R 프로그래밍 #11

2019. 05. 28

한국생명공학연구원
김하성

Sequence analysis II

Genbank file parsing
IRanges / GenomicRanges packages
Feature views in genome

The NCBI sequence database

The National Centre for Biotechnology Information (NCBI) (www.ncbi.nlm.nih.gov)

The European Molecular Biology Laboratory (EMBL) Sequence Database (www.ebi.ac.uk/embl)

In Japan, the DNA Data Bank of Japan (DDBJ; www.ddbj.nig.ac.jp).

Dengue fever

Dengue virus: DEN-1, DEN-2, DEN-3, and DEN-4

Accession no.: NC_001477, NC_001474, NC_001475 and NC_002640

U.S. National Library of Medicine
National Center for Biotechnology Information

Log in

Search NCBI databases Help

NC_001477 Search

Results found in 9 databases for "NC_001477"

Dengue virus 1, complete genome
10,735 bp genomic DNA.
Type: 1. Old_name: Dengue virus type 1. Clone: 45AZ5.
Accession: NC_001477.1 GI: 9626685
GenBank FASTA Graphics Gene

Literature		
Books	0	books and reports
MeSH	0	ontology used for PubMed indexing
NLM Catalog	0	books, journals and more in the NLM Collections
PubMed	1	scientific and medical abstracts/citations
PubMed Central	23	full-text journal articles

Genes		
EST	0	expressed sequence tag sequences
Gene	1	collected information about gene loci
GEO DataSets	0	functional genomics studies
GEO Profiles	0	gene expression and molecular abundance profiles
HomoloGene	0	homologous gene sets for selected organisms

Health		
ClinVar	0	human variations of clinical significance
dbGaP	0	genotype/phenotype interaction studies
GTR	0	genetic testing registry
MedGen	0	medical genetics literature and links
OMIM	0	online mendelian inheritance in man
PubMed Health	0	clinical effectiveness, disease and drug reports

Genomes		
Assembly	1	genome assembly information
BioCollections	0	museum, herbaria, and other biorepository collections
BioProject	0	biological projects providing data to NCBI
BioSample	0	descriptions of biological source materials
Clone	0	genomic and cDNA clones
dbVar	0	genome structural variation studies
Genome	1	genome sequencing projects by organism
GSS	0	genome survey sequences
Nucleotide	19	DNA and RNA sequences
Probe	15	sequence-based probes and primers
SNP	0	short genetic variations

PopSet	0	sequence sets from phylogenetic and population studies
UniGene	0	clusters of expressed transcripts

Proteins		
Conserved Domains	0	conserved protein domains
Identical Protein Groups	1	protein sequences grouped by identity
Protein	17	protein sequences
Protein Clusters	0	sequence similarity-based protein clusters
Sparcle	0	functional categorization of proteins by domain architecture
Structure	0	experimentally-determined biomolecular structures

Chemicals		
BioSystems	0	molecular pathways with links to genes, proteins and chemicals
PubChem BioAssay	0	bioactivity screening studies
PubChem Compound	0	chemical information with structures, information and links
PubChem Substance	0	deposited substance and chemical information

reutils

- https://github.com/gschofl/reutils

https://github.com/gschofl/reutils

reutils

build passing | build failing | downloads 600/month | CRAN 0.2.3

`reutils` is an R package for interfacing with NCBI databases such as PubMed, Genbank, or GEO via the Entrez Programming Utilities (EUtils). It provides access to the nine basic *eutils*: `einfo`, `esearch`, `esummary`, `epost`, `efetch`, `elink`, `egquery`, `espell`, and `ecitmatch`.

Please check the relevant usage guidelines when using these services. Note that Entrez server requests are subject to frequency limits. Consider obtaining an NCBI API key if are a heavy user of E-utilities.

