Read an external file of poly(A) sites and analyze it with the movAPA package

Xiaohui Wu, Wenbin Ye, Tao Liu, Hongjuan Fu

Last modified 2020-12-25

Contents

1 Overview

This documentation describes how to read an external file of poly(A) sites and analyze it with movAPA. We used the model species – Arabidopsis for demonstration. First we can download a poly(A) site list from PlantAPAdb. Here we just downloaded poly(A) site clusters (PACs) for demostration. A PAC is already the group of nearby cleavage sites.

Demo file 1: PACs with genome annotation (3 replicates). Download the data (arabidopsis_thaliana.SRP093950_amp.high_confidence.PAC.annotation.tpm.csv) here.

Demo file 2: PACs in bed format with only coordinates. Download the data here.

These data files and the Arabidopsis TAIR10 gff3 file can also be downloaded from here.

2 Read the file of PACs with genome annotation

movAPA implemented the *PACdataset* object for storing the expression levels and annotation of PACs from various conditions/samples. Almost all analyses of poly(A) site data in movAPA are based on the *PACdataset*. The “counts” matrix is the first element in the array list of *PACdataset*, which stores non-negative values representing expression levels of PACs. The “colData” matrix records the sample information and the “anno” matrix stores the genome annotation or additional information of the poly(A) site data.

