Acta Crystallographica Section D
**Biological Crystallography**
ISSN 0907-4449

# The *Buccaneer* software for automated model building. 1. Tracing protein chains

**Kevin Cowtan**

Department of Chemistry, University of York, Heslington, York YO10 5DD, England

Correspondence e-mail: cowtan@ysbl.york.ac.uk

A new technique for the automated tracing of protein chains in experimental electron-density maps is described. The technique relies on the repeated application of an oriented electron-density likelihood target function to identify likely $C^{\alpha}$ positions. This function is applied both in the location of a few promising 'seed' positions in the map and to grow those initial $C^{\alpha}$ positions into extended chain fragments. Techniques for assembling the chain fragments into an initial chain trace are discussed.

## 1. Introduction

Automated building of atomic models of protein structures from electron density is an important element of a high-throughput structure-solution environment and a useful tool in a non-automated environment. Current automated building tools incorporate a range of ideas, some of which have been purpose-designed for automation and others that have been adopted from techniques which already exist in graphical model-building programs.

The approach to automated model building described here incorporates one new technique, the use of an oriented electron-density likelihood target function to identify likely $C^{\alpha}$ positions, along with a range of methods adapted from existing approaches. The resulting combination of methods is very simple and yet shows significant promise as the basis for a new automated model-building system. Before the new developments are described in detail, some other approaches which have been influential in this work will be discussed.

### 1.1. Graphical model-building tools

Much of the fundamental work on which current automated model building depends is drawn from the work of Jones and coworkers (*e.g.* Jones, 2004) on graphical tools for model building and in particular the *O* software. Two approaches are used. The first involves the calculation of a 'skeleton' of ridge-lines connecting peaks in the electron density. The skeleton is then interpreted in terms of $C^{\alpha}$ positions, which commonly occur near branch points in the skeleton (Jones *et al.*, 1991). The second involves the location of secondary-structure features, in particular helices and strands, by performing a six-dimensional rotation and translation search with an idealized fragment and evaluating the electron density at the atomic centres (Kleywegt & Jones, 1997). These features provide a starting point from which the rest of the protein chain may be traced.

doi:10.1107/S0907444906022116

Another important contribution by Jones and coworkers is the docking of the protein sequence to the main-chain trace (Zou & Jones, 1996). This step provides validation of the chain direction and is often necessary before completion of the main-chain trace because flexible loops connecting the more easily interpreted core regions of the protein may not be visible in the electron density. Zou & Jones (1996) score possible side-chain types by a combination of rotomer fitting and a real-space residual and then 'slide' the known sequence against the residue scores to find the most likely match.

Oldfield went on to develop graphical chain-tracing tools to perform assisted and automated building (Oldfield, 2002). Secondary-structure features are identified by geometrical analysis of the skeleton ridge-lines and these features are then automatically grown to model the loop regions of the molecule by automatic identification of branch points in the skeleton which extend the chain fragments. The resulting method stands out from subsequent automatic procedures because of its speed. Oldfield suggests that the procedure is limited in most cases to data at resolutions better than 4.0 Å resolution (Oldfield, 2003).

### 1.2. Non-graphical model-building tools

Automated electron-density interpretation by the identification of atoms with electron-density peaks has a long history in the field of small-molecule direct methods and has also been applied to macromolecules at high resolution (see, for example, Sheldrick *et al.*, 2001). The *ARP*/*wARP* package has extended this approach to work at successively lower resolutions (Morris *et al.*, 2002). At lower resolutions, atoms are not resolved and therefore individual atomic peaks disappear; however, it is still possible to construct (under-determined) atomic models which account for the observed data. Morris *et al.* (2002) apply information about protein geometry to select plausible C$^\alpha$ atoms from these redundant models and then conduct an exhaustive search of possible routes through the resulting list of candidate C$^\alpha$ atoms to identify a best trace. This approach works reliably when data is available to 2.5 Å resolution and in some cases to worse resolutions (Cohen *et al.*, 2004). Automated sequence docking and refinement lead to a near-complete model in many cases.

Another approach to the problem of limited resolution is to search for structures larger than atoms. The template-convolution method (Kleywegt & Jones, 1997) is an example of this and inspired the Fourier-based *FFFear* method (Cowtan, 1998), later applied as an electron-density-based likelihood function (Cowtan, 2001) for locating secondary-structure features and larger domains. A similar approach was later adopted by Terwilliger (2001) for the location of secondary-structure elements, implemented in the *RESOLVE* phase-improvement and model-building software. The secondary-structure elements may then be grown and joined to complete the structure by adding residues in conformations consistent with geometrical constraints (Terwilliger, 2003). One particularly powerful technique employed in *RESOLVE* is the building of two additional residues at a time, with the best combined electron-density fit for the pair of residues determining the final position for the first of the two residues. This 'look-ahead' approach is more reliable than building a single residue on the basis of density alone.

The *CAPRA* software of Ioerger & Sacchettini (2002) has some significant parallels with the current work in that it uses pattern-recognition techniques to identify likely C$^\alpha$ positions in the electron-density map. An electron-density skeleton is calculated and orientation-invariant features of the electron density in a 4 Å sphere about a candidate point are processed using a neural network to identify which points on the skeleton are most likely to represent C$^\alpha$ positions. The chain is then traced by selecting connected candidate positions using the scores and geometrical constraints. This approach is effective at 2.8 Å resolution or better (Ioerger & Sacchettini, 2002).

## 2. Method

### 2.1. Overview

The approach to chain tracing described here is built on the idea of locating likely C$^\alpha$ positions and extending these into a chain. The first step resembles the *CAPRA* approach, but with one very significant difference: *CAPRA* locates likely C$^\alpha$ positions on the basis of orientation-independent density features, whereas *Buccaneer* uses an orientation-dependent measure. This has two benefits.

(i) The result of the search is a list of oriented amino-acid groups, rather than just positions. This provides additional directional information to assist the process of assembling the amino acids into chains.

(ii) Since orientation-dependent information is not being excluded from the identification of the C$^\alpha$ positions, the target function may be more sensitive. However, this is offset by limitations imposed upon the target function by the search algorithm.

One other difference between this and some previous implementations is that the whole calculation takes place in 'crystal' space, in which the space-group symmetry and cell repeat are implicit. As a result, there is no need to 'locate' the molecule in the cell before building is attempted, since all symmetry copies of any atom are by definition built simultaneously. The implementation in crystal space is a benefit of the use of the 'Clipper' crystallographic libraries (Cowtan, 2003).

The discussion here is given in terms of locating C$^\alpha$ groups, where a C$^\alpha$ group is considered to include the C$^\alpha$ atom, the bonded N, C and H atoms and the C$^\beta$ atom when present, these atoms forming a rigid group. However, exactly the same techniques are equally applicable to the location of planar peptide groups (C$^\alpha$, C, O, N, C$^\alpha$) or of nucleotides for the tracing of DNA and RNA.

Likely C$^\alpha$ positions will be located using a density-likelihood function, which will score possible positions and orientations in the electron-density map in a six-dimensional search. Each possible configuration will be scored according to how well the density features reproduce the density features of real $C^{\alpha}$ groups in a simulated electron-density map for a known structure.

