Module 4.2: Learn - Coding

Overview

Having run the differential gene expression pipeline on the untrimmed data, this week we will run it on the trimmed data. This section is a guide to help you do this. Your assigned trimming data set will have a gene counts file in the geneCounts subdirectory of our class shared drive on Agave.

Steps to run the differential expression pipeline with your trimming data set

Step 1: Copy differential expression pipeline code into Week 4 directory

It is good practice to keep data and code used to generate that data in one directory so that you can be certain of what steps were done when you come back and look at the analysis much later. In your code directory `/home/<ASUrite_id>/CURE_2022_Scripts`, make a copy of `DE_Pipeline_UntrimmedData.Rmd` from the `Week_3` folder to the `Week_4` folder and change the name to `DE_Pipeline_TrimmedData.Rmd`.

Navigate to your scripts directory:

```
cd /home/<ASUrite_id>/CURE_2022_Scripts
```

List contents of `Week_3` directory and make sure you see the code for analyzing the untrimmed data from the previous module `DE_Pipeline_UntrimmedData.Rmd`:

```
ls Week_3
```

Make a copy of that code from the `Week_3` directory to `Week_4` subdirectory:

```
cp Week_3/DE_Pipeline_UntrimmedData.Rmd Week_4
```

List contents of Week_4 directory to see if the copy worked and if so navigate to that directory:

```
ls Week_4
```

```
cd Week_4
```

Rename code to state that it will be changed to run the trimmed data:

```
mv DE_Pipeline_UntrimmedData.Rmd DE_Pipeline_TrimmedData.Rmd
```

It would certainly be possible to modify the input files in the same Rmd as we worked with in Week_3, but by cop
new directory, we make sure that we don't overwrite anything if we accidentally forget to change output file name
files will be put in the directory you run the Rmd from by default.

Step 2: Change the author to include yourself in the Rmd header

Begin an RStudio session from Agave and open up your newly created DE_Pipeline_UntrimmedData.Rmd u
At the top of the Rmd, add your name to the author section. We want you to add code to this Rmd to investigate
data; it's important to note when you have made a contribution to code.

Step 3: Describe your objectives

It is good practice to indicate the purpose of your code. A good way to do that in an Rmd is to add a description s
will see that we have added descriptions between chunks of R code which each section marked with a ## for He
Add a new description section stating that this script will be used to analyze the trimmed data so you state the ob
right from the start. You can indicate important details like what trimmed data you were assigned and any import
information you think will be helpful for anyone looking at this code in the future (even yourself! It's easy to forget

Step 4: Set paths to input data

In the `WorkingDirectory` chunk, you will need to change the working directory to your `Week_4` subdirectory w
data code is and where your output files will be stored.

The input_directory variable will need to be set to the path of the directory containing the gene counts matrix file y
trimming data in the class shared drive For example, the path to the gene counts matrix for the untrimmed data

```
/data/mwilsons/GenomicsResearchExperience/Placenta_SexDiff/geneCounts/trimq_NONE_m
```

The `stats/` directory will have to be changed to match the location of your assigned trimming data set.

You can see that this path will be used in the `DataSetup` chunk to fill the counts variable using the `geneCounts`
the assigned data directory you set. The pheno variable has been set to the full path to the untrimmed data and
because the sample name and genetic sex contained in this file should stay the same regardless of data trimmin

Step 5: Change the result file label

At the end of the `DataSetup` chunk, there is a place to set a label that will be in all of your result files. Change tl
to your assigned trimming parameter data set. Please use the same `trimq_XXX_minlen_YYY` format we did w
sets during data generation.

Step 6: Run the pipeline

From here you should be able to go through the differential expression pipeline with the trimmed data just as we
data. Because our input data is formatted in just the same way, we can process it in exactly the same way.   You
figures, lists, and output files generated in the `Week_4` subdirectory for the trimmed data as you did for the untrim
this is unclear, please ask for help using Slack.

