Machine Learning and Artificial Intelligence in MX

Melanie Vollmar

Diamond Light Source Ltd

UK

Outline

- What is machine learning?
- What is needed?
- Why machine learning and artificial intelligence?
- The trouble with data
- Training data and METRIX database for exploratory work
- Experimental phasing success
- Molecular replacement success
- Map traceability
- Future plans

What is machine learning?

- Part of the field of artificial intelligence
- Statistical methods are implemented in algorithms to allow a computer to “learn”
- Learning is data driven, i.e. predictions are made on learned information and is not hard-coded in advance by a developer
- Generally a processes of pattern recognition through defining features
- i.e. targeted adverts when you use Google

What is machine learning?

Supervised learning
Training data has a known solution and has been labeled accordingly, i.e. yes/no, red/green/yellow/blue, 0/1, a distinct value, gray-scale, doesn't have to be binary, for example tumor biopsy

Unsupervised learning
No prior knowledge given and the algorithm identifies the features necessary to make predictions, i.e. clustering similar as in BLEND

Semi-supervised learning
Where only part of the training data has been assigned a label due to missing information

Re-inforcement learning
Learning from mistakes; deep-learning → way too expensive → leave to Google

What is machine learning?

supervised learning to train a model to make predictions

classification and regression

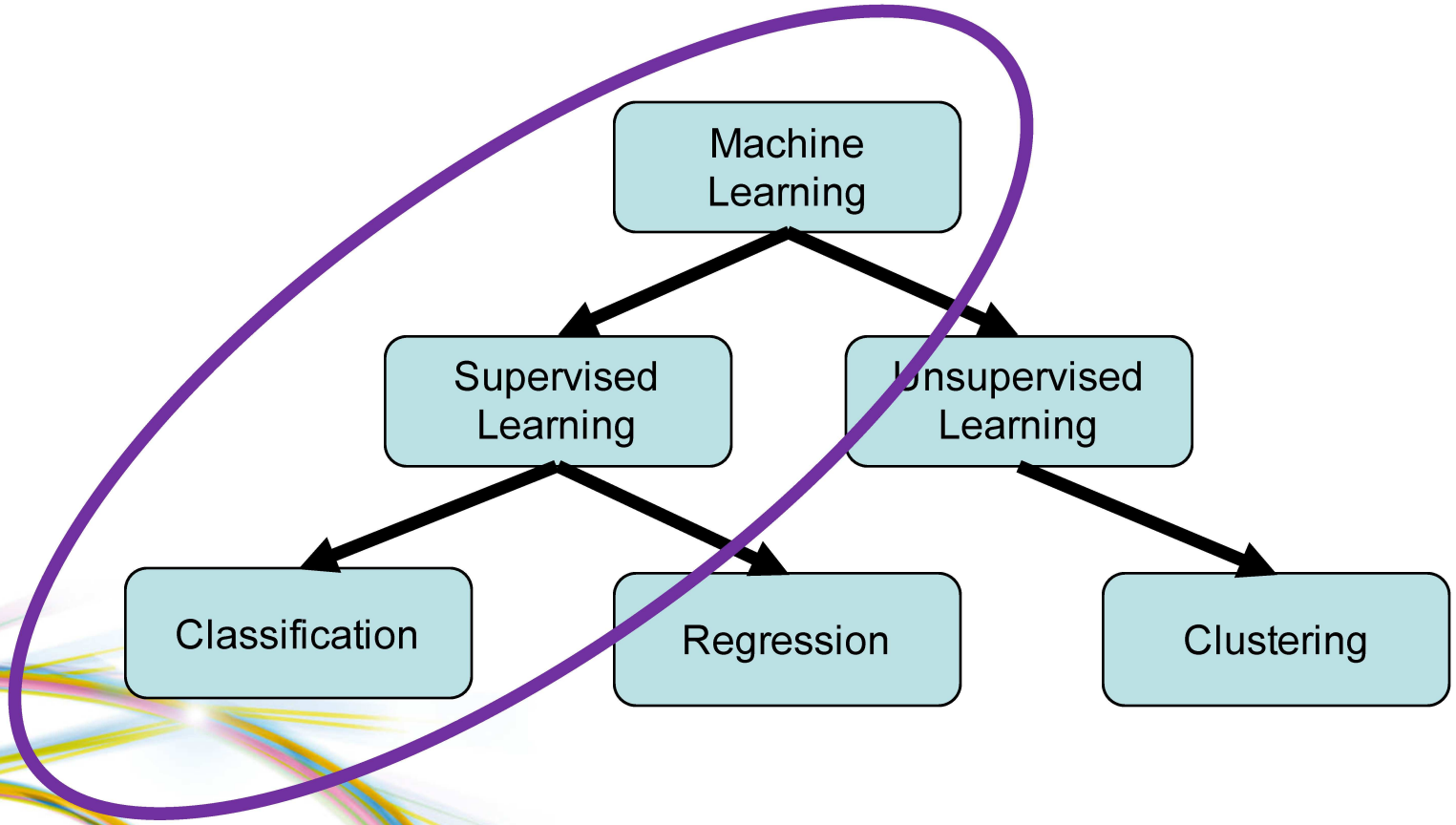
One or multiple distinct classes, e.g. is a biopsy sample malignant or benign, is the object in the picture a dog or a house

A distinct value, e.g. the market value of a house in a certain geographical area, most likely value of the £££ tomorrow

CCP4 diamond

What is machine learning?

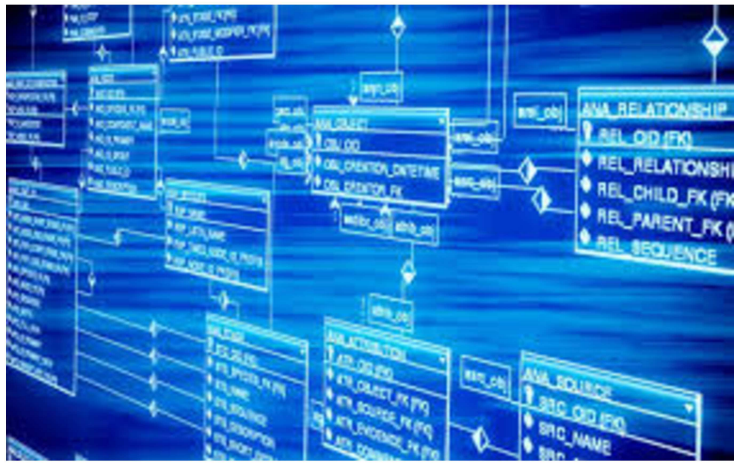
Supervised vs unsupervised; Classification vs regression



What is needed?

