

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



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

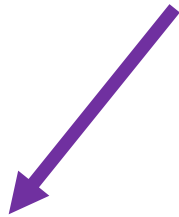
Re-inforcement learning

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

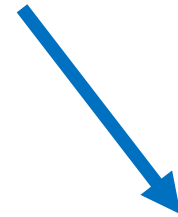


What is machine learning?

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

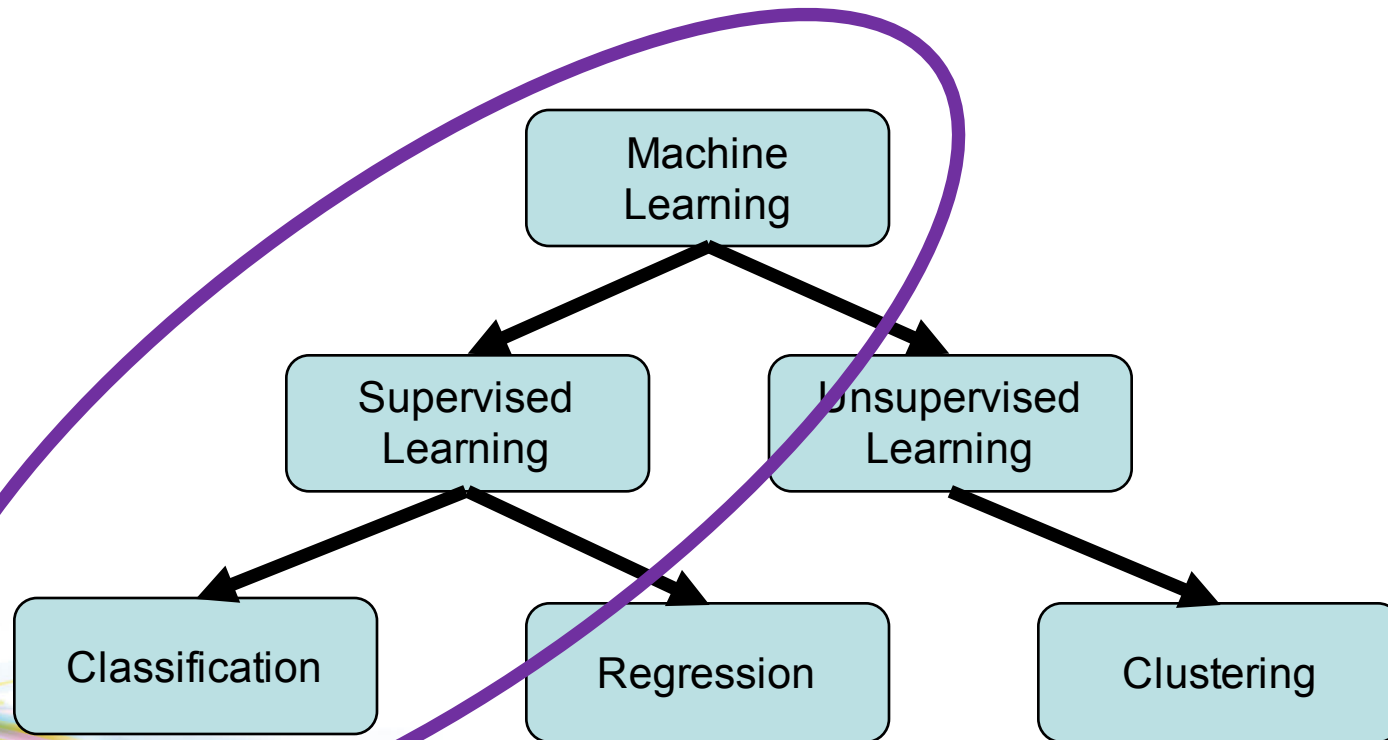


A distinct value, e.g. the
market value of a house in a
certain geographical area,
most likely value of the £££
tomorrow



What is machine learning?

Supervised vs unsupervised; Classification vs regression



What is needed?

