



Science and  
Technology  
Facilities Council

# CCP-EM

## 8th Icknield Workshop

Tom Burnley CCP-EM/STFC  
4th November 2024  
RAL/DLS



# What is CCP-EM?



Support users and developers in computational aspects of biological EM

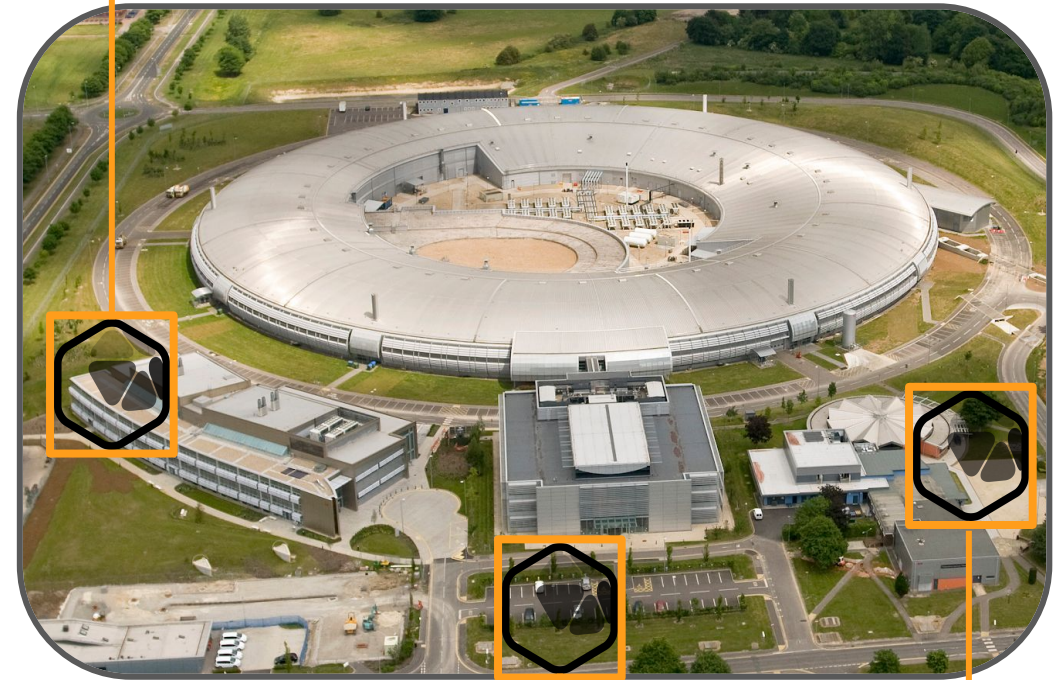
Based at STFC RAL national laboratory

CCP-EM software suite

- EM community (>3.5K subscribers)
- Software users (>1200 downloads, >30 industrial licences)
- Support developers (>10 external groups)

