MANTA:
Microbial Assemblage
Normalized Transcript Analysis
(Version 1.8.0)

Adrian Marchetti
amarchetti@unc.edu

David M. Schruth

October 14, 2013

1 Licensing

2 Installation

Building the *manta* package from source requires that you have the proper dependency packages, *caroline* and *edgeR*, installed from CRAN and Bioconductor respectively. This can typically be accomplished via the following commands from within the R command line environment:

```
install.packages('caroline')   # CRAN install
biocLite('http://bioconductor.org/biocLite.R')
biocLite('edgeR')      # Bioconductor install
```

Currently, however, the most recent development version of the *edgeR* package (> v2.5) is recommended. It must be downloaded from `http://bioconductor.org/packages/2.10/bioc/` and built manually. The *manta* and *edgeR* packages should be installed from these source archives using the following commands:

```
R CMD INSTALL edgeR_x.y.z.tar.gz
R CMD INSTALL manta_x.y.z.tar.gz
```

After a successful installation the *manta* package can be loaded in the normal way: by starting R and invoking the following `library` command:

```
> library(manta)
```

3 Introduction

One of the most difficult aspects of working with environmental shotgun sequence data has been the challenge of assigning reads to specific taxa that are collectively part of a mixed community assemblage. Continued advances in sequencing technologies–both in read lengths and throughput–have improved taxonomic assignment certainties and boosted the significance of differential expression [DE] analyses between samples. Key challenges, however, still remain in performing robust DE analysis on taxonomic subsets of these community level samples. Selection-induced shifts in the species composition of the underlying community of study are unfortunate side-effects of a multi-condition experiment and can alter measured taxonomic proportions and confound DE analysis. This is also applicable when comparing sequence libraries from environmental surveys in space and time. Employing robust normalizing techniques to control for these shifts is important when attempting to perform comparative transcriptomics on a taxonomic subset of a metatranscriptome. Equally important is deciding upon appropriately filtered taxonomic subsets for valid DE analysis. This document provides an introduction to the *manta* R package, which enables comparative metatranscritomic statistical analysis by helping users decide how to appropriately subset mixed transcript collections and thereby aide in disentangling taxonomic abundance shifts from the underlying within-species expression differences for subsequent DE analysis.

4 The MANTA object

The MANTA object is derivative of the DGEList object (from the *edgeR* package) but additionally keeps track of taxonomic meta-information and summaries of these taxonomic bins. As with its parent class, the DGEList object, the two required inputs for manta object instantiation are: 1) read counts organized into a gene (rows) by lane (columns) matrix where each row and column has a unique name and 2) a dataframe detailing provenance information for each column of the count matrix (at the very minimum, a 'groups' parameter binning the conditions of counts into replicates of individual conditions).

```
> cts.path <- system.file("extdata","PapaGO-BWA.counts-diatoms.tab", package="manta")
> cts <- read.delim(cts.path)
> samples <- makeSampleDF(cts)
> obj.simple <- manta(counts= cts, samples = samples)
> print(obj.simple)
>
```

5 Data Input

Although the *manta* object can be instantiated for non-meta transcriptomic analysis (as demonstrated in the previous section), it is often the case that the

original raw data is more complicated, comes in a variety of formats, and/or contains additional information that would not otherwise be utilized. In this section we outline several different ways that (meta) transcriptomic (and associated) data can be read into the *manta* object to enable subsequent visualization and analysis.

5.1 Counts

The simplest way to create a *manta* object is from raw count data. These counts must be annotated minimally with gene names and ideally with taxonomic (meta) information as well. The function `counts2manta` (which calls the underlying `tableMetas` and `tableMetaSums` functions) is the method you should use when converting counts.

```
> cts.path <- system.file("extdata","PapaGO-BWA.counts-diatoms.tab", package="manta")
> cts <- read.delim(cts.path)
> cts.annot.path <- system.file("extdata","PapaGO-BWA.annot-diatoms.tab", package="manta")
> cts.annot <- read.delim(cts.annot.path, stringsAsFactors=FALSE)
> obj <- counts2manta(cts, annotation=cts.annot,
+                     a.merge.clmn='query_seq', agg.clmn='what_def', meta.clmns=c('family',
+                     gene.clmns=c('what_def','kid','pathway'))
```

Notice that the secondary annotation table can (and often will) be completely different dimensions than the cts table.

