## affyPLM: the threestep function

# Ben Bolstad bmb@bmbolstad.com http://bmbolstad.com

October 13, 2015

#### Contents

1	Introduction	1
2	Using the threestep function	1
3	Methods available in threestep	2

#### 1 Introduction

This document describes the threestep function which is part of the affyPLM package. The threestep function is an alternative method of computing expression measures (which can also be generated using expresso, rma and other functions from the affy package). In the threestep framework an expression measure consists of three steps: background/signal adjustment, normalization and then summarization. One important thing to notice is that all expression measures returned by threestep are all in the log<sub>2</sub> scale.

After starting R, the package should be loaded using this will load affyPLM as well as the affy package and its dependencies.

### 2 Using the threestep function

The simplest method of calling the threestep function is to call it with no arguments beyond an *AffyBatch*.

- > require(affydata)
- > data(Dilution)
- > ##FIXME: remove the next line

```
> Dilution = updateObject(Dilution)
> eset <- threestep(Dilution)</pre>
```

this will compute the conventional RMA expression measure.

But we can go beyond this by altering any of the three steps. This can be done using the three parameters background.method, normalize.method and summary.method. For example

```
> eset <- threestep(Dilution, background.method = "MASIM",
+ normalize.method="quantile",summary.method="tukey.biweight")</pre>
```

computes an expression measure where we background correct/signal adjust by breaking the array into grids doing a location dependent adjustment, then subtracting the "ideal mismatch" from the perfect match. Normalization is done using the quantile normalization method and then summarization using a 1-step Tukey Biweight.

An even more unconventional expression measure is to background correct/signal adjust by subtracting the ideal mismatch, carry out no normalization, then summarize by taking the 2nd largest PM (which in this case will really be the largest PM after subtracting the ideal mismatch).

```
> eset <- threestep(Dilution, background.method = "IdealMM",
+ normalize="quantile",summary.method="log.2nd.largest")</pre>
```

#### 3 Methods available in threestep

The following tables outline the methods that are available for each of the threesteps.

Background methods				
Option	Name			
RMA.2 (default)	RMA convolution model background			
RMA.1	RMA convolution model background			
MAS	MAS 5.0 location dependent background			
IdealMM	Ideal Mismatch			
MASIM	MAS 5 and Ideal Mismatch			
GCRMA	GCRMA background			

Normalization methods			
Option	Name		
quantile (default)	Quantile normalization		
scaling	scaling normalization		

Summarization methods			
Option	Name		
median.polish (default)	Median polish		
tukey.biweight	1 step tukey biweight		
average.log	Average of Logs		
log.average	Log of Average		
median.log	Median of Logs		
log.median	Log of Median		
log.2nd.largest	2nd largest PM		
lm	Linear Model		
rlm	Robust Linear Model		