Core funding from MRC since 2012

CCP-EM & CCP4 | RCaH

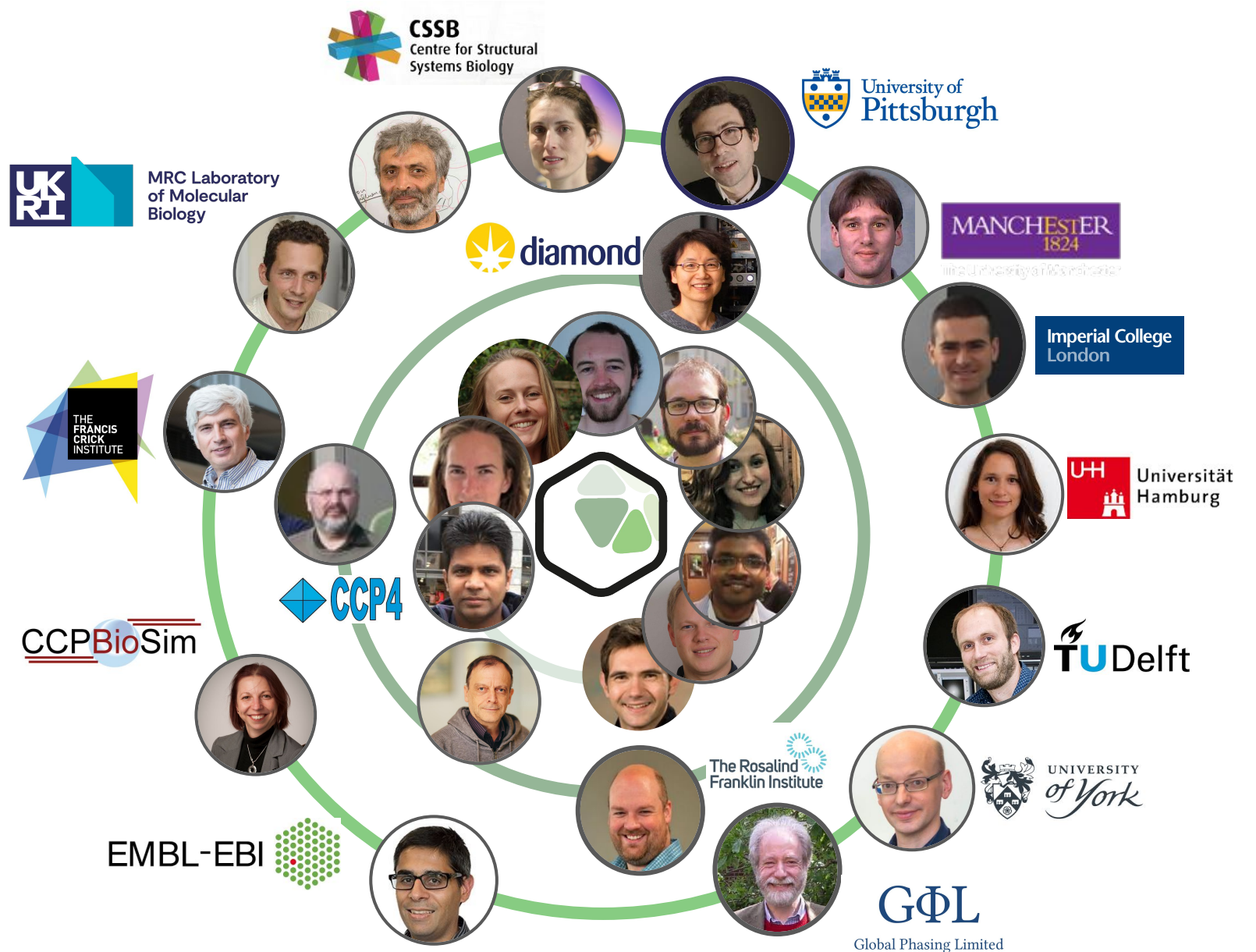


Franklin

eBIC | DLS

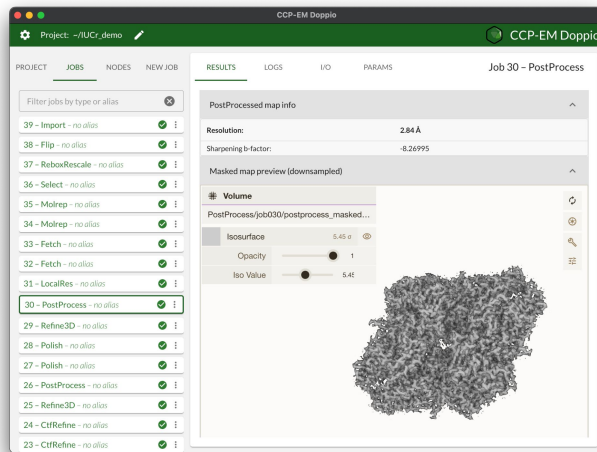


# Collaborative Computational Project





# Collaborative Computational Project



Tom  
Burnley



George  
Coldstream



Joel  
Greer



Lauren  
Giles



Matt  
Iadanza



Agnel  
Joseph



Sony  
Malhotra



Jola  
Mirecka



Colin  
Palmer



Rangana  
Warshamanage



Martyn  
Winn



# *Collaborative Computational Project*



**Yuriy  
Chaban**



**Lorna  
Malone**



**Karen  
Davies**



**Dan  
Hatton**



**Stephen  
Riggs**



# CCP-EM software suite

30+ tasks in a common Python framework

Raw micrographs to validated atomic model

Uses some CCP4 programs

Download from [ccpem.ac.uk](http://ccpem.ac.uk)

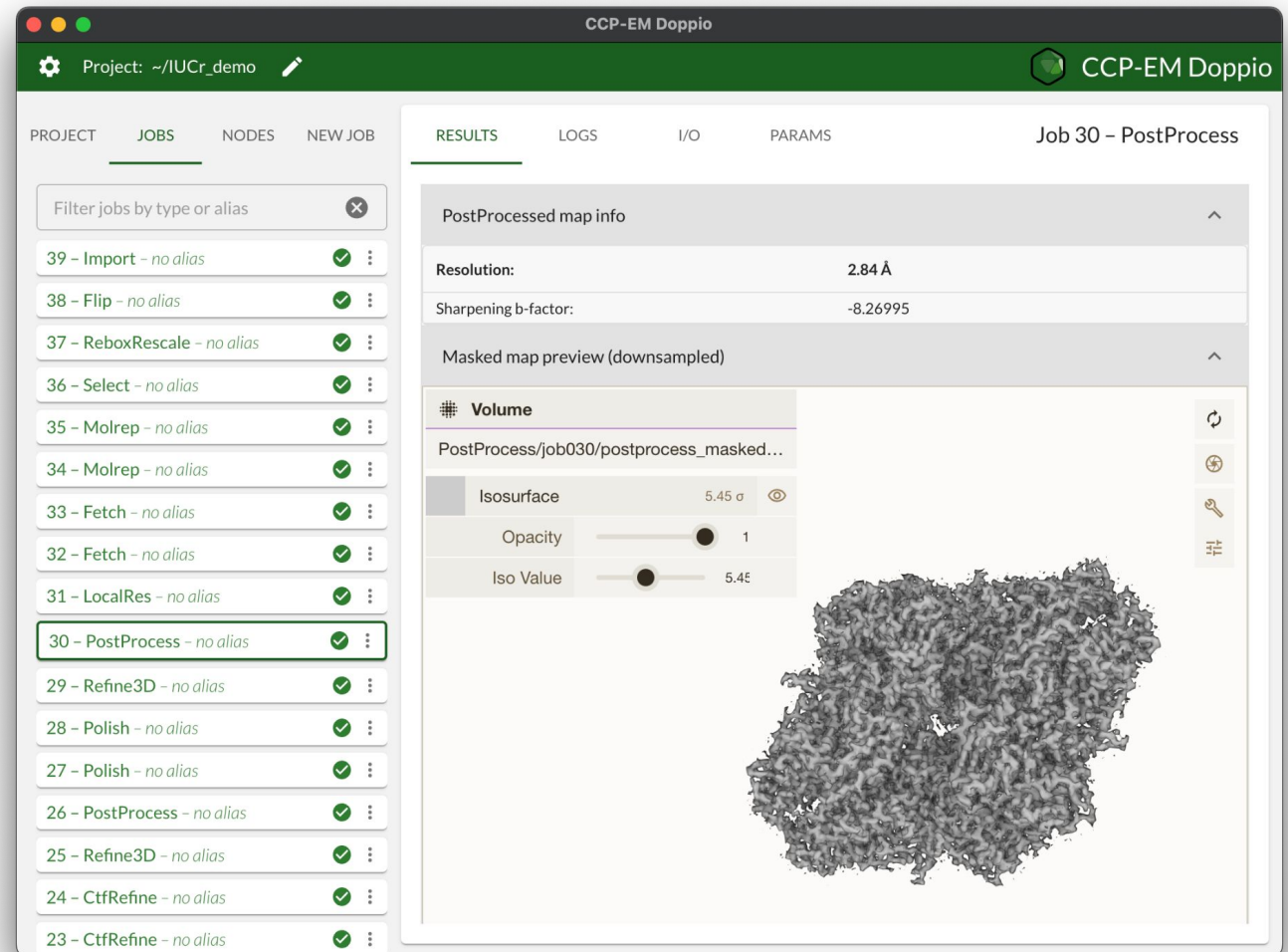
Linux & Mac

Free for academic use, fee for commercial

Bugs & requests:

[ccpem@stfc.ac.uk](mailto:ccpem@stfc.ac.uk)

Recommend using latest nightly



The screenshot displays the CCP-EM Doppio web interface. The top navigation bar includes 'PROJECT', 'JOBS', 'NODES', and 'NEW JOB'. The 'JOBS' tab is active, showing a list of 30 jobs. Job 30, 'PostProcess', is highlighted. The right-hand panel shows the 'RESULTS' for Job 30 - PostProcess, including 'PostProcessed map info' with a resolution of 2.84 Å and a sharpening b-factor of -8.26995. Below this is a 'Masked map preview (downsampled)' section showing a 3D volume rendering of a protein structure. The volume is displayed as a grey mesh, and the interface includes controls for 'Isosurface' (set to 5.45 σ) and 'Opacity' (set to 1). The 'Iso Value' is also set to 5.45. The interface is titled 'CCP-EM Doppio' and shows the project path as '~/UCr\_demo'.

