Introduction to the easyVAF R package

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

[1]Cancer Center Biostatistics Core, University of Colorado Denver|Anschutz Medical Campus

*Junxiao.Hu@CUAnschutz.edu

2022-08-09

Abstract

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. VAFs on targeted loci under different (experimental) conditions are often compared. We present our R package ‘ esayVAF’ for parametric and non-parametric comparison of VAFs among multiple treatment groups.

Contents

1 easyVAF overview

Note: if you use easyVAF in published research, please cite:

> Junxiao Hu,Vida Alami, Yonghua Zhuang, Dexiang Gao. “easyVAF, a R package for VAF comparison among groups”. Journal of Open Source Software, 2022. (*Submitted*)

1.1 easyVAF package

The easyVAF package has the following dependencies:

The current version of the easyVAF package includes three (external) functions:

- **QCchecking()**: Quality checking for biological variability among samples.
- **Taronetest()**: Test for overdispersion in Poisson and Binomial Regression Models.
- **VAFmain()**: comparison of VAFs among N groups.

More details on above functions can be found in the package manual.

2 easyVAF workflow

We recommend VAF analysis work flow as following:

- 1). Start with exploratory plots for Variant Allele Count, Read Depth, and VAF for quality checking (i.e., unexpected biological variability, batch effect, technical effect);
- 2). Conduct 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).The main comparison of VAFs will be conducted as described below:
  - a). For each locus, the good-ness of fit test for binomial distribution (overdispersion) is conducted first;
  - b). Appropriate method (model-based or non-parametric) will be selected to perform the VAF comparison among treatment groups;
  - c). The raw and adjusted p-values will be reported for each locus, accompanied with the estimated VAFs, difference in VAFs and the corresponding confidence intervals (only available for two group comparisons).

2.1 Example dataset

In this document, we illustrate a standard workflow of VAF comparisons with a mouse VAF dataset.

```
library(easyVAF)
library(knitr)
data(dat)
names(VAF)
```

```
[1] “Locus” “vc” “dp” “chrom” “sample” “group”
```

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

Table 1: (#tab:example data)VAF data example

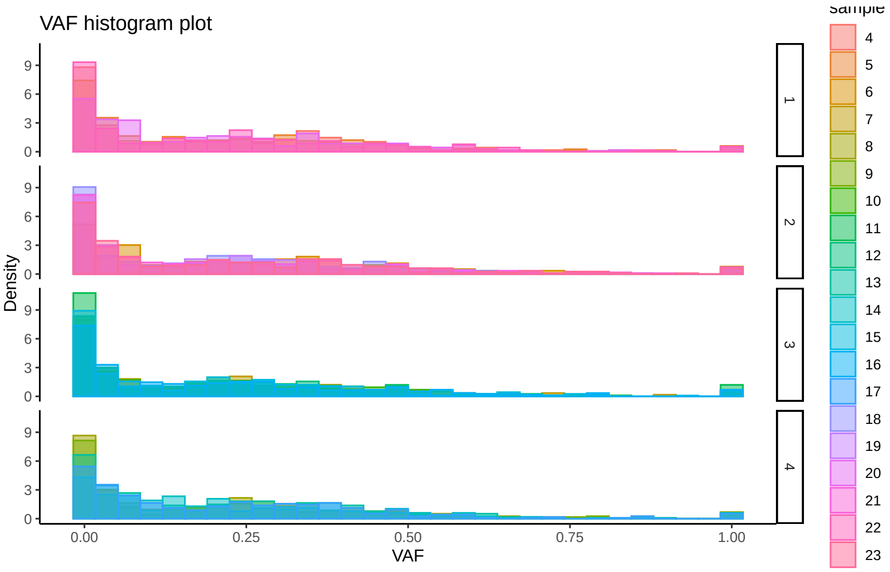
Locus	vc	dp	chrom	sample	group
38	0	49	1	10	3
40	0	22	1	10	3
1	6	16	1	10	3
6	0	8	1	10	3
15	0	35	1	10	3
19	8	31	1	10	3

The data contains the following columns:

- locus: locus ID
- vc: variant count (to calculate VAF)
- dp: read depth (to calculate VAF)
- chrom: chromosome information (for linkage disequilibrium adjustment in QC test, if desired)
- sample: mouse ID (for QC test)
- group: treatment group

