Information-Driven Structural Modelling of Protein–Protein Interactions

ARTICLE *in* METHODS IN MOLECULAR BIOLOGY · JANUARY 2015

Impact Factor: 1.29 · DOI: 10.1007/978-1-4939-1465-4_18 · Source: PubMed

3 AUTHORS:

João P G L M Rodrigues
Stanford University
20 PUBLICATIONS **244** CITATIONS
SEE PROFILE

Ezgi Karaca
European Molecular Biology Laboratory
16 PUBLICATIONS **297** CITATIONS
SEE PROFILE

Alexandre M J J Bonvin
Utrecht University
224 PUBLICATIONS **8,948** CITATIONS
SEE PROFILE

For publication in Methods in Molecular Biology: Molecular Modelling of Proteins.

Ed. Andreas Kukol. Humana Press Inc.

Information-driven structural modelling of protein-protein interactions

"Information-driven Protein Docking"

João P.G.L.M. Rodrigues, Ezgi Karaca and Alexandre M.J.J. Bonvin*

Computational Structural Biology Group, Bijvoet Center for Biomolecular Research, Faculty of Science - Chemistry, Utrecht University, Padualaan 8, 3584 CH Utrecht, the Netherlands.

* Phone: +31.30.2533859, Fax: +31.30.2537623, Email: a.m.j.j.bonvin@uu.nl

i. Summary

Protein-protein docking aims at predicting the three-dimensional structure of a protein complex starting from the free forms of the individual partners. As assessed in the CAPRI community-wide experiment, the most successful docking algorithms combine pure laws of physics with information derived from various experimental or bioinformatics sources. Of these so-called 'information-driven' approaches, HADDOCK stands out as one of the most successful representatives. In this chapter, we briefly summarize which experimental information can be used to drive the docking prediction in HADDOCK, and then focus on the docking protocol itself. We discuss and illustrate with a tutorial example a 'classical' protein-protein docking prediction, as well as more recent developments for modelling multi-body systems and large conformational changes.

ii. Keywords

Biomolecular interactions; information-driven docking; conformational changes; multi-body docking; HADDOCK; molecular modelling.

1. Introduction.

Docking is defined as the modelling of the three dimensional (3D) structure of a molecular complex from its known unbound constituents. It was developed to aid in the structural elucidation of transient or weak interactions, which can be challenging to characterize experimentally due to, for example, difficulties in crystallization or because the molecular weight rules out a thorough classical NMR analysis. The advent of explicit treatment of molecular flexibility, together with better and more efficient algorithms for both sampling and scoring, has earned docking a solid reputation amongst experimentalists. In turn, this attention brought new challenges such as the prediction of large molecular assemblies, protein-nucleic acid complexes, high-throughput predictions of entire metabolic pathways, or understanding the molecular origins of binding affinity and specificity ***(1, 2)***.

Docking predictions rely usually on a combination of shape complementary and some energy functions to define the conformational landscape of the interacting molecules and identify near-native models, respectively. Since most of these functions are borrowed from molecular dynamics and structure prediction software, they suffer from the same limitations due to their approximate character***(3)***. To gain the upper hand, available (experimental) information was incorporated into the algorithms to greatly enhance their accuracy. In fact, this approach – information-driven docking - has become the most reliable and successful in the docking community, as shown in the latest assessment of the community-wide docking assessment experiment (CAPRI)***(4)(5)***.

Traditionally, information was incorporated in docking predictions as a post-sampling filter. This filter approach is simple and straightforward: it first blindly generates a large pool of models and then removes or penalizes models that do not agree with the data. Its disadvantage lies in the need for a large number of solutions that should cover, ideally, the entire conformational search space. Often, as in the case of large or extremely flexible systems, the computational cost associated with exhaustive sampling of the search space is prohibitively high. Therefore, an alternative is to incorporate the data directly during the sampling stage – the so-called *information-driven docking*. This effectively biases the algorithm to visit only regions of the interaction space that respect the information contained in the data and thus enriches the pool of generated models with 'correct' models, provided of course the information is correct (which can be a pitfall of data-driven approaches)***(6)***

HADDOCK is an information-driven docking software ***(7)*** that was originally adapted from the NMR automated structure determination approach ARIA***(8)***. It uses CNS (Crystallography and NMR System) ***(9, 10)*** as computation engine and, as such, has access to a variety of energy functions that allow inclusion a variety of NMR-derived parameters such as chemical shift perturbations, NOE distances, residual dipolar couplings, and pseudo-contact shifts to drive the docking prediction (please see the Chapter in this book about *"NMR-based modelling and refinement of protein 3D structures"* by Vranken *et al.* for a description of the various NMR data). Throughout the years, HADDOCK has been extended to support other types of information commonly generated in the 'wet-lab' such as mutagenesis, chemical cross-

linking, and SAXS, as well as 'dry-lab' interface predictions from bioinformatics methods***(11-13)***. How this information is incorporated into HADDOCK depends on its characteristics. For instance, information such as NMR chemical shift perturbation (CSP), alanine scanning mutagenesis, chemical cross-linking, or EPR distances are translated into distance restraints. These restraints can have different levels of accuracy and ambiguity (one-to-many or one-to-one relationships, i.e. some highly ambiguous, e.g. from CSP, some very specific, e.g. EPR distances). These are generally implemented as an additional term in the energy function used to describe the conformational landscape. Lower resolution sources of information such as SAXS, Cryo-EM, or collision cross-section data (CCS) from ion-mobility mass spectrometry are currently used only for scoring in HADDOCK; this is done by measuring the discrepancy between the structural properties back-calculated from the generated models and the experimentally measured values. Information about the radius of gyration of the molecule can also be extracted from SAXS data; this one-dimensional value can in principle also be restrained during the sampling.

In this chapter, we will focus on *information-driven docking* with HADDOCK and describe how different types of data can be used in the modelling. We will illustrate this in a protocol example making use of HADDOCK and NMR chemical shift perturbation data to model the phosphoryl transfer complex between the signal transducing proteins HPr and IIa(glucose) of E.coli (PDB entry 1GGR)***(14)***.

2. Theory.

This section briefly discusses various useful information sources, how these can be used to drive docking predictions, and describes the HADDOCK strategy to produce structural models of biomolecular complexes. Since HADDOCK uses CNS for its structure calculations, details on the implementation of particular restraint type is best found in the CNS website (http://cns-online.org/v1.3) or related publications ***(9, 10)***.

2.1 Sources of information for data-driven docking

2.1.1 Common NMR Structural Information sources

The majority of data derived from NMR, such as NOE inter-proton distances, hydrogen bonds, residual dipolar couplings, relaxation diffusion anisotropy, pseudo-contact shifts and paramagnetic relaxation enhancements, can be applied directly to docking. Please refer to the chapter *"NMR-based modelling and refinement of protein 3D structures"* for a detailed explanation of these restraints.

Some NMR experiments and other techniques particularly useful for the structural elucidation of interactions are shortly described below.

2.1.2 NMR Chemical Shift Perturbations

Chemical Shift Perturbations are a simple strategy to identify regions of a protein that interact with the partner molecule. Atomic nuclei register changes in their chemical environment as small perturbations in their well-defined

chemical shifts. By labelling one of the components of the complex and titrating the other partner unlabelled, it is possible to follow which chemical shifts are displaced when compared to the spectra of the isolated partner, since the proximity of the partner will alter the chemical environment of those residues in the interface. The reverse experiment (first partner unlabelled and second labelled) can be done to map the interface on the other partner. The downside of this technique is that it cannot distinguish between perturbations caused by the proximity of the partner molecule and those caused by allosteric effects, conformational changes upon binding, or solvent reshuffling at the interface.

2.1.3 NMR Cross-Saturation

In cross-saturation experiments, one of the interacting partners is $^{2}H/^{15}N$ uniformly labeled. The only observable protons are those that can exchange back with protons of water (e.g. amide protons). Upon irradiation with a radiofrequency pulse, the unlabeled protein protons become instantly saturated due to spin diffusion effects. Afterwards, cross-relaxation phenomena transfer this saturation to neighboring interfacial protons on the labeled protein, reducing their peak intensity in a ^{15}N HSQC spectrum. Reversing the labeling on the partners allows this experiment to pinpoint accurate information on the binding interface. Since this experiment relies on direct through-space interactions, it is more reliable than chemical shift perturbation experiments, particularly for interactions where large conformational changes occur upon binding.

