The handle http://hdl.handle.net/1887/22715 holds various files of this Leiden University dissertation.

**Author:** Waterreus, Willem-Jan  
**Title:** Software developments in automated structure solution and crystallographic studies of the Sso10a2 and human C1 inhibitor protein  
**Issue Date:** 2013-12-05
Recent advances in **Crank**

**Abstract**

For its first release in 2004, **Crank** was shown to effectively detect and phase anomalous scatterers from SAD data. Since then, **Crank** has significantly improved and many more structures can be built automatically with single or multiple wavelength anomalous diffraction or SIRAS data. Here, we discuss the new algorithms we have developed that lead to the substantial improvements and show **Crank**’s performance on over one hundred real data sets. The latest version of **Crank** is freely available for download at [http://www.bfsc.leidenuniv.nl/software/crank/](http://www.bfsc.leidenuniv.nl/software/crank/) and from CCP4 ([http://www.ccp4.ac.uk/](http://www.ccp4.ac.uk/)).


The author of this thesis implemented utilities for deployment of massive parallel tests on openPBS; implemented a generic framework for parsing log files and automated analysis of extracted statistics; performed minor debugging of the **Crank** program support scripts; evaluated, summarized and presented the results of parallel tests.
2. Recent advances in Crank

2.1 Introduction

Currently, many software packages are available to automatically solve structures. The main aim of Crank (Ness et al. 2004) is to provide a user friendly and automated system incorporating the latest computational developments in all stages of structure solution by experimental phasing. Crank is not a monolithic system: users can define pipelines from a choice of many different programs. Figure 2.1 shows the current steps that Crank can perform and the programs that users can select to perform the task. The externally developed programs that Crank can interface with are SHELXC (Sheldrick 2008), SHELXD (Schneider and Sheldrick 2002), SHELXE (Sheldrick 2002), DM (Cowtan 1994), PARROT (Cowtan 2010), PIRATE (Cowtan 2000), BUCCANEER (Cowtan 2006), ARP/wARP (Langer et al. 2008) both of which iterate with REFMAC (Murshudov et al. 2011) and RESOLVE (Terwilliger 2000, 2002).

We are the main authors of the programs AFRO (Pannu et al., in preparation) for \( F_A \) calculation, CRUNCH2 (de Graaff et al. 2001) for substructure detection, Bp3 (Pannu and Read 2004) for substructure phasing, SOLOMON (Abrahams and Leslie 1996) for density modification, MULTICOMB (Skubák et al. 2010) for phase combination and co-authors of the program REFMAC. These programs use multivariate maximum likelihood methods that allow the observed diffraction data and any current models to be considered simultaneously at any stage in the structure solution process. Thus, the wealth of information contained in the observed diffraction data can be used directly throughout the structure solution process and not approximated or ignored as current approaches do after constructing an initial electron density map.

Below, we provide a brief intuitive description of the novel methods in various steps in experimental phasing that we have developed since our first publication on Crank. We show the power of combining all of these new methods on over one hundred real single- and multiple-wavelength anomalous diffraction (SAD/MAD) and single isomorphous replacement with anomalous scattering (SIRAS) data sets run automatically with minimal user input in Crank.
2.1. Introduction

Input: reflections

Substructure factor amplitude estimation
AFRO, SHELXC

Substructure detection
CRUNCH2, SHELXD

Phasing
Bp3, SHELXE

Density modification
SOLOMON, PARROT, PIRATE, RESOLVE

Model building (& refinement)
BUCCANEER (& REFMAC), ARP/wARP (& REFMAC), RESOLVE

Output: model coordinates

Figure 2.1: The steps in the structure solution process from substructure factor amplitude estimation to model building. The bottom half of each node shows the programs that CRANK can interface with in that step. In the final phase the model can be refined by iterating either BUCCANEER or ARP/wARP with REFMAC.
The programs and methods we develop are not only available in CRANK, but also AUTORickshaw (Panjikar et al. 2005) and ARP/wARP. Furthermore, the original methods we have developed have also been re-written in mathematically identical forms in both phenix.refine and phaser (Adams et al. 2010).