Installation

You can install the released version of reutils from CRAN with:

```
install.packages("reutils")
```

Install the development version from `github` using the `devtools` package.

```
require("devtools")
install_github("gschofl/reutils")
```

Please post feature or support requests and bugs at the issues tracker for the reutils package on GitHub.

Important functions

With nine E-Utilities, NCBI provides a programmatical interface to the Entrez query and database system for searching and retrieving requested data

Important functions

With nine E-Utilities, NCBI provides a programmatical interface to the Entrez query and database system for searching and retrieving requested data

Each of these tools corresponds to an `R` function in the reutils package described below.

`esearch`

`esearch` : search and retrieve a list of primary UIDs or the NCBI History Server information (queryKey and webEnv). The objects returned by `esearch` can be passed on directly to `epost` , `esummary` , `elink` , or `efetch` .

`efetch`

`efetch` : retrieve data records from NCBI in a specified retrieval type and retrieval mode as given in this table. Data are returned as XML or text documents.

`esummary`

`esummary` : retrieve Entrez database summaries (DocSums) from a list of primary UIDs (Provided as a character vector or as an `esearch` object)

`elink`

`elink` : retrieve a list of UIDs (and relevancy scores) from a target database that are related to a set of UIDs provided by the user. The objects returned by `elink` can be passed on directly to `epost` , `esummary` , or `efetch` .

`einfo`

`einfo` : provide field names, term counts, last update, and available updates for each database.

`epost`

`epost` : upload primary UIDs to the users's Web Environment on the Entrez history server for subsequent use with `esummary` , `elink` , or `efetch` .

Download NC_001477.1 sequence

Dengue virus: DEN-1, DEN-2, DEN-3, and DEN-4
Accession no.: NC_001477, NC_001474, NC_001475 and NC_002640

```
library(eutils)

acc <- c("NC_001477", "NC_001474", "NC_001475", "NC_002640")
ep <- epost(acc, "nuccore")
ef <- efetch(ep, retmode = "text", rettype = "fasta")
nc <- content(ef)
nc

## write the sequences to a file
write.table(nc, file="den.fasta", quote=F, col.names=F, row.names=F)

## read the sequences
den.seqs <- readDNAStringSet("den.fasta")
```

Download genbank format data

```
library(reutils)

acc <- c("NC_001477", "NC_001474", "NC_001475", "NC_002640")
for(i in 1:length(acc)){
  ef <- efetch(acc[i], "nuccore", retmode = "text", rettype = "gb")
  write(content(ef),file=paste(acc[i], ".gb", sep=""))
  Sys.sleep(1)
  cat(i, "/", length(acc), "\n");flush.console()
}
```

Parsing genbank format data

```
library(genbankr)

acc <- c("NC_001477", "NC_001474", "NC_001475", "NC_002640")
acc_files <- paste(acc, ".gb", sep="")

dg_list <- vector("list", length(acc_files))
dg_list[[1]] <- parseGenBank(file = acc_files[1])
```

Exercise 11-1

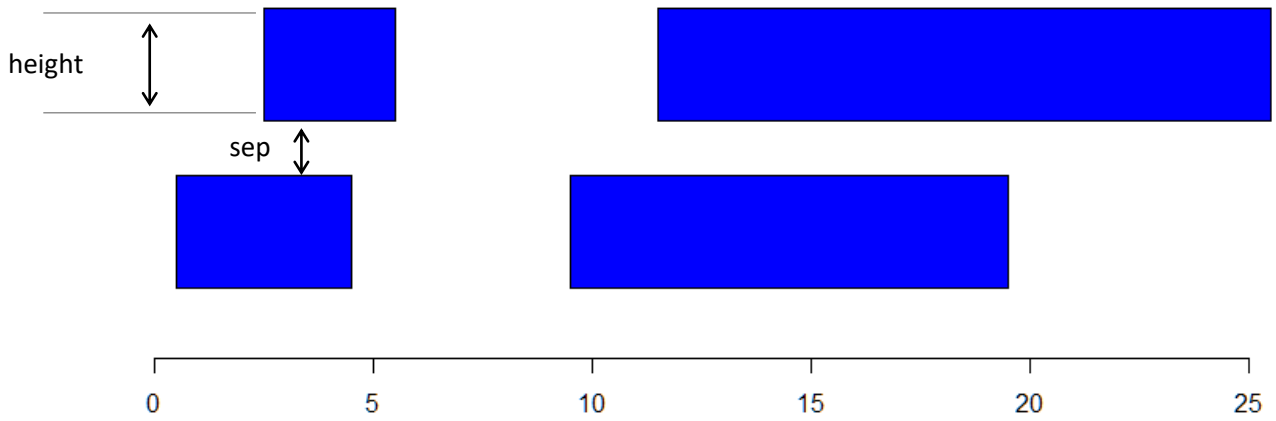
Parsing all files
Use 'for' loop to parse all the files