2.1.4 Hydrogen/Deuterium Exchange

H/D exchange provides information on the solvent accessible residues of a protein. In a deuterated medium, amide protons exposed to the solvent exchange rapidly while those buried by the protein structure do not. Upon interaction, the interface of the proteins also becomes inaccessible to solvent exchange. Following this event by either NMR with ^{15}N HSQC spectra or by mass spectrometry reveals the solvent-accessible surface of the bound complex and, indirectly, the interfacial residues.

2.1.5 NMR Pseudocontact Shifts

Pseudocontact shift (PCS) experiments require a paramagnetic ion attached to the protein, much like paramagnetic relaxation enhancement experiments, and are usually measured in ^{15}N HSQC or ^{13}C HSQC spectra***(15)***. When comparing a reference (diamagnetic) spectrum with a paramagnetic spectrum recorded in the presence of, for example, a paramagnetic lanthanide ion, PCSs can be measured as the differences in the chemical shifts between both spectra. Intramolecular PCSs can be used to optimize the ΔX tensor parameters of the protein to which the lanthanide is attached, while intermolecular PCSs can be used to obtain the anisotropic tensor ΔX parameters with respect to the second protein. Since both ΔX tensors are theoretically equal, they can be used to derive relative orientations between the two interacting proteins. Furthermore, given the distance-dependence of the PCS effect, this experiment also provides (long-range) distance information between the proteins.

2.1.6 Residual dipolar couplings

Residual dipolar couplings (RDCs) are a chemical phenomenon manifested as an increase or decrease in the magnitudes of multiplet splittings that can be seen in undecoupled NMR spectra***(16)***. These couplings can be measured in solution by inducing a weak alignment of the molecule, which can be done using a variety of methods***(17)***. RDCs provide information on the orientation of the internuclear vector of the two atoms for which the RDC is measured (e.g. N-C, N-H) with respect to the three global axes of the alignment tensor. This information can be used in the context of docking to orient the binding partners with respect to each other, thus reducing the number of degrees of freedom to be sampled during the docking calculation ***(18, 19)***.

2.1.7 Long Distance Information

NOE-derived distances are typically short, below 5-6Å. Larger distances can be reported by other experimental sources, such as chemical cross-linking detected by mass spectrometry (the distance depends on the linker length and flexibility), EPR (20-80Å), FRET (~50Å). These can be used to define upper limits in the docking predictions. These techniques been used, for instance, in the characterization of very large molecular assemblies that are not amenable for NMR***(20-22)***.

2.1.8 Low-resolution Shape Information (Cryo-EM, SAXS, CCS)

Low-resolution information obtained through Cryo-EM, SAXS, and CCS experiments provide overall shape information that can be used either to limit the conformational space search in sampling stages, or to filter ‘incorrect’

models after the sampling. Cryo-EM information is often simpler and faster to obtain compared to NMR experiments, in particular for very large multi-body assemblies like chaperonins or the nuclear pore complex, since there are less requirements for sample preparation and there is no peak assignment step to be carried out, although it still requires plenty of manual intervention and can become challenging when conformational heterogeneity is present. The resulting electron density maps allow, depending on the resolution, the identification and positioning of the interacting partners in the density map. SAXS provides a scattering curve and allows the calculation of the radius of gyration of the assembly. It can also be used to generate molecular envelopes that can be used in a similar manner as Cryo-EM maps. Collision Cross Section (CCS) data from ion mobility mass spectrometry provides a rotationally averaged two-dimensional projection of the molecules. Currently, HADDOCK implements SAXS (and CCS data) as a filter***(23)***. These can be best used after the rigid-body energy minimization step, and only for complexes whose components show asymmetry in their molecular shapes.

2.1.9 Bioinformatics Predictions

In the absence of experimental data, there is a wealth of information stored in sequence and structure databases that can be manipulated to provide predictions on the interface of the complex***(11)***. Of the many algorithms and web-servers available to predict protein interfaces, one is routinely used in tandem with HADDOCK: CPORT***(24)***. This server makes use of the structure of the free proteins to search for possible interfacial residues by sequence and structure homology. Other approaches based on correlated mutation from

evolutionary records have been proposed that predict unambiguous contacts between interacting partners***(25, 26)***.

2.2 Protein-Protein HADDOCKing

2.2.1 Docking protocol

Docking and flexible refinement in HADDOCK are performed in three successive stages:

- *it0*: Rigid-Body Energy Minimization (RBEM)
- *it1*: Semi-Flexible Simulated Annealing (SA) in Torsion Angle Space (TAD/SA)
- *water*: Restrained Molecular Dynamics in Explicit Solvent

These are preceded by a structure/topology generation stage that rebuilds missing atoms if necessary and a post-processing stage in which various energy terms, restraint violations and intermolecular contact are analysed.

2.2.1.1 Rigid-Body Energy Minimization (RBEM, it0)

In the initial docking stage, the interacting partners are first separated in space and each is randomly rotated around its centre of mass to remove any orientational bias. They are then subjected to a rigid-body energy minimization protocol, where first only the orientation of the partners is optimized, and then both rotations and translations are allowed, effectively resulting in the docking of the molecules. Given the fast calculation of each docking model at this stage, it is typically worth generating a large number of models to cover the interaction space. By default, 1000 are written to disk,

although 10000 are sampled – each model is the result of five internal docking trials with, for each, the 180^{o}-rotated solution around the normal to the interface being sampled as well. These models are then ranked according to the HADDOCK score (see below), and a fraction of these is selected for further flexible refinement – typically 200.

2.2.1.2 Semi-Flexible Simulated Annealing in Torsion Angle Space (TAD/SA, it1)

The second stage of the HADDOCK protocol fine-tunes each complex by flexible refinement of its interface. This second stage starts with a rigid-body SA step to optimize the orientation of the components. Then, the side chains of the interface – automatically defined for each docking model as all residues within 10Å of a partner molecule – are allowed to move in a second SA stage. A third and final SA stage optimizes both backbone and side-chains of the interface residues to allow for some conformational rearrangements. Finally, a short energy minimization in Cartesian space relaxes the models. A new ranking of the models is produced at this step, but, usually, all models are allowed to undergo the third and final refinement step in explicit solvent.

2.2.1.3 Restrained Molecular Dynamics in Explicit Solvent (water)

By default, both the RBEM and TAD/SA stages do not include any explicit description of the solvent (see **Note 1**). The docking is performed in vacuum with a dielectric constant (ε) of 10 for the electrostatic Coulomb energy term (should be set to 78 in case of protein-DNA docking). To improve the network of hydrogen bonds and electrostatic interactions at the interface, as well as

increase the realism of the prediction, the third and final step of the HADDOCK protocol is a short restrained molecular dynamics simulation in Cartesian space in a shell of explicit solvent, either TIP3P (water, 8Å shell) or DMSO (lipid-mimic, 12.5Å shell).

2.2.2 The HADDOCK Score

All structure calculations in HADDOCK are bound to a set of energetics terms, which together form the HADDOCK score. This score is a weighted sum of terms whose weights depend on the stage of the HADDOCK protocol:

- *(it0)* $E = 0.01\ E_{vdW} + 0.1\ E_{elec} + 1.0\ E_{desolv} - 0.01\ BSA + 0.01\ E_{AIR}$
- *(it1)* $E = 1.0\ E_{vdW} + 1.0\ E_{elec} + 1.0\ E_{desolv} - 0.01\ BSA + 0.1\ E_{AIR}$
- *(water)* $E = 1.0\ E_{vdW} + 0.2\ E_{elec} + 1.0\ E_{desolv} + 0.01\ E_{AIR}$

, where E_{vdW} and E_{elec} represent Lennard-Jones and Coulomb potentials, E_{desolv} is an empirical desolvation term developed by Fernandez-Recio et. al***(27)***, BSA is the buried surface area of the model in Ångstrom, and E_{AIR} is the energy reflecting the accordance of the model to the input restraints (the distance-based ones). Other terms might be included, such as E_{sym} for symmetry restraints or E_{RDCs} for residual dipolar coupling, depending on the application.

