Advanced Data Manipulation: Exercises

Exercise set 1

1. Display the data where the gene ontology biological process (the `bp` variable) is "leucine biosynthesis" (case-sensitive) *and* the limiting nutrient was Leucine. (Answer should return a 24-by-7 data frame – 4 genes × 6 growth rates).
2. Gene/rate combinations had high expression (in the top 1% of expressed genes)? *Hint:* see `?quantile` and try `quantile(ydat$expression, probs=.99)` to see the expression value which is higher than 99% of all the data, then `filter()` based on that. Try wrapping your answer with a `View()` function so you can see the whole thing. What does it look like those genes are doing? Answer should return a 1971-by-7 data frame.

Exercise set 2

1. First, re-run the command you used above to filter the data for genes involved in the "leucine biosynthesis" biological process *and* where the limiting nutrient is Leucine.
2. Wrap this entire filtered result with a call to `arrange()` where you'll arrange the result of #1 by the gene symbol.
3. Wrap this entire result in a `View()` statement so you can see the entire result.

Exercise set 3

Here's a warm-up round. Try the following.

1. Show the limiting nutrient and expression values for the gene ADH2 when the growth rate is restricted to 0.05. *Hint:* 2 pipes: `filter` and `select`.

```
## # A tibble: 6 × 2
##    nutrient expression
##       <chr>      <dbl>
## 1   Glucose       6.28
## 2   Ammonia       0.55
## 3 Phosphate      -4.60
## 4   Sulfate      -1.18
## 5   Leucine       4.15
## 6    Uracil       0.63
```

2. What are the four most highly expressed genes when the growth rate is restricted to 0.05 by restricting glucose? Show only the symbol, expression value, and GO terms. *Hint:* 4 pipes: `filter`, `arrange`, `head`, and `select`.

```
## # A tibble: 4 × 4
##   symbol expression                  bp                             mf
##    <chr>      <dbl>               <chr>                          <chr>
## 1   ADH2       6.28       fermentation* alcohol dehydrogenase activity
## 2  HSP26       5.86 response to stress*       unfolded protein binding
## 3   MLS1       5.64    glyoxylate cycle       malate synthase activity
## 4   HXT5       5.56    hexose transport  glucose transporter activity*
```

3. When the growth rate is restricted to 0.05, what is the average expression level across all genes in the "response to stress" biological process, separately for each limiting nutrient? What about genes in the "protein biosynthesis" biological process? *Hint:* 3 pipes: `filter`, `group_by`, `summarize`.

```
## # A tibble: 6 × 2
##    nutrient meanexp
##       <chr>   <dbl>
## 1   Ammonia   0.943
## 2   Glucose   0.743
## 3   Leucine   0.811
## 4 Phosphate   0.981
## 5   Sulfate   0.743
## 6    Uracil   0.731
```

```
## # A tibble: 6 × 2
##    nutrient meanexp
##       <chr>   <dbl>
## 1   Ammonia  -1.613
## 2   Glucose  -0.691
## 3   Leucine  -0.574
## 4 Phosphate  -0.750
## 5   Sulfate  -0.913
## 6    Uracil  -0.880
```

Exercise set 4

That was easy, right? How about some tougher ones.

First, some review. How do we see the number of distinct values of a variable? Use `n_distinct()` within a `summarize()` call.

```
ydat %>% summarize(n_distinct(mf))
```

```
## # A tibble: 1 × 1
##   `n_distinct(mf)`
##              <int>
## 1             1086
```

1. Which 10 biological process annotations have the most genes associated with them? What about molecular functions? *Hint:* 4 pipes: `group_by`, `summarize` with `n_distinct`, `arrange`, `head`.

```
## # A tibble: 10 × 2
##                                                               bp     n
##                                                            <chr> <int>
## 1                                     biological process unknown   269
## 2                                           protein biosynthesis   182
## 3                             protein amino acid phosphorylation*    78
## 4                                          protein biosynthesis*    73
## 5                          cell wall organization and biogenesis*    64
## 6  regulation of transcription from RNA polymerase II promoter*    49
## 7                          nuclear mRNA splicing, via spliceosome    47
## 8                                                    DNA repair*    44
## 9                                           aerobic respiration*    42
## 10                                        ER to Golgi transport*    42
```

```
## # A tibble: 10 × 2
##                                         mf     n
##                                      <chr> <int>
## 1              molecular function unknown   886
## 2      structural constituent of ribosome   185
## 3                         protein binding   107
## 4                             RNA binding    63
## 5                        protein binding*    53
## 6                            DNA binding*    44
## 7            structural molecule activity    43
## 8                         GTPase activity    40
## 9  structural constituent of cytoskeleton    39
## 10          transcription factor activity    38
```

