# affyILM

March 24, 2012

---

`ILM-class` *Class to contain the results of an ilm calculation*

---

## Description

S4 object to contain intensities, probe concentrations, annotations, and summarized expression values.

## Slots

`Ipm`: A `"matrix"` of size (number of probes) x (number of CEL-files). Each columns corresponds to one CEL-file with the raw PM.

`I0`: A `"matrix"` of size 1x1 or (number of probes) x (number of CEL-files). In the current release, I0 is set to 0 (default value). In the future releases, I0 will include estimates of background intensities.

`probe.concs`: A `"matrix"` of size (number of probes) x (number of CEL-files) holding the concentration for each probe (picoMolar) computed according to the Langmuir model.

`exprSummary`: A `"list"` with 1 slots, `Probe.Set` and `Cluster.Set`, each containing either NA of a `matrix` of size (number of sets) x (number of CEL-files). Each slot has been included to provide summarized expression values (using medianpolish or transposed medianpolish), computed at the level of Probe Sets (old and new generation arrays), or Cluster Sets (only for whole gene arrays, if probe table contains `"Cluster.Set.Name"` definition, added manually)

`se.exprSummary`: A `"list"` with 1 slots, `Probe.Set` and `Cluster.Set`, each containing either NA of a `matrix` of size (number of sets) x (number of CEL-files). Each slot has been included to provide standard deviation estimates for summarized expression values, computed for each Probe Set (old and new generation arrays), or Cluster Set (only for whole gene arrays, if probe table contains `"Cluster.Set.Name"` definition, added manually)

`satLim`: Is the `"numeric"` saturation limit of the intensities of the Langmuir Isotherm, i.e. where the concentration is high or the probe-target binding free energies are large. The default value of `A` is 10000.

`deltaG.pm`: A `"matrix"` providing the DNA-RNA hybridization free energies (deltaG) computed for each PM probe sequence.

`deltaGp.pm`: A `"matrix"` providing the RNA-RNA hybridization free energies (deltaG) for each PM probe sequence.

`info`: A `"list"` containing several annotations : `"ncol"` and `"nrow"` provide the chip dimensions, `"cdfName"` provides information on the array model,`"pmindex"` contains the indices of the probes together with the name of the Probe Set, `"alpha"` is a matrix with intermediate value used by the Langmuir Model (used by the function `plotILM`), and `"probe.table"` contains the probe annotation table.

## Methods

`[` Subset ILM objects

**getIntens** Get background intensity (of a particular probe set)

**getProbeConcs** Get probes concentrations in pM

**getExprSummary** Get summarized expression values

**getSDSummary** Get estimates for summarized expression values

**plotIntens** Plot probe intensities of selected probe set

**plotILM** Plot Langmuir Isotherm for selected probe set

## Author(s)

Myriam Kroll, Fabrice Berger and Enrico Carlon

## Examples

```
showClass("ILM")
```

---

`ilm-methods` *Methods to access the results of ilm.*

---

## Description

These methods allow to access the results of the function `ilm` stored in an object of type `ILM`.

## Usage

```
getIntens(object,y)
getProbeConcs(object,y)
getExprSummary(object,y,z)
getSDSummary(object,y,z)
```

## Arguments

| | |
|---|---|
| `object` | An object of type `ILM` |
| `y` | A character string or vector of probe set name(s) |
| `z` | A character string specifying the type of set that is requested |

## Details

"getIntens()" is used to access the intensity values.

The probe concentrations are calculated on behalf of the Langmuir model. For each probe (of a probeset) the concentration is estimated in picoMolar and can be accessed via "getProbeConcs()". Medianpolish, Transposed Medianpolish or Median can then be used to compute probeset summarized expression values.

The results can be accessed via "getExprSummary()" and the associated standard deviation is accessed via "getSDSummary()". z, if specified, can take two values : Probe.Set and Cluster.Set (the last one has been created for the definition of cluster sets in the last generation of arrays, whole-gene).

If y=NULL, the results for all probe sets are shown.

## Value

A "matrix"

An object of the class ILM when subsetting "["

## Author(s)

Myriam Kroll, Fabrice Berger and Enrico Carlon

## See Also

ILM

## Examples

```
## Locate and read in CEL-file
path <- system.file("rawData", "FusionSDK_HG-Focus", "HG-Focus", "2.Calvin",
    package="AffymetrixDataTestFiles")
file1 <- file.path(path,"HG-Focus-1-121502.CEL")
## Calculation of background estimates and expression values (concentrations)
result <- ilm(file1)
## Background intensities of all probes
getIntens(result)
## Background intensities for one or more probesets
getIntens(result,"203561_at")
getIntens(result,c("203561_at","40359_at"))

## Get concentrations of all probesets
getProbeConcs(result)
## Get concentrations (in picoMolar)
getProbeConcs(result,"203561_at")
getProbeConcs(result,c("203561_at","40359_at"))

## Subsetting
result["203561_at"]
```

`ilm` *affyILM*

## Description

This function is the working horse of the package and is used as an overall function to calculate the background intensities as well as the concentrations.