5.2 Annotation Alignment

Another very common and convenient source format for *manta* object instantiation is tabular output from an alignment program such as BLAST. It is required that the alignment data frame have at least two columns at a minimum: a gene name column and a condition column. To get full use out of the package, however, it is recommended that taxonomic meta information is also included. The `align2manta` function is the method of choice for converting this alignment into a manta object.

```
> align.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.tab", package="manta")
> annot.diatoms <- read.delim(align.path, stringsAsFactors=FALSE)
> obj <- align2manta(annot.diatoms, cond.clmn='treatment', agg.clmn='what_def',
+                         gene.clmns=c('what_def','kid','pathway'),
+                         meta.clmns=c('family','genus_sp'))
```

5.3 SEAStAR Format

A third possibility (a special case of count input, detailed above in 5.1) is the Armbrust Lab's SEASTaR format which can easily be converted into counts via `readSeastar` or the `seastar2manta` functions: where the later is a wrapper for the former.

```
> conditions <- caroline::nv(factor(x=1:2, labels=c('ambient','plusFe')),c('ref','obs'))
> ss.names <- caroline::nv(paste('Pgranii-',conditions,'.seastar', sep=''), conditions)
> ss.paths <- system.file("extdata",ss.names, package="manta")
> df <- readSeastar(ss.paths[1])
>
```

5.4 PPlacer Format

A fourth input option is pplacer format, the output of the (shotgun sequencing read) phylogenetic placement program by the same name. This format consists of a directory of gene named folders, each of which contains an underlying SQLite taxonomic database file.

```
> KOG.SQLite.repo <- system.file("extdata","pplacer", package="manta")
> obj.KOG <- pplacer2manta(dir=KOG.SQLite.repo,
+                          groups=c('coastal','costal','DCM','surface','upwelling'),
+                          norm=FALSE, disp=FALSE
+                          )

KOG0035, KOG0119, KOG0521,

>
```

6 Assemblage Subsetting & Compositional Bias

One key assumption of assessing DE genes in a mixed community of organisms is that they have similar differential genetic responses between conditions. It is possible, however, that organisms within a community may *not* respond in the same way under varied experimental conditions, violating this assumption (of no-compositional bias). We have therefore developed a test to determine if a particular community sample is composed of biased taxonomic subsets.

To test this assumption, we suggest begining with a barchart-style visualization using a `compbiasPlot`

```
> compbiasPlot(obj, pair=conditions, meta.lev='genus_sp', meta.lev.lim=7)
```

