“A Novel Sparse Compositional Technique Reveals Microbial Perturbations”

(Published in February 12th, 2019)

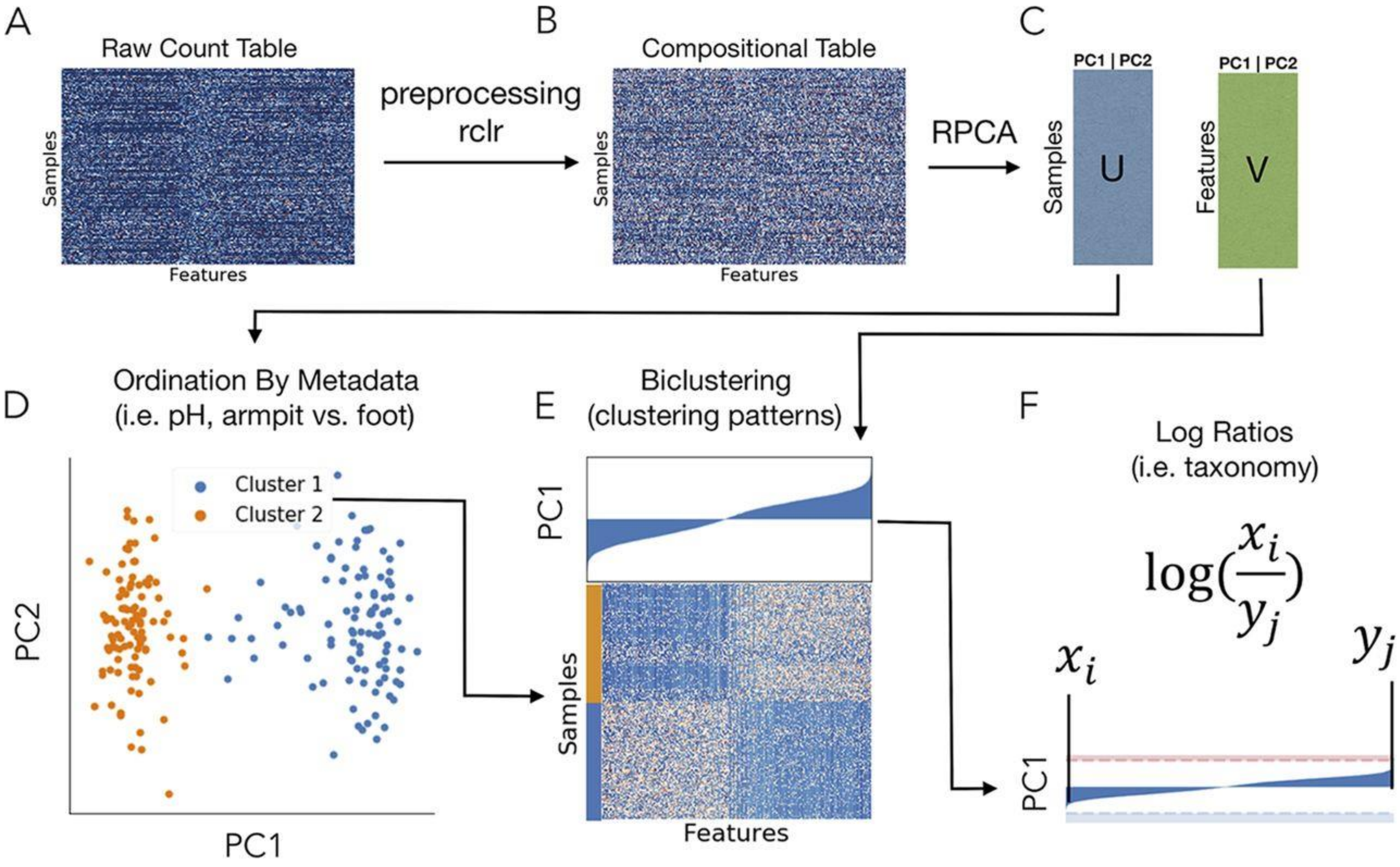
Authors: Cameron Martino, James T. Morton, Clarisse A. Marotz, Luke R. Thompson, Anupriya Tripathi, Rob Knight, Karsten Zengler

Presenter: Itamar Curiel

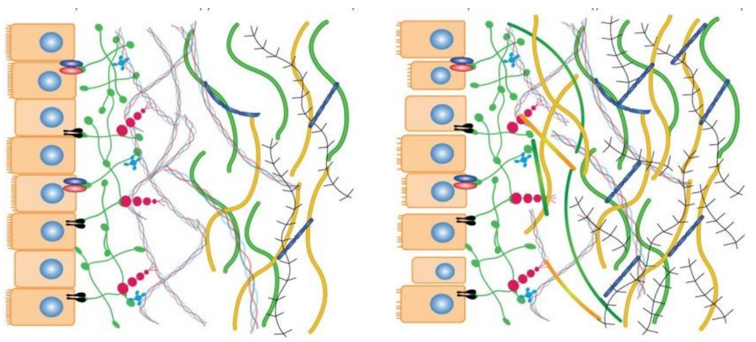
A Novel Sparse Compositional Technique Reveals Microbial Perturbations

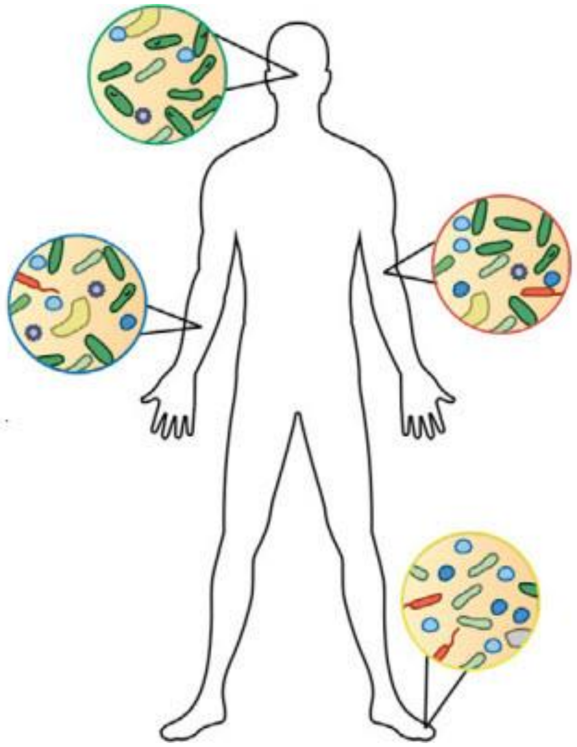
ANSCTRMP?

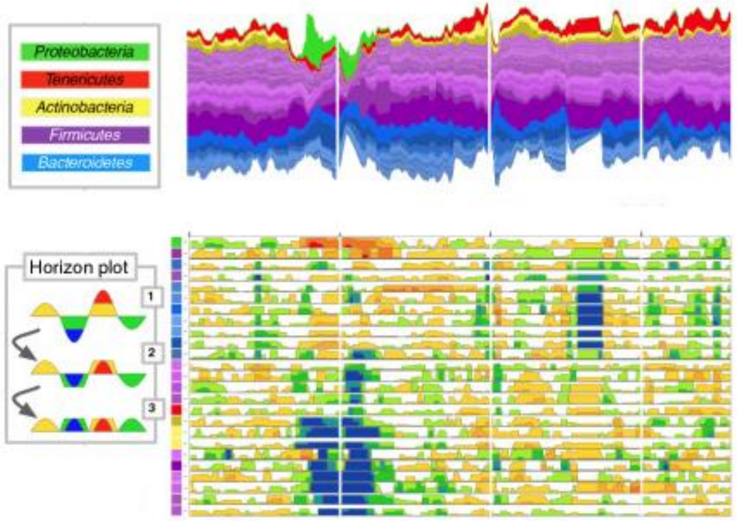
TL;DR

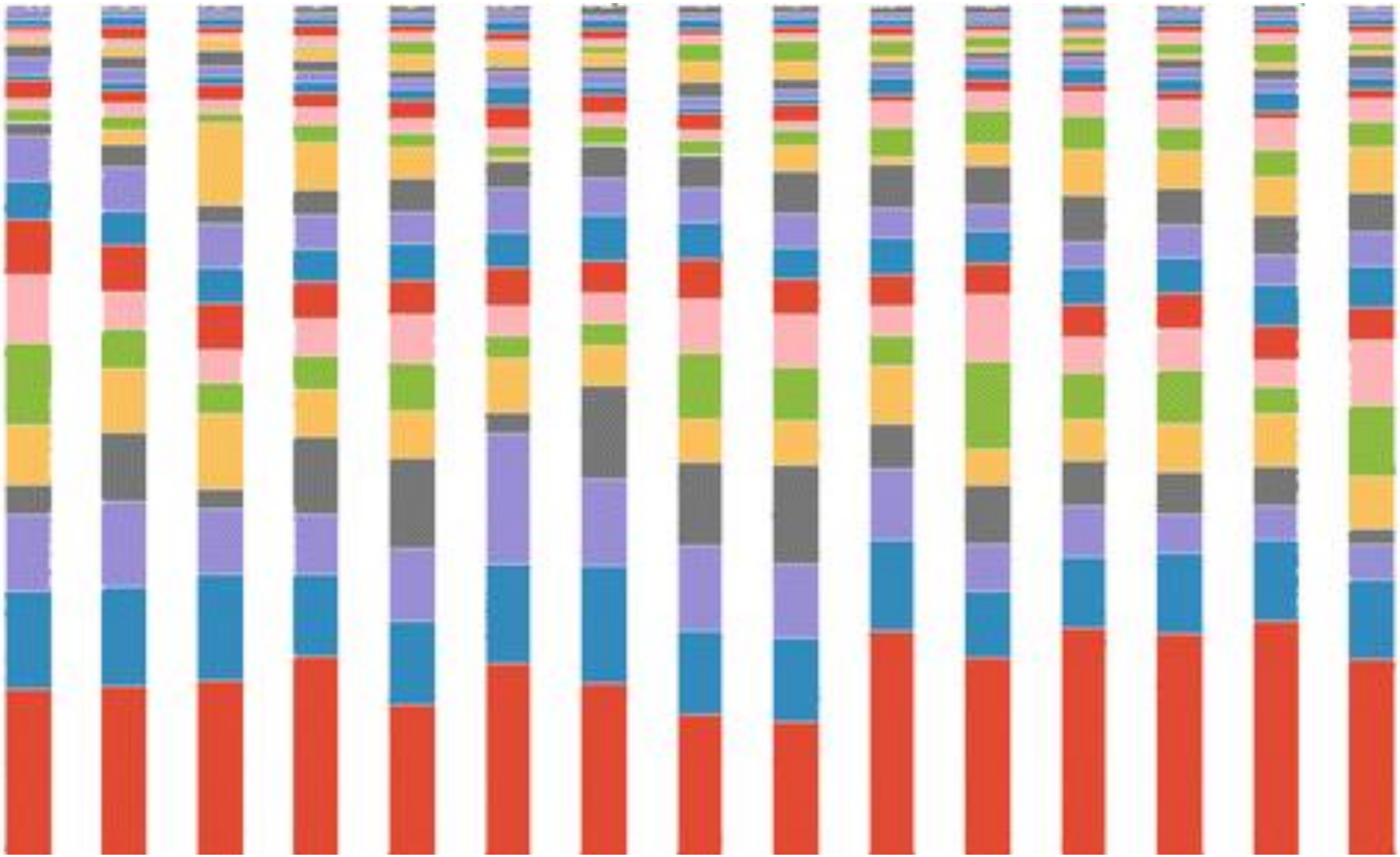


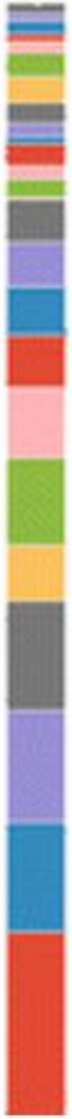
A
Raw Count Table
Samples
Features
preprocessing
rclr
B
Compositional Table
Samples
Features
RPCA
C
PC1 | PC2
Samples
U
PC1 | PC2
Features
V
D
Ordination By Metadata
(i.e. pH, armpit vs. foot)
Cluster 1
Cluster 2
PC2
PC1
E
Biclustering
(clustering patterns)
PC1
Samples
Features
F
Log Ratios
(i.e. taxonomy)
$\log(\frac{x_i}{y_j})$
x_i
y_j
PC1






