Weighted expected binomial model

Anthony Aylward

2018-07-31

Prepare the data.

```
library(chenimbalance)
total_reads <- rowSums(accb[, c("cA", "cC", "cG", "cT")])
data <- data.frame(
  total = total_reads,
  allelicRatio = sapply(
    1:nrow(accb),
    function(i) {
      accb[[paste("c", accb[["ref"]][[i]], sep = "")]][[i]] / total_reads[[i]]
    }
  )
)
head(data)
#>   total allelicRatio
#> 1    45    1.0000000
#> 2    59    0.5423729
#> 3   114    0.4736842
#> 4    53    0.5094340
#> 5   119    0.5042017
#> 6    21    0.0952381
```

Empirical distribution

Compute the empirical allelic ratio distribution

```
binSize <- 40
bins <- pretty(0:1, binSize)
minN <- 6
maxN <- min(2500, max(data[["total"]]))
apropor <- length(data[["total"]][data[["total"]] <= 2500]) / nrow(data)
empirical <- empirical_allelic_ratio(
  data,
```

```
  bins,
  maxN = maxN,
  minN = minN,
  plot = TRUE
)
```

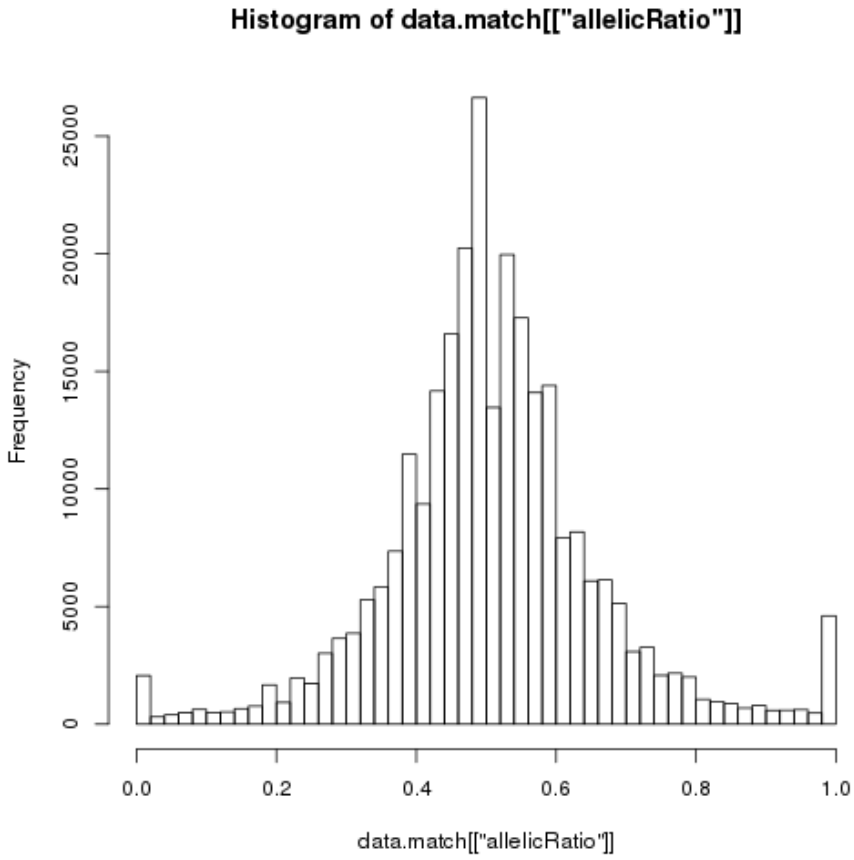

Figure 1: plot of chunk unnamed-chunk-2

Expected Binomial and Beta-Binomial distributions

Compute the weighted expected binomial distribution

```
w <- weight_by_empirical_counts(data[["total"]])
d_combined_sorted_binned <- nulldistrib(
  w,
```

```
  minN = minN,
  binSize = binSize
)
```

Compute the sum of squared errors for the empirical distribution vs the weighted expected binomial distribution.

```
sse <- sum((empirical - d_combined_sorted_binned[,2])^2)
sse
#> [1] 0.005153999
```

Choose the overdispersion parameter for the beta-binomial distribution

```
w_grad <- graded_weights_for_sse_calculation(r_min = 0, r_max = 1, bins = bins)
overdispersion_details <- choose_overdispersion_parameter(
  w_grad,
  w,
  empirical,
  sse
)
head(overdispersion_details[["b_and_sse"]])
#>         b         sse
#> [1,] 0.0 0.005153999
#> [2,] 0.1 0.006505881
#> [3,] 0.2 0.013358063
#> [4,] 0.0 0.000000000
#> [5,] 0.0 0.000000000
#> [6,] 0.0 0.000000000
```

Generate a plot of the weighted expected binomial and weighted expected beta-binomial distributions overlaid on the empirical distribution

```
plot_distributions(
  minN,
  maxN,
  bins,
  empirical,
  d_combined_sorted_binned,
  overdispersion_details[["e_combined_sorted_binned"]],
  yuplimit = 0.15
)
```

`overdispersion_details` is a list whose elements include the chosen value of `b` and the sum of squared errors.

```
paste(
  "b_chosen =",
  overdispersion_details[["b_choice"]],
  ", SSE_chosen =",
  overdispersion_details[["sse"]]
```

