PAUL SCHERRER INSTITUT
PSI

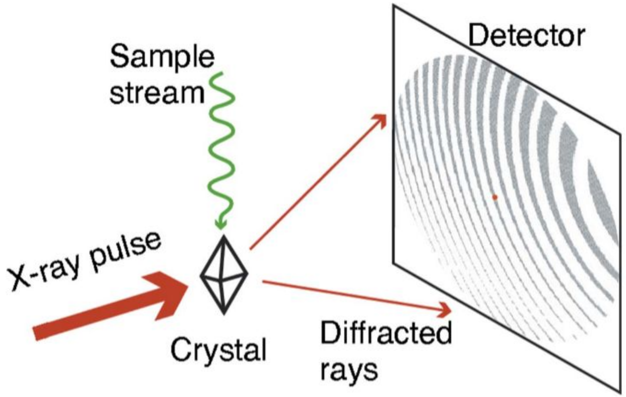
Piero Gasparotto :: DevCon

The REDML project

AWI Bi-Monthly Meeting – 17th Aug 2022

What and why?

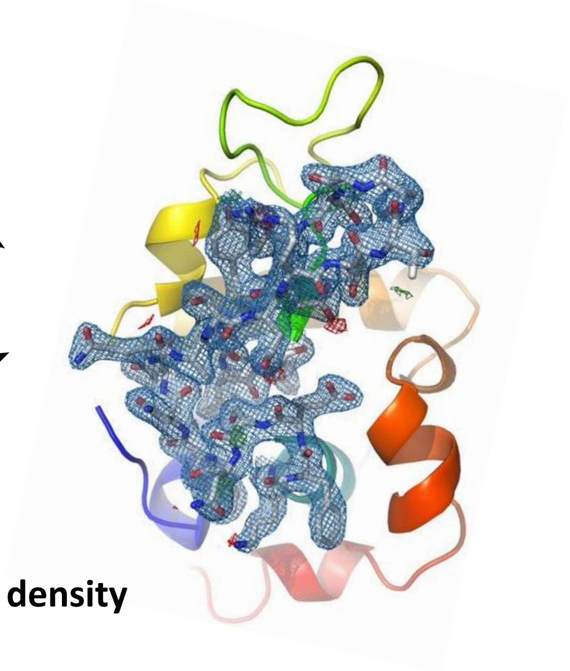
(Time-resolved) Serial crystallography



REDML

- Spot detection
- Indexing
- Integration

- Phasing

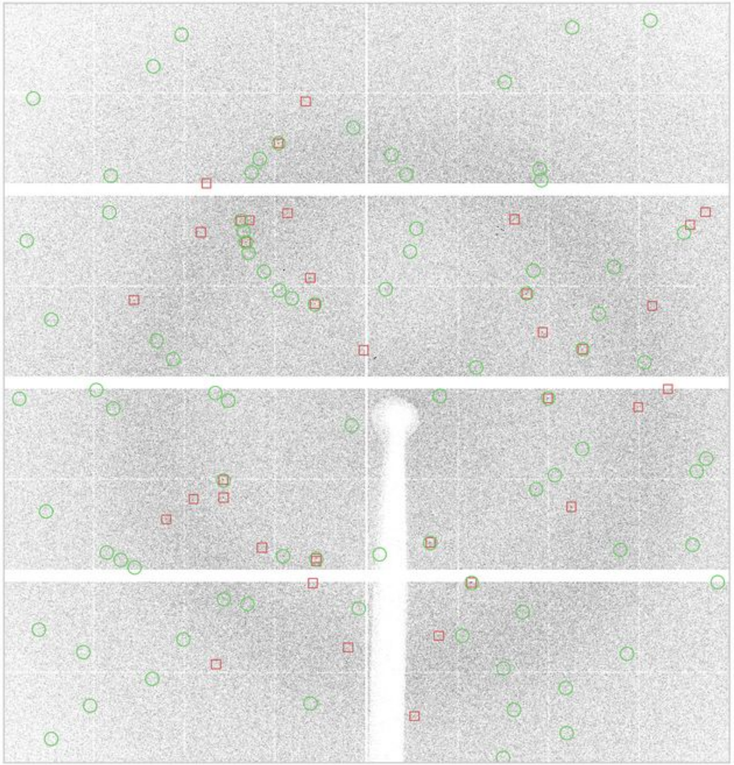


Electron density

It is important to arrange for **data processing capabilities that produce real time feedback**, in order to understand the characteristics of the experimental results. Full data analysis, on a time scale significantly shorter than the data collection, permits key indicators to be monitored so that experimental parameters can be adjusted before the available sample and allotted beam time are exhausted.

