Universitat Ramon Llull

DOCTORAL THESIS

Title: ARCIMBOLDO, a supercomputing method for crystallographic *ab initio* protein structure solution below atomic resolution.

Presented by: Dayté Dayana Rodríguez Martínez

Centre: IQS School of Engineering

Department: Bioengineering

Directed by: Prof. Dr. Isabel Usón Finkenzeller

C. Claravall, 1-3
08022 Barcelona
Tel. 936 022 200
Fax 936 022 249
E-mail: urlsc@sec.url.es
www.url.es

Universitat Ramon Llull

TESIS DOCTORAL

Título: ARCIMBOLDO, un método de supercomputación para la determinación cristalográfica *ab initio* de estructuras de proteínas con resolución inferior a la atómica.

Realizada por Dayté Dayana Rodríguez Martínez

en el Centro: IQS Escuela de Ingeniería

y en el Departamento: Bioingeniería

Dirigida por: Prof. Dr. Isabel Usón Finkenzeller

C. Claravall, 1-3
08022 Barcelona
Tel. 936 022 200
Fax 936 022 249
E-mail: urlsc@sec.url.es
www.url.es

Universitat Ramon Llull

TESI DOCTORAL

Títol: ARCIMBOLDO, un mètode de supercomputació per a la determinació cristal·logràfica *ab initio* d'estructures de proteïnes amb resolució inferior a l'atòmica.

Realitzada per : Dayté Dayana Rodríguez Martínez

en el Centre: IQS Escola d'Enginyeria

i en el Departament: Bioenginyeria

Dirigida per Prof. Dr. Isabel Usón Finkenzeller

C. Claravall, 1-3
08022 Barcelona
Tel. 936 022 200
Fax 936 022 249
E-mail: urlsc@sec.url.es
www.url.es

Contents

List of Figures

List of Tables

Abbreviations

aa	**a**mino**a**cid
AU	**A**symmetric **U**nit
CC	**C**orrelation **C**oefficient
CPU	**C**entral **P**rocessing **U**nit
FCSCL	**F**undación **C**entro de **S**upercomputación de **C**astilla y **L**eón
FOM	**F**igures **O**f **M**erit
HTC	**H**igh **T**hroughput **C**omputing
IP	**I**nfrastructure **P**rovider
MAD	**M**ulti-wavelength **A**nomalous **D**ispersion
MIR	**M**ultiple **I**somorphous **R**eplacement
MIRAS	**M**ultiple **I**somorphous **R**eplacement plus **A**nomalous **S**cattering
MPE	**M**ean **P**hase **E**rror
MR	**M**olecular **R**eplacement
NFS	**N**etwork **F**ile **S**ystem
NIS	**N**etwork **I**nformation **S**ystem
NMR	**N**uclear **M**agnetic **R**esonance
PDB	**P**rotein **D**ata **B**ank
RaaS	**R**esults **a**s **a** **S**ervice
RIP	**R**adiation damage **I**nduced **P**hasing
rmsd	**r**oot **m**ean **s**quare **d**eviation
SP	**S**ervice **P**rovider
SU	**S**ervice **U**ser
SAD	**S**ingle-wavelength **A**nomalous **D**ispersion
SIR	**S**ingle **I**somorphous **R**eplacement
SIRAS	**S**ingle **I**somorphous **R**eplacement plus **A**nomalous **S**cattering
VO	**V**irtual **O**rganization
VRE	**V**irtual **R**esearch **E**nvironment

Dedicated to my daughter, my mother and my sister.

Chapter 1

Introduction

1.1 The phase problem

Crystallography is arguably the most powerful of all structural techniques. Through the determination of the electron density map, it is able to deliver an accurate three-dimensional view of the biomacromolecular structures. Albeit, in order to calculate electron densities

$$\rho(xyz) = \frac{1}{V}\sum_{h}\sum_{k}\sum_{l}\underbrace{|F(hkl)|}_{Amplitudes}\cos 2\Pi(hx+ky+lz-\underbrace{\phi(hkl)}_{Phases})$$

both magnitudes and phases of the x-ray diffraction maxima are required, but only the intensities are available from the diffraction experiment, hence, the density function cannot be calculate without retrieving the phases. This fundamental problem in crystallography is also known as the crystallographic phase problem [1].

1.1.1 Current methods to overcome the phase problem

In the case of small molecules -that is for chemical crystallography- the phase problem can usually be solved directly. In protein crystals diffraction is limited by their intrinsic nature while the number of variables required to determine the structure is much larger. The term resolution refers to the maximum resolution to which a macromolecular crystal diffracts, that is the minimum spacing between parallel planes of atoms from which diffraction can still be recorded. More closely spaced planes lead to reflections further from the center of the diffraction pattern. In the present work, atomic resolution will be used to design data collected to 1.2Å or beyond, high resolution to data of 2Å or better: the resolution to which 50% of the crystal structures deposited with the Protein Data Bank [2,3] diffracts. The mathematical complexity of the problem rules out an analytical solution, and it has to be reduced in order to establish a rough initial electron density map or model, what is known as phasing. In macromolecular crystallography there are

several methodologies to retrieve the missing phases involving both experimental and computational methods:

Experimental Phasing

It is based on using substructures of anomalous scatterers or heavy atoms, to provide reference phases for the full structure to be determined. Crystals can be modified to incorporate heavier elements,[4,5] so that differences induced by their presence can be determined. For crystals containing -or derivatized to contain- anomalous scatterers, diffraction data can be recorded at wavelengths selected to affect the scattering behaviour of this particular element. Max Perutz[6] and John Kendrew[7] were the first to exploit the presence of heavy atoms in proteins for phasing. To introduce the heavy atoms into the crystals, they can be soaked in a heavy atom solution. Alternatively, the protein can be derivatized previously to crystallization or crystallization can be accomplished in the presence of the heavy atom. Ideally, the heavy atoms occupy ordered positions within the crystal, as their affinity to particular chemical environments drives their association to the protein, while only minimally disrupting the three-dimensional structure of the protein, that is native and derivative crystals are isomorphous, keeping the native crystal spacegroup and unit cell dimensions within a small percentage. Heavy atoms contribute significantly to the diffraction, making their contribution easy to detect. The difference of the diffraction pattern of the derivative and native crystals can be used to determine the positions of the heavy atoms. With direct, Patterson or dual-space recycling methods (see below) the heavy atom substructure can be solved.

This approach to phasing is called **Single Isomorphous Replacement** (SIR), the trigonometric relationships involving the phases and structure factors of native, derivative and substructure are fulfilled by two possible phase values. Thus, the problem is underdetermined since the number of variables is larger than the available equations, therefore, further experimental information or constraints must be applied in order to solve the structure. By using at least one other heavy atom derivative the phase ambiguity can be solved, this method is called **Multiple Isomorphous Replacement** (MIR). The attempts to get a proper derivative could be unsuccessful for lack of specific binding, insufficient derivatization or lack of isomorphism, along with other error sources. In addition, these derivative data are usually discarded once the target structure is solved, which together with the fact that the number of experiments that must be carried out to find a good substructure is unpredictable, poses an impractical increase in the time scale, resource needs and experimental effort of the crystallographic study.

The lack of isomorphism issue can be solved by recording anomalous data at various wavelengths on the same crystal as long as radiation damage[8] does not prevent it. Macromolecules, being chiral by nature, invariably crystallize in one of the 65 chiral, and thus acentric spacegroups. In the presence of anomalous scatterers, at appropriate wavelengths, Friedel's Law breaks down in acentric spacegroups. It states that the intensities of the h, k, l and $-h$, $-k$, $-l$ reflections (called Friedel pairs) are equal. The consequence

of Friedel's Law is that even if the space group lacks a center of symmetry, the diffraction pattern is centrosymmetric and it is not possible to tell by diffraction whether an inversion center is present or not.
The reason for Friedel's rule is that, according to the geometrical theory of diffraction, the diffracted intensity is proportional to the square of the modulus of the structure factor $|F_h|^2$. The structure factor is given by

$$F_h = \sum_j f_j \exp(-2\pi i \mathbf{h}.\mathbf{r}_j)$$

where f_j is the atomic scattering factor of atom j, $\mathbf{h}$ the reflection vector and $\mathbf{r}_j$ the position vector of atom j. If the atomic scattering factor, f_j, is real, the intensities of the h, k, l and -h, -k, -l reflections are equal, it follows:

$$|F_h|^2 = F_h F_h^* = F_h F_{\bar{h}} = |F_{\bar{h}}|^2$$

However, if the crystal is absorbing due to anomalous dispersion, the atomic scattering factor is complex andd

$$F_{\bar{h}} \neq F_h^*$$

The anomalous dispersion corrections take into account the effect of absorption in the scattering. The total scattering factor of an atom is a combination of different contributions, and at particular wavelength ranges the absorption effect is large enough to be exploited:

$$f(\lambda) = f^o + f'(\lambda) + if''(\lambda)$$

f^o, the normal scattering factor -which is independent of the wavelength- and f' and f'', the anomalous scattering factors -which account for real and imaginary components- that change with the wavelength. For X-ray energy values where resonance exists, f' usually decreases dramatically the real scattering component, while the value of the imaginary component f'' is positive[9].
If the anomalous signal is recorded at a single wavelength value, the phasing procedure is called **Single-wavelength Anomalous Dispersion**[10,11] (SAD), if otherwise the anomalous signal is detected at different wavelength values, it is called **Multiple-wavelength Anomalous Dispersion**[12] (MAD). Whereas MAD would render perfect phases in the absence of errors, SAD gives -as in the SIR case- two possible phase values, but even so this starting information may be sufficient to lead to a solved structure applying valid approximations, structural constraints and sophisticated mathematical treatments. Both procedures require that the percentage of difference between the anomalous component and the normal one produces a significant signal that can be distinctly separated from the noise. Although much lower values can lead to successful phasing, typical values of around 3%[13] can be exploited.

Many recombinant proteins can be obtained in prokaryotic expression systems while replacing methionine residues by selenium-methionine[14]. Selenium is an appropriate anomalous scatterer as its absorption edge corresponds to a wavelength that is readily accessible at synchrotron beamlines. Radiation damage[15] induces errors that might hinder the determination of reliable differences within the anomalous experiment. It is present even in crystals held at cryogenic temperatures (100K), the longer the crystal is exposed to X-ray radiation the higher the effect of the radiation damage increasing the amorphous or disordered volume of the crystal sample. The underlying cause of this damage is that the energy lost by the beam in the crystal -owing to either the total absorption, or the inelastic scattering of a fraction of the X-rays as they pass through the crystal- induces physical changes and chemical reactions in the sample.
MIRAS/SIRAS are experimental phasing techniques developed to solve crystallographic structures by combining isomorphous replacement and anomalous dispersion. This combination leads to an overdetermined problem by introducing more equations. Anomalous signal can be obtained from heavy atoms at suitable wavelengths, therefore single/multiple isomorphous replacement with anomalous signal scattering is powerful combining the strength of both methods if the appropriate experimental setup is available.
Finding the position of the heavy atoms and/or anomalous scatterers can be used to provide reference phases[16] for the structure to be solved. This has a comparable effect to introducing or differentiating a small molecule structure within the target macromolecule, which can be effectively solved by Patterson or direct methods. If the coordinates of the substructure can be obtained, phases for the whole structure can be derived and a starting model can be built[17,18] into the resulting electron density map.

Molecular Replacement

Relies upon knowledge from a previously solved similar structure to provide the missing phases and takes a different approach: locating a large number of lighter atoms (at least, most of the main-chain) as a rigid group[19]. It is based on the fact that macromolecules of similar sequence tend to have very similar structures by conserving their folds[20].
Possible orientations and positions of the solved model are tried in the unit cell of the unknown crystal[21] until the predicted diffraction best matches the observed ones aided by the Patterson function or Maximum-likelihood methods[22]. Then the unknown phases are estimated from the phases of the known model and an initial map is calculated with the "borrowed" phases and the observed amplitudes which are going to contribute in the rebuilding/refinement process supplying the information that will make the model closest to the target.
This approach becomes more powerful as the available structural knowledge deposited in the Protein Data Bank[2,3] grows larger. It is fast, efficient and highly automated though only useful when a close enough structure is available, which also introduces the problem of model bias because phases usually contain more structural information than the diffraction intensity,

and consequently, especially at low resolution, the calculated structure may be biased towards the phasing model.

The above mentioned methods are general methodologies that involve automated computational techniques at the final steps of their schedules, but, to achieve phasing, they all depend on experimental and/or particular, theoretical starting information at the initial stage of their layouts.
Methodologies demanding less initial experimental effort or particular structural knowledge, and thus more shifted to automated computational techniques would constitute a clear advantage. In this scenario, general ***ab initio* methods**, able to solve biomacromolecules using only a native dataset of amplitudes, without the use of a previously known related structure or phase information from isomorphous heavy atoms or anomalous scatterer derivatives would be desirable. Pure *ab initio* methods may not involve specific structural information, however, general features that are not specic to the structure in question can be utilized [23].
The ***ab initio*** field involves several approaches:

Patterson: the convolution of the electron density with itself can be calculated through the Fourier transform of the experimentally measured diffraction intensities [24]. This function does not contain phase information, but it has valuable information about interatomic vectors that can be used to determine their relative positions. The height of the maxima are proportional to the number of electrons of the atoms involved. Once a Patterson map is available, a correct interpretation may allow to get the absolute position of some atoms and use them to find the rest. This is feasible for heavy atoms within small molecules but does not scale well for macromolecules as the peaks originated from individual pairs of atoms are less prominent and heavily overlapped. Nevertheless, using its interpretation to aid other sophisticated techniques has been crucial in the solution of complex problems [25].
Direct methods: the unmeasured phases are not independent and relationships among them can be drawn for a particular set of measured intensities, based on probability theory derived from the application of nonnegativity and atomicity constraints [26]. This method exploits the fact that the crystallographic problem is heavily overdetermined at atomic resolution, which means that there are many more measured intensities than parameters that are necessary to describe an atomic model and therefore, it requires high resolution datasets. It is generally effective in the solution of small molecules up to 200 atoms [27] and has been widely used and implemented in crystallographic computer programs giving birth to a new generation of methods based on direct methods:

Dual space recycling: currently, recycling between real and reciprocal space is exploited in crystallography in a number of scenarios, but its original application to atomic resolution macromolecular phasing referred to an intensive-computational direct methods implementation based on the minimal principle, which states that a particularly simple function of the phases takes on its constrained minimal value for the correct set of phases [28].

While classical direct methods, in their original form, work entirely in reciprocal space, dual space recycling switches back and forth between real and reciprocal spaces to effectively enforce the atomicity constraint in both formalisms. This approach has come to be known as *Shake and Bake*[28,29] and consists on alternate trials of local minimization technique to be applied in reciprocal space (*Shaking*) with atomicity and nonnegativity constraints explicitly imposed in real space (*Baking*).

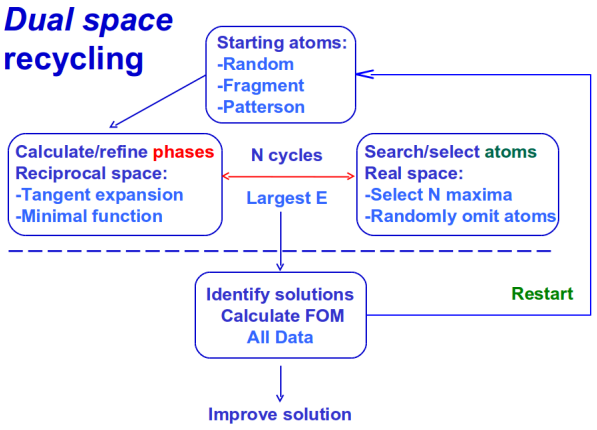

Figure 1.1: Dual Space recycling algorithm[23,30].

It still depends on high resolution information, yet successful on extending the power of direct methods into the scope of small macromolecules (≈1,000 equal atoms[23] and 2,000 atoms for structure containing heavy atoms[31]).
Charge flipping: a member of the dual space recycling family. It follows an iterative Fourier cycle that unconditionally modifies the calculated electron density and structure factors in dual spaces. The real-space modification simply changes the sign of the charge density below a threshold, which is the only parameter of the algorithm. Changing the sign of electron density below a small positive threshold simultaneously forces positivity, and introduces high-frequency perturbations to avoid getting stuck in a false minimum. In reciprocal space, observed moduli are constrained using the unweighed observed amplitudes (Fobs) map, while the total charge is allowed to change freely[32,33].
This is a very simple four-step cyclic algorithm for exploring the phase space that may be described as follows:

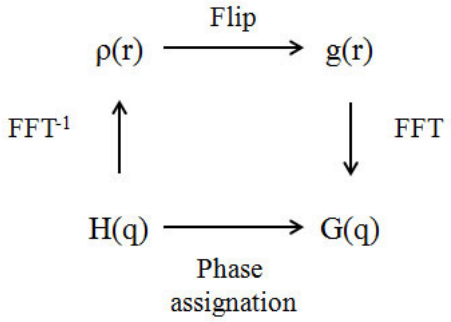

Figure 1.2: Charge flipping algorithm.

1. $\varrho(r) \rightarrow g(r)$: Real-space modification of the electron-density map, obtained by flipping the low-density region.
2. $g(r) \rightarrow G(q)$: Fourier transform to reciprocal space.
3. $G(q) \rightarrow H(q)$: Reciprocal-space modification of calculated structure factors, in practice replaced by the observed amplitudes.
4. $F(q) \rightarrow \varrho(r)$: Inverse Fourier transform to real space.

The method is able to solve large structures and even proteins if high resolution data are available and a couple of heavy atoms (calcium or heavier) are present [34].
VLD: is named after *vive la difference* and allows the recovery of the correct structure starting from a random model through cyclic combinations of the electron densities of the random selected model and the expected ideal difference Fourier synthesis [35].
Once a random model has been selected, the difference electron density should provide a map well correlated with the ideal one. Suitably modified and combined with the electron density of the model, it should define a new electron density no longer completely random, which, refined by electron-density modification to meet its expected properties (e.g. positivity, atomicity, solvent properties etc.) may lead to a better structural model.
Roughly speaking, the difference electron density provides a difference model, which is refined via cycles of density modification. The method is able to solve medium-size molecules and proteins of up to 9,000 atoms containing heavy atoms [31], provided the data have high, though not necessarily atomic resolution.

Considering that experimental methods are based on additional laboratory experiments, which at the same time introduce extra cost and time into the structure determination project, the methods that rely more on the computational stage appear to be more desirable, especially favoured by the continuous evolution of hardware and software nowadays.

1.2 Computational scenario

Given that the crystal structures of the first macromolecules were determined over 50 years ago[6,7,36], crystallography has started exploiting computing almost from its cradle. This has motivated a highly efficient use of resources in the evolution of crystallographic software. Methods programmed to run on computing centres 40-30 years ago were highly efficient, fast enough to be transferred to run on personal computers with risk, pentium or xeon processors. Thus, a scientific community regularly using computing, has largely turned its back on supercomputing. Still, the use of massive calculations allows to solve problems in completely different ways as algorithms -that would be prohibitive on a desktop machine- become amenable. The advantage is not to calculate the same faster, but rather to enable a different approach.

1.2.1 Grid computing/Supercomputing

Supercomputers are an extremely powerful type of computing systems (hardware, systems software and applications software) the most powerful available at a given time. They are composed by a large number of CPUs often functioning in parallel, actually, most of them are really multiple computers that perform parallel processing. Supercomputers have huge storage capacity and very fast input/output capability, but they are very expensive and so primarily used for scientific and engineering work[37].
In the absence of a supercomputer or just to optimize the project budget, high performance/throughput computing can be performed in networked frameworks as long as it is possible to divide the main goal into smaller and individual tasks. Grid computing can arise as an attainable alternative to supercomputers for solving scientific problems, backed up by strong reasons as cost reductions and resource availability.
A computing grid is a distributed system that supports a virtual research environment (VRE) across different virtual organizations[38] (VO), it well encases the collaborative work in the scientific world as sharing large volume of data among research groups is the norm.
A virtual organization merge in a set of individuals and/or institutions defined by sharing rules. The VOs can vary in purpose, scope, size, duration, structure, etc., but they all concern about highly flexible sharing relationships and sophisticated and precise levels of control over how shared resources are used[39].
This distributed infrastructure offers a useful tool to fulfil the needs of each client by squeezing the hardware capabilities of geographically dispersed resources, bringing together different operative systems and hardware platforms whose performance is higher on average, creating a virtual supercomputer which has the capacity to execute the tasks that, otherwise, would not be achievable by its single part-machines apart.
Multisolution approaches are a perfect target for such technology given that they provide the possibility of decomposing the problem into smaller computing tasks

affordable by way of parallel computing. Grid-computing environment better suits organizations that are already used to high performance/throughput computing or somehow oriented to distributed computing. A grid computing system comes together with administration and accounting resources issues being as it involves multiple members, which requires proportionality between their contributions.
The grid technology offers protocols, services and tools that accomplish the mentioned requirements by means of:

- Security solutions.
- Resource management and monitoring protocols and services.
- Data management services that locate and transport datasets between storage systems and applications.

The model of a computing grid enables the availability of an efficient network infrastructure, larger computing power and additional hardware for the different-constituent VOs. It can be described by its components as a layer-based structure:

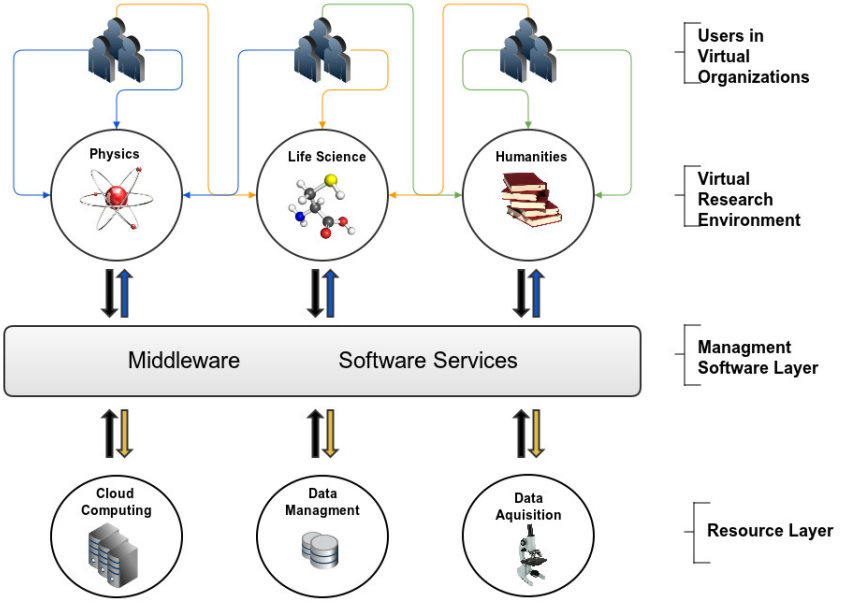

Figure 1.3: Layered structure of grids [38].

The hardware forms the bottom layer and consists of heterogeneous resources like processors, storage, networks and possibly sensors, including the corresponding system software. It is encapsulated by the management software layer, a domain-independent software that manages the VOs, guarantees data privacy and continuous interaction between resources while hiding to the user most aspects of hardware layer complexity. The third layer is the visible part of the VRE for a given problem domain. It is a domain-specific software layer, dedicated to the

needs of the various VOs within a VRE, that uses the services of the management software layer. Any user-specific application forms the top layer and only accesses the services of the discipline-specific software layer[38].

1.2.2 Middleware

In the field of computing systems, a job is the unit of work given to the operating system, *e.g.*, a job could be the run of an application program, and it is usually executed without further user intervention. A similar term is task, that can also be applied to interactive work.
Middleware software is key for the optimal functioning of a grid computing system, its role within the management software layer is to connect the different parts of the grid while offering at least elementary services such as job execution, security, information and file transferring. However, it may offer as well more advanced services including file, task and information management. Middleware software should hide the underlying complexity of the grid as much as possible, but should provide enough flexibility that individual control about a file transfer can be issued[40].
The **job execution services** must be able to submit a job and to check its status at a any given time. Jobs must have a unique identifier so that they can be properly distinguished.
Security services should involve authentication and authorization to protect resources and data on the grid. Because of the large number of resources in the grid, it is important to establish policies and technologies that allow the users to sign-on through a single interaction with the Grid.
Information services provide the ability to query for information about resources and participating services on the grid and must be able to deal with the diverse nature of queries directed to it.
File transfer services must be available to move files from one location to another sophisticatedly enough to allow access to high-performance and high-end mass storage systems. Often such systems are coupled with each other and it should be possible to utilize their special capabilities in order to move files between each other under performance considerations.
Aside from the aforementioned, a variety of more advanced services can be developed. For example, users may want to profit from advanced file transfer services that dynamically adapt to changing network conditions and fault situations without user intervention; organizing and scheduling subtasks as a result from a comparative study, where multiple and similar jobs were run, should be performed by a sophisticated service that utilizes the capabilities of the grid by reusing the current generation of grid services. Therefore, grid task management is a complex issue as part of any grid middleware software:

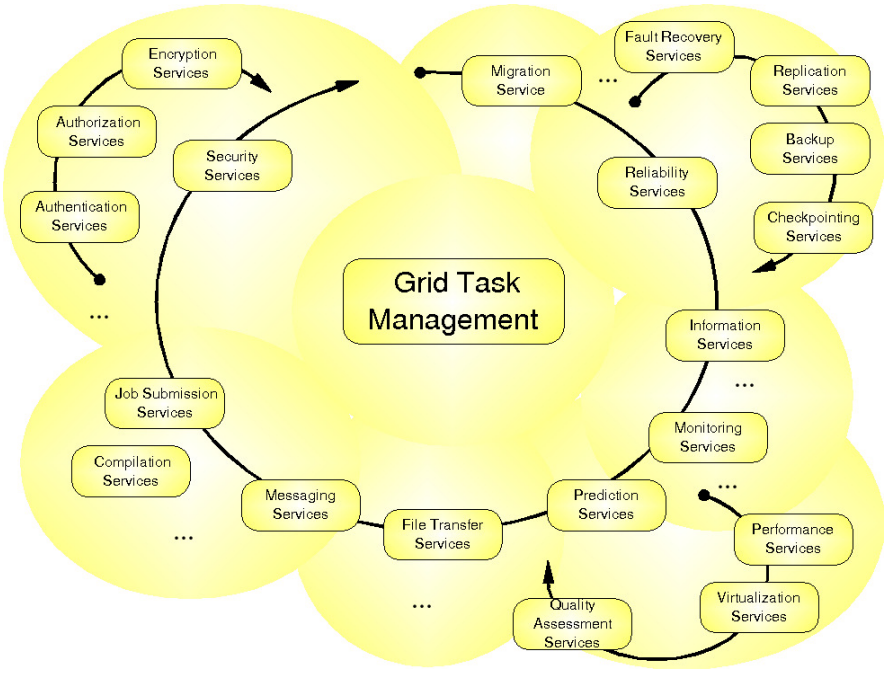

Figure 1.4: Complexity of Grid task management [40].

There are several approaches of grid computing software now available: HTCondor [41], ARC [42], Globus [43], UNICORE [44], etc. Each one is unique because the interpretation of protocols and standards can vary significantly as well as the underlying platforms.

1.2.3 Cloud computing

Nowadays, computing is transforming from applications running in individual computers to services demanded by users from anywhere in the world, offering results based on the user requirements and served without regard to where they are hosted or how they are delivered. Clouds are promising to provide services to users without reference to the infrastructure on which they are hosted.
A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more computing resources based on service-level agreements established through negotiations between the service provider and consumers [45].
It presents the opportunity to reduce or eliminate costs bound to in-house provision of those services by making them available on demand and removing the need of heavy invests, plus difficulties in building and maintaining complex computing infrastructures.
The **service providers** make services accessible to the **service users** through web-based interfaces. Clouds aim to outsource the provision of the computing infrastructure required to host services. This infrastructure is offered "as a service" by **infrastructure providers**, moving computing resources from the SPs to the IPs, so the SPs can gain in flexibility and reduce costs [46].

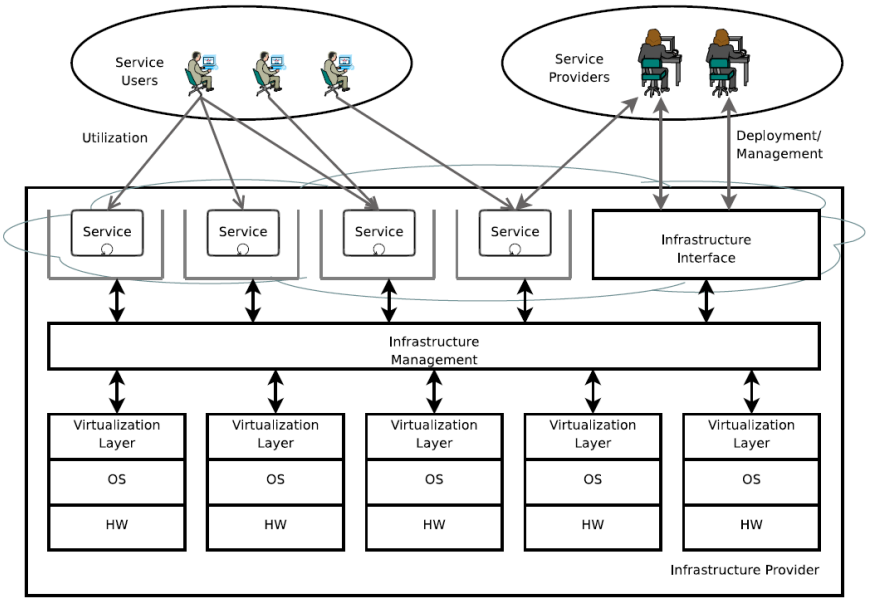

Figure 1.5: Cloud actors[46].

It is a new way of providing and utilizing resources, but by no means should be considered a competitor to grid-computing technology as both approaches can benefit from each other[38].

1.3 The ARCIMBOLDO approach to solve the phase problem

ARCIMBOLDO materializes a new approach to render macromolecular *ab initio* phasing generally successful, rather than limited to exceptional cases, through the exploitation of parallel computing. Its application has afforded the practical results reflected in the publications attached.
The aim of this thesis is to lay out the development of the method in the course of the last years, which started by using molecular replacement (MR) to place small secondary structure models such as the ubiquitous main-chain helix against a native dataset of amplitudes, and to subject the resulting phases to a density modication and autotracing procedure in a supercomputing frame.
The algorithm is conceived as a multisolution method that generates many starting hypotheses, therefore it is controlled by sophisticated filters to limit the number of parallel jobs that have to be calculated within tractable limits and to discriminate successful solutions automatically through reliable figures of merit.
Even using rather complete homologous models, the calculations for the rotation and translation searches enclosed in the classic MR methodology are already set in a multisolution environment. Such technique applied to a smaller scattering mass involves a huge amount of calculations as figures of merit become unreliable

and many more indistinguishable, partial solutions have to be pursued, requiring a more powerful computational resources.
Nevertheless, the tasks to be executed are very easy to parallelize and suitable for a grid environment. The highly heterogeneous hardware and operative systems characterizing the limited computing resources we had access to at the beginning of our project, attracted our attention to an open-source high throughput product named HTCondor[47] that allows to run multiple instances of the same software, just slightly varying the input, on a large number of otherwise idle or sub-utilized processors, regardless of their heterogeneity.
A condor-grid was then configured and exploited for our experiments at the Institute of Macromolecular Biology of Barcelona (IBMB in Spanish) and later a collaboration was established with the Foundation of Supercomputing Center of Castile and León (FCSCL in Spanish). The condor-grid environment tested at IBMB was then exported to Calendula, a supercomputer with hardware characteristics compatible with our first experimental workspace.
Macromolecular crystallographic structures have always entailed a difficult computing problem to solve. This is originated by the existence of two important differences with small molecule crystals: size and resolution. Both of them hinder direct methods when applied to proteins and although they are generally effective for small molecules, methods based on probabilistic phase relations have been relegated to a small number of cases, notably large antibiotics, with characteristics closer to small molecules than to macromolecules that precluded the use of standard protein methods[23].
Of the two barriers just mentioned, resolution is the most difficult to overcome. The available resolution determines the amount of information obtained from a given crystal form in the diffraction experiment, the total number of unique diffraction intensities being higher, the higher the resolution. The difference in resolution will be reflected in the quality of the electron density map, a higher data resolution comporting a higher level of detail in the electron density map, which in turn leads to higher accuracy of the positions of the atoms in the model determined.
In the present work we have pursued the relaxation of resolution values. Protein or peptide structures of small size can not be solved by direct methods in the absence of atomic resolution. The size limitation for direct methods is well found in the mathematical background, and though size is a constant parameter for each structure, data resolution depends on the diffraction experiment and is thus conditioned by the crystallization procedure, the researcher skills, and the diffraction platform. Nevertheless, the possibilities to improve the diffraction properties of a given crystal form are limited as well by the intrinsic nature of the crystal. Not just the resolution limit, but also the completeness of the data is a main factor for *ab initio* methods to succeed, experiments have proved that even extrapolated values of non measured data improves the quality of the final electron-density map[48].
Lacking the advantages derived from atomicity in the absence of high resolution data prevents the use of direct methods, but at lower resolution this constraint can be substituted by the knowledge that macromolecular structures are composed by smaller fragments of known geometry (like α-helices), whose number and length is predictable from the protein amino acid sequence. This information can be

used to constrain the phasing procedure. Tests with the density modification program ACORN[49] have proved that placing 13% of a structure can be enough to succeed with density modication and the idea of enforcing stereochemistry as a powerful constraint has been implemented with highly effective results in the autotracing algorithms of popular crystallographic software such as RESOLVE[50], ARP/wARP[51] or SHELXE[52]. All these elements have been channelled into tackling the objectives described in the next section.

