Prova scritta di Metodi Informatici per la Biologia

DATA: **NOME:** **COGNOME:**

MATRICOLA:

Di seguito sono elencate, per le diverse parti del Corso, alcune domande con 4 possibile relative risposte, delle quali UNA SOLA è vera. Ciascuno studente dovrà scegliere, per ogni domanda, UNA SOLA possibile risposta, barrando il quadrato corrispondente nella colonna a destra.

Parte Generale:	
1) Perchè la RAM si chiama così (Random Access Memory)?:	
• La velocità di accesso alle singole celle varia a caso	☐
• I dati sono inseriti al suo interno in maniera disordinata	☐
• Il tempo necessario per accedere ad una cella è lo stesso, indipendentemente dalla posizione della cella nella sequenza	☐
• Nessuna delle precedenti risposte	☐
2) Un algoritmo è:	
• Una successione di operazioni elementari che possono essere eseguite da un calcolatore	☐
• Una informazione da elaborare rappresentata in un formato che consenta al programma di operare su di essa	☐
• Una descrizione grafica della logica del programma	☐
• Nessuna delle precedenti risposte	☐
3) Definiamo effettiva per un esecutore la soluzione di un problema quando:	
• L'esecutore è in grado di interpretare la descrizione di tale soluzione e associare ad essa le azioni che deve compiere per eseguirla	☐
• L'esecutore è in grado di compiere tali azioni completandone l'esecuzione in un tempo infinito	☐
• Due soggetti giudicano come effettiva la stessa soluzione di un problema ma poi compiono azioni che producono risultati differenti	☐
• Nessuna delle precedenti risposte	☐
4) La CPU:	
• è composta dai soli registri	☐
• è composta da una Unità di controllo, una Unità aritmetico logica e dai registri	☐
• è un componente unico non composto da sottoparti	☐
• comprende le periferiche	☐
5) Il processo di sviluppo di un programma è organizzato in:	
• Formalizzazione -> Programmazione -> Analisi -> Traduzione	☐
• Programmazione -> Analisi -> Formalizzazione -> Traduzione	☐
• Traduzione - > Analisi -> Formalizzazione -> Programmazione	☐
• Analisi -> Traduzione -> Formalizzazione -> Programmazione	☐
6) Perché una descrizione formale di una soluzione sia un algoritmo devono essere soddisfatti i /il seguenti/e requisiti/o:	
• Finitezza	☐
• Generalità	☐
• Non ambiguità	☐
• Tutti e tre i precedenti	☐

R Reference Card 2.0

V 2 by Matt Baggott, matt@baggott.net
V 1 by Tom Short, t.short@ieee.org
Material from *R for Beginners* by permission of Emmanuel Paradis.

Getting help and info

help(topic) documentation on topic
?topic same as above; special chars need quotes: for example **?'&&'**
help.search("topic") search the help system; same as **??topic**
apropos("topic") the names of all objects in the search list matching the regular expression "topic"
help.start() start the HTML version of help
summary(x) generic function to give a "summary" of x, often a statistical one
str(x) display the internal structure of an R object
ls() show objects in the search path; specify pat="pat" to search on a pattern
ls.str() str for each variable in the search path
dir() show files in the current directory
methods(x) shows S3 methods of x
methods(class=class(x)) lists all the methods to handle objects of class x
findFn() searches a database of help packages for functions and returns a data.frame (*sos*)

Other R References

CRAN task views are summaries of R resources for task domains at: cran.r-project.org/web/views Can be accessed via *ctv* package
R FAQ: cran.r-project.org/doc/FAQ/R-FAQ.html
R Functions for Regression Analysis, by Vito Ricci: cran.r-project.org/doc/contrib/Ricci-refcard-regression.pdf
R Functions for Time Series Analysis, by Vito Ricci: cran.r-project.org/doc/contrib/Ricci-refcard-ts.pdf
R Reference Card for Data Mining, by Yanchang Zhao: www.rdatamining.com/docs/R-refcard-data-mining.pdf
R Reference Card, by Jonathan Baron: cran.r-project.org/doc/contrib/refcard.pdf