PAUL SCHERRER INSTITUT

What and why?



1. Real-time steering of experiment
2. Data reduction

Predicted lattice

Strong reflections

Less than 3% of the image

↓

Lossy/lossless compression scheme

Compression is tricky…

Garbage to gold: getting good results from bad data

By Tom Fleischman

July 26, 2018

A team led by physics professors Sol Gruner and Veit Elser began their recent research by seeking data other researchers had discarded as unusable.

Crazy, you say? To prove their idea was valid, the Cornell scientists needed data that was deemed too unclear – or "noisy" – to be used. The scientists who originally acquired the data were only able to use the best images – about 5 percent of the hundreds of thousands they collected – and threw the rest away. The Cornell group proved that these "garbage" images actually were golden.

https://news.cornell.edu/stories/2018/07/garbage-gold-getting-good-results-bad-data

Automated spot-finding

- Tedious search of the best parameters

```
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks10
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks10.stream
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks6
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks6.stream
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks7
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks7.stream
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks8
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks8.stream
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks9
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco1.mpeaks9.stream
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco2.mpeaks10
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco2.mpeaks10.stream
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco2.mpeaks6
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco2.mpeaks6.stream
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco2.mpeaks7
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco2.mpeaks7.stream
lyso_5_1000Hz_dtz170_data_000049.th10.snr3.0.mpixco2.mpeaks8
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco1.mpeaks7.stream
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco1.mpeaks8
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco1.mpeaks8.stream
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco1.mpeaks9
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco1.mpeaks9.stream
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks10
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks10.stream
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks6
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks6.stream
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks7
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks7.stream
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks8
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks8.stream
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks9
lyso_5_1000Hz_dtz170_data_000049.th13.snr3.5.mpixco2.mpeaks9.stream
lyso_5_1000Hz_dtz170_data_000049.th13.snr4.0.mpixco1.mpeaks10
lyso_5_1000Hz_dtz170_data_000049.th13.snr4.0.mpixco1.mpeaks10.stream
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco1.mpeaks10
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco1.mpeaks10.stream
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco1.mpeaks6
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco1.mpeaks7
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco1.mpeaks7.stream
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco1.mpeaks8
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco1.mpeaks8.stream
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco1.mpeaks9
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco1.mpeaks9.stream
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco2.mpeaks10
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco2.mpeaks10.stream
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco2.mpeaks6
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco2.mpeaks6.stream
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco2.mpeaks7
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco2.mpeaks7.stream
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco2.mpeaks8
lyso_5_1000Hz_dtz170_data_000049.th16.snr4.5.mpixco2.mpeaks8.stream
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco1.mpeaks7.stream
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco1.mpeaks8
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco1.mpeaks8.stream
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco1.mpeaks9
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco1.mpeaks9.stream
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks10
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks10.stream
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks6
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks6.stream
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks7
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks7.stream
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks8
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks8.stream
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks9
lyso_5_1000Hz_dtz170_data_000049.th6.snr5.0.mpixco2.mpeaks9.stream
lyso_5_1000Hz_dtz170_data_000049.th7.snr3.0.mpixco1.mpeaks10
lyso_5_1000Hz_dtz170_data_000049.th7.snr3.0.mpixco1.mpeaks10.stream
```

- Mostly offline processing, but faster online processing is required (JUNGFRAU 4 Mpixel at the full 2 kHz frame rate continuously produce 16.8 GB/s)
- Non trivial to detect spots in large/interesting proteins – SNR ~1.2/1.4 – 3% hits
- Masking problems: mask is defined manually

Automated spot-finding

Different approaches:

- **Local spot-finding**

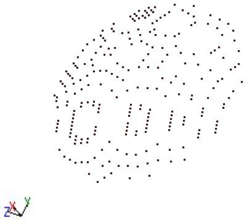
  The model works only in the surrounding of a pixel

Pro: simple models, fast to train, fast to execute, good in detecting strong signals

Cons: cannot capture long-range correlation within the full image, difficult to distinguish a Bragg reflection from any other strong pixel

- **Global spot-finding**

  The model works on the full image.



