Vignette for the *CellMapper* R Package

Brad Nelms, Levi Waldron, and Curtis Huttenhower

October 17, 2016

Contents

1 Overview

Cell type-specific gene expression plays a defining role in cellular function and differentiation. There are many situations where identifying which genes are most strongly expressed in one cell type relative to others can provide important biological insights. *CellMapper* infers which genes are strongly expressed in one cell type relative to others by identifying genes that share an expression profile similar to an established set of cell type-specific markers, referred to here as 'query genes'. CellMapper can incorporate information from heterogeneous samples that contain mixtures of many different cell types, and can provide accurate predictions even when a cell type has not been isolated [**?**].

This vignette provides an introduction to gene-driven analysis using the *CellMapper* R package and the accompany data package, *CellMapperData*. It contains guidelines on how to select all inputs to the algorithm (such as query genes and gene expression data), directions for running CellMapper using microarray data from the *CellMapperData* package, and advanced examples where custom microarray datasets are used.

2 Getting Started

To use the *CellMapper* package, *R* and *Bioconductor* must first be installed. A command line version of *R* can be downloaded from www.r-project.org, or for those more comfortable working with a graphical user interface environment, there are several options; we recommend *RStudio* (www.rstudio.com). Both *R* and RStudio are available for Windows, Mac, and Linux, and documentation to help with installation can be found on the websites. After *R* is installed, the *Bioconductor* base package can be installed as described at bioconductor.org/install/.

Next, the CellMapper package can be downloaded and installed by running the following code from within R:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("CellMapper")
```

We also recommend installing the companion data package, *CellMapperData*, which contains pre-processed microarray data from several large gene expression datasets. The data in this package is derived from 18,932 human and 1,332 mouse microarray experiments performed on a wide range of samples, including whole organs, purified cell populations, cell lines, and 900 distinct sub-regions of the adult human brain. *CellMapper* can be used to search these datasets individually or in aggregate, depending on which tissue and/or species is most relevant to the cell type(s) of interest. More details about this data can be found in the package documentation. The data package is available from *Bioconductor*'s *ExperimentHub*, and can be accessed as follows:

```
> library("ExperimentHub")
> hub <- ExperimentHub()
> x <- query(hub, "CellMapperData")
> x
```

```
ExperimentHub with 6 records
# snapshotDate(): 2016-08-08
# $dataprovider: GEO, ArrayExpress, Allen Brain Atlas
# $species: Homo sapiens, Mus musculus
# $rdataclass: CellMapperList
# additional mcols(): taxonomyid, genome, description, coordinate_1_based,
#   maintainer, rdatadateadded, preparerclass, tags, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["EH170"]]'

          title
  EH170 | Pre-processed microarray data from the Allen Brain Atlas
  EH171 | Pre-processed microarray data from the Affymetrix HG-U133PlusV2 platform
  EH172 | Pre-processed microarray data from the Affymetrix HG-U133A platform
  EH173 | Pre-processed microarray data from the Affymetrix MG-U74Av2 platform
  EH174 | Pre-processed microarray data from the human small and large intestine
  EH175 | Pre-processed microarray data from the human kidney
```

After installing and loading *ExperimentHub*, the individual CellMapperData datasets can be loaded using their *ExperimentHub* accession numbers. For instance, the Allen Brain Atlas dataset is stored under accession number 'EH170' and can be downloaded using the following code:

```
> BrainAtlas <- hub[["EH170"]]
> BrainAtlas
```

```
An object of class "CellMapperList"
#  Provide as input to the 'CMsearch' function of the 'CellMapper' package
#  Derived from an expression dataset with 20787 genes and 3702 samples
#     Dataset source:  'Allan Brain Atlas'
#  The type of gene ID used is: 'Human Entrez IDs'
#     Example gene IDs:  '733', '735', '740', '741', '744', '745', ...
```

For general information about using *R*, the *R* and *Bioconductor* websites contain extensive documentation, and many helpful tutorials can be found online.

3 Planning a CellMapper Analysis

CellMapper requires, at a minimum, two inputs: (i) a set of gene expression data and (ii) one or more 'query genes' specific to the cell type of interest. In most cases, the pre-processed microarray data from the *CellMapperData* package will be sufficient, and so the main decision involves choosing the query gene(s).

3.1 Gene Expression Data

CellMapper performs best with large sample sizes in excess of 500 arrays, and so it is generally advisable to use large datasets that combine expression data from multiple sources. The *CellMapperData* package contains data from three large microarray meta-analyses that span a range of human and mouse samples, and one dataset from the Allen Brain Atlas focused on the adult human brain. For most non-brain cell types, we recommend searching the three large meta-analyses as described in **??**, while for brain cell types we recommend searching the Allen Brain Atlas dataset as described in **??**. *CellMapper* can also be applied to custom microarray datasets as described in **??**, or to data from a specific tissue or organ as described in **??**.

3.2 Query Genes

A set of cell type-specific marker genes, called 'query genes', are used to define a reference expression profile for a given cell type. Then other genes that display a similar expression profile to the query genes are identified, as these genes are likely to be expressed in the same cell type. *CellMapper* was designed to accurately predict cell type-specific expression with as little as a single query gene, and is compatible with standard cell type-specific markers used in a variety of experimental techniques, including immunohistochemistry, flow cytometry, and promoter-directed (Cre-lox) conditional knockout mice.

The most important criterion when choosing a query gene is that the gene is expressed only in the cell type of interest. It is important to consider potential expression in any other cell type that might be present in the samples contained within a dataset. For example, when searching the Allen Brain Atlas data, which contains exclusively samples from the brain, a good query gene must not be expressed in any other cell type present in the brain. When searching any of the three meta-analyses datasets from the *CellMapperData* package, which contain samples from essentially every organ, a good query gene must not be expressed in any other mammalian cell type.

If multiple suitable marker genes are available for a cell type, *CellMapper* can be applied with >1 query gene. However, it is better to use a single carefully chosen query gene than to include additional query genes of questionable quality.

To identify potential query genes, a great place to start is by searching the JAX mice database (http://cre.jax.org/) to see if there are any conditional knockout mice strains available for the cell type of interest. This database lists many mice strains where the Cre-recombinase gene is expressed using a cell type-specific promoter. Because these strains express the Cre transgene in every cell type where the chosen promoter is active, most conditional knockout mice have been thoroughly evaluated to characterize which cell types the promoter driving Cre expression is active in. If there are no conditional knockout mice available for a cell type, many antibody suppliers provide detailed lists of antibodies that target genes expressed specifically in different cell types.

3.3 Other Parameters (optional)

CellMapper applies an SVD-based filter as a pre-processing step to highlight biologically important signals in the microarray data. The *CellMapper* R package allows the strength of this pre-processing filter to be tuned using two parameters: an alpha parameter, which ranges from 1 (no filter) to 0 (strong filter), and a query-driven weight parameter, which can either be set to TRUE (query-driven weight filter is ON) or FALSE (query-driven weight filter is OFF). These parameters are discussed in more detail in the CellMapper manuscript and the package documentation. In general, both parameters can be safely left at their default values (alpha $= 0.5$ and query-driven weight set to TRUE).

4 Running CellMapper

4.1 With a Single Microarray Dataset

To run a CellMapper analysis, the first step is to start R and load the *CellMapper* package:

```
> library(CellMapper)
```

As an initial example, we will demonstrate a CellMapper analysis to find genes expressed in GABAergic neurons using the query gene glutamate decarboxylase 1 (GAD1). GAD1 is an enzyme that catalyzes the final step in the biosynthesis of gamma-aminobutyric acid (GABA, the neurotransmitter that defines the GABAergic class of neurons, and is widely used as a specific marker for this neuron lineage.

For this example, we will use a large microarray dataset from the Allen Brain Atlas [**?**]. This dataset is available from the *CellMapperData* on *ExperimentHub*:

```
> library(ExperimentHub)
> hub <- ExperimentHub()
> query(hub, "CellMapperData")
```

