nMyo: an R package and shiny applet to visualise the scRNA-seq expression profiles of mouse cardiac non-myocytes

Efthymios Motakis & Diana Low

05/01/2020

1. Background

We have developed the **nMyo** R shiny applet and associated R package to visualise the single-cell RNAseq expression profiles of 2031 high-quality mouse cardiac non-myocytes. The quality control, data generation and normalisation, cluster annotation and differential expression steps are briefly discussed in the subsequent paragraphs. **nMyo**'s input is the outcome of the counts normalisation, cell characterisation and differential expression analysis pipelines (the respective input format is described below). Practically, the user selects a gene ID and our software generates a series of interactive plots for data viusalisation and exploratory analysis.

1.1 Cell isolation and RNA sequencing

The cells were dissociated from the whole left ventricle of 3 Sham-operated and 3 MI-induced mice. The cell isolation was based on a FACS-scRNAseq protocol for global, non-biased RNA sequencing performed with the SMART-seq2 technology on Illumina 4000.

1.2 From fastq to raw counts

The raw single-cell, paired-end reads in fastq format of the originally 2,272 mouse non-myocytes (1,152 MI and 1,120 Sham) were initially processed with FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/) for quality control at the base and sequence level. The Nextera adapters were removed by `Trimmomatic` (Bolger et al, 2014) and the PCR-deduplication was perfomred by `FastUniq` (Hu et al, 2012). The GENCODE M12 annotated transcripts were quantified by `Kallisto v.43` (Bray et al, 2016). The conversion of transcript to gene counts (and transcript length corrected counts) was performed by the `tximport` package in R.

1.3 Overview of the main data analysis

We performed QC at the gene expression level, retaining samples with more than 200,000 reads, less than 15% of the reads mapped to mitochondrial RNA and more than 1500 detected genes. The samples that did not pass the QC were considered of low quality and they were removed from further analysis. The transcript-length corrected gene counts of 26,942 genes and 2,031 high-quality cells (957 Sham and 1,074 MI) were normalized to a set of endogenous reference genes that were identified by intersecting the top non-differentially expressed genes (Risso et al, 2014) and least variable genes of `scatter` (HVG function) in R. To identify the cell types, we run 2D `t-SNE` on the normalised data using the first 20 PC components (Rtsne; van der Maaten, 2014) and clustered the data by `Affinity Propagation` (Frey and Dueck, 2007). The clusters were annotated based on evidence from the significant up-regulation of known markers of major cardiac cell types combined with gene ontology functions of upregulated genes. Eleven distinct cell types were identified, i.e. cardiac fibroblasts (440 Sham and 390 MI), endothelial cells (326 Sham and 258 MI), endothelial ECM cells (61 Sham and 65 MI), lymphatic endothelial (10 Sham and 34 MI), T-cells (16 Sham and 39 MI), B-cells (20 Sham and 17MI), two machrophage populations (Pop1: 4 Sham and 69 MI; Pop2: 17 Sham and 18 MI), smooth muscle

cells (12 Sham and 19 MI), pericytes (36 Sham and 46 MI) and neutrophils (5 Sham and 22 MI). A small cluster of 10 Sham and 7 MI cells had an unpredicted cell type. The edgeR differential expression analysis of MI vs Sham or CellType i vs Rest raw counts was performed after accounting for technical differences between plates and cell cycle phase via the `cyclone()` function of `scran` R package (FDR = 10%).

2. The input matrices

nMyo expects as input a set of .txt files of a specific format which we will briefly describe here.

