

HowTo plw

Magnus Åstrand

May 3, 2016

1 Introduction

This vignette describes how to use *plw*, an R implementation of the Probe level Locally moderated Weighted median-t (PLW) method (?) for finding differentially expressed genes. PLW uses an empirical Bayes model taking into account the dependency between variability and intensity-level. A global covariance matrix is also used allowing for differing variances between arrays as well as array-to-array correlations, and thus PLW performs weighted analysis. PLW is specially designed for Affymetrix type arrays (or other multiple-probe arrays). Instead of making inference on probe-set summaries, comparisons are made separately for each perfect-match probe and are then summarized into one score for the probe-set. The Locally Moderated Weighted-t (LMW) method, applying the model of PLW on probe-set summaries or data from single probe arrays, is also implemented in the *plw* package. See ? for details on PLW and LMW, and ??, ?, and ? for details on weighted analysis for microarrays. PLW is demonstrated in Sections ?? to ??, and LMW in Section ??.

2 Data

The R-package *plw* depends on the *affy* package, available from the Bioconductor¹ project, which is loaded automatically when loading *plw*:

```
> require(plw)
```

The *affy* package contains functions for reading CEL-file data into an **AffyBatch** object using the function **ReadAffy**. It also contains functions for doing low-level analysis, such as background correction, normalization, and calculating expression indexes. For example, the **rma** function performs all three steps in one call and returns an **ExpressionSet** object holding RMA expression indexes. For further details on how to read CEL-file data into R use

```
> help(ReadAffy)
```

In this vignette the PLW method is demonstrated using the **AffySpikeU95Subset** data set of 6 arrays and 1016 probe-sets. The data set was loaded using the **ReadAffy** function and is included in the *plw* package. **AffySpikeU95Subset** is a sub-set of the Affymetrix U95 Latin-Square spike-in data set of 59 arrays and 12626 probe-sets. For these data there are 16 known differentially expressed genes/probe-sets ?, of which all 16 are included in **AffySpikeU95Subset**. The data set is loaded using

```
> data(AffySpikeU95Subset)
```

```
> AffySpikeU95Subset
```

¹<http://bioconductor.org/>

```

AffyBatch object
size of arrays=182x182 features (18 kb)
cdf=HGU95subset1016 (1016 affyids)
number of samples=6
number of genes=1016
annotation=hgu951016
notes=

```

3 Running PLW

The `AffySpikeU95Subset` data set use data from groups a and b of the Affymetrix U95 Latin-Square spike-in data set. Here we show how to do a comparison of these two groups. The fifth letter of the CEL-file names holds the group assignment of each array which we can inspect using the `pData` function

```
> pData(AffySpikeU95Subset)
```

	sample
1521a99hpp_av06.CEL	1
1532a99hpp_av04.CEL	2
2353a99hpp_av08.CEL	3
1521b99hpp_av06.CEL	4
1532b99hpp_av04.CEL	5
2353b99hpp_av08r.CEL	6

We define a design using the function `model.matrix`, and a contrast matrix for comparing groups a and b.

```

> group<-factor(rep(letters[1:2],each=3))
> design<-model.matrix(~group-1)
> contrast<-matrix(c(1,-1),1,2)

```

```
> design
```

	groupa	groupb
1	1	0
2	1	0
3	1	0
4	0	1
5	0	1
6	0	1

```

attr("assign")
[1] 1 1
attr("contrasts")
attr("contrasts")$group
[1] "contr.treatment"

```

```
> contrast
```

	[,1]	[,2]
[1,]	1	-1

Now we are ready to use the `plw` function.

```
> plwFit<-plw(AffySpikeU95Subset,design=design,contrast=contrast,epsilon=1e-05)
```

```
> plwFit
```

```
Call:
```

```
plw(x = AffySpikeU95Subset, design = design, contrast = contrast,      epsilon = 1e-05)
```

```
Number of arrays      : 6
Number of probe-sets  : 1016
Number of PM probes   : 16256
Number of knots for v: 6
m parameter           : 9.328
Df for probe t-stat.  : 13.3
Convergence status    : TRUE
Number of iterations   : 51 12
```

From the output we can see that steps 1 and 2 of the procedure used in `plw` required 51 and 12 iterations, respectively (see `?` for details of the procedure). The estimated value for the m -parameter is 9.328 and the degrees of freedom for the moderated t-statistics is 13.3.