Essentials:

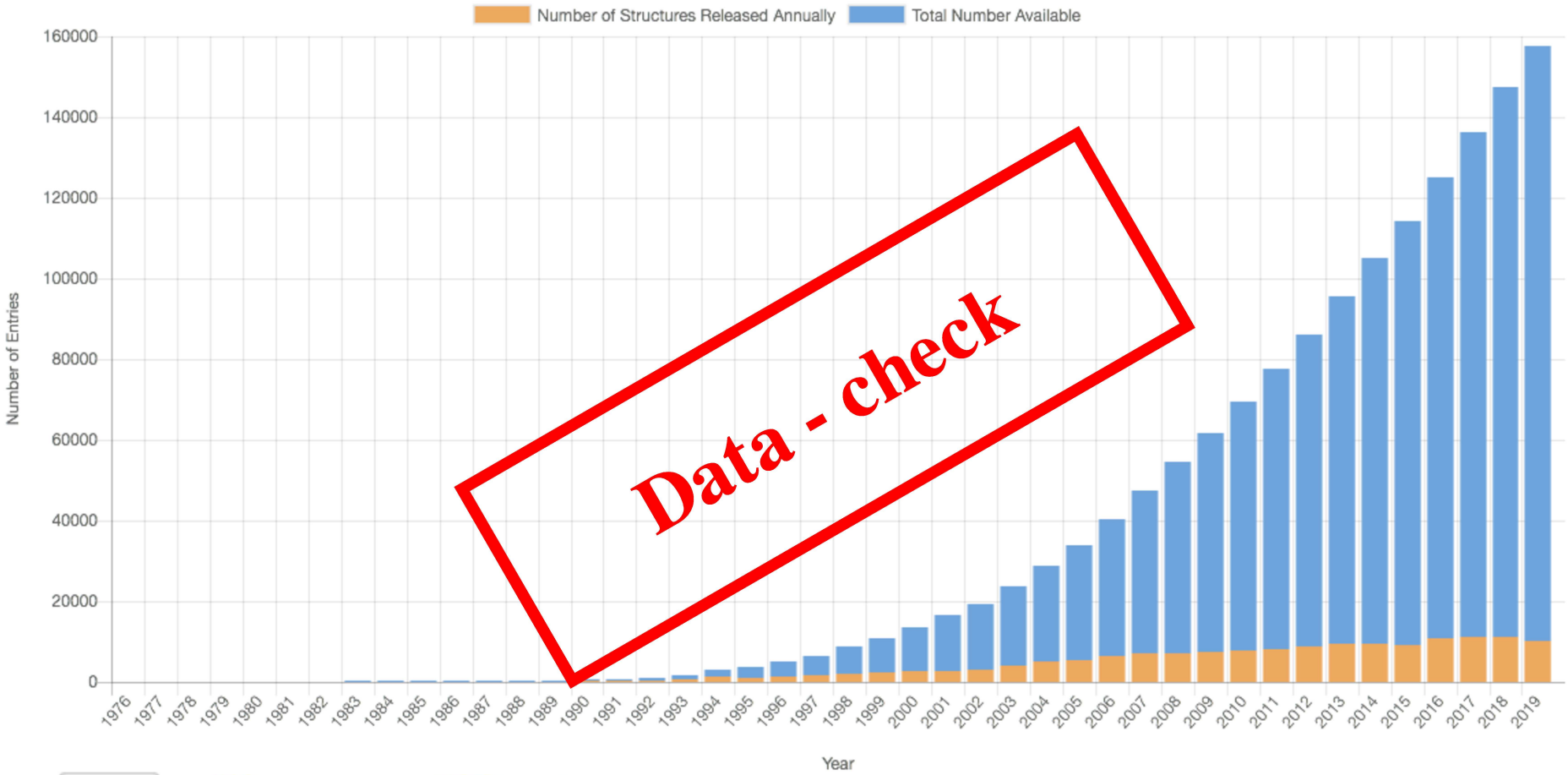
- Lots of clean and curated data in a database (i.e. PDB, ISPyB/Synchweb, lab notebook)
- Decent computing infrastructure, hardware and software
- Someone who puts it all together



CCP4 diamond

What is needed?

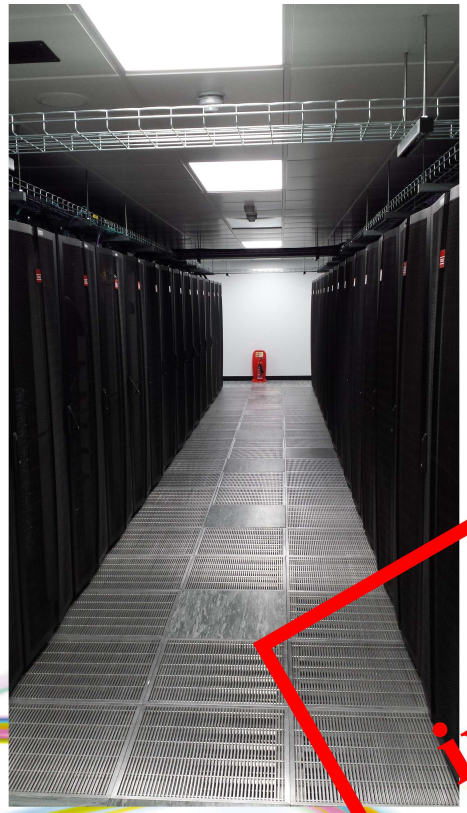
~158000 released structures in the PDB



CCP4

diamond

What is needed?



Computing infrastructure - check

+

Keras

python™

CCP4

diamond

What is needed?

Why machine learning and artificial intelligence?

Diamond operates ~210 days/year

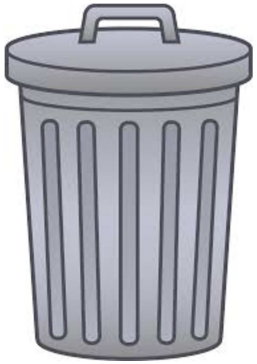
During operation 23 hrs/day

Diamond collects ~28,800 datasets/year

Diamond data results in ~900 structures/year

3%

97%



CCP4

diamond

Why machine learning and artificial intelligence?