2.2.3 Clustering of Final Solutions

HADDOCK models are not analyzed on a per model basis. Instead, it is assumed that groups of structurally similar models with overall low HADDOCK

score are the best representatives of the near-native conformation. To form these groups, HADDOCK offers two clustering algorithms. The default choice in HADDOCK2.2 is a contact-based algorithm that groups models based on the similarity of their contact networks at the interface – fraction of common contacts (FCC)***(28)***. This is particularly efficient for large molecules, multi-body assemblies, and symmetrical complexes. The other choice is a standard RMSD-based clustering algorithm***(29)*** that uses the backbone interface-ligand RMSD as a distance measure. The backbone interface-ligand RMSD is calculated by first fitting the models on the interface of the first component of the complex (which should be the largest). Then, the RMSD is computed on the interface of the remaining components. The defaults cut-offs for each algorithm are 0.75 for FCC clustering and 7.5Å for backbone interface-ligand RMSD.

2.3 Restraints implemented in HADDOCK

2.3.1 Ambiguous Interaction Restraints (AIRs)

Several experimental techniques provide information on residues that are potentially involved in the interaction, but fail to report on the specificity of the residue pair interactions (i.e. they produce surface patches but not the specific pairwise residue contacts). HADDOCK implements this information as Ambiguous Interaction Restraints (AIRs). This concept, similar to ambiguous NOEs***(30)***, is designed to create an attraction between the interfaces during the docking without favoring a particular orientation. To support this implementation, HADDOCK divides interacting residues in two classes –

active and passive – that differ in their contribution to the binding event. Active residues are usually those involved directly in the interaction, for which there is strong experimental or predictive information, while passive residues comprise those that are surface accessible neighbors of active residues. Active residues will be forced to be at the interface, otherwise generating a restraint violation, while passive residues may or may not be at the interface (if not, no violation is generated). AIRs are generated between each active residue on one partner and all active and passive residues on the other partner(s). The total number of AIRs is equal to the sum of active residues. False negatives (missing information) are dealt with usually by automatic definition of passive residues, while false positives can be minimized by randomly removing a fraction (50%, by default) of the restraints for each docking trial. Internally, HADDOCK (or rather CNS) uses lists of active and passive residues for each interacting partner to define an *effective distance* for each active residue. This distance is defined between an active residue *i* of protein *A* and all active and passive residues of protein *B* as,

$$d_{iAB}^{eff} = \left(\sum_{m_{iA}=1}^{N_{A\ atom}} \sum_{k=1}^{N_{resB}} \sum_{n_{kB}=1}^{N_{B\ atom}} \frac{1}{d_{m_{iA} n_{kB}}^{6}} \right)^{-\frac{1}{6}}$$

where $N_{A\ atom}$ indicates all atoms of residue *i* on protein *A*, N_{resB} indicates each active and passive residue on protein *B*, $N_{B\ atom}$ indicates all atoms of the residue *k* on protein *B*, and *d* is the Euclidean distance between atoms m_{iA} and n_{kB}. The effective distances are restrained to a target value by a flat

bottom harmonic potential with upper and lower bounds. This potential switches to a linear potential beyond the upper bound to avoid large forces that could cause the calculations to fail. By default, HADDOCK sets the effective upper bound to 2Å and the lower bound to 0Å (the van der Waals energy term prevents potential atom overlap), meaning that if an effective distance d^{eff} greater than 2Å separates pair of restrained residues, these feel an attractive force. This seemingly small distance is a by-product of the mathematical formula (it grows smaller with the number of distances entering the sum), and in reality translates to real distances of 3-5Å between atoms of the active/passive residues in the model.

2.3.2 Unambiguous Distance Restraints

When the information is sufficiently accurate to allow unambiguous pairing of atoms or residues between partner molecules, it is possible to incorporate it in HADDOCK as unambiguous distance restraints. These use the same functional form of the distance restraining potential as the AIRs (flat bottom harmonic potential with upper and lower bounds transitioning to a linear potential after the upper bound to avoid large forces). In practice, ambiguous and unambiguous boils down to the syntax in which the restraints are written: either with multiple atom selections linking one residue or group of atoms to many in the partner molecule, or just to one specific pair of atoms. The main difference is that unambiguous restraints are never randomly discarded while ambiguous restraints are by default. Ambiguous and unambiguous distance restraints can therefore be combined and written in any of the two types of distance restraint files HADDOCK reads. Next to these two types, HADDOCK

also allows the definition of a third class of restraints as hydrogen bond restraints. These are treated in a similar manner as unambiguous restraints. Each restraint file can be activated or deactivated at various stages of the protocol.

2.3.3 Symmetry Restraints

If there is *a priori* information that hints or determines that the complex assumes a symmetrical arrangement, it is possible to incorporate this in the docking prediction and thus restrict quite substantially the conformational search space. HADDOCK supports several types of cyclic and dihedral symmetries, implemented through combinations of symmetrical distance restraints***(31, 32)***: C2, C3, C4, D2, and C5. Next to the symmetry restraints, non-crystallographic symmetry restraints (NCS restraints) can also be defined: these ensure that two molecules are similar without imposing any symmetry operation between them, equivalent to restraining the RMSD between the two molecules to be zero.

2.3.4 Centre of Mass Restraints

To ensure compactness of solutions, for example when using symmetry restraints without experimental information, HADDOCK allows the definition of distance restraints between the geometric centres of mass the molecules (based only on CA atoms). These so-called centre of mass restraints can also be used when there is no information about the binding interface (*ab initio* docking), albeit with a lower chance of success since the only factor in play are the physical terms of the scoring function.

2.3.5 Other orientational restraints.

Next to the distance-based restraints define above, HADDOCK supports a variety of orientational restraints from NMR, including residual dipolar couplings***(19)*** and relaxation anisotropy data that are useful in defining the relative orientation of the molecules, and pseudo-contact shifts that provide both distance and orientational information***(33)***. For details refer to the Chapter in this issue about *"NMR-based modelling and refinement of protein 3D structures"*.

3. Materials

The following 'Methods' describes an example protocol on the usage of HADDOCK to model the interaction between two proteins by using experimental information. In order to follow the example, the reader must have a number of software programs installed on his computer, together with the data provided in the Extra Materials (extras.springer.com). The protocol should be run on a Linux-based system (or under Mac OSX).

3.1 HADDOCK v2.2

HADDOCK can be obtained free of charge for academic users at http://nmr.chem.uu.nl/haddock. In principle, little is required to install and configure HADDOCK (detailed instructions are provided with the software). Also, a mailing-list is available to all current and potential users to provide advice and troubleshooting for any problem(s) encountered during installation and usage (http://groups.yahoo.com/group/haddock-discuss). Furthermore, the WeNMR project provides several tutorials, tips, and a help center related to HADDOCK and might also be of interest to all users (http://www.wenmr.eu/wenmr/support/documentation/nmr-services/haddock). Finally, a web server version of HADDOCK is also available, free of charge for academic users that offers a user-friendly interface to all the powerful features of this docking software (http://www.haddocking.org).

3.2 Crystallography and NMR Suite (CNS) v1.3

CNS can be obtained free of charge for academic users at http://cns-online.org/v1.3. The usage of particular experimental information to drive the

docking, such as pseudo-contact shifts (PCS) requires the installation of an additional module and re-compilation of CNS. Otherwise, the binaries provided at the web address are sufficient to run HADDOCK. HADDOCK v2.2 is designed for CNS v1.3, so ensure that the versions of the software are appropriate.

3.3 ProFit

The ProFit software calculates the root mean square deviation of atomic coordinates between molecules. It is designed to 'be the ultimate protein least squares fitting program' and it supports a powerful zone selection syntax. It can be obtained free of charge for academic users at the author's website (http://www.bioinf.org.uk/software/profit/).

3.5 NACCESS

NACCESS is a software program that uses the Lee & Richards method***(34)*** to calculate the solvent accessible area of a molecule from its three-dimensional PDB coordinates. It is available free of charge to academic and non-profit organisations (http://bioinf.manchester.ac.uk/naccess/).

3.6 PyMOL

PyMOL is an open-source molecular visualization system offering a large number of features and extensible through Python scripts. It is available in several formats depending on the affiliation and needs of the user at www.pymol.org.

