Visualization and evaluation of RNA-seq data normalization – the Normalization Visualization Tool

Thomas Eder[1,2], Thomas Rattei[1]

[1] CUBE Division of Computational Systems Biology,
Department of Microbiology and Ecosystem Science,
University of Vienna, Althanstrasse 14, 1090 Vienna, Austria.
[2] Ludwig Boltzmann Institute for Cancer Research,
Waehringer Strasse 13A, 1090 Vienna, Austria.

January 12, 2016

Abstract

Differential expression analysis, between two samples, is a common task in the analysis of RNA-Seq data. In order to identify differential expressed genes it is crucial that the two compared data sets are normalized. For the task of normalizing gene expression data there are multiple methods available. But all of them are based on certain assumptions that can or can not be meet by the data which should be normalized. An important question is, how the normalization affects the data and which normalization method and its assumptions blend well with the expression data. The *NVT* package provides a fast and simple way to analyze multiple normalization methods via visualization and correlation values. This vignette explains the use of the package and demonstrates a typical work flow.

NVT version: 0.3

If you use *NVT* in published research, please cite:

T. Eder, T.Rattei: **NVT: a fast and simple test for the evaluation of RNA-Seq normalization strategies**. *Bioinformatics* 2016, **15**:550.
http://dx.doi.org/10.1186/s13059-014-0550-8

Contents

1 Installation 3

2 Standard work flow 3
2.1 Load data 3
2.1.1 Load expression data 3
2.1.2 Load gene length data 3
2.2 Generate NVTdata objects 3
2.2.1 Normalize the NVTdata 4
2.3 Visualize expression data 4
2.3.1 Simple plot 4
2.3.2 Advanced plot 4
2.4 Correlation values 4
2.4.1 Pearson correlation 4
2.4.2 RMSD and MEA correlation 4
2.5 Test all methods 4

3 Acknowledgments 4

4 Session Info 4

1 Installation

Download the latest NVT package from: https://github.com/NexusX/NVT/releases and install it with the following command.

```
install.packages(file.path("/home/user/Downloads/","NVT_1.0.tar.gz"),
repos=NULL, type="source")
```

2 Standard work flow

2.1 Load data

We need some gene expression data, load the example data provided with the package

```
library("NVT")
```

2.1.1 Load expression data

load example data

```
data(mylen)
data(myexp1)
data(myexp2)
mylist1=c('CT462','CT115','CT045','CT678')
```

2.1.2 Load gene length data

Instead of using a simple flat file, it is also possible to load the gene or exon length data directly from an annotation file in gtf or gff format.

```
mygffpath<-system.file("extdata", "Ctr-D-UW3CX.gff", package = "NVT")
mylen <- NVTloadgff(mygffpath,"gff3","gene","locus_tag")
head(mylen)
```

```
##       Length
## CT001    273
## CT002    303
## CT003   1476
## CT004   1467
## CT005   1092
## CT006    570
```

2.2 Generate NVTdata objects

```
mynvt <- NVTinit(mylist1,myexp1,myexp2,"RPKM",mylen)
```

2.2.1 Normalize the NVTdata

```
mynorm <- NVTnormalize(mynvt)

## [1] "RPKM normalization!"
```

2.3 Visualize expression data

2.3.1 Simple plot

```
png(filename = "figure/simpleplot.png", width = 580, height = 580, units = "px", pointsize = 12)
NVTplot(mynorm,1)
dev.off()

## png
##   2
```

2.3.2 Advanced plot

```
png(filename = "figure/advancedplot.png", width = 580, height = 580, units = "px", pointsize = 12
NVTadvancedplot(mynorm,2,2,4)
dev.off()

## png
##   2
```

2.4 Correlation values

2.4.1 Pearson correlation

2.4.2 RMSD and MEA correlation

2.5 Test all methods

3 Acknowledgments

We have benefited in the development of *NVT* from the help and feedback of many individuals,

