STRUCTURAL BIOLOGY

ISSN 2059-7983

Comparison of automated crystallographic model-building pipelines

Emad Alharbi,[a,b]* Paul S. Bond,[c] Radu Calinescu[a] and Kevin Cowtan[c]

[a]Department of Computer Science, University of York, Heslington, York YO10 5GH, England, [b]Department of Information Technology, University of Tabuk, Tabuk, Saudi Arabia, and [c]Department of Chemistry, University of York, Heslington, York YO10 5DD, England. *Correspondence e-mail: emad.alharbi@york.ac.uk

Received 14 June 2019
Accepted 4 November 2019

Edited by K. Diederichs, University of Konstanz, Germany

Keywords: structure solution; model building; software.

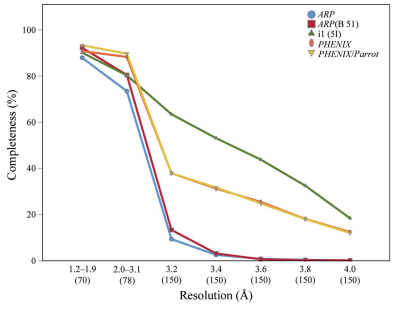
Supporting information: this article has supporting information at journals.iucr.org/d

A comparison of four protein model-building pipelines (*ARP/wARP*, *Buccaneer*, *PHENIX AutoBuild* and *SHELXE*) was performed using data sets from 202 experimentally phased cases, both with the data as observed and truncated to simulate lower resolutions. All pipelines were run using default parameters. Additionally, an *ARP/wARP* run was completed using models from *Buccaneer*. All pipelines achieved nearly complete protein structures and low R_{work}/R_{free} at resolutions between 1.2 and 1.9 Å, with *PHENIX AutoBuild* and *ARP/wARP* producing slightly lower *R* factors. At lower resolutions, *Buccaneer* leads to significantly more complete models.

1. Introduction

The automation of protein model building began with the release of *ARP/wARP* in the late 1990s (Perrakis *et al.*, 1999; Lamzin & Wilson, 1993; Morris *et al.*, 2003; Langer *et al.*, 2013), and has rapidly advanced through the development of additional protein model-building pipelines. These pipelines include *Buccaneer* (Cowtan, 2006, 2008), *PHENIX AutoBuild* (Terwilliger *et al.*, 2008), *SHELXE* (Sheldrick, 2008, 2010; Thorn & Sheldrick, 2013; Usón & Sheldrick, 2018) and a major new version of *ARP/wARP* (Langer *et al.*, 2008). Judging by the numbers of Web of Science citations during 2017 and 2018, *ARP/wARP* (286 citations), *Buccaneer* (304 citations) and *PHENIX AutoBuild* (217 citations) are all widely used; *SHELXE* was cited 9548 times within the same time period (with all citation counts being based on the papers listed above).

Complex optimization problems such as building protein structures can be tackled using multiple approaches. As such, different protein-building pipelines employ different steps and algorithms, may refine their intermediate structures using difference refinement programs such as *REFMAC* (Murshudov *et al.*, 2011) or *phenix.refine* (Afonine *et al.*, 2012) and yield different results for the same data. The comparison detailed here sheds light on some of these differences by examining the completeness of protein structures, R_{work}/R_{free} values and execution times using *ARP/wARP*, *Buccaneer*, *PHENIX AutoBuild* and *SHELXE*. Performed for data sets with resolutions ranging from 1.2 to 4.0 Å, this comparison provides insights into the strengths and weaknesses of the different pipelines, which may be of use to users seeking to address specific problem data sets, as well as to developers seeking to improve their own algorithms or to build new meta-pipelines which exploit the complementary strengths of the different algorithms.



As scientists are inevitably affected by cognitive biases, including self-serving biases, this study would ideally have been conducted by an independent party, similar to the study of van den Bedem *et al.* (2011). However, independent researchers often lack the motivation to perform detailed tool comparisons. For us, further development of the *Buccaneer* methods required a better understanding of their limitations, and thus we conducted our own comparison. We acknowledge that its results may have been impacted by biases in our study, and we make those sources of bias that we are aware of explicit in the discussion.

2. Pipelines and methods

2.1. *ARP/wARP*

ARP/wARP was the first fully automated pipeline for building protein models from electron-density maps. Initially limited to high resolutions of better than 2.3 Å (Perrakis *et al.*, 1999), *ARP/wARP* was subsequently extended to resolutions of 2.7 or 2.8 Å (Langer *et al.*, 2008). More recent versions have further enlarged the useful range of resolutions (Chojnowski, 2019). *ARP/wARP* is integrated with *CCP*4, and therefore can be used from the *CCP*4 graphical user interfaces (GUIs). Additionally, *ARP/wARP* has a web service interface for remote running, which enables access to resources beyond those available on the users' local machines.

The *ARP/wARP* approach starts by placing free atoms in the electron-density map. Free atoms are atoms that do not have a chemical identity but are likely to develop one during model building and refinement. The approach then traces the main protein chain via an algorithm (Morris *et al.*, 2002) that uses modified depth-first search techniques. Next, *ARP/wARP* uses a rotamer library and a downhill simplex algorithm to fit the side chains into the map density. Finally, the missing parts of the protein model are completed by matching C^{α} segments from known models and choosing those that best fit the density of the working model. Following the building stage, the model is refined with *REFMAC* and the calculated map is used for further *ARP/wARP* building cycles.

2.2. *Buccaneer*

Buccaneer is a command-line protein model-building tool developed by Cowtan (2006). Its subsequent integration with the Collaborative Computational Project Number 4's *CCP*4 software suite (Winn *et al.*, 2011) provided *Buccaneer* with a graphical user interface through the *CCP*4*i* (Potterton *et al.*, 2003) and *CCP*4*i*2 (Potterton *et al.*, 2018) GUIs.

The *Buccaneer* algorithm is built around a likelihood target function for the identification of likely C^{α} positions. This function is used to find a small set of 'seed' residues and then to grow these seeds into chain fragments using Ramachandran restraints. Overlapping chain fragments are merged and are docked into the sequence on the basis of a further application of the likelihood target function to the identification of the side-chain type (Cowtan, 2006, 2008). Model building is iterated with refinement in *REFMAC* (Murshudov *et al.*, 2011).

2.3. *PHENIX AutoBuild*

PHENIX AutoBuild is part of the *PHENIX* software suite for the automated modelling of molecular structures. Using a GUI based on the main *PHENIX* GUI, *PHENIX AutoBuild* facilitates the interactive specification of protein model-building parameters, with default values automatically provided for most parameters. Additionally, command-line access is available to enable the integration of *PHENIX AutoBuild* with other tools.

