AGED Vignette on TCGA

Michael Sweeney, Luke Torre-Healy

11/27/2020

Load TCGA expression matrix

```r
data("TCGA_PAAD")
TCGA_PAAD$ex = TCGA_PAAD$ex[1:1000,]
TCGA_PAAD$featInfo = TCGA_PAAD$featInfo[1:1000,]
```

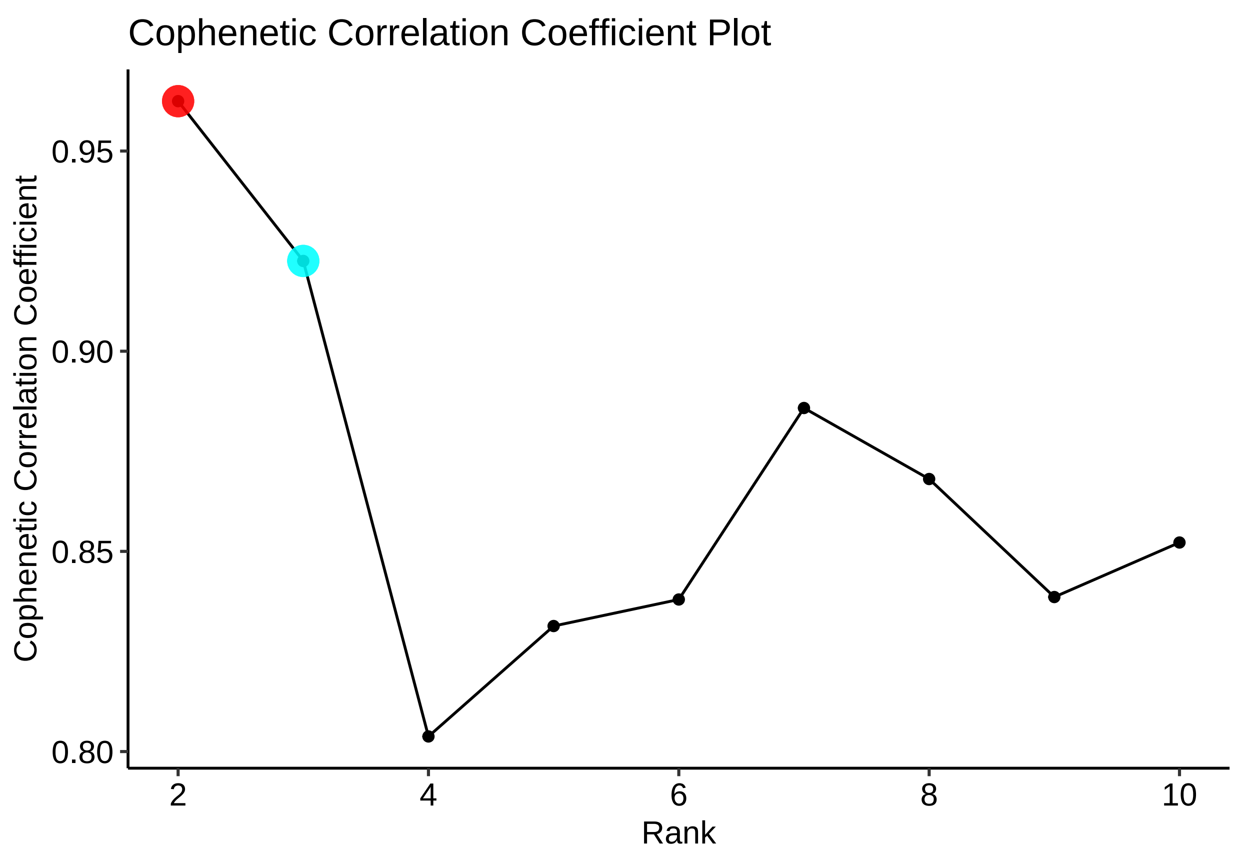
Prep for NMF

First, for a trial run, trim your data so you aren't overloading your memory with computationally intensive NMF. Also, pre-normalize your data using VST or log if it is not already.

```r
df = as.matrix(TCGA_PAAD$ex)
df = apply(df,2,as.integer)
rownames(df) = TCGA_PAAD$featInfo$SYMBOL
df = DESeq2::varianceStabilizingTransformation(df)
```

Next, estimate your ideal rank using the built in cophenetic correlation coefficient plot generator. The generator will highlight up to two key ranks: the maximum rank will be highlighted in red, and the rank directly before the steepest slope to the next rank (before the first positive slope is witnessed) will be highlighted in cyan. If both of these ranks are the same rank, it will be highlighted in purple. If the range of ranks you selected is not a consecutive set of numbers, only the maximum rank will be highlighted.

```r
cophenetic_generator(df, rank_range = 2:10, .options = 'p4')
```



