User Manual: inctePareto

Yingying Cao

2019-10-08

Contents

```
require(intePareto)
```

Introduction

The R package intePareto is designed to do integrative analysis when matched RNA-Seq and ChIP-Seq data are available from the same set of samples. The main steps include the matching of RNA-Seq and ChIP-Seq data on gene level, quantitativey integrate both data types and prioritize genes using Pareto optimaztion.

Method

1. bam to counts/rpm
intePareto transform the aligned bam files into counts or rpm (reads per million) that fall into specific region like promoter or genebody.

```
data("promoter", package = "intePareto")
file.bam <- system.file("extdata", "SRR925640.bam",
                        package = "intePareto")
# note: this is just a subsampled test bam file to show how it works
b2c <- bam2counts(bamFile = file.bam,
                  region = promoter,
                  fragLength = 180)
tail(b2c)
#> [1] 0 0 1 1 0 0
b2r <- bam2rpm(bamFile = file.bam,
               region = promoter,
               fragLength = 180)
tail(b2r)
#> [1]   0.000   0.000 366.166 366.166   0.000   0.000
```

2. match of RNA-Seq and ChIP-Seq data on gene lelvel
Since one gene usually have several promoters and most of them have overlaps on the region, we use two strategies (1)highest: the promoters with maximum value or (2) weighted.mean: the weighted mean of several promoter to represent the signal of one specific histone modification for one gene. Through this approach we get the match of RNA-Seq and ChIP-Seq on gene level.

3. quantitativey integrate RNA-Seq and ChIP-Seq data
After the match of RNA-Seq and ChIP-Seq on gene level, the integration of these two types of data is conducted through the calculation of lFC (log2FoldChange), the lFC is shrunk to zero when the counts are low, dispersion is high, or the number of replicates is small, which is implemented by DESeq2, and then Z score $Z.score = (lFC.RNAseq \mid sd(lFC.RNAseq)) * (lFC.ChIPseq \mid sd(lFC.ChIPseq))$ is calculated for each histone modifiaction mark which measures the correlation of chnages in RNA-Seq and changes in ChIP-Seq for this mark.

4. prioritization
Take the Z scores of several different histone modifications as input, the prioritization of genes based on Z scores can be formulated as multiobjective optimization problem and solved with Pareto optimization

Input

The input to intePareto is two data.frame objects with meta data of preprocessed RNA-Seq data and meta data with preprocessed ChIP-Seq data separately.

The RNA-Seq meta data.frame at least two columns:
1. condition: identifier of the condition to which each sample belongs (e.g. wild.type or tet2.out).
2. files: the exact address of the files contains the tsv file which is the output of RNA-Seq preprocessed with Kallisto.

```
data("test_rna_meta", package = "intePareto")
test_rna_meta
#>         SRR condition                          files
#> 1 SRR925874 wild.type ./inst/extdata/SRR925874.tsv
#> 2 SRR925875 wild.type ./inst/extdata/SRR925875.tsv
#> 3 SRR925878  tet2.out ./inst/extdata/SRR925878.tsv
#> 4 SRR925879  tet2.out ./inst/extdata/SRR925879.tsv
```

The ChIP-Seq meta data.frame at least three columns:
1. mark: the mark of histone modifications (e.g. H3K4me3 or H3K27ac).
2. condition: identifier of the condition to which each sample belongs (e.g. wild.type or tet2.out).
3. files: the exact address of the aligned bam files.

```
data("test_chip_meta", package = "intePareto")
test_chip_meta
#>         SRR    mark condition                          files
#> 1 SRR925640 H3K4me3 wild.type ./inst/extdata/SRR925640.bam
#> 2 SRR925642 H3K27ac wild.type ./inst/extdata/SRR925642.bam
#> 3 SRR925643 H3K27ac wild.type ./inst/extdata/SRR925643.bam
#> 4 SRR925651 H3K4me3 wild.type ./inst/extdata/SRR925651.bam
#> 5 SRR925674 H3K4me3  tet2.out ./inst/extdata/SRR925674.bam
#> 6 SRR925675 H3K27ac  tet2.out ./inst/extdata/SRR925675.bam
#> 7 SRR925680 H3K4me3  tet2.out ./inst/extdata/SRR925680.bam
#> 8 SRR925682 H3K27ac  tet2.out ./inst/extdata/SRR925682.bam
```