Genbank

```
LOCUS       NC_001474              10723 bp
DEFINITION  Dengue virus 2, complete genome.
ACCESSION   NC_001474
VERSION     NC_001474.2
DBLINK      BioProject: PRJNA485481
KEYWORDS    RefSeq.
SOURCE      Dengue virus 2
  ORGANISM  Dengue virus 2
            Viruses; Riboviria; Flaviviridae; I
REFERENCE   1  (bases 1 to 10723)
  AUTHORS   Kinney,R.M., Butrapet,S., Chang,G.
            Bhamarapravati,N. and Gubler,D.J.
  TITLE     Construction of infectious cDNA cl
            16681 and its attenuated vaccine d
  JOURNAL   Virology 230 (2), 300-308 (1997)
   PUBMED   9143286
REFERENCE   2  (bases 1 to 10723)
  CONSRTM   NCBI Genome Project
  TITLE     Direct Submission
  JOURNAL   Submitted (01-NOV-2007) National C
            Information, NIH, Bethesda, MD 208
REFERENCE   3  (bases 1 to 10723)
  AUTHORS   Kinney,R.M., Butrapet,S., Chang,G.
            Bhamarapravati,N. and Gubler,D.J.
  TITLE     Direct Submission
  JOURNAL   Submitted (28-JAN-1997) Division o
            Diseases, National Center for Infe
            Disease Control and Prevention, Pu
            Department of Health and Human Ser
            Collins, CO 80522, USA
COMMENT     REVIEWED REFSEQ: This record has b
            reference sequence was derived fro
            On Nov 1, 2007 this sequence versi
            The mature peptides were added by
            annotations for Dengue virus with
            Yamshchikov (Southern Research Ins
            COMPLETENESS: full length.
FEATURES             Location/Qualifiers
     source          1..10723
                     /organism="Dengue virus 2"
                     /mol_type="genomic RNA"
                     /strain="16681"
                     /db_xref="taxon:11060"
                     /country="Thailand"
                     /collection_date="1964"
     5'UTR           1..96
     stem_loop       2..70
                     /note="stem-loop A (SLA)"
     regulatory      71..80
                     /regulatory_class="other"
                     /note="oligo U track spac
     regulatory      81..96
                     /regulatory_class="promot
                     /note="5' upstream AUG re
     stem_loop       81..95
                     /db_xref="VBRC:35925"
     mat_peptide     6376..6756
                     /gene="POLY"
                     /locus_tag="DENV_gp1"
                     /gene_synonym="polyprotein gene"
                     /product="nonstructural protein NS4A"
                     /protein_id="NP_739588.2"
                     /db_xref="VBRC:35926"
     mat_peptide     6757..6825
                     /gene="POLY"
                     /locus_tag="DENV_gp1"
                     /gene_synonym="polyprotein gene"
                     /product="protein 2K"
                     /protein_id="NP_739593.2"
                     /db_xref="VBRC:35927"
     mat_peptide     6826..7569
                     /gene="POLY"
                     /locus_tag="DENV_gp1"
                     /gene_synonym="polyprotein gene"
                     /product="nonstructural protein NS4B"
                     /protein_id="NP_739589.2"
                     /db_xref="VBRC:35928"
     mat_peptide     7570..10269
                     /gene="POLY"
                     /locus_tag="DENV_gp1"
                     /gene_synonym="polyprotein gene"
                     /product="RNA-dependent RNA polymerase NS5"
                     /note="methyltransferase component of capping enzyme;
                     nonstructural protein NS5"
                     /protein_id="NP_739590.2"
                     /db_xref="VBRC:35929"
     stem_loop       116..132
                     /note="capsid region hairpin (cHP)"
     regulatory      134..144
                     /regulatory_class="other"
                     /note="5' conserved sequence (CS); also called cyclization
                     sequence"
     3'UTR           10273..10723
     ncRNA           10299..10723
                     /ncRNA_class="lncRNA"
                     /product="sfRNA1"
                     /note="subgenomic flavivirus RNA"
     stem_loop       10303..10368
                     /note="flaviviral nuclease-resistant RNA 1 (fNR1); also
                     called stem-loop 1 or xrRNA1"
     ncRNA           10372..10723
                     /ncRNA_class="lncRNA"
                     /product="sfRNA2"
                     /note="subgenomic flavivirus RNA"
     stem_loop       10376..10441
                     /note="flaviviral nuclease-resistant RNA 2 (fNR2); also
                     called stem-loop 2 or xrRNA2"
     ncRNA           10449..10723
                     /ncRNA_class="lncRNA"
                     /product="sfRNA3"
```

