**## Bioconductor packages for Sequencing anlaysis {#highlight1 .smaller}**

- Data structures: IRanges, GenomicRanges, Biostrings, BSgenome

- Input/Ouput: ShortRead, Rsamtools, GenomicAlignments and rtracklayer (GTF,GFF,BED)

- Annotation: GenomicFeatures, BSgenome, biomaRt, TxDb.\*, org.\*

- Alignment: Rsubread, Biostrings

- Accessing Database: SRAdb & GEOquery

- ChIP-seq peak identification, motif discovery and annotation: ChIPQC, chipseq, ChIPseqR, ChIPpeakAnno, DiffBind, rGADEM, BayesPeak, MotifDb, SeqLogo.

- RNA-seq and Differential expression analysis: Rsubread, GenomicAlignments, edgeR, DESeq2, DEXseq, goseq

- SNP: snpStats, SeqVarTools, GGtools

- Work-flows: ReportingTools, easyRNASeq, ArrayExpressHTS, oneChannelGUI

**## File formats in NGS**

- FASTQ (Fasta with quality)

- SAM/BAM -

- BED

- Wiggle/bedgraph

**## BAM/SAM {.smaller}**

BAM/SAM consist two sections: Header and Alignment

Header:

Meta data (reference genome, aligner), starts with “@”

Alignment:

- <b>QNAME</b>: ID of the read (“query”)

- <b>FLAG</b>: alignment flags

- <b>RNAME</b>: ID of the reference (typically: chromosome name)

- <b>POS</b>: Position in reference (1-based, left side)

- <b>MAPQ</b>: Mapping quality (as Phred score)

- <b>CIGAR</b>: Alignment description (mismatch, gaps etc.)

- <b>RNEXT</b>: Mate/next read reference sequence name

- <b>MPOS</b>: Mate/next read position

- <b>TLEN</b>: observed Template Length

- <b>SEQ</b>: sequence of the read

- <b>QUAL</b>: quality string of the read

## BioC Annotation Packages{.smaller}

Annotation packages can be broadly classified in to gene-centric, genome-centric and web-based annotation packages

Gene-centric annotation packages (AnnotationDbi):

- Organism level packages: contains gene annotation for entire organism. Follows “org.XX.YY.db” naming pattern (Ex: org.Hs.eg.db)

- General System biology data: KEGG.db (association between pathways and genes), GO.db (Gene ontology term and genes) and ReactomeDb

- Platform level packages: Annotation for a specific platform (ex: Affymetrix HGU133A microarray).

Genomic-centric GenomicFeatures packages:

- TranscriptDB contains chromosomal positions for genes in the entire transcriptome specific to a genome build, ex: TxDb.Hsapiens.UCSC.hg19.knownGene. These packages allow access to various features on transcriptome, including exons, genes and transcripts coordinates

Web-based annotation services:

-Annotation web services: Retrieving annotation from the web. biomaRt provides interface to query web-based `biomart' resource for genes, sequence, SNPs, and etc.

## AnnotationDbi Accessor Functions

- <b>columns </b>What kind of annotation available in AnnotationDb object.

- <b>keytypes </b>Displays which type of identifiers can be passed in to select function.

- <b>keys </b> returns keys (index) for the database contained in the AnnotationDb object. Its used along with keytypes in select function to retrieve interested annotation

- <b>select</b> will retrieve the annotation data as a data.frame based on the supplied keys, keytypes and columns.

Package name = Annotation object

We will see an example of how to retrieve annotation from gene-centric organism level annotation package (org.Hs.eg.db)

## TranscriptDB (TxDb) packages

- TxDb packages provide access genomic coordinates to various transcript-related features from UCSC and Biomart data sources.

- TxDb objects contains relationship between mRNA transcripts, exons, CDS and their associated identifiers

- Txdb objects can be accessed using GenomicFeatures package

- TxDb packages follows specific naming scheme, ex: TxDb.Mmusculus.UCSC.mm9.knownGene

We will explore TxDb package for Mouse mm9 genome from UCSC. We will first install the package and load in to our current working space.

## TxDb.Mmusculus.UCSC.mm9.knownGene

By default, the annotation object will have same name as package name. We will create an alias for convenience.

## TxDb.Mmusculus.UCSC.mm9.knownGene

Most common operations performed on TxDb objects are retrieving exons, transcripts and CDS genomic coordinates. The functions transcripts, exons, and cds return the coordinates for the group as GRanges objects.

## TxDb.Mmusculus.UCSC.mm9.knownGene

TxDb package also provides interface to access how genomic features are related to each other. Ex: Access all transcripts or exons associated to a gene. Such grouping can be achieved by transcriptsBy, exonsBy, and cdsBy functions

## TxDb.Mmusculus.UCSC.mm9.knownGene {.smaller}

Other interesting functions: intronsByTranscript, fiveUTRsByTranscript and threeUTRsByTranscript

We can save the annotation object and label them appropriately to facilitate reproducible research:

GenomicFeatures also provides functions to create TxDb objects directly from UCSC and Biomart databases: <b>makeTranscriptDbFromBiomart</b> and <b>makeTranscriptDbFromUCSC</b>

##Annotations from the web – biomaRt

biomaRt package oﬀers access to biomart based online annotation resource referred as ‘marts’. Each mart has several datasets. getBM function can be used to retrieve annotation from the biomarts.