PHENIX AutoBuild accepts several types of input, experimental phases, an existing model and a model whose sequence differs by less than 5% from that of the target model, and performs different procedures for each input type. The steps in its fully automated pipeline include density modification, model building and refinement (Terwilliger, 2000, 2002, 2003; Liebschner *et al.*, 2019). These *PHENIX AutoBuild* steps are not executed sequentially, as the density modification is repeated after refinement to exploit information from the built model.

Early in the structure-determination procedure, *PHENIX AutoBuild* scores models using a metric based on the number of residues built, the number of residues that match the protein sequence and the number of chains (Terwilliger *et al.*, 2008). Later, when their $R_{\rm work}$ value drops below a pre-set value, the models are scored mainly using $R_{\rm work}$. Refinement of the built structures is performed using *phenix.refine* (Afonine *et al.*, 2012), a refinement tool from the *PHENIX* suite.

2.4. *SHELXE*

SHELXE is a program for main-chain tracing and density modification from experimental phases and molecular replacement (Sheldrick, 2010; Thorn & Sheldrick, 2013). Backbone tracing begins by finding seven-residue α-helices and extending them in both directions whenever possible. The latest version of *SHELXE* has been extended to find up to 14 residues (Usón & Sheldrick, 2018). The traced chains are then cut at their closest points of contact and the N-termini and C-termini are joined together. Finally, new estimated phases are calculated from the traced residues and combined with the initial phases for use in the next cycle of density modification and tracing (Sheldrick, 2010).

SHELXE scores a built structure using a correlation coefficient (CC) calculated from structure factors from the trace against native data. A CC of above 25% at 2.5 Å resolution indicates that *SHELXE* may have found a correct solution (Usón & Sheldrick, 2018).

2.5. Data sets

We used 202 real data sets (van den Bedem *et al.*, 2011) with resolutions between 1.2 and 3.2 Å (Fig. 1), as well as synthetic data sets obtained through simulating each of the original data sets at resolutions of 3.2, 3.4, 3.6, 3.8 and 4.0 Å. The 202 data sets used are a subset of the 770 data sets from van den Bedem *et al.* (2011). A total of 230 structures were available to the authors, of which 229 had one or more data sets from

experimental phasing. A single data set, with the highest r.m.s.d. of local map r.m.s.d., was chosen for each structure. There is no guarantee that the chosen data set is the same one as used for the final deposited structure, but in order to check this the deposited coordinates were refined against the chosen data set using *REFMAC* v.5.8.0158 in *CCP*4 v.7.0.045 (Murshudov *et al.*, 2011). 11 structures failed owing to large differences between cells and one structure failed owing to a serine residue being labelled UNK. A further 15 structures were removed as they had very high *R* factors after refinement. Five of the deposited structures (PDB entries 2a9v, 2ash, 2awa, 2o5r and 2pnk) have their structure-determination method listed as a combination of MAD and molecular replacement, and one (PDB entry 2fcl) has only molecular replacement. In these cases the deposited structure may contain some model bias from the search model used by the original author. This simulation involved inflating the *B* factors of the structure-factor amplitudes and removing the reflections with resolutions higher than the target resolution. Inflation of *B* factors was carried out by first downloading a list of all structures in the PDB, each with a resolution and an average *B* factor. A linear fit was then performed, which gave a gradient of 32.8 Å that was used to inflate the *B* factors by the difference in resolution. This modification resulted in the reduction of the resolution of the electron-density map to that of the simulated resolution. This process produced 1009 synthetic data sets: five synthetic data sets at the lower resolutions mentioned above for each original data set, except for a single data set in which the original resolution was already 3.2 Å. This gave 1211 data sets in total. The 52 data sets that had previously been used in the development of *Buccaneer*[1] were excluded, along with the synthetic data sets obtained from them.

The density of both the original and synthetic data sets was then modified using *Parrot* (Cowtan, 2010) for three density-modification types: heavy-atom NCS (HA-NCS) determined using S-atom or Se-atom positions from the deposited model, molecular-replacement NCS (MR-NCS) determined using all atoms of the deposited model and no NCS (NO-NCS). The three groups of 1211 data sets (*i.e.* 3633 data sets in total) created in this way were used in the comparison.

The following PDB entries were used in the comparison (the omitted data sets are marked with asterisks): 1o6a*, 1vjf*, 1vjn*, 1vjr*, 1vjv*, 1vjx*, 1vjz*, 1vk2*, 1vk3*, 1vk4*, 1vk8*, 1vk9*, 1vkb*, 1vkd*, 1vkh*, 1vkm*, 1vkn*, 1vku*, 1vky*, 1vkz*, 1vl0*, 1vl4*, 1vl5*, 1vl6*, 1vlc*, 1vli*, 1vll*, 1vlm*, 1vlo*, 1vlu*, 1vm8, 1vme*, 1vmf*, 1vmg*, 1vmi*, 1vp4*, 1vp7*, 1vp8*, 1vpb*, 1vpm*, 1vpy*, 1vpz*, 1vqr*, 1vqs*, 1vqy*, 1vqz*, 1vr0*, 1vr3*, 1vr5*, 1vr8*, 1vra, 1vrb*, 1z82*, 1z85*, 1zbt, 1zkg, 1zko, 1ztc, 1zy9, 1zyb, 2a2m, 2a3n, 2a6a, 2a6b, 2a9v, 2aam, 2afb, 2aj6, 2aj7, 2ajr, 2aml, 2anu, 2ash, 2avn, 2awa, 2b8m, 2ess, 2etd, 2eth, 2etj, 2ets, 2f4l, 2f4p, 2fcl, 2fea, 2ffj, 2fg0, 2fg9, 2fna, 2fno, 2fqp, 2fur, 2fzt, 2g0t, 2gb5, 2gfg, 2ghr, 2ghs, 2gjg, 2glz, 2gm6, 2gno, 2gnr, 2go7, 2gpj, 2gvh, 2gvk, 2h1q, 2hag, 2hcf, 2hdo, 2hh6, 2hhz, 2hi0, 2hoe, 2hq7, 2hr2, 2hsb, 2hti, 2huh, 2huj, 2hx1, 2hxv, 2hyt, 2i51, 2i5i, 2i8d, 2i9w, 2ia7, 2ich, 2ifx, 2ig6, 2ii1, 2iiu, 2ilb, 2inb, 2isb, 2it9, 2itb, 2nlv, 2nuj, 2nwv, 2nyh, 2o08, 2o1q, 2o2g, 2o2x, 2o3l, 2o5r, 2o62, 2o7t, 2o8q, 2obn, 2obp, 2oc5, 2oc6, 2od4, 2od5, 2od6, 2ogi, 2oh1, 2oh3, 2okc, 2okf, 2ooc, 2ooj, 2op5, 2opk, 2opl, 2ord, 2osd, 2otm, 2ou6, 2ouw, 2owp, 2oyo, 2ozg, 2ozj, 2p10, 2p1a, 2p4g, 2p4o, 2p7i, 2p8j, 2p97, 2pbl, 2pc1, 2pg3, 2pg4, 2pgc, 2pim, 2pke, 2pn1, 2pn2, 2pnk, 2ppv, 2pr7, 2prr, 2prv, 2prx, 2pv4 and 2pw4.

