
WELCOME!
Summer School will begin at
8:30 a.m. PDT

EMSL

Summer School Day 5: Integration Methods and Machine Learning

Daniel Claborne & Samantha Erwin

Digital Intelligence; One Health and Biodefense

07.28.2023

8:30-9:45 a.m.	**Introduction to Machine Learning for Biological Applications**	**Sam Dixon**
9:45-10:15	Machine Learning in Pathogen Discovery	Isabelle O'Brien
10:15-10:30	Networking Break	
10:30-11:30	The What and Why of Multi-Omics Integration	Daniel Claborne
11:30-12:15	Case Study with MixOmics	Samantha Erwin

EMSL

Introduction to Machine Learning

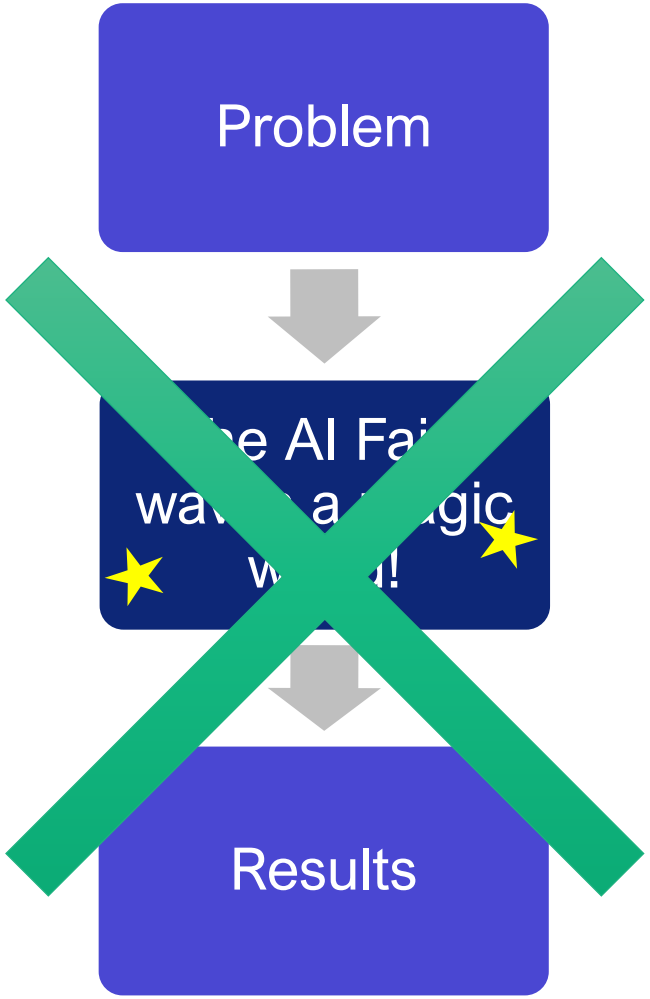
Samuel Dixon
Data Scientist

Machine Learning Myths

Equivalently: “Can we use AI to understand X?”

Requirements for Success

- **Scientific or Mission Goal**
- **Data** – you need data to learn
- **Quantitative metrics** – you need to know if you are getting closer to success
- **Partnership** – between data scientist and domain scientist/subject matter expert

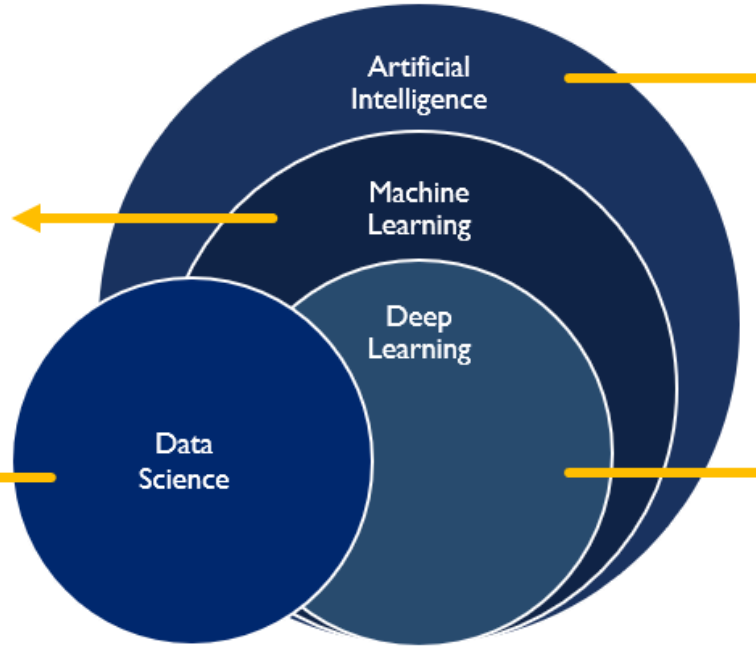


Equivalently: “Can we use AI to understand X?”

Artificial Intelligence

Machine Learning is a subset of AI, and builds a model based on training data to make predictions

Data Science is a subset of AI. It is an area of statistics, scientific methods, etc. to extract meaning and insights from data..



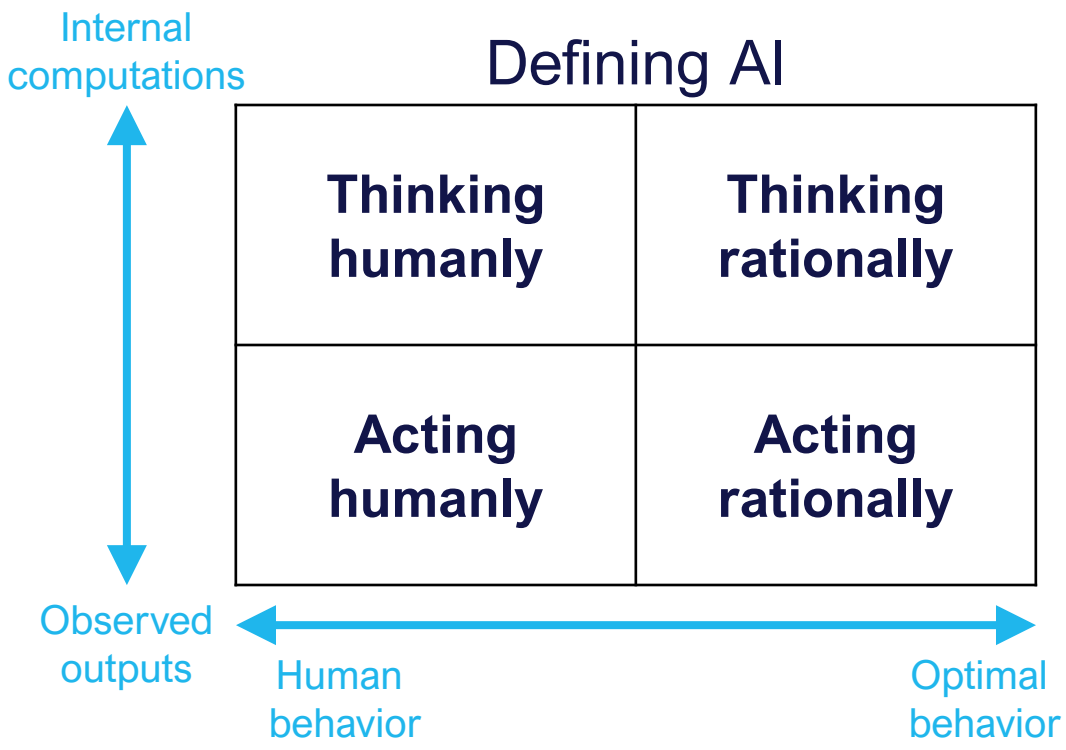
Artificial Intelligence means creating smart machines to mimic human behavior

Deep learning is a subset of ML, a class of ML algorithms to solve complex problems.

https://studyopedia.com/data-science/difference-datascience-machinelearing-ai-dl/

Artificial intelligence, machine learning and data science. Retrieved from "Data Science: Concepts and Practice" by Kotu, V., Deshpande, B. (2019). (2 ed.): Morgan Kaufmann. p. 3.

Artificial Intelligence



"The theory and development of computer systems able to perform tasks that **normally require human intelligence,** such as visual perception, speech recognition, decision-making, and translation between languages."

Two Historical Classes of AI

Symbolic

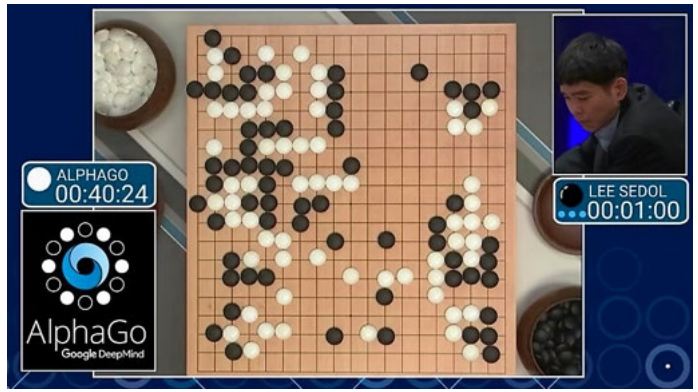
Handcrafted knowledge, rule-based systems, formal logic, causal models, ontologies

Largely driven by human-coded representations of prior knowledge

Statistical

Machine learning, probabilistic models, inductive inference

Largely driven by observational data

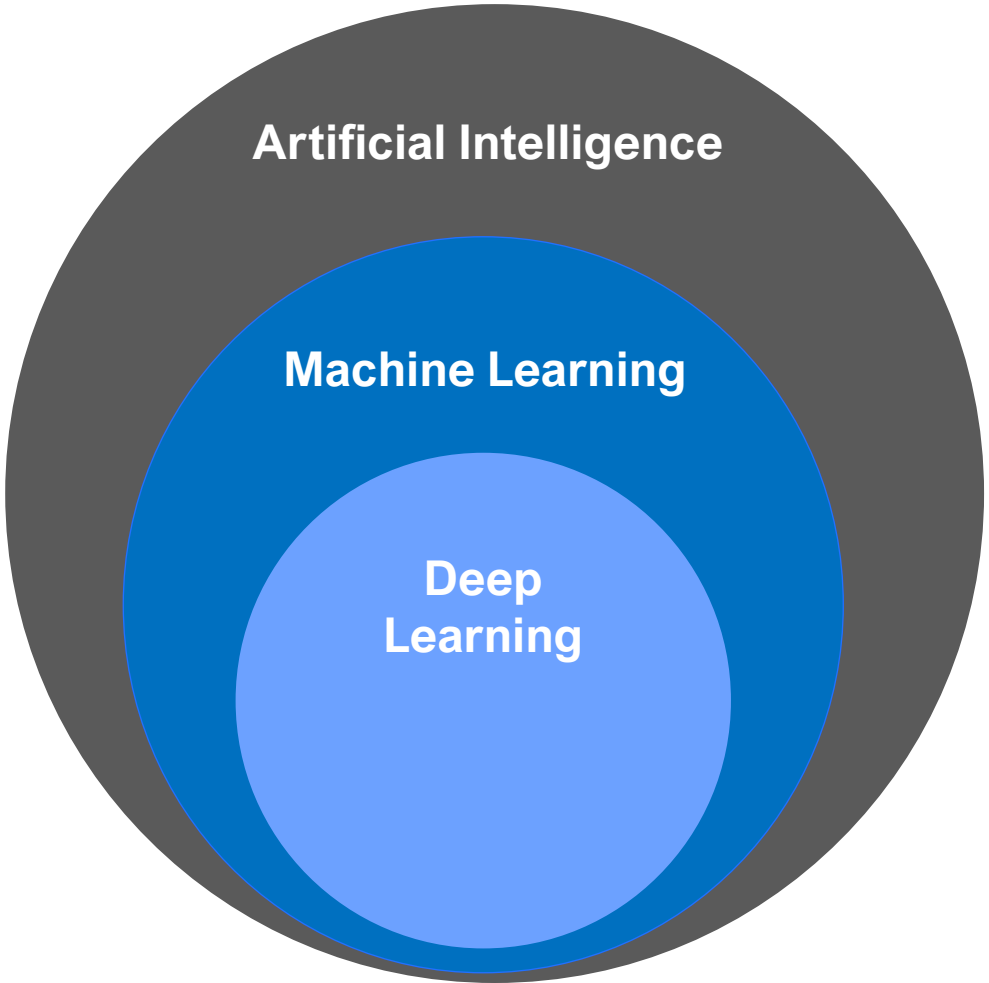


What is Machine Learning?

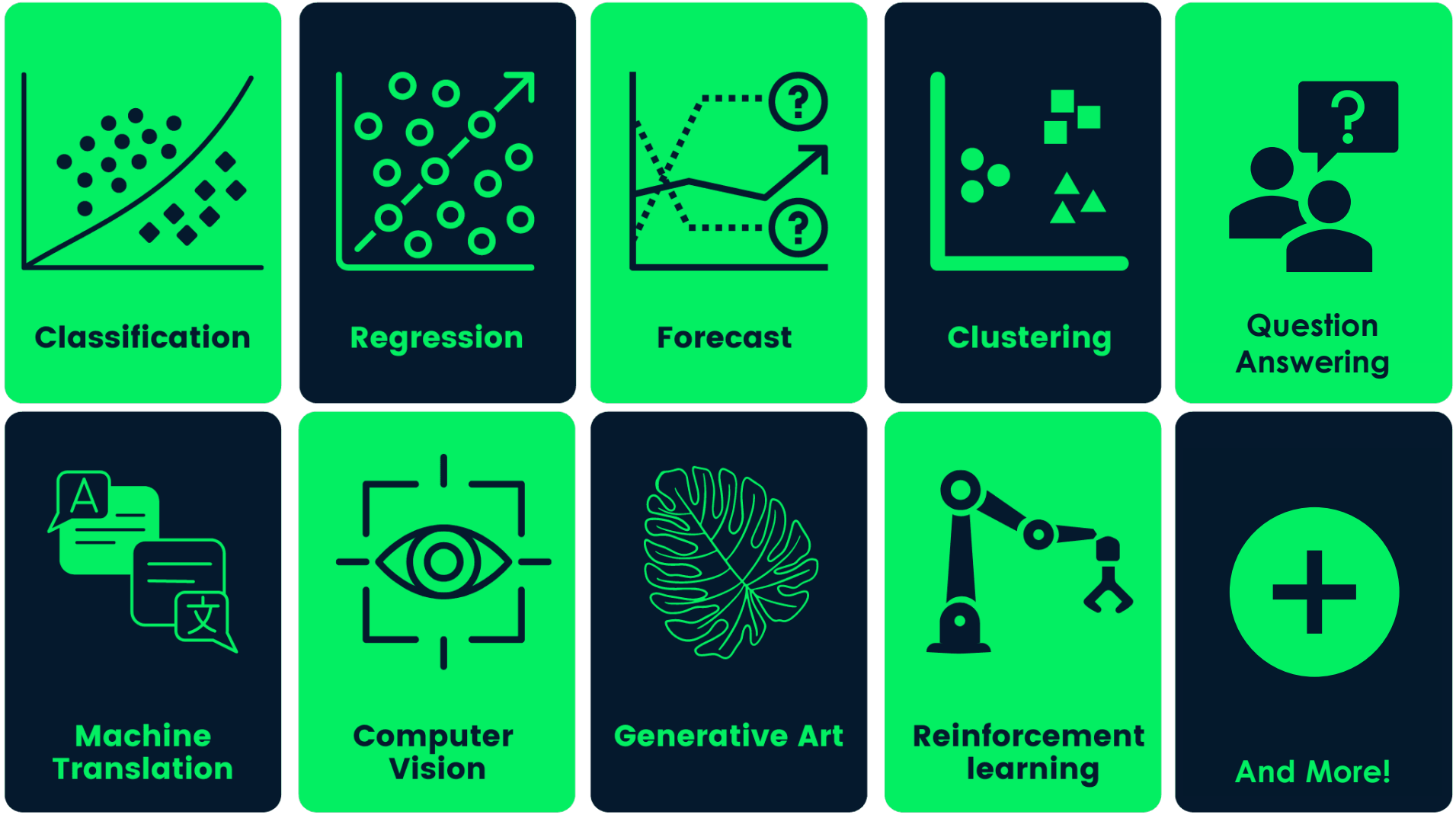
ML algorithms build on **statistical patterns observed in data**

Types of ML Models:

- Linear / Logistic Regression
- Random Forests
- Support Vector Machines
- Bayesian Models
- Neural Networks



What can we do with Machine Learning?



What is Machine Learning?

*A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.*

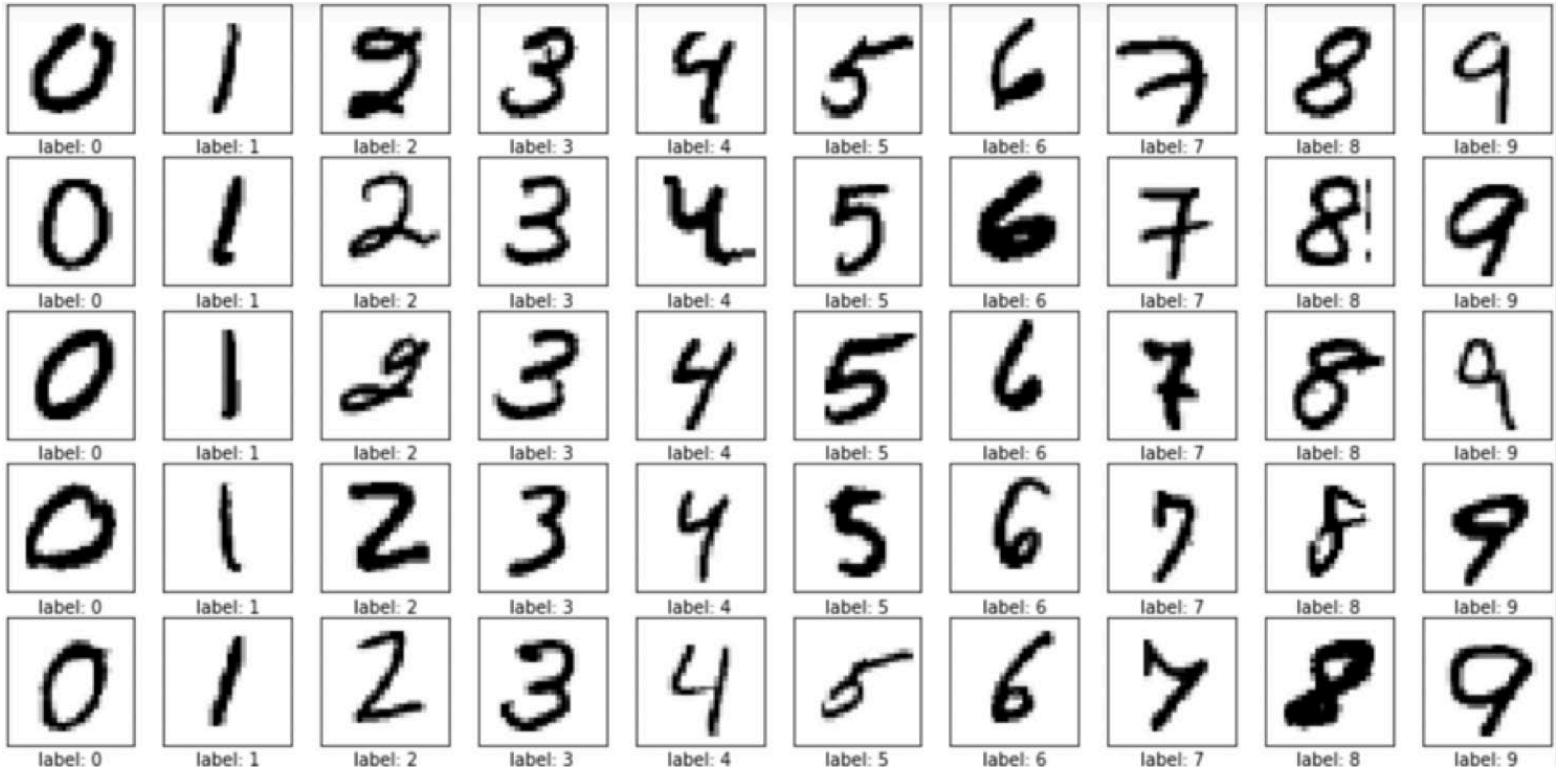
– Tom Mitchell, *Machine Learning*, 1997

Example Problem: Handwriting Recognition

Task (T): Recognizing and classifying handwritten numbers within images

Performance measure (P): Percent of numbers correctly classified

Experience (E): Database of handwritten numbers with given classifications



Example adapted from Tom Mitchell, *Machine Learning*, 1997

Data from MNIST database, http://yann.lecun.com/exdb/mnist/

How Does Machine Learning Work?

Data / Experience

$X = \{x_1, x_2, \ldots, x_n\}$	Y

Model

$$Y=f(x)$$

Loss Function

$$\varepsilon = \frac{1}{N}\sum_{i=0}^{n} g(f(x_i) - Y_i)$$

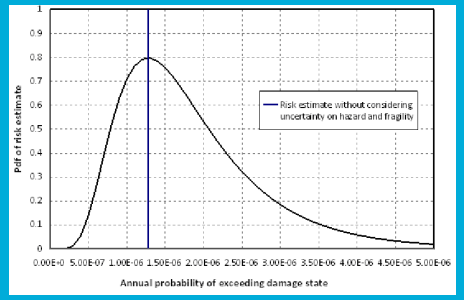
Learning Algorithm

```
if (settings[0].compareTo("s")==0) {
    if (name.compareTo("") !=0) {
        name += "_";
    }
    name+- etr.getString(settings[1]);
} else if (setting [0].compareTo("d") == 0){
    if (name.compareTo("") !=0) {
        name += "_";
    }
    name += DateUtils.format(etr.getDate(settings[1])
} else if (setting [0].compareTo("d") == 0){
    if (name.compareTo("") !=0) {
        name += "_";
    }
```

Parameterized Model

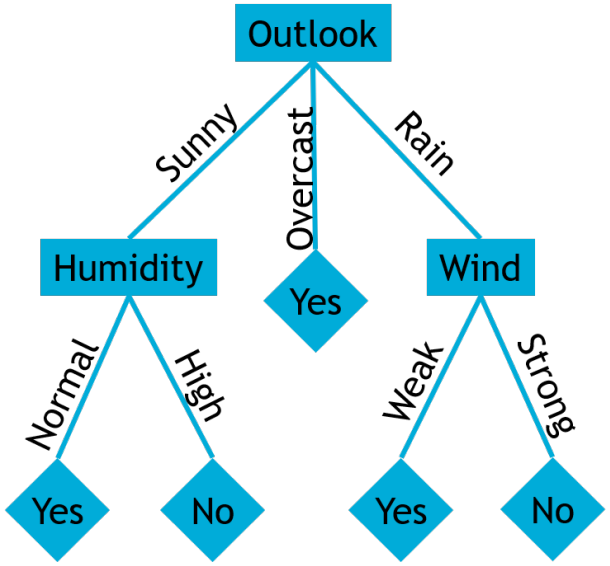
$$f(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots$$

Predictions & Evaluation



Learning Example : Decision Trees

- **Task**: Determine if Bill will play tennis given weather observations
- **Performance Metric**: Prediction accuracy
- **Experience**: Past observations



Day	Outlook	Temperature	Humidity	Wind	Play Tennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example from Mitchell, T.M. (1997). *Machine Learning*.

Learning Example: Data Preprocessing and Feature Engineering

- Many learning algorithms take a set or sequence of ***vectors*** as input
  - Raw data needs to be encoded in this format
  - For many data types, there are existing encoding conventions
- ***Feature engineering*** uses domain knowledge to create these encodings
  - Highly manual and time consuming
  - Quality of learned model often dependent on feature encodings

Example: Play Tennis?

