

The HADDOCK web server for data-driven biomolecular docking

Sjoerd J de Vries, Marc van Dijk & Alexandre M J J Bonvin

Bijvoet Center for Biomolecular Research, Science Faculty, Utrecht University, Utrecht, The Netherlands. Correspondence should be addressed to A.M.J.J.B. (a.m.j.j.bonvin@uu.nl).

Published online 15 April 2010; doi:10.1038/nprot.2010.32

Computational docking is the prediction or modeling of the three-dimensional structure of a biomolecular complex, starting from the structures of the individual molecules in their free, unbound form. HADDOCK is a popular docking program that takes a data-driven approach to docking, with support for a wide range of experimental data. Here we present the HADDOCK web server protocol, facilitating the modeling of biomolecular complexes for a wide community. The main web interface is user-friendly, requiring only the structures of the individual components and a list of interacting residues as input. Additional web interfaces allow the more advanced user to exploit the full range of experimental data supported by HADDOCK and to customize the docking process. The HADDOCK server has access to the resources of a dedicated cluster and of the e-NMR GRID infrastructure. Therefore, a typical docking run takes only a few minutes to prepare and a few hours to complete.

INTRODUCTION

Biomolecular complexes are the molecular machines of the cell. To fully understand how the various units work together to fulfill their tasks, structural knowledge at an atomic level is required. Classical structural methods such as nuclear magnetic resonance (NMR) and X-ray crystallography provide this knowledge, but often encounter difficulties when it comes to complexes. Nevertheless, even in cases in which structure determination is difficult, valuable information about complexes can still be obtained from a variety of both experimental and predictive approaches. By combining this information with computational approaches such as docking, useful insights can be obtained on biomolecular interactions¹.

Docking

Docking is the modeling or prediction of the structure of a complex based on its known constituents. During recent years, many docking methods have been developed, including advanced methods that take into account conformational change on binding (for a review, see ref. 2). To assess the performance of docking methods in predicting the structures of protein complexes, the Critical Assessment of Predicted Interactions (CAPRI) experiment has been set up (<http://www.ebi.ac.uk/msd-srv/capri>). CAPRI has proven to act as an important catalyst in the development of biomolecular docking approaches, with the result that accurate predictions for complexes that were considered beyond the limits of docking methodology a few years ago can nowadays be obtained^{3,4}. Currently, the main difficulty in docking is the fact that the conformation of the starting structure (unbound structure, homology model) may be a poor approximation of the conformation in the bound form. The incorporation of structural flexibility into docking algorithms in order to deal with this problem remains an open challenge².

In docking, there are two main strategies that can be followed, namely, *ab initio* docking or data-driven docking. *Ab initio* docking only considers the coordinates of the starting structures, disregarding any experimental knowledge about the system; however, experimental information is often used to filter the docking results. In contrast, in data-driven docking, experimental or predicted information is driving the docking process directly. Among

all the docking methods participating in CAPRI, only HADDOCK⁵⁻⁷ follows a true data-driven strategy.

HADDOCK

HADDOCK is a docking method driven by (experimental) knowledge, in the form of information about the interface region between the molecular components and/or their relative orientations. This information can be derived, e.g., from mutagenesis, mass spectrometry or a variety of NMR experiments (chemical shift perturbation (CSP), residual dipolar couplings (RDCs) or hydrogen/deuterium exchange, or, if available, classical NMR distance restraints (NOEs) can be included as well). When experimental information is sparse or absent, bioinformatic interface predictions can also be used^{8,9}. The 2.0 version of HADDOCK⁶ supports nucleic acids¹⁰ and small molecules; it can deal with a wide range of experimental data⁷ and provides improved docking protocols. HADDOCK has been applied to a variety of problems including protein–protein, protein–nucleic acid, protein–oligosaccharide and protein–small molecule complexes. Unlike many other docking programs, HADDOCK allows for conformational change of the molecules during complex formation, not only of the side chains but also of the backbone. In addition, HADDOCK directly supports the docking of NMR structures and other Protein Data Bank (PDB) structures containing multiple models.

The coordinates of more than 60 biomolecular complexes solved using HADDOCK have been deposited in the PDB¹¹. HADDOCK has also performed very well in the CAPRI blind docking experiment (11 out of 15 complexes achieved the root mean square deviation (RMSD) < 2 Å criterion for a CAPRI two-star evaluation)^{6,7} and is currently the most cited biomolecular docking program¹². The software has been licensed to over 650 laboratories around the world.

The HADDOCK docking protocol

In HADDOCK, experimental data are entered in the form of active and passive residues (**Box 1**). These are converted by HADDOCK into Ambiguous Interaction Restraints (**Box 2**) used to drive the

BOX 1 | ACTIVE RESIDUES AND PASSIVE RESIDUES

Before docking, one must supply, for every molecule, a list of active residues (residues that are known to make contact within the complex) and a list of passive residues (residues that potentially make contact). Note: the Expert and Guru interfaces also allow docking without specifying any active and passive residues (see **Box 2**). On the basis of these residues, HADDOCK generates a set of ambiguous interaction restraints (**Box 2**). In principle, every residue could be made passive, but in practice, they can be limited to residues that are around active residues and on the protein surface, saving computation time. One has the option to automatically define the passive residues from the user-specified active residues, taking all residues that are both on the surface (relative surface accessibility of either main chain or side chain is above 15%, as determined by NACCESS³⁷) and within a radius of 6.5 Å of any active residue. This radius can be changed in the Expert and Guru interfaces.

Active residues are usually residues for which there is experimental evidence that they are in the interface. In principle, any active residue that, during the docking, is not contacting any active or passive residue on the partner molecule gives an energy penalty.

