DTU

Alexander Gmeiner

Processing, analysing and preparing genomic data for epidemiological analyses and modelling

The exercise(s)

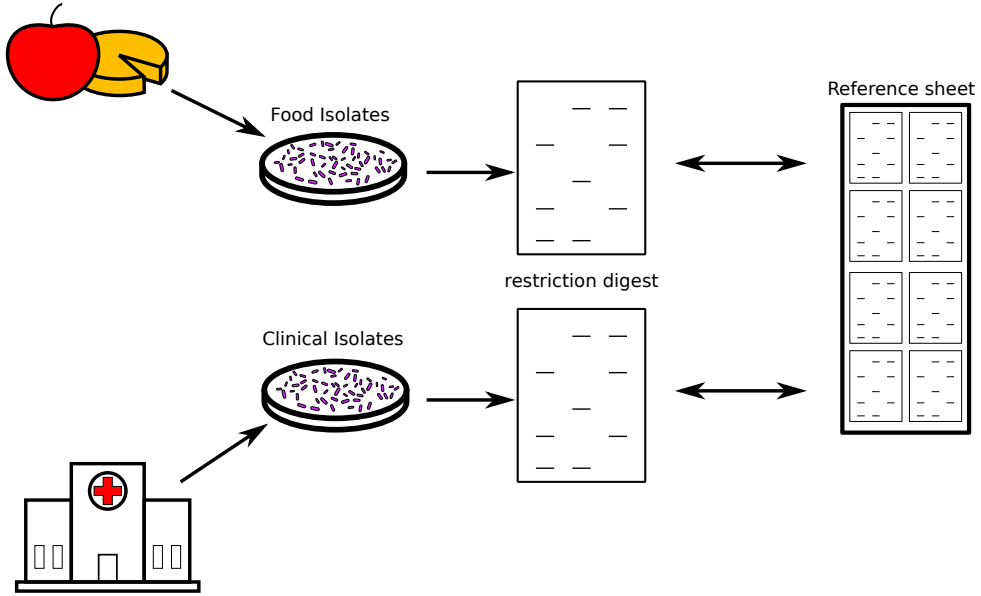
Listeria monocytogenes (LM)

- Listeriosis one of highest case fatality rate (20%-30%)[2]
- *L. monocytogens* is quite resistant concerning environmental surrounding and grows well at cool storage temperatures[3]
- Strict governmental restrictions
  – Denmark (all Listeria samples found in the food industry and in clinic have to be WGS)

1. Listeriosis. https://www.who.int/news-room/fact-sheets/detail/listeriosis.
2. Liu, D., Lawrence, M. L., Ainsworth, A. J. & Austin, F. W. Comparative assessment of acid, alkali and salt tolerance in Listeria monocytogenes virulent and avirulent strains. FEMS Microbiology Letters **243**, 373–378 (2005).

What we have…

- 57 WGS assembled samples
  - From Denmark
  - Food industry and clinics



What we are looking for…

- We are trying to predict virulence (i.e. harmfulness)

$$clinical\ frequency = \frac{\#clinical\ isolates}{\#clinical\ isolates + \#food\ isolates}$$

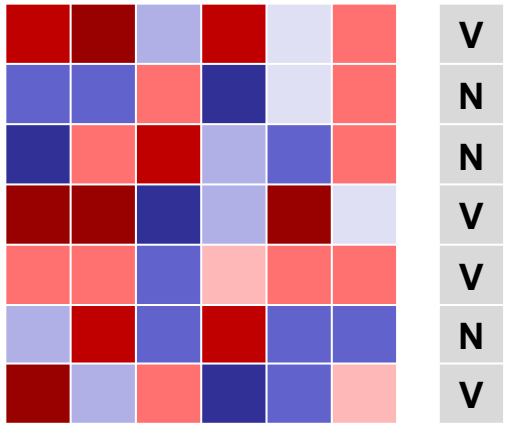
low

high

Workflow

Features
%-identity Vir. Genes

Outcome
Clin. Freq



$$\textit{clinical frequency} = \frac{\#\ \textit{clinical isolates}}{\#\ \textit{clinical isolates} + \#\ \textit{food isolates}}$$