Outlook: {sunny, overcast, rain} or
{sunny, partly cloudy, mostly cloudy, cloudy, drizzle, rain, downpour} or
RGB image from TennisCam

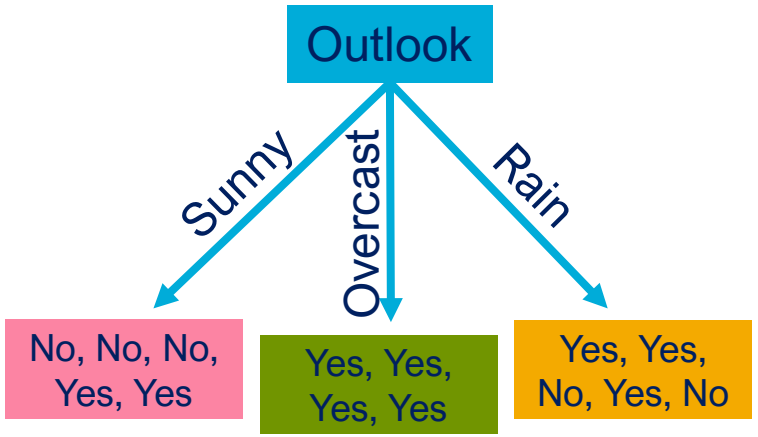
Temperature: {hot, mild, cool} or
{hot, warm, mild, cool, cold} or
{-20F, -19F, … , 114F, 115F} or
continuous

Learning Example: Decision Trees

General approach:

- Split the data based on information theory (entropy)
- Entropy measures the distribution of positive and negative examples in each block
- Greedy search through attribute (feature) space

Gain = Entropy all data − Sum of Entropies after split



Day	Outlook	Temperature	Humidity	Wind	Play Tennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

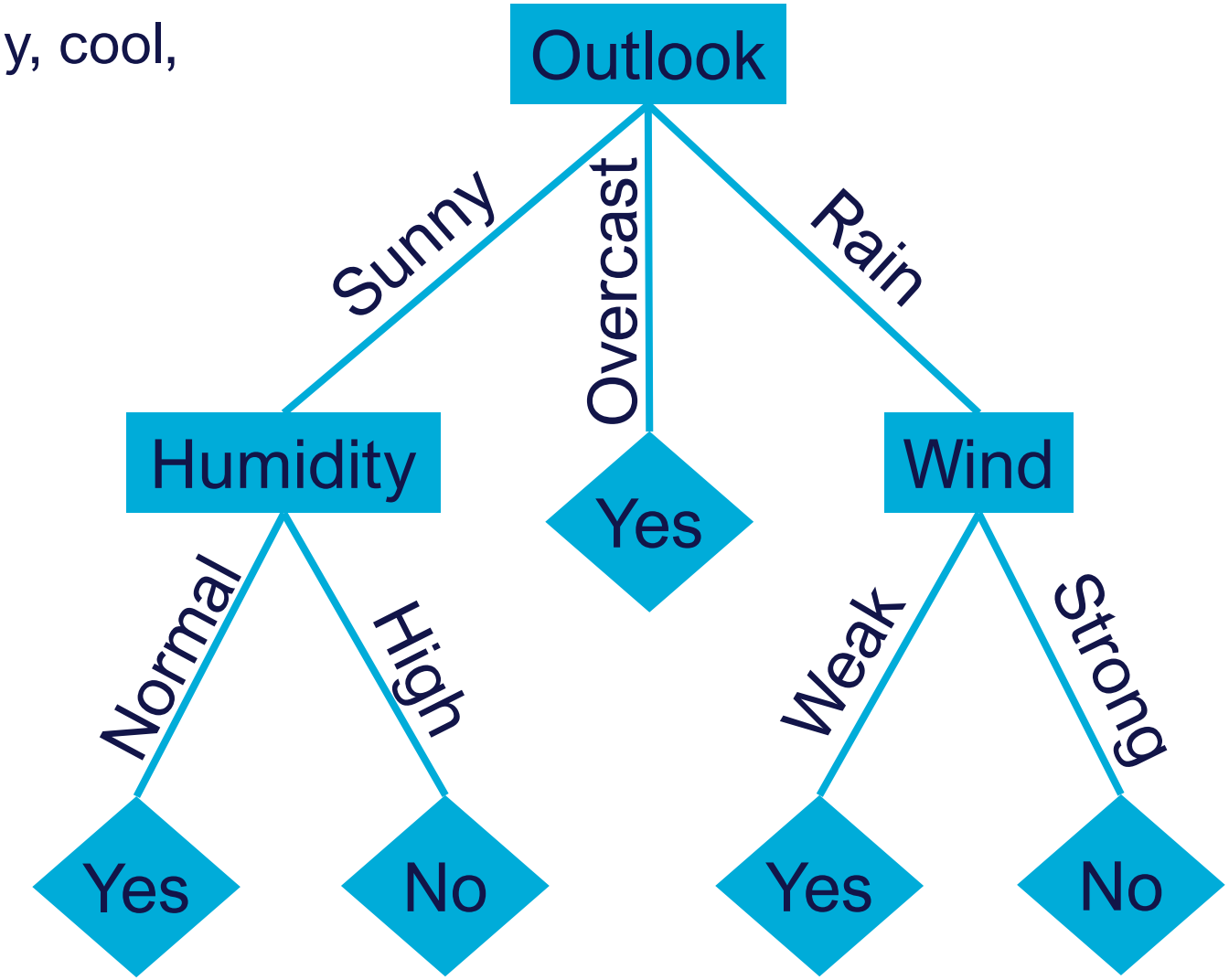
G=0.247 G=0.029 G=0.152 G=0.048

Learning Example: Decision Trees

(Day 15) What will happen on a sunny, cool, humid, windy day?

Many design decisions affect performance:

- Training data (number and quality of examples)
- Which variables describe the data
- Splitting criterion
- Binary versus multivariate splits
- What to do with numeric variables
- Stopping criterion



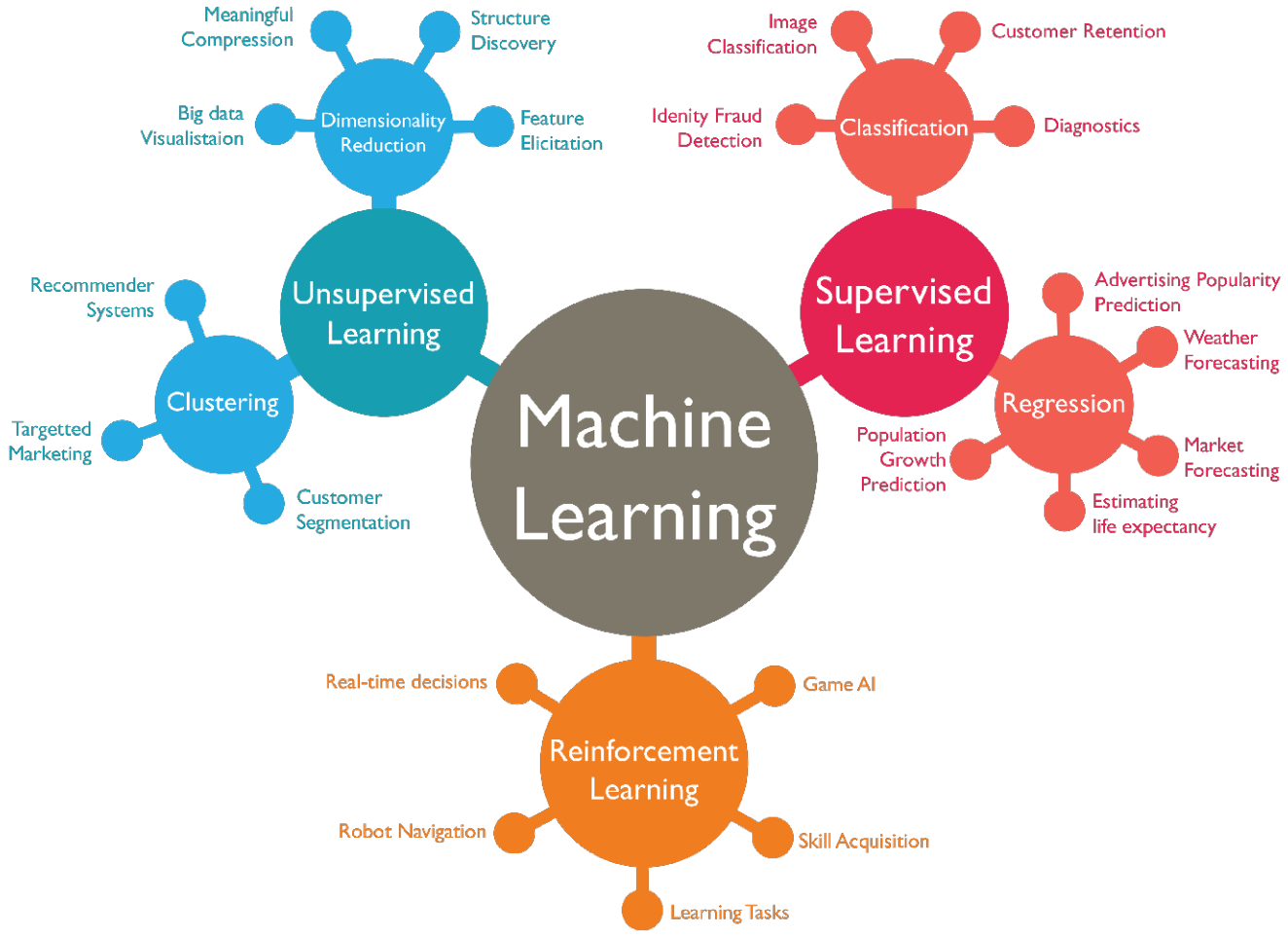
What's the Question?

- Machine learning models are often just a component of a larger system you must clearly define what you want to get out of the ML model itself.
- For a model to be effective it must answer the correct question.
  - Where does it fit in the larger system?
  - What task does the model accomplish?
- Other considerations:
  - What domain knowledge is relevant?
  - What data is available?

Who Knows your Problem?

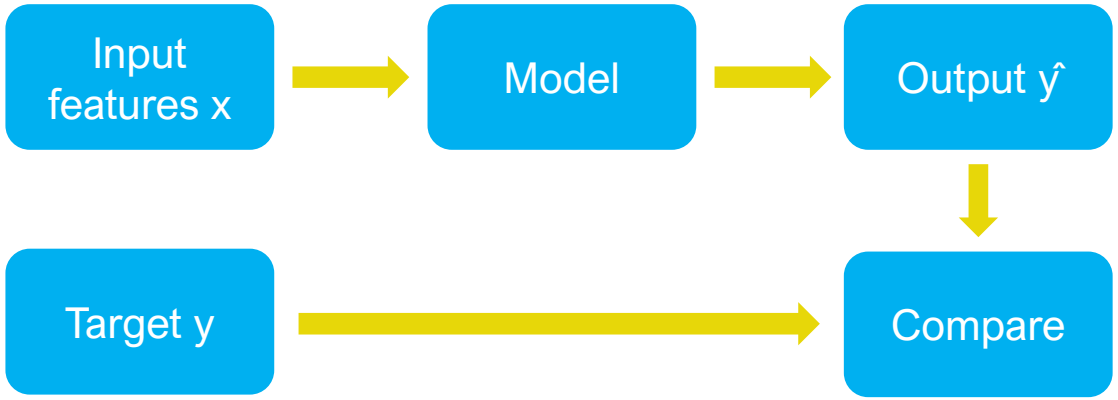
- ML experts know models
  - Identify approaches given a task and data
  - Process data for models
  - Develop and train models
- SMEs know the domain details
  - What the raw data looks like
  - How the physical sensor/system works
  - What the general goals of their field are
- Mission SMEs know the mission
  - What the real-world goal is
  - What success looks like when applying the whole system

Types of Machine Learning Problems



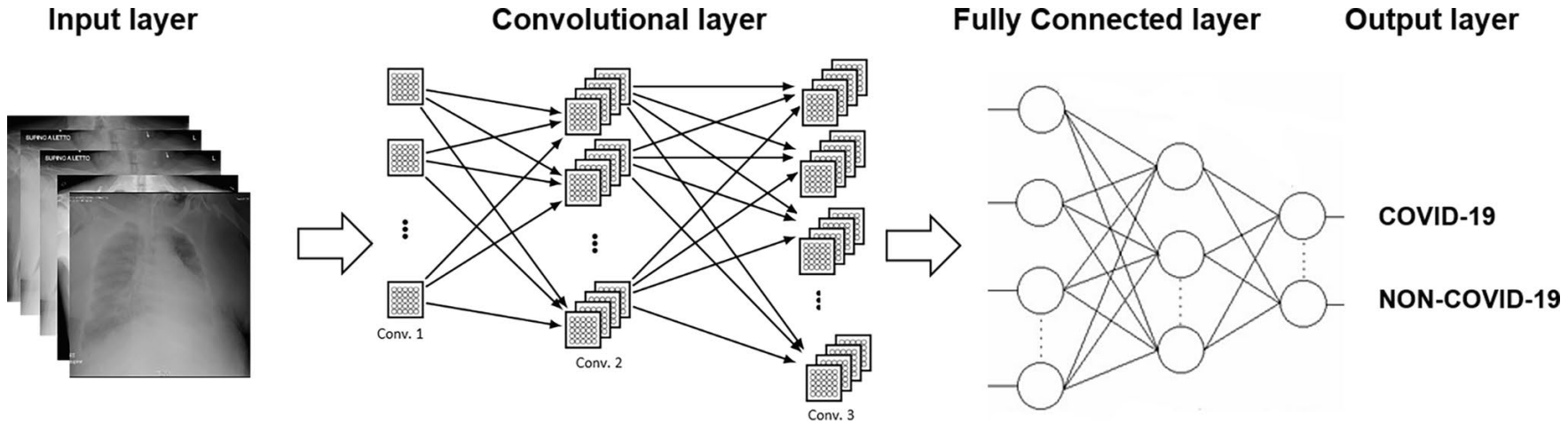
Supervised Learning

- “Supervised Learning is when the algorithm gets a copy of the answer key.”
- Requires labeled datapoints
  - Number of examples depends on task and algorithm
  - More is always better!
- Supervised models trained on representative data
  - New inputs/outputs look like old inputs/outputs
  - No guarantee you can generalize to new types of data
    or new parts of the input space



COVID-19 detection from chest x-rays

- Binary classification: *is pneumonia due to Covid-19 or other causes?*
- Results:
  - 98% accuracy
  - 98% sensitivity, 97% specificity



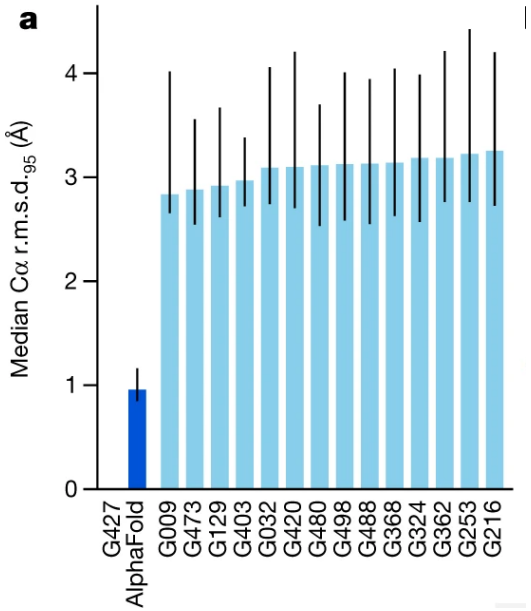
COVID-19 classification in X-ray chest images using a new convolutional neural network: CNN-COVID (2022)

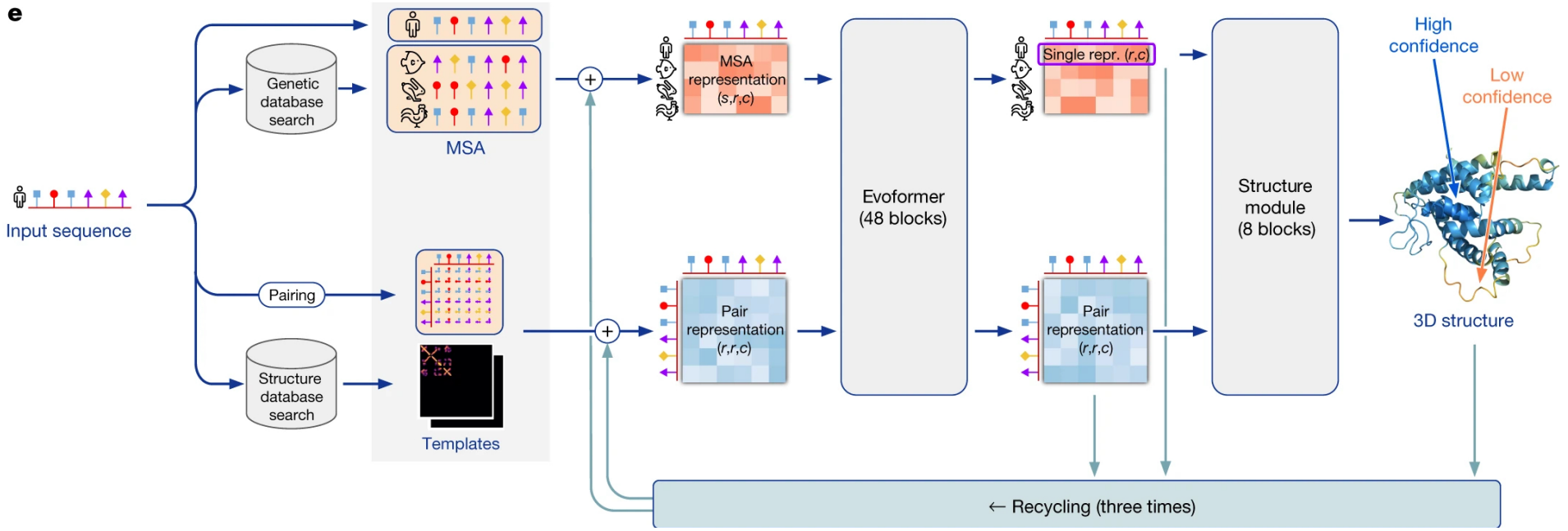
https://link.springer.com/article/10.1007/s42600-020-00120-5

AlphaFold

Predicts 3D protein structure with transformers

- Inputs: sequences
- Outputs: cartesian coordinates of folded protein
- Transformer applied to *embeddings* of sequence and structure

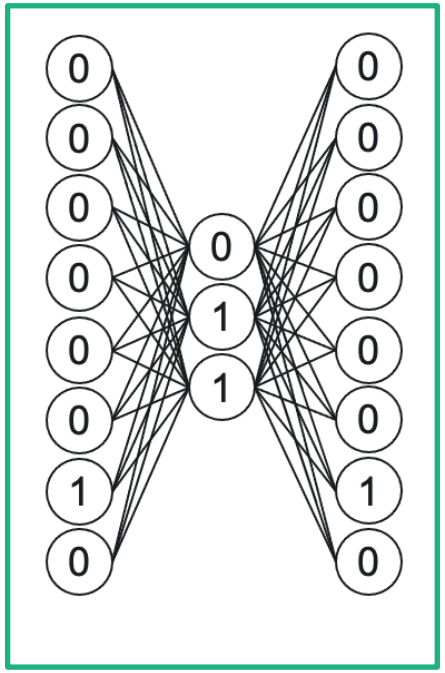


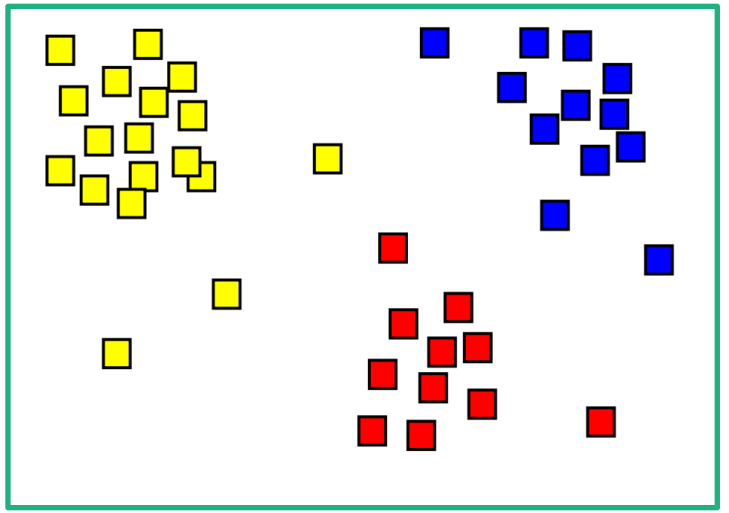


"Highly accurate protein structure prediction with AlphaFold," *Nature*, 2021. https://www.nature.com/articles/s41586-021-03819-2

Unsupervised Learning

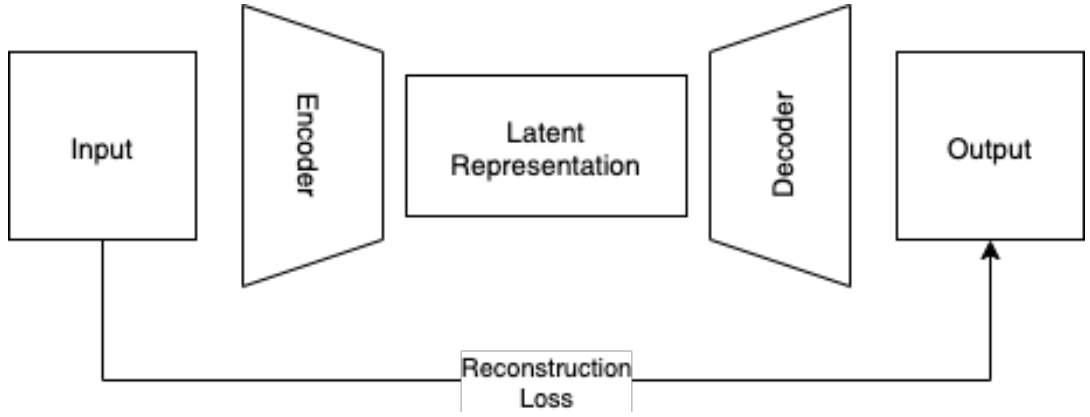
- **Inputs -> ???**
  - What if we don't have any labels?
  - Replace outputs with another goal
- Many different approaches:
  - Autoencoders
    (Capture regularities through data compression)
  - Clustering algorithms
    (Group by similarity)
  - Semi- & Self-supervised learning
    (Create your own labels!)
  - Generative Adversarial Networks
    (Create new examples to drive training)
- Useful for many tasks, including anomaly detection



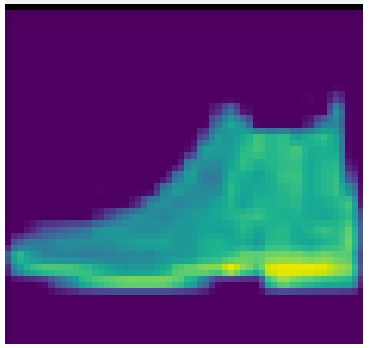


Autoencoder

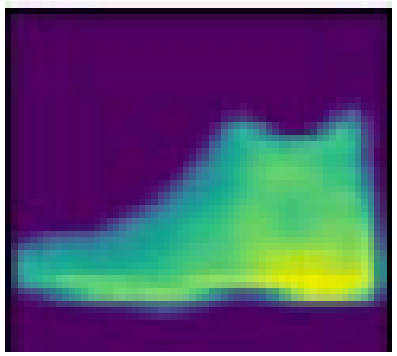
- Unsupervised models learning latent representations are often used as part of supervised methods
- With an auto encoders the **Goal** is to reduce the dimensionality of the input and recover it
- Dimensionality reduction is at heart a data compression strategy
- Neural networks can learn very rich representations of data



Original

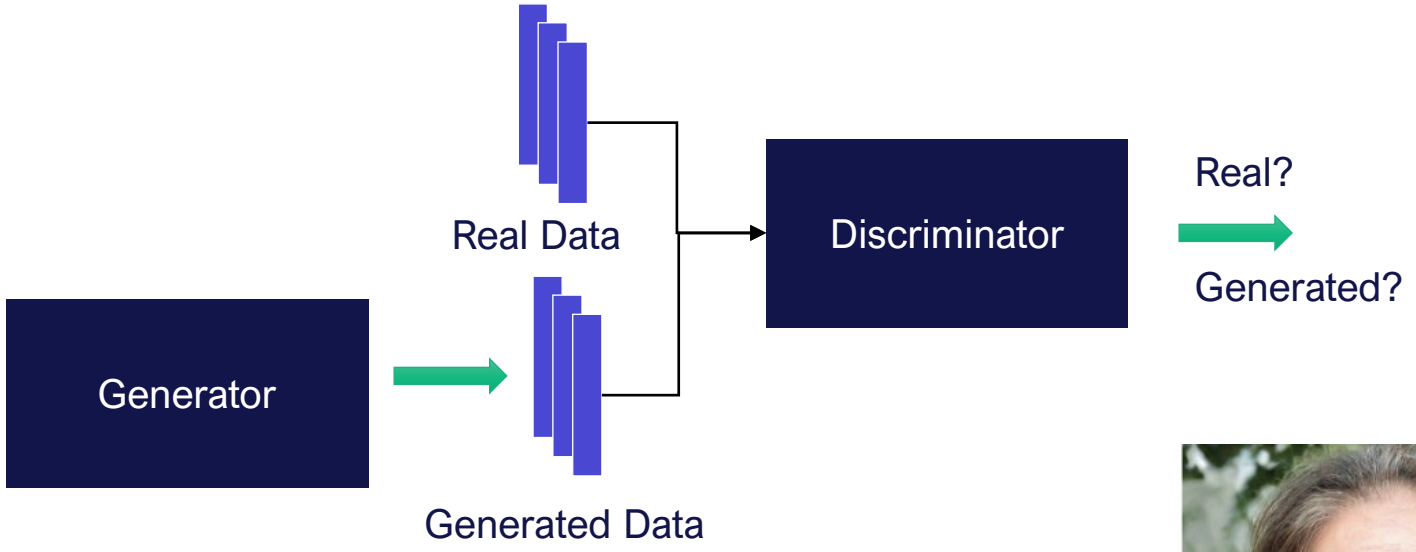


Reconstruction



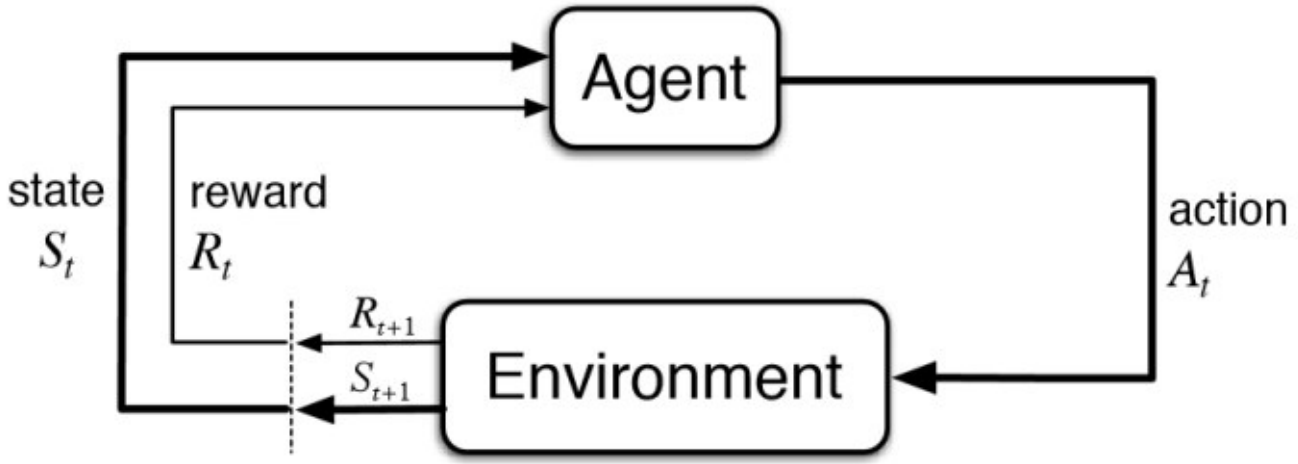
Generative Adversarial Networks

- Essential idea: Connect two models with competing objectives
  - Generator: Create realistic data
  - Discriminator: Distinguish between real and generated data