! CAUTION It is an error to designate all residues of a protein as active. It is also important to limit active residues only to those residues that can physically make contact, i.e., those that are on the protein surface.

In most cases, the conversion of experimental restraints into active residues is straightforward. For example, in the case of mutagenesis data, those residues should be selected whose mutation leads to an abolishment or decrease of binding; in the case of chemical shift perturbation (CSP) data, those residues should be selected that give a significant change in chemical shift when the protein is titrated with its binding partner. It remains the user's responsibility to make a proper interpretation of the data. For example, in the case of mutagenesis data, one must check that the mutant still folds properly, and in the case of CSP data, one must check that the changes in chemical shifts are not due to some possibly remote conformational change in the protein. In all cases, only those residues should be made active that are on the surface of the protein and hence able to make physical contacts with the partner molecule.

Most of the experimental data used in HADDOCK can be noisy, causing some residues to be defined as active, whereas in reality, they are not making any contact with the partner molecule (false positives). To deal with noisy data, in HADDOCK, it is possible to discard at random a fraction of the restraints (there is a separate restraint for every active residue). In fact, this option is enabled by default, with 50% of the restraints discarded. In HADDOCK, the docking process is repeated several thousands of times, starting from different initial random orientations, and also with a different set of restraints being discarded. This means that some of the docking solutions, by chance, will be driven only by restraints that are correct, leading to correct docking solutions. HADDOCK relies then on the fact that these correct solutions should have better scores to discriminate them from solutions driven by wrong restraints.

The Expert and Guru interfaces give more control over how active and passive residues are treated, allowing to modify or even disable the random removal of restraints. The percentage of restraints that is discarded is equal to $100/n$, where n is the number of restraint partitions. By default, the number of partitions is two, meaning that $100/2 = 50\%$ of the restraints are discarded. With the HADDOCK server, the number of partitions does not need to be an integer, so that any percentage of restraints can be discarded. Note: in the Expert and Guru interfaces, restraint files can also be uploaded instead of being specified in the form of active and passive residues. In that case, the number of partitions must be integer.

docking. HADDOCK automatically generates the topology of the molecules that are to be docked. The docking protocol then consists of three stages, namely, a rigid-body energy minimization, a semi-flexible refinement in torsion angle space and a final refinement in explicit solvent. After each of these stages, structures are scored and ranked, and the best structures are kept for the next stage. The HADDOCK score is a weighted sum of van der Waals, electrostatic, desolvation and restraint violation energies together with buried surface area. For every stage, the number of structures and the scoring weights can be modified by the user, as well as several parameters controlling the docking/refinement protocol such as temperature, number of time steps and force constants. HADDOCK makes use of CNS (Crystallographic and NMR system)¹³ as its structure calculation engine. In addition, HADDOCK allows the user to define the protonation states of histidines and several forms of symmetry restraints. During the torsion angle refinement, to save computation time, flexibility is only taken into account for so called semi-flexible segments and during part of the refinement. By default, these semi-flexible segments are determined automatically, by identifying the parts of the molecule that are involved in complex formation. It is possible to define manually the semi-flexible segments and also fully flexible segments that are flexible throughout the refinement. Finally, the HADDOCK program supports the simultaneous

docking of up to six molecules, as any combination of proteins, nucleic acids, peptides and other biomolecules.

The HADDOCK server

When run on a local machine, the main limitations of HADDOCK are that it requires a Unix environment with CNS¹³ and various other programs installed, and that the user has some technical expertise with these programs. In addition, a typical HADDOCK run can take hundreds of CPU hours, requiring a cluster of computers to run in reasonable time. These requirements might not be a problem for NMR structural biologists, who are used to perform structure calculations with CNS or Xplor-NIH¹⁴ and form a good share of the HADDOCK user base. However, they may present a barrier to the use of data-driven docking by a wider community. To overcome this barrier, a web server interface to HADDOCK is presented here. This web server interface implements the full set of parameters supported by the HADDOCK program and provides the same amount of control over the docking process as running HADDOCK on a local machine. However, owing to the validation and automation provided by the server, the protocol of setting up a docking run is greatly simplified. **Figure 1** shows a flowchart of data-driven docking using the HADDOCK server.

Here we will describe the different web interfaces and the various steps and parameters of the HADDOCK server, explaining them in

BOX 2 | AMBIGUOUS INTERACTION RESTRAINTS

In HADDOCK, experimental information is incorporated in the form of Ambiguous Interaction Restraints (AIRs). This drives the docking in a similar manner as ambiguous distance restraints based on NOE data drive the structural calculation of an NMR structure (see e.g., Linge *et al.*³⁸). AIRs are defined by HADDOCK based on the active and passive residues defined by the user (**Box 1**). For every active residue, a single AIR restraint is defined between that residue and all active and passive residues on the partner molecule. An explicit AIR energy term is introduced into the calculation through a soft-square harmonic potential (E_{AIR}) that depends on an effective distance. The latter is calculated through the following formula:

$$d_{iAB}^{\text{eff}} = \left(\frac{N_{\text{atoms}} N_{\text{res}} N_{\text{atoms}}}{\sum_{m_{iA}=1} \sum_{k=1} \sum_{n_{kB}=1} d_{m_{iA}n_{kB}}^6} \right)^{-\frac{1}{6}}$$

where *A* and *B* are molecules, *i* iterates over all distance restraints, N_{atoms} indicates all atoms of a given residue and N_{res} the sum of active and passive residues for a given protein. This means that the effective distance for an AIR between two single atoms would be equal to the actual distance; however, as the number of atoms grows, so does the number of distances taken into consideration, causing the effective distance to become shorter and shorter.