We've also written a simple F-test function for the manta object which can be run as follows:

```
> compbiasTest(obj, meta.lev='genus_sp')

Analysis of Variance Table

Response: R
            Df Sum Sq Mean Sq F value    Pr(>F)
subtaxa      6   99.3 16.5438  6.6569 5.267e-07 ***
Residuals 2710 6735.0  2.4852
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

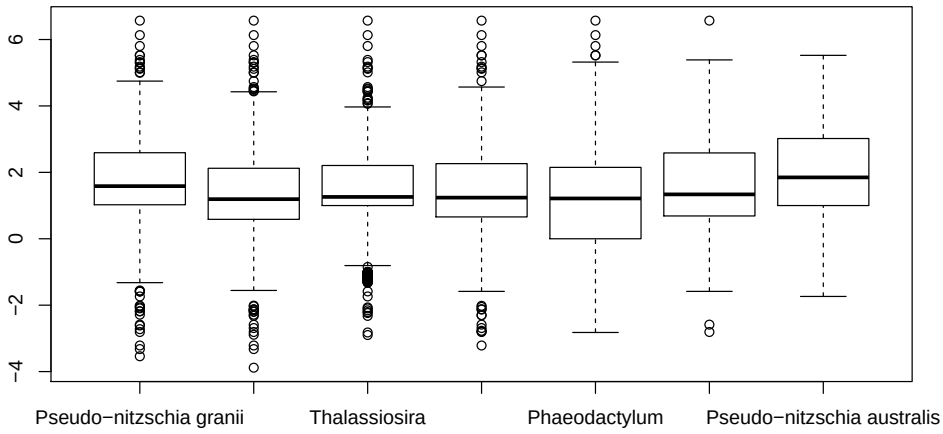

Figure 1: compbiasPlot(obj)

If the test is significant, it's advisable to remove the outlier population(s) before performing subsequent DE analysis. In the example above, the boxplots suggest that both Pseudo-nitzschia granii & P.australis have housekeeping genes with fold change distributions different from the other diatom species detected in this dataset. And indeed, `compbiasTest` confirms that these differences are probably not due to chance. Once this has been determined, we can subset out and re-test like so:

```
> annot.diatoms.sub <- subset(annot.diatoms,  !genus_sp %in% paste('Pseudo-nitzschia',c('gra
> obj.sub <- align2manta(annot.diatoms.sub, cond.clmn='treatment', agg.clmn='what_def',
+                                  gene.clmns=c('what_def','kid','pathway'),
+                                  meta.clmns=c('family','genus_sp'))
> compbiasTest(obj.sub, meta.lev='genus_sp')

Analysis of Variance Table

Response: R
           Df Sum Sq Mean Sq F value  Pr(>F)
subtaxa     4   20.7  5.1651  2.2152 0.06506 .
Residuals 2058 4798.4  2.3316
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

As you can see, the same test on this new subset of diatoms fails to reject

the null hypothesis of no housekeeping gene change in expression. We can now move on to assessing significant differential expression of the genes in this subset, without the misinforming species confounding the analysis. By focusing on a more homogenous, similarly responding subset of species, we can greatly improve the power of the DE analysis.

7 Normalization and Differential Expression

The *edgeR* and *manta* packages both provide support for normalization and estimation of DE genes. Prerequisite calculations for normalization factors and dispersion coefficients (respectively) can be carried out like so:

```
> obj.sub <- calcNormFactors(obj.sub)
> obj.sub <- estimateCommonDisp(obj.sub)
```

Because our example dataset doesn't have any replicates, estimateCommonDisp returns NA. Although it is not advised to proceded on to performing inferential statistics without replicates, the *edgeR* manual offers several alternatives to assuming that biological variability is absent. For the purposes of example we use *edgeR*'s `estimateGLMCommonDisp`.

```
> obj.sub <- estimateGLMCommonDisp(obj.sub, method="deviance", robust=TRUE , subset=NULL)
> obj.sub$common.dispersion

[1] 0.4450603
```

DE analysis is carried out on the normalized and dispersion estimated object in the standard way using the `exactTest` function.

```
> test <- exactTest(obj.sub)  # edgeR
```

Other likelihood based tests such as `glmFit` and `glmLRT` also run in a similar fashion.

