Analyzing NGS data with R/Bioconductor

NGS analysis for gene regulation and epigenomics

Physalia 2021

Jacques Serizay

Recap on Bigwig

- Importing a bigwig is done with rtracklayer::import('….bw')
- You can import a bigwig as a Rle with rtracklayer::import("….bw", as = "Rle")

Recap on GRanges

- `Importing a bed file is done with rtracklayer::import('….bed')`

Genome sequence / Gene annotations

- Were do you get it from? Which database?
  - UCSC?
  - NCBI?
  - Ensembl?
  - Gencode?
  - iGenomes?

Genome sequence / Gene annotations

- UCSC?
- NCBI?
- Ensembl?
- Gencode?
- iGenomes?

The iGenomes are a collection of reference sequences and annotation files for commonly analyzed organisms. The files have been downloaded from Ensembl, NCBI, or UCSC. Chromosome names have been changed to be simple and consistent with the download source. Each iGenome is available as a compressed file that contains sequences and annotation files for a single genomic build of an organism.

Species	Source	Build(s)		
Saccharomyces cerevisiae (Yeast)	Ensembl	R64-1-1	EF4	EF3
	NCBI	build3.1	build2.1	
	UCSC	sacCer3	sacCer2	

https://support.illumina.com/sequencing/sequencing_software/**igenome**.html

Genome sequence / Gene annotations

- UCSC?
- NCBI?
- Ensembl?
- Ge
-

The iGenomes are a collection of refer… files have been downloaded from Ensembl, NCBI, or UCSC. Ch… the download source. Each iGenome is available as a compressed file th… an organism.

		Build(s)		
	Ensembl	R64-1-1	EF4	EF3
	NCBI	build3.1	build2.1	
	UCSC	sacCer3	sacCer2	

KNOW YOUR ORGANISM!

https://support.illumina.com/sequencing/sequencing_software/**igenome**.html

SummarizedExperiment

- An NGS analysis workflow typically involves:
    - Defining features of interest (e.g. gene annotations for RNA-seq or accessibility peaks for ATAC-seq)
    - Counting reads overlapping with each feature
    - Performing differential analysis
    - Extracting results

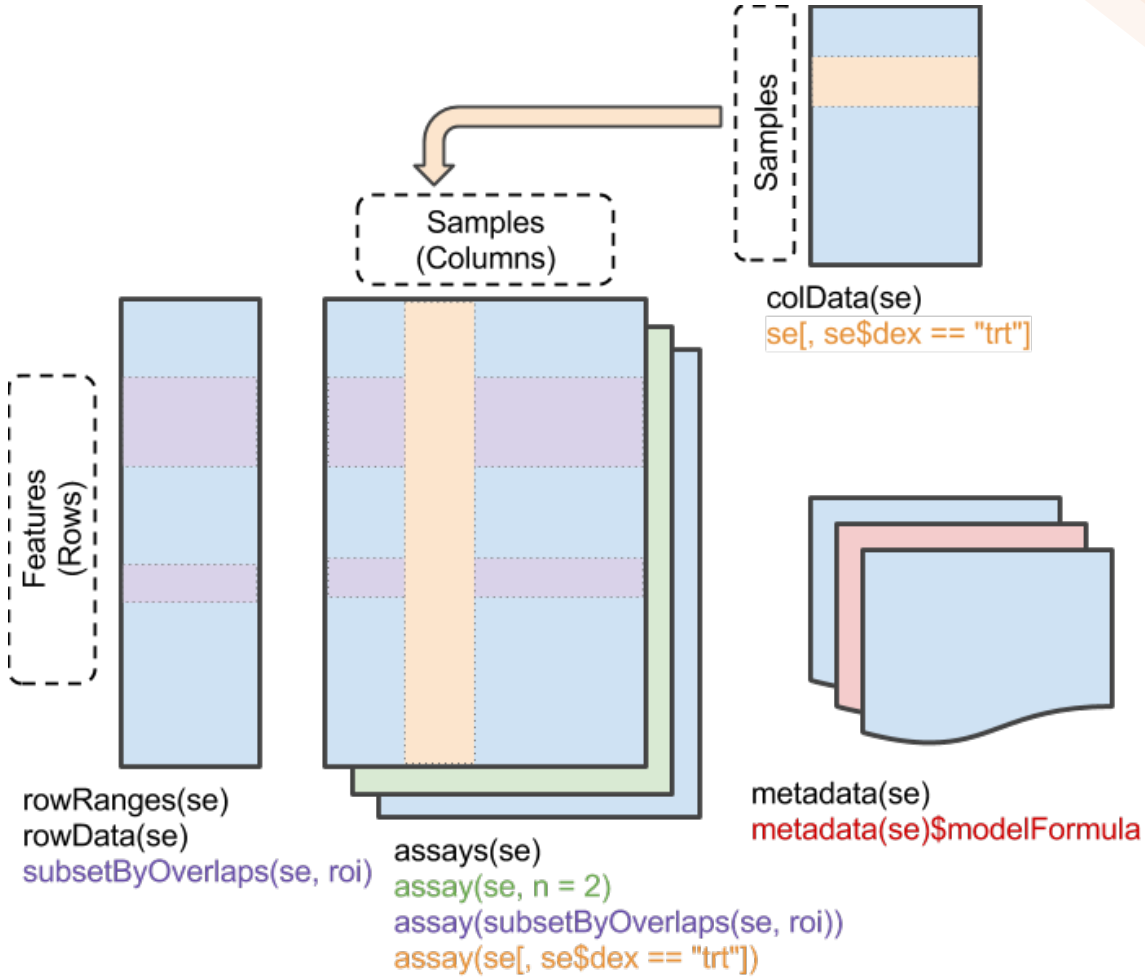
SummarizedExperiment

Published: 29 January 2015

Orchestrating high-throughput genomic analysis with Bioconductor

Wolfgang Huber ✉, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, Raphael Gottardo, Florian Hahne, Kasper D Hansen, Rafael A Irizarry, Michael Lawrence, Michael I Love, James MacDonald, Valerie Obenchain, Andrzej K Oleś, Hervé Pagès, Alejandro Reyes, Paul Shannon, Gordon K Smyth, Dan Tenenbaum, Levi Waldron & Martin Morgan -Show fewer authors