Reinforcement Learning

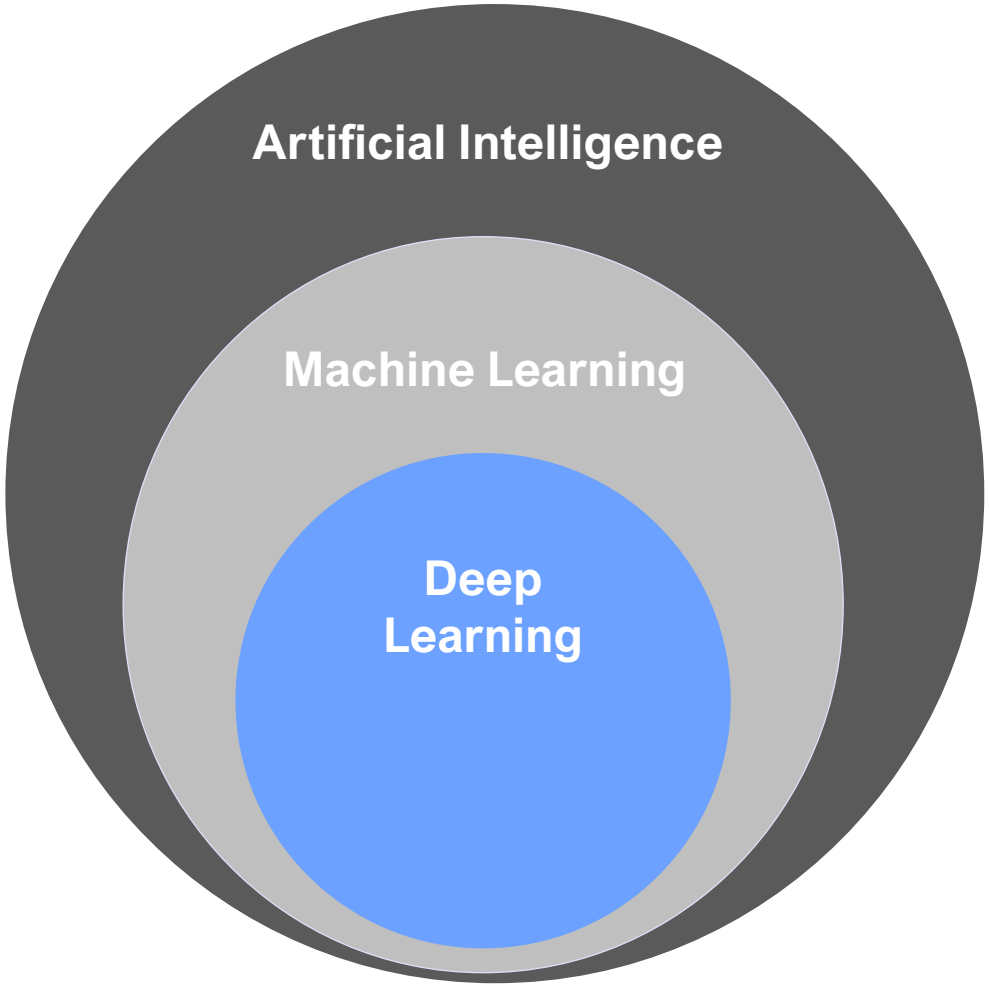
- Learns from trial and error
- Reinforcement learning is based on rewarding desired behaviors and/or punishing undesired ones
- Goal is to learn a **policy** - that maximizes the expected reward according to a **value-function**



https://icml.cc/2016/tutorials/deep_rl_tutorial.pdf

What is Deep Learning?

- Specific subset of machine learning algorithms
- Made from successive layers composed of simpler models. The “deep” learning refers to these layers of the model
- Achieves state-of-the-art performance in many domains



ML vs DL: Traditional ML Pipelines

Goh et al. (2017). Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models

ML vs DL: End-to-End Learning

Goh et al. (2017). Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models

Why Deep Learning and Why Now?

- **Compute**

  Deep learning models are significantly more computationally expensive compared to their classical ML counterparts. Until recently, computing resources could only train very simple models. Modern GPU resources have greatly expanded the scale of models that can be trained.

- **Data**

  Bigger models require more data to successfully train. The availability of massive and carefully annotated datasets has allowed deep learning to show its effectiveness on a wide variety of models.

- **Code**

  Many new tools have been developed that abstract basic algorithms and allow more developers and researchers to leverage compute resources to train models on massive datasets without needing to worry about low level implementation details.

Questions?

8:30-9:45 a.m.	Introduction to Machine Learning for Biological Applications	Sam Dixon
9:45-10:15	**Machine Learning in Pathogen Discovery**	**Isabelle O'Brien**
10:15-10:30	Networking Break	
10:30-11:30	The What and Why of Multi-Omics Integration	Daniel Claborne
11:30-12:15	Case Study with MixOmics	Samantha Erwin

EMSL

Machine Learning in Pathogen Discovery

Isabelle O’Bryon

Thank you to Becky Hess

OmniScreen: Friend or Foe?

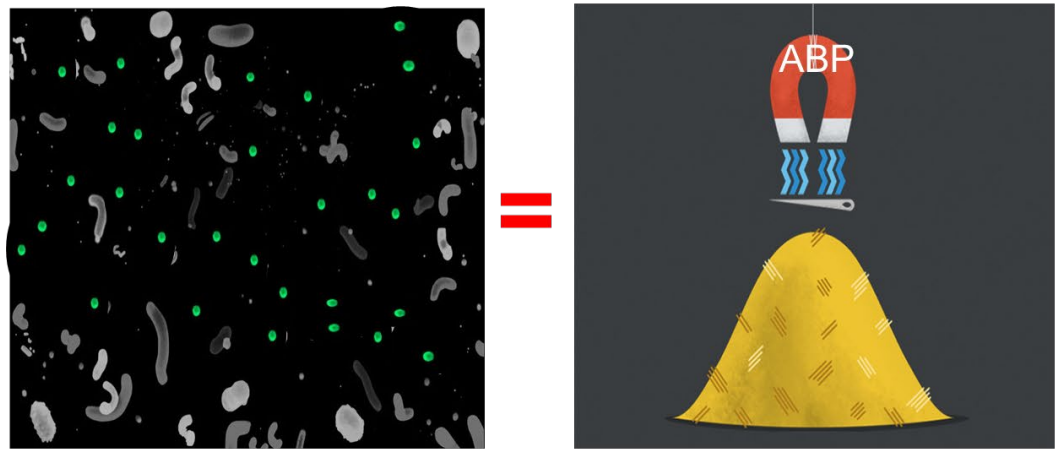
- Defense Advanced Research Projects Agency (DARPA) issued a “Friend or Foe challenge:
  - Can we isolate pathogens without altering the sample of phenotype? (1)
- Unknown bacteria can be introduced to human immune cells and an immune response can be validated by a skilled microbiologist expert using microscopy
- The OmniScreen project aims to create a machine learning model for microscopy images that will automate the classification of pathogenic and non-pathogenic bacteria

OmniScreen Project

- Part 1 is to enrich for the pathogenic phenotype
- Part 2 is to challenge the potentially pathogenic bacteria to human cell lines
- Part 3 is to evaluate the if the bacteria was pathogenic to the human cells using microscopy
- Part 4 (future work) is to evaluate if the bacteria was pathogenic to the human cells using proteomics

Part 1: Pathogen Extraction and Enrichment

A potential pathogen is a needle in a haystack



Enriching for pathogens simplifies detection

Enrichment

Pathogens display phenotypes related to virulence

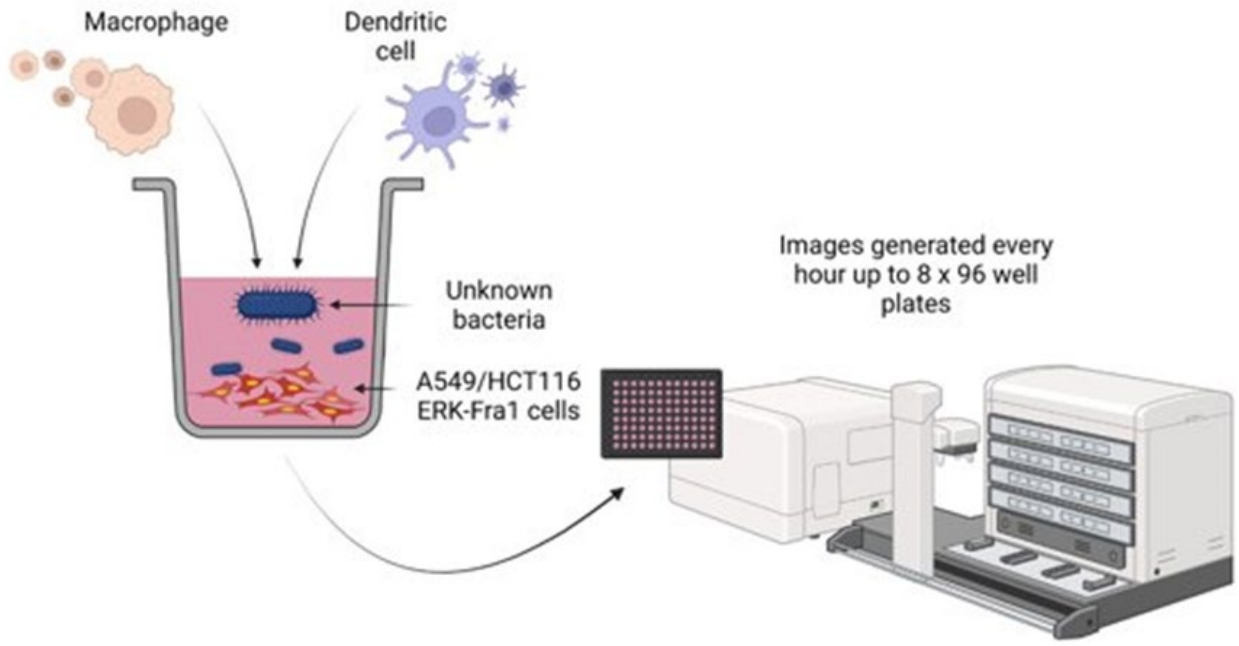
Interrogate a mixture limited to the species most likely to be pathogenic

Increase the likelihood of finding a single pathogen

Activity based probes (ABPs) applied to living microbes

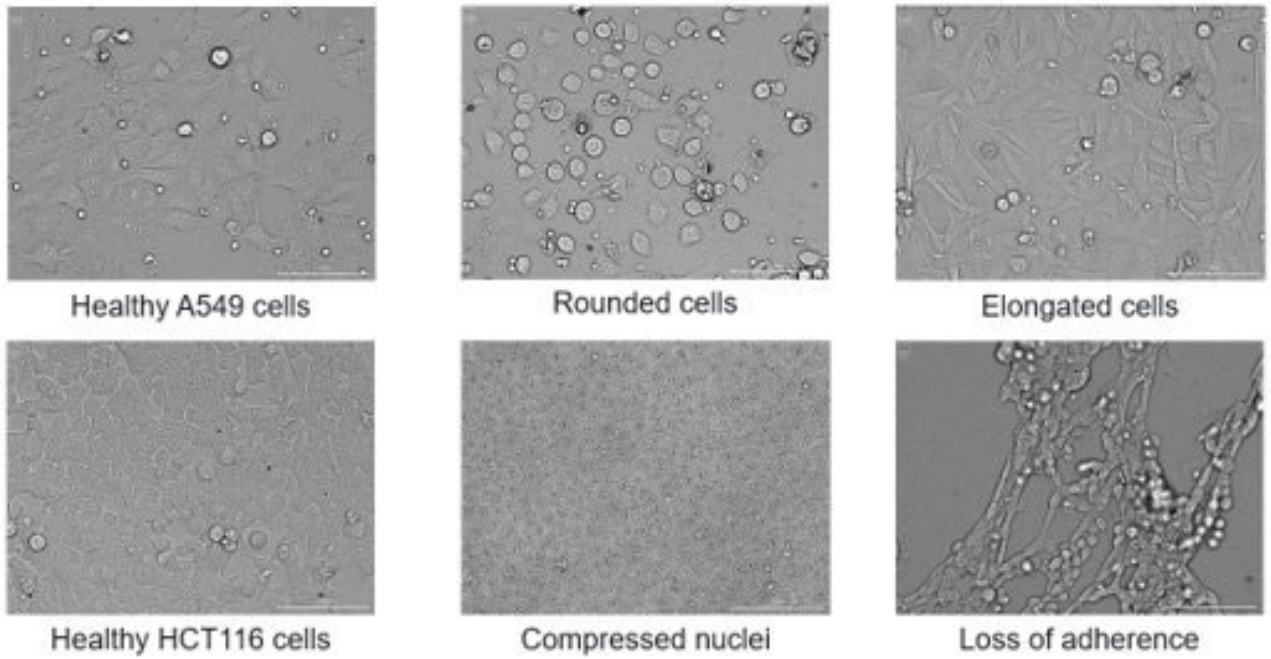
Part 2: Challenge the Cell Lines

- The isolated bacteria is labeled with a probe and added to either the lung or gut cell line
- Gut and lung cell lines were challenged by 45 bacterial strains that are known friends and foes

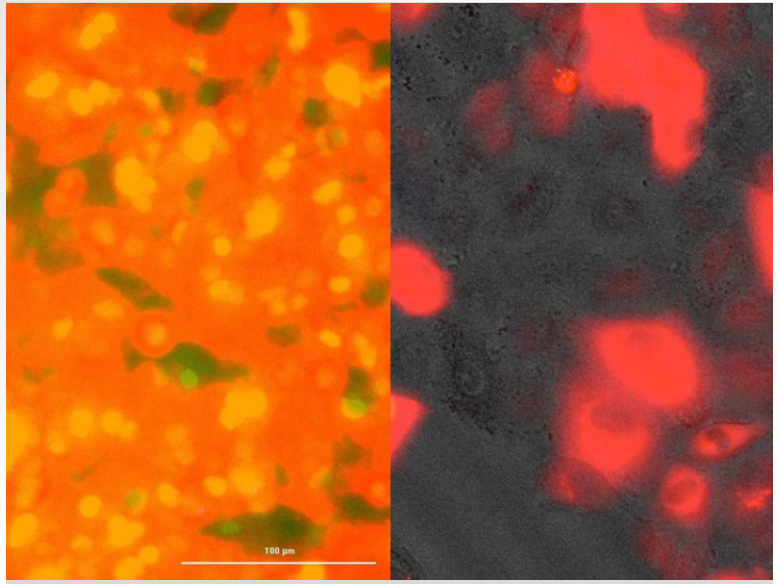


Part 3: Evaluate Pathogenicity

- Images were taken every 24 hours for several days
- Here, we can see healthy growing cells are seen to adhere strongly to the dish, but sickly cells lose structural integrity and start to lift away from the dish
- However, because of the amount of time it takes an expert to evaluate the pathogenicity a machine model was developed



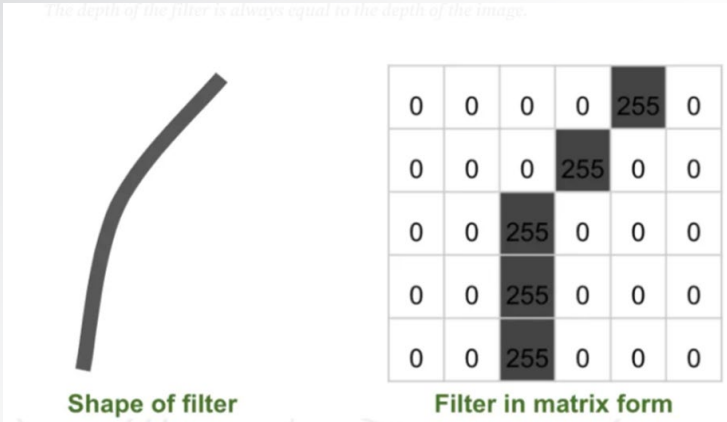
Why a machine learning model is applicable here

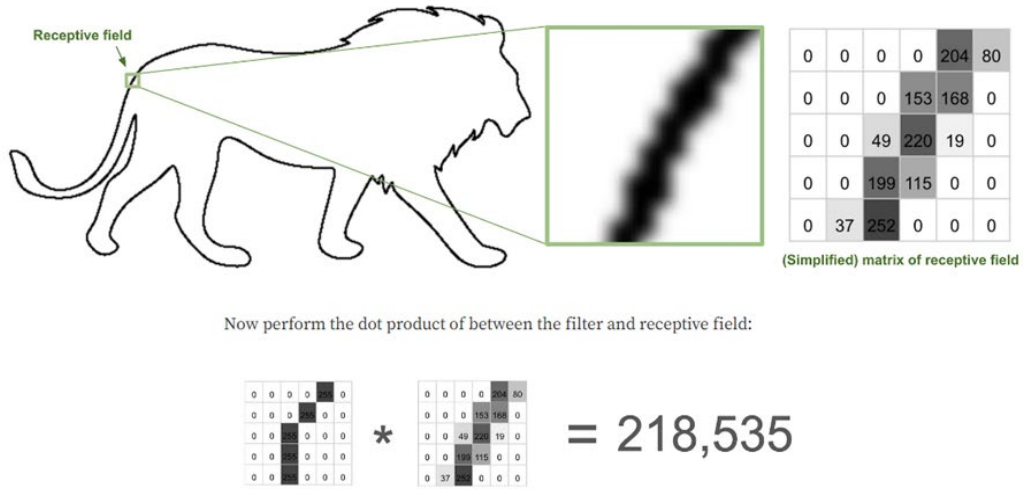


An example image from microscopy (1)

- The experiment design was able to have ground truth information by introducing known pathogen and non-pathogen bacteria
- There were a total of 14,000 images taken for training the model
- There is a well-defined question that is being asked
  - “Is there an immune response to bacteria?”
- A convolutional neural network (CNN) model was used

Overview of Convolutional Neural Network (CNN)

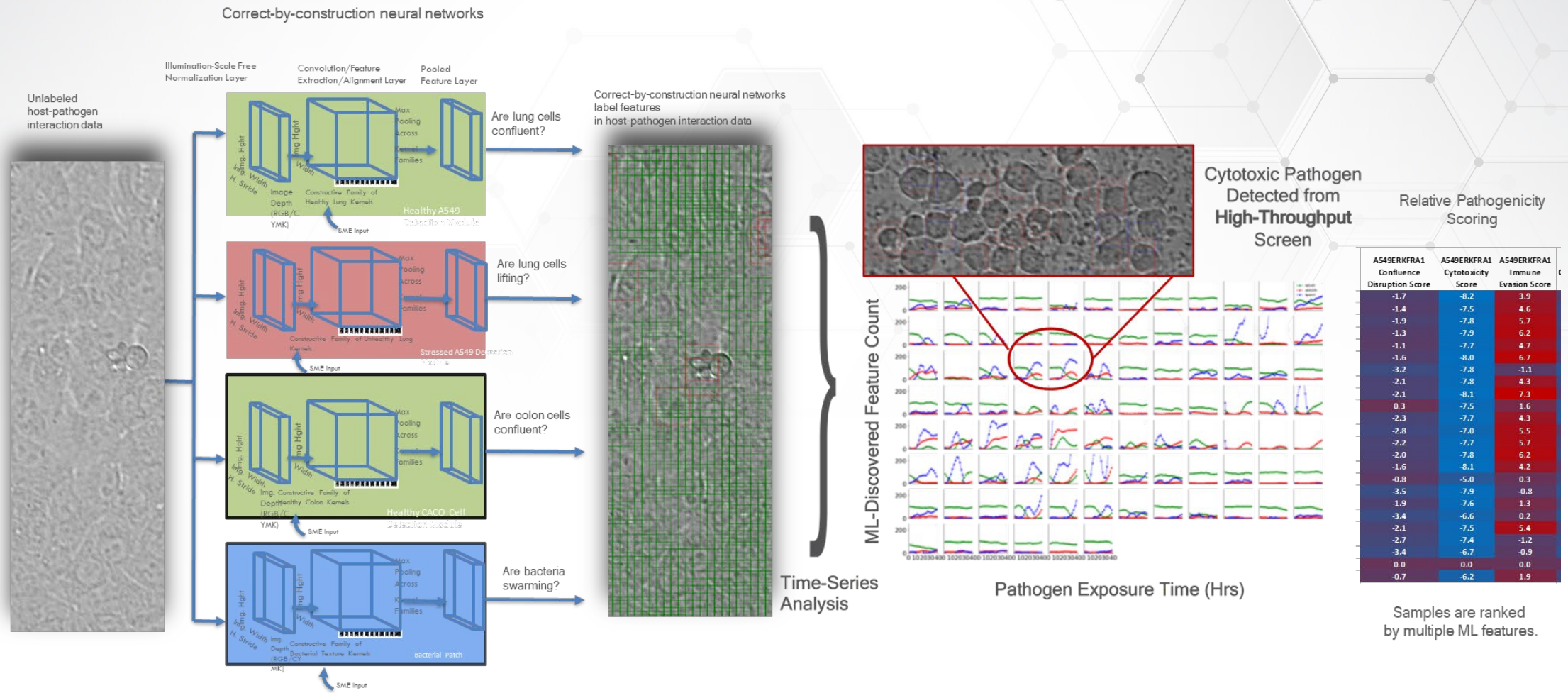




- CNN is often used for image processing models
- It is good at learning pattens and then evaluating if those pattens are found

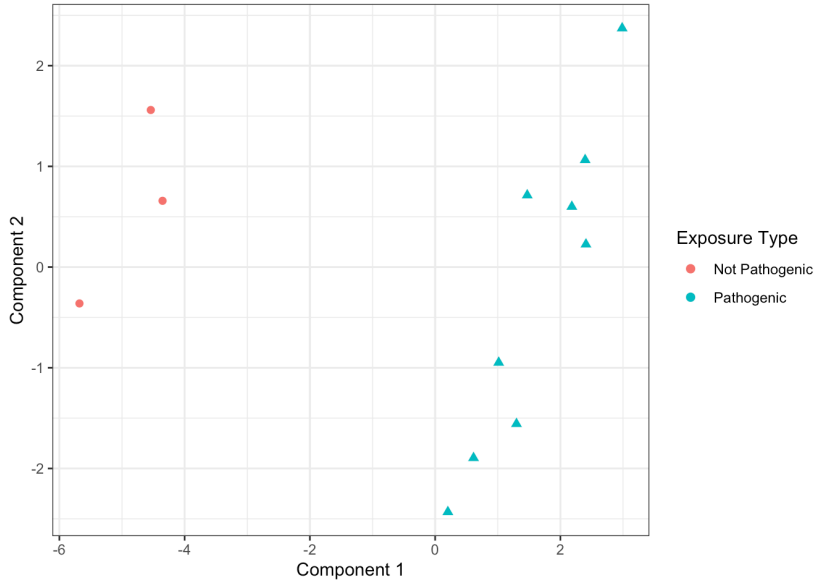
https://towardsdatascience.com/conv-nets-for-dummies-a-bottom-up-approach-c1b754fb14d6

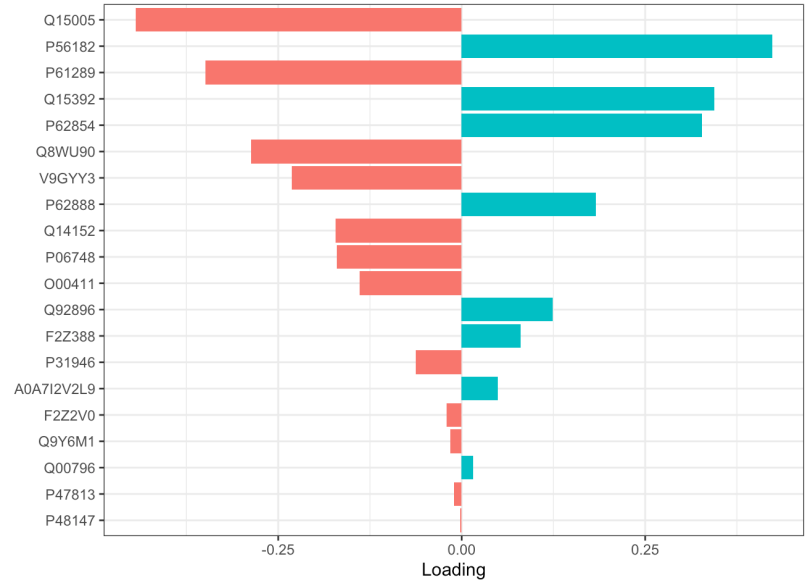
OmniScreen CNN Model Architecture



Part 4: Proteomics (Future work)