Chapter 2

Goal Settings

Ab initio, direct methods[23], solve the phase problem through probability theory derived from the positivity of the electron density map and atomicity constraints. To be satisfied, these constraints require atomic resolution data and although they perform exceedingly well for small molecules, in the field of macromolecules their applicability is hindered by the requirement of exceptionally high resolution, further complicated by the large number of atoms to be located.
However, the fact that proteins are composed by building blocks of known geometry like α-helices and β-strands, which can be predicted from their amino acid sequence, provides a possible general alternative to atomicity, as a means of bringing in prior stereochemical information that can and must be used to extend the classical *ab initio* macromolecular phasing methods.
The main subject of this project has been the development of a new computational method of general applicability at medium resolution -around 2Å- to overcome the phase problem relying only on a dataset of native diffraction amplitudes and without previous detailed structural knowledge or measurements of heavy atoms or anomalous scatterer derivatives.
We focused our attention in mainly helical macromolecular proteins given the tendency of α-helices to be more rigid in their main-chain geometry than β-strands, the almost ubiquitous presence of α-helical main-chain fragments in protein structures (78% of the structures deposited in the Protein Data Bank[2,3] contain this secondary structure element), and their predictability from the known amino acid sequence or even from data features such as the Patterson function or the Wilson plot even in the absence of sequence information.
The planned objectives for this thesis were:

- To achieve macromolecular *ab initio* phasing, using only a dataset of native amplitudes and without detailed structure knowledge, measurements of heavy atoms or anomalous scatterer derivatives.

- To relax the previous stringent requirement for atomic resolution data, targeting a typical macromolecular resolution limit, fulfilled by half of the structures deposited in the PDB: 2Å.

- To combine the advantages of some of the currently available phasing methods, integrating complementary sources of phase information into a new method of general applicability.
- To exploit grid and supercomputing resources in a multisolution frame, designing a parallel tasks environment in order to decrease the time cost.
- To develop the method testing it on already known macromolecular structures.
- To solve previously unknown protein structures with resolution, size and composition beyond the scope of the aforementioned methods when independently working.
- To develop and optimize different branches of the method depending on the starting information available.
- To provide a tool accessible to the standard crystallographer lacking specialized supercomputing knowledge and/or hardware resources.

Chapter 3

Materials and Methods

ARCIMBOLDO[53] is a methodology of general applicability for macromolecular phasing at medium resolution, around 2Å, which works in a multisolution frame by combining the correct location of accurate small model-fragments with the program Phaser[54] and density modication and autotracing with the program SHELXE[55]. Given its multisolution nature, it requires grid/supercomputing environments.

3.1 Phaser

Phaser[54] is a widely used MR program that has been developed by Randy Read's group at the Cambridge Institute for Medical Research (CIMR) in the University of Cambridge. It can be found associated to the CCP4[56] suite of programs for macromolecular structure determination as well as to PHENIX software suite[57]. MR in Phaser is based on maximum likelihood probability theory target functions[58], to better take into account the incompleteness and inaccuracy of the search model. This suits well the case of small main-chain fragments, necessarily representing a very partial fraction of the correct structure.
Maximum likelihood states that the best model is most consistent with the observations: experimental data and prior knowledge. Consistency is measured statistically by the probability that the observations should have been made; if the model is changed to make the observations more probable, the likelihood goes up indicating that the model agrees better with the data. One way to think about likelihood is imagining that data have not been measured yet, a model is available with various parameters to adjust (coordinates and B-factors in the case of crystallography), as well as some model concerning the sources of error and how they would propagate. This allows to calculate the probability of any possible set of measurements and to see how well they agree with the model.
Probability distributions in experimental science are often Gaussian, and then likelihood becomes equivalent to the least-squares formulation. In the case of crystallography, the (phased) structure factors have Gaussian distributions, but

the phase information is lost when intensities are measured, and the distributions of the measurements are no longer Gaussian. This is why it is necessary to go to first principles and apply likelihood.
The standard Phaser procedure possesses a high degree of automation, however, the step by step algorithm can be divided in convenient blocks of **rotation**, **translation**, **packing**, **re-scoring** and **refinement and phasing** tasks, that can be invoked using traditional CCP4 keyword-style input.
Each of these processes depends on specic input lines, some of which are common to all of them:

- Mandatory
  - Phaser task to execute
  - MTZ file containing observed data.
  - Labels contained in the MTZ file and used for the structure factor amplitudes and their standard deviations.
  - Model(s) in the form of a pdb file.
  - Molecular weight of the protein.
  - Number of molecules in the asymmetric unit (AU).
- Optional
  - Potential solution(s) to be used.

The output files indicate the particular figure of merit (FOM) for every process each solution has passed through:

RFZ: Rotation Function Z-score.

TFZ: Translation Function Z-score.

PAK: Number of clashes in the packing.

LLG: Log-Likelihood-Gain.

Signal-to-noise is judged using the **Z-score**, which is computed by comparing the LLG values from the rotation or translation search with LLG values for a set of random rotations or translations. The mean and the root mean square deviation (rmsd) from the mean are computed from the random set, then the Z-score for a search peak is defined as its LLG minus the mean, all divided by the rmsd, *i.e.* **the number of standard deviations above (or below) the mean**.
For the rotation function, the correct solution may be in the list with a Z-score under 4, and will not be identified until a translation function is performed and picks out the correct solution.

For the translation function the correct solution will generally have a Z-score over 5 and be well separated from the rest of the solutions. The table provided in the Phaser manual gives a rough guide to interpreting TF Z-scores, these values are to be trusted in the conventional MR context, but very incomplete models represent a different case, to be determined from systematic molecular replacement trials.

TF Z-score	**Have I solved it?**
less than 5	no
5-6	unlikely
6-7	possibly
7-8	probably
more than 8*	definitely
*6 for 1st model in monoclinic spacegroups	

Table 3.1: Indication of a structure being solved through the TFZ-score values according to the Phaser manual.

When searching for multiple components, the signal may be low for the first few components but, as the model becomes more complete, the signal should become stronger. Finding a clear solution for a new component is a good sign that the partial solution to which that component was added was indeed correct. In our particular use scenario, the fragments to be located invariably constitute a very small fraction of the total structure, although they tend to be very accurate.

3.1.1 Rotation

The rotation task is performed to obtain the orientation of a model expressed in α, β and γ Euler angles, the notation ϕ, θ, ψ is also frequently used.

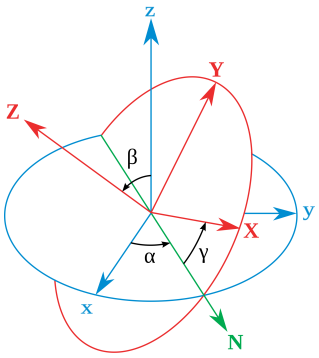

Figure 3.1: Euler angles representing rotations:xyz is the original frame, XYZ is the rotated one and the intersection of **xy** and **XY** is represented by **N**.

To understand how a rotation is represented by such angles, let us define **xyz** as the axes of the original coordinate system, **XYZ** as the axes of the rotated one and **N** as the intersection of the **xy** and **XY** coordinate planes. Within this reference frame, α is the angle between the *x-axis* and the *N-axis*, β is the angle between the *z-axis* and the *Z-axis* and γ is the angle between the *N-axis* and the *X-axis* [59].

The rotation likelihood function can be approximated through two different functions: the brute-force or fast rotation. In both cases a rotation function for the model is calculated on a grid of orientations covering the rotational asymmetric unit for the spacegroup of the crystal. The function used for a brute search [60] represent a better approximation to the likelihood target and is thus, very exhaustive and slow to compute, therefore recommended only when the space for the search can be restricted using additional information. The fast rotation function [61] represents a signicant speed improvement, at the price of calculating one term less.

In addition to the mandatory input parameters mentioned above, Phaser rotation searches also demand:

- Rotation mode (brute or fast).

- Model to orient against diffraction data.

Its output is written to files with extensions `.out` and `.rlist`. The latter contains a condensed summary of the solutions produced in the run, exhaustively detailed in `.out` files. Files with extension `.rlist` contain the set(s) of orientations that best satisfy the chosen rotation function together with the observed data, sorted by a given FOM, and selected after applying a given cutoff, *e.g*:

```
SOLU SET

SOLU TRIAL ENSEMBLE ensemble1 EULER 138.693 77.189 91.309 RFZ 3.68

SOLU TRIAL ENSEMBLE ensemble1 EULER 141.203 81.674 91.277 RFZ 3.67

SOLU TRIAL ENSEMBLE ensemble1 EULER 140.744 79.499 89.329 RFZ 3.62

SOLU TRIAL ENSEMBLE ensemble1 EULER 148.954 83.869 39.361 RFZ 3.60
```

Each line contains the label associated to the oriented model (highlighted in bold) and the calculated values for α, β and γ Euler angles that will become the input for the translation search. At the end of the line, underlined, the value of the Rotation Function Z-score (rotation FOM) is expressed.
The example displayed corresponds to a set of solutions calculated for the 1st orientation search of a model, but, for subsequent fragments, a rotation can also be calculated once a model-fragment has been placed. This information might come from different sources, for example, additional knowledge concerning the position of a substructure or a previous solution that is not in itself enough to achieve successful phasing, but can contribute with partial information to a combined search. In such cases, the output file of potential orientations for the placement of a new model-fragment might look like this:

```
SOLU SET RFZ=2.9 TFZ=4.4 PAK=0 LLG=7 LLG=12 TFZ==0.5

SOLU 6DIM ENSE ensemble1 EULER 134.873 78.270 219.973 FRAC 0.49572 0.54601 0.96696 BFAC -
3.32103

SOLU TRIAL ENSEMBLE ensemble1 EULER 141.203 81.674 91.277 RFZ 3.53

SOLU TRIAL ENSEMBLE ensemble1 EULER 138.693 77.189 91.309 RFZ 3.51

SOLU TRIAL ENSEMBLE ensemble1 EULER 140.283 81.862 91.429 RFZ 3.45
```

The FOMs underlined, correspond to the tasks performed to locate the 1st fragment and resulting in a 6 dimensional solution, which in its turn is stressed in bold and expressed by means of the name of the placed model, its rotation Euler angles, its fractional coordinates, and refined B factor.
The "bold" solution is fixed during the rotation search of the next fragment and taken into account for the current fragment search. In the example, it constrains the calculations for the given rotation function, whose alternative results are the lines that can be found right after the line in bold, with each corresponding Rotation Function Z-score stressed in bold.

3.1.2 Translation

Once a fragment is oriented, the translation task can supply its position by means of three values corresponding to fractional coordinates. Translation, as well as rotation, can be run in two modalities within Phaser: brute-force or fast translation. The translation function is calculated on a hexagonal grid of positions and, like previously mentioned on rotation, the translation brute-force search [60] is time and compute-resource demanding, which makes it preferable to use it to refine the results of a previous fast translation search, when the location can be more or less predicted. Latest Phaser versions actually perform these brute-force rotation and translation searches to refine the peaks of the fast functions by default. A speed improvement is achieved in any case with the likelihood-enhanced translation functions as approximations to the full likelihood target [62].
Phaser translations searches also require as input:

- Translation mode (brute or fast).
- Model to locate.
- Rotation solution(s).

The output files of a translation search, `.sol` contain the set(s) of solutions sorted-by-FOM, that comprise orientations and locations for the provided model(s) best fitting the calculated equations against observed data:

SOLU SET RFZ=3.5 **TFZ=4.3**

SOLU 6DIM ENSE ensemble1 EULER 141.297 82.863 92.195 FRAC 0.25620 0.70221 0.98149 BFAC 0.00000

SOLU SET RFZ=3.6 **TFZ=4.5**

SOLU 6DIM ENSE ensemble1 EULER 140.283 81.862 91.429 FRAC 0.27330 0.70034 0.98783 BFAC 0.00000

SOLU SET RFZ=2.9 **TFZ=4.5**

SOLU 6DIM ENSE ensemble1 EULER 116.901 36.346 347.424 FRAC 0.19616 0.75040 0.35943 BFAC 0.00000

SOLU SET RFZ=3.4 **TFZ=4.4**

SOLU 6DIM ENSE ensemble1 EULER 82.975 38.802 302.573 FRAC 0.40305 0.75660 0.44776 BFAC 0.00000

Each solution contains the name of the oriented model, the calculated rotation Euler angles, fractional coordinates and B factors. The top solutions according to a given FOM will become input for the packing pruning task.

3.1.3 Packing

The packing filtering is a process where Phaser checks whether the potential solutions and their symmetry equivalents overlap geometrically and therefore are physically unreasonable. It is an important pruning step where the dispersed solutions after individual rotations and translations must be gathered together in

packages of a manageable number of solutions so that the Phaser clash-check can evaluate if all of them are valid solutions, or just overlapped, or technically impossible given the size of the asymmetric unit.
The number of solutions to be grouped together can be customized by the user, but changing our default values should be unnecessary. Phaser packing within ARCIMBOLDO works smoothly with the common input detailed above, and with the intention not to overload the network sending and receiving unnecessary data, a lighter MTZ file is created from the original one using the mtzdmp utility available through the CCP4 installation. The new MTZ file contains only the header and not the information concerning to reflections of the original MTZ file and it is going to be used specically for the packing moment.
Each requested packing may return a `.sol` file, but sometimes the clash test returns no solution fitting this constraint. The output contains the set(s) of solutions that have passed the test, they all subsequently become input for the re-scoring task:

SOLU SET RFZ=3.5 TFZ=4.3 LLG=9 **PAK=0**

SOLU 6DIM ENSE ensemble1 EULER 141.297 82.863 92.195 FRAC 0.25620 0.70221 0.98149 BFAC 0.00000

SOLU SET RFZ=3.6 TFZ=4.5 LLG=9 **PAK=0**

SOLU 6DIM ENSE ensemble1 EULER 140.283 81.862 91.429 FRAC 0.27330 0.70034 0.98783 BFAC 0.00000

SOLU SET RFZ=2.9 TFZ=4.5 LLG=9 **PAK=0**

SOLU 6DIM ENSE ensemble1 EULER 116.901 36.346 347.424 FRAC 0.19616 0.75040 0.35943 BFAC 0.00000

SOLU SET RFZ=3.4 TFZ=4.4 LLG=9 **PAK=0**

SOLU 6DIM ENSE ensemble1 EULER 82.975 38.802 302.573 FRAC 0.40305 0.75660 0.44776 BFAC 0.00000

3.1.4 Re-scoring

Phaser allows to compare rotated and translated solutions coming from the same or different input files. The Log-Likelihood Gain mode resorts the grouped solutions by calculating the log-likelihood gain under specified conditions, *i.e.* resolution, identity of the model and number of copies in the asymmetric unit. It returns re-scored `.sol` file(s), sorted according to the new calculated value of the LLG. It does not check for identity among different solutions:

SOLU SET RFZ=3.5 TFZ=4.3 **LLG=9**

SOLU 6DIM ENSE ensemble1 EULER 141.297 82.863 92.195 FRAC 0.25620 0.70221 0.98149 BFAC 0.00000

SOLU SET RFZ=3.6 TFZ=4.5 **LLG=9**

SOLU 6DIM ENSE ensemble1 EULER 140.283 81.862 91.429 FRAC 0.27330 0.70034 0.98783 BFAC 0.00000

SOLU SET RFZ=2.9 TFZ=4.5 **LLG=9**

SOLU 6DIM ENSE ensemble1 EULER 116.901 36.346 347.424 FRAC 0.19616 0.75040 0.35943 BFAC 0.00000

SOLU SET RFZ=3.4 TFZ=4.4 **LLG=9**

SOLU 6DIM ENSE ensemble1 EULER 82.975 38.802 302.573 FRAC 0.40305 0.75660 0.44776 BFAC 0.00000

3.1.5 Refinement and Phasing

Phaser can subject each of the fragments composing a solution to a rigid-body refinement. In addition, it identifies equivalent solutions, which are pruned taking into account crystallographic symmetry and possible changes of origin.
This is a slow and thus demanding step, calculations can be distributed in packages with a feasible number of solutions according to the RAM allocated to the computing node. The number of solutions to be refined and analysed simultaneously on a single machine can be defined: a very large number scales up the time taken in the calculation and may lead to process failure if the RAM is insufficient, on the other hand, equivalent solutions are pruned only after refinement has been performed.

SOLU SET RFZ=2.9 TFZ=4.4 LLG=7 **LLG=12**

SOLU 6DIM ENSE ensemble1 EULER 134.873 78.270 219.973 FRAC 0.49572 0.54601 0.96696 BFAC -3.32103

SOLU SET RFZ=2.8 TFZ=4.3 LLG=8 **LLG=12**

SOLU 6DIM ENSE ensemble1 EULER 113.981 37.016 350.787 FRAC 0.21074 0.75605 0.36052 BFAC -3.43524

SOLU SET RFZ=2.9 TFZ=4.2 LLG=8 **LLG=12**

SOLU 6DIM ENSE ensemble1 EULER 139.343 68.933 218.556 FRAC 0.38722 0.74061 0.14966 BFAC -3.43052

SOLU SET RFZ=2.9 TFZ=4.8 LLG=9 **LLG=11**

SOLU 6DIM ENSE ensemble1 EULER 122.982 36.100 342.544 FRAC 0.29818 0.75969 0.54681 BFAC -3.09906

Phaser may be directed to output coordinate and files containing map coefficients and phases in `.pdb` and `.mtz` formats, respectively, in addition to the described solution files.

3.2 SHELXE

SHELXC, SHELXD and SHELXE[63] are stand-alone executables designed to provide simple, robust and efficient experimental phasing of macromolecules by the SAD, MAD, SIR, SIRAS and RIP methods. They are particularly suitable for use in automated structure-solution pipelines.
SHELXE was designed to provide a route from substructure sites found by SHELXD to an initial electron density map. While most density modication programs were developed originally for use within a low resolution scenario based on solvent flattening and histogram matching, SHELXE was conceived from the high resolution end and attempts to bring in stereochemical knowledge, through the sphere of inuence[52] and free lunch[48,49,64] algorithms to improve a preliminary map by density modication.
The latest released version iterates between density modication and generation of a poly-ala trace[63], enabling an interpretable map and partial model to be obtained from weak initial phase information.
The preliminary map to start a density modication with SHELXE must not necessarily originate from heavy atoms or anomalous scatterers, but can also be obtained

by a coarse solution of MR[65] after the placement of small model fragments like theoretical helices of polyalanine. It can be said that the structure is solved if longer chains are traced with a correlation coefficient between the observed and native data better than 25% when the resolution of the data extends to around 2Å.
To start from a MR model without other phase information, the `.pdb` file from MR should be renamed `xx.pda` and input to SHELXE, *e.g.*

```
shelxe xx.pda -s0.5 -a20
```

In ARCIMBOLDO, molecular replacement is executed by Phaser, the output used is a collection of possible solutions expressed in terms of the name of the model and the proposed rotation and position values (see 3.1.5). After re-sorting the refinement and phasing solutions and selecting the top ones, a further calculation is carried out to transform the original coordinates of the model by applying each rotation + translation solution to the atoms in the pdb file. The results are recorded in files with the extension `.pda`, which are the ones needed for the density modication and autotracing jobs.
If anomalous data are available, a MR solution can be combined with an anomalous map as starting point for further iterative density modication and poly-ala tracing with SHELXE. These two approaches can be combined, a `.res` file must be provided being the equivalent of a `.pda` file, but containing information about the anomalous model. If the model is an anomalous substructure, native and anomalous data in hkl format (`xx.hkl` and `yy.hkl`) and substructure in res format (`yy.res`) are requested:

```
shelxe xx yy -s0.5 -z -a3
```

or the phases from the MR model are used to generate the heavy atom substructure. This is used to derive experimental phases that are then combined with the phases from the MR model. Native and anomalous data (`xx.hk` and `yy.hkl`) and models (`xx.pda` and `yy.res`) are required:

```
shelxe xx.pda yy -s0.5 -a10 -h -z
```

The following SHELXE options are the most common ones within the ARCIMBOLDO scenario described in the present work (SHELXE defaults in brackets):

Syntax	Function
-aN	N cycles autotracing [off]
-dX	truncate reflection data to X Angstroms [off]
-eX	add missing 'free lunch' data up to X Angstroms [dmin+0.2]
-h or **-hN**	(N) heavy atoms also present in native structure [-h0]
-i	invert spacegroup and input (sub)structure or phases [off]
-mN	N iterations of density modification per global cycle [-m20]
-o or **-oN**	prune up to N residues to optimize CC for xx.pda [off]
-q	search for α-helices
-sX	solvent fraction [-s0.45]
-tX	time factor for helix and peptide search [-t1.0]
-vX	density sharpening factor [default dependent on resolution]
-wX	add experimental phases with weight X each iteration [-w0.2]
-yX	highest resol. in Å for calc. phases from xx.pda [-y1.8]

Table 3.2: Most common SHELXE options within ARCIMBOLDO scope.

For options available in later SHELXE versions, running the program with no arguments provides a list of defaults.

3.3 HTCondor

HTCondor[47] is an open-source software, developed in the University of Wisconsin, that allows High Throughput Computing (HTC) on large collections of distributed resources by combining dispersed computational power into one resource.
It enables Linux, MacOS and Windows ordinary users to do extraordinary computing by enforcing the premise that large computing power can be achieved inexpensively with collections of small devices rather than an expensive, dedicated supercomputer. Every node participating in the system remains free to contribute as much or as little as other constraints may impose or advice.
This specialized workload management system for compute-intensive jobs, provides **job queueing mechanism**, **scheduling policy**, **priority scheme**, **resource monitoring** and **resource management**[41], which are similar functionalities to that of a more traditional batch queueing system, but HTCondor succeeds in areas where traditional scheduling systems fail. For example, HTCondor is able to integrate resources from different networks into a single HTCondor pool, besides, it can be configured to share the machine resources with non-HTCondor jobs and to prioritize them (even to the point of migrating the HTCondor job to a different machine) based on checking if keyboard and mouse, or CPU are idle. If a job is running on a workstation when the user returns and hits a key, or the machine crashes or becomes unavailable to HTCondor, the job is not lost, but can migrate to a different workstation without the user intervention.
This configuration, respecting primary use of the nodes by their owners and ensuring that only residual, idle time will be dedicated to ARCIMBOLDO calculations has allowed us to expand our limited computing power adding resources of other research groups (either from our own network or from an external one) that might not be averse to contributing with their workstations under such conditions of use.
While developing ARCIMBOLDO, the experiments were initially carried out in a local grid that is currently formed by 22 workstations clustering 114 cores. All of them are GNU/Linux machines of 64bit architecture, with 2, 4, 6 or 8 cores

each, and memory ranging from 1,5 - 4GB per core. One of these machines (owing 4 cores) belongs to an external network and contributes to the calculations only during night. Of the rest, 8 machines, with 8 cores each, are completely dedicated to HTCondor tasks, while the others contribute only whenever any of their cores is idle.

Along with the ARCIMBOLDO version developed to run on the local grid described, a version has been programmed to run on the powerful, homogeneous system provided by a supercomputer. The facility used in the course of this thesis has been a pool provided by the Calendula supercomputer from the FCSCL in León (Spain). Its architecture, is based on Xeon Intel chips and HTCondor was installed, as well as the software required by ARCIMBOLDO (Phaser, mtzdmp and SHELXE). The local prototype was successfully exported and ARCIMBOLDO is currently running in Calendula, using 240 dedicated cores running GNU/Linux, under a 64bit architecture, with 2GB of memory per core.

Additional useful features of HTCondor include that it does not require a Network File System (NFS) to access centralized data over the network; on behalf of the user, HTCondor can transfer the files corresponding to a job. It is also independent of centralized authentication system, like NIS.

In a heterogeneous pool -where only some workstations belong to a common NFS- the user does not need to know whether the files must be transferred or not. HTCondor would smoothly transfer files if the machine that submits the job and the one that executes it do not have a common NFS. Furthermore, also programs and libraries can be transferred and do not need to be regularly accessible from the executing machine. Our local pool includes 8 workstations (34 cores) within the same NFS, which means that files are transferred if one of this group is paired with one of the rest. In Calendula, files are never transferred since there is NFS available.

HTCondor implements a flexible framework for matching resource requests (jobs) with resource offers (machines) using a mechanism similar to classified advertisements in newspapers (ClassAds). There can be defined job requirements and job preferences as well as machine requirements and machine preferences.

None of our pools has a policy of machine preferences, but the jobs produced by ARCIMBOLDO can vary regarding their requirements, for example, heavy tasks are directed to machines with larger memory.

HTCondor does not require a centralized submission machine, users can submit their jobs from any machine configured as a submitter, which will contain its own job queue. In addition, the grid nodes can play different roles at the same time, a submission machine can be as well an execution machine; once a job is submitted, based on its ClassAds, HTCondor chooses when and where to execute it, monitors its progress, and ultimately informs the user upon completion.

While the supercomputer environment was set up with a single dedicated submitter, in the local pool the jobs can be submitted from 7 workstations which are simultaneously execution machines. Even though the same workstation can play several roles, our pools have a dedicated central manager; in the local pool this role is played by our weakest machine. A machine of 1GB of memory can perfectly perform as central manager. Our local central manager is a single core 32bit machine of 1GB memory, running GNU/Linux, its power would have not contributed

significantly to the grid, but would rather delay the total time to achieve a goal.

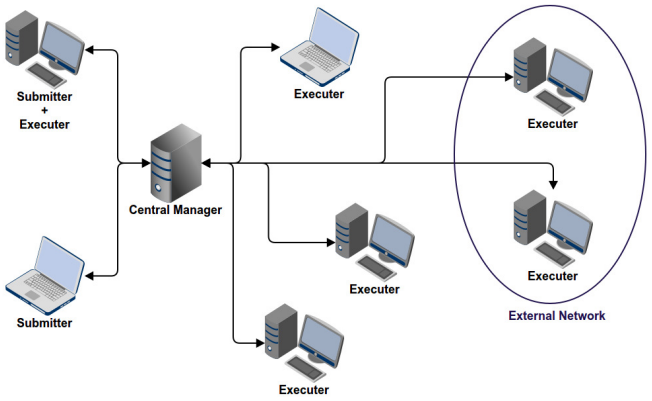

Figure 3.2: Example of an HTCondor grid. Dispersed computing-resources communicating through HTCondor.

Users can assign priorities to their submitted jobs in order to control their execution order. A "nice-user" mechanism requests the use of only those machines which would have otherwise been idle. This kind of jobs does not even compete with other HTCondor jobs, that is, whenever some non nice-user HTCondor job is requiring a resource, it has priority over a machine occupied by a nice-user job. Administrators, as well, may assign priorities to users enabling a fair share, strict order, fractional order, or a combination of policies.
When jobs are submitted to an agent, it is responsible for remembering them in persistent storage while finding resources willing to run those jobs. Agents (job submitters) and resources (job executers) advertise themselves to a matchmaker (central manager), which is responsible for introducing potentially compatible agents and resources. Once introduced, an agent is responsible for contacting a resource and verifying that the match is still valid.

Grid computing requires both planning and scheduling[47]. Planning is the acquisition of resources by users and scheduling is the management of a resource by its owner. There must feedback between planning and scheduling and HTCondor uses matchmaking to bridge the gap between planning and scheduling.

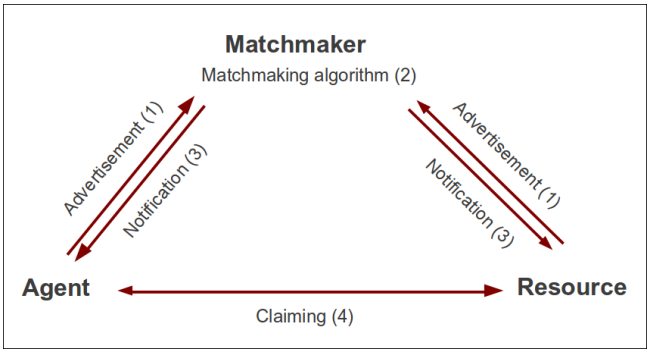

Figure 3.3: HTCondor matchmaking[47].

Matchmaking creates opportunities for planners and schedulers to work together while still respecting their essential independence. It requires 4 steps:

1. Agents and resources advertise their characteristics and requirements in ClassAds.

2. A matchmaker scans the known ClassAds and creates pairs that satisfy each other constraints and preferences.

3. The matchmaker informs both parties of the match.

4. Matched agent and resource establish contact, possibly negotiate further terms, and then cooperate to execute a job.

Job ClassAd	**Machine ClassAd**
MyType = "Job" **TargetType** = "Machine" **Requirements** = "((Memory < 2004) && (Arch == "X86_64"))" **Rank** = "kflops"	**MyType** = "Machine" **TargetType** = "Job" **Requirements** = "(LoadAvg <= 0.3) && (KeyboardIdle > (15*60)))" **Rank** = "Owner == user1 || Owner == user2" **OpSys** = "Linux" **Memory** = "2010" **Arch** = "Intel"

Table 3.3: Example of ClassAds matching.

Each ClassAd contains a **MyType** attribute describing what type of resource it represents, and a **TargetType** attribute that specifies the type of resource desired in a match. Jobs advertising want to be matched with machine advertising and vice-versa. The HTCondor matchmaker assigns significance to two special attributes,**Requirements** and **Rank**. **Requirements** indicates a constraint and **Ranks** measures for both, machine and job, the desirability of a match (where higher numbers mean better matches). It is used to choose among compatible matches, for example, in this case, the job requires a 64bit computer with a memory capacity higher than 2004MB. Among all such computers, the customer prefers those with higher relative floating point performance.
Since the **Rank** is a user-specified metric, any expression may be used to specify the desirability of the match. Similarly, the machine may place constraints and preferences on the jobs that it will run by setting the machine configuration. In order for two ClassAds to match, both requirements must evaluate to "True".
A job is submitted for execution to HTCondor using the `condor_submit` command. It takes as an argument the name of the submit description file, which should contain commands and keywords to direct the queuing of jobs. In the submit description file, the user defines everything HTCondor needs to execute the job. Within ARCIMBOLDO, the job requirements are contained in files named after the process they will run, plus `.cmd` extension.
Within the same submitting file, it is very easy to submit multiple runs of a program, with different input and output files. To run the same program **N** times on **N** different input data sets, the data files must be arranged such that each run reads its own input, and each run writes its own output.
The keyword `queue` inside the submitting file states the value of the total jobs to be submitted (**N**), that in turn are individually accessible through `$(Process)`. For example, if `queue 4`, `$(Process)` takes values of 0, 1, 2 and 3.
A submitting file containing the following keywords

```
input = $(Process).sh
transfer_input_files = data.mtz, $(Process).pdb, $(Process).sh
output = $(Process).out
queue 100
```

would submit 100 jobs, where the number of the process determines the input files taken to execute each single job, and the name of the output to produce.
Once submitted, the jobs can be managed and monitored through several commands available to the HTCondor user, `condor_status` and `condor_q` are very handy when tracking the jobs and the grid resources. `condor_status` can be executed from any HTCondor machine and can be used to monitor and query the pool. It shows, among other, detailed information about the names of the machines, their number of cores, architecture and operating system, and resumed information of the total claimed, unclaimed, matched, and preempted machines. `condor_q` is related to the queue of jobs, therefore it can only be executed on submitter machines. It displays the status of all jobs in the local queue by detailing the job ID (related to `$(Process)`), the owner of each job, the time of submission, etc. Both commands may modify their output by specifying available options that can be used to customize the user experience.

Chapter 4

Results and discussion

4.1 ARCIMBOLDO: the central hypotheses underlying the method

Crystallography provides an accurate and detailed three-dimensional view into the macromolecular structures that has not been surpassed yet by any other structural technique. Nevertheless, the structural model resulting from the crystallographic determination cannot be directly calculated from data collected during an X-ray experiment due to what is known as the phase problem: only the diffracted intensities and not the phases from the X-ray diffracted beams can be obtained while the phases are essential for the structure solution.
Crystallography relies upon different methodologies that perform calculations on the available experimental data and previous knowledge to retrieve the so yearned for phases. MR and the measurement of derivatives are the most widely extended crystallographic methods to solve the phase problem on proteins, but they require beyond the native dataset, additional experiments or previous specific knowledge. This complicates and enlarges the timescale of the project.
To phase a structure, starting only from the diffracted intensities without additional stereochemical or experimental information, general *ab initio* methods must be developed. Direct methods are invariably effective in the field of small molecules crystallography, however, given their strong dependency on atomic resolution data, in the case of macromolecules they have been confined to a few favourable cases. The significantly large number of atoms to be determined in a macromolecular structure multiplies the complexity of the problem for a direct method approach since it is based on probabilistic relations inversely proportional to the square of the number of atoms. Also, protein crystals generally contain a high proportion of disordered solvent, which results in impaired periodicity, smaller crystal size, lower diffraction signal to noise ratio, etc. All this factors, intrinsic to macromolecular crystals, tend to limit the diffraction resolution below that required for direct methods.
The pioneering work of Weeks and Hauptman[28] on dual-space recycling methods

allowed the breakthrough of macromolecular phasing through direct methods, extending its scope from 200 independent atoms to ≈ 1,000 atoms structures without atoms heavier than sulphur and diffracting to atomic resolution [23].
Although atomic resolution is exceptionally rare to be found on macromolecular datasets, the technological advances in beamline and crystallization technologies, together with the increase in data-collection, time availability, and beamline speed, are leading to better determined datasets to higher resolutions. Thus, the proportion of medium resolution data is gaining ground in the PDB entries.
The presence of elements heavier than sulphur, such as metals in the active centre or playing structural roles, constitutes a favourable circumstance for an *ab initio* approach. It has allowed, as well, to successfully solve a structure of 1,283 atoms including holmium, diffracting at 1.92Å [66] and 7,890 atoms including 8 gold atoms with resolution up to 1.65Å.
Of resolution and size barriers, the first one appears to be the most difficult to overcome. Reported cases suggest that even extrapolation of those reflections that have been missed during the X-ray experiments, is preferable to not accounting for such unmeasured data [48,49,64] In the frame of density-modification algorithms, even extrapolating data beyond the experimental diffraction limit has proven to be effective [35,52,67,68].
Within the scenario just described, the scope of this work has focussed on developing an *ab initio* phasing method, able to succeed using medium resolution data by exploiting different available sources of information. It has been applied successfully on test and previously unknown cases at resolutions comprised between 1.2 and 2Å, that is the span where resolution is no longer atomic but still high, and yet attainable in 50% of crystallographic determinations [2,3].
The central ideas underlying the method developed in the course of this work derive from the proposition that enforcing **secondary structure** rather than atomicity opens a new way to phase macromolecular structures *ab initio* at medium resolution. Any model fragment predictable from the amino acid sequence might be useful, α-helices, β-strands, and base pairs are common fragments to macromolecular structures that can be used at this initial stage of the phasing procedure. By combining the location of small fragments with the molecular replacement program Phaser, with the density modification and autotracing procedures of the program SHELXE, phases close enough to the true structure can be obtained which allows a further automatic or manual building of the complete structure.
Predictable fragments, such as a 14 amino acids main-chain α-helix, represent a very small fraction of the scattering mass in the structure and will account for a low percentage of the measured diffraction. As a consequence, even if correctly placed, figures of merit characterizing their placement, at medium resolution, will not be very sensitive. Given that most structures will require placement of several correct fragments, a large number of indistinguishable hypotheses has to be generated to enable that correct ones are included in the set. The method will require pushing all hypotheses on to the point where you can recognize their correctness. The procedure can be compared to the experience of solving a jigsaw puzzle having a vision problem and lacking the proper glasses: you patiently try different positions of possible pieces and discard those with edges that do not fit at all. However, chances are that several pieces have the proper shape to encase in the

puzzle frame, but do not match the drawing pattern. Unfortunately, your fuzzy vision prevents you from recognizing the correctly placed pieces from the wrongly placed ones.
In order to proceed, the most promising hypotheses should be pushed to the point where their correctness could be recognized. In a macromolecular scenario, density modification is effective in revealing and identifying the true portrait of the protein being solved when the starting hypothesis is close enough. In our example of the puzzle, finding a pair of glasses that do not provide a 100% perfect vision, but help to better distinguish some extra details allowing to successfully trace the rest of the pattern, is analogous to the task performed by density modification in the currently described method. As long as it cannot be known which fragments are correctly placed, the procedure will need to take all partial molecular replacement solutions to density modification, coupled to main-chain autotracing, in order to produce a reliable figure of merit: CC of the traced fragment and number of fragments traced.
Given the complexity of the problem, the amount of calculations demanded exceeds the volume that can be computed on a single workstation, even a powerful one with many cores: supercomputing is needed before such an algorithm can be pursued. Even with supercomputing not all possible hypotheses can be calculated, far more possible hypotheses can be proposed than calculated, therefore a sophisticated design is essential in order to trim down the pool of intermediate solutions, identifying early on and pursuing only those more likely to succeed.
Even if the method is rooted in the *ab initio* principles, it should be recalled that the main aim is to phase the structures under study and there is no reason to prevent exploiting all the information and data that might be available: thus if anything is known about the problem or additional data are available they should be incorporated into the frame of the method. Prior stereochemical knowledge of weak anomalous signal constitute the most frequent source of complementary information that can and should be blended into the method.
The algorithm and program incorporating the elements just enunciated have been named after the Italian painter Giuseppe Arcimboldo (1526 or 1527 - July 11, 1593), who developed a painting technique that illustrates well the basic idea of our method. In his still-life paintings, small objects like fruits, vegetables, flowers, etc, are disposed in such a manner that a "hidden" portrait of a subject can be recognized (see Fig. 4.1). The ARCIMBOLDO method composes hypotheses out of secondary structure elements that, if properly arranged, reveal the true portrait of the protein. When the initial hypothesis is close enough to the true "portrait" of the protein, density modification will reveal its structure, while most of the trials computed will remain "still-life" pictures.
Although in ARCIMBOLDO any predicted fragment can be used as starting hypothesis, the method is focused mainly in exploiting polyalanine α-helices as they comprise the more rigid, ubiquitous arrangement of atoms, thus it may fit well the core of any helix independently of its sequence. Furthermore, reducing the side-chains to the CB in alanine, favours having incomplete but extremely accurate stereochemical information rather than having a complete but imperfect model. If the calculated phases are accurate enough, electron density revealing the side-chains should emerge in the map.