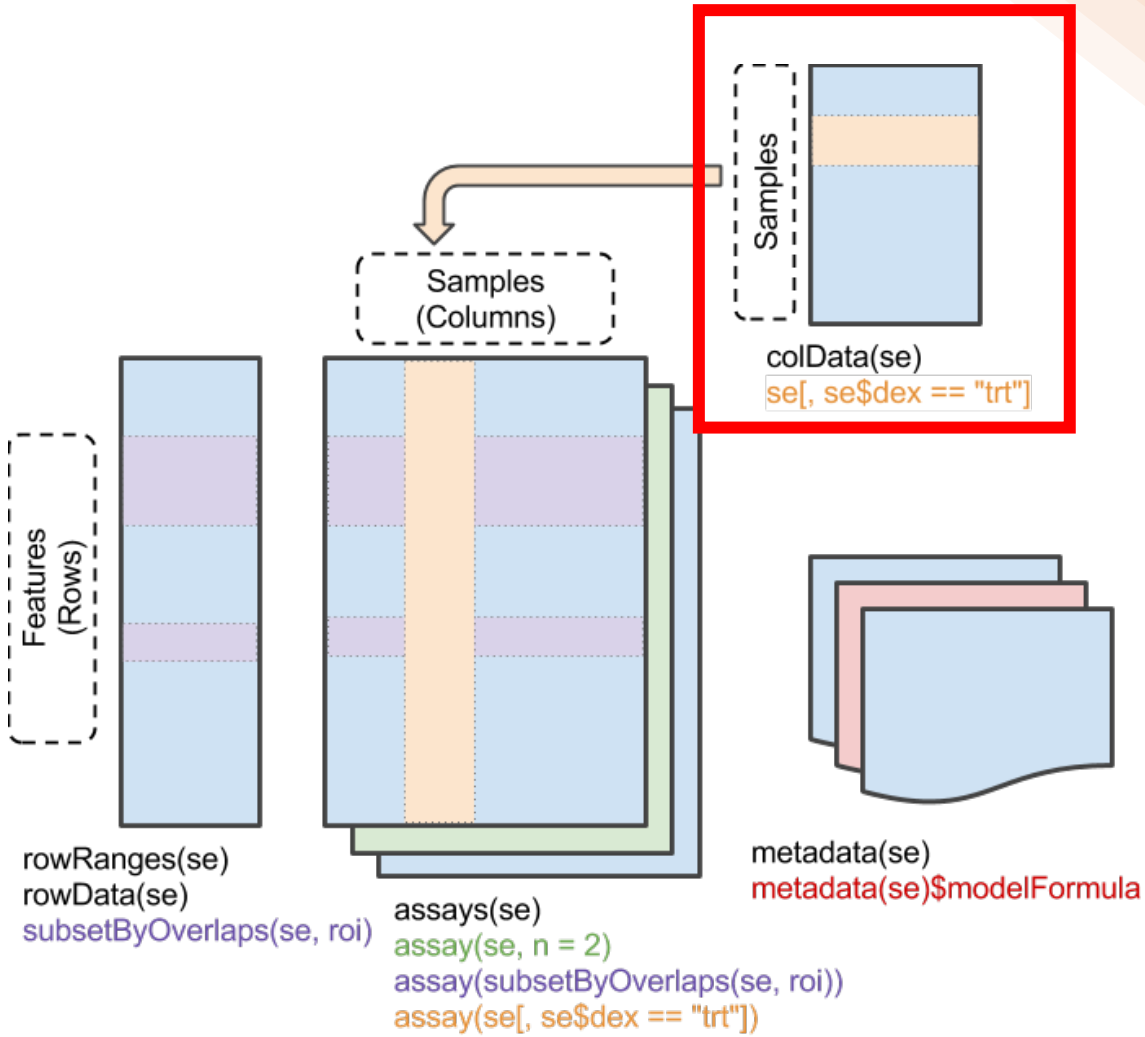
Nature Methods **12**, 115–121(2015) | Cite this article



SummarizedExperiment

- colData(): Annotations on each column, as a DataFrame.