# CCP-EM workflow at Icknield

Map  
Optimisation

Model  
Building

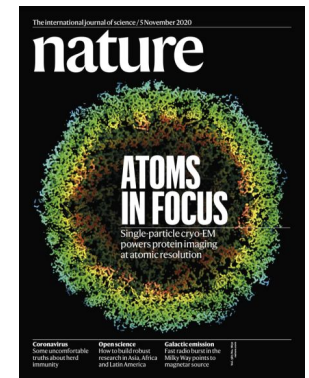
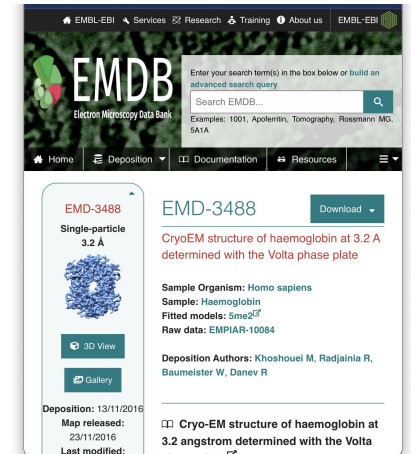
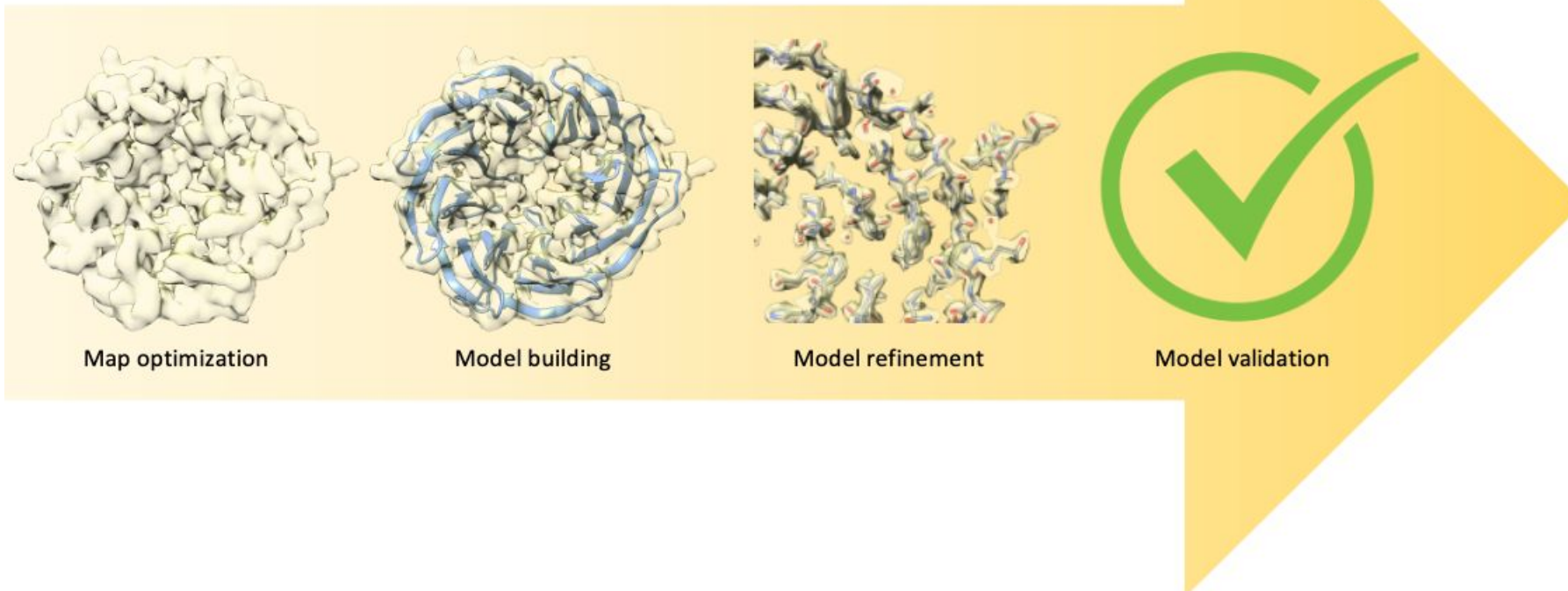
Model  
Docking

Automated  
Refinement

Interactive  
Refinement

Map/Model  
Validation

Deposition



# CCP-EM workflow at Icknield

## Map Optimisation

LocScale



## Model Building

ModelCraft



Model Angelo



## Model Docking

EM\_placement

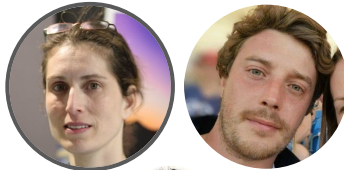


## Automated Refinement

Refmac Servalvat



TEMPy -REFF



## Interactive Refinement

Coot Moorhen



ChimeraX/ ISODLE



## Map/Model Validation

CCP-EM Validation



Find/Check MySeq



Privateer



## Deposition

EMDB



AF-DB & 3DBeacons



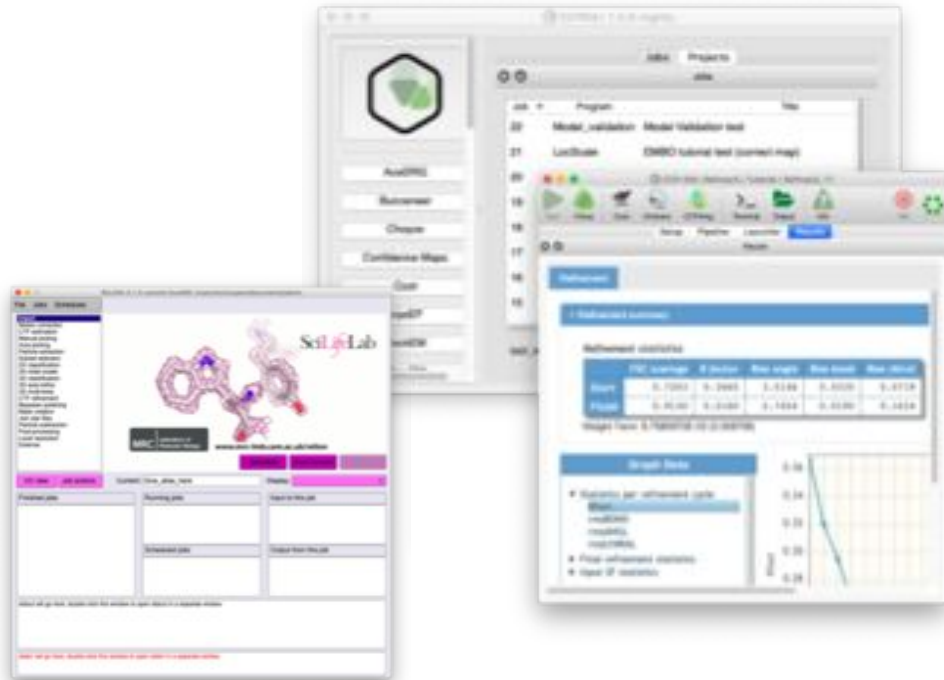
sharpen... build... refine... validate...  
repeat...



