

HAPPy

Heavy Atom Phasing in Python



Dan Rolfe, Paul Emsley, Charles Ballard
Maria Turkenburg, Eleanor Dodson

What is it?

- 😊 New automated experimental phasing pipeline.
- 😊 Replace and expand on the capabilities of Paul Emsley's Chart package.

It will...

- 😊 Take integrated and merged experimental data amplitudes (post-TRUNCATE), de-twinned, consistently indexed.
- 😊 Determine the heavy atom structure and phase probabilities.
- 😊 Optimize the density map to give interpretable map.
- 😊 Build structure.
- 😊 First release will handle SAD data only. MAD, MIR, MIRAS modes later.

Implementation

- 😊 Written in Python. Uses EMMA from cctbx, Driver from XIA.
- 😊 Employ existing packages for the various stages of the structure solution.
- 😊 Use CCP4 programs where possible.
- 😊 Modular. Different modules for HA search, phasing, density modification/phase improvement will be interchangeable.
- 😊 SHELXD for heavy atom substructure, MLPHARE, PHASER, BP3 for SAD phasing, PIRATE for phase improvement. Will add DM module.
- 😊 BUCCANEER and COOT for model building in the future.
- 😊 Designed to cooperate with other automation packages e.g. take output from automated data processing software DNA/XIA-DP (Graeme Winter).
- 😊 Use well-defined APIs and data formats where data exchange is necessary.

- 😊 Will use and is contributing to the next generation CCP4(i) database (Peter Briggs, Wanjuan (Wendy) Yang).
- 😊 Working on performance testing suite, including reference structure comparisons for test data sets using Clipper utilities. Enables tracking of HAPPy performance over time.

What it already does

Input (XML description, MTZs)

Prepare and analyse
Resolution limits, anomalous difference limits

For each space group

For each limits set

Find heavy atom sites (SHELXD)

Select best substructure (SHELXD CC, ABS)

Try to identify correct substructure hand (ABS)

Using each phasing program (MLPHARE, PHASER, BP3)

For each remaining substructure hand

Determine initial phases, refine substructure

Try to choose correct substructure hand
Use map and refined substructure score

For each remaining substructure hand

Improve phases with PIRATE

Try to choose correct substructure hand
Use improved map and PIRATE statistics

Choose best final map
Use map and PIRATE statistics

What's still needed for SAD release (not exhaustive)

- ☺ More rigorous substructure determination.
More intelligent use of SHELXD - run it for longer when needed .
- ☺ Better logic for which phasing programs to use when.
e.g Try quick solution (e.g. MLPHARE).
Try slower more sophisticated method (e.g. PHASER) if MLPHARE fails.
- ☺ Improved and new scoring of phased data and maps.
For hand selection (CFFT not helping, trying ABS, PIRATE).
For deciding if we've got a good enough final result (trying PIRATE)
Use connectivity, skeletonizability, DM statistics
- ☺ More intelligent choice of resolution limits?
- ☺ More intelligent values/defaults, e.g. min HA separation depends on the atom.
- ☺ Lots of testing going on now will lead to further tuning and improvements..

Interface

Currently:

😊 Command line from XML input and MTZs.

Before release:

😊 CCP4i interface.

Will also:

😊 Start directly from processing pipeline e.g. DNA/XIA through XML input description and MTZs.

Data formats (so far)

- ☺ Input MTZs and XML, PIR/FASTA sequences...
- ☺ Output:
 - ☺ Reflection files: MTZ.
 - ☺ Substructures: XML, PDB.
 - ☺ Sequences: PIR.
 - ☺ HAPPy data, e.g. history, knowledge, phased result metadata: XML.
 - ☺ Whatever is necessary to talk to other software...
- ☺ HAPPy XML formats:
 - ☺ Reflect HAPPy objects, make HAPPy easy to develop and expand.
 - ☺ Informed by and designed to easily translate into related structure solution file formats and database schemas.