2. How many distinct genes are there where we know what process the gene is involved in but we don't know what it does? *Hint:* 3 pipes; `filter` where `bp!="biological process unknown" & mf=="molecular function unknown"`, and after `select`ing columns of interest, pipe the output to `distinct()`. The answer should be **737**, and here are a few:

```
## # A tibble: 737 × 3
##    symbol                                                              bp
##     <chr>                                                           <chr>
## 1    SFB2                                           ER to Golgi transport
## 2    EDC3                           deadenylylation-independent decapping
## 3    PER1                                   response to unfolded protein*
## 4   PEX25                         peroxisome organization and biogenesis*
## 5    BNI5                                                    cytokinesis*
## 6   CSN12 adaptation to pheromone during conjugation with cellular fusion
## 7   SEC39                                               secretory pathway
## 8    ABC1                                         ubiquinone biosynthesis
## 9   PRP46                          nuclear mRNA splicing, via spliceosome
## 10   MAM3                      mitochondrion organization and biogenesis*
## # ... with 727 more rows, and 1 more variables: mf <chr>
```

3. When the growth rate is restricted to 0.05 by limiting Glucose, which biological processes are the most upregulated? Show a sorted list with the most upregulated BPs on top, displaying the biological process and the average expression of all genes in that process rounded to two digits. *Hint:* 5 pipes: `filter`, `group_by`, `summarize`, `mutate`, `arrange`.

```
## # A tibble: 881 × 2
##                                                bp meanexp
##                                             <chr>   <dbl>
## 1                                   fermentation*    6.28
## 2                                glyoxylate cycle    5.29
## 3  oxygen and reactive oxygen species metabolism    5.04
## 4                             fumarate transport*    5.03
## 5                        acetyl-CoA biosynthesis*    4.32
## 6                                 gluconeogenesis    3.64
## 7                       fatty acid beta-oxidation    3.57
## 8                               lactate transport    3.48
## 9                            carnitine metabolism    3.30
## 10                            alcohol metabolism*    3.25
## # ... with 871 more rows
```

4. Group the data by limiting nutrient (primarily) then by biological process. Get the average expression for all genes annotated with each process, separately for each limiting nutrient, where the growth rate is restricted to 0.05. Arrange the result to show the most upregulated processes on top. The initial

result will look like the result below. Pipe this output to a `View()` statement. What's going on? Why didn't the `arrange()` work? *Hint:* 5 pipes: `filter`, `group_by`, `summarize`, `arrange`, `View`.

```
## Source: local data frame [5,257 x 3]
## Groups: nutrient [6]
##
##     nutrient                                           bp meanexp
##        <chr>                                        <chr>   <dbl>
## 1    Ammonia                         allantoate transport    6.64
## 2    Ammonia                        amino acid transport*    6.64
## 3  Phosphate               glycerophosphodiester transport    6.64
## 4    Glucose                                 fermentation*    6.28
## 5    Ammonia                          allantoin transport    5.56
## 6    Glucose                              glyoxylate cycle    5.29
## 7    Ammonia                         proline catabolism*    5.14
## 8    Ammonia                               urea transport    5.14
## 9    Glucose oxygen and reactive oxygen species metabolism    5.04
## 10   Glucose                          fumarate transport*    5.03
## # ... with 5,247 more rows
```