2.1 Data read

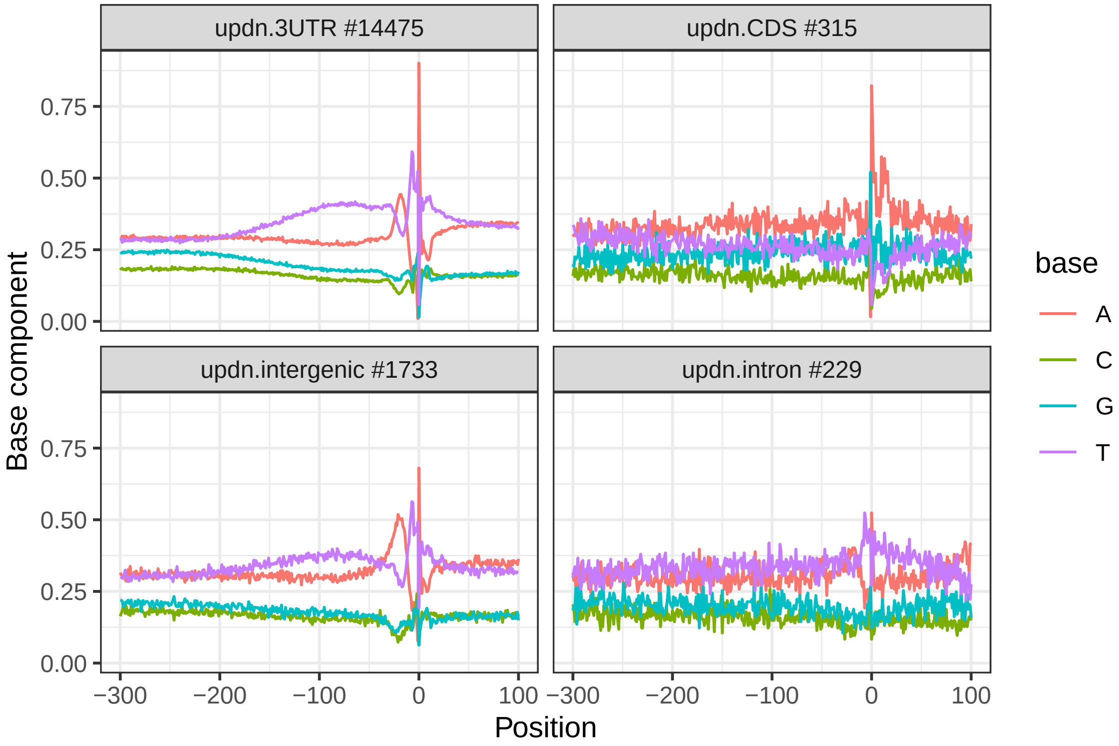
```r
library(movAPA, warn.conflicts = FALSE, quietly=TRUE)
filename <- 'arabidopsis_thaliana.SRP093950_amp.high_confidence.PAC.annotation.tpm.csv'
pac=read.csv(filename,stringsAsFactors =F)
pac <- pac[,-1]
## Rename annotation columns.
## In a PACdataset, the annotation column names must be named as
##(gene/gene_type/ftr/ftr_start/ftr_end/UPA_start/UPA_end).
## Other non-sample columns will be also retained
## in the @anno slot of the PACdataset.
pac=dplyr::rename(pac, UPA_start = 'start', UPA_end='end', gene_type='biotype')
colnames(pac)
#>  [1] "chr"              "UPA_start"        "UPA_end"          "strand"
#>  [5] "PAnum"            "tot_tag"          "coord"            "refPAnum"
#>  [9] "ftr"              "gene_id"          "gene_type"        "ftr_start"
#> [13] "ftr_end"          "upstream_id"      "upstream_start"   "upstream_end"
#> [17] "downstream_id"    "downstream_start" "downstream_end"   "Amp311_R1"
#> [21] "Amp311_R2"        "Amp311_R3"        "average"

## Describe the sample columns and corresponding group(s) in a data.frame
colData=as.data.frame(matrix(c('Amp','Amp','Amp'), ncol=1,
                                 dimnames=list(paste0('Amp311_R',1:3), 'group')))

## Read the PAC file into a PACdataset
PACds=readPACds(pacFile=pac, colDataFile=colData, noIntergenic=FALSE, PAname='PA')
#> 16959 PACs

PACds
#> PAC# 16959
#> gene# 0
#>             nPAC
#> 3UTR       14475
#> 5UTR          78
#> CDS          315
#> exon         129
#> intergenic  1733
#> intron       229
#> sample# 3
#> Amp311_R1 Amp311_R2 Amp311_R3 ...
#> groups:
#> @colData...[3 x 1]
#>           group
#> Amp311_R1   Amp
#> Amp311_R2   Amp
#> @counts...[16959 x 3]
#>     Amp311_R1 Amp311_R2 Amp311_R3
#> PA1  1.881794  3.730031  3.874707
#> PA2  4.390854  1.017281  6.027323
#> @colData...[3 x 1]
#>           group
#> Amp311_R1   Amp
#> Amp311_R2   Amp
#> @anno...[16959 x 20]
#>     chr UPA_start UPA_end strand PAnum tot_tag coord refPAnum        ftr
#> PA1   1      5846    5922      +    13      34  5895        6       3UTR
#> PA2   1     13924   13935      +     4      27 13926       23 intergenic
#>       gene_id      gene_type ftr_start ftr_end upstream_id upstream_start
#> PA1 AT1G01010 protein_coding      5631    5899        <NA>             NA
#> PA2 AT1G03987         lncRNA     11372   23121   AT1G03987          11101
#>     upstream_end downstream_id downstream_start downstream_end  average
#> PA1           NA          <NA>               NA             NA 3.162177
#> PA2        11372     AT1G01040            23121          31227 3.811819
```

2.2 Statistics

After read the data into a *PACdataset*, users can use many functions in movAPA for removing internal priming artifacts, polyA signal analysis, etc. Please follow the vignette of "movAPA_on_rice_tissues" for more details.

```
# For example, users can remove internal priming artifacts
library("BSgenome.Athaliana.TAIR.TAIR9")
bsgenome <- BSgenome.Athaliana.TAIR.TAIR9

# Please make sure the chromosome name of your PAC data is the same as the BSgenome.
seqnames(bsgenome) <- c(1:5,’Mt’,’Pt’)
seqnames(bsgenome)
#> [1] "1"  "2"  "3"  "4"  "5"  "Mt" "Pt"

PACdsIP=removePACdsIP(PACds, bsgenome, returnBoth=TRUE,
                      up=-10, dn=10, conA=6, sepA=7)
#> 5086 IP PACs; 11873 real PACs
length(PACdsIP$real)
```

```
#> [1] 11873
length(PACdsIP$ip)
#> [1] 5086

# Base compostions and k-grams
faFiles=faFromPACds(PACds, bsgenome, what='updn', fapre='updn',
                    up=-300, dn=100, byGrp='ftr')
#> 14475 >>> updn.3UTR.fa
#> 1733 >>> updn.intergenic.fa
#> 229 >>> updn.intron.fa
#> 129 >>> updn.exon.fa
#> 78 >>> updn.5UTR.fa
#> 315 >>> updn.CDS.fa

faFiles=c("updn.3UTR.fa", "updn.CDS.fa", "updn.intergenic.fa", "updn.intron.fa")
## Plot single nucleotide profiles using the extracted sequences
## and merge all plots into one.
plotATCGforFAfile (faFiles, ofreq=FALSE, opdf=FALSE,
                   refPos=301, mergePlots = TRUE)
```