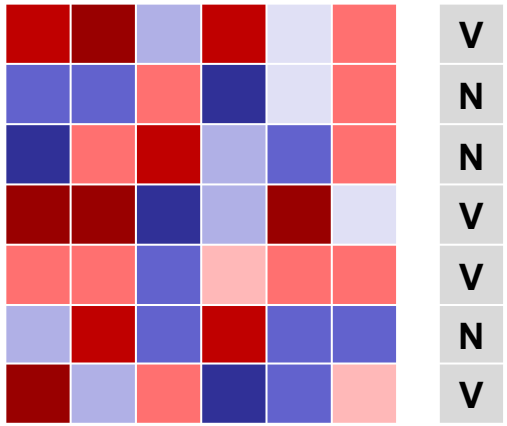
Workflow

Absence/presence

- Virulence genes
- Pangenome genes
- Kmers
- Snps

Features
%-identity Vir. Genes

Outcome
Clin. Freq



V
N
N
V
V
N
V

$$\text{clinical frequency} = \frac{\#\ \text{clinical isolates}}{\#\ \text{clinical isolates} + \#\ \text{food isolates}}$$

Workflow

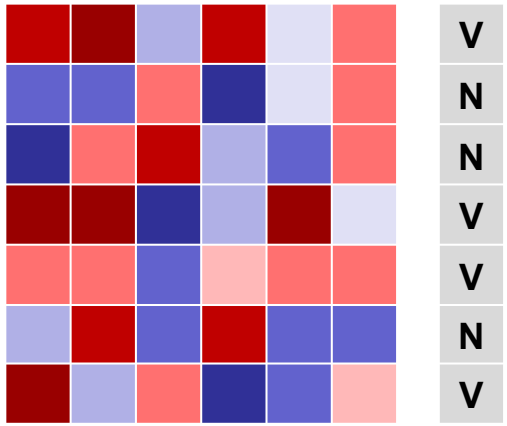
Absence/presence

- Virulence genes
- Pangenome genes
- Kmers
- Snps

Database for known virulence genes of ***LM***

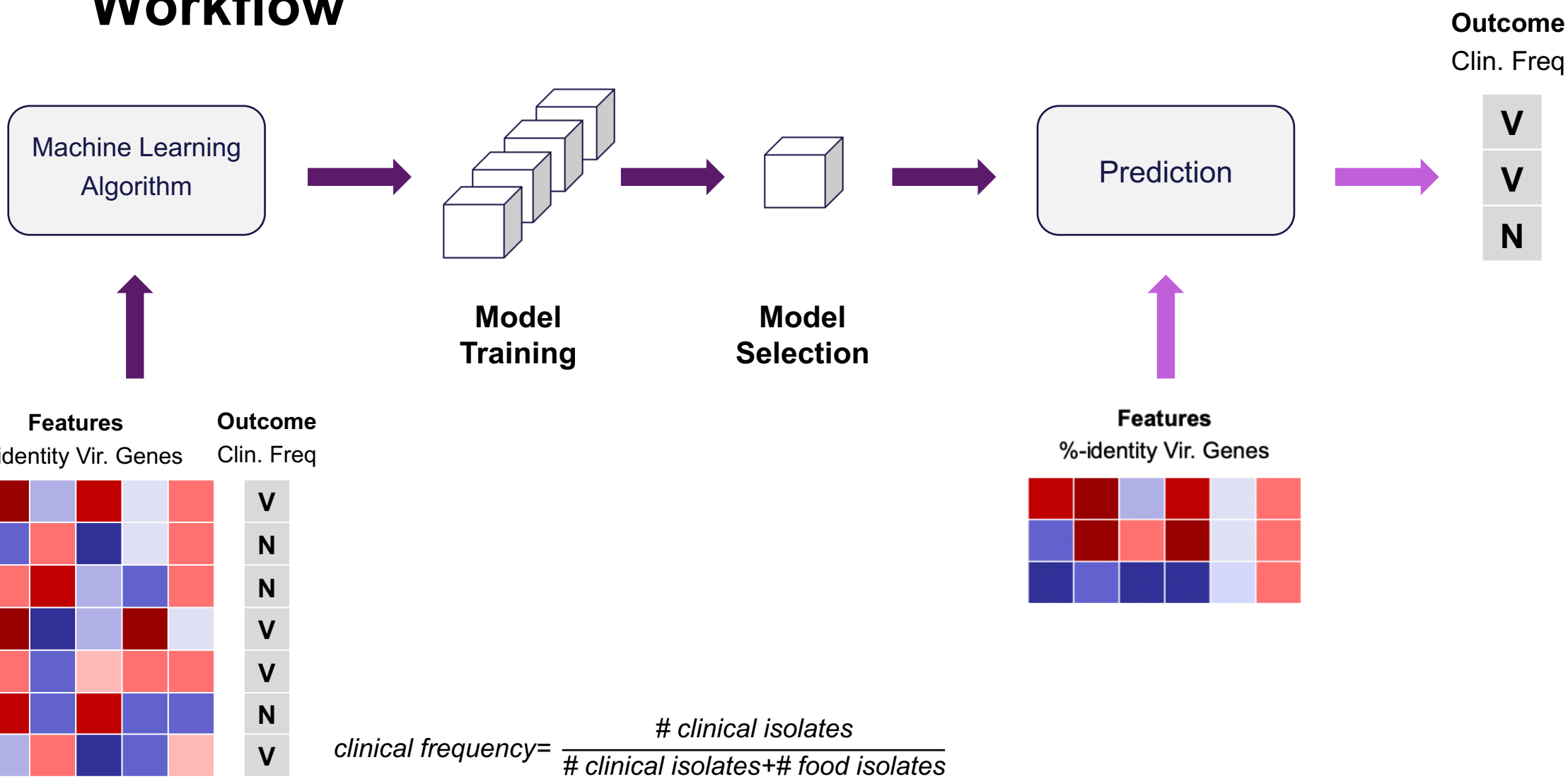
Features
%-identity Vir. Genes

Outcome
Clin. Freq



$$\textit{clinical frequency} = \frac{\#\ \textit{clinical isolates}}{\#\ \textit{clinical isolates} + \#\ \textit{food isolates}}$$

Workflow



$$\text{clinical frequency} = \frac{\#\ \text{clinical isolates}}{\#\ \text{clinical isolates} + \#\ \text{food isolates}}$$

Exercise setup

1. **Prepare the WGS data (C2)**
   - Blast database against individual samples
   - Parse the blast database output

2. **How to build our ML model (theoretical)**
   - Train a Random Forrest ML model

3. **Predict newly found strains (C2)**
   - Use pre-trained model to predict virulence of newly found strains

