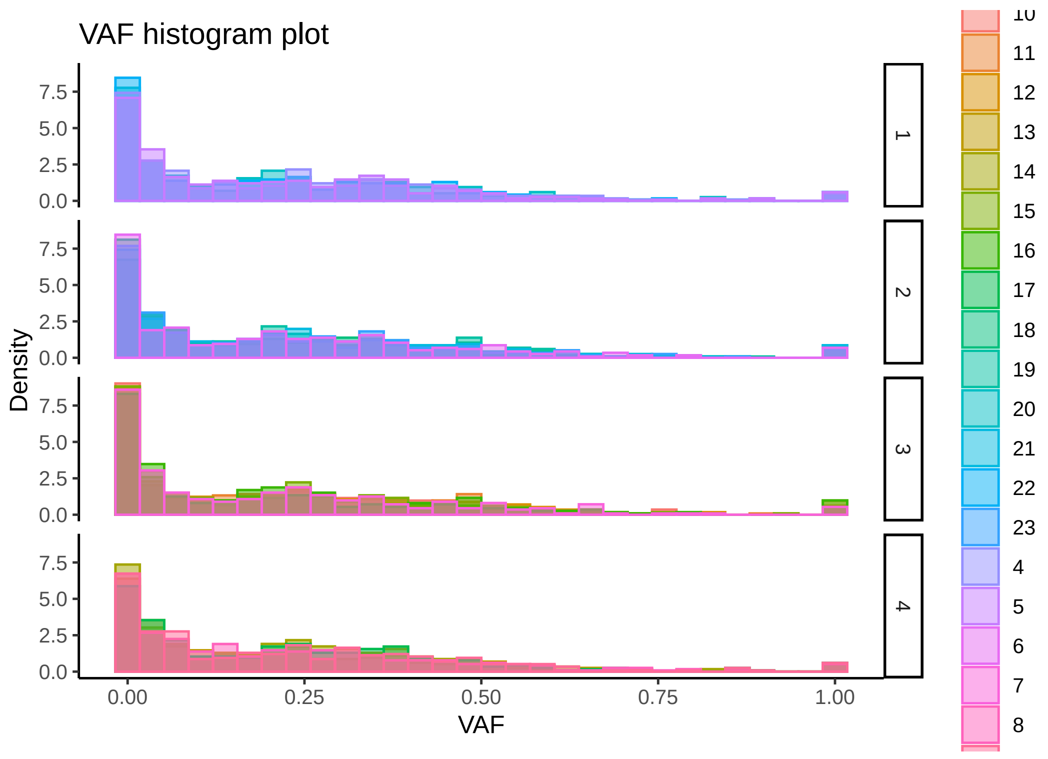
Locus	vc	dp	chrom	sample	group
38	0	37	1	10	3
38	1	36	1	11	3
38	0	46	1	12	3
38	0	37	1	15	3
38	1	36	1	16	3
38	0	37	1	7	3

where Locus contains the locus ID, vc contains variant count, dp contains read depth, chrom contains the chrom ID, sample contains the sample ID, and group contains group ID.

Quality test results

In the "Quality of Sample" tab, view the distribution of VAF by group and the likelihood ratio test results. The R code and results using the package directly are shown below.

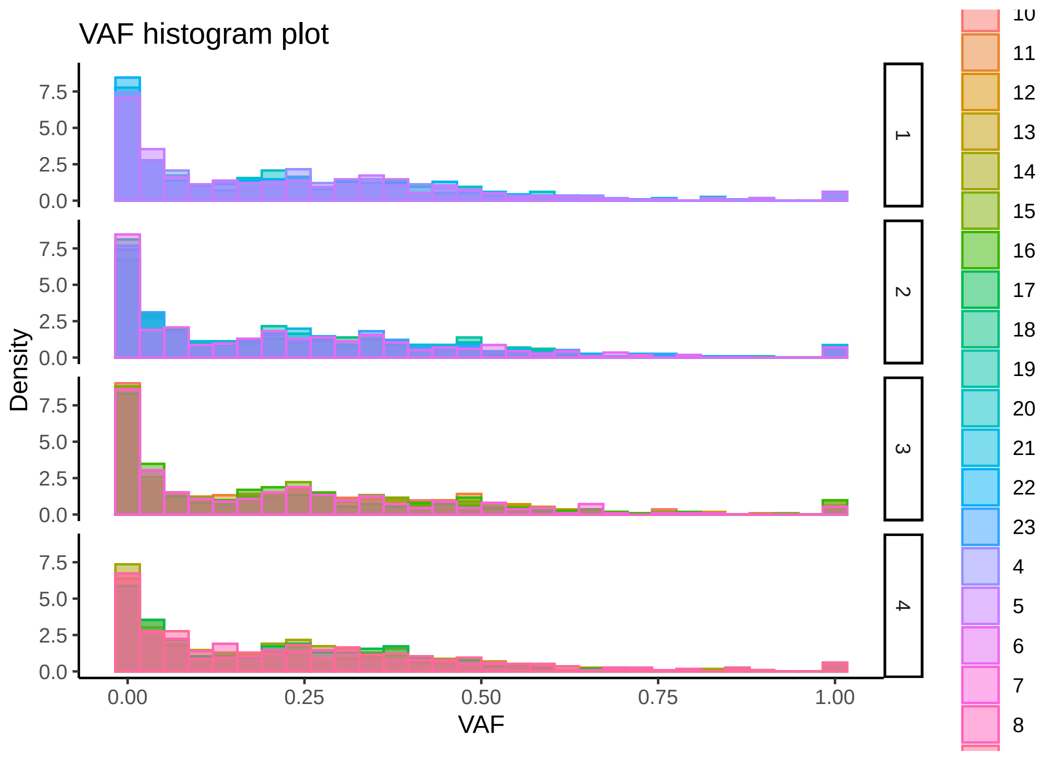
```
rslt <- QCchecking(data=VAF, method="lm")
```



```
print(kable(rslt))
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
6643	347.6179	NA	NA	NA	NA
6659	347.8898	-16	-0.2719647	0.3248289	0.9946621