An upper limit to the effective distance (typically 2 Å) is enforced by HADDOCK. If this limit is exceeded, the AIR energy becomes positive and the active residue experiences an attractive force toward the active and passive residues of the partner molecule. If not, the restraint is satisfied and the AIR energy and attractive force are zero for that restraint. The exact relation between effective distance and energy is described in a paper by Nilges *et al.*³⁹. As many atom–atom distances inversely contribute to the effective distance, an AIR restraint is typically satisfied, depending on the level of ambiguity, if a residue comes within 3–4 Å of any active or passive residue of the partner molecule. In this way, (putative) interface residues are forced to make contact with (a surface region on) the partner protein, but not with any specific partner residue.

Normally, restraints are generated automatically by the HADDOCK server, based on the active and passive residues. The Expert and Guru interfaces also allow one to upload custom restraints files (in CNS format (see **Box 4**)). These custom restraints can be intermolecular, but also intramolecular, e.g., to preserve secondary structure or the correct coordination of an ion or cofactor.

! CAUTION The HADDOCK server checks the uploaded restraints for syntax errors, but it is still one's responsibility to make sure that the restrained atoms and residues are actually present in the structure.

Restraints defined from active and passive residues are removed at random from a fraction of the docking trials (see **Box 1**). However, one can exclude uploaded restraints from this removal process by uploading them under 'unambiguous restraints'.

Finally, in the Expert and Guru interfaces, one can dock without any restraints at all (*ab initio*), relying on the HADDOCK scoring function to discriminate between right and wrong solutions. The various HADDOCK *ab initio* docking options are described in Step 9 of the protocol. The number of rigid-body structures should be increased to at least 5,000 or even more if one is docking without restraints.

relation to the docking process. Common error messages returned by the server are discussed in the TROUBLESHOOTING section. In our experience, the accuracy of docking is influenced not only by the amount of information but also by the quality of the starting structure and the strength of the interaction. The researcher would need to judge the reliability of the docking result and to relate it with other experimental data.

Considering the already large HADDOCK user base, we expect the HADDOCK server to become a useful tool in the modeling of biomolecular complexes. The server has been available since June 2008. As of December 2009, 534 users have registered and the server has processed more than 3450 docking runs, excluding users and runs from our own lab (over 5650 runs in total including those carried out by our lab). The HADDOCK server has participated in CAPRI rounds 15–19 (targets 32–42), making an acceptable (interface RMSD < 4 Å and fraction of native contacts > 10%) prediction for a protein–RNA complex and a protein–protein complex (target 34, unpublished structure; <http://www.ebi.ac.uk/msd-srv/capri/round15/round15.html>; target 42, unpublished structure; <http://www.ebi.ac.uk/msd-srv/capri/round19/round19.html>), a two-star prediction (interface RMSD < 2 Å and fraction of native contacts > 30%) for another protein complex (target 41, unpublished structure; <http://www.ebi.ac.uk/msd-srv/capri/round19/round19.html>) and a three-star prediction (interface RMSD < 1 Å and fraction of native contacts > 50%) for another protein–protein complex (target 40, PDB code 3E8L; <http://www.ebi.ac.uk/msd-srv/capri/round18/round18.html>).

The web server portal provides the user with four different web interfaces:

The HADDOCK server web interfaces

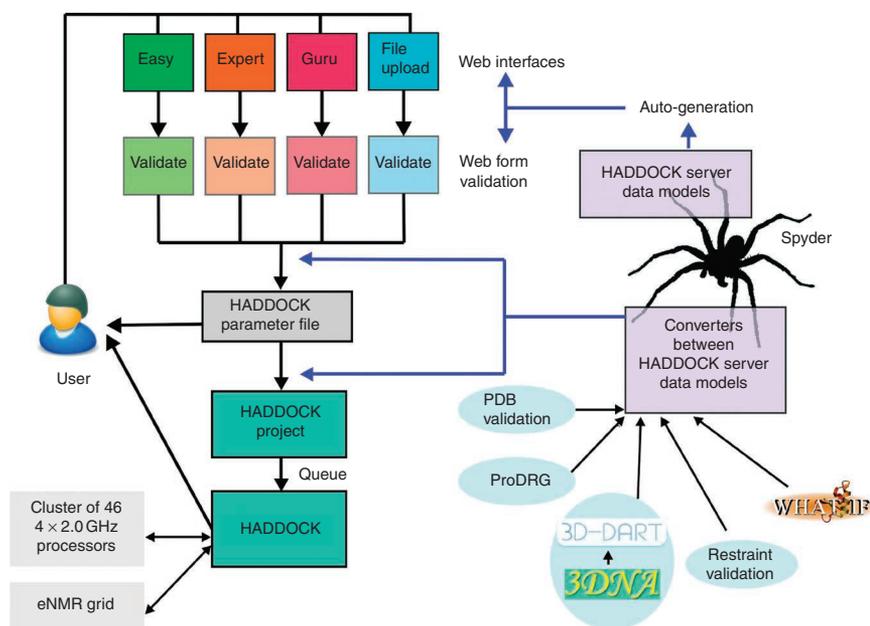
The web server portal provides the user with four different web interfaces:

- The *Easy interface* requires only starting structures and lists of interacting residues. This interface hides the majority of the parameters, implicitly setting them at their default values or automatically determining them when needed. This is sufficient for the most common applications of HADDOCK, such as docking driven by CSPs, mutagenesis data or bioinformatic interface predictions⁸.
- The *Expert interface* allows the more advanced user to upload custom distance, hydrogen bond and dihedral angle restraint files, and gives control over additional parameters such as the number of structures to be calculated, the protonation state of histidines and the manual definition of flexible segments.
- The *Guru interface* gives full control over all aspects of HADDOCK, including force constants for the various restraint energy

PROTOCOL

Figure 1 | Flowchart of the HADDOCK server.