## Usage

```
ilm(celfiles,satLim=10000,scale.method="linear",scale.target="concs",cdf.name=NU
```

## Arguments

`celfiles` filenames of CEL-files

`satLim` Saturation Limit of the Langmuir isotherm (determined by scanner). Default value 10000.

`scale.method` Method used to scale the values between arrays. "linear" (default) uses a coefficient to adjust slopes of all pairwise comparisons to unity, and "linear.stat" makes use of both median and median absolute deviation to scale the values.

`scale.target` Specify on which value the scaling step has to be performed. Authorized values are "intens" and "concs", respectively refering to probe intensities and probe concentrations.

`cdf.name` Name of a package providing the chip definition file. By default, this information is extracted from the chip-model provided by the CEL files. This parameter can be used to specify alternative definition files.

`probe.table` A matrix providing probe annotation table, with the same structure than the probe package provided on Bioconductor. Manual creation of probe tables, or subsets, can be used in place of alternative definition files and probe packages. An additional column, named "Cluster.Set.Name", can be added for new whole-gene arrays.

`probe.name` Name of a package providing the probe annotation table. By default, this information is extracted from the chip-model provided by the CEL files. This parameter can be used to specify alternative probe table packages.

`na.replace` Experimental. The methods that can be used to replace missing values, for each probe set. na.replace should be a list with 2 slots, each containing a function. The first slot should contain the function used to missing probe concentrations due to missing probe intensities (i.e. median), and the second slot should contain the function used to compute probe concentrations when the missing value is due to intensity values that are higher than the saturation limit defined by satLim (i.e. max). Those function are then used to compute replacement values based on concentration values of the same probe set.

`bgcorrect` Not used in the current release. Default value is FALSE. This parameter will be use in the next releases to compute the background correction.

`summarize.level` Parameter for summarization (character). Default value is "none", to avoid this step. Other accepted values are "probeset" for traditionale definition of probe sets, "cluster" if a Cluster.Set.Name column has been manually added to the probe table and provided with parameter `probe.table`, or "both" to run summarization using both definitions.

`summary.method`
: Parameter for summarization (character). Default value is "none", to avoid this step. Other accepted values are "median" to use the median of probe concentrations, "medpolish" to use the medianpolish on the log2 of concentrations, and "tmedpolish" for the transposed medianpolish procedure.

`summary.na.rm`
: Parameter for summarization (logical). Default value is TRUE, sothat to be able to compute summarized values when data contains missing values.

`dgDRpairs`
: Experts only. Allow to tune the computation of DNA/RNA hybridization free energies. dgDRpairs is used to provide list of deltaG values for di-nucleotides.

`dgRRpairs`
: Experts only. Allow to tune the computation of RNA/RNA hybridization free energies. dgRRpairs is used to provide list of deltaG values for di-nucleotides.

`beta`
: Experts only. Allow to tune the computation of concentrations using the Langmuir Isotherm. beta = 1/RT allow to change the effective hybridization temperature.

## Details

The Langmuir Isotherm is used as a model to estimate probe concentrations from measured PM intensities, thanks to the computation of sequence-specific DNA/RNA hybridization free energies.

## Value

An object of class `ILM`

## Note

The AffymetrixDataTestFiles-package must be installed to run examples.

## Author(s)

Myriam Kroll, Fabrice Berger, Gerard Barkema and Enrico Carlon

## References

KM Kroll, E Carlon and GT Barkema (2009), Linear method for fast background subtraction in oligonucleotide microarrays Algorithms for Molecular Biology 2009, 4:15 G Mulders, GT Barkema and E Carlon, Inverse Langmuir method for oligonucleotide microarray analysis, BMC Bioinformatics (2009) 10, 64

## See Also

`getIntens`, `getProbeConcs`, `getExprSummary`, `getSDSummary`

## Examples

```
## Locate and read in CEL-file
path <- system.file("rawData", "FusionSDK_HG-Focus", "HG-Focus", "2.Calvin",
    package="AffymetrixDataTestFiles")
file1 <- file.path(path,"HG-Focus-1-121502.CEL")
## Calculation of background estimates and expression values (concentrations)
result <- ilm(file1)
## show all
show(result)
```

```
## per probeset (example probeset randomly chosen)
result["AFFX-r2-P1-cre-5_at"]

## Analysis of two files
file2 <- file.path(path,"HG-Focus-2-121502.CEL")
result2 <- ilm(c(file1,file2))
```

---

`plotILM` *plotILM*

---

## Description

Illustrate the Langmuir Isotherm for selected probe set.