A vital element of the calculation is the preparation of this simulated electron-density map. For the likelihood target function to be valid, the simulated electron density must be on the same scale, represent broadly similar thermal motion and have the same size and type of noise features as the electron-density map to be interpreted. This simulation process is a complex calculation in itself and is the basis for both the *Pirate* statistical phase-improvement software and the *Buccaneer* chain-tracing software and will be described in another paper (Cowtan, 2006).

The whole calculation can therefore be described in terms of four steps.

(i) Finding initial $C^{\alpha}$ 'seed' positions.

(ii) Growing 'seed' positions into chain fragments.

(iii) Joining chain fragments into chains.

(iv) Pruning of clashing chains.

Each of these will be discussed in more detail in the following sections.

### 2.2. Finding initial $C^{\alpha}$ 'seed' positions

The aim of the finding step is to locate a few very probable $C^{\alpha}$ positions in the electron-density map for use as seed points from which longer chains will be grown. This process is related to the location of $\alpha$-helices in the *RESOLVE* model-building software. The location of the $C^{\alpha}$ 'seed' positions requires a six-dimensional search in both position and orientation. The accomplishment of this search in reasonable time places some constraints on the type of target function which can be used. The approach adopted here is to use a target function for which the translational search may be achieved with a few fast Fourier transforms (FFTs) and to perform an FFT-based translation search for every possible orientation of the density target.

A suitable target function for this type of calculation is the weighted density agreement function described by Cowtan (1998), which described how a localized density agreement function could be efficiently calculated using FFTs. This work was extended (Cowtan, 2001) by the use of the same function to calculate a density likelihood function in the presence of noise. The analysis here follows the same approach, with the exception that the term introduced there to account for the noise in the target map is replaced by the use of a simulated noisy map in the construction of the likelihood target function.

The search function is constructed using Bayes theorem,

$$P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model})P(\text{model})}{P(\text{data})}. \quad (1)$$

In this case the data are the electron-density map and the model is a specific placement of the search density for the $C^{\alpha}$ group. Let $F$ represent the case that the electron density arises from a correctly positioned and oriented $C^{\alpha}$ group and $\overline{F}$ represent the case that the electron density arises from any other source (*i.e.* an incorrectly positioned $C^{\alpha}$ group or density arising from a completely different source). Then, the probability of a correctly positioned $C^{\alpha}$ group given an individual density value from the map is given by

$$P[F|\rho(\mathbf{x})] = \frac{P[\rho(\mathbf{x})|F]P(F)}{P[\rho(\mathbf{x})]}. \quad (2)$$

$P[\rho(\mathbf{x})]$ is the probability of the 'observed' map density at $\mathbf{x}$. It may be calculated as a marginal distribution of $P[\rho(\mathbf{x}), C]$, $C \in (F, \overline{F})$, *i.e.*

$$\begin{aligned} P[\rho(\mathbf{x})] &= P[\rho(\mathbf{x}), F] + P[\rho(\mathbf{x}), \overline{F}] \\ &= P[\rho(\mathbf{x})|F]P(F) + P[\rho(\mathbf{x})|\overline{F}]P(\overline{F}). \end{aligned} \quad (3)$$

It is more likely that an electron-density value will arise from any other source than from a correctly positioned and oriented $C^{\alpha}$ group, therefore $P(\overline{F})$ will dominate over $P(F)$. Neglecting both this first term and also the prior probabilities $P(F)$ and $P(\overline{F})$, which will be assumed to be uniform, (2) becomes

$$P[F|\rho(\mathbf{x})] \simeq \frac{P[\rho(\mathbf{x})|F]}{P[\rho(\mathbf{x})|\overline{F}]}. \quad (4)$$

There are a number of $C^{\alpha}$ groups in the reference map, each represented by a different pattern of electron-density values in the region around it. For each position in the region of a standard $C^{\alpha}$ group placed at the origin, a distribution of electron densities will be calculated based on the different values appearing in that position relative to the $C^{\alpha}$ atom when considering all the $C^{\alpha}$ groups in the reference structure. The probability of a particular electron-density value given a particular correctly positioned $C^{\alpha}$ group will be approximated by a Gaussian whose mean is the expected electron density and whose variance is given by the variance of the distribution of densities at that position when calculated over all the $C^{\alpha}$ atoms in the simulated reference map. These will be termed $\rho_{\rm frag}(x)$ and $\sigma_{\rm frag}(x)^2$.

The probability of an observed density value arising from a correctly positioned $C^{\alpha}$ group is then

$$P[\rho(\mathbf{x})|F] \propto \exp\left\{-\frac{[\rho(\mathbf{x}) - \rho_{\rm frag}(\mathbf{x}')]^2}{2\sigma_{\rm frag}(\mathbf{x}')^2}\right\}, \quad (5)$$

where $\mathbf{x}'$ is the coordinate relative to the $C^{\alpha}$ which maps to the point $\mathbf{x}$ in the map under the current translation and orientation of the $C^{\alpha}$ group.

The probability of an observed density arising from some other source than a correctly positioned $C^{\alpha}$ group is estimated from the simulated reference density map by examining the density in regions not correlated with $C^{\alpha}$ features (but avoiding solvent). If the mean and variance of such uncorrelated density are given by $\rho_{\rm rand}$ and $\sigma_{\rm rand}$, then

$$P[\rho(\mathbf{x})|\overline{F}] \propto \exp\left\{-\frac{[\rho(\mathbf{x}) - \rho_{\rm rand}]^2}{2\sigma_{\rm rand}^2}\right\}. \quad (6)$$

Substituting these expressions in (4) and discarding the constant terms gives

$$P[F|\rho(\mathbf{x})] \propto \frac{\exp\left\{-\dfrac{[\rho(\mathbf{x})-\rho_{\rm frag}(\mathbf{x}')]^2}{2\sigma_{\rm frag}(\mathbf{x}')^2}\right\}}{\exp-\left\{\dfrac{[\rho(\mathbf{x})-\rho_{\rm rand}]^2}{2\sigma_{\rm rand}^2}\right\}}$$
$$\propto \exp\left\{-\frac{[\rho(\mathbf{x})-\rho''(\mathbf{x}')]^2}{2\sigma''(\mathbf{x}')^2}\right\}, \qquad (7)$$

where

$$\rho''(\mathbf{x}') = \frac{\sigma_{\rm rand}^2\rho_{\rm frag}(\mathbf{x}') - \sigma_{\rm frag}(\mathbf{x}')^2\rho_{\rm rand}}{\sigma_{\rm rand}^2 - \sigma_{\rm frag}(\mathbf{x}')^2}$$

and

$$\sigma''(\mathbf{x}')^2 = \frac{\sigma_{\rm frag}(\mathbf{x}')^2\sigma_{\rm rand}^2}{\sigma_{\rm rand}^2 - \sigma_{\rm frag}(\mathbf{x}')^2}.$$