Figure 4.1: Rudolf II painted by G. Arcimboldo as Vertumnus.[69].

Dual-space recycling atomic resolution methods based on direct methods have equal chances of producing solutions with either chirality, that is, the solution can correspond to one out of two chemically identical molecules being mirror images of each other. Fragments composed by two atoms, such as disulphide bridges are non-chiral and again, two different, yet indistinguishable substructures may be determined. For alanine, for the mirror image to occur, the groups bonded to the central carbon would have to be swapped to the other side of the carbon.
A good illustration of this are the human hands, both, with identical components but non superposables, are a good example to explain chirality. Each mirror image is known as the L-isomer or the R-isomer (like a left hand or a right hand) or r and s which is used by chemists. In proteins only the L-enanthiomer of alanine is found.

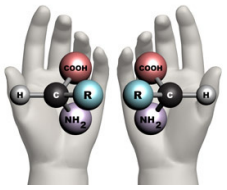

Figure 4.2: Chirality in two enantiomers of a generic amino acid.

Hence, protein fragments such as α-helices are chiral, fixing in its turn the chirality of the resulting structure. This would be true for any other secondary structures but α-helices, are the most rigid, predictable from sequence, as well as the most prevalent ones. Their ubiquitous presence in proteins makes them the perfect target when looking for model fragments.
Helices observed in proteins generally range from 4 - 40 residues long, but a typical helix contains about 14 amino acids, besides, larger helices tend to bend resulting in less regular forms and therefore in less predictable models. MR techniques, applied at lower resolution to such small portions of the total scattering mass, produce a large number of solutions including not correct ones. Regular FOMs in such context are not longer reliable[70], precluding their use to clearly discriminate correct solutions within a MR performance.

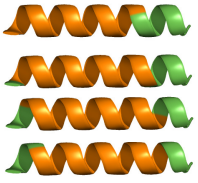

Figure 4.3: Small model helix correctly fitting a larger target helix at several positions.

On the other hand, since the models used for the search are very small, they are accurate, but a small helix can be accommodated several ways in a larger one, *e.g.* given a sliding window of 1aa, a 14aa helix might fit a 20aa one in 7 different positions, provided that they are not severely bent. Furthermore, a helix of 14aa is such a general secondary structure element that, in common proteins, can be found forming part of almost all the helices of the structure, thus increasing the number of possible solutions and forcing to pursue a huge amount of hypotheses. Consequently, the ARCIMBOLDO approach has to be performed as a multi-solution algorithm and is thus highly demanding in terms of computing power.

4.1.1 Underlying algorithms

The ideas outlined in the previous section have been implemented with the practical technical design described in this section. The tasks performed will be the location of model fragments through molecular replacement, coupled to density modification and autotracing. Here, the flow of the algorithm programmed in ARCIMBOLDO in the course of this work will be described, whereas the next section will detail the results obtained in tests cases that have guided the choice of setup finally adopted, and that have eventually led to the solution of unknown structures.
The free HTCondor middleware was selected in view of its flexibility to handle heterogeneous hardware architecture, operating system and portability from our local development pool to exploiting the method in a supercomputer. HTCondor allows to access grid/supercomputing technology providing the necessary platform for ARCIMBOLDO to deal with a large number of hypotheses. The algorithm is ideally suited for parallel computing, as the problem can be easily divided into multiple tasks to be run in parallel, aggregation of results and decision making for subsequent steps can be designed without interfering with the number crunching calculations, for which highly optimized programs are already available. This minimizes the overhead time paid in aggregation, choosing a flow design that will balance tasks to prevent idle time or superfluous computation.
The molecular replacement layer of ARCIMBOLDO is performed by the program Phaser, widely used in an automated mode for solving macromolecular structures in the classic molecular replacement way. This automated strategy comprises a sequential combination of functions that can be invoked separately, allowing to create a parallel schedule of tasks adaptable to a grid infrastructure. The sequence of Phaser modes that ARCIMBOLDO uses are: **rotation search**, **translation search**, **packing check**, **re-scoring of solutions** and **rigid body refinement**. In general, the models that ARCIMBOLDO uses for the MR search are required in multiple instances within the algorithm. This fact naturally introduces the idea of defining a customized search (depending on the model to locate in the structure) to be repeated as many times as the fragment is expected. The exception is the 1st ARCIMBOLDO round to locate a model fragment, which starts from nothing but the model and the diffraction data, where parallelization of the tasks is unnecessary.
This initial calculations are not intensive enough to require a complex design since they can be safely executed on a single machine, therefore, although some of the advantages of distributed computing offered by the HTCondor grid are used (for example, targeting a more powerful resource than the jobs submitting-machine), the procedure for locating the 1st model fragment flows sequentially and is slightly different from all subsequent rounds to locate the same or other models. At this stage, it is important to produce a variety of solutions to constitute a diverse base for subsequent calculations.

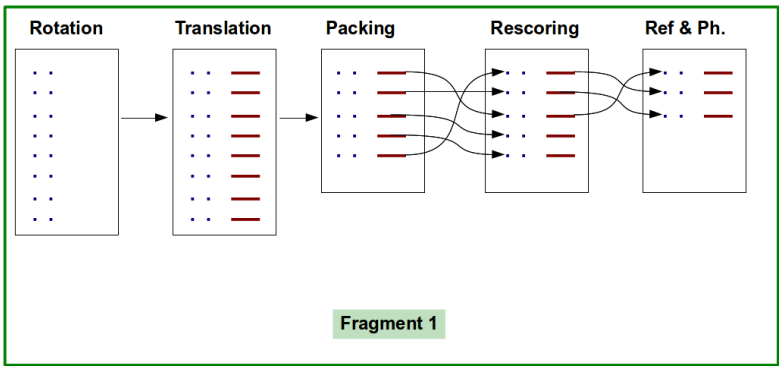

Figure 4.4: Simplified procedure of ARCIMBOLDO attempt to locate the 1st model fragment.

The above diagram represents a detailed description of the molecular replacement layer in the general procedure for the 1st round to locate a model by ARCIMBOLDO. The rotation search produces a single `.rlist` file with the calculated three-dimensional rotation solutions (represented by the dotted lines in the figure) that become input for the translation search.
The translation search finds positions characterized by 3 coordinates (beelines) for the provided rotation angles. The resulting `.sol` file, containing 6 dimensional solutions, is sent to a packing check. From the packing results, some solutions are rejected according to the given constraints and a `.sol` file with less solutions is passed to the re-scoring step and later to a rigid body refinement. At this point the 6-dimensional solutions are refined and duplicated ones are pruned, the resulting list is sorted according to the LLG.
Once a 1st fragment has been placed, the subsequent cycles of model location share a common design. The design strives to keep the number of calculations within tractable limits, terminating some of the open, less promising threads while keeping some variety. They all start with a rotation search that uses the output of the previous rigid body refinement prune, which implies that the starting information for the rotation searches is larger since no dependable criterion is available to unequivocally identify the best partial solutions.
The number of hypotheses tends to increase exponentially, but must be kept within affordable limits even using supercomputing, where the available computing power is larger. Efficiency is not simply a matter of reducing the program runtime, but rather of rendering the problem tractable at all, time and resources must be invested in pursuing those solutions more likely to succeed. To this end, control parameters are introduced to prevent the number of solution from exploding.
Although a resolution around 2Å is necessary and the higher its limit, the easier for the density modification process to succeed, the need to use full resolution dataset to find the rotation and translation values for a model fragment had not been assessed previously. Each of this processes sets to determine 3 values, thus being on principle highly overdetermined. Limiting resolution reduces the time needed for the calculations, however, the optimal value could vary from one fragment to another or depending on the particular dataset. To explore and fine-tune this effect in the case of small secondary structure fragments, and to be able to

control it in production, a specific variable of ARCIMBOLDO was introduced. The next section shows the results related to its effects.

Hypotheses that seed a new fresh round of model search must be divided in tractable portions of solutions in order to balance the workload within the grid environment. Since the time needed for the calculations of a single rotation search is very short and a grid architecture provides several computing nodes, the method spreads each partial solution over the grid and associates the index order of the refined solutions with its corresponding rotation file.

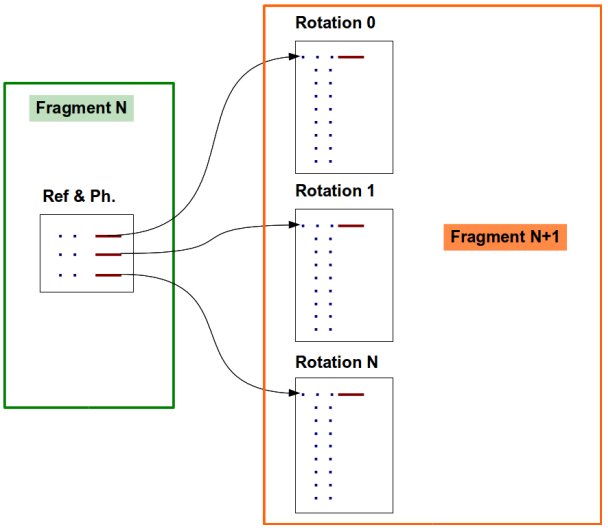

Figure 4.5: Rotation search layer of ARCIMBOLDO attempt to locate any N+1 fragment.

The partial solutions acting as seeds for the fragment N+1 search are depicted in Fig. 4.5 as the heading dotted-straight lines in the rotation files. The shifted dotted lines represent the euler angles for possible N+1 fragment locations calculated within the rotation search.
The summed partial solutions can be seen as the puzzle pieces that, when correctly assembled, allow the density modification procedure to eventually show the final picture. In every cycle of model placement ARCIMBOLDO sets to incorporate one more piece to the puzzle through calculations led by the previously placed model fragments.

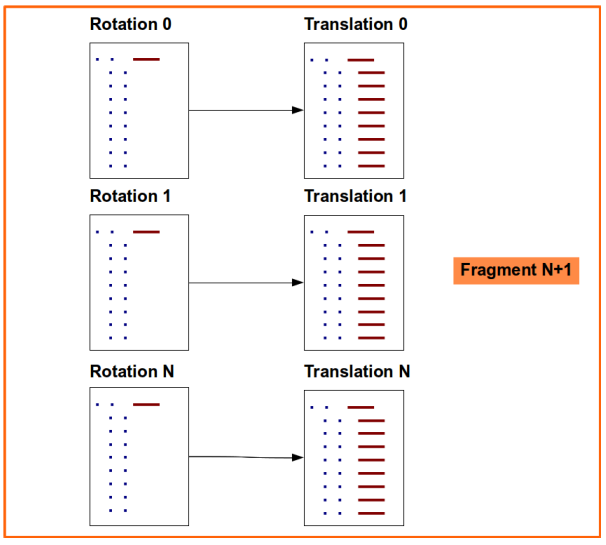

Figure 4.6: Translation search layer of ARCIMBOLDO attempt to locate any N+1 fragment.

Depending on the hardware diversity of the available grid, some of the rotation results might take more time to be calculated than others. The HTCondor grid manages the correct execution of the jobs, but its performance -for the purpose of ARCIMBOLDO- is mainly limited to the data reception, job execution and output returning. ARCIMBOLDO is in charge of checking whenever a rotation job is finished to send its output as input for a translation search in order to obtain 6-dimensional solutions of the N+1 fragment location. By the time the translation search finishes, for each rotation run there will be one translation file containing multiple solutions.

Frequently, the number of solutions and output files is large and none of the molecular replacement FOMs is discriminating until a certain percentage of the final structure is correctly placed. During the development of the method (see in particular the EIF5 case described in next section), analyses showed that solutions leading to the final structure tend to be contained in translation output files with comparatively few solutions. A procedure was incorporated into the ARCIMBOLDO algorithm, to select a sample of translation output files and calculate its averaged number of solutions (see Fig. 4.7). Once the average established, all translation resulting files containing a larger number of solutions are completely discarded, whereas from the rest, the top solutions (typically 70) are retained. With this value, the available information can be efficiently trimmed. Otherwise, the less promising translation results, containing the largest number of solutions would flood the computing nodes.

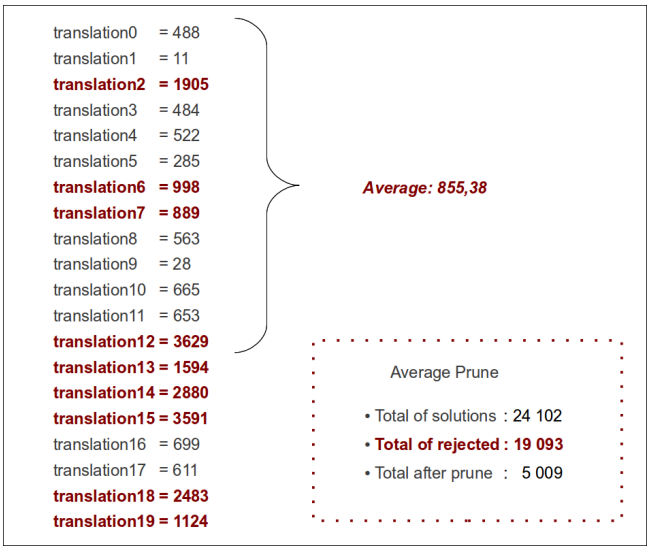

Figure 4.7: Averaged solutions prune for translation results.

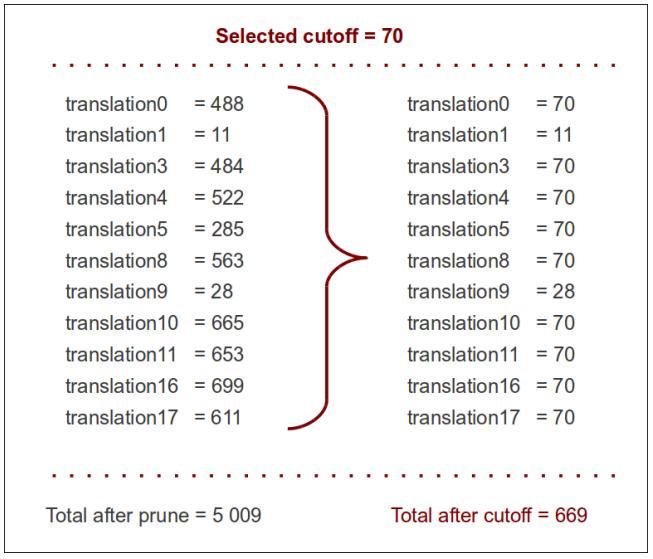

Figure 4.8: Translation solutions selection after cutoff.

Even if pruning based on the average number of translations helps to discard unpromising partial solutions, the remaining ones still tend to constitute too large a number to pursue. In the course of fine-tuning on test cases, results suggested that even though small fragments at medium resolution are not unequivocally scored through figures of merit (LLG, Z-SCORE, CC), they provide useful indications. Thus, solutions leading to the final structure tend to be positioned towards the top of the most promising output-files.

The rationale behind not simply scoring all possible solutions and keeping the top is that within a multisolution method, preserving some variability is desirable, and as the same model helix, perfectly placed on two different parts of the structure will be characterized by different FOMs, it is preferable to trim discontinuously

through the re-scored files. Both the number of solutions contained in the output files, as well as the index of the best solutions within each file, can vary significantly depending in a complex way on several parameters such as the size of the structure, crystal symmetry and data resolution, chosen parameters for rotation and translation searches. etc.The optimal choice of parameters may have to be optimized after feedback from a preliminary run is available. Defaults based on the ARCIMBOLDO experience are provided in the program where a number of top solutions can be selected before moving forward with an improved number of hypotheses.
Although the Fig. 4.8 describes the cutoff trimming after a **translation search**, the same procedure can be applied after **packing**, **re-scoring**, and **refinement and phasing** steps. With the cutoff selection the number of solutions per file can be greatly reduced, but, creating input files containing only the selected top group of solutions is inefficient.
In the case of Phaser modes performing very fast calculations (packing checks and re-scoring of solutions), a large number of files with a low number of solutions is inefficient, as the overhead paid in file transfer, handling, queueing, etc, is not compensated by the parallelization. Conversely, small jobs constitute a handicap for the Phaser pruning mode "refinement and phasing", which is more profitable when using populated input files more likely to contain redundant solutions that are discarded once identified.

In addition, the number of solutions contained in the accepted output files can greatly vary from one file to another (compare `translation0` against `translation1` in Fig. 4.8). It is necessary to create a more homogeneous sample, consequently, after the translation search and for each one of the subsequent Phaser modes within the ARCIMBOLDO algorithm, input files should be redimensioned grouping a number of solutions, in such a manner, that the weakest computer of the grid can deal with them when performing a high memory demanding task, yet containing a number of solutions appropriate for the Phaser pruning processes.

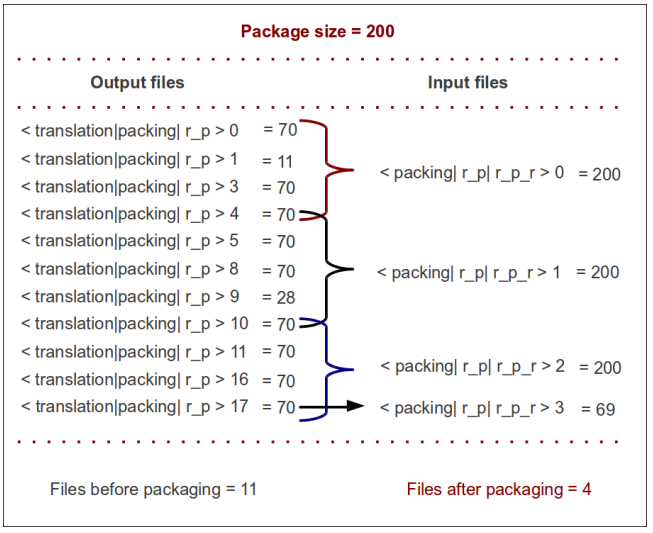

Figure 4.9: Redistribution and packaging of packing, re-scoring and refinement and phasing input files.

In general, when moving forward from one Phaser stage to the next, all the output files are collected and treated as a mass of solutions which is divided in packs of equal number of entries. Afterwards, the algorithm proceeds with a new redistribution of the available information.

After each complete round of fragment location, a density modification and autotracing run can be performed based on the partial solutions produced by molecular replacement. The program SHELXE attempts to phase the whole protein structure starting from the Phaser partial solutions obtained after a rigid body refinement.
Since there can be several output-files from the rigid body refinement, ARCIMBOLDO collects all of them and offers the possibility of re-sorting the partial solutions by a different FOM than the default LLG in Phaser. The sorting criterion affects both, the new round of fragment search, and the density modification and autotracing runs of the current fragment.
Starting from different partial solutions calculated by Phaser and expanded by SHELXE, the final structure can be successfully solved several times, but once a solution is available there is no interest in achieving more solutions. On that account and since the SHELXE jobs take more time to be finished than the Phaser tasks, an express lane was designed to expand only a number of top solutions while the input to expand the remaining ones is created, compressed and stored in a file to be revisited if the program finishes without achieving a solution from the first selection of trials.
The expansions are sent in parallel and addressed to the most powerful machines of the grid pool and, since the expansion results do not affect at all further fragments location (if any) the next round of model placement is immediately started after the SHELXE jobs are launched and while they are being calculated.

4.1.2 Solution of test cases.

To develop and parameterize *ab initio* phasing, 4 test cases were selected in different spacegroups, at resolutions ranging from 1.2 to 2Å, and protein sizes from 87 to 368 amino acids in the asymmetric unit. 1.2Å is regarded as the limit for atomic resolution, whereas 2Å was our goal at the onset of this project. The chosen cases are:

1. **CopG (2CPG)**[71]: 1.2Å. Three molecules with 43 amino acids each, 129 amino acids, 1015 atoms strand, helix, helix, 45% solvent, $C222_1$.
2. **Oxidized bacteriophage T4 glutaredoxin (Thioredoxin)** (1ABA): 1.45Å, one molecule with 87 amino acids, 728 protein and 152 ligand plus solvent atoms, fold contains three helices and 4 strands, 45% solvent, $P2_12_12_1$.
3. **Glucose isomerase (1MNZ)**[72]: 1.54Å, in house data, 385 amino acids, TIM barrel, 3433 atoms, 48% solvent, I222.
4. **Eukaryotic translation Initiation Factor 5 (EIF5) C-terminal domain (2IU1)**[73]: 1.7Å, one molecule with 179 amino acids, α-helical, 1473 atoms, 45% solvent, $P2_12_12_1$

Details on the data and tests described individually.

4.1.2.1 CopG, on the border of atomic resolution.

CopG was a particular test case where the resolution of the experimental diffraction data was 1.2Å. Despite the availability of such unusually high resolution data, an *ab initio* solution with SHELXD could not be achieved. The characteristics of this case (small size and nearly atomic resolution) were used to establish a first proof of principle and develop various of the previously described parameters within ARCIMBOLDO.

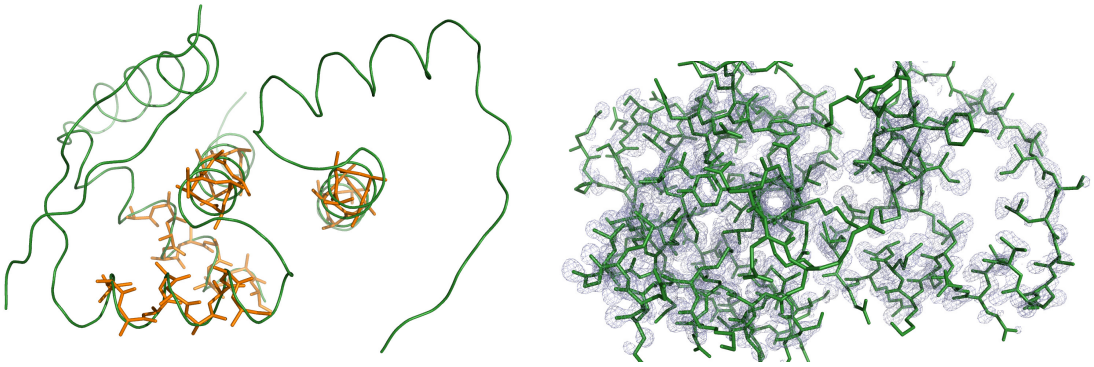

Figure 4.10: Left: Backbone traced with SHELXE for CopG from a set of 4 polyalanine α-helices located with Phaser. Right: detail of the SHELXE map for the CopG solution.

The AU of the orthorhombic $C222_1$ CopG contains three monomers of 43aa each (1015 protein atoms in total) and 40% solvent, six helices (of ∼12 amino acids) are present.
Using default parameters but full resolution for an automatic Phaser run, to locate a theoretical model helix made up of 10 alanines, combining rotation, translation, packing and refinement, yielded 4 correctly placed helices and failed to find the fifth and sixth helices. The reason is that the default search grid is too coarse when dealing with such small fragments. It can be overcome by adopting a mesh of 1-3 degrees for the rotation function and 0.5-0.7Å for the translation search, but a finer grid for the translation search requires more memory and the 4 located fragments are enough start information for SHELXE to obtain phases that would permit to build the whole structure.
Even 2 out of the 20 partial solutions containing a single helix of 10aa are enough to phase the structure (MPE: 29.1°) even though they do not correspond to the solutions characterized by the highest LLG figure of merit. Thus, if even at atomic resolution correct solutions cannot be perfectly discriminated by their FOMs, a multisolution approach is imperative to ensure structure solution.
The 4 correctly placed helices can be extended judging on the LLG. Re-scoring LLG on a set of model helices of different lengths superimposed on the 10aa search model, allows to extend the original fragment if the LLG value increases, thus completing three helices of 14 amino acids and one of 12 (LLG = 98 for the initial 40 residues substructure, LLG = 136 for the improved 54 residues one). After thus completing the model, helices 5 and 6 can be found as well.

The effect of resolution on our tests appears somewhat erratic. One would naturally expect that using the highest possible resolution would lead to optimal results with as perfect a model as a main-chain helix, since it entails an increase in the number of data per parameter to be determined, but this is not the case and previous work anticipated it[74]. Limiting the resolution may be beneficial in the rotation search, even with low rmsd models. In the case of CopG, truncating the resolution of the data from the available 1.2 to 2.1Å for the rotation function still yields correctly oriented (Ala)10 helices, for which the translation function works well truncating the data to 1.5Å, whereas no translation solution is achieved for these rotations when data are truncated to either 1.8Å or 2.1Å.

Res. cut-off rot.	**Res. cut-off trans.**	**Sol. 2H**	**Sol. 3H**	**Sol. 4H**	**Sol. 5H**	**Sol. 6H**	**Solved?**
2.1	1.2	110	1207	20	2	2	2H
1.8	2.1	129	787				2H
2.1	2.0	18	100	195	295	476	No
2.5	2.5	175	2787	29536			No
2.1	1.8	69	393	1249	2433		No

Table 4.1: ARCIMBOLDO results for protein CopG depending on resolution cutoffs.

Res. cutoff rot.	Resolution cutoff for the rotation search
Res. cutoff trans.	Resolution cutoff for the translation search
Sol. (#)H	Number of MR solutions containing # helices

On the other hand, performing the rotation search with data truncated to 1.8Å and using these rotations to determine translations with data truncated to 2.1Å did work, yielding a complete solution from the expansion of some substructures composed of 2 fragments.
Truncating the resolution for both rotation and translation search to 2.5Å did not yield useful substructures for phasing.

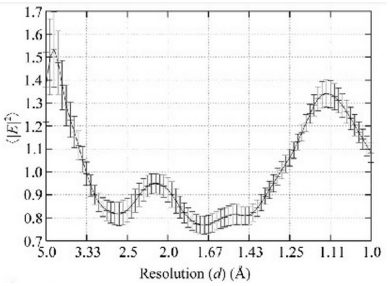

Figure 4.11: Averaged squared normalized structure factor amplitudes over 700 protein structures with standard deviations calculated from the population of individual $|E|^2$ profiles[75].

These results are in agreement with the study of Morris and Bricogne[75], where for over 700 protein structures deposited in the PDB, the behaviour as a function of the resolution of the averaged squared normalized structure factor amplitudes was examined.
They suggested that one possible explanation for the sharp dependence of direct methods on atomic resolution could lie in the fact that the resolution shells below 1.2Å contain the stereochemical information derived from the interatomic bond distances. Correspondingly, the resolution shells below 2.4Å would contain the values of most 1-3 interatomic distances in a protein and their projections perpendicular to the diffraction planes.
This is reflected in the local maxima shown in the distribution of average squared normalized structure-factor amplitudes as a function of resolution in these regions.

4.1.2.2 Glutaredoxin, as a test for an α-β structure at 1.45Å

Glutaredoxin was a test case of a small protein with more strands than helices, for which coordinates and experimental data can be found deposited in the PDB under the entry 1ABA. Its asymmetric unit contains 87 amino acids (880 atoms including solvent and ligands) and 45% solvent in the orthorhombic spacegroup $P2_12_12_1$. Three helices of 10, 12 and 14 amino acids are present. Data to a maximum resolution of 1.45Å are available.
At the time, the possibility to combine different fragments had not yet been implemented, so the fragment search was set up to locate three helices of 10 alanines, restricting the resolution for the rotation search to 2.1Å and using data to the full 1.45Å resolution for the translation search with Phaser. The rotation search was carried out in 2° steps and translation in 0.7Å steps.
Combining every rotation peak with every translation peak would soon lead to an unmanageable number of solutions. In order to control the flow of the search, the following limits were set: for every rotation or translation search, peaks under 75% of top were rejected, as is the default in Phaser. Furthermore, from each translation run after the first fragment, no more than 70 solutions were further pursued. After the packing check, surviving substructures were divided in groups of 400 solutions for rigid body refinement against the full resolution and pruning of duplicates. Only the 100 top solutions were kept. Expansion to the full structure with SHELXE works with substructures made up of 1, 2 and 3 helices. The structure was solved starting from 1 helix (50 atoms), in 3.6% of the cases, starting with 2 helices in 6.6% of the cases and from 3 helices in 15% of the substructures. In every SHELXE attempt, starting from phases derived from the partial structure, 5 runs of density modification made up of 30 cycles each were interspersed with autotracing. The density sharpening parameter (`v`) was set to 0 and reflections were extrapolated to a resolution of 1Å.

4.1.2.3 Glucose isomerase, a TIM barrel protein at 1.54Å

Of particular interest is the location of model fragments when the resolution of the data is high but worse than atomic and the size of the protein with no heavier elements than sulphur present exceeds a thousand atoms, as under such circumstances dual-space recycling *ab initio* is not expected to succeed.
In this test, data collected in-house for Glucose Isomerase to a maximum resolution of 1.54Å were used. The AU contains 368 amino acids, so both for its size and resolution the problem would be well beyond the reach of dual-space recycling *ab initio* methods. The protein fold is that of a TIM barrel, so that it contains a large proportion of helical structure, including some long helices (over 20 amino acids). For this structure, a complete helix of 23 amino acids (186 atoms) perfectly positioned is enough to provide starting phases that will allow SHELXE to solve the structure (the MPE slowly evolves from 77° to 25° in the course of 1000 cycles). In practice, more than a single helix will be needed, as it will be neither perfectly positioned nor complete (with side-chains). But as a start, the location of such a fragment (residues 150-172 from the final refined structure) was undertaken.

Using the full resolution of the data for the rotation search did not render the correct orientation in a rotation search, whereas this could be found truncating the data to 2.1Å, albeit not as the first solution (83% of top peak). When the rotation search is undertaken using data up to 2.6Å the rotation search fails as well.
Translation search using the full resolution (1.54Å) of the rotation solutions from data truncated to 2.1Å leads to the correct solution, further improved by rigid body refinement. Nevertheless, this experimentally placed substructure does not succeed in solving the structure with the equivalent SHELXE run rendering a solution with a MPE stuck at 71° whereas the incorporation of data extrapolation to 1Å and iterative autotracing accomplishes final phases with an MPE of 19.1°. So, although it is clear that a very small fraction of a structure may suffice to phase it, it appears to be strongly dependent on how perfect this substructure is. In general, more than one fragment will have to be located to make up for the deviation from the real structure, which cannot be known a priori.

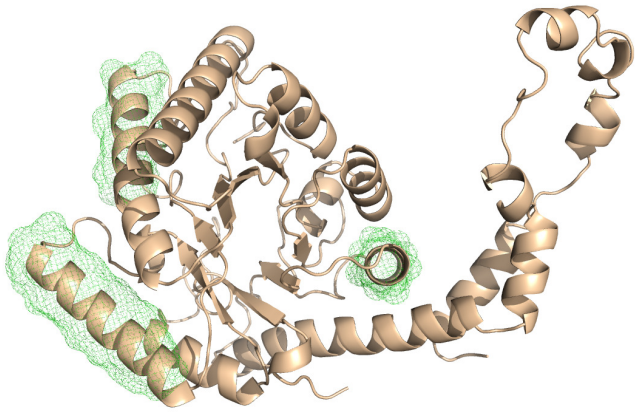

Figure 4.12: GI solution. Located models with highest FOM clustered around the same three helices in the target structure.