Operators

<-	Left assignment, binary
->	Right assignment, binary
=	Left assignment, but not recommended
<<-	Left assignment in outer lexical scope; not for beginners
$	List subset, binary
-	Minus, can be unary or binary
+	Plus, can be unary or binary
~	Tilde, used for model formulae
:	Sequence, binary (in model formulae: interaction)
::	Refer to function in a package, i.e, pkg::function; usually not needed
*	Multiplication, binary
/	Division, binary
^	Exponentiation, binary
%x%	Special binary operators, x can be replaced by any valid name
%%	Modulus, binary
%/%	Integer divide, binary
%*%	Matrix product, binary
%o%	Outer product, binary
%x%	Kronecker product, binary
%in%	Matching operator, binary (in model formulae: nesting)
! x	logical negation, NOT x
x & y	elementwise logical AND
x && y	vector logical AND
x | y	elementwise logical OR
x || y	vector logical OR
xor(x, y)	elementwise exclusive OR
<	Less than, binary
>	Greater than, binary
==	Equal to, binary
>=	Greater than or equal to, binary
<=	Less than or equal to, binary

Packages

install.packages("pkgs", lib) download and install pkgs from repository (lib) or other external source
update.packages checks for new versions and offers to install
library(pkg) loads pkg, if pkg is omitted it lists packages
detach("package:pkg") removes pkg from memory

Indexing vectors

x[n]	nth element
x[-n]	all but the nth element
x[1:n]	first n elements
x[-(1:n)]	elements from n+1 to end
x[c(1,4,2)]	specific elements
x["name"]	element named "name"
x[x > 3]	all elements greater than 3
x[x > 3 & x < 5]	all elements between 3 and 5
x[x %in% c("a","if")]	elements in the given set

Indexing lists

x[n]	list with elements n
x[[n]]	nth element of the list
x[["name"]]	element named "name"
x$name	as above (w. partial matching)

Indexing matrices

x[i,j]	element at row i, column j
x[i,]	row i
x[,j]	column j
x[,c(1,3)]	columns 1 and 3
x["name",]	row named "name"

Indexing matrices data frames (same as matrices plus the following)

X[["name"]]	column named "name"
x$name	as above (w. partial matching)

Input and output (I/O)

R data object I/O

data(x) loads specified data set; if no arg is given it lists all available data sets
save(file,...) saves the specified objects (...) in XDR platform-independent binary format
save.image(file) saves all objects
load(file) load datasets written with save

Database I/O

Useful packages: *DBI* interface between R and relational DBMS; *RJDBC* access to databases through the JDBC interface; *RMySQL* interface to MySQL database; *RODBC* ODBC database access; *ROracle* Oracle database interface driver; *RpgSQL* interface to PostgreSQL database; *RSQLite* SQLite interface for R

7) I Linguaggi Assembly:	
• Sono versioni binarie dei linguaggi hardware	☐
• Sono versioni simboliche dei linguaggi hardware	☐
• Sono linguaggi hardware	☐
• Nessuna delle precedenti risposte	☐
8) Il Firmware è:	
• Un software integrato direttamente in un componente elettronico	☐
• Il marchio del produttore del personal computer	☐
• Un programma modificabile dall'utente	☐
• Un programma che gestisce la memoria	☐
9) L'interprete dei dati è:	
• Un software di traduzione linguistica	☐
• Un software che converte tra loro vari linguaggi di programmazione	☐
• Un software che fa da tramite tra le periferiche di input e il processore	☐
• Un componente hardware	☐
10) La tecnica di programmazione:	
• E' legata ad uno specifico linguaggio	☐
• Non è legata ad un linguaggio e si può abbinare a qualsiasi linguaggio	☐
• Non è legata ad un linguaggio, anche se alcuni linguaggi sono ideali per essere abbinati ad una specifica tecnica di programmazione	☐
• Nessuna delle precedenti risposte	☐
11) Il Dithering è:	
• Un software per la dettatura	☐
• Una tecnica per creare l'illusione della profondità di colore in immagini dotate di una tavolozza limitata	☐
• Una tecnica per comprimere un'immagine senza (apparentemente) diminuirne la qualità	☐
• Nessuna delle precedenti risposte	☐
12) Nell'algebra di Boole, la seguente definizione: (3 E' UN NUMERO PARI) & (3 E' MINORE DI 2) è:	
• VERA	☐
• FALSA	☐
• Indeterminata	☐
• assurda	☐
13) Nell'algebra di Boole, la seguente definizione: (3 E' UN NUMERO DISPARI) | (5 E' MINORE DI 4) è:	
• VERA	☐
• FALSA	☐
• Indeterminata	☐
• assurda	☐
14) Nell'algebra di Boole, l'UNIONE dell'Insieme N (Numeri interi) e dell'Insieme R (Numeri Reali) è:	
• Coincidente con l'Insieme N	☐
• Insieme vuoto	☐
• sottoinsieme R dei numeri Reali positivi	☐
• Coincide con l'insieme R	☐
15) Nell'ambito delle tecniche GIS, un BUFFER è:	
• Un'operazione che definisce una distanza costante rispetto all'entità di riferimento	☐
• Un'operazione che definisce una linea creata sulla base di una distanza generalmente, ma non necessariamente, costante rispetto all'entità di riferimento	☐
• Un'operazione che definisce un'entità areale creata sulla base di una distanza generalmente, ma non necessariamente, costante rispetto all'entità di riferimento	☐
• Nessuna delle precedenti risposte	☐