2.6. Method of comparison

A comparison was conducted of the following versions of the four protein-building pipelines described in Sections 2.1–2.4: *PHENIX AutoBuild* v.1.14, *Buccaneer* in *CCP*4*i*, *ARP*/*wARP* v.8 and *SHELXE* v.2019/1. All binary files were obtained from *CCP*4 v.7.0.066 and were run with the default parameters set by the developers of the pipeline. *ARP*/*wARP* was run without the $R_{\rm free}$ flag, in line with the tool's documentation, and automatically includes a secondary-structure building step in cases where the resolution is worse than 2.7 Å. *PHENIX AutoBuild* by default builds three models at each step, leading to improved results at the cost of computing time. Additionally, the comparison considered several pipeline variants with nondefault parameters as listed below.

(i) *ARP*/*wARP* with the $R_{\rm free}$ flag set and using as initial models the models built by *Buccaneer* in *CCP*4*i*, as one known limitation of *Buccaneer* is its use of fewer model-finalization steps.

(ii) *PHENIX AutoBuild* with density-modified phases (using *Parrot*; Cowtan, 2010).

(iii) *SHELXE* with density-modified phases (using *Parrot*; Cowtan, 2010).

(iv) *SHELXE* (with and without density-modified phases) variants with the `-t` flag set to 20 as a higher value is recommended in the tool's documentation.

Table 1 shows the short names used for these pipeline variants in the rest of the paper.

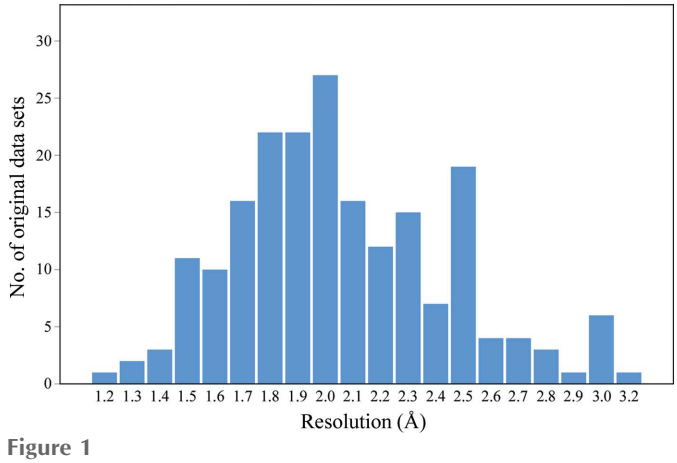


Figure 1
Resolutions of the 202 original data sets.

[1] These 52 data sets were analysed for a secondary study in which we assessed the efficiency of choosing training data sets for pipeline development (supporting information Sections S3 and S4).

Table 1
Pipeline variants used in the comparison.

Short name	Long name
ARP	*ARP*/*wARP*
ARP(B 5I)	*ARP*/*wARP* after *Buccaneer* in *CCP4i* using the default five iterations
i1(5I)	*Buccaneer* in *CCP4i* using five iterations
PHENIX/*Parrot*	*PHENIX AutoBuild* runs after Parrot (density-modified phases)
SHELXE/*Parrot*	*SHELXE* runs after *Parrot* (density-modified phases)
PHENIX	*PHENIX AutoBuild* fed by density-unmodified phases
SHELXE	*SHELXE* fed by density-unmodified phases

Each execution of a pipeline received two inputs: a reflection data file comprising the result of an experimental phasing calculation and the sequence file of the relevant protein. *SHELXE* did not receive the sequence file because it is not required. The model-building task was then submitted as a job to a 173-node high-performance cluster with 7024 Intel Xeon Gold/Platinum cores and a total memory of 42 TB. Each job involved building one protein model and was stopped if it did not complete within 48 h. There was no resource sharing between jobs.

Following model building, a 'zero-cycle' *REFMAC* run was used to calculate $R_{\rm work}/R_{\rm free}$ in order to avoid the confounding effects of different scaling and solvent parameterizations in different refinement programs. *REFMAC* was run with default parameters. The quality of the starting phases was assessed using the weighted *F*-map correlation between the initial map and the phases from the refined deposited model. A structure completeness measure was obtained for the final model by calculating the percentage of residues in the processed deposited model from the PDB whose C^{α} atoms have the same residue type as, and coordinates within 1.0 Å of, the corresponding residue in the built model. *SHELXE* completeness was only calculated for C^{α} atoms in correct positions within 1.0 Å because *SHELXE* only builds the main chain.

A tool was developed to automate the execution of the pipelines and the analysis of their results. To ensure the reproducibility of the study, the execution of all pipeline variants was repeated for a sample of 30 structures. The results (provided as supporting information) did not vary significantly when the pipelines were rerun with the same inputs. Additionally, a series of tests searching for errors that might have occurred during the running or analysis stages were performed; for example, the running parameters from log files were verified for possible errors in the parameter settings.

Three measures were used to compare the protein models built by different pipelines: structure completeness, $R_{\rm work}/R_{\rm free}$ and pipeline-execution time. $R_{\rm work}/R_{\rm free}$ values were rounded to two decimal places and completeness was rounded to the nearest whole number.

For both completeness and $R_{\rm work}/R_{\rm free}$, and for each pair of pipelines, we report the percentage of data sets for which one pipeline yields better models than the other and the percentage of data sets for which one pipeline yields models which are at least 5% better than the models produced by the other pipeline. (Cases in which the results are equivalent or better by between 1% and 4% are reported in the supporting information.) The results obtained for the real data sets used in the comparison and for the data sets truncated to simulate lower resolutions are reported separately. For execution time, we report the mean pipeline-execution times partitioned into classes based on their structure sizes.

3. Results

3.1. Overview

The results described here were obtained by comparing the protein structures successfully built by each of the pipeline variants in Table 1. For the first four pipeline variants in the table we used all 3633 data sets obtained as described in the previous section. For *PHENIX AutoBuild* and *SHELXE* no prior density modification was run and the results were compared with the NO-NCS results from the other pipelines. *SHELXE* variants were not run on synthetic data sets because this is not recommended, and therefore *SHELXE* is omitted from the synthetic data sets comparison.