```r
rslt <- QCchecking(data=VAF, method="lmer")
```



```r
print(kable(rslt))
```

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
m0	6	-1407.286	-1366.460	709.6430	-1419.286	NA	NA	NA
m1	22	-1381.044	-1231.349	712.5219	-1425.044	5.757834	16	0.9905067

Analysis results

For analyses on exactly two groups, the app displays a three-panel graph of the loci comparing different selection methods in the "Statistical testing" tab. The user can zoom in and out, hover over the points in the graph to see the locus ID, click on individual points or use the lasso tool to select points, and see selected loci in the corresponding results table. For three or more groups, only the loci table is available. The R code and results using the package directly are shown below.

```r
data(VAF)

#4 groups
groups <- unique(VAF$group)[c(1:4)]
rslt <- VAFmain(data=VAF, groups=groups)
rslt$P.value <- as.numeric(rslt$P.value)
toploci <- head(rslt[order(as.numeric(rslt$P.value)),
                     c("ID",
                       "P.value",
                       "Overdispersion",
                       "p.adjust",
                       "sig.diff.fdr")], n=10)

print(kable(toploci, row.names=F,
            caption="Top 10 significantly different loci,
                     multiple group comparison",
            digits=3))
```

Table 4: Top 10 significantly different loci, multiple group comparison

ID	P.value	Overdispersion	p.adjust	sig.diff.fdr
329	0.001	Yes	0.208	No difference
331	0.002	Yes	0.208	No difference
177	0.002	Yes	0.208	No difference
325	0.003	Yes	0.208	No difference
273	0.003	Yes	0.208	No difference
39	0.006	Yes	0.223	No difference
332	0.006	Yes	0.223	No difference
26	0.007	Yes	0.223	No difference
303	0.008	Yes	0.223	No difference
267	0.008	Yes	0.223	No difference

```
#2 groups
groups <- unique(VAF$group)[c(1:2)]
rslt <- VAFmain(data=VAF, groups=groups)
rslt$P.value <- as.numeric(rslt$P.value)
toploci <- head(rslt[order(as.numeric(rslt$P.value)),
                     c("ID",
                       "Effect.size",
                       "95% CI",
                       "p.adjust",
                       "sig.diff.fdr",
                       "Change.direction")], n=10)