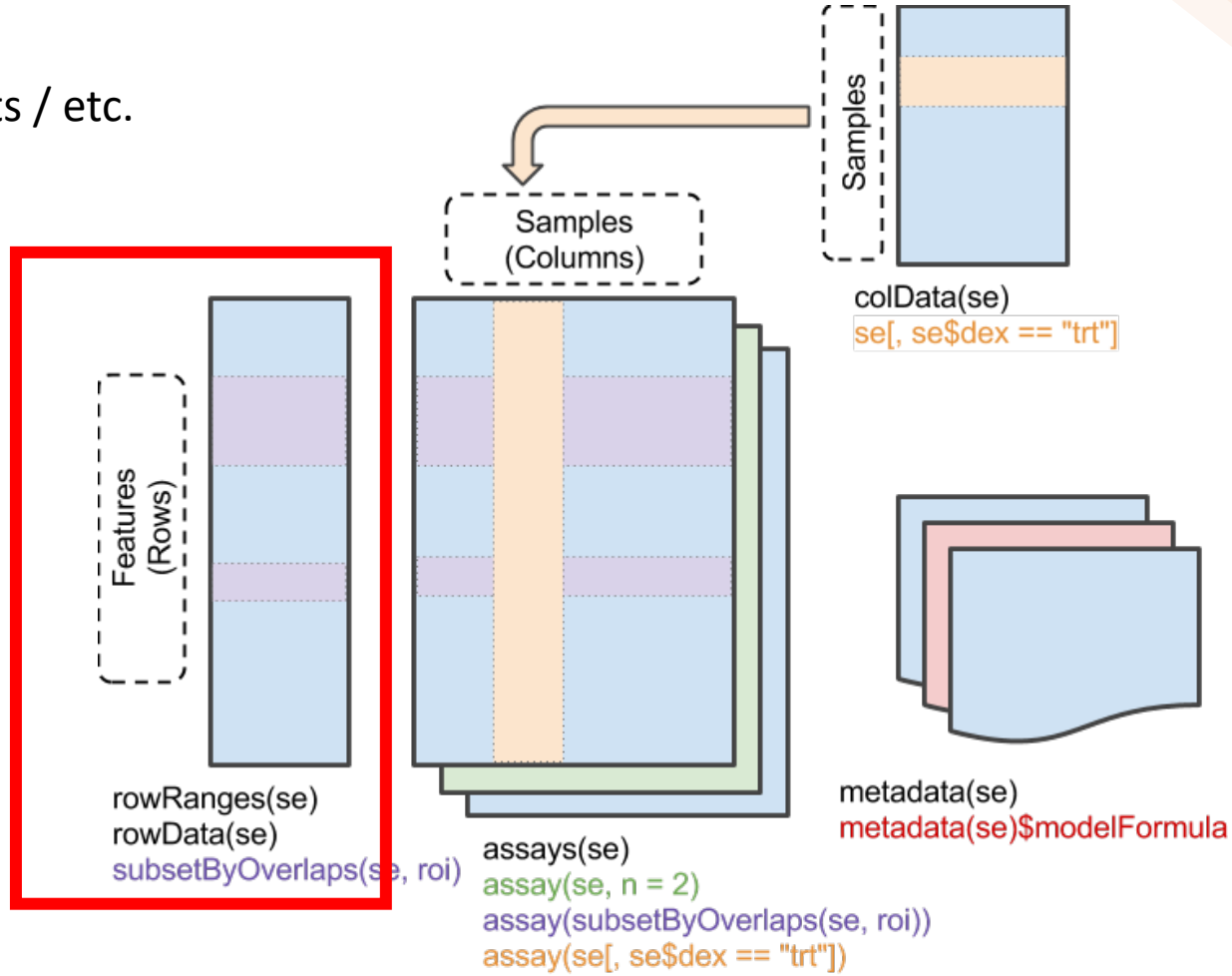
  E.g., description of each sample



SummarizedExperiment

- rowData/rowRanges(): Annotations on each row.

  E.g., coordinates of gene / exons /peaks in transcripts / etc.



SummarizedExperiment

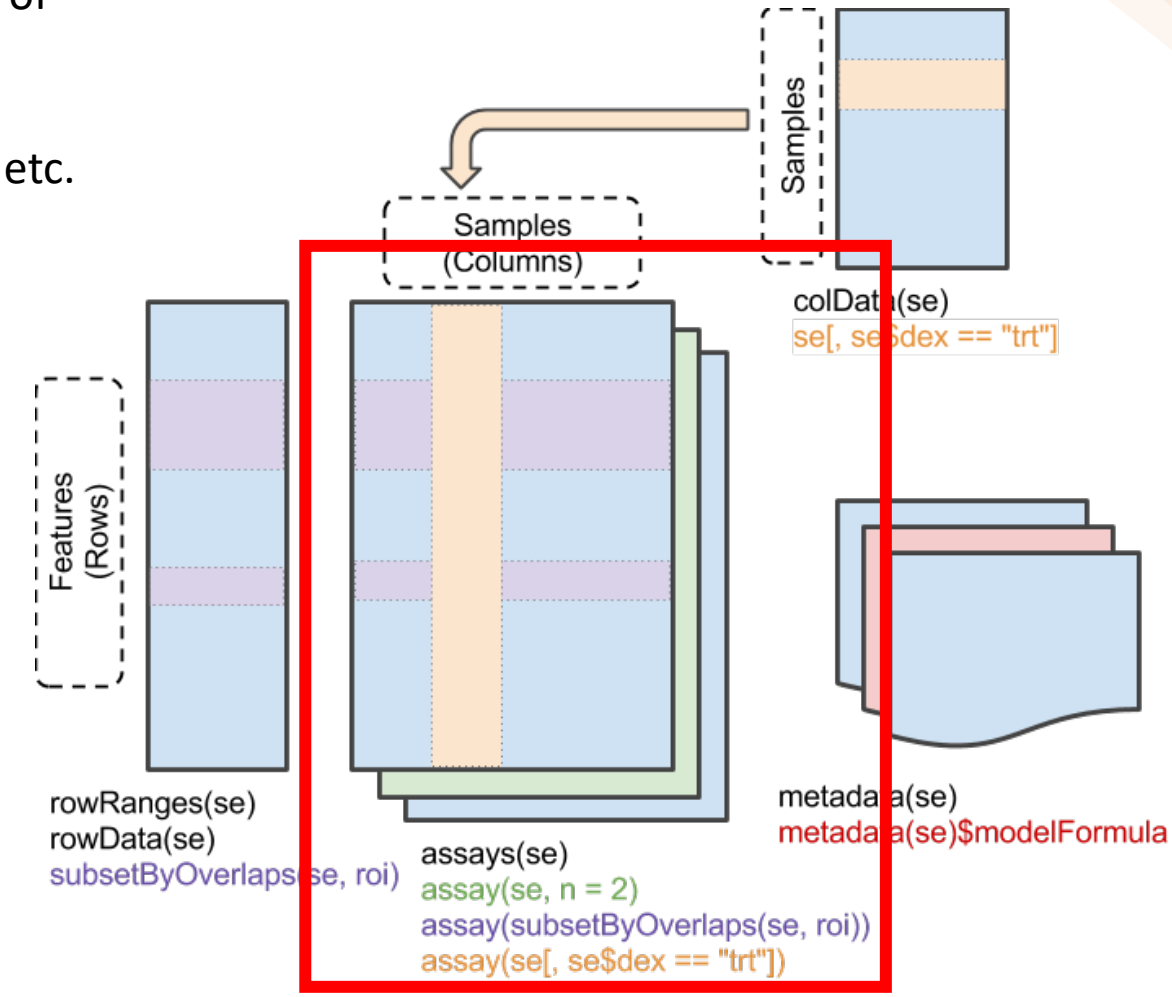
- assay(), assays(): A matrix-like or list of matrix-like objects of identical dimension