Before we start

There are two main directories:

- `data`
- `scripts`

Each has 3 sub-directories (for the different parts):

- `prep_input`
- `ML_model_training`
- `pred_virulence`

Before we start

The ”.q” files are THE ONES TO USE with the ”`qsub`” command. They contain all the required headers for the queuing system.

The exercises some python scripts that I coded myself.

They are in the `scripts/sub_directory/required_scripts/` and have a ”.py” ending.

Before we start

Copy the scripts directory to your personal directory

```
cp -r /home/projects/course_23262/course/week08/scripts/
/home/projects/course_23262/people/youruser/.
```

1. Prepare Input

Goals of Exercise 1:

- Preparing an input matrix for the ML model

Columns: Virulence Genes

Rows: LM samples

Genome	Vir_gene_1	Vir_gene_2	Vir_gene_X
LM_sample_1	95,78	100	
LM_sample_2	91,71	100	
LM_sample_3	91,71	100	
LM_sample_4	95,78	100	
LM_sample_5	86,52	100	
LM_sample_6	98,75	100	

Percent Identities from blast mapping

Mapping

- Aim:

  Look for absence/presence of virulence genes from the database in the samples

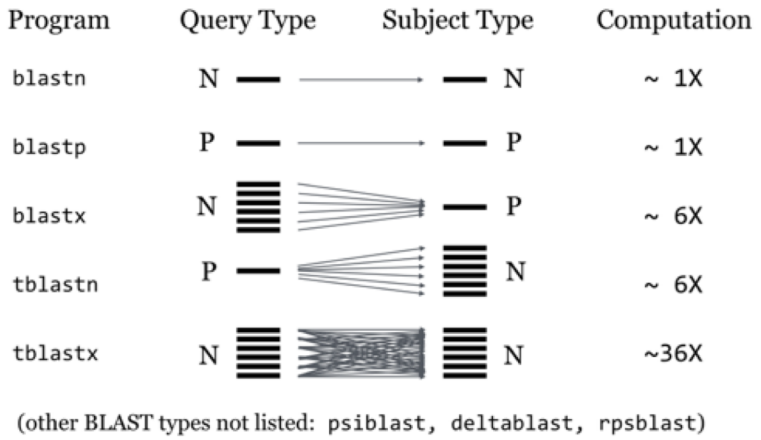
- Script:
  - `run_jobarray_blast.q` (/scripts/prep_input/)

