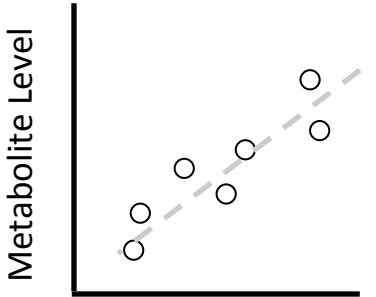
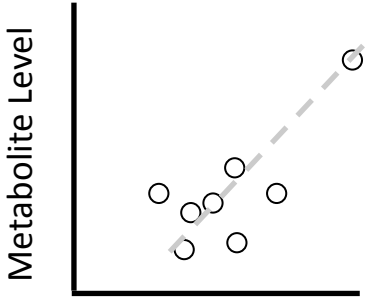
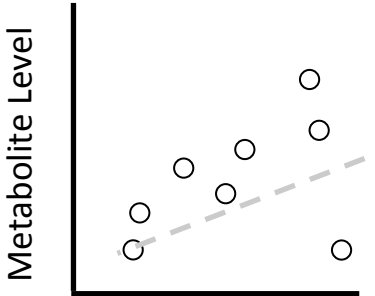
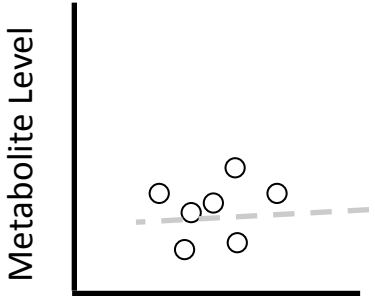
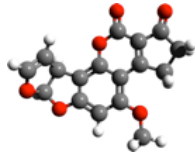


# Correlations Under the Influence (CUTIE)

Code Review 4.8.19

Kevin Bu

	True Correlation	False Correlation
Significant Correlation ( $p < 0.05$ )	<p>✓ True Positive (Non-CUTIE)</p>  <p>Metabolite Level</p> <p>Bacteria Abundance</p>	<p>✗ False Positive (CUTIE)</p>  <p>Metabolite Level</p> <p>Bacteria Abundance</p>
Insignificant Correlation ( $p > 0.05$ )	<p>✗ False Negative</p>  <p>Metabolite Level</p> <p>Bacteria Abundance</p>	<p>✓ True Negative</p>  <p>Metabolite Level</p> <p>Bacteria Abundance</p>



**Metabolite**

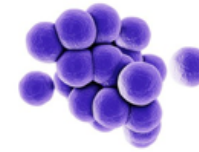
Glycine

Citrate

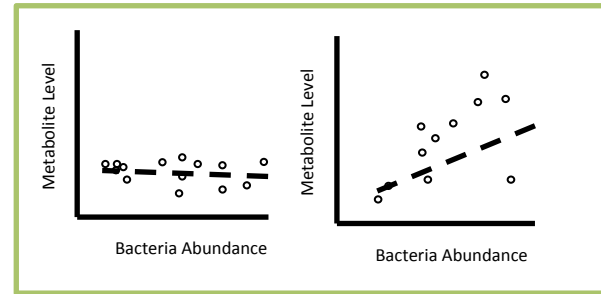
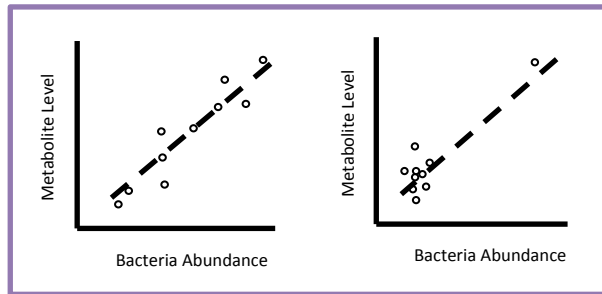
**Bacteria**

Ecoli

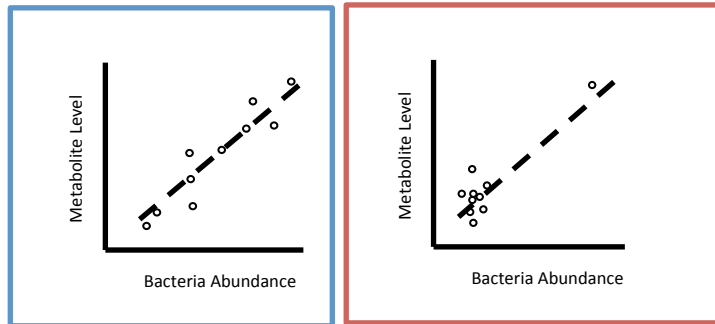
Shigella



...  
**Compute Pairwise Correlations**  
...