7.1 Finding Outliers

One useful function built into *manta*'s underlying *edgeR* package is the `topTags` function, which returns the most extreme up and down regulated DE genes as sorted by either p.value or fold change. Another function, `outGenes` (in the *manta* package), provides additional (cut-off based) functionality for identifying these fold change outliers as well as those with high average absolute read count.

```
> topTags(test, n=5)

Comparison of groups:  plusFe-ambient
                                                                what_def
magnesium chelatase                                  magnesium chelatase
```

```
small subunit ribosomal protein S30e small subunit ribosomal protein S30e
small subunit ribosomal protein S29e small subunit ribosomal protein S29e
actin, other eukaryote                             actin, other eukaryote
taurine dioxygenase                                   taurine dioxygenase
                                        kid  pathway
magnesium chelatase                  K03403     <NA>
small subunit ribosomal protein S30e K02983 Ribosome
small subunit ribosomal protein S29e K02980     <NA>
actin, other eukaryote               K10355     <NA>
taurine dioxygenase                  K03119     <NA>
                                           logFC   logCPM
magnesium chelatase                     8.265404 13.33390
small subunit ribosomal protein S30e    7.521348 12.69618
small subunit ribosomal protein S29e    6.965635 12.24269
actin, other eukaryote                 -3.437644 13.66593
taurine dioxygenase                     6.529179 11.90461
                                            PValue FDR
magnesium chelatase                  0.004885506   1
small subunit ribosomal protein S30e 0.014920741   1
small subunit ribosomal protein S29e 0.031374603   1
actin, other eukaryote               0.050537155   1
taurine dioxygenase                  0.055485482   1

> out <- outGenes(test)

obs/ref (up-regulated)
                                              R        A
magnesium chelatase                  8.265404 13.33390
small subunit ribosomal protein S30e 7.521348 12.69618
small subunit ribosomal protein S29e 6.965635 12.24269
                                           PValue adj.p
magnesium chelatase                  0.004885506     1
small subunit ribosomal protein S30e 0.014920741     1
small subunit ribosomal protein S29e 0.031374603     1

ref/obs (down-regulated)
    R  A PValue adj.p
NA NA NA     NA    NA

high abundance (house-keeping)
                       R        A    PValue adj.p
tubulin alpha -1.126268 16.03549 0.4467385     1
fucoxanthin    1.756782 14.41336 0.2582533     1
```

8 RA Plots

Another feature the *manta* package offers is the ability to create sophisticated plots for visualizing DE. Improving upon *edgeR* which contains `maPlot` for creating Magnitude Amplitude [MA] plots from a pair of count vectors, the *manta* package offers native plotting support for its *manta* object, instead employing `raPlot` to generate Ratio Average [RA] plots, as implemented in the *caroline* package. RA plots are nearly identical to MA plots with the exception of the distinctive geometric 'ray' (arrow) shape caused by the way the condition unique points are included, via an added epsilon factor, in the plot. The diagonal positioning of these library-unique wings (forming the head of the arrow), aligns points in the wings to fold-change and amplitude levels of non-wing points that have similar p-values.

8.1 MANTA RA-plots

Any meta information available in the 'meta' slot of the manta object can be included in the RA plot as pie-chart overlays depicting the taxonomic distribution of any particular gene on the plot. Significance of DE is easily visualized in the plot via the 'borders' parameter. The normalization line can be added using the mm parameter or the mm slot on the plotted manta object.

```
> obj.sub$nr <- nf2nr(x=obj.sub, pair=conditions)
> plot(obj.sub, main='Diatom Gene Expression\n at Ocean Station Papa', meta.lev='genus_sp',
```

8.2 Pseudo RA-plots

Another simple way to plot the differential expression data is directly plot the raw R (logFC) and A (logCPM) values output from `exactTest`. This can be a convenient solution when plotting count data with replicates (something that is currently not supported in `raPlot` or `maPlot`).

```
> with(test$table, plot(logCPM, logFC))
>
```