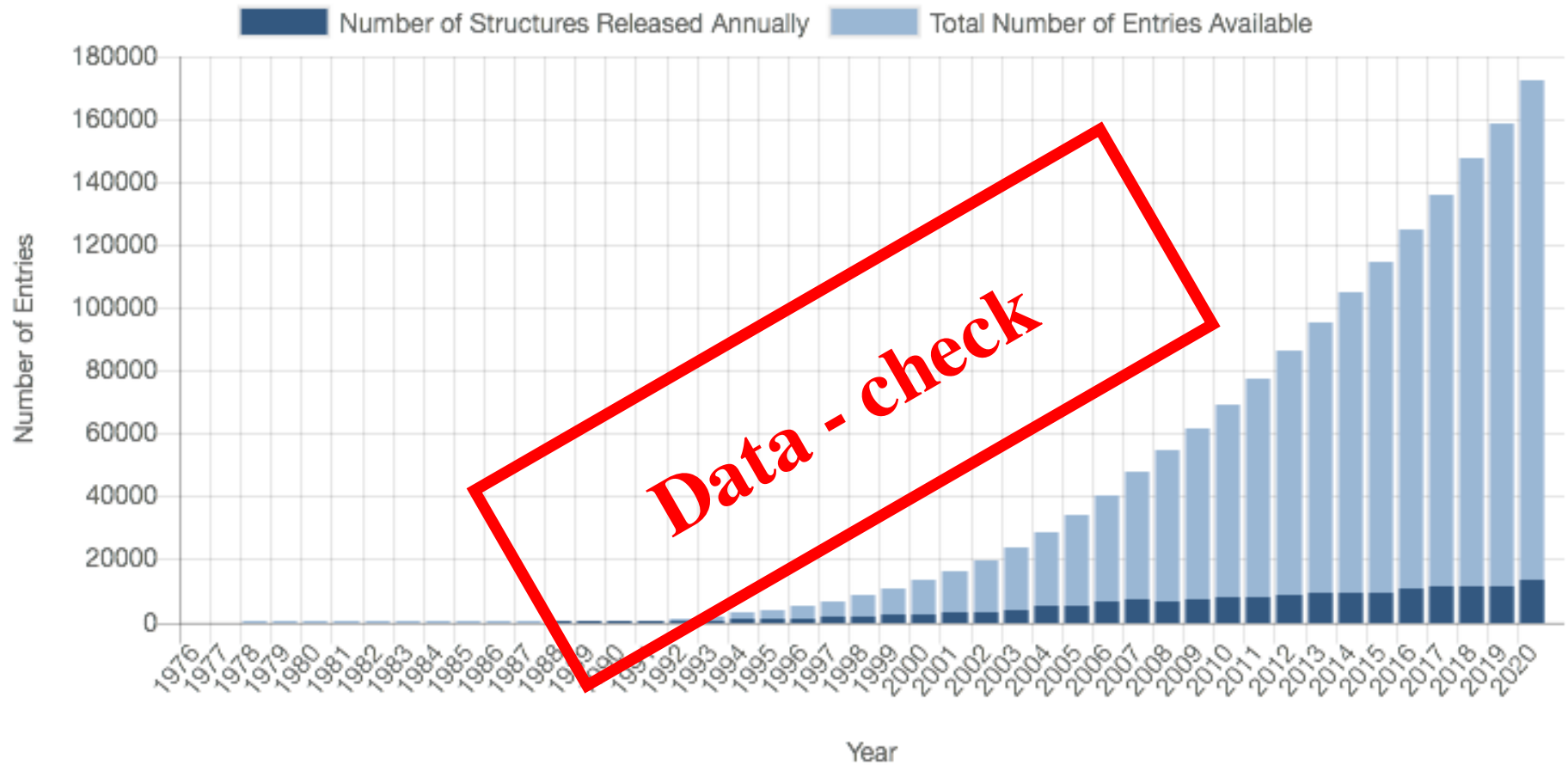
Essentials:

- 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



What is needed?

~170000 released structures in the PDB



Year

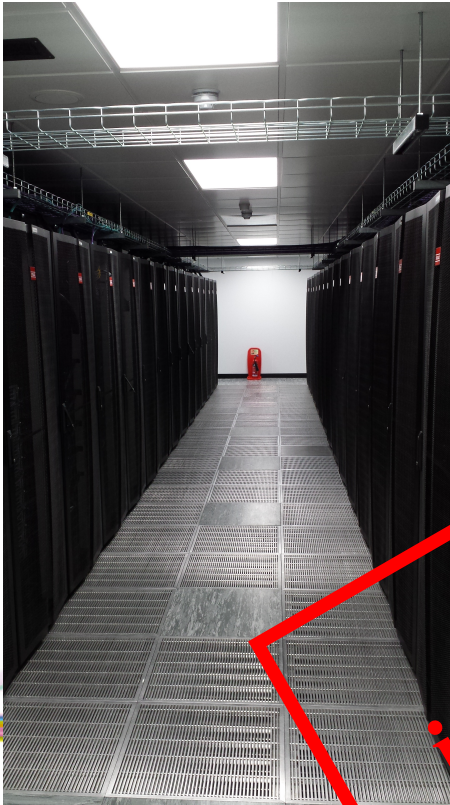


CCP4

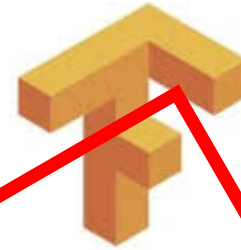


diamond

What is needed?



Computing
infrastructure - check



+



Keras



CCP4



diamond

What is needed?



Why machine learning and artificial intelligence?

Diamond operates ~210 days/year

During operation 23 hrs/day

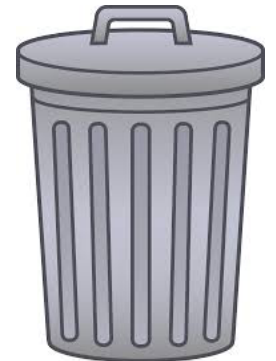
Diamond collects ~28,800 datasets/year

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



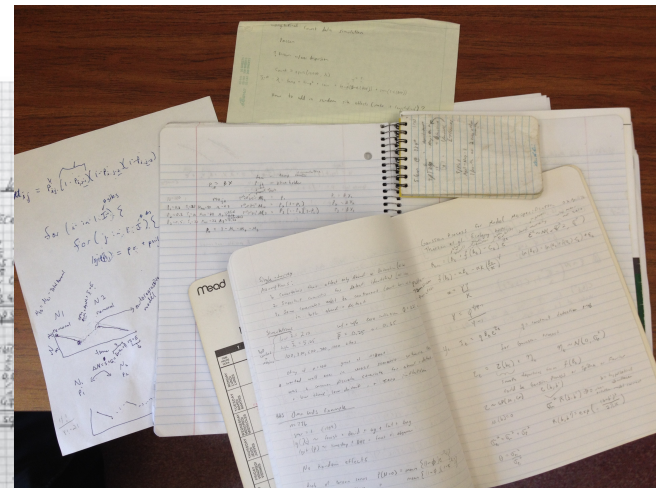
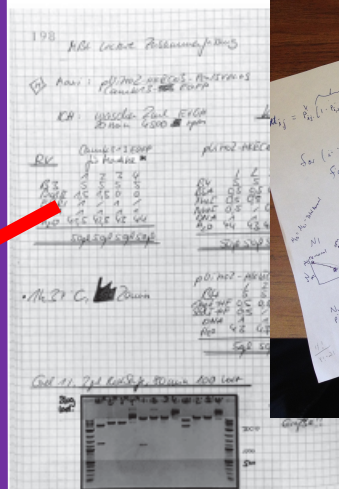
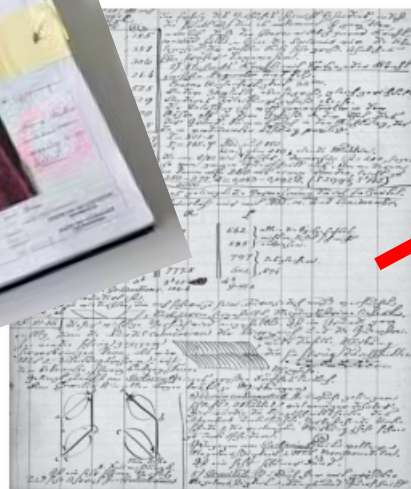
Combined produce ~600 datasets/day 126,000