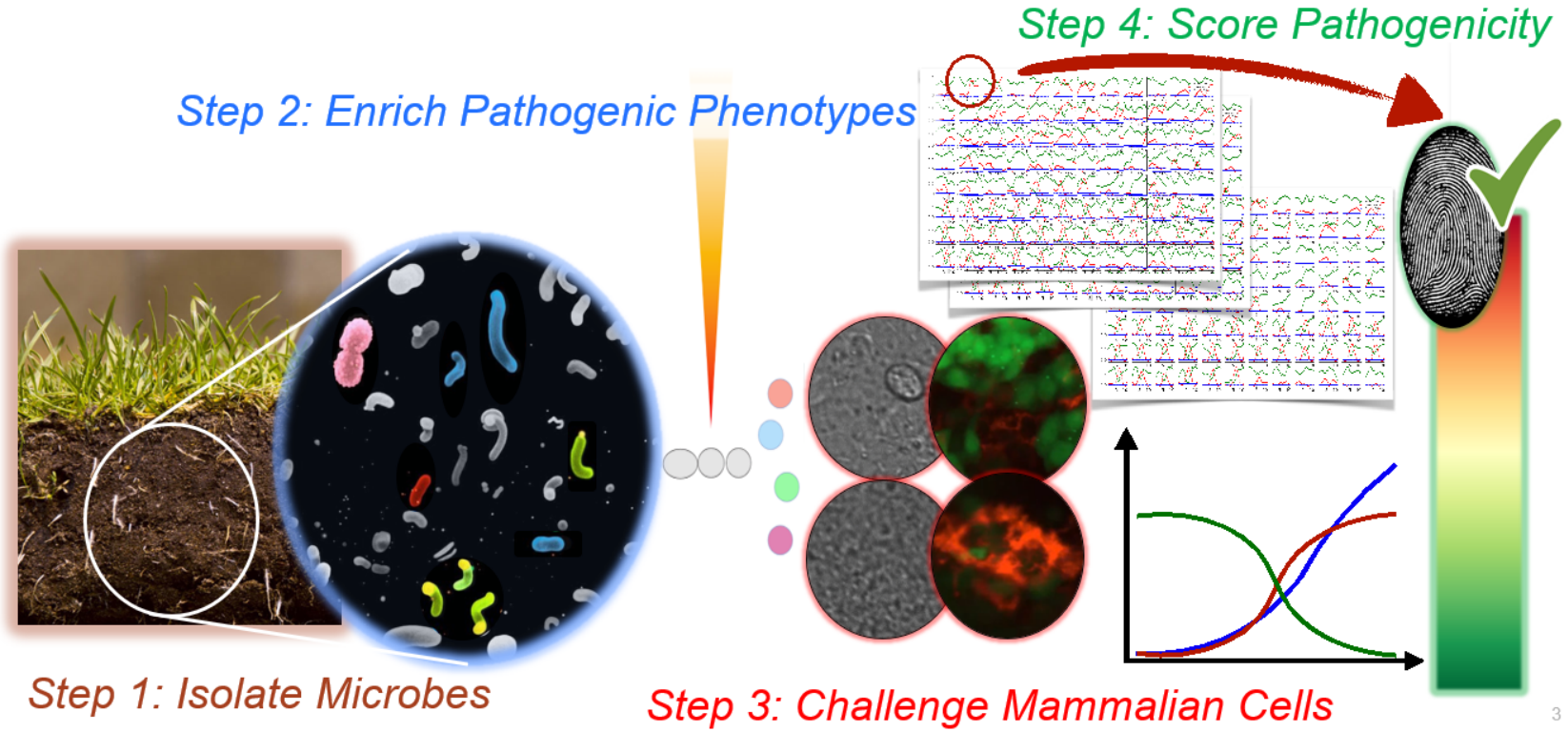
- Proteomics using mass spectrometry can yield many different protein identifications and quantifications
- These proteins can be used as the data points
- A model can be trained to look for trends that we as people are unable to tell are significant using standard differential expression methods



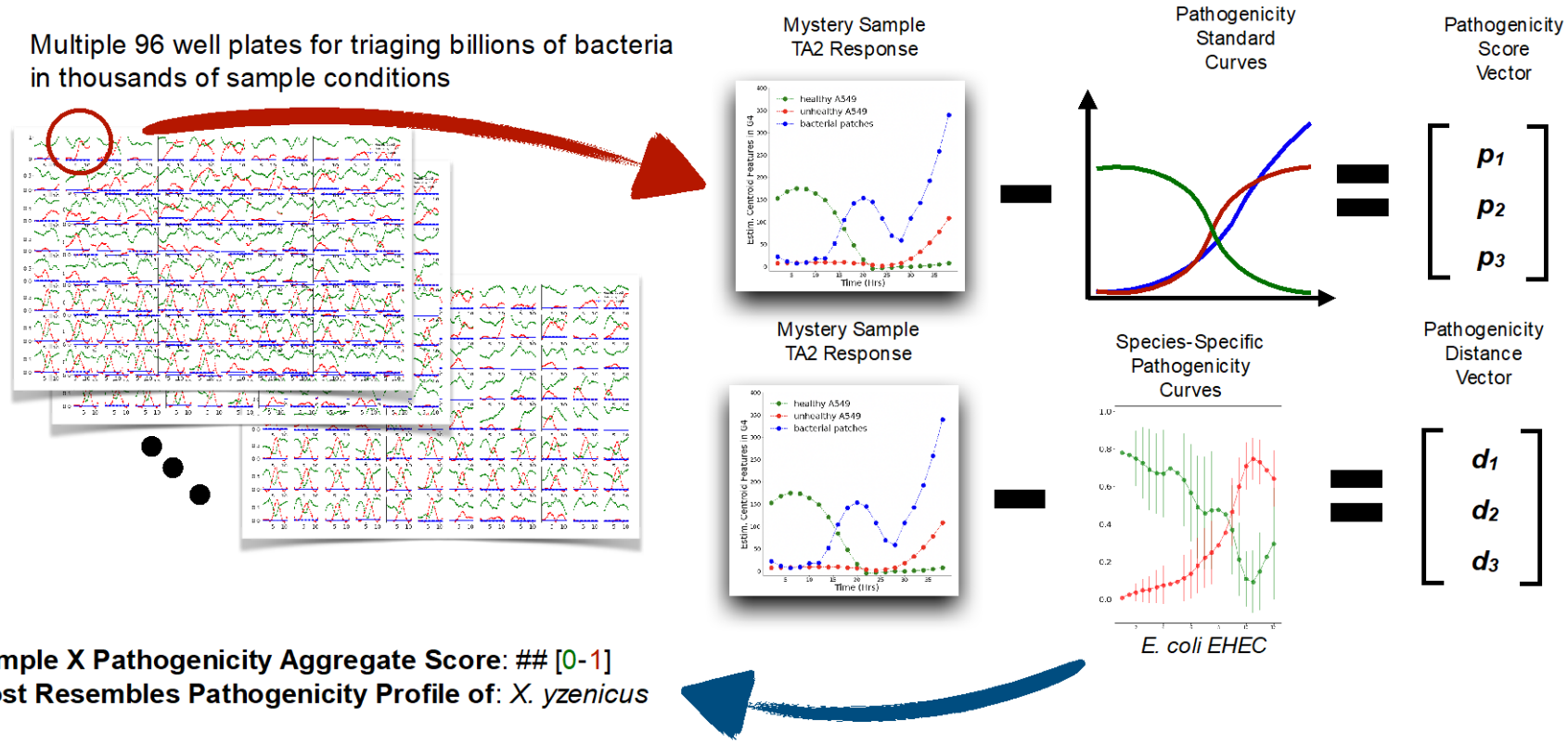


Appling the model to real world data

- Artic permafrost samples and searching for pathogenic bacteria



- The experimental samples were compared and scored against their relative functional distance from a standard reference strain



Current Results

By tracking distinct cellular features within a single sample, they were able to:

- 1) Evaluate a sample's capacity for immune cell evasion, by quantifying bacterial growth in the presence of immune cells
- 2) Evaluate a sample's cytotoxicity, by quantifying the increase in unhealthy epithelial cells over time
- 3) Evaluate a sample's natural capacity to colonize epithelial culture, by quantifying the natural growth of the bacterial sample.
- Approximately 22 of the samples exhibited pathogenic response across multiple feature dimensions in the presence of a simulated immune response.

Sources

https://www.pnnl.gov/news-media/glowing-progress-pathogen-discovery

OmniScreen team:

Becky Hess,
PNNL

Rob Egbert,
PNNL

Enoch Yeung,
UC Santa Barbara

Questions?


Networking Break
10:15 – 10:30 a.m.

8:30-9:45 a.m.	Introduction to Machine Learning for Biological Applications	Sam Dixon
9:45-10:15	Machine Learning in Pathogen Discovery	Isabelle O'Brien
10:15-10:30	Networking Break	
10:30-11:30	**The What and Why of Multi-Omics Integration**	**Daniel Claborne**
11:30-12:15	Case Study with MixOmics	Samantha Erwin

EMSL

The What and Why of Multi-Omics

Daniel Claborne
Data Scientist

Intro

Motivation

What is multi-omics and why is it important?

What are some methods for extracting insight from multi-omics?

What motivates the choice of different methods? I'll give some examples that hopefully give some intuition about this.

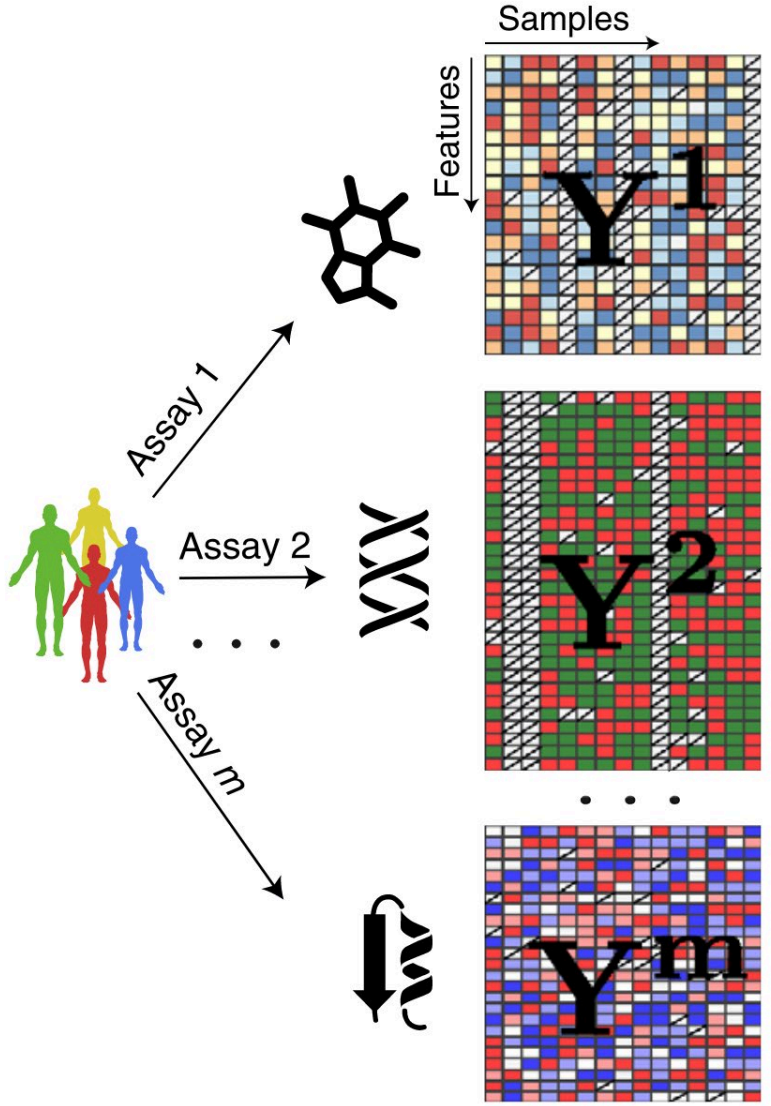
Intro

What is Multi-omics?

What is Multi-omics?

- What it sounds like!

Samples are taken across multiple different biomolecule types or “views”.



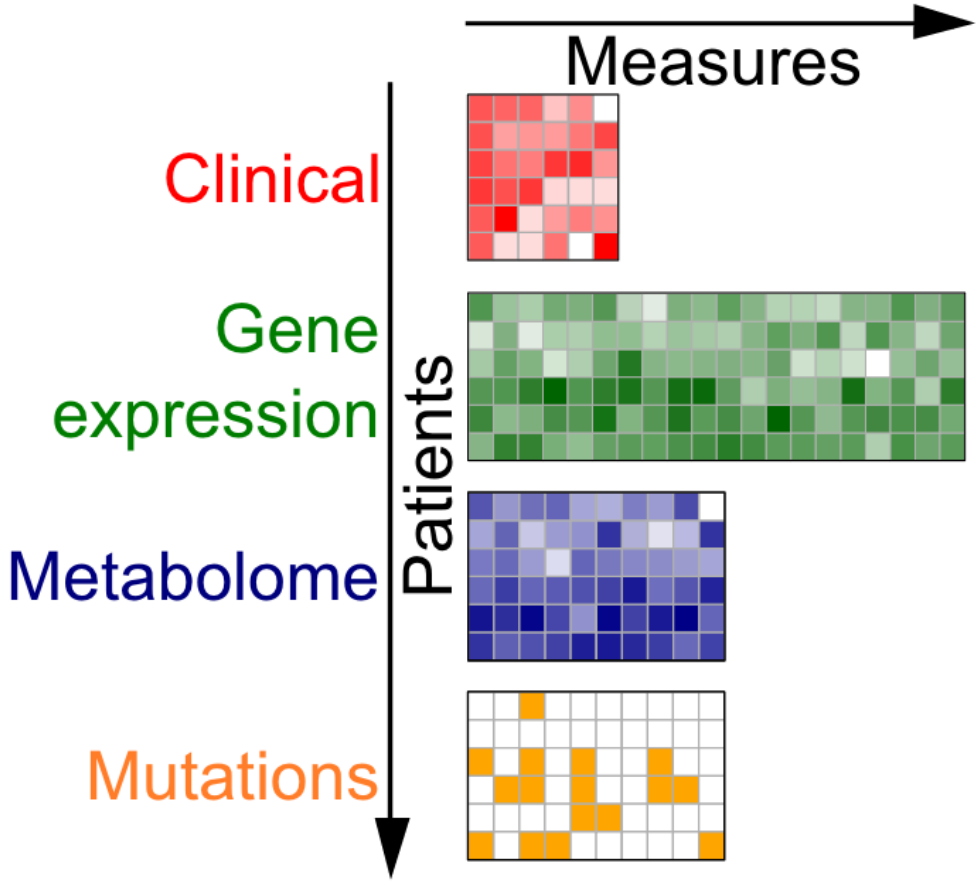
R. Argelaguet *et al.*, “Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets,” *Molecular Systems Biology*, vol. 14, no. 6, p. e8124, Jun. 2018, doi: 10.15252/msb.20178124.

What is Multi-omics?

- What it sounds like!

The ‘Views’ or sets of features for each sample can come from a variety of sources, including `omics, clinical measures (e.g. blood pressure), or even more complex data such medical scans.

Here we’ll focus on the `omics piece of the input.



S. Pai, S. Hui, R. Isserlin, M. A. Shah, H. Kaka, and G. D. Bader, “netDx: interpretable patient classification using integrated patient similarity networks,” *Molecular Systems Biology*, vol. 15, no. 3, p. e8497, Mar. 2019, doi: 10.15252/msb.20188497.

Intro

What is Multi-omics?

- More concretely, we take these features and try to analyze them “simultaneously”
- More complete understanding of the biological system
- More powerful predictors for use cases such as cancer subtype prediction.

Intro

What is Multi-omics?

Problems?

- Large numbers of features in an already usually p >> n problem
- Different data types can be hard to combine (e.g. continuous vs count data)
- Missing value problem is exacerbated by completely missing views
- Nature of the relationship between datasets can be difficult to model

Integration

How can we combine the different views?

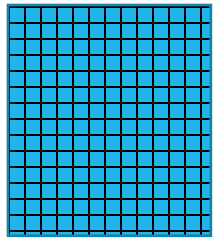
A good high-level partitioning of the ways we categorize integration schemes is early, middle, and late integration.

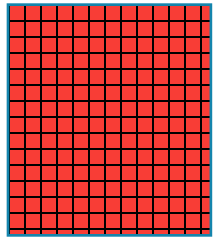
Other categorizations include ‘mixed’, ‘intermediate’ and ‘hierarchical’.

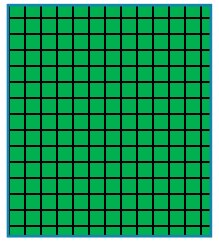
The point of these schemes is to force the algorithm to learn in a certain way that hopefully reflects actual biological interactions.

Early Integration

Early Integration

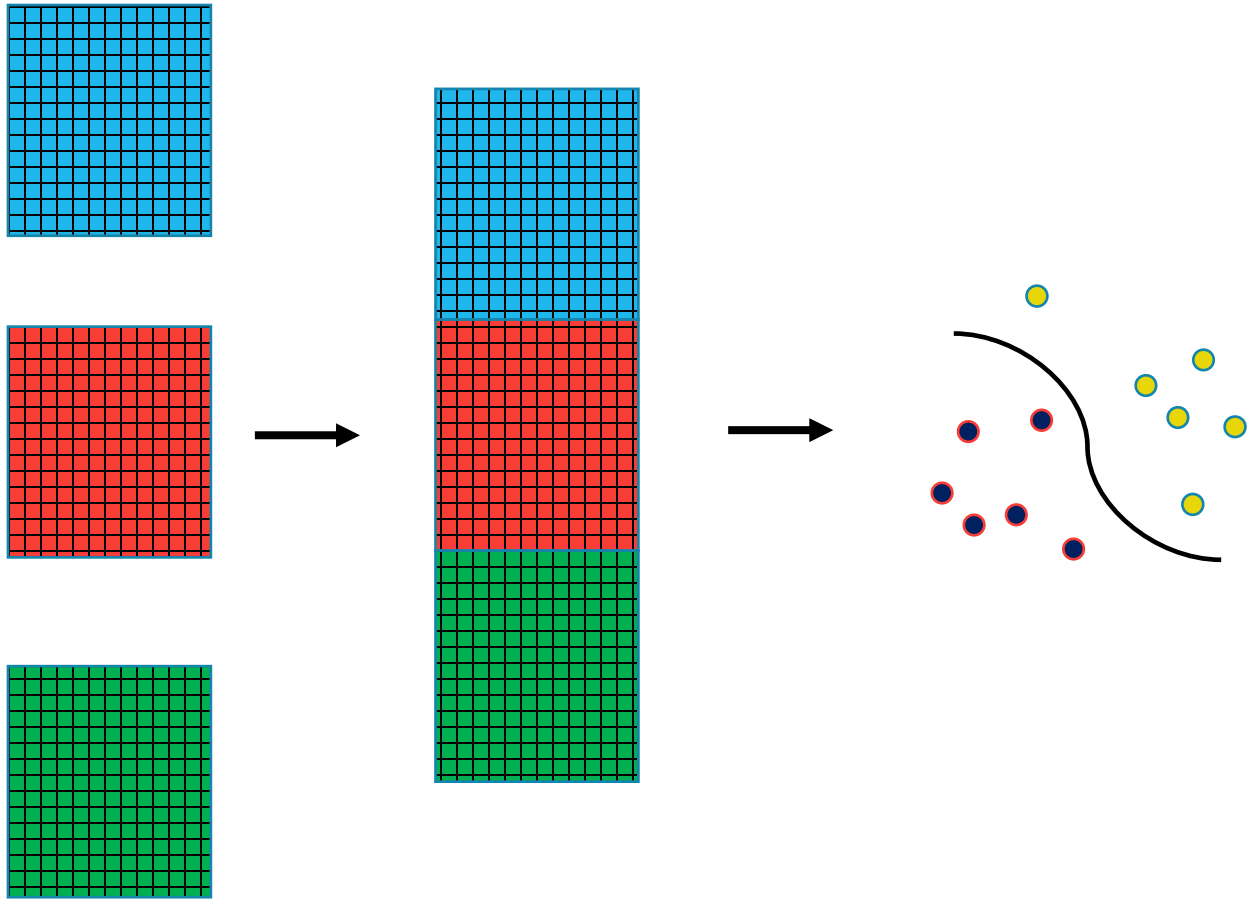






Early Integration

Concatenate and predict



Early Integration

Problems?

- Imbalance in the data may cause learning algorithms to pay too much attention to one of the views.
- Ignores data-specific distributions of the views.
- The missingness of a particular view is problematic.
- High dimensionality problem is exacerbated.

Middle Integration

Intermediate/Mixed Integration (Middle Integration)

The next reasonable option is to combine `omics at some ‘middle’ point.

Distinction between methods is subtle, and it is better to talk about specific cases rather than a whole class of algorithms.

Middle Integration

Intermediate – Separate Networks

Plan 1: Separately process each network and then combine through more statistical learning or heuristics.

This has the advantages of (usually) performing dimensionality reduction and allowing for the removal of data differences across views.

Middle Integration

Intermediate representation methods

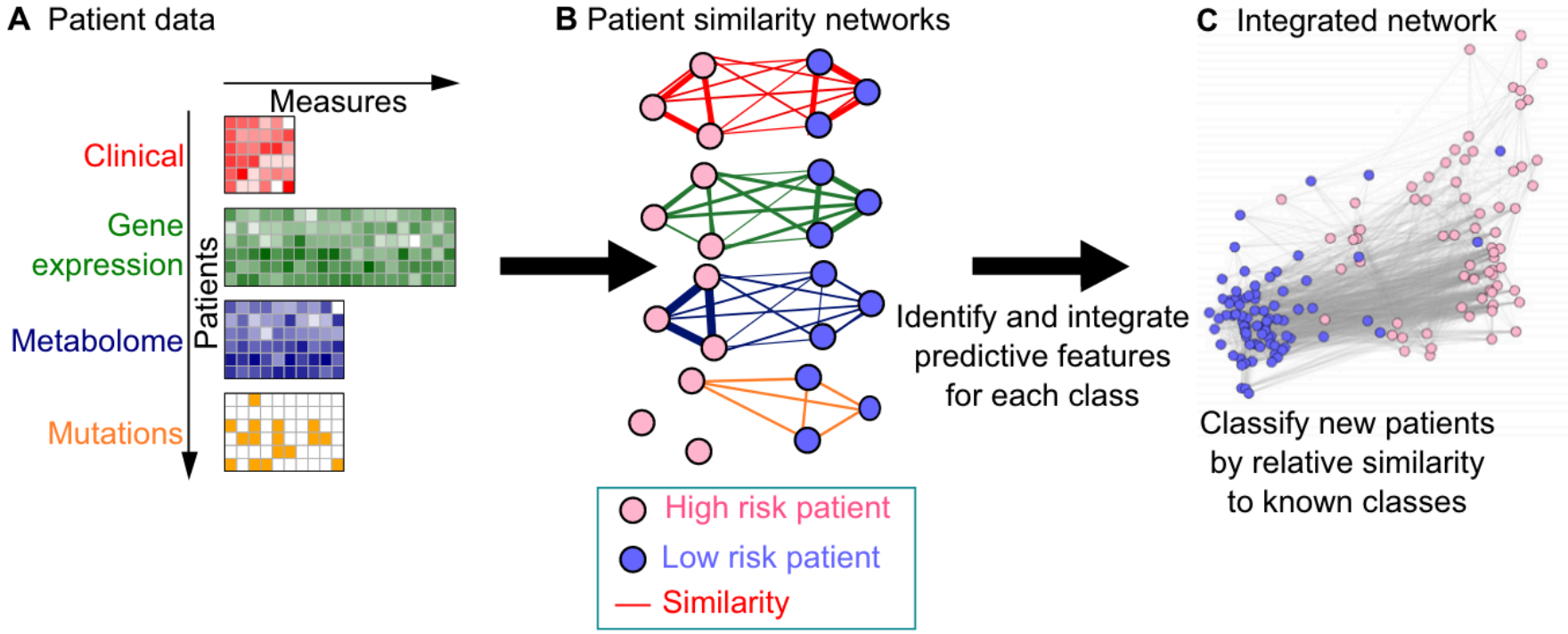
Plan 1: Separately process each network and then combine through more statistical learning or heuristics.