- Data needed:
  - Sample assemblies (/course/week08/data/prep_input/input_assblies/)
  - Virulence genes database (/course/week08/data/prep_input/database/List_virulence_db.txt)

Let’s look at the script

- Jobarrays
  - Nice to use when you have to run the same script for multiple samples.
  - Specify with “-t” flag
  - Always limit number of simultaneously running jobs “%” after your list of jobarray numbers

Mapping



- Task:

  In **run_jobarray_blast.q** replace the ”XXX”

```
XXX -query ${db} -subject ${sample_f} -outfmt 6 -out ${out_f} -evalue
0.001 -max_hsps 1
```

1. Replace the ”XXX” in the ”outdir” variable with your user name

2. Decide which blast+ version to use

   (Tip: Find out if the db/sample sequences are protein or nucleotide sequences.
   Then have a look on the website or use the picture in the top right.
   https://open.oregonstate.education/computationalbiology/chapter/command-line-blast/)

Mapping

- Task:

  Submit **run_jobarray_blast.q** to the queue

```
qsub ./run_jobarray_blast.q
```

Have a look at the output files **List_*_blast_out.txt** in the ”blast_out/” directory

Parsing

- Aim:

  Parse the blast mapping output file and convert into tabular format

- Script:
  - run_parallel_parse_blastout.q `(/scripts/prep_input/)`

- Data needed:
  - Blast output files `(/blast_out/List_*_blast_out.txt)`
  - Virulence genes database `(/course/week08/data/prep_input/database/List_virulence_db.txt)`

Parsing

- Task:

  In **run_parallel_parse_blastout.q** replace the ”XXX”

```
# set the directory
indir=”XXX"
```

(Tip: The indir variable should lead to the directory where you stored the output files from the mapping with blast. DON’T FORGET A “/” AT THE END OF THE ABSOLUTE PATH)

Parsing

- Task:

  Submit **run_parallel_parse_blastout.q** to the queue

```
qsub ./ run_parallel_parse_blastout.q
```

Have a look at the output file **parse_blast_out.csv** in the ”blast_out/” directory