2 days/dataset → 1200 days
→ 3 years of working 365days/year
→ 1 PhD



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



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

Python 2.7

SQLite 3

XIA2 using DIALS for intensity
integration and AIMLESS for data
reduction

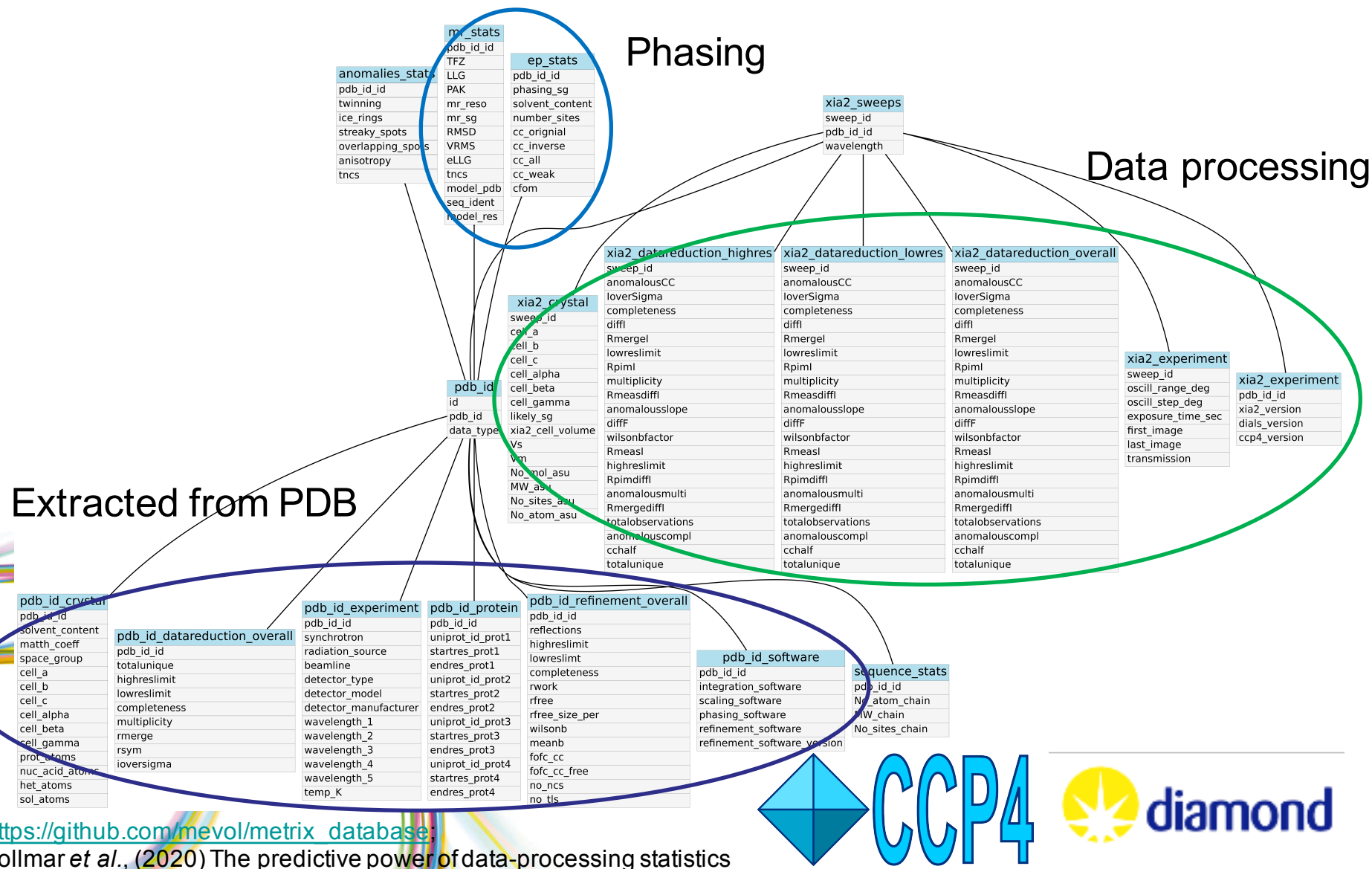
SHELXC/D/E
Experimental phasing pipeline