In principle, it would be possible to do a brute-force generation of substructures and test their expansion with SHELXE in all cases, but the need to locate several fragments simultaneously strongly increases the number of parameters and thus the calculation time required in such an approach. In the case of GI, the search for ideal model helices represents a more realistic scenario, looking for (Ala)12 helices, the first rotation search yields a high number of orientations, but all the highest solutions cluster around the same three helices in the model.
Selecting the helix cluster that corresponds to the 18 amino acids long helix comprising residues 64-81 in the model, re-scoring the LLG of the rotation solutions (LLG=2.40 for the best solution) on longer helices of 16 amino acids brings the maximum value up to LLG=3.29. For (Ala)18 LLG further increases to 3.66 whereas for (Ala)20 it drops to 3.44. This apparent sensitivity of the scoring function to the correctness of the fragment is worth exploiting as the length of straight helix the rotation corresponds to is not known until the structure is solved. No satisfactory translation solutions could be found for the (Ala)18 helix, which further indicates the necessity of using the rotation function to improve the search models, making them closer to the real structure by introducing curvature.

4.1.2.4 EIF5, solution of a mainly helical, 179 amino acids protein at 1.7Å

The structure of the Elongation Initiation Factor 5 was solved using synchrotron data collected to a resolution of 1.7Å, by the automated procedure setup with the strategy and parameters derived from our previous exhaustive tests sparsely summarized above.
As the structure was already determined, we knew it to contain 11 α-helices of various lengths ranging from 7 to 21 amino acids and different degrees of deviation from the ideal straight helix. Choosing a fragment contained in most of these secondary structure elements, the search was set up to locate 7 helices of 12 alanines, restricting the resolution for the rotation search to 2.1Å and using the full 1.7Å resolution for the translation search and rigid body refinement with PHASER. The rotation search was carried out in 3° steps and translation in 0.7Å steps.
Combining every rotation peak with every translation peak would soon lead to an unmanageable number of solutions derived from a combinatorial explosion. Indeed, by the time the search had reached the fourth fragment to be placed, over 50.000 parallel jobs would be required. In order to control the flow of the program, the following limits were set: for every rotation or translation search, peaks under 75% of top were rejected, as is the default in PHASER. Furthermore, from each translation run after the first fragment, no more than 70 solutions were further pursued. After the packing check, surviving substructures were divided in groups of 800 solutions for rigid body refinement and pruning of equivalent solutions. Only the 100 top solutions were kept. Expansion to the full structure with SHELXE was attempted with substructures made up of 3, 4 and 5 helices.

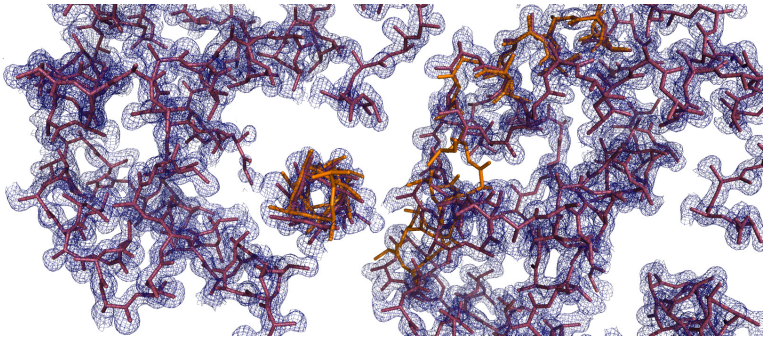

Figure 4.13: EIF5 solution. Calculated electron density map around deposited coordinates of EIF5.

The structure was solved starting from 4 helices (240 atoms), yielding a MPE of 31° against the deposited model. Figure 4.13 displays the resulting electron density map and the coordinates of the deposited model. In every SHELXE attempt, starting from phases derived from the partial structure calculated to a resolution of 1Å, 4 runs of density modification made up of 20 cycles each were interspersed with main-chain autotracing. The density sharpening parameter (`v`) was set to 0 and reflections were extrapolated beyond the experimental limit to a resolution of 1Å.

Substructures of 5 fragments also lead to an equivalent solution whereas no solution was achieved starting from 3 helices (180 atoms) even though the number of SHELXE iterations was doubled. The ARCIMBOLDO procedure was stopped beyond the 5th fragment.

4.1.3 Extension to incorporate stereochemical information into ARCIMBOLDO

The principles underlying ARCIMBOLDO can be exploited within three different scenarios beyond the originally proposed *ab initio* case. If no more than the native data and the amino acid sequence are known, the secondary structure prediction can be exploited to suggest the presence and length of α-helices, however, if additional information is available, it should be desirable to incorporate it into the frame of the method. ARCIMBOLDO has been extended to use all the previously available partial structural information: fragments that are more sophisticated than the ubiquitous main-chain α-helix can be proposed by modelling side-chains onto the main-chain, or extracted from low-homology models, because even when a conventional MR attempt fails, the target structure and the homologous one can share general fold and local similarities that can be exploited.
There are several possibilities to formulate more detailed hypotheses than a bare main-chain helix, derived from the secondary structure prediction, side-chains may be modelled on a predicted helix and various combinations of the most frequent -and thus probable[76]- conformations may be set up. Following this idea, in the ARCIMBOLDO code a procedure was implemented to use the design protocol within the Rosetta Software[77], it was intended to create more specific starting hypotheses that contributed with more scattering mass from the onset. As the side-chain conformations or the sequence of the helices targeted cannot be known a priori, the idea was again to pursue several hypotheses and proceed with the best scoring ones. In practice, one needs to model several sets of side-chains conformations onto a main-chain model, creating a library of hypotheses to later select the most encouraging one and proceed as in a regular *ab initio* case.
For the ARCIMBOLDO method we set up Rosetta to optimize all sequence positions in the main-chain model by fixing the backbone and applying the rotamer packing subprotocol. In Rosetta, side-chains are considered in a discrete set of favourable conformation called rotamers. Rotamer packing does not perform any backbone or sequence alteration, limiting itself to the optimal packing of side-chain rotamers. The algorithm changes the rotamer of a single amino acid and makes use of an energy function for evaluating the favorability of the new conformation. Since the starting information is so scarce, when considering an isolated helix, of totally unknown environment, in our scenario it is more important to try a large number of hypotheses with high variability than to attempt unconstrained optimization leading to very few, similar hypotheses. A large population of probable starting fragments is preferable, as they are later subjected to several selection procedures. For that reason, the number of fixed backbone simulations was set up to 1000, expanding the rotamer library by including deviations in the torsion

angles with a neighbour cutoff low enough to apply to all residues, regardless of packing.
Unfortunately for the ARCIMBOLDO approach, the Rosetta algorithms are designed to provide a very optimized prediction of protein structures, which is in contradiction with our need of producing a library of variable starting hypotheses. The Rosetta-ARCIMBOLDO coupling for the generation of side-chains did not lead to promising results, although the implementation is still available in the code for possible new approaches in the future.
From this experience we decided to create our own libraries of starting hypotheses, also because the criterion to create the library could originate in very diverse contexts and we would want flexibility. Typical scenarios could span structures belonging to a family with a structurally conserved active site that could be used as a search fragment, low homology related structures that failed as classical molecular replacement models but could provide small fragments close enough to the target structure without it being possible to predict beforehand which should have low enough geometrical deviation to the target structure, or simply the amino acid sequence of the predicted helices in a number of combinations of the most favourable rotamers that would pass a packing check. Using each and every one of these models as a start for a full ARCIMBOLDO run would considerably increase the total computation time, so it would be necessary to attempt to identify promising start hypotheses early on. Ideally, scoring alternative fragments at the stage of the first rotation or translation search would be convenient since it would be achievable without increasing the total running time. At this stage, jobs are launched sequentially in the standard procedure, as in the absence of previous hypotheses, the search of the first helix is a concatenation of single processes. Thus, we set to test the rotation and translation performance of a library of alternative models at different resolutions to determine whether comparable fragments could be discriminated on any figure of merit, and under what circumstances.
The following tests were performed with PRD2, an all-helical structure comprising 220 residues in the asymmetric unit and diffracting to 1.95Å, whose solution *ab initio* will be described in more detail, as it was the first unknown protein determined with ARCIMBOLDO[53]. By the time the use of libraries was implemented, the structure had already been solved, and thus here it is considered as a test structure.
Figures 4.14 and 4.15 show the values of LLG and Z-score from the rotation search for the various model candidates at resolutions ranging from 3.5 to 2Å in 0.1° steps. The helix selected to vary the sequence corresponds to the first one to be located in the search for a polyalanine helix, thus ensuring that the comparison of maximum figures of merit will be meaningful, as different helices will give rise to different FOM values, influenced for instance by their B-values.
The models compared are:

- **real_part.pdb** (green) is a fragment comprising residues Leu74 to Gln87 cut out from monomer A in the final structure but with artificial B-factors set to the same uniform value as in each of the model fragments.
- **14alaninas.pdb** (blue) is the ideal helix composed of 14 alanine residues.

- **14ala_sidechains.pdb** (yellow) is the above helix with side-chains in the most represented conformers from Leu74 to Gln87, modelled with SCWRL4 [78].
- **14ala_side-chains_better.pdb** (red) is the same 14aa main-chain helix with the side-chains in the standard conformers that are closest to the final structure.

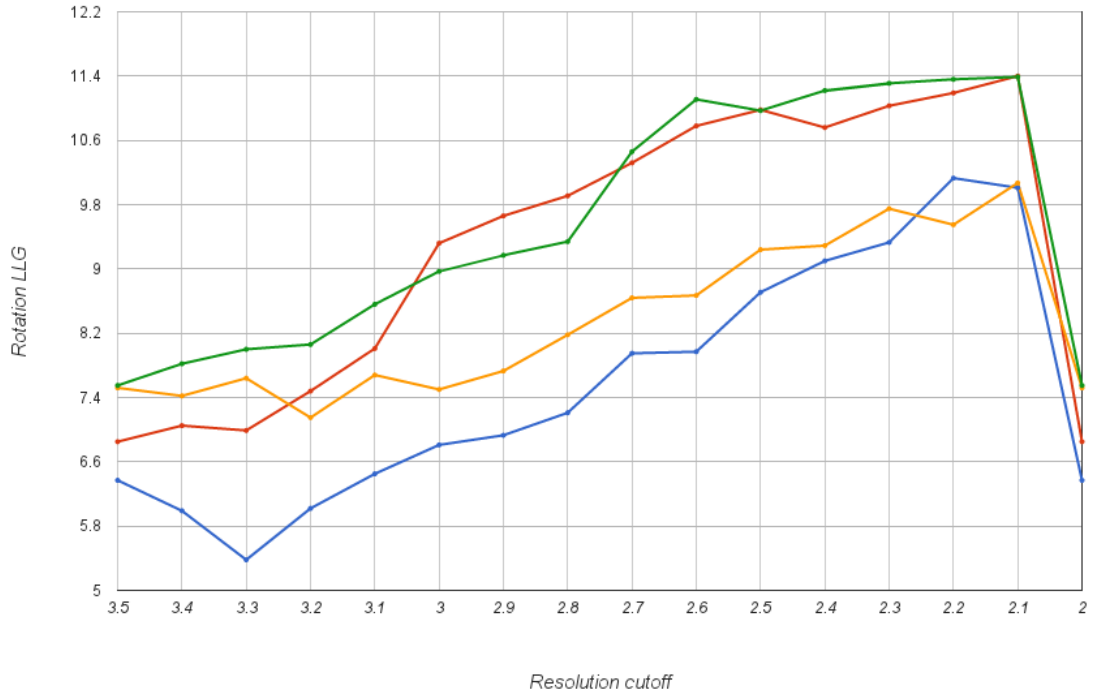

Figure 4.14: LLG values rendered by the Phaser fast rotation function at varying resolution for alternative models corresponding to the same helix. Green: real_part, blue: 14alaninas, yellow: 14ala_sidechains, red: 14ala_side-chains_better

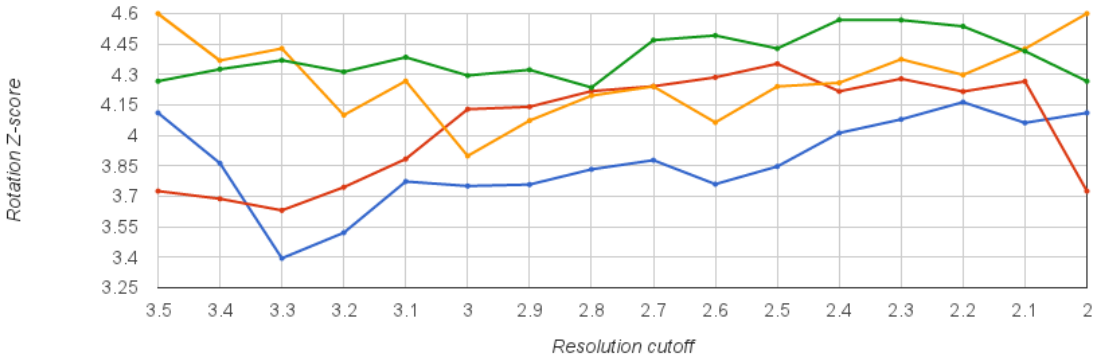

Figure 4.15: Z-score values rendered by the Phaser fast rotation function at varying resolution for alternative models corresponding to the same helix. Green: real_part, blue: 14alaninas, yellow: 14ala_sidechains, red: 14ala_side-chains_better

In general, the LLG values of the candidates with side-chains are higher than the polyalanine model -as expected- due to the increase in number of atoms and thus in scattering mass of the fragments. At the high resolution end and up to 2.7Å the discrimination by LLG appears to reflect reality: the green line corresponding to the perfect fragment marks the top of the graphic, as it does at low resolution, below 3Å. Remarkable, though, is the fact that at 2Å results are misleading. This could be due to the same effect previously discussed, that renders the performance of the rotation function more reliable between 2.1 and 2.4Å, or simply because given the resolution limit of 2Å in this dataset, the test could be influenced by the noise level in this last shell. The effect of individual conformers is sensitive at high resolution but not error-free and should not be overinterpreted. Discrimination can be aided by Z-score but it does not provide a good basis for comparison unless it shows distinctly high values. It is rather in the context of the translation function that it identifies the most promising results.
The rotation function LLG values offers a preliminary test to rank the alternative hypotheses. It can be quickly performed since the resolution shell needed for selecting the most promising candidate among similar models corresponds to the same to be used for the rotation search. The flow of the lines in the figure for the LLG comparison shows that it works best at high resolution between 2.1 and 2.4Å, but the fact that even at lower resolution the perfect model scores better than the approximations suggests that geometry improvement against the data as early as the rotation stage can be further developed.
An ARCIMBOLDO run initiated against the discussed library and set to choose the model with the highest LLG value for the first rotation function, selects the model with the optimal conformers and, upon placing it twice, is able to succeed in phasing the structure, showing that the approximation standard conformers provide is good enough for the purpose of the method. Furthermore, whereas three polyalanine helices were needed to phase the structure, just two fragments of the more complete model achieve the same result.
It has to be remarked at this stage that although perfect models perform best, phasing can be nevertheless obtained with models with errors as well as with slightly misplaced models. ARCIMBOLDO provides the possibility of testing libraries of alternative models by sending parallel jobs to calculate the first rotation and/or translation functions and choosing LLG or Z-score from either of these functions to select the model to be adopted on the location stage of the first fragment(s).
The library of hypotheses may be explicitly input into the script or passed in the form of the path to an external file. That file must list either several PDB hypotheses or several `.tar.gz` files containing multiple PDB files. The `.tar.gz` option was mainly created for choosing among collections of similar ensembles that, at the same time, could be classified and divided following another criterion, *e.g.* helices with different degrees of curving, helices with the same side-chains in different conformations, equal length fragments cut out from different homologues...
Once the preliminary test is finished, the model selected is used as a model to locate through the same original *ab initio* procedure. Naturally, in contrast to the ubiquitous main-chain helix, as particularized fragments are being used, they will need to be combined with other different particular or general models. Thus,

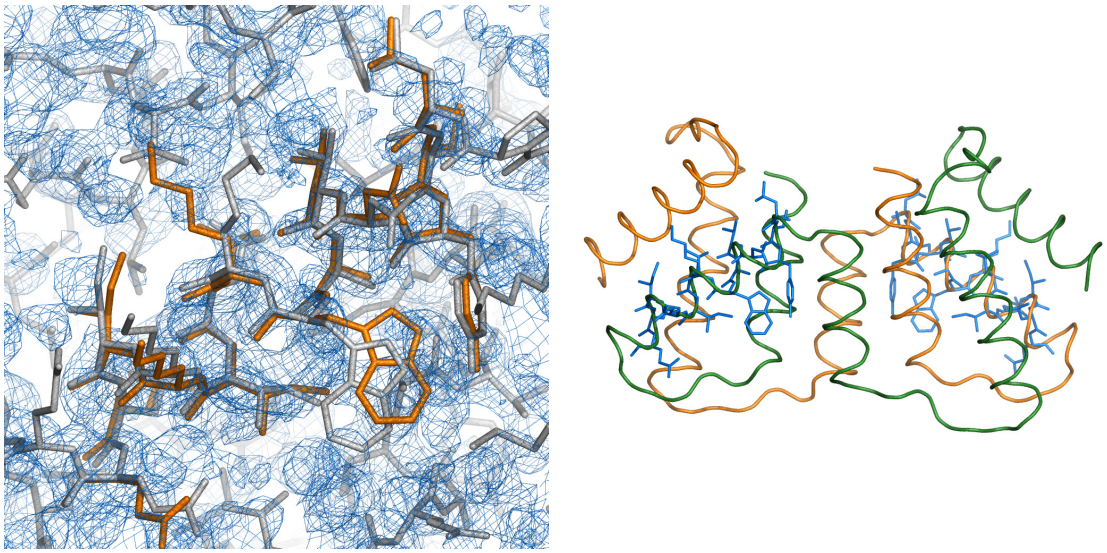

Figure 4.16: Left: PRD2 density map surrounding the deposited coordinates and the superposed two model helices with side-chains that allowed the final solution. Right: Solution of the test case after placing two copies of the model with optimal conformers and expanding from this stage.

ARCIMBOLDO also allows to perform searches specifying any combination of different models, which is useful as well to locate helices of various lengths following a secondary structure prediction.

4.1.4 Extension to incorporate complementary experimental information into ARCIMBOLDO

Experimental phases of anomalous substructures can be exploited as well and we envisaged three ways to incorporate their practical use in ARCIMBOLDO. Two of them are already implemented while the 3rd one is not yet automated.

1. Provided that there is anomalous difference or MAD data available, a run with ARCIMBOLDO can be set up to locate anomalous fragments like disulphide bridges, heavy-atom clusters, etc.

2. In case that an anomalous substructure has been previously determined by another program (for example SHELXD) a run with ARCIMBOLDO can be set up to use that previous information to constrain the location of model fragments, like α-helices, and to ensure a common origin to both sources of information. For this cases, the use of Phaser brute rotation/translation functions might be helpful since the location of the substructure can be used as a starting partial solution to restrict the space for the search of a given model.

3. If partial fragment phases are available -achievable by using ARCIMBOLDO- an anomalous cross-Fourier map can be calculated from the current density modification phases in order to determine the anomalous substructure.

Combination of both sources of information: fragments located and weak experimental phases provides better maps and recycling may be used to further improve the anomalous structure.

It is possible to search for substructures made up of anomalous scatterers or heavy atoms provided a suitable model is known a priori. Although this is not a frequent scenario, our test case contains several anomalous scatterers in a known geometry as it belongs to a family characterized by a fold with a known disulphide-bridge pattern and high variability in the functional loops, where the coordinates for the S atoms can be taken from an homologous structure.

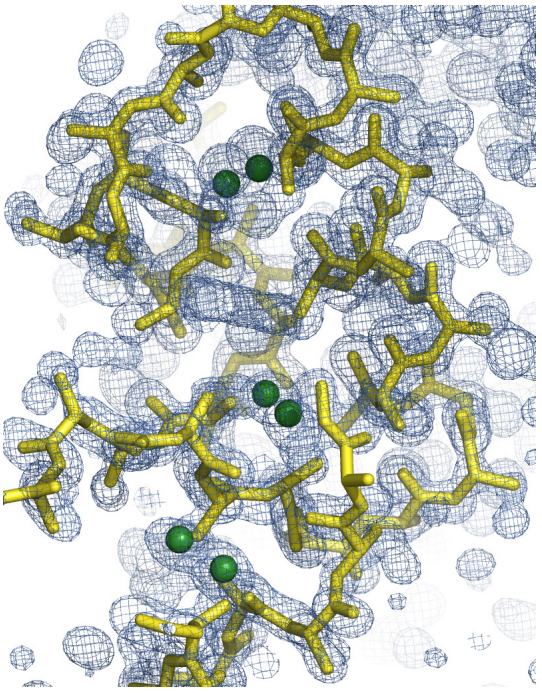

Figure 4.17: Test case of viscotoxin A1 phased locating the anomalous substructure of the S atoms (green spheres) in hellethionin D (NMR structure; PDB entry 1nbl[79]). Electron-density map after expansion, shown in blue contoured at 1sigma and including data extrapolated to 1.0Å, and the polyalanine trace of the solution obtained for viscotoxin A1 after the first fragment.

The viscotoxin A1 (VTA[80]) structure in spacegroup $P4_32_12$ provides such a suitable case. Data recorded to 1.25Å resolution using an in-house CuK system show significant anomalous signal derived from six cysteines involved in three disulphide bridges present in each of the viscotoxin molecules. A fragment consisting of the six cysteines can be extracted from another PDB entry displaying the viscotoxin fold, such as the NMR structure of hellethionin D[79]. In the fragment, the remaining atoms of the cysteine residues should be retained with occupancy 0. They are not part of the fragment as they do not present anomalous diffraction, but are still useful to compute packing clashes and discard impossible solutions.

Search and optimization with this anomalous fragment was performed cutting the anomalous data to a resolution of 2Å. The `.mtz` file passed on to Phaser must contain the δF or FA data and their standard deviations. These columns are set as F and SIGF in the ARCIMBOLDO script. The first fragment produces 20 translation solutions, of which five are unique and have similar figures of merit. At the expansion stage, SHELXE uses the anomalous substructure and the file containing the anomalous differences and phase shifts to phase the native data, in combination with fragments, if present. In this case, 30 cycles of density modification and three cycles of autotracing already bootstraps with one six- S-atom substructure. If the second anomalous fragment is searched for, the correct solution is even clearer from the figures of merit (LLG = 180 versus 119 and TFZ = 8.6 versus 5.2 for the next best). Fig. 4.21 shows the density map and main-chain trace obtained as well as the sulphur substructure.

4.1.5 Practical use of ARCIMBOLDO and tutorial

The development of ARCIMBOLDO has allowed the solution of over a score previously unknown crystal structures where conventional approaches in the hands of competent crystallographers had failed. In this sections the use of ARCIMBOLDO in its different modes will be described in technical detail and discussed with various datasets. Structures first solved by ARCIMBOLDO or test data used in its development have been chosen to best describe, both the results obtained and its application by users, in the way of a tutorial that should guide parameterization of other cases. All the information available should be used to enforce ARCIMBOLDO's success when dealing with larger structures and poorer data; the following figure shows a simplified graphical summary of the method, that helps to explain the ARCIMBOLDO flow depending on the starting information. The same figure will be used to describe each test case by highlighting the path that should be followed depending on the circumstances.

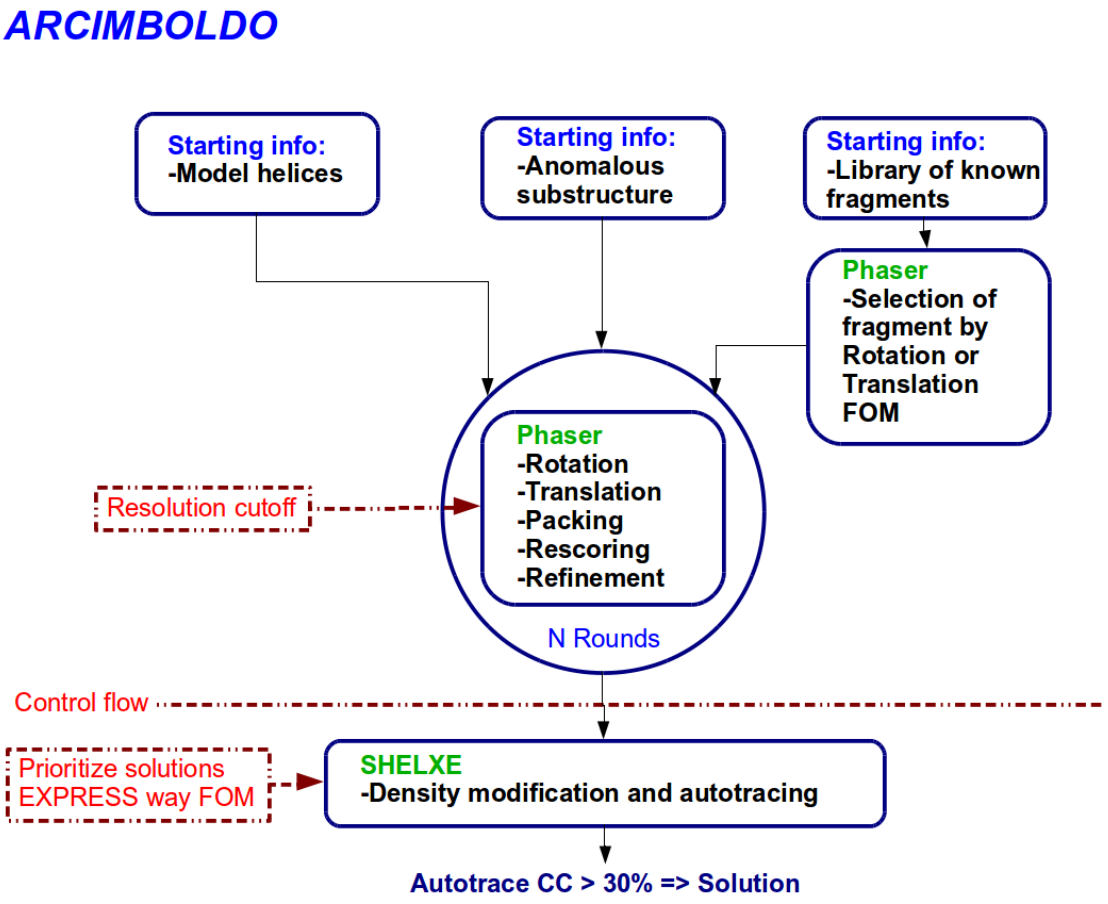

Figure 4.18: Schematic flow of the ARCIMBOLDO procedure.

4.1.5.1 Preliminary considerations

- For a successful *ab initio* solution, the higher the resolution available the better, but in general, complete data of good quality to a resolution of 2Å are required for an ARCIMBOLDO solution. Often -for the rotation function and occasionally for the translation function- resolution limits between 2.5 - 2.1Å are effective at the fragment search stages, however, the density modification expansion from a very reduced part of the total scattering mass is greatly enhanced through the availability of higher resolution and its success is instrumental to identify the correct solutions.

- Completeness of the data is of capital importance for ARCIMBOLDO to succeed. In general, crystallographers tend to use data coming from the best crystal, but a dataset averaged from different crystals -even though the isomorphism of some of them may not be as perfect as would be desirable- will offer a higher redundancy and possibly a 100% completeness at least in most resolution shells.

- To run this tutorial some computational requirements must be fulfilled, a HTCondor-grid[47] must be available, as well as the crystallographic programs Phaser[54] and mtzdump, within CCP4 suite[56] and SHELXE[55].

- All the command line expressions in the present work correspond to a Linux environment with tcsh.

- The ARCIMBOLDO algorithm is written in Perl and although its use to solve a structure does not require to be an expert Perl programmer, there are some basic concepts that should be known before its use:

  - The ; is used as a statement delimiter.
  - A scalar variable is a plain, simple, one-dimensional value. It is always preceded by the $ sign and can hold values of number or string types.
  - An array is a single entity made up of a collection of scalars, each single value contained in an array has an associated and unique key that identifies it. Arrays are lists indexed by sorted numbers starting from any number, although usually from 0. In Perl, they are defined by names preceded by @ and to access their elements $ is used, followed by the name of the array, plus the index number of the element enclosed in square brackets.

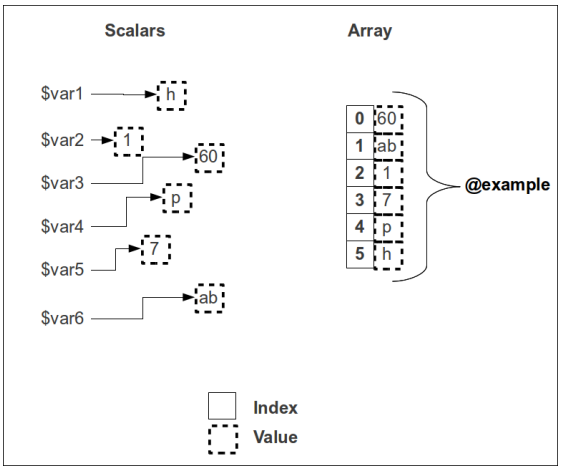

Figure 4.19: Graphical representation of scalars, array and hash concepts.

For example, using the information provided in the figure:

$$\$var1 = h$$
$$\$example[2] = 1$$

– Single quotation marks are used to enclose data that must be taken literally, they are frequently used in ARCIMBOLDO code to define paths. It is possible to quote without quotes using the following notation:

$$\mathbf{q}(text\_to\_quote);$$

- All the paths to the input files provided to download are relative to the folder where ARCIMBOLDO is downloaded.
- When selecting the steps for the sampling of rotation and translation functions, one can imagine them as represent the hole dimensions of the mesh in a fishing net. To get small fishes -our model fragments- a finer mesh is needed, thus, it is necessary to select a fine sampling in order to find the correct location of the models. However, a balance must be kept because a finer mesh requires more calculations and no intrinsic hard limits are provided to the number of jobs being run. Different systems could handle different loads. ARCIMBOLDO sampling defaults are empirical values that have been established in our test cases while developing the method, they can be kept or modified according to the user needs, but in any case correspond to smaller values than Phaser's defaults.

4.1.5.2 Setup for general variables

There are some setup variables that should remain constant for every run once defined for a given work environment: program paths, the decision of producing or not output files that are not essential to the flow of ARCIMBOLDO and some HTCondor variables suited to the particular grid being used:

- Program paths:
  - `$phaser_location`: holds the path to the program Phaser, which performs the fragment placement strategy of the ARCIMBOLDO method.
  - `$mtzdmp_location`: holds the path to the mtzdmp script which invokes the program mtzdump, belonging to the CCP4 suite. It is used to read the .mtz binary file containing experimental diffraction data.
  - `$shelxe`: holds the path to the program SHELXE, which is in charge of the density modification and autotracing stage of ARCIMBOLDO.
- Output:
  - `$xyzout` and `$hklout`: Switches on/off the output of coordinates and mtz files containing the phasing information for every step of the model location. It is recommended to set them off

    ```
    $xyzout = 'XYZOUT OFF';
    $hklout = q(HKLOUT OFF);
    ```

    in order to optimize speed and memory space because the mentioned files are not used by ARCIMBOLDO. Hence, unless the user is concerned about the content of these files, `$xyzout` and `$hklout` can remain OFF.
- HTCondor Grid performance:
  - `$long_log`: Can take a value of 0 or 1 and it is meant to produce concise or verbose log files, mainly about HTCondor jobs:

    ```
    $long_log = 0;
    ```

    however, it shows information of interest for ARCIMBOLDO developers rather than for crystallographers. Thus, for the purpose of users it should be set as concise (0).
  - `$nice_user`: Switches "True" or "False" to indicate whether the HTCondor jobs can compete for the computer resources (False) always following the HTCondor central manager priority standards among the HTCondor users, or must be scheduled with low priority (True), waiting or leaving the resources as soon as other HTCondor jobs require them.

    ```
    $nice_user      = q(False);
    ```
  - `$requirements`: Absolutely dependent on the computing resources available in the HTCondor-grid. This variable specifies the particular resources where HTCondor jobs must be run, for example limiting jobs to be run on Intel CPUs with dedicated memory over 2010Mb:

    ```
    $requirements = q((memory > 2010) && (Arch == "INTEL"));
    ```

An extensive explanation of the ClassAd mechanism of HTCondor can be found in the HTCondor documentation, but in general, this aspects should be the concern of the system manager setting up the condor grid rather than of particular users.

4.1.6 *Ab initio* case results on the first new structure phased, 3GHW.

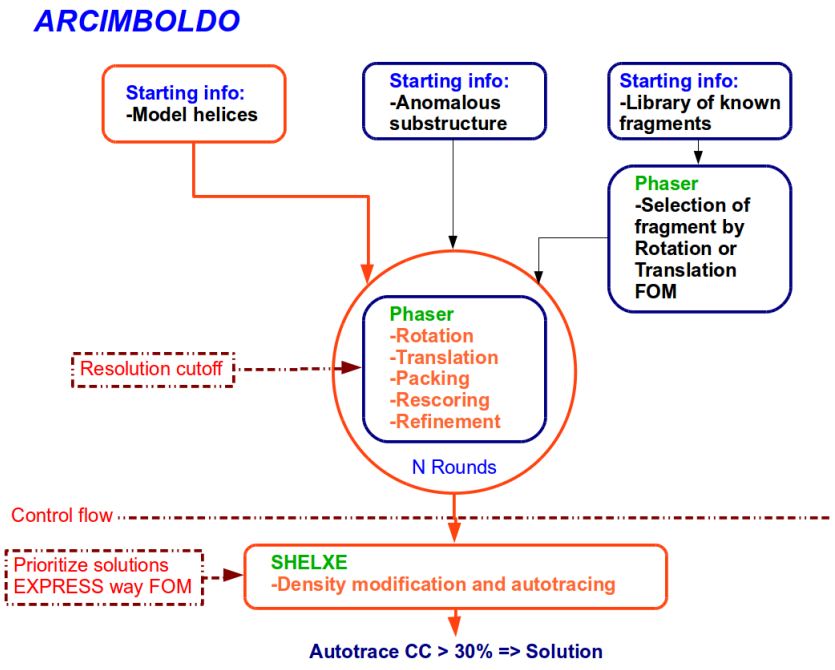

Figure 4.20: Schematic procedure for ARCIMBOLDO *ab initio* path.

