



## How do I choose between using *affy* or *oligo* package for preprocessing microarrays?



Hello,

0



I'm looking for an **automated** way to choose between preprocessing microarray datasets with the *affy* or *oligo* package. I'm developing a pipeline that automates the acquisition and processing of data from ArrayExpress, and neither package is one-size-fits all. My main question is thus:

- **Is there a way to *automatically* determine which package is preferable, either from the name of a platform (e.g. "[HuGene-1\_1-st] Affymetrix Human Gene 1.1 ST Array", "Affymetrix GeneChip Human Genome U133 Plus 2.0 [HG-U133\_Plus\_2]"), or from the header of a CEL file?**

I will also include some answers that I've found which may be helpful if you're arriving from Google:

- Should I use *oligo* or *affy*?
  - Try *oligo* first, then if it doesn't work, try *affy*.\*
  - *oligo* works for newer platforms and the popular old platforms. *affy* won't work for new platforms such as the Gene ST and Exon ST arrays.
  - Some datasets cause an error in *oligo* but still work with *affy*; I think this has to do (sometimes? always?) with custom CDFs in the dataset.
- What differences are there between the two, if both of them work?
  - The expression matrices produced by each are almost identical. \*\*
  - *oligo*'s `read.celfiles()` uses 33% less memory than *affy*'s `read.affybatch()`.\*\*\* Since this step is the most memory-demanding of a microarray analysis, and a big dataset can easily suck up tens of gigabytes of memory and reduce your computer to a thrashing mess, this can be significant.
  - `affy::rma()` is often 10% - 50% quicker than `oligo::rma()`

If anyone has any other reasons to choose one over the other, please do let me know.

Footnotes:

\* They're quite easy to change between. For a vector of `rawfilepaths`:

- `rawbatch = read.celfiles(rawfilepaths) ; RMA = oligo::rma(rawbatch)`
- `rawbatch = read.affybatch(rawfilepaths) ; RMA = affy::rma(rawbatch)`

\*\* For older chips, the expression matrices produced are virtually identical (the differences are just rounding error or similar). The only real difference I found was for Human Gene 1.0 ST, in which *oligo* produced an expression table with 33,297 rows, vs. 32,321 rows from *affy*.

\*\*\* I tested files from E-MTAB-1724 and found the relationships between number of files and peak memory usage (in GiB) to be:

For read.affybatch():  $0.0263 * \text{length}(\text{rawfilepaths}) + 0.102$   
 For read.celfiles():  $0.0176 * \text{length}(\text{rawfilepaths}) + 0.139$   
 ( $R^2 > 0.999$  for both)

Another difference, which won't affect most people, is that the output object from *oligo*'s read.celfiles() is far larger than that from *affy*'s read.affybatch(). However, it's so much smaller (tens to hundreds of megabytes) than the peak memory usage (gigabytes to tens of gigabytes) that you shouldn't worry about it unless you're collecting a lot of these.

microarray affymetrix microarrays affy oligo • 1.8k views

[ADD COMMENT](#) • [link](#) 3.8 years ago [stevesheridan](#) • 0 • updated 3.8 years ago [James W. MacDonald](#) ⚡ 55k



0



These days I can't think of a compelling reason that you would need to use affy in lieu of oligo. Benilton added a generic array class, maybe two release cycles ago, that allows one to use non-standard CDFs like the MBNI re-mapped CDFs that you mention. And if you go to MBNI's download site, you can see that they have a pdInfoPackage for (AFAIK) all of the CDFs that they have generated.

The only reason I can think of for using affy is if you want to use one of the packages (like frma) that rely on affy rather than oligo. This is particularly true for any of the random-primer based arrays, which share probes across multiple probesets, and which the affy package does not handle correctly.

But if you have a particular array for which affy works, but oligo does not, please let me know.

[ADD COMMENT](#) • [link](#)

3.8 years ago [James W. MacDonald](#) ⚡ 55k

➔ [Login](#) before adding your answer.

## Similar Posts

### Memory error building index •

6.1 years ago [hmabril](#) • 0 • updated 6.1 years ago [Martin Morgan](#) ⚡ 25k

<p>I get the following error when running the following command in r:</p> <pre> buildindex(basename=&quot;my\_index&quot;; reference=&qu...

### 64-bit CPUs and bioconductor •

17.2 years ago [Shinichiro Wachi](#) • 50 • updated 17.2 years ago [David Henderson](#) • 20

<div class='preformatted'>Hello, Now that both PC and Mac have 64-bit CPU capable of addressing several gigabytes of RAM, will bioconducto...

### 64-bit CPUs and bioconductor •

17.2 years ago [Shinichiro Wachi](#) • 50 • updated 17.2 years ago [David Henderson](#) • 20

<div class='preformatted'>Hello, Now that both PC and Mac have 64-bit CPU capable of addressing several gigabytes of RAM, will bioconducto...

### FW: PLANdbAffy + Alternative Exon Annotation +XPS, aroma, oligo, RMAExpress •

10.0 years ago [Ramil Nurtdinov](#) • 10 • updated 9.9 years ago [brislav misovic](#) • 120

<div class='preformatted'>Dear colleagues My experience with R BioConductor and Affymetrix Human Exon 1.0 ST array started from oligo pack...

## Content

[Search](#)  
[Users](#)  
[Tags](#)  
[Badges](#)

## Help

[About](#)  
[FAQ](#)

## Access

[RSS](#)  
[API](#)  
[Stats](#)

Use of this site constitutes acceptance of our [User Agreement](#) and [Privacy Policy](#).

Powered by the  version 2.3.3