All pipeline variants successfully completed the analysis of over 99% of both the original and synthetic data sets. The remaining runs did not complete within 48 h (a time limit that we set in our experiments), failed owing to insufficient memory or crashed. In all of these cases the pipeline variant was rerun with its memory quota and time limit increased until it either succeeded or a limit of 20 GB of allocated memory and 48 h were reached. As shown in Tables 2 and 3, only very few runs did not complete (even after this memory increase), and most of these produced intermediate protein models that we used in our comparison. The data sets marked 'Failed' in the tables were excluded from the comparison (for all pipeline variants). The numbers of different types of 'complete' and 'intermediate' models used in the comparison are reported above each table.

Including noncrystallographic averaging improves the starting phases for structures where NCS is present, but it does not significantly affect the conclusions of this work because the completeness is not significantly affected. Given that the differences between NCS and NO-NCS cases are small, the poorer-phased NO-NCS data sets will be considered in the remainder of the comparison.

Using the correct solvent fraction in *SHELXE* improves its results, but does not significantly affect the results when compared with other pipeline variants. A default fraction of solvent of 0.45 is used in the comparison.

3.2. Structure completeness

Tables 4 and 5 report the percentages of models for which each pipeline variant achieved a structure completeness that is higher and at least 5% higher, respectively, than the other pipeline variants. Note that the two figures associated with a pair of pipeline variants in Table 4 do not always add up to 100% because some of the models are generated with the same structure completeness (rounded to the next integer) by the two pipeline variants. For example, the structure

Table 2
Complete and intermediate models produced by the seven pipeline variants for the original data sets, where '(T)' and '(C)' denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Models used in the comparison: 149 HA-NCS, 149 MR-NCS and 148 NO-NCS.

	HA-NCS			MR-NCS			NO-NCS		
Pipeline variant	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
ARP	201	1(T) 0(C)	0	202	0(T) 0(C)	0	202	0(T) 0(C)	0
ARP(B 5I)	202	0(T) 0(C)	0	201	1(T) 0(C)	0	202	0(T) 0(C)	0
i1(5I)	202	0(T) 0(C)	0	202	0(T) 0(C)	0	202	0(T) 0(C)	0
PHENIX/*Parrot*	198	2(T) 1(C)	1	200	0(T) 1(C)	1	199	1(T) 1(C)	1
SHELXE/*Parrot*	202	0(T) 0(C)	0	201	1(T) 0(C)	0	200	2(T) 0(C)	0
PHENIX	—	—	—	—	—	—	199	1(T) 0(C)	2
SHELXE	—	—	—	—	—	—	200	2(T) 0(C)	0

Table 3
Complete and intermediate models produced by the five pipeline variants for the synthetic resolution data sets, where '(T)' and '(C)' denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Models used in the comparison: 750 HA-NCS, 750 MR-NCS and 750 NO-NCS.

	HA-NCS			MR-NCS			NO-NCS		
Pipeline variant	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
ARP	1008	1(T) 0(C)	0	1007	2(T) 0(C)	0	1008	1(T) 0(C)	0
ARP(B 5I)	1005	4(T) 0(C)	0	1006	3(T) 0(C)	0	1003	6(T) 0(C)	0
i1(5I)	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0
PHENIX/*Parrot*	1002	7(T) 0(C)	0	1004	5(T) 0(C)	0	1001	8(T) 0(C)	0
PHENIX	—	—	—	—	—	—	1001	7(T) 0(C)	1

Table 4
Structure completeness comparison for the models generated from the original NO-NCS data sets.

Each row corresponds to a pipeline variant and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	*ARP*	*ARP*(B 5I)	i1(5I)	*PHENIX*/*Parrot*	*PHENIX*	*SHELXE*	*SHELXE*/*Parrot*
ARP	0	23	33	39	37	68	61
ARP(B 5I)	45	0	40	43	43	76	73
i1(5I)	57	45	0	46	49	77	72
PHENIX/*Parrot*	49	44	45	0	46	80	77
PHENIX	48	39	41	32	0	78	72
SHELXE	26	15	20	16	16	0	34
SHELXE/*Parrot*	32	22	24	17	22	57	0

Table 5
Structure completeness comparison for the models generated from the original NO-NCS data sets.

Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	*ARP*	*ARP*(B 5I)	i1(5I)	*PHENIX*/*Parrot*	*PHENIX*	*SHELXE*	*SHELXE*/*Parrot*
ARP	0	6	15	11	14	45	40
ARP(B 5I)	24	0	20	16	16	53	53
i1(5I)	28	17	0	16	16	56	48
PHENIX/*Parrot*	28	20	26	0	14	61	55
PHENIX	28	18	23	7	0	57	51
SHELXE	17	7	11	7	7	0	9
SHELXE/*Parrot*	21	12	17	5	10	32	0

completeness of 23% of the *ARP* models was higher than that of the corresponding *ARP*(B 5I) models, and 45% of the *ARP*(B 5I) models had a higher structure completeness than that of the *ARP* models; thus, the remaining 32% of the models built by *ARP* and *ARP*(B 5I) had the same structure completeness, after rounding.

As shown in the first of these tables, *ARP*/*wARP* built 37% of the data sets better than *PHENIX AutoBuild*, while *PHENIX AutoBuild* did better for 48% of the data sets, which means that 15% of the data sets are equal in their completeness. *Buccaneer* in *CCP*4*i* built more than half of the data sets with higher completeness compared with *ARP*/*wARP*. The default five-cycle *Buccaneer* runs typically produce less complete models than *PHENIX AutoBuild*.

Table 5 shows the number of cases in which one pipeline variant achieved 5% or higher structural completeness than another. By this measure, for every pipeline variant there are at least 3% of cases in which that pipeline produces a significantly more complete model than another pipeline; however, a similar general pattern is shown to the previous comparison.

Running *ARP*/*wARP* after *Buccaneer* can impact the results. Comparing *ARP*/*wARP* after *Buccaneer* in *CCP*4*i* (five iterations) with *ARP*/*wARP* alone showed a 5% improvement in completeness in a quarter of cases, with only a few cases of a comparable decrease in completeness. Using *PHENIX AutoBuild* after *Parrot* showed a small benefit of the additional density-modification step; 14% of the data sets were built better, compared with 7% that were built worse.

Comparison of *SHELXE* with the other pipeline variants shows that over half of the data sets are typically built better by other pipeline variants even when the 5% improvement comparison level is considered. *SHELXE* built 16% of the data sets better than *PHENIX AutoBuild*, but this number decreased to 7% for the 5% improvement comparison level. *SHELXE* after *Parrot* showed some improvements when compared with the other pipeline variants; however, for the 5% improvement comparison level these variants built over 40% of the data sets better than *SHELXE* after *Parrot*.