# CCP-EM 2.0: pipeliner & Doppio



# CCP-EM version 1.x



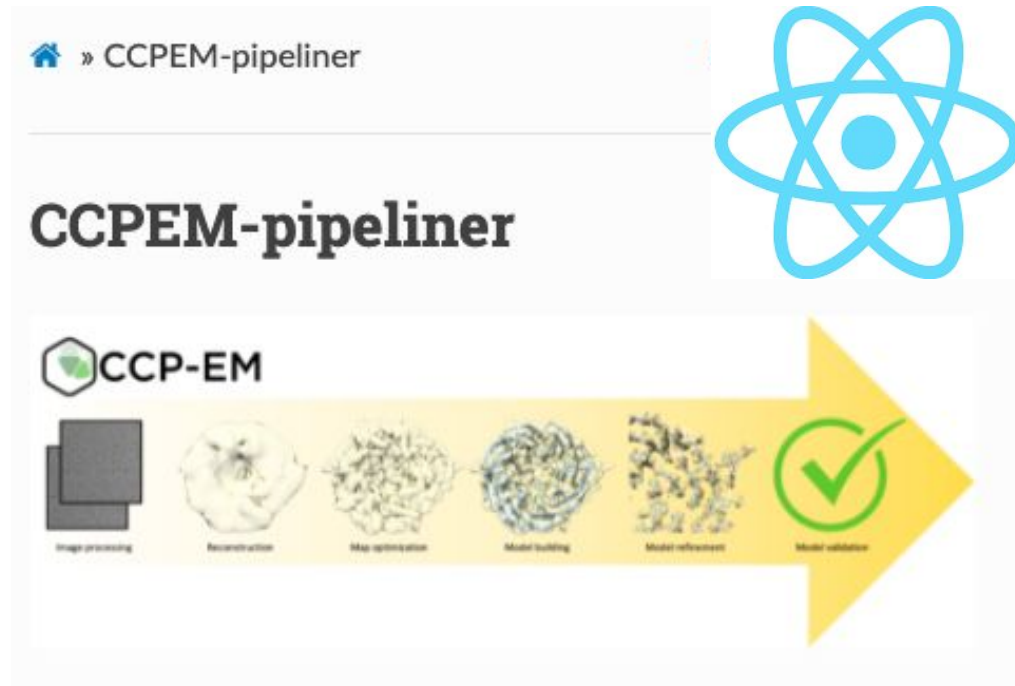
## **CCP-EM**

- Map post processing
- Atomic modelling
- PyQt5 GUI
- No data model
- Basic project management

## **RELION**

- Single particle reconstruction
- FLTK GUI
- Data model
- DAG project management

# CCP-EM version 2.x



## *Images to structures*

- Python pipeliner python API
- JS react GUI
  
- Updated data model
- Project management
  
- Single particle
- Atomic modelling
- Tomography (coming soon)