Xchem facility → refer to Ailsa's talk

unattended data collection → some of you had data collected

↓

Combined produce ~600 datasets/day

126,000

2 days/dataset → 1200 days

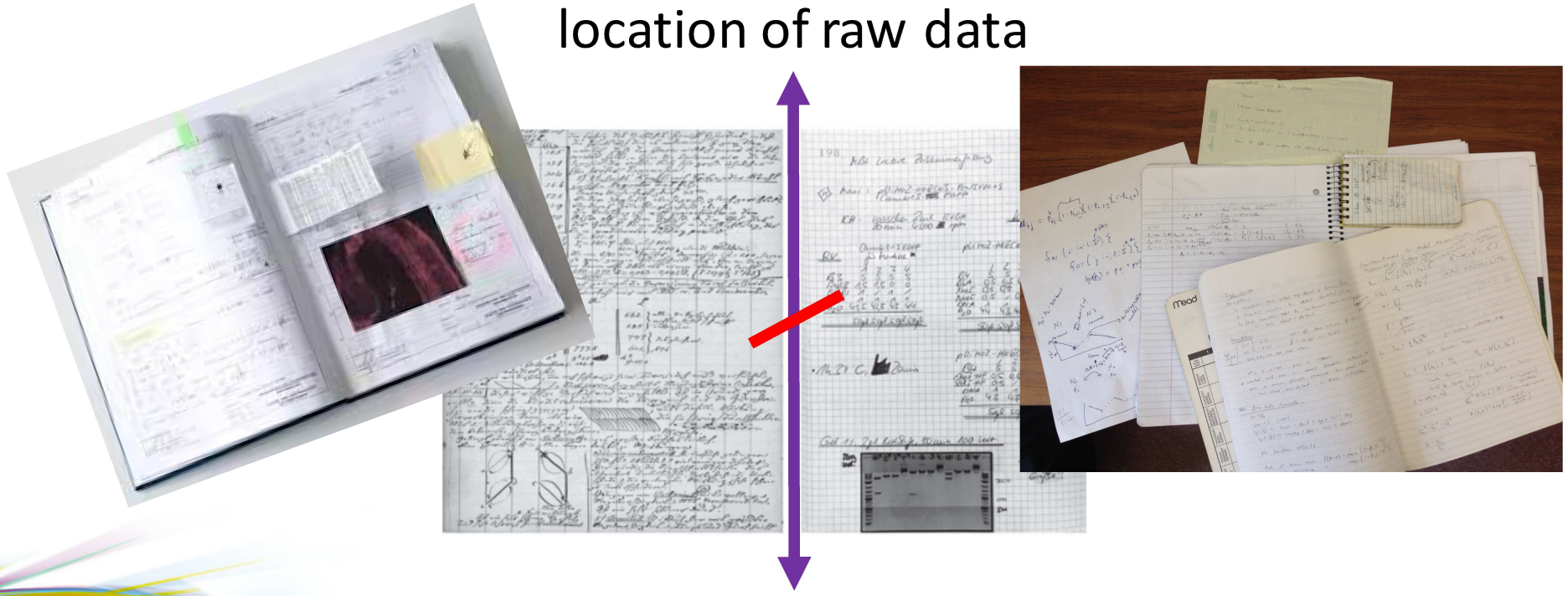
→ 3 years of working 365days/year

→ 1 PhD

CCP4 diamond

The trouble with data

ISPyB/Synchweb has data collection details and image location of raw data



PDB has refined, atomic coordinates, structure factors and phases

Inconsistent crystallographic metrics across the PDB

Most diffraction data not public

CCP4 diamond

Training data and METRIX database

Data sources

JCSG → 507 structures
SGC → 303 structures

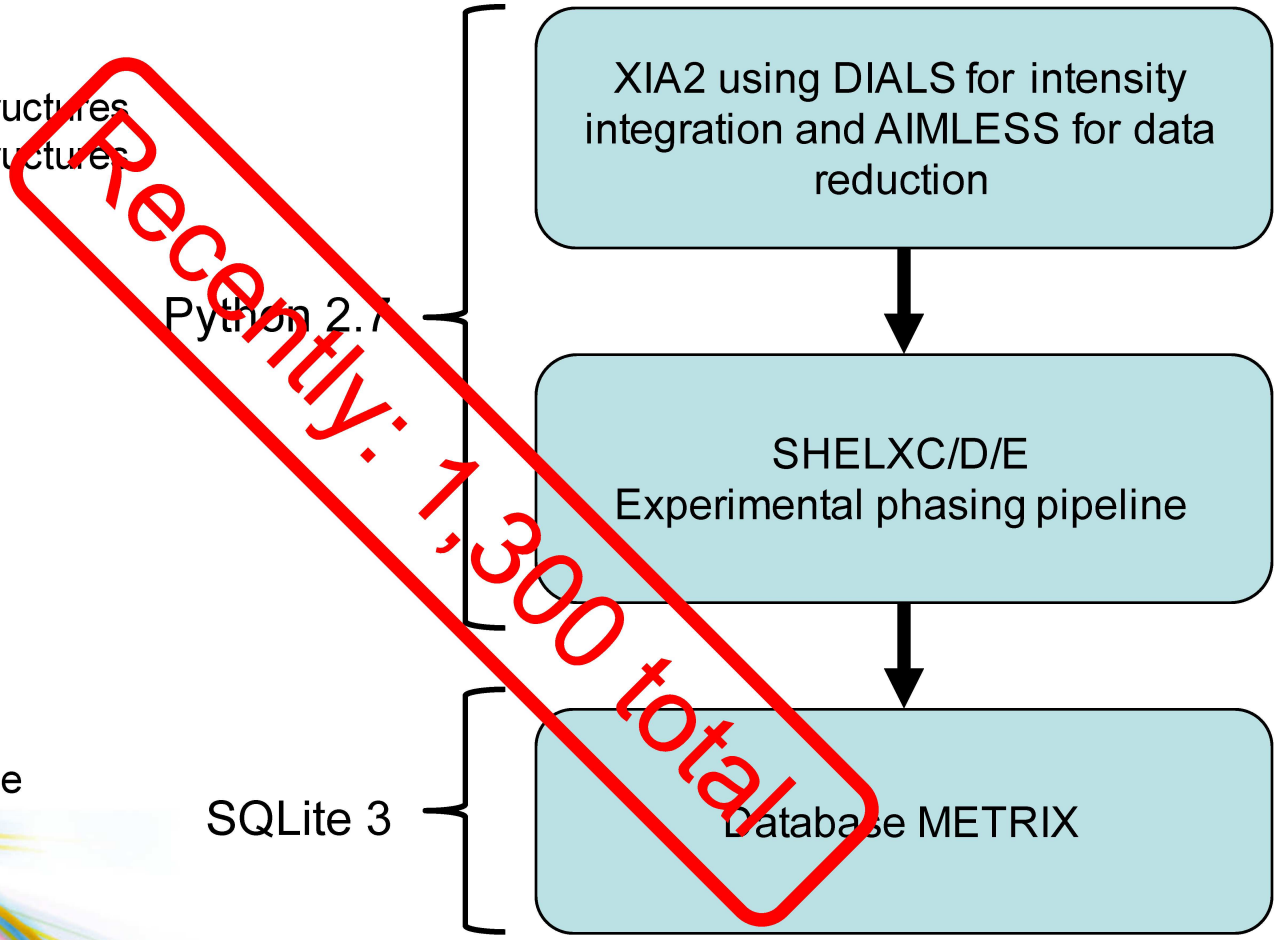
Phasing method:
S/MAD → 446
Native → 364

