The MIMOSA Package

Greg Finak <gfinak@fhcrc.org>

May 2, 2014

1 Background on ICS Assays

In a vaccine trial setting, intracellular cytokine staining assays measure whether individuals respond to a vaccine. They do so by measuring how antigen-specific T–cells respond to antigen stimulation. Antigen-specific T–cells that respond to antigen express one or multiple cytokines (i.e., IFNg, IL2, TNFa, IL4). The strength of response is measured as the fraction of antigen-specific T-cells expressing the cytokine or cytokine combination, usually relative to an unstimulated control sample.

2 The MIMOSA Package

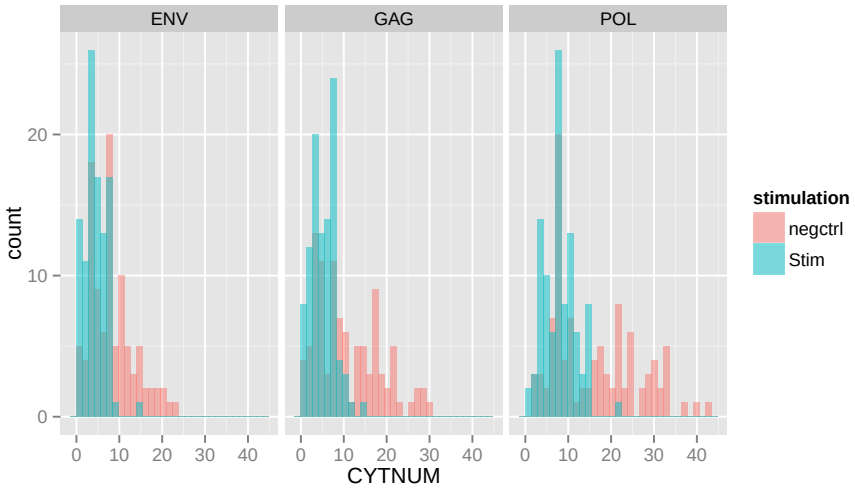
The MIMOSA package provides a framework for analysing ICS assay data (as well as other types of single-cell assays that generate paired count data). Given a cytokine, and counts of cytokine positive and negative cells for stimulated and control samples from multiple individuals, the MIMOSA package will fit a two–component Beta–Binomial or Dirichlet-Multinomial mixture model to the data. Importantly, input to the model are *paired* observations of *stimulated* and *unstimulated* samples from the same experimental unit (i.e., individual or subject). In this sense, the model is unlike the other mixture of Beta-Binomials or Dirichelt-Multinomials models available through BioConductor or CRAN.

2.1 The MIMOSA Model

MIMOSA fits a two-component mixture model. The two components represent two competing hypotheses for the data. Under the null hypothesis of no response, the proportion of cytokine expressing cells in the stimulated sample is equal to the proportion of cytokine expressing cells in the unstimulated sample. The first component models this non-response. Under the alternative hypothesis of response, the proportion of cytokine positive cells in the stimulated sample is greater than the proportion of cytokine positive cells in the control sample (MIMOSA also supports the two-sided hypothesis of non-equality). The second component models this response. Under the null hypothesis, both stimulated

and unstimlated sample counts would be generated by the non-response component, under the alternative hypothesis, the unstimulated sample counts are generated by the non-response component, while the stimualted sample counts are generated by the response component. Since each individual assayed by ICS can be either a *responder* or *non–responder*, the whole data set is a mixture of responding and non–responding individuals.

Cell counts in ICS assays are typically small, as are the proportions. MIMOSA uses a Bayesian modelling approach to borrow strength across individuals in order to improve the sensitivity of the model in identifying responders and non–responders.



3 Preparing the Data

MIMOSA fits a model to count data (for example, cell counts extracted from flow cytometry data). Data is generally in tabular form and has been extracted from processed and gated flow cytometry (FCS) files. *MIMOSA* is bundled with some simulated intracellular cytokine staining (ICS) data with three antigens. Magnitude of response is constant, as is the response rate, but the number of cells collected varies for the three antigens (ENV>GAG>POL). There are 100 subjects total (25 responders and 25 non-responders) in the simulated data sets.