2. Recent developments in Crank

2.2 Substructure determination

After diffraction data has been indexed and merged, $|F_A|$ values are calculated for input to substructure detection programs. $|F_A|$ values are the amplitudes of structure factors corresponding to the heavy atoms to be located. For single-wavelength anomalous diffraction (SAD) data, most programs use the absolute value of Bijvoet differences, $\Delta F = ||F^+| - |F^-||$ as an estimate for $|F_A|$. Burla et al. (2002) proposed employing multivariate joint probability distributions to obtain the expected value for $|F_A|$ in an equation that contains three integrals. In order to obtain an analytical solution to the integrals, Burla et al. (2002) assume the “Bijvoet phases” are equal. We have obtained an expression requiring only one numerical integration without making this assumption. This approach has been implemented in the program AFRO and performs satisfactorily. Details of the implementation and test results will be shown elsewhere (Pannu et al., in preparation). The development version of AFRO containing the multivariate $|F_A|$ calculation is available in the latest version of CRANK and can be used as input for either CRUNCH2 or SHELXD.

Within CRANK, methods exist to validate whether a correct substructure has been determined and to terminate the substructure detection step early. If a threshold value for a statistic used by the substructure detection program has been reached or if a significant deviation exists between the best and worst score in different trials, the substructure detection program will successfully terminate before running all trials. CRANK also provides an alternate and independent assessment
of whether a correct substructure solution has been located: an option exists to run the substructure phasing program Bp3 quickly in “check” mode and examine likelihood based statistics to determine whether a correct and complete substructure has been found. The statistic that CRANK uses is a Luzzati (1952) parameter: if the average Luzzati parameter is greater than a threshold value (the default is 0.7), it is assumed that the full substructure has been found and substructure detection is terminated. Using likelihood methods to validate substructure detection has been available in CRANK for over three years (Pannu et al. 2007) and this approach has been appreciated by phenix developers, who recently adopted it in their own suite (Paul Adams, CCP4 bulletin board, 31 July 2010).

2.2.2 Substructure phasing

To incorporate anomalous phase information, heavy atom refinement programs such as SHARP (Bricogne et al. 2003) or MLPHARE (Collaborative Computational Project Number 4 1994; Otwinowski et al. 1991) use a Gaussian function on observed Bijvoet differences ($\Delta F = |F^+| - |F^-|$) centered on the “calculated” Bijvoet difference that is determined from an assumed value of the “true” structure factor and the heavy atom structure factor (Matthews 1966; North 1965). Since, in general, the “true” structure factor is not known for a SAD or multiple-wavelength anomalous diffraction (MAD) experiment, SHARP integrates out the amplitude and phase of the true structure factor. Furthermore, the estimate of measurement error for Bijvoet differences is determined by merging the measurement errors for Friedel pairs ($\sigma_{\Delta F} = \sqrt{\sigma_{F^+}^2 + \sigma_{F^-}^2}$), leading to suboptimal use of experimental information.

To input the observed structure factors directly, it is necessary to consider a joint probability of all observations given a current model. We have previously shown that this method provides better results over other approaches for the case of SAD (Ness et al. 2004; Pannu and Read 2004) as implemented in Bp3. We have recently shown that better results may be obtained by deriving a multivariate function for SIRAS
2. Recent advances in Crank (Skubák et al. 2009) which will be released in the next version of Crank.

2.2.3 Density modification

In the density modification (DM) procedure, the density modified map is iteratively combined with the initial map obtained from experimental phasing. Current methods assume that these two maps are independent and propagate the initial map’s phase information indirectly through Hendrickson-Lattman coefficients (Hendrickson and Lattman 1970). We have applied a multivariate analysis that considers the observed Friedel pairs directly for a SAD experiment, accounts for the correlation between the initial and density modified map and refines the errors that can occur in a SAD experiment. Results on many test cases show a significant improvement over the current state of the art (Skubák et al. 2010): the maps produced by the multivariate phase combination algorithm lead to many more structures being built automatically.

Despite the improvements in the quality of electron density maps, figures of merit remained escalated after DM. To obtain more accurate FOMs, we have recently developed and implemented a new cross-validated scheme for accurate error parameter estimation in likelihood based phase combination. The method leads to more reliable phase probability statistics from DM and results in a further improvement in subsequent model building. In addition, the more accurate FOMs enable a more reliable hand determination or identification of incorrect non-crystallographic symmetry (NCS) operators used in DM (Skubák and Pannu 2011). These developments have been implemented in a new phase combination program called Multicomb and can be used in conjunction with either Solomon or Parrot.