Resolution range:
1.05 – 3.8Å

Detector type:
CCD, PAD

X-Ray source:
Synchrotron, in-house

Protein:
6 – 100kDa

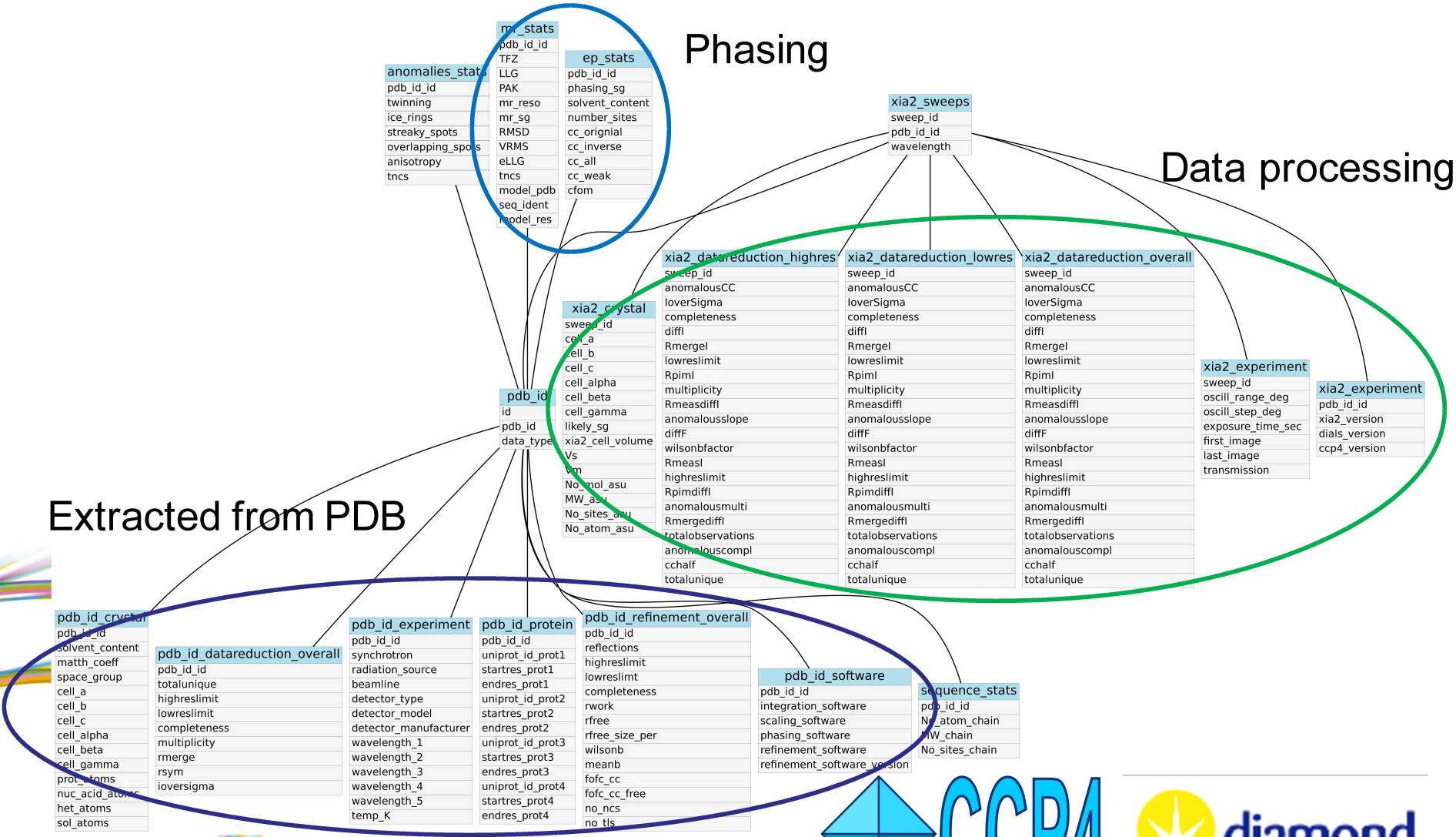


https://github.com/ccp4/metrix-database

CCP4 diamond

Training data and METRIX database

METRIX content and schema



https://github.com/ccp4/metrix-database

Aim

Is there predictive power in crystallographic metrics (multiplicity, completeness, different R values, high and low resolution limit)

If yes, are they useful for anything

If yes then:

Create a set of tools using machine learning to help users/crystallographers to solve their protein structures.

During data collection, e.g. giving recommendations

During data analysis, e.g. data reduction and phasing

Experimental phasing success

Pre-assessment and classifiers tried

Pre-assessment tried
- Linear Pearson's correlation coefficients
- Recursive feature elimination

Classifiers tried
- Support vector machine with linear kernel
- Support vector machine with RBF kernel
- Decision tree
- Decision tree with Bagging
- Decision tree with AdaBoost
- Random forest
- Extreme random forest

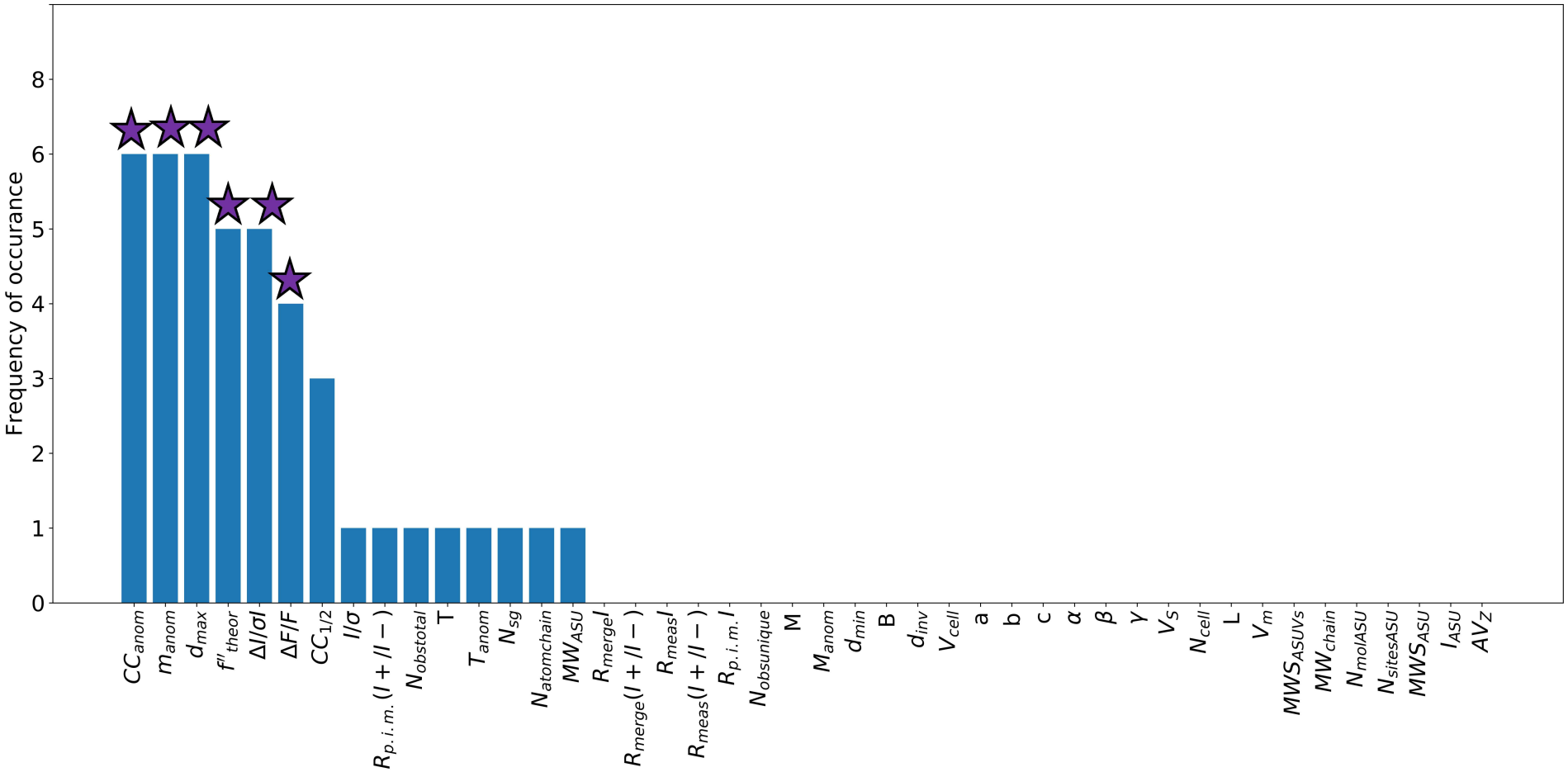
Combined results to identify most important decision making features;
Then retrain all classifiers and assess their performance;
Python 3.x

703 samples; stratified test-train split (20/80)
3-fold cross-validation (20/80 split)

