neutralitytestr: Testing a neutral evolutionary model on cancer sequencing data

Marc Williams

2018-04-10

Load package

After you have installed the package, you can load the package via the normal command. This will also load some test data.

```
library(neutralitytestr)
```

The test data files include 2 simulated VAF distributions, one under a neutral evolutionary model and one with a non neutral evolutionary model. The test data are vectors of variant allele frequency, and are named VAFselection and VAFneutral. First of all we print the first few elements of one of these vectors.

```
head(VAFselection)
```

```
## [1] 0.3669725 0.4056604 0.4226804 0.3333333 0.4200000 0.3700000
```

The test data were generated using an evolutionary model of cancer which produces synthetic sequencing data. The test data here were generated to approximately mimic whole genome sequencing data. There are ~ 5000 mutations in both datasets and the data was "sequenced" to 100X. Also included was normal contamination of 20%, that is why we observe the peak at roughly 0.4 ($2 \times 0.4 = 0.8$) which represents the clonal mutations in the sample, that is mutations present in every cancer cell. For both data the effective mutation rate (or equivalently the per tumour doubling mutation rate) is 200 corresponding to a WGS base per rate of $\sim 7 \times 10^{-8}$. For the `VAFneutral` data a neutral model where all cells have the same fitness was used, for the `VAFselection` data there is a single subclone at frequency 0.44 hence the peak at around 0.2 in the VAF spectrum (as will be seen below).

`neutralitytest` object

The basic functionality of the `neutralitytestr` package is achieved by creating a `neutralitytest` object. The `neutralitytest` object contains a range of metrics to test for neutrality, and makes plotting histograms and cumulative distributions to visualize the output easy. The neutralitytest function takes a vector of VAFs and an upper and lower limit for the frequency range over which we wish to test whether the data is consistent with a neutral model, and then calculates all 4 metrics.

```
s <- neutralitytest(VAFselection, fmin = 0.1, fmax = 0.25)
```

We can the print a summary of the neutralitytest object for the synthetic data with selection. This prints out all values and associated p-values for all the metrics. The p-values are the p-values under the null model of neutral evolution.

```
summary(s)
```

```
## Summary of neutrality metrics:
##
## Area:
##   value =  0.2029819 , p-value =  0.007
## Kolmogorov Distance:
##   value =  0.3357769 , p-value =  0.008
## Mean distance:
##   value =  0.202308 , p-value =  0.007
## R^2:
##   value =  0.9373371 , p-value =  0.009
##
## Effective mutation rate =  462.9867
```

And we can do the same for the neutral synthetic data.

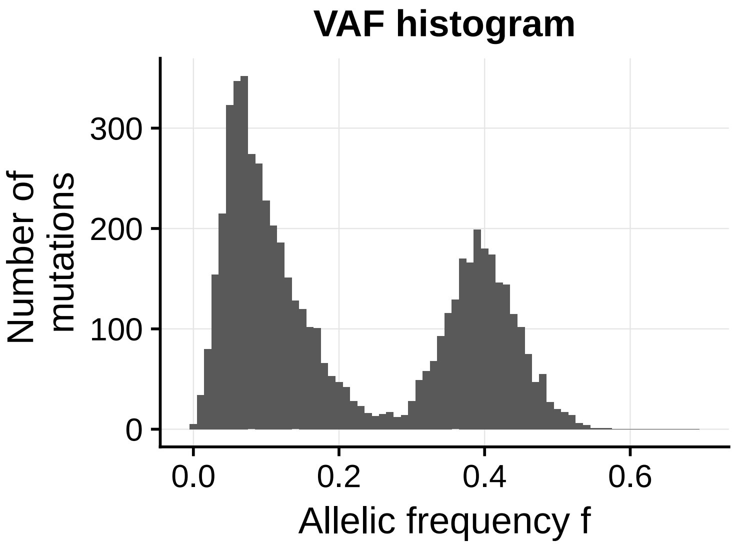
```
n <- neutralitytest(VAFneutral,fmin = 0.1, fmax = 0.25)
summary(n)
```

```
## Summary of neutrality metrics:
##
## Area:
##   value =  0.03067203 , p-value =  0.679
## Kolmogorov Distance:
##   value =  0.09036603 , p-value =  0.641
## Mean distance:
##   value =  0.04131414 , p-value =  0.595
## R^2:
##   value =  0.98993 , p-value =  0.404
##
## Effective mutation rate =  216.7985
```

Plotting

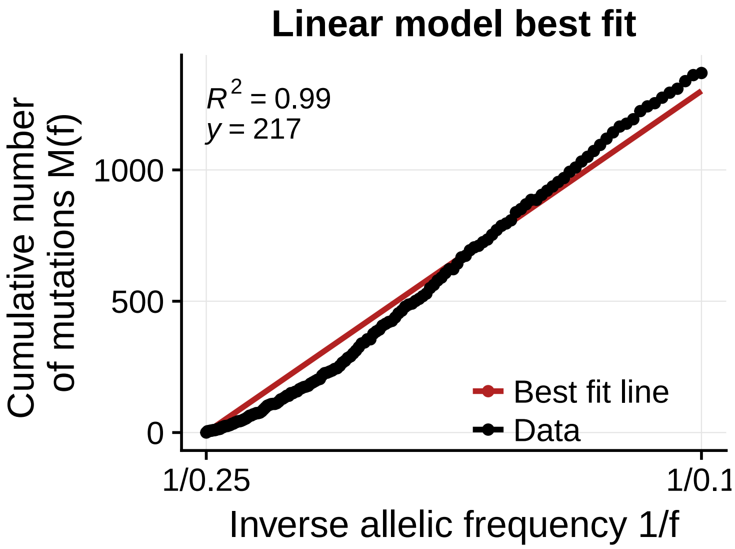
The `neutralitytest` object makes plotting simple. Plotting uses `ggplot`, so plots can easily modified using the usual `ggplot` syntax. First of all we can plot a histogram of the VAFs.

```
vaf_histogram(n)
```