Proteobacteria
Tenericutes
Actinobacteria
Firmicutes
Bacteroidetes
Horizon plot
1
2
3





	SPECIES	FRACTION
■	Bacillus Thuringiensis	0.001
■	Campylobacter jejuni	0.001
■	Staphylococcus epidermidis	0.002
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	…	…
■	Methanobacterium extroquens	0.076
■	Bordetella bronchiseptica	0.076
■	Propionibacterium propionicus	0.122

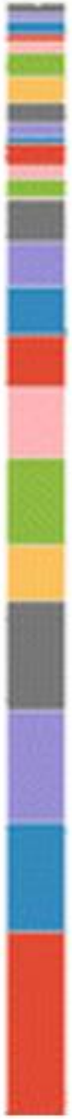
$$x' = [0.003, 0.001, 0.002, \dots, 0.081, 0.078, 0.122]$$

(species found in this sample)

$$x = [0.003, \mathbf{0}, \mathbf{0}, \mathbf{0}, 0.001, \mathbf{0}, 0.002, \dots, 0.122, \mathbf{0}, \mathbf{0}]$$

(all species)

- All non-negative!
- Values add up to 1!
- May contain zeros!

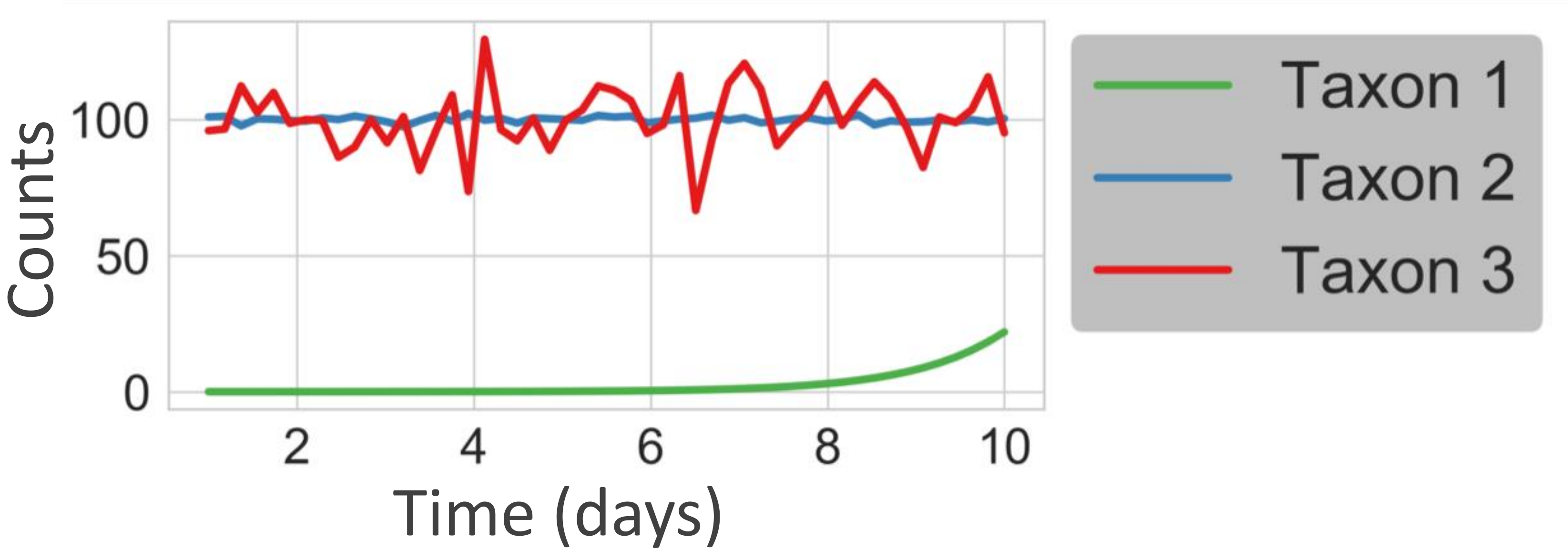


		SPECIES	FRACTION
x_1	🟦	Bacillus Thuringiensis	0.003
x_2	🟥	Campylobacter jejuni	0.001
x_3	🩷	Staphylococcus epidermidis	0.002
⋮	🟩	…	…
⋮	🟧	…	…
⋮	⬛	…	…
⋮	🟪	…	…
⋮	🟦	…	…
⋮	🟥	…	…
⋮	🩷	…	…
⋮	🟩	…	…
⋮	🟧	…	…
⋮	⬛	…	…
⋮	🟪	…	…
⋮	🟦	…	…
⋮	🟥	…	…
⋮	🩷	…	…
⋮	🟩	…	…
⋮	🟧	…	…
⋮	⬛	…	…
⋮	🟪	Methanobacterium extroquens	0.081
⋮	🟦	Bordetella bronchiseptica	0.078
x_n	🟥	Propionibacterium propionicus	0.122

The Problems

- High-dimensionality – our vectors are long
- Compositionality – if one species goes up, the rest go down
  - Non-normality - lack of a good “anchor” species/sample to fix compositionality
- Sparsity – lots of 0s in our data
  - Because species does not exist
  - Because species exists in small amounts and we undersampled
  - Because there’s too much of the other species
  - Because of an error in the process
- Some species are very impactful even in small amounts

Example



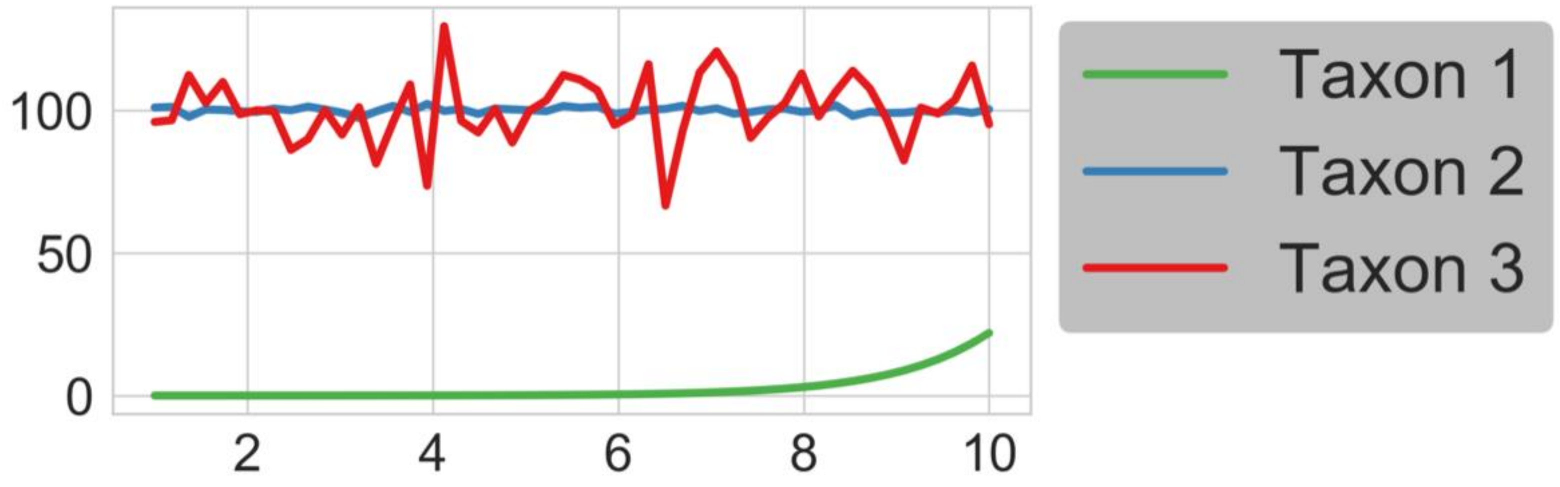
Taxon 1 starts very low but increases exponentially over time

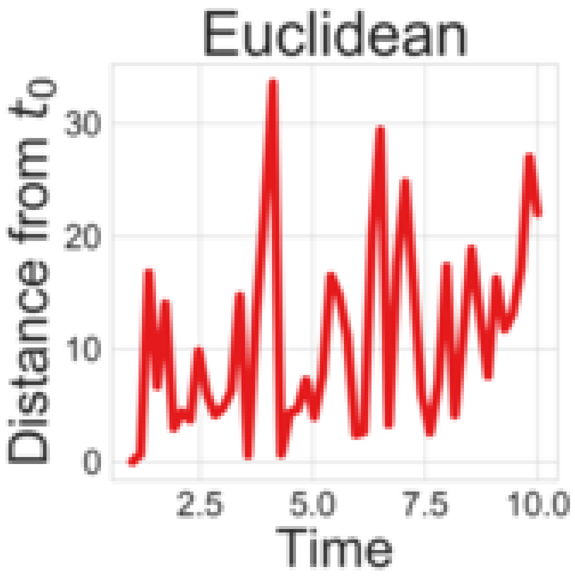
Taxon 2 starts high and remains there, stable