https://github.com/ccp4/metrix_ml

Experimental phasing success

Important decision making features



d_{max} → low resolution cut-off

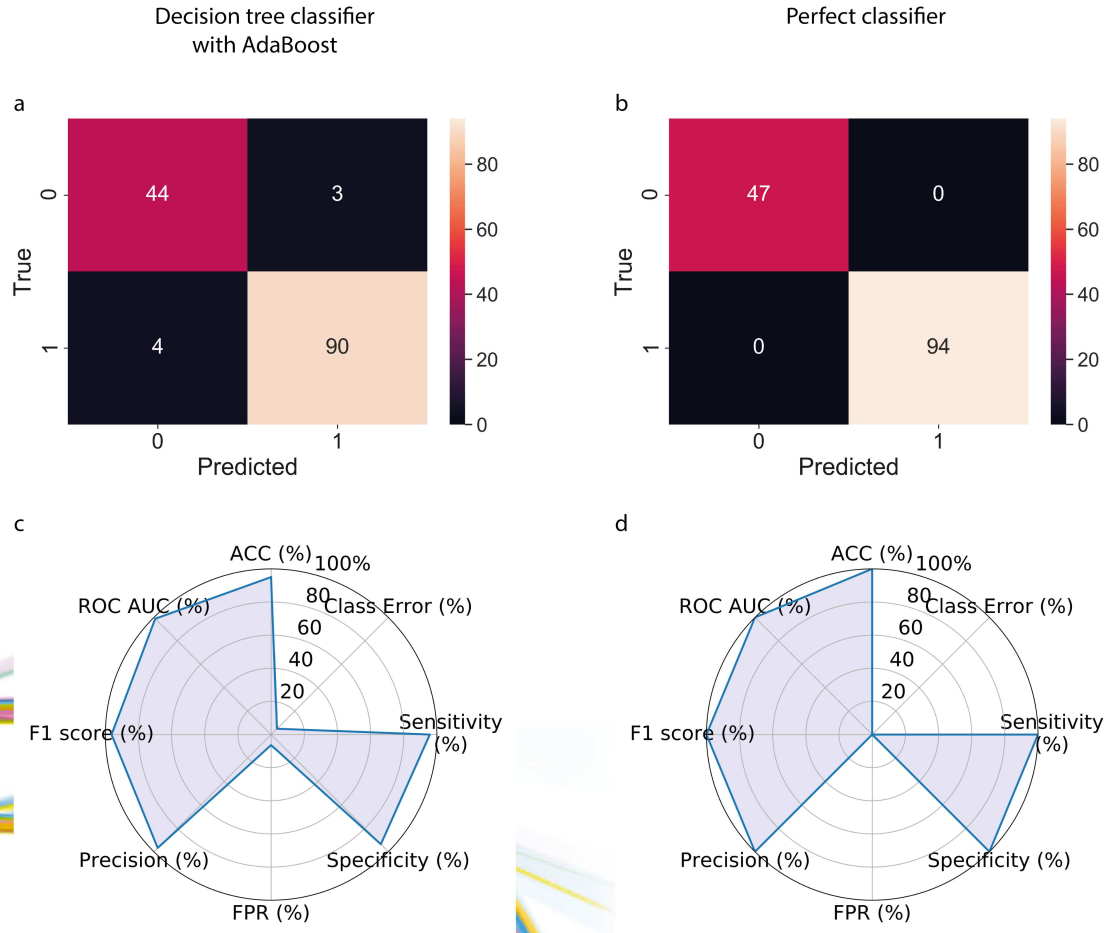
d_{min} → high resolution cut-off

https://github.com/ccp4/metrix_ml

CCP4 diamond

Experimental phasing success

Classifier assessment and performance



	Decision tree with AdaBoost	Perfect classifier
ACC (%)	95	100
Class Error (%)	5	0
Sensitivity (%)	96	100
Specificity (%)	94	100
FPR (%)	6	0
Precision (%)	97	100
F1 score (%)	96	100
ROC AUC (%)	99	100
TP test set	90	94
TN test set	44	47
FP test set	3	0
FN test set	4	0

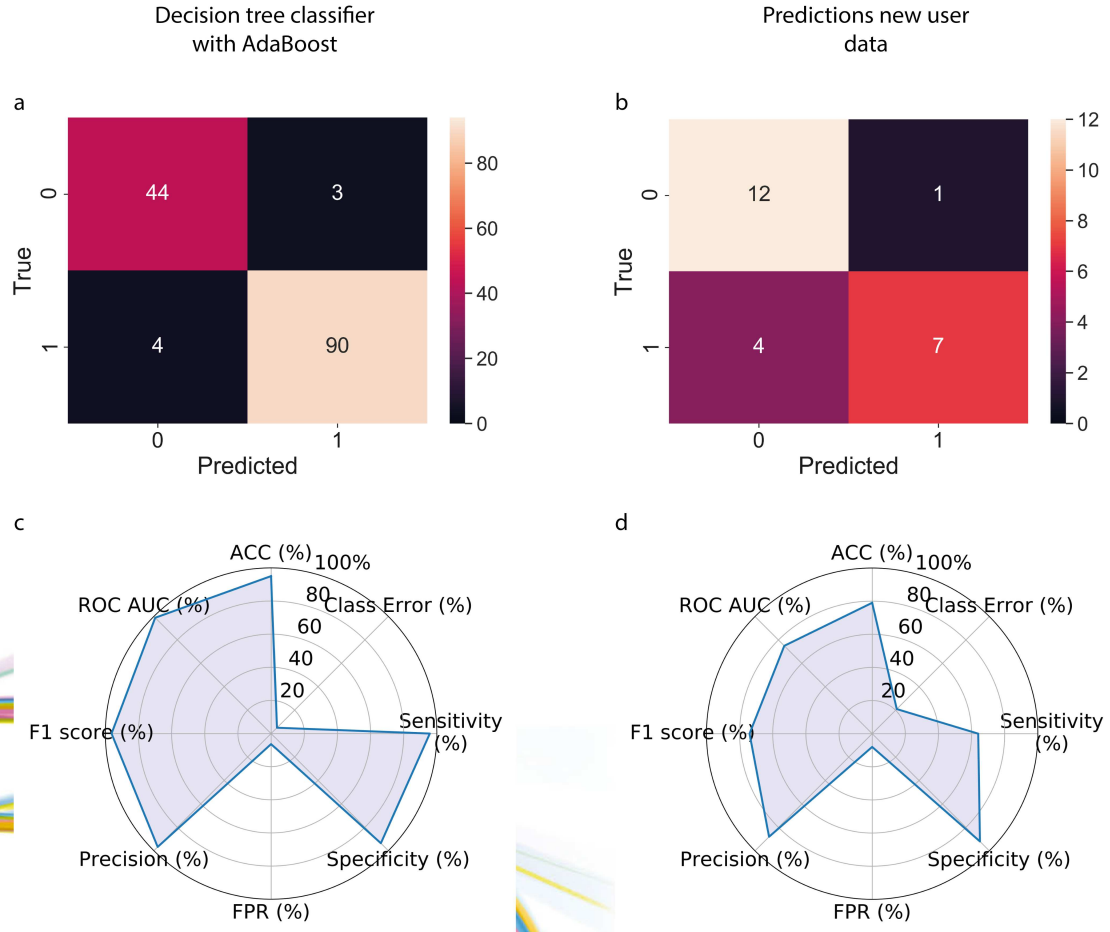
https://github.com/ccp4/metrix_ml

CCP4

diamond

Experimental phasing success

Predictions for new user data



	Decision tree with AdaBoost	New user data
ACC (%)	95	79
Class Error (%)	5	21
Sensitivity (%)	96	64
Specificity (%)	94	92
FPR (%)	6	8
Precision (%)	97	88
F1 score (%)	96	74
ROC AUC (%)	99	75
TP test set	90	7
TN test set	44	12
FP test set	3	1
FN test set	4	4

Probability cut-off for class 1: 80%

https://github.com/ccp4/metrix_ml

Molecular replacement success

Pre-assessment and classifiers tried

Pre-assessment tried
- Linear Pearson's correlation coefficients
- Recursive feature elimination

Classifiers tried
- Support vector machine with linear kernel
- Support vector machine with RBF kernel
- Decision tree
- Decision tree with Bagging
- Decision tree with AdaBoost
- Random forest
- Extreme random forest