3 Read the file of PACs with only coordinates

In this section, we show how to read a list of polyA sites with only coordinates. Here we use the file in bed format for demonstration.

3.1 Data read

```
library(movAPA)
## Read a BED file
pac=read.table(’arabidopsis_thaliana.SRP093950_amp.high_confidence.PAC.bed’,
               header=F, stringsAsFactors = F)
head(pac)
#>   V1    V2    V3 V4 V5
#> 1  1  5846  5922  .  +
#> 2  1 13924 13935  .  +
#> 3  1 31127 31190  .  +
#> 4  1 74000 74111  .  +
#> 5  1 76647 76709  .  +
#> 6  1 89687 89851  .  +
# We only keep the chr/strand/coord, here we used the start position as the coord.
colnames(pac)=c(’chr’,’coord’,’x’,’dot’,’strand’)
pac=pac[,c(’chr’,’strand’,’coord’)]

# We don’t have any expression level of the sample,
# so we only read the PAC list and set the expression as 1.

## Read the PAC file into a PACdataset
PACds=readPACds(pacFile=pac, colDataFile=NULL, noIntergenic=FALSE, PAname=’PA’)
#> 16959 PACs

PACds
#> PAC# 16959
#> sample# 1
#> tag ...
#> groups:
#> @colData...[1 x 1]
#>      group
#> tag group1
#> @counts...[16959 x 1]
#>     tag
#> PA1   1
#> PA2   1
#> @colData...[1 x 1]
#>      group
#> tag group1
#> @anno...[16959 x 3]
#>     chr strand coord
#> PA1   1      +  5846
#> PA2   1      + 13924
```