Finally, the probability indications for the presence of a C$^\alpha$ group on the basis of each individual density value in the map are combined to give an overall indication of the probability of a the C$^\alpha$ group being present with the given translation and orientation,

$$P(F|\rho) = \prod_x P[F|\rho(\mathbf{x})]. \qquad (8)$$

It is more convenient to calculate the logarithm of this expression,

$$\log P(F|\rho) = \sum_x \log P[F|\rho(\mathbf{x})]$$
$$= \sum_x -\left\{\frac{[\rho(\mathbf{x})-\rho''(\mathbf{x}')]^2}{2\sigma''(\mathbf{x}')^2}\right\} + c. \qquad (9)$$

The resulting function may be efficiently calculated for a single orientation as a function of position in the cell using an FFT approach. Let the translation search function, which gives the agreement between the C$^\alpha$ group density (in the current orientation) and the electron density as a function of translation, be called $t(x)$. As a simplification, let $\mu''(x) = 1/[2\sigma''(\mathbf{x}')^2]$. The search function may then be written as

$$t(x) = \sum_{x'} \mu''(x')[\rho''(x') - \rho(x'-x)]^2 \qquad (10)$$
$$= \sum_{x'} \mu''(x')\rho''(x')^2 - 2\mu''(x')\rho''(x')\rho(x'-x)$$
$$+ \mu''(x')\rho(x'-x)^2.$$

Note that in the expansion the first term is independent of $x$ and so is only calculated once, whereas the second two terms are convolutions and may therefore be efficiently calculated in reciprocal space as follows,

$$t(x) = \sum_y \mu''(y)\rho''(y)^2 + (1/V)\mathcal{F}\{\mathcal{F}^{-1}[\mu''(x)]\mathcal{F}^{-1}[\rho(x)^2]^*$$
$$- 2\mathcal{F}^{-1}[\mu''(x)\rho''(x)]\mathcal{F}^{-1}[\rho(x)]^*\}, \qquad (11)$$

where $\mathcal{F}$ represents the Fourier transform, $\mathcal{F}^{-1}$ the inverse Fourier transform and * complex conjugation. If the Fourier coefficients of the density and squared density are precalculated, then the translation function may be calculated by three Fast Fourier Transforms (FFTs) per orientation. Since the C$^\alpha$ group has no symmetry, the FFTs must be performed in $P1$.

The electron-density target function for the location of a C$^\alpha$ group is determined by considering the electron density within a 4 Å sphere around each C$^\alpha$ in the simulated reference map. This radius was initially inspired by *CAPRA* (Ioerger & Sacchettini, 2002) and subsequent testing proved it to be a good choice for this method too. The density means and variances are calculated on a fine (0.5 Å) orthogonal grid.

Fig. 1 shows the mean and variance density for a typical search model. The mean density shows the expected pattern of density around the atoms of the C$^\alpha$ group, with weaker C$^\beta$ density and bulges in likely C$^\gamma$ directions. However, the variance density shows more interesting features, in particular that the most conserved density is concentrated not only at the main-chain atomic sites, but also at low-density positions between the atoms. This highlights the power of the *FFFear* search function to select both high- and low-density positions. Note also the hollows around the C$^\beta$ at common C$^\gamma$ sites.

The six-dimensional search is performed over every possible translation and orientation of the C$^\alpha$ group and the highest scoring matches are assumed to be correct. Each position and orientation is then refined by a simplex algorithm search and then stored as a 'seed' position for chain growth. As a default, one seed position is stored for every five residues expected in the final model, although this parameter is not very critical.

### 2.3. Growing 'seed' positions into chain fragments

The 'seed' C$^\alpha$ groups are grown into chains by adding additional C$^\alpha$ groups both before and after the seed group in positions which optimize the log-likelihood fit to density for the new group while not disobeying the constraints of the Ramachandran plot. The same log-likelihood function is used for evaluating C$^\alpha$ positions added by growth as for the initial finding stage; however, it is now evaluated in real space for each candidate position and orientation instead of using the FFT approach.

For the purposes of this calculation, the Ramachandran plot classified by residue type and contoured at two levels: a frequency of >0.0005 rad$^{-2}$ describing an 'allowed' region and a frequency of >0.01 rad$^{-2}$ describing a 'favoured' region, using the imprecise but commonplace terminology.

The growing process proceeds as follows. To grow a single residue in the forward direction, a search is conducted over the 'allowed' values of the Ramachandran angle $\psi$ for the current residue and $\varphi$ for the next residue. The angles are searched with a uniform angle step of 20°, rejecting any $\psi$ values forbidden by the Ramachandran plot. (When building the first new residue in a chain, no information is available concerning the first $\varphi$.) Next, a second residue is built, using a coarser angle search of 30°, but again applying Ramachandran constraints. The best combined log-likelihood score for the two residues is used to select the position of the first residue. The second residue is discarded, having served its sole purpose in validating the position of the first.

This two-residue look-ahead approach is similar to that of Terwilliger (2003). The Ramachandran data used here is from the 'Top 500' structures database of Lovell *et al.* (2003). For the first residue any 'allowed' conformation for any residue type is allowed, whereas for the second only 'favoured' conformations for non-Gly residues are allowed.

Building in the reverse direction occurs in exactly the same manner, except for the reversal of the Ramachandran angles.

A cutoff threshold for the log-likelihood function is required to determine when to stop growing the chain in either direction. This cutoff is established through an effective *ad hoc* procedure: for each of the initial seed points, three residues are grown in a forward direction. It is then assumed that 90% of the resulting terminal C$^\alpha$ atoms will be correct. The scores for the terminal C$^\alpha$ atoms are sorted and the value separating the worst 10% from the remainder is used as the cutoff. This is a crude *ad hoc* criterion which provides only a rudimentary coupling to the quality of the map; however, in practice it is effective in providing useful fragments for processing by the subsequent stages.

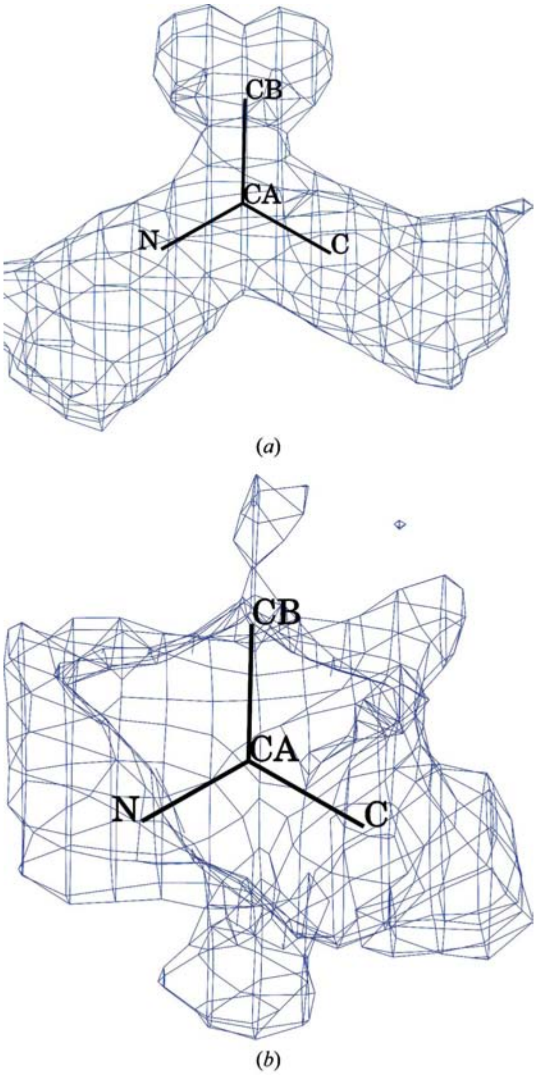

**Figure 1**
Representation of *Buccaneer* target function for a C$^\alpha$ group showing regions of (*a*) high mean density and (*b*) low variance (*i.e.* strongly conserved) density. Figures generated using *CCP4MG* (Potterton *et al.*, 2002).