Fig. 2 shows the mean structure completeness for different ranges of data-set resolutions across both the original and synthetic data sets. As expected, the pipeline variants achieved the best results at 1.2–1.9 Å, and the completeness of the models was significantly poorer at 4.0 Å. *ARP*/*wARP* dropped rapidly at 3.2 Å (synthetic data sets) and decreased to nearly zero completeness at 4.0 Å. In contrast, for *Buccaneer* in *CCP*4*i* the completeness degrades only slowly as the resolution drops below 3.1 Å. *PHENIX AutoBuild* produces the most complete models when using the original data resolution; however, its completeness falls between those of *Buccaneer* and *ARP*/*wARP* for the resolution-truncated data sets. The pipelines were affected by *F*-map correlation, with lower completeness at an *F*-map correlation of 0.53 or lower (Fig. 3).

Fig. 4 shows the mean numbers of residues which were built incorrectly grouped into bins based on the data-set resolution. Achieving high structure completeness leads to the generation of a large number of incorrect residues. For example, *Buccaneer* in *CCP*4*i* built more residues incorrectly than other pipeline variants; for example, a fraction of 0.50 of the residues were incorrect at 4.0 Å, while *PHENIX AutoBuild* only reached a fraction of 0.20 of incorrect residues at the same resolution. *ARP*/*wARP* and *PHENIX AutoBuild* built almost no incorrect residues between 1.2 and 1.9 Å.

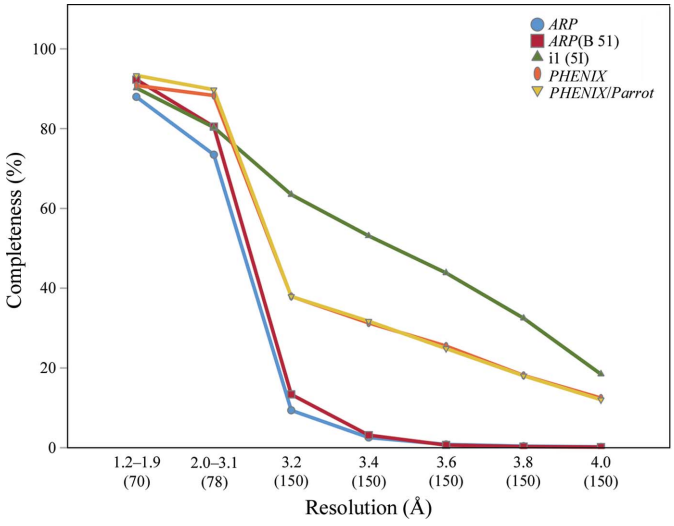


Figure 2
Mean completeness for the protein models built for all NO-NCS data sets. The data sets are grouped into bins based on their resolution, with the number of data sets in each bin shown in parentheses under the graph.

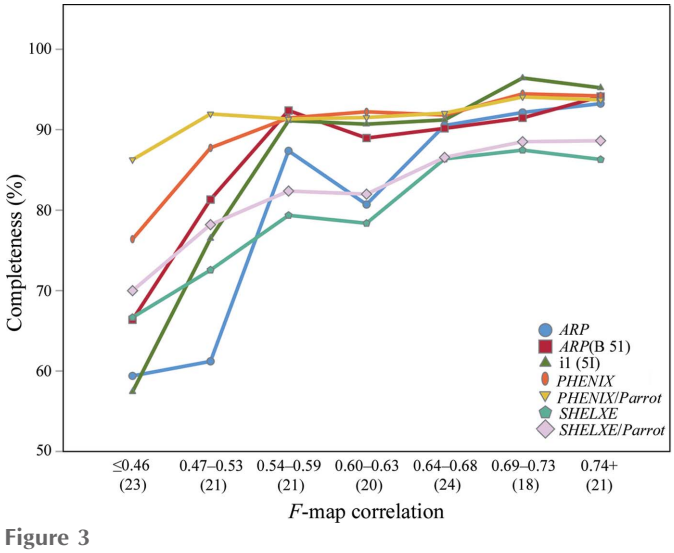


Figure 3
Mean completeness for the models built for the original NO-NCS data sets, grouped into bins based on their initial map correlation (*F*-map correlation); the number of data sets in each bin is reported in parentheses under the graph.

3.3. R_{work} and R_{free}

Tables 6 and 7 show the R_{work}/R_{free} results for the pipeline variants at the two levels of comparison (*i.e.* better and at least 5% better). If R_{free} was not used, no results are reported. *ARP*/*wARP* and *PHENIX AutoBuild* gave results which better explain the X-ray observations than *Buccaneer*. *Buccaneer* in *CCP*4*i* built less than 10% of the data sets with lower R_{work}/R_{free} compared with *PHENIX AutoBuild*, which built 93% of the models with lower R_{work}/R_{free} than the *Buccaneer* pipeline. The performance of *ARP*/*wARP* and *SHELXE* can only be compared with the other pipelines in terms of R_{work} owing to R_{free} not being used, and the results of

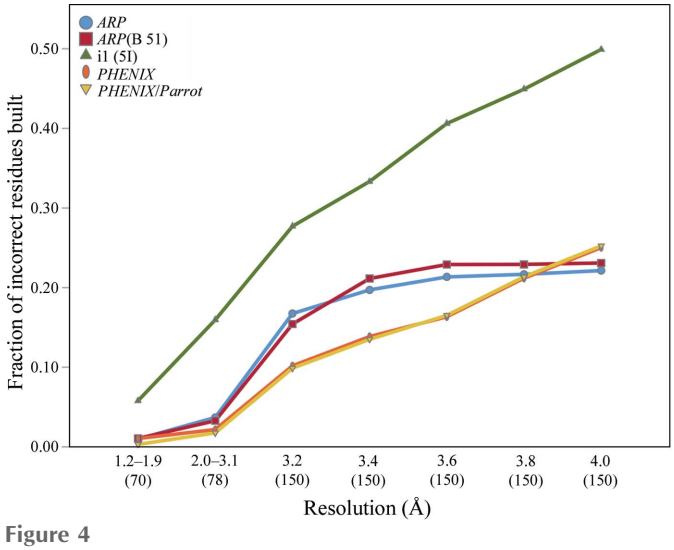


Figure 4
Mean residues incorrectly built for the protein models built for all NO-NCS data sets. The data sets are grouped into bins based on their resolution, with the number of data sets in each bin shown in parentheses under the graph. The number of residues incorrectly built was normalized by dividing by the number of residues in the deposited model.

ARP/*wARP* were closer to those achieved by *PHENIX AutoBuild* than to those with *Buccaneer*. *ARP*/*wARP* built 94% of the models with lower R_{work}, while *Buccaneer* only built 5% of the models with lower R_{work} (Table 6). When considering only cases in which R_{work} or R_{free} change by more than 5% (Table 7), there are comparatively few differences between *ARP*/*wARP* and *PHENIX AutoBuild*, but both outperform the *Buccaneer* pipeline in a significant proportion of cases. All pipeline variants built at least 97% of the models with lower R_{work}/R_{free} compared with *SHELXE* variants, which built 3% of the models with lower R_{work} in the best scenario. These results remain almost the same when the 5% improvement comparison level is considered. Using *SHELXE* after *Parrot* improved R_{work}, but it did not significantly improve the results when compared with other pipeline variants.