3.2 Annotation

After read the data into a *PACdataset*, users can use movAPA for annotation first. Please download the genome annotation file of Arabidopsis TAIR 10 in gff3 format from the tair website.

```
athGFF="./Arabidopsis_thaliana.TAIR10.42/Arabidopsis_thaliana.TAIR10.42.gff3"
# First we parse the gff3 file.
gff=parseGff(athGFF)
# Please make sure the chromosome name of your PAC data is the same as
# the gff file (and the BSgenome)
head(gff$anno.need)
#>   seqnames start  end width strand    source           type score phase
#> 4        1  3631 3759   129      + araport11 five_prime_UTR    NA    NA
#> 5        1  3631 3913   283      + araport11           exon    NA    NA
#> 6        1  3760 3913   154      + araport11            CDS    NA     0
#> 7        1  3996 4276   281      + araport11           exon    NA    NA
#> 8        1  3996 4276   281      + araport11            CDS    NA     2
#> 9        1  4486 4605   120      + araport11           exon    NA    NA
#>            ID Alias              Name        biotype description   gene_id
#> 4        <NA>                    <NA> protein_coding        <NA> AT1G01010
#> 5        <NA>       AT1G01010.1.exon1 protein_coding        <NA> AT1G01010
#> 6 AT1G01010.1                    <NA> protein_coding        <NA> AT1G01010
#> 7        <NA>       AT1G01010.1.exon2 protein_coding        <NA> AT1G01010
#> 8 AT1G01010.1                    <NA> protein_coding        <NA> AT1G01010
#> 9        <NA>       AT1G01010.1.exon3 protein_coding        <NA> AT1G01010
#>   logic_name      Parent transcript_id constitutive ensembl_end_phase
#> 4       <NA> AT1G01010.1          <NA>         <NA>              <NA>
#> 5       <NA> AT1G01010.1          <NA>            1                 1
#> 6       <NA> AT1G01010.1          <NA>         <NA>              <NA>
#> 7       <NA> AT1G01010.1          <NA>            1                 0
#> 8       <NA> AT1G01010.1          <NA>         <NA>              <NA>
#> 9       <NA> AT1G01010.1          <NA>            1                 0
#>   ensembl_phase           exon_id rank  protein_id Is_circular
#> 4          <NA>              <NA> <NA>        <NA>        <NA>
#> 5            -1 AT1G01010.1.exon1    1        <NA>        <NA>
#> 6          <NA>              <NA> <NA> AT1G01010.1        <NA>
#> 7             1 AT1G01010.1.exon2    2        <NA>        <NA>
#> 8          <NA>              <NA> <NA> AT1G01010.1        <NA>
#> 9             0 AT1G01010.1.exon3    3        <NA>        <NA>
# You can also save the parsed gff file as an rda object for further use.
# save(gff, file=’TAIR10.gff.rda’)
# Annotate the PAC data
PACds=annotatePAC(PACds, gff)
PACds
#> PAC# 16959
#> gene# 11769
#>             nPAC
#> 3UTR       12927
#> 5UTR          89
#> CDS          450
#> exon         140
#> intergenic  3088
#> intron       265
#> Mean 3UTR length of PACs (bp): 217
#> sample# 1
#> tag ...
#> groups:
#> @colData...[1 x 1]
#>      group
#> tag group1
#> @counts...[16959 x 1]
#>        tag
#> PA9399   1
#> PA9400   1
#> @colData...[1 x 1]
#>      group
#> tag group1
#> @anno...[16959 x 19]
#>        chr strand    coord        ftr      gene_type ftr_start  ftr_end
#> PA9399   1      - 10031958 intergenic protein_coding  10031985 10014256
#> PA9400   1      - 10041642       3UTR protein_coding  10041576 10041837
#>             gene gene_start gene_end gene_stop_codon upstream_id upstream_start
#> PA9399 AT1G28530   10031985 10035638        10032127   AT1G28530       10031985
#> PA9400 AT1G28570   10041576 10044258        10041838        <NA>             NA
#>        upstream_end downstream_id downstream_start downstream_end
#> PA9399     10035638     AT1G28480         10013434       10014256
#> PA9400           NA          <NA>               NA             NA
#>        three_UTR_length three_extend
#> PA9399              169           27
#> PA9400              196           NA
#> @supp...[1]
#> stopCodon
```

3.3 Statistics

After read the data into a *PACdataset*, users can use many functions in movAPA for removing internal priming artifacts, polyA signal analysis, etc. Please follow the vignette of “movAPA_on_rice_tissues” or the above example for more details.