2.2 Quality checking for biological variability among samples

```
rslt <- QCchecking(data=VAF, method="lm")
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
rslt
```

Analysis of Variance Table

Model 1: vaf ~ group/sample Model 2: vaf ~ group Res.Df RSS Df Sum of Sq F Pr(>F) 1 6649 345.74
2 6665 346.21 -16 -0.47018 0.5651 0.9115

2.3 Tarone test: overdispersion tests

We use Tarone test to examine overdispersion in Poisson and Binomial Regression Models.

```
Tarone.test(sum(VAF$dp),sum(VAF$vc))
```

```
Tarone's Z test
```

data: sum(VAF$vc)successesfromsum(VAF$dp) trials z = -0.70711, p-value = 0.4795 alternative hypothesis: true dispersion parameter is greater than 0 sample estimates: proportion parameter 0.1808001

2.4 Comparison of VAFs among N groups

We perform the comparisons for all four groups as illustration.

```
library(easyVAF)
#4 groups
groups <- unique(VAF$group)[c(1:4)]
rslt <- VAFmain(data=VAF, groups=groups)
rslt$P.value <- as.numeric(rslt$P.value)
names(rslt)
```

[1] “ID” “Read.Depth.3” “Variant.Count.3” “VAF.3”
[5] “Read.Depth.4” “Variant.Count.4” “VAF.4” “Read.Depth.2”
[9] “Variant.Count.2” “VAF.2” “Read.Depth.1” “Variant.Count.1” [13] “VAF.1” “P.value” “Test” “Effect.size”
[17] “95% CI” “Overdispersion” “p.adjust” “sig.diff”
[21] “sig.diff.fdr” “sig.change.20” “Change.direction”

```
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))
```

```


groups <- unique(VAF$group)[c(1:2)]
rslt <- VAFmain(data=VAF, groups=groups)
rslt$P.value <- as.numeric(rslt$P.value)
names(rslt)
```

Table 2: Top 10 significantly different loci, multiple group comparison

ID	P.value	Overdispersion	p.adjust	sig.diff.fdr
93	0.000	No	0.014	Difference
25	0.001	No	0.065	No difference
84	0.001	No	0.065	No difference
219	0.001	No	0.065	No difference
302	0.001	No	0.079	No difference
216	0.002	No	0.079	No difference
329	0.002	No	0.079	No difference
51	0.002	No	0.082	No difference
135	0.003	No	0.091	No difference
42	0.003	No	0.091	No difference

```
[1] “ID” “Read.Depth.3” “Variant.Count.3” “VAF.3”
[5] “Read.Depth.4” “Variant.Count.4” “VAF.4” “P.value”
[9] “Test” “Effect.size” “95% CI” “Overdispersion”
[13] “p.adjust” “sig.diff” “sig.diff.fdr” “sig.change.20”
[17] “Change.direction”
```

```
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 3: Top 10 significantly different loci, two groups

ID	Effect.size	95% CI	p.adjust	sig.diff.fdr	Change.direction
93	Diff in prop = 0.256	(0.09, 0.422)	0.107	No difference	Group1 > Group2
135	Diff in prop = -0.011	(-0.02, -0.002)	0.107	No difference	Group1 < Group2
45	Diff in prop = 0.061	(0.022, 0.1)	0.107	No difference	Group1 > Group2
149	Diff in prop = 0.102	(0.036, 0.167)	0.107	No difference	Group1 > Group2
59	Diff in prop = 0.014	(0.005, 0.023)	0.107	No difference	Group1 > Group2
50	Diff in prop = -0.162	(-0.269, -0.055)	0.107	No difference	Group1 < Group2
112	Diff in prop = -0.117	(-0.194, -0.039)	0.107	No difference	Group1 < Group2
215	Diff in prop = 0.137	(0.046, 0.228)	0.107	No difference	Group1 > Group2
89	Diff in prop = 0.076	(0.022, 0.13)	0.191	No difference	Group1 > Group2
315	Diff in prop = -0.061	(-0.107, -0.016)	0.191	No difference	Group1 < Group2

3 References

Junxiao Hu,Vida Alami, Yonghua Zhuang, Dexiang Gao. “easyVAF, a R package for VAF comparison among groups”. Journal of Open Source Software, 2022. (*Submitted*)