Harshlight: HowTo

Maurizio Pellegrino, Yupu Liang

October 17, 2016

Contents

1 Introduction

This document describes briefly how to use the `Harshlight` package.

1.1 Background

Analysis of hybridized microarrays starts with scanning the fluorescent image. The quality of data scanned from a microarray is affected by a plethora of potential confounders, which may act during printing/manufacturing, hybridization, washing, and reading. For high-density oligonucleotide arrays (HDONAs) such as Affymetrix GeneChip oligonucleotide (Affy) arrays, each chip contains a number of probes specifically designed to assess the overall quality of the biochemistry, whose purpose is, e.g., to indicate problems with the biotinylated B2 hybridization. Affymetrix software and packages from Bioconductor project for R provide for a number of criteria and tools to assess overall chip quality, such as percent present calls, scaling factor, background intensity, raw Q, and degradation plots. However, these criteria and tools have little sensitivity to detect localized artifacts, like specks of dust on the face of the chip, which can substantially affect the sensitivity of detecting physiological (i.e., small) differences. In the absence of readily available safeguards to indicate potential physical blemishes, researchers are advised to carefully inspect the chip images visually. Unfortunately, it is impossible to visually detect any but the starkest artifacts against the background of hundreds of thousands of randomly allocated probes with high variance in affinity.

`Harshlight` analyzes the content of Affymetrix microarray data stored in `.CEL` files, detecting and eliminating the artifacts that microarray images present on their surface.

1.2 Get started

This document outlines how to get the data from `.CEL` files and analyze it with `Harshlight`.

As a start, the package needs to be loaded in your `R` session.

```
R> library(Harshlight) ##load the Harshlight package
```

note: you need to have the `affy` package in your library in order for Harshlight to work.

2 How-to

2.1 Reading .CEL files

Harshlight analyzes an Affybatch object, derived from your `.CEL` files. Use the function `ReadAffy` included in the `affy` package to read the information of .CEL files.

```
R> abatch <- ReadAffy(celfile.path = "path_to_CEL_file")
```

This will read all the .CEL files found in "path_to_CEL_file" and will store their information in `abatch` (Affybatch object). For more information on how to use `ReadAffy` refer to the `affy` help page, `help(ReadAffy)`.

3 Harshlight

3.1 Functions

The `Harshlight` package contains several functions for the detection of different kinds of artifacts. Once the affected probes are detected, the values of those probes in the chip are substituted (see `Analysis` or the R help file for the package).

`Harshlight` is the main function that allows the detection of extended artifacts (i.e. blemishes that affect the overall chip), compact defects (i.e. blemishes that affect most of the probes in a circumscribed area), and diffuse defects (i.e. blemishes that cover large areas on the surface of the chip but do not affect all the probes in that area). The defects are detected in that order; all blemishes found in a round of detection are eliminated before the next round starts.

`HarshExt` is used to detect only extended blemishes; neither compact nor diffuse blemishes are considered.

`HarshComp` is used when only the detection of extended and compact blemishes is desired, while diffuse blemishes are ignored.

Harshlight makes use of several user-tunable parameters in order to best detect the blemishes on the chips. For a more detailed description of the parameters refer to the package help page.

3.2 Analysis

The functions found in `Harshlight` perform the analysis detecting the different blemishes previously described. Once found, the user has the option to substitute the defects with two values: the median value of the same pixel in the other chips (default), or NA. The first option is preferred, for example, when the microarray batch will be subject to other analyses that do not accept NA values (e.g. rma).

```
R> abatch.Harshlight <- Harshlight(affy.object = abatch, na.sub = FALSE)
```

To run `Harshlight` on a sample `affybatch` object, download the file example.rda in your working directory from our website http://asterion.rockefeller.edu/Harshlight/ and load it into your environment.

```
R> load('example.rda')
```

Then run `Harshlight` and store the result in another object.

```
R> harsh.affybatch <- Harshlight(my.affybatch,report.name='MyReport.ps')
```

`Harshlight` writes a report specified by the variable `report.name` at the end of the analysis, with a summary of what was found in the microarrays. The file is in `.ps` format: this can be read by readers such as `Ghostscript`.

The results of the analysis are stored in `harsh.affybatch`. This is an `affybatch` object in which the affected values were substituted. Once this is done, the `affybatch` object can be written back into `.CEL` files using the add-on package `Helper` (donwloadable from the web site http://asterion.rockefeller.edu/Harshlight/).

You can then continue analyzing your microarray chips with other programs. For example:

```
R> harsh.mas5 <- mas5(harsh.affybatch)
R> harsh.rma  <- rma(harsh.affybatch)
```

4 References

For more information on the algorithm behind the package, see the reference below.

```
Harshlight: a "corrective make-up" program for microarray chips
Mayte Suarez-Farinas*, Maurizio Pellegrino*, Knut M Wittkowski and Marcelo O Magnasco,
*These two authors contributed equally
```

For problems, bug reports, and questions, write to:

```
Maurizio Pellegrino, mpellegri@berkeley.edu
```