- CorrCounts: a matrix of the normalised gene counts of 26,942 genes and 2,031 cells. The first column (column name: `Marker`) contains the gene IDs in the format *EnsemblID:GeneSymbol*. The column names of the other columns are the unique cell IDs.
- design_table: a matrix summarising the various cell characteristics (experimentally known or predicted) of our dataset. The first column (column name: `SampleID`) contains the unique cell IDs depicted in the corrCounts matrix (in the same order). Some important experimentally known characteristics are the sequencing `Plate` and the cell experimental `Condition` (Sham and MI). The rest have been estimated from the data. The most important are: the cell `Library Size`, the percentage of reads mapped to `mitochondrial` genes, the number of `detected genes`, the estimated `cell cycle` phases, the t-SNE dimensions (`Dim1`, `Dim2` and `Dim3`) and the estimated `cell types`.
- DE: a matrix with the differential expression analysis estimates. The first column (column name: `Marker`) has the gene IDs formatted as in the CorrCounts matrix. Other essential statistics are the `logFC`, `logCPM`, `PValue` and `FDR` that are directly obtained from edgeR. The last two columns, `cell type` and `comparison`, contain values to characterise and separate the pairwise tests. For example, CellType = `ShamMI` and Comparison = `Fibroblast-Rest` indicate the comparison of Fibroblast vs the rest of the cell types across all cells while CellType = `Fibroblast` and Comparison = `Sham-MI` the comparison of Sham vs MI in fibrblasts.

3. The nMyo R package

The **nMyo** R package can be installed as from GitHub. This will automatically install and load all dependencies. The data visualisation is performed in 3 steps, i.e. data loading, gene selection and data visualisation. Here, we briefly describe the functions of interest. More detailed information can be obtained from the package help pages (e.g. `?readData` for the help page of the readData() function).

3.1 Data loading

The data loading step is done as:

```
data(nMyo_Data)
data<-readData(nMyo_Data,Markers_file=NULL,is.Exact=TRUE,logFC_cutoff=0,FDR_cutoff=1)
```

```
## [1] "***** The data have been successfully loaded! *****"
```

```
# listed components
names(data)
```

```
##  [1] "Counts"             "Design"             "DEstats"
##  [4] "Annotation"         "Status"             "Exact_Marker_Match"
##  [7] "logFC"              "FDR"                "Output_Folder"
## [10] "Dimensions"         "temp"               "filteredData"
## [13] "Date_Stamp"
```

```
# the CorrCounts
data$Counts[1:5, 1:5]
```

```
##                            RMH2238  RMH2239 RMH2240 RMH2241  RMH2244
## ENSMUSG00000000001:Gnai3 12.93526 10.92297       0       0 4.684664
## ENSMUSG00000000028:Cdc45  0.00000  0.00000       0       0 0.000000
## ENSMUSG00000000031:H19    0.00000  0.00000       0       0 0.000000
## ENSMUSG00000000037:Scml2  0.00000  0.00000       0       0 0.000000
## ENSMUSG00000000056:Narf   0.00000  0.00000       0       0 0.000000
```

```
# the design_table
data$Design[1:5, ]
```

```
##         SampleID Plate  Mouse Condition LibrarySize Spike_n   Spike_p
## RMH2238  RMH2238    P3 A-Sham      Sham     2833319   14423 0.4698843
## RMH2239  RMH2239    P3 A-Sham      Sham     1934676    3074 0.1470934
## RMH2240  RMH2240    P3 A-Sham      Sham     2395588   20208 0.7688614
## RMH2241  RMH2241    P3 A-Sham      Sham     1964128    4306 0.2030815
## RMH2244  RMH2244    P3   A-MI        MI     1826143    5780 0.2929754
##         Ribosomal_p Mitochondrial_p Detected_genes Batch CycloneGroups
## RMH2238    6.451095       0.7728347           3331    B1            G1
## RMH2239    6.691361       0.5857417           3934    B1            G1
## RMH2240    7.676135       0.4091615           3248    B1            G1
## RMH2241    6.799127       0.3647072           3795    B1            G1
## RMH2244    6.580491       0.5633947           3581    B1           G2M
##              Dim1        Dim2       Dim3 AP_clusters Louvain_clusters   Jaccard
## RMH2238  5.519352 -0.02171482  1.0218681         AP5               L7 0.7973589
## RMH2239 -2.576475 -2.25756127 -0.6227282         AP9               L1 0.6145859
## RMH2240 -2.262321 -2.08032755 -1.5056269        AP29               L0 0.6100619
## RMH2241 -1.913847 -1.49179762 -4.1076580         AP1               L1 0.7724455
## RMH2244 -5.198106  5.72491869  2.5965975        AP14               L3 0.9679335
##             AP_type    CellType Seurat_groups Colors_CellType Colors_Condition
## RMH2238      Stable Endothelial   Endothelial         #F8766D          #00B0F6
## RMH2239     Pattern  Fibroblast    Fibroblast         #00B0F6          #00B0F6
## RMH2240     Pattern  Fibroblast    Fibroblast         #00B0F6          #00B0F6
## RMH2241      Stable  Fibroblast    Fibroblast         #00B0F6          #00B0F6
## RMH2244 Very_stable     MF_Pop2       MF_Pop2         #00BA38          #F8766D
```

```
# the DE
data$DEstats[1:5, ]
```

```
##                     Marker logFC     logCPM PValue   FDR CellType  Comparison
## 1 ENSMUSG00000000001:Gnai3 -1.29 5.11441659 0.1580 0.297   ShamMI B_cell-Rest
## 2 ENSMUSG00000000028:Cdc45 -6.40 0.41089721 0.0660 0.189   ShamMI B_cell-Rest
## 3   ENSMUSG00000000031:H19 -5.59 0.23841731 0.0293 0.132   ShamMI B_cell-Rest
## 4 ENSMUSG00000000037:Scml2 -3.94 0.09094684 0.0484 0.167   ShamMI B_cell-Rest
## 5  ENSMUSG00000000056:Narf -1.66 1.70434303 0.3100 0.459   ShamMI B_cell-Rest
```

```
# the data stamp (for file storage)
data$Date_Stamp
```

```
## [1] "Mon_Jun__12:16:18_1"
```