3.7 Grace (xmgrace)

Grace is a simple 2D plotting software program that can be obtained free of charge here: http://plasma-gate.weizmann.ac.il/Grace

4. Methods

4.1 Modelling of complexes by information-driven docking using HADDOCK.

We describe here the use of the HADDOCK2.2 package for the modelling of a protein-protein complex using chemical shift perturbation data. We will use data from the `haddock2.2/examples/e2a-hpr` directory. You should first copy this directory to the directory you are working in (*see* **Note 2**):

```
cp -r $HADDOCK/examples/e2a-hpr .
```

4.1.1 Preparation of PDB files and input data

Make sure that your input models are compliant with the PDB format, particularly, the presence of an `END` statement as the last line of the file. Furthermore, the segment identifier (characters 73-76 in each `ATOM` statement) and chain identifier fields (character 22 in each `ATOM` statement) should be empty strings (i.e. filled with spaces). If you use a crystal structure, make sure that there are no double occupancies or residue insertions. If you are using an ensemble of models, split the file in individual files that contain only one structure (*see* **Note 3**).

As input data, you should combine chemical shift perturbation data (or other data indicating residues at the interface) and solvent accessibility data

calculated with NACCESS: use only those residues that have both a high enough chemical shift perturbation (*see* **Note 4**) and a high enough relative accessibility. In the example, the residue solvent accessibilities calculated with NACCESS are already provided in the files `e2a_1F3G.rsa` and `hpr/hpr_rsa_ave.lis` (the latter containing the average for the 10 starting models for hpr). From these files you can select the residues with high enough (e.g. >~40%) accessibility (*see* **Note 5**). You could calculate the accessibility values yourself using the following command:

```
naccess e2a_1F3G.pdb
```

4.1.2 Definition of active and passive residues

Passive residues are defined as the solvent accessible surface neighbours of active residues. To define and visualize them you can use a molecular visualization program, for example PyMOL,

```
pymol e2a_1F3G.pdb
```

Start by colouring the active residues, for example in red. Then, filter out the residues with a low solvent accessibility, using either the output of NACCESS, recommended, or an embedded tool of the visualization program (e.g. `get_area` command in PyMOL). Next, select all surface neighbours within a certain cut-off radius (e.g. 5Å), and that are solvent accessible, to define the passive residues and colour them for example in green. In the `e2a-hpr` example, several PyMOL scripts are provided with the respective residues already coloured according to this scheme: `e2a_pymol_active.pml`, `e2a_pymol_active_passive.pml` and similar for hpr. You can load these scripts in PyMOL using the following commands:

```
pymol e2a_1F3G.pdb
```

Then, in the PyMOL command line, type:

```
@e2a_pymol_active.pml
```

You will use the active and passive residues for both molecules to generate Ambiguous Interaction Restraints (AIRs); for this go to the HADDOCK GenTBL service (http://haddock.chem.uu.nl/services/GenTBL/) and follow the instructions. You should save the resulting file as `ambig.tbl` in the working directory; note that, in the `e2a-hpr` example directory, an example file names `e2a-hpr_air.tbl` is already present and can be used for comparison (*see* **Note 6**).

4.1.3 Setup of a new run: new.html

To set up a new run, go to the project setup page on http://www.nmr.chem.uu.nl/haddock, click on "start a new project" and follow the instructions. Depending on the experimental data you have available, you can input various data files such as ambiguous restraints, unambiguous restraints, RDCs etc. PCS restraints are not yet supported in the website, but an example case is provided with the HADDOCK software. After saving the `new.html` file to disk, type `haddock2.2` in the same directory. This will generate a run directory containing all necessary information to run haddock. An example of a `new.html` file can be found in the `e2a-hpr` directory as `new.html-refe`. (*see* **Note 7**) and is displayed below. Such a file can in principle also be created by manual editing.

```
<html>
<head>
```

```
<title>HADDOCK - start</title>
</head>
<body bgcolor=#ffffff>
<h2>Parameters for the start:</h2>
<BR>
<h4><!-- HADDOCK -->
AMBIG_TBL=./e2a-hpr_air.tbl<BR>
HADDOCK_DIR=../../<BR>
N_COMP=2<BR>
PDB_FILE1=./e2aP_1F3G.pdb<BR>
PDB_FILE2=./hpr/hpr_1.pdb<BR>
PDB_LIST2=./hpr-files.list<BR>
PROJECT_DIR=./<BR>
PROT_SEGID_1=A<BR>
PROT_SEGID_2=B<BR>
RUN_NUMBER=1<BR>
submit_save=Save updated parameters<BR>
</h4><!-- HADDOCK -->
</body>
</html>
```

4.1.4 Run.cns

The next step is to define all parameters to perform the docking run. For this, enter the newly created directory:

```
cd run1
```

You will find a file called `run.cns` containing all the parameters to run the docking, and which deserves special attention. You need to edit this file and define a few parameters such as the location of the CNS executable and the queue command to use. Other options such as the semi-flexible segments at the interface, or fully flexible segments (*see* **Note 8**), the number of models to generate at each stage, the clustering algorithm and cut-off, and the force constants for the several energy terms are also defined there. You can edit your `run.cns` file manually or via “project setup” on

http://www.nmr.chem.uu.nl/haddock. More information is available via the "run.cns" option in the manual section on http://www.nmr.chem.uu.nl/haddock.

4.1.5 Docking run

To actually start the docking run with HADDOCK type in the directory containing the `run.cns` file (*see* **Note 9**).

```
haddock2.2 >& haddock.out &
```

As more extensively explained in **Theory** section before and "the docking" section in the HADDOCK manual, the entire protocol consists of three stages. An initial topology and structure generation step validates and builds the structure files to be used in the docking. The initial models as provided by the user are written to `data/sequence/`.

- *Topology and model generation*. The resulting topologies (*.psf) and coordinates (*.pdb) files are written to the `begin/` directory (*see* **Notes 10 and 11**). There is one output file per chain – `generate_X.out`, where X is the segment identifier given in `run.cns` – that must be checked for errors if there is a problem at this stage (*see* **Note 12**).
- *Randomization and rigid body energy minimization.* The docked models are written to `structures/it0/`. When all models have been generated, HADDOCK will write the PDB files with names sorted according to the HADDOCK score (weights defined in the `run.cns`) to `file.cns`, `file.list` and `file.nam` in the same directory. The number of trials (`ntrials`, by default 5) and the sampling of 180

degrees rotated solutions (`rotate180_0`, by default true) can be modified in `run.cns`.

- *Semi-flexible simulated annealing*. The best models after rigid body docking (defined at `structures_1` in `run.cns` and by default 200) will be subjected to a semi-flexible simulated annealing (SA) in torsion angle space. The temperatures and number of steps for the various stages are also defined in `run.cns`. The resulting refined models are written into `structures/it1`. The numbering of the file names reflects their rank from the previous step (e.g. `complex_1.pdb` is the refined best ranked structure in `it0` according to the HADDOCK score). At the end of the calculation, HADDOCK generates the `file.cns`, `file.list` and `file.nam` files as in the previous stage (*see* **Note 13**). At the end of this stage, the models are analysed and the results are placed in the `structures/it1/analysis` directory (see the analysis section below).
- *Flexible explicit solvent refinement.* The choice of the solvent in which to refine the models is defined in `run.cns` (`solvent`) and can be either `water` or `dmso`. The resulting models are written in the `structures/it1/water` directory. The numbering in the files here matches that of the previous stage (e.g. `complex_1w.pdb` is the water refined `complex_1.pdb` of `it1`). At the end of the explicit solvent refinement, HADDOCK generates the `file.cns`, `file.list` and `file.nam` files. Finally, the models are analysed and the results are placed in the `structures/it1/water/analysis` directory (see the analysis section).

4.1.6 Automatic Analysis

A number of analysis scripts are automatically run after the semi-flexible and explicit solvent refinement stages and the results placed in `structures/it1/analysis` and `structures/it1/water/analysis`, respectively. Here we discuss a few of the most relevant output files.

- `e2a-hpr_fcc.disp`: contains the pairwise FCC matrix; this file is used as input for FCC clustering. If the clustering algorithm is RMSD, then the filename is `e2a-hpr_rmsd.disp`. The FCC measure, unlike RMSD, is asymmetric (FCC(AB) != FCC(BA)) so it produces a full matrix.
- `cluster.out`: contains the clusters generated from the abovementioned matrix. The clusters are numbered according to their size (number of models in the cluster) and not according to their HADDOCK score. This is related to the algorithm used to cluster the models.
- `noe.disp`: contains the number of distance restraints violations per structure and averaged over the ensemble over all distance restraint classes and for each class (unambiguous, ambiguous, hbonds) separately. Similar files are generated when you have RDC (`sani.disp`), relaxation anisotropy (`dani.disp`) or PCS (`pcs.disp`) restraints.
- `energies.disp`: contains the various energy terms per model and averaged over the ensemble.