# CCP-EM software plan

GUI



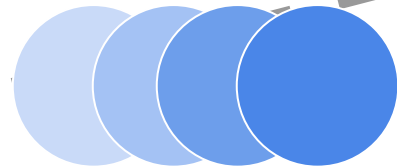
CCP-EM Doppio

Manager



CCP-EM Pipeliner

Tasks



Relion Refine



LocScale



Flex-EM



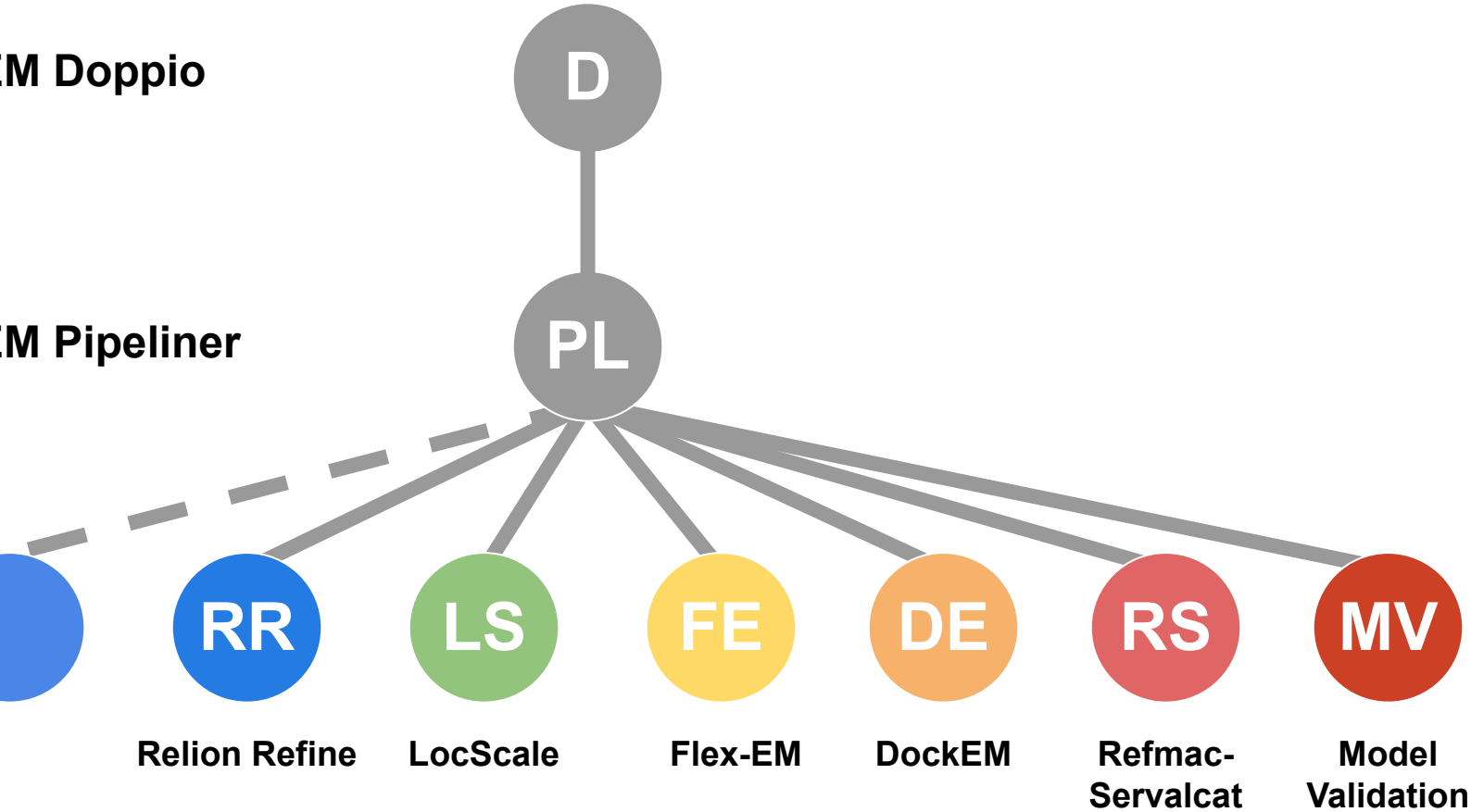
DockEM



Refmac-Servalcat

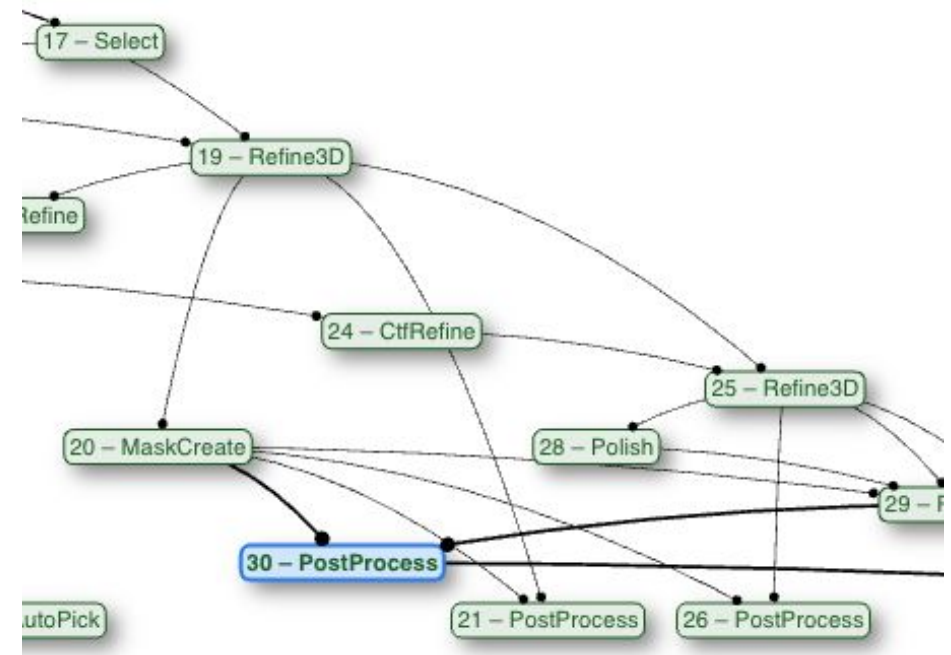
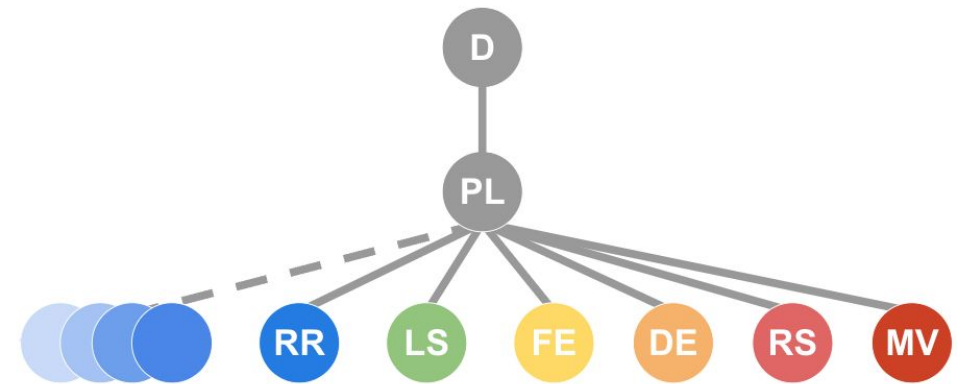


Model Validation



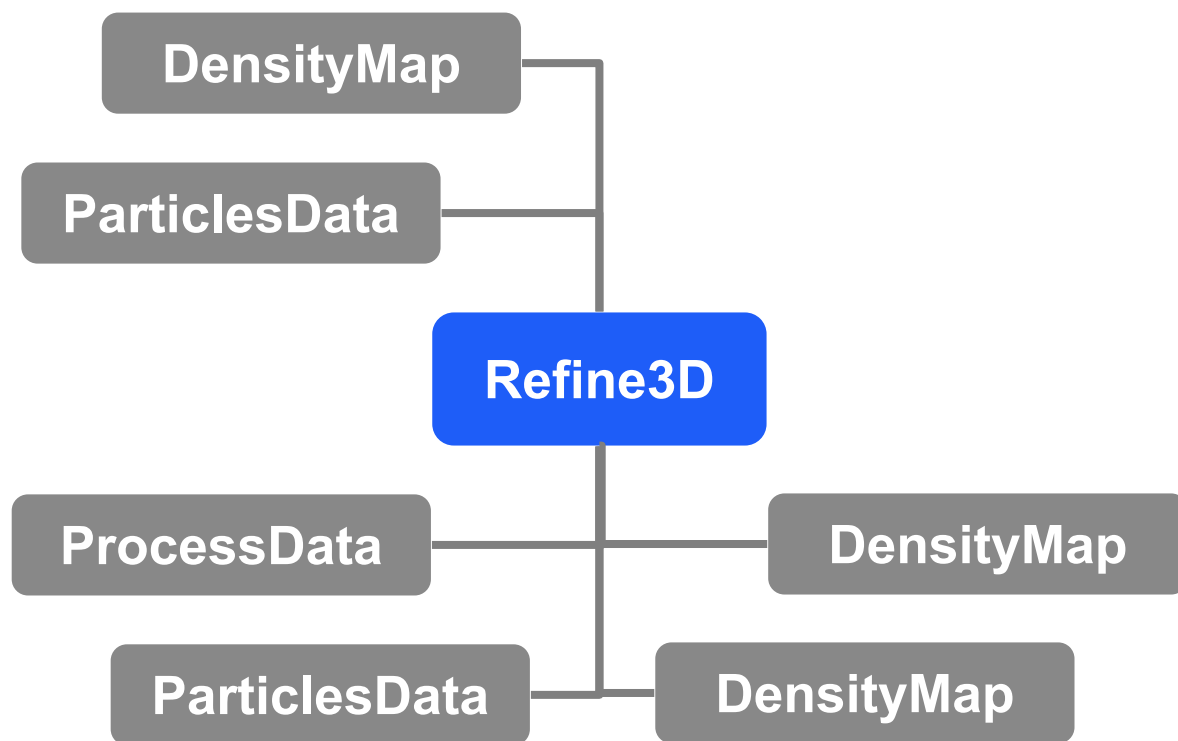
# ccpem-pipeliner

- **Business logic layer decoupled from tasks and UI**
- Python 3
- MPL 2.0 license
- <https://gitlab.com/ccpem/ccpem-pipeliner>
- Directly Acyclic Graph dataflow
  - Metadata tracking
  - Archiving
  - Jobs have input/output nodes



# Job Nodes

*Each job has input and output nodes & nodes are specific data types*



CCP-EM Doppio

RESULTS LOGS I/O PARAMS Job 25 - Refine3D

Open with: CHIMERA X COOT PDF VIEWER RELION DISPLAY TEXT EDITOR UCSF CHIMERA

Inputs to this job:

- CtfRefine/job024/particles\_ctf\_refine.star ParticlesData relion ctfrefine
- Refine3D/job019/run\_class001.mrc DensityMap relion refine3d

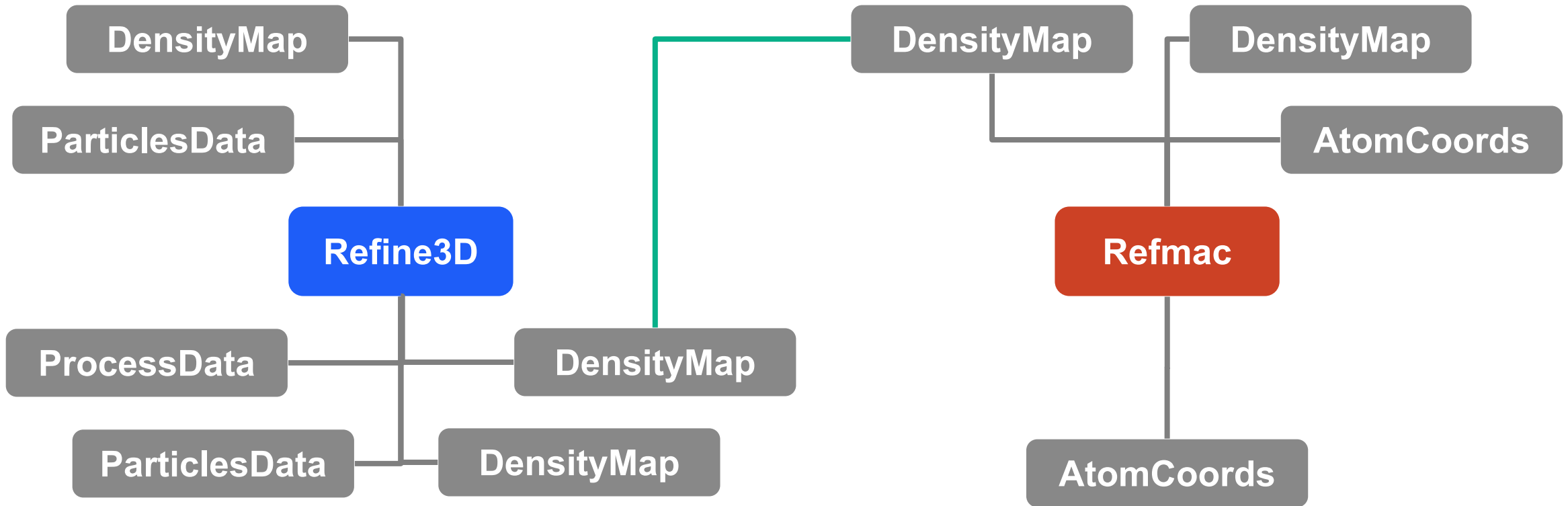
Outputs from this job:

- Refine3D/job025/run\_data.star ParticlesData relion refine3d
- Refine3D/job025/run\_optimiser.star ProcessData relion optimiser refine3d
- Refine3D/job025/run\_half1\_class001\_unfil.mrc DensityMap relion halfmap refine3d
- Refine3D/job025/run\_class001.mrc DensityMap relion refine3d



# Job Nodes

*Each job has input and output nodes & nodes are specific data types*



# Automating complex workflows



- **CLI**

```
#!/user/bin/bash
pipeliner --start_new_project
pipeliner --schedule_job Import_job.star
pipeliner --schedule_job MotionCorr_job.star
pipeliner --schedule_job CtfFind_job.star

pipeliner --run_schedule --name Schedule1 --jobs Import/job001/ job002 CtfFind/new_alias --min_between 15
--nr_repeats 3 --min_wait_before 2 -sec_wait_after 15

pipeliner --metadata_report job003
```