Highlighted in orange is described the simplest case for the ARCIMBOLDO procedure: the *ab initio* path. It can be used when all the available previous information is reduced to a set of native data and the amino acid sequence. From this, secondary-structure prediction algorithms [81–83] can derive the number and length of expected α-helices and β-strands.
α-Helices tend to be very constant in their main-chain geometry, especially over a short range of protein residues (10-14aa). In contrast, the higher variability and shorter span of β-strands make them less useful as search fragments and, so far, no *ab initio* successes have been attained locating single main-chain strands.

4.1.6.1 Input files

The input files for this case are available for download from `http://chango.ibmb.csic.es/ARCIMBOLDO/` under the name of **PRD2: *ab initio* test case**. They consist of:

The native dataset in CCP4 mtz format: `stefan.mtz`. Any .mtz file to be used under the ARCIMBOLDO - *ab initio* scenario must contain the structure factor amplitudes and their standard deviations; they are usually associated with the label notations "F" and "SIGF" respectively, but labels in the `.mtz` file can be arbitrarily assigned.

The model fragment to be located in pdb format: `th14.pdb`. Any `.pdb` file to be used by ARCIMBOLDO must contain the atomic coordinates for the atoms of the model.

The native dataset in SHELX hkl format: `stefan.hkl`. Any `.hkl` file to be used must contain a reflection list with intensities or amplitudes and their standard deviations. It can be obtained from the `.mtz` file using the mtz2various program of the CCP4 suite or from other formats with the program xprep.

The instruction file for the density modification procedure through SHELXE: `stefan.ins`. Any .ins file to be used by ARCIMBOLDO must contain information about the cell, symmetry and parameters to control the SHELXE run.

4.1.6.2 Data analysis

PRD2 (PDB entry 3GWH[53]) was the 1st previously unknown macromolecular structure solved using ARCIMBOLDO. It was solved, as described in the Nature Methods publication, by pure *ab initio* in spacegroup $P2_1$ with diffraction data recorded to 1.95Å. Its AU contains 222 protein residues making up 10 helices, 8 of them with lengths comprised between 14 and 20aa.

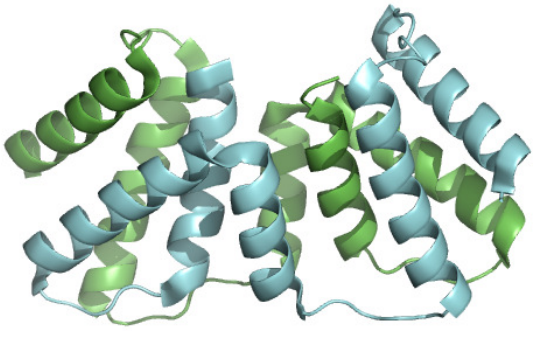

Figure 4.21: *Ab initio* PRD2 solution after model building and refinement, PDB code 3GHW.

Classic approaches of MR failed trying to solve the PRD2 crystal structure using structures from available homologues as search fragments. ARCIMBOLDO succeeded using a poly-ala helix of 14aa as search model. The program Phaser was used to search 8 times for the model fragment and the resulting MR partial solutions were expanded by the density modification and autotracing algorithms of SHELXE. Solutions were obtained already from some combinations of three helices, so that ARCIMBOLDO can be terminated before completing the programmed search.

Data were collected by Christian Grosse at 100K on a home source with a copper rotating anode, 3 circle diffractometer and CCD Bruker detector at the Structural Chemistry Department in the Univertity of Göttingen. A brief analysis of the

PRD2 diffraction data with the program XPREP from the SHELXTL software package[84] shows a 13.7 signal to noise value. A value of I/SIGMA less than 5 indicates that the overall data intensity is weak, I/SIGMA around 10 is a typical value and I/SIGMA greater than 20 points to strong intensity implying that, possibly, the resolution limit has been underestimated and higher resolution could have been recorded. However, I/SIGMA is just an indicative parameter that might be affected by several causes. Among others, a spacegroup (SG) with high symmetry can lead to underestimated sigma values, producing higher I/SIGMA values, in any case it should not be taken as a categorical indicator of data quality.

Lattice exceptions:	P	A	B	C	I	F	Obv	Rev	All
N (total) =	0	12316	12287	12293	12312	18448	16389	16413	24624
N (int>3sigma) =	0	8117	7982	8027	8140	12063	10850	10780	16213
Mean intensity =	0.0	10.0	10.2	10.1	9.8	10.1	10.3	10.1	10.2
Mean int/sigma =	0.0	13.6	13.6	13.7	13.5	13.6	13.7	13.6	13.7

Figure 4.22: XPREP output of ISIGMA for PRD2.

Data quality can be assessed looking at the values of completeness, redundancy, signal to noise and regression or correlation coefficients calculated from equivalent reflections.

Resolution	#Data	#Theory	%Complete	Redundancy	Mean I	Mean I/s	Rmerge	Rsigma
Inf - 8.09	193	195	99.0	1.82	83.7	62.11	0.0113	0.0151
8.09 - 5.35	451	452	99.8	1.91	22.9	47.36	0.0141	0.0174
5.35 - 4.22	647	647	100.0	1.93	42.0	55.36	0.0101	0.0152
4.22 - 3.69	641	641	100.0	1.94	31.7	49.71	0.0133	0.0165
3.69 - 3.35	640	640	100.0	1.95	20.8	41.38	0.0175	0.0195
3.35 - 3.10	657	657	100.0	1.95	12.4	32.49	0.0224	0.0255
3.10 - 2.91	651	651	100.0	1.96	9.1	27.33	0.0232	0.0299
2.91 - 2.77	634	634	100.0	1.95	6.6	22.85	0.0324	0.0377
2.77 - 2.65	627	627	100.0	1.96	6.2	20.63	0.0385	0.0420
2.65 - 2.54	677	677	100.0	1.96	5.0	15.33	0.0476	0.0586
2.54 - 2.45	664	665	99.8	1.96	4.2	11.22	0.0694	0.0818
2.45 - 2.38	584	585	99.8	1.96	3.8	9.78	0.0787	0.0983
2.38 - 2.31	657	657	100.0	1.97	3.4	8.03	0.0921	0.1202
2.31 - 2.25	629	629	100.0	1.96	3.1	6.41	0.1463	0.1550
2.25 - 2.19	686	686	100.0	1.92	2.8	5.21	0.1701	0.1898
2.19 - 2.14	668	668	100.0	1.90	2.5	4.61	0.1592	0.2158
2.14 - 2.09	697	697	100.0	1.86	2.3	3.83	0.2035	0.2573
2.09 - 2.05	613	613	100.0	1.84	2.4	3.54	0.3858	0.2716
2.05 - 2.01	658	658	100.0	1.82	1.7	2.74	0.2653	0.3741
2.01 - 1.98	531	531	100.0	1.82	1.6	2.55	0.2776	0.3960
1.98 - 1.95	650	673	96.6	1.73	1.3	2.13	0.2897	0.4662
2.05 - 1.95	1839	1862	98.8	1.79	1.5	2.47	0.2761	0.4090
Inf - 1.95	12855	12882	99.8	1.91	10.2	18.90	0.0381	0.0460

Figure 4.23: PRD2 XPREP analysis of intensities statistics.

The line next to last states the statistics for the last resolution range and indicates whether ARCIMBOLDO has any chance to succeed on the available data. In this case the redundancy values are artificially low as data were processed and

merged previously, but friedel pairs are conserved. Nevertheless, for raw data, the higher this value the better. The same line also shows that the data are practically complete even at the resolution limit and the value of I/SIGMA close to 3 indicates that the intensity of the reflections in the last resolution range is strong enough to be clearly distinguished from the noise.

4.1.6.3 Definition of variables for ARCIMBOLDO/Phaser

After assessing the data, the next step would be the definition of ARCIMBOLDO variables. Assuming that the environment variables (see Setup for general variables 4.1.5.2) are already defined, we can focus on parameters that particularly describe the ARCIMBOLDO experiment. The path to the file containing the experimental diffraction data must be provided through the variable `$mtz_file`. In the available *ab initio* example the variable contains a path relative to the downloaded folder:

```
$mtz_file = q(stefan.mtz);
```

The same `mtz` file contains labels determining the structure factor amplitudes and their standard deviations. Phaser needs this information to proceed with the fragment search, the labels can be extracted from the binary `.mtz` file using the CCP4 program mtzdump by typing in the command line:

```
mtzdump stefan.mtz
```

The required labels can be read from the output shown in the figure below, the OVERALL FILE STATISTICS table. The ones corresponding to structure factor amplitudes and their standard deviations must correspond to types "F" and "Q", respectively:

```
OVERALL FILE STATISTICS for resolution range   0.001 -   0.264
=======================


Col Sort    Min    Max    Num      %     Mean     Mean   Resolution   Type Column
num order              Missing complete          abs.   Low    High       label

  1 ASC    -19     18      0  100.00     -2.1      7.2  35.98   1.95   H  H
  2 NONE     0     33      0  100.00     12.4     12.4  35.98   1.95   H  K
  3 NONE     0     19      0  100.00      7.4      7.4  35.98   1.95   H  L
  4 NONE   0.0   19.0      0  100.00     9.55     9.55  35.98   1.95   I  FreeR_flag
  5 NONE  10.5 1132.3     39   99.70    85.41    85.41  24.27   1.95   F  F
  6 NONE   0.8   27.0     39   99.70     4.92     4.92  24.27   1.95   Q  SIGF
  7 BOTH  -0.0   -0.0  12316    4.31     0.00     0.00  21.79   1.95   D  DANO
  8 BOTH   0.0    0.0  12316    4.31     0.00     0.00  21.79   1.95   Q  SIGDANO
  9 NONE  10.5 1132.3     39   99.70    85.41    85.41  24.27   1.95   G  F(+)
 10 NONE   0.8   27.0     39   99.70     4.92     4.92  24.27   1.95   L  SIGF(+)
 11 NONE  10.5 1132.3  12316    4.31   102.74   102.74  21.79   1.95   G  F(-)
 12 NONE   1.0   25.1  12316    4.31     6.25     6.25  21.79   1.95   L  SIGF(-)
 13 BOTH     1      1     39   99.70      1.0      1.0  24.27   1.95   Y  ISYM


No. of reflections used in FILE STATISTICS    12871
```

Figure 4.24: Example of OVERALL FILE STATISTICS table from mtzdmp output.

Their extraction cannot be automated, given that an `.mtz` file might contain several datasets, that is, different sets of amplitudes and their associated errors. Thus the user needs to make the choice which to use. After identifying the labels, they are passed to `$labin` as follows:

```
$labin = q(LABIN  F=F SIGF=SIGF);
```

The label for the structure factor amplitudes must be placed after "`F=`" and for the standard deviations after "`SIGF=`". The composition of the crystals required by Phaser can be expressed in two alternative ways. Given that a secondary structure prediction is needed before using ARCIMBOLDO and that the molecular weight can be easily calculated from the available protein sequence, all the ARCIMBOLDO tests have been performed using the convention for molecular weight and the expected number of copies of the molecule in the AU. The molecular weight value for PRD2 example follows "`PROTEIN MW `" and the number of copies of the protein in the AU is written after "`NUMBER `":

```
$composition = q(COMPOSITION PROTEIN MW 12000 NUMBER 2);
```

The resolution range must be set up for both, rotation and translation searches, since our results on test proteins previously described [53] have shown that the best resolution to find the correct rotation does not have to be the most suitable for the translation search.
When locating a helix, its periodicity and the bond distances favour that correct rotations are already identified at resolutions of 2.5Å, but using full resolution is advantageous for translation searches.
For this example, both, rotation and translation searches will use data with resolution limited to 2.0Å as shown at the declaration lines in

```
$resolution_rot = q(RESOLUTION 99.0  2.0);
$resolution     = q(RESOLUTION 99.0  2.0);
```

The 1st number may correspond either to the lower resolution available or equal to 99.0 to indicate that no low resolution cutoff will be used.
The next variables involve the main point of the experiment: the definition of the target model and the instructions to perform the search. In the case of PRD2 it was necessary to locate 3 times a 14aa helix using the Phaser fast rotation and fast translation functions, afterwards, a SHELXE expansion of the partial solutions succeeded in solving the crystal structure.
The model search is a concept described by several parameters, it can be seen as the summation of the model description and the parameterization of its search. To describe the model the main parameters relate to the choice and characteristics of the model itself. It must be declared as an anomalous or conventional fragment (not anomalous), the anomalous nature of the fragment is irrelevant for the model location as it will simply be located against anomalous differences or MAD FA values declared in the LABEL definition, but it is essential for its treatment in the density modification step.

Another parameter that describes the model is its identity with the unknown structure in terms of the rmsd. The default 100% identity should be kept while working with polyalanine helices, as they are expected to identically fit a substructure within the whole structure. For comparison purposes, it is preferable to stick to this value even when models incorporate side-chains or other stereochemical guesses are hazarded.
The fragment search is described by the target to be used in the search: the fast or brute rotation/translation functions and the number of times the model(s) must be located. In the current case only one type of model needs to be located, but complex protein structures might require a more sophisticated search approach, where different models might be offered as initial hypotheses to be placed alternatively or sequentially. In order to provide suitable descriptions for all these scenarios, the mentioned parameters describing models, their anomalous nature, rmsd, rotation and translation function targets, and instances a given model should be located are controlled by array type variables (reference in page 59) emulating linked lists commanded by the list of models. A graphical illustration to help understanding the connections relating such lists might be the following one:

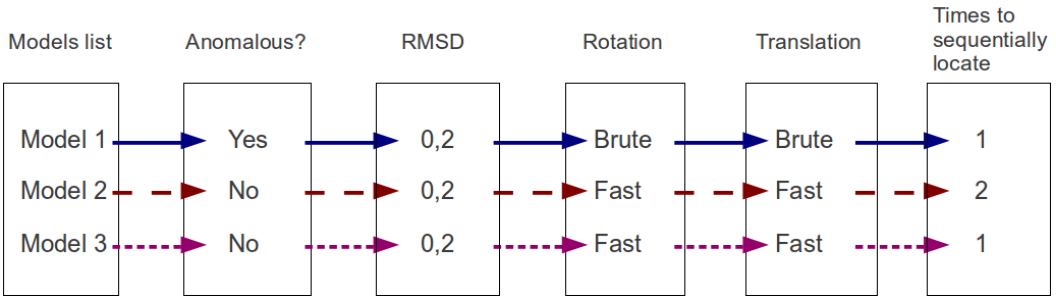

Figure 4.25: Graphical illustration of model search parameterization.

Adapting it to the PRD2 case, the graphical interpretation would be reduced to:

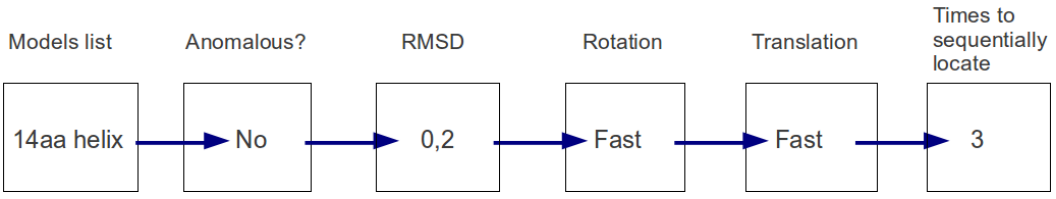

Figure 4.26: Graphical approach of PRD2 search parametrization.

In the figure describing the algorithm, the type of the linking arrows represent the array indices, all the arrays start the indexation at 1 and their respective names are `@pdb_file`, `@an_pdb`, `@rotf`, `@traf`, `@ident` and `@search` :

- `@pdb_file`: holds the path(s) to the file(s) with the atomic coordinates of the model fragment(s) indexed by the order in which they should be located. In

the case of PRD2, the same 14 alanines helix must be placed 3 times before succeeding, therefore there is only one model to be defined:

```
$pdb_file[1] = q(th14.pdb);
```

- `@an_pdb`: delimits whether or not the models in `@pdb_file` must be treated as describing a structure of anomalous scatterers for the density modification procedure to be run with SHELXE, the accepted values are 1 or 0, representing true or false. PRD2 is not an anomalous case, and to link this information to the model helix defined in `@pdb_file`, the index [1] must coincide:

```
$an_pdb[1] = 0;
```

- `@rotf`: defines the Phaser rotation function to be used when searching for the correct orientation of the model fragment(s). Accepted values are frf, for fast rotation function, or brf, for brute rotation function. PRD2 was solved applying the fast rotation function to the helix declared in index 1 of `@pdb_file`, therefore:

```
$rotf[1] = q(frf);
```

- `@traf`: defines the Phaser translation function to be used when attempting to find the correct position of the model(s) once rotated. Accepted values are ftf, for fast translation function, or btf, for brute translation function. PRD2 was solved applying the fast translation function to the helix declared in index 1 of `@pdb_file`, consequently:

```
$traf[1] = q(ftf);
```

- `@ident`: defines the rms coordinate error expected for the model versus the target structure to be solved. Since the models used in ARCIMBOLDO are very small, they need to be very accurate as well, so that locating them may rely on the high resolution data; this value can be set up to 0.2:

```
$ident[1] = q(0.2);
```

- `@search`: defines the number of times each model-fragment has to be located. It is expressed as a range by typing the 2 limiting numbers of the search range within a parenthesis. In the PRD2 structure, the 14aa helix can be found 8 times, however, correctly placing 3 helices is enough to phase the whole structure, thus, lowest value in the search range is 1 and highest is 3:

```
$search[1] = q(1,3);
```

The selected numbers are reflected in the folder structure containing the output of the run. This topic is further explained.
To find the correct orientation and position of the model fragment(s), a sampling mesh is required, default Phaser values being unsuitable for our particular scenario. It is provided in terms of degrees for the rotation and angstroms for the translation and can be seen as the holes of a fisher net to be used for "fishing" the model fragments. ARCIMBOLDO models are usually "small fish", accordingly, the sampling must be set up as an unusually fine net, while avoiding to overload the grid infrastructure to preserve an affordable time execution. In the PRD2 case, the rotation step to set up the sampling for the fast rotation function was defined as 1.5° and the step chosen for the fast translation function was 0.7Å

```
$rot_frf_sampling = q(SAMPLING ROT 1.5);
$tra_ftf_sampling = q(SAMPLING TRA 0.7);
```

The helix to be located is smaller than most of the expected helices in the structure, hence, the number of potentially correct solutions can be very large because a 14aa helix can accommodate 7 times in a 20aa helix, just by displacing each solution from the next one by 1aa.
Furthermore, not all "fished" solutions are correct. For such small fragments, FOMs characterizing their location are unreliable, therefore, given the resolution and the size of ARCIMBOLDO model fragments, the correct solutions cannot be easily discriminated from among the majority of wrong locations. The method moves forward with a coarse selection of possible solutions and prepares several input packages to be spread over a HTCondor-grid and evaluated by Phaser MR sub-processes (rotation, translation, packing, re-scoring and refinement and phasing). After every sub-process cycle, the solutions are limited, if necessary, to keep their number within tractable limits. There are a couple of variables to set up Phaser to accept solutions after a rotation search: `$final_rot` and `$save_rot`. Both were declared in PRD2 case following Phaser defaults to select and save only those solutions over a threshold corresponding to 75% of the highest peak. Usually this value should not be changed unless the packing stage discards too many solutions:

```
$final_rot = q(FINAL ROT SELECT PERCENT 75.0);
$save_rot  = q(SAVE ROT SELECT PERCENT 75.0);
```

The translation search requires the analogous parameters `$final_tra` and `$save_tra` that were configured as well in PRD2 case to select and save solutions over 75% of the maximum translation peak:

```
$final_tra = q(FINAL TRA SELECT PERCENT 75.0);
$save_tra  = q(SAVE TRA SELECT PERCENT 75.0);
```

Aside from the Phaser peak selection parameters, there are other variables, intrinsic to the ARCIMBOLDO design, to keep the number of solutions under control in

the course the parallel individual processes that build the fragment location procedure. Running many hypotheses in parallel is prone to run out of control, even if ARCIMBOLDO can run 10000 jobs in parallel on most grid systems, 50000 parallel jobs become intractable; besides, computing time should be invested where it may succeed.
Since the intermediate solutions offered by Phaser are sorted according to the LLG and given that while developing the method our tests aimed for a way to ensure variability of hypothesis while keeping those most likely to develop into solutions, cutoff variables were introduced according to the following criteria:

1. **Rotation**: the 1st round of rotation search usually starts from nothing but the native dataset. However, from the 2nd round on (remember that for PRD2 ARCIMBOLDO needs to locate the model three times) the rotation search is performed to complete solutions where fragments were already positioned and refined by Phaser refinement and phasing procedure.
   The refined solutions accepted to a new round of rotation search can be limited by the variables `$rot_limit` and `$rot_sec_limit`. If `$rot_limit = -1` all the refined solutions are accepted and `$rot_sec_limit` is ignored. For safety reasons, it is nevertheless advisable to adopt the default value for `$rot_limit` cutoff. Even if figures of merit characterizing rotations cannot be trusted, solutions with high scores are more likely to be correct and should be further pursued. Such solutions will be located in the first output files, as they are sorted according to FOMS. On the other hand, in a multisolution frame, trying over many similar solutions may be less effective than ensuring enough variability of start hypotheses. Therefore, as correctly placed fragments may have diverse FOMS if they belong to different areas of the structure, the choice is to keep all solutions from the first files and the top ones from every remaining file.
   To this aim, ARCIMBOLDO keeps the solutions of the top refined files, until `$rot_limit` number of solutions are counted. At that point, the solution selection changes to only `$rot_sec_limit` top solutions per each output file of refinement and phasing. In PRD2 case all the refined solutions were taken and naturally `$rot_sec_limit` was ignored:

   ```
   $rot_limit        = -1;
   ```

2. **Translation**: solution pruning was performed before preparing the input for the rotation search and given that, except in spacegroup P1, the translation search is complementary to rotation to describe the position of a model fragment -otherwise a fragment with the rotation angles and lacking the X, Y, Z positions is meaningless- every resulting rotation solution is further pursued through a translation search.

3. **Truncating translation solutions**: at this point, recalling that correct translations tend to appear in sets of few solutions, another selection can be performed: small sets will be accepted in their entirety, while large -less likely- sets will be pursued only for their best scored solutions. Before

checking whether oriented and displaced solutions pack correctly in the unit cell, accepted solutions can be limited by `$trans_limit`. Again, a value of -1 indicates that all the solutions are taken to the packing check, but for PRD2 the variable was defined to select only the top 70 solutions of each translation output file. If a file with less solutions than `$trans_limit` exists, the cut off is filled with solutions of the subsequent translation file(s) if any.

```
$trans_limit = 70;
```

In addition to the translation cutoff limit, there is another variable that can combine a user constraint and dynamic values resulting from the experiment: `$sample`. It concerns only the translation search stage of the method as it derives from empirical analyses showing that, frequently, translation files containing less solutions are recipients of the correct solutions.
Whenever `$sample > 0`, a constraint is set to select a population out of the first `$sample` translation output files. The sample is used to calculate the average number of translation solutions and the resulting number is used to utterly reject the translation output files containing a number of solutions exceeding the calculated average. To solve PRD2 structure the population defined to calculate the average number of translations per output file was of 20 files:

```
$sample = 20;
```

4. **Packing**: because packing is a pruning process itself, there is no rationale in further trimming its output. Moreover, it is a very fast process. That is the reason PRD2 was set up to accept all the packing solutions despite the existence of the cutoff variable `$packing_limit`:

```
$packing_limit = -1;
```

5. **Re-scoring**: refinement and phasing: After grouping and sorting the obtained solutions, they are sent to the Phaser refinement procedure, which is very effective on pruning duplicated solutions, but also very time and memory consuming. On that account, solutions are re-scored and, before being passed on for refinement, their number can be limited by `$r_p_limit`. Solutions ordered after the cut off are discarded.
In the PRD2 case solutions surviving the packing filter are grouped in packages of 800, and from the sorted output-file only 70 top solutions will be accepted for refinement:

```
$pack_size = 800;
$r_p       = 70;
```

The same variable, `$pack_size` is used to pack a fixed number of solutions when preparing the input files for packing, re-scoring, and refinement and phasing. In contrast, rotation and translation input files are prepared containing individual

solutions in order to produce a dense population to be analysed and sieved by the following MR subprocesses. The different choices detailed for rotation, translation, packing, re-scoring, and refinement and phasing input files correspond to an amenable number of solutions, neither so low that correct solutions are missed, nor so large that finishing the tasks becomes unmanageable. Especially, for refinement and re-scoring the memory allocated to the process may be exceeded if the number of solutions to be compared is too large, which would lead the whole ARCIMBOLDO process to crash.
The set up through the variable `$pack_size` depends as well on the number of available computing nodes in the grid and their power, the file transfer time between nodes and the average time needed for executing each Phaser task. The default values provided in the program are safe but may be further optimised to adapt them to grids different from ours.

4.1.6.4 Density modification with SHELXE

This is the stage where actual solutions can be recognized. Still, too low a fraction of the correct structure will not lead to a map and trace from which the solution can be recognised. In general, it is not worth to attempt expansion from less than 10% of the total main-chain at a resolution of 2Å. This value is orientative and should be relaxed for higher resolution or large solvent content, therefore, after each fragment placement, a density modification expansion can be set up or skipped. The following variables are related to the SHELXE intervention. Again, the experimental diffraction data must be provided, but this time in SHELX hkl format. An array type variable was chosen to hold such files, because, in some cases, anomalous data are available along with the native diffraction data and they can be used as complementary sources of phasing information. The datasets must be provided through the array `@hkl_file`, where index 0 is reserved for the native data and index 1 for anomalous data if present. Since, PRD2 is a pure *ab initio* case, only index 0 is used:

```
$hkl_file[0] = q(stefan.hkl);
```

Along with the `.hkl` file the procedure requires a SHELX format instruction file for the density modification task. This `.ins` file must be declared through `$ins_file` and for the case of PRD2:

```
$ins_file = q(stefan.ins);
```

The variable `$shelxe_i` accepts 1 or 0 value. It is provided for cases where the fragment to be located is not chiral, for instance in the case of a disulphide bridge or a centrosymmetric cluster it is impossible to distinguish the substructure from its enantiomer from the anomalous difference data and therefore SHELXE must be run twice, testing either possibility. Since in the case of PRD2 the helix fragment establishes the chirality, it should be switched off:

```
$shelxe_i = 0;
```

For the density modification job, the arguments to parameterize the run must be provided through the variable `$arguments` if they should differ from SHELXE defaults. A detailed explanation of the meaning of each argument can be found in SHELXE help, but in the context of ARCIMBOLDO it is advisable to follow different defaults and the relevant parameters are listed in a previous section.
The parameters used for the SHELXE expansion that solved PRD2 are 30 cycles of density modification without sharpening, alternating with three rounds of auto-tracing, twentyfold increased time for the search of α-helices and tripeptides with a solvent content of 45%, deriving phases from the fragments to the resolution limit of 1.9Å and extrapolating missing reflections up to 1.7Å:

```
$arguments = q(-m30 -v0 -y1.9 -a3 -t20 -e1.7 -q -s0.45);
```

As the expansion stage is slow, it constitutes one of the main bottlenecks in the procedure, but it is also the stage where a solution can be attained. Thus, after each fragment location round, an express lane has been established to prioritize a small number hypotheses most likely to succeed. In PRD2, the expansion is undertaken on the 10 solutions with the highest Z-score characterizing their translation function through the combination of `$sort` and `$prio`.
The variable `$sort` holds a FOM that can be "TZS" or "LLG" which is used to order the fragment solutions after the rigid body refinement stage. TZS forces ARCIMBOLDO to sort them by their translation Z-score value and LLG is for cases where the LLG after refinement and phasing is preferred. Phaser outputs refined solutions sorted by LLG, but in general, as several input files are sent through the grid, the intrinsic Phaser sorting comparison will be limited to solutions within the same output-files, so, to customize the sort consistently, ARCIMBOLDO gathers all the refinement and phasing outputfiles -as if it would be the result of a single job- and sorts their solutions according to `$sort`.
To avoid performing density modification to all the available solutions, the most promising ones can be discriminated through `$prio` and expanded, while the rest are kept compressed in the `$expand.tar.gz` file, to be expanded at the end of the ARCIMBOLDO run if the selected ones fail. To expand all solutions, a value of -1 can be assigned to `$prio`. In PRD2 case the decision was to sort solutions according to their translation Z-score value and to expand only the 10 top refined solutions:

```
$sort = 'TZS';
$prio = 10;
```

For the general case the selection of the FOM can vary depending on the results, the latest version of Phaser, currently, advises to rely mainly on the LLG values, rather than on the translation Z-score.

4.1.6.5 Lauching ARCIMBOLDO and checking results

After setting up all the parameters for the job, ARCIMBOLDO can be launched from the downloaded folder by typing in the command line:

```
(perl ./arcimboldo.pl > log) >& log.err &
```

This will use the Perl version installed in your system to run ARCIMBOLDO in background, and redirect the standard output to `log` while the errors, if any, would be redirected to `log.err`. The ARCIMBOLDO job will create several files and folders whose structure will further presented.
It can be said that the structure is solved once a `CC > 25%` characterizing the main-chain trace is reached at the density modification stage, and longer chains than those provided are traced. At present, if no error appears, ARCIMBOLDO runs until all the provided models are located and expanded as many times as ordered. Future plans include making ARCIMBOLDO recursively check the density modification results to stop once the structure is solved, but currently, the CC value must be manually checked through the command line by moving to the project folder where the job was launched and typing:

```
grep CC ens1_frag*/expand/*pdb | sort -k7 -n
```

This looks through all the pdb files created after applying density modification and autotracing, finds the lines where the CCs are expressed, and sorts them by their 7th column value, which corresponds to the CC value.

```
sirius:/home/grid/run4/ab_initio/new_phaser/res_2.0> grep CC ens1_frag*/expand/*pdb | sort -k7 -n
ens1_frag1/expand/test3.pdb:TITLE  test3.pdb   Cycle   3   CC =  8.61%    54  residues in   5  chains
ens1_frag1/expand/test8.pdb:TITLE  test8.pdb   Cycle   3   CC =  8.78%    60  residues in   6  chains
ens1_frag1/expand/test7.pdb:TITLE  test7.pdb   Cycle   3   CC =  9.64%    86  residues in   8  chains
ens1_frag1/expand/test2.pdb:TITLE  test2.pdb   Cycle   1   CC =  9.79%    58  residues in   6  chains
ens1_frag2/expand/test6.pdb:TITLE  test6.pdb   Cycle   1   CC = 10.84%    68  residues in   6  chains
ens1_frag1/expand/test4.pdb:TITLE  test4.pdb   Cycle   1   CC = 11.29%    60  residues in   7  chains
ens1_frag1/expand/test6.pdb:TITLE  test6.pdb   Cycle   1   CC = 11.49%    42  residues in   4  chains
ens1_frag2/expand/test9.pdb:TITLE  test9.pdb   Cycle   1   CC = 11.78%    54  residues in   4  chains
ens1_frag1/expand/test9.pdb:TITLE  test9.pdb   Cycle   1   CC = 11.79%    59  residues in   6  chains
ens1_frag1/expand/test0.pdb:TITLE  test0.pdb   Cycle   3   CC = 11.98%    75  residues in   8  chains
ens1_frag1/expand/test1.pdb:TITLE  test1.pdb   Cycle   2   CC = 12.53%    80  residues in   7  chains
ens1_frag1/expand/test5.pdb:TITLE  test5.pdb   Cycle   3   CC = 12.57%    95  residues in  11  chains
ens1_frag2/expand/test0.pdb:TITLE  test0.pdb   Cycle   2   CC = 12.76%    80  residues in   6  chains
ens1_frag2/expand/test5.pdb:TITLE  test5.pdb   Cycle   3   CC = 14.23%    67  residues in   6  chains
ens1_frag2/expand/test2.pdb:TITLE  test2.pdb   Cycle   2   CC = 14.74%    80  residues in   7  chains
ens1_frag2/expand/test7.pdb:TITLE  test7.pdb   Cycle   3   CC = 15.48%   103  residues in  10  chains
ens1_frag2/expand/test1.pdb:TITLE  test1.pdb   Cycle   3   CC = 15.73%    96  residues in   8  chains
ens1_frag2/expand/test4.pdb:TITLE  test4.pdb   Cycle   2   CC = 15.85%    97  residues in  10  chains
ens1_frag2/expand/test3.pdb:TITLE  test3.pdb   Cycle   1   CC = 16.40%    87  residues in   9  chains
ens1_frag2/expand/test8.pdb:TITLE  test8.pdb   Cycle   3   CC = 27.72%   117  residues in   9  chains
ens1_frag3/expand/test4.pdb:TITLE  test4.pdb   Cycle   3   CC = 36.76%   145  residues in   9  chains
ens1_frag3/expand/test1.pdb:TITLE  test1.pdb   Cycle   3   CC = 37.14%   146  residues in  10  chains
ens1_frag3/expand/test8.pdb:TITLE  test8.pdb   Cycle   3   CC = 37.16%   143  residues in   9  chains
ens1_frag3/expand/test9.pdb:TITLE  test9.pdb   Cycle   3   CC = 38.30%   145  residues in  10  chains
ens1_frag3/expand/test7.pdb:TITLE  test7.pdb   Cycle   3   CC = 38.32%   150  residues in   9  chains
ens1_frag3/expand/test2.pdb:TITLE  test2.pdb   Cycle   2   CC = 38.37%   139  residues in   8  chains
ens1_frag3/expand/test6.pdb:TITLE  test6.pdb   Cycle   3   CC = 38.75%   155  residues in   9  chains
ens1_frag3/expand/test0.pdb:TITLE  test0.pdb   Cycle   3   CC = 38.77%   141  residues in   8  chains
ens1_frag3/expand/test3.pdb:TITLE  test3.pdb   Cycle   3   CC = 40.64%   151  residues in   7  chains
ens1_frag3/expand/test5.pdb:TITLE  test5.pdb   Cycle   3   CC = 40.76%   149  residues in   9  chains
```

Figure 4.27: CC Analysis of traced pdb files.