IRanges

```
library(IRanges)

ir <- IRanges(start=c(1,3,12,10), end=c(4,5,25,19))
ir
length(ir)
start(ir)
end(ir)
width(ir)
range(ir)

## plot
height <- 1
xlim = c(min(start(ir)), max(end(ir)))
bins = disjointBins(ir)

plot.new()
plot.window(xlim, c(0, max(bins) * height))
ybottom = bins * height - height
rect(start(ir), ybottom, end(ir), ybottom + height, col = "blue")
axis(1)
```



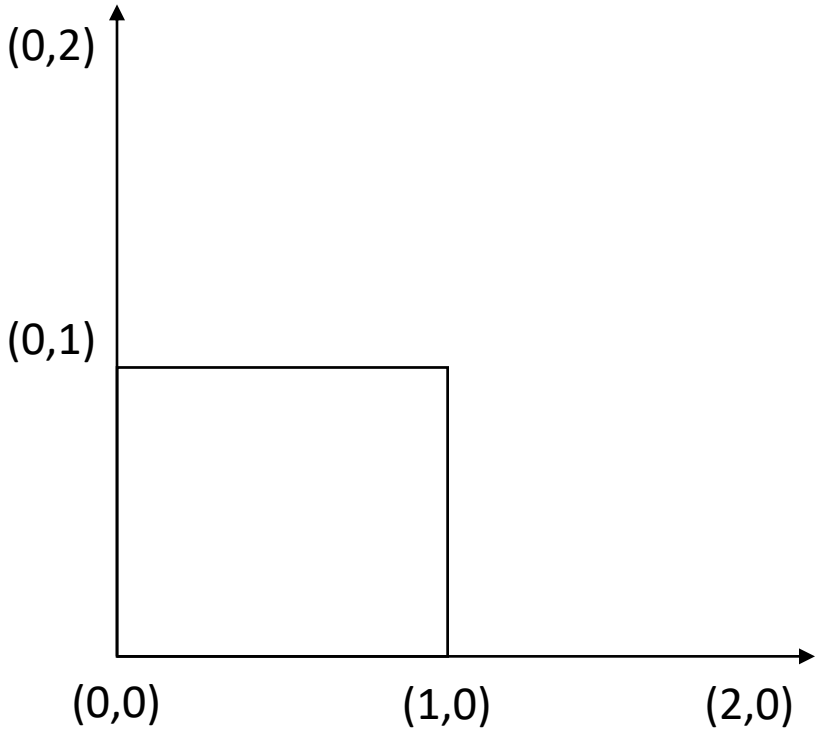
Exercise 11-2

Draw a box

- plot.new()
- plot.window()
- rect()

plot.window(xlim, ylim, log = "", asp = NA, ...)