Taxon 3 starts high and remains there, but very unstable

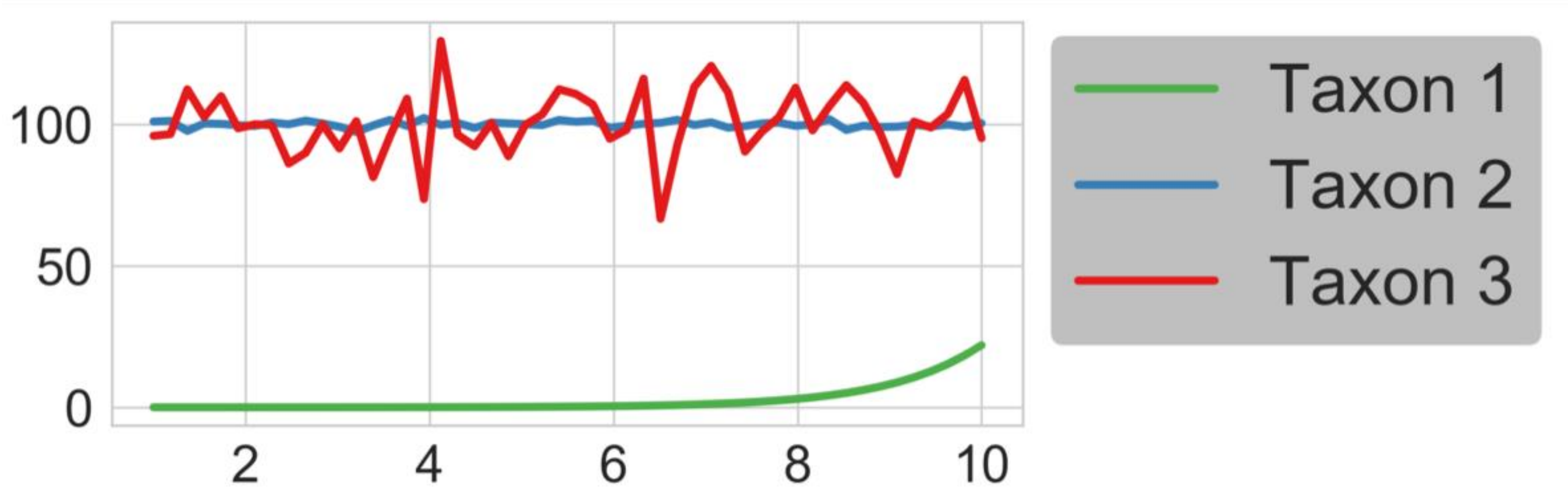
We want to notice the trend of Taxon 1 increasing!

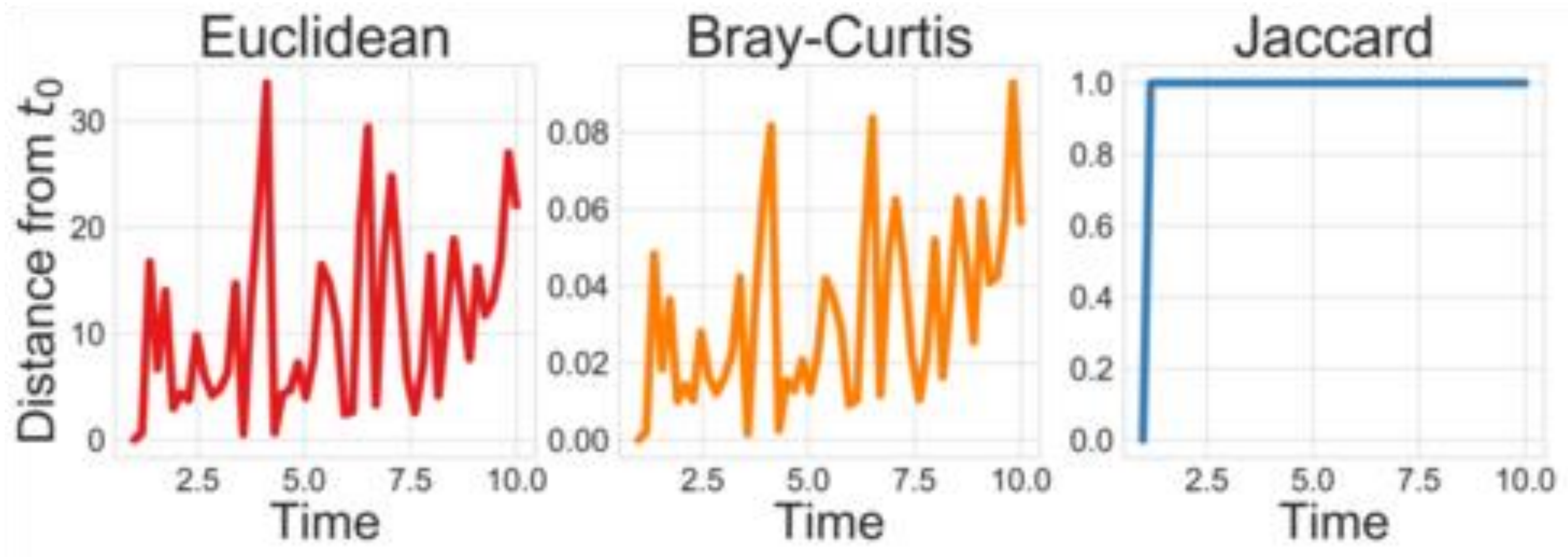
Example



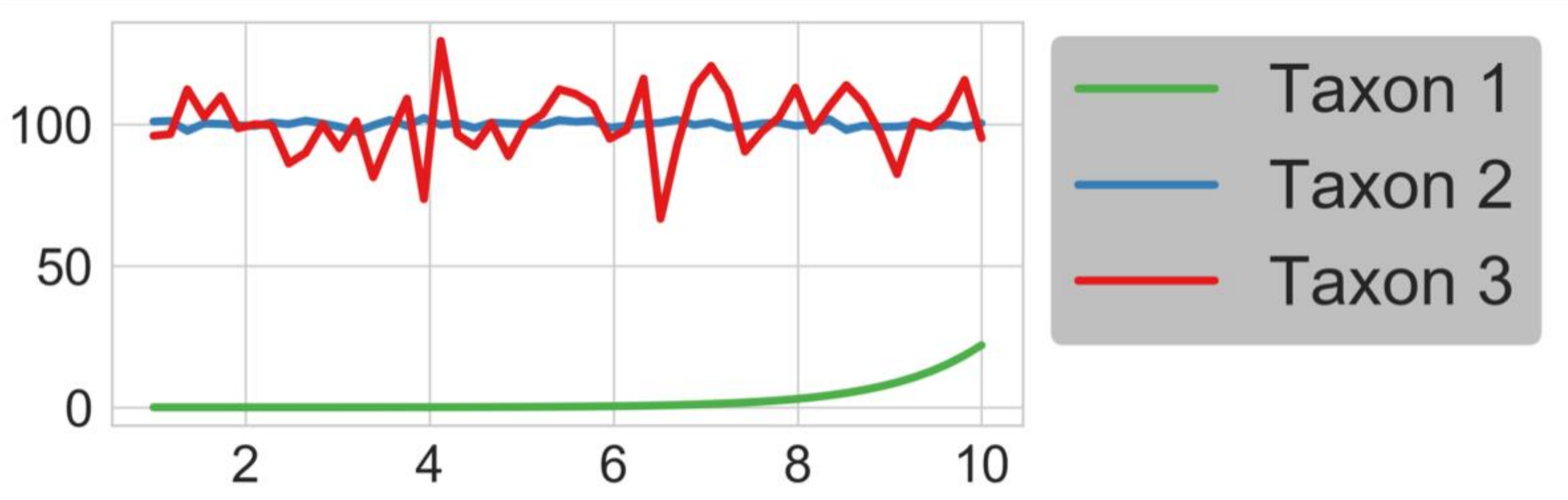

Euclidean

Example

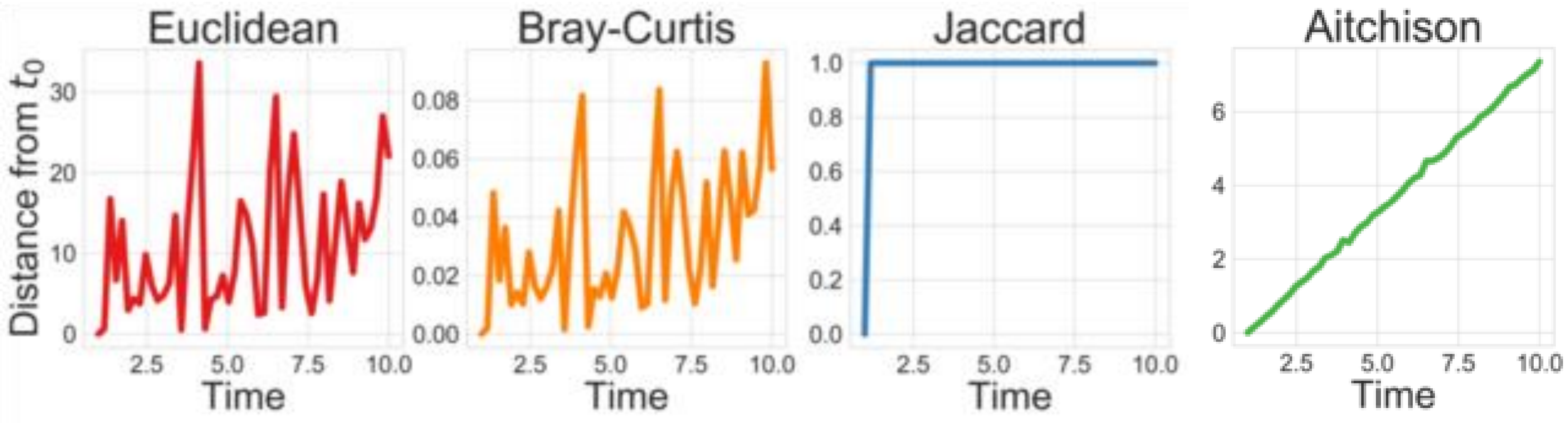

100
50
0
2
4
6
8
10
Taxon 1
Taxon 2
Taxon 3


Euclidean
Bray-Curtis
Jaccard
Distance from t_0
30
20
10
0
0.08
0.06
0.04
0.02
0.00
1.0
0.8
0.6
0.4
0.2
0.0
2.5
5.0
7.5
10.0
Time
Time
Time