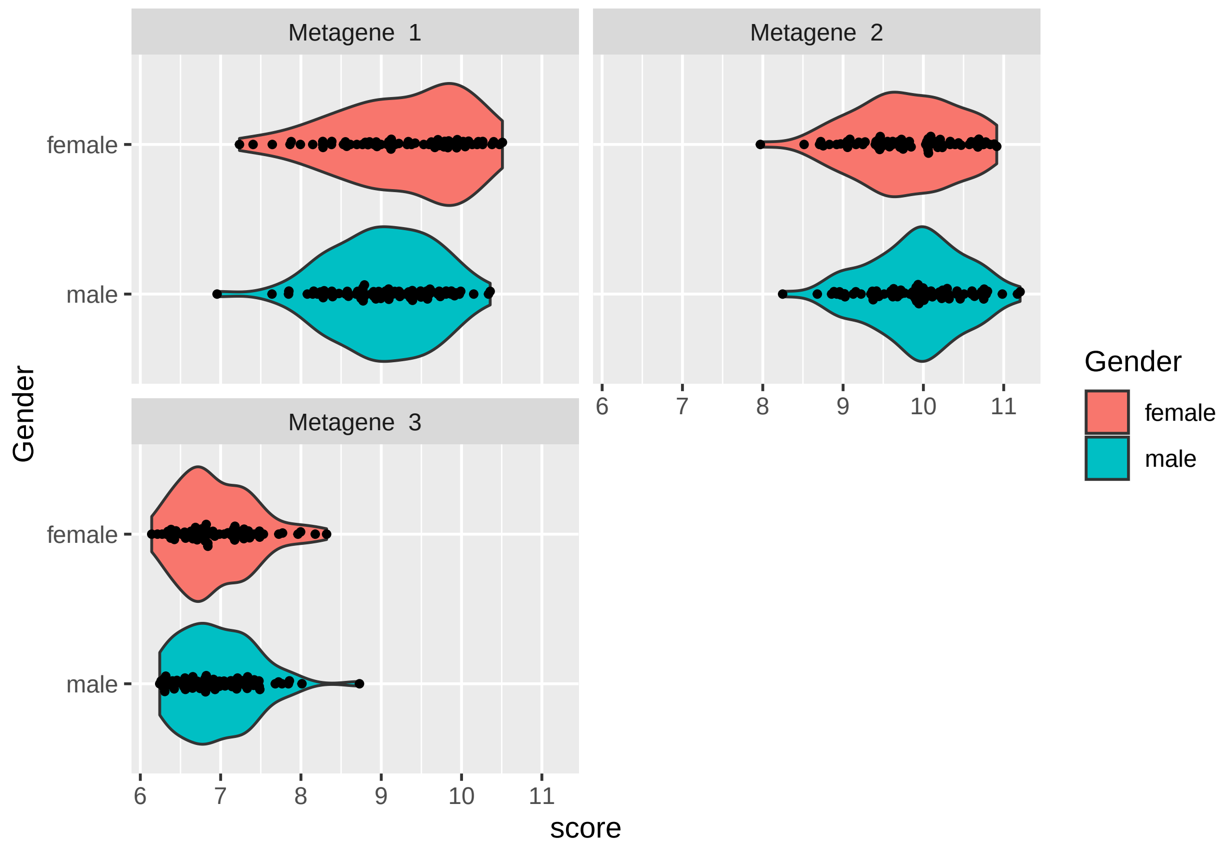
Run NMF using aged()

The call to aged() will likely produce a variety of warnings regarding gene mappings. This is normal as not all genes will be able to be mapped perfectly. Such warnings are muted in this vignette.

```
x = aged::aged(data = df,
               rank = 3,
               .options = 'p4',
               # transformation_type = 'vst',
               category = "H",
               gsea_barcodes = F)
## [1] "Performing NMF with rank 3..."
## [1] "Normalizing NMF results..."
## [1] "Calculating barcode genes for each metagene..."
## [1] "Starting GSEA..."
## [1] "Number of significant sets: 4"
## [1] "Number of significant sets: 9"
## [1] "Number of significant sets: 12"
```