16) In ambiente R, la seguente funzione:

Csum <- function(vseries){

Other file I/O

read.table(file), read.csv(file), read.delim("file"), read.fwf("file") read a file using defaults sensible for a table/csv/delimited/fixed-width file and create a data frame from it.

write.table(x,file), write.csv(x,file) saves x after converting to a data frame

txtStart and **txtStop**: saves a transcript of commands and/or output to a text file (*TeachingDemos*)

download.file(url) from internet

url.show(url) remote input

cat(..., file="", sep=" ") prints the arguments after coercing to character; sep is the character separator between arguments

print(x, ...) prints its arguments; generic, meaning it can have different methods for different objects

format(x,...) format an R object for pretty printing

sink(file) output to file, until sink()

Clipboard I/O

File connections of functions can also be used to read and write to the clipboard instead of a file.

Mac OS: **x <- read.delim(pipe("pbpaste"))**

Windows: **x <- read.delim("clipboard")**

See also **read.clipboard** (*psych*)

Data creation

c(...) generic function to combine arguments with the default forming a vector; with recursive=TRUE descends through lists combining all elements into one vector

from:to generates a sequence; ":" has operator priority; 1:4 + 1 is "2,3,4,5"

seq(from,to) generates a sequence by= specifies increment; length= specifies desired length

seq(along=x) generates 1, 2, ..., length(along); useful in for loops

rep(x,times) replicate x times; use each to repeat "each" element of x each times; rep(c(1,2,3),2) is 1 2 3 1 2 3; **rep(c(1,2,3),each=2)** is 1 1 2 2 3 3

data.frame(...) create a data frame of the named or unnamed arguments data.frame (v=1:4, ch= c("a","B","c","d"), n=10); shorter vectors are recycled to the length of the longest

list(...) create a list of the named or unnamed arguments; list(a=c(1,2),b="hi", c=3);

array(x,dim=) array with data x; specify dimensions like dim=c(3,4,2); elements of x recycle if x is not long enough

matrix(x,nrow,ncol) matrix; elements of x recycle factor(x,levels) encodes a vector x as a factor

gl(n, k, length=n*k, labels=1:n) generate levels (factors) by specifying the pattern of their levels; k is the number of levels, and n is the number of replications

expand.grid() a data frame from all combinations of the supplied vectors or factors

Data conversion

as.array(x), as.character(x), as.data.frame(x), as.factor(x), as.logical(x), as.numeric(x), convert type; for a complete list, use **methods(as)**

Data information

is.na(x), is.null(x), is.nan(x); is.array(x), is.data.frame(x), is.numeric(x), is.complex(x), is.character(x); for a complete list, use **methods(is)**

x prints x

head(x), tail(x) returns first or last parts of an object

summary(x) generic function to give a summary

str(x) display internal structure of the data

length(x) number of elements in x

dim(x) Retrieve or set the dimension of an object; **dim(x) <- c(3,2)**

dimnames(x) Retrieve or set the dimension names of an object

nrow(x), ncol(x) number of rows/cols; **NROW(x), NCOL(x)** is the same but treats a vector as a one-row/col matrix

class(x) get or set the class of x; **class(x) <- "myclass";**

unclass(x) removes the class attribute of x

attr(x,which) get or set the attribute which of x

attributes(obj) get or set the list of attributes of obj

Data selection and manipulation

which.max(x), which.min(x) returns the index of the greatest/smallest element of x

rev(x) reverses the elements of x

sort(x) sorts the elements of x in increasing order; to sort in decreasing order: **rev(sort(x))**

cut(x,breaks) divides x into intervals (factors); breaks is the number of cut intervals or a vector of cut points

match(x, y) returns a vector of the same length as x with the elements of x that are in y (NA otherwise)

which(x == a) returns a vector of the indices of x if the comparison operation is true (TRUE), in this example the values of i for which x[i] == a (the argument of this function must be a variable of mode logical)

choose(n, k) computes the combinations of k events among n repetitions = n!/[(n − k)!k!]