  rows: refer to **rowRanges**: genes, genomic coordinates, etc.

  columns: refer to **colData**: samples, cells, etc.

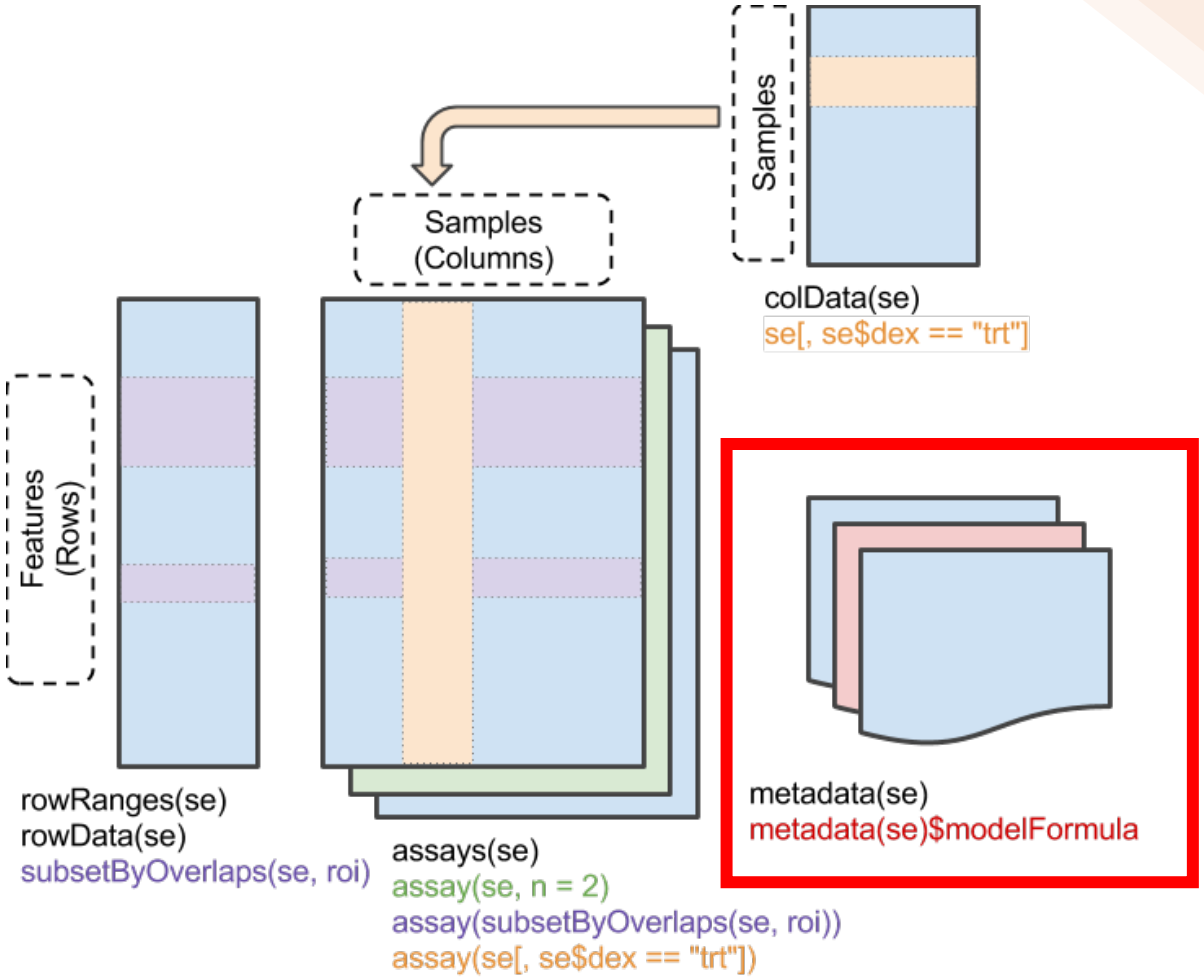
  Implements dim(), dimnames() and 2-dimensional [,]

  Can be several assays!!!