```
ExperimentHub with 6 records
# snapshotDate(): 2016-08-08
# $dataprovider: GEO, ArrayExpress, Allen Brain Atlas
# $species: Homo sapiens, Mus musculus
# $rdataclass: CellMapperList
# additional mcols(): taxonomyid, genome, description, coordinate_1_based,
#   maintainer, rdatadateadded, preparerclass, tags, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["EH170"]]'

         title
  EH170 | Pre-processed microarray data from the Allen Brain Atlas
  EH171 | Pre-processed microarray data from the Affymetrix HG-U133PlusV2 platform
  EH172 | Pre-processed microarray data from the Affymetrix HG-U133A platform
  EH173 | Pre-processed microarray data from the Affymetrix MG-U74Av2 platform
  EH174 | Pre-processed microarray data from the human small and large intestine
  EH175 | Pre-processed microarray data from the human kidney
```

As can be seen above, the Allen Brain Atlas data is stored under accession 'EH170'. This dataset can be loaded using the following command:

```
> BrainAtlas <- hub[["EH170"]]
```

Next, we will set our query and control genes. The datasets from *CellMapperData* all use human Entrez IDs, and so we need to define all genes using Entrez identifiers. First, define a variable containing our choice of query gene, GAD1 (Entrez ID = 2571):

```
> query <- "2571"
```

Then run the analysis using the CMsearch function. CMsearch provides the core functionality of the CellMapper package, and can be run to predict GABAergic genes using our selected dataset and query gene:

```
> GABAergic <- CMsearch(BrainAtlas, query.genes = query)
```

CMsearch returns a matrix, in rank order, containing the false discovery rate (FDR) that each gene is co-expressed with the cell type-specific query gene. To view the top results for this analysis within R, use the head function:

```
> head(GABAergic)
```

```
       Gene           FDR
1      2572 1.294645e-102
2    140679  4.534742e-61
3      5368  3.064491e-18
4 100130256  5.136594e-18
5    137970  4.537342e-12
6      5121  7.151401e-12
```

These results can be saved as a .csv file and loaded into excel if desired:

```
> write.csv(GABAergic, file = "CellMapper analysis for GABAergic neurons.csv")
```

4.2 With Multiple Microarray Datasets

In this example, we will search for genes expressed in simple epithelial cells using the epithelial marker gene KRT8. Simple epithelia can be found in many different organs, and so we will pool results from the three meta-analysis datasets contained within the *CellMapperData* package. First, load the three meta-analysis datasets from *CellMapperData*:

```
> Engreitz <- hub[["EH171"]]
> Lukk <- hub[["EH172"]]
> ZhengBradley <- hub[["EH173"]]
```

The dataset from [**?**] contains data from mouse microarray samples. ZhengBradley is a pre-processed version where the mouse Entrez IDs have been replaced with Entrez IDs for their human orthologs. This allows human Entrez IDs to be used for every search and there is no need to worry about mapping orthologs between species.

Next, we will run CMsearch using the query gene to KRT8 (Entrez ID 3856):

```
> query2 <- "3856"
> SimpleEpithelia <- CMsearch(list(Engreitz, Lukk, ZhengBradley), query.genes =
+ query2)
> head(SimpleEpithelia)
```

```
  Gene          FDR
1 3875 2.082440e-76
2 3880 3.122666e-36
3 1366 1.969737e-21
4 1365 3.950937e-21
5 1999 6.546830e-19
6 4072 1.913438e-18
```

The output from the multiple dataset analysis is identical to the output from individual CellMapper searches, and contains matrix, in rank order, containing the Entrez ID and false discovery rate (FDR) for each gene. These results can be viewed from within R using the head function, or saved as a .csv file to import into a spreadsheet software such as Excel, as described at the end of **??**.

5 Advanced Examples

5.1 Using Custom Microarray Data

CellMapper can also be run using custom microarray data. In this case, the data must first be pre-processed with the CMprep function. CMprep accepts microarray data in either of two formats: a matrix of expression data with genes as rows and samples as columns, or a *Bioconductor* ExpressionSet object. Most microarray normalization packages in *Bioconductor* return the results as an ExpressionSet object, and this output can be passed directly to the CMprep function.

For this example, we will use a dataset of human leukemia samples from the *ALL* data package. This dataset was selected solely for illustrative purposes, and is not necessarily an ideal choice for CellMapper. To load this dataset, install the ALL package and run the following lines of code:

```
> library(ALL)
> data(ALL)
```

Then pre-process the ALL expression dataset with the CMprep function:

```
> prepped.data <- CMprep(ALL)
```

For the large datasets ($>$ 1000 arrays), the CMprep function can take over an hour to run on a personal laptop. All inputs to the CellMapper algorithm, including the choice of query genes and algorithm parameters, are selected after the pre-processing step is completed. Thus, the pre-processing step only needs to be performed once for any new dataset, and the results can then be saved and accessed at a later time for each CellMapper search:

```
> save(prepped.data, file = "Custom Data for CellMapper.Rdata")
```

This saved data file can be later loaded into R with the load function:

```
> load("Custom Data for CellMapper.Rdata")
```

After processing the data, CellMapper can be applied using the CMsearch function as described in **??**. It is important to select query and control genes using the same identifiers as provided for the original microarray dataset. The ALL dataset uses Affymetrix probeset IDs rather than Entrez gene IDs, and so all query genes should be selected using their

corresponding Affymetrix probeset IDs. For example, CellMapper can be used to search the ALL pre-processed data using the probeset "1000_at" as a query gene:

```
> out <- CMsearch(prepped.data, query.genes = "1000_at")
> head(out)
```

```
        Gene         FDR
1   36129_at 0.006478953
2   38813_at 0.052214771
3   37307_at 0.060395306
4   33388_at 0.070458714
5   35653_at 0.077434668
6 36546_r_at 0.117601147
```

Alternatively, probesets can be mapped to genes before running the CMprep function, and then all CellMapper searches can be performed using gene identifiers.

5.2 Accessing Sample Metadata to Restrict Search to a Single Organ or Tissue

Some cell types only have markers available that are specific within one organ, but are not specific organism-wide. In such cases, *CellMapper* can still be applied provided the search is restricted to data from the organ or tissue where the markers are specific. For brain cell types, the search can be restricted simply by selecting the Allen Brain Atlas dataset as described in **??**. For non-brain cell types, however, a custom microarray dataset must be used that is restricted to the organ where markers are available.

In this example, we illustrate how to restrict CellMapper to a specific organ by accessing sample metadata. We will search for genes specifically expressed in enteroendocrine cells (EECs) using an intestine-specific subset of the microarray datasets from [**?**] and [**?**]. EECs are a rare intestinal cell type, comprising $<1\%$ of the gut epithelium. Chromogranin A (CHGA) is the most established genetic marker of EECs within the intestine, but it is also expressed by neurons and other endocrine cell types in other tissues and so does not make a good marker for EECs organism-wide. We will overcome this problem by selecting only intestine microarray datasets to analyze.

The original data from [**?**] and [**?**] are available from ArrayExpress (E-MTAB-62) and GEO (GSE64985), respectively. For convenience, we have also deposited these datasets to *ExperimentHub* in the package *HumanAffyData*:

```
> query(hub, "HumanAffyData")
```

```
ExperimentHub with 2 records
# snapshotDate(): 2016-08-08
# $dataprovider: ArrayExpress, GEO
# $species: Homo sapiens
# $rdataclass: ExpressionSet
# additional mcols(): taxonomyid, genome, description, coordinate_1_based,
#   maintainer, rdatadateadded, preparerclass, tags, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["EH176"]]'

          title
  EH176 | GEO accession data GSE64985 as an ExpressionSet
  EH177 | ArrayExpress accession data E-MTAB-62 as an ExpressionSet
```

First, load the data from E-MTAB-62 (*ExperimentHub* accession 'EH177'):

```
> E.MTAB.62 <- hub[["EH177"]]
> E.MTAB.62
```

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 12496 features, 5372 samples
  element names: exprs
```

```
protocolData: none
phenoData
  sampleNames: GSM23227.CEL 1229968152.CEL ... 676426699.CEL (5372 total)
  varLabels: OperatorVariation DataSource ... ArrayDataFile (16 total)
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu133a
```