na.omit(x) suppresses the observations with missing data (NA)

na.fail(x) returns an error message if x contains at least one NA

complete.cases(x) returns only observations (rows) with no NA

unique(x) if x is a vector or a data frame, returns a similar object but with the duplicates suppressed

table(x) returns a table with the numbers of the different values of x (typically for integers or factors)

split(x, f) divides vector x into the groups based on f

subset(x, ...) returns a selection of x with respect to criteria (..., typically comparisons: x$V1 < 10); if x is a data frame, the option select gives variables to be kept (or dropped, using a minus)

sample(x, size) resample randomly and without replacement size elements in the vector x, for sample with replacement use: replace = TRUE

sweep(x, margin, stats) transforms an array by sweeping out a summary statistic

prop.table(x,margin) table entries as fraction of marginal table

xtabs(a b,data=x) a contingency table from cross-classifying factors

replace(x, list, values) replace elements of x listed in index with values

```
csum=0
for(i in 1:length(vseries)) csum=csum+vseries[i]
return(csum)
}
```

serve a:

- Calcolare la lunghezza di un vettore numerico ☐
- Calcolare la somma di tutti gli elementi di un vettore numerico ☐
- Calcolare la somma di alcuni elementi (specificati dall'utente) di un vettore numerico ☐
- Nessuna delle precedenti ☐

17) Nell'ambito delle tecniche GIS, una QUERY è:

- l'uso di operatori elementari in sequenza allo scopo di risolvere problemi spaziali complessi ☐
- la sovrapposizione di due o più strati informativi (layer) ☐
- L' operazione di estrazione di informazioni sia spaziali che alfanumeriche da un GIS ☐
- Un attributo dei dati ☐

18) In cartografia, la Proiezione è:

- Il metodo per sovrapporre due o più strati informativi (layer) ☐
- Il sistema di riferimento dei dati ☐
- Una tecnica di compressione dei dati ☐
- Il procedimento mediante il quale si passa dalla superficie sferica della Terra ad una superficie bidimensionale ☐

19) Un automa cellulare è:

- è un modello matematico usato per descrivere l'evoluzione di sistemi complessi discreti, studiati in teoria della computazione, matematica, fisica e biologia ☐
- Un individuo virtuale di un modello ☐
- La rappresentazione matematica di una cellula vivente ☐
- Un software di analisi dei dati ☐

20) Quale tra i seguenti word processor appartiene alla filosofia WYSIWYM (What You See Is What You Mean):

- Microsoft Word ☐
- Notepad ☐
- Adobe Acrobat ☐
- Latex ☐

Data reshaping

merge(a,b) merge two data frames by common col or row names
stack(x, ...) transform data available as separate cols in a data frame or list into a single col
unstack(x, ...) inverse of stack()
rbind(...) , cbind(...) combines supplied matrices, data frames, etc. by rows or cols
melt(data, id.vars, measure.vars) changes an object into a suitable form for easy casting, (*reshape2* package)
cast(data, formula, fun) applies fun to melted data using formula (*reshape2* package)
recast(data, formula) melts and casts in a single step (*reshape2* package)
reshape(x, direction...) reshapes data frame between 'wide' (repeated measurements in separate cols) and 'long' (repeated measurements in separate rows) format based on direction

Applying functions repeatedly

(m=matrix, a=array, l=list; v=vector, d=dataframe)
apply(x,index,fun) input: m**;** output: a or l**;** applies function fun to rows/cols/cells (index) of x
lapply(x,fun) input l**;** output l**;** apply fun to each element of list x
sapply(x,fun) input l**;** output v**;** user friendly wrapper for **lapply()**; see also **replicate()**
tapply(x,index,fun) input l output l; applies fun to subsets of x, as grouped based on index
by(data,index,fun) input df; output is class "by", wrapper for tapply
aggregate(x,by,fun) input df; output df; applies fun to subsets of x, as grouped based on index. Can use formula notation.
ave(data, by, fun = mean) gets mean (or other fun) of subsets of x based on list(s) by

plyr **package functions** have a consistent names: The first character is input data type, second is output. These may be d(ataframe), l(ist), a(rray), or _(discard). Functions have two or three main arguments, depending on input:

a*ply(.data, .margins, .fun, ...)
d*ply(.data, .variables, .fun, ...)
l*ply(.data, .fun, ...)