We load the data and have a look:

```
require(MIMOSA)
head(ICS)

##    NSUB CYTNUM ANTIGEN CYTOKINE TCELLSUBSET UID  Ntot Stim
## 1 79992      8     ENV     IFNg         CD4   1 80000  POL
## 2 79992      8     ENV     IFNg         CD4   2 80000  POL
## 3 79992      8     ENV     IFNg         CD4   3 80000  POL
```

```
## 4 79991      9     ENV     IFNg          CD4   4 80000  POL
## 5 79992      8     ENV     IFNg          CD4   5 80000  POL
## 6 79994      6     ENV     IFNg          CD4   6 80000  POL
```

The data contains six columns: CYTNUM and NSUB which are the measured cytokine-positive and negative cell counts. CYTOKINE is the measured cytokine for that row, TCELLSUBSET is the subset of T–cellls measured in that row, ANTIGEN is the antigen stimulation used for that observation, and UID is an experimental unit identifier (identifying a unique subject). We see we have 100 subjects within each combination of T-cell subset, antigen, and cytokine.

While MIMOSA doesn't expect any fixed column names, it does expect the data to be in the above format (column names can vary). The rest of the information is passed in to the constructor which will shape the data into a form MIMOSA can work with.

We have taken advantage of the the ExpressionSet object in Bioconductor to wrap up the data and facilitate extraction of different subsets of the data for model fitting. So, the first step to fitting a MIMOSA model is to wrap up the data in an ExpressionSet object. We provide the ConstructMIMOSAExpressionSet() method to facilitate this. We'll construct such an object below, then we'll explain the details.

```
E <- ConstructMIMOSAExpressionSet(ICS, reference = ANTIGEN %in% "negctrl", measure.columns =
    "NSUB"), other.annotations = c("CYTOKINE", "TCELLSUBSET", "ANTIGEN", "UID",
    "Ntot"), default.cast.formula = component ~ UID + ANTIGEN + CYTOKINE + TCELLSUBSET +
    Ntot, .variables = .(TCELLSUBSET, CYTOKINE, UID, Ntot), featureCols = 1,
    ref.append.replace = "_REF")
```

3.1 What do these arguments mean?

Most importantly, we need to tell the model which variables hold our positive and negative cell counts, which variables uniquely identify an observation, and which observations represent the controls (untreated) samples for each subject.

reference= is an *R expression* that evaluates to a logical vector which is TRUE for each element of the data representing an unstimulated or untreated sample. In the case above, the expression ANTIGEN%in%"negctrl" indicates that all samples labelled *negctrl1* in the antigen column will be treated as our reference or baseline for the test.

variables= is a *dotted pairlist* (which is just a list of variable names in brackets preceded by a period), which identifies the variables that should be used to group observations into unique, subject-specific groups that go together. Above, we have UID (the subject id), TCELLSUBSET because we measure multiple T-cell subsets per subject, and CYTOKINE since we measure multiple cytokines per subject as well. Notice that we DO NOT include ANTIGEN here. That is

because, within each group defined above, we use the SAME negative control ANTIGEN stimulation.

measue.columns= identifies which columns in the data hold our positive and negative cell counts.

other.annotations= is a vector of column names that you want to include in the resulting expression set pheno data.

default.cast.formula= is a *formula* that should always begin with *component* on the left-hand-side. The default should be fine in most cases. This formula describes how to reshape the data frame before it is encapsulated in an ExpressionSet. Again, the default should be fine for most uses.

featureCols= identifies which columns in the reshaped data frame hold the feature information. Features, in our case, are the positive and negative cell counts. If you use the default cast formula, then the default value of featureCols is fine for you.

ref.append.replace= Is a substring used internally to distinguish between stimulated and unstimulated samples while constructing the expression set. You shouldn't need to worry about this unless your data already has positive and negative cell counts from stimulated and unstimulated samples on the same row. In that case, they should share a common name for the positive and negative parts, and a suffix to distinguish the stimulated and unstimulated counts. *i.e., NSUB,CYTNUM,NSUB_REF,CYTNUM_REF*, where ref.append.replace="_REF".