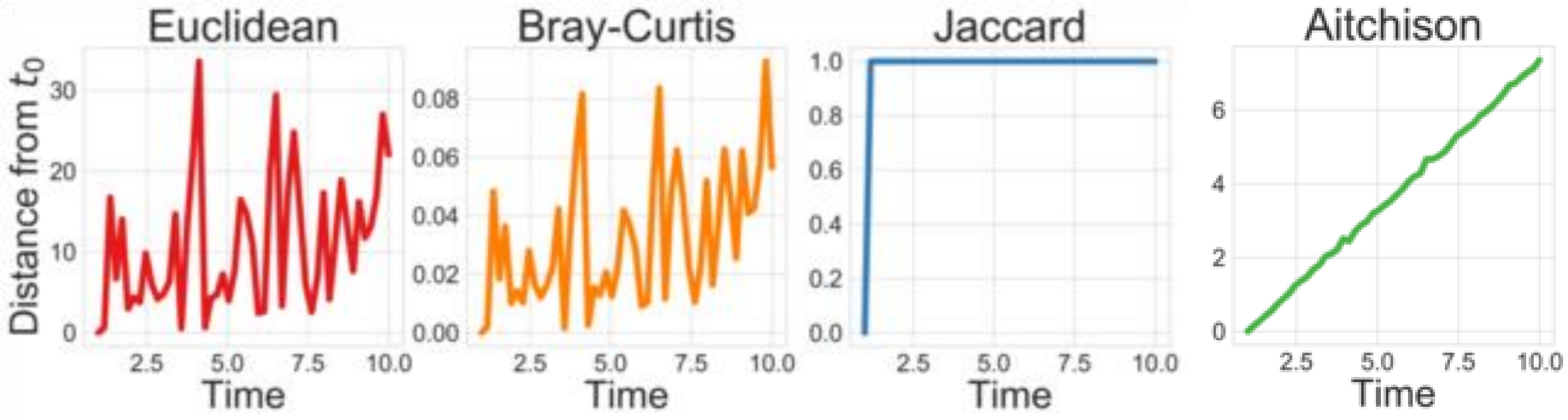
Example



100
50
0
2
4
6
8
10
Taxon 1
Taxon 2
Taxon 3



Euclidean
Distance from t_0
30
20
10
0
2.5
5.0
7.5
10.0
Time
Bray-Curtis
0.08
0.06
0.04
0.02
0.00
2.5
5.0
7.5
10.0
Time
Jaccard
1.0
0.8
0.6
0.4
0.2
0.0
2.5
5.0
7.5
10.0
Time
Aitchison
6
4
2
0
2.5
5.0
7.5
10.0
Time



$$\sqrt{\sum_{k=1}^{d}(x_k-y_k)^2}$$

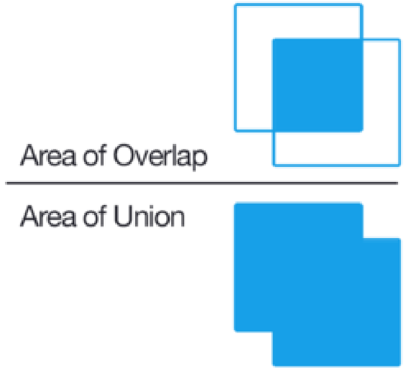
(L2 norm of the delta)

$$1-\sum_{k=1}^{d}\min(x_k,y_k)$$

(sum of lesser values for each species, assuming unit vectors. $\sum x_k = 1$)

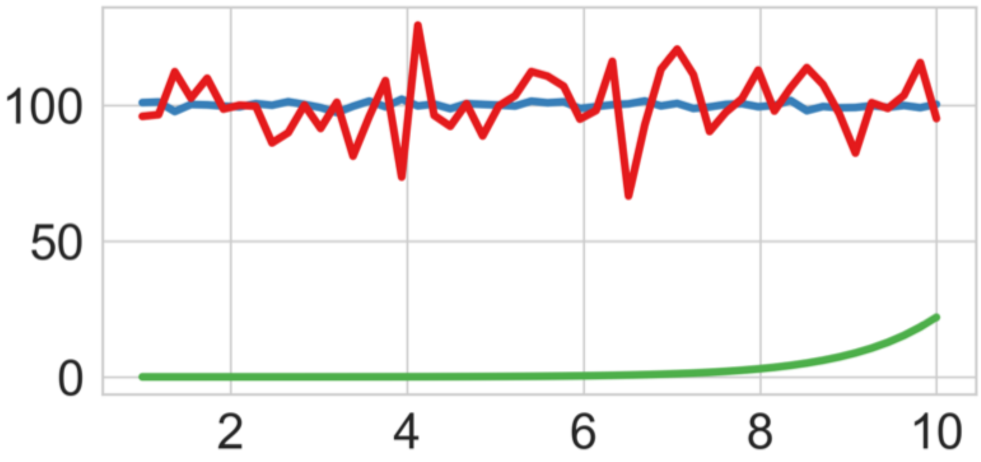
(transforms data into binary yes/no)

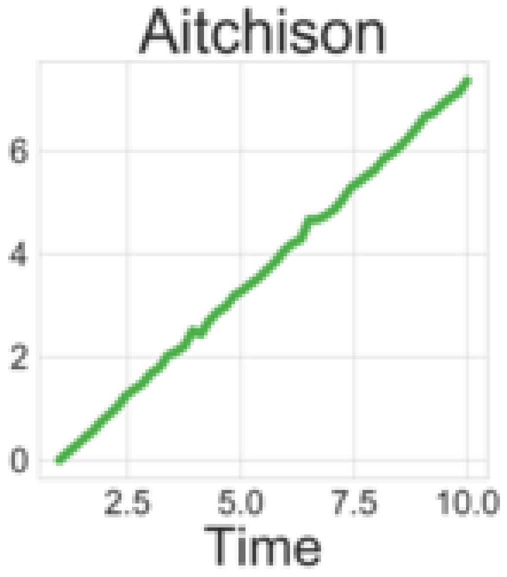
$$1-\frac{\sum_{k=1}^{d}\delta_{x_k=y_k}}{d}$$



$$\sqrt{\sum_{k=1}^{d}\left(\log\frac{x_k}{g(x)}-\log\frac{y_k}{g(y)}\right)^2}$$

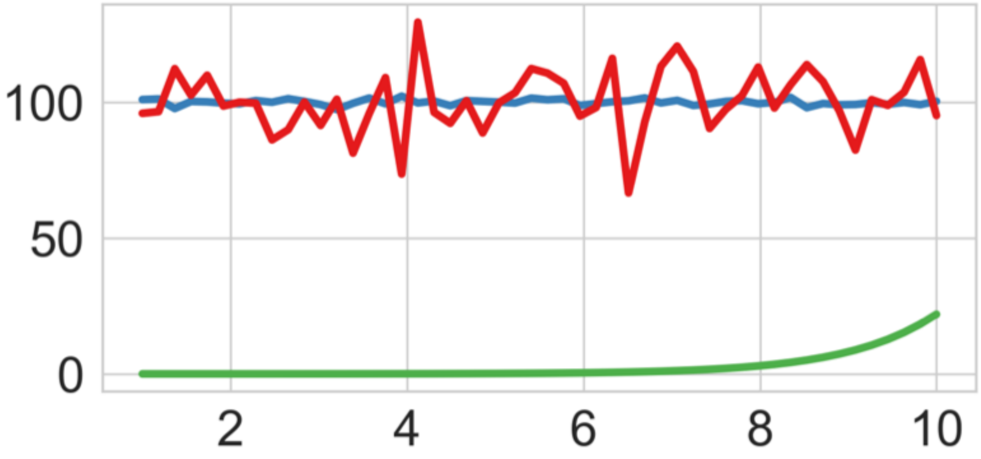
$$g(x)=\sqrt[d]{\prod_{k=1}^{d}x_k}$$

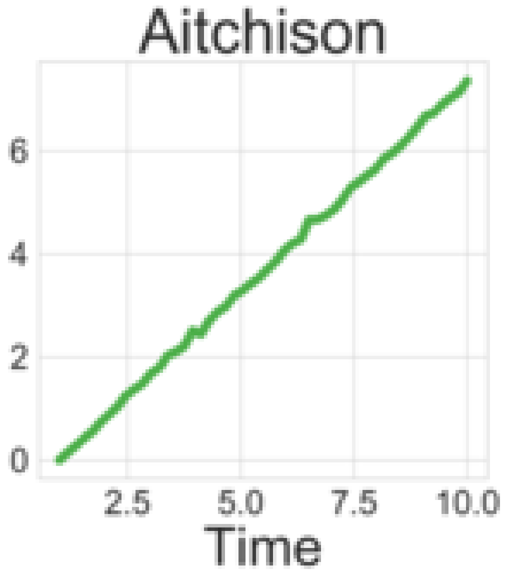




$$\sqrt{\sum_{k=1}^{d}\left(\log\frac{x_k}{g(x)}-\log\frac{y_k}{g(y)}\right)^2}$$

$$g(x)=\sqrt[\mathrm{d}]{\prod_{k=1}^{d}x_k}$$





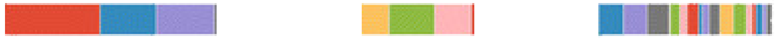
$$\sqrt{\sum_{k=1}^{d}\left(\log\frac{x_k}{g(x)}-\log\frac{y_k}{g(y)}\right)^2}$$

- Scale invariant!

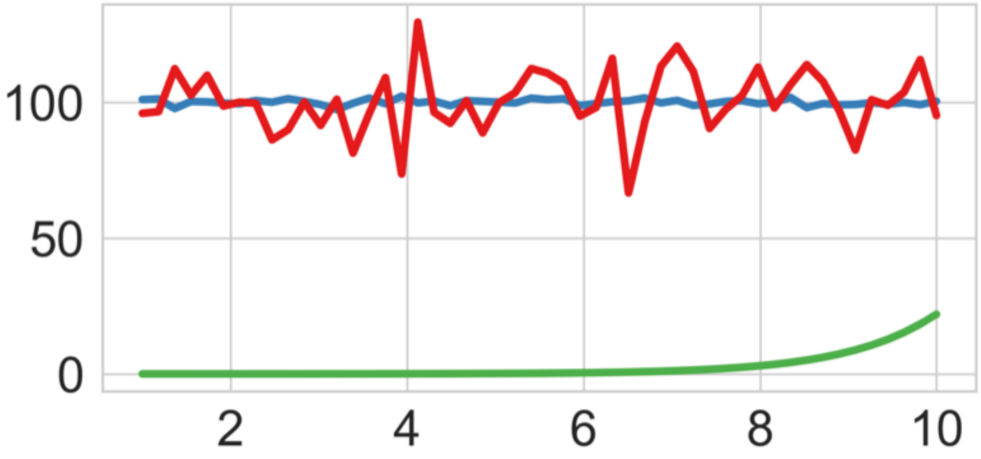
- Perturbation invariant!

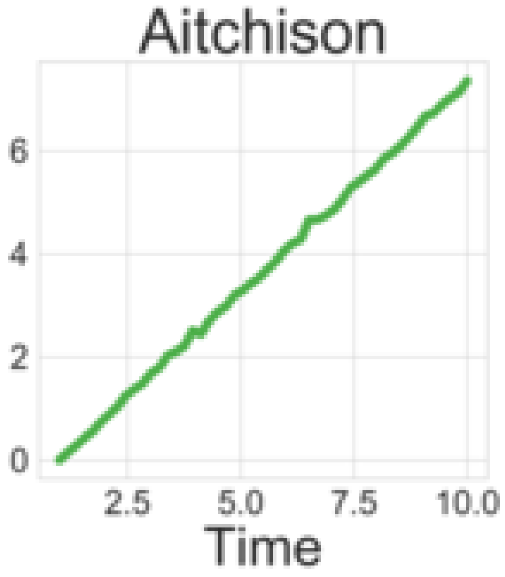
- Permutation invariant!

- Subcompositional dominant!