- `ana_*.lis`: there is a set of files called ana*.lis where * can be dihed_viol, dist_viol_all, hbond_viol, hbonds, nbcontacts, noe_viol_all, noe_viol_ambig, noe_viol_unambig. The 'viol' refers to violations, and those files contain listings of violations including the number of times a restraint is violated as well as the average distance and violation per restraint. In addition, `ana_hbonds.lis` gives a listing of hydrogen bonds, and `ana_nbcontacts.lis` a listing of non-bonded contacts.
- `ene-residue.disp`: contains intermolecular energies for all interface residues.
- `nbcontacts.disp`: contains non-bonded contacts.

4.1.7 Manual Analysis

An important part of the analysis needs to be performed manually. A number of analysis scripts and programs are provided in the `tools` directory. These allow you to collect various statistics on the generated models and more importantly to perform the clustering of solutions and their analysis on a per-cluster basis.

- *Collecting statistics of the models with `ana_structure.csh`:* This script should be run once the `file.list` file has been created. It extracts various energy terms, violation statistics, and the buried surface area from each PDB file and calculates the RMSD of each structure compared to the lowest energy one (if the location of ProFit is defined (see installation and software links on http://www.nmr.chem.uu.nl/haddock)). The output are several files named "`structures*.stat`" that contain the same information

sorted in different ways. Usually, the most important file is `structures_haddock-sorted.stat`. From this file, you can generate a plot of the HADDOCK score as a function of the RMSD to the lowest energy model and investigate if the run produces an 'energy funnel', meaning that the low energy models should have small RMSD values and the high energy models should have large RMSD values (*see* **Figure 1**). A script called `make_ene-rmsd_graph.csh` is provided in `$HADDOCKTOOLS` and it produces a Grace compatible plot file. Specify two columns to extract data from and a filename:

```
$HADDOCKTOOLS/make_ene-rmsd_graph.csh 3 2 structures_haddock-sorted.stat
```

This will generate a file called `ene_rmsd.xmgr`, which you can display using xmgrace:

```
xmgrace ene_rmsd.xmgr
```

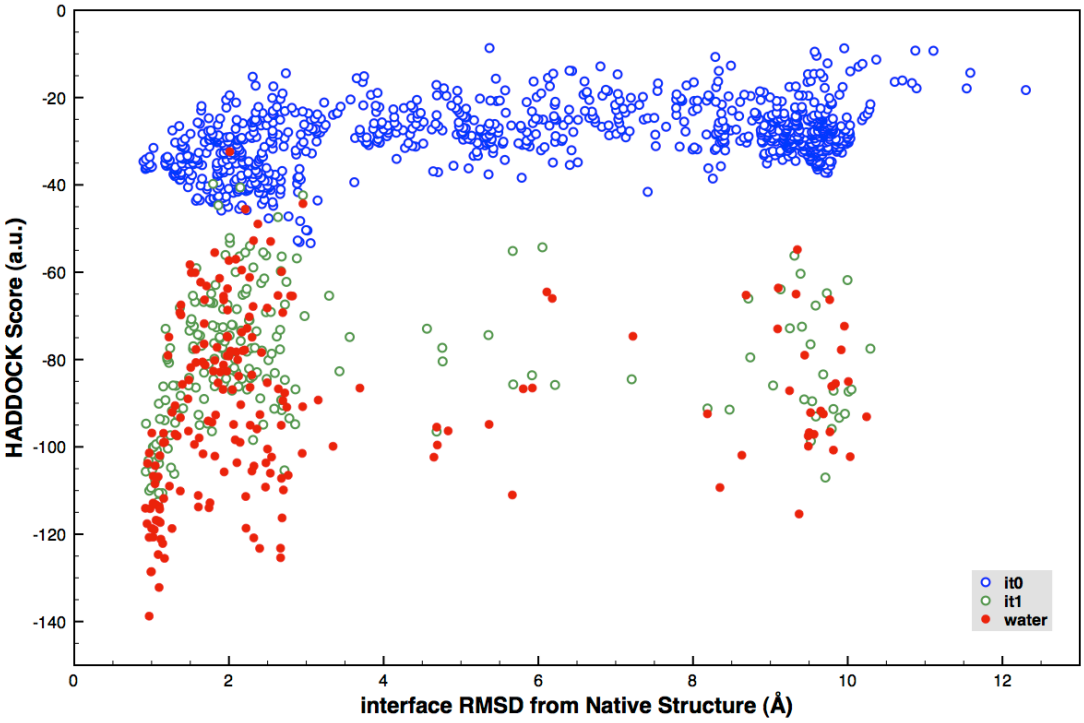

Figure 1. Plot of HADDOCK scores versus interface RMSD from the reference complex (PDB ID 1GGR) for the three stages of the docking protocol (blue, green, and red, for it0, it1, and water refinement respectively). One can clearly see a funnel at low RMSD values, corresponding to near-native solutions, becoming more apparent after flexible refinement.

- *Clustering of solutions*: The clustering is run automatically in `it1/analysis` and `it1/water/analysis`, based on the criteria defined in the `run.cns` file. However, try using different cut-offs for the clustering since it is difficult to know *a priori* the best RMSD/FCC cut-off. This value depends on the system under study and the number of experimental restraints used to drive the docking (*see* **Note 14**). For FCC clustering, the script to use it `cluster_fcc.py`, while for RMSD clustering, use the C program `cluster_struc` (this should have been compiled during the installation of HADDOCK). The scripts read the appropriate `e2a-hpr_*.disp` file

containing the pairwise matrix and generate clusters. The usage is (in the `analysis` directory):

```
python cluster_fcc.py e2a-hpr_fcc.disp cut-off [options] >cluster.out
```

or

```
cluster_struc [-f]  e2a-hpr_rmsd.disp cut-off min_size >cluster.out
```

Here cut-off indicates the FCC/RMSD cut-off to determine if two models belong in the same cluster. For FCC clustering, there are several options that can be modulated (type `python cluster_fcc.py -h` for the list and their explanation). In the RMSD clustering script, `min_size` is the minimum number of models in a cluster (typically a number like 4) and -f is an optional full-linkage clustering algorithm.

In either case, the output looks like the following:

```
  Cluster 1 -> 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 23 24 27 28
43
  Cluster 2 -> 25 26 29 32 34 35 57 71 73 20 21 44 39 46
  …
```

The numbers after the arrow correspond to the rank in `file.nam`. The 'sorted' models are also in the `analysis` directory. For example, 2 corresponds to the second model in the analysis directory, which is the second structure listed in `file.list` in `it1` or `it1/water`.

– *Analysis of the clusters with `ana_clusters.csh`:* This script takes the output of the clustering script, by default `analysis/cluster.out`, to perform an analysis of the various clusters, calculating average energies,

RMSDs, and buried surface area per cluster. The following runs the analysis on all clusters:

```
$HADDOCKTOOLS/ana_clusters.csh [-best #] analysis/cluster.out
```

The `-best #` is an optional argument to generate additional files with cluster averages calculated only on the best (#) ranked models of a cluster according to their HADDOCK score. This recommended option allows removing the dependency of the cluster averages on the size of the respective clusters (*see* **Note 15**). The `ana_clusters.csh` script analyses the clusters in a similar way as the `ana_structures.csh` script, but in addition generates average values over the models belonging to one cluster. It creates a number of files for each cluster containing the cluster number `clustX` in the name. It also creates files containing various averages over the clusters, `cluster_xxx.txt`; these contain the average and standard deviation of various terms such as intermolecular energy (`xxx=ene`) etc. Also, files combining all the above information and sorted based on various criteria are provided: `clusters.stat` that contains the various cluster averages and `clusters_xxx-sorted.stat` where `xxx` is the energy term according to which the values are sorted (e.g. `xxx=ene` for intermolecular energy, etc.). Again, the most relevant output file is `clusters_haddock-sorted.stat`, or rather `clusters_haddock-sorted.stat_bestX`.

- *Rerunning the HADDOCK analysis on a cluster basis*: Having performed the cluster analysis, you can now rerun the HADDOCK analysis for the best models of each cluster to obtain violation and energetics details and statistics.