Figs. 5 and 6 show the R_{work} and R_{free} obtained for different resolution ranges. As shown in the tables, *PHENIX AutoBuild* achieved the best values at 1.2–1.9 Å, with the results degrading significantly over 3.2 Å. The results of *Buccaneer* degrade more gradually to 4.0 Å. R_{free} increased in the same manner as R_{work}. *ARP*/*wARP* produces very good R_{work} values at all resolutions, although the authors caution that overfitting is a problem in the dummy-atom model. Nonetheless, R_{free} (for the hybrid *Buccaneer* + *ARP*/*wARP* runs, where it is available) is also better than for the other pipelines at lower resolutions, in contrast to the completeness results. This suggests that the dummy-atom model has significant predictive power in explaining the X-ray observations, even when it cannot be interpreted in terms of sequenced protein chain.

Table 6
Comparison of R_{work}/R_{free} (rounded to two decimal places) for the models generated from the original NO-NCS data sets.

Each row shows the percentage of models that a pipeline variant built with lower R_{work} or R_{free} than each other pipeline variant.

Pipeline variant	*ARP*	ARP(B 5I)	i1(5I)	*PHENIX*/*Parrot*	*PHENIX*	*SHELXE*	*SHELXE*/*Parrot*
ARP R_{work}	0	22	94	34	37	100	100
ARP R_{free}	—	—	—	—	—	—	—
ARP(B 5I) R_{work}	45	0	99	44	45	100	100
ARP(B 5I) R_{free}	—	0	85	52	50	—	—
i1(5I)R_{work}	5	0	0	3	3	97	97
i1(5I) R_{free}	—	11	0	3	5	—	—
PHENIX/*Parrot* R_{work}	47	31	95	0	27	99	99
PHENIX/*Parrot* R_{free}	—	43	93	0	31	—	—
PHENIX R_{work}	43	32	93	22	0	99	99
PHENIX R_{free}	—	45	93	31	0	—	—
SHELXE R_{work}	0	0	3	1	1	0	19
SHELXE R_{free}	—	—	—	—	—	—	—
SHELXE/*Parrot* R_{work}	0	0	3	1	1	42	0
SHELXE/*Parrot* R_{free}	—	—	—	—	—	—	—

Table 7
Comparison of R_{work}/R_{free} (rounded to two decimal places) for the models generated from the original NO-NCS data sets.

Each row shows the percentage of models that a pipeline variant built with R_{work} or R_{free} at least 5% lower than each other pipeline variant.

Pipeline variant	*ARP*	*ARP*(B 5I)	i1(5I)	*PHENIX*/*Parrot*	*PHENIX*	*SHELXE*	*SHELXE*/*Parrot*
ARP R_{work}	0	3	52	5	7	100	100
ARP R_{free}	—	—	—	—	—	—	—
ARP(B 5I) R_{work}	5	0	60	6	6	100	100
ARP(B 5I) R_{free}	—	0	60	12	16	—	—
i1(5I) R_{work}	0	0	0	0	1	95	94
i1(5I) R_{free}	—	1	0	0	1	—	—
PHENIX/*Parrot* R_{work}	5	3	54	0	2	99	99
PHENIX/*Parrot* R_{free}	—	13	57	0	2	—	—
PHENIX R_{work}	4	2	55	1	0	99	98
PHENIX R_{free}	—	11	57	1	0	—	—
SHELXE R_{work}	0	0	1	1	1	0	0
SHELXE R_{free}	—	—	—	—	—	—	—
SHELXE/*Parrot* R_{work}	0	0	1	0	1	1	0
SHELXE/*Parrot* R_{free}	—	—	—	—	—	—	—

3.4. Pipeline-execution time

Fig. 7 shows the mean execution times that the pipeline variants required to build the protein models for the original NO-NCS data sets from our comparison. *Buccaneer* in *CCP*4*i* was the fastest pipeline over all structure sizes. *ARP*/*wARP* averaged less than 50 min to build a small structure, making it the second fastest pipeline after *Buccaneer*. Using *Buccaneer* in *CCP*4*i* models as initial models for *ARP*/*wARP* slowed the building of the models compared with the normal run of *ARP*/*wARP*, with averages slightly higher than for normal *ARP*/*wARP*. *PHENIX AutoBuild*, after *Parrot* and without *Parrot*, was the slowest pipeline, with averages of around 200 min to build small structures and more than 1600 min for large structures. *SHELXE* required execution times between those of *ARP*/*wARP* and *PHENIX AutoBuild*, achieving the smallest average when building small structures, but with execution times that increased to over 200 min when building large structures.

4. Discussion

Comparisons of the different model-building pipelines against a range of observed data sets, both at the original resolution and after simulated resolution reduction, highlight different strengths and weaknesses of the different software. These may be used to guide users in choosing the most appropriate software for their problem and developers in the improvement

of their software or the construction of hybrid pipelines using multiple tools.

Comparison of the model completeness, as assessed by the fraction of the model α carbons built to within 1.0 Å of the correct location and assigned the correct residue type, suggests that at better than 3.1 Å resolution *PHENIX AutoBuild* achieves the most complete models, with *Buccaneer* and *ARP/wARP* producing successively less complete models. *PHENIX AutoBuild* was developed mainly against data at better than 3.0 Å resolution (Bunkóczi *et al.*, 2015).

At worse than 3.1 Å resolution *Buccaneer* substantially outperforms the other pipelines, with *PHENIX AutoBuild* giving an intermediate performance and *ARP/wARP* only building a small proportion of residues when averaged across many structures. This is consistent with expectations given that the original design criterion for *Buccaneer* was that it should be more robust against reduced resolution. Running *ARP/wARP* after *Buccaneer* leads to results which are worse than those from *Buccaneer*, suggesting that the residues successfully sequenced by *Buccaneer* are not being retained by *ARP/wARP*.

When comparing model completeness against initial map quality for the original resolution data sets, all of the pipelines perform well when the initial phases are good (correlation of >0.64). The best results are obtained using *PHENIX AutoBuild*, especially after initial phase improvement using *Parrot* (Cowtan, 2010). This suggests that phase improvement in *Parrot* is in some way complementary to the statistical phase improvement which is incorporated in the *PHENIX AutoBuild* pipeline (Terwilliger *et al.*, 2008). *SHELXE* also showed improved model building when starting from phases improved by *Parrot*.

When comparing *R* factors the conclusions are somewhat different. *ARP/wARP* produces the lowest *R* factors across all resolution ranges, and produces dramatically lower *R* factors at worse than 3.1 Å resolution. *PHENIX AutoBuild* comes close to *ARP/wARP* at better than 3.2 Å resolution. *SHELXE* produced the highest $R_{\rm work}$ because it only built the main chain. Sequence assignment and side-chain modelling are likely to significantly reduce the *R* factors as long as the chains built by *SHELXE* do not contain too many tracing errors.