5. Let's try to further process that result to get only the top three most upregulated biolgocal processes for each limiting nutrient. Google search "dplyr first result within group." You'll need a `filter(row_number()......)` in there somewhere. *Hint:* 5 pipes: `filter`, `group_by`, `summarize`, `arrange`, `filter(row_number()...`. *Note:* dplyr's pipe syntax used to be `%.%` before it changed to `%>%`. So when looking around, you might still see some people use the old syntax. Now if you try to use the old syntax, you'll get a deprecation warning.

```
## Source: local data frame [18 x 3]
## Groups: nutrient [6]
##
##     nutrient                                           bp meanexp
##        <chr>                                        <chr>   <dbl>
## 1    Ammonia                         allantoate transport    6.64
## 2    Ammonia                        amino acid transport*    6.64
## 3  Phosphate               glycerophosphodiester transport    6.64
## 4    Glucose                                 fermentation*    6.28
## 5    Ammonia                          allantoin transport    5.56
## 6    Glucose                              glyoxylate cycle    5.29
## 7    Glucose oxygen and reactive oxygen species metabolism    5.04
## 8     Uracil                          fumarate transport*    4.32
## 9  Phosphate                vacuole fusion, non-autophagic    4.20
## 10   Leucine                                 fermentation*    4.15
## 11 Phosphate          regulation of cell redox homeostasis*    4.03
## 12   Leucine                          fumarate transport*    3.72
## 13   Leucine                              glyoxylate cycle    3.65
## 14   Sulfate                        protein ubiquitination    3.40
## 15   Sulfate                          fumarate transport*    3.27
## 16    Uracil                         pyridoxine metabolism    3.11
## 17    Uracil                         asparagine catabolism*    3.06
## 18   Sulfate                  sulfur amino acid metabolism*    2.69
```

6. There's a slight problem with the examples above. We're getting the average expression of all the biological processes separately by each nutrient. But some of these biological processes only have a single gene in them! If we tried to do the same thing to get the correlation between rate and expression, the calculation would work, but we'd get a warning about a standard deviation being zero. The correlation coefficient value that results is `NA`, i.e., missing. While we're summarizing the correlation between rate

and expression, let's also show the number of distinct genes within each grouping.

```
ydat %>%
  group_by(nutrient, bp) %>%
  summarize(r=cor(rate, expression), ngenes=n_distinct(symbol))
```

```
## Warning in cor(c(0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05,
## 0.05, : the standard deviation is zero
```

```
## Source: local data frame [5,286 x 4]
## Groups: nutrient [?]
##
##    nutrient                                      bp       r ngenes
##       <chr>                                   <chr>   <dbl>  <int>
## 1   Ammonia             'de novo' IMP biosynthesis*  0.3125      8
## 2   Ammonia  'de novo' pyrimidine base biosynthesis -0.0482      3
## 3   Ammonia 'de novo' pyrimidine base biosynthesis*  0.1670      4
## 4   Ammonia       35S primary transcript processing  0.5080     13
## 5   Ammonia      35S primary transcript processing*  0.4240     30
## 6   Ammonia                    acetate biosynthesis  0.4677      1
## 7   Ammonia                      acetate metabolism  0.9291      1
## 8   Ammonia                     acetate metabolism* -0.6855      1
## 9   Ammonia                  acetyl-CoA biosynthesis -0.8512      1
## 10  Ammonia   acetyl-CoA biosynthesis from pyruvate  0.0951      1
## # ... with 5,276 more rows
```

7. Take the above code and continue to process the result to show only results where the process has at least 5 genes. Add a column corresponding to the absolute value of the correlation coefficient, and show for each nutrient the singular process with the highest correlation between rate and expression, regardless of direction. *Hint:* 4 more pipes: `filter`, `mutate`, `arrange`, and `filter` again with `row_number()==1`. Ignore the warning.

```
## Source: local data frame [6 x 5]
## Groups: nutrient [6]
##
##    nutrient                                      bp     r ngenes  absr
##       <chr>                                   <chr> <dbl>  <int> <dbl>
## 1   Glucose telomerase-independent telomere maintenance -0.95      7  0.95
## 2   Ammonia telomerase-independent telomere maintenance -0.91      7  0.91
## 3   Leucine telomerase-independent telomere maintenance -0.90      7  0.90
## 4 Phosphate telomerase-independent telomere maintenance -0.90      7  0.90
## 5    Uracil telomerase-independent telomere maintenance -0.81      7  0.81
## 6   Sulfate                   translational elongation*  0.79      5  0.79
```