Some forms of intermediate representation include:

- Kernels
- Graphs
- Neural Network Layers

Middle Integration

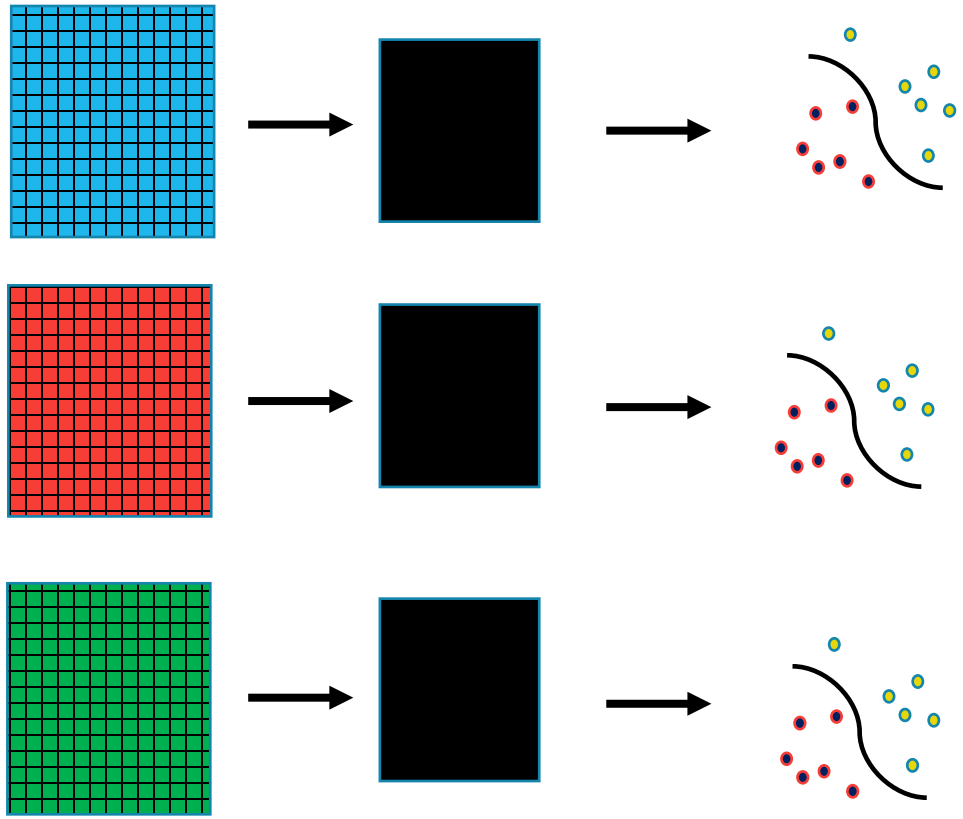
A graph intermediate representation.



S. Pai, S. Hui, R. Isserlin, M. A. Shah, H. Kaka, and G. D. Bader, "netDx: interpretable patient classification using integrated patient similarity networks," *Molecular Systems Biology*, vol. 15, no. 3, p. e8497, Mar. 2019, doi: 10.15252/msb.20188497.

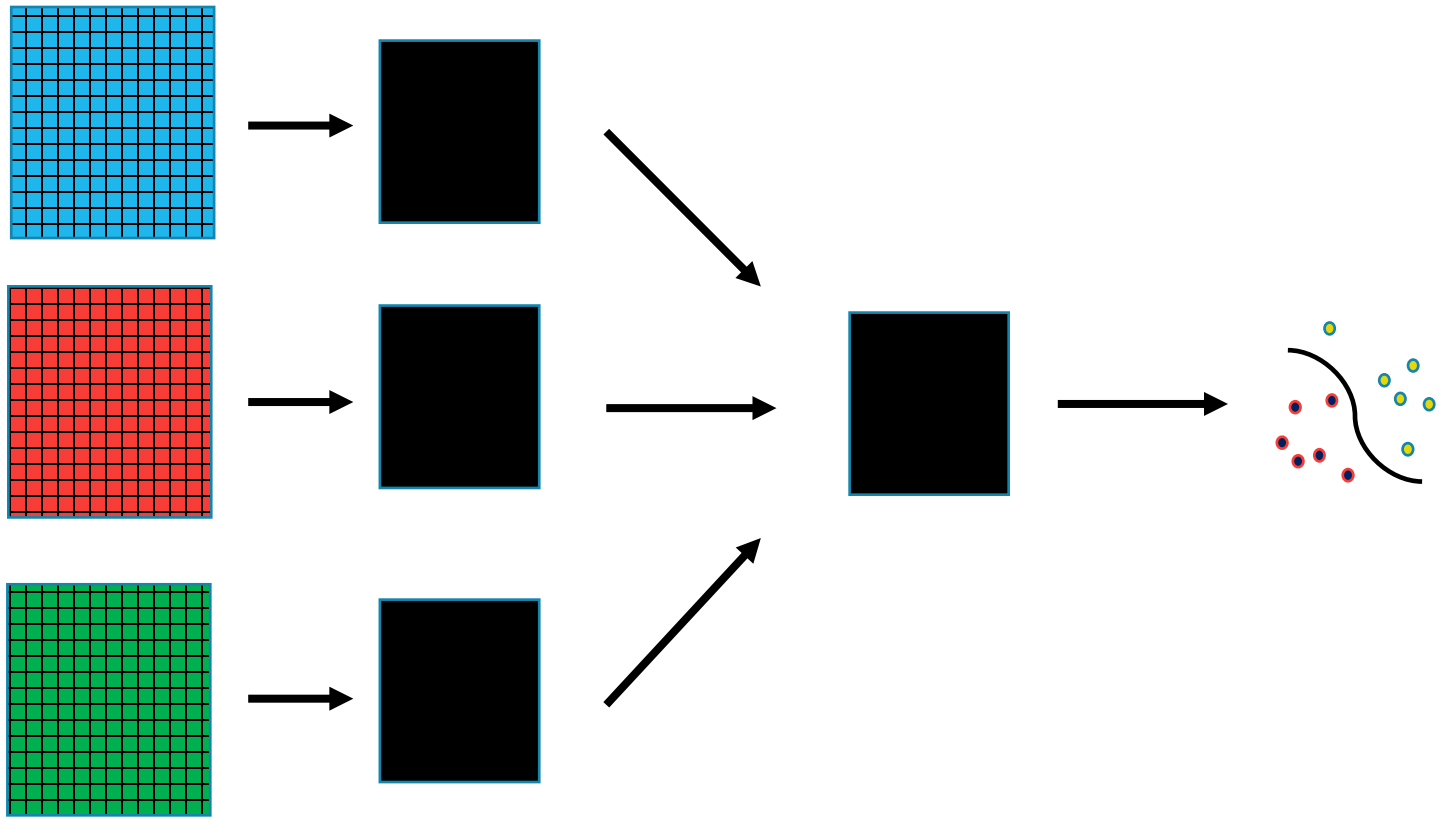
Middle Integration

Example of modeling considerations



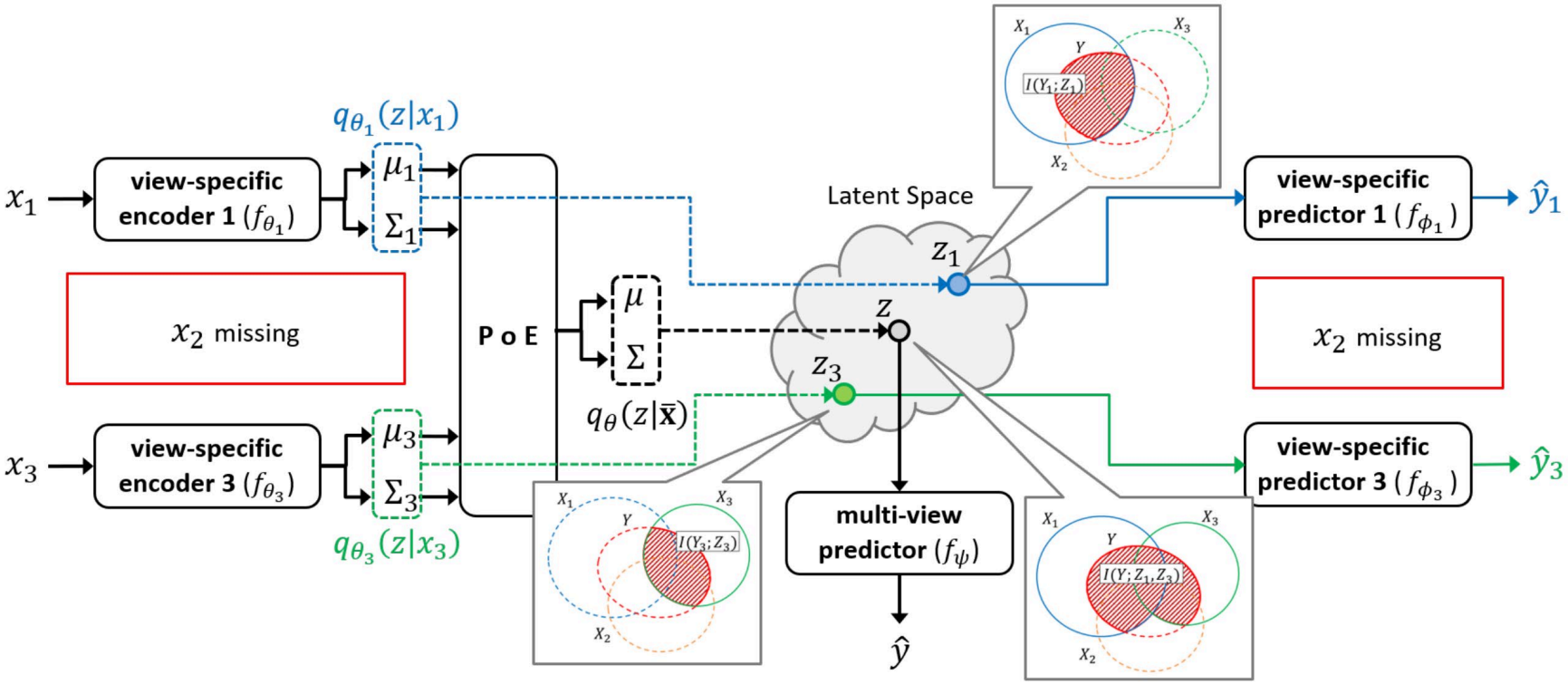
Middle Integration

Example of modeling considerations



Middle Integration

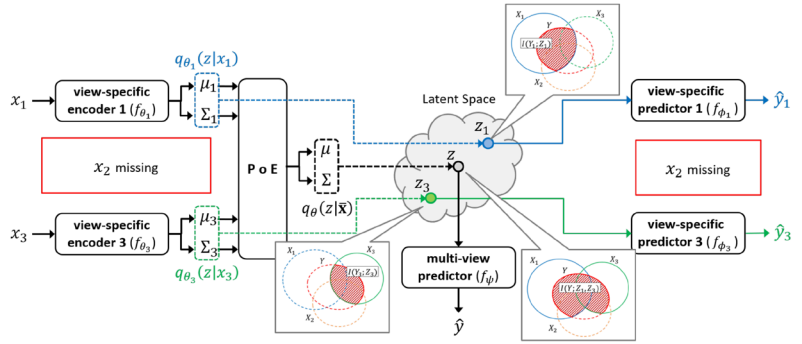
A Deep-Learning Based Intermediate Representation



Lee, C. & van der Schaar, M. A Variational Information Bottleneck Approach to Multi-Omics Data Integration. Preprint at https://doi.org/10.48550/arXiv.2102.03014 (2021).

Middle Integration

A Deep-Learning Based Intermediate Representation



- Combines each view through a 'product of experts'
- Handles the view-missing problem by training in the presence of missing views.
- Requires view-specific predictors to also perform well by themselves.

Middle Integration

Intermediate/Mixed Integration

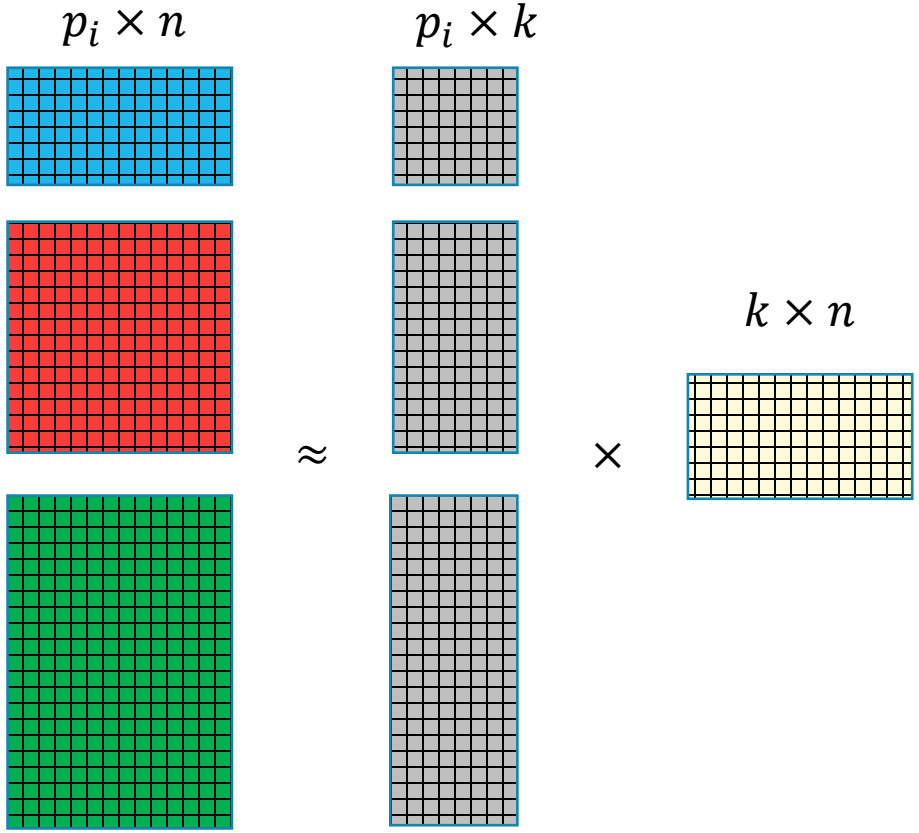
Plan 2: Assume that all views share some common ‘latent space’. Almost invariably some matrix factorization method.

Advantages are that is does not need prior transformation to learn the shared representation, however it is more sensitive to heterogeneity in the data.

Middle Integration

Intermediate/Mixed Integration

Plan 2: Assume that all views share some common ‘latent space’. Almost invariably some matrix factorization method.



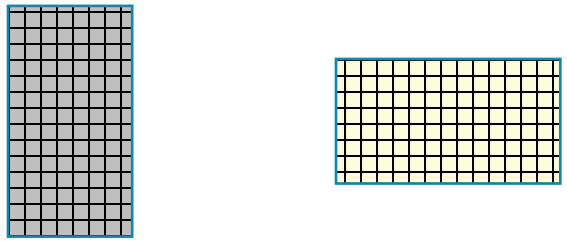
Middle Integration

Intermediate/Mixed Integration

Plan 2: Assume that all views share some common ‘latent space’. Almost invariably some matrix factorization method.

Answer questions such as

- How much variance is explained by each hidden factor?
- Which factors drive variation across multiple views? Only one view?



Middle Integration

Intermediate/Mixed Integration

Plan 2: Assume that all views share some common ‘latent space’.

Issues?

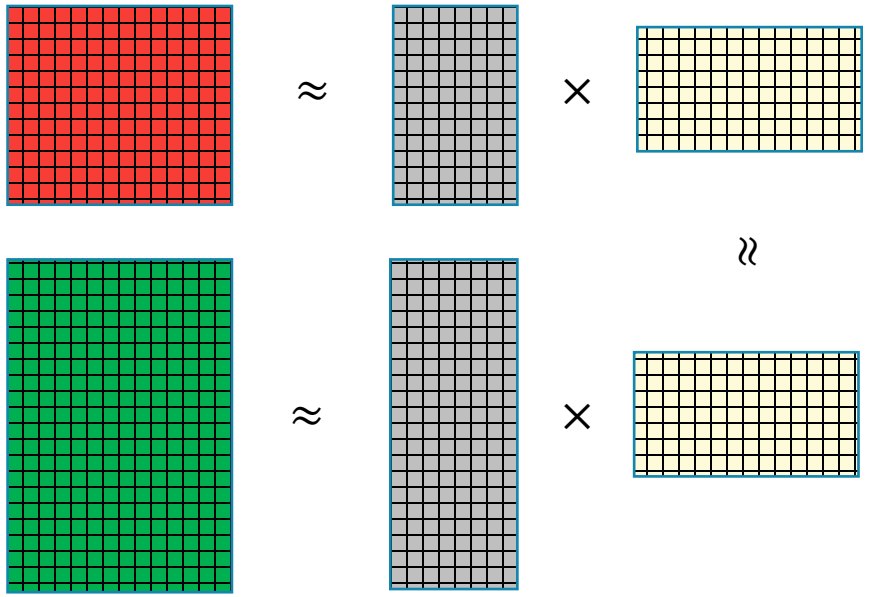
- Sensitive to differences in the views
- Assumption of common latent space reasonable?

Middle Integration

Intermediate/Mixed Integration

Plan 2: Assume that all views share some common ‘latent space’.

Preview: SPLS, look for common sources of variation in two datasets.

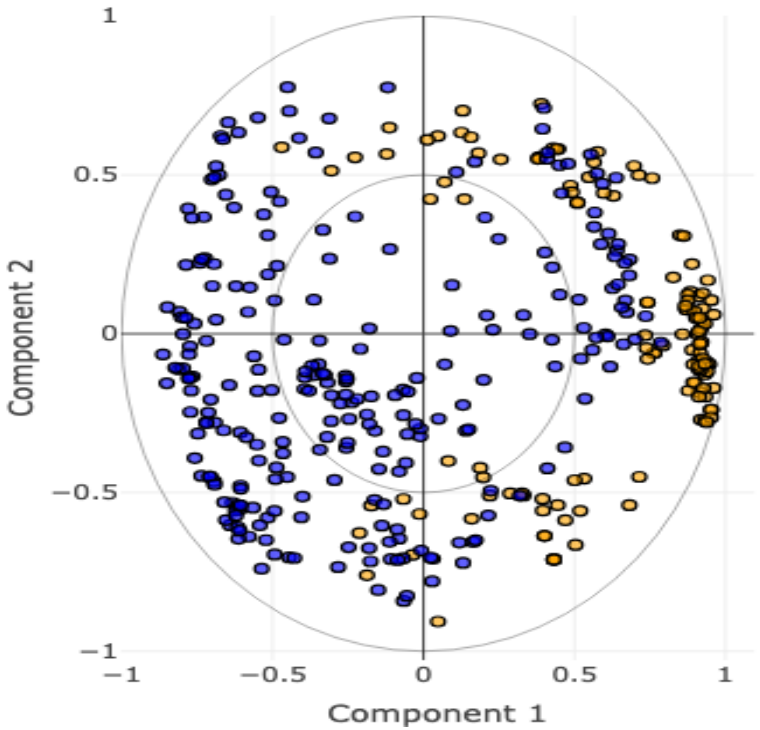


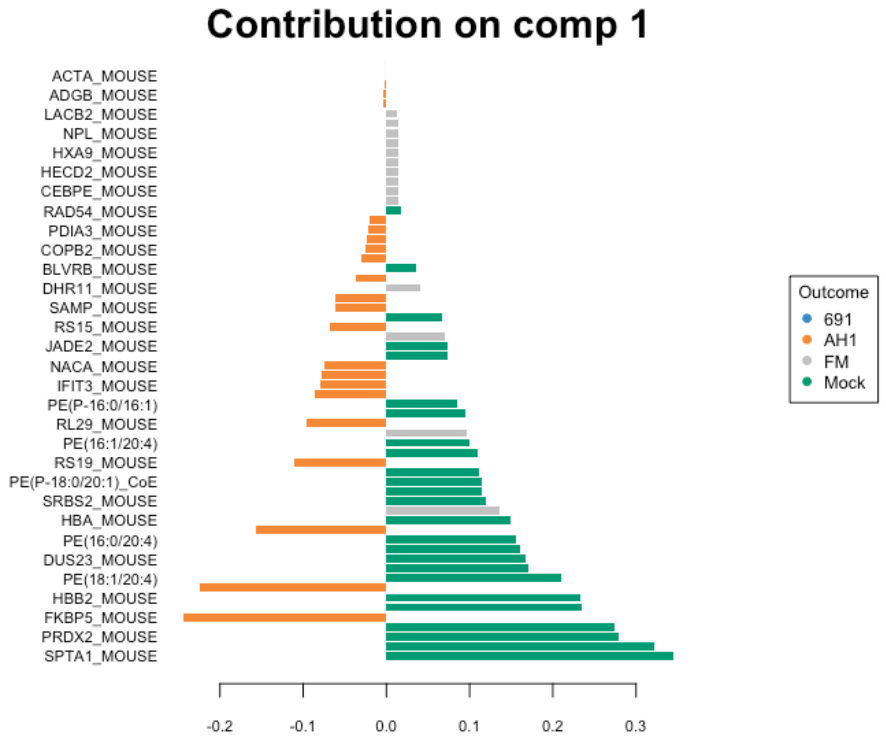
Middle Integration

Intermediate/Mixed Integration

Plan 2: Assume that all views share some common ‘latent space’.

Preview: SPLS, look for common sources of variation in two datasets.

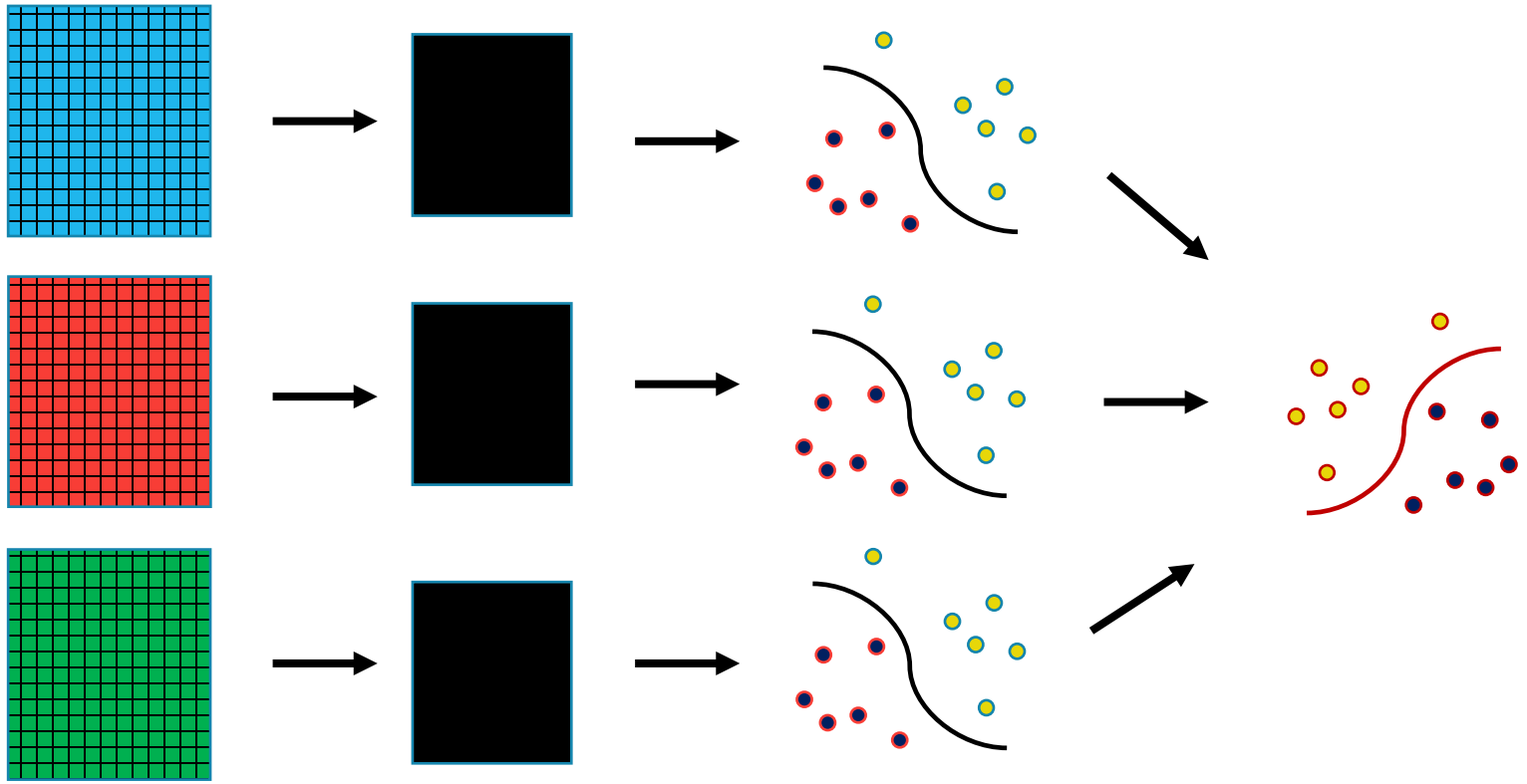




Late Integration

Late integration

Run each view through its own model and aggregate the results at the end.



Late Integration

Late integration pros and cons

Run each view through its own model and aggregate the results at the end.