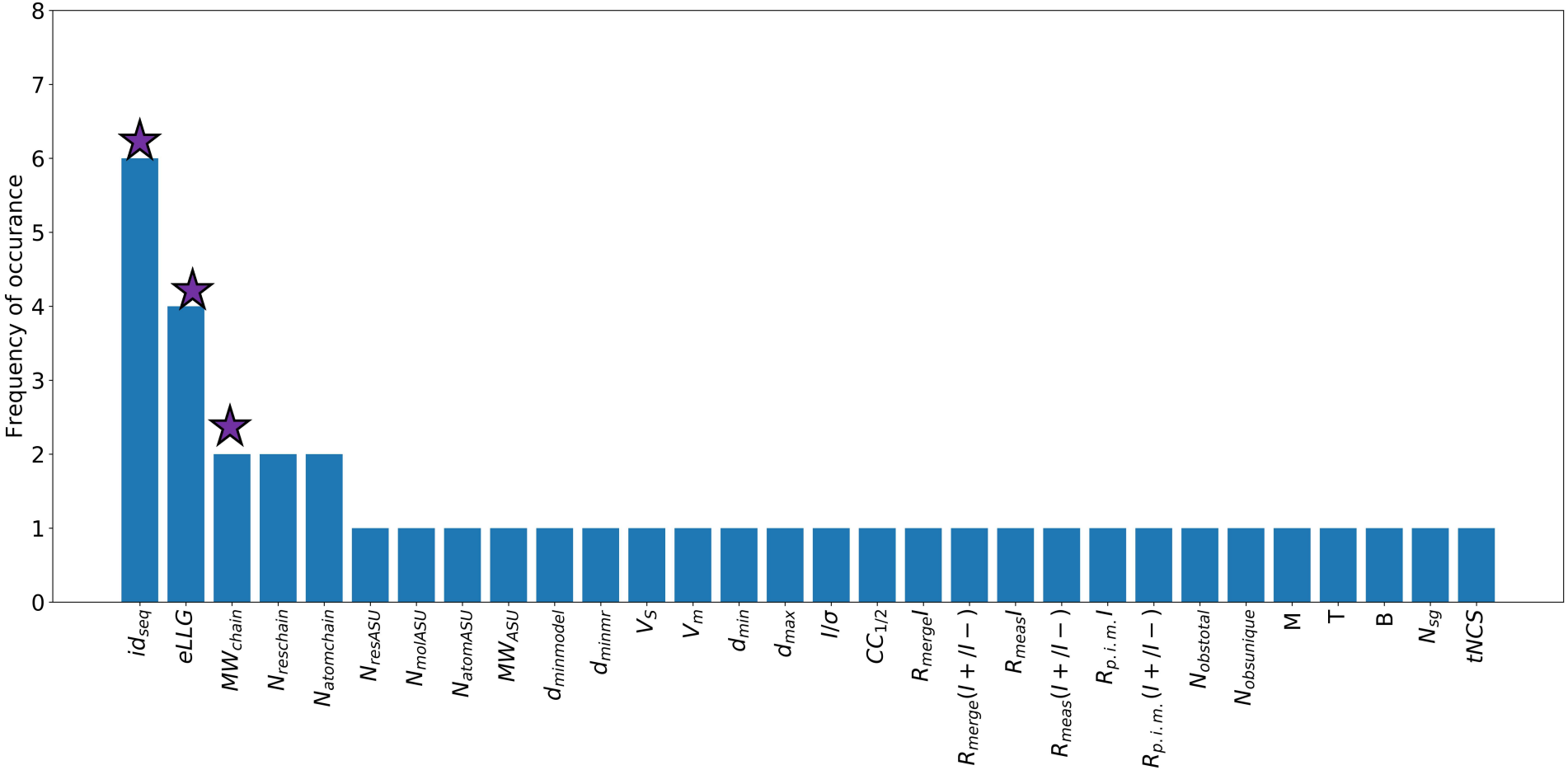
Combined results to identify most important decision making features;
Then retrain all classifiers and assess their performance;
Python 3.x

1020 samples; stratified test-train split (20/80)
3-fold cross-validation (20/80 split)