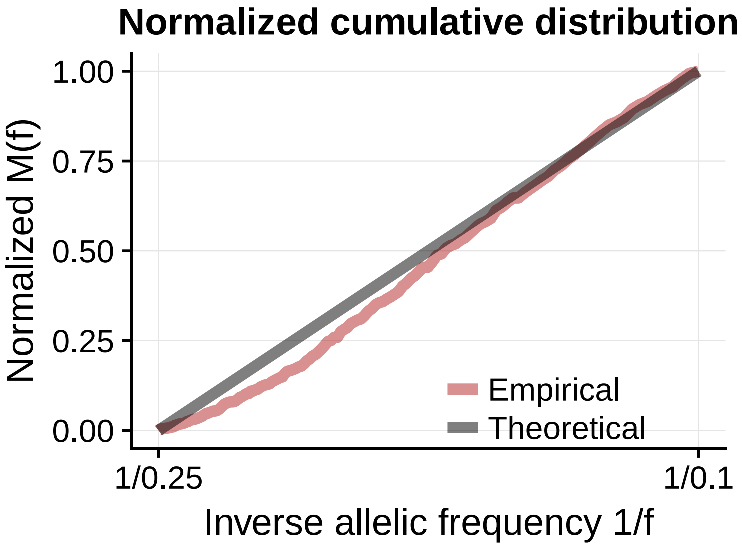
We can also plot the cumulative distribution along with the least squares best fit line, from which we get the R^2 value and the estimated mutation rate.

```
lsq_plot(n)
```



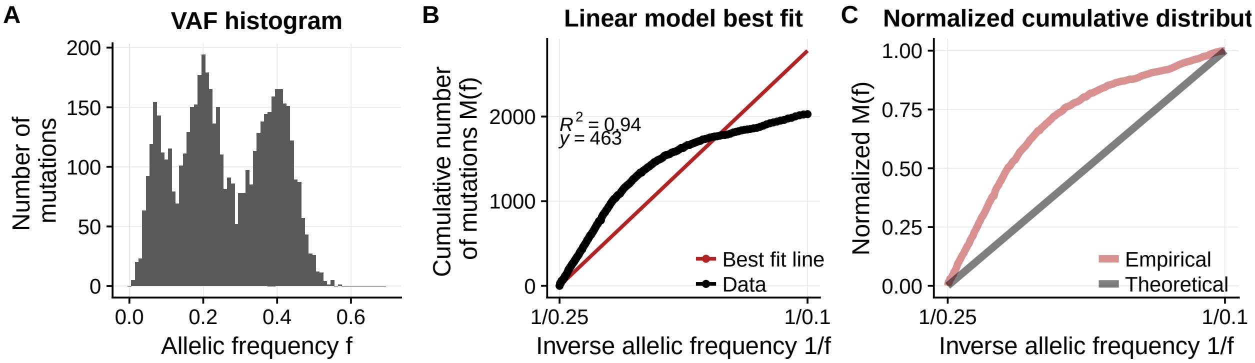
And finally the normalized cumulative distribution, which is used to calculate the kolmogorov distance and the area metrics.

```
normalized_plot(n)
```



Invoking the plot command will plot all 3 plots together, all these plots are generated using ggplot2, they can therefore be modified as any other ggplot object would be including saving etc.

```
gout <- plot_all(s)
gout
```



Summary of neutrality metrics

`neutralitytestr.R` calculates values for 4 different metrics from which we can deduce whether a given dataset is likely to be driven by a neutral evolutionary process or not. These metrics are all based the cumulative distributions of mutations, M(f), which under a neutral model follows the following equation:

$$M(f) = \frac{\mu}{\beta}\left(\frac{1}{f} - \frac{1}{f_{max}}\right) \tag{1}$$

where f is the frequency of mutations, μ is the mutation rate, β is the proportion of divisions that results in 2 surviving offspring and f_{max} is the maximum frequency over which we conduct the analysis. The first metric we calculate is the R^2 value from the best fit line of the linear model described by equation (1).

The other metrics are based on a normalized version of equation (1) which removes the mutation rate dependency. The equation for the normalized $M(f)$ is

$$M(f) = \frac{\frac{1}{f} - \frac{1}{f_{max}}}{\frac{1}{f_{min}} - \frac{1}{f_{max}}} \tag{2}$$

Theoretically, any dataset can be compared to the curve described by equation (2). We can then calculate the area between the data and this curve, the kolmogorov distance between the two and the mean distance of all points and this curve. For further details on the theoretical background see *Williams, Werner et al. Nat. Gen. 2016.*