Several optimizations are used to improve the performance of this approach. For the full angle search, the log-likelihood function is approximated by only using a subset of the grid points in the calculation. Since each calculation requires a density interpolation from the target map, this saves a significant amount of time. The best 50 conformations of the first residue are used to build the second residue and the best 30 combined scores are then rescored using all the points in the log-likelihood function. Finally, the Ramachandran angles for the best solution are refined using a simplex algorithm search.

### 2.4. Joining chain fragments

At this stage of the model-building process the model consists of many overlapped chain segments which may or may not be consistent with one another. From these, a single consistent model must be constructed either for visual assessment or for use in conventional refinement programs. This is achieved in two steps: joining of consistent fragments, followed by pruning of inconsistent fragments.

The joining stage merges overlapped fragments wherever this is possible and makes some initial selections between fragments when multiple possible merges are possible. The calculation proceeds as follows.

Firstly, every chain segment is split into a series of overlapping fragments, each containing three residues; *i.e.* a chain of $n$ residues is split into $n - 2$ fragments of three residues, with each tri-residue fragment overlapping its neighbours by two residues.

Next, multiple traces of the same chain segment are merged by combining any pair of tri-residues for which all three C$^\alpha$ atoms match to within 2.0 Å. The combination is achieved by averaging all the coordinates of each main-chain atom of each tri-residue. This leads to a model in which multiple consistent traces of the same chain segment have been removed.

Next, the tri-residues are examined to see how they can be reassembled into chains. A search is conducted over every pair of tri-residues to identify any pair for which the second and third C$^\alpha$ atoms of the first tri-residue match the first and second C$^\alpha$ atoms of the second tri-residue, to within 2.0 Å. Every such pair is marked as a potential join.

A problem arises when a single tri-residue joins to several possible precursors or successors. At this point a decision must be made about the correct routing of the chain. Following the example of Cohen *et al.* (2004), the different possible routings of the chain are considered and that which yields the longest non-looped chain is assumed to be correct. An assumption here is that tracing is more likely to skip residues than to insert extra residues.

Identification of the longest possible trace through a list of multiply linked tri-residue fragments is a problem of finding the longest path through a directed graph. This is a simple computational problem which is conventionally solved by a dynamic programming technique called 'critical path analysis'.

However, the conventional implementations must be adjusted to deal with the possibility of looped chains in the hypothetical trace. The implementation is therefore as follows.

Each remaining tri-residue is considered to be a numbered node in a directed graph which may have zero or more predecessors and zero or more successors.

(i) All nodes are labeled with a pair of integers, a 'path length' and a 'predecessor pointer', both of which are set to the dummy value −1.

(ii) All nodes which are not the successor of any other node are therefore chain starts; they have their 'path length' set to 0, and are added to a queue of nodes to be considered.

(iii) While there remains a node on the list to be considered, the first node on the list is examined. All the possible successor nodes are considered in turn. For each possible successor, the following tests are conducted:

(1) A check is made by searching the predecessors of this node to ensure that the successor does not already appear in the trace up to this point, and therefore will not form a loop.

(2) The value of the 'path length' for the current node is compared against the value for the successor node: If the path through the current node to the successor is longer (*i.e.* the 'path length' is greater), then the successor node is updated by the following steps:

(*a*) The 'path length' of the successor node is set to the 'path length' of the current node plus one.

(*b*) The 'predecessor pointer' of the successor node is set to the number of the current node, so that the chain may be traced backwards.

(*c*) The successor node is added to the end of the queue of nodes to be processed.

(iv) When no more nodes remain to be considered, the longest chain is then extracted by searching for the node with the greatest 'path length', and then tracing backwards through its predecessors until the start of the chain is found. This is the longest contiguous chain trace in the map. These nodes may be removed, and the calculation repeated with the remaining nodes until no more segments of more than 5 residues are found.

The calculation is very fast and leads to the longest possible chain trace through the given fragments in the case where there are no loops. In the case of looped chains, the results are not guaranteed to be optimal; however, they are usually optimal or near-optimal.

Once a set of chains have been traced through the tri-residues, the final atomic coordinates are assembled from the coordinates of the successive tri-residues in the chain. A weighted combination of all the overlapped atoms is used in order to achieve a smooth transition from one tri-residue to the next and thus maintain connectivity in the merged model. The weighting of each tri-residue decreases linearly from the central C$^\alpha$ towards its extremities.

### 2.5. Pruning of clashing chain fragments

The previous step will have merged all consistent chain fragments, selecting a single path where fragments branch in different directions. There still remains the problem of inconsistent chain fragments, including two common cases: firstly the case where two fragments trace the same chain in opposite directions and secondly the case where chains cross or clash without any commonality.

Both of these cases are handled by a simple pruning step. Each chain is compared against every other chain, noting any cases where any pair of C$^\alpha$ atoms approach to within 2.0 Å. Any clashing C$^\alpha$ atoms are removed from the shorter chain. Any segments of the shorter chain which are less than five residues in length are then deleted.