https://github.com/ccp4/metrix_ml

Molecular replacement success

Important decision making features



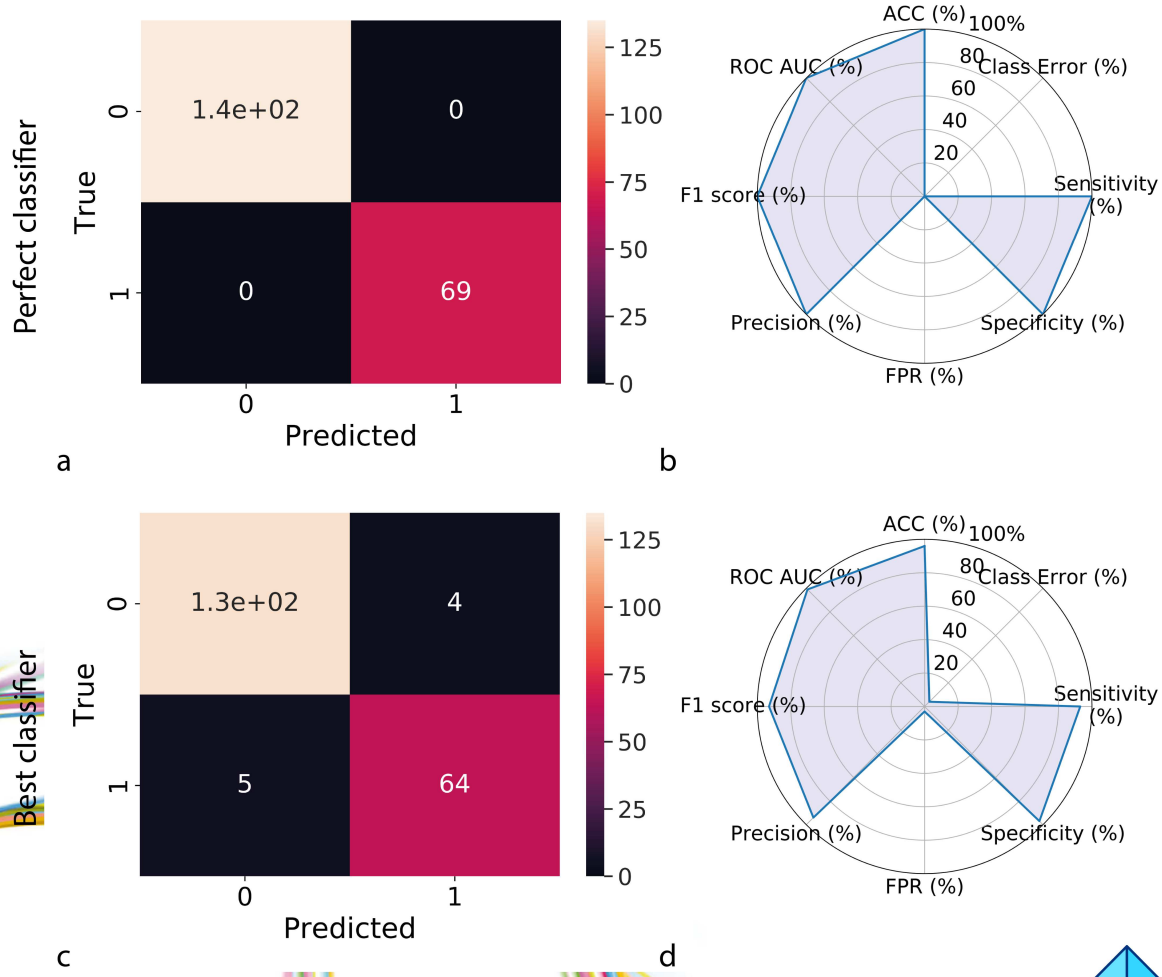
https://github.com/ccp4/metrix_ml

CCP4

diamond

Molecular replacement success

Classifier assessment and performance



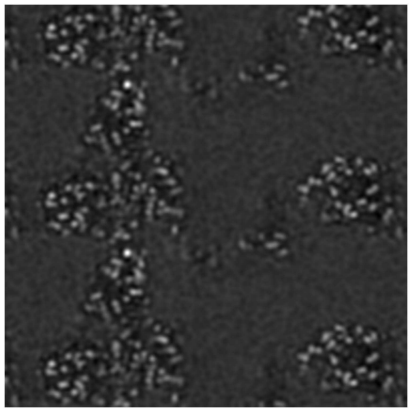
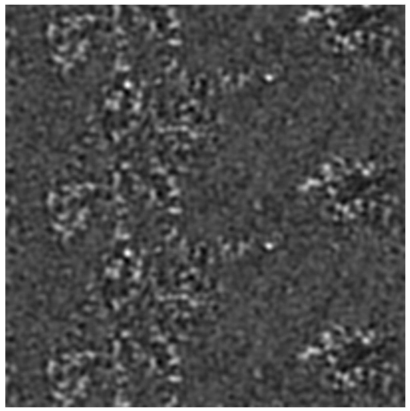
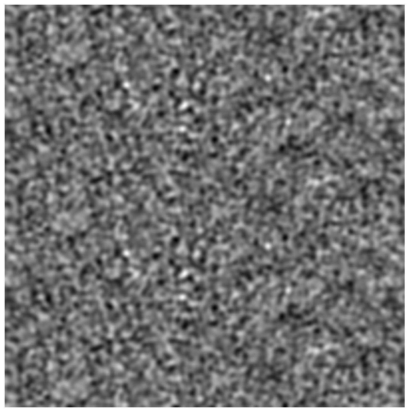
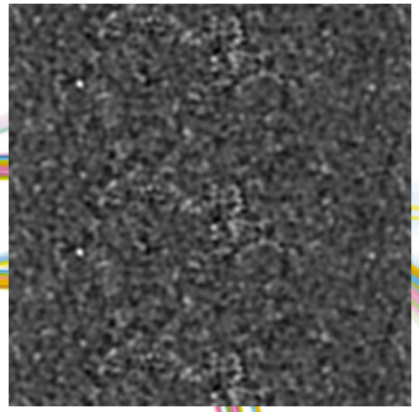
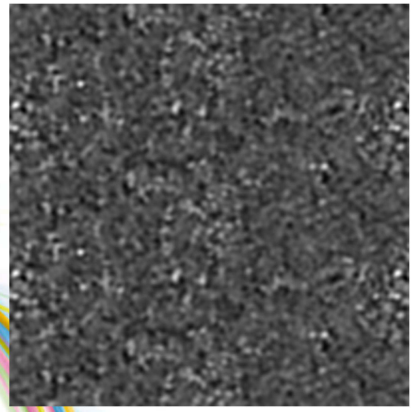
	Decision tree with AdaBoost	Perfect classifier
ACC (%)	96	100
Class Error (%)	4	0
Sensitivity (%)	93	100
Specificity (%)	97	100
FPR (%)	3	0
Precision (%)	94	100
F1 score (%)	93	100
ROC AUC (%)	99	100
TP test set	64	69
TN test set	131	135
FP test set	4	0
FN test set	5	0

https://github.com/ccp4/metrix_ml

CCP4 diamond

Map traceability

”Good” vs “bad” map PDB entry: 4DNK

solvent flattening backbone tracing	solvent flattening no backbone tracing	no solvent flattening no backbone tracing	
			“good”
			“bad”

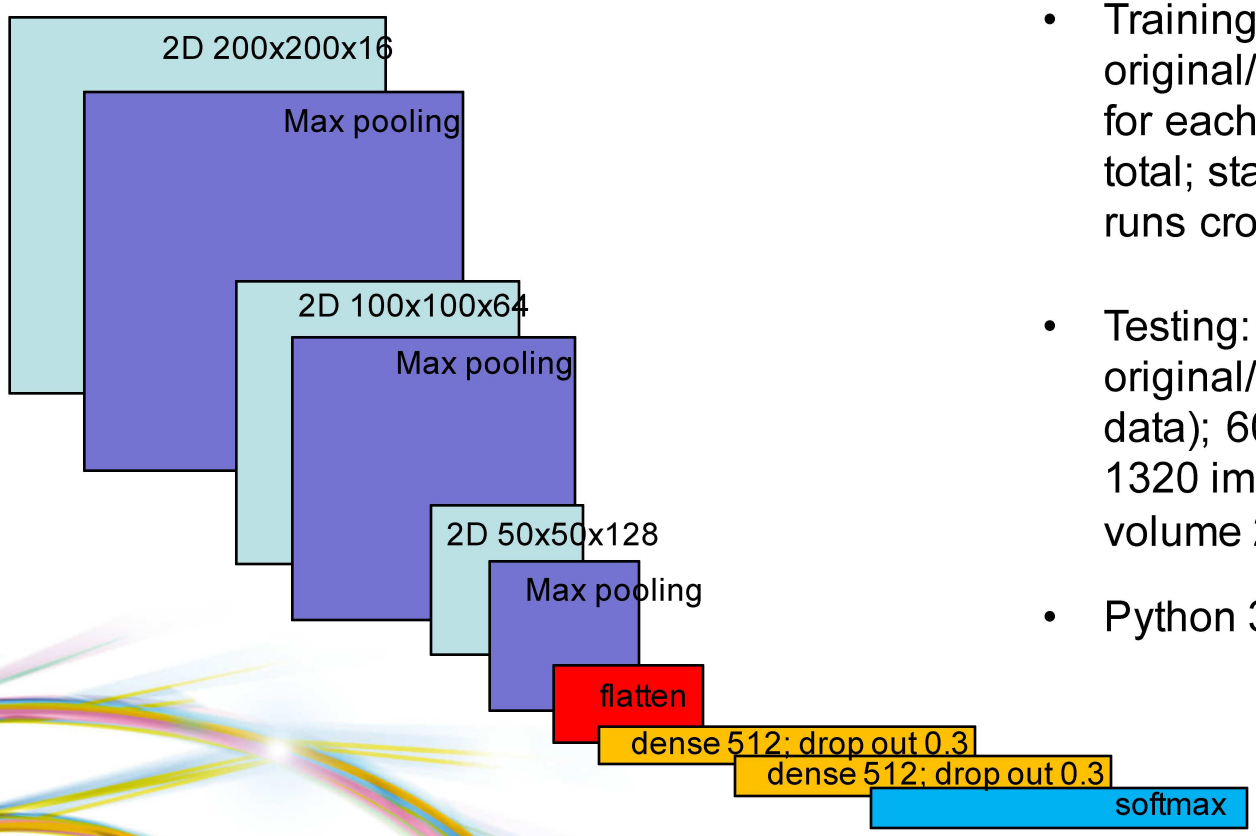
https://github.com/DiamondLightSource/python-topaz3