## Usage

```
plotILM(object,y,z,...)
```

## Arguments

| | |
|---|---|
| `object` | An object of type `ILM` |
| `y` | A probe set |
| `z` | The name of a sample |
| `...` | Graphical parameters can be given as arguments to `par` |

## Details

`plotIntens` plots one graph for each CEL-file. Note that it is only possible to plot one probe set at a time (if y (or z) is a vector, only the first value is used). If y=NULL or z=NULL, there is no output. Optional plot.error argument (numeric) can be used to define the illustration of the variability. plot.error=1 computes the error on the log scale. plot.error=2 (default value) computes the error on the concentrations. plot.error=3 computes the error on the concentrations, for 2 intervals. Note that error curves can only be plotted if concentration is higher than error (negative concentration does not exist!).

## Author(s)

Myriam Kroll, Fabrice Berger and Enrico Carlon

## See Also

`ILM`,`plotIntens`

## Examples

```
path <- system.file("rawData", "FusionSDK_HG-Focus", "HG-Focus", "2.Calvin",
    package="AffymetrixDataTestFiles")
file1 <- file.path(path,"HG-Focus-1-121502.CEL")
file2 <- file.path(path,"HG-Focus-2-121502.CEL")
result2 <- ilm(c(file1,file2))
## plot output
plotILM(result2,y="203561_at",z="HG-Focus-2-121502.CEL")
```

## plotIntens *plotIntens*

### Description

Plot the PM intensity and the calculated background of one probe set.

### Usage

```
plotIntens(object,y,z,...)
```

### Arguments

| | |
|---|---|
| object | An object of type ILM |
| y | A probe set |
| z | The name of a sample |
| ... | Graphical parameters can be given as arguments to par |

### Details

plotIntens plots one graph for each CEL-file. Note that it is only possible to plot one probe set at a time (if y (or z) is a vector, only the first value is used). If y=NULL or z=NULL, there is no output.

### Author(s)

Myriam Kroll, Fabrice Berger and Enrico Carlon

### See Also

ILM,plotILM

### Examples

```
path <- system.file("rawData", "FusionSDK_HG-Focus", "HG-Focus", "2.Calvin",
    package="AffymetrixDataTestFiles")
file1 <- file.path(path,"HG-Focus-1-121502.CEL")
file2 <- file.path(path,"HG-Focus-2-121502.CEL")
result2 <- ilm(c(file1,file2))
## plot output
plotIntens(result2,y="203561_at",z="HG-Focus-2-121502.CEL")
```

# Index


*Topic **classes**
ILM-class, 1
*Topic **manip**
ilm, 4
*Topic **methods**
ilm-methods, 2
plotILM, 6
plotIntens, 7
[,ILM-method *(ilm-methods)*, 2

getExprSummary, *5*
getExprSummary *(ilm-methods)*, 2
getExprSummary,ILM,character-method *(ilm-methods)*, 2
getExprSummary,ILM,missing-method *(ilm-methods)*, 2
getExprSummary,ILM-method *(ilm-methods)*, 2
getExprSummary-methods *(ilm-methods)*, 2
getIntens, *5*
getIntens *(ilm-methods)*, 2
getIntens,ILM,character-method *(ilm-methods)*, 2
getIntens,ILM,missing-method *(ilm-methods)*, 2
getIntens,ILM-method *(ilm-methods)*, 2
getIntens-methods *(ilm-methods)*, 2
getProbeConcs, *5*
getProbeConcs *(ilm-methods)*, 2
getProbeConcs,ILM,character-method *(ilm-methods)*, 2
getProbeConcs,ILM,missing-method *(ilm-methods)*, 2
getProbeConcs,ILM-method *(ilm-methods)*, 2
getProbeConcs-methods *(ilm-methods)*, 2
getSDSummary, *5*
getSDSummary *(ilm-methods)*, 2
getSDSummary,ILM,character-method *(ilm-methods)*, 2
getSDSummary,ILM,missing-method *(ilm-methods)*, 2
getSDSummary,ILM-method *(ilm-methods)*, 2
getSDSummary-methods *(ilm-methods)*, 2

ILM, *2*, *3*, *5–7*
ILM *(ILM-class)*, 1
ilm, *2*, 4
ILM-class, 1
ilm-methods, 2

par, *6*, *7*
plotILM, 6, *7*
plotILM,ILM,character-method *(plotILM)*, 6
plotILM,ILM,missing-method *(plotILM)*, 6
plotILM,ILM-method *(plotILM)*, 6
plotIntens, *6*, 7
plotIntens,ILM,character-method *(plotIntens)*, 7
plotIntens,ILM,missing-method *(plotIntens)*, 7
plotIntens,ILM-method *(plotIntens)*, 7

show *(ilm-methods)*, 2
show,ILM-method *(ilm-methods)*, 2