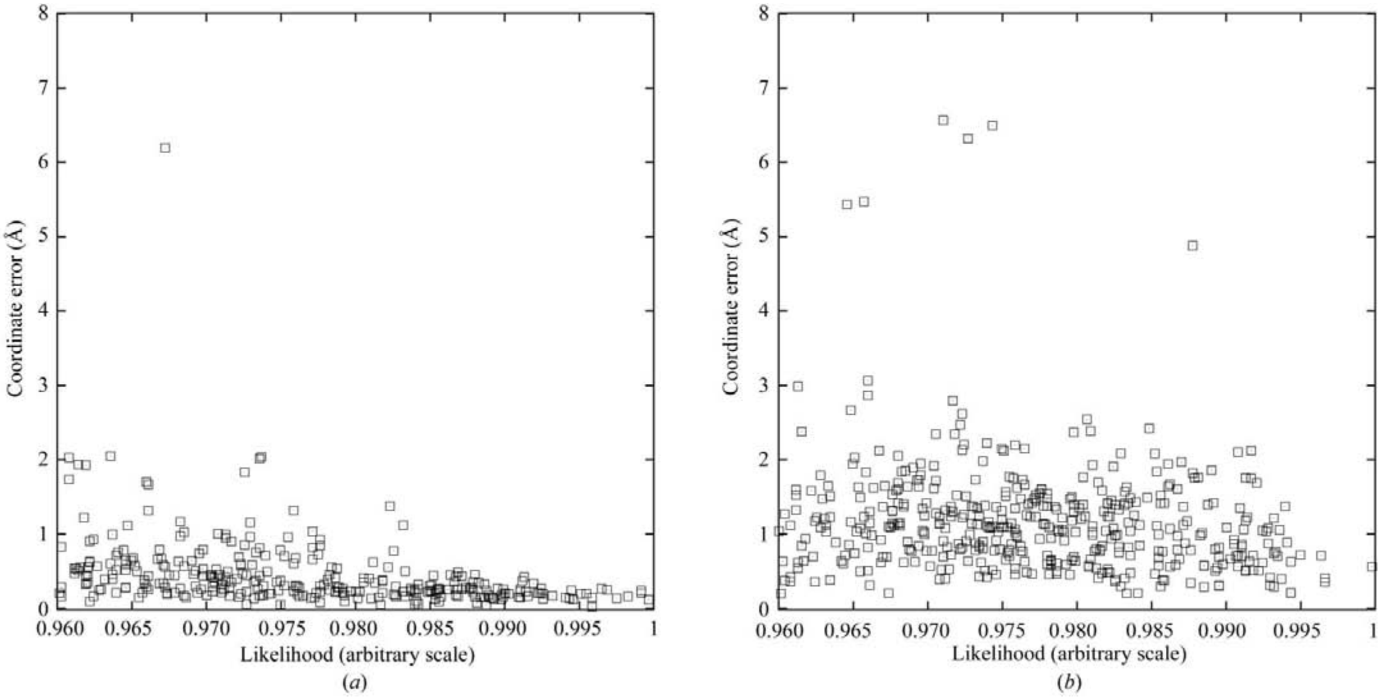

**Figure 2**
Distance to the nearest true C$^\alpha$ as a function of *Buccaneer* likelihood score (on an arbitrary scale; larger is better) for (*a*) a good high-resolution map (PDB code 1z92) and (*b*) a poorer low-resolution map (PDB code 1vrb). Results are plotted for the best C$^\alpha$s found by the six-dimensional *FFFear* search in each case. The likelihood score is a reliable indicator of C$^\alpha$ position when density is good, but picks a mixture of good and bad positions when density is poor.

One aim of this approach is to encourage chain tracing in the right direction, under the assumption that reversed chain traces will tend to be shorter than forward traces owing to the use of the Ramachandran constraint in the chain-growing step. (This is of course only true in loop regions since helices and strands may be traced in either direction without violating Ramachandran constraints; however, in practice the approach is effective.)

### 2.6. Results

The procedure has been implemented in a software package called *Buccaneer* using the Clipper crystallographic libraries (Cowtan, 2003). The implementation is extremely simple, involving about 2000 lines of C++ code. The software as outlined here is incomplete in comparison to other software in the field: no refinement of the model or recycling takes place to complete the model and no sequence docking or side-chain building is performed. As a result, the software is not comparable to competing methods at this point. However, some initial results obtained using real data can provide some indications to the capabilities of the method.

The procedure was tested using 58 structures from the Joint Center for Structural Genomics (JCSG) data archive (Joint Center for Structural Genomics, 2006). This is a database of structures solved by largely automated methods. The chosen structures were solved using experimental phasing. For each structure, the JCSG software pursued multiple phasing paths using different software and parameters. A single initial phasing set was chosen for each structure by automatically selecting a structure on the basis of the statistics of the electron-density map. The selection criteria were crude, however, and so in some cases poor, low-resolution or even wrong phasing has been selected; all of these were kept as a means to test the behaviour of the software.

The selected set of experimental phases for each structure was then subjected to three cycles of phase improvement using the *Pirate* software (Cowtan, 2000). The resulting phases were used as a starting point for the *Buccaneer* chain-tracing calculation. The calculation for a single structure took between 2 and 30 min on a 2.4 GHz PC, depending on the volume of the asymmetric unit and resolution.

The quality of the starting data is described in terms of the data resolution and of the *E*-map correlation with the map from the final refined structure, the latter being a measure of phase error weighted by *E* value and figure of merit.

As an initial test, the performance of the '$C^\alpha$-finding' step was examined. The six-dimensional *FFFear* search was used to identify the most probable $C^\alpha$ positions in both a good 2.0 Å map (PDB code 1z82) and a poorer low-resolution map (PDB code 1vrb). The likelihood score for the best matches was compared against the distance in angstroms from the candidate position to the nearest $C^\alpha$ in the solved structure. The results are shown in Fig. 2. Note that with a good map the likelihood function accurately identifies $C^\alpha$ positions. In the poorer map there are a number of wrong positions identified along with the correct ones, although 75% of the candidate positions are still within 1.5 Å of a true $C^\alpha$ position. At lower resolutions, later stages in the chain-tracing calculation will have to remove fragments traced from incorrect candidates.

**Table 1**
Results of applying *Buccaneer* chain tracing to density modified phases for 58 JCSG data sets.

The columns give the deposition code, number of residues in the deposited model, resolution and *E*-map correlation for the starting density-modified phases and the completeness and accuracy of the resulting chain trace.