print(kable(toploci, row.names=F,
            caption="Top 10 significantly different loci, two groups",
            digits=3))
```

Table 5: Top 10 significantly different loci, two groups

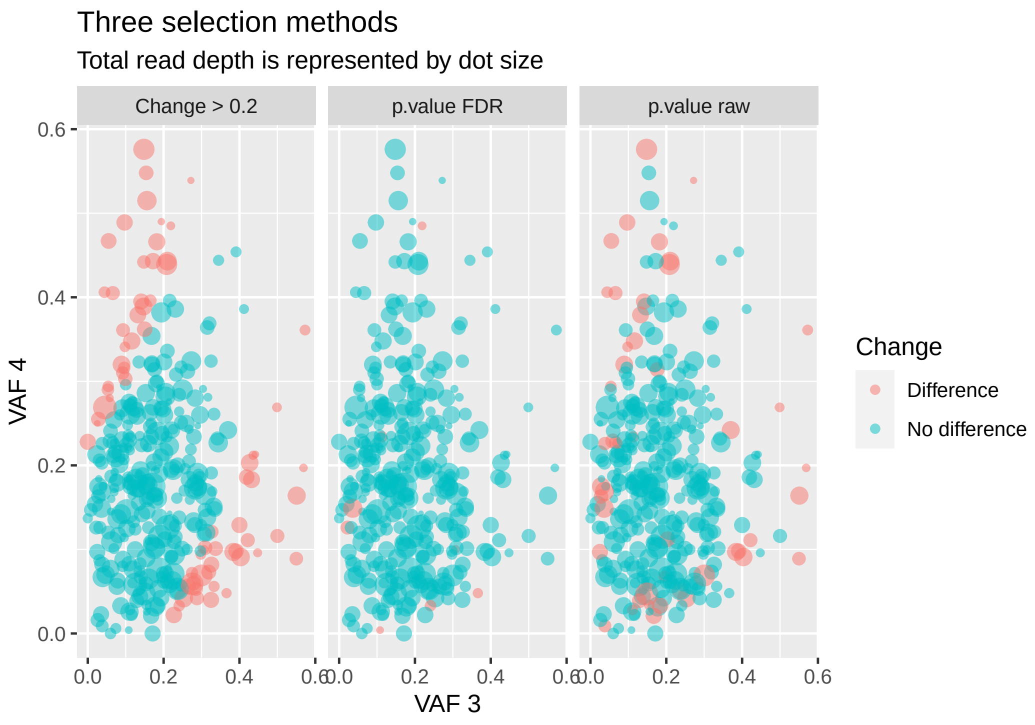
ID	Effect.size	95% CI	p.adjust	sig.diff.fdr	Change.direction
23	Diff in prop = 0.319	(0.237, 0.401)	0.000	Difference	Group1 > Group2
54	Diff in prop = 0.208	(0.143, 0.273)	0.000	Difference	Group1 > Group2
271	Diff in prop = -0.104	(-0.144, -0.064)	0.000	Difference	Group1 < Group2
320	Diff in prop = -0.266	(-0.358, -0.173)	0.000	Difference	Group1 < Group2
178	Diff in prop = 0.207	(0.125, 0.289)	0.000	Difference	Group1 > Group2
123	Diff in prop = 0.105	(0.029, 0.18)	0.001	Difference	Group1 > Group2
273	Diff in prop = -0.113	(-0.154, -0.073)	0.035	Difference	Group1 < Group2
26	Diff in prop = -0.121	(-0.188, -0.054)	0.035	Difference	Group1 < Group2
287	Diff in prop = 0.387	(0.313, 0.46)	0.168	No difference	Group1 > Group2
304	Diff in prop = 0.372	(0.281, 0.464)	0.176	No difference	Group1 > Group2

```r
dat1 <- rslt[,c("Read.Depth.3",
                "VAF.3",
                "Read.Depth.4",
                "VAF.4",
                "sig.diff")]
dat1$method <- "p.value raw"
names(dat1)[5] <- "Change"
dat2 <- rslt[,c("Read.Depth.3",
                "VAF.3",
                "Read.Depth.4",
                "VAF.4",
                "sig.diff.fdr")]
dat2$method <- "p.value FDR"
names(dat2)[5] <- "Change"
dat3 <- rslt[,c("Read.Depth.3",
                "VAF.3",
                "Read.Depth.4",
                "VAF.4",
                "sig.change.20")]
names(dat3)[5] <- "Change"
dat3$method <- "Change > 0.2"

pdat <- rbind(dat1,dat2,dat3)


pdat$Total.Depth <- (as.numeric(pdat$Read.Depth.3)+
                     as.numeric(pdat$Read.Depth.4))/
                     max(as.numeric(pdat$Read.Depth.3)+
                     as.numeric(pdat$Read.Depth.4))*5
p <- ggplot(pdat,
            aes(x = as.numeric(VAF.3),
                y = as.numeric(VAF.4),
                color=Change))+
  geom_point(alpha = 0.5, size = pdat$Total.Depth)+
  scale_size_area()+
  xlab(paste("VAF", groups[1]))+
  ylab(paste("VAF", groups[2]))+
  facet_grid(cols = vars(method))+
  ggtitle("Three selection methods",
          subtitle = "Total read depth is represented by dot size")

print(p)
```



Citations

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(1), 289–300.

Chang, W., Cheng, J., Allaire, J., Xie, Y., McPherson, J., & others. (2017). Shiny: Web application framework for r. *R Package Version*, *1*(5), 2017.

Chang, W., Park, T., Dziedzic, L., Willis, N., & McInerney, M. (2018). Shinythemes: Themes for shiny. *R Package Version*, *1*(2).

Fisher, R. E. (1970). RA statistical methods for research workers. *Edinburgh: Oliver & Boyd. Ed*, *4*, I932.

Gohel, D., & Skintzos, P. (2016). Ggiraph: Make "ggplot2" graphics interactive. *R Package Version 0.3*, *2*.

Lesnoff, M., Lancelot, & R. (2012). *Aod: Analysis of overdispersed data*. https://cran.r-project.org/package=aod

Müller, K., & Wickham, H. (2021). *Tibble: Simple data frames*. https://CRAN.R-project.org/package=tibble

Pearson, K. (1900). X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *50*(302), 157–175.

R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Tarone, R. E. (1979). Testing the goodness of fit of the binomial distribution. *Biometrika*, *66*(3), 585–590.

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H., François, R., Henry, L., & Müller, K. (2021). *Dplyr: A grammar of data* *manipulation*. https://CRAN.R-project.org/package=dplyr

Wilke, C. O. (2020). *Cowplot: Streamlined plot theme and plot annotations for 'ggplot2'*. https://CRAN.R-project.org/package=cowplot

Xie, Y., Cheng, J., & Tan, X. (2022). *DT: A wrapper of the JavaScript library 'DataTables'*. https://CRAN.R-project.org/package=DT