To run this analysis, we need the cluster-specific `file.nam_clust#`, `file.list_clust#` and `file.cns_clust#` files. A script in the `tools/` directory called `make_links.csh` will move the original `file.nam`, `file.list` and `file.cns` files to `file.nam_all`, `file.list_all`, `file.cns_all` and the same with the `analysis` directory. It will then create links to the appropriate files (`file.nam_clust#`, ...) and to a new `analysis_clust#/` directory.

For example, to rerun the analysis for the best 10 models of the first cluster type in the `water` directory:

```
$HADDOCKTOOLS/make_links.csh clust1_best10
cd ../../..
haddock2.2
```

The `cd` command brings you back into the main run directory from where you start again HADDOCK. Only the analysis of the best 10 models of the first cluster in the water will be run. Once this is finished, go to the respective analysis directory and inspect the various files. The RMSD from the average models should now be low (check `rmsave.disp`).

Having run the HADDOCK analysis on a cluster basis for each cluster, you should now have new directories in the water directory, called `analysis_clustX_best10`. Each of these analysis directories contains now cluster specific statistics. You can also visualize the clusters, using for example PyMOL. We provide a Perl script in the `tools/` directory, `joinpdb`, which allows concatenation of the various PDB files into one single ensemble file:

```
    $HADDOCKTOOLS/joinpdb -o e2a-hpr_clust1.pdb e2a-
hprfit_*.pdb
    pymol e2a-hpr_clust1.pdb
```

In general, the top ranked models of the cluster with the lowest HADDOCK score are considered the representatives of the biological system. However, scoring in docking remains a difficult problem and we do recommend, if possible, using additional independent information to validate the results (e.g. mutagenesis data). The selected model should explain as much as possible all what is known about the system (*see* **Figure 2**).

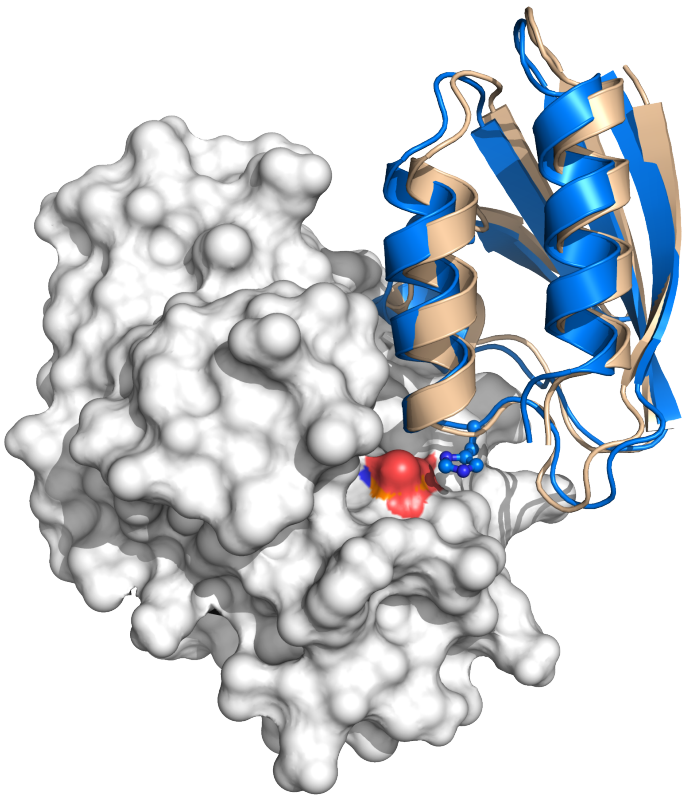

Figure 2. Superimposition of the top model of the best scoring cluster onto the native structure (PDB ID 1GGR). The molecules were superimposed on backbone atoms of E2A, which is shown in white surface representation with the phosphorylated histidine coloured according to the atom types (blue, red, and orange, for nitrogen, oxygen, and phosphorous, respectively). The HPR molecules are shown in cartoon representation (the

model in blue and the native in peach) and the histidine residue involved in the phosphate transfer in ball-and-stick. The model is in excellent agreement with the native structure (interface RMSD = 0.97Å). The proximity of the two histidines across the interface, which was not defined as a restraint in HADDOCK, is consistent with the biological function of this phosphor-transfer complex.

4.2 Other docking scenarios

Although the previous example illustrates a canonical dimer docking, HADDOCK also supports more advanced protocols. Users can model large macromolecular complexes, address substrate-induced conformational changes, and deal with extremely flexible peptides. These protocols are briefly explained below.

4.2.1 Multi-Body Docking

HADDOCK allows users to include up to six molecules and dock them simultaneously***(31)***. Multi-body HADDOCKing, as this protocol is called, follows the same rules of the original pairwise docking protocol requiring only that each molecule is restrained to at least another one of the system. The restraints between the different molecules are defined with the same syntax described in the case of dimer docking, and can be generated via the web server indicated before: http://haddock.chem.uu.nl/services/GenTBL/. Here, we recommend the user to also turn on centre-of-mass restraints, in order to ensure the compactness of the resulting models. If there is any cyclic and/or dihedral symmetry present, the user can activate the built-in symmetry restraints and impose them between and/or within each molecule (*see* **Section 2.3.3**).

4.2.2 Flexible Multi-Domain Docking

Modelling binding-induced large conformational changes is a major challenge of the docking community, since it requires sampling a vast and intricate conformational space. Unfortunately, addressing such a number of degrees of freedom is often out of reach for most of the current sampling methods, including the one of HADDOCK. To tackle this challenge, we developed a special application of the multi-body docking protocol***(35)*** that divides to conquer: the flexible partner is cut at hinge regions, and thus dissected into rigid domains that allow HADDOCK to sample a wider range of motions during rigid-body energy minimization.

The identification of the hinge regions can be carried out using normal mode analysis, such as that provided by the web server HingeProt (*see* **Note 16**). To ensure molecular integrity and biological realism, we define connectivity restraints (in the form of distance restraints) between the separated domains. These are first defined with a maximum distance of 10Å, to allow sampling of large range of motions. They are then shortened to a peptide bond distance (1.3Å) at the water refinement step. Imposing different connectivity restraints is possible by submitting both `unambig.tbl` and `hbonds.tbl` restraint files (a copy except for the connectivity distance), and changing the stages when these are active in `run.cns` (options `unambig_firstit (0)`, `hbond_firstit (2)` and `unambig_lastit (1)`, `hbind_lastit (2)`). It is also necessary to define the artificial termini as uncharged and the first three residues starting from the 'cut' hinge as fully flexible.

4.2.3 Protein-Peptide Docking

Albeit the other end of the size spectrum, small systems such as peptides are also challenging regarding sampling. Their extreme flexibility and the many conformations they can adopt upon binding makes them challenging to model and require usually long molecular dynamics simulations or other advanced sampling methods, none of which is possible or feasible, time-wise, for use in HADDOCK.

To cover the conformational landscape of peptides, we developed a shortcut approach. In this custom-tailored protocol, the peptide is provided as an ensemble of three most common conformations: α-helical, β-strand, and polyproline II. (*see* **Note 17**). Additionally, the number of MD steps in the flexible refinement stage needs to be increased four-fold to improve sampling efficiency (from 500/500/1000/1000 to 2000/2000/4000/4000). Finally, the peptide is defined entirely as fully flexible and the clustering algorithms are adapted for small molecules (*see* **Note 14**).

Notes

1. HADDOCK has a special feature – solvated docking – that allows water molecules to be introduced at the interface of the complex for entire duration of the docking protocol. This feature should only be used when the experimental information is accurate enough to drive the docking and the interface is expected to be ‘wet’. In short, solvated docking starts by surrounding each molecule by a shell (approximately 4Å wide) of water molecules, optimized via a short MD simulation, prior to the RBEM stage.

After the minimization, all water molecules not at the interface are removed. At the interface, only a fraction of the molecules is kept (by default 25%), with the removal being carried out via a biased Monte Carlo sampling method whose criteria is based on a statistical potential of amino acid – water contact propensities. Finally, energetically unfavourable water molecules (those with a positive intermolecular energy) are removed, which might lead to a complete desolvation of the interface, and another round of RBEM is performed to optimize the final complex. The remaining of the HADDOCK protocol remains unchanged, with the difference that interfacial water molecules might be included in the further refinement. We refer the reader to the following references for an in-depth explanation of solvated docking in HADDOCK: ***(36-39)***.