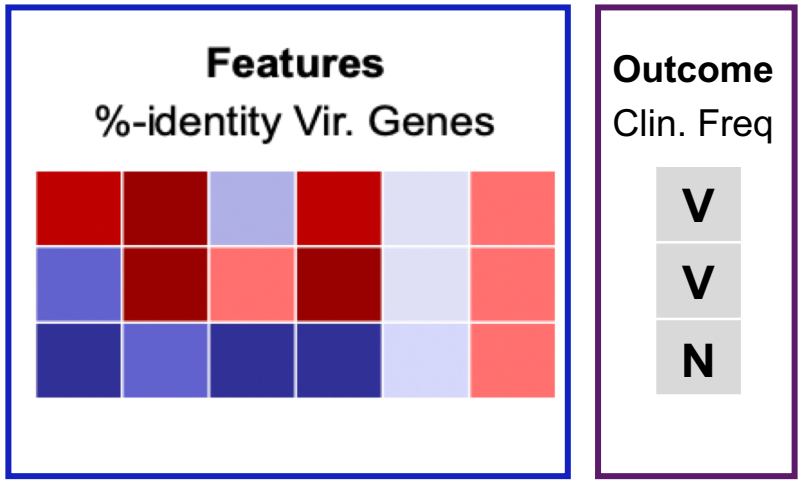
2. ML model building

Encoding & Grouping

Encoding

Make input more suitable for machine learning

e.g., normalize all features to be between a specific range; in general 0 and 1

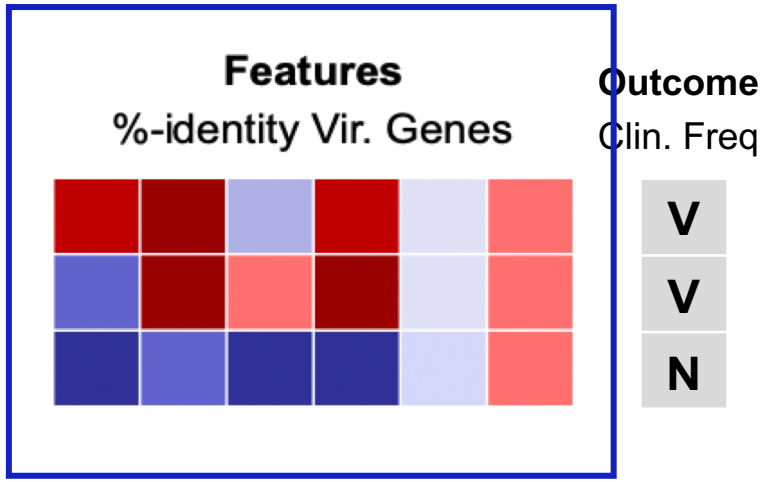


Grouping

To move from a Regression problem to a Classification problem

Encoding

- Use absolute percent identities
  e.g., 90% → 0.90



Genome	ActA	Agr
0	95,78	100,00
1	95,78	100,00
2	86,52	100,00

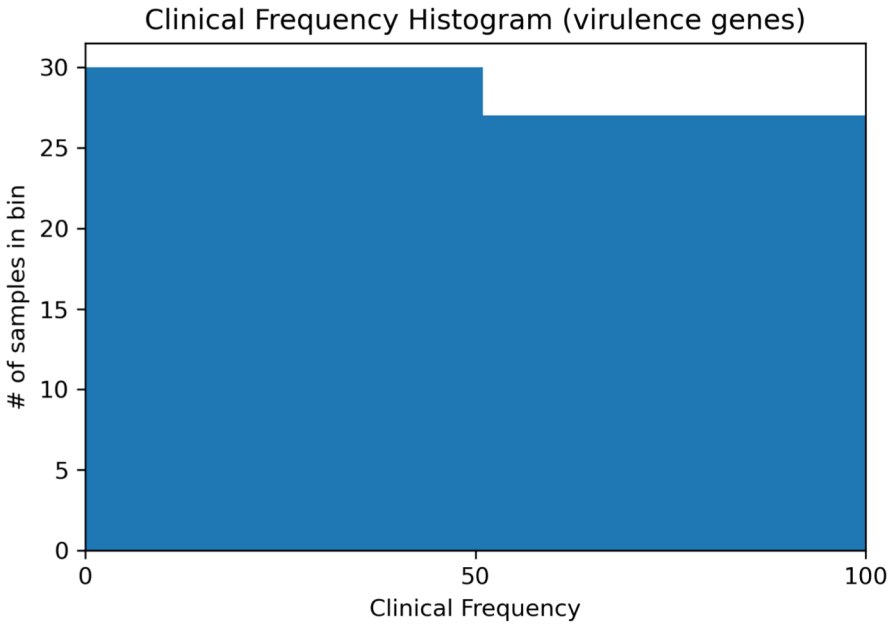
Genome	ActA	Agr
0	0,9578	1
1	0,9578	1
2	0,8652	1

Pros: No arbitrary bin boundaries (original values); Values are between 0 and 1

Cons: We might have noisy input

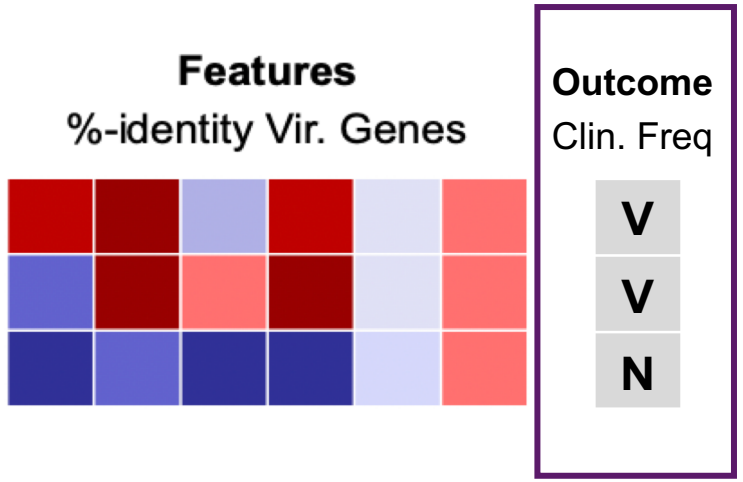
Grouping

- Bin Clinical frequency in categories – e.g., 3 classes



<= 50 > 50

Non-Virulent Virulent



$$\text{clinical frequency} = \frac{\#\ \text{clinical isolates}}{\#\ \text{clinical isolates} + \#\ \text{food isolates}}$$