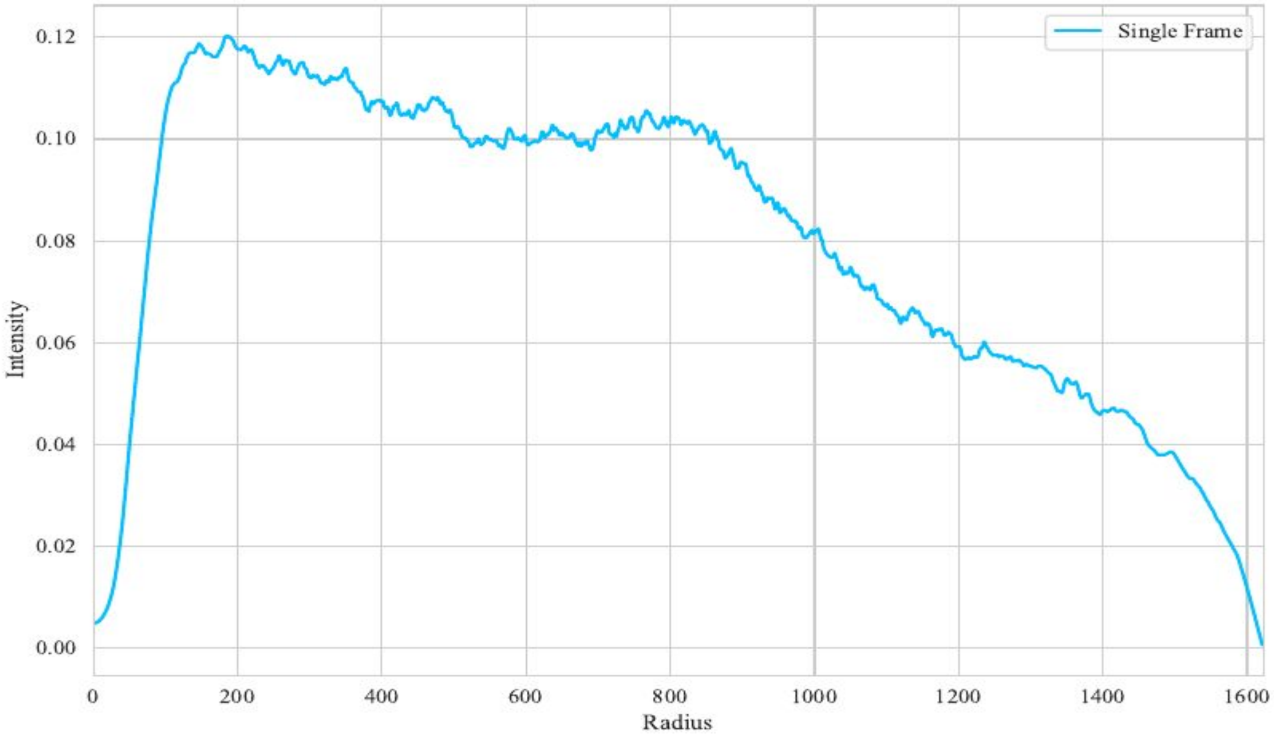
Pro: can capture long-range correlations, indexing can be implicitly taught to the model

