The LumiR User Guide

Renan Sauteraud*

June 10, 2013

Contents

*rsautera@fhcrc.org

1 Introduction

1.1 Technology

Luminex xMAP technology is a multiplex assay using flow cytometry to concurrently measure up to 500 analytes in a single reaction volume.

1.2 LumiR

`LumiR` is an R package that provides data structure and functions to read, store and analyze Luminex xMAP experiment.

As with any R package, it should first be loaded in the session

```
> library(LumiR)
```

2 Softwares

Even though Luminex is the company delivering the beads, a wide range of partners offer customizable kits and solutions for acquisition and/or analysis of the data. While the technology is identical, the output depends greatly on the software used for the acquisition. In its current version, `LumiR` can read the data from three different vendors:

2.1 Luminex

Luminex's own acquisition software xPONENT produces one csv file per well. These files contain the raw bead level information: bead id, fluorescence measured as well as the fluorescences used to map the bead. `LumiR` can read data from xPONENT versions 1.x and 3.x.

2.2 MiraiBio

MiraiBio also developped a set of tools for the acquisition and analysis of Luminex xMAP platform. The acquisition software 'MasterPlex CT' currently in version 1.2 creates one file per well. These binary files with a .lxb extension are based on the format FCS 3.0 and contain the bead level information. Along with these files comes a summary in xml format with a .lxd extension. It contains some information regarding the setup used for the experiment, the matching of the bead id with the analyte name and some basic calculations such as the MFI or the bead count in each well for each analyte.

2.3 BIO-RAD

Bio-Plex Manager Software yields files with a proprietary format. However, it allows the user to exports the results in XML files that can be processed by `LumiR`. The output is a single XML file per experiment gathering all informations available.

3 Requirements

In order to read the data into R, the package require the data to be organized in a specific file tree. Additional user-provided information is necessary to run most of the functions in this package.

3.1 Folders architecture

Regardless of which software produced the data, the required organization of the folder remains the same. The experiment should be located in a folder (root) with the raw bead level files for each plate in a subfolder named after the plate name or ID. The mapping files should be placed at the root.

3.2 Bead events extraction

Classic analysis methods based on the MFI do not use the bead level fluorescence intensities. Thus, depending on the software this information may not be saved by default. For a more detailed documentation on how to save and extract the bead level information, refer to the other vignettes of this package.

- raw_data_xPONENT
- raw_data_MasterPlex
- raw_data_BioPlex

3.3 Mapping files

There are 3 mapping files parsed when the data are read into R. These csv files contain information about the samples, the analytes and the layout of the experiment. There are three different way of generating these information:

- User submitted files that respect the expected format of each mapping file.
- Generating the files using `setup_template` and manually edit the information before reading.
- Skipping this step altogether. The reading function will extract as much information as possible from the data. The user will have to edit the R object afterward in order to access the most advanced functions of `LumiR`.

3.3.1 Analyte mapping file

In the raw data file, that are not usually exposed to the user, the beads are referenced using numbers or 'beadID' (or 'bid') instead of the actual analyte name. The number used for an analyte vary depending on the vendor and the kit used.

In order to display the actual analyte name, the package will look for a file 'analyte.csv' that will map each ID to a name. It should contain two columns **analyte** and **bid**. If this file is not submitted by the user, `LumiR` will still be able to read the data, but the beadID will be displayed instead of the actual analyte name. Data exported from BioPlex are the exception, as the .xml file also contains information regarding the analytes used. Alternately, when using MasterPlex CT, the reading function will look for a file with an extension '.lxd' and attempt to extract this information. (More information on this file and how to retrieve it is available in the MasterPlex vignette)

Finally, it is worth noting that in case of a user submitted file, only the analyte listed in the mapping file will be part of the R object. If more analytes are found in the data, they will be discarded and a warning will be thrown. This can be useful in a big multpilex experiment where only a handful of analytes are of interest to the user.

3.3.2 Phenotype mapping file

In the same fashion, `LumiR` will scan the root folder, looking for a file 'phenotype.csv'. This file is where information about the samples should be added. In its minimal form, the file contains four columns **plate**, **filename**, **well** and **sample_id**. These are the required information to match a file with a well (sample). The sample_id is a unique ID based on the other information used to tag a sample in the experiment. Additional, user provided information such as sample_name, treatment or ptid should be added as new columns.