- **Python API**

```
from pipeliner.api.manage_project import PipelinerProject

my_project = PipelinerProject()

my_project.schedule_job("Import_job.star") # adds Import/job001/ to the pipeline
my_project.schedule_job("MotionCorr_job.star") # adds MotionCorr/job002/ to the pipeline
my_project.schedule_job("CtfFind_job.star") # adds AutoPick/job003/ to the pipeline

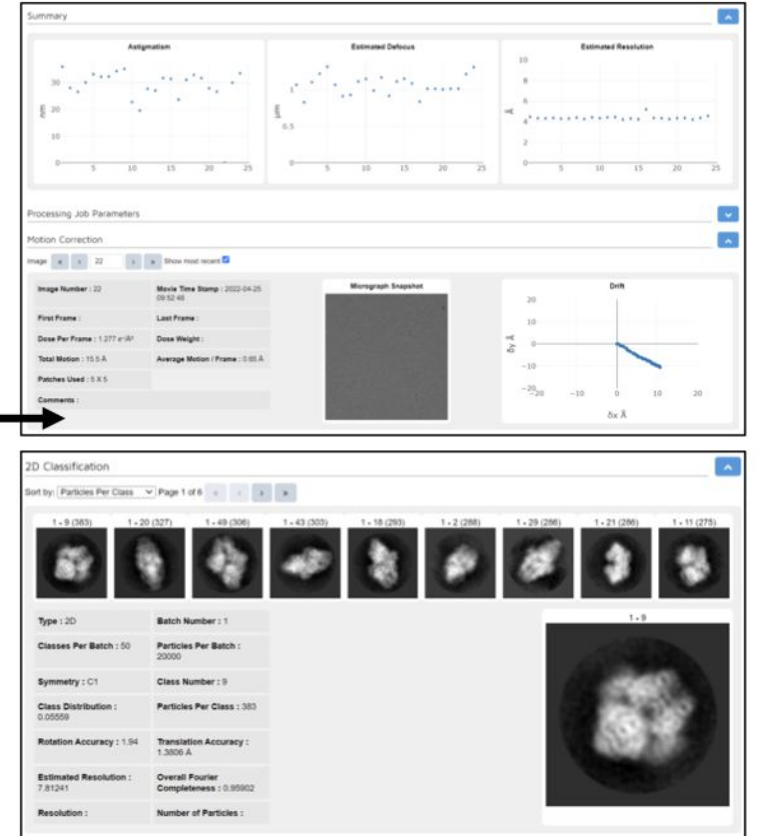
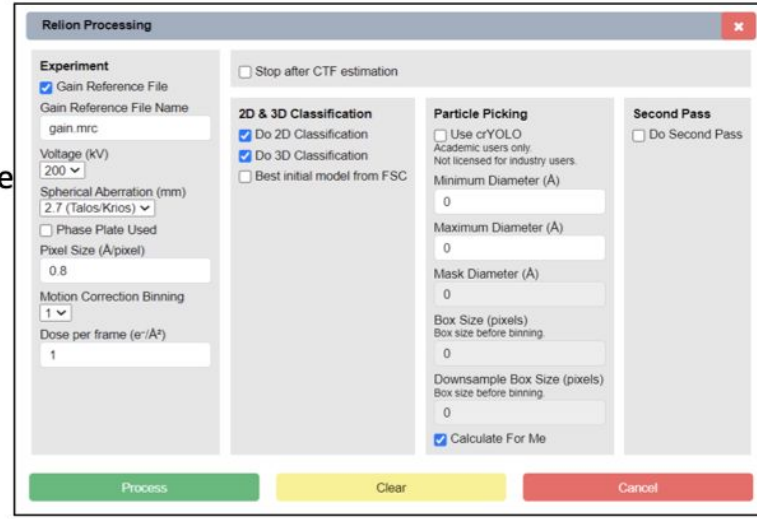
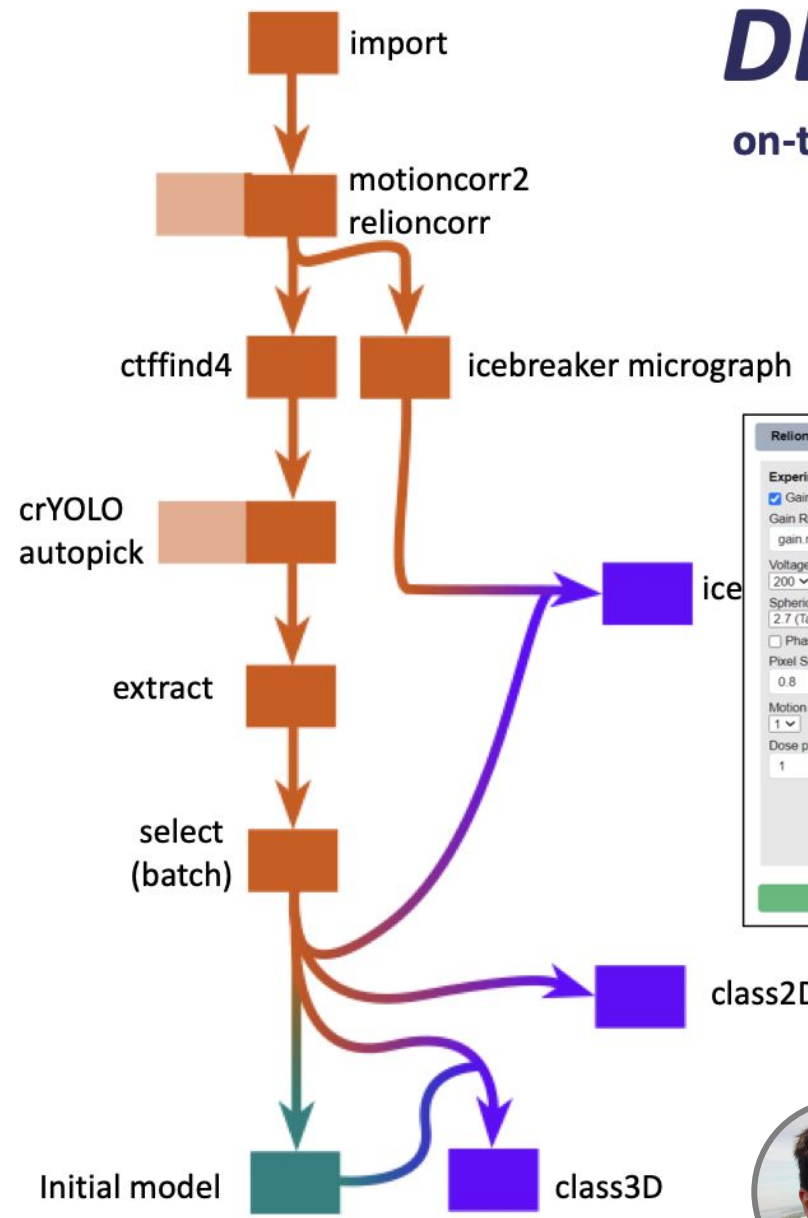
my_project.run_schedule(
    name="Schedule1",
    job_ids=["Import/job001/", "MotionCorr/job002/", "CtfFind/job003/"],
    nr_repeat=3,
    minutes_wait=15,
    minutes_wait_before=2,
    seconds_wait_after=15,
)
my_project.get_network_metadata("CtfFind/job003/", "my_metadata.json") # returns metadata
# and all upstream jobs
```

# DLS/eBIC Processing

on-the-fly batched preprocessing and evaluation



ISPyB



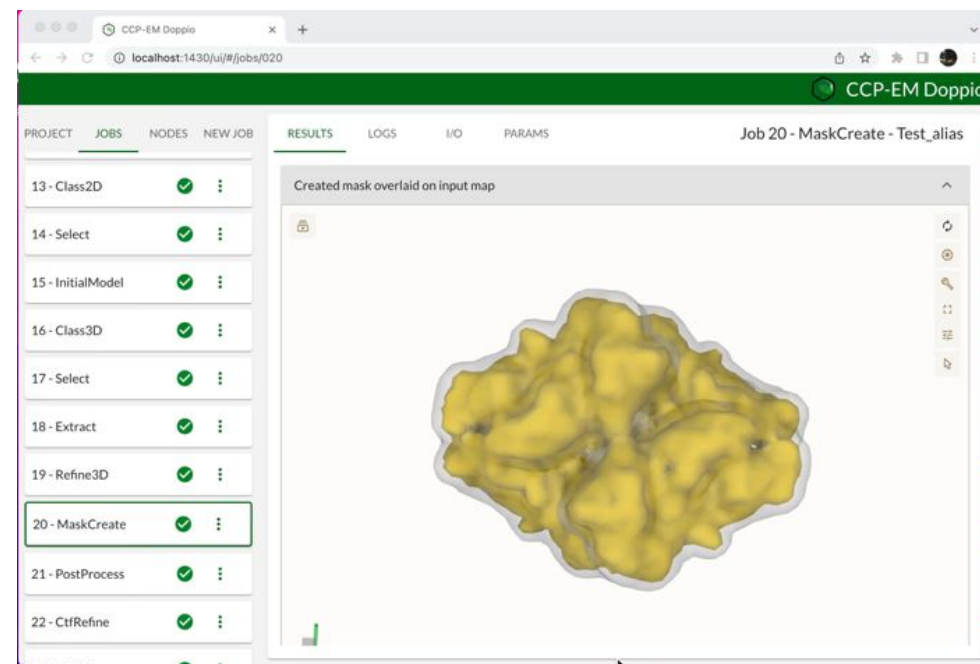
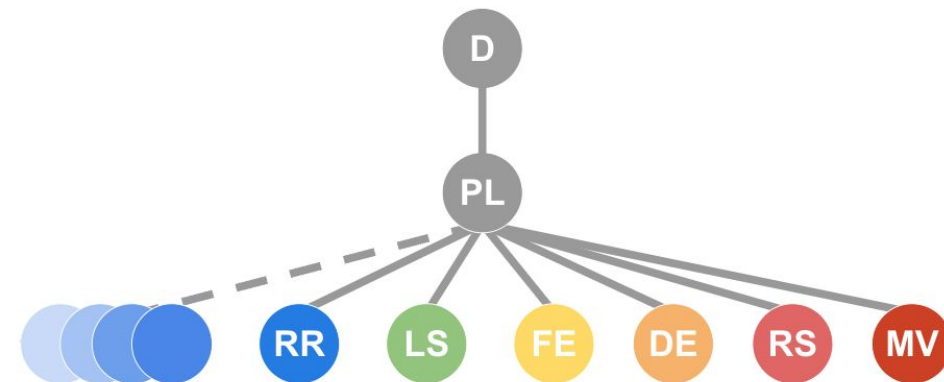
Dan Hatton Diamond Light Source