SummarizedExperiment

- metadata(): List of unstructured metadata describing the overall content of the object.



BPPARAM

- Genomic analyses require heavy resources
- Generally, benefits from parallelization

BPPARAM

- Genomic analyses require heavy resources
- Generally, benefits from parallelization

- BiocParallel is a Bioconductor package designed to reduce the complexity faced when developing and using software that performs parallel computations

- BiocParallel aims to provide a unified interface to existing parallel infrastructures where code can be easily executed in different environments

BPPARAM

- Declaring configurations:

registered()

bpparam()

register(…, default = TRUE)

```
 > BiocParallel::registered()
$MulticoreParam
class: MulticoreParam
  bpisup: FALSE; bpnworkers: 6; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA
  cluster type: FORK

$SnowParam
class: SnowParam
  bpisup: FALSE; bpnworkers: 6; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA
  cluster type: SOCK

$SerialParam
class: SerialParam
  bpisup: FALSE; bpnworkers: 1; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA

 > BiocParallel::bpparam()
class: MulticoreParam
  bpisup: FALSE; bpnworkers: 6; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA
  cluster type: FORK
```

BPPARAM

- Declaring configurations:

registered()

bpparam()

register(…, default = TRUE)

MulticoreParam()

SerialParam()

SnowParam()

```
 > BiocParallel::register(MulticoreParam(workers = 4), default = TRUE)
 > BiocParallel::registered()
$MulticoreParam
class: MulticoreParam
  bpisup: FALSE; bpnworkers: 4; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA
  cluster type: FORK

$SnowParam
class: SnowParam
  bpisup: FALSE; bpnworkers: 6; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA
  cluster type: SOCK

$SerialParam
class: SerialParam
  bpisup: FALSE; bpnworkers: 1; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA

 > BiocParallel::bpparam()
class: MulticoreParam
  bpisup: FALSE; bpnworkers: 4; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA
  cluster type: FORK
```

BPPARAM

- Execute in parallel:

bplapply()	bplapply(1:4, FUN)
bpiterate()	bpiterate(ITER, FUN, ...)

```
> BiocParallel::registered()
$MulticoreParam
class: MulticoreParam
  bpisup: FALSE; bpnworkers: 6; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA
  cluster type: FORK

$SnowParam
class: SnowParam
  bpisup: FALSE; bpnworkers: 6; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA
  cluster type: SOCK

$SerialParam
class: SerialParam
  bpisup: FALSE; bpnworkers: 1; bptasks: 0; bpjobname: BPJOB
  bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
  bpRNGseed: ; bptimeout: 2592000; bpprogressbar: FALSE
  bpexportglobals: TRUE
  bplogdir: NA
  bpresultdir: NA
```

Additional packages in Bioconductor

- Rsamtools: interacting with BAM files
- GenomicAlignements: counting BAM files over Granges
- Many others….

	Package	Maintainer	Title	Rank
	BiocVersion	Bioconductor Package Maintainer	Set the appropriate version of Bioconductor packages	1
	BiocGenerics	Bioconductor Package Maintainer	S4 generic functions used in Bioconductor	2
*	S4Vectors	Bioconductor Package Maintainer	Foundation of vector-like and list-like containers in Bioconductor	3
*	IRanges	Bioconductor Package Maintainer	Foundation of integer range manipulation in Bioconductor	4
	Biobase	Bioconductor Package Maintainer	Biobase: Base functions for Bioconductor	5
	zlibbioc	Bioconductor Package Maintainer	An R packaged zlib-1.2.5	6
	XVector	Hervé Pagès	Foundation of external vector representation and manipulation in Bioconductor	7
*	AnnotationDbi	Bioconductor Package Maintainer	Manipulation of SQLite-based annotations in Bioconductor	8
*	GenomeInfoDb	Bioconductor Package Maintainer	Utilities for manipulating chromosome names, including modifying them to follow a particular naming style	9
*	BiocParallel	Bioconductor Package Maintainer	Bioconductor facilities for parallel evaluation	10
	DelayedArray	Hervé Pagès	A unified framework for working transparently with on-disk and in-memory array-like datasets	11
*	SummarizedExperiment	Bioconductor Package Maintainer	SummarizedExperiment container	12
*	GenomicRanges	Bioconductor Package Maintainer	Representation and manipulation of genomic intervals	13
	limma	Gordon Smyth	Linear Models for Microarray Data	14
*	Biostrings	H. Pagès	Efficient manipulation of biological strings	15
*	biomaRt	Mike Smith	Interface to BioMart databases (i.e. Ensembl)	16
	annotate	Bioconductor Package Maintainer	Annotation for microarrays	17
*	Rsamtools	Bioconductor Package Maintainer	Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import	18
	genefilter	Bioconductor Package Maintainer	genefilter: methods for filtering genes from high-throughput experiments	19
	graph	Bioconductor Package Maintainer	graph: A package to handle graph data structures	20
*	GenomicAlignments	Bioconductor Package Maintainer	Representation and manipulation of short genomic alignments	21
	BiocFileCache	Lori Shepherd	Manage Files Across Sessions	22
	Rhtslib	Bioconductor Package Maintainer	HTSlib high-throughput sequencing library as an R package	23
	edgeR	Yunshun Chen, Gordon Smyth, Aaron Lun, Mark Robinson	Empirical Analysis of Digital Gene Expression Data in R	24
*	rtracklayer	Michael Lawrence	R interface to genome annotation files and the UCSC genome browser	25
*	GenomicFeatures	Bioconductor Package Maintainer	Conveniently import and query gene models	26
*	DESeq2	Michael Love	Differential gene expression analysis based on the negative binomial distribution	27
	Rhdf5lib	Mike Smith	hdf5 library as an R package	28
	geneplotter	Bioconductor Package Maintainer	Graphics related functions for Bioconductor	29