4 Analysing PLW output

There are three functions for displaying the ranking of probe-sets with respect to differential expression, `topRankSummary`, `plotSummaryT`, and `plotSummaryLog2FC`. All three show results for a given number of top ranking probe-sets (e.g. probe-set ranked 1-20), for a specific list of ranks (e.g. probe-set ranked 1,5, and 7), or for a specific list of probe-sets. For example we can display the result for the 16 spiked-in probsets.

```
> topRankSummary(plwFit,genes=spikedProbesU95)
```

	Rank	Median t	Q1-t	Q3-t	Med. log2FC
37777_at	16	-1.032	-2.11	-0.5477	-0.323
684_at	61	-0.702	-1.64	-0.0697	-0.145
1597_at	54	-0.709	-1.56	-0.0339	-0.138
38734_at	8	-3.948	-4.94	-1.7735	-0.666
39058_at	10	-3.148	-4.32	-2.4664	-0.562
36311_at	4	-5.612	-6.89	-3.7220	-0.794
36889_at	9	-3.657	-4.82	-1.4953	-0.650
1024_at	3	-5.719	-6.84	-5.1256	-0.895
36202_at	2	-6.059	-7.06	-5.3868	-0.827
36085_at	5	-5.394	-6.08	-4.3263	-0.569
40322_at	11	-2.627	-3.25	-1.9677	-0.250
407_at	13	-1.203	-2.51	-0.1519	-0.353
1091_at	12	-1.703	-3.50	-0.7309	-0.165
1708_at	1	37.206	31.92	45.3687	7.049
33818_at	7	-4.718	-4.86	-3.3829	-0.512
546_at	6	-4.759	-5.90	-2.3678	-0.695

We can also display results for probe-sets ranked 11 to 20,

```
> topRankSummary(plwFit,genesOfRank=11:20)
```

	Rank	Median t	Q1-t	Q3-t	Med. log2FC
40322_at	11	-2.627	-3.251	-1.968	-0.250
1091_at	12	-1.703	-3.500	-0.731	-0.165
407_at	13	-1.203	-2.513	-0.152	-0.353
36400_at	14	1.126	0.530	1.565	0.273

33040_at	15	1.053	0.156	2.100	0.311
37777_at	16	-1.032	-2.107	-0.548	-0.323
31642_at	17	1.026	0.700	1.972	0.342
39311_at	18	1.008	0.168	1.324	0.165
39045_at	19	-0.996	-1.400	0.126	-0.133
33527_at	20	0.967	0.306	1.286	0.348

Alternatively, we can display the result for the 20 top ranking probe-sets,

```
> topRankSummary(plwFit,nGenes=20)
```

	Rank	Median	t	Q1-t	Q3-t	Med. log2FC
1708_at	1	37.206	31.924	45.369	7.049	
36202_at	2	-6.059	-7.058	-5.387	-0.827	
1024_at	3	-5.719	-6.845	-5.126	-0.895	
36311_at	4	-5.612	-6.886	-3.722	-0.794	
36085_at	5	-5.394	-6.085	-4.326	-0.569	
546_at	6	-4.759	-5.895	-2.368	-0.695	
33818_at	7	-4.718	-4.856	-3.383	-0.512	
38734_at	8	-3.948	-4.941	-1.774	-0.666	
36889_at	9	-3.657	-4.818	-1.495	-0.650	
39058_at	10	-3.148	-4.317	-2.466	-0.562	
40322_at	11	-2.627	-3.251	-1.968	-0.250	
1091_at	12	-1.703	-3.500	-0.731	-0.165	
407_at	13	-1.203	-2.513	-0.152	-0.353	
36400_at	14	1.126	0.530	1.565	0.273	
33040_at	15	1.053	0.156	2.100	0.311	
37777_at	16	-1.032	-2.107	-0.548	-0.323	
31642_at	17	1.026	0.700	1.972	0.342	
39311_at	18	1.008	0.168	1.324	0.165	
39045_at	19	-0.996	-1.400	0.126	-0.133	
33527_at	20	0.967	0.306	1.286	0.348	

The other two functions plot individual values for each perfect-match probe together with the median value. The `plotSummaryT` plots t-statistics, whereas `plotSummaryLog2FC` plots logged fold-change values, as shown in Figures ?? and ??, respectively.

```
> plotSummaryT(plwFit,genes=spikedProbesU95)
```



Figure 1: T-statistics for spiked-in probsets.

```
> plotSummaryLog2FC(plwFit,nGenes=15)
```



Figure 2: Logged fold-change values for the 15 top ranking probe-sets.