rect(xleft, ybottom, xright, ytop, density = NULL, angle = 45,…)



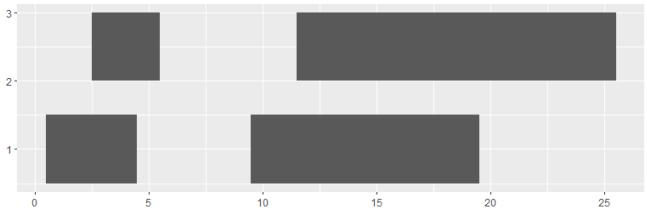
IRanges with ggplot

```
library(ggplot2)

height <- 1
bins <- disjointBins(ir)
ybottom <- bins*height - height
df <- data.frame(ybottom = ybottom,
           xleft = start(ir) - sep,
           xright = end(ir) + sep,
           ytop = ybottom + height)
ggplot(df, aes(xmax = xright, xmin = xleft, ymax = ytop, ymin = ybottom)) +
  geom_rect()
```

geom_rect uses the locations of the four corners (xmin, xmax, ymin and ymax)
geom_tile uses the center of the tile and its size (x, y, width, height)

aes Construct aesthetic mappings

```
> df
  ybottom xleft xright ytop
1     0.5   0.5    4.5  1.5
2     2.0   2.5    5.5  3.0
3     2.0  11.5   25.5  3.0
4     0.5   9.5   19.5  1.5
> aes(xmax = xright, xmin = xleft, ymax = ytop, ymin = ybottom)
* xmax -> xright
* xmin -> xleft
* ymax -> ytop
* ymin -> ybottom
```



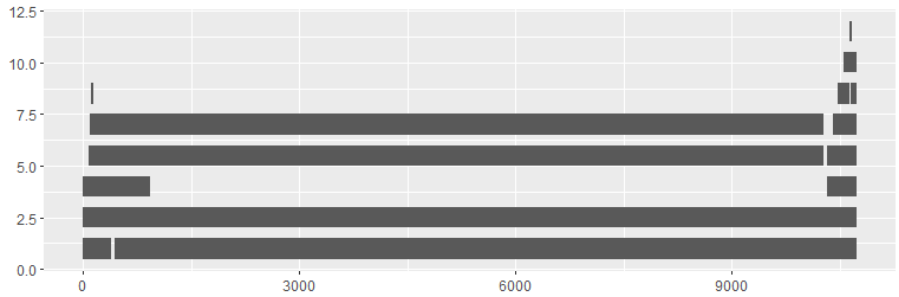
Exercise 11-3

Generate a function named “plotRanges”
Input parameter: ir, height, sep
output: ggplot
run by “plotRanges(ir)”

Feature plot

```
sel_dg <- dg_list[[1]]

sel_dg$FEATURES[[1]]$type
sel_dg$FEATURES[[1]]$start
sel_dg$FEATURES[[1]]$end
sel_dg$FEATURES[[1]]$strand
sel_dg$FEATURES[[1]]$product
```

```
n <- length(sel_dg$FEATURES)
start.pos <- rep(0, n)
end.pos <- rep(0, n)

for(i in 1:n){
  start.pos[i] <- sel_dg$FEATURES[[i]]$start
  end.pos[i] <- sel_dg$FEATURES[[i]]$end
}

ir <- IRanges(start=start.pos, end=end.pos)
plotRanges(ir)
```