DESeq2 analysis workflow

- 1) Which files??
  - Need alignment files (.bam files) → Typically listed in a sample sheet

sample	timepoint	replicate	bam
SacCer3_00_rep1	00	1	Share/day04/_bam-files//SacCer3_00_rep1_SRR1822155_noChrM_q10.bam
SacCer3_00_rep2	00	2	Share/day04/_bam-files//SacCer3_00_rep2_SRR1822156_noChrM_q10.bam
SacCer3_15_rep1	15	1	Share/day04/_bam-files//SacCer3_15_rep1_SRR1822157_noChrM_q10.bam
SacCer3_15_rep2	15	2	Share/day04/_bam-files//SacCer3_15_rep2_SRR1822158_noChrM_q10.bam
SacCer3_30_rep1	30	1	Share/day04/_bam-files//SacCer3_30_rep1_SRR1822159_noChrM_q10.bam
SacCer3_30_rep2	30	2	Share/day04/_bam-files//SacCer3_30_rep2_SRR1822160_noChrM_q10.bam
SacCer3_45_rep1	45	1	Share/day04/_bam-files//SacCer3_45_rep1_SRR1822161_noChrM_q10.bam
SacCer3_45_rep2	45	2	Share/day04/_bam-files//SacCer3_45_rep2_SRR1822162_noChrM_q10.bam
SacCer3_60_rep1	60	1	Share/day04/_bam-files//SacCer3_60_rep1_SRR1822163_noChrM_q10.bam
SacCer3_60_rep2	60	2	Share/day04/_bam-files//SacCer3_60_rep2_SRR1822164_noChrM_q10.bam

DESeq2 analysis workflow

- 1) Which files??
    - Need genomic features → Typically imported in R as Granges from .gtf or .bed

```
peaks <- read.table('Share/day04/_yapc-files/stress-response_0.05.bed') %>%
        as_tibble() %>%
        setNames(c('seqnames', 'start', 'end', 'yapc_scores', 'score', 'strand', 'summit', 'summit2')) %>%
        GenomicRanges::makeGRangesFromDataFrame(keep.extra.columns = TRUE)
```

DESeq2 analysis workflow

```
> peaks
GRanges object with 6513 ranges and 2 metadata columns:
         seqnames        ranges strand |    summit      peakID
            <Rle>     <IRanges>  <Rle> | <integer> <character>
     [1]        I       78-552      * |        78      peak_1
     [2]        I      1193-1481    * |      1193      peak_2
     [3]        I      2415-2738    * |      2415      peak_3
     [4]        I      6342-6539    * |      6342      peak_4
     [5]        I      9304-9545    * |      9304      peak_5
     ...      ...            ...  ... .       ...         ...
  [6509]      XVI 941520-941693      * |    941520   peak_6509
  [6510]      XVI 942541-942742      * |    942541   peak_6510
  [6511]      XVI 943128-943286      * |    943128   peak_6511
  [6512]      XVI 944265-944521      * |    944265   peak_6512
  [6513]      XVI 946186-946398      * |    946186   peak_6513
  -------
  seqinfo: 16 sequences from an unspecified genome; no seqlengths
```

DESeq2 analysis workflow

- 2) Need to count reads mapping over each features
    - We can use `GenomicAlignments::summarizeOverlaps()`

```
counts <- GenomicAlignments::summarizeOverlaps(,
        features = peaks,
        reads = Rsamtools::BamFileList(sampleTable$bam ,
        ignore.strand = TRUE,
        singleEnd = FALSE
)
```

DESeq2 analysis workflow

- 3) Need to perform differential coverage analysis
    - We can use `DESeq2::DESeq()`

```
library(DESeq2)
dds <- DESeqDataSet(counts, design = ~ timepoint)
dds <- DESeq(dds)
```

DESeq2 analysis workflow

- 3) Need to perform differential coverage analysis
  - We can use `DESeq2::DESeq()`

```
library(DESeq2)
dds <- DESeqDataSet(counts, design = ~ timepoint)
dds <- DESeq(dds)
```

```
> colData(counts)
DataFrame with 10 rows and 4 columns
         sample   timepoint replicate              bam
     <character> <character> <numeric>      <character>
1  SacCer3_00_rep1          00         1 Share/day04/_bam-fil..
2  SacCer3_00_rep2          00         2 Share/day04/_bam-fil..
3  SacCer3_15_rep1          15         1 Share/day04/_bam-fil..
4  SacCer3_15_rep2          15         2 Share/day04/_bam-fil..
5  SacCer3_30_rep1          30         1 Share/day04/_bam-fil..
6  SacCer3_30_rep2          30         2 Share/day04/_bam-fil..
7  SacCer3_45_rep1          45         1 Share/day04/_bam-fil..
8  SacCer3_45_rep2          45         2 Share/day04/_bam-fil..
9  SacCer3_60_rep1          60         1 Share/day04/_bam-fil..
10 SacCer3_60_rep2          60         2 Share/day04/_bam-fil..
```