With the parameterization discussed, after expansion of the 1st fragment, corresponding to 4% of the total main-chain, there is no indication of the structure being solved, but after density modification at the 2nd stage, one of the solutions shows a promising CC of 27.72% and all expanded solutions of the models composed of three fragments show CC higher 30%.
This crystal structure was originally solved with an older version of Phaser, capable of solving the structure with resolution limits of 2.5Å for the rotation and translation searches. Understandably, identical results are irreproducible with the current Phaser version and currently the cutoff can be set to 2.0Å in order to solve the same structure starting from an even less complete partial structure: it is solved after placing 2 fragments instead of the original 3, but relying on better resolution for the fragment searches. It should be taken into consideration that one of the difficulties of solving unknown cases is that they will be suboptimally parameterised, whereas parameterisation is easily optimised for a test case.

4.1.6.6 ARCIMBOLDO folder structure

As ARCIMBOLDO is a strategic combination of MR and density modification procedures, separated folders are created for the different models and stages. Typing `ls -lart` in the command line at the location of the project folder where an ARCIMBOLDO job for PRD2 was run must show the following files and folders:

```
arcimboldo.pl
condor_queue
ens1_frag1
ens1_frag2
ens1_frag3
log
log.err
packing.mtz
Sol.tra
stefan.hkl
stefan.ins
stefan.mtz
th14.pdb
```

Figure 4.28: List of files and folders of PRD2 *ab initio* solution.

Shown in green appear the downloaded files that were used for PRD2 example and that have been previously discussed. In blue are the folders that contain the MR data, and the black ones are files created after launching the job:
condor_queue is an auxiliary file created by ARCIMBOLDO to control that the jobs eventually sent to the HTCondor grid have been completed.
log and **log.err** were asked to be created through the command line launching ARCIMBOLDO.
packing.mtz is a lighter mtz file created by ARCIMBOLDO, out of the original mtz data file, but lacking information about the reflections because it is unnecessary for the packing procedure and it helps to avoid overloading the network when working in environments where files and folders are not shared through a Network File System (NFS).
Sol.tra is a file created to hold statistics on the results of the translation searches, it is updated after every translation search and each update can be interpreted in 2 parts.
If the average option was selected, the 1st part of `Sol.tra` reports the average number of translation solutions derived after each rotation file. These figures are going to be used to prune the translation files. Even if no pruning based on the average is going to be performed, the file is created anyway, because it can be used to study the results and, perhaps, improve the strategy in subsequent runs if a

solution is not achieved from the initial parameterization.
The Fig. 4.29 shows the first lines of the table in `Sol.tra`, highlighting areas of interest. Target folder points to the path where the translation solutions were analysed. In the 2nd line appears the population chosen to calculate the average number of solutions (provided by the user through `$sample`). The body of this part of the table contains, to the left, the translation files enclosed in the restricted sample, and to the right, the number of solutions each of those files contains. At the bottom of the table the calculated average for the given sample is shown.

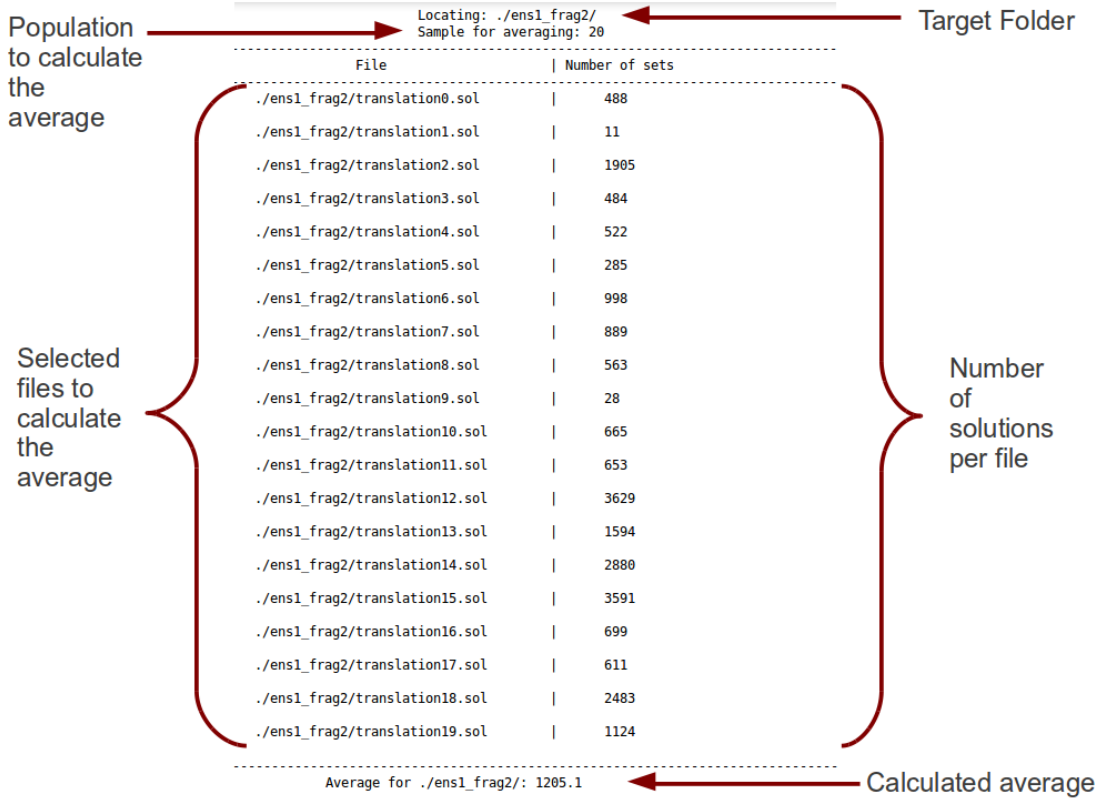

Figure 4.29: Averaging parameters for translation files pruning.

Whenever `$sample = -1`, meaning that averaging is not selected, this part of the table shows the same information for each of the translation solution files obtained.
A remarkable fact that can be observed in this table is that for some files the number of solutions is significantly lower in comparison with others, for example, `translation1.sol`. This can be a promising sign on the way to structure solution because files with less partial solutions tend to have better FOM and to be related to the final structure solution. Because of the solution packaging procedure, promising files also tend to be located towards the top of the produced output, allowing to eliminate unpromising, highly populated files by averaging establishing an average value as cutoff by sampling the first files.
The 2nd part of the table is partially shown in the following figure and it can be exemplified through the highlighted cases:

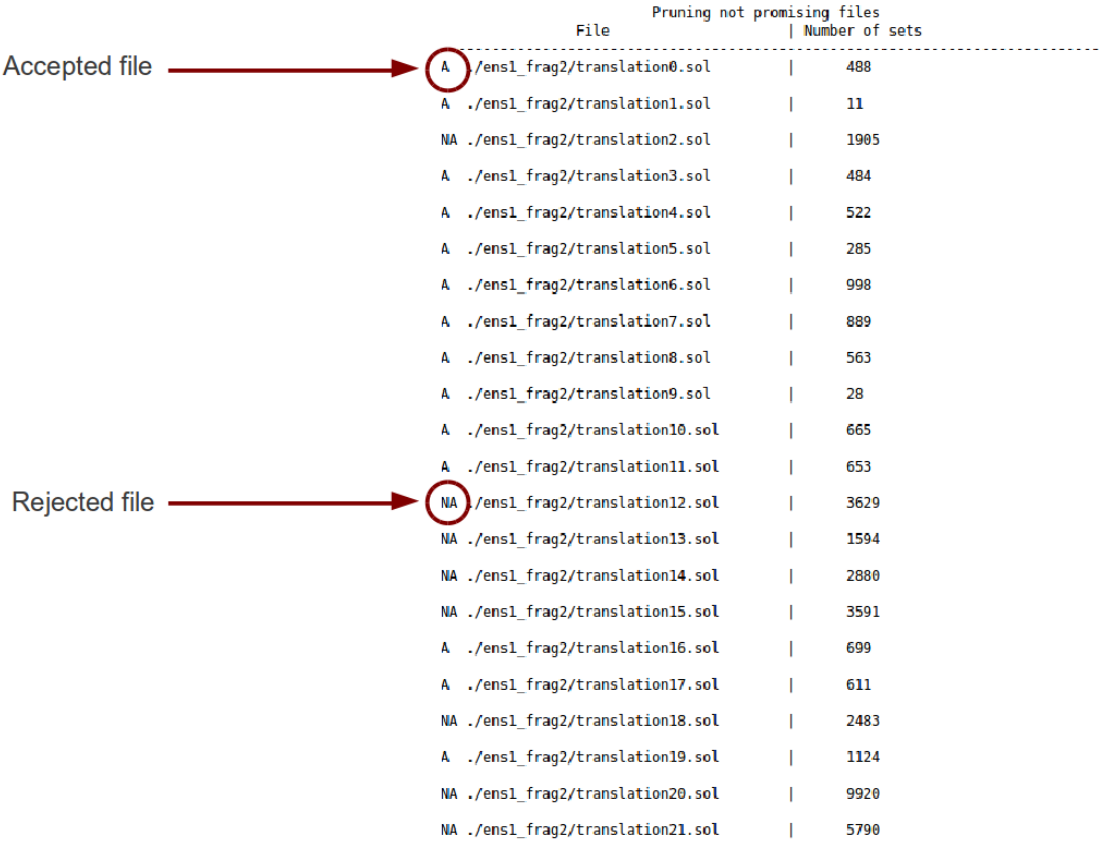

Figure 4.30: Averaging pruning for translation files.

The information presented is the same as in the 1st part, but an extra column is added this time. Left of the table, each line starts with a label that can be either "A" or "NA" (after "Accepted" or "Not Accepted"), which corresponds to the comparison of the number of solutions per file with the calculated average. If the number of solutions is lower than the calculated average, the file is kept, otherwise the line is flagged as "NA" and the file is skipped and deleted.

4.1.6.7 MR folder structure and its input/output files

Phaser naming convention is adopted for the ARCIMBOLDO file and folder structure, thus, a "model fragment" within ARCIMBOLDO is equivalent to an "ensemble" in Phaser. Aside from the initially provided ones, the input and output files corresponding to every Phaser task can be found in folders named after the number of the model fragment (`ensemble #`) plus the number of times the model was located by Phaser (`fragment #`). Consequently, in the PRD2 run:

Folder ens1_frag1 corresponds to the first round (fragment 1) to locate the 14aa helix (ensemble 1).

Folder ens1_frag2 attempts to enhance the partial structure by combining the selected Phaser results in ens1_frag1 folder, with a 2nd instance (fragment 2) to locate the same model (ensemble 1).

Folder ens1_frag3 attempts to solve the structure by combining the Phaser results in ens1_frag2 folder, with a 3rd instance (fragment 2) of the same model (ensemble 1).

The folders mentioned above contain various types of files:
Input files

- `.sh` files: are written by ARCIMBOLDO. They contain the given keywords to execute the Phaser subprocesses. If available, fixed and/or partial ensemble location(s) appear.

Output files

- `.rlist` files: Contain partial solutions calculated by Phaser resulting from a given rotation function. Rotation solutions are expressed in the form of 3 values corresponding to Euler angles.
- `.sol` files: Contain solutions returned by Phaser after every subprocess except a rotation. Solutions contained in .sol files are 6 dimension solutions composed by 3 Euler angles plus 3 fractional coordinates.
- `.out` files: Contain the regular Phaser log of the run.

The root names of those input/output files are:

1. `Rotation` => `rotation#.(sh|rlist|out)`
2. `Translation` => `translation#.(sh|rlist|out)`
3. `Packing` => `packing#.(sh|sol|out)`
4. `Re-scoring` => `r_p#.(sh|sol|out)`
5. `Refinement and phasing` => `r_p_r#.(sh|sol|out)`

were `#` corresponds to cardinal indexes and establishes a relation among `.sh`, `.sol` and `.out` files with the same root name, within the same folder.

Input files for applying MR to PRD2 through the grid.
Since the files containing the basic data that Phaser needs for performing a MR search were already discussed, the following descriptions correspond to those `.sh` files created to manage Phaser jobs, using its keyword interface, through a HT-Condor grid.
The `.sh` files are composed by the Phaser keywords and the values defining each subprocess, plus fixed ensemble locations if already determined, in which case they are described in 3 lines that, in this text, are highlighted in bold. An example of a fragment location provided within a `.sh` file might be:

```
SOLU SET RFZ=3.0 TFZ=4.3 PAK=0 LLG=11 LLG=13
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 38.981 99.713 259.712 FRAC -1.28005 0.49264 0.28429
    BFAC -6.93452
```

The 1st line declares the corresponding FOMs calculated after each Phaser subprocess (rotation, translation, packing, re-scoring and rigid body refinement) for the fragment solution proposed in the last line:

- RFZ : Z-score value resulting from the chosen rotation function.
- TFZ: Z-score value resulting from the chosen translation function.
- PAK: number of clashes revealed in the packing check.
- LLG: calculated log likelihood gain values at re-scoring or refinement and phasing processes.

The 2nd line shows the spacegroup provided in the mtz file and used for the calculations.
The 3rd line expresses the solution in terms of Euler angles, fractional coordinates and calculated B-factors.
Rotation .sh files: the only difference between the rotation `.sh` files in `ens1_frag1` and the ones in the rest of the folders is that those in `ens1_frag1` do not contain any partial solutions because they represent, precisely, the first search steps:

```
MODE MR_FRF
HKLIN "stefan.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test rotation for Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
    100.0
SEARCH ENSEMBLE ensemble1

SAMPLING ROT 1.5
FINAL ROT SELECT PERCENT 75.0
SAVE ROT SELECT PERCENT 75.0
ROOT "rotation0"
END
EOF-phaser
```

rotation0.sh in ens1_frag1

```
MODE MR_FRF
HKLIN "stefan.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test rotation for Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
    100.0
SEARCH ENSEMBLE ensemble1
SOLU SET RFZ=3.0 TFZ=4.3 PAK=0 LLG=11
    LLG=13
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 38.981
    99.713 259.712 FRAC -1.28005 0.49264
    0.28429 BFAC -6.93452
SAMPLING ROT 1.5
FINAL ROT SELECT PERCENT 75.0
SAVE ROT SELECT PERCENT 75.0
ROOT "rotation0"
END
EOF-phaser
```

rotation0.sh in ens1_frag2

Translation .sh files: these files always contain calculated rotation angles taken

from the rotation output files within the same folder and its corresponding Z-score value (both underlined):

```
MODE MR_FTF
HKLIN "stefan.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test Translation for Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
     100.0
TRANSLATE FULL

SOLU SET



SOLU TRIAL ENSEMBLE ensemble1
     EULER 308.005 82.264 231.445 RFZ 3.96
SOLU TRIAL ENSEMBLE ensemble1
     EULER 51.300 80.603 78.523 RFZ 3.92
SOLU TRIAL ENSEMBLE ensemble1
     EULER 51.320 79.307 76.836 RFZ 3.89
SOLU TRIAL ENSEMBLE ensemble1
     EULER 50.346 78.102 77.657 RFZ 3.87
SOLU TRIAL ENSEMBLE ensemble1
     EULER 50.606 78.227 77.849 RFZ 3.87
SOLU TRIAL ENSEMBLE ensemble1
     EULER 50.250 80.122 77.725 RFZ 3.86
(...)
SAMPLING TRA 0.7
FINAL TRA SELECT PERCENT 75.0
SAVE TRA SELECT PERCENT 75.0
ROOT "translation0"
END
EOF-phaser
```

translation0.sh in ens1_frag1

```
MODE MR_FTF
HKLIN "stefan.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test Translation for Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
     100.0
TRANSLATE FULL

SOLU SET RFZ=3.0 TFZ=4.3 PAK=0 LLG=11
     LLG=13
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 38.981
     99.713 259.712 FRAC -1.28005 0.49264
     0.28429 BFAC -6.93452
SOLU TRIAL ENSEMBLE ensemble1
     EULER 308.005 82.264 231.445 RFZ 3.65
SOLU TRIAL ENSEMBLE ensemble1
     EULER 51.320 79.307 76.836 RFZ 3.63
SOLU TRIAL ENSEMBLE ensemble1
     EULER 51.300 80.603 78.523 RFZ 3.61
SOLU TRIAL ENSEMBLE ensemble1
     EULER 308.178 82.989 233.013 RFZ 3.60
SOLU TRIAL ENSEMBLE ensemble1
     EULER 308.718 82.726 233.401 RFZ 3.60
SOLU TRIAL ENSEMBLE ensemble1
     EULER 308.448 82.858 233.208 RFZ 3.60
(...)
SAMPLING TRA 0.7
FINAL TRA SELECT PERCENT 75.0
SAVE TRA SELECT PERCENT 75.0
ROOT "translation0"
END
EOF-phaser
```

translation0.sh in ens1_frag2

Packing .sh files: at the packing stage, 6-dimensional solutions are evaluated. The partial solutions are syntactically equivalent to the fixed stressed in bold, with the difference that FOMs were calculated only for rotation and translation (underlined). All partial solutions are still subject to change when they reach the stage of the refinement and phasing evaluation within the same folder:

```
MODE MR_PAK
HKLIN "packing.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test Packing for Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
PACK SELECT ALLOW
PACK CUTOFF 0
PACK DISTANCE 3.0
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
    100.0
SOLU SET RFZ=3.8 TFZ=3.5

SOLU SPAC P 1 21 1



SOLU 6DIM ENSE ensemble1 EULER 307.355
    79.237 232.888 FRAC 0.32793 0.00000
    0.14330 BFAC 0.00000
SOLU SET RFZ=3.7 TFZ=3.5

SOLU SPAC P 1 21 1



SOLU 6DIM ENSE ensemble1 EULER 306.824
    79.491 232.511 FRAC 0.33316 0.00000
    0.13475 BFAC 0.00000
SOLU SET RFZ=3.8 TFZ=3.5

SOLU SPAC P 1 21 1



SOLU 6DIM ENSE ensemble1 EULER 307.090
    79.365 232.700 FRAC 0.33057 0.00000
    0.13904 BFAC 0.00000
(...)
ROOT "packing0"
END
EOF-phaser
```

packing0.sh in ens1_frag1

```
MODE MR_PAK
HKLIN "packing.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test Packing for Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
PACK SELECT ALLOW
PACK CUTOFF 0
PACK DISTANCE 3.0
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
    100.0
SOLU SET RFZ=3.1 TFZ=4.2 PAK=0 LLG=12
    LLG=14 RFZ=3.1 TFZ=6.6
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 215.220
    158.429 51.922 FRAC -0.82226 0.48897
    0.32647 BFAC -7.22268
SOLU 6DIM ENSE ensemble1 EULER 293.225
    24.825 356.224 FRAC 0.23483 0.34305
    0.27611 BFAC 0.00000
SOLU SET RFZ=3.1 TFZ=4.2 PAK=0 LLG=12
    LLG=14 RFZ=3.1 TFZ=6.7
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 215.220
    158.429 51.922 FRAC -0.82226 0.48897
    0.32647 BFAC -7.22268
SOLU 6DIM ENSE ensemble1 EULER 295.421
    24.894 353.705 FRAC 0.24492 0.34551
    0.28712 BFAC 0.00000
SOLU SET RFZ=3.1 TFZ=4.2 PAK=0 LLG=12
    LLG=14 RFZ=3.1 TFZ=6.7
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 215.220
    158.429 51.922 FRAC -0.82226 0.48897
    0.32647 BFAC -7.22268
SOLU 6DIM ENSE ensemble1 EULER 287.650
    25.278 1.004 FRAC 0.24359 0.35118
    0.26456 BFAC 0.00000
(...)
ROOT "packing0"
END
EOF-phaser
```

packing0.sh in ens1_frag2

Re-scoring .sh files: after packing, the remaining `.sh` files to be analysed here, are very similar in their syntax. What makes the difference is only the additional FOM, which in this case is derived from the packing clash test (underlined):

```
MODE MR_LLG
HKLIN "stefan.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test Refinement and Phasing for Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
     100.0
SOLU SET RFZ=3.8 TFZ=3.5 PAK=0

SOLU SPAC P 1 21 1



SOLU 6DIM ENSE ensemble1 EULER 232.645
     100.763 52.888 FRAC -1.32793 0.50000
     -0.14330 BFAC 0.00000
SOLU SET RFZ=3.7 TFZ=3.5 PAK=0

SOLU SPAC P 1 21 1



SOLU 6DIM ENSE ensemble1 EULER 233.176
     100.509 52.511 FRAC -1.33316 0.50000
     -0.13475 BFAC 0.00000
SOLU SET RFZ=3.8 TFZ=3.5 PAK=0

SOLU SPAC P 1 21 1



SOLU 6DIM ENSE ensemble1 EULER 232.910
     100.635 52.700 FRAC -1.33057 0.50000
     -0.13904 BFAC 0.00000
(...)
ROOT "r_p0"
END
EOF-phaser
```

r_p0.sh in ens1_frag1

```
MODE MR_LLG
HKLIN "stefan.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test Refinement and Phasing for Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
     100.0
SOLU SET RFZ=3.8 TFZ=2.2 PAK=0 LLG=11
     LLG=13 RFZ=3.4 TFZ=5.9 PAK=0
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 230.757
     98.897 53.509 FRAC -1.15285 0.50213
     -0.17859 BFAC -6.90145
SOLU 6DIM ENSE ensemble1 EULER 48.876
     73.015 76.182 FRAC 0.15786 -0.67380
     -0.98755 BFAC 0.00000
SOLU SET RFZ=3.8 TFZ=2.2 PAK=0 LLG=11
     LLG=13 RFZ=3.3 TFZ=5.9 PAK=0
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 230.757
     98.897 53.509 FRAC -1.15285 0.50213
     -0.17859 BFAC -6.90145
SOLU 6DIM ENSE ensemble1 EULER 49.713
     72.763 75.857 FRAC 0.17597 -0.66622
     -0.98000 BFAC 0.00000
SOLU SET RFZ=3.8 TFZ=2.2 PAK=0 LLG=11
     LLG=13 RFZ=3.4 TFZ=5.8 PAK=0
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 230.757
     98.897 53.509 FRAC -1.15285 0.50213
     -0.17859 BFAC -6.90145
SOLU 6DIM ENSE ensemble1 EULER 48.042
     73.272 76.501 FRAC 0.13995 -0.68121
     -0.99536 BFAC 0.00000
(...)
ROOT "r_p0"
END
EOF-phaser
```

r_p0.sh in ens1_frag2

Refinement and phasing .sh files: the partial solutions are described, and sorted as well, by a combination of the previous FOMs plus the LLG calculated during the re-scoring procedure:

```
MODE MR_RNP
HKLIN "stefan.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test 2 Refinement and Phasing for
     Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
     100.0
SOLU SET RFZ=3.8 TFZ=3.5 PAK=0 LLG=13

SOLU SPAC P 1 21 1



SOLU 6DIM ENSE ensemble1 EULER 232.645
     100.763 52.888 FRAC -1.32793 0.50000
     -0.14330 BFAC 0.00000
SOLU SET RFZ=3.7 TFZ=3.5 PAK=0 LLG=13

SOLU SPAC P 1 21 1



SOLU 6DIM ENSE ensemble1 EULER 233.176
     100.509 52.511 FRAC -1.33316 0.50000
     -0.13475 BFAC 0.00000
SOLU SET RFZ=3.8 TFZ=3.5 PAK=0 LLG=13

SOLU SPAC P 1 21 1



SOLU 6DIM ENSE ensemble1 EULER 232.910
     100.635 52.700 FRAC -1.33057 0.50000
     -0.13904 BFAC 0.00000
(...)
ROOT "r_p_0"
END
EOF-phaser
```

r_p_r0.sh in ens1_frag1

```
MODE MR_RNP
HKLIN "stefan.mtz"
HKLOUT OFF
LABIN  F=F SIGF=SIGF
TITLE Test 2 Refinement and Phasing for
     Grid
COMPOSITION PROTEIN MW 12000 NUMBER 2
RESOLUTION 99.0 2.5
XYZOUT OFF
ENSEMBLE ensemble1 PDBFILE th14.pdb IDENT
     100.0
SOLU SET RFZ=3.8 TFZ=2.2 PAK=0 LLG=11
     LLG=13 RFZ=3.4 TFZ=5.9 PAK=0 LLG=35
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 230.757
     98.897 53.509 FRAC -1.15285 0.50213
     -0.17859 BFAC -6.90145
SOLU 6DIM ENSE ensemble1 EULER 48.876
     73.015 76.182 FRAC 0.15786 -0.67380
     -0.98755 BFAC 0.00000
SOLU SET RFZ=3.8 TFZ=2.2 PAK=0 LLG=11
     LLG=13 RFZ=3.3 TFZ=5.9 PAK=0 LLG=35
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 230.757
     98.897 53.509 FRAC -1.15285 0.50213
     -0.17859 BFAC -6.90145
SOLU 6DIM ENSE ensemble1 EULER 49.713
     72.763 75.857 FRAC 0.17597 -0.66622
     -0.98000 BFAC 0.00000
SOLU SET RFZ=3.8 TFZ=2.2 PAK=0 LLG=11
     LLG=13 RFZ=3.4 TFZ=5.8 PAK=0 LLG=34
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 230.757
     98.897 53.509 FRAC -1.15285 0.50213
     -0.17859 BFAC -6.90145
SOLU 6DIM ENSE ensemble1 EULER 48.042
     73.272 76.501 FRAC 0.13995 -0.68121
     -0.99536 BFAC 0.00000
(...)
ROOT "r_p_r0"
END
EOF-phaser
```

r_p_r0.sh in ens1_frag2

In the root directory, it is possible to find transient verb+.cmd+ files. They are created/deleted by ARCIMBOLDO and contain the HTCondor commands to submit the Phaser jobs, for example, addressing a rotation order to the grid through `rotation.cmd` might look like:

```
executable              = given_path_to_phaser
input                   = rotation$(Process).sh
universe                = vanilla
nice_user               = given_condor_mode
notification            = error
initialdir              = ens#_frag#
output                  = rotation$(Process).out
transfer_input_files    = stefan.mtz,th14.pdb
should_transfer_files   = IF_NEEDED
when_to_transfer_output = ON_EXIT
requirements            = given_condor_requirements
rank                    = kflops
queue #
```

The values for the keywords `executable, nice_user, transfer_input_files` and `requirements` have been explained previously in the tutorial.
The lines for `input` and `output` share their root names, as well as the variable $(Process), which is used by HTCondor to assign process numbers to the individual jobs. These numbers grow from 0 to N-1, where N is the number of input files that were created following the defined constraints for pruning and packaging the

solutions, and coincides as well with the # after `queue`.
The keywords `universe`, `notification`, `should_transfer_files`, `when_to_transfer_output` and `rank` are related to the HTCondor environment and are set up according to the convention suggested in the HTCondor manual, but they can be configured differently depending on the environment and the experience of the user.

universe: defines an execution environment for the HTCondor jobs. Vanilla universe is useful to run Shell scripts, transfer files, as well as images of programs to be run.

notification: owners of HTCondor jobs are notified by e-mail when certain events occur, in the above case notifications are sent only on abnormal exit status of the job.

should_transfer_files and **when_to_transfer_output**: their combination enable the file transfer mechanism, should_transfer_files explicitly enables or disables the file transfer mechanism and when_to_transfer_output tells HTCondor when output files are to be transferred back. It depends on whether the machines in the grid share or not an NFS system.

rank: defines a criterion to sort the available machines in the grid to assign the top one(s) to the submitted job(s) according to their `requirements`. It is useful in an heterogeneous environment such as our local development pool but meaningless if all cores in a pool are equivalent, as in Calendula.

The value of `initialdir` tells HTCondor where to find input files and where to transfer back the output files. Its value is internally determined by ARCIM-BOLDO depending on the step of the algorithm when the `.cmd` file is created.

Output files while applying MR to PRD2 through the grid.
When a number of Phaser jobs is submitted to the grid through a `.cmd` file and its related `.sh` files, the same amount of output files should be returned, except for packing submissions where the packing test have determined that no solution passes the check. The number of jobs launched for rotation will be equalled by the subsequent translations, but in all the other MR stages output-solutions are shifted and redistributed fulfilling the given cutoff and packaging constraints to preserve some variability and create different input batches for the next round of `.sh` files. Once the solutions pass the refinement and phasing check, they become the fixed ones that were highlighted in bold in the previous `.sh` file explanation.

A fragment of the rotation output file (`.rlist`) that produced the above described translation .sh file for PRD2 case:

```
SOLU SET
SOLU TRIAL ENSEMBLE ensemble1 EULER
     308.005  82.264 231.445 RFZ 3.96
SOLU TRIAL ENSEMBLE ensemble1 EULER 51.300
      80.603  78.523 RFZ 3.92
SOLU TRIAL ENSEMBLE ensemble1 EULER 51.320
      79.307  76.836 RFZ 3.89
SOLU TRIAL ENSEMBLE ensemble1 EULER 50.346
      78.102  77.657 RFZ 3.87
SOLU TRIAL ENSEMBLE ensemble1 EULER 50.606
      78.227  77.849 RFZ 3.87
SOLU TRIAL ENSEMBLE ensemble1 EULER 50.250
      80.122  77.725 RFZ 3.86
(...)
```

rotation0.rlist in ens1_frag1

```
SOLU SET RFZ=3.0 TFZ=4.3 PAK=0 LLG=11 LLG
     =13
SOLU SPAC P 1 21 1
SOLU 6DIM ENSE ensemble1 EULER 38.981
     99.713 259.712 FRAC -1.28005 0.49264
     0.28429 BFAC -6.93452
SOLU TRIAL ENSEMBLE ensemble1 EULER
     308.005  82.264 231.445 RFZ 3.65
SOLU TRIAL ENSEMBLE ensemble1 EULER 51.320
      79.307  76.836 RFZ 3.63
SOLU TRIAL ENSEMBLE ensemble1 EULER 51.300
      80.603  78.523 RFZ 3.61
SOLU TRIAL ENSEMBLE ensemble1 EULER
     308.178  82.989 233.013 RFZ 3.60
SOLU TRIAL ENSEMBLE ensemble1 EULER
     308.718  82.726 233.401 RFZ 3.60
SOLU TRIAL ENSEMBLE ensemble1 EULER
     308.448  82.858 233.208 RFZ 3.60
(...)
```

rotation0.rlist in ens1_frag2

The content of translation, packing, re-scoring and refinement and phasing output files (`.sol`) is the highlighted portion of packing, re-scoring, refinement and phasing, and rotation input files previously discussed in pages 79 - 82.

4.1.6.8 Density modification folder structure and input/output files

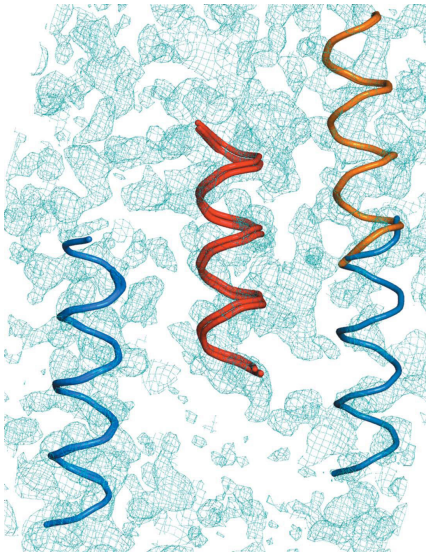

Figure 4.31: 3 sets of 3 helices that lead so successful phasing of PRD2.

The expansions are held in the `expand` folder within the fragment location folders (`ens*_frag*`). A PRD2 solution was found after the 3rd round of placed and expanded fragments, as can be established from the value of the Correlation Coefficient between the partial structure against the native data, which is contained in the `.lst` and `.pdb` output files produced by SHELXE. A value over 25% for the CC indicates that the structure was clearly solved.

A "post-mortem" analysis on the PRD2 case revealed that for the 1st and 2nd fragment, no solution showed a MPE against the final structure phases better than 87°. For the third fragment, one of the top 10 solutions showed an MPE of 57°.

Once a solution is found, there is no point in expanding the rest of the hypotheses, but, in the PRD2 case, if expansion of all 153 solutions consisting of three fragments is performed, the structure can be phased in two more cases. Fig. 4.31 below displays the 3 sets of fragments. The helices in red and orange are common to all three winning solutions and the blue ones are different in each case, although the two on the left overlap over a large span (initial MPE of the fragment phases of 63°) and the one on the right is rather incorrectly placed (initial MPE of 74°). The final map, shown in cyan (including data extrapolated to 1.7Å) reveals that their positions are otherwise extremely accurate.

4.1.7 Libraries of alternative model-fragments case tutorial

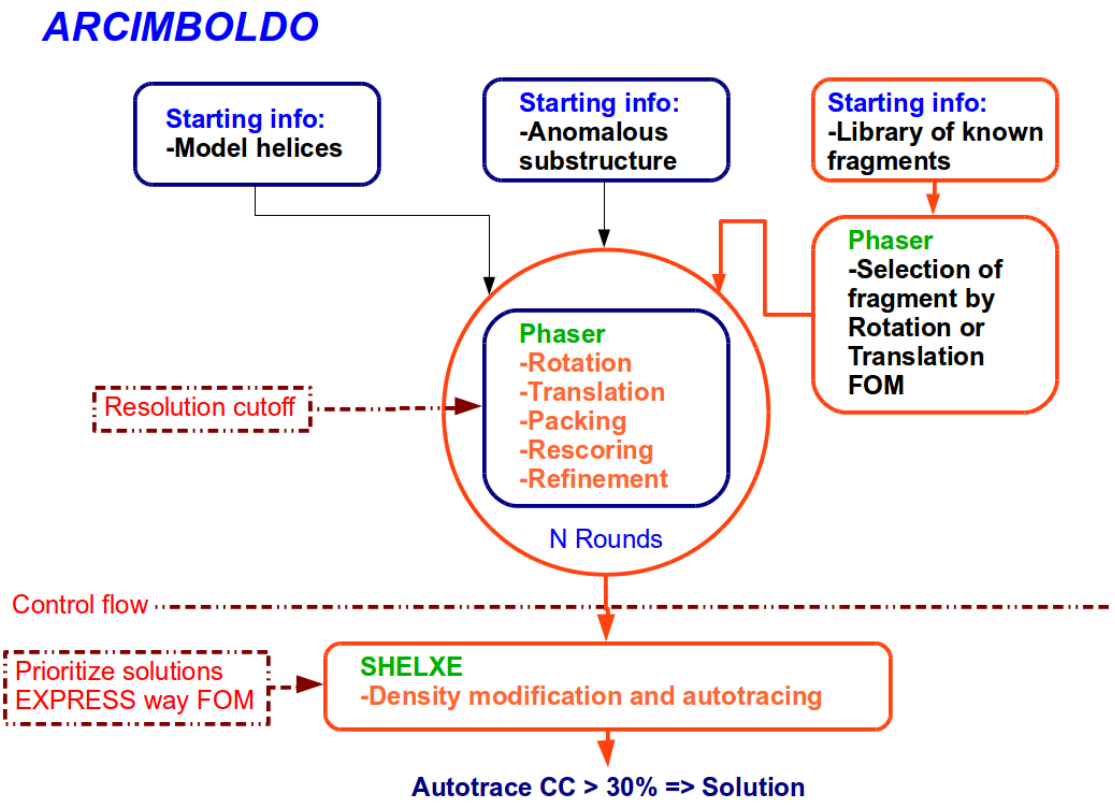

Figure 4.32: Simplified procedure for ARCIMBOLDO alternative model-fragments path.

