Leggere e scrivere dati in R

Corso di Bioinformatica

Nicola Vitacolonna
Corso di Laurea in Biotecnologie

Preparare una sessione di lavoro

- Creare una cartella (ad esempio, sul Desktop) per contenere i dati, le analisi e i risultati
- In RStudio, impostare lo spazio di lavoro (*workspace*) alla cartella creata al passo precedente (supponiamo di averla chiamata `prima_analisi`), con un comando del tipo:

```
setwd("C:/Documents and Settings/nicola/Desktop/prima_analisi")
```

oppure dal menú Session > Set Working Directory > Choose Directory…

- Verificare che lo spazio di lavoro sia impostato in modo corretto, con il comando:

```
getwd()
```

- Per stampare l'elenco dei file contenuti nella cartella, usare il comando `dir()`

```
dir()
```

Formati di dato che è possibile caricare in R

- File di testo in formato tabulare (in cui i campi sono tipicamente separati da tabulazioni o virgole) (`.csv`, `.tsv`)
- Sessioni di R (tipicamente file con suffisso `.rda`)
- File Excel (mediante package come `xlsx` o `XLConnect`)
- File prodotti da altri software statistici (S, Stata, etc…)
- File HTML, XML, JSON, etc…
- Dati da database

Leggere di file di testo in formato tabulare

- Con la funzione `read.table()`
- Carica i dati in RAM - non adatta per una quantità di dati maggiore della memoria disponibile
- Parametri importanti:
  - `header` (`TRUE/FALSE`): specifica se le colonne della tabella sono etichettate
  - `sep`: carattere usato per separare i campi
  - `nrows`: massimo numero di righe da leggere (nel caso in cui non si voglia leggere l'intero contenuto del file)
- La funzione `read.table()` crea un data frame a partire dai dati letti dal file

Esempio

Tabella con campi separati da tabulazione:

```
Nome    Voto
Anna    28
Fabio   26
Lina    25
```

```
risultatiEsame <- read.table("esami.tsv", header = TRUE)
risultatiEsame
```

```
##    Nome Voto
## 1  Anna   28
## 2 Fabio   26
## 3  Lina   25
```

Esempio

Tabella con campi separati da virgola:

```
Nome,Voto
Anna,28
Fabio,26
Lina,25
```

```
risultatiEsame <- read.table("esami.csv", header = TRUE, sep = ",")
risultatiEsame
```

```
##    Nome Voto
## 1  Anna   28
## 2 Fabio   26
## 3  Lina   25
```

In questo caso, si può usare la piú conveniente funzione `read.csv("esami.csv")`

Esempio

Tabella senza intestazione e `;` come separatore. In questo caso, poiché il file è privo d'intestazione, è opportuno etichettare le colonne dopo aver caricato i dati:

```
Anna;28
Fabio;26
Lina;25
```

```
risultatiEsame <- read.table("esami.txt", sep = ";", header = FALSE)
names(risultatiEsame) <- c("Nome", "Voto")
risultatiEsame
```

```
##    Nome Voto
## 1  Anna   28
## 2 Fabio   26
## 3  Lina   25
```

Stimare la quantità di memoria

Per avere una stima approssimata della quantità di memoria occupata dai dati si può assumere quanto segue:

- un valore numerico occupa 8 byte
- una stringa occupa $8N$ byte, dove N è il numero di caratteri

Ad esempio, una tabella con $1.5 \cdot 10^6$ righe e 120 colonne, contenente solo valori numerici, occupa approssimativamente:

$$1.5 \cdot 10^6 \cdot 120 \cdot 8 = 1.44 \cdot 10^9 \text{ byte} \approx 1.34 \text{ GB}$$

tenuto conto che $1 \text{ GB} = 2^{30} \text{ byte}$

Salvare un data frame

- Con la funzione `write.table()`
- I dati sono salvati nella cartella di lavoro corrente sotto forma di file di testo

```
write.table(risultatiEsame, "copia_tabella.csv", sep = ",")
```

- Per convenienza, R offre anche la funzione `write.csv()`:

```
write.csv(risultatiEsame, "copia_tabella.csv")
```

Salvare e caricare una sessione di lavoro

- È possibile salvare gli oggetti nel *workspace* con le funzioni `save()` e `save.image()`
- `save()` salva soltanto gli oggetti specificati
- `save.image()` salva l'intero contenuto del *workspace*

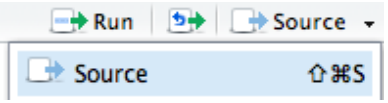
```
risultatiEsame <- read.csv("esami.csv")
nomi <- risultatiEsame$Nome
save(nomi, file = "nomi_studenti.rda")  # Per convenzione, il suffisso è .rda
```

- Gli oggetti cosí salvati possono essere nuovamente caricati in memoria con `load()`

```
load("nomi_studenti.rda")
```

- La funzione `load()` aggiunge gli oggetti caricati dal file a quelli già presenti nel *workspace*, sovrascrivendo quelli con lo stesso nome

Salvare ed eseguire script

- Uno **script** è una sequenza di comandi R contenuta in un file
- Tipicamente, gli script R sono salvati in file con suffisso `.r`
- Per creare uno script in RStudio, scegliere File > New > R Script. Nella finestra che compare, è possibile scrivere una sequenza arbitraria di comandi R. I comandi *non* sono eseguiti immediatamente, come avviene nella console.
- Lo script può essere salvato cliccando sull'icona 
- Lo script può essere eseguito cliccando su 

- La sola linea corrente (quella in cui si trova il cursore) può essere eseguita in modo isolato cliccando su 

Alcune risorse di dati

- NCBI (http://www.ncbi.nlm.nih.gov)
- Protein Data Bank (http://www.rcsb.org/)
- Gapminder (http://www.gapminder.org/data/)
- Istat (http://www.istat.it/)