Cons: slower execution, possible problems with input sizes, slow training

1. Background removal -> 2. Local spotfinding -> 3. Global spotfinding -> 4. Fast indexing

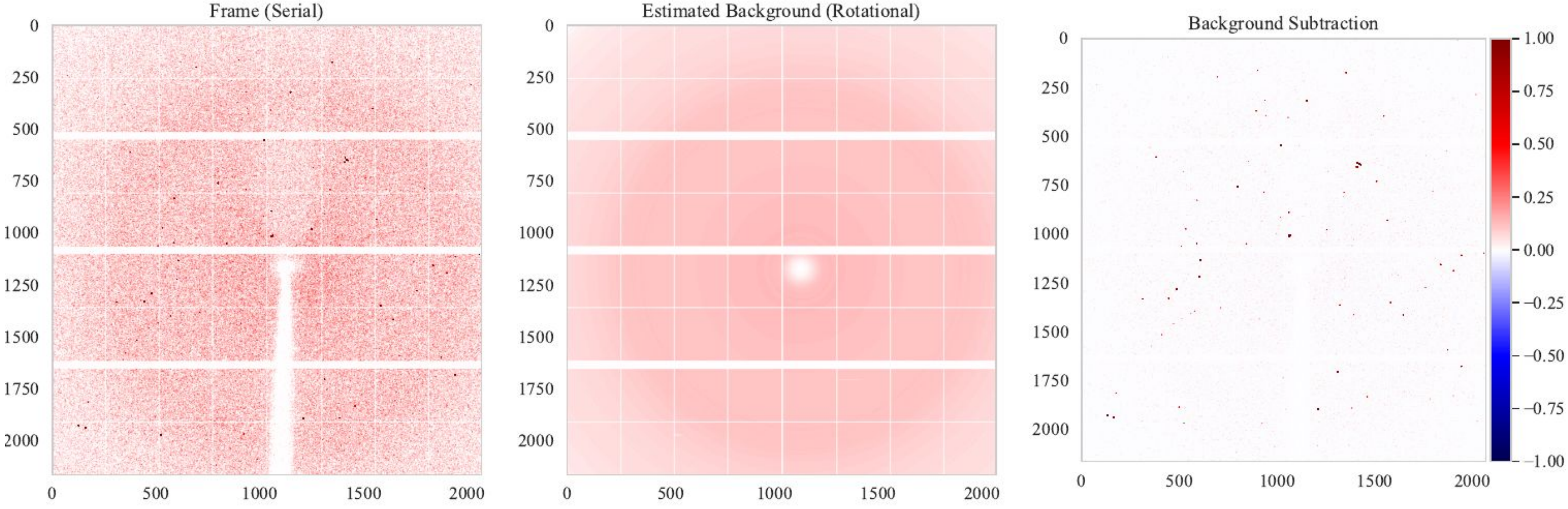
Automated spot-finding: background removal

1. Radial background Estimation



Automated spot-finding: background removal

2. Background subtraction



Automated spot-finding: local spotfinding

Methods:

Supervised: **LinearSVM, KernelSVM**, FFNN, **CNN**

Unsupervised: Dictionary learning

Uncertainty estimation: Ensemble modeling

Inputs:

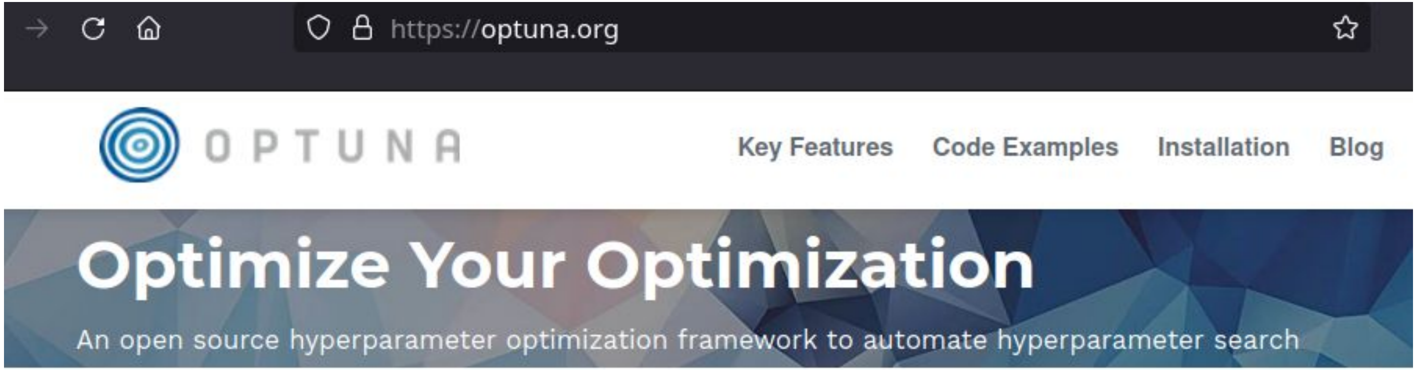
- Raw counts, Square root, Log counts, Gaussian Filter Ratio
- Different sizes (9x9, 21x21), Centered

