PLGEM overview

Mattia Pelizzola and Norman Pavelka

November 8, 2007

In this document a brief tutorial on the **Power Law Global Error Model** (plgem; N. Pavelka *et. al.*, BMC Bioinformatics, 2004 Dec 17;5(1):203) is provided. A `wrapper` (*run.plgem*) exists that performs all the steps necessary to obtain a list of differentially expressed genes (DEG), starting from a dataset of class *ExpressionSet*. It automatically attempts to find the best solution at each step, and requires only modest or no input decisions by the user. For this purpose we use the microarray dataset used in the paper, where four replicates of LPS stimulated dendritic cells (LPS) are compared to sixteen replicates of untreated dendritic cells (C):

```
> library(plgem)
> data(LPSeset)
> LPSdegList <- run.plgem(esdata = LPSeset)
```

For advanced users, the individual functions are also provided in this tutorial and we will use them now, step by step.

The first step is the `fitting of the model` on a microarray dataset. By fitting the model we will obtain a mathematical relationship that allows us to determine the expected variability associated to a given expression value.

PLGEM can only be fitted on a set of replicates of a same experimental condition, therefore we first need to choose which condition to use for the fitting step. In our dataset two conditions are provided: C and LPS. Usually the most replicated one is chosen, therefore we choose the first condition C, that contains 16 replicates and corresponds to a *fit.condition* of 1. Moreover it is necessary to set p and q parameters.

Briefly p represents the number of intervals used to partition the expression value range of the dataset; we observed that p can be modified over a wide range of values, without any major effects on the final results, except when it is chosen to close to the total number of genes on the microarray. The default of 10 will be OK for most purposes.

q is the quantile of the location-dependent spread used to fit the model. The default of q is set to 0.5, because this represents the median value, which is what you are looking for when modeling the variability. We recommend to only modify this parameter for very special purposes, e.g. for the determination of empirical confidence intervals of standard deviation.

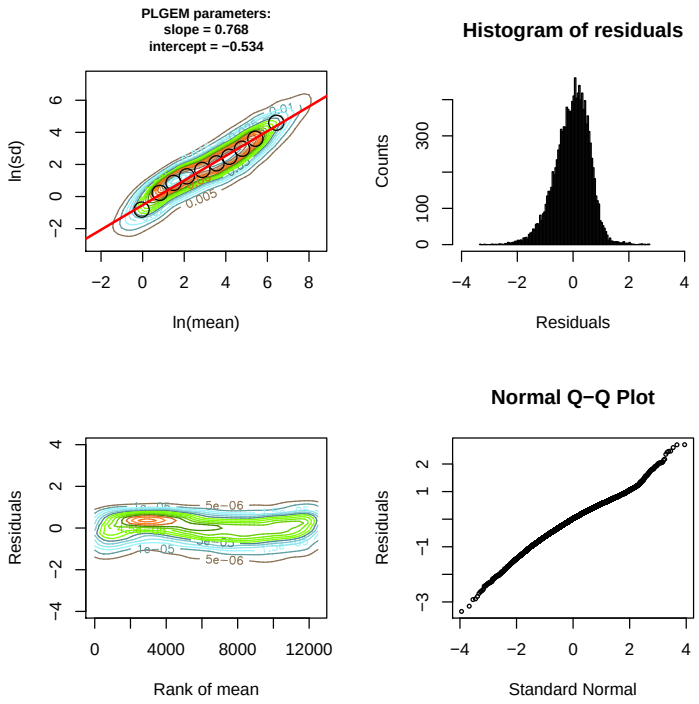
Finally setting *plot.file*=TRUE saves a png image to a file instead of plotting it, therefore we won't change its default.