DESeq2 analysis workflow

- 3) Need to perform differential coverage analysis

```
> dds <- DESeqDataSet(counts, design = ~ timepoint)
> dds <- DESeq(dds)
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
> dds
class: DESeqDataSet
dim: 6513 10
metadata(1): version
assays(4): counts mu H cooks
rownames: NULL
rowData names(37): summit peakID ... deviance maxCooks
colnames: NULL
colData names(5): sample timepoint replicate bam sizeFactor
```

DESeq2 analysis workflow

- 4) Extract results
    - We can use `DESeq2::results()`
    - DESeq2 extracts results for pairwise comparison between 2 conditions

```
> contrasts
   15_v_00   30_v_00   45_v_00   60_v_00   30_v_15   45_v_15   60_v_15   45_v_30   60_v_30   60_v_45
[1,] "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint"
[2,] "15"      "30"      "45"      "60"      "30"      "45"      "60"      "45"      "60"      "60"
[3,] "00"      "00"      "00"      "00"      "15"      "15"      "15"      "30"      "30"      "45"
```

DESeq2 analysis workflow

```
> contrasts
    15_v_00    30_v_00    45_v_00    60_v_00    30_v_15    45_v_15    60_v_15    45_v_30    60_v_30    60_v_45
[1,] "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint"
[2,] "15"       "30"       "45"       "60"       "30"       "45"       "60"       "45"       "60"       "60"
[3,] "00"       "00"       "00"       "00"       "15"       "15"       "15"       "30"       "30"       "45”

> results(dds, contrast = contrasts[, 2])
log2 fold change (MLE): timepoint 30 vs 00
Wald test p-value: timepoint 30 vs 00
DataFrame with 6513 rows and 6 columns
     baseMean log2FoldChange     lfcSE      stat      pvalue       padj
    <numeric>      <numeric> <numeric>  <numeric>   <numeric>  <numeric>
1   1643.7457      -0.900131  0.225114   -3.99856 6.37297e-05 0.002809493
2    106.1707      -0.876365  0.313066   -2.79930 5.12139e-03 0.066589377
3    134.4141      -0.753498  0.311437   -2.41942 1.55451e-02 0.133487340
4    203.8053      -1.094019  0.227780   -4.80296 1.56338e-06 0.000140911
5     67.2922      -1.001061  0.336647   -2.97362 2.94309e-03 0.046569804
...       ...            ...       ...        ...         ...        ...
6509   36.1437    -0.0313828  0.417826 -0.0751098   0.9401274   0.979001
6510  603.5468     -0.4020281  0.181778 -2.2116386   0.0269916   0.185475
6511   42.8792    -0.3776449  0.378523 -0.9976807   0.3184342   0.633629
6512   15.2541    -0.5818438  0.642764 -0.9052221   0.3653477         NA
6513   37.8941    -0.8396196  0.427293 -1.9649731   0.0494173   0.258224
```

DESeq2 analysis workflow

```
> contrasts
     15_v_00   30_v_00   45_v_00   60_v_00   30_v_15   45_v_15   60_v_15   45_v_30   60_v_30   60_v_45
[1,] "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint"
[2,] "15"        "30"        "45"        "60"        "30"        "45"        "60"        "45"        "60"        "60"
[3,] "00"        "00"        "00"        "00"        "15"        "15"        "15"        "30"        "30"        "45"

> results(dds, contrast = contrasts[, 2])
log2 fold change (MLE): timepoint 30 vs 00
Wald test p-value: timepoint 30 vs 00
DataFrame with 6513 rows and 6 columns
     baseMean log2FoldChange     lfcSE      stat      pvalue       padj
    <numeric>      <numeric> <numeric> <numeric>   <numeric>  <numeric>
1    1643.7457      -0.900131  0.225114  -3.99856 6.37297e-05 0.002809493
2     106.1707      -0.876365  0.313066  -2.79930 5.12139e-03 0.066589377
3     134.4141      -0.753498  0.311437  -2.41942 1.55451e-02 0.133487340
4     203.8053      -1.094019  0.227780  -4.80296 1.56338e-06 0.000140911
5      67.2922      -1.001061  0.336647  -2.97362 2.94309e-03 0.046569804
...         ...            ...       ...       ...         ...        ...
6509   36.1437    -0.0313828  0.417826 -0.0751098   0.9401274   0.979001
6510  603.5468     -0.4020281  0.181778 -2.2116386   0.0269916   0.185475
6511   42.8792     -0.3776449  0.378523 -0.9976807   0.3184342   0.633629
6512   15.2541     -0.5818438  0.642764 -0.9052221   0.3653477         NA
6513   37.8941     -0.8396196  0.427293 -1.9649731   0.0494173   0.258224
```

DESeq2 analysis workflow

```
> contrasts
     15_v_00    30_v_00    45_v_00    60_v_00    30_v_15    45_v_15    60_v_15    45_v_30    60_v_30    60_v_45
[1,] "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint"
[2,] "15"       "30"       "45"       "60"       "30"       "45"       "60"       "45"       "60"       "60"
[3,] "00"       "00"       "00"       "00"       "15"       "15"       "15"       "30"       "30"       "45"

> results(dds, contrast = contrasts[, 2])
log2 fold change (MLE): timepoint 30 vs 00
Wald test p-value: timepoint 30 vs 00
DataFrame with 6513 rows and 6 columns
     baseMean log2FoldChange     lfcSE      stat     pvalue       padj
    <numeric>      <numeric> <numeric> <numeric>  <numeric>  <numeric>
1   1643.7457      -0.900131  0.225114  -3.99856 6.37297e-05 0.002809493
2    106.1707      -0.876365  0.313066  -2.79930 5.12139e-03 0.066589377
3    134.4141      -0.753498  0.311437  -2.41942 1.55451e-02 0.133487340
4    203.8053      -1.094019  0.227780  -4.80296 1.56338e-06 0.000140911
5     67.2922      -1.001061  0.336647  -2.97362 2.94309e-03 0.046569804
...       ...            ...       ...       ...        ...        ...
6509   36.1437   -0.0313828  0.417826 -0.0751098  0.9401274   0.979001
6510  603.5468    -0.4020281  0.181778 -2.2116386  0.0269916   0.185475
6511   42.8792   -0.3776449  0.378523 -0.9976807  0.3184342   0.633629
6512   15.2541   -0.5818438  0.642764 -0.9052221  0.3653477         NA
6513   37.8941   -0.8396196  0.427293 -1.9649731  0.0494173   0.258224
```