ML training workflow

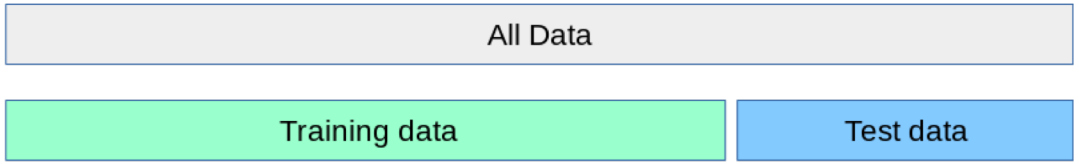
1. Preprocess input matrix

All Data

Adapted from: https://scikit-learn.org/stable/_images/grid_search_cross_validation.png; 2021/10/03,11:35

ML training workflow

1. Preprocess input matrix
2. Train-/test data (0.80/0.20)



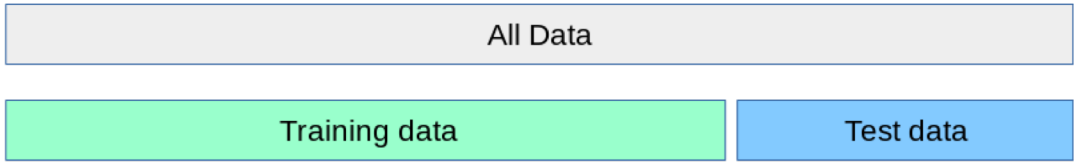
Adapted from: https://scikit-learn.org/stable/_images/grid_search_cross_validation.png; 2021/10/03,11:35

ML training workflow

1. Preprocess input matrix

2. Train-/test data (0.80/0.20)
   For each estimator/algorithms:

   1) Hyper parameters search



Adapted from: https://scikit-learn.org/stable/_images/grid_search_cross_validation.png; 2021/10/03,11:35

Model training (important concepts)

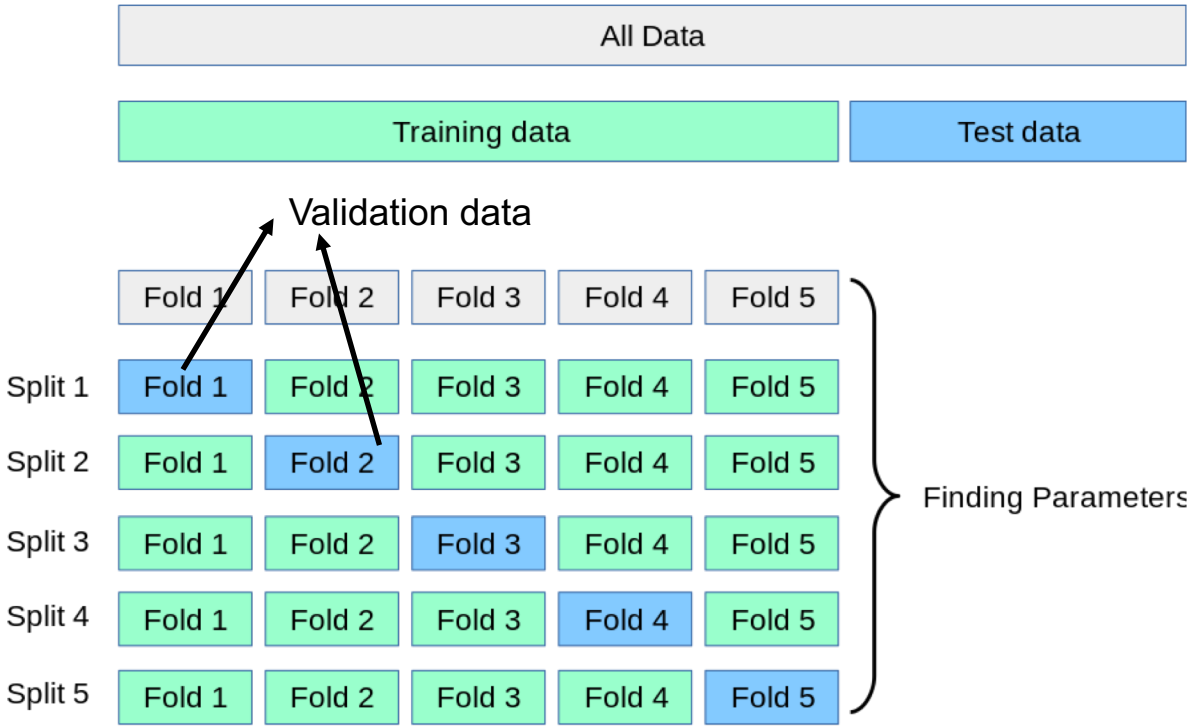
- Hyperparameters
  - parameters that can not be derived from data itself
    e.g., depth of RF
  - these are usually tuned by searching over a grid of multiple parameters values to find the best combination that maximizes the performance

