

1 Introduction

This document describes the use cases and API's for the program Mosflm. Mosflm is an unusual case, because there are number of significant functions it performs - each of these will have to be a well identified use case.

1.1 Functionality

The program mosflm has the following key pieces of functionality which need to be exposed *via* the API:

- Autoindexing of diffraction patterns.
- Estimation of mosaic spread.
- Refinement of unit cell.
- Integration of diffraction patterns.
- Calculation of data collection strategies (deprecated.)

1.2 Input

Each of these tasks requires a certain amount of common input - it can only make sense to standardise on the API for this. Of particular importance are the content of the image header and the frames to use in processing.

2 Rules

2.1 Introduction

There are a couple of rules which are useful when running mosflm, and these should be encoded in the *wrapper* rather than the rest of the expert system because they are specific to this program.

These rules pertain to:

- Indexing errors, in particular lattice assignment.

2.2 Indexing Errors & Lattice Assignment

Prototype data set: 1VR9 native (12847) from the JCSG. This is correctly C2 symmetry, but is pseudo-I222.

Autoindexing with mosflm, even with the “refine” keyword assigned, gives a lattice of I222. However, when this is set the mosaic spread estimation & refinement fails, which is a pointer that there is something wrong. The estimation fails saying “you’ll have to guess” or something, and is then refined to a negative value in cell refinement. This should be interpreted as

a pointer that the lattice has too high a symmetry. Asserting a spacegroup of C2 corrects this, so clearly this is something important to build into the wrapper.