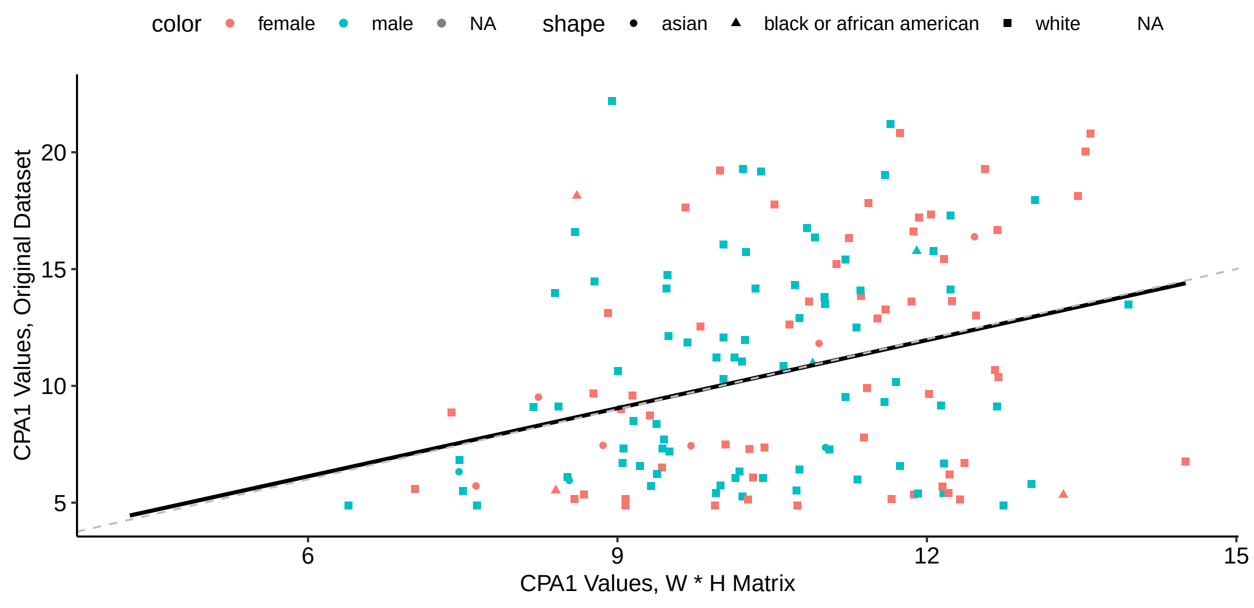
Visualize metagenes by gender in original dataset using violin_generator

```
aged::violin_generator(aged_results = x,
                data = df,
                batch = TCGA_PAAD$sampInfo$Gender,
                var_axis = "Gender")
```



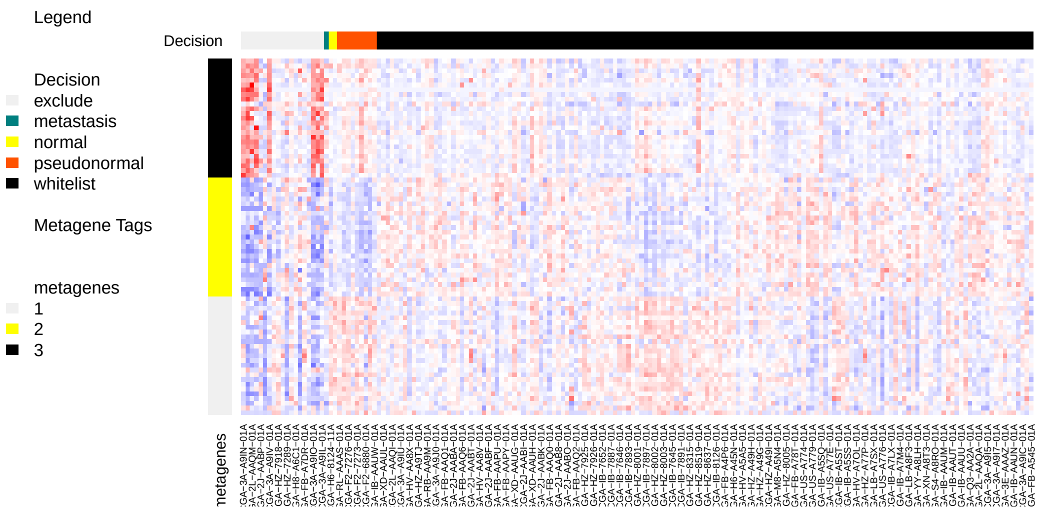
Visualize expression of specific gene by gender and race using the scatterplot generator

```
aged::scatterplot_generator(aged_results = x,
                            data = df,
                            gene = "CPA1",
                            y_axis = "wh",
                            color = TCGA_PAAD$sampInfo$Gender,
                            shape = TCGA_PAAD$sampInfo$Race,
                            ellipse = FALSE)
## Warning: Removed 35 rows containing missing values (geom_point).
```



Visualize metagene expression by batch, treatment, decision, etc. using the heatmap generator

```
aged::heatmap_generator(aged_results = x,
                        data = df,
                        samp_info = TCGA_PAAD$sampInfo,
                        batches = c("Decision"),
                        specific_order = c("Decision"))
```


Legend
Decision
Decision
exclude
metastasis
normal
pseudonormal
whitelist
Metagene Tags
metagenes
metagenes