DESeq2 analysis workflow

```
> contrasts
     15_v_00    30_v_00    45_v_00    60_v_00    30_v_15    45_v_15    60_v_15    45_v_30    60_v_30    60_v_45
[1,] "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint" "timepoint"
[2,] "15"       "30"       "45"       "60"       "30"       "45"       "60"       "45"       "60"       "60"
[3,] "00"       "00"       "00"       "00"       "15"       "15"       "15"       "30"       "30"       "45"

> results(dds, contrast = contrasts[, 2])
log2 fold change (MLE): timepoint 30 vs 00
Wald test p-value: timepoint 30 vs 00
DataFrame with 6513 rows and 6 columns
     baseMean log2FoldChange     lfcSE      stat      pvalue       padj
    <numeric>      <numeric> <numeric> <numeric>   <numeric>  <numeric>
1   1643.7457      -0.900131  0.225114  -3.99856 6.37297e-05 0.002809493
2    106.1707      -0.876365  0.313066  -2.79930 5.12139e-03 0.066589377
3    134.4141      -0.753498  0.311437  -2.41942 1.55451e-02 0.133487340
4    203.8053      -1.094019  0.227780  -4.80296 1.56338e-06 0.000140911
5     67.2922      -1.001061  0.336647  -2.97362 2.94309e-03 0.046569804
...        ...            ...       ...       ...         ...        ...
6509   36.1437   -0.0313828  0.417826 -0.0751098   0.9401274   0.979001
6510  603.5468    -0.4020281  0.181778 -2.2116386   0.0269916   0.185475
6511   42.8792   -0.3776449  0.378523 -0.9976807   0.3184342   0.633629
6512   15.2541   -0.5818438  0.642764 -0.9052221   0.3653477         NA
6513   37.8941   -0.8396196  0.427293 -1.9649731   0.0494173   0.258224
```

Regularized log counts

- DESeq2 is primarily designed to **estimate fold-change** between conditions, **NOT** actual abundance!
- Rlog (regularized log) can be used to approximate gene expression levels

Regularized log counts

- Rlog transforms the count data to the **log2 scale** in a way which

    1) minimizes differences between samples for rows with small counts

    2) normalizes with respect to library size

Regularized log counts

```
> rlogs <- rlog(dds, blind = FALSE)
> rlogs
class: DESeqTransform
dim: 6513 10
metadata(1): version
assays(1): ''
rownames: NULL
rowData names(38): summit peakID ... maxCooks rlogIntercept
colnames(10): 1 2 ... 9 10
colData names(5): sample timepoint replicate bam sizeFactor
> dim(assay(rlogs, 1))
[1] 6513   10
```

Final results

- Mean of rlog values across replicates
- Fold-changes between relevant timepoints
- Associated adjusted p-values

```
> glimpse(results)
Rows: 6,513
Columns: 25
$ rlog_00      <dbl> 11.180315, 6.908934,
$ rlog_15      <dbl> 10.438408, 6.729950,
$ rlog_30      <dbl> 10.550133, 6.471479,
$ rlog_45      <dbl> 10.547565, 6.705362,
$ rlog_60      <dbl> 10.335557, 6.664081,
$ L2FC_15_v_00 <dbl> -1.062311696, -0.3507
$ L2FC_30_v_00 <dbl> -0.90013109, -0.87636
$ L2FC_45_v_00 <dbl> -0.902906533, -0.3446
$ L2FC_60_v_00 <dbl> -1.17312240, -0.47517
$ L2FC_30_v_15 <dbl> 0.162180610, -0.52560
$ L2FC_45_v_15 <dbl> 0.1594051627, 0.00614
$ L2FC_60_v_15 <dbl> -0.110810706, -0.1244
$ L2FC_45_v_30 <dbl> -0.0027754472, 0.5317
$ L2FC_60_v_30 <dbl> -0.272991316, 0.40118
$ L2FC_60_v_45 <dbl> -0.2702158684, -0.130
$ padj_15_v_00 <dbl> 9.503337e-05, 5.06491
$ padj_30_v_00 <dbl> 2.809493e-03, 6.65893
$ padj_45_v_00 <dbl> 4.710445e-03, 6.36697
$ padj_60_v_00 <dbl> 2.520754e-05, 3.46970
$ padj_30_v_15 <dbl> 8.704961e-01, 5.71741
$ padj_45_v_15 <dbl> 9.800032e-01, 9.99908
$ padj_60_v_15 <dbl> 0.9143673089, 0.92795
$ padj_45_v_30 <dbl> 0.9986153, 0.9986153,
$ padj_60_v_30 <dbl> 0.729105509, 0.708307
$ padj_60_v_45 <dbl> 0.2818529, NA, NA, NA
```