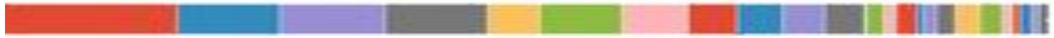
$$g(x)=\sqrt[\mathrm{d}]{\prod_{k=1}^{d}x_k}$$



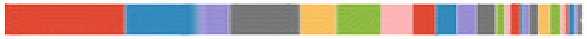


$$\sqrt{\sum_{k=1}^{d}\left(\log\frac{x_k}{g(x)}-\log\frac{y_k}{g(y)}\right)^2}$$

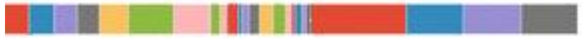
- Scale invariant!



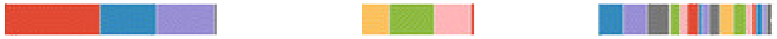
- Perturbation invariant!



- Permutation invariant!



- Subcompositional dominant!



$$g(x)=\sqrt[d]{\prod_{k=1}^{d}x_k}$$

→ Fits all four requirements suggested by…. Dr. John Aitchison… ⍰

CLR transform – Centered Log Ratio transform

$$\text{distance} = \sqrt{\sum_{k=1}^{d}\left(\log\frac{x_k}{g(x)} - \log\frac{y_k}{g(y)}\right)^2}$$

$$g(x) = \sqrt[\mathrm{d}]{\prod_{k=1}^{d} x_k} = e^{\left(\frac{1}{d}\sum_{k=1}^{d}\log x_k\right)} = e^{\overline{\log x}}$$

CLR transform – Centered Log Ratio transform

$$\text{distance} = \sqrt{\sum_{k=1}^{d}\left(\log\frac{x_k}{g(x)} - \log\frac{y_k}{g(y)}\right)^2} = \sqrt{\sum_{k=1}^{d}\left(\text{clr}(x_k) - \text{clr}(y_k)\right)^2}$$

$$g(x) = \sqrt[\text{d}]{\prod_{k=1}^{d} x_k} = e^{\left(\frac{1}{d}\sum_{k=1}^{d}\log x_k\right)} = e^{\overline{\log x}}$$

$$\text{clr}(x_k) = \log\frac{x_k}{g(x)} = \log x_k - \log g(x) = \log x_k - \overline{\log x}$$

$$\text{clr}(x) = \log x - \overline{\log x}$$

CLR transform – Centered Log Ratio transform

$$\text{distance} = \sqrt{\sum_{k=1}^{d}\left(\log\frac{x_k}{g(x)} - \log\frac{y_k}{g(y)}\right)^2} = \sqrt{\sum_{k=1}^{d}\left(\text{clr}(x_k) - \text{clr}(y_k)\right)^2}$$

$$g(x) = \sqrt[\text{d}]{\prod_{k=1}^{d} x_k} = e^{\left(\frac{1}{d}\sum_{k=1}^{d}\log x_k\right)} = e^{\overline{\log x}}$$

$$\text{clr}(x_k) = \log\frac{x_k}{g(x)} = \log x_k - \log g(x) = \log x_k - \overline{\log x}$$

$$\text{clr}(x) = \log x - \overline{\log x}$$

But wait, what about zeros...?

$$\log(0) = \texttt{undefined}$$

- just ignore them
- ignore them when calculating geometric mean / logarithmic average
- ignore them when calculating CLR
- call it “Robust CLR” – rCLR

$$g_r(x) = \sqrt[|\Omega_x|]{\prod_{k\in\Omega_x}^{|\Omega_x|} x_k} \qquad \mathrm{rclr}(x_k) = \log\frac{x_k}{g_r(x)} \text{ ... or } \texttt{undefined} \text{ if } x_k = 0$$

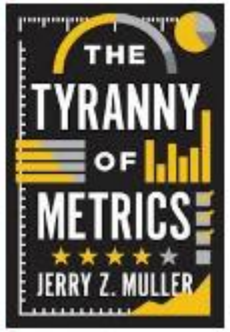
…Okay, but…we still have **`undefined`**s in our data

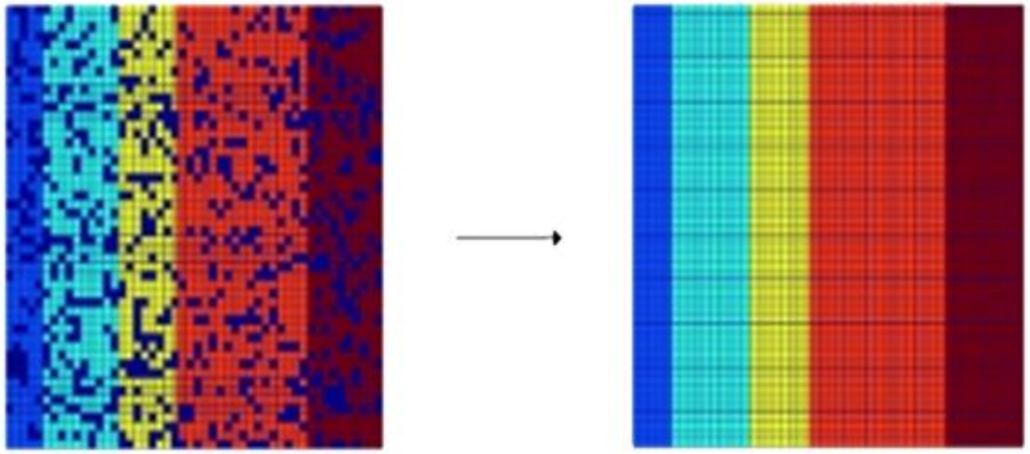
- Pretend that they're missing values!
- Reconstruct the vectors as if these species' data is missing
- This is called "**Matrix Completion**"

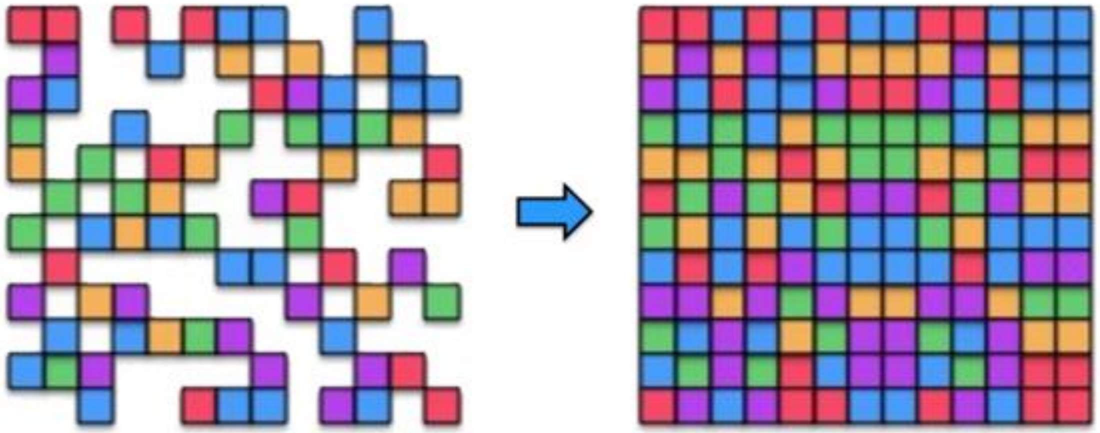
(where each column in the matrix is a different sample vector)

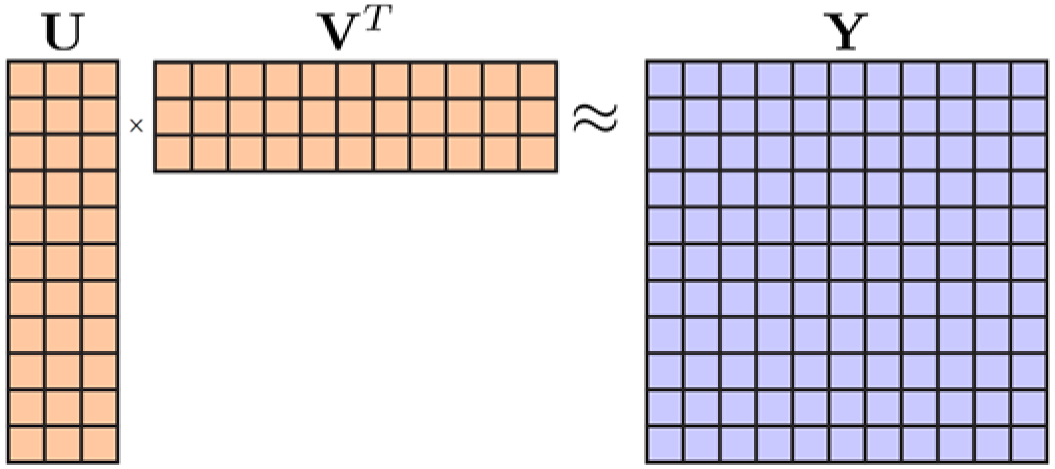
$$\begin{pmatrix} 1 & ? & ? & 4 & ? \\ ? & 2 & 5 & ? & ? \\ ? & ? & 4 & 5 & ? \\ 5 & ? & ? & ? & 4 \end{pmatrix}$$

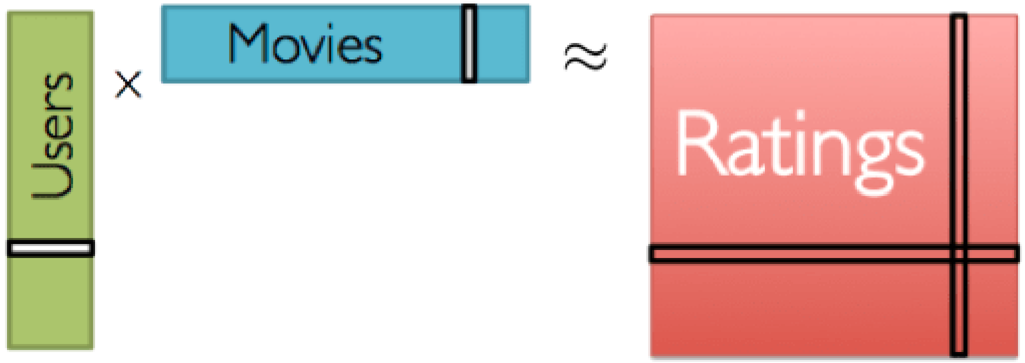
NETFLIX

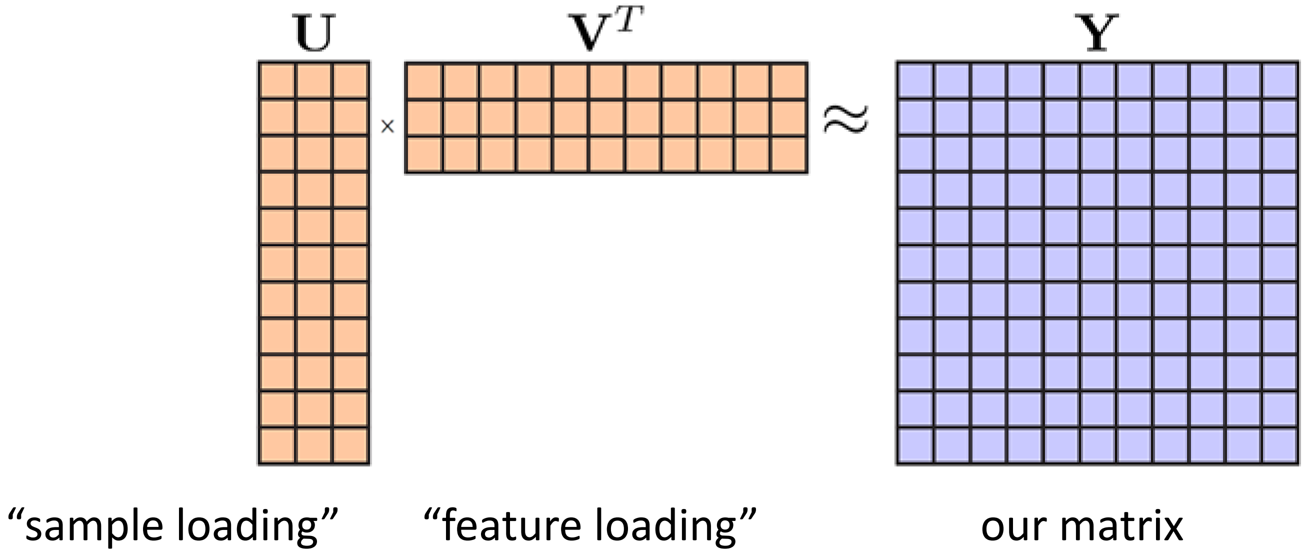
						• • •
Alice	1			4		
Bob		2	5			
Carol			4	5		
Dave	5				4	
⋮						