Training:

Balanced/Unbalanced sets, Cross entropy, Focal loss, Dice loss, Weighted cross entropy

Automated spot-finding

- General vs Specialized models: one can try to have a single model to work well in all cases, or for each system we can retrain a new specialized model

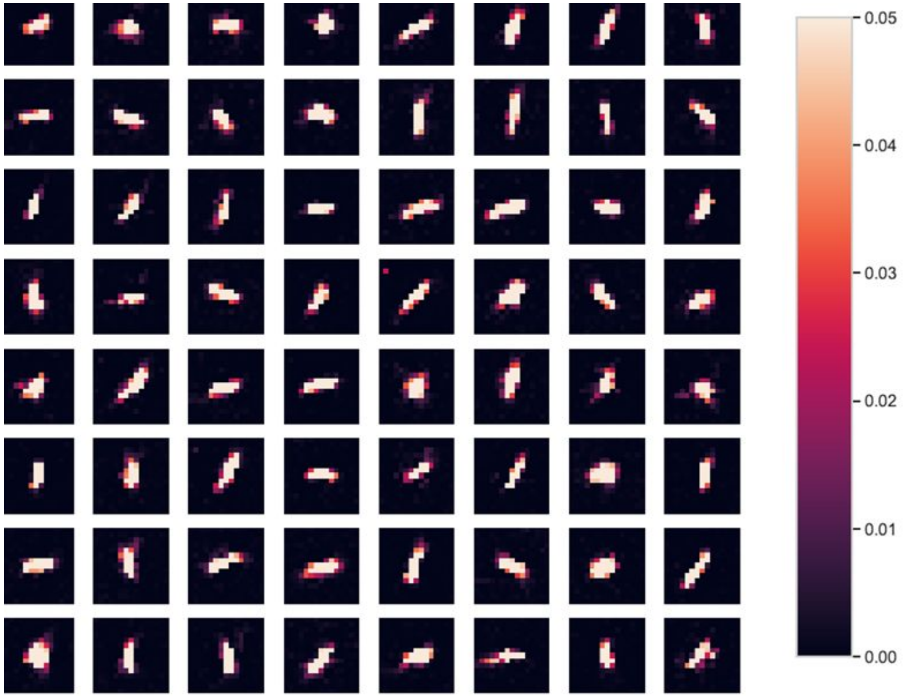


PAUL SCHERRER INSTITUT PSI

Automated spot-finding: supervision?