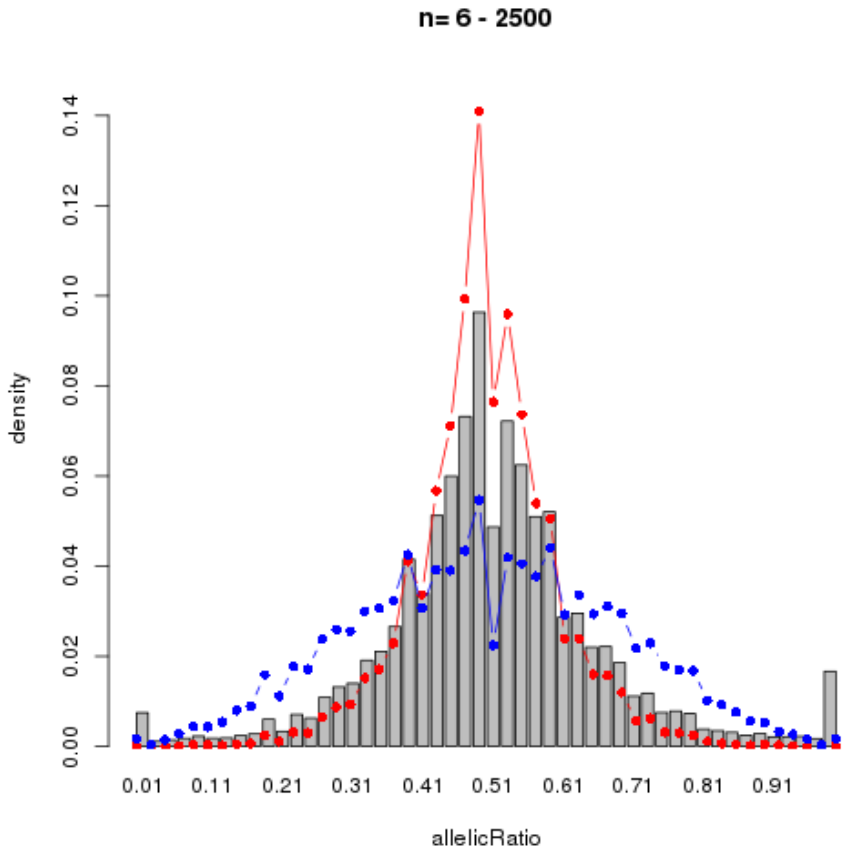

Figure 2: plot of chunk unnamed-chunk-6

```
)
#> [1] "b_chosen = 0.1 , SSE_chosen = 0.006505880670026 29"
```

Optimize the overdispersion parameter

```
optimized_overdispersion_details <- optimize_overdispersion_parameter(
  w_grad,
  overdispersion_details[["b_and_sse"]],
  overdispersion_details[["b_choice"]],
  overdispersion_details[["sse"]],
  empirical,
  overdispersion_details[["counter"]],
  minN = minN,
  binSize = binSize
)
plot_distributions(
  minN,
  maxN,
  bins,
  empirical,
  d_combined_sorted_binned,
  optimized_overdispersion_details[["e_combined_sorted_binned"]],
  yuplimit = 0.15
)
```

Check the optimized value

```
list(
  b = optimized_overdispersion_details[["b_choice"]],
  sse = optimized_overdispersion_details[["sse"]]
)
#> $b
#> [1] 0.01875
#>
#> $sse
#> [1] 0.0004222515
```

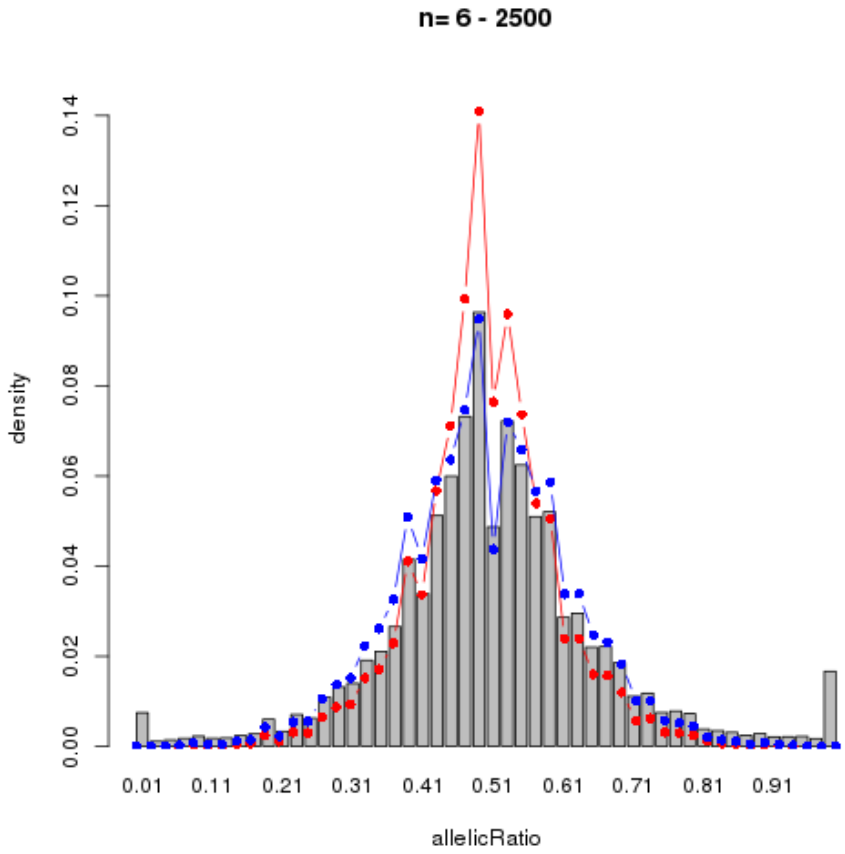

Figure 3: plot of chunk unnamed-chunk-8