How To Find Genes Close to Specified Genes

October 22, 2008

Introduction

In some cases you have certain genes of interest and you would like to find other genes that are *close* to the genes of interest. This can be done using the `genefinder` function.

You need to specify either the index position of the genes you want (which row of the expression array the gene is in) or the name (consistent with the `featureNames` of the ExpressionSet).

A vector of names can be specified and matches for all will be computed. The number of matches and the distance measure used can all be specified. The examples will be carried out using the artificial data set, `sample.ExpressionSet`.

Two other options for `genefinder` are `scale` and `method`. The `scale` option controls the scaling of the rows (this is often desirable) while the `method` option controls the distance measure used between genes. The possible values and their meanings are listed at the end of this document.

```
> library("genefilter")
> data(sample.ExpressionSet)
> igenes <- c(300, 333, 355, 419)
> closeg <- genefinder(sample.ExpressionSet, igenes, 10, method = "euc",
+     scale = "none")
> names(closeg)

[1] "31539_r_at" "31572_at"   "31594_at"   "31658_at"
```

The Affymetrix identifiers (since these were originally Affymetrix data) are `31539_r_at`, `31572_at`, `31594_at` and `31658_at`. We can find the nearest genes (by index) for any of these by simply accessing the relevant component of `closeg`.

```
> closeg$"31539_r_at"

$indices
 [1] 220 425 457 131 372 137 380 231 161  38

$dists
 [1] 70.30643 70.94069 71.66043 71.66962 73.55186 73.66967 74.77823 77.42745
 [9] 77.86960 83.57073

> Nms1 <- featureNames(sample.ExpressionSet)[closeg$"31539_r_at"$indices]
> Nms1

 [1] "31459_i_at"      "31664_at"        "31696_at"        "31370_at"
 [5] "31611_s_at"      "31376_at"        "31619_at"        "31470_at"
 [9] "31400_at"        "AFFX-TrpnX-3_at"
```

You could then take these names (from `Nms1`) and the *annotate* package and explore them further. See the various HOWTO's in annotate to see how to further explore your data. Examples include finding and searching all PubMed abstracts associated with these data. Finding and downloading associated sequence information. The data can also be visualized using the *geneplotter* package (again there are a number of HOWTO documents there).

Parameter Settings

The `scale` parameter can take the following values:

none No scaling is done.

range Scaling is done by $(x_i - x_{(1)})/(x_{(n)} - x_{(1)})$.

zscore Scaling is done by $(x_i - \bar{x})/s_x$. Where s_x is the standard deviation.

The `method` parameter can take the following values:

euclidean Euclidean distance is used.

maximum Maximum distance between any two elements of x and y (supremum norm).

manhattan Absolute distance between the two vectors (1 norm).

canberra The $\sum(|x_i - y_i|/|x_i + y_i|)$. Terms with zero numerator and denominator are omitted from the sum and treated as if the values were missing.

binary (aka asymmetric binary): The vectors are regarded as binary bits, so non-zero elements are *on* and zero elements are *off*. The distance is the proportion of bits in which only one is on amongst those in which at least one is on.

1 Session Information

The version number of R and packages loaded for generating the vignette were:

- R version 2.8.0 (2008-10-20), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US;LC_NUMERIC=C;LC_TIME=en_US;LC_COLLATE=en_US;LC_MONETARY=C;LC_MESSAGES=en_US;`
- Base packages: base, datasets, graphics, grDevices, methods, splines, stats, tools, utils
- Other packages: Biobase 2.2.0, class 7.2-44, genefilter 1.22.0, survival 2.34-1
- Loaded via a namespace (and not attached): annotate 1.20.0, AnnotationDbi 1.4.0, DBI 0.2-4, RSQLite 0.7-0