Steps to compare results from trimmed and untrimmed data

Now that you have generated a differentially expressed gene list using your trimmed data, we can start to use the
learned about in the first section of this module to compare the results from the untrimmed data. We have provid
template code `CompareTwo.Rmd in the Week_4` subdirectory to get started. We expect that you will chang

your personal copy of this Rmd but we hope that the code we have provided will serve as examples to help you s
own.

While developing this code, we created a fake dataset by adding random numbers to the untrimmed data output
do the differential expression analysis on the real trimming data sets. You will need to replace this fake data file r
data files in your `/home/<ASUrite_id>/CURE_2022_Scripts/` directory. Feel free to keep the fake data in t
learning purposes if you would like, but make sure that you are using your real trimmed data differential gene exp
get the real results.

Setting your input directories [*SetDirectory* chunk]

In addition to setting your working directory to `Week_4`, you will need to specify where the results from your analy
untrimmed data results should be in your `Week_3` directory and your trimmed data results should be in `Week_4`;
accordingly.

Measuring the amount of reads trimmed away [*CompareNumReads* chunk

In this chunk, we will compare the number of reads using output from the `flagstat` option of `samtools` which v
command line during data generation. This output will be in the stats subdirectory of the class shared data drive i
for geneCounts with a directory for each data set:

```
/data/mwilsons/GenomicsResearchExperience/Placenta_SexDiff/stats/
```

As you run the code in this chunk, you will read the stats files and extract the number of reads for each sample. V
average number of reads across the samples before and after trimming and use that to calculate a percent loss a
have greater than 25% loss of sequence reads for this data which we know to be of good quality, this is evidence
values of `trimq` and `minlen` are too stringent and that this might negatively impact the identification of differenti

Storing lists of differentially expressed genes [*UntrimmedDEG* and *Trimme*

In these modules we will use subsetting in R to apply thresholds to multiple hypothesis corrected p-values (adjus
change between female and male placentas to store lists of differentially expressed genes so we can compare th
our differential expression pipeline contain these values for differences between females versus males, so we wil

files as tables and analyze them in this follow-up code. For our differentially expressed gene lists, we filter for ge
p-value threshold of less than 0.05. Since log fold change was calculated as females divided by males, we know
change (> 0) indicates higher expression in female placentas and a negative log fold change (< 0) indicates high
placentas.

Comparing the number of differentially expressed genes in untrimmed vers [*DEGBarGraph* chunk]

To simply look at the number of differentially expressed genes identified in the trimming data set versus the untrir
use a bar graph. Here we simply use the length function in R to get the number of genes in the list that we store
create a bar plot. As we have seen, you can play around with other plotting types with `ggplot` to see if there are
displaying the comparison between untrimmed and trimmed results.

Venn diagram of Overlap between lists of differentially expressed genes [*V*

In this chunk, we use the Reduce function to calculate the overlap (intersection) between differentially expressed
ggVennDiagram function in the ggVennDiagram package to draw the Venn diagram. This function automatically
diagram to show where most of the genes fall in red (common to both untrimmed and trimmed or unique to one c

Visualizing changes in fold change after trimming [*FCchange* chunk]

As a way of looking at how fold change between females and males is affected by trimming, we can plot fold cha
in both conditions and use lines to show the direction of change. In this chunk, we include a small list of genes a
add gene names to this list if you would like to plot more (make sure that the gene name is present in the data by
data objects). After finding the fold change of the gene from the untrimmed and trimmed data sets, we use `ggpl`
scatter plot with the x-axis representing the two data sets and `geom_line` to add lines to connect the values of t
data sets. If we see more lines pointing up, this would indicate that fold change of those selected genes tends to
trimming, for example. Again, you can use other ways of visualizing this information, but this was just a suggesti

Correlation between fold change using untrimmed versus trimmed [*Correla*

To visualize the correlation between fold change of female to male average expression, this chunk shows how to
both data sets in a scatter plot and add a correlation line to it. This module will also show you how to do hypothe
`cor.test` function. The output for this function gives the coefficients we would need to accept the alternative hy
variables (female to male fold change from untrimmed and from trimmed) are related to one another.