2. HADDOCK must be correctly installed for the `$HADDOCK` environment variable to be defined. Check the installation instructions provided with the software.

3. If your input PDBs contains missing segments, this might lead to domains drifting away during the refinement stage. To avoid this, simply define a few unambiguous distance restraints between CA atoms from the various 'sub-domains', setting the actual measured distance as a target distance and the bounds to 0.0. The same can be done to ensure that an ion coordination geometry is properly maintained. Missing residues at the interface or in hinge regions must be handled with extreme care not to compromise the biological integrity of the models. Missing atoms, on the

other hand, are not problematic since HADDOCK rebuilds them based on the topology files of the force field, as long as the residue name is defined in them. Alos, termini charges are very important for the docking protocol, as they can lead to artificial interactions. By default, termini are charged but they can be neutralized by using an appropriate linkage file (`protein-allhdg5-4-*.link` files in the `toppar/` directory). For example, to have both termini of molecule *A* uncharged, simply add in run.cns (option `prot_link_A`) the appropriate linkage file (`protein-allhdg5-4-noter.link`). Another important point concerns ions; if proper care is not taken, they can be problematic during the torsion angle dynamics stage. HADDOCK has an in-built mechanism that defines artificial bonds to 'chelate' the ion to the protein but it relies on proper ion naming. Check these names in the `covalions.cns` script and add yours if necessary. Also, make sure that their name in the PDB file matches the ion names defined in the `ion.top` file in the `toppar/` directory. To avoid that a N- or C-terminal patch be applied to them, they should also be defined in the `topallhdg5.4.pep` file (look for the "`first IONS`" and "`last IONS`" statements).

4. We have developed an automated method to discriminate the significant CSP – SAMPLEX***(40)***. It compares two sets of chemical shifts from two different samples (e.g. bound/unbound), and using the three-dimensional structure of the molecules, returns the confidence for each residue to be in a perturbed or unperturbed state.

5. The accessibility cut-off is not a hard limit; check the accessibilities and possibly include residues with lower accessibilities but with functionally important groups.

6. The syntax of the restraints is what determines their (un)ambiguous character, not the filename where they are stored: `ambig.tbl`, `unambig.tbl`, or `hbonds.tbl`. This allows for example to mix unambiguous and ambiguous restraints in the same file. The difference lies in the random removal option (`noecv=true`), which is applied only to `ambig.tbl`. In principle, one should use `ambig.tbl`; `unambig.tbl` and `hbonds.tbl` could be used, for example, to provide extra NOEs or other data (e.g. FMD connectivity restraints) for which one wants to use different force constants or for which there is exceptional certainty.

7. An important setting in `new.html` is the value of `N_COMP`. This should be set equal to the number of components of the complex (2 in case of a dimer, 3 for a trimer, etc.). Note that it can also be set to 1, in which case HADDOCK can be used for refinement instead of docking.

8. HADDOCK allows the definition of fully flexible regions: these are treated as fully flexible throughout all stages, except the initial rigid-body docking. This should be useful for cases where part of a structure are disordered or unstructured or when docking small flexible molecules onto a protein.

9. This command causes HADDOCK to run in the background and all output to be redirected to the `haddock.out` file. If at some stage HADDOCK stops producing new models and the run is not yet finished, search for error messages in the output files: `gunzip xxx.out.gz` where `xxx.out.gz` is a particular output file, and look for ERR in this file. Also, kill the current HADDOCK process:

```
ps -ef | grep haddock
kill -9 id
```

   Here `id` is the process id that is returned by the `ps -ef` command.

   You can restart a HADDOCK run, but before doing that, make sure to delete any *.out file from the run directory (see **Note 12**). HADDOCK will proceed from where it was in the calculations.

10. The OPLS force field used by HADDOCK is a mixed united/all-atom force field. This means that protons do not have van der Waals parameters of their own. Instead these are accounted for in the heavy atom parameters to which they are attached. In HADDOCK, by default, non-polar hydrogen atoms are deleted in order to speed up the calculation. This does not affect the resulting models significantly since the missing hydrogen atoms are actually accounted for in the united atoms parameters. You can change this behaviour by setting `delenph=true` in `run.cns`. This should be done in case classical NOE distance restraints are used, or in situations where the hydrogen atoms are extremely relevant (e.g. small molecule docking).

11. Topology generation is often the most problematic stage in HADDOCK. While most amino acids and their most common modifications are supported in HADDOCK, small ligands and exotic modifications to amino acids or nucleic acid bases that are not described in the force field will give an error and halt the docking protocol. A list of the supported modified amino acids is given here: http://haddock.science.uu.nl/services/HADDOCK/library.html. For those molecules not in the list, the user is left to generate their own parameterization scheme, using for example PRODRG***(41)***, ACPYPE***(42)***, or ATB***(43)***, and provide the necessary files (topology and parameters in CNS format) in the `toppar/` directory: `ligand.top` and `ligand.par`. Molecule parameterization is not simple though, and must be approached and carried out with extreme care and expertise (for a 'best-practices' guide for parameterization, although under a different force-field, check the following reference: ***(44)***).

12. If a particular stage of HADDOCK fails, in case of a model not being generated, the run being stopped accidentally, etc., re-issuing the command `haddock2.2` might not be enough. HADDOCK makes use of the output files to control the flow of the docking run. When a particular step is initiated, HADDOCK write a `.job` file and a `.out` file, and when it is completed, the .out file is compressed to a `.out.gz` file. To safely restart a HADDOCK run, remove all `.out` files prior to the issuing of the `haddock2.2` command.

13. A typical error at this stage is that a couple of models in `it1` are not successfully generated. Often, this can be solved by changing the random seed in `run.cns` (`iniseed`, by default 917) and restart HADDOCK (see **Note 12**). Otherwise, try to decrease the `timestep` (e.g. 0.001 instead of 0.002) and/or the temperature of the first two SA stages (e.g. 1000 or 500K instead of 2000). HADDOCK, by default, tries this automatically in case a model fails. If none of this works, simply copy the missing models from the `it0` directory so that the run can proceed. This can be done using the `copy-missing.csh` script provided in the `tools` directory with as arguments the file root name and the number of the missing model.

14. If only a small fraction of the models do fall into clusters, try decreasing the cut-off in case of FCC clustering, or increasing it in case of RMSD clustering. If all models fall in one single cluster, and the restraints are not that restrictive, try the reverse. This is particularly relevant for protein-small molecule docking, for which a tailored FCC clustering algorithm using small molecule atoms – protein residue contacts is available. For the RMSD clustering, due to the nature of interface-ligand RMSD, the resulting RMSD values are larger than would be obtained by fitting on all chains of the complex, which explains the large cut-off value that is used by default (7.5 Å).

15. It is better to use matching number of models (e.g. 4) to compare the cluster statistics in order to remove cluster size effects. In our experience,

the size of a cluster does not always correlate with its quality / score and as such cannot be used as an indicator of the quality of the cluster.

16. The choice of the hinge region(s) where to ‘cut’ the molecules should be made with the structural integrity of the molecule in mind. As such, hinges are favoured if located at the end of α-helices and β-strands, or in loops. The experimental temperature factors should also be taken into account when deciding between possible hinges. The molecule should then be cut at the first peptide bond following the predicted hinge region.

17. The peptide conformations can be generated using PyMOL and setting the φ/ψ dihedral angles according to the desired secondary structure: -57° and -47° for α-helix, -139° and -135° for β-strand, and -78° and -149° for polyproline II.