This dataset is stored as a Bioconductor ExpressionSet object, and includes both expression data and phenotype data. Lukk, et al. (2010) manually curated all samples in the dataset, and provided extensive phenotypic information about each sample. This phenotype data can be used to select samples from a specific organ, and can be accessed with the pData function:

```
> pDat <- pData(E.MTAB.62)
```

We are interested in the column of this phenotype data labeled 'OrganismPart', because this column contains information about the organ or tissue of origin for each sample. For example, the first 30 categories in the 'OrganismPart' column are:

```
> unique(pDat$OrganismPart)[1:30]
```

```
 [1] kidney                                mammary gland
 [3] lung
 [5] bone marrow                           blood
 [7] skeletal muscle                       adipose tissue
 [9] adipose tissue from abdomen           adipose tissue from abdomen and thigh
[11] Trachea                               Abdominus
[13] amygdala                              Leukocyte
[15] Blood                                 cerebellum
[17] bladder                               bladder mucosa
[19] bone                                  brain
[21] bronchial epithelia, current smoker   bronchial epithelia
[23] bronchial epithelia, former smoker    lung transplant
[25] muscle                                colon
[27] oral                                  brain caudate nucleus
[29] caudate nucleus                       Cord blood
131 Levels:  Abdominus adipose tissue ... yolk sac
```

After examining all 131 categories in this column, we find three that are relevant to enteroendocrine cells: 'colon', 'colon mucosa', and 'small intestine'. To select the subset of the Lukk dataset associated with these sample categories, first create a variable that identifies these samples:

```
> samples <- which(pDat$OrganismPart %in% c("colon", "colon mucosa",
+ "Small intestine"))
```

Then access the expression data for these samples using the Biobase exprs function:

```
> Lukk_unprocessed.gut <- exprs(E.MTAB.62)[, samples]
```

The new variable, Lukk_unprocessed.gut, now contains a matrix of expression values with Entrez IDs as rows and 130 intestine-specific samples as columns. To pre-process this data for CellMapper, use the CMprep function as described in **??**:

```
> Lukk.gut <- CMprep(Lukk_unprocessed.gut)
```

Next, we will prepare an intestine-specific subset of the GSE64985 dataset. Load the GSE64985 data (*ExperimentHub* accession 'EH176'), and create a variable with the phenotypic data using the same approach described above for the E-MTAB-62 dataset:

```
> GSE64985 <- hub[["EH176"]]
> pDat <- pData(GSE64985)
```

For the Engreitz dataset, the phenotypic data contains two columns containing the 'title' and 'description' of each sample entry from GEO. To identify intestinal samples from this dataset, we will use a text mining approach to find key words in the title or description associated with each sample entry. The key words 'colon' and 'intestin' successfully distinguish most intestinal samples:

```
> keywords <- c("colon", "intestin")
> select = grepl(paste(keywords, collapse = "|"),
+         pDat$title, ignore.case = TRUE) |
+         grepl(paste(keywords, collapse = "|"),
+         pDat$description, ignore.case = TRUE)
```

This code will treat the keywords as substrings without regard to case. For example, the keyword 'intestin' will return results that contain 'intestine', 'INTESTINE', 'intestinal', etc. This returns 582 samples containing the selected key words. To extract this subset of the data, use the exprs function:

```
> Engreitz_unprocessed.gut = exprs(GSE64985)[,select]
```

Then pre-process the data:

```
> Engreitz.gut = CMprep(Engreitz_unprocessed.gut)
```

Finally, we will set our query gene to CHGA (Entrez ID 1113) and run CMsearch using the pre-processed intestine-specific datasets using the same approach as described in **??**:

```
> query = "1113"
> EECs <- CMsearch(list(Lukk = Lukk.gut, Engreitz = Engreitz.gut), query.genes =
+         query)
> head(EECs)
```

```
   Gene        FDR
1  2641 0.01798850
2  2981 0.03023993
3 11148 0.05040129
4   766 0.05040129
5  8671 0.05040129
6 63928 0.05040129
```