Several processing and validation routines are plugged into the HADDOCK server framework. Uploaded Protein Data Bank data are checked for correct formatting. If a structure contains unconnected multiple bodies (e.g., a dimeric receptor), restraints are automatically defined to prevent the various bodies to drift away during the docking. In case of proteins, chemical groups other than standard amino acids are detected. For a few of those groups, CNS¹³ parameters are already present in HADDOCK, but for all others, the PRODRG server³⁵ is contacted automatically to retrieve CNS parameters. In addition, if they have not been supplied by the user, histidine protonation states are automatically defined by querying the WHATIF server²⁵. In case of nucleic acids, modules from the 3D-DART framework⁴⁰ are used to parse and validate the supplied structures. Through its bindings to X3DNA^{41,42}, the framework extracts Watson–Crick base pairing from the starting structure and automatically defines restraints to preserve the helical shape of the nucleic acid during the docking. Once the data have been validated, the run is written on the server disk as a HADDOCK project directory and added to a queue. The user is kept informed of the current status of his/her run by email and a link to a unique web page is given in which the run can be monitored and the final results are presented. After the docking, the structures are clustered and the resulting clusters ranked according to the HADDOCK score. A web page containing cluster statistics is generated, from which the user can download the top four members of each clusters or preview them using Jmol³². Optionally, the entire HADDOCK run can be downloaded for local manual analysis by the user.



terms, time steps and temperatures for the simulated annealing refinement stage and weights of the various terms entering the scoring function; it further offers support for symmetry, RDCs and anisotropy restraints.

- Finally, the parameter file upload interface allows a one-step launch of the docking run, providing a simple way of building pipelines from other applications. The generation of such parameter files has already been implemented into the CCPN Analysis NMR software suite¹⁵ and the MTMDAT program for the analysis of mass spectrometry data resulting from partial digestion of complexes¹⁶.

Figure 2 shows the HADDOCK server Easy interface. **Table 1** shows a comprehensive overview of the parameters used in HADDOCK that are accessible from the various interfaces. An example web form with filled-in data is available at <http://haddock.chem.uu.nl/services/HADDOCK/haddockserver-demo.html>.

Comparison with existing docking methods

A number of other docking servers have already been developed, such as ClusPro¹⁷, GRAMM-X¹⁸, PatchDock/FireDock^{19–21}, Hex^{22,23} and RosettaDock²⁴. All of these docking servers implement *ab initio* docking methods. Among those, only PatchDock/FireDock and the RosettaDock server can deal with structural flexibility but only at the side chain level. Several servers can deal with nucleic acids, but none of these allow for any structural flexibility. In contrast, HADDOCK allows for full structural flexibility of both side chains and backbone, for proteins, nucleic acids and any other biomolecules.

If HADDOCK is to be used for predictive modeling, the results should be compared with those of *ab initio* modeling, as described in the TROUBLESHOOTING section.

Limitations of the HADDOCK server

The HADDOCK server supports the most common modified amino acids (such as phosphoSer/Thr/Tyr and acetylated Met) but does not, at present, allow users to provide their own parameter files for modified residues or bases. This is because HADDOCK uses the topallhdg format convention for CNS amino acid parameters and this convention must be followed strictly to be able, e.g., to build peptide bonds. A solution planned for the next version of HADDOCK is to give users the option to upload a full CNS .psf file containing all topologies and parameters of the entire molecule.

Time

A HADDOCK docking run typically takes a few hundreds of CPU hours. This computational power is provided by a dedicated cluster. In addition, the HADDOCK server can also make use of additional GRID computing resources that have been deployed across Europe within the FP7 e-NMR e-Infrastructure European project (<http://www.enmr.eu>).

Availability

The HADDOCK web server is available at <http://haddock.chem.uu.nl/Haddock>. It is free for nonprofit users, but, because of the large amount of CPU time involved and owing to CNS licensing issues, it requires registration through a web form. Access is initially given to the Easy Interface. Access to the more advanced interfaces is granted on request.

The GRID-enabled version of the HADDOCK web server is available at <http://haddock.chem.uu.nl/enmr>. To register for this version, a valid certificate provided by a national registration authority is required (see <http://www.enmr.eu/eNMR-registration>).

MATERIALS

EQUIPMENT

- Atomic resolution structures or high-quality homology models of the biomolecules that are to be docked, in PDB format
- A computer with Internet access and a web browser
- Experimental or predicted data, giving information about the residues in the interface between the biomolecules and/or their relative orientations

PROCEDURE

Register at the HADDOCK server website

1| This step can be performed using option A or B, depending on whether or not one wants to use the e-NMR GRID for docking calculations. The GRID contains a much larger number of computers than our local cluster, hence, in principle, docking runs can finish faster with the GRID-enabled version of the HADDOCK server, especially if the cluster is busy. However, for registration with the GRID-enabled version, a valid personal certificate must be obtained and loaded in one's web browser.

(A) Normal registration

- (i) The registration form at the HADDOCK server web site (<http://haddock.chem.uu.nl/Haddock>) must be filled. Registration is free, the only requirement is that the application comes from a nonprofit user. The username and password will be sent by email.

(B) GRID-enabled registration

- (i) A personal certificate and register with the e-NMR virtual organization (see <http://www.enmr.eu/eNMR-registration>) must be obtained.
- (ii) The certificate must be loaded in one's web browser.
- (iii) The registration form at the GRID-enabled HADDOCK server web site (<http://haddock.chem.uu.nl/enmr>) must be filled. The username and password will be sent by email.