Database METRIX

Recently: 1,300 total

Training data and METRIX database

METRIX content and schema



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

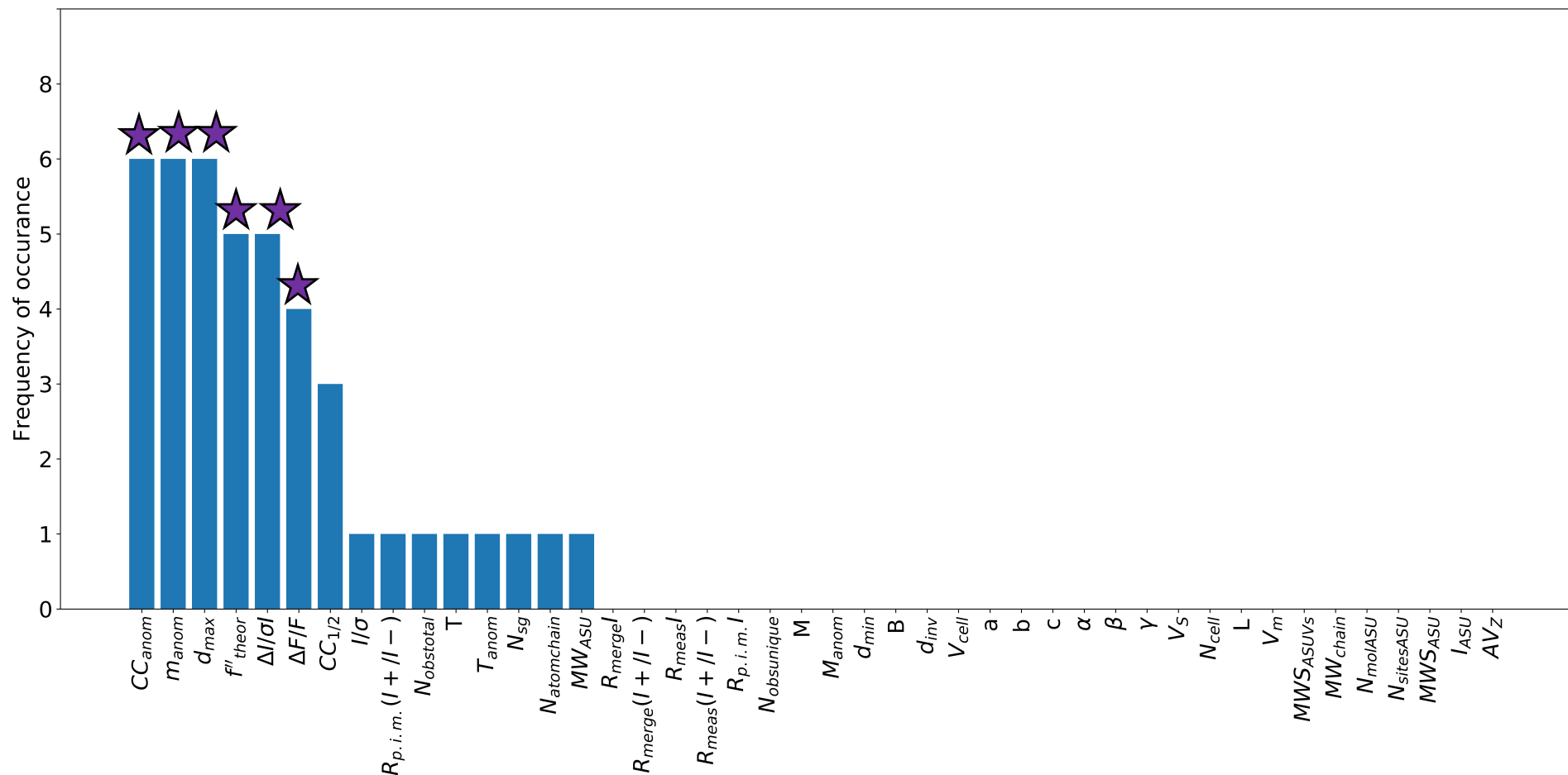
Combined results to identify most important decision making features;
Then retrain all classifiers and assess their performance;
Python 3.x

703 samples; stratified test-train split (20/80)

3-fold cross-validation (20/80 split)

Experimental phasing success

Important decision making features



d_{max} → low resolution cut-off

d_{min} → high resolution cut-off

https://github.com/mevol/metrix_ml;

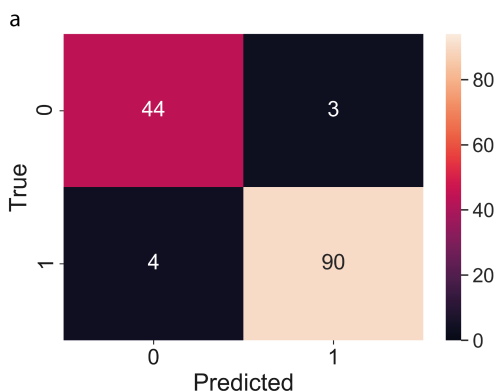
Vollmar et al., (2020) The predictive power of data-processing statistics