ML training workflow

1. Preprocess input matrix

2. Train-/test data (0.80/0.20)
   For each estimator/algorithms:

   1) Hyper parameters search
      – Filter features with low amount of information
      – stratified 5-Fold cross-validation



Adapted from: https://scikit-learn.org/stable/_images/grid_search_cross_validation.png; 2021/10/03,11:35

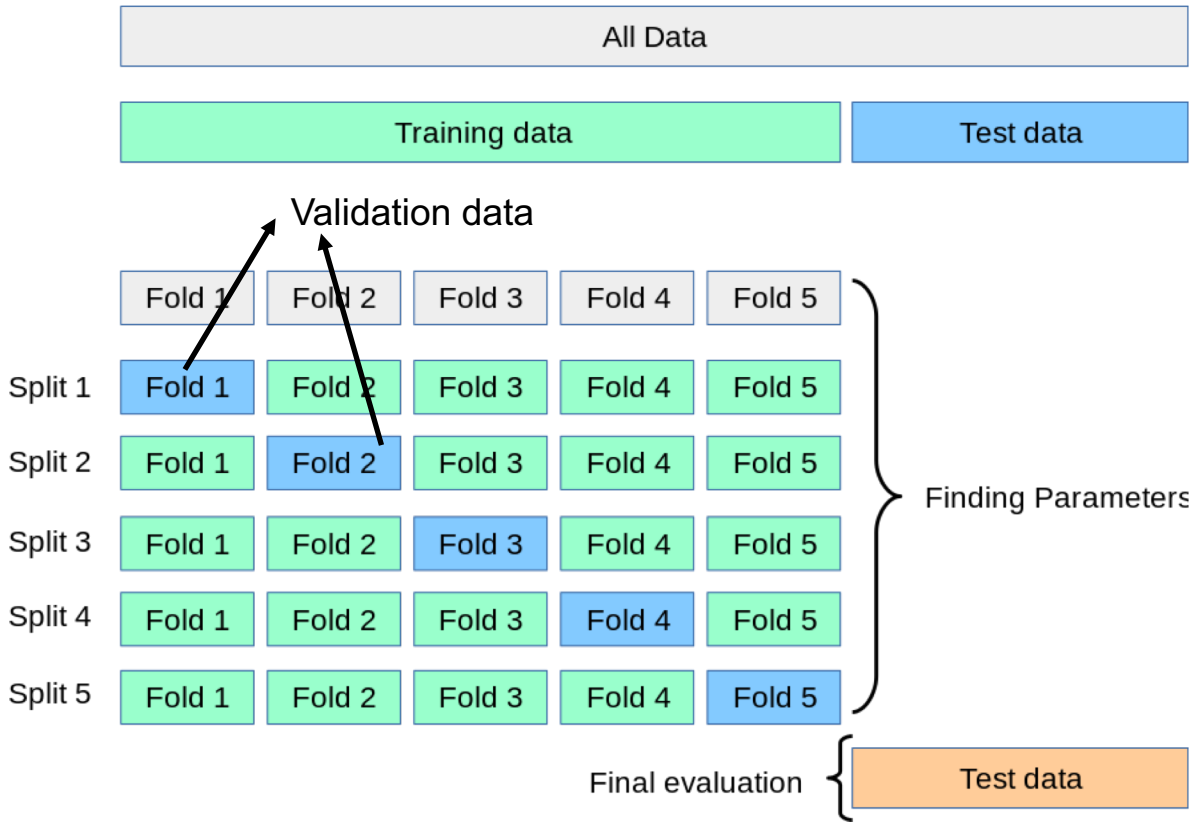
ML training workflow

1. Preprocess input matrix

2. Train-/test data (0.80/0.20)
   For each estimator/algorithms:

   1) Hyper parameters search
      – Filter features with low amount of information
      – stratified 5-Fold cross-validation

   2) Predict on the test data



Adapted from: https://scikit-learn.org/stable/_images/grid_search_cross_validation.png; 2021/10/03,11:35