CCP4

diamond

Map traceability

cNN



- Training: 400 maps or 200 original/inverse pairs; 60 slices (20 for each axes); 24,000 images in total; standard volume 200Å^3; 5 runs cross-validation 20/80 split
- Testing: 22 maps or 11 original/inverse pairs (~5% of total data); 60 slices (20 for each axes); 1320 images in total; standard volume 200Å^3
- Python 3.x

https://github.com/DiamondLightSource/python-topaz3

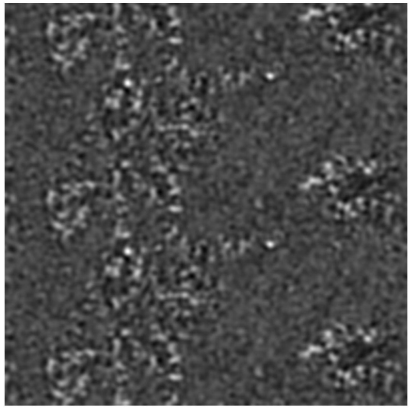
CCP4 diamond

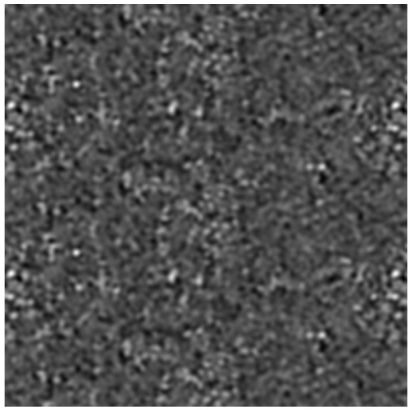
Map traceability

cNN assessment and performance

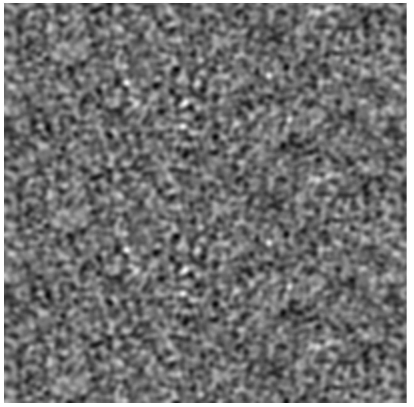
	Train: traced Test: traced	Train: traced Test: solvent, no build
ACC (%)	96	63
Class Error (%)	4	37
Sensitivity (%)	94	67
Specificity (%)	98	58
FPR (%)	2	42
Precision (%)	98	62
TP test set	621	442
TN test set	648	385
FP test set	12	275
FN test set	39	218

solvent flattening
no backbone tracing





no solvent flattening
no backbone tracing



https://github.com/DiamondLightSource/python-topaz3

CCP4

diamond

Map traceability

Implementation

```
/dls/<beamline>/data/<year>/<visit>/processed/<your_folder>/
<sample_folder>/<dataset_folder>/fast_ep/topaz3/
```

```
[ghp45345@cs03r-sc-serv-16 topaz3]$ cat avg_predictions.json
{
    "Original": {
        "0": 0.012835011336551228,
        "1": 0.987164831533432
    },
    "Inverse": {
        "0": 0.9392266973853112,
        "1": 0.060773291307587 3
    }
}[ghp45345@cs03r-sc-serv-16 topaz3]$ pwd
/dls/i04-1/data/2019/mx26335-
5/processed/GLKEIG_BrPhe/PACT_D1/PACT_D1_1_/fast_ep/topaz3
```

Protein can be build in original hand:
Confidence: 99%

Protein can be build in inverse hand:
Confidence: 6%

https://github.com/DiamondLightSource/python-topaz3

CCP4 diamond

Future plans

- Molecular replacement or experimental phasing success

–Feedback through Synchweb/ISPyB

–Include other software

- Map traceability

–Applying filters such as Gaussian, mean and median; data augmentation

–deep cNN for image denoising (Deep Image Prior, Ulyanov, D., Vedaldi, A., Lempitsky, V., https://arxiv.org/abs/1711.10925)

–ResNet and others

–From 2D to 3D

- On the side

–General integration into and querying from Synchweb/ISPyB

–Integration into CCP4 or some of its individual programs

–Expanding crystallographic data analysis framework

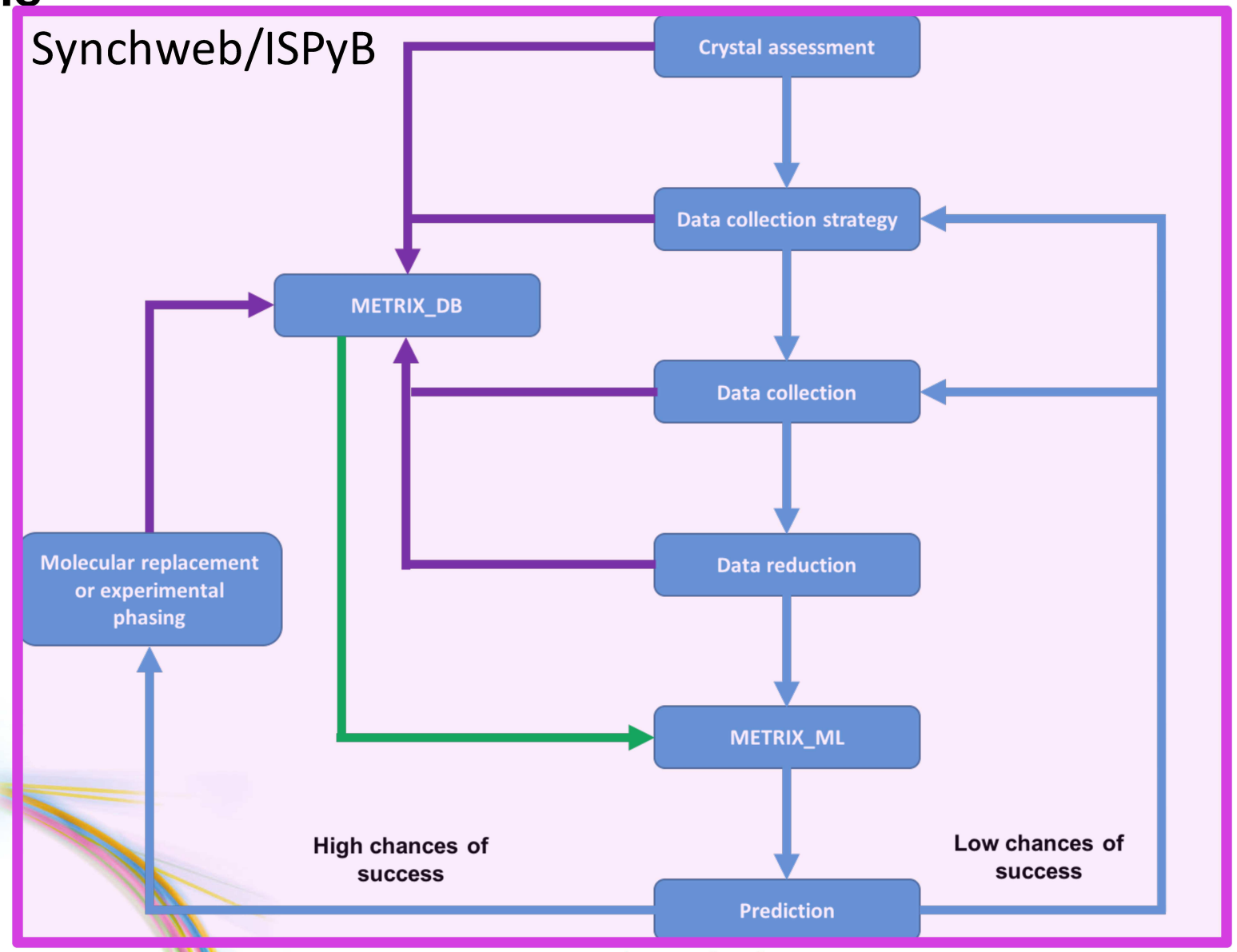
–Expanding METRIX, including public access

–Point/space group classifier

–Add other bioinformatics prediction tools

https://github.com/DiamondLightSource/python-topaz3

Future plans



https://github.com/DiamondLightSource/python-topaz3

CCP4

diamond

Acknowledgements:

Diamond Light Source:
James Parkhurst
Jenna Elliott (summer student 2018)
Tim Guite
Dominic Jaques (summer student 2016)
Gwyndaf Evans
Irakli Sikharulidze

CCP4:
David Waterman
Eugene Krissinel

MRC-LMB:
Garib Murshudov

University of Newcastle:
Arnaud Baslé

Vollmar, M., Parkhurst, J. M., Jaques, D., Baslé, A., Murshudov, G. N., Waterman, D. and Evans, G. (2019) IUCrJ, The predictive power of data processing statistics, submitted