Determine the appropriate interface

2| One must choose between the Easy, Expert or Guru web interface. The Easy interface allows to specify starting structures and enter the restraints in the form of active and passive residues. This is the most common docking case. If one has specific experimental data, or if one needs more control over the docking process, the Expert or Guru interface must be used. **Table 1** shows an overview of all parameters and the interfaces that support them. Note that access to the Expert or Guru interface can be enabled from one's account, for which an email must be sent to haddocking@gmail.com.

Upload the starting structure

3| Unfold the menu 'First molecule' and specify the starting structure of the biomolecule in PDB format. See **Box 3** for more information on starting structures in HADDOCK.

? TROUBLESHOOTING

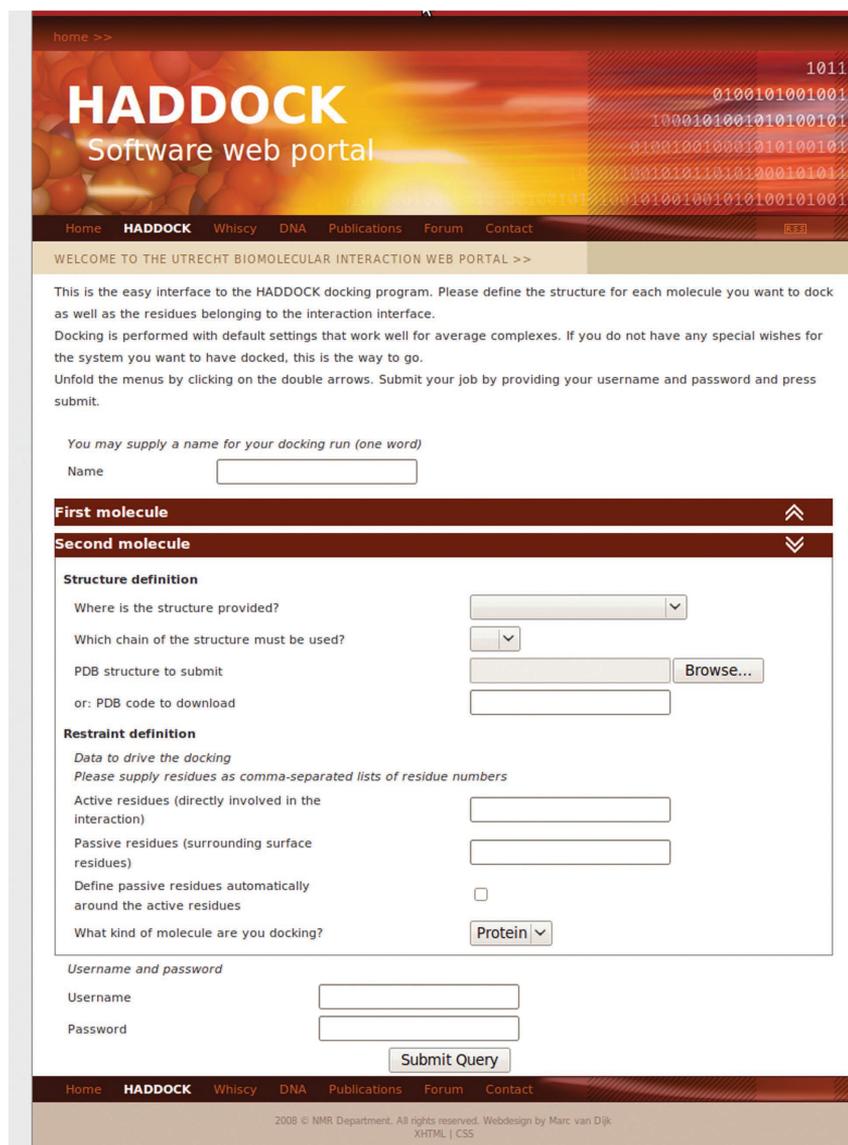


Figure 2 | The HADDOCK server Easy interface. This interface is sufficient for the most common applications of HADDOCK.

TABLE 1 | Data-driven docking parameters supported by the various HADDOCK web interfaces.

| | Easy interface | Expert interface | Guru/upload interface |
|---|------------------------|----------------------------------|----------------------------------|
| Active and passive residue lists | Supported | Supported | Supported |
| Docking from ensemble structures | Supported | Supported | Supported |
| Protein–nucleic acid docking | Supported | Supported | Supported |
| Cofactors and modified amino acids ^a | Supported | Supported | Supported |
| Semi-flexible segment definition | Automatic | Automatic or manual | Automatic or manual |
| Fully flexible segment definition | No | Supported | Supported |
| Histidine protonation states | Automatic ^b | Automatic ^b or manual | Automatic ^b or manual |
| Custom CNS distance restraints | No | Supported | Supported |
| Custom hydrogen bond restraints | No | Supported | Supported |
| Custom dihedral angle restraints | No | Supported | Supported |
| Preservation of nucleic acid base pairing and backbone conformation | Automatic | Automatic or manual | Automatic or manual |
| <i>Ab initio</i> docking | No | Supported | Supported |
| Nonpolar hydrogens | No | Supported | Supported |
| Random removal of restraints | Automatic | Manual | Manual |
| Number of structures to dock and to refine | Automatic | Manual | Manual |
| DMSO refinement | No | Supported | Supported |
| Solvated docking | No | Supported, automatic parameters | Supported, manual parameters |
| Epsilon (electrostatic scaling constant) | Automatic | Manual | Manual |
| Clustering | Automatic | Manual | Manual |
| Symmetry | No | No | Supported |
| Restraints energy constants | Automatic | Automatic | Manual |
| Relaxation anisotropy restraints | No | No | Supported |
| Residual dipolar couplings | No | No | Supported |
| Energy and interaction parameters | Automatic | Automatic | Manual |
| Scoring parameters | Automatic | Automatic | Manual |
| Randomization of starting orientations | Automatic | Automatic | Manual |
| Refinement-only protocol | No | No | Supported |
| Temperature and timesteps of the various refinement stages | Automatic | Automatic | Manual |

^aSmall molecule parameters are automatically retrieved from the PRODRG server²⁵. ^bThe most likely protonation states are automatically retrieved from the WHATIF server²⁵.