Among the `data` components we find the CorrCounts (`Counts`), design (`Design`) and differential expression (`DEstats`) matrices as well as other automatically generated information such as: `Annotation` with the geneIDs, `logFC` with the logFC cut-off for preliminary marker filtering based on the differential expression analysis results (default is 0), `FDR` with the FDR cut-off for preliminary marker filtering based on the differential expression analysis results (default is 1), `Output folder` specifying the folder that stores the results and `Date stamp` that assigns unique filenames for storage.

3.2 Gene selection

Next, the user selects the gene for visualisation by its Ensembl ID or its gene symbol or the combination *EnsemblID:GeneSymbol*. Any of the genes present in the CorrCounts matrix can be selected. Below, we indicate some examples:

```
data_Postn <- MarkerQuery(Data = data, marker = "Postn")
```

```
## [1] "Gene ENSMUSG00000027750:Postn is selected for visualisation."
```

Parameter `Data` accepts the outcome of readData() function while parameter `marker` takes the ID of interest. The above example shows the selection of *Postn* gene for further analysis. Alternatively, the user can select the respective Ensembl ID as:

```
data_Postn_alt <- MarkerQuery(Data = data, marker = "ENSMUSG00000027750")
```

```
## [1] "Gene ENSMUSG00000027750:Postn is selected for visualisation."
```

or he can use the combination of the two as:

```
data_Postn_alt2 <- MarkerQuery(Data = data, marker = "ENSMUSG00000027750:Postn")
```

```
## [1] "Gene ENSMUSG00000027750:Postn is selected for visualisation."
```

To account for small typos in the `marker` input, **nMyo** finds the best match associated to the selected gene, e.g.:

```
data_Postn_err <- MarkerQuery(Data = data, marker = "Post")
```

```
## [1] "Gene Post does not exactly match any of the existing IDs! The top match ENSMUSG00000027750:Postn
```

If the selected gene is not present in the data, **nMyo** generates an appropiate error and the analysis stops, prompting the user to select another ID:

```
data_err <- MarkerQuery(Data = data, marker = "Tp53")
```

```
## [1] "This gene ID does not match any of the existing IDs!"
```