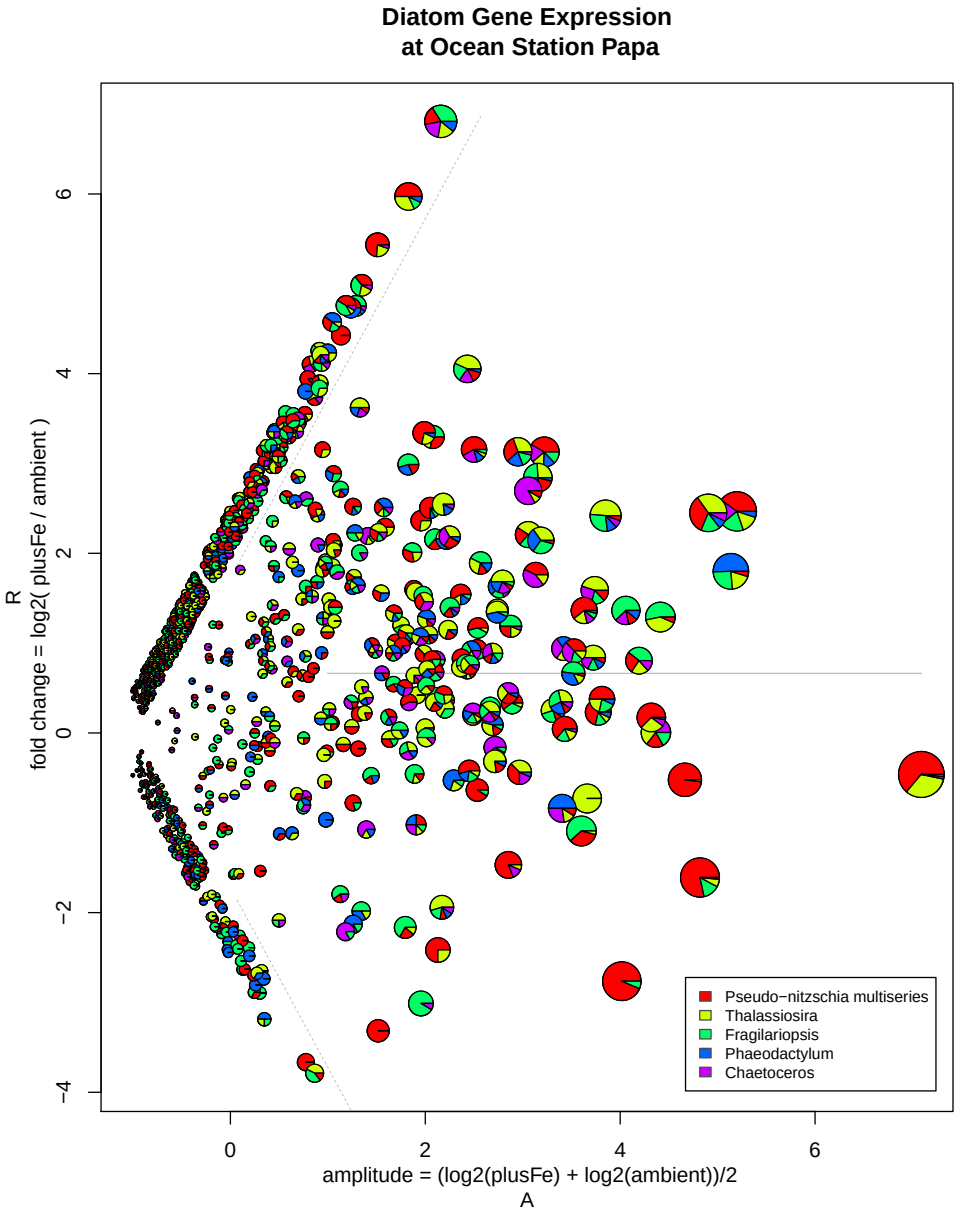

Figure 2: an example of plot(manta) call.

