QUICK REFERENCE

matR (0.9.9): metagenomics analysis tools for R

mg-rast@mcs.anl.gov

installation and help

`dependencies(prompt=)`	obtain supporting packages (with or without interactive prompting)
`msession$setAuth(file=)`	enter key for private data (use `file` to read from disk)
`demo(package="matR"); demo2("demo")`	list demos; or step through demo named `"demo"`
`data(package="matR")`	list metagenome collections prepackaged
`?command; library(help="matR"); msession$debug()`	get help and examples for `command`; see index of help topics; create debug report

metagenome collections

`cc <- collection(x, ..., file=)`	create collection with sample IDs as in `x` or `file`, and data views specified by `...`
`samples(cc); projects(cc)`	see samples in a collection, or projects appearing (fully or partially) in a collection
`views(cc)`	see data views included in a collection
`names(cc); names(cc) <- x`	see sample names; or name or rename samples
`groups(cc); groups(cc) <- x`	see sample grouping; or group samples of a collection
`cc[sub]; as(x, "collection")`	subset collection, retaining samples `sub`; formally convert matrix `x` to collection

data views in a collection

`view.parameters; view.descriptions; view.defaults`	see parameters that define data views and meaning of valid values; or see default views
`cc$raw`	extract data view named `raw` as an ordinary matrix
`cc[["raw", full=, plain=]]`	extract data view named `raw` with options
`views(cc); viewnames(cc) <- x`	see or rename data views of a collection
`x <- c(annot=, entry=, level=, source=, hit=)`	describe a data view by selecting values for view parameters
`cc$new <- x`	add data view named `new`, as described by `x`, to an existing collection

metadata

`metadata(cc)`	all metadata of a collection
`metadata(cc)["str"]`	metadata fields matching specifier `"str"`
`metadata(cc)["str1", "str2", ..., bygroup=]`	fields matching any of given specifiers; `NA` inserted as needed when `bygroup=TRUE`
`mm <- metadata(x, file=)`	retrieve (only) metadata of samples specified by ID by `x` or `file`

analysis utilities (`x` is `matrix`)

`remove.singletons(x, lim.entry=, lim.row=)`	replace entries with zero and/or remove rows below a threshold
`normalize(x, method=)`	apply logarithmic transformation, scale, and center to [0,1]
`dist(x, y=, groups=, method=)`	compute distances among columns of `x`; mean-pairwise among `groups`; or from `y` to each
`sigtest(x, groups, test=, fdr.level=, qvalue=)`	apply significance `test` across rows to `groups` of columns of `x`
`randomize(x, n=, method=, seed=, FUN=)`	apply `FUN` to each of `n` permutations of `x` according to `method`

analysis routines (`cc` is `collection`)

`boxplot(cc, ..., view=)`	summarize distribution of annotations per sample; graphical parameters as in *boxplot*
`pco(cc, comp=, method=, ..., view=, rows=)`	compute and plot principal coordinates; graphical parameters as in *plot* and *points*
`heatmap(cc, ..., view=, rows=)`	compute and view heatmap dendrogram; graphical parameters as in *gplots::heatmap.2*
`parcoord(cc, groups=, test=, p.lim=, n.lim=, ..., view=, rows=)`	compute and plot `n.lim` parallel coordinates (or all with p-value less than `p.lim`)