When comparing $R_{\rm free}$ a similar pattern emerges, although at worse than 3.1 Å resolution the free *R* factors from the *Buccaneer* + *ARP/wARP* pipeline show a more modest gain over the other pipelines. [On the basis of developer recommendations and our tests, no free set is used when running *ARP/wARP* on its own (*ARP/wARP* 8.0 User Guide; https://www.embl-hamburg.de/ARP/Manual/UserGuide8.0.html)].

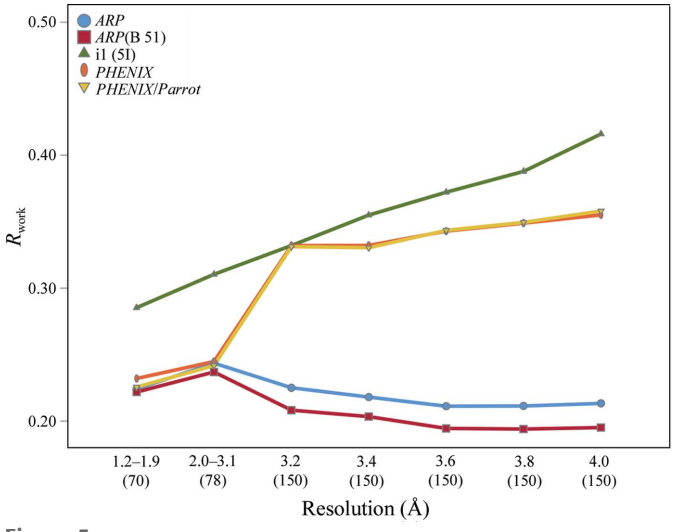


Figure 5
Mean protein model $R_{\rm work}$ for the NO-NCS data sets partitioned into classes based on their resolution. The number of data sets in each class is indicated in parentheses under the graph.

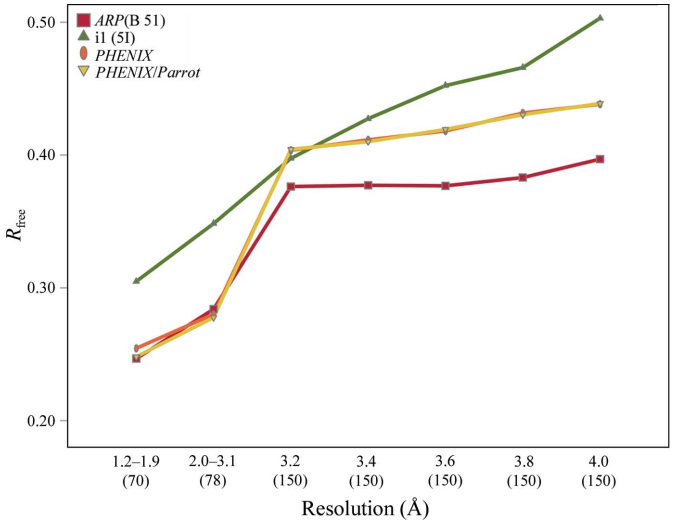


Figure 6
Mean protein model $R_{\rm free}$ for the NO-NCS data sets partitioned into classes based on their resolution. The number of data sets in each class is indicated in parentheses under the graph.

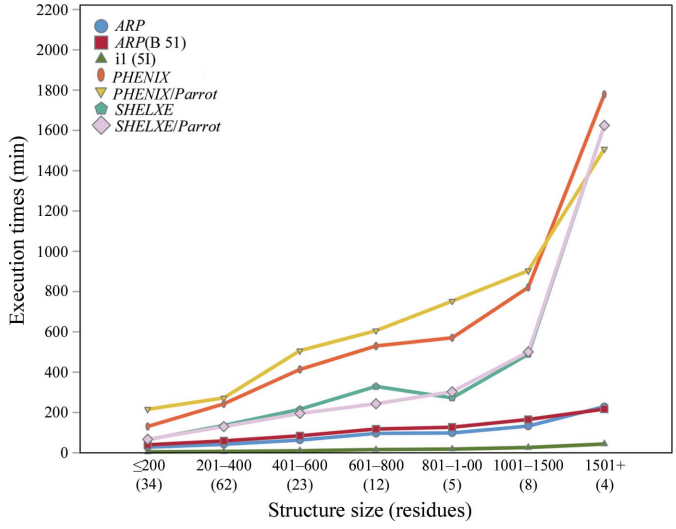


Figure 7
Mean pipeline-execution times for the original NO-NCS data sets partitioned into classes based on their structure sizes. The number of data sets in each class is indicated in parentheses under the graph.

The differing conclusions concerning the effectiveness of *ARP*/*wARP* from the three metrics are connected to the methodology. The use of dummy atoms in the *ARP*/*wARP* calculation allows the observations to be fitted very well and potentially overfitted (Cohen *et al.*, 2008); however, the portion of the model represented by dummy atoms does not contribute to the completeness score used here. The good $R_{\rm free}$ values obtained from *ARP*/*wARP* show that the dummy-atom model has significant explanatory power at lower resolutions even when the dummy atoms cannot be explained in terms of sequenced main chain. This suggests that improved results may be possible either by using *ARP*/*wARP* as a preliminary step for another method or by further development of the methods for interpreting the dummy-atom model.

The performance of a model-building algorithm is determined by multiple factors: the ability of the method to interpret an initial map, the ability of the pipeline to improve this map in the light of the model built so far and the amount of finalization (for example waters, *cis*-peptides and so on) which is performed by the pipeline. The results presented here suggest that *Buccaneer* may be the most effective tool for classifying features in the initial map, especially at lower resolution, but lacks the finalization tools which are present in *ARP*/*wARP* and *PHENIX AutoBuild*, and therefore leads to higher *R* factors. This suggested the use of *ARP*/*wARP* to finalize the *Buccaneer* model; however, the model sequence tends to be lost at lower resolutions, limiting the benefit of this approach. *PHENIX AutoBuild* has however successfully implemented *Buccaneer* as an optional preliminary step (not tested here).

The model-building pipelines show considerable variability in performance from structure to structure, making the *a priori* recommendation of a single method for a given data set difficult. The speed and ease of use of the model-building pipelines mean that users seldom need to try and anticipate which software will be most suitable; instead, most users are likely to use whichever software is most convenient for them. The results presented here may be of use in deciding which pipeline to try next in the case where the first option is unsuccessful. *ARP*/*wARP* and *PHENIX AutoBuild* are likely to be better options at better than 3.1 Å resolution, where their advanced model-finalization tools lead to lower *R* factors. As the resolution drops below 3.1 Å, *Buccaneer* is more likely to produce the most complete model; however, manual editing to remove wrongly built structure is also required.