Match of RNA-Seq and ChIP-Seq data on gene lelvel

The results of this step is a list with the following three items.
1. res.rna, a data frame contains RNA-Seq counts
2. res.chip, a data frame contains ChIP-Seq counts
3. matched.data, a dataframe contains matched RNA-Seq counts and ChIP-Seq counts

```
# get the exact place of the tsv files for RNA-Seq and
# bam files for ChIP-Seq data.
test_rna_meta$files <- system.file("extdata",
                                   paste0(test_rna_meta$SRR,".tsv"),
                                   package = "intePareto")


test_chip_meta$files <- system.file("extdata",
                                    paste0(test_chip_meta$SRR,".bam"),
                                    package = "intePareto")
```

```
# match of RNA-Seq and ChIP-Seq data on the gene level
# through "weighted.mean" strategy.
res <- doMatch(rnaMeta = test_rna_meta,
               chipMeta = test_chip_meta,
               region = "promoter",
               method = "weighted.mean",
               ensemblDataset = "mmusculus_gene_ensembl")
```

Quantitativey integrate RNA-Seq and ChIP-Seq data

```
data("res", package = "intePareto")
df_final <- doIntegration(res = res,
                          ref = "wild.type",
                          type = "apeglm",
                          apeAdapt = FALSE)

head(df_final)
#>                RNAseq.log2FoldChange H3K27ac.log2FoldChange
#> 2610507B11Rik         -0.1028945691            -0.04255326
#> Aamp                   0.0594382012             0.01925989
#> Acaca                  0.1628259986             0.07590317
#> Acad9                 -0.0738736278             0.04960663
#> Acat2                  0.0002867839            -0.06513978
#> Acin1                  0.0148561743            -0.08441788
#>                H3K4me3.log2FoldChange   z.H3K27ac    z.H3K4me3
#> 2610507B11Rik           -0.057192888  0.37817773  0.532225902
#> Aamp                    -0.062393084  0.09887583 -0.335400447
#> Acaca                    0.091914295  1.06746656  1.353531414
#> Acad9                    0.022099341 -0.31651901 -0.147648923
#> Acat2                    0.075096253 -0.00161351  0.001947756
#> Acin1                    0.009091038 -0.10832084  0.012214677
```