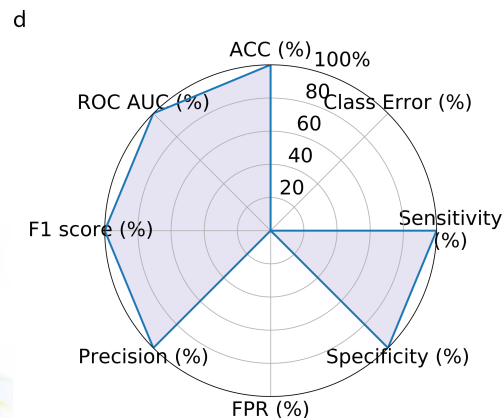
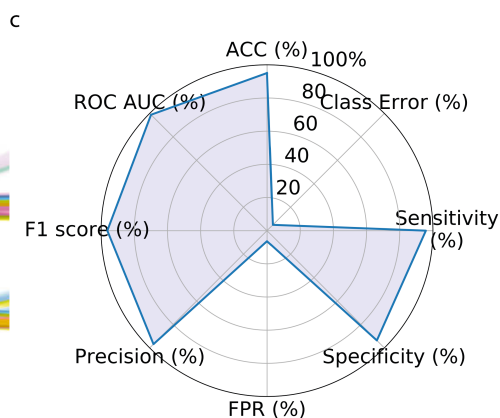
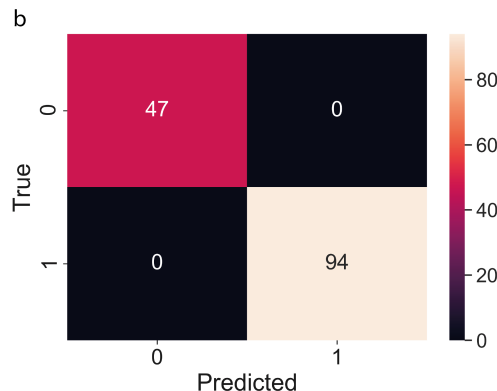
Experimental phasing success

Classifier assessment and performance

Decision tree classifier
with AdaBoost



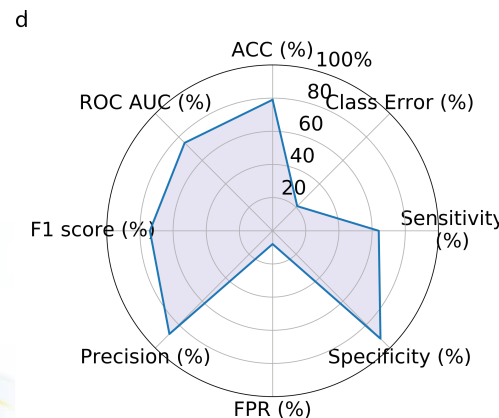
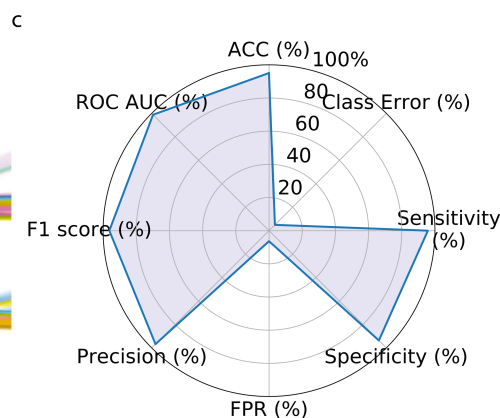
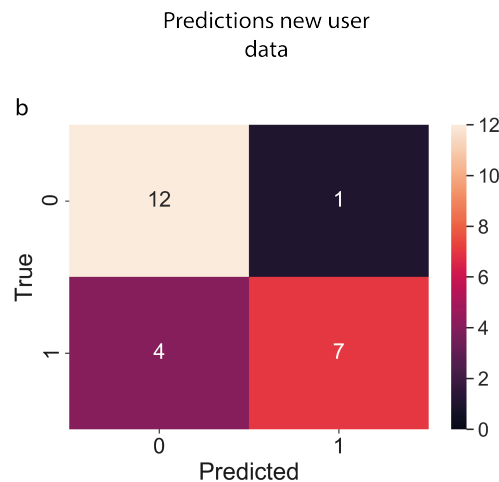
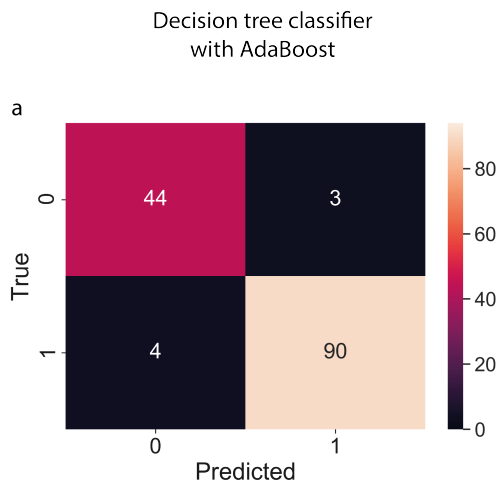
Perfect classifier



	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

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/mevol/metrix_ml;

Vollmar *et al.*, (2020) The predictive power of data-processing statistics



Diamond data analysis pipelines

Strategy estimations are carried out on datasets with fewer than 20 images

Data reduced using fast_dp and [xia2](#)

If a pdb file is provided, difference maps can be generated by [DIMPLE](#)

If a sequence file is provided, [MrBUMP](#) will run automatically to find a molecular replacement solution

If data integration highlights the possibility of anomalous signal and an [anomalous scatterer is specified in ISPyB](#), experimental phasing is carried out with Fast ep and/or Big ep

Input data:

/dls/i03/data/2020/cm26458-2/processed/gw/screen_collect_test/insu_11/insu_11_1_/04530c95-23a5-4f40-9c26-2f0ae1668f79/xia2-3dii/xia2.json

Output file: /dls/i03/data/2020/cm26458-

2/tmp/zocalo/gw/screen_collect_test/insu_11/insu_11_1_/bacfa167-4a55-49a-821b-5a0afa352bd5/ep_predict/ep_prediction.json

Results:

```
{'class': 0,
  'failure': 93.932,
  'metrics': {'Anomalous correlation': '0.28758',
              'Anomalous slope': '1.10816',
              'Low resolution limit': '20.74482',
              'dF/F': '0.07956',
              'dl/s(dI)': '1.03441',
              'f''': '0.23264'},
  'success': 6.068,
  'threshold': 80.0}}
```