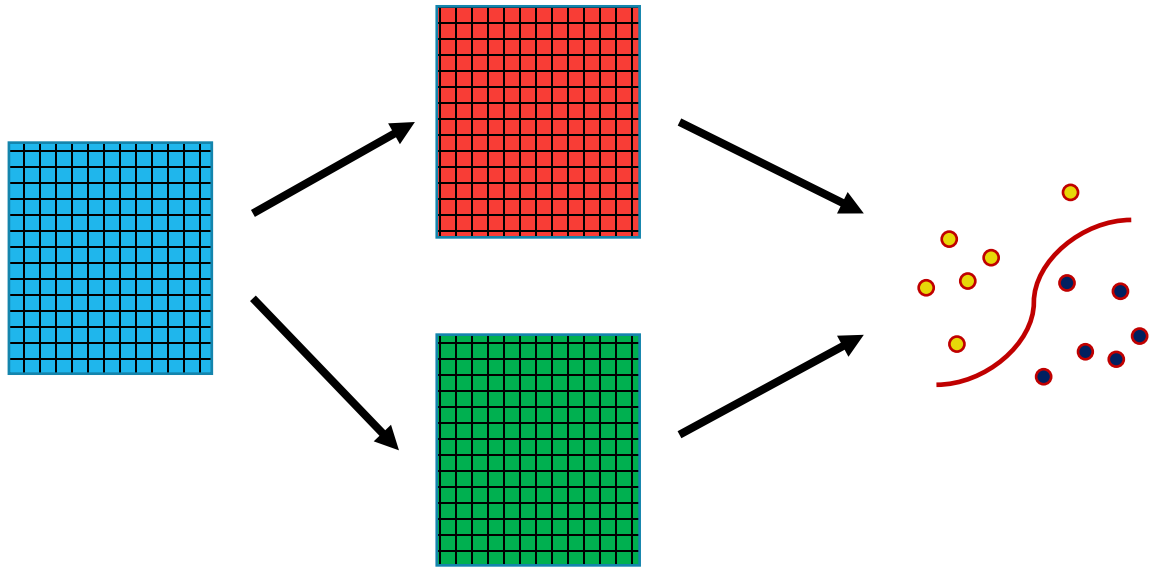
- Can easily utilize existing tools for single-omics and combine results.
- Possibly achieves the best performance on things like prediction tasks.
- Doesn’t learn relationships between the different omics.
- Fairly unpalatable from an explainability standpoint.

Hierarchical Integration

Modeling hierarchical relationships

Sometimes we might want to model the ‘direction’ of causality.

For example, we might want to specify that genes affect the outcome *through* proteins.



Conclusion

- Multi-omics is crucial in understanding complex biological systems.
- There is great diversity in the goals of multi-omics studies, as well as the methods to attack each problem.
- The complexity of relationships between different levels of the omics hierarchy requires careful modeling considerations so as to avoid spurious conclusions.

Questions?

8:30-9:45 a.m.	Introduction to Machine Learning for Biological Applications	Sam Dixon
9:45-10:15	Machine Learning in Pathogen Discovery	Isabelle O'Brien
10:15-10:30	Networking Break	
10:30-11:30	The What and Why of Multi-Omics Integration	Daniel Claborne
11:30-12:15	**Case Study with MixOmics**	**Samantha Erwin**

EMSL

MixOmics Case Study

Samantha Erwin
Data Scientist

Overview

- My background
- *C. difficile* introduction
- Experimental design
- Data
- Analysis:
  - MixOmics
  - Glasso
  - Mechanistic Model

My Background

- Undergraduate:
  - B.S. Mathematics, Murray State University
  - Undergraduate research
- Graduate:
  - M.S./Ph.D. Mathematics, Virginia Tech
  - Applied math focus
  - CompBio Summer School
  - Summer Intern at Los Alamos National Lab and
- Postdoc:
  - Population Health, NC State Vet School
  - Computational modeling group

- Computational Scientist at ORNL: 2019-2022
- Data Scientist and Team Lead at PNNL: 2022 - Current

AMERICAN SOCIETY FOR MICROBIOLOGY | mSphere®

RESEARCH ARTICLE
Host-Microbe Biology

Shifts in the Gut Metabolome and *Clostridium difficile* Transcriptome throughout Colonization and Infection in a Mouse Model

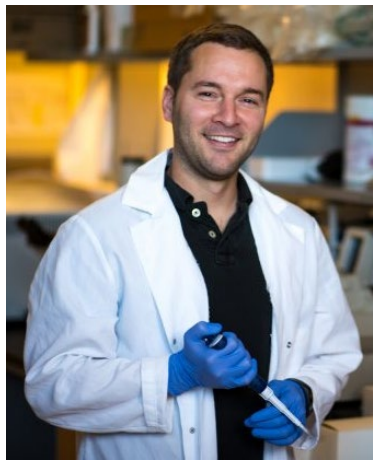
Joshua R. Fletcher,[a] Samantha Erwin,[a] Cristina Lanzas,[a] Casey M. Theriot[a]

AMERICAN SOCIETY FOR MICROBIOLOGY | mSphere®

RESEARCH ARTICLE
Host-Microbe Biology

Shifts in the Gut Metabolome and *Clostridium difficile* Transcriptome throughout Colonization and Infection in a Mouse Model

Joshua R. Fletcher,[a] Samantha Erwin,[a] Cristina Lanzas,[a] Casey M. Theriot[a]



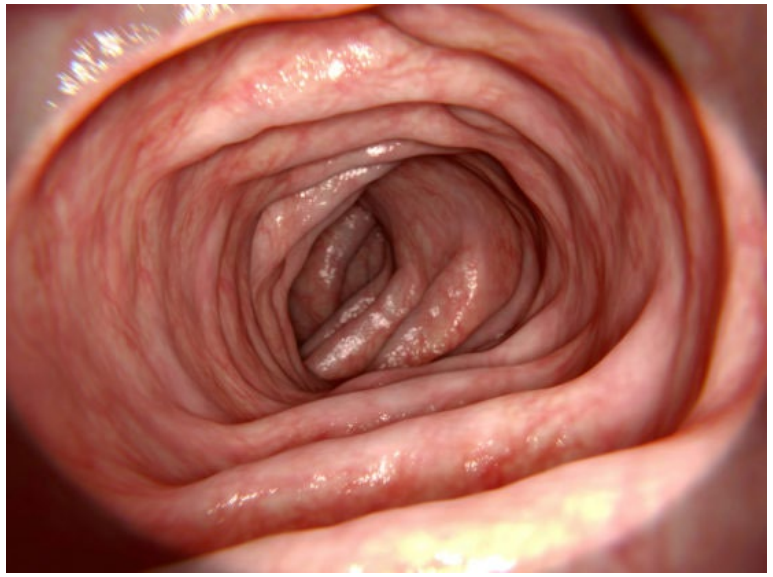
Josh Fletcher

Cristina Lanzas

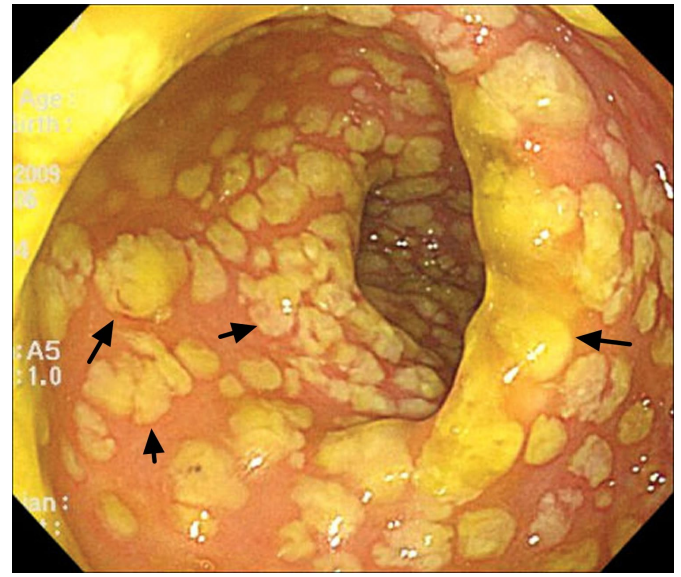
Casey Theriot

Introduction

- *C. difficile* is the most common cause of healthcare-acquired infection in humans
- Antibiotics deplete your healthy gut microbiota & if exposed to *C. difficile* you'll get infected
- Diarrhea, pseudomembranous colitis, and death
- Recurrence in 30% of cases

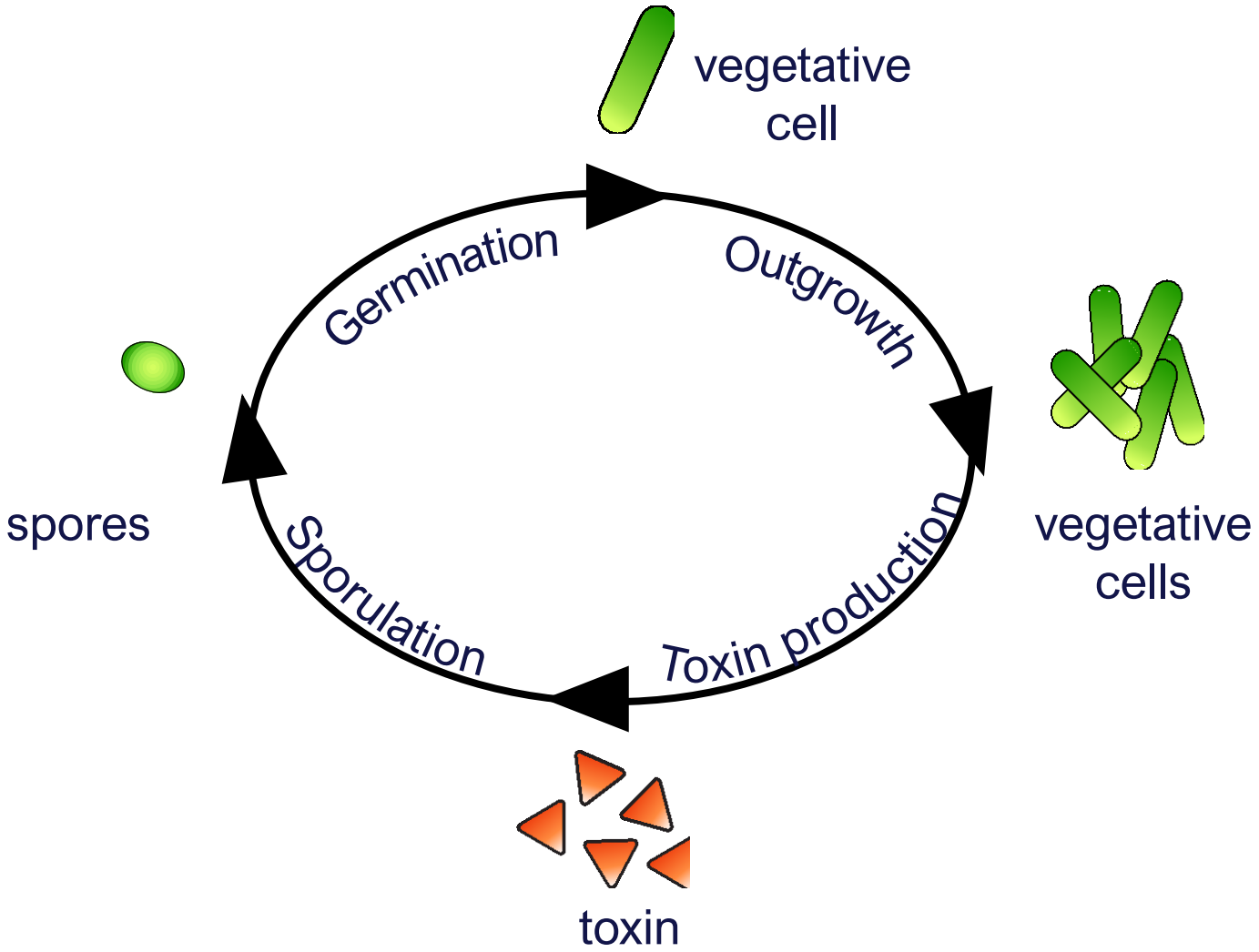


Healthy colon



Pseudomembranous colitis

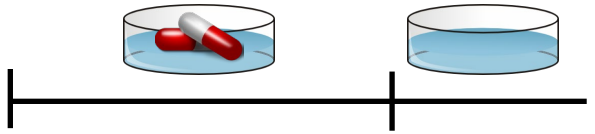
C. difficile lifecycle

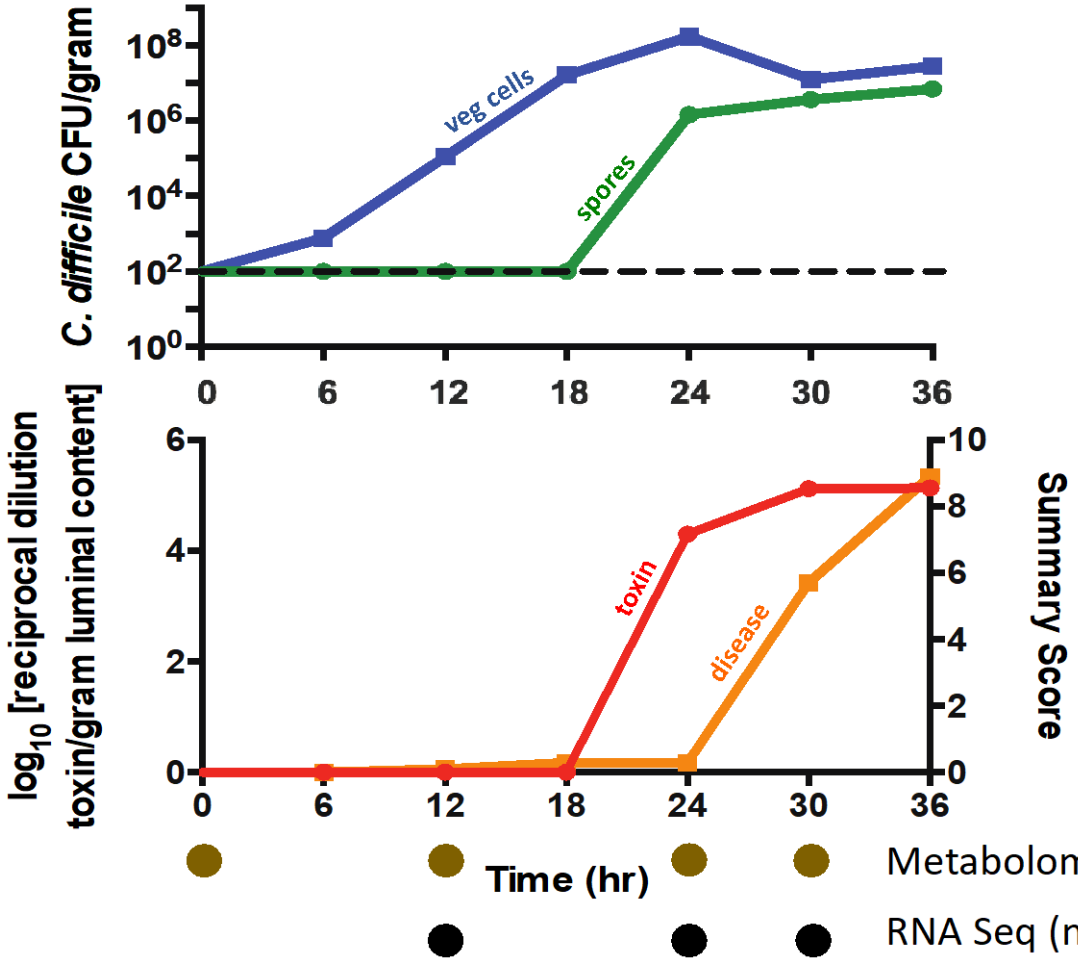


Toxin production

- Toxins function by damaging the intestinal mucosa and cause symptoms of C. difficile infection
- Toxin production appears to be triggered by nutrient depletion
- Few studies have explored the complex environment of the antibiotic-depleted host gut in vivo
- We want to mathematically study what specific mechanisms are leading to toxin production.

C. difficile life cycle in a mouse model





Metabolomics (n=8 per time point)

RNA Seq (n=3-4 per time point)

Experimental design

- Mice were treated with the antibiotic cefoperazone and challenged with *C. difficile* VPI 10463 2 days later (0 hour)
- Mice were euthanized at 0 hour (pre-infection), 12 hour, 24 hour, and 30 hour.
- Cecal contents were analyzed via untargeted metabolomics at 0 hour, 12 hour, 24 hour, and 30 hour with a sample size of n=8 at each time point.
- RNAseq (transcriptomics) of *C. difficile* was completed from paired cecal contents at 12 hour, 24 hour, and 30 hour with a sample size of n=3-4 at each time point.
- $$$$$

Metabolomics

- Metabolites are small molecules that are the intermediate products of metabolic reactions
- Metabolomics provide measurements of every metabolite in the host metabolome
- Cecal samples were processed by Metabolon where metabolites were identified by automated comparison of the ion features to a reference library of chemical standard entries.
- Measured 638 unique metabolites on 32 samples

METABOLON®

Where **knowing** comes to **life**™

Missing data

- 9.7% of the metabolomics matrix entries are missing
- Data could be missing because:
  - Sample preparation error
  - Below limit of detection
  - The value is truly 0
- Previous studies compared multiple methods for filling in missing data and found K–nearest neighbor algorithm to be the most robust (Do et al, 2018)
- We use the samples at each time point as the nearest neighbors

Transcriptomics

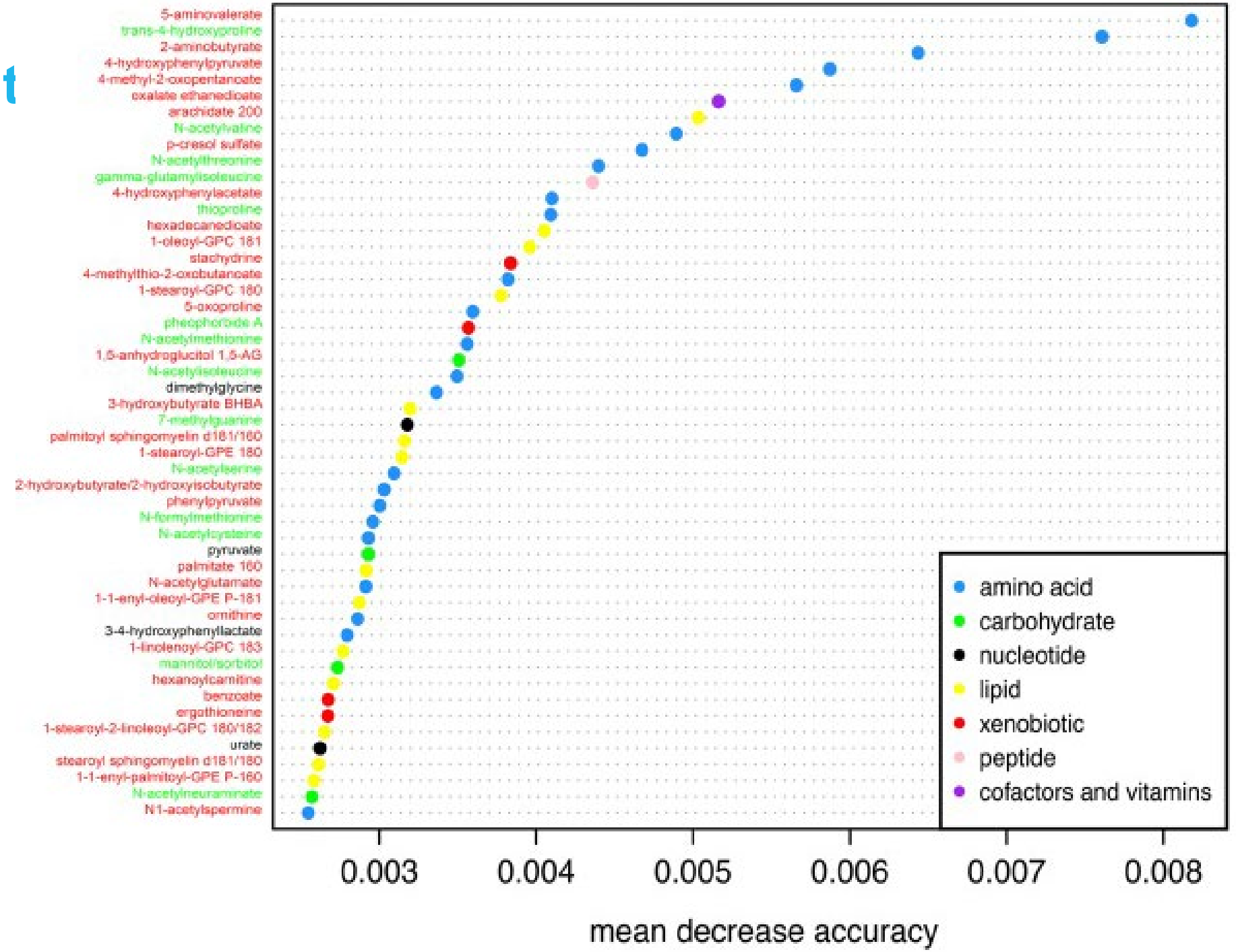
- The set of all RNA molecules in *C. difficile*
- Whole transcriptome shotgun sequencing approach
- RNAseq (transcriptomics) of *C. difficile* was completed at the University of Michigan DNA Sequenceing Core
- Measured 3769 genes across 11 samples
  - These 11 samples are a subset of the same samples from the metabolomics data.
  - 12 hour (n=3), 24 hour (n=4), and 30 hour (n=4)

Results

Current data techniques with metabolomics

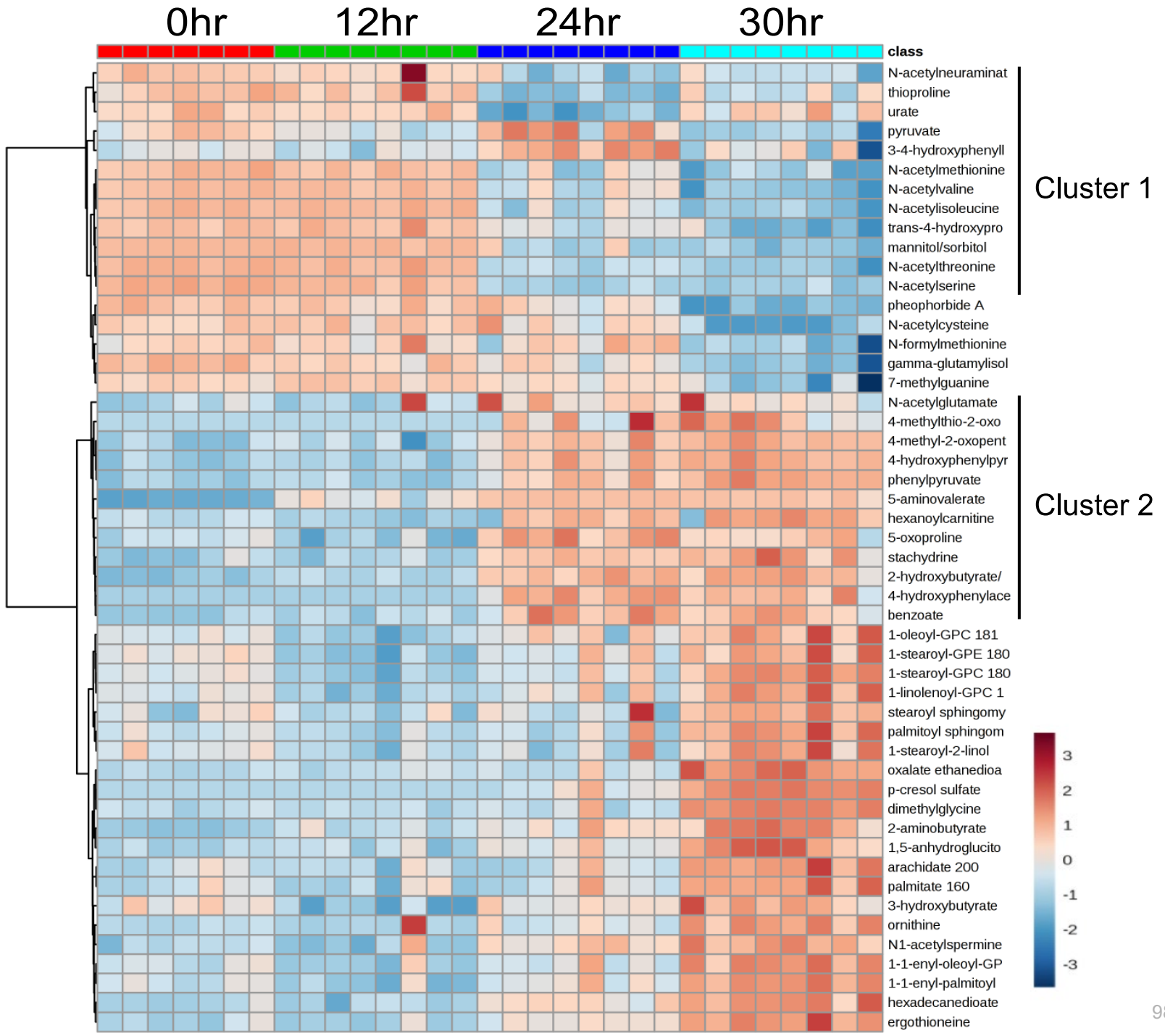
Random Forest

Top 50 metabolites identified by Random Forest analysis. The mean accuracy value decrease is a measure of how much predictive power is lost if a given metabolite is removed or permuted in the Random Forest algorithm; Metabolite names are labeled red if their level increased throughout infection, black if they were variable, and green if the level decreased