$\mathbf{U}$
×
$\mathbf{V}^T$
≈
$\mathbf{Y}$


Users
×
Movies
≈
Ratings



In the paper, the matrix completion algorithm that was applied is called **OptSpace:**

$$\text{optimize for } \min_{\text{U,V}}(|Y - USV^T|_2^2) \quad \text{(not including new entries)}$$

Summary

1. Transform our samples with the **Robust Centered Log Ratio transform**
2. Put all of the samples in a matrix, side by side
3. Run **Matrix Completion**
4. Result 1: a new complete matrix, representing what we “should see”
5. Result 2: two smaller matrices U and V, which represent important features and important samples

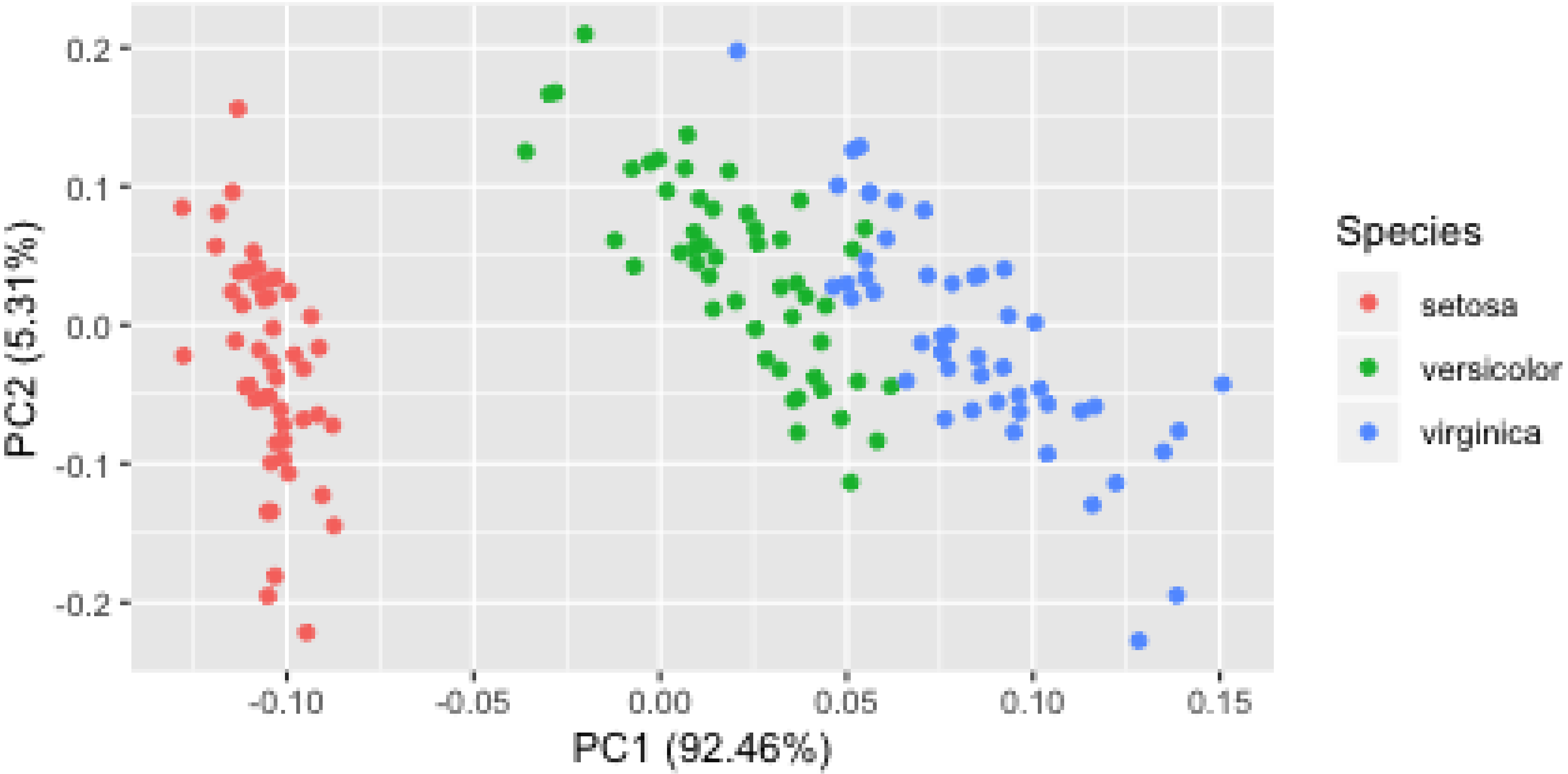
Now we can finally calculate distances between vectors in this matrix!

But instead of doing that, we’ll do Ordination – Principal Component Analysis.

PCA - Principal Component Analysis

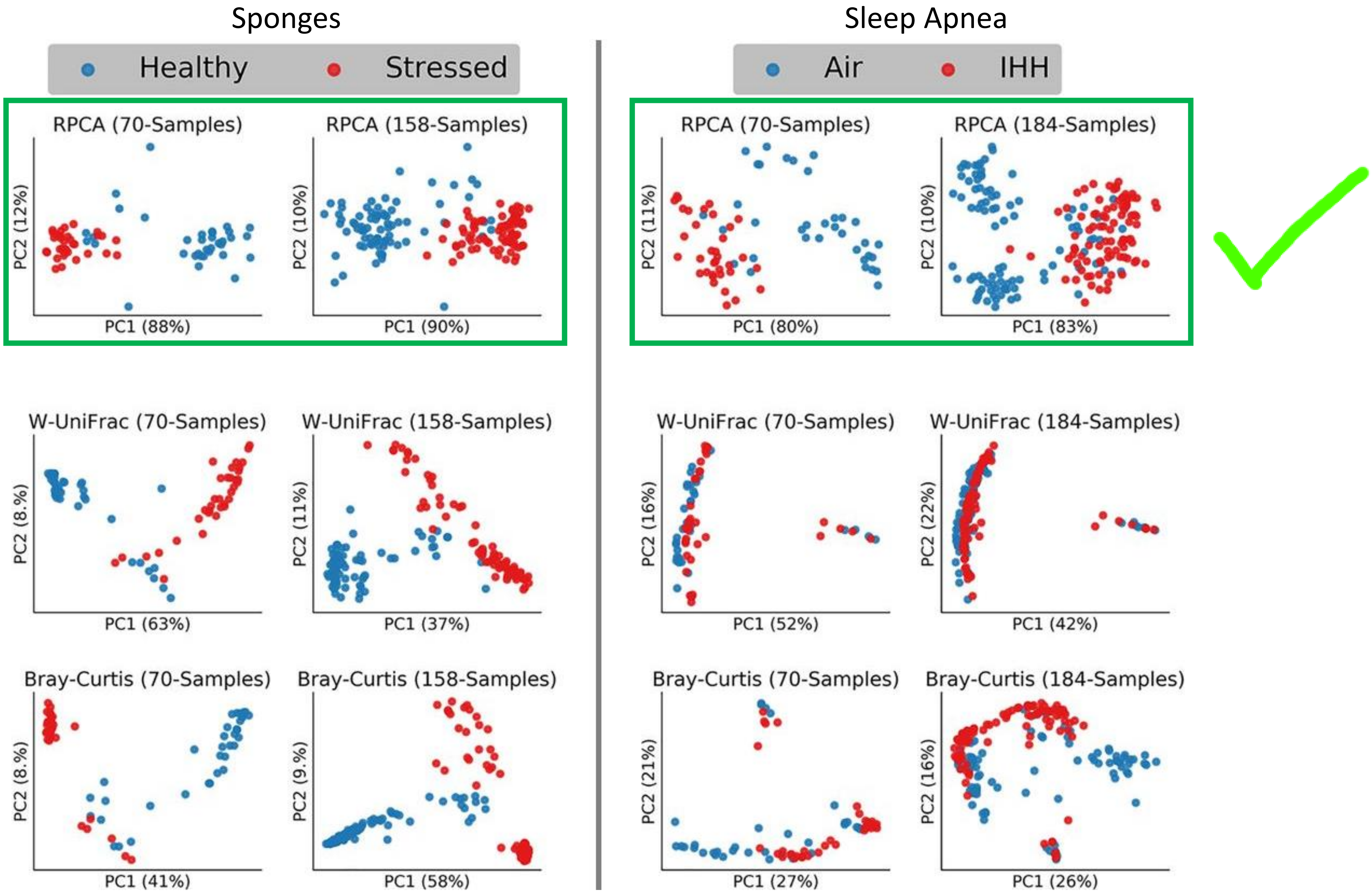
- PCA is when you try to find order in your chaos
- You search for components (e.g. groups of species) that explain the data, name them PC1, PC2, PC3, etc. in order of how much they each explain variance in the data, and then make a graph that shows how right you are
- Usually, only PC1 and PC2 are shown, because 2D graphs are easy
- Each axis represents a component of the data that explains the difference, and has a percentage number showing how much it does
- We want percentages to be high in each axis, relative to what's left to explain (after using the other axes)