| PDB code | No. of residues | Resolution (Å) | *E*-map correlation | Completeness (%) | Accuracy (%) |
|---|---|---|---|---|---|
| 1vr8 | 135 | 1.75 | 0.912 | 52 | 100 |
| 1vqs | 551 | 1.80 | 0.807 | 60 | 100 |
| 1vp8 | 190 | 1.53 | 0.916 | 57 | 100 |
| 1vmg | 83 | 1.46 | 0.808 | 85 | 100 |
| 1vlo | 364 | 1.70 | 0.798 | 68 | 100 |
| 1vku | 85 | 1.94 | 0.718 | 47 | 100 |
| 1vk4 | 283 | 1.91 | 0.882 | 63 | 100 |
| 1vr5 | 1072 | 1.73 | 0.805 | 81 | 99 |
| 1vme | 802 | 1.80 | 0.923 | 75 | 99 |
| 1z82 | 624 | 2.00 | 0.786 | 90 | 98 |
| 1vlc | 362 | 2.46 | 0.774 | 75 | 98 |
| 1vqz | 330 | 1.99 | 0.686 | 78 | 97 |
| 1vpm | 460 | 1.66 | 0.828 | 86 | 97 |
| 1vp4 | 836 | 1.82 | 0.671 | 74 | 97 |
| 1vkm | 1752 | 2.60 | 0.800 | 79 | 97 |
| 1vkh | 513 | 2.30 | 0.790 | 88 | 97 |
| 1vk2 | 190 | 2.10 | 0.786 | 73 | 97 |
| 1vmf | 407 | 1.73 | 0.695 | 86 | 96 |
| 1vr0 | 708 | 2.49 | 0.807 | 72 | 95 |
| 1vmi | 329 | 2.32 | 0.646 | 83 | 95 |
| 1vpb | 437 | 1.80 | 0.883 | 78 | 94 |
| 1vp7 | 417 | 3.00 | 0.741 | 90 | 94 |
| 1vk8 | 373 | 2.00 | 0.705 | 58 | 93 |
| 1vjf | 167 | 2.60 | 0.892 | 88 | 93 |
| 1vpy | 251 | 2.40 | 0.681 | 85 | 92 |
| 1vlm | 414 | 2.20 | 0.638 | 78 | 92 |
| 1vl5 | 906 | 1.85 | 0.724 | 78 | 92 |
| 1vjx | 149 | 2.30 | 0.620 | 92 | 92 |
| 1vqy | 847 | 2.40 | 0.687 | 74 | 91 |
| 1vkz | 782 | 2.90 | 0.799 | 71 | 91 |
| 1vkn | 1351 | 2.45 | 0.871 | 83 | 91 |
| 1z85 | 428 | 2.12 | 0.621 | 84 | 90 |
| 1vjz | 325 | 2.50 | 0.792 | 85 | 89 |
| 1o6a | 168 | 1.90 | 0.791 | 90 | 88 |
| 1vli | 358 | 2.38 | 0.700 | 81 | 87 |
| 1vl6 | 1486 | 2.80 | 0.794 | 75 | 87 |
| 1vpz | 113 | 2.05 | 0.701 | 69 | 85 |
| 1vk3 | 586 | 2.80 | 0.630 | 71 | 84 |
| 1vjv | 367 | 2.65 | 0.900 | 70 | 84 |
| 1vr3 | 179 | 2.06 | 0.789 | 92 | 82 |
| 1vlu | 792 | 3.00 | 0.644 | 76 | 81 |
| 1vrb | 1224 | 3.20 | 0.628 | 59 | 80 |
| 1vl4 | 856 | 2.32 | 0.735 | 67 | 80 |
| 1vll | 642 | 3.00 | 0.779 | 66 | 76 |
| 1vkd | 1956 | 2.60 | 0.705 | 62 | 76 |
| 1vky | 563 | 3.00 | 0.664 | 71 | 72 |
| 1vk9 | 147 | 2.70 | 0.713 | 89 | 67 |
| 1vjn | 383 | 3.00 | 0.857 | 63 | 66 |
| 1vkb | 147 | 1.90 | 0.704 | 69 | 64 |
| 1zej | 282 | 2.00 | 0.751 | 84 | 53 |
| 1vjr | 261 | 2.50 | 0.592 | 86 | 43 |
| 1vqr | 1101 | 3.00 | 0.528 | 27 | 35 |
| 1vpg | 301 | 2.10 | 0.450 | 3 | 23 |
| 1vpj | 356 | 1.74 | 0.401 | 20 | 16 |
| 1vl0 | 842 | 2.50 | 0.222 | 14 | 15 |
| 1vkw | 217 | 2.00 | 0.074 | 2 | 4 |
| 1vr9 | 242 | 2.00 | 0.042 | 0 | 1 |
| 1vjo | 377 | 2.00 | 0.018 | 2 | 1 |

Next, the whole procedure was used to trace connected chains. The quality of the *Buccaneer* model is described in terms of the proportion of the known structure which was correctly built (*i.e.* completeness) and the proportion of the built model which was correct (*i.e.* accuracy). These were calculated by counting the proportion of real C$^\alpha$ atoms correctly built and the proportion of built C$^\alpha$ which were correct. For the purposes of this analysis, a correctly built C$^\alpha$ is one which is within 1.9 Å of a true C$^\alpha$ position in the known structure and has a neighbour which is in turn within 1.9 Å of a neighbouring C$^\alpha$ in the known structure. (1.9 Å was chosen as half the distance separating two C$^\alpha$ atoms, a registration error of up to half a residue.)

Table 1 describes the results for the 58 test structures in terms of number of residues, the quality measures of the