3.3 Data visualisation

nMyo can generate four types of interative plots for gene expression visualisation and exploration. First, we will see the t-SNE scatterplot depicting the expression levels of the selected gene with a colour gradient (blue-to-red). Mouse-over on the plot reveals the estimated normalised expression level, the experimental condition and the estimated cell type of each cell. By default, double clicking on any of the dots highlights all cells of the same cell type (controlled by the parameter `highlight.by`). Alternative values of `highlight.by` are possible to bring forth other aspects of the data (e.g. `Condition` or any other factor of the design_table). The logical parameters `show.plot` and `save.plot` determine whether the plot will be shown on screen and/or stored as an html (interactive) file. Finally, clicking on the gene ID of the plot legend directs the user to the respective NCBI page of the gene.

```
scatter_Postn <- doScatterDR(Data = data_Postn,
                             highlight.by = "CellType",
                             show.plot = TRUE,
                             save.plot = FALSE)
```

```
## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plot.ly/r/reference/#scatter-mode
```

A split-violin plot for each cell type (default; controlled by the parameter `grouping.by`) splitted by `Condition` (default; controlled by the parameter `grouping2.by`) can be generated as:

```
violin_Postn <- doViolin(Data = data_Postn,
                         grouping.by = "CellType",
                         grouping2.by = "Condition",
                         show.plot = TRUE,
                         save.plot = FALSE)
```

Other values of `grouping.by` and `grouping2.by` are also possible (essentially any of the factors of the design_table). For example, setting `grouping2.by = NULL` generates a simple violin plot for each cell type. A constraint is currently imposed on `grouping2.by` that can only accept factors with 2 levels (e.g. `Condition` is either Sham or MI).

```
violin_Postn2 <- doViolin(Data = data_Postn,
                         grouping.by = "CellType",
                         grouping2.by = "",
                         show.plot = TRUE,
                         save.plot = FALSE)
```

```
## [1] "***** The grouping2.by variable does not exist in the Design file. Setting grouping2.by = NULL!
```

The Volcano and MA plots are governed by a similar set of parameters. An interactive volcano plot can be generated as:

```
volcano_Postn <- doVolcano(Data = data_Postn,
                           filters = "Comparison=='Fibroblast: Sham-MI'",
                           logFC = 1,
                           FDR = 0.01,
                           show.plot = TRUE,
                           save.plot = FALSE)
```

```
## Adding markers to mode; otherwise symbol would have no effect.
```

Parameter `filters` specifies the pairwise comparison to be shown. There are several alternatives:

```
# all pairwise comparisons (to be used in filter parameter)
names(table(data$DEstats$Comparison))
```

```
##  [1] "B_cell-Rest"                "B_cell: Sham-MI"
##  [3] "Endothelial-Rest"           "Endothelial: Sham-MI"
##  [5] "Endothelial_ECM-Rest"       "Endothelial_ECM: Sham-MI"
##  [7] "Fibroblast-Rest"            "Fibroblast: Sham-MI"
##  [9] "Lymph_Endothelial-Rest"     "Lymph_Endothelial: Sham-MI"
## [11] "MF_Pop1-Rest"               "MF_Pop1: Sham-MI"
## [13] "MF_Pop2-Rest"               "MF_Pop2: Sham-MI"
## [15] "Neutrophil-Rest"            "Neutrophil: Sham-MI"
## [17] "Pericyte-Rest"              "Pericyte: Sham-MI"
## [19] "SMC-Rest"                   "SMC: Sham-MI"
## [21] "T_cell-Rest"                "T_cell: Sham-MI"
## [23] "Unknown-Rest"               "Unknown: Sham-MI"
```

The differentially expressed genes (the `logFC` and `FDR` parameters control the cutoffs) are highlighted in different color along with the selected gene which is highlighted with a large triangle. Clicking on it, the user is redirected to the accosiated NCBI page. In the same way, one can generate the MA plot:

```
ma_Postn <- doMA(Data = data_Postn,
                 filters = "Comparison=='Fibroblast: Sham-MI'",
                 logFC = 1,
                 FDR = 0.01,
```

```
                show.plot = TRUE,
                save.plot = FALSE)
```

```
## Adding markers to mode; otherwise symbol would have no effect.
```

```
## A marker object has been specified, but markers is not in the mode
## Adding markers to the mode...
```

4. The nMyo R shiny applet

The **nMyo** web applet can be accessed with

```
run_nMyo()
```

The applet offers to the computationally inexperienced users an alternative, easy-to-use way to visualise our non-myocyte dataset at the cost of less flexibility. It only requires a prior installation of R or R studio in their system.

