|  |  |
| --- | --- |
| **Script name** | 1-Optimum bandwidth using cross validation approach.R |
| **Input file** | Original intensity data after normalisation. (100 highly variable genes) |
| **Output file** | Bandwidth with minimum error. |
| **Description** | Script is for identifying optimum parameter (bandwidth) for locfit function using cross validation approach. Here we select top 100 highly variable genes (1st part of code) and iterate the loop over genes followed by each time point. 2nd part of program demonstrated that we initialise bandwidth value with broad range (1 to 20) and identify optimum bandwidth. Last part of code is similar as 2nd code except narrow range of bandwidth values (0.5 to 2.5). In each iteration for one gene we hide one time point value (say tp1)and build the model based on remaining 8 time points (e.g tp2 to tp9). Then, we are predicting value of unknown time point (i.e. tp1) based on preformed model. Error will be predicted value minus original value. Optimum bandwidth will be having minimum squared error will as number of genes. |
|  |  |
| **Script name** | 2-Locfitting of whole data using optimum bandwidth value.R |
| **Input file** | Original intensity data after normalisation. |
| **Output file** | LOCFITTED data for all genes and all time points |
| **Description** | First program gives optimum bandwidth parameter which will be used in this script to convert all original data into locfitted data. We are predicting all time point values at optimum bandwidth. |
|  |  |
| **Script name** | 3-Identifying highly variable genes across 2 strains based on STDEV filter.R |
| **Input file** | LOCFITTED data for all genes and all time points |
| **Output file** | LOCFITTED data of highly variable genes |
| **Description** | Here, purpose is to find out highly variable genes (standard deviation above 0.5). Genes which did not show high variation in both the strains (MKT1 and S288c) will not useful for analysis and here we get rid of those genes. Genes showing high variation (> 0.5) in at least one of the strain will be useful and we catch only those subset of genes. |
|  |  |
| **Script name** | 4-Row centering of data.R |
| **Input file** | LOCFITTED data of highly variable genes (both strains or single strains) |
| **Output file** | Row centered data |
| **Description** | Manually add LOCFITTED data of highly variable genes from both strains in single file (18 arrays)and run the script. It calculate mean for each gene and then each value minus mean will be returned. After row centering again separate each strain data into separate file. |
|  |  |
| **We used MeV software to create a heatmaps. You can draw heatmaps in R as well.** | |
|  |  |
| **Script name** | 5-Identify tmax for each highly variable gene.R |
| **Input file** | Row centered data of individual strains |
| **Output file** | tmax value for each gene as discrete time points |
| **Description** | Before drawing heatmap, we ordered the genes based on the time of induction (tmax) of gene. Using this script, we were identifying tmax and tmin for each gene in both the strains. Then, manually ordered the genes based on tmax (1 to 9) and within one tmax cluster genes were arranged based on tmin. |
|  |  |
| **Script name** | 6-Making data continuous to plot heatscatterplot.R |
| **Input file** | Original intensity data after normalisation. |
| **Output file** | tmax of each gene but in continues manner |
| **Description** | We want tmax of genes in continues manner. For that we predicted expression values for middle time points where actual experiments were not done. Basically we increase dimensions of our data from 9 time points to many and then calculate tmax value. |
|  |  |
| **Script name** | 7-heatscatterplot.R |
| **Input file** | tmax value in continues manner |
| **Output file** | heatscatterplot image |
| **Description** | We used LSD bioconductor package to identify genes with different tmax acroass starins. |
|  |  |
| **Script name** | 8-Comparing expression profile of individual gene across 2 strains.R |
| **Input file** | Row centering data files for both the strains |
| **Output file** | Expression profile graph for each gene |
| **Description** | Here we plot line graph to visualise expression pattern of single gene in two strains. |
|  |  |
| **Script name** | 9-Comparing expression lines drawn at different bandwidths along with original expression values.R |
| **Input file** | Original intensity data after normalisation. |
| **Output file** | Expression profile graph to visualise smoothness of data at various bandwidth values |
| **Description** | Expression profile graph to visualise smoothness of data at various bandwidth values |