Triggering thresholds for BigEP

resolution < 3.5Å

Average CC_anom > 25%

Average d"/sig > 0.9

Autocorrelation of CC_anom or d"/sig > 0.8



6-months evaluation

Run1_2020; Run2+3 (predictor was retrained with historic data plus data from run1_2020)

metric	After attempting EP				Adding trusted “0” predictions			
	3dii		dials		3dii		dials	
ACC (%)	23	71	21	55	51	93	51	93
Class error (%)	77	29	79	45	49	7	49	7
Sensitivity (%)	15	59	5	5	15	59	5	5
Specificity (%)	25	75	24	98	57	95	56	99
FPR(%)	75	25	76	2	43	5	44	1
Precision (%)	5	47	1	67	5	47	1	67
F1-score (%)	7	54	2	9	7	54	2	9
TP test set	4	24	1	2	4	24	1	2
TN test set	27	83	27	49	104	551	107	549
FP test set	80	27	84	1	80	27	84	1
FN test set	22	17	19	41	22	17	19	41

Verdict

Question:

Why is there such a large difference in the number of TP predictions for the two different integration packages?

Answer:

The data handed to XIA2 to calculate the stats is identical. However, the two integration engines XDS and DIALS handle the data differently despite using the same metric.

Consequence:

- Generally, values found for the DIALS engine are lower than for the XDS engine
- A low value for DIALS may denote a success while the same value for XDS will be a failure
- XDS has a higher false-positive rate
- DIALS has large number of samples predicted as class “0” or failure

metric	3dii	dials
CC_{anom}	0.53683	0.03829
m_{anom}	1.07437	0.34764
d_{max}	54.94927	55.01078
$\Delta F/F$	0.04635	0.06293
$\Delta I/\sigma(\Delta I)$	1.09233	0.44770
f''_{theor}	0.63423	0.63423
failure	2.414	40.308
success	97.586	59.692
class	1	0

Data example:

ins_2_4_ → test dataset commonly used for software development; both samples are build successfully



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)

Molecular replacement success

Important decision making features

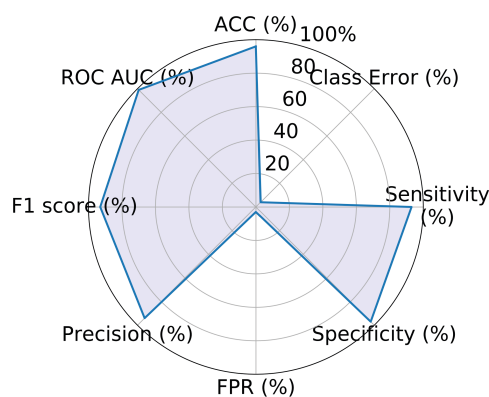
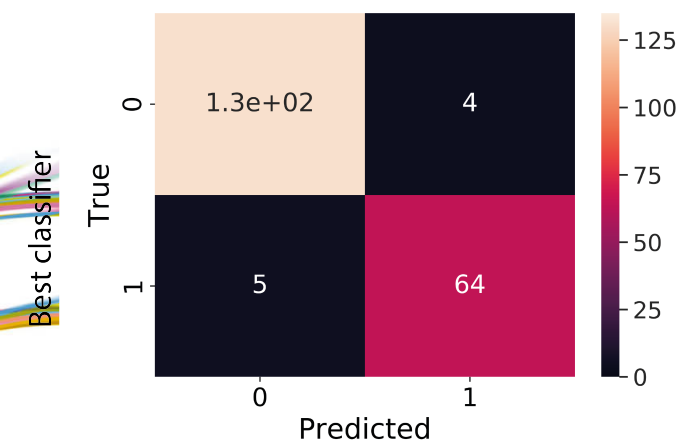
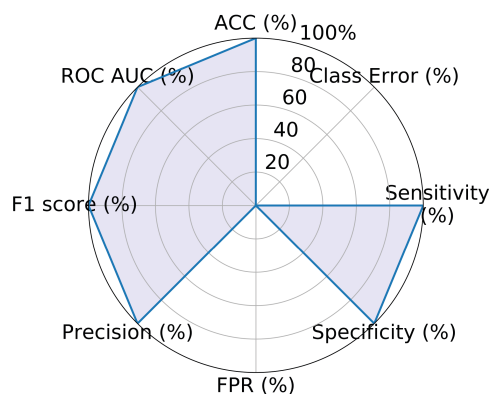
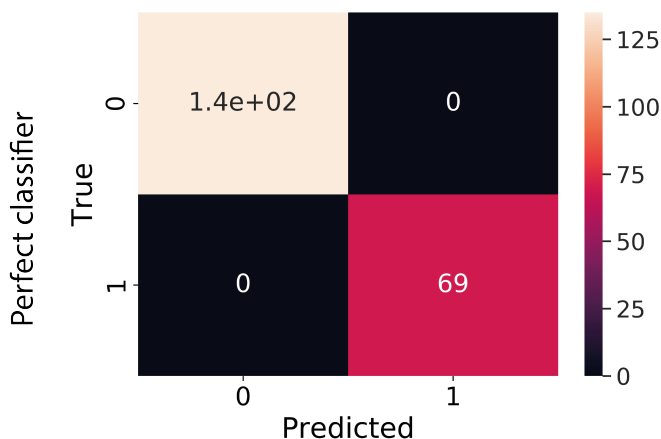


https://github.com/mevol/metrix_ml;