Heat map

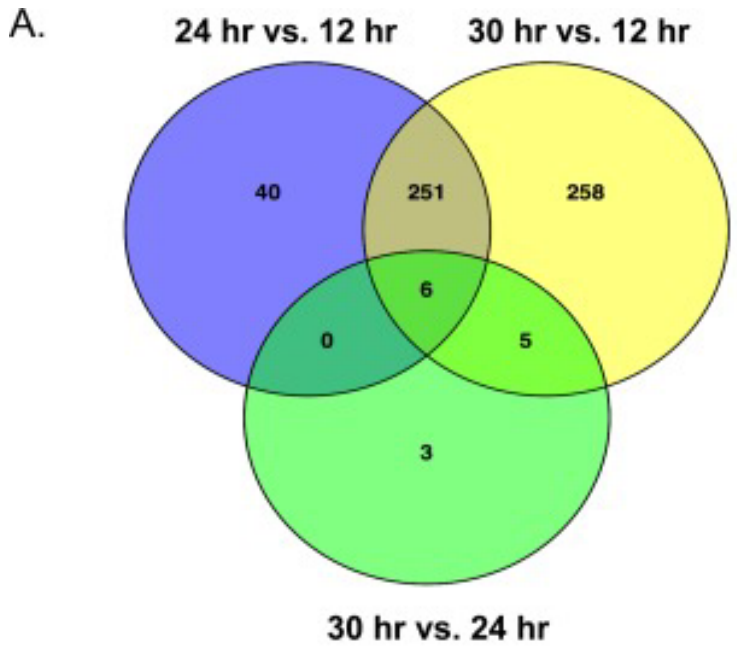
- Heat maps showing relative abundances of top metabolites based on a random forest analysis used to classify samples into time point categories
- Unsupervised hierarchical clustering was used to cluster metabolites with similar abundance profiles over time
- Suggests 5-aminovalerate has highest predictive accuracy and increases throughout infection

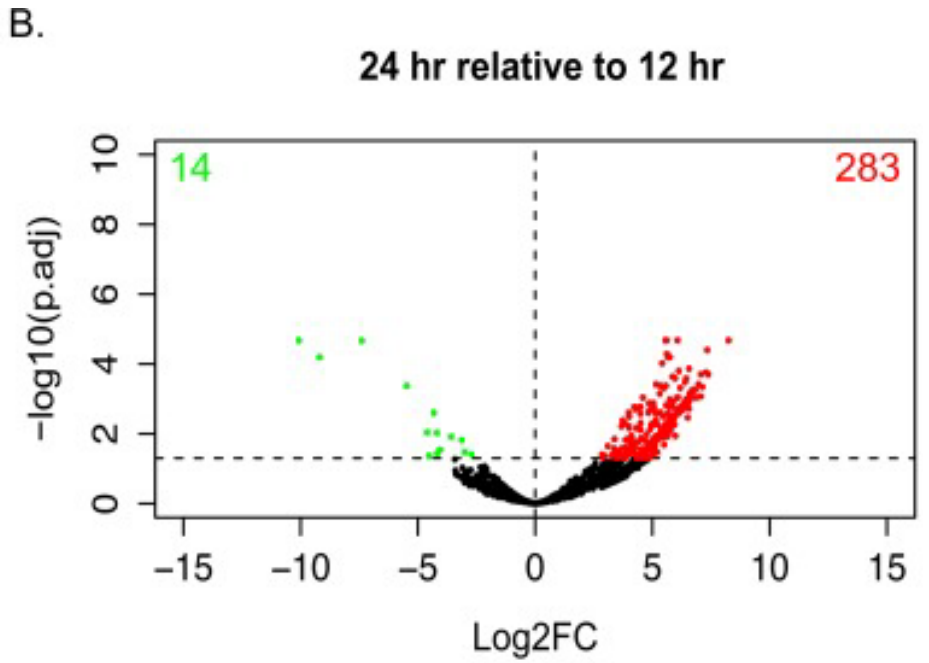


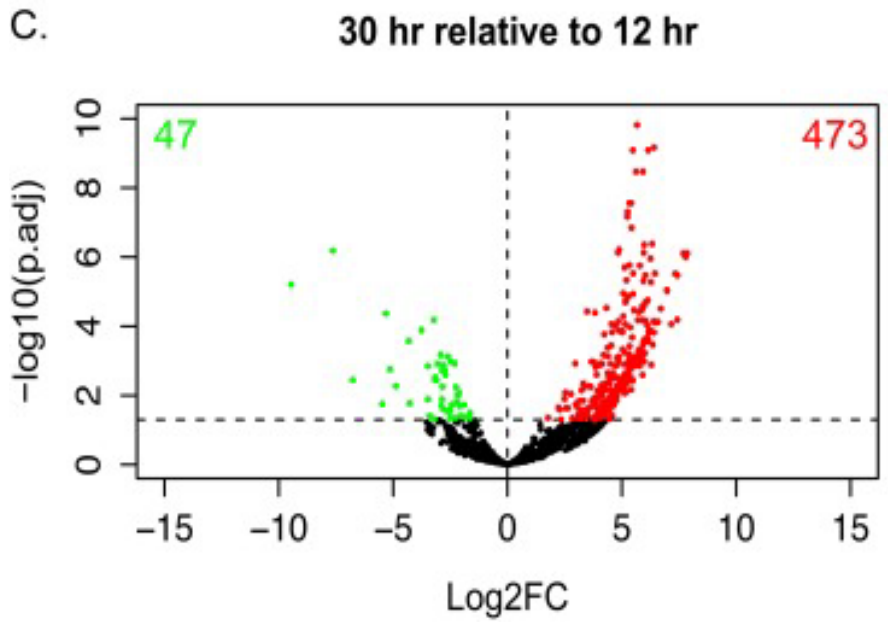
Results

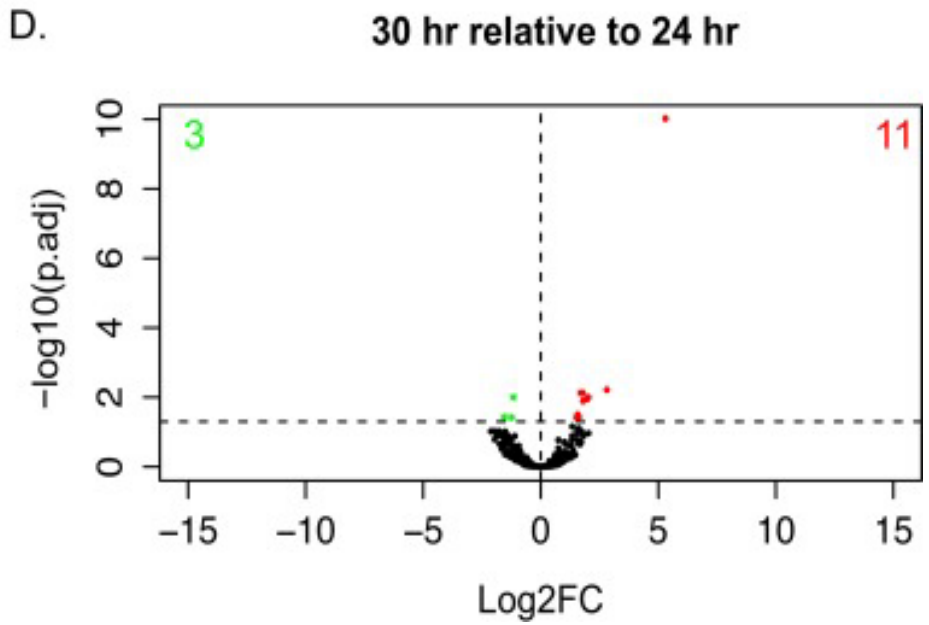
Current data techniques with transcriptomics

- (A) Venn diagram showing the differentially expressed genes that were shared or unique between the three time points.
- (B to D) Volcano plots highlighting genes whose transcript levels changed by greater than 2-fold
  - Red genes had increased transcript levels, green had decreased levels.



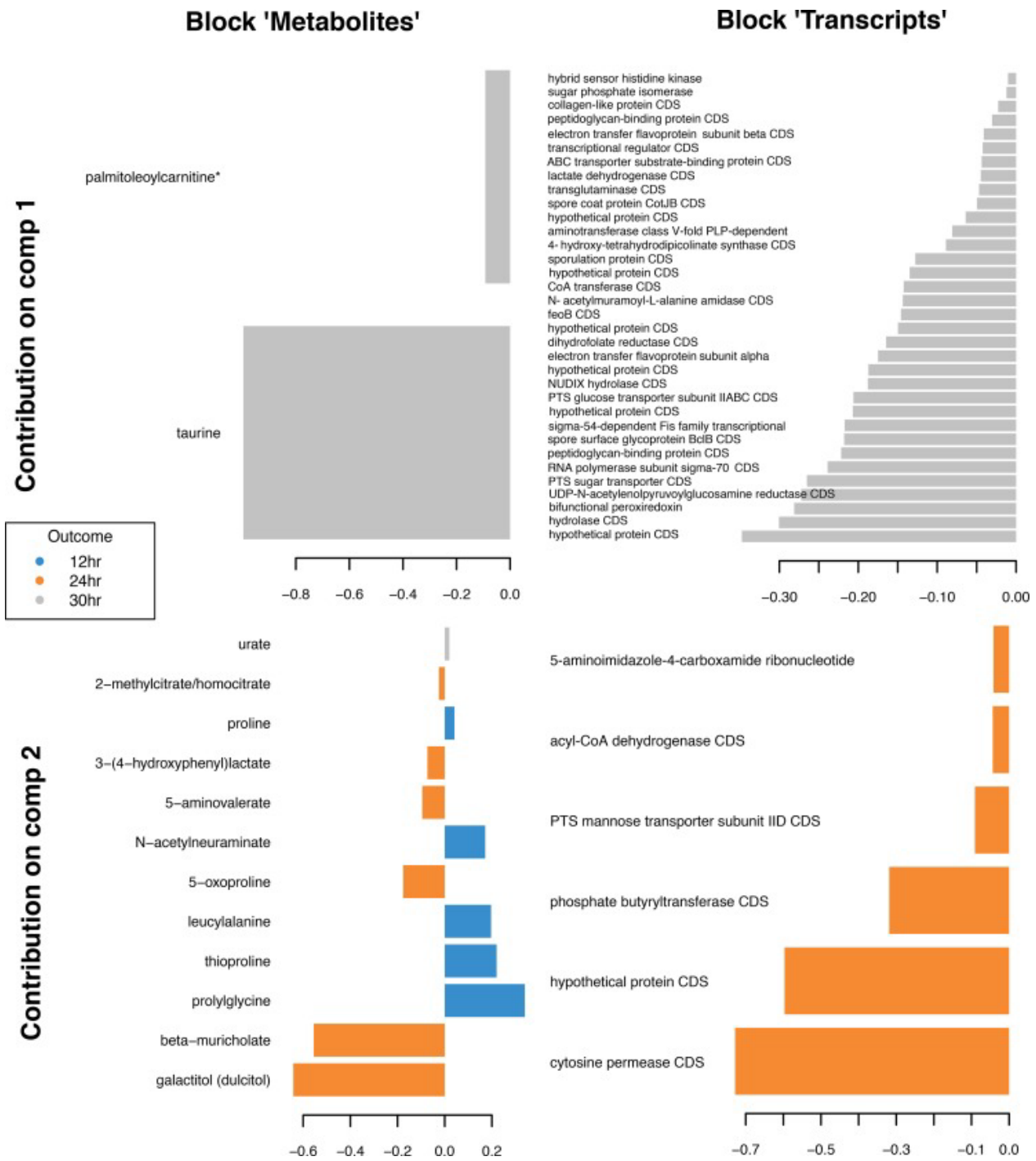




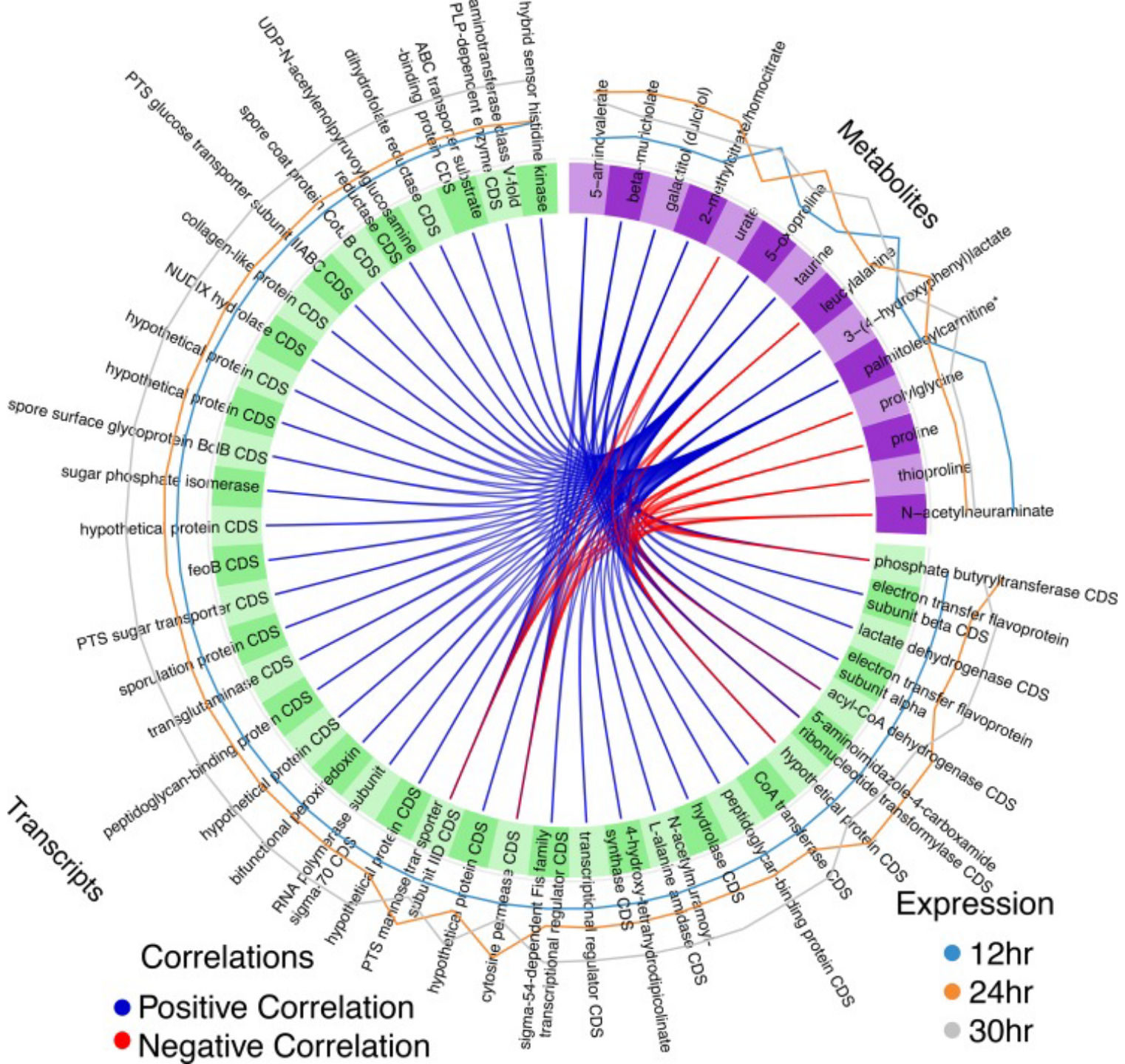


Results

Multivariate-based integration of the gut metabolome and *C. difficile* transcriptome throughout colonization and infection using MixOmics



- Multivariate-based analysis of the gut metabolome and C. difficile transcriptome during colonization and infection.
- The top row indicates the features in the first component for the metabolites (left) and transcripts (right).
- The bottom row indicates the features in the second component for the metabolites (left) and transcripts (right).
- The color indicates the expression levels of each variable according to each class where blue represents 12 h, orange represents 24 h, and gray represents 30 h.



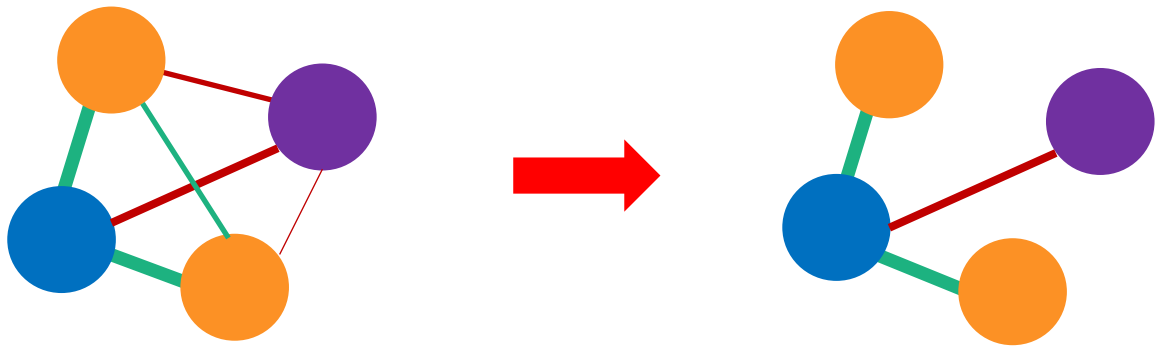
- Correlations between the metabolome and C. difficile transcriptome.
- A Circos plot displays the positive and negative correlations between the selected features with blue and red lines, respectively
- The metabolites are indicated in purple (top right quadrant), and the transcripts are indicated in green.
- The outer lines indicate the expression levels of each variable according to each class where blue represents 12 h, orange represents 24 h, and gray represents 30 h.

Sparsity

- High dimensional data is difficult because there is a low sample size and a lot of measurements
- We cannot analyze all of the data, and some of it is noisy
- Graphical models are useful to understand dependencies within a data set
- Using sparsity allows the use of networks as synthesis tools, particularly for datasets containing a large number of variables such as omics datasets.
- Penalized methods make networks sparse (fewer entries) so that only the most important associations in the population are included

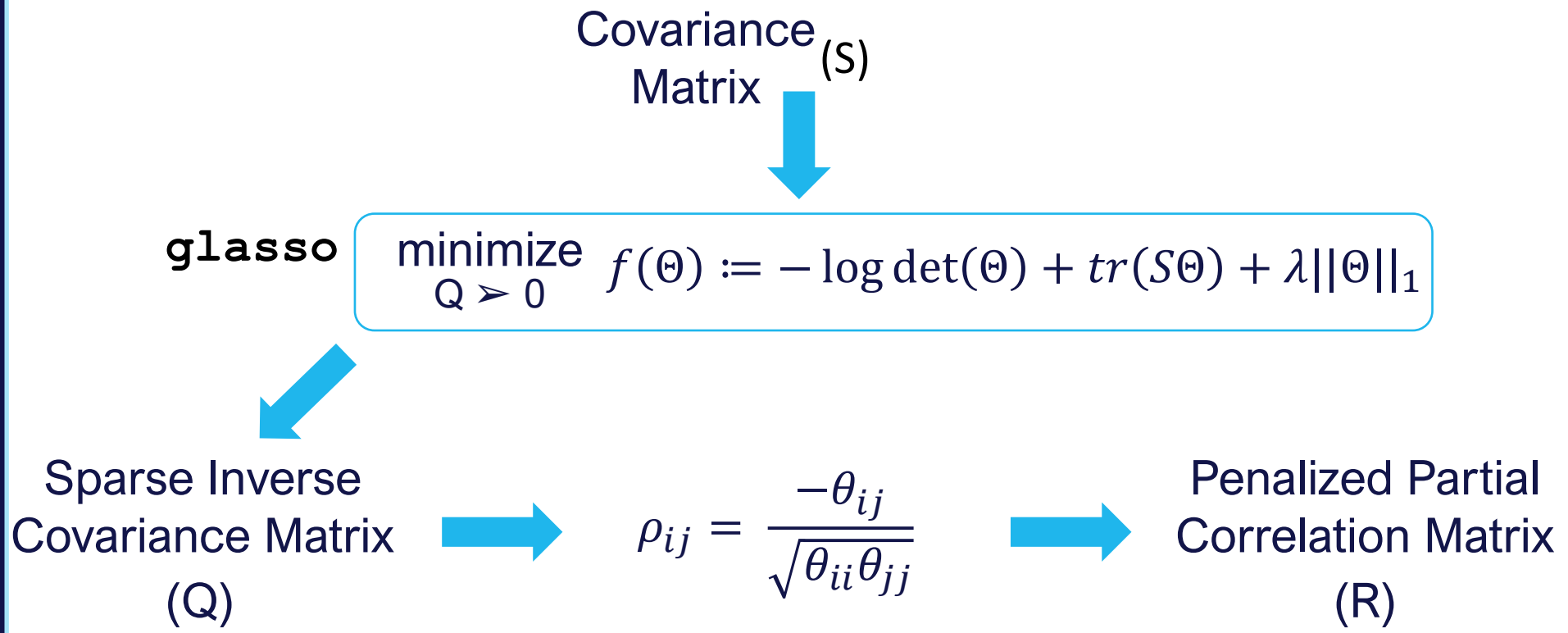
Sparsity

- Graphical models are useful to understand dependencies within a data set
- Using sparsity allows the use of networks as synthesis tools, particularly for data sets containing a large number of variables such as omics data sets.
- Penalized methods make networks sparse (fewer entries) so that only the most important associations in the population are included



Sparse graphical models

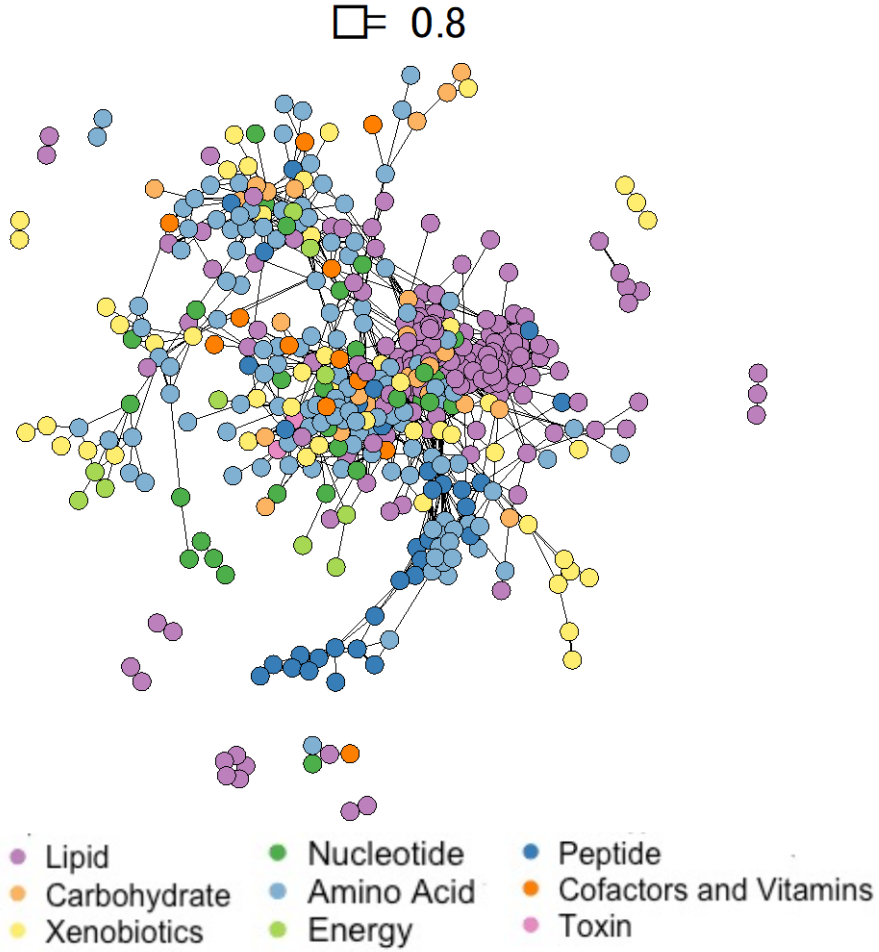
- Network structure was estimated from data using the graphical least absolute selection and shrinkage operator (glasso, Friedman, Hastie & Tibshirani, 2008)

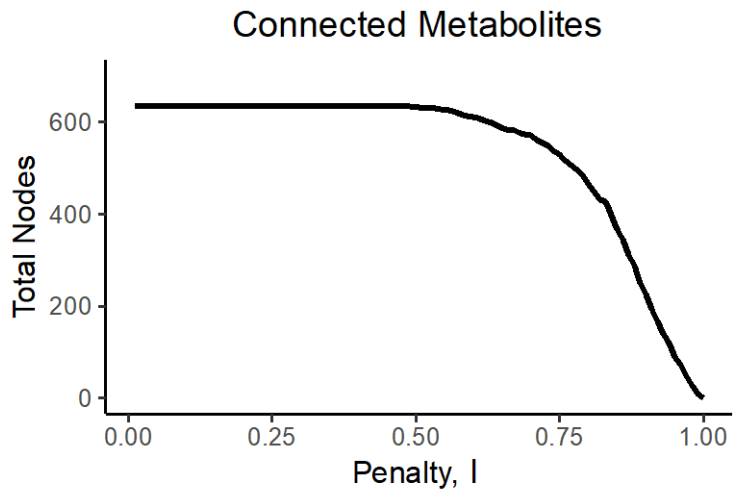


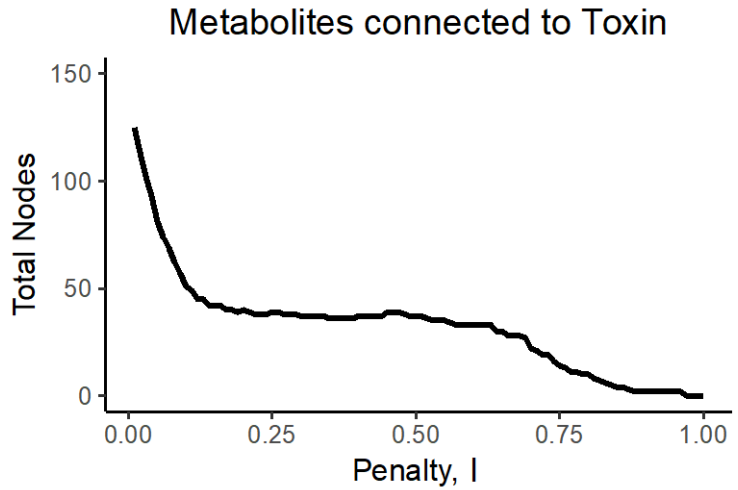
- Partial correlation matrix removes correlations of confounding variables