Here again, if no file is given, this information will be added automatically, based on the folder structure and the filenames.

3.3.3 Layout mapping file

The last mapping file is used to get information about the design of the experiment. In a multiplate experiment, it is assumed that all plates have the same layout, therefore the file only have one line per well (96max).Three columns are required: the **well**, **sample_type** (standard/control/background/unknown) and the **concentration** The concentration is the expected concentration of the standards and should be set to 'NA' for all other wells, with the exception of the background wells that can be set to either 0 or 'NA'. Naturally, the information regarding the location and concentration is required for the standard curve fitting.

If no file is provided, each well will be defaulted to the type 'unknown' and have 'NA' concentration.

In R, this information is merged with the phenotype information.

4 setup_templates

The easiest way to create the three mapping files described above is to use the `setup_templates` function. It takes two arguments: a path and a list of the templates to create. The path should be set to the root of the experiment folder. The possible list of templates is 'analyte', 'phenotype' and 'layout'. By default, all templates are selected but if they exists a warning will be thrown and they will not be created.

An analyte and layout mapping file are provided as an example in the exdata folder of this package. The function can be used to create the missing 'phenotype.csv'.

```
> XP.path <- system.file("extdata", package = "LumiR")
> setup_templates(XP.path, templates = "phenotype")
```

As stated previously, this step can be skipped without affecting the reading. However, writing these templates provides an easy way to create mapping file that respect the required format and they can be easily modified prior to the reading.

5 blum

The first step is to read the data into R using the function `read.experiment`. Assuming the folder to read meets all the requirements, the function will automatically detect what software has been used for the acquisition. `read.experiment` returns an object of class `blum`. This object contains the bead level data and all the information extracted from the mapping files.

As an example, we provide two plates of the Listeria Project generated by Ofir Goldberger[1], Mark M. Davis[1], John-Demian Sauer[2] and Daniel A. Portnoy[3].

```
> XP.path <- system.file("extdata", package = "LumiR")
> out <- capture.output(bl <- read.experiment(XP.path))
> bl

An object of class blum with 27 analytes:
         CHEX.1 CHEX.2 CHEX.3 ... IL.7 IL.8 IP10
929587 measures of expression in 184 wells, on 2 plates.
And slots: phenoData featureData exprs
```