Moving on from its first use for the *ab initio* case, ARCIMBOLDO has expanded to other scenarios enabling it to tackle larger structures on poorer data. The same frame as that used for *ab initio*, allows the exploitation of other sources of previous stereochemical knowledge, such as low homology model fragments. It is possible to search for different models, either sequentially (by defining several ensembles for the *ab initio* path, *e.g.* looking 1st for a helix of 20aa and afterwards for a helix of 14aa) or by evaluating different hypotheses in parallel.
The path followed in order to test libraries of alternative model fragments will be described through an example and is highlighted in orange. It can be used to exploit any particular stereochemical knowledge available. The same example provided for the *ab initio* case, PRD2 in spacegroup P21 (PDB entry 3gwh) is used here.
The input files for this case are available for download from `http://chango.ibmb.csic.es/ARCIMBOLDO/` under the name of **PRD2: Alternative models test case**. When setting up the procedure, the main differences are to be found in the declaration of the variables for the collection of ensembles to be tested before selecting the most promising one according to a FOM. The tutorial contemplates 4 alternative models but libraries with 10.000 models can be evaluated. Beyond

that number, the condor installation may run into problems to handle all the jobs created. Nevertheless, ARCIMBOLDO supports the successive evaluation of a list of libraries, thus a much larger number of models can be compared.
Given that the current tutorial is based on the same protein as that used to explain the *ab initio* case, the explanation for some of the input-files is omitted here, and pages where they were previously discussed are quoted:

- The native dataset in CCP4 mtz format: `stefan.mtz`
- The model fragments to be located in pdb format. Unlike the *ab initio* case, 4 model fragments are provided as alternative hypotheses:
  1. `14alaninas.pdb`: a file containing the atomic coordinates for the same model polyalanine-helix with 14 residues used in the previous case.
  2. `14ala_side-chains.pdb`: a file containing the atomic coordinates for a helix with side-chains in the most represented conformers from Leu74 to Gln87 modelled with the program SCWRL4[[78]].
  3. `14ala_side-chains_better.pdb`: a file containing the atomic coordinates for the same helix with the side-chains in the standard conformers that are closest to the final structure.
  4. `real_part.pdb`: a file containing the atomic coordinates for the real helix cut out of monomer A in the final structure but with artificial B-factors.
- A plain text-file containing a list with the above mentioned model fragments: `likely_ens.ls`. Under Linux, this file can be created from the command line in the folder containing the files with the coordinates of the model-fragments by typing:

```
ls -1 *pdb > likely_ens.ls
```

- The native dataset in SHELX hkl format: `stefan.hkl`
- The instruction file for the density modification procedure through SHELXE: `stefan.ins`

4.1.7.1 Data analysis

Since the diffraction data provided to describe the alternative model fragments path are the same used in the *ab initio* path tutorial, please, refer to the previous tutorial for an explanation on how to analyse if the available data are suitable for ARCIMBOLDO.

4.1.7.2 Definition of variables for ARCIMBOLDO/Phaser

Most of the input data for this tutorial is common to that provided for the *ab initio* path explanation. Formally, the current case can be seen as an extension of the previously described one. For variables remaining unaffected, references are offered.
Once the specific setup of the environment (see Setup for general variables above) is defined, the variables defining the data and target structure `$mtz_file, $labin` and `$composition` have to be defined as previously described.
The resolution range must be set up for both, rotation and translation searches, in this tutorial it will be truncated to 2.1Å for both, rotation and translation, which can be seen at the end of the declaration lines. The 99.0 indicates that no low resolution cutoff was set up.

```
$resolution_rot = q(RESOLUTION 99.0 2.1);
$resolution     = q(RESOLUTION 99.0 2.1);
```

Next steps involve the definition of the starting hypotheses and how to set up their search. Comparing the results of this case with those described in the *ab initio* tutorial, the main difference is that PRD2 structure is solved twice after correctly placing 2 times the selected helix of the alternatives provided, rather than requiring the fine placement of 3 main-chain helices by the *ab initio* procedure. The cause is that the fragments, from which the expansion procedure starts, contain more information because side-chains are included and there are 2 copies of the molecule in the AU. Nevertheless, to illustrate the user in the definition of a combined search of different model fragments, the search for this tutorial will be set to look twice for the 1st ensemble and once for a 2nd one. To define the conditions for the model searches, it must be kept in mind that the array type variables that describe the models and their use are linked by their indices, *i.e.* In ARCIMBOLDO, content from different arrays, but sharing the same index number, is connected. Each of these arrays starts indexing at 1:

- `@pdb_file`: holds the paths to the files with the atomic coordinates of the model fragments, indexed by the order in which they should be located. For this tutorial 2 model fragments must be located, however, at the beginning of the procedure the 1st ensemble is still undefined, instead, a list of likely ensembles is available. In order to make ARCIMBOLDO run a test to select the most promising ensemble within a collection, the definition corresponding to index 1 of `@pdb_file` must be empty:

  ```
  $pdb_file[1] = q();
  ```

  The method (following an algorithm that is further discussed) will select the ensemble with a highest FOM over the likely hypotheses listed in the provided input file `likely_ens.ls` and will use it to fill the value of this (for the moment) empty variable. To define a combined search of different model fragments, a 2nd ensemble to be located is presented. The 2nd hypothesis is

predetermined, thus it is explicitly given in `$pdb_file[2]` as a model helix of 14 aminoacids:

```
$pdb_file[2] = q(14alaninas.pdb);
```

In general, due to the combinatorial explosion derived from combining all given hypothesis with a variety of new hypotheses, the strategy of selecting one ensemble out of a library of possibilities should be limited to the 1st ensemble (`$pdb_file[1]`) in the list of models to locate (`@pdb_file`). Starting from several possible locations of an ensemble to look for the possible position of each model within a library, would derive in an intractable number of grid jobs.

- `@an_pdb` :sets whether or not the models in `@pdb_file` must be treated as anomalous scatterers by the density modification procedure with SHELXE, the accepted values are 1 or 0 meaning true or false. The models provided for solving PRD2 are not anomalous fragments, thus, to link this information with the reserved indexes in `@pdb_file`:

```
$an_pdb[1] = 0;
$an_pdb[2] = 0;
```

- `@rotf`: defines the Phaser rotation function to be used to find the correct orientation of the model fragment(s). In general, the fast rotation function will be used to both search ensembles, therefore:

```
$rotf[1] = q(frf);
$rotf[2] = q(frf);
```

- `@traf`: defines the Phaser translation function to be used when trying to find the correct position of the model(s) once rotated. PRD2 was solved applying the fast translation function to both search ensembles, consequently:

```
$traf[1] = q(ftf);
$traf[2] = q(ftf);
```

- `@ident`: defines the percent sequence identity of the model to the real sequence. For any ARCIMBOLDO case this value will be 100%, given that the model fragments used by the method are required to be very small and accurate. Particularly, the declaration of the percent sequence identity of both expected helices in PRD2 is:

```
$ident[1] = q(100.0);
$ident[2] = q(100.0);
```

- `@search`: defines the number of times the models have to be located. It is expressed as a range and composed by the 2 limiting numbers of the

search range within a parenthesis. To solve PRD2 structure following this tutorial, placing twice the 1st ensemble would be enough, but, to explain a combined-search case at the same time, a 2nd ensemble was provided. To set up the combined-search environment ARCIMBOLDO is customized to locate 2 times the 1st ensemble and once the 2nd one:

```
$search[1] = q(1,2);
$search[2] = q(1,1);
```

The folder structure will be derived from the combination of the ensemble indexes and the search ranges as shown below.

Particular variables for the alternative models case: the 'likely' variables.

In a case where alternative hypotheses are available, the key variable for defining the model-fragments is not `@pdb_file` any more, but `@likely_ens`. While `@pdb_file` must be used to define the list of models expected to be sequentially located, `@likely_ens` must be used to define the list of alternative models to be evaluated. The set of likely models can be proposed in 2 different ways:

1. Manually, by defining each index of `$likely_ens[1]` and starting from index 1 as follows:

```
$likely_ens[1][1] = q(14alaninas.pdb);
$likely_ens[1][2] = q(real_part.pdb);
$likely_ens[1][3] = q(14ala_side-chains.pdb);
$likely_ens[1][4] = q(14ala_side-chains_better.pdb);
```

2. Through a text file containing the names of all potential pdb models, which in turn can be created using the command ls -1 (see page 87). In this case the name of the file must be assigned to the index 0 as follows:

```
$likely_ens[1][0] = q(likely_ens.ls);
```

Notice that, to add values to the list of alternative models, 2 indexes are used unlike when adding values to `@pdb_file` where only 1 index is used. This is due to the fact that they are different data types: `@pdb_file` is an array while `@likely_ens` is an array of arrays. There is a cross reference between the indexes of `@likely_ens` and `@pdb_file` which means that, provided that `$pdb_file[1] = q();` whatever is declared in index 1 of `@likely_ens` is related to index 1 of `@pdb_file`. In practice, `@likely_ens` can be used to define alternatives for more than 1 ensemble because it has no limiting-index, but, as previously explained while defining `@pdb_file`, seeking for the most promising hypothesis is only functional for ensemble 1, which leaves space for defining exclusively the index 1 of `@likely_ens`, the reason why the 1st index of `@likely_ens` is fixed to 1. When internally the most promising

ensemble contained in the array `$likely_ens[1]` is detected, the empty definition of `$pdb_file[1]` will be filled with the selected model and ARCIMBOLDO will proceed as it would do in a regular *ab initio* case. After defining likely models, ARCIMBOLDO needs to be told whether they are a single collection of similar pdbs (value = 0) or a collection of `.tar.gzip` files (value = 1) grouping models alike:

```
$likely_ens_gz[1] = 0;
```

The files in this ARCIMBOLDO example are not compressed, being just four files. When scoring alternative hypotheses by a given FOM, they should be matchable, for instance, a 30aa helix will get a higher LLG than a 14aa one because of their sizes. It is clear that helices of so different sizes are not comparable, but, for complex cases where clustering the alternatives models by subcategories is feasible, compressed files for each cluster can be provided. Examples of such subcategories might be: helices with different degrees of curving, helices with side-chains in different conformations, fragments cut out from different homologues, etc. Attending to this, the current example could be designed to divide the likely models into 2 groups:

- `Without_side-chains.tar.gz`, containing the helix without side-chains.
- `With_side-chains.tar.gz`, containing the rest of the models.

in which case the set of likely models could be proposed through a text file containing the names of all the available clusters, that, under Linux, can be created from the folder containing the compressed models, using the command:

```
ls -1 *tar.gz > compressed_likely_ens.ls
```

and assigning it to the index 0 of `@likely_ens` as follows:

```
$likely_ens[1][0] = q(compressed_likely_ens.ls);
```

Each cluster will be decompressed to a folder named `ens1_eval`, tests are performed in the mentioned folder and after evaluating their results, the model with the highest FOM will be kept to be compared to the homologous one of the next group. When all the clusters are evaluated, the final ensemble is taken and used as in an *ab initio* routine case. The current example fits in the last stage of the procedure explained, when the last cluster is already decompressed and what remains to be done is to test the alternatives and select the most attractive one. The criteria to select the ensemble is composed by 2 variables: `$sel_best_after` and `$fom`. The 1st one accepts 'rotation' or 'translation' values to indicate which one of those Phaser functions is going to be used to discriminate the final model. These are fast and accurate tasks that, combined with the power of the parallelism offered by a grid, make amenable the calculation and comparison of a large number of jobs. "Post-mortem" analyses of several solved structures has shown that the final solutions are usually favoured by their rotation and translation FOMs at initial stages. The variable `$fom` states which FOM is going to be used for sorting the

solutions before keeping the top one, it can take 'LLG' or 'Z-score' values. In the case of PRD2 the selected ensemble arose from a selection after the fast rotation function where solutions were sorted by LLG:

```
$sel_best_after = q(rotation);
$fom            = 'LLG';
```

Limiting and pruning variables.
To find the correct orientation of the helices in PRD2 case, the rotation step to set up the sampling for the fast rotation function was increased to 2.0°, slightly larger than the previously used for the *ab initio* case:

```
$rot_frf_sampling = q(SAMPLING ROT 2.0);
```

This number establishes a compromise between time and accuracy. Since evaluating a function over a whole library of models increases the computational and time requirements, the choice was to slightly increase its value to speed up the evaluation. The variables `$tra_ftf_sampling`, `$final_rot`, `$save_rot`, `$final_tra` and `$save_tra` retained the same values as those used in *ab initio* tutorial. As empirical values, they are found to perform satisfactorily and are thus left unless packing eliminates too many solutions:

```
$tra_ftf_sampling = q(SAMPLING TRA 0.7);
$final_rot = q(FINAL ROT SELECT PERCENT 75.0);
$save_rot  = q(SAVE ROT SELECT PERCENT 75.0);
$final_tra = q(FINAL TRA SELECT PERCENT 75.0);
$save_tra  = q(SAVE TRA SELECT PERCENT 75.0);
```

Again, as in the *ab initio* case, pruning intermediate solutions is important to prevent flooding the grid. In this case all the previous values were kept except for the cut-off variables before rotation and refinement and phasing:

1. Rotation: A "post-mortem" on the results of the *ab initio* case showed that a careful cut-off before a new round of rotation might improve the speed and efficiency of the procedure. Therefore, for this case, instead of taking every solution, the refined solutions accepted to a new round of rotation search were primarily limited to 50. Once the number is reached only the 5 top solutions of each refinement and phasing `.sol` file passed to the next stage:

   ```
   $rot_limit     = 50;
   $rot_sec_limit = 5;
   ```

   This procedure was designed to avoid collecting all the obtained solutions to perform an additional re-scoring step. In this way a tractable and heterogeneous sample is built from the most promising results, together with the top solutions of the remaining files.

2. Refinement and phasing: In each re-scoring output-file, the number accepted solutions was slightly increased to the top 100 per file, since the number and overall power of the machines in the grid increased while developing the method:

```
$r_p = 100;
```

Depending on the characteristics of the pool this number may be adapted. The variables `$trans_limit` and `$packing_limit` remained as in the *ab initio* tutorial:

```
$trans_limit   = 70;
$packing_limit = -1;
```

4.1.7.3 Density modification with SHELXE

Given that the input data for the density modification scenario is exactly the same to that used in the *ab initio* case and that the SHELXE parametrization is mainly based on the experimental data and the preliminary idea of the structure, most of the definitions for the variables retained their previous values. Details are given for the arguments used at the density modification stage, for the rest of the variables only their values are offered below:

```
$hkl_file[0] = q(stefan.hkl);
$ins_file    = q(stefan.ins);
$shelxe_i    = 0;
```

The arguments used for the SHELXE expansion that solved PRD2 are 30 cycles of density modification without sharpening, alternating with three rounds of autotracing searching for α-helices, spending 10 times as much as the default for random seeding of helices and peptides initial positions, with a solvent content of 40%, deriving phases from the fragments to the resolution limit of 2Å and extrapolating missing reflections up to 1.7Å. The memory range was increased 3 times the default to fit the size of the problem according to the experimental and extrapolated data:

```
$arguments   = q(-m30 -v0 -a3 -q -t10 -s0.4 -y2 -e1.7 -l3);
```

The express lane was designed like in the *ab initio* case, the expansion is attempted on the 10 solutions with the highest Z-score characterizing their translation function:

```
$sort = 'TZS';
$prio = 10;
```

4.1.7.4 ARCIMBOLDO folder structure

As the ARCIMBOLDO procedure is a strategical combination of MR and a density modification, separated folders are created for both strategies.

4.1.7.5 MR folder structure and input/output files

The folder structure and their contents parallel the ones in 4.1.6.7. The main difference is that in this tutorial data can be found for the search of a second ensemble. The input and output files corresponding to the Phaser tasks to locate the main-chain helix of 14 aminoacids will be found in a folder named after the number of the model fragment (ensemble 2) plus the number of times the model was located by Phaser (fragment 1). On that account:

`ens1_frag1`: corresponds to the first round to locate (fragment 1), the most promising helix out of the library provided (ensemble 1).

`ens1_frag2`: contains solutions composed by combining those in the `ens1_frag1` folder with a second position (fragment 2) for the same model (ensemble 1).

`ens2_frag1`: corresponds to the first round to locate (fragment 1), the main-chain helix (ensemble 2) after the location of 2 copies of the most promising helix out of the provided library.

4.1.7.6 Density modification folder structure and input/output files

The expansions are held in the `expand` folder within the fragment location folders (`ens*_frag*`). A PRD2 solution was found after the 2nd round of placed and expanded solutions for the selected ensemble from the given collection of likely hypotheses. Of the 10 two-fragment solutions that expanded through density modification, one clearly led to a recognizable solution with tracing 103aa characterized by CCs of 26.5%. A second solution tracing 79aa with a CC of 16.1% could be developed into a full solution through more cycles of iterated density modification and autotracing. A "post-mortem" analysis indicated that their MPEs compared to the final structure were 58 and 72°, respectively.
As searching for successive fragments is much more time consuming than performing many single fragment rotations, it may be more effective to invest time initially to screen through fragments with side-chains in all possible standard conformer combinations, that will not clash, than to have to place more fragments. Unfortunately, solving fragments may not always be unequivocally identified through such early stage FOM but, in any case, it may be useful to score the models to be tried.

4.2 Macromolecular structures solved with ARCIMBOLDO

ARCIMBOLDO was first released in September 2009[53]. The structure of PRD2 (3GHW) at 1.95Å resolution, whose case has been described in detail in the tutorial section, was the first previously unknown structure to be determined with this method. Since then, the method is available to download free of charge for academic users. The factors limiting its practical use might be the requirement of a pool of computers running Condor middleware, as its use is intrinsic to the program, as well as the necessity to ease its exploitation by non-expert users providing standard parameterization guidelines with robust defaults. The present work aims to provide such a guide for future reference, based on the successfully solved structures. Plans for extension to other middleware platforms will be developed in the future.
During this initial period a score of previously unknown macromolecular structures were solved, covering a wide range of spacegroups, sizes twice as large as the regular protein solved by classic methods, and resolution varying from 1.3 to 2.0Å, along with a few exceptions phased at lower resolution, as shown in the table below. These correspond to particular cases where the method could be further pushed by combination with other sources of information or algorithms that are not yet a standard part of ARCIMBOLDO.

No.	Protein from	spacegroup	Nres	Fragment*	d(Å)
1	P. Czabotar	$P3_121$	120	1H14	1.30
2	M. Graille	$P2_1$	310	1H16	1.45
3	K. V. Hecke	P432	165	2H14	1.60
4	J. Hermoso	$P6_1$	50	1H10	1.70
5	J. M. Pereda	C2	240	C.F. 2H17	1.70
6	K. Zeth	$P2_1$	428	C.F. 2H16	1.70
7	D. H. Cavalcante, K. Gruber	$P2_1$	204	2H14	1.70
8	S. Trakhanov	$P2_12_12_1$	144	1H14	1.75
9	V. Arcus (4E1P)	$P2_1$	112	2H12	1.80
10	K. Zeth	$P3_121$ twin	74	1H20	1.90
11	K. Zeth (4AEQ)	$C222_1$	90	1H12	1.90
12	R. Bunker	P1	200	M. H.	1.95
13	S. Becker (3GHW)	$P2_1$	222	3H14	1.95
14	A. Thorn, G.M.Sheldrick (3SZS)	I422	327	2nmr31	1.95
15	J. Hermoso (2Y8P)	$C222_1$	378	2hom85	2.00
16	X. Gomis-Rüth	$P2_12_12_1$	700	F. + Se-MAD	2.00
17	N. Verdaguer	$P6_322$	50	3H14	2.10
18	O. Mayans	$P2_1$	240	H. with SC	2.10
19	C. Artola, J. Hermoso	$P2_1$	700	F., mod + Buster	2.70
20	N. Valadares, R. Garrat	Pseudo-merohedral	60-240	M.H., twinned	1.60-2.8

*Fragments legend.
#H#: # of helices of # aa.
C.F.: Composite fragment: library of models containing more than one secondary structure fragment.
M.H.: Model helices.
2nmr31: Two fragment of 31aa from an NMR model.
2hom85: Two fragment of 85aa from a low homology model.
H. with SC: Helix with side-chains.
F., mod + Buster: Model fragments refined with Buster.

Table 4.2: Summary of previously unsolved structures phased by ARCIMBOLDO.

The table is sorted according to the resolution of the diffraction data since it is the most limiting barrier for *ab initio* solution of protein structures, and also within

ARCIMBOLDO, availability of high resolution data is a favourable factor, greatly aiding the process.
It should be noted that beyond the characterization of the resolution limit, data quality plays a major role: it pays off to collect the best possible data a given crystal form can yield, in terms of completeness, multiplicity, absence of overloads, ice rings, low background, appropriate cryo-conditions, etc[85,86]. Anisotropic scaling of the data[87] is performed by default both in Phaser and in Shelxe.
All the structures in the table contain at least one helix, and there is clearly a majority of all-helical proteins, but the percentage of secondary structure in a few cases deviates considerably from this trend. For instance, the $C222_1$ structure solved on 1.9Å data is a case where β-strands prevail and there is only one helix. In practice, not only do helices constitute ideal search fragments, but tracing their main-chain is also more robust than for beta strands, hence, it constitutes a circumstance favouring all stages in the procedure. Naturally, the most frequently search model used is the theoretical polyalanine helix of 14aa; occasionally, slightly shorter or larger helices are used, but shorter fragments lead to starting from too litle information and are less effective while for longer helices, deviation from the template leading to a detrimental increase in rmsd to the true structure start to become noticeable and evaluating libraries becomes imperative. In fact solving the first all-β structure has been a challenge requiring a fundamental extension to the method, that falls outside the scope of the present work[88].
As for the spacegroups represented in the table, they reflect in part the more frequent symmetries found in the protein database but the method works with generality. Low symmetry is said to favour higher resolution, and it clearly enhances the rotation function signal, but on the other hand higher symmetry helps data scaling, correction of systematic errors and completeness, leading to better quality experimental data[3,89].
A brief description of the particularities of individual structures solved, the parameterization employed and keys to success follows.
The first case is the structure of a 4-helix bundle, with excellent diffraction data of close to atomic resolution. It is solved with the default parameters discussed, from just a single placed helix of 14 alanines. Publication is still pending.
High resolution of 1.45Å for the all-α protein constituting the second case, is also instrumental in the possibility of phasing it from a bare 6% of the total main-chain. Nevertheless, this solution is borderline, and actually placing a second helix led to a clearer, more complete solution which was developed into the eventually built and refined model used in the structural study. Two of the helices are extremely long (30 and 40 residues, respectively), which favours autotracing to extend the correctly placed fragment into a much larger starting. Publication of the structure is under review.
The 3rd protein in the table proceeds from K. V. Hecke in Leuven and is also not published yet. Even though the dataset suffers from the presence of ice rings, affecting its completeness, the cubic symmetry leading to high multiplicity and the high resolution compensate this flaw. In this case, a model helix of 14aa needed to be correctly placed two times to allow a successful expansion of the fragments to the final solution.

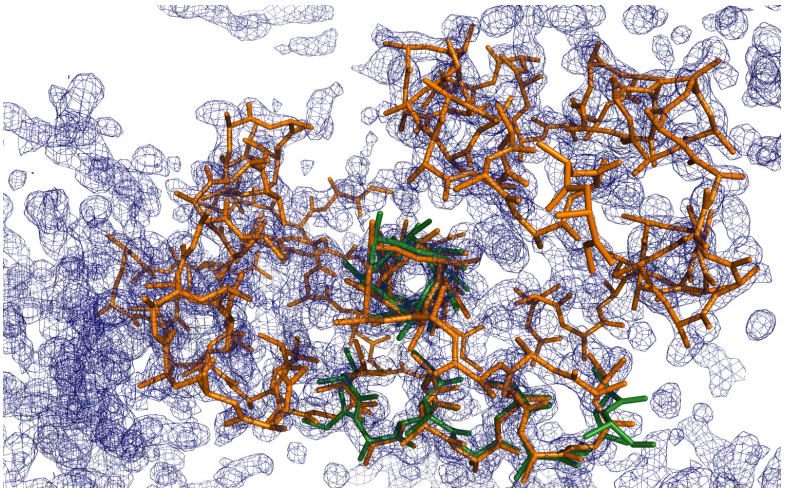

Figure 4.33: Located model helices superposed to the final solution of Hecke's structure.

To find the correct rotation and translation of the model, data was trimmed to 2.5Å and the steps for evaluating rotation and translation functions were set to 2.0° and 0.7Å respectively. Peaks were accepted over the threshold of 75% of the highest peak.
The first MR attempts to locate the model helix produced 15 posible solutions that were not expanded, but directly sent to a new round of rotation and translation searches, to calculate the location of a 2nd copy of the same helix.
After translation coordinates were retrieved, there were available 15 output files containing a number of solutions ranging from 30 - 2310. From these files, only the 70 top solutions were accepted and regrouped in files of maximum 100 solutions to be packing checked.

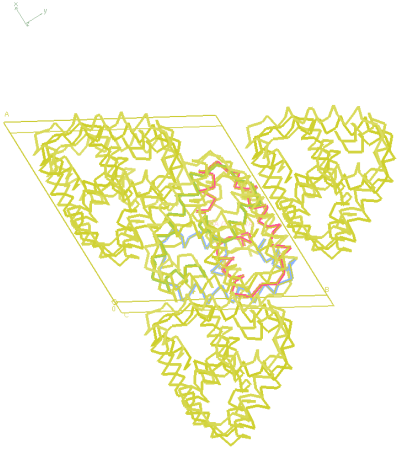

Figure 4.34: Endless polymeric chains in the $P6_1$ crystals of J. Hermoso's group, with a single repeat in the asymmetric unit.

Solutions that survived the packing test were again clustered in files of 100 solutions, later re-sorted by their LLG value. From those, the top 80 per file were once more repackaged in files of 100 solutions and rigid body refined.
The resulting 255 refined solutions were expanded by 100 cycles of density modification without sharpening, alternating with 3 cycles of main-chain autotracing, deriving phases from the fragments with resolution up to 2.0Å and extrapolating missing reflections up to 1.7Å.
From the 255 expanded solutions, 41 showed a CC between 42.34 - 46.92%, including the top 10. The number of residues traced in the winning solutions vary from 125-143 in 2-5 chains.
The 4th structure shown in table was solved within a collaboration with the group of J. Hermoso and it presented a curious crystallographic puzzle: the asymmetric unit is too small to host the protein derived from

the construct expressed, which contained within its full sequence, three highly identical repeats (see Fig. 4.34), thus solution was first attempted in the monoclinic spacegroup $P2_1$, with equal a and c (within the experimental error) and β close to 120, anticipating a pseudo-merohedral twinning but this is not the case. The structure was finally solved in the higher symmetry spacegroup $P6_1$, as the AU contains a single repeat from an apparently infinite polymer, while the non-symmetric part of the sequence has been proteolyzed. One of the advantages of *ab initio* phasing, is that relying on little information is helpful when unexpected and unforeseeable changes have affected the structure to be determined.
The fifth and sixth structures, determined from data provided by Jose Mara de Pereda in C2 and Kornelius Zeth in $P2_1$, both at a resolution of 1.7Å, have been solved with the program BORGES[88], that combines the ARCIMBOLDO algorithm with customized libraries. The development of BORGES in our research group is beyond the scope of this thesis but related inasmuch as it takes the Arcimboldo method one step further to achieve phasing enforcing unspecific tertiary, rather than secondary structure.
The libraries, composed of contiguous main-chain helices in antiparallel and parallel disposition, respectively were necessary, since in these structures the packing check at the stage where a second fragment is added discards all partial solutions. Providing and scoring several different models requires in practice a sophisticated approach, to jointly evaluate all the results, in terms of geometry of the locations as well as for the figures of merit characterizing them. The preliminary work discussed in this thesis in the case of helices with side-chains modeled in different conformers provided proof of principle for the viability of scoring and selecting optimal structural hypotheses but requires its own particular algorithm to create the required libraries as well as to make the huge volume of calculations amenable.
The 18th entry in the table, a structure determined in cooperation with Olga Mayans in Liverpool[90] could be solved with polyalanine helices where part of the sequence had been modeled in various conformers. In this case, the difficulty was derived from the somewhat more modest resolution, of 2.1Å but especially from the highly anisotropic nature of the data. Indeed, the diffraction pattern reminded somewhat of the typical DNA diffraction images, as beyond the anisotropy, the direction of the helices was not to be overseen but distinguishing correct from pseudo-solutions proved extremely hard. Side-chains probably played a role in setting apart the correct solutions and allowing density modification and autotracing to bootstrap and in any case, led to a clear cut solution and an interpretable map. The addition of side-chains information to the starting hypothesis was successfully applied on Mayans structure. Libraries of alternative helices with side-chains (modeled from a predicted main-chain helix or set up by the most frequent conformers) can be input to the method to let it choose the most promising model candidate after a rotation or translation FOM. Even homologues too poor for succeeding with the standard MR procedure can offer valuable information regarding to the folding or particular local information. This last approach was followed in the case of the protein MltE[91] (PDB entry 2Y8P), crystallized in $C222_1$ in the group of Juan Hermoso at the Instituto Rocasolano, CSIC in Madrid. Data were available to a resolution of 2Å. The available, low homology models were unsuccessful as molecular replacement search fragments, but served to propose to

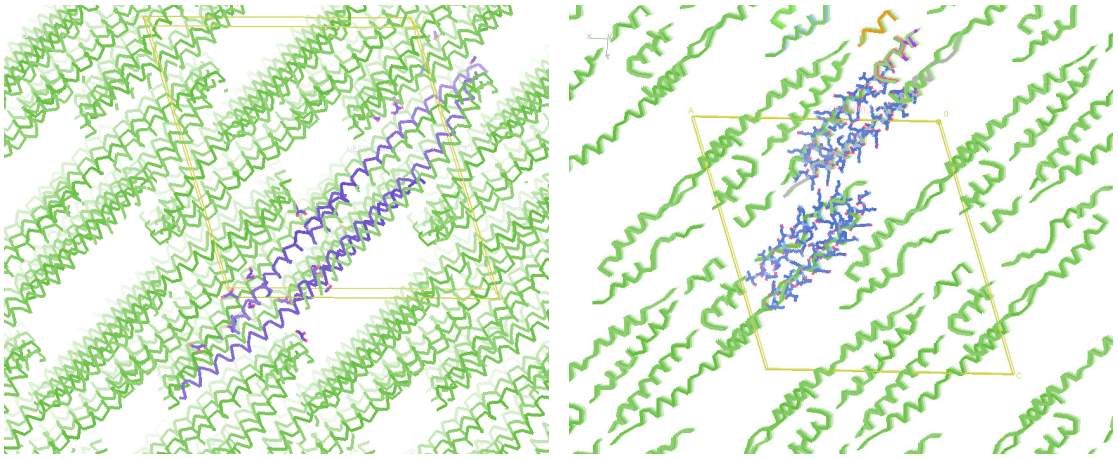

Figure 4.35: Structure of a Coiled Coil at 2.1Å from the group of Mayans. The packing accounts for the high anisotropicity of the data, phasing was achieved from the helix with modelled side-chains shown in blue. Left: ARCIMBOLDO solution; Right: final model.

ARCIMBOLDO alternative small models cut from them, in the aim of exploiting more specific information than mere secondary structure prediction. Extensive ARCIMBOLDO trials with systematically truncated models and further trimming of located fragments to improve the correlation coefficient of model versus data, finaly identified an 85 residues substructure from which the whole structure could be completed through iterative density modification and autotracing of the resulting electron density map.

The structure of a surfactant protein, Lv-ranaspumin from the frog Leptodactylus vastus, was solved *ab anitio* from two fragments of 14 alanine helices. Being a protein isolated directly from from the frog, rather than recombinant, the sequence was unknown, although its all-helical nature was anticipated from circular dichroism experiments and the presence of helices could also be evidenced in the Patterson functions[92] of the three crystal forms obtained. The low resolution crystals could were not appropriate for the ARCIMBOLDO method, but the 1.7Å data rendered a straightforward solution.

Vic Arcus' problem case, Lsr2 from Mycobacterium tuberculosis[93], contains two rather curved helices. As a consequence, the structure could only be solved with models of 14 alanines, shorter than the predicted helix and from the many combinations of resolution cutoffs tried, only one worked. The structure in the monoclinic spacegroup $P2_1$ was solved at 1.8Å. A 14-residue long model polyalanine α-helix was used as the search fragment to arrive at the *ab initio* solution. Location of the first fragment in the structure rendered 26 solutions, characterized by very similar figures of merit. None of the 10 selected for the expressway was effective in expanding to the full structure. Neither did the expansion of the 10 prioritized solutions of two fragments with SHELXE among the 95 partial solutions located with PHASER, lead to a solution. Still, a difference of 4% in the correlation coefficient (CC) of a few solutions, compared to the CC values for the rest led to their identification as promising candidates. Iterations of pdb optimization (eliminating residues from the trace whenever this lead to an increase in the CC for the fragment), and expansion with SHELXE (using 30 cycles of density modification,

no sharpening, extrapolating missing reflections to 1Å and 3 cycles of autotracing) afforded main-chain traces with CC values over 30% in 1 of the 10 selected cases. This initial structure represented a solution from which the remainder of the model could be built.
The $P3_12_12$ structure from the group of Kornelius Zeth at 1.9Å along with the structures form N. Valadares and R. Garrat in Brazil, all presented different cases of twinning, from merohedral to pseudomerohedral. As in the last case, the main problem was to derive a complete, interpretable solution from a large set of partially correct, yet stuck solutions. Publication is underway.
Collicin M Immunity protein, CMI structure[94] (PDB entry 4AEQ) was one of the first unknown macromolecular structures solved by ARCIMBOLDO. The protein crystallized under several conditions but appropriate data from the diffraction experiment was available only from one of the crystals.