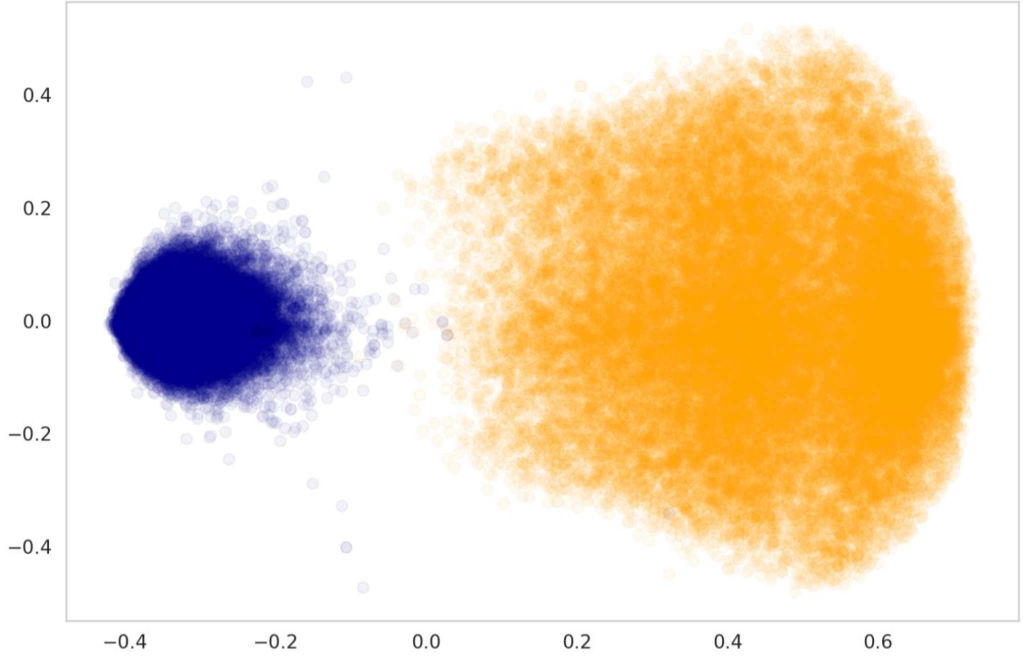
1. Initial labels provided by XDS, Crystfel or DIALS
2. Semi-supervised: label are learnt in unsupervised fashion using dictionary learning

Example of learnt templates



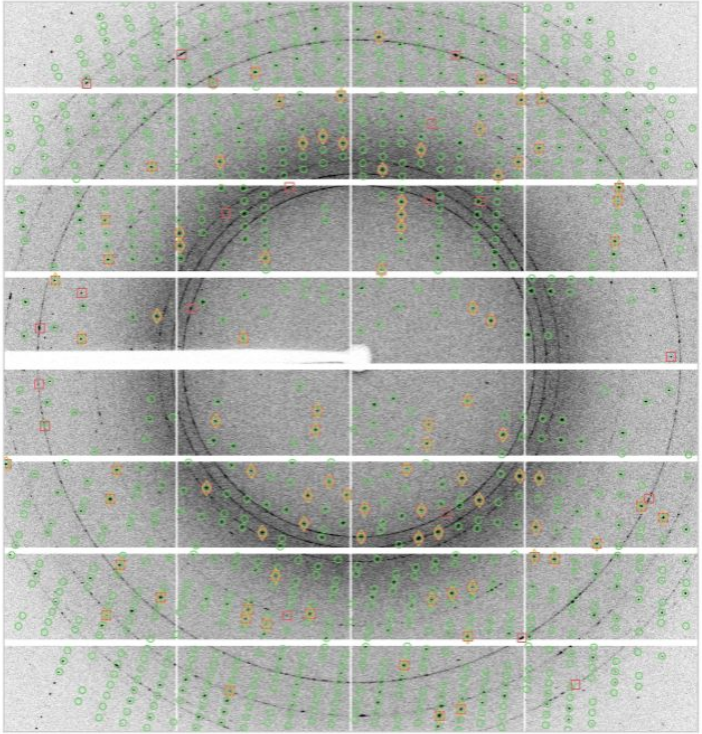
Automated spot-finding: binary or multiclass?

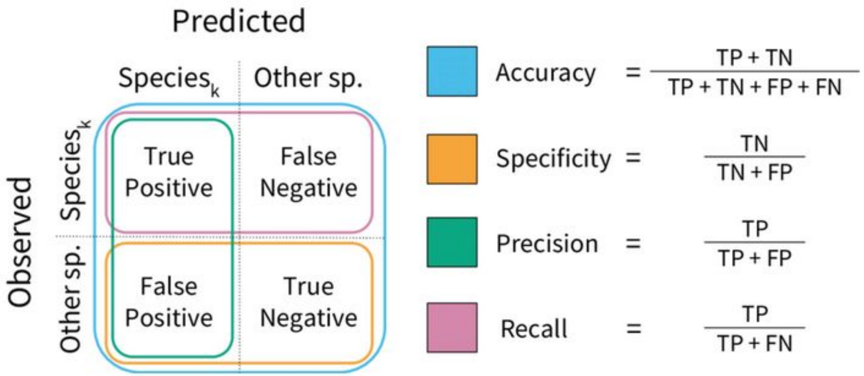
Finding strong reflections seems to be an easy task for ML. We can check this by using dimensionality reduction (KernelPCA) on the flattened image vector. Classes in the high-dimensional manifold are well-separated, meaning that also simple/fast linear models can capur strong signals