Enter active and passive residues

4 | Decide which residues must be in the interface between the molecules, based on the experimental information. Enter these residues in the ‘Active residues’ field as comma-separated numbers. Passive residues are residues that might be in the interface. Passive residues are defined in the same way, but they can be omitted if one enables the ‘Define passive residues automatically around the active residues option’. See **Box 1** for more details.

? TROUBLESHOOTING

Define histidine protonation states

5 | This optional step is only available in the Expert and Guru interfaces. Every histidine in a protein has three possible protonation states: a proton on the delta nitrogen (HISD), a proton on the epsilon nitrogen (HISE) or a proton on both nitrogens (HIS+). The first two protonation states are neutral, whereas the last one carries a positive charge. By default, the WHATIF server²⁵ is queried automatically to obtain the most likely protonation state for each histidine, but this can be overruled. If one knows the protonation state of one or more histidines, it can be defined in the web form. Editing the histidines in the PDB file will not work.

BOX 3 | STARTING STRUCTURES

Provide the starting structures in the Protein Data Bank (PDB) format¹¹. There are a number of requirements that must be followed. All of them (except overall starting structure quality) are validated by the HADDOCK server, which will give an error message if a requirement is not met.

Ensembles of structures: The HADDOCK server can deal directly with NMR ensembles and other PDB files that contain multiple models, such as ensembles derived from molecular dynamics simulations. However, all models must be completely identical in terms of number of atoms, residue numbering and nomenclature.

Nonoverlapping numbering: During the docking process, every molecule is reduced to a single chain. Therefore, in case of a multichain molecule, one can select a chain to be used or select all the chains. However, in the latter case, there must be no overlap in numbering between the chains. For example, a molecule with chain A residue 1–100 and chain B residue 101–200 is acceptable; a molecule with residue 1–100 for both chains A and B is not. If this is the case, one must renumber the PDB structure.

Numerical residue identifiers: All residue numbers must be numerical; values such as '69A' are not supported. If they are present in the PDB, one must renumber the PDB structure.

Elemental ions: Elemental ions can be included in HADDOCK, but their proper charge must be specified in both the residue name and the atom name. For example, for Fe^{3+} , the atom name (PDB columns 13–16) must be FE+3, whereas the residue name (PDB columns 18–20) must be FE3.

Cofactors and modified amino acids: Cofactors and modified amino acids can be directly used in HADDOCK. Modified amino acids must be specified in the PDB as ATOM lines. They must be present in the HADDOCK library of modified amino acids, listed at <http://haddock.chem.uu.nl/services/HADDOCK/library.html>. Selenomethionine residues (MSEs) are automatically converted to methionine. If the protein contains an unsupported modified amino acid, the docking run cannot proceed. One could alternatively define it as 'cofactor' with HETATM lines in the PDB file; in that case, topology and parameters will be queried automatically from the PRODRG server³⁵. The modified amino acid will however be unconnected to the remaining of the protein and might drift away during docking. To prevent this problem, one can define distance restraints that mimic a peptide bond between the modified amino acid and the rest of the protein.

Cofactors must be specified in the PDB as HETATM lines. For cofactors, force field parameters (bonds, angles, electric charges and nonbonded parameters) are obtained automatically from the PRODRG server³⁵. Note that PRODRG only supports N, C, O, S, P, Cl, I, Br and F atoms. However, the PRODRG server will not be contacted if they are present in the HADDOCK library (currently, however, no cofactor parameters are contained in the library). Therefore, if one has better cofactor parameters that one would like to use during the docking, one might contact us so that they can be added to the library. These parameters should however be properly validated and should be provided in the correct CNS format.

Nucleic acids: In case of nucleic acids, modules from the 3D-DART framework⁴⁰ are used to parse and validate the supplied structures. Through its bindings to X3DNA^{41,42}, the framework extracts Watson–Crick base pairing from the starting structure and automatically defines restraints to preserve the helical shape of the nucleic acid during the docking.

Starting structures for double stranded DNA can be easily generated using the 3D-DART web server developed by our group⁴⁰ (<http://haddock.chem.uu.nl/dna/dna.php>). Note: modified nucleic acids and cofactors bound to nucleic acids are currently not supported.

Starting structure quality: One must make sure that the input structures are of sufficiently high quality. A PDB may contain some missing atoms, which are built by CNS before docking. Clashes and improbable conformations are also tolerated to some extent because a short energy minimization is performed before docking. However, too many of these errors will cause the topology generation step to fail. Therefore, if one wants to use homology models with HADDOCK, one must make sure that they are carefully refined.

A high quality of the structure also means that it must be a fairly good approximation of the conformation of the biomolecule (protein, nucleic acid and small molecule) as it occurs in the actual complex. There are two cases in which this requirement may be violated: first, a poor homology model with a large deviation from the true, unbound structure; second, the biomolecule undergoes a large conformational change on complexation.