Discussion

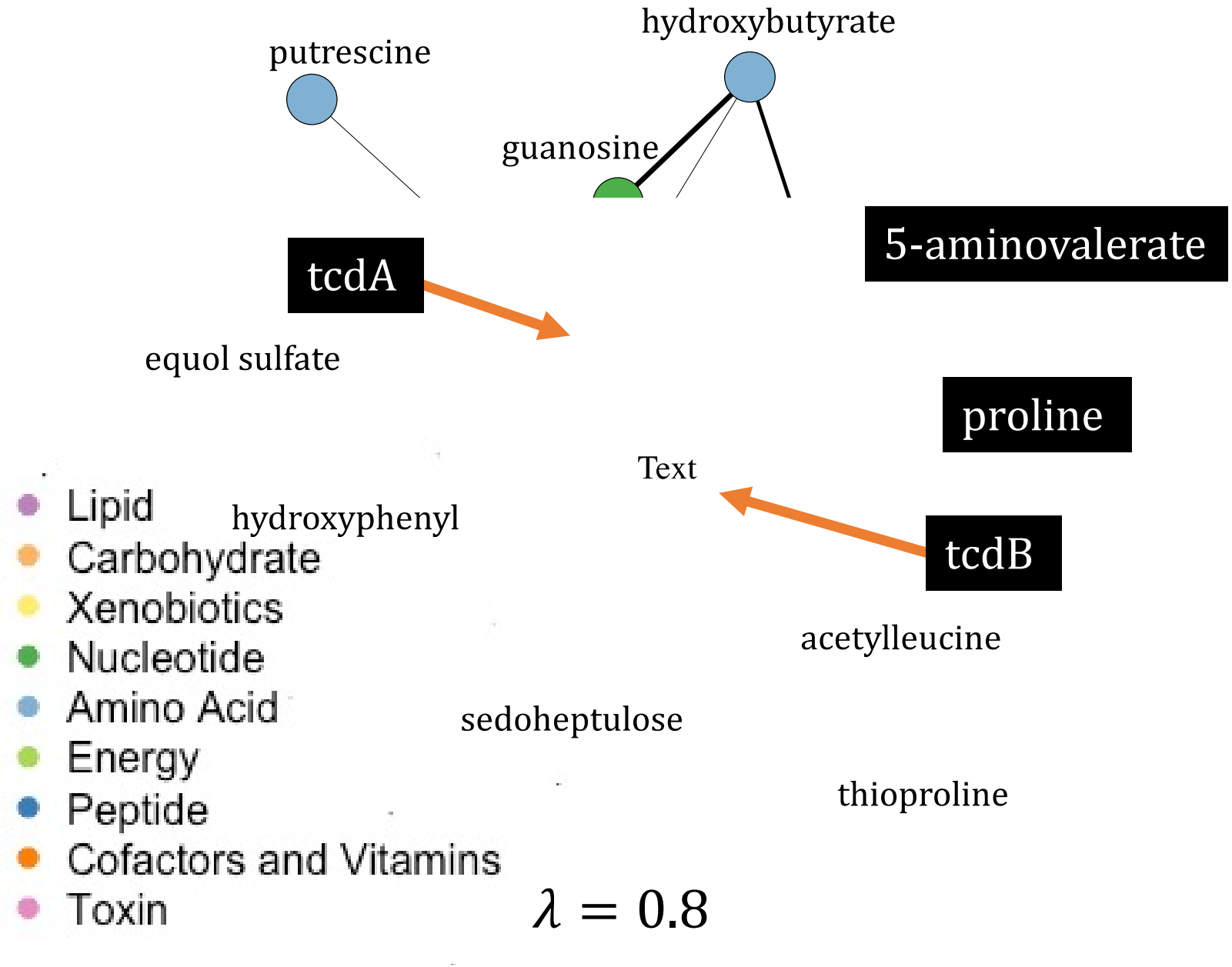
- Left panel shows a sparse graph with $\lambda = 0.8$.
- The right panel shows a graph of the total metabolites in a graph as λ is increased (top) and the metabolites connected to toxin as λ is increased (bottom)





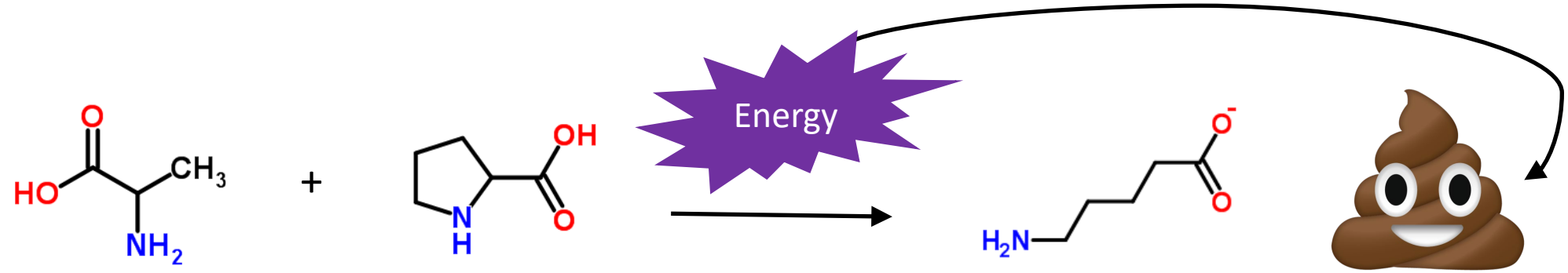


What's connected to toxin?

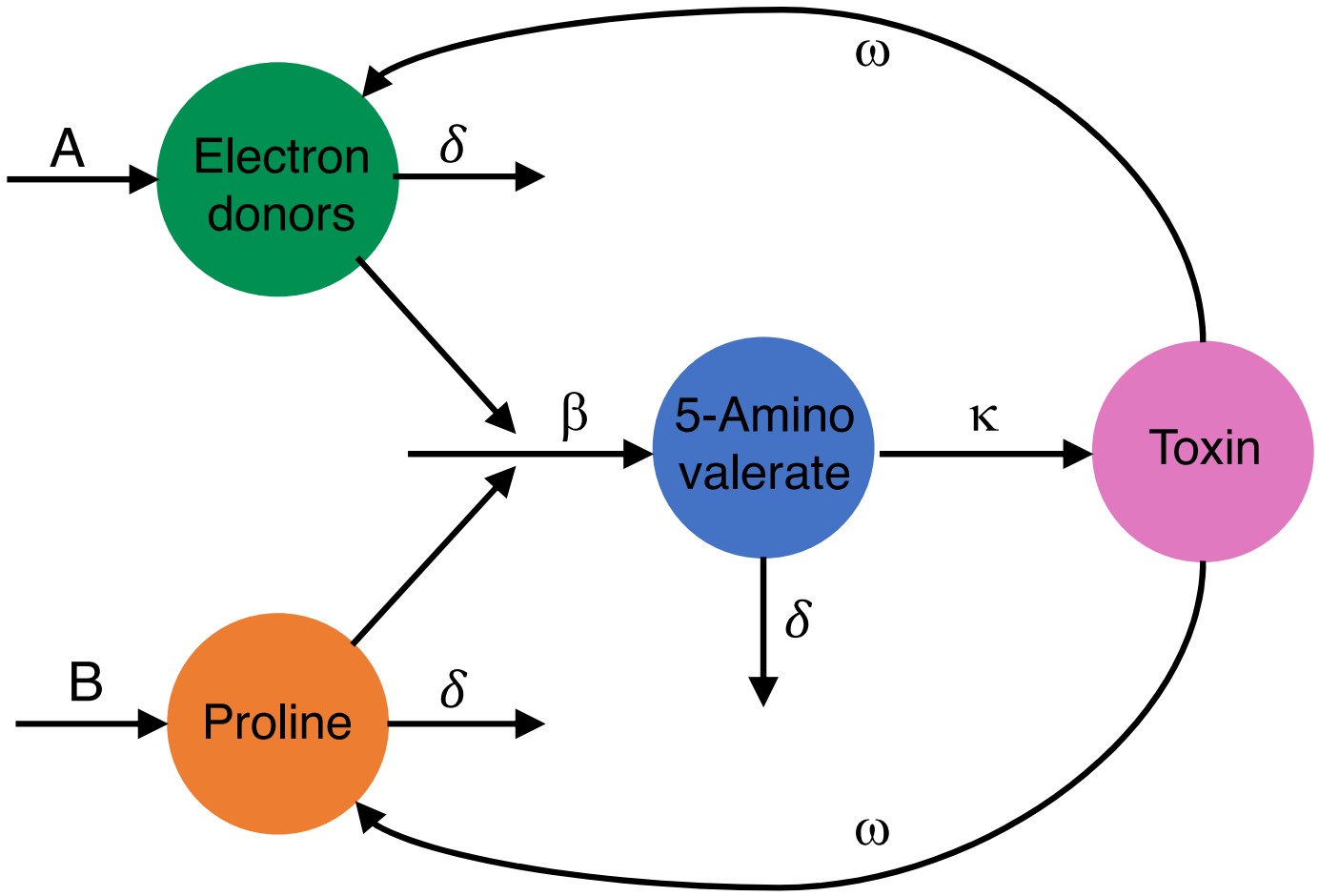


Metabolic reactions lead to energy for *C. difficile*

- The metabolites in our sparse networks are involved in the Stickland reaction
- This reaction is thought to provide energy to *C. difficile* which will grow and produce toxin and subsequent infection
- There are numerous different amino acids that could be contributing electrons to the proline pathway



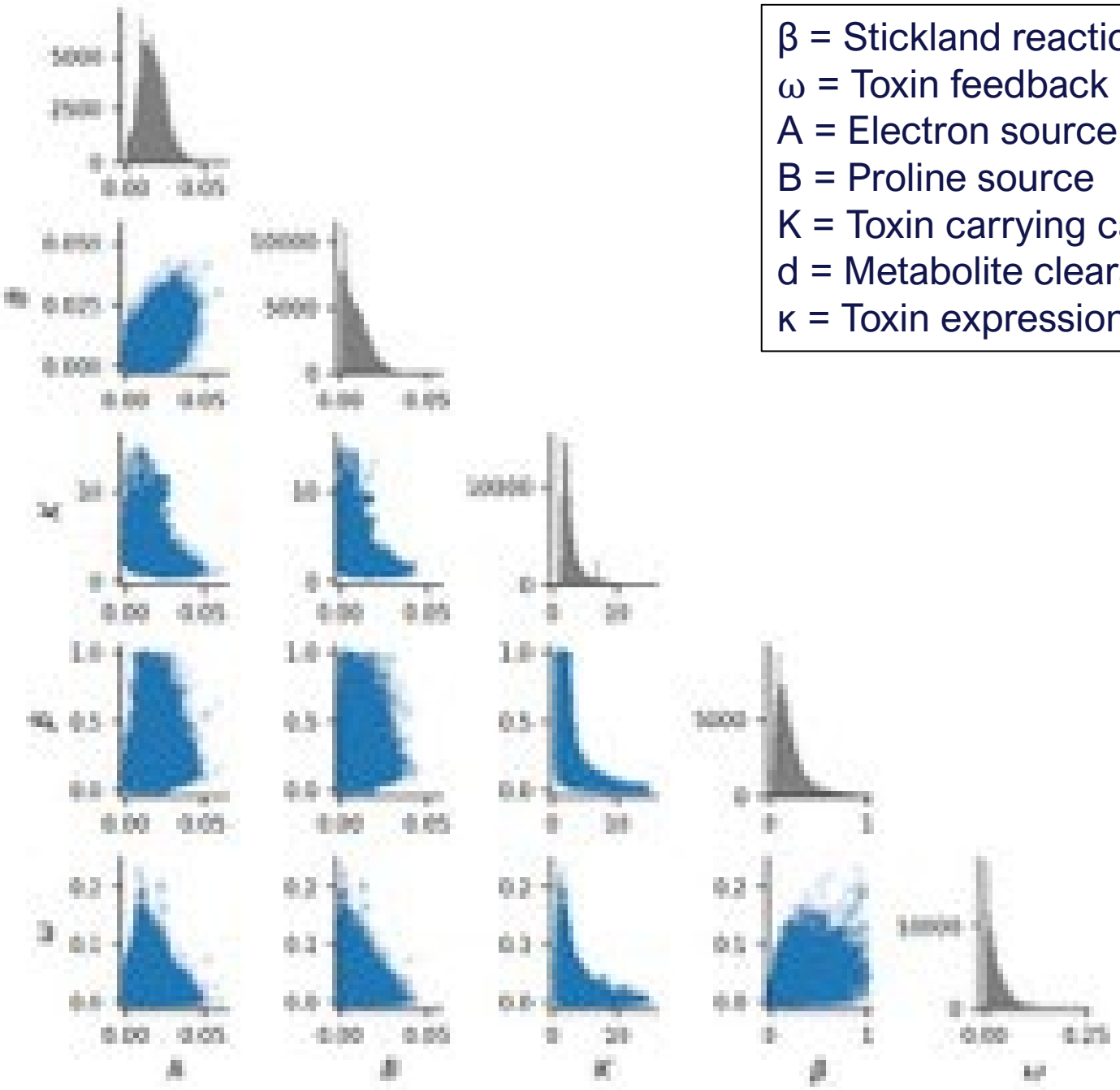
Stickland reaction model



$$\frac{dE}{dt} = A + \omega TE - \beta EP - \delta E,$$
$$\frac{dP}{dt} = B + \omega TP - \beta EP - \delta P,$$
$$\frac{dS}{dt} = \beta EP - \delta S,$$
$$\frac{dT}{dt} = \kappa T\left(1 - \frac{T}{KS}\right) - \omega T(E + P),$$

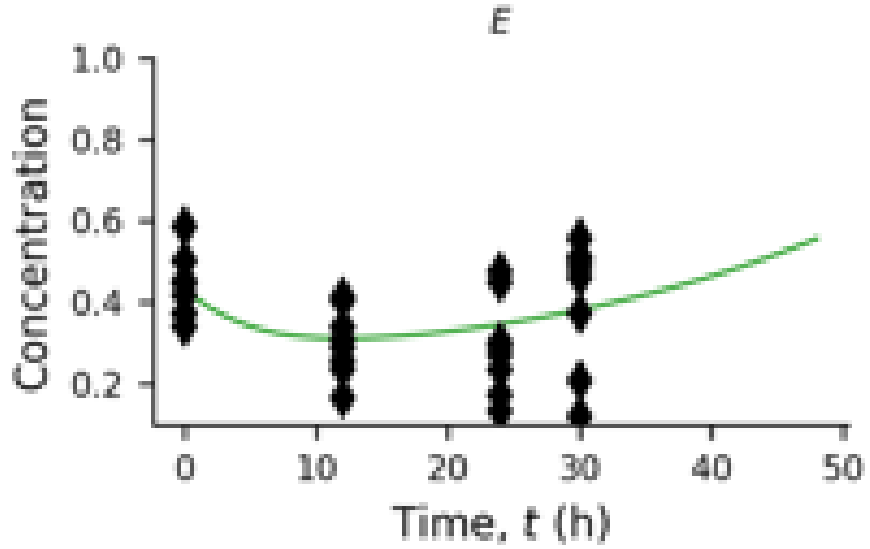
- E = leucine, isoleucine, and valine
- P = proline
- S = 5-aminovalerare

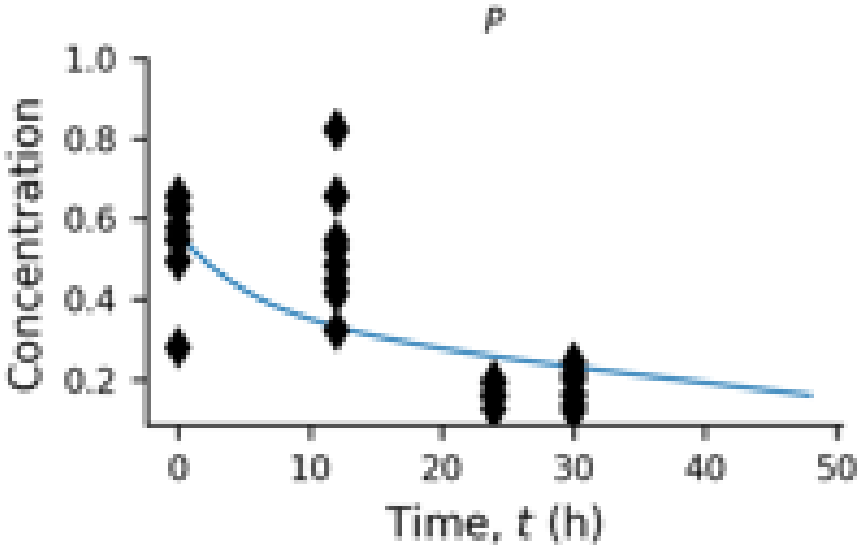
- We fix parameters d and k.
- We use median values from the data for the initial conditions
- We fit b, w, A, B and K

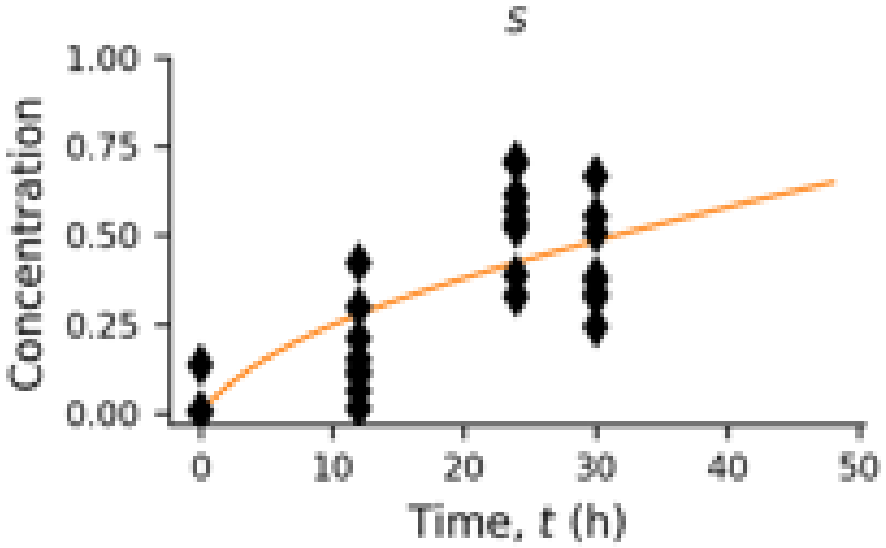


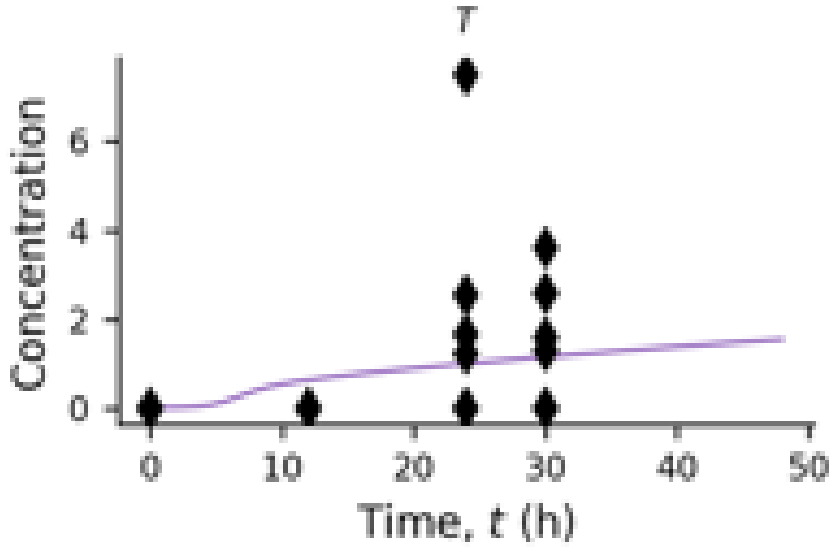
β = Stickland reaction rate
ω = Toxin feedback
A = Electron source
B = Proline source
K = Toxin carrying capacity
d = Metabolite clearance
κ = Toxin expression/growth

Stickland reaction and toxin production

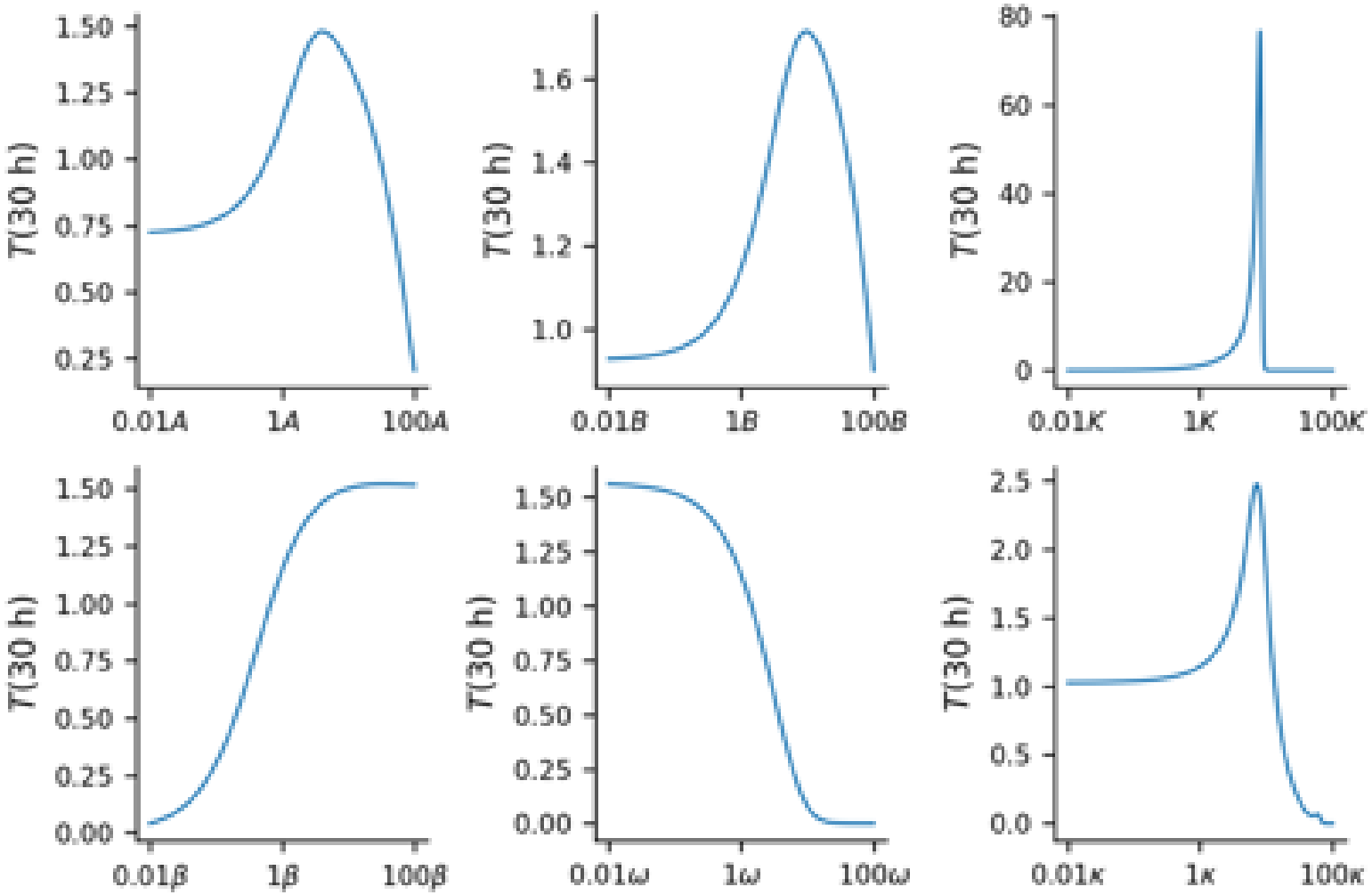








Changes in proline and electron donors decrease toxin production



β = Stickland reaction rate	K = Toxin carrying capacity
ω = Toxin feedback	d = Metabolite clearance
A = Electron source	κ = Toxin expression/growth
B = Proline source	

Conclusion

- Sparse graphical models are a suitable to develop a mechanistic model based on omics data
- The Stickland reaction maybe an influential mechanism driving toxin production
- Our model suggests adjusting the inflow of metabolites used in the Stickland reaction as a primary intervention strategy

Questions?

Afternoon Session

1:30-2:30 p.m.	Basic Machine Learning Workflow	Samantha Erwin
2:30-2:40	Networking Break	
2:40-4:00	‘Omics Integration with SPLS	Daniel Claborne
4:00-4:05	Closing Remarks	Lisa Bramer