Exercise 11-4

Plot only for the feature type of “mat_peptide”
Use “if”

GenomicRanges

Rle: Run length encoding

- A simple data compression method to represent a long sequence
- Instead of saving the whole sequence, it stores the consecutive elements with the same value as a single value and count

```
x <- Rle(values=c("a","b","c"), lengths=c(2,3,4))
x
as.character(x)
?GRanges
```

GRanges-class {GenomicRanges} R Documentation

GRanges objects

Description

The GRanges class is a container for the genomic locations and their associated annotations.

Details

GRanges is a vector of genomic locations and associated annotations. Each element in the vector is comprised of a sequence name, an interval, a strand, and optional metadata columns (e.g. score, GC content, etc.). This information is stored in four components:

`seqnames`

a 'factor' Rle object containing the sequence names.

`ranges`

an IRanges object containing the ranges.

`strand`

a 'factor' Rle object containing the strand information.

GRanges example

Required fields:

- seqnames: Rle object for sequence name, e.g., the chromosome number.
- ranges: IRanges object for locations.

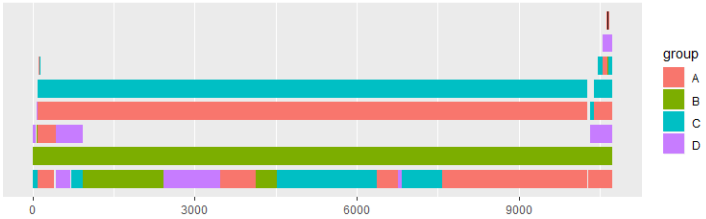
Other fields: strand, elementMetadata for other information.

```
library(GenomicRanges)
gr <- GRanges(seqnames = Rle(c("chr1", "chr2"), c(2,3)),
              ranges = IRanges(start=1:5, end=6:10),
              strand = Rle(strand(c("-", "+", "+", "-")), c(1,1,2,1)),
              score = 1:5,
              GC = seq(1, 0, length=5))
```

```
GRanges object with 5 ranges and 2 metadata columns:
      seqnames    ranges strand |     score        GC
         <Rle> <IRanges>  <Rle> | <integer> <numeric>
  [1]     chr1       1-6      - |         1         1
  [2]     chr1       2-7      + |         2      0.75
  [3]     chr2       3-8      + |         3       0.5
  [4]     chr2       4-9      + |         4      0.25
  [5]     chr2      5-10      - |         5         0
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

GRanges with ggbio and ggplot

```
n <- length(sel_dg$FEATURES)
start.pos <- rep(0, n)
end.pos <- rep(0, n)
dstr <- rep("", n)

for(i in 1:n){
  start.pos[i] <- sel_dg$FEATURES[[i]]$start
  end.pos[i] <- sel_dg$FEATURES[[i]]$end
  dstr[i] <- sel_dg$FEATURES[[i]]$strand
}

snames <- Rle(c("Chr1"), n)
dstrand <- Rle(strand(dstr))
ir <- IRanges(start=start.pos, end=end.pos)

gr <- GRanges(seqnames=snames, ranges=ir, strand=dstrand)

ibrary(ggbio)
ggplot(gr) + geom_rect()


gr <- GRanges(seqnames=snames, ranges=ir, strand=dstrand, group=sample(LETTERS[1:4], length(ir), T))

ggplot(gr) + geom_rect(aes(fill=group))
ggplot(gr) + layout_circle(geom="rect", aes(fill=group))
```



Exercise 11-5

Generate a synthetic genome with 100 features

- Use GRanges
- Each feature starts from random position within 1 to 300
- Length of each features ranging from 50 to 200
- Random strand
- Each feature belongs to one of four groups, A, B, C, D

Plot using geom_rect and layout_circle

Next

- Sequence analysis in R III
- Sequence alignment, clustering, testing