! CAUTION The accuracy of the energies and scores used in HADDOCK is highly dependent on the quality of the structure. If the starting structure is a poor approximation of the molecule's conformation in the bound form, the docking may finish without error, but the results will not be meaningful. There is no hard limit for the amount of deviation that can be tolerated, i.e., the maximum amount is dependent on the system and on the quality of the restraints. However, in general, backbone RMSD values of up to 1 Å are easily dealt with, whereas deviations of 2 Å or more are usually problematic. It is therefore crucial that one has at least some estimate of the conformational change that is taking place upon complex formation.

Define flexible segments

6| This optional step is only available in the Expert and Guru interfaces. HADDOCK allows the definition of semi-flexible segments, which are made flexible during the later stages of the simulated annealing stage (it1). Semi-flexible segments are also the segments among which random patches can be defined (see Step 9 for more details). By default, HADDOCK defines the semi-flexible segments automatically and separately for each structure, based on an analysis of the intermolecular contacts after the rigid-body stage (it0). HADDOCK also allows fully flexible segments: these are made flexible throughout the entire simulated annealing stage. By default, there are no fully flexible segments.

If one wants to manually define specific semi-flexible and/or fully flexible segments, define them here.

7| Repeat Steps 3–6 for the second molecule. The following steps (Steps 8–14) are only for the Expert/Guru interface. Proceed to ‘Submit your docking run’ for the Easy interface (Step 15).

Customize the restraints

8| (Optional) Unfold the menus ‘Distance restraints’ and ‘Dihedral and hydrogen bond restraints’. Upload custom distance restraints, dihedral restraints or hydrogen bond restraints in CNS format. These restraints can be intermolecular but also intramolecular. The server will validate the syntax of each file. See **Box 2** for extra information. There are two fields, namely, ‘ambiguous restraints’ and ‘unambiguous restraints’, to upload distance restraints; ‘ambiguous restraints’ is for distance restraints that are to be subjected to random removal, such as active residues (see **Box 1** for more details). Distance restraints uploaded as ‘unambiguous restraints’ are not subjected to random removal. Dihedral and hydrogen bond restraints are also not subjected to random removal.

9| (Optional) Unfold the menu ‘Distance restraints’. Choose one or more *ab initio* docking modes to use. There are three modes supported, namely, random restraints, center-of-mass restraints and surface contact restraints. These modes can be combined. Center-of-mass restraints and surface contact restraints can be used in addition to experimental restraints to enforce binding. Random restraints are incompatible with experimental restraints.

(A) Random restraints

- (i) Check this option to define patches of active residues selected randomly among all surface residues in both partners. If semi-flexible segments are defined, random patches are confined to these segments.

(B) Center-of-mass restraints

- (i) Check this option to enforce the centers-of-mass of both molecules to be as close as possible. Change the force constant to give more or less weight to this restraint.

(C) Surface contact restraints

- (i) Check this option to enforce the surfaces of both molecules to make at least one contact. Change the force constant to give more or less weight to this restraint.

▲ **CRITICAL STEP** If one chooses to use the *ab initio* mode, we do recommend the use of center-of-mass restraints as the default. Random restraints are mostly useful in case of antibody–antigens, because patches are selected only from the semi-flexible segments. This allows the selection of fully random patches on the antigen, whereas the random patches on the antibody can be limited to the CDR loops. Surface contact restraints are most useful as a light restraint to prevent drifting away.

10| (Optional) Unfold the menu ‘Distance restraints’. Customize the random removal of restraints. One must enter the number of partitions for restraint removal. The percentage that is removed is 100/(number of partitions). The number of partitions can be noninteger. Random removal of restraints provides a way to deal with noisy experimental data and predictions. See **Box 2** for more details on restraints in HADDOCK.

Customize the sampling parameters

11| (Optional) Unfold the menu ‘Sampling parameters’. Set the number of structures to dock during the rigid-body stage, the flexible annealing stage and water refinement. In addition, specify the solvent to use during water refinement (water or dimethyl sulfoxide, DMSO) and the epsilon constant for the electrostatic forces during the first two stages (a value of 1 corresponds to the value of epsilon in a vacuum). The default settings for these last two parameters are optimized for soluble proteins, i.e., a value of 10 (a compromise between the epsilon values for solvent and protein interior) is used for the implicit solvent parts of the protocol, whereas epsilon is set to 1 for the final explicit solvent refinement.

! **CAUTION** The last two stages each take about 100 times more computation time per structure than the rigid-body stage. Please do not set the number of structures for those stages higher than necessary.

Define symmetry restraints

12| (Optional) Unfold the menus ‘Non-crystallographic symmetry restraints’ and ‘Symmetry restraints’. Enable the use of symmetry. HADDOCK supports noncrystallographic symmetry as well as C2, C3 and C5 symmetry. In all cases, the symmetry can either be intramolecular or intermolecular. Specify the segments that are related by symmetry operations. The server will verify that they contain the same number of residues.

▲ **CRITICAL STEP** Note that for noncrystallographic symmetry restraints, the selections should contain exactly the same number and type of atoms.

Define RDCs and diffusion anisotropy (DANI) restraints

13| (Optional) Unfold the menus 'Residual Dipolar Couplings' and 'Relaxation Anisotropy' and enter the experimental parameters. RDCs can be specified in two different forms, namely, directly (SANI)²⁶ or as intervector angle restraints²⁷.

In the case of VEAN restraints, one must upload an intervector projection angle restraints file. The generation of these restraints is implemented in a slightly modified version of a python script (*dipolar_segid.py*) kindly provided by Dr. Helen Mott and Dr. Wayne Boucher (Cambridge University). This script is distributed with the HADDOCK program.