Example

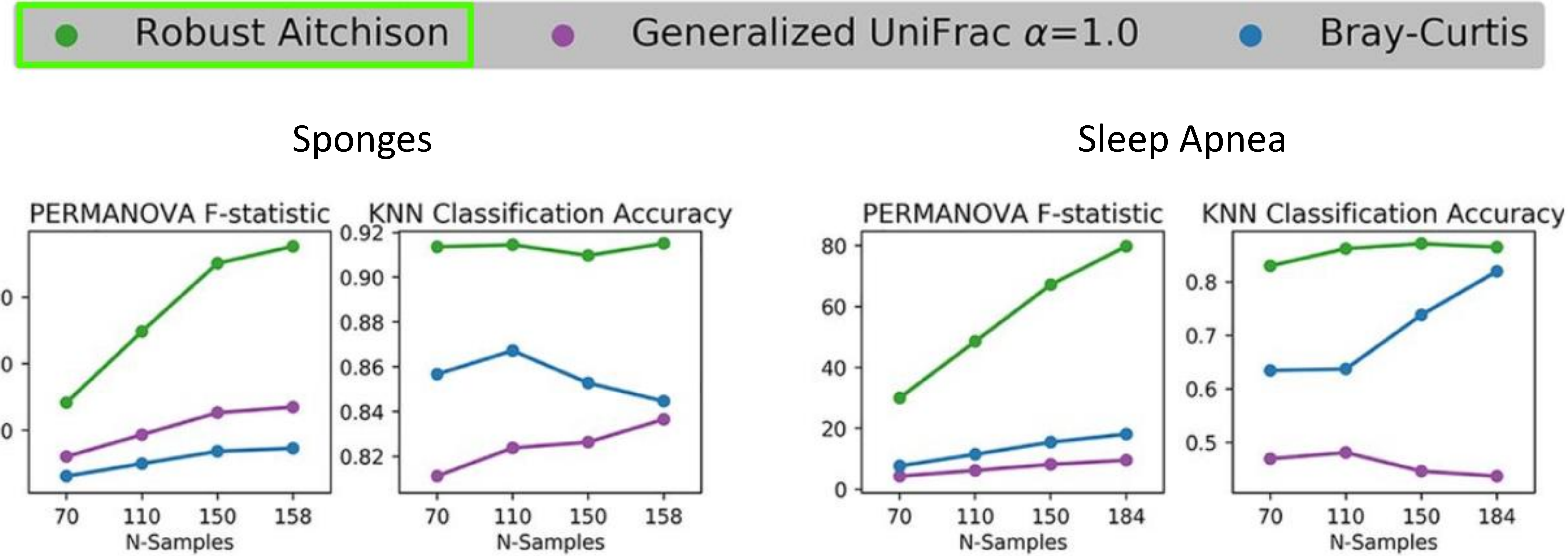


PC2 (5.31%)
0.2
0.1
0.0
-0.1
-0.2
PC1 (92.46%)
-0.10
-0.05
0.00
0.05
0.10
0.15
Species
setosa
versicolor
virginica

Results


Sponges
● Healthy
● Stressed
RPCA (70-Samples)
PC2 (12%)
PC1 (88%)
RPCA (158-Samples)
PC2 (10%)
PC1 (90%)
W-UniFrac (70-Samples)
PC2 (8.%)
PC1 (63%)
W-UniFrac (158-Samples)
PC2 (11%)
PC1 (37%)
Bray-Curtis (70-Samples)
PC2 (8.%)
PC1 (41%)
Bray-Curtis (158-Samples)
PC2 (9.%)
PC1 (58%)
Sleep Apnea
● Air
● IHH
RPCA (70-Samples)
PC2 (11%)
PC1 (80%)
RPCA (184-Samples)
PC2 (10%)
PC1 (83%)
✓
W-UniFrac (70-Samples)
PC2 (16%)
PC1 (52%)
W-UniFrac (184-Samples)
PC2 (22%)
PC1 (42%)
Bray-Curtis (70-Samples)
PC2 (21%)
PC1 (27%)
Bray-Curtis (184-Samples)
PC2 (16%)
PC1 (26%)