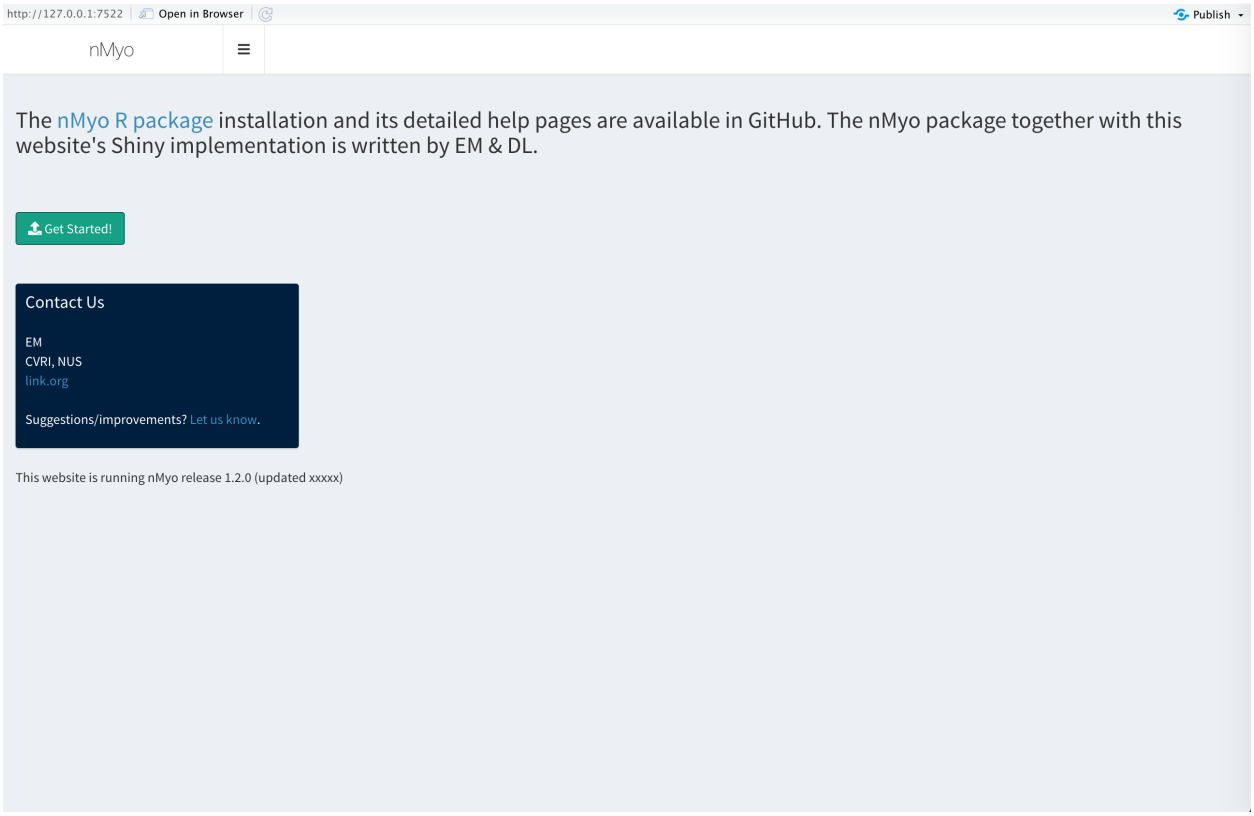

Figure 1: nMyo welcome screen

The three input data tables are automatically loaded upon **nMyo**'s initialization. By clicking on *Get Started!* the user is redirected to the Quick Analysis tab.