4 Session Info

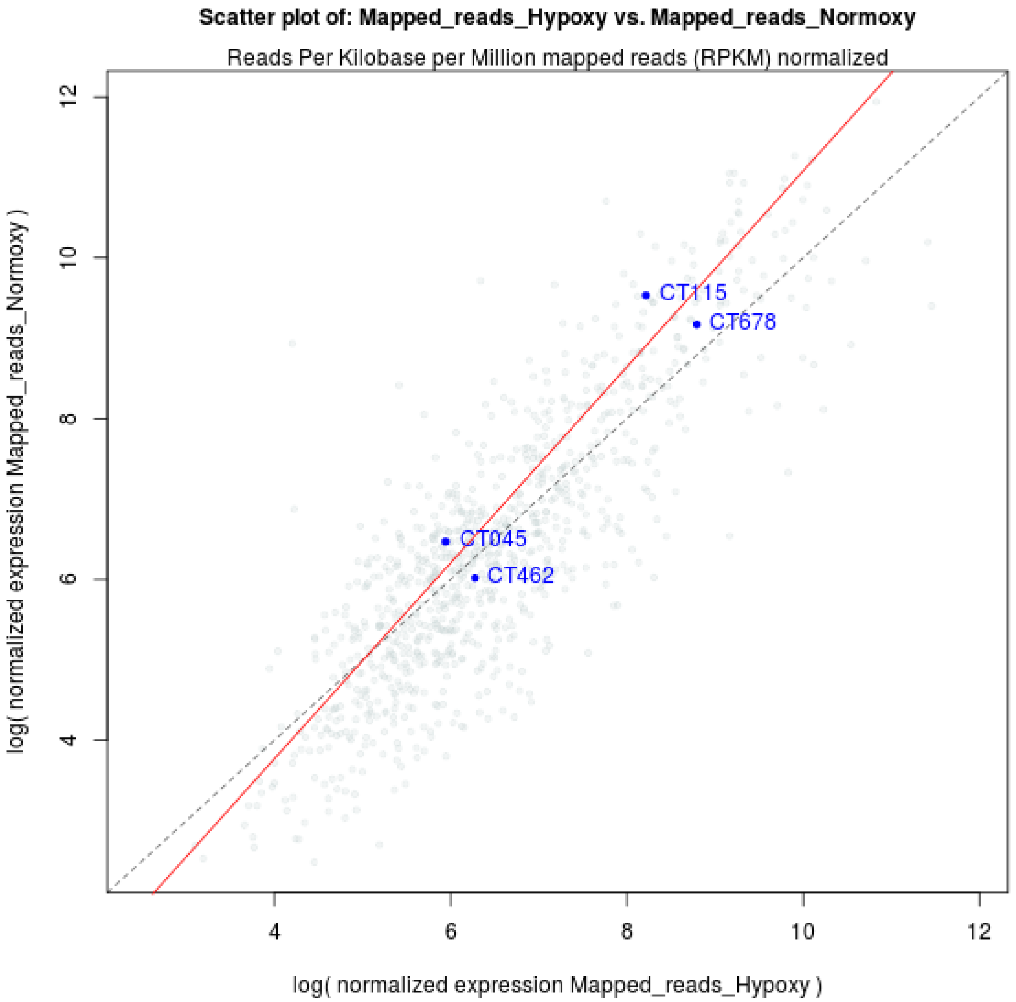

Figure 1: **Simple XY-Scatter-Plot.** This plot shows the simple NVT plot function result

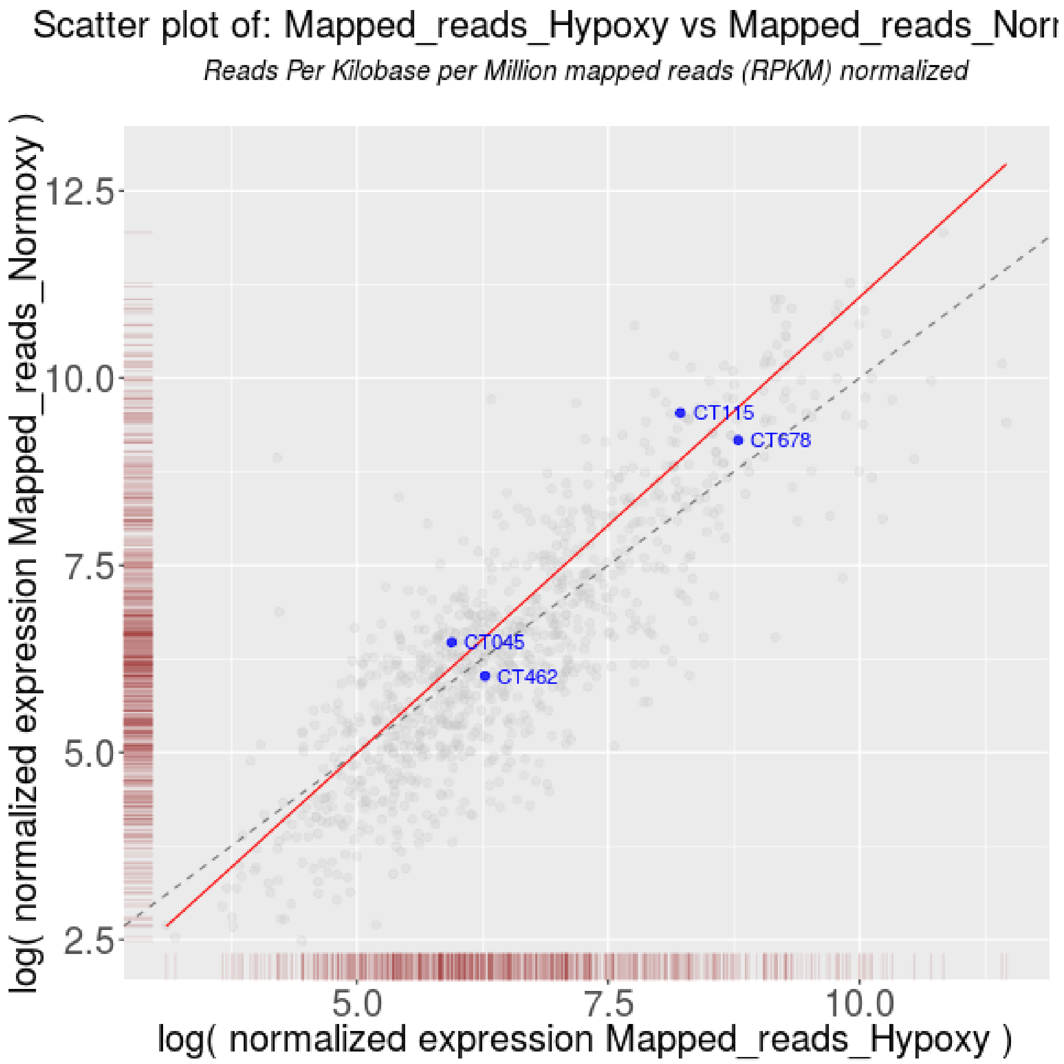

Figure 2: **Advanced XY-Scatter-Plot.** This plot shows the advanced NVT plot function result