Assignment 11

Identify Expression-Modulating Variants

Bio5488 4/5/19
Link genotype to phenotype?

Expression quantitative trait loci (eQTLs)

- **Definition:** region w/ DNA sequence variants influence expression level of 1/more genes

- **Data collection for eQTL analysis:**
  - Genotype: SNP microarray / Whole genome sequencing
  - Expression: Expression microarray / RNA sequencing

- **Statistical test**
  - 1. Group individual by allele they carry
  - 2. Compare difference in expression between groups
Expression quantitative trait loci (eQTLs)

Classification & Interpretation

• eQTLs can have local / distant effect

• Possible causal relationships:

MPRA: high throughput direct identification

Tewhey, R. Cell.
Assignment 11: What to do?

• **Filter** out variants with insufficient barcode labeling

• **Count # barcodes** in both pDNA and cDNA - only count barcodes matched perfectly

• **Assign** barcode counts back to variants
• **Calculate** normalized expression level
• **Compare** the difference between reference and alternative alleles (log-transformed fold change)
• **Determine** which fold-changes are significantly different
Requirements

• Due **Friday (4/12/17)** before class

• Your submission folder should contain:
  • Scripts:
    filter_variants.py, count_barcodes.py, analyze_MPRA.py

  • Output files:
    filtered_variant_to_barcode.txt, pDNA_count.txt, cDNA_count.txt, variant_fold_change.txt
Unix Basics II
Count newline, word, and byte

\( (\text{wc}) \)

\text{wc} counts line, byte, character, word... in a file

\$ \text{cat} \ B.txt
Apple      Red
Banana     Yellow
Kiwi       Green

\$ \text{wc} \ -l \ B.txt
3

Count by line
Combine two files with one shared column (join)

• When you have two files each with one column storing same set of values, e.g.:

$ cat A.txt
Apple    Red
Banana   Yellow
Kiwi     Green
Orange   Orange

$ cat B.txt
Apple    Crispy
Banana   Sweet
Kiwi     Sour

• `join` will combine all columns from 2 files together

$ join -1 1 -2 1 -t $’\t’ A.txt B.txt

Apple    Red    Crispy
Banana   Yellow Sweet
Kiwi     Green   Sour

Tells `join` to use tab as delimiter
Replace or remove specific characters (tr)

- By default, output from `join` is delimited by space:
  ```
  $ join -1 1 -2 1 A.txt B.txt
  Apple Red Crispy
  Banana Yellow Sweet
  Kiwi Green Sour
  ```
  This looks ugly 😞

- `tr` replaces the 1st set in input with the 2nd set
  ```
  $ join -1 1 -2 1 A.txt B.txt | tr '' \t
  Apple Red Crispy
  Banana Yellow Sweet
  Kiwi Green Sour
  ```
  Pipe
  Replace space with tab
Pass output for another command as input ("|")

- **Pipe**: Pass the output of one command to another for further processing
  - What will the code below provide?
  ```bash
  $ join -1 1 -2 1 A.txt B.txt | head -n1 | tr ' ' '
' | wc -l
  ```
Print the lines in sorted order

(sort)

- Sorting all lines based on sort key extracted from each line
  - -k: by column of
  - -n: rank by number
  - -r: in reverse order

  **Hint:** `join` requires the two files to be combined sorted by the column with values to join.

- Say if we want to sort the joined file by the last column

  ```
  $ join -1 1 -2 1 -t $'\t' A.txt B.txt | sort -k3,3
  Apple  Red  Crispy
  Kiwi   Green Sour
  Banana Yellow Sweet
  ```
Happy coding!