2.2.4 Automated model building and refinement

The incorporation of experimental phase information was previously shown to improve refinement (Pannu et al. 1998). However, the likelihood function developed, typically denoted maximum likelihood Hendrickson-Lattman (MLHL), propagates the external phase informa-
tion via Hendrickson-Lattman coefficients (Hendrickson and Lattman 1970). Thus, the MLHL function is dependent on the accuracy and reliability of the coefficients that are input. Furthermore, in its derivation, the MLHL function assumes that the experimental phase information (represented by Hendrickson-Lattman coefficients) is independent from the calculated structure factor. This assumption is questionable, as the experimental phase information is used to build an initial model. To overcome these issues, we considered and derived a multivariate likelihood function for SAD (Skubák et al. 2005) and SIRAS (Skubák et al. 2009) experiments. The likelihood functions take as input the diffraction data directly, the heavy atom coordinates and the calculated structure factors and accounts for correlation between them. Compared to the other likelihood functions in Refmac, more models are built automatically in ARP/WARP with the multivariate functions. The SAD and SIRAS functions in Refmac are available in Crank both in model building with ARP/WARP and Buccaneer.

2.2.5 Integration of programs and steps

To support the integration of the different programs it interfaces with, Crank has a plug-in architecture and communicates between plugins via an XML file. At the moment, there are two methods available to generate an XML file that Crank uses to run a pipeline: the program gcx (Ness et al. 2004) and a CCP4i graphical user interface. Both interfaces to Crank can be run with only minimal input: an mtz file with the relevant column labels specified, a sequence file and the name, expected number and f' and f" values for the heavy atoms. However, users can customize settings for individual programs, define custom made pipelines using any programs at each step and define the start and end step for a particular pipeline.

The program gcx allows Crank to be run from command line with a simple unix script: more information on it can be obtained from the program’s documentation (http://www.ccp4.ac.uk/html/gcx.html). The test cases described below were run with gcx. Most users are
2. Recent advances in Crank

likely to run Crank via the CCP4i GUI. The most convenient way to view a Crank log file is via the baubles system that can be initiated with the “View Annotated Logfile in a Web Browser” option in the CCP4i GUI. Documentation for Crank can be found at the CCP4 wiki (http://www.ccp4wiki.org/) which includes information on how to best interpret the log files. Figure 2.2 shows the CCP4i GUI with its few required fields.
2.3 Methods

Here, we test the new methods described above on a wide range of real SAD, MAD and SIRAS merged diffraction data sets. For our tests, only the intensities or structure factor amplitudes, along with the sequence for a protein monomer, the number of substructure atoms expected per monomer and the $f'$ and $f''$ values for the substructure atoms were input. CRANK performed substructure detection using AFRO and CRUNCH2, BP3 for substructure phasing and SOLOMON with MULTICOMB was used for DM. Three cycles of BUCCANEER iterated with REFMAC were used for automated model building with iterative refinement. All options or parameters were default in all programs. The defaults set by CRANK depend upon the particular experiment: for SAD data, AFRO uses the multivariate $|F_A|$ value calculation, MULTICOMB uses the multivariate SAD function for phase combination in DM while BUCCANEER uses the SAD function implemented in REFMAC. For SIRAS data, AFRO calculates $|F_A|$ from either the anomalous signal or using isomorphous differences by determining which signal is greater. BP3 uses the uncorrelated SIRAS function described previously (Pannu et al. 2003), SOLOMON uses MLHL phase combination in MULTICOMB, while BUCCANEER uses the multivariate SIRAS function in REFMAC. Finally, for MAD data, AFRO chooses the wavelength with the greatest anomalous signal and calculates multivariate $F_A$ values from it. Similar to SIRAS data, SOLOMON uses MLHL phase combination in MULTICOMB to perform DM and BUCCANEER uses the MLHL likelihood function in REFMAC for model refinement.

In the test cases below, the previous version of CRANK, 1.3, is tested with the current version, 1.4. The main differences between the two versions is the development version of AFRO that calculates multivariate $|F_A|$ values given SAD data and the use of MULTICOMB for phase combination in DM which are both introduced in version 1.4.

In total, we report results from 116 real data sets from several different sources listed in Appendix A. The data sets provide a wide range of resolution (from 0.94 to 3.29 Å) and anomalous scatterers, including se-
2. Recent advances in Crank

Figure 2.3: Improvement of Crank version 1.4 compared with Crank version 1.3 in terms of the fraction of the model built for SAD, MAD and SIRAS data sets. The fraction of the model built is defined as the fraction of the coordinates that are within 1 Å of the coordinates of the structure deposited in the PDB.

Of the 116 data sets, 63 are MAD data sets, 46 are SAD data sets, and 7 are SIRAS data sets.