4 Fitting the Model

While the above is a little more complex than usual, in one call we get our ExpressionSet object with samples on the colums and features on the rows. Features are the positive and negative cell counts. Samples are defined by the combination of different T-cell subsets, antigen stimulations, cytokines measured, and subjects. Here we can take advantage of the formula interface to fit our model.

```
set.seed(100)
result<-MIMOSA(NSUB+CYTNUM~UID+TCELLSUBSET+CYTOKINE+Ntot|ANTIGEN,
               data=E, method="EM")
```

There's two things to note about the above function call. The formula NSUB+CYTNUM UID+TCELLSUBSET+CYTOKINE—ANTIGEN will fit a separate model for each ANTIGEN to the cell counts of all subjects. CYTOKINE and TCELLSUBSET are not included in the conditioning because there is only one level of each of these factors. The left hand side contains NSUB+CYTNUM, which are our negative and positive cell counts. The model expects, specifically, that the first component represents the negative cell counts, so NSUB (or whatever you call your negative cell counts) should always come first. The ref and subset arguments are expressions that evaluate to a logical

vector and specify which subset of observations represent the referece for the contrast, and which represent the treatment. These have been automatically recoded as "Reference" and "Treatment" by the previous call that created the ExpressionSet, such that the reference are the "negctrl1" samples, and the treatment is any other antigen stimulation. Together, the ref and subset arguments allow you to fit the model to whichever subset of data you may be interested in analyzing.

Finally, the *method* argument can be either "EM" (which is faster) or "MCMC", which is slower, but more stable. The call takes additional arguments that are passed on to the model fitting function.

4.1 Results of IFNg Fit

Model fitting returns an list of "MIMOSAResult" objects (MIMOSAResultList), each of which contains a slot for the Z's, W's and a "result" slot whose content is either an MDMixResult or MCMCResult object, depending on the method used to fit the data.

Since here we are talking about ICS data, we are typically interested in the computed z's from the model, i.e. the posterior probabilities of response for each subject, as a function of the effect size. The z's can be accessed directly via the @z slot of the MIMOSAResult object. The effect size can be computed from the empirical proportions of cells, or from the sampled posterior p's when using MCMC.

The fdr function computes the false discovery rate from the posterior probabilities, useful for selecting observations which show a significant difference at some threshold value.

The w's can be inerpreted as the fraction of subjects that respond, or not, to the stimulation.

```
lapply(result, function(x) table(fdr(x) < 0.01))
```

```
## $ENV
##
## FALSE  TRUE
##    63    37
##
## $GAG
##
## FALSE  TRUE
##    69    31
##
## $POL
##
## FALSE  TRUE
##    86    14
```