4 Session Information

The session information records the versions of all the packages used in the generation of the present document.

```
sessionInfo()
#> R version 3.6.0 (2019-04-26)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 18363)
#> 
#> Matrix products: default
#> 
#> locale:
#> [1] LC_COLLATE=Chinese (Simplified)_China.936 
#> [2] LC_CTYPE=Chinese (Simplified)_China.936   
#> [3] LC_MONETARY=Chinese (Simplified)_China.936
#> [4] LC_NUMERIC=C                              
#> [5] LC_TIME=Chinese (Simplified)_China.936    
#> 
#> attached base packages:
#> [1] stats4    parallel  stats     graphics  grDevices utils     datasets 
#> [8] methods   base     
#> 
#> other attached packages:
#>  [1] BSgenome.Athaliana.TAIR.TAIR9_1.3.1000
#>  [2] movAPA_0.1.0                          
#>  [3] DEXSeq_1.32.0                         
#>  [4] DESeq2_1.26.0                         
#>  [5] SummarizedExperiment_1.16.1           
#>  [6] DelayedArray_0.12.3                   
#>  [7] BiocParallel_1.20.1                   
#>  [8] matrixStats_0.57.0                    
#>  [9] GenomicFeatures_1.38.2                
#> [10] AnnotationDbi_1.48.0                  
#> [11] Biobase_2.46.0                        
#> [12] ggbio_1.34.0                          
#> [13] BSgenome_1.54.0                       
#> [14] rtracklayer_1.46.0                    
#> [15] Biostrings_2.54.0                     
#> [16] XVector_0.26.0                        
#> [17] ggplot2_3.3.2                         
#> [18] data.table_1.13.2                     
#> [19] RColorBrewer_1.1-2                    
#> [20] GenomicRanges_1.38.0                  
#> [21] GenomeInfoDb_1.22.1                   
#> [22] IRanges_2.20.2                        
#> [23] S4Vectors_0.24.4                      
#> [24] BiocGenerics_0.32.0                   
#> [25] reshape2_1.4.4                        
#> [26] dplyr_1.0.2                           
#> 
#> loaded via a namespace (and not attached):
```

```
#>   [1] colorspace_1.4-1         hwriter_1.3.2            ellipsis_0.3.1
#>   [4] biovizBase_1.34.1        htmlTable_2.1.0          base64enc_0.1-3
#>   [7] dichromat_2.0-0          rstudioapi_0.11          farver_2.0.3
#>  [10] bit64_4.0.5              splines_3.6.0            geneplotter_1.64.0
#>  [13] knitr_1.30               Formula_1.2-4            Rsamtools_2.2.3
#>  [16] annotate_1.64.0          cluster_2.1.0            dbplyr_1.4.4
#>  [19] png_0.1-7                graph_1.64.0             BiocManager_1.30.10
#>  [22] compiler_3.6.0           httr_1.4.2               backports_1.1.10
#>  [25] assertthat_0.2.1         Matrix_1.2-18            lazyeval_0.2.2
#>  [28] htmltools_0.5.0          prettyunits_1.1.1        tools_3.6.0
#>  [31] gtable_0.3.0             glue_1.4.2               GenomeInfoDbData_1.2.2
#>  [34] rappdirs_0.3.1           Rcpp_1.0.5               vctrs_0.3.4
#>  [37] xfun_0.19                stringr_1.4.0            lifecycle_0.2.0
#>  [40] ensembldb_2.10.2         statmod_1.4.35           XML_3.99-0.3
#>  [43] zlibbioc_1.32.0          scales_1.1.1             VariantAnnotation_1.32.0
#>  [46] hms_0.5.3                ProtGenerics_1.18.0      RBGL_1.62.1
#>  [49] AnnotationFilter_1.10.0  yaml_2.2.1               curl_4.3
#>  [52] memoise_1.1.0            gridExtra_2.3            biomaRt_2.42.1
#>  [55] rpart_4.1-15             reshape_0.8.8            latticeExtra_0.6-29
#>  [58] stringi_1.4.6            RSQLite_2.2.1            genefilter_1.68.0
#>  [61] checkmate_2.0.0          rlang_0.4.8              pkgconfig_2.0.3
#>  [64] bitops_1.0-6             evaluate_0.14            lattice_0.20-41
#>  [67] purrr_0.3.4              labeling_0.4.2           GenomicAlignments_1.22.1
#>  [70] htmlwidgets_1.5.2        bit_4.0.4                tidyselect_1.1.0
#>  [73] GGally_2.0.0             plyr_1.8.6               magrittr_1.5
#>  [76] R6_2.4.1                 generics_0.0.2           Hmisc_4.4-1
#>  [79] DBI_1.1.0                pillar_1.4.6             foreign_0.8-71
#>  [82] withr_2.3.0              survival_3.2-7           RCurl_1.98-1.2
#>  [85] nnet_7.3-14              tibble_3.0.4             crayon_1.3.4
#>  [88] OrganismDbi_1.28.0       BiocFileCache_1.10.2     rmarkdown_2.5
#>  [91] jpeg_0.1-8.1             progress_1.2.2           locfit_1.5-9.4
#>  [94] grid_3.6.0               blob_1.2.1               digest_0.6.27
#>  [97] xtable_1.8-4             openssl_1.4.3            munsell_0.5.0
#> [100] askpass_1.1
```