References

1. A.S. Melquiond, E. Karaca, P.L. Kastritis, et al. (2012) Next challenges in protein–protein docking: from proteome to interactome and beyond, *Wiley Interdisciplinary Reviews: Computational Molecular Science*. **2**:642–651.
2. P.L. Kastritis and A.M. Bonvin (2013) Molecular origins of binding affinity: seeking the Archimedean point, *Current Opinion in Structural Biology*. Advanced Online Publication.
3. T. Schlick, R. Collepardo-Guevara, L.A. Halvorsen, et al. (2011) Biomolecular modeling and simulation: a field coming of age, *Quarterly Reviews of Biophysics*. **44**:191–228.
4. J. Janin (2013) The Targets of CAPRI Rounds 20-27, *Proteins: Structure, Function, and Bioinformatics*. Advanced Online Publication.
5. M.F. Lensink and J. Janin (2013) Docking, Scoring and Affinity Prediction in CAPRI, *Proteins: Structure, Function, and Bioinformatics.* Advanced Online Publication.
6. S.J. de Vries, A.S.J. Melquiond, P.L. Kastritis, et al. (2010) Strengths and weaknesses of data-driven docking in critical assessment of prediction of interactions, *Proteins: Structure, Function, and Bioinformatics.* **78**:3242–3249.
7. C. Dominguez, R. Boelens, and A.M.J.J. Bonvin (2003) HADDOCK: a protein-protein docking approach based on biochemical or biophysical information, *Journal of the American Chemical Society.* **125**:1731–1737.
8. J.P. Linge, M. Habeck, W. Rieping, et al. (2003) ARIA: automated NOE assignment and NMR structure calculation, *Bioinformatics (Oxford, England).* **19**:315–316.
9. A.T. Brünger, P.D. Adams, G.M. Clore, et al. (1998) Crystallography & NMR system: A new software suite for macromolecular structure determination, *Acta Crystallographica Section D: Biological Crystallography.* **54**:905–921.
10. A.T. Brunger (2007) Version 1.2 of the Crystallography and NMR system, *Nature Protocols.* **2**:2728–2733.
11. S.J. de Vries and A.M.J.J. Bonvin (2008) How proteins get in touch: interface prediction in the study of biomolecular complexes, *Current Protein and Peptide Science.* **9**:394–406.
12. E. Karaca and A.M.J.J. Bonvin (2013) Advances in integrative modeling of biomolecular complexes, *Methods (San Diego, Calif)*. **59**:372–381.
13. C. Schmitz, A.S. Melquiond, S.J. de Vries, et al. (2012) Protein-Protein Docking with HADDOCK, *NMR of Biomolecules: Towards Mechanistic Systems Biology (1st ed), Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim.* 521–535.
14. G. Wang, J.M. Louis, M. Sondej, et al. (2000) Solution structure of the phosphoryl transfer complex between the signal transducing proteins

HPr and IIA(glucose) of the Escherichia coli phosphoenolpyruvate:sugar phosphotransferase system, *EMBO J.* **19**:5635–5649.

15. I. Bertini, V. Calderone, L. Cerofolini, et al. (2012) The catalytic domain of MMP-1 studied through tagged lanthanides, *FEBS letters.* **586**:557–567.
16. A. Bax (2003) Weak alignment offers new NMR opportunities to study protein structure and dynamics, *Protein science : a publication of the Protein Society.* **12**:1–16.
17. J.H. Prestegard, C.M. Bougault, and A.I. Kishore (2004) Residual dipolar couplings in structure determination of biomolecules, *Chemical Reviews.* **104**:3519–3540.
18. N. Tjandra (1997) Direct Measurement of Distances and Angles in Biomolecules by NMR in a Dilute Liquid Crystalline Medium, *Science (New York, NY).* **278**:1111–1114.
19. A.D.J. van Dijk, D. Fushman, and A.M.J.J. Bonvin (2005) Various strategies of using residual dipolar couplings in NMR-driven protein docking: Application to Lys48-linked di-ubiquitin and validation against 15N-relaxation data, *Proteins: Structure, Function, and Bioinformatics.* **60**:367–381.
20. N. Kalisman, C.M. Adams, and M. Levitt (2012) Subunit order of eukaryotic TRiC/CCT chaperonin by cross-linking, mass spectrometry, and combinatorial homology modeling, *Proceedings of the National Academy of Sciences of the United States of America.* **109**:2884–2889.
21. U.B. Choi, P. Strop, M. Vrljic, et al. (2010) Single-molecule FRET-derived model of the synaptotagmin 1-SNARE fusion complex, *Nature Publishing Group.* **17**:318–324.
22. A.T. Brunger, P. Strop, M. Vrljic, et al. (2011) Three-dimensional molecular modeling with single molecule FRET, *Journal Of Structural Biology.* **173**:497–505.
23. E. Karaca and A.M.J.J. Bonvin (2013) On the usefulness of ion-mobility mass spectrometry and SAXS data in scoring docking decoys, *Acta Cryst (2013).* **D69**, **683-694**:1–12.
24. S.J. de Vries and A.M.J.J. Bonvin (2011) CPORT: a consensus interface predictor and its performance in prediction-driven docking with HADDOCK, *PloS one.* **6**:e17695.
25. M. Weigt, R.A. White, H. Szurmant, et al. (2009) Identification of direct residue contacts in protein-protein interaction by message passing, *Proceedings of the National Academy of Sciences of the United States of America.* **106**:67–72.
26. D.S. Marks, T.A. Hopf, and C. Sander (2012) Protein structure prediction from sequence variation, *Nature biotechnology.* **30**:1072–1080.
27. J. Fernández-Recio, M. Totrov, and R. Abagyan (2004) Identification of protein-protein interaction sites from docking energy landscapes,

Journal of Molecular Biology. **335**:843–865.

28. J.P.G.L.M. Rodrigues, M. Trellet, C. Schmitz, et al. (2012) Clustering biomolecular complexes by residue contacts similarity, *Proteins: Structure, Function, and Bioinformatics.* **80**:1810–1817.
29. X. Daura, K. Gademann, B. Jaun, et al. (1999) Peptide folding: when simulation meets experiment, *Angewandte Chemie International Edition.* **38**:236–240.
30. M. Nilges and S.I. O'Donoghue (1998) Ambiguous NOEs and automated NOE assignment, *Progress in nuclear magnetic resonance spectroscopy.* **32**:107–139.
31. E. Karaca, A.S.J. Melquiond, S.J. de Vries, et al. (2010) Building macromolecular assemblies by information-driven docking: introducing the HADDOCK multibody docking server, *Molecular & Cellular Proteomics.* **9**:1784–1794.
32. M. Nilges (1993) A calculation strategy for the structure determination of symmetric dimers by 1H NMR, *Proteins: Structure, Function, and Bioinformatics.* **17**:297–309.
33. C. Schmitz and A.M.J.J. Bonvin (2011) Protein-protein HADDocking using exclusively pseudocontact shifts, *Journal of Biomolecular NMR.* **50**:263–266.
34. B. Lee and F.M. Richards (1971) The interpretation of protein structures: estimation of static accessibility, *Journal of Molecular Biology.* **55**:379–400.
35. E. Karaca and A.M.J.J. Bonvin (2011) A multidomain flexible docking approach to deal with large conformational changes in the modeling of biomolecular complexes, *Structure (London, England : 1993).* **19**:555–565.
36. A.D.J. van Dijk and A.M.J.J. Bonvin (2006) Solvated docking: introducing water into the modelling of biomolecular complexes, *Bioinformatics (Oxford, England).* **22**:2340–2347.
37. P.L. Kastritis, A.D.J. van Dijk, and A.M.J.J. Bonvin (2012) Explicit treatment of water molecules in data-driven protein-protein docking: the solvated HADDOCKing approach, *Methods in molecular biology (Clifton, NJ).* **819**:355–374.
38. P.L. Kastritis, K.M. Visscher, A.D.J. van Dijk, et al. (2013) Solvated protein-protein docking using Kyte-Doolittle-based water preferences, *Proteins: Structure, Function, and Bioinformatics.* **81**:510–518.
39. M. van Dijk, K.M. Visscher, P.L. Kastritis, et al. (2013) Solvated protein-DNA docking using HADDOCK, *Journal of Biomolecular NMR.* **56**:51–63.
40. M. Krzeminski, K. Loth, R. Boelens, et al. (2010) SAMPLEX: automatic mapping of perturbed and unperturbed regions of proteins and complexes, *BMC bioinformatics.* **11**:51.
41. A.W. Schüttelkopf and D.M.F. van Aalten (2004) PRODRG: a tool for

high-throughput crystallography of protein-ligand complexes, *Acta Crystallographica Section D: Biological Crystallography.* **60**:1355–1363.

42. A.W. Sousa da Silva and W.F. Vranken (2012) ACPYPE - AnteChamber PYthon Parser interfacE, *BMC research notes.* **5**:367.

43. A.K. Malde, L. Zuo, M. Breeze, et al. (2011) An automated force field topology builder (ATB) and repository: version 1.0, *Journal of Chemical Theory and Computation.* **7**:4026–4037.

44. J.A. Lemkul, W.J. Allen, and D.R. Bevan (2010) Practical considerations for building GROMOS-compatible small-molecule topologies, *Journal of chemical information and modeling.* **50**:2221–2235.