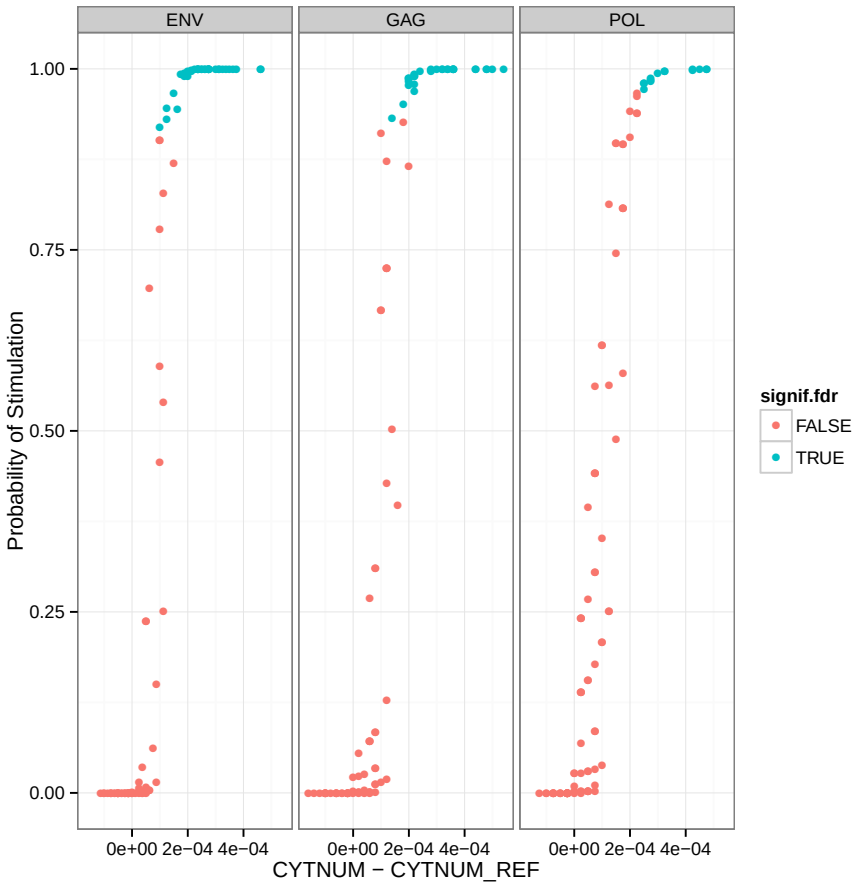
```
getW(result)
```

```
##              ENV    GAG    POL
## w.nonresp 0.5575 0.5921 0.6351
## w.resp    0.4425 0.4079 0.3649
```

4.2 Plotting the Results

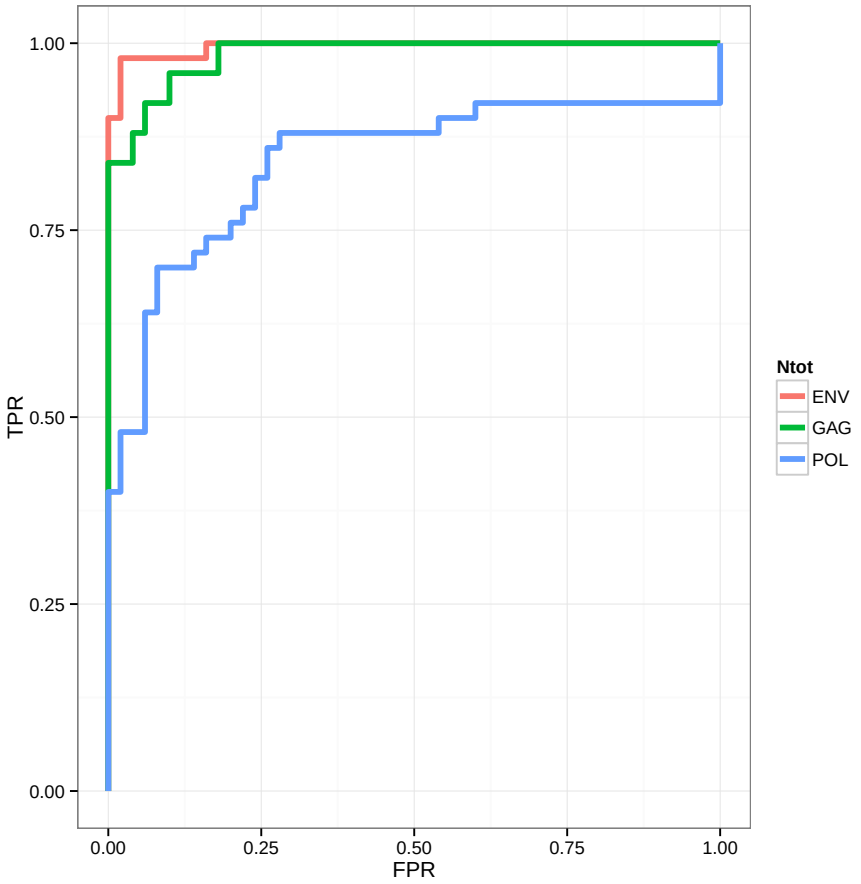
Volcano plots tend to be useful to visualize the results.

```
volcanoPlot(result, effect_expression = CYTNUM - CYTNUM_REF, facet_var = ~ANTIGEN)
```