Vollmar *et al.*, (2020) The predictive power of data-processing statistics

Molecular replacement success

Classifier assessment and performance



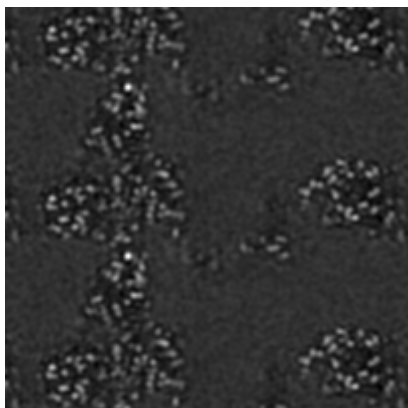
	Decision tree with AdaBoost	Perfect classifier
ACC (%)	96	100
Class Error (%)	4	0
Sensitivity (%)	93	100
Specificity (%)	97	100
FPR (%)	3	0
Precision (%)	94	100
F1 score (%)	93	100
ROC AUC (%)	99	100
TP test set	64	69
TN test set	131	135
FP test set	4	0
FN test set	5	0



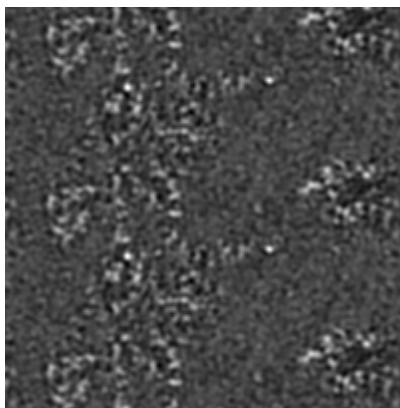
Map traceability

“Good” vs “bad” map PDB entry: 4DNK

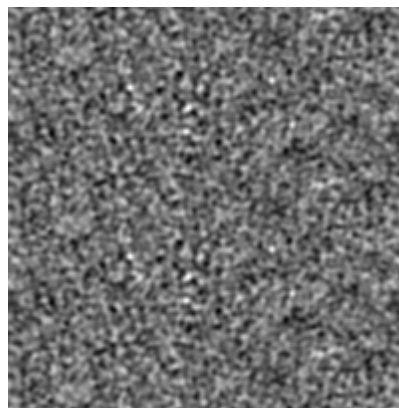
solvent flattening
backbone tracing



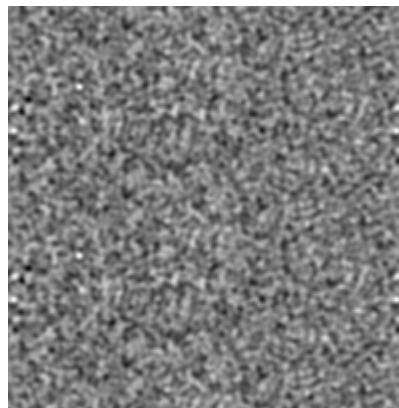
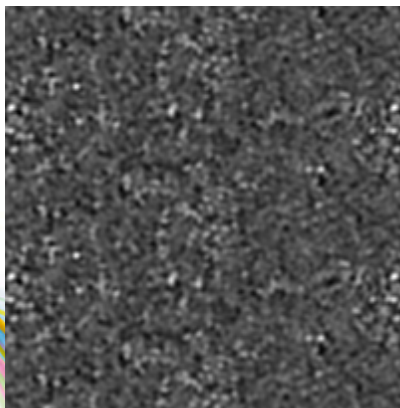
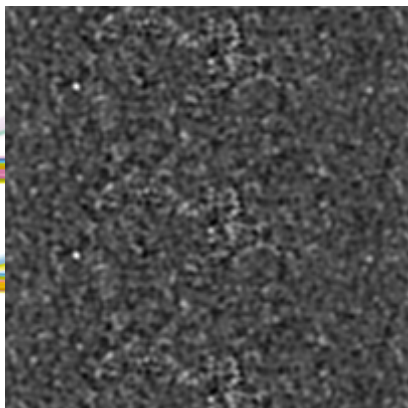
solvent flattening
no backbone tracing



no solvent flattening
no backbone tracing



“good”