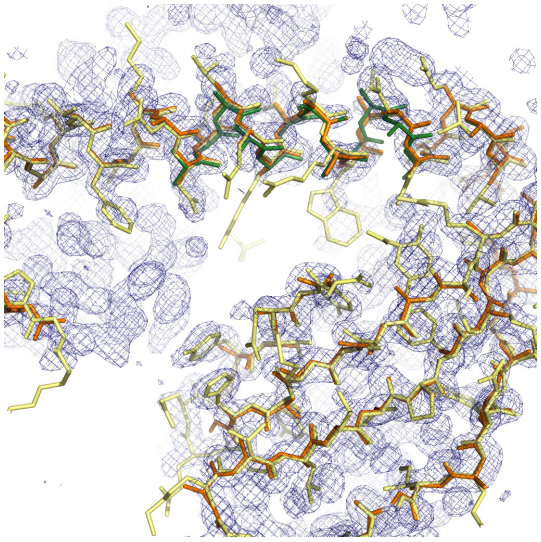

Figure 4.36: Solution of the CMI structure. The search model is shown in green, SHELXE trace in orange, the final model in yellow and the electron density map of the solution, contoured at 1sigma is shown as a blue mesh.

Previous molecular replacement attempts using the standard approaches and the available homologues failed, while ARCIMBOLDO succeeded through its *ab initio* path, using crystal data at 1.95Å resolution.
The monomer contained in the asymmetric unit comprises 90aa and shows a mixed -fold with a 27aa long N-terminal α-helix, followed by a four-stranded strongly bent anti-parallel β-sheet of 14 topology. The extended β-strands 2 and 3 are enclosed by two smaller β-strands 1 and 4 and an extended loop structure L2 is observed between the 1 and 2 strands. The small β-sheet wraps around the N-terminal helix of the symmetry related molecule of the complex.
Dimerization of this fold is mediated by an extended interface of hydrogen bond interactions between the α-helix and the four-stranded β-sheet of the symmetry related molecule. Two intermolecular disulfide bridges covalently connect this dimer to further lock this complex.
Notably, the 3D topology (in particular the long α-helix) is in close agreement with the secondary structure prediction of CMI, which determined the selection of the model fragment (a poly-ala helix of 12aa) for the CMI crystallographic structure solution through ARCIMBOLDO. The method was configured to locate only one copy of the 12aa α-helix using resolution data trimmed to 2.5Å for both, rotation and translation Phaser searches. The sampling steps for these functions were set up to 2.0° and 0.7Å respectively, and the accepted peaks for both were chosen over the threshold of 75% of their respective highest peaks.
Taking into account the number of residues of the chosen model-fragment, the size of the α-helix contained in the final structure, and the fact that both are straight

helices, the 12aa model could have been perfectly placed 16 times in the target helix by relatively displacing the model just 1aa. With the selected parameterization, Phaser proposed 83 locations of the model, that were used to seed as many SHELXE jobs, that enclosed 100 cycles of density modification without sharpening, alternating with 3 cycles of main-chain autotracing, deriving phases from the fragments with resolution up to 2.0Å and extrapolating missing reflections up to 1.7Å.

MR solution No.	CC after expansion	Number of traced aa
62	**47.84**	**83**
36	**47.34**	**80**
23	**47.29**	**88**
10	**43.30**	**80**
1	19.72	35
14	14.39	27
52	13.88	56

Table 4.3: Analysis of CMI Phaser and SHELXE results.

From the 83 MR partial solutions sorted by LLG after refinement, 4 expanded ones showed CC over 43%, with a number of main-chain traced residues ranging from 80 to 88 of the 90 contained in final structure (highlighted in bold in Table 4.3).

At the time CMI was solved, ARCIMBOLDO lacked the "SHELXE express lane", later designed to pursue the top solutions after rigid body refinement, however, the results show that the suggested default to prioritize expanding the top 10 refined solutions would have succeeded anyway, because the 10th solution (highlighted in bold in Table 4.4) was a "winning" one. Still, it is worth remarking that solutions are not necessarily attained from the fragments placed with the highest FOMs.

MR solution No.	CC after expansion
1	19.72
2	6.97
3	7.86
4	12.67
5	4.32
6	7.97
7	7.00
8	5.48
9	8.53
10	**43.30**

Table 4.4: Analysis of CMI top 10 Phaser results.

A promising CC of 19.72% for the expansion of 1st MR partial solution, placed at the 5th position in Table 4.3, and with a gap of 5.33 compared to the 6th top solution of the sorted list (for the comparison, see underlined row in Table 4.3), suggests that a number of additional cycles can make the difference. Doubling the original number of cycles for either density modification or autotracing (-m or -a) results in a solution of the crystal structure. Finally, the expanded solutions allowed to build and refine the few missing residues and side-chains of the deposited and published structure, as can be judge by the figure below.

ARCIMBOLDO's role in the solution of the large MECR2[95] protein from the group of Xavier Gomis at the IBMB-CSIC, was combined with the exploitation of available weak, experimental phases and with previous knowledge of the expected topology of the fold. Case using available anomalous data. Indeed, the selenium substructure could be solved from anomalous differences collected at the peak wavelength, and the fold could be dismembered in secondary structure fragments

that were searched for through brute-force rotation and translation searches in the geometrical range and orientation constrained by the fold hypothesis and the anomalous scatterers as markers.
The virus protein from the group of Nuria Verdaguer, at the IBMB-CSIC was a favourable case for its small size, even if the resolution limit does not quite reach the 2Å required in general. The standard *ab initio* method and defaults could be used to phase the structure.
The above described cases, together with the wide range of symmetries and resolution span listed in the table prove the generality of the method. Further development will be needed to extend its performance to larger structures diffracting to more modest resolution but the cases described comprise a wide variety of scenarios and open the door to further improvement of the original method. Even some protein structures presenting twin pathologies, which is a circumstance that hinders the structure solution, have been solved by ARCIMBOLDO.
The 2.7Å structure from Cecilia Artola in the research group of Juan Hermoso (Rocasolano, CSIC), again constitutes an isolated case, that combines ARCIMBOLDO solutions with modeling of side-chains with SCWRL4[78] and refinement of the partial solutions with the program BUSTER[96]. Then, SHELXE can be started from a file containing the phase information derived from the refinement program, encoded in the form of Hendrickson-Lattman coefficients, instead of generating sigma-A weighted phases from a pdb model supplied as `.pda`.
Also an external user, Richar Bunker, succeeded in using our program to phase his case, the C-terminal domain of the telomere length regulator protein (pdb code 4BJS) and let us know upon its publication in Cell[97]. Feedback from users is particularly appreciated, especially when it reports on a successful use of the program. ARCIMBOLDO has been licensed for more than 380 users all over the world, but many will have been hampered in their use of the program for the need to use supercomputing. The method, then, has resulted in a more ambitious project that requires a more powerful platform for calculations. It is currently developed in collaboration with Caton Alternative systems, that offers a supercomputing environment for the development of the method, and it is financed by the Centre for Technological Development of Industry (CDTI in Spanish) under its Science Industry program. The aim of this collaboration is to expedite, to structural biologists, a simple way to access to supercomputing centres and information technology tools, extending the available crystallographic software to the exploitation of massive parallel-calculations. In this way, the scientific community will dispose of a global and profitable solution of software, middleware and hardware for a specific problem, accessible from anywhere in the world. The idea is to develop a simple graphical interface -focusing on users that lack specific knowledge of supercomputing- that allows, in a short period of time, to remotely use the method to solve macromolecular structures using a RaaS model.

Chapter 5

Summary and Conclusions

- A new method of general applicability for *ab initio* phasing of macromolecular structures using only a medium resolution dataset of native amplitudes and without a detailed structure knowledge, measurements of heavy atoms or anomalous scatterer derivatives has been developed.
- The method has been implemented in a program named ARCIMBOLDO.
- The atomicity dependency of classical *ab initio* phasing methods has been substituted by the enforcement of secondary structure constraints through multisolution location of small models of given geometry combined with density modication and autotracing.
- At the resolution of 2Å or better, targeted by the method, solutions can be unequivocally identified by the number of residues traced and the correlation coefficient of the trace against the experimental data.
- The high demand of CPU time and memory and storage resources required for calculations can be effectively provided and optimized within a grid environment, through parallel tasks and a smart design of the problem guiding the program run through different FOMs.
- Complete data of good quality to a resolution of 2Å are required, although 2.5 - 2.1Å is enough (and sometimes even preferable to higher resolution) for the task of locating model helices, while at the same time speeding up the performance of the method. The density modication and autotracing algorithms are greatly enhanced by the presence of data at higher resolution and they are key to the identification of correct solutions.
- Our method has been most successful exploiting helical fragments, but does not require the majority of the structure to be helical.
- The 1700-atom previously unknown structure of PRD2 at 1.95Å starting from 210 atoms (barely 12% of the total scattering mass) was the first case to be solved *ab initio* with ARCIMBOLDO.

- ARCIMBOLDO was successfully applied to an heterogeneous group of protein structures (test cases as well as previously unknown proteins) with resolution ranging from 1.3 - 2.1Å. The protein sizes were varying from 50 - 428 residues and we can state that the method is independent of the symmetry as structures belonging to nine different spacegroups were solved.

- Two exceptional cases were solved with the help or ARCIMBOLDO at lower resolutions but they corresponded to cases where additional stereochemical or experimental information was available and by no means represent the scope of the method; rather, they provide proof of principle of ways how it can be further developed in the future.

- Some of the cases correspond to pathological data such as non-merohedral and merohedral or pseudo-merohedral twins.

- Extensions to the original method, in order to exploit and combine with model fragment search all starting information available, were introduced into the program ARCIMBOLDO.

- We have implemented the possibility to search for different model fragments of known geometry, either sequentially or by evaluating different fragments, including extensive libraries in which case ARCIMBOLDO is able to select the most promising hypothesis according to FOMs tied to the fragment(s) location procedure.

- Experimental phase information should be exploited whenever possible and three ways to accomplish it have been proposed.

    1. To search for anomalous fragments against MAD or SAD data.
    2. To search for normal fragments once an anomalous substructure is known.
    3. To determine the anomalous substructure once initial phases are known.

- An implementation of ARCIMBOLDO has been taylored to run on the supercomputer Calendula at the FCSCL (`http://www.fcsc.es`), so that the scientic community shall dispose of a highly automated, global solution of software, middleware and hardware for a specific problem. Users with access to an HTCondor grid may download ARCIMBOLDO from our web, `http://chango.ibmb.csic.es/ARCIMBOLDO`.

Appendix A

Scientific production

Publications, scientific and technical documents

1. B. Franke; A. Gasch; **D.D. Rodríguez**; M. Chami; M.M. Khan; R. Rudolf; J. Bibby; A. Hanashima; J. Bogomolovas; E. von Castelmur; D.K. Rigden; I. Usón; S. Labeit; O. Mayans. *Molecular basis for the fold organization and sarcomeric targeting of the muscle atrogin MuRF1.* Open Biology. *under review* 2013. Type of production: Article Format: Journal

2. M. Sammito; C.L. Milln; **D.D. Rodríguez**; I.M de Ilarduya; K. Meindl; I. De Marino; G. Petrillo; R.M. Buey; J.M. de Pereda; K. Zeth; G.M Sheldrick; I. Usón. *Exploiting tertiary structure through local folds for crystallographic phasing.* Nature Methods. *in press* 2013 doi:10.1038/nmeth.2644. Type of production: Article Format: Journal

3. I Usón; C.L. Millan; M. Sammito; K. Meindl; I.M de Ilarduya; I De Marino; **D.D. Rodríguez**. Phasing through location of small fragments and density modification with ARCIMBOLDO. *Present and Future Methods for Biomolecular Crystallography*: The Structural Path to Defence against CBRN Agents. In Advancing methods for Biomolecular crystallography, edited by Randy J. Read, Alexandre G. Urzhumtsev and Vladimir Y. Lunin, 2012. Type of production: Chapters of books Format: Book

4. K. Propper; K. Meindl; **D.D. Rodríguez**; M. Sammito; B. Dittrich; G.M. Sheldrick; E. Pohl; I. Usón. *DNA - Protein Complex Structure Prediction. ARCIMBOLDO structure-solution with DNA - Protein fragment subsets.* Acta Crystallographica Section A. 68, pp. s121. 2012. Type of production: Scientific and technical document or report

5. **D.D. Rodríguez**; M. Sammito; K. Meindl; I.M. de Ilarduya; M. Potratz; G.M. Sheldrick; I. Usón. *Practical structure solution with ARCIMBOLDO.* Acta Crystallographica Section D: Biological Crystallography. 68 - 4, pp. 336 - 343. International Union of Crystallography, 2012. Type of production: Article Format: Journal

6. I. Usón; S.I. Patzer; **D.D. Rodríguez**; V. Braun; K. Zeth. *The crystal structure of the dimeric colicin M immunity protein displays a 3D domain swap.* Journal of Structural Biology. Elsevier, 2012. Type of production: Article Format: Journal

7. **D.D. Rodríguez**; C. Grosse; S. Himmel; C. Gonzalez; I.M. de Ilarduya; S. Becker; G.M. Sheldrick; I. Usón. *Crystallographic ab initio protein structure solution below atomic resolution.* Nature methods. 6 - 9, pp. 651 - 653. Nature Publishing Group, 2009. Type of production: Article Format: Journal

Appendix B

Posters presentations

- Title: *ARCIMBOLDO goes Super: ab initio phasing on the supercomputer calendula fcscl.*
  Name of the conference: International Union of Crystallography XXII Congress and General Assembly
  City: Madrid, Community of Madrid, Spain
  Date: 22/08/2011 - 30/08/2011
  Organising institution: International Union of Crystallography
  I Uson; **D.D. Rodríguez**; M Samito; I.M de Ilarduya; K Meindl.

- Title: *BORGES: a tool to generate customised, secondary structure libraries for phasing.*
  Name of the conference: International Union of Crystallography XXII Congress and General Assembly
  City: Madrid, Community of Madrid, Spain
  Date: 22/08/2011 - 30/08/2011
  M Samito; **D.D. Rodríguez**; I.M de Ilarduya; K Meindl; I Uson.

- Title: *New features in ARCIMBOLDO: a tutorial.*
  Name of the conference: International Union of Crystallography XXII Congress and General Assembly
  City: Madrid, Community of Madrid, Spain
  Date: 22/08/2011 - 30/08/2011
  D.D. Rodríguez; M Samito; I.M de Ilarduya; K Meindl; I Uson.

- Title: *Phasing of an unpredicted palindromic coiled-coil motif.*
  Name of the conference: International Union of Crystallography XXII Congress and General Assembly
  City: Madrid, Community of Madrid, Spain
  Date: 22/08/2011 - 30/08/2011
  B Franke; **D.D. Rodríguez**; I Uson; O Mayans.

- Title: *DNA - Protein Complex Structure Prediction.*
  Name of the conference: Meeting of the American Crystallographic Association

City: New Orleans, United States of America
Date: 28/05/2011 - 02/06/2011
K. Propper; K. Meindl; **D.D. Rodríguez**; M. Sammito; B. Dittrich; G.M. Sheldrick; E. Pohl; I. Uson.

Bibliography

[1] HA Hauptman. The phase problem of x-ray crystallography. *Reports on Progress in Physics*, 54(11):1427, 1991.

[2] FC Bernstein, TF Koetzle, GJB Williams, EF Meyer Jr, MD Brice, JR Rodgers, O Kennard, T Shimanouchi, and M Tasumi. The protein data bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology*, 112(3):535–542, 1977.

[3] HM Berman, J Westbrook, Z Feng, G Gilliland, TN Bhat, H Weissig, IN Shindyalov, and PE Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[4] GA Petsko. Preparation of isomorphous heavy-atom derivatives. *Methods in Enzymology*, 114:147–156, 1985.

[5] E Garman and JW Murray. Heavy-atom derivatization. *Acta Crystallographica Section D*, 59(11):1903–1913, 2003.

[6] MF Perutz. Isomorphous replacement and phase determination in non-centrosymmetric space groups. *Acta Crystallographica*, 9(11):867–873, 1956.

[7] JC Kendrew, G Bodo, HM Dintzis, RG Parrish, H Wyckoff, and DC Phillips. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature*, 181(4610):662–666, 1958.

[8] EF Garman. Radiation damage in macromolecular crystallography: what is it and why should we care? *Acta Crystallographica Section D*, 66(4):339–351, 2010.

[9] JM Bijvoet, AF Peerdeman, and AJ Van Bommel. Determination of the absolute configuration of optically active compounds by means of x-rays. *Nature*, 168(4268):271–272, 1951.

[10] E Dodson. Is it jolly SAD? *Acta Crystallographica Section D*, 59(11):1958–1965, 2003.

[11] Z Dauter, M Dauter, and EJ Dodson. Jolly sad. *Acta Crystallographica Section D*, 58(3):494–506, 2002.

[12] WA Hendrickson and CM Ogata. Phase determination from multiwavelength anomalous diffraction measurements. *Methods in Enzymology*, 276:494–523, 1997.

[13] JL Smith. Determination of three-dimensional structure by multiwavelength anomalous diffraction:. *Current Opinion in Structural Biology*, 1(6):1002–1011, 1991.

[14] DB Cowie and GN Cohen. Biosynthesis by escherichia coli of active altered proteins containing selenium instead of sulfur. *Biochimica et Biophysica Acta*, 26(2):252–261, 1957.

[15] RBG Ravelli and EF Garman. Radiation damage in macromolecular cryocrystallography. *Current opinion in structural biology*, 16(5):624–629, 2006.

[16] E de La Fortelle and G Bricogne. Maximum-likelihood heavy-atom parameter refinement for multiple isomorphous replacement and multiwavelength anomalous diffraction methods. *Methods in enzymology*, 276:472–494, 1997.

[17] TA Jones. A graphics model building and refinement system for macromolecules. *Journal of Applied Crystallography*, 11(4):268–272, 1978.

[18] TA Jones, JY Zou, SW Cowan, and M Kjeldgaard. Improved methods for building protein models in electron density maps and the location of errors in these models. *Acta Crystallographica Section A*, 47(2):110–119, 1991.

[19] MG Rossmann and DM Blow. The detection of sub-units within the crystallographic asymmetric unit. *Acta Crystallographica*, 15(1):24–31, 1962.

[20] CB Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[21] DM Blow. Molecular replacement: Noncrystallographic symmetry. In MG Rossmann and E Arnold, editors, *International Tables for Crystallography*, volume F, chapter 13, pages 263–268. 1st edition, 2001.

[22] P Evans and A McCoy. An introduction to molecular replacement. *Acta Crystallographica Section D*, 64(1):1–10, 2007.

[23] GM Sheldrick, HA Hauptman, CM Weeks, R Miller, and I Usón. Direct methods: *ab initio* phasing. In MG Rossmann and E Arnold, editors, *International Tables for Crystallography*, volume F, chapter 16, pages 333–345. 1st edition, 2001.

[24] AL Patterson. A fourier series method for the determination of the components of interatomic distances in crystals. *Physical Review Letters*, 46: 372–376, 1934.

[25] MG Rossmann and E Arnold. Patterson and molecular-replacement techniques. In U Shmueli, editor, *International Tables for Crystallography*, volume B, chapter 2, pages 235–263. 2nd edition, 2001.

[26] C Giacovazzo. Direct methods. In U Shmueli, editor, *International Tables for Crystallography*, volume B, chapter 2, pages 210–234. 2nd edition, 2001.

[27] I Usón and GM Sheldrick. Advances in direct methods for protein crystallography. *Current opinion in structural biology*, 9(5):643–648, 1999.

[28] R Miller, GT De Titta, R Jones, DA Langs, CM Weeks, and HA Hauptman. On the application of the minimal principle to solve unknown structures. *Science*, 259:1430–1430, 1993.

[29] CM Weeks, GT DeTitta, R Miller, and HA Hauptman. Application of the minimal principle to peptide structures. *Acta Crystallographica Section D*, 49 (1):179–181, 1993.

[30] GM Sheldrick, CJ Gilmore, HA Hauptman, CM Weeks, R Miller, and I Usón. Direct methods: *ab initio* phasing. In MG Rossmann, E Arnold, and Himmel M, editors, *International Tables for Crystallography*, volume F, chapter 16, pages 413–432. 2nd edition, 2011.

[31] C Frazão, L Sieker, GM Sheldrick, V Lamzin, J LeGall, and MA Carrondo. *Ab initio* structure solution of a dimeric cytochrome c 3 from desulfovibrio gigas containing disulfide bridges. *Journal of Biological Inorganic Chemistry*, 4(2):162–165, 1999.

[32] G Oszlányi and A Suto. *Ab initio* structure solution by charge flipping. *Acta Crystallographica Section A*, 60(2):134–141, 2004.

[33] G Oszlanyi and A Suto. The charge flipping algorithm. *Acta Crystallographica Section A*, 64(1):123–134, 2007.

[34] C Dumas and A van der Lee. Macromolecular structure solution by charge flipping. *Acta Crystallographica Section D*, 64(8):864–873, 2008.

[35] MC Burla, C Giacovazzo, and G Polidori. From a random to the correct structure: the vld algorithm. *Journal of Applied Crystallography*, 43(4):825–836, 2010.

[36] JD Watson and FHC Crick. Molecular structure of nucleic acids. *Nature*, 171 (4356):737–738, 1953.

[37] Encyclopedia Britannica. Encyclopedia britannica online, 2013. URL `http://www.britannicAcom/EBchecked/topic/574200/supercomputer`.

[38] U Schwiegelshohn, RM Badia, M Bubak, M Danelutto, S Dustdar, F Gagliardi, A Geiger, L Hluchy, D Kranzlmüller, and E Laure. Perspectives on grid computing. *Future Generation Computer Systems*, 26(8):1104–1115, 2010.

[39] I Foster, C Kesselman, and S Tuecke. The anatomy of the grid - enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15:2001, 2001.

[40] G von Laszewski and K Amin. Grid middleware. In Q. Mahmoud, editor, *Middleware for Communications*, chapter 5, pages 109–130. Wiley Online Library, 2004.

[41] T Tannenbaum, D Wright, K Miller, and M Livny. Condor – a distributed job scheduler. In Thomas Sterling, editor, *Beowulf Cluster Computing with Linux*, pages 307–350. MIT Press, 2001.

[42] M Ellert, M Grønager, A Konstantinov, B Kónya, J Lindemann, I Livenson, JL Nielsen, M Niinimäki, O Smirnova, and A Wäänänen. Advanced resource connector middleware for lightweight computational grids. *Future Generation computer systems*, 23(2):219–240, 2007.

[43] I Foster. Globus toolkit version 4: Software for service-oriented systems. *Journal of computer science and technology*, 21(4):513–520, 2006.

[44] A Streit, P Bala, A Beck-Ratzka, K Benedyczak, S Bergmann, R Breu, JM Daivandy, B Demuth, A Eifer, and A Giesler. Unicore 6 recent and future advancements. *Annals of Telecommunications*, 65(11):757–762, 2010.

[45] R Buyya, CS Yeo, S Venugopal, J Broberg, and I Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.

[46] LM Vaquero, L Rodero-Merino, J Caceres, and M Lindner. A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1):50–55, 2008.

[47] D Thain, T Tannenbaum, and M Livny. Distributed computing in practice: The condor experience. *Concurrency and Computation: Practice and Experience*, 17(2-4):323–356, 2005.

[48] R Caliandro, B Carrozzini, GL Cascarano, L De Caro, C Giacovazzo, and D Siliqi. Phasing at resolution higher than the experimental resolution. *Acta Crystallographica Section D*, 61(5):556–565, 2005.

[49] Y Jia-xing, MM Woolfson, KS Wilson, and EJ Dodson. A modified *ACORN* to solve protein structures at resolutions of 1.7Å or better. *Acta Crystallographica Section D*, 61(11):1465–1475, 2005.

[50] TC Terwilliger. Automated main-chain model building by template matching and iterative fragment extension. *Acta Crystallographica Section D*, 59(1):38–44, 2003.

[51] A Perrakis, R Morris, and VS Lamzin. Automated protein model building combined with iterative structure refinement. *Nature Structural Biology*, 6: 458–463, 1999.

[52] GM Sheldrick. Macromolecular phasing with shelxe. *Zeitschrift für Kristallographie*, 217(12/2002):644–650, 2002.

[53] DD Rodríguez, C Grosse, S Himmel, C González, IM de Ilarduya, S Becker, GM Sheldrick, and I Usón. Crystallographic *ab initio* protein structure solution below atomic resolution. *Nature methods*, 6(9):651–653, 2009.

[54] AJ McCoy, RW Grosse-Kunstleve, PD Adams, MD Winn, LC Storoni, and RJ Read. Phaser crystallographic software. *Journal of applied crystallography*, 40(4):658–674, 2007.

[55] GM Sheldrick. A short history of *SHELX*. *Acta Crystallographica Section A*, 64(1):112–122, 2008.

[56] MD Winn, CC Ballard, KD Cowtan, EJ Dodson, P Emsley, PR Evans, RM Keegan, EB Krissinel, AGW Leslie, and A McCoy. Overview of the ccp4 suite and current developments. *Acta Crystallographica Section D*, 67 (4):235–242, 2011.

[57] PD Adams, PV Afonine, G Bunkoczi, VB Chen, IW D, N Echols, JJ Headd, LW Hung, GJ Kapral, and RW Grosse-Kunstleve. Phenix: a comprehensive python-based system for macromolecular structure solution. *Acta Crystallographica Section D*, 66(2):213–221, 2010.

[58] AJ McCoy. Liking likelihood. *Acta Crystallographica Section D*, 60(12):2169–2183, 2004.

[59] Wikipedia. Euler angles — wikipedia, the free encyclopedia, 2013. URL `http://en.wikipediAorg/wiki/Euler_angles`.

[60] RJ Read. Pushing the boundaries of molecular replacement with maximum likelihood. *Acta Crystallographica Section D*, 57(10):1373–1382, 2001.

[61] LC Storoni, AJ McCoy, and RJ Read. Likelihood-enhanced fast rotation functions. *Acta Crystallographica Section D*, 60(3):432–438, 2004.

[62] AJ McCoy, RW Grosse-Kunstleve, LC Storoni, and RJ Read. Likelihood-enhanced fast translation functions. *Acta Crystallographica Section D*, 61(4): 458–464, 2005.

[63] GM Sheldrick. Experimental phasing with *SHELXC/D/E*: combining chain tracing with density modification. *Acta Crystallographica Section D*, 66(4): 479–485, 2010.

[64] I Usón, CEM Stevenson, DM Lawson, and GM Sheldrick. Structure determination of the *O*-methyltransferase NovP using the 'free lunch algorithm' as implemented in *SHELXE*. *Acta Crystallographica Section D*, 63(10):1069–1074, 2007.

[65] A Thorn and GM Sheldrick. Extending molecular-replacement solutions with shelxe. *Acta Crystallographica Section D*, 69(11):2251–2256, 2013.

[66] R Caliandro, B Carrozzini, GL Cascarano, L De Caro, C Giacovazzo, A Mazzone, and D Siliqi. *Ab initio* phasing of proteins with heavy atoms at non-atomic resolution: pushing the size limit of solvable structures up to 7890 non-H atoms in the asymmetric unit. *Journal of Applied Crystallography*, 41 (3):548–553, 2008.

[67] MC Burla, R Caliandro, MC Camalli, GK Cascarano, L De Caro, C Giacovazzo, D Polidori, GS, and R Spagna. Il milione: a suite of computer programs for crystal structure solution of proteins. *Journal of Applied Crystallography*, 40(3):609–613, 2007.

[68] MC Burla, B Carrozzini, GL Cascarano, C Giacovazzo, and G Polidori. Advances in the vld algorithm. *Journal of Applied Crystallography*, 44(6):1143–1151, 2011.

[69] Wikipedia. Giuseppe arcimboldo — wikipedia, the free encyclopedia, 2013. URL `http://en.wikipediAorg/w/index.php?title=Giuseppe_Arcimboldo&oldid=537180609`.

[70] M Fujinaga and RJ Read. Experiences with a new translation-function program. *Journal of Applied Crystallography*, 20(6):517–521, 1987.

[71] FX Gomis-Rüth, M Solà, P Acebo, A Párraga, A Guasch, R Eritja, A González, M Espinosa, G del Solar, and M Coll. The structure of plasmid-encoded transcriptional repressor copg unliganded and bound to its operator. *The EMBO journal*, 17(24):7404–7415, 1998.

[72] E Nowak, S Panjikar, and PA Tucker. to be published.

[73] C Bieniossek, P Schütz, M Bumann, A Limacher, I Usón, and U Baumann. The crystal structure of the carboxy-terminal domain of human translation initiation factor eif5. *Journal of molecular biology*, 360(2):457–465, 2006.

[74] E Alexopoulos, A Küsel, George M Sheldrick, U Diederichsen, and I Usón. Solution and structure of an alternating D,L-peptide. *Acta Crystallographica Section D*, 60(11):1971–1980, 2004.

[75] RJ Morris and G Bricogne. Sheldrick's 1.2å rule and beyond. *Acta Crystallographica Section D*, 59(3):615–617, 2003.

[76] B Kuhlman and D Baker. Native protein sequences are close to optal for their structures. *Proceedings of the National Academy of Sciences*, 97(19): 10383–10388, 2000.

[77] F DiMaio, TC Terwilliger, RJ Read, A Wlodawer, G Oberdorfer, U Wagner, E Valkov, A Alon, D Fass, and HL Axelrod. Improved molecular replacement by density-and energy-guided protein structure optization. *Nature*, 473(7348): 540–543, 2011.

[78] GG Krivov, MV Shapovalov, and RL Dunbrack Jr. Improved prediction of protein side-chain conformations with scwrl4. *Proteins: Structure, Function, and Bioinformatics*, 77(4):778–795, 2009.

[79] AG Milbradt, F Kerek, L Moroder, and C Renner. Structural characterization of hellethionins from helleborus purpurascens. *Biochemistry*, 42(8):2404–2411, 2003.

[80] A Pal, JE Debreczeni, M Sevvana, T Gruene, B Kahle, A Zeeck, and GM Sheldrick. Structures of viscotoxins a1 and b2 from european mistletoe solved using native data alone. *Acta Crystallographica Section D*, 64(9): 985–992, 2008.

[81] LJ McGuffin, K Bryson, and DT Jones. The psipred protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000.

[82] PK Mehta, J Heringa, and P Argos. A simple and fast approach to prediction of protein secondary structure from multiply aligned sequences with accuracy above 70%. *Protein Science*, 4(12):2517–2525, 2008.

[83] J Söding, A Biegert, and AN Lupas. The hhpred interactive server for protein homology detection and structure prediction. *Nucleic acids research*, 33(suppl 2):W244–W248, 2005.

[84] GM Sheldrick and PC SHELXTL. Version 6.12, an integrated system for solving, refining, and displaying crystal structures from diffraction data; siemens analytical x-ray instruments. *InC: Madison, WI*, 2001.

[85] Z Dauter. Efficient use of synchrotron radiation for macromolecular diffraction data collection. *Progress in biophysics and molecular biology*, 89(2):153–172, 2005.

[86] Z Dauter. Carrying out an optal experiment. *Acta Crystallographica Section D*, 66(4):389–392, 2010.

[87] S Sheriff and WA Hendrickson. Description of overall anisotropy in diffraction from macromolecular crystals. *Acta Crystallographica Section A*, 43(1):118–121, 1987.

[88] M Sammito, C Millán, DD Rodríguez, IM de Ilarduya, K Meindl, I De Marino, G Petrillo, RM Buey, JM de Pereda, and K Zeth. Exploiting tertiary structure through local folds for crystallographic phasing. *Nature methods*, 2013. doi: 10.1038/nmeth.2644.

[89] P Evans. Scaling and assessment of data quality. *Acta Crystallographica Section D*, 62(1):72–82, 2005.

[90] AG Franke, DD Rodríguez, M Chami, MM Khan, R Rudolf, J Bibby, A Hanashima, J Bogomolovas, E Castelmur, DK Rigden, I Usón, S Labeit, and O Mayans. under review.

[91] C Artola-Recolons, C Carrasco-López, LI Llarrull, M Kumarasiri, E Lastochkin, IM de Ilarduya, K Meindl, I Usón, S Mobashery, and JA Hermoso. High-resolution crystal structure of mlte, an outer membrane-anchored endolytic peptidoglycan lytic transglycosylase from escherichia coli. *Biochemistry*, 50(13):2384–2386, 2011.

[92] R Caliandro, D Dibenedetto, GL Cascarano, A Mazzone, and G Nico. Automatic-helix identification in patterson maps. *Acta Crystallographica Section D*, 68(1):1–12, 2011.

[93] EL Summers, K Meindl, I Usón, AK Mitra, M Radjainia, R Colangeli, D Alland, and VL Arcus. The structure of the oligomerization domain of lsr2 from mycobacterium tuberculosis reveals a mechanism for chromosome organization and protection. *PlOS one*, 7(6):e38542, 2012.

[94] I Usón, SI Patzer, DD Rodríguez, V Braun, and K Zeth. The crystal structure of the dimeric colicin m immunity protein displays a 3d domain swap. *Journal of structural biology*, 178(1):45–53, 2012.

[95] P Arede, T Botelho, T Guevara, I Usón, DC Oliveira, and FX Gomis-Ruth. Structure-function studies of the staphylococcal methicillin resistance antirepressor, mecr2. *Journal of Biological Chemistry*, 2013.

[96] OS Smart, TO Womack, C Flensburg, P Keller, W Paciorek, A Sharff, C Vonrhein, and G Bricogne. Exploiting structure similarity in refinement: automated ncs and target-structure restraints in buster. *Acta Crystallographica Section D*, 68(4):368–380, 2012.

[97] T Shi, RD Bunker, S Mattarocci, C Ribeyre, M Faty, H Gut, A Scrima, U Rass, SM Rubin, and D Shore. Rif1 and rif2 shape telomere function and architecture through multivalent rap1 interactions. *Cell*, 153(6):1340–1353, 2013.