Prioritization of genes based on Z scores with Pareto optimization

```
# chosse the first 3 fronts
objective <- data.frame(mark = c("z.H3K27ac","z.H3K4me3"),
                        obj = c("max","max"), stringsAsFactors = FALSE)
nr.fronts <- 3
res_final <- doPareto(df_final = df_final,
                      objective = objective,
                      nr.fronts = nr.fronts)
head(res_final)
#>         RNAseq.log2FoldChange H3K27ac.log2FoldChange
#> Kdm5a               0.2608758             0.11572511
#> Pnn                 0.3807880             0.21205779
#> Exosc10             0.2234160             0.20071534
#> Ppan                0.2758964             0.08579278
#> Pwp1               -0.3072403            -0.13721852
#> Qrich1             -0.2251835            -0.22278891
```

```
#>          H3K4me3.log2FoldChange z.H3K27ac  z.H3K4me3 front
#> Kdm5a               0.192876889  2.607546  4.5506722     1
#> Pnn                 0.105475127  6.974421  3.6324101     1
#> Exosc10            -0.080872108  3.873160 -1.6340847     2
#> Ppan                0.136129635  2.044407  3.3967240     2
#> Pwp1               -0.009256043  3.641342  0.2571965     2
#> Qrich1              0.239661111  4.333120 -4.8808520     2
```

```
sessionInfo()
#> R version 3.6.1 (2019-07-05)
#> Platform: x86_64-pc-linux-gnu (64-bit)
#> Running under: Ubuntu 16.04.6 LTS
#>
#> Matrix products: default
#> BLAS:   /usr/lib/libblas/libblas.so.3.6.0
#> LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
#>
#> locale:
#>  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
#>  [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=C
#>  [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=en_US.UTF-8
#>  [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
#>  [9] LC_ADDRESS=C               LC_TELEPHONE=C
#> [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#> [1] intePareto_0.0.1
#>
#> loaded via a namespace (and not attached):
#>   [1] bitops_1.0-6                matrixStats_0.55.0
#>   [3] bit64_0.9-7                 httr_1.4.1
#>   [5] progress_1.2.2              RColorBrewer_1.1-2
#>   [7] GenomeInfoDb_1.20.0         numDeriv_2016.8-1.1
#>   [9] tools_3.6.1                 backports_1.1.4
#>  [11] R6_2.4.0                    rpart_4.1-15
#>  [13] Hmisc_4.2-0                 DBI_1.0.0
#>  [15] lazyeval_0.2.2              BiocGenerics_0.30.0
#>  [17] colorspace_1.4-1            nnet_7.3-12
#>  [19] apeglm_1.6.0                prettyunits_1.0.2
#>  [21] tidyselect_0.2.5            gridExtra_2.3
#>  [23] DESeq2_1.24.0               curl_4.0
#>  [25] bit_1.1-14                  compiler_3.6.1
#>  [27] Biobase_2.44.0              htmlTable_1.13.1
#>  [29] DelayedArray_0.10.0         scales_1.0.0
#>  [31] checkmate_1.9.4             genefilter_1.66.0
#>  [33] stringr_1.4.0               digest_0.6.20
#>  [35] Rsamtools_2.0.0             foreign_0.8-72
#>  [37] rmarkdown_1.15              XVector_0.24.0
#>  [39] base64enc_0.1-3             pkgconfig_2.0.2
#>  [41] htmltools_0.3.6             bbmle_1.0.20
#>  [43] htmlwidgets_1.3             rlang_0.4.0
#>  [45] rstudioapi_0.10             RSQLite_2.1.2
#>  [47] rPref_1.3                   BiocParallel_1.18.1
#>  [49] acepack_1.4.1               dplyr_0.8.3
#>  [51] RCurl_1.95-4.12             magrittr_1.5
#>  [53] GenomeInfoDbData_1.2.1      Formula_1.2-3
#>  [55] Matrix_1.2-17               Rcpp_1.0.2
#>  [57] munsell_0.5.0               S4Vectors_0.22.0
```

```
#>  [59] stringi_1.4.3              yaml_2.2.0
#>  [61] MASS_7.3-51.4              SummarizedExperiment_1.14.1
#>  [63] zlibbioc_1.30.0            plyr_1.8.4
#>  [65] grid_3.6.1                 blob_1.2.0
#>  [67] parallel_3.6.1             crayon_1.3.4
#>  [69] lattice_0.20-38            Biostrings_2.52.0
#>  [71] splines_3.6.1              annotate_1.62.0
#>  [73] hms_0.5.1                  locfit_1.5-9.1
#>  [75] zeallot_0.1.0              knitr_1.24
#>  [77] pillar_1.4.2               igraph_1.2.4.1
#>  [79] GenomicRanges_1.36.0       biomaRt_2.40.4
#>  [81] geneplotter_1.62.0         stats4_3.6.1
#>  [83] XML_3.98-1.20              glue_1.3.1
#>  [85] evaluate_0.14              latticeExtra_0.6-28
#>  [87] data.table_1.12.2          RcppParallel_4.4.3
#>  [89] vctrs_0.2.0                gtable_0.3.0
#>  [91] purrr_0.3.2                assertthat_0.2.1
#>  [93] ggplot2_3.2.1              emdbook_1.3.11
#>  [95] xfun_0.9                   xtable_1.8-4
#>  [97] coda_0.19-3                survival_2.44-1.1
#>  [99] tibble_2.1.3               GenomicAlignments_1.20.1
#> [101] AnnotationDbi_1.46.1       memoise_1.1.0
#> [103] IRanges_2.18.2             cluster_2.1.0
```