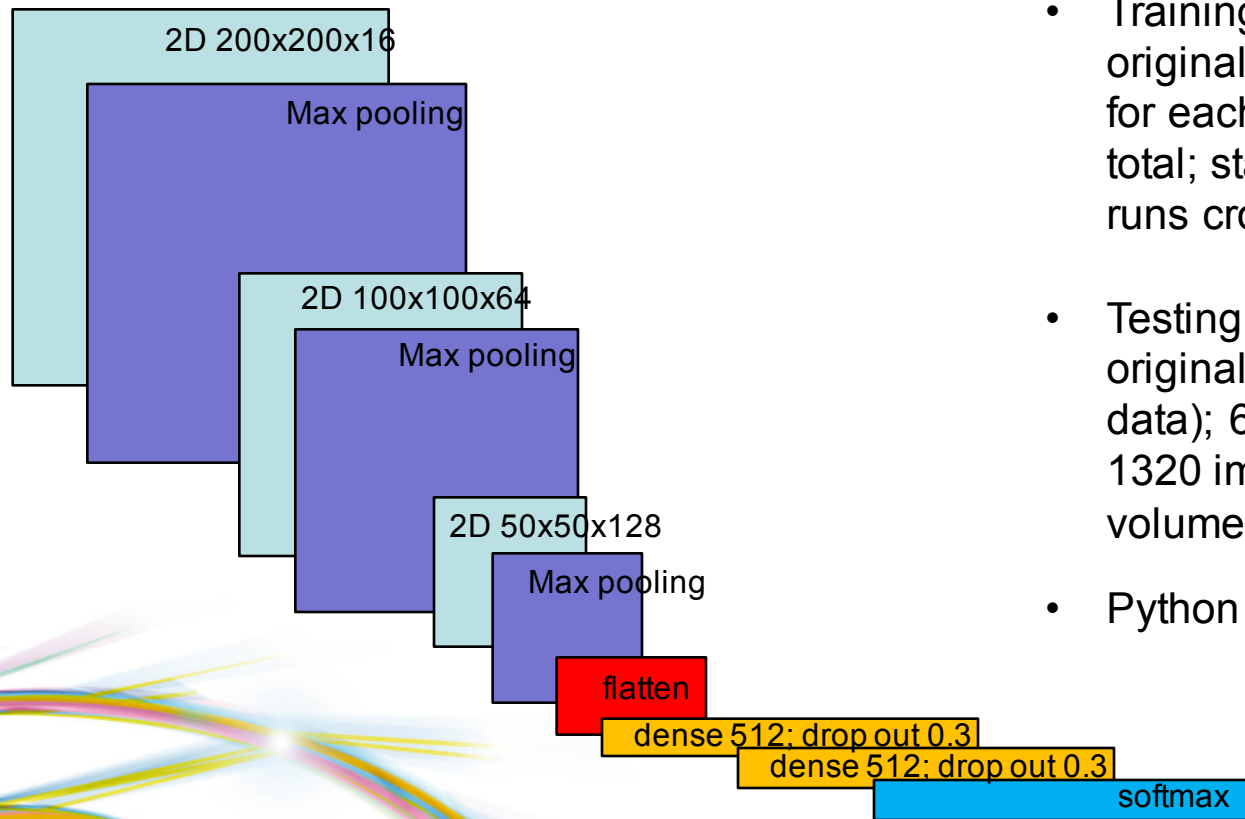


“bad”



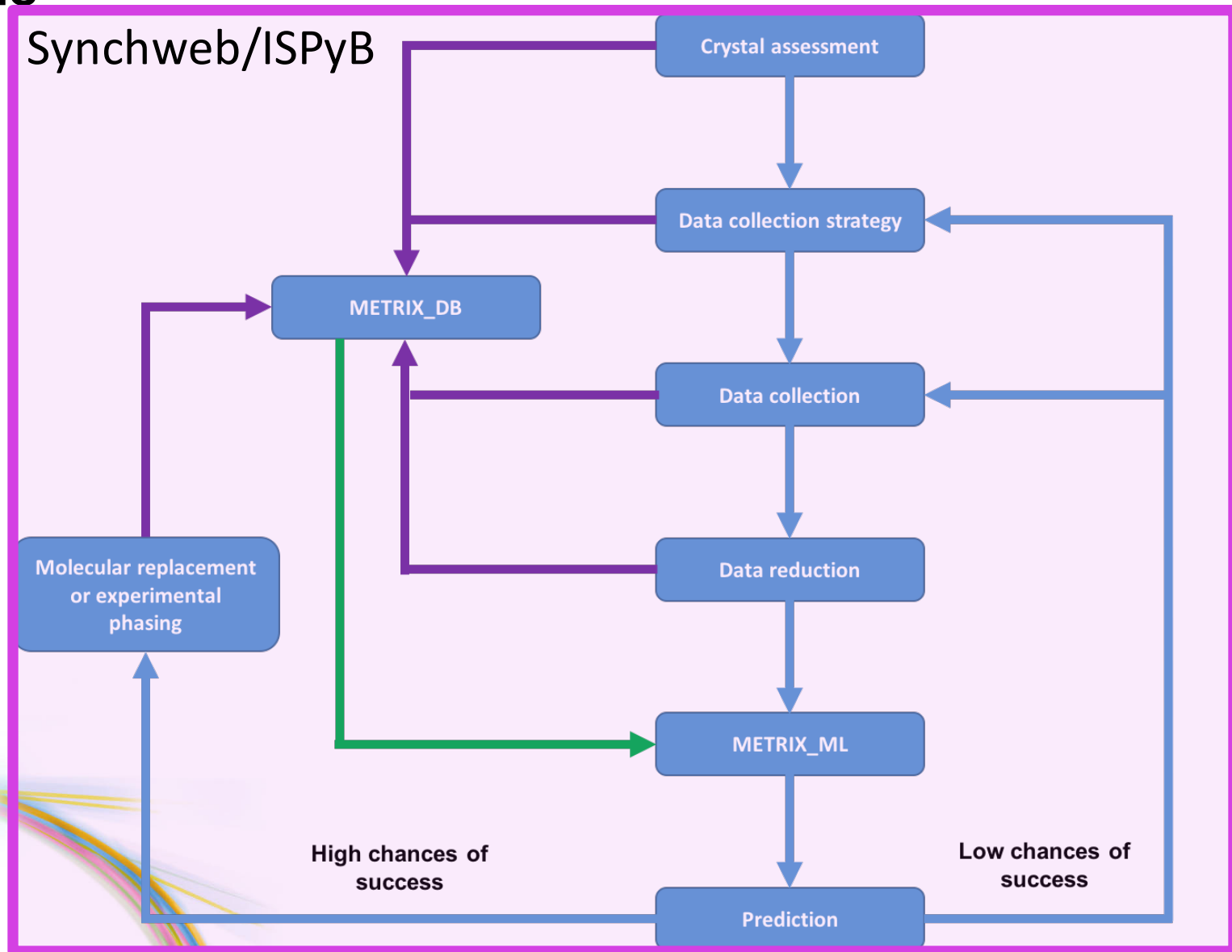
Map traceability

cNN



- Training: 400 maps or 200 original/inverse pairs; 60 slices (20 for each axes); 24,000 images in total; standard volume $200\text{\AA}^3$; 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 $200\text{\AA}^3$
- Python 3.x

Future plans



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

Rob Nicholls

University of Newcastle:

Arnaud Baslé