LC_TBOX - Local Correlation Toolbox

The local correlation toolbox is a set of matlab's functions and variables built to compute local correlation coefficients for:

- a set of objects characterized by two distance matrices;
- a set of genes characterized by expression profiles and gene ontology annotations.

The first option is the central application of this toolbox, and is accessible through the `lc_calc.m` function. It requires the user to previously prepare two distance matrices of the same size, containing all the pair-wise distances between a given set of objects. It offers three different local correlation coefficients:

- pearson coefficient;
- ratio coefficient;
- score coefficient.

The second option is an example of the possible use of local correlation coefficients. In this case local correlation is measured between gene expression distances and gene annotation distances. The function `lc_expgo.m` carries out this computation if the user has a list of gene identifiers (from yeast, mouse or arabidopsis genomes) and a corresponding gene expression data matrix (where rows should correspond to genes and columns to experimental conditions).

We now describe the inputs and outputs of these two functions.

- **Function `lc_calc.m`**

To use this function the user should type the next expression in the matlab's command line (when the current working directory is …`\lc_tbox\`):

```
[lc,p,tc]=lc_calc(d1,d2,lcfun)
```

`d1` and `d2` should be two symmetric distance square matrices of the same size (n x n) (containing distances in two different data spaces between n objects) and `lcfun` must be `'ratio'`, `'pearson'` or `'score'`, selecting the corresponding local correlation coefficient to be used. The function returns: `lc`, a column vector of size n, containing the local correlation coefficients for each of the n objects; `p`, a column vector of size n, containing the p values associated with the local correlation coefficients in `lc`; and `tc` a total correlation coefficient between distance values in `d1` and `d2` using the same coefficient formula.

- **Function `lc_expgo.m`**

To use this function the user should type the next expression in the matlab's command line (when the current working directory is …`\lc_tbox\`):

```
[lc,p,tc]=lc_expgo(genelist,genome,expdata,disttype,lcfun,
annot)
```

`genelist` should be a cell array of strings containing identifiers of genes (affymetrix chip references, gene common names or according to the genome database nomenclature), `genome` can be `1`, `2`, `3` or `4`, selecting the genome to which `genelist` belongs (yeast, mouse, arabidopsis or other), `expdata` should be a matrix with n rows, each corresponding to one of the n genes in `genelist`, containing expression values in different experimental conditions (columns), `disttype` can be `‘euc’` or `‘corr’`, defining the way to calculate distances between gene expression profiles, either through Euclidean or correlation distances, `lcfun` must be `‘ratio’`, `‘pearson’` or `‘score’`, selecting the corresponding local correlation coefficient to be used. This function also returns three variables: `lc`, `p` and `tc`, but now each of these variables is structured, containing three fields: `.f`, `.p` and `.c`, corresponding to function, process and cellular component GO categories. Inside each of these fields, the variables have the same dimensions and the same meanings as they had for the `lc_calc.m` function.

But the `lc_expgo` contains an extra input variable that was not explained, the `annot` variable. This input variable is optional if `genome` is `1`, `2` or `3`, but is necessary if `genome` is `4`. It should contain the GO codes for the annotations of the genes in `genelist`. `Annot` should be structured with three fields: `.f`, `.p` and `.c`, each correspondind to the three GO categories. Inside each field there should be a two column matrix. Numbers in the first column are the gene index in `genelist` (so they can only take values between 1 and n, the number of genes in `genelist`) and the numbers in the second column should be GO codes. This way, one given gene can be annotated with more than one code, and not every gene in `genelist` needs to have annotations.

- **Test data**

The `lc_tbox` directory contains three `.mat` files with test data. `ylist.mat` contains a list of 195 yeast gene references, `ydata.mat` contains the corresponding gene expression data matrix. `Yannot.mat` is an `annot` type variable, containing the GO codes that are associated with the genes in `ylist.mat`.