**Examine all correlations with  $p < 0.05^*$**

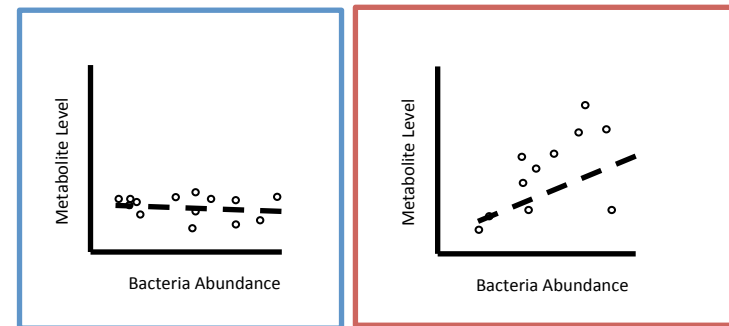


**Perform CUTIE analysis**

Differentiate **non-CUTle's (TP)** from **CUTle's (FP)**



**Examine all correlations with  $p > 0.05^*$**



**Perform rCUTIE analysis**

Differentiate **TN's** from **FN's**

## Milestone 1: Code Review 4.8.19

- (1) Tutorial (in the README)
- (2) Unit tests for all core functions

# CUTIE file structure

CUTie/

cutie/

\_\_init\_\_.py

output.py

parse.py

utils.py

statistics.py

README.md

setup.py

scripts/

calculate\_cutie.py

configs/

tests/

config\_defaults.ini

data/

Mapping.Pneumotype.Multiomics.RL.NYU.w\_metabolites.w\_inflamm.txt

otu\_table\_MultiO\_merged\_\_\_L6.txt

n=50,alpha=0.6.csv

lungpt\_1pc\_point\_unit\_test\_kkc1fdr0.05/

# Issues

- 1. Parse.py (Nan)
  - Check PEP8 compliance (md5Checksum, parse\_input, process\_df, read\_taxa, parse\_minep, parse\_config)
- 2. Output.py (Zach)
  - 2a. Unit tests for generation of matrices (print\_matrix, print\_Rmatrix, print\_true\_false\_corr, report\_results, generate\_pair\_matrix)
  - 2b. Check for PEP8 compliance, potentially shorten graphing functions (graph\_subsets, generate\_dfs, plot\_dfs, plot\_figure, plot\_pdist, plot\_logp\_and\_logpfold, plot\_corr, plot\_corr\_sets, diag\_plots)
  - 2c. Unit tests for log handling (init\_log, write\_log)
- 3. Utils.py (David)
  - Unit tests for auxiliary functions (return\_indicators, remove\_nans, initialize\_stat\_dicts, get\_param, calculate\_intersection)
- 4. Statistics.py (4a: David, 4b-e: Kevin)
  - 4a. Unit tests for zero-handling (multi\_zeros, multi\_replacement, zero\_replacement, log\_transform)
  - 4b. Unit tests for initial stats (assign\_statistics, initial\_stats\_SLR, initial\_stats\_MINE, set\_threshold, indicator, get\_initial\_corr)
  - 4c. Unit tests for pointwise stat functions (resample1\_cutie\_pc, resample1\_cutie\_sc, cookd, dffits, dsr, return\_influence, pointwise\_comparison, calculate\_FP\_sets, compute\_pc, compute\_sc, compute\_kc)
  - 4d. Unit tests for MINE-specific handling (compute\_mine, str\_to\_pvalues, binarySearchBins)
  - 4e. Unit tests for CUTIE, bootstrap, and jackknife functions (updatek\_cutie, cutiek\_true\_corr, evaluate\_correlation\_k, update\_rev\_extrema\_rp, init\_var\_indicators, jackknifek\_cutie, bootstrap\_cutie, resamplek\_cutie, get\_pCI, test\_CI)