# CCP-EM *Doppio*



- **JavaScript UI**
- React.js / FastAPI / Material UI
- Remote browser app or local Electron app
- Task UIs auto generated from Python plugins
- Full support for **Image Processing** through to **Model Validation**
- <https://gitlab.com/ccpem/doppio>
- STFC dual license
- Funding: BEIS / STFC BID / CCP-EM



# Doppio demo



CCP-EM Doppio

Project: ~/IUCr\_demo

PROJECT JOBS NODES NEW JOB

Filter jobs by type or alias

- 39 - Import - no alias ✓
- 38 - Flip - no alias ✓
- 37 - ReboxRescale - no alias ✓
- 36 - Select - no alias ✓
- 35 - Molrep - no alias ✓
- 34 - Molrep - no alias ✓
- 33 - Fetch - no alias ✓
- 32 - Fetch - no alias ✓
- 31 - LocalRes - no alias ✓
- 30 - PostProcess - no alias ✓**
- 29 - Refine3D - no alias ✓
- 28 - Polish - no alias ✓
- 27 - Polish - no alias ✓
- 26 - PostProcess - no alias ✓
- 25 - Refine3D - no alias ✓
- 24 - CtfRefine - no alias ✓
- 23 - CtfRefine - no alias ✓

RESULTS LOGS I/O PARAMS Job 30 - PostProcess

PostProcessed map info

Resolution: 2.84 Å

Sharpening b-factor: -8.26995

Masked map preview (downsampled)

# Volume

PostProcess/job030/postprocess\_masked...

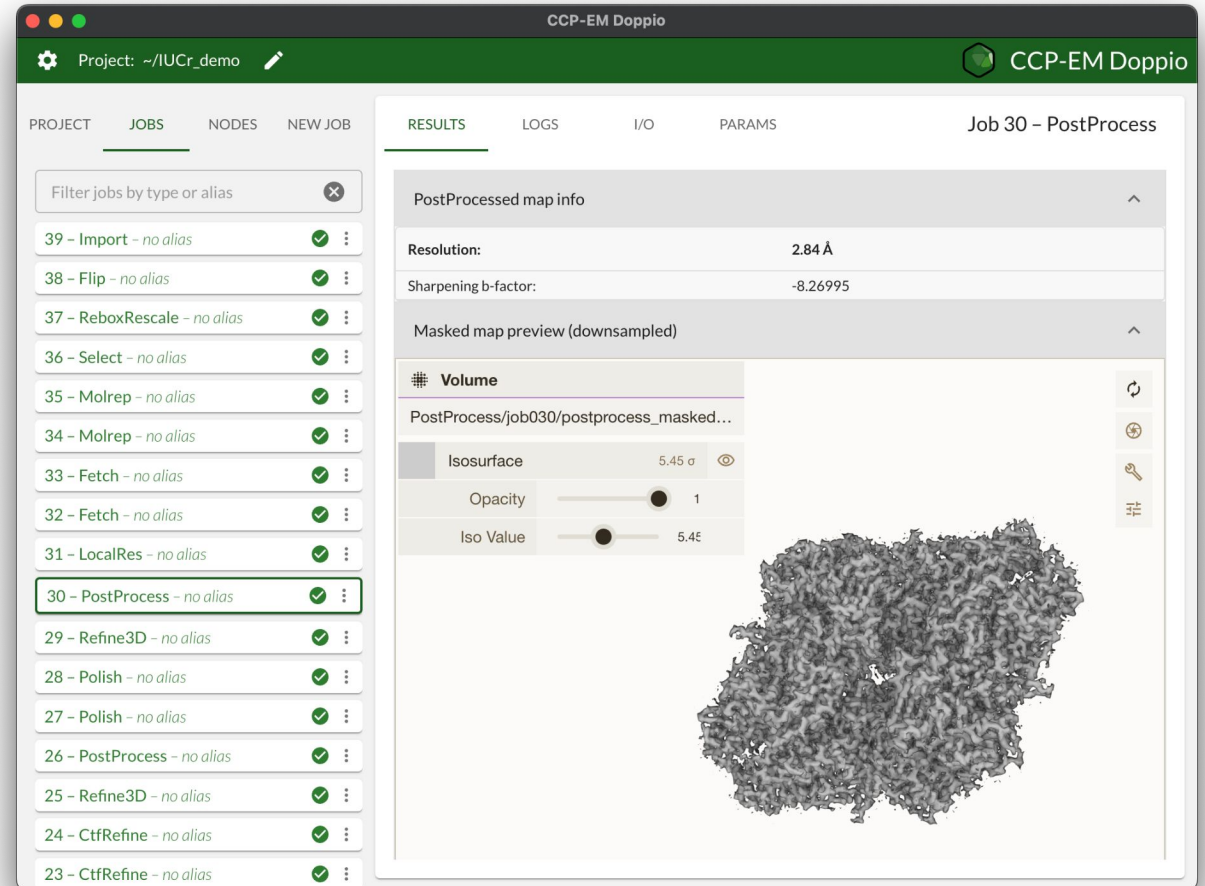
Isosurface 5.45  $\sigma$

Opacity 1

Iso Value 5.4 $\epsilon$

# CCP-EM Doppio

- **Doppio 1.1.0**
  - Linux and Mac
  - Requires CCP-EM 1.x
  - Optionally CCP4 9.x & Relion 5.0
- **Future plans:**
  - Tomography STA Relion 5 pipeline
  - Conda based installation
  - Automated metadata deposition



# Kaizen



Kaizen (Japanese: 改善, "improvement") is a concept referring to business activities that continuously improve all functions and involve all employees.

CCP-EM Core Team

CCP4 Core Team

STFC Business & Innovation

CCP-EM Commercial License Holders

CCP-EM Collaborators

CCP-EM Users

Website: [www.ccpem.ac.uk](http://www.ccpem.ac.uk)

Mailing list: [www.jiscmail.ac.uk/ccpem](http://www.jiscmail.ac.uk/ccpem)

Twitter: @ccp\_em

Email: [ccpem@stfc.ac.uk](mailto:ccpem@stfc.ac.uk)