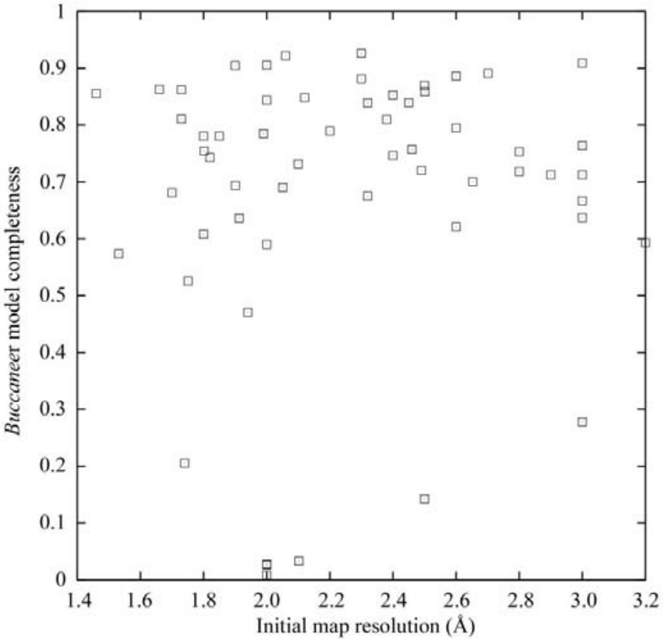

**Figure 3**
Completeness of the *Buccaneer* models for 58 JCSG test structures as a function of resolution.

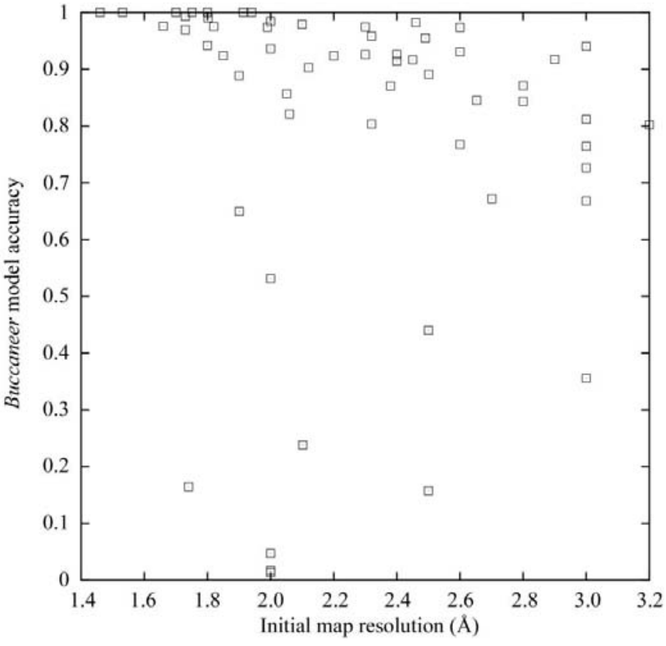

**Figure 4**
Accuracy of the *Buccaneer* models for 58 JCSG test structures as a function of resolution.

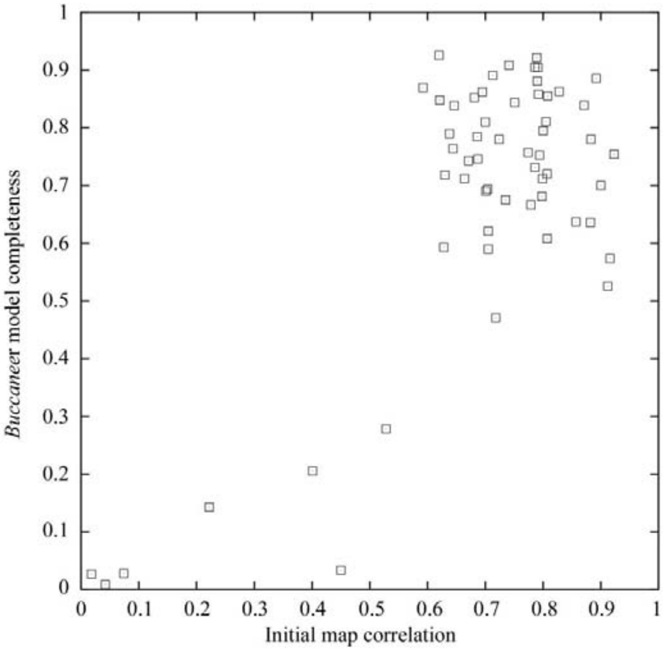

**Figure 5**
Completeness of the *Buccaneer* models for 58 JCSG test structures as a function of the quality of the initial phases (*E*-map correlation).

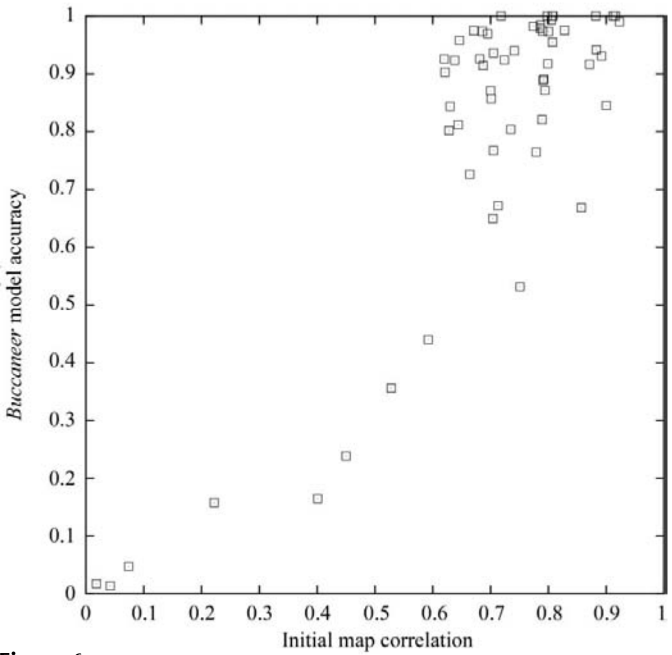

**Figure 6**
Accuracy of the *Buccaneer* models for 58 JCSG test structures as a function of the quality of the initial phases (*E*-map correlation).

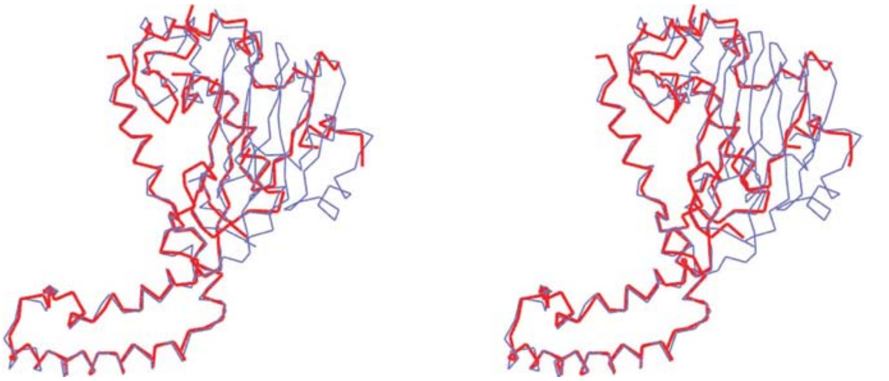

**Figure 7**
*Buccaneer* trace of the *A* subunit of 1vrb at 3.2 Å. The true structure is shown in thin blue lines and the *Buccaneer* trace in thicker red lines.

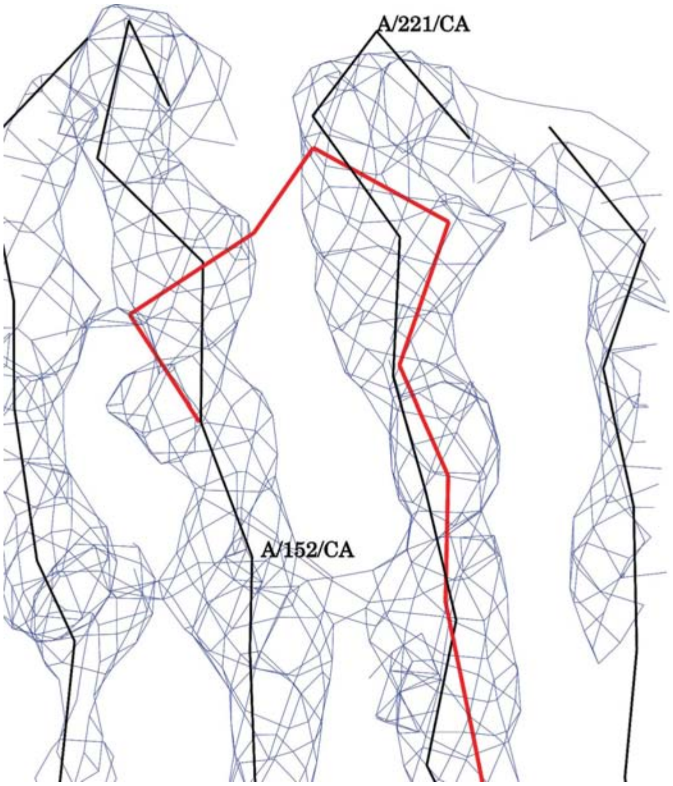

**Figure 8**
*Buccaneer* tracing error in the *A* subunit of 1vrb at 3.2 Å. The true structure is shown in thin black lines and the *Buccaneer* trace in thicker red lines.

starting data and the quality measures for the *Buccaneer* chain trace. The same data are visualized in Figs. 3, 4, 5 and 6.

Completeness varies between 0 and 92% and accuracy between 0 and 100%. Note that neither completeness or accuracy vary strongly as a function of the resolution of the starting data. There is a slight drop in accuracy for the lowest resolution models, but completeness remains consistent. The method appears to be usable at least to the 3.2 Å low-resolution limit of the data available for these tests.

However, completeness and accuracy are strongly related to the quality of the initial phases. An initial *E*-map correlation of less than 0.6 leads to a poor model. From these results, it can be concluded that the method is not strongly sensitive to the data resolution, but is sensitive to the quality of the phases. Thus, the method appears to be complementary to *ARP*/*wARP* (Cohen *et al.*, 2004), which is more sensitive to the data resolution but can give results with quite poor phases.

For the purposes of automated model building, it is common for a first model to be incomplete and to be extended in a recycling process with refinement and map calculation. The accuracy of the initial model is therefore probably more important than its completeness.