Moreover it is possible to evaluate the fitting of the model setting the option *fittingEval*. If this argument is set to TRUE, a multi-panel plot is produced

where the residuals of the model are evaluated. A good fit is characterized by a near-normal distribution and an horizontal symmetric ranking-plot of the residuals.

```
> LPSfit <- plgem.fit(data = LPSeset, covariateNumb = 1, fit.condition = 1,
+     p = 10, q = 0.5, plot.file = FALSE, fittingEval = TRUE, verbose = TRUE)

fitting PLGEM..
samples extracted for fitting:
   conditionName
C1             C
C2             C
C3             C
C4             C
determining modelling points...
fitting data and modelling points...
done with fitting PLGEM.
```



The next step is the `computation of the Signal to Noise ratio` (STN) statistic for the detection of differential expression. The STN is determined using the model-derived spread estimates instead of the data-derived ones. Therefore it is necessary to give to the *plgem.obsStn* function the model parameters *slope* and *intercept* determined during the model fitting that are contained in the value returned by *plgem.fit* function. By default, all experimental conditions (according to the values of the *covariateNumb* covariate defined in the phenoData slot of the ExpressionSet) are compared to the first condition (baseline).

If the condition to be treated as the baseline is not the first one, you can change this by modifying the argument *baseline.condition*. A matrix of observed STN is determined, where its dimensions are the number of probesets and the number of comparisons that can be performed in the dataset. In this case, the dataset contains only one condition to be compared to the baseline, therefore the matrix is reduced to a vector of observed PLGEM-STN.

```
> LPSobsStn <- plgem.obsStn(data = LPSeset, covariateNumb = 1,
+     baseline.condition = 1, plgemFit = LPSfit, verbose = TRUE)

calculating observed PLGEM-STN statistics:found 1 condition(s) to compare to the baseline.
working on baseline C ...
C1 C2 C3 C4
working on condition LPS ...
LPS1 LPS2
done with calculating PLGEM-STN statistics.
```

In order to get an estimate of the distribution of the test statistic under the null-hypothesis of not differential expression, a `resampled statistic` is determined using the method described in the paper. The number of iterations of the resampling step should be correlated with the total number of replicates that are present in the data set. If this argument is set to *automatic*, the number of iterations is automatically determined based on the total number of possible combinations. In this case, an upper threshold of 500 iterations is set to avoid excessive computation time. This should be fine for most purposes.

```
> LPSresampledStn <- plgem.resampledStn(data = LPSeset, plgemFit = LPSfit,
+     iterations = "automatic", verbose = TRUE)

calculating resampled PLGEM-STN statistics:found 1 condition(s) to compare to the baseline
baseline samples:
C1 C2 C3 C4
resampling on samples:
C1 C2 C3 C4
Using  500  iterations...
working on cases with  2  replicates...
     Iterations: 20  40  60  80  100  120  140  160  180  200  220  240  260  280  300  32
done with calculating resampled PLGEM-STN statistics.
```

Finally `DEG are selected` by comparing the observed statistics to the distribution of resampled STN. For this purpose two non-symmetrical thresholds are defined at the given significance level *delta*. *Delta* is roughly an estimate of the False Positive Rate (FPR). Therefore if 10000 probesets are evaluated with a *delta* of 0.001, 10 probesets are expected to be selected as DEG by chance. Therefore, if 500 DEG are selected, 10 of them are expected to be false positives, but obviously it is not possible to know which one they are.

This function returns a list with a number of items is equal to the number of different significance levels *delta* used as input. In this case the default single value of 0.001 was used, so the list will contain only one item at this level. This item is again a list, whose number of items correspond to the number of performed comparisons, i.e. the number of conditions in the starting ExpressionSet minus the baseline, in this case again only one. In each list-item the values are the observed STN and the names are the DEG probeset ids.

```
> LPSdegList <- plgem.deg(observedStn = LPSobsStn, plgemResampledStn = LPSresampledStn,
+     delta = 0.001, verbose = TRUE)

selecting significant DEG:found 1 condition(s) compared to the baseline.
Delta =  0.001
        Condition =  LPS
delta: 0.001 condition: LPS found 354 DEG
done with selecting significant DEG.

> LPSdegList[["0.001"]][["LPS"]][1:5]

  100012_at 100213_f_at   100277_at   100278_at 100342_i_at
  -4.782786    3.729672    6.607890    3.643951    3.784421
```

Finally, the obtained list of DEG can be written on the disk using the *plgem.write.summary* function.