The `plw` function uses an empirical bayes model with an inverse-gamma prior for the unknown variances, where the scale parameter of the inverse-gamma prior is modeled as a function of mean intensity. With the `varHistPlot` function we can compare the fitted distribution for $\log(s^2)$ with the observed data, and with the `scaleParameterPlot` function we can look at the fitted curve for the scale parameter ν of the inverse-gamma prior. See Figures ?? and ??, respectively.

```
> varHistPlot(plwFit)
```



Figure 3: Comparing the fitted distribution for $\log(s^2)$ with the observed data.

```
> scaleParameterPlot(plwFit)
```



Figure 4: Fitted curve for the scale parameter ν of the inverse-gamma prior.

5 LMW on two-color microarray data

In ? the LMW method is used on RMA expression indexes, and `example(lmw)` shows how to use LMW on Affymetrix or other one-color array data. This section demonstrates how to use LMW on the ApoAI data-set (?), comparing 8 ApoAI knockout mice with 8 normal mice using a set of $n = 16$ two-color cDNA-arrays. Data was pre-processed as described in (?) and the analysis presented here is based on the 6068 genes (out of 6226) having no missing values.

```
> source("http://www.math.chalmers.se/~astrandm/plw/GetApoAIdata.R")
> RG <- GetApoAIdata()
> require(limma)
> MA <- normalizeWithinArrays(RG)
> rownames(MA$M) <- MA$genes$Name
> ii <- apply(is.na(MA$M), 1, any)
> MA$A <- MA$A[!ii,]
> MA$M <- MA$M[!ii,]
```

Arrays 1 to 8 is the control group with mRNA from normal mice, whereas arrays 9 to 16 are from the knockout group. Thus, we specify a design and contrast matrix for the comparison of knockout mice with the control group of normal mice.

```
> design <- cbind("Control-Ref"=1, "KO-Control"=MA$targets$Cy5=="ApoAI KO")
> contrast <- matrix(0:1, ncol=2)

> design
> contrast
```

The analysis using LMW is done using the mean intensity of the sum of logged green and red signal, respectively, to model the scale parameter of the inverse-gamma prior for error variances. Also, the spline-knots for the scale-parameter function are set using a set of sample quantiles (10, 30, 50, 70, and the 90% quantile) of the mean intensity instead of the default knots computing using an internal function.

```
> meanX <- apply(MA$A, 1, mean)
> knots <- quantile(meanX, seq(0.1, 0.9, by=0.2))
> lmwFit <- lmw(MA$M, design=design, contrast=contrast, meanX=meanX, knots=knots)

> lmwFit
```

From the fitted model we can select the top 10 ranked genes from the analysis,

```
> topRankSummary(lmwFit, nGenes=10)
```

and inspect the model fit for the inverse-gamma prior together with the estimated scale-parameter curve,

References

- M. Åstrand, P. Mostad, and M Rudemo. Empirical bayes models for multiple probe type arrays at the probe level. Technical report, Chalmers University of Technology and Göteborg University, Department of Mathematical Statistics, 2007a. URL <http://www.math.chalmers.se/Math/Research/Preprints/2007/27.pdf>.
- M. Åstrand, P. Mostad, and M Rudemo. Improved covariance matrix estimators for weighted analysis of microarray data. *J. Comput. Biol.*, Accepted, appearing in number 10, 2007b.
- Matthew J. Callow, Sandrine Dudoit, Elaine L. Gong, Terence P. Speed, and Edward M. Rubin. Microarray Expression Profiling Identifies Genes with Altered Expression in HDL-Deficient Mice. *Genome Res.*, 10(12):2022–2029, 2000.
- Leslie M. Cope, Rafael A. Irizarry, Harris A. Jaffee, Zhijin Wu, and Terence P. Speed. A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics*, 20(3):323–331, 2004.
- Erik Kristiansson, Anders Sjögren, Mats Rudemo, and Olle Nerman. Weighted analysis of paired microarray experiments. *Stat. Appl. Genet. Mol. Biol.*, 4(1):article 30, 2005.
- Erik Kristiansson, Anders Sjögren, Mats Rudemo, and Olle Nerman. Quality optimised analysis of general paired microarray experiments. *Stat. Appl. Genet. Mol. Biol.*, 5(1):article 10, 2006.
- Anders Sjögren, Erik Kristiansson, Mats Rudemo, and Olle Nerman. Weighted analysis of general microarray experiments. *BMC Bioinformatics*, 8(1):article 387, 2007.