

Metaprior documentation:

The basic call to the Lirnet algorithm looks like:

```
params_out = lirnet (ydata,assign, maxiters, nreguls, params_in)
```

Inputs:

1. ydata: struct

- e : expression data struct
 - unrm - # genes x # arrays - unnormalized expression data
 - nrm - normalized expression data (each gene will have 0 mean, unit variance)
 - locs - location of gene on chromosome
 - reguls - # genes x 1 - indicator of whether each gene is a regulator
- g : genotype struct
 - unrm - # markers x # arrays, binary
 - nrm - normalized
 - locs - location on chromosome
- mf : metafeatures of genotype markers, struct
 - sm - # markers x # metafeatures - for every marker, sum of metafeature values over all snps corresponding to that marker
 - average: - # markers x # metafeatures - for every marker, average of metafeature values over all snps corresponding to that marker
 - nrm: cell of length # markers.
 - For each entry, metafeature values in matrix #snps-in-marker x # metafeatures
 - unrm: same as above but unnormalized
 - mindex - cell array of length # markers, each cell contains an array
 - entry 1 - marker index
 - entry 2 - chromosome
 - entry 3 - indices of SNPs corresponding to that marker
- mf_pair : pairwise (marker-module pairs) metafeatures of genotype markers, cell array of length = # metafeatures.
 - each cell contains nrm (normalized), and unrm (un-normalized) values for one feature, so it is cell array of length # markers, each containing
 - matrix #snps-in-marker x # modules
- rf : metafeatures for regulators
 - nrm - normalized metafeatures - #genes x #metafeatures
 - unrm - unnormalized metafeature values - # genes x # metafeatures
- rf_pair : pairwise metafeatures for regulators, cell array of length = # metafeatures.
 - each cell contains nrm (normalized), and unrm (un-normalized) values for one feature, so it is #genes x # modules
- *** note that for the above structures, only the "nrm" version of each piece of data is used by the code, even when an un-normalized version of the data exists

2. assign: array, #genes x 1, telling which module each gene is assigned to

3. maxiters : integer, the maximum number of iterations of the metaprior loop

4. nreguls: array, the indices of genes which are **not** regulators

5. params: optional input. see output definition, has same structure as returned

params_out

Outputs:

params_out: struct

- mw - learned regression weight matrix for genotype markers (#modules x #genotype markers)
- rw - learned regression weight matrix for regulators (#modules x #genes)
- mw0 - genotype marker weight matrix used in the first iteration
- rw0 - regulator weight matrix used in the first iteration
- mb - learned metaprior for genotype metafeatures (#metafeatures x 1)
- rb - learned metaprior for regulator metafeatures (#metafeatures x 1)
- mb_pair - learned metaprior for pairwise genotype metafeatures (# pairwise metafeatures)
- rb_pair - learned metaprior for pairwise regulator metafeatures (# pairwise metafeatures)
- ma - the weights on the genotype features calculated using the learned metapriors mb and mb_pair - # markers x #modules
- ra - the weights on the regulator features calculated using the learned metapriors rb and rb_pair - # genes x # modules
- mb1, mb0 - bounds on L1 weights for genotype markers
- rb1, rb0 - bounds on L1 weights for regulators