Given that the software pipelines perform differently on different problem types, the results of any test will inevitably be biased by the choice of test data. In this case, data sets from the Joint Center for Structural Genomics (JCSG; van den Bedem *et al.*, 2011) were used; other JCSG data were also used in the development of *Buccaneer*, although these data sets were excluded from the results presented here. It is possible that this has led to some element of 'tuning' of *Buccaneer* to work on JCSG-sourced data, although the use of different programs for different structures within the JCSG pipeline may mitigate this. Similarly, the resolution-truncation protocol used for low-resolution tests may lead to different results compared with genuine low-resolution data sets. In our case, the resolution-truncation procedure leads to better phases at low resolution than from a real low-resolution data set. Finally, the evaluation criteria also dictate the results; in particular the counting of correctly placed and sequenced α carbons appears to penalize *ARP*/*wARP* at lower resolutions compared with the results of *R*-factor comparisons. Which model is more desirable will depend on the needs of the downstream user.

5. Data and methods

The comparison tool code, the structures built by the pipelines and log files, and the data used are available at https://doi.org/10.15124/d4cb35df-a42d-4365-b539-9868730d165f.

Acknowledgements

This project was undertaken on the Viking Cluster, which is a high-performance compute facility provided by the University of York. We are grateful for computational support from the University of York's High Performance Computing service, Viking and the Research Computing team.

Funding information

This work is funded by University of Tabuk (Emad Alharbi), the White Rose BBSRC DTP in Mechanistic Biology (BB/M011151/1; Paul S. Bond) and the BBSRC (BB/S005099/1; K. Cowtan).

References

Afonine, P. V., Grosse-Kunstleve, R. W., Echols, N., Headd, J. J., Moriarty, N. W., Mustyakimov, M., Terwilliger, T. C., Urzhumtsev, A., Zwart, P. H. & Adams, P. D. (2012). *Acta Cryst.* D**68**, 352–367.
Bedem, H. van den, Wolf, G., Xu, Q. & Deacon, A. M. (2011). *Acta Cryst.* D**67**, 368–375.
Bunkóczi, G., McCoy, A. J., Echols, N., Grosse-Kunstleve, R. W., Adams, P. D., Holton, J. M., Read, R. J. & Terwilliger, T. C. (2015). *Nat. Methods*, **12**, 127–130.
Chojnowski, G. (2019). *Methods Underlying Extension of MR Solutions in ARP/wARP*. Presentation at the CCP4 Study Weekend.
Cohen, S. X., Ben Jelloul, M., Long, F., Vagin, A., Knipscheer, P., Lebbink, J., Sixma, T. K., Lamzin, V. S., Murshudov, G. N. & Perrakis, A. (2008). *Acta Cryst.* D**64**, 49–60.
Cowtan, K. (2006). *Acta Cryst.* D**62**, 1002–1011.
Cowtan, K. (2008). *Acta Cryst.* D**64**, 83–89.
Cowtan, K. (2010). *Acta Cryst.* D**66**, 470–478.
Lamzin, V. S. & Wilson, K. S. (1993). *Acta Cryst.* D**49**, 129–147.
Langer, G., Cohen, S. X., Lamzin, V. S. & Perrakis, A. (2008). *Nat. Protoc.* **3**, 1171–1179.
Langer, G. G., Hazledine, S., Wiegels, T., Carolan, C. & Lamzin, V. S. (2013). *Acta Cryst.* D**69**, 635–641.
Liebschner, D., Afonine, P. V., Baker, M. L., Bunkóczi, G., Chen, V. B., Croll, T. I., Hintze, B., Hung, L.-W., Jain, S., McCoy, A. J., Moriarty, N. W., Oeffner, R. D., Poon, B. K., Prisant, M. G., Read, R. J., Richardson, J. S., Richardson, D. C., Sammito, M. D., Sobolev, O. V., Stockwell, D. H., Terwilliger, T. C., Urzhumtsev, A. G., Videau, L. L., Williams, C. J. & Adams, P. D. (2019). *Acta Cryst.* D**75**, 861–877.
Morris, R. J., Perrakis, A. & Lamzin, V. S. (2002). *Acta Cryst.* D**58**, 968–975.

Morris, R. J., Perrakis, A. & Lamzin, V. S. (2003). *Methods Enzymol.* **374**, 229–244.
Murshudov, G. N., Skubák, P., Lebedev, A. A., Pannu, N. S., Steiner, R. A., Nicholls, R. A., Winn, M. D., Long, F. & Vagin, A. A. (2011). *Acta Cryst.* D**67**, 355–367.
Perrakis, A., Morris, R. & Lamzin, V. S. (1999). *Nat. Struct. Biol.* **6**, 458–463.
Potterton, E., Briggs, P., Turkenburg, M. & Dodson, E. (2003). *Acta Cryst.* D**59**, 1131–1137.
Potterton, L., Agirre, J., Ballard, C., Cowtan, K., Dodson, E., Evans, P. R., Jenkins, H. T., Keegan, R., Krissinel, E., Stevenson, K., Lebedev, A., McNicholas, S. J., Nicholls, R. A., Noble, M., Pannu, N. S., Roth, C., Sheldrick, G., Skubak, P., Turkenburg, J., Uski, V., von Delft, F., Waterman, D., Wilson, K., Winn, M. & Wojdyr, M. (2018). *Acta Cryst.* D**74**, 68–84.
Sheldrick, G. M. (2008). *Acta Cryst.* A**64**, 112–122.
Sheldrick, G. M. (2010). *Acta Cryst.* D**66**, 479–485.
Terwilliger, T. C. (2000). *Acta Cryst.* D**56**, 965–972.
Terwilliger, T. C. (2002). *Acta Cryst.* D**58**, 2082–2086.
Terwilliger, T. C. (2003). *Acta Cryst.* D**59**, 38–44.
Terwilliger, T. C., Grosse-Kunstleve, R. W., Afonine, P. V., Moriarty, N. W., Zwart, P. H., Hung, L.-W., Read, R. J. & Adams, P. D. (2008). *Acta Cryst.* D**64**, 61–69.
Thorn, A. & Sheldrick, G. M. (2013). *Acta Cryst.* D**69**, 2251–2256.
Usón, I. & Sheldrick, G. M. (2018). *Acta Cryst.* D**74**, 106–116.
Winn, M. D., Ballard, C. C., Cowtan, K. D., Dodson, E. J., Emsley, P., Evans, P. R., Keegan, R. M., Krissinel, E. B., Leslie, A. G. W., McCoy, A., McNicholas, S. J., Murshudov, G. N., Pannu, N. S., Potterton, E. A., Powell, H. R., Read, R. J., Vagin, A. & Wilson, K. S. (2011). *Acta Cryst.* D**67**, 235–242.