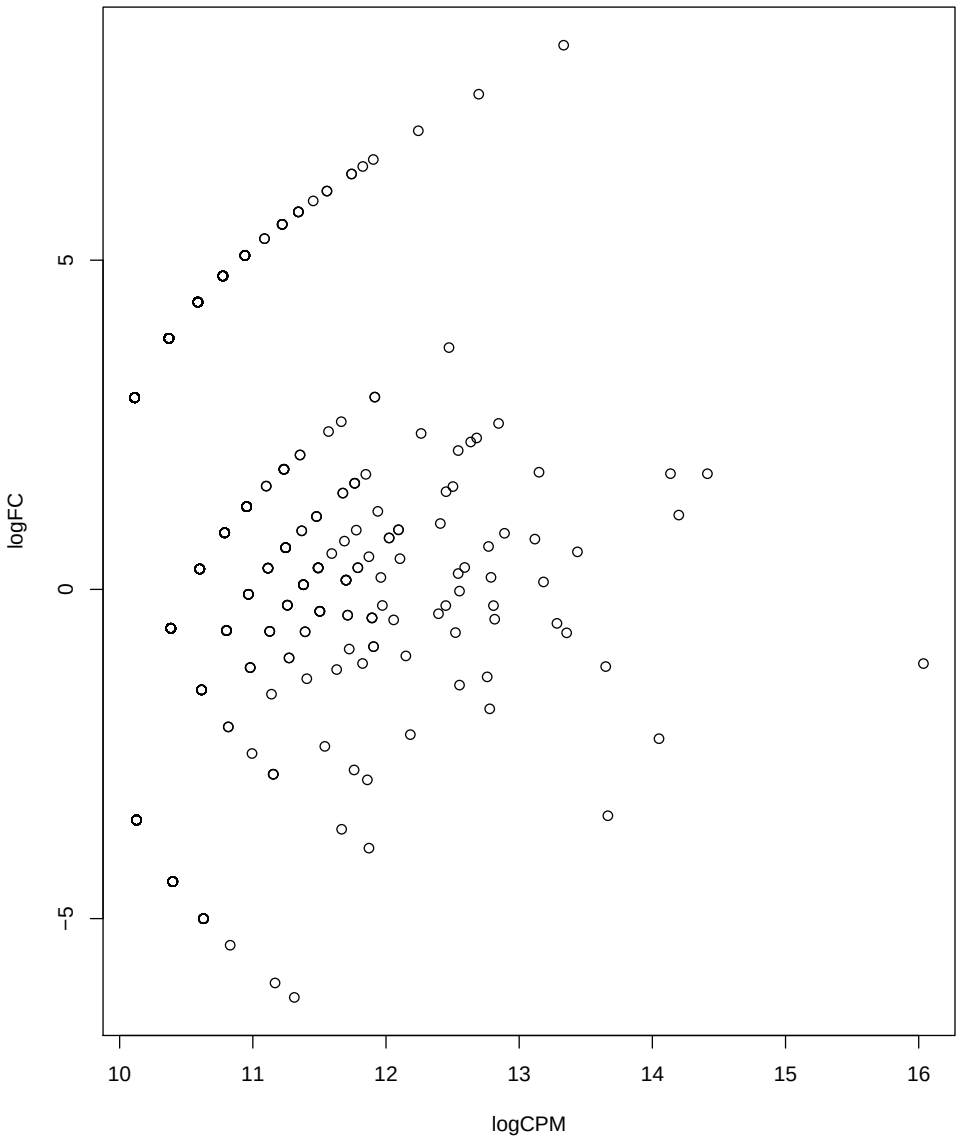

Figure 3: plot(logCPM, logFC)

8.3 Annotating via HyperPlot

A new R function for annotating scatterplots with interactive hover text and links is also compatible with the manta object.

```
> caroline::hyperplot(obj, annout=out, browse=FALSE)
```

References

Marchetti A, Schruth DM, Durkin CA, Parker MS, Kodner RB, Berthiaume CT, Morales R, Allen A, and Armbrust EV (2012). Comparative metatranscriptomics identifies molecular bases for the physiological responses of phytoplankton to varying iron availability. Proceedings of the National Academy of Sciences 109, no.6, E317

Matsen FA, Kodner RB, and Armbrust EV (2010). pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. BMC Bioinformatics 11, 538

Robinson MD, McCarthy DJ and Smyth GK (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26, 139-140

Robinson MD and Smyth GK (2007). Moderated statistical tests for assessing differences in tag abundance. Bioinformatics 23, 2881-2887