XML input file

```
<?xml version='1.0'?>
<!DOCTYPE project SYSTEM
"http://www.ccp4.ac.uk/HAPPY/xml/schemas/project.dtd">

<project name="gilu" phasing_mode="SAD" date_created="2005-09-21"
        created_by="Maria Turkenburg's own fair hands">

  <target>
    <consider_space_group_list>
      <space_group>I 2 2 2</space_group>
    </consider_space_group_list>
    <monomers>
      <monomer name="8xia">
        <sequence>
          MNYQTPEDR FTFGLWTVGW QGRDPFGDAT RRALDPVESV QRLAELGAHG VTFHDDDLIP
          FGSSDSEREE HVKRFRQALD DTGMKVPMAT TNLFTHPVFK DGGFTANDRD VRRYALRKTI
          RNIDLAVELG AETYVAWGGR EGAESGGAKD VRDALDRMKE AFDLLGEYVT SQGYDIRFAI
          EPKPNEPRGD ILLPTVGHAL AFIERLERPE LYGVNPEVGH EQMAGLNFPH GIAQALWAGK
          LFHIDLNGQN GIKYDQDLRF GAGDLRAAFW LVDLLESAGY SGPRHFDFKP PRTEDFDGVW
          ASAAGCMRNY LILKERAFAF RADPEVQEAL RASRLDELAR PTAADGLQAL LDDRSAFEEF
          DVDAAAARGM AFERLDQLAM DHLLGARG
        </sequence>
      </monomer>
    </monomers>
  </target>
</project>
```

```
</target>

<native name="8xia">
  <obs type="non-anom" file="sfdata/gilu.mtz">
