NPBin

Anthony Aylward

2018-03-20

We will analyze a subset of one of the sample dataset for illustration purposes.

```r
library(npbin)

minimum_coverage <- 5 # minimum total coverage allowed
n_cores <- 47 # the number of cores to be used, can ONLY be 1 if run on Windows.

dt <- atac
colnames(dt)
#>  [1] "chr"             "location"        "m"
#>  [4] "xm"              "winning.chip"    "motif"
#>  [7] "pval.mat.atSNP"  "pval.pat.atSNP"  "pval.rank.atSNP"
#> [10] "winnig.motif"    "potential_TP"    "potential_FP"

dt.ct <- data.table(dt)[m >= minimum_coverage, ]
#> Error in data.table(dt): could not find function "data.table"
```

for illustration purpose, keep only the 2000 points, remove this line will end up with analyzing the whole data set. It could be slow if only one core is used.

```r
dt.ct <- dt.ct[1:2000, ]
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
dt.ct[, p_hat:=xm / m]
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
n <- nrow(dt.ct)
#> Error in nrow(dt.ct): object 'dt.ct' not found
```

NPBin

```r
n_breaks <- 11 # number of breaks
spline_order <- 4 # order of splines
breaks <- seq(0, 1, length.out = n_breaks)
```

```r
pi_init <- hist(
  dt.ct[, p_hat],
  breaks = seq(0, 1, length.out = n_breaks + spline_order - 3),
  plot = FALSE
)[["density"]] # initialized the weights using the histogram of p_hat
#> Error in hist(dt.ct[, p_hat], breaks = seq(0, 1, length.out = n_breaks + : object 'dt.ct
```

estimate the overall model

```r
overall_model_estimate <- emBinBspl(
  dt.ct[, xm],
  dt.ct[, m],
  breaks = breaks,
  k = spline_order,
  pi.init = pi_init,
  ncores = n_cores,
  err.max = 1e-3,
  iter.max = 200
)
#> Error in emBinBspl(dt.ct[, xm], dt.ct[, m], breaks = breaks, k = spline_order, : object
```

estimate the null model

```r
null_model_estimate <- estNull(
  dt.ct[, xm],
  dt.ct[, m],
  overall_model_estimate,
  init = NULL,
  iter.max = 200,
  ncores = n_cores,
  ub = rep(log(1e4), 2),
  err.max = 1e-4
)
#> Error in estNull1(mod, pseq, ncores = ncores): object 'overall_model_estimate' not found
dt.ct[,
  fnp := null_model_estimate[["f"]]
][,
  f0np := null_model_estimate[["f0"]]
][,
  locfdrnp := null_model_estimate[["locfdr"]]
][,
  fdrnp := locfdr2FDR(locfdrnp)
][,
  ranknp := rank(locfdrnp, ties.method = "max")
]
```

```
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
names(null_model_estimate)
#> Error in eval(expr, envir, enclos): object 'null_model_estimate' not found

null_model_estimate$coef.null
#> Error in eval(expr, envir, enclos): object 'null_model_estimate' not found
```

Empirical Bayes test using p_hat

```
pct0 <- 0.45
empirical_bayes_beta_hat <- ebBeta(
  dt.ct[, xm],
  dt.ct[, m],
  dt.ct[, p_hat],
  breaks = breaks,
  k = spline_order,
  pi.init = pi_init,
  pct0 = pct0,
  init = NULL,
  iter.max = 200,
  err.max = 1e-4,
  ncores = n_cores
)
#> Error in emBspl(x, m, p, breaks = breaks, k = k, pi.init = pi.init, ncores = ncores, : ol
dt.ct[,
  fhat := empirical_bayes_beta_hat[["f"]]
][,
  f0hat := empirical_bayes_beta_hat[["f0"]]
][,
  locfdrhat := empirical_bayes_beta_hat[["locfdr"]]
][,
  fdrhat := locfdr2FDR(locfdrhat)
][,
  rankhat := rank(locfdrhat, ties.method = "max")
]
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
names(empirical_bayes_beta_hat)
#> Error in eval(expr, envir, enclos): object 'empirical_bayes_beta_hat' not found
```

null parameters of EBE

```
empirical_bayes_beta_hat[["coef.null"]]
#> Error in eval(expr, envir, enclos): object 'empirical_bayes_beta_hat' not found
```

Binomial test

```
p_binomial <- sapply(
  1:n,
  function(y) binom.test(dt.ct[y, xm], dt.ct[y, m])[["p.value"]]
)
#> Error in lapply(X = X, FUN = FUN, ...): object 'n' not found
dt.ct[,
  pvbin := p_binomial
][,
  fdrbin := p.adjust(pvbin, method = "BH")
][,
  rankbin := rank(pvbin, ties.method = "max")
]
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
```

Evaluate the results using motifs

number of potential TP defined based on motif

```
dt.ct[, sum(potential_TP)]
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
```

number of potential FP defined based on motif

```
dt.ct[, sum(potential_FP)]
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
```

find the number of TP and FP in top ranked SNPs

```
dt.ct[,
  tpnp := rank2nhit(ranknp ,potential_TP)
][,
  fpnp := rank2nhit(ranknp, potential_FP)
]
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
dt.ct[,
  tp_hat := rank2nhit(rankhat, potential_TP)
][,
  fp_hat := rank2nhit(rankhat, potential_FP)
]
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
dt.ct[,
  tpbin := rank2nhit(rankbin, potential_TP)
][,
  fpbin := rank2nhit(rankbin, potential_FP)
]
#> Error in eval(expr, envir, enclos): object 'dt.ct' not found
```

plot the accuracy measure as in the main paper. We presented a zoom-in version in the main paper to the top 20%, because usually there are not many ALI SNPs. Note that the default of the demo only select a subset of the data for illustration purposes. Thus the figure may not an exact replica of the one in the paper. To reproduce the results in the paper, please use the whole dataset

```r
cbfpalette <- c(
  "#D55E00",
  "#0072B2",
  "#CC79A7",
  "#009E73",
  "#E69F00",
  "#56B4E9",
  "#F0E442"
)
plotidac <- c(' NPB','EBE','Binom')
```