ML training workflow

1. Preprocess input matrix

2. Train-/test data (0.80/0.20)
   For each estimator/algorithms:

   1) Hyper parameters search
      – Filter features with low amount of information
      – stratified 5-Fold cross-validation

   2) Predict on the test data

   x 30

3. Perform statistical analysis on predictions (Bootstrapping)

How to evaluate our Model

- 2 classes
  - Virulent
  - Non-virulent

How to evaluate our Model

- 2 classes
  - Virulent
  - Non-virulent

ACTUAL

PREDICT

	1	0
1	TP	FP
0	FN	TN

How to evaluate our Model

- 2 classes
  - Virulent → Positive class (1)
  - Non-virulent → Negative class (0)

ACTUAL

PREDICT

	1	0
1	TP	FP
0	FN	TN

How to evaluate our Model

- 2 classes
  - Virulent → Positive class (1)
  - Non-virulent → Negative class (0)

ACTUAL

PREDICT

	1	0
1	TP	FP
0	FN	TN

True Positive (TP): predicted virulent, actually virulent
True Negatives (TN): predicted non-virl., actually non-virl.

False Positive (FP): predicted virulent, actually non-virl.
False Negative (FN): predicted non-virl., actually virulent

Performance measures

PREDICT \ ACTUAL	1	0
1	TP	FP
0	FN	TN

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Number of correct descriptions/All predictions

$$Recall/Sensitivity = \frac{TP}{TP + FN}$$

Which proportion of positive class got correctly classified

$$Precision = \frac{TP}{TP + FP}$$

Proportion of predicted positives that are actually true positives

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Harmonic mean of Precision and Recall

CancerDiscover: A configurable pipeline for cancer prediction and biomarker identification using machine learning framework
Akram Mohammed, Greyson Biegert, Jiri Adamec, Tomáš Helikar; bioRxiv 182998; doi: https://doi.org/10.1101/182998

Performance measures

PREDICT \ ACTUAL	1	0
1	TP	FP
0	FN	TN

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Number of correct descriptions/All predictions

$$Recall/Sensitivity = \frac{TP}{TP + FN}$$

Which proportion of positive class got correctly classified

$$Specificity = \frac{TN}{TN + FP}$$

Which proportion of negative class got correctly classified

$$Precision = \frac{TP}{TP + FP}$$

Proportion of predicted 1's that are actually true 1's

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Harmonic mean of Precision and Recall

$$Area\ Under\ the\ Curve\ (AUC)$$

measures ability of a classifier to distinguish between classes

CancerDiscover: A configurable pipeline for cancer prediction and biomarker identification using machine learning framework
Akram Mohammed, Greyson Biegert, Jiri Adamec, Tomáš Helikar; bioRxiv 182998; doi: https://doi.org/10.1101/182998

Exercise

- Encoding
  - Absolute percent identities
- Grouping (Clinical Frequency)
  - 2 Groups (virulent; non-virulent)
- ML model
  - RF
  - 2-layer cross-validation
    - Train-test split (0.80/0.20)
    - 5-Fold cross-validation
- Performance measure
  - F - score

3. Virulence prediction

Let’s test our model

- Aim:

  To see how well our model performs on predicting the virulence of new found samples

- Script:
  - pred_LM_virulence.py `(/scripts/pred_virulence/)`

- Data needed:
  - New sample assemblies `(/data/pred_virulence/new_samples/ new_LM_sample_*.fna)`
  - Pre-trained RF model `(/data/pred_virulence/trained_RF_model.pickle)`

Virulence prediction

- Task:

  Figure out what input **pred_LM_virulence.py** needs

```
python3 pred_LM_virulence.py -h
```

Virulence prediction

- Task:

  Run **pred_LM_virulence.py**

```
python3 pred_LM_virulence.py –ind XXX –outd XXX
```

(NOTE!: pred_LM_virulence.py is a python script that submits to the queue of C2 automatically. You don’t need to qsub the script. Run as shown above and replace the ‘XXX’)

Look at output file **virulence_prediction_virgenes_out.csv**

Looking at the output

- What are the predicted classes?
- Do you think that is right?

Looking at the output

- What are the predicted classes?

- Do you think that is right?

- What could we do to have a better performing model?
  - Use more Data
  - Use different ML model
  - Include data from more countries
  - Explore different levels (pangenome, kmers, snps)

Wrap up

Take home

- Cross-Validation always!
- Independent test sets is great to further validate the model
- Your model is only as good as your data
- ML doesn’t have to be complicated
  BUT the devil is in the detail