2.4 Results and Discussion

Figure 2.3 shows the fraction of the backbone built within 1 Å of the final deposited structure for each of these data sets for the current version of Crank (1.4) versus the previous version (1.3). In total, 77 out of 116 structures have greater than sixty percent of the structure built correctly and of these 77 structures, 66 are built to over eighty percent completeness.
2.4. Results and Discussion

An example of an automatically built structure with a weak signal is GerE (Ducros et al. 2001). The structure of GerE was originally solved with a four wavelength seleno-methionine MAD data set collected at 2.7 Å and a native data set diffracting to 2.1 Å. CRANK version 1.3 could build the structure from just the peak data set to a high degree, but failed to build the structure with just the SAD inflection data set. CRANK version 1.4 can build the structure to a high degree using either the peak or inflection data set. We are unaware of any other automated package or collection of algorithms that can build GerE using either the peak or inflection data set automatically. To give an indication of the anomalous signal, Figure 2.4 plots the Bijvoet ratio (i.e. $|\Delta F|/|F|$) as a function of resolution bin for the GerE peak and inflection wavelength data: the overall Bijvoet ratios for the peak and inflection data are 0.167 and 0.139, respectively.

For the 77 structures that were built automatically, substructure determination successfully terminated early in 69 of the cases. For 33 of the 69 cases, the Luzzati parameter statistics in BP3 allowed the early termination, while the remaining 36 cases the complete substructure was validated by an analysis of the CRUNCH2 statistics.

2.4.1 Analysis of data sets that were not automatically built

39 of the 116 data sets could not be built automatically by CRANK. 19 of the 39 data sets failed at substructure detection and could be built automatically if the resolution cutoff in CRUNCH2 was changed or SHELX-C and SHELX-D was used in substructure detection. It should also be noted that the five cases that could not be built in version 1.4 but were successful in version 1.3 are all due to the changes in the substructure detection algorithm. These tests will be used to further debug and improve the development version of the multivariate $|F_A|$ calculation in AFRO.

For 5 of the 39 cases, CRANK in conjunction with a new SIRAS function for phasing leads to building when the current “uncorrelated”
Figure 2.4: Plot of the Bijvoet ratios as a function of resolution for the peak and inflection wavelength data of the GerE test case. The peak and inflection wavelength are shown with a solid and dashed line, respectively.

function in Bp3 had failed to produce an automatically traceable map. The multivariate SIRAS function for phasing will be released in the next version of Crank.

The remaining 15 cases could not be built automatically or manually in Crank. For 7 of these cases, Mueller-Dieckmann et al. (2007) had also failed to build the structures. Similarly, 4 other cases consisted of SAD experiments using derivative data sets from SIRAS experiments also containing a very weak signal. It is very likely that no currently available methods can build these structures and new methods need to be developed to build structures from such weak data. The remaining 4 cases that could not be built are from the JCSG repository: these structures can be built with currently available methods and the given data. The reasons why Crank fails to build these data sets are yet to
be determined.

2.5 Conclusions and future developments

Because of the new methods we have developed, CRANK can build many more structures automatically and can build structures where current methods fail. CRANK’s robustness is shown by the large number of data sets we use in this test that require very minimal input.

CRANK’s CCP4i GUI is easy to use, but does have some limitations. Firstly, log files are only updated once a particular step in the pipeline has finished. Secondly, users can not manually stop a current step and proceed to a next step. Instead the pipeline can only be terminated and the CRANK run must be restarted from the beginning. Furthermore, although, CRANK has an interface to COOT (Emsley et al. 2010), it cannot show real time updates of a model as a CRANK run proceeds. All of these shortcomings are being addressed and a new pyQt (http://www.riverbankcomputing.co.uk/software/pyqt/intro) interface for CRANK is currently being developed in collaboration with CCP4.

Although having an easy to use and powerful interface is important, the first priority for CRANK will always be developing better methods to solve data sets that elude current methods. For the case of MAD data, current approaches in CRANK and elsewhere use univariate, uncorrelated likelihood functions for $F_A$ calculation, substructure phasing and the MLHL function for DM and automated model building and refinement. Obviously, a multivariate MAD function could address the shortcomings in current approaches and could lead to structures where current methods fail.

In the case of SAD data, the multivariate functions used in substructure phasing, DM and model refinement only differ in the number of input variables and the parametrization. Although current algorithms separate these steps, the common mathematical framework suggests that all the information could be used simultaneously and combined optimally in a unified process using a single mathematical function, possibly resulting in substantial improvements.