Automated spot-finding: performance metrics

We need a way to compare different labels

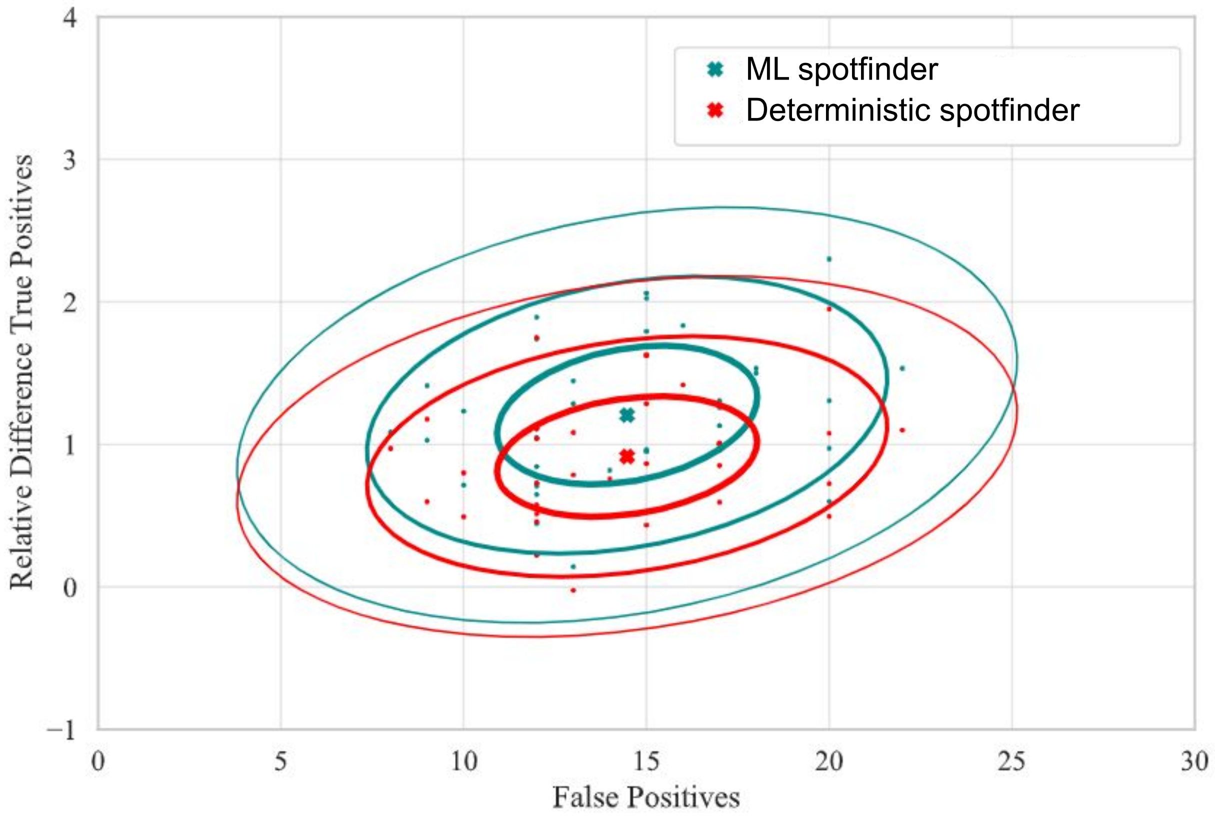




COLSPOT:

F1: 0.153 | Precision: 0.802 | Recall: 0.085 | FP: 21.000

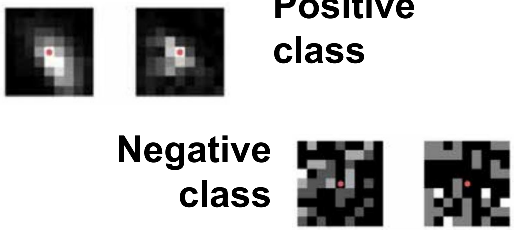
Automated spot-finding: performance metrics



Automated spot-finding: local & global

ONLINE
Fast & Small images // binary classifier

Local Spotfinder

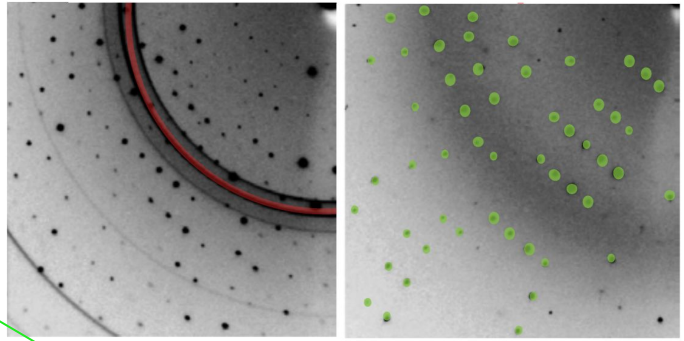


Training from experiments

Pixel-wise classification: spot vs no spot

ONLINE/OFFLINE
Slow & Large full patterns // multiclass

Global Spotfinder



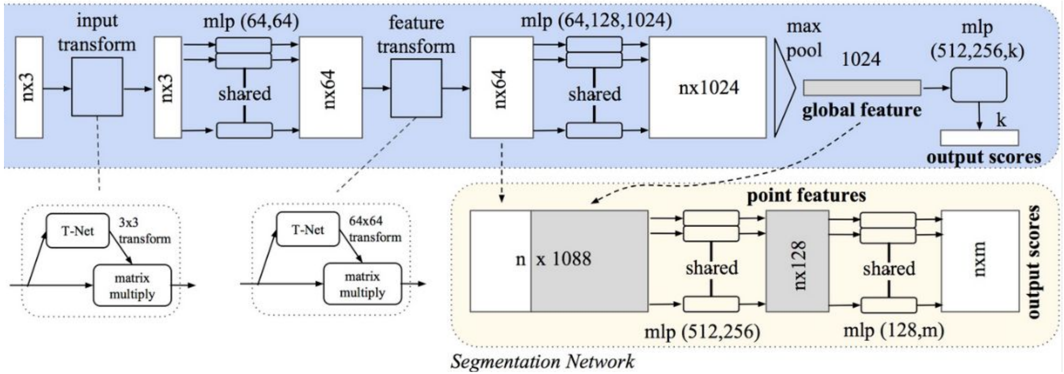
Training also from simulations

Separate & classify point clouds
(ice, defects, different crystals)

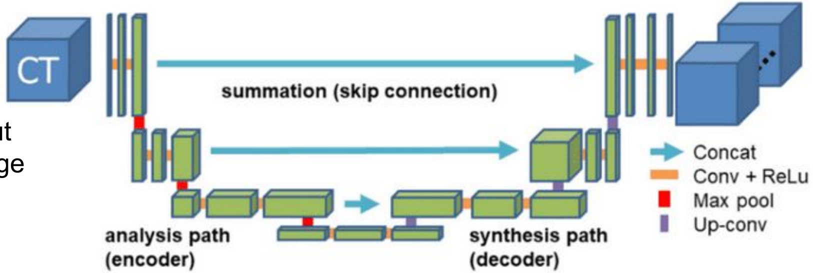
Automated spot-finding: global spot finding

Segmentation problem: split patterns that are indexable from those that are not.

- Segmentation of 3D point clouds in the reciprocal space – **PointNet** (https://arxiv.org/abs/1612.00593)



- Segmentation of point of patterns in 2D images – **UNet** (https://arxiv.org/abs/1612.00593)



Input image

One output for each class

Acknowledgements

My thanks go to

- Alun, Markus
- HC Stadler
- Filip, Greta (SLS)
- Benjamin, Luis Barba, Taulant (SDSC)