Three commonly used functions with ply functions are **summarise()**, **mutate(), and transform()**

Math

Many math functions have a logical parameter na.rm=FALSE to specify missing data removal.

sin,cos,tan,asin,acos,atan,atan2,log,log10,exp
min(x), max(x) min/max of elements of x
range(x) min and max elements of x
sum(x) sum of elements of x
diff(x) lagged and iterated differences of vector x
prod(x) product of the elements of x
round(x, n) rounds the elements of x to n decimals
log(x, base) computes the logarithm of x
scale(x) centers and reduces the data; can center only (scale=FALSE) or reduce only (center=FALSE)
pmin(x,y,...), pmax(x,y,...) parallel minimum/maximum, returns a vector in which ith element is the min/max of x[i], y[i], . . .
cumsum(x), cummin(x), cummax(x), cumprod(x) a vector which ith element is the sum/min/max from x[1] to x[i]
union(x,y), intersect(x,y), setdiff(x,y), setequal(x,y), is.element(el,set) "set" functions
Re(x) real part of a complex number
Im(x) imaginary part
Mod(x) modulus; **abs(x)** is the same
Arg(x) angle in radians of the complex number
Conj(x) complex conjugate
convolve(x,y) compute convolutions of sequences
fft(x) Fast Fourier Transform of an array
mvfft(x) FFT of each column of a matrix
filter(x,filter) applies linear filtering to a univariate time series or to each series separately of a multivariate time series

Correlation and variance

cor(x) correlation matrix of x if it is a matrix or a data frame (1 if x is a vector)
cor(x, y) linear correlation (or correlation matrix) between x and y
var(x) or **cov(x)** variance of the elements of x (calculated on n − 1); if x is a matrix or a data frame, the variance-covariance matrix is calculated
var(x, y) or **cov(x, y)** covariance between x and y, or between the columns of x and those of y if they are matrices or data frames

Matrices

t(x) transpose
diag(x) diagonal
%*% matrix multiplication
solve(a,b) solves a %*% x = b for x solve(a) matrix inverse of a
rowsum(x), colsum(x) sum of rows/cols for a matrix-like object (consider **rowMeans(x)**, **colMeans(x)**)

Distributions

Family of distribution functions, depending on first letter either provide: r(andom sample) ; p(robability density), c(umulative probability density),or q(uantile):

rnorm(n, mean=0, sd=1) Gaussian (normal)
rexp(n, rate=1) exponential
rgamma(n, shape, scale=1) gamma
rpois(n, lambda) Poisson
rweibull(n, shape, scale=1) Weibull
rcauchy(n, location=0, scale=1) Cauchy
rbeta(n, shape1, shape2) beta
rt(n, df) 'Student' (t)
rf(n, df1, df2) Fisher-Snedecor (*F*) **(!!!²)**
rchisq(n, df) Pearson
rbinom(n, size, prob) binomial
rgeom(n, prob) geometric
rhyper(nn, m, n, k) hypergeometric
rlogis(n, location=0, scale=1) logistic
rlnorm(n, meanlog=0, sdlog=1) lognormal
rnbinom(n, size, prob) negative binomial
runif(n, min=0, max=1) uniform
rwilcox(nn, m, n), rsignrank(nn, n) Wilcoxon

Descriptive statistics

mean(x) mean of the elements of x
median(x) median of the elements of x
quantile(x,probs=) sample quantiles corresponding to the given probabilities (defaults to 0,.25,.5,.75,1)
weighted.mean(x, w) mean of x with weights w
rank(x) ranks of the elements of x
describe(x) statistical description of data (in *Hmisc* package)
describe(x) statistical description of data useful for psychometrics (in *psych* package)
sd(x) standard deviation of x
density(x) kernel density estimates of x

Ambiente R: Problema

Descrivere all'interno del box sottostante, il procedimento in R per:

1) Generare una matrice di 10 righe e 5 colonne, contenente numeri a caso tra 1 e 100;
2) Calcolare la somma di ciascuna riga della matrice, in modo da ottenere un vettore di tali somme;
3) Trovare il valore massimo di tale vettore.

E' consigliato l'uso del prontuario dei comandi allegati. Qualsiasi procedimento descritto, purchè porti al risultato corretto, sarà considerato soddisfacente per ottenere una buona valitazione. Sono ammessi commenti ai comandi elencati.