The right part of the screen allows the user to download the input matrices and see at a glance the package's functionality. As before, the user is enabled to select the gene ID of interest and load it in the system. The *Plot Options* tab will be subsequently activated enabling the user to select one of the four data visualisation plots.

By default, the t-SNE higlights the cell types (`highlight.by = CellType`), the violin is done by cell type

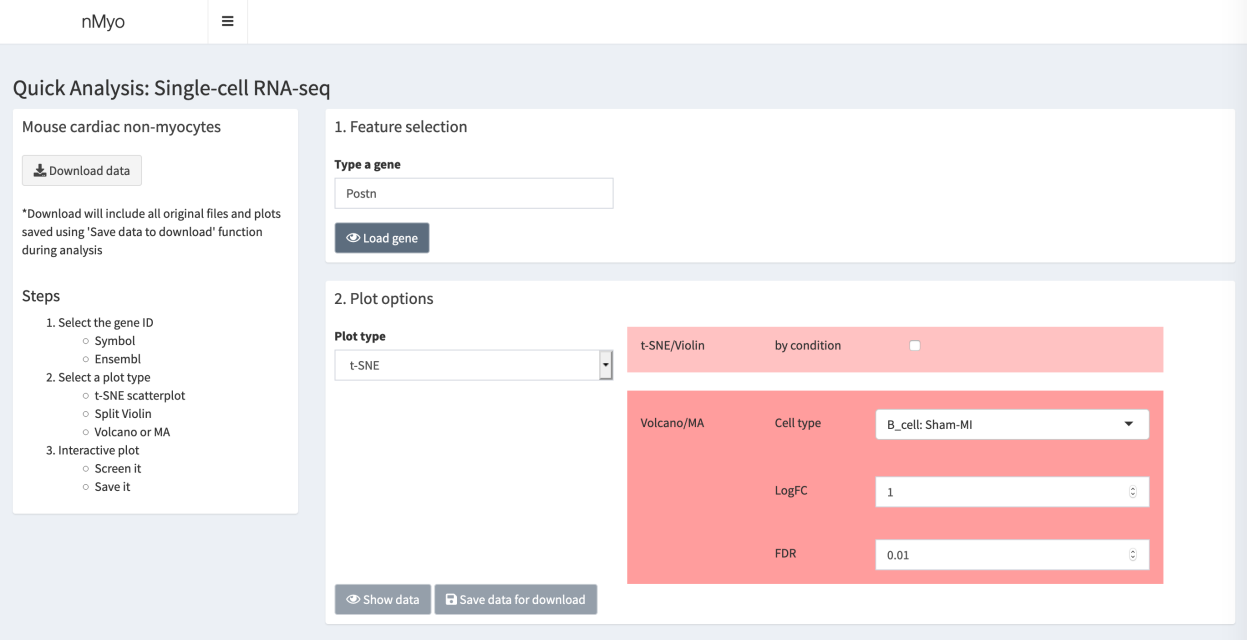

Figure 2: nMyo welcome screen

only (not split) and the MA/Volcano plots highlight the differentially expressed genes at logFC = 1 and FDR = 0.01. Clicking on the `by condition` checkbox activates the `Conditions` highlight in the t-SNE and the split by condition in the violin plot. The `cell type`, `logFC` and `FDR` parameters can be adjusted accoring t the user's needs. The user can either see the plot at the bottom of the screen (`Show data`) and/or save it (`Save data`) as an htlm interactive file located in the `Data/InteractivePlots` subfolder of the package.

5. References

- Bolger et al. "Trimmomatic: a flexible trimmer for Illumina sequence data". *Bioinformatics* 2014; 30:2114-2120.
- Hu et al. "FastUniq: a fast de novo duplicates removal tool for paired short reads". *PLoS One* 2012; 7:e52249.
- Bray et al. "Near-optimal probabilistic RNA-seq quantification". *Nature Biotechnology* 2016; 34:525-527.
- van der Maaten. "Accelerating t-SNE using Tree-Based Algorithms". *Journal of Machine Learning Research* 2014; 15:3221-3245.
- Frey and Dueck. "Clustering by passing messages between data points". *Science* 2007; 315:972-976.