## GenomicRanges

- GenomicRanges provides data structure for efficiently storing genomic coordinates

- Collection of genes coordinates

- Transcription factor binding sites (ChIP-Seq peaks)

- Collection of aligned sequencing reads

- Builds on top of Interval Ranges (IRanges) package and lays foundation for NGS analysis.

- IRanges are collection of integer interval and GenomicRanges extends IRanges by including chromosome and strand.

- Provides collection of functions for accessing and manipulating Genomic coordinates

- Use cases: Identifying TF binding overlap, counting sequencing reads overlap with a gene

- Main classes: GRanges and GRangesList

## Run Length Encoding (RLE)

- Run length encoding is a data compression technique

- Efficiently encoding the redundant information – especially genomics coordinates

- <b>Rle</b> function used to create Rle instance

The above Rle can be interpreted as a run of length 1 of chr1, followed by run length of 3 of chr2, followed by run length of 2 of chr1 and followed by run length of 4 of chr3

Coverage: Number of ranges over each nucleotide. GenomicRanges computes the coverage of GRanges object and returns as Rle object.

## Constructing GRanges object

GRanges class represents a collection of genomic features with single start and end location on the genome. GRanges object can be cretated using <b>GRanges</b> function.

## Constructing GRanges object

- The coordinates in GRanges are 1-based and left-most (start of a read will always be left-most coordinate of the read regardless of which strand the read aligned to).

- Additional data stored beyond genomic coordinates (separated by “|”) are called metadata. In our case metadata column contains score and GC content.

- Metadata columns are optional and can be extracted from GRanges object using <b>mcols</b> function.

## GenomicRangesList

- To represent hierarchical structured data, ex: Exons in a transcript

- List-like data structure

- Each element of the list is GRanges instance

## Operations on GenomicRanges

Opetaions Functions/methods

------------- ---------------------------

Accessors start, end, width, names, mcols, length

Extraction GR[i], GRL[[i]], head, tail

Overlaps findOverlaps, countOverlaps, nearest, precede, follow

Arithmetic shift, resize, distance, distanceToNearest

## Finding overlapping regions

- One of the common tasks in NGS data analysis

- Ex: Identify transcription factor binding sites overlap with promoters

- <b>findOverlaps</b> function find intervals overlap between two GRanges object.

- First argument is query and second argument is subject

- Function usage: \*\*<b>function(query,subject)</b>\*\*

- Output of findOverlaps is a 'Hits' object indicating which of the query and subject overlap.

## Finding overlapping regions

</b> subsetByOverlaps </b> extracts the query intervals overlap with subject intervals

## Finding overlapping regions

Other interesting functions: <b>nearest</b> and <b>distanceToNearest</b>

## Counting overlapping regions

<b> countOverlaps</b> tabulates number of subject intervals overlap with each interval in query, ex: counting number of sequencing reads overlap with genes in RNA-Seq

## Computing Coverage

<b>coverage</b> calculates how many ranges overlap individual positions in the genome. <b>coverage</b> function returns the coverage as Rle instance.

## Reading Sequence alignments (BAM/SAM)

Methods for reading BAM/SAM

-readAligned from ShortRead package

– Accept multiple formats – BAM, export

- Reads all files in a directory

- Reads base call qualities, chromosome, position, and strand

-scanBam from Rsamtools package

- readGAlignments

## Reading Sequence alignments (BAM/SAM)

- GenomicRanges package defines the GAlignments class – a specialised class for storing set of genomic alignments (ex: NGS data)

- GenomicAlignments package defines efficient ways of reading and storing short genomic alignments as GAlignments objects

- Only BAM support now – future version may include other formats

- The readGAlignments function takes an additional argument, <b>param</b> allowing the user to customise which regions and which fields to read from BAM

## Tips for using BioC

## GenomicRanges

* Rle – data compression technique - efficiently encoding the redundant information
* Coverage

- Operations on GRanges

## Accessing and manipulating BAM/SAM

## Visualisation

Exercises:

1. Print few Uniprot identifiers from org.Hs.eg.db
2. Print all chromosomes from org.Mm.eg.db
3. Retrieve gene name, chromosome and Ensembl gene identifiers for “HEBP2” and “PRND”
4. Retrieve gene symbol and chromosome coordinates for genes in chromosome 2 for human using org.Hs.eg.db
5. Retrieve gene annotation for genes in chromosome 10 from Ensembl biomart and build GRanges object
6. Build a GRanges object with Transcription Start Sites (TSS) for all genes in hg19 using Ensembl biomart
7. Create two GRanges object, one with gene coordinates for hg19 and another for TF1 binding sites from
   1. Compute TF1 binding sites overlap with TSS ± 1kb
   2. Create a histogram of TF1 binding site distance to TSS