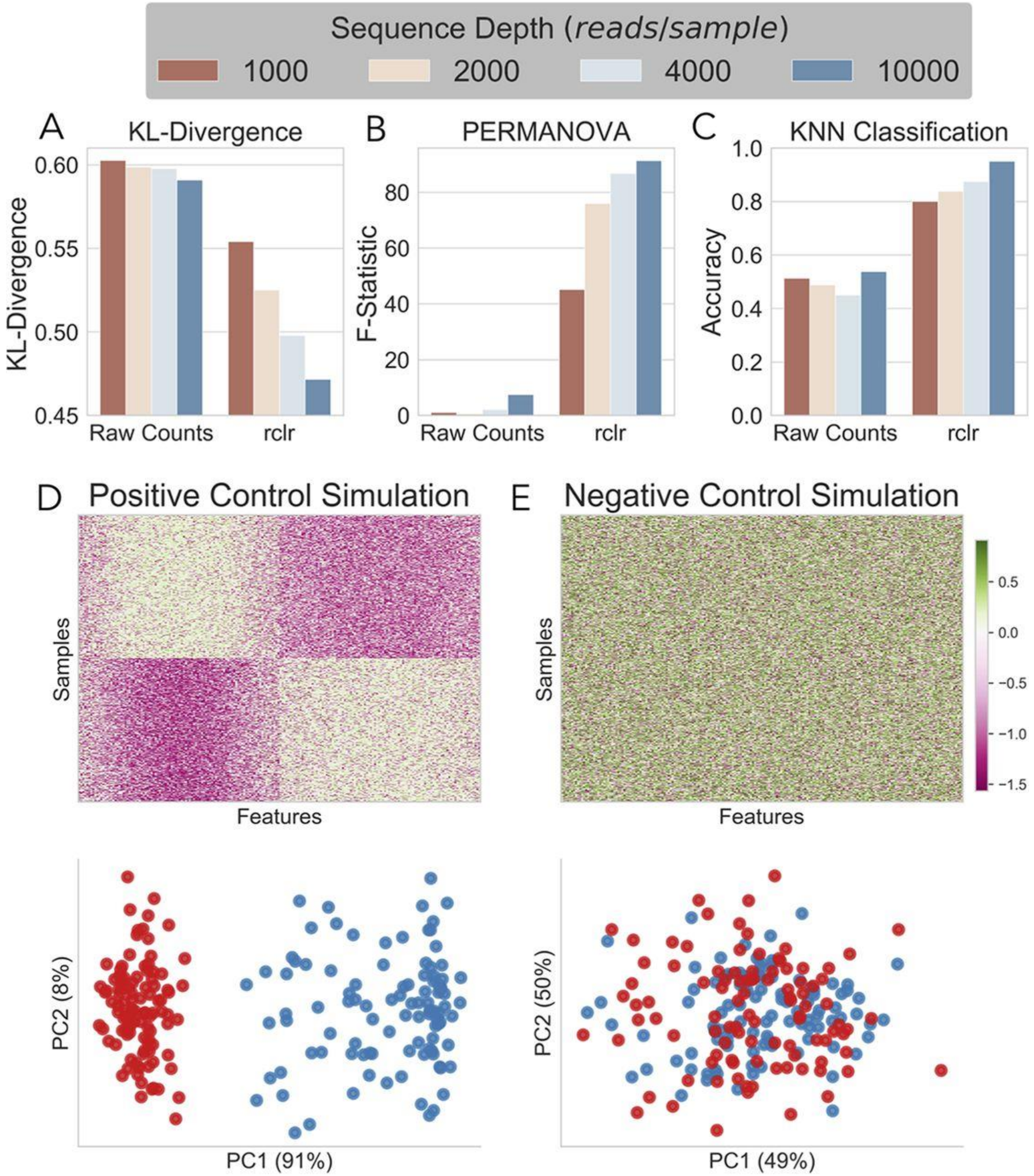
Results



Sponges

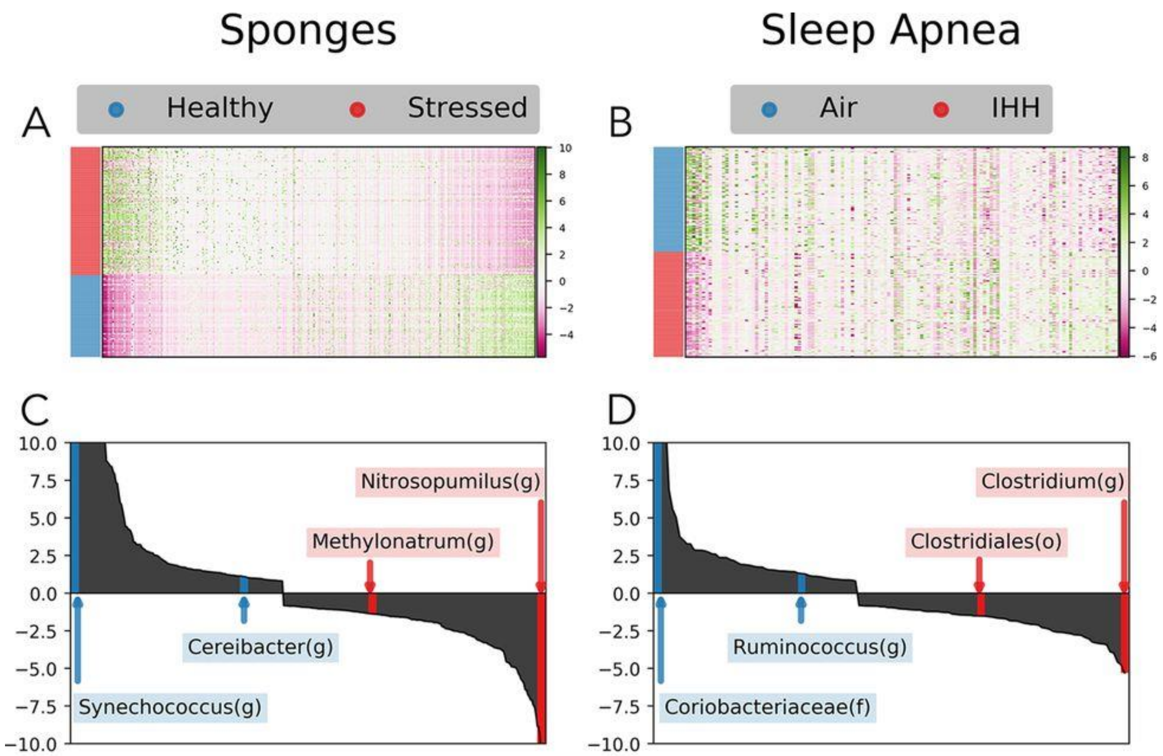
Sleep Apnea

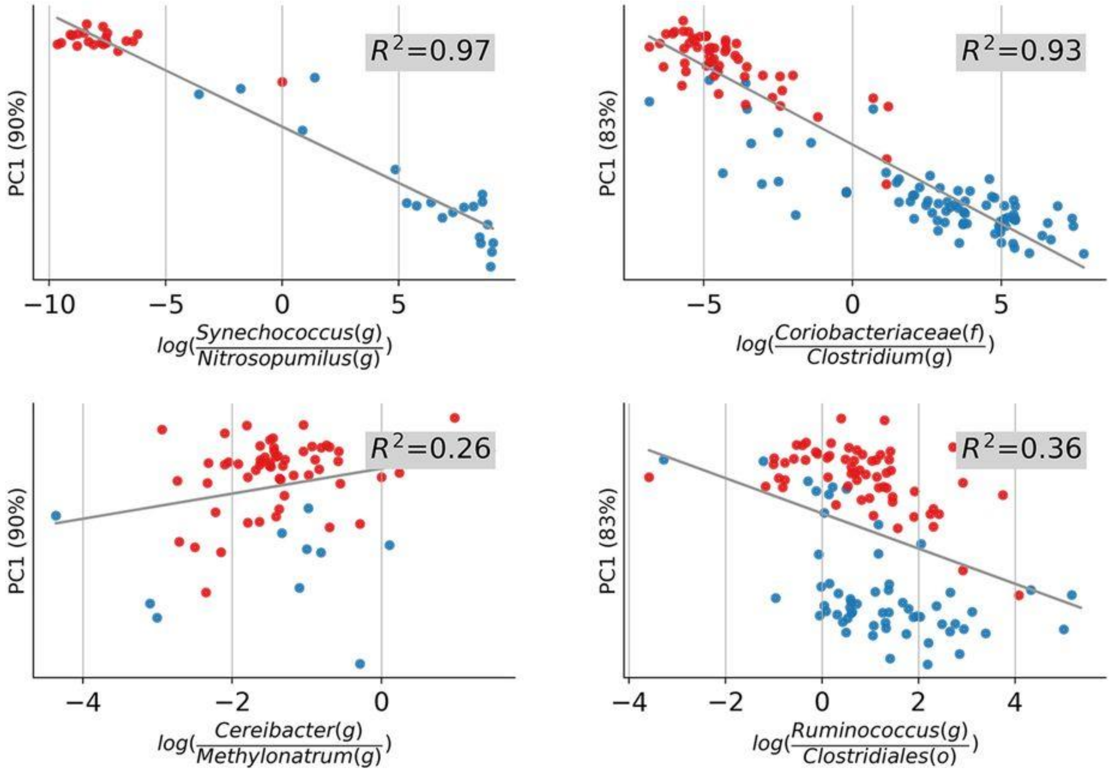
Results



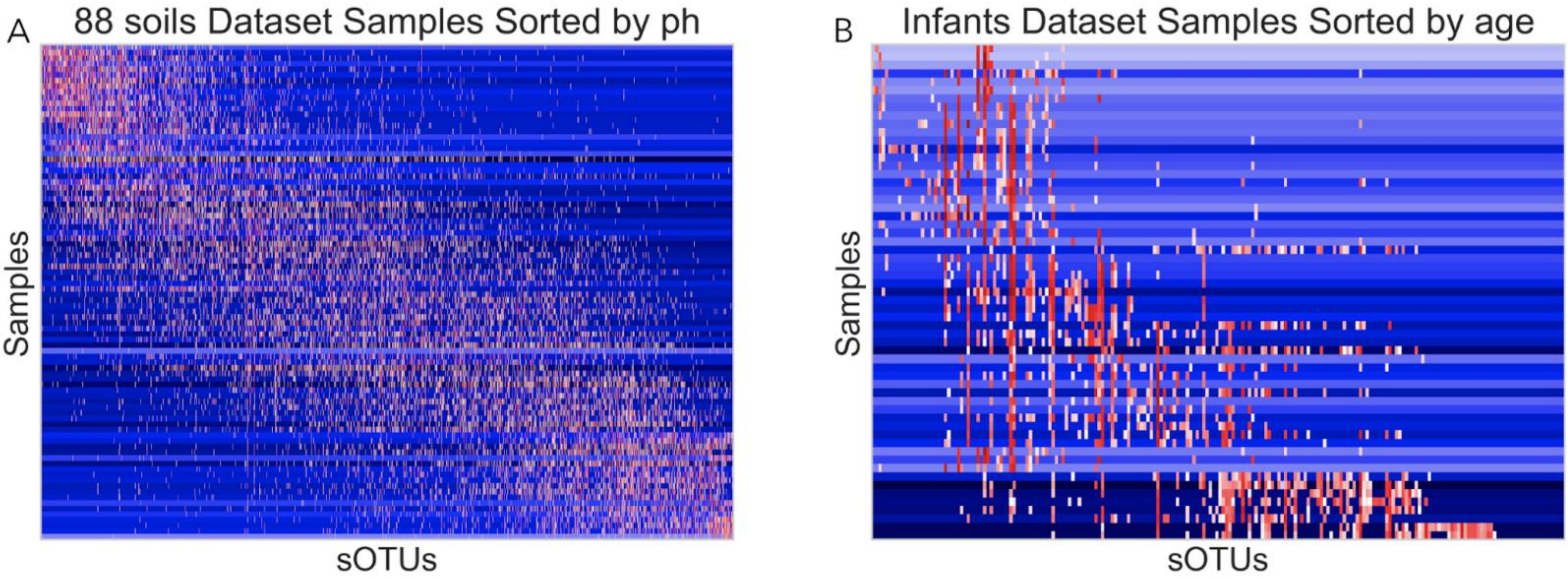
Sequence Depth (*reads/sample*)
1000
2000
4000
10000
A
KL-Divergence
KL-Divergence
0.60
0.55
0.50
0.45
Raw Counts
rclr
B
PERMANOVA
F-Statistic
80
60
40
20
0
Raw Counts
rclr
C
KNN Classification
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Raw Counts
rclr
D
Positive Control Simulation
Samples
Features
E
Negative Control Simulation
Samples
Features
0.5
0.0
−0.5
−1.0
−1.5
PC2 (8%)
PC1 (91%)
PC2 (50%)
PC1 (49%)

Results


Sponges
Sleep Apnea
A
● Healthy
● Stressed
B
● Air
● IHH
10
8
6
4
2
0
−2
−4
−6
C
D
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
Nitrosopumilus(g)
Methylonatrum(g)
Cereibacter(g)
Synechococcus(g)
Clostridium(g)
Clostridiales(o)
Ruminococcus(g)
Coriobacteriaceae(f)


PC1 (90%)
R^2=0.97
−10
−5
0
5
$log(\frac{Synechococcus(g)}{Nitrosopumilus(g)})$
PC1 (83%)
R^2=0.93
−5
0
5
$log(\frac{Coriobacteriaceae(f)}{Clostridium(g)})$
PC1 (90%)
R^2=0.26
−4
−2
0
$log(\frac{Cereibacter(g)}{Methylonatrum(g)})$
PC1 (83%)
R^2=0.36
−4
−2
0
2
4
$log(\frac{Ruminococcus(g)}{Clostridiales(o)})$

Potential problems – high-rank matrices!



Paper suggests to only use with rank 2-3, for ≤100 samples

Conclusion

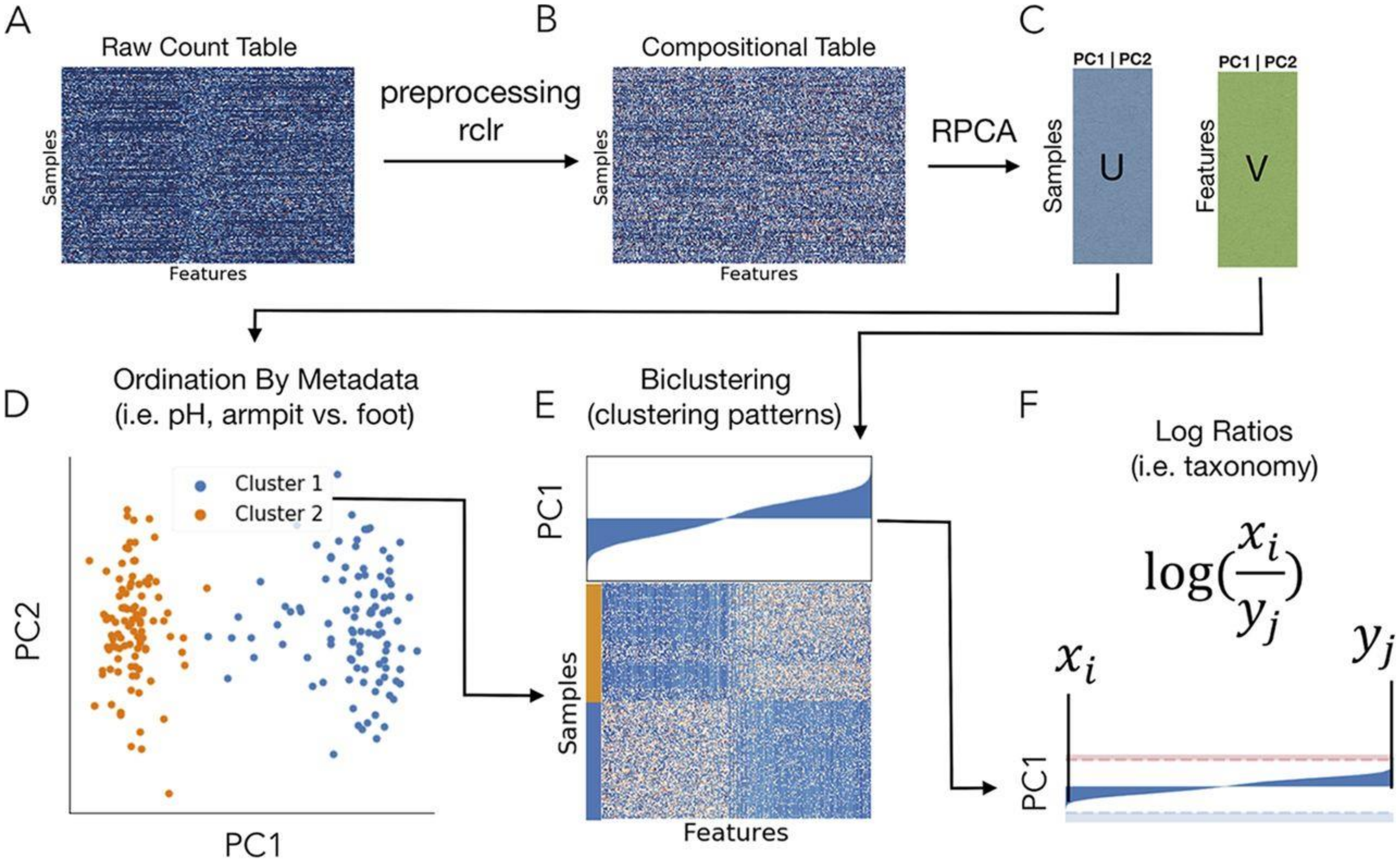
- Using Aitchison distance (with CLR transformation) is very good
- Using “robustness” and then matrix completion is pretty good on datasets where the rank is expected to be low
- Not good for “gradient-like” datasets, with a high-rank structure. It might cause misleading results
- Overfitting is possible but unlikely if we keep to low ranks (2-3)

(also the code for this is open-source, as usual)

Discussion Points

- Is this usually useful? Do many data sets actually have a low rank?
- All of the results are on small datasets and subsets of datasets, with a small amount of samples. Does this still work with very large sample sizes? Is it computationally slow?
- Why does matrix completion work so well on zeros? It feels like dark magic.

The End



(now you can finally understand this thing)