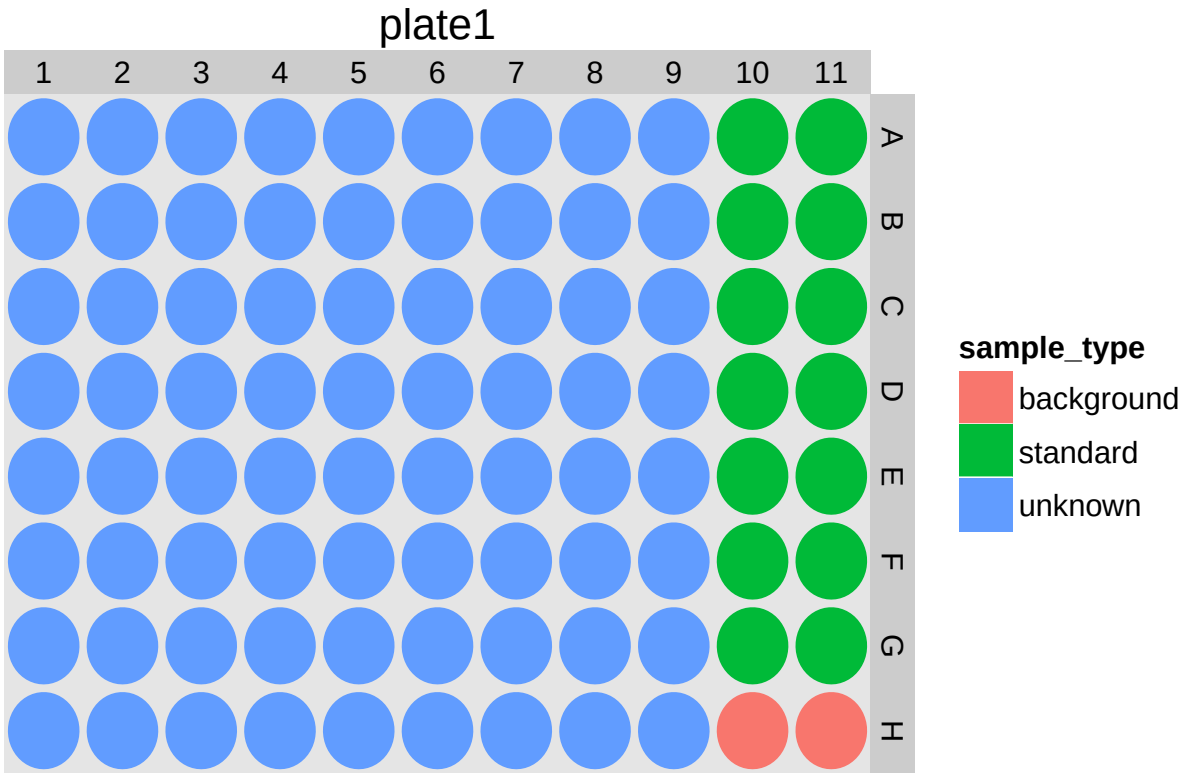
Naturally, `blum` objects have accessors to the phenotype information (`pData`), the analytes (`fData`) and the fluorescence measurements (`exprs`).

```
> head(pData(bl))

  well  plate   filename         sample_id sample_type concentration
1   A1 plate1 Run001.csv  Run001_plate1_A1     unknown            NA
2   A1 plate2 Run001.csv  Run001_plate2_A1     unknown            NA
3  A10 plate1 Run073.csv Run073_plate1_A10    standard          5000
4  A10 plate2 Run073.csv Run073_plate2_A10    standard          5000
5  A11 plate1 Run081.csv Run081_plate1_A11    standard          5000
6  A11 plate2 Run081.csv Run081_plate2_A11    standard          5000

> head(fData(bl))

  analyte bid
1  CHEX.1  97
2  CHEX.2  98
3  CHEX.3  99
4  CHEX.4 100
5 EOTAXIN  52
6   G.CSF  46
```

`LumiR` makes use of the `ggplot2` package to quickly visualize the design of an experiment. The `plot_layout` method can plot phenotype data for the selected plate.

```
> p <- plot_layout(bl, plate = "plate1", fill = "sample_type")

> print(p)
```

[1]Stanford
[2]University of Wisconsin-Madison
[3]UC Berkeley



Here, we use colors to plot the sample_type for each well of plate1.
This method uses `ggplot2`'s `geom_polygon` function, therefore, the same arguments are valid here.

6 slum

The second step is to summarize the data using `slummarize`. It returns an object of class `slum`. It is used for 'classic' standardization methods as it only contains the median fluorescence intensity (MFI) for each well and not the raw bead events. For this step, the package needs a user provided 'layout.csv' file with at least the location and expected concentration of the standards. The file provided in the extdata folder of this package is an example of such file.

```
> sl <- slummarize(bl)
```

In the expression slot, the bead events are replaced by the MFI for each analyte/sample. Additionaly, the concentration calculated from each MFI is stored in the object. For a quick view of the concentration of each analyte in each well:

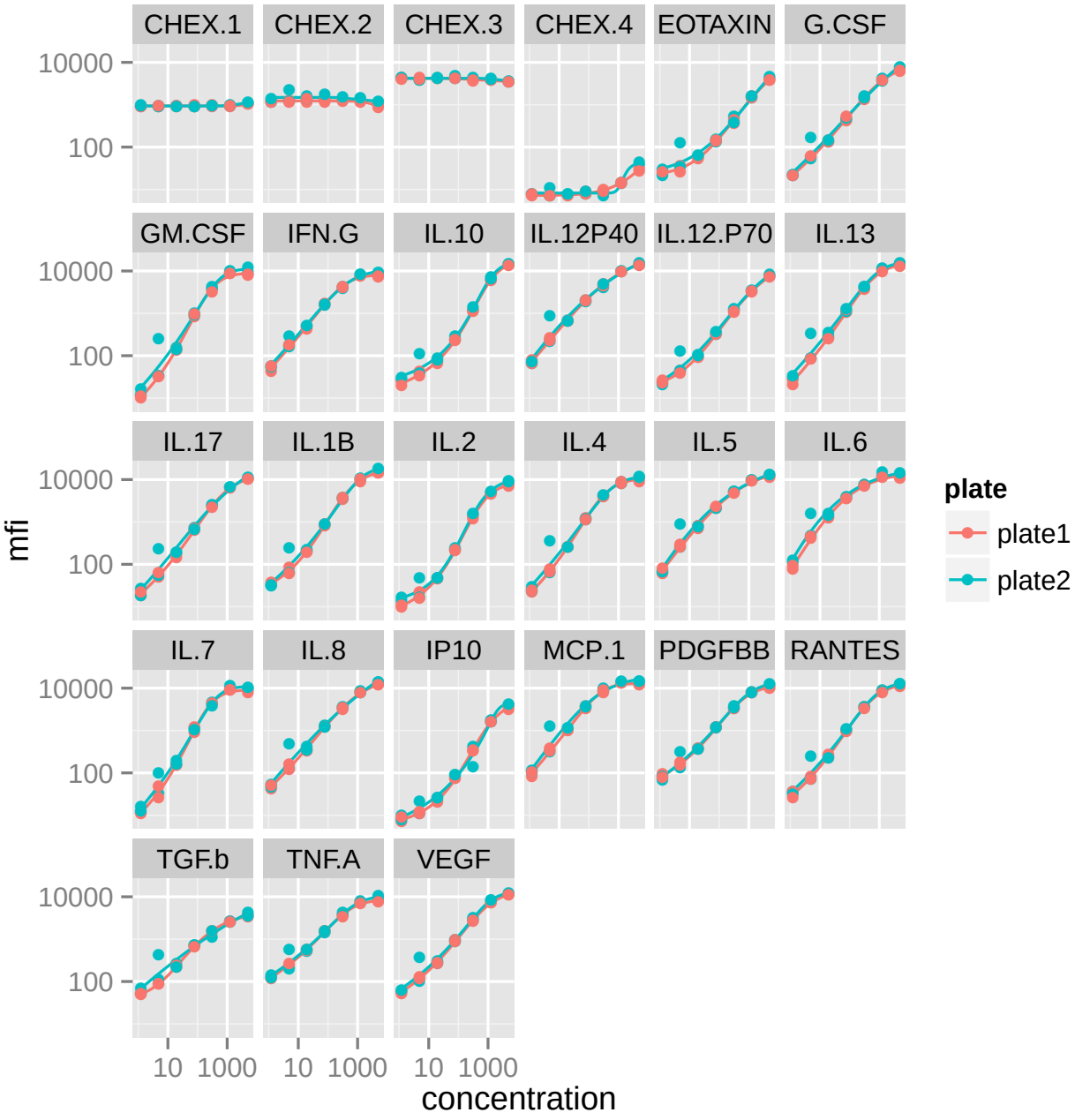
```
> concentration(sl)
```

Since `slum` objects retain the phenotype information from the mapping file, they may be used as first argument in `plot_layout`.

`slummarize` does the standard curve fitting using a 5-Parameters Logigistic curve by default. `LumiR` defines a new method to be used in `ggplot2` for standard curve visualisation.

In the following example, we melt the object into a big data.frame in order to display the measured FIs of the standards on the same plot using `ggplot2::geom_point`.

```
> library(ggplot2)
> msl <- melt(sl)
> msl.ss <- subset(msl, tolower(sample_type) == "standard")
> p <- ggplot(msl.ss, aes(color = plate), alpha = 0.5) + scale_x_log10() +
+     scale_y_log10() + facet_wrap(~analyte) + geom_sc(sl) + geom_point(aes(x = concentration,
+     y = mfi))

> print(p)
```



Here again, any argument will be passed to the underlying `ggplot2`'s method `geom_line`.

7 output

An slum object can be written to a file following ImmPort MBAA (Multiplex Bead Array Assays) format. First, the name of the organisation producing the data should be added using the `set_center` function.

```
> sl <- set_center(sl, "FHCRC")
```

Then, `writeMBAA` can be used to write a csv file that can be opened using Excel and contain the fields required by ImmPort.

```
> writeMBAA(sl, outfile = "./MBAA.results.csv")
```