The model for the lowest resolution data set in the test, 1vrb (Joint Center For Structural Genomics, unpublished work), shows some interesting features. One subunit of the *Buccaneer* model from the 3.2 Å experimental phasing data set is shown in Fig. 7. Note that helical regions of the molecule have been well traced; even at low resolution the precision of atom placement is high, often within 0.3 Å. This is to be expected since helical conformations are more common and more uniform than other conformations and so contribute more strongly to the likelihood density target. Non-helical regions are much more variable and the precision of the chain traces is accordingly much lower. This suggests a future approach involving the use of different density targets for growth in different regions of the Ramachandran plot.

Other differences between the *Buccaneer* and final models are worth noting. Fig. 8 shows a typical auto-tracing error where the chain trace has jumped between strands by means of side-chain density. Note this also illustrates how *Buccaneer* building can be counter-intuitive in comparison to programs which seek high density.

Fig. 9 shows a place where *Buccaneer* has built a loop which was missing from the original model, along with the electron density. Comparison of the number of residues inserted against the sequence of the final model suggests that the trace is correct. The loop density is present, but the connectivity is only evident when the contour level is lowered. This case highlights another feature of *Buccaneer*. Since the likelihood target function keys on expected low-density features as well as high density features, *Buccaneer* is capable of building regions where the electron density is low. This feature has a cost: *Buccaneer* can also overinterpret solvent in terms of protein features in some cases. This can be seen in Table 1 in the cases where the completeness of the model is high but the accuracy is low (*e.g.* 1vjr), *i.e.* in addition to correctly tracing the protein region, *Buccaneer* has built protein chain in the solvent region. These cases can be trivially identified using the ‘Density fit analysis’ feature of the *Coot* model-building program (Emsley & Cowtan, 2004) and will be implemented in future developments of *Buccaneer*.

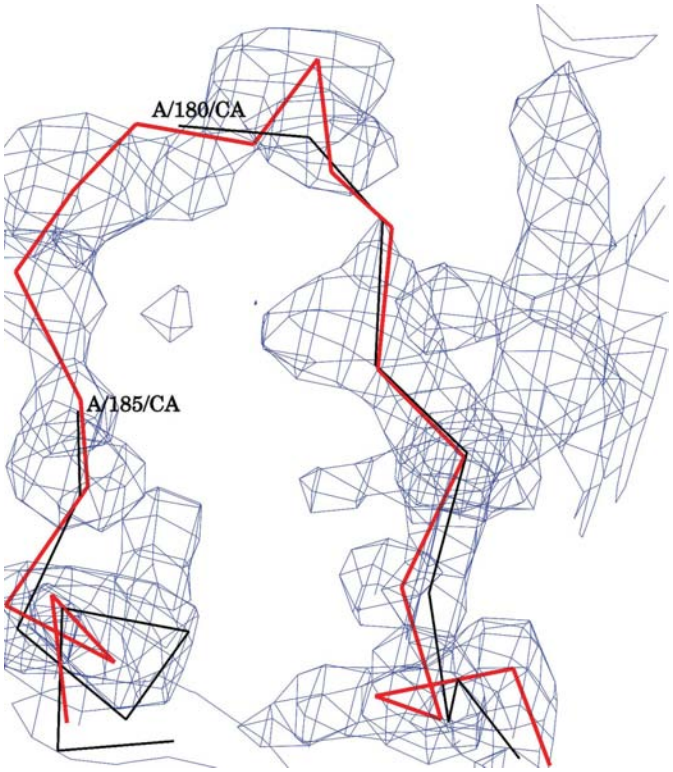

**Figure 9**
*Buccaneer* trace for a loop missing in the deposited structure. The true structure is shown in thin black lines and the *Buccaneer* trace in thicker red lines.

## 3. Conclusions

The chain-tracing approach described here is extremely simple, relying on the application of a single likelihood function in several different ways to trace protein main chains in experimentally phased electron-density maps. The method is reasonably fast, taking minutes to an hour, and can give a partial trace even at low resolutions (*i.e.* worse than 3.0 Å). However, the method is dependent on the quality of the initial experimental phasing and phase improvement.

The method as presented here is incomplete, lacking implementations for sequence docking, removal of incorrectly traced features, refinement of the resulting mode or recycling to model completion. However, the initial results suggest that the approach described here provides a suitable basis for future development.

The author would like to thank P. Emsley and E. Dodson for their helpful suggestions and the JCSG data archive for providing a source of well curated test data. This work was supported by The Royal Society under a University Research Fellowship.

## References

Cohen, S. X., Morris, R. J., Fernandez, F. J., Ben Jelloul, M., Kakaris, M., Parthasarathy, V., Lamzin, V. S., Kleywegt, G. J. & Perrakis, A. (2004). *Acta Cryst.* D**60**, 2222–2229.
Cowtan, K. D. (1998). *Acta Cryst.* D**54**, 750–756.
Cowtan, K. D. (2000). *Jnt CCP4/ESF–EACBM Newsl. Protein Crystallogr.* **38**, 7.
Cowtan, K. D. (2001). *Acta Cryst.* D**57**, 1435–1444.
Cowtan, K. D. (2003). *IUCr Comput. Commun. Newsl.* **2**, 4–9.
Cowtan, K. D. (2006). In preparation.
Emsley, P. & Cowtan, K. D. (2004). *Acta Cryst.* D**60**, 2126–2132.
Ioerger, T. R. & Sacchettini, J. C. (2002). *Acta Cryst.* D**58**, 2043–2054.
Joint Center for Structural Genomics (2006). *JCSG Data Archive.* http://www.jcsg.org/datasets-info.shtml.
Jones, T. A. (2004). *Acta Cryst.* D**60**, 2115–2125.
Jones, T. A., Zou, J.-Y., Cowan, S. W. & Kjeldgaard, M. (1991). *Acta Cryst.* A**47**, 110–119.
Kleywegt, G. J. & Jones, T. A. (1997). *Acta Cryst.* D**53**, 179–185.
Lovell, S., Davis, I., Adrendall, W., de Bakker, P., Word, J., Prisant, M., Richardson, J. & Richardson, D. (2003). *Proteins*, **50**, 437–450.
Morris, R. J., Perrakis, A. & Lamzin, V. S. (2002). *Acta Cryst.* D**58**, 968–975.
Oldfield, T. J. (2002). *Acta Cryst.* D**58**, 487–493.
Oldfield, T. J. (2003). *Acta Cryst.* D**59**, 483–491.
Potterton, E., McNicholas, S., Krissinel, E., Cowtan, K. & Noble, M. (2002). *Acta Cryst.* D**58**, 1955–1957.
Sheldrick, G., Hauptman, H., Weeks, C., Miller, R. & Usón, I. (2001). *International Tables for Crystallography*, Vol. *F*, edited by M. G. Rossmann & E. Arnold, pp. 333–351. Dordrecht: Kluwer Academic Publishers.
Terwilliger, T. C. (2001). *Acta Cryst.* D**57**, 1755–1762.
Terwilliger, T. C. (2003). *Acta Cryst.* D**59**, 38–44.
Zou, J.-Y. & Jones, T. A. (1996). *Acta Cryst.* D**52**, 833–841.