For SANI and DANI²⁸ restraints, one must specify the axial (D) and rhombic (R) components of the alignment tensor and provide a restraint file defining the restraints with respect to an alignment tensor. For more information on orientational restraints, see **Box 4**, *Residual dipolar coupling and diffusion anisotropy restraint files*.

In addition, for DANI restraints, one must supply nitrogen and proton frequencies.

HADDOCK supports up to five sets of RDC and DANI restraints. For each set of restraints, one can control the weight during each stage of the docking protocol. For more details on the use of RDCs and DANI restraints in HADDOCK, see van Dijk *et al.*^{29,30}.

14| (Optional) Unfold the menu 'Solvated docking'. Enable solvated docking and specify the parameters. Solvated docking provides a way to include water molecules in all stages of the docking process. Refer to the paper of van Dijk *et al.*³¹ for a description of all the parameters that are involved.

Submit the docking run

15| Enter the username and password that are obtained during registration. Press the submit button. The web form will be validated for inconsistencies.

● **TIMING** This should take no more than 1 min. If there are any inconsistencies, an error message will be shown, with instructions on how to fix the error. If this happens, press the Go Back button in the browser, edit the parameters and re-submit the web form.

Monitor the docking process

16| One can see a message that the data have been received and are being processed. A link to the parameter file will be presented. Save this file, as it contains all the data and parameters, and allows to re-submit the docking run at any given time using the File Upload interface.

17| Wait until an email is received with the result of the processing of the data.

● **TIMING** This should take no more than 10 min. If everything goes off successfully, one will receive a message that the docking run has been added to the queue.

? TROUBLESHOOTING

18| Wait until an email is received with the result of the docking.

● **TIMING** A successful docking run takes between 1 h and several days to complete. This depends on the size of the interacting partners, the type and number of restraints and the number of docking runs on the server. Typical execution time is about 2 h if no other docking is running. Docking runs running on the GRID are much less affected by the number of other docking runs. In any case, one will be presented with a link to the result page for the docking run.

? TROUBLESHOOTING

Get the docking results

19| Follow the link to the result page. After a successful docking run, the clustered docking solutions will be displayed here. The clusters are numbered according to cluster size, i.e., cluster 1 is the largest cluster and cluster 2 the second largest. The clusters are sorted by HADDOCK score, i.e., the cluster with the lowest HADDOCK score is on top of the web page and is the most likely solution according to HADDOCK. Only the top 10 clusters are displayed. For every cluster, the various energies that make up the HADDOCK score are displayed. The top four structures of every cluster can be downloaded or viewed in a web browser using Jmol³².

! **CAUTION** The HADDOCK score (given in arbitrary units) cannot be used to predict binding affinities or compare different complexes. It should only be used to compare different solutions for a given complex.

? TROUBLESHOOTING

20| (Optional) The entire docking run, containing structures from all docking stages, is available for download in the form of a zipped archive. This is recommended only for users who are familiar with HADDOCK. If the HADDOCK software has been installed on a local machine, the HADDOCK analysis and clustering steps can then be re-done with user-defined parameters.

▲ **CRITICAL STEP** This step is highly recommended in case of internal symmetry within one of the partner molecules because the clustering in HADDOCK does not take this symmetry into account.

BOX 4 | CUSTOM RESTRAINT FILES

In HADDOCK, restraints are in CNS¹³ format, consisting of a CNS (sub)statement, one or more atom selections and one or more floating-point values. The Expert and Guru interfaces allow the upload of several kinds of restraints files. For more information on CNS syntax in general, see http://cns-online.org/v1.21/syntax_manual/frame.html. For more information on CNS atom selections, see http://cns-online.org/v1.21/tutorial/language/atom_sele_basic/text.html and http://cns-online.org/v1.21/syntax_manual/html/atom-selection.html.

The following custom restraints can be uploaded:

Distance restraint files

This file format is used in the uploading of ambiguous distance restraints, unambiguous distance restraints and hydrogen bond restraints. The format of a distance restraint is: *assign <atom selection> <atom selection> <target distance> <lower distance margin> <higher distance margin>*. For example:

```
assign (resid 2 and segid A)
      (resid 10 and segid B) 2.0 2.0 0.0
```

This indicates that residue 2A and 10B should have an effective distance of 2 Å or less. More information can be found in the HADDOCK paper¹.

Dihedral restraint files

The format of a dihedral restraint is: *assign <atom selection> <atom selection> <atom selection> <atom selection> <force constant> <target dihedral angle> <error range> <exponent>*. For example:

```
assign (segid B and resid 48 and name N) (segid B and resid 48 and name CA)
      (segid B and resid 48 and name CB) (segid B and resid 48 and name CG)
      1.0 180.0 30.0 2
```

Residual dipolar coupling restraint files

Residual dipolar couplings can be defined in two manners in CNS, namely, directly (SANI)²⁶ or as intervector angle restraints²⁷.

In SANI, the residual dipolar coupling restraint is defined between the observed pair of nuclei (e.g., N-H) and an alignment tensor.

The tensor will be automatically added to the coordinate file by HADDOCK. The restraints should refer to the correct tensor atom name, residue name (ANI) and residue number: they must be numbered between 995 and 999 and containing the atoms OO, Z, X and Y.

The format of a residual dipolar coupling restraint is as follows:

```
assign (resid 999 and resname ANI and name OO)
      (resid 999 and resname ANI and name Z)
      (resid 999 and resname ANI and name X)
      (resid 999 and resname ANI and name Y)
      (segid B and resid 2 and name N)
      (segid B and resid 2 and name HN) - 12.1010 0.2000
```

The first four selections contain the alignment tensor, the