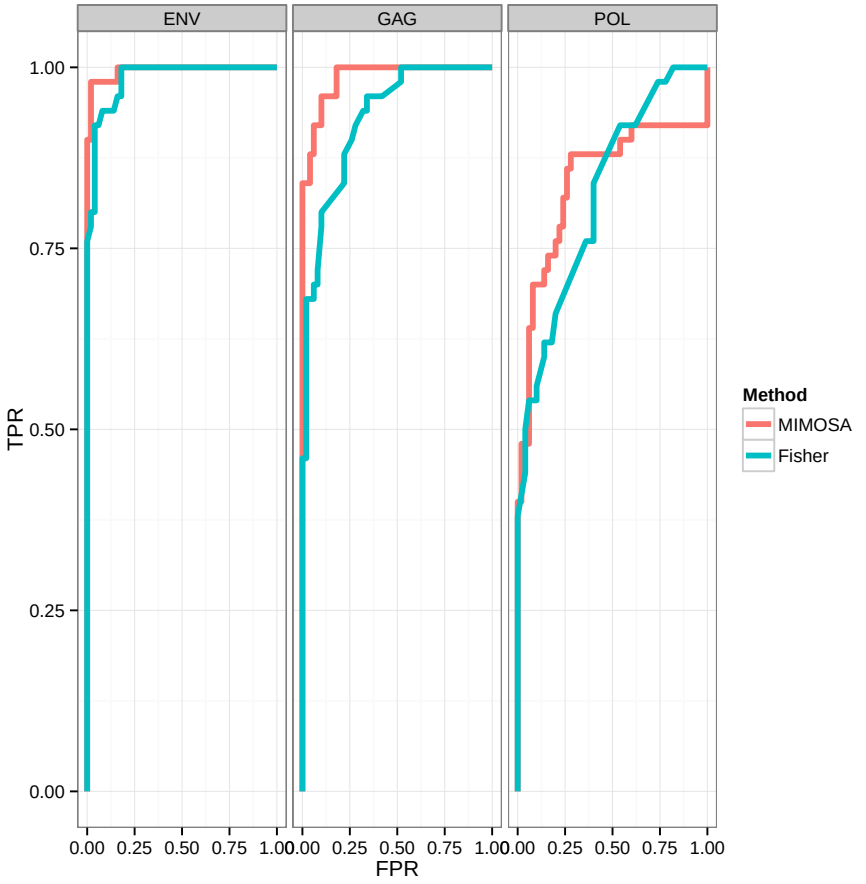


Above, you see that as the effect size increases, the model can better discriminate between responders and non-responders. If we look at ROC curves, we observe the same effect.

```
ggplot(ROC) + geom_line(aes(x = FPR, y = TPR, color = Ntot), lwd = 1.5) + theme_bw()
```



We can compare MIMOSA to Fisher's exact test:



MIMOSA can be applied to paired count data, and is not limited to ICS assays. It has been successfully used with EliSpot assays for epitope mapping as well as Fluidigm single-cell gene expression,and is applicable to any single-cell data where where cells can be thresholded as positive or negative for one or more markers and counted. Multivariate counts are supported (i.e. multiple positive categories), although MIMOSA will only test two-sided hypotheses in that case, and only globally, (i.e. you'll know there is a difference, but now which subset exhibits the difference).