    <columns type="F" F="F_GILU" SIGF="SIGF_GILU"/>
  </obs>
</native>

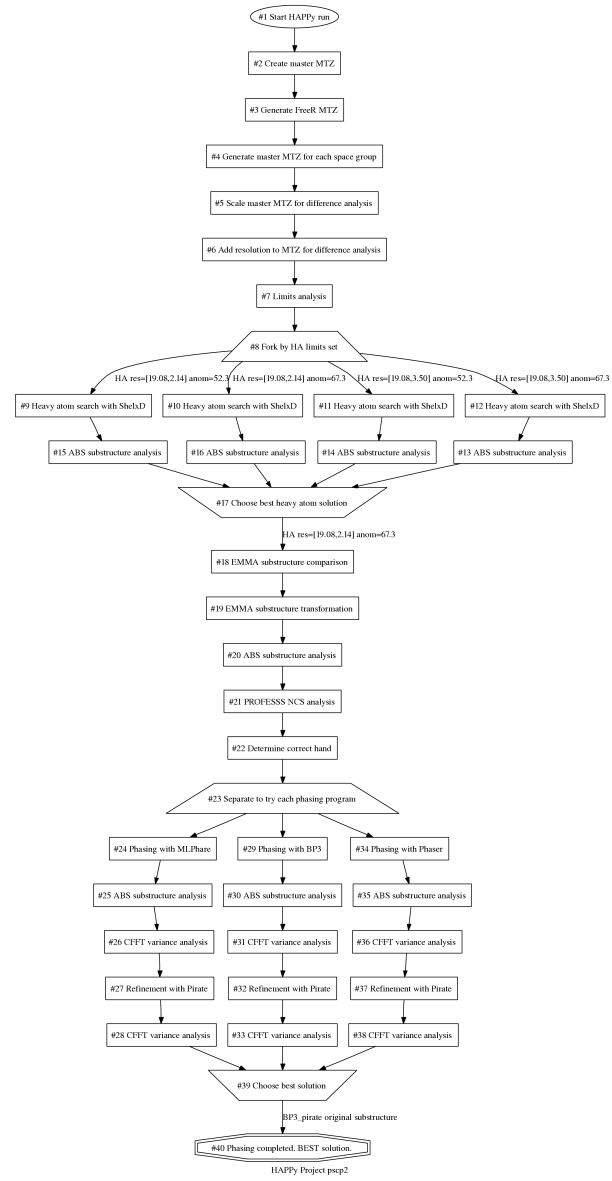
<derivative name="Mn" atom="Mn">
  <estimated_nsites>1</estimated_nsites>
  <obs type="peak" file="sfdata/gilu.mtz">
    <columns type="AN0" Fplus="F_GILU(+)" SIGFplus="SIGF_GILU(+)"
      Fminus="F_GILU(-)" SIGFminus="SIGF_GILU(-)"/>
  </obs>
</derivative>

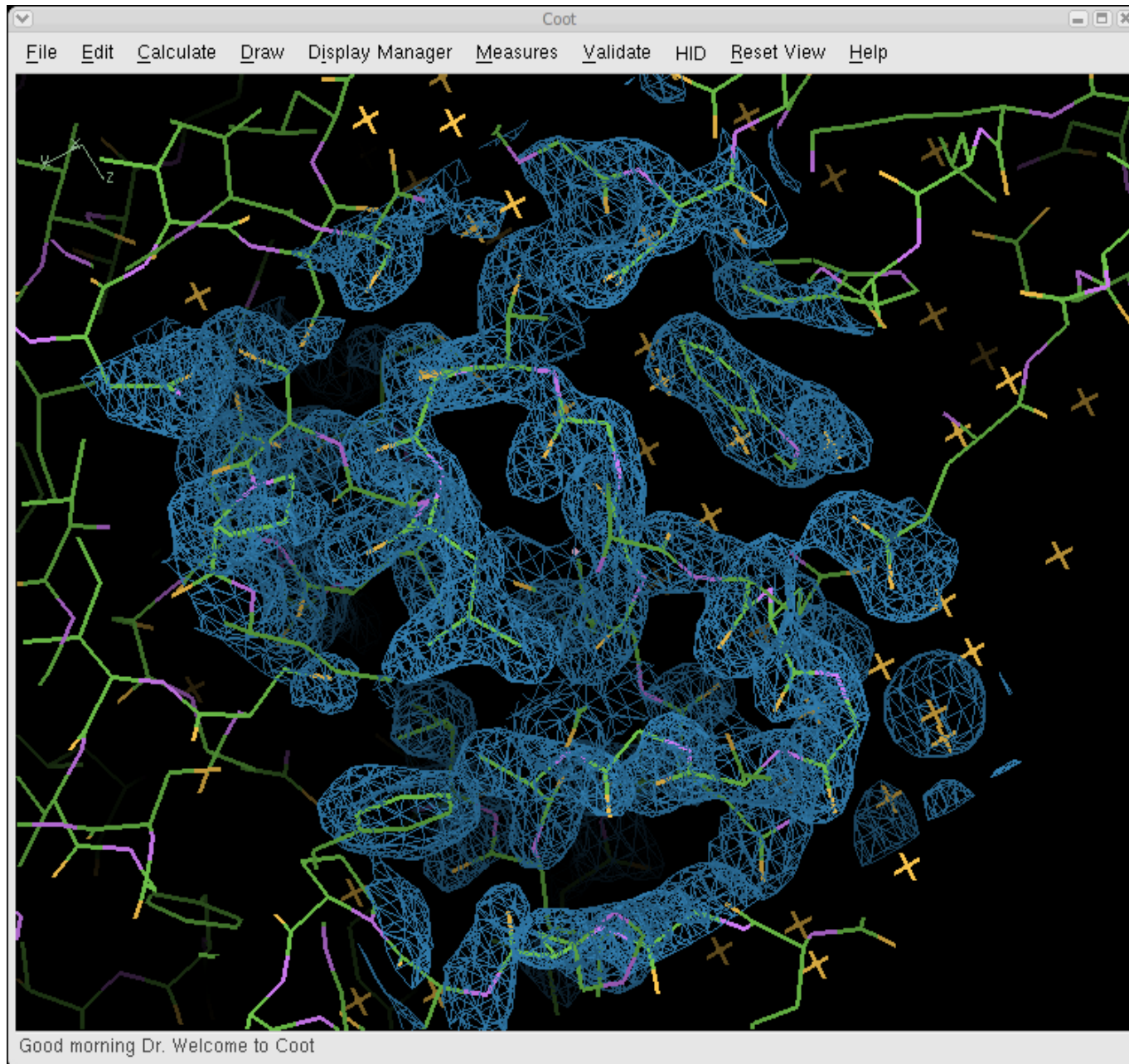
<freer file="sfdata/gilu.mtz" label="FreeR_flag"/>

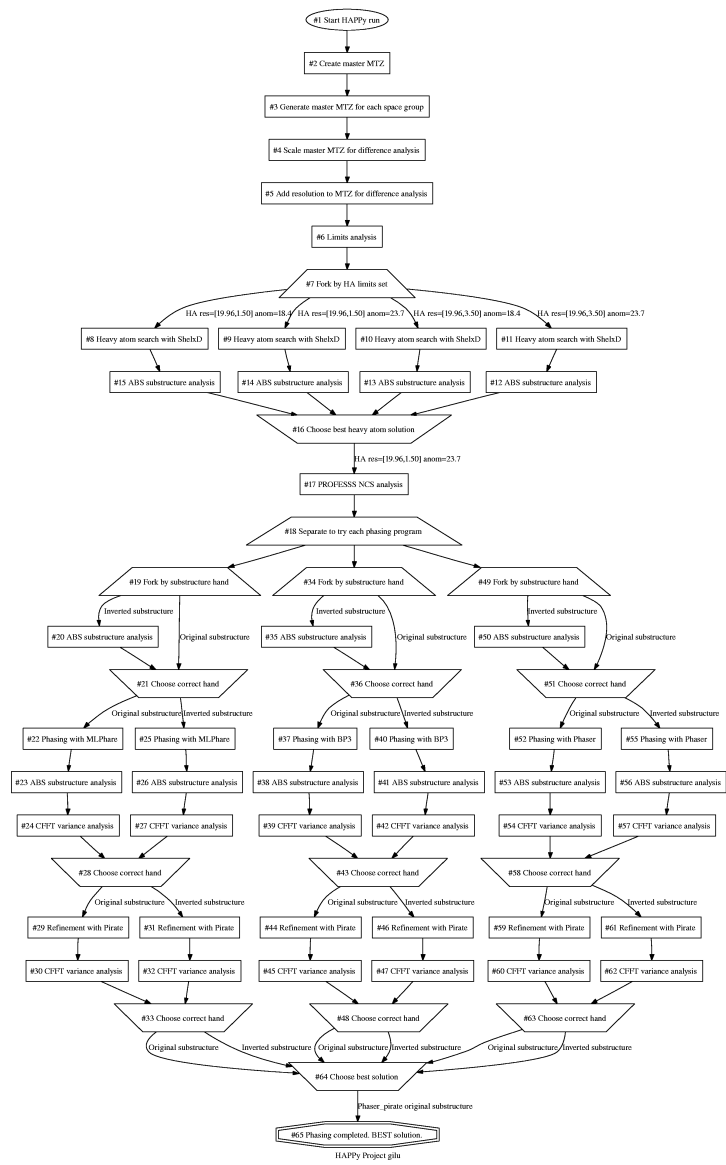
</project>
```

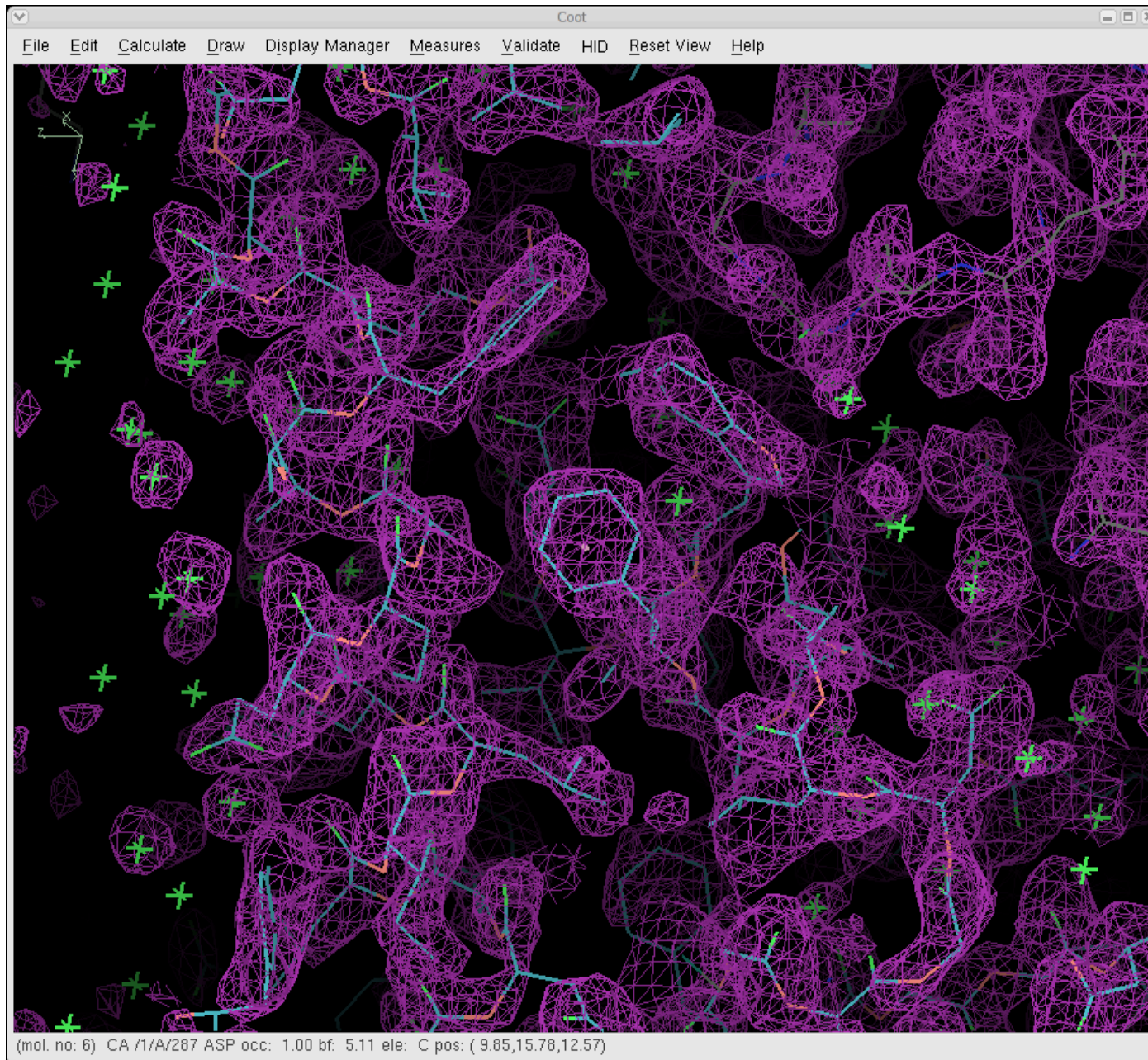
Substructure XML

```
<?xml version='1.0'?>
<substructure crystal="Mn" type="INITIAL" id="1" inverted="False">
  <space_group>UNQID:23_I222</space_group>
  <program>SHELXD</program>
  <hand>None</hand>
  <atom id="1">
    <atom_id>Mn1</atom_id>
    <element_name>Mn</element_name>
    <x>0.082214</x>
    <y>0.632553</y>
    <z>0.066377</z>
    <anomalous_occupancy>1</anomalous_occupancy>
    <bfactor>
      <iso>20</iso>
    </bfactor>
  </atom>
  <atom id="2">
    ...
  </atom>
  <score type="SHELXD">1.39</score>
  <score type="SHELXD_CCweak">19.94</score>
  ...
</substructure>
```









Future plans

- 😊 Multiple entry points (e.g. provide initial substructure).
- 😊 Further phasing modes.
- 😊 Model building.
- 😊 Modules for other substructure determination software.
- 😊 Effective use of clusters and multiple CPU machines.
- 😊 Integration with next generation CCP4(i) database.
- 😊

Further information

- 😊 Website www.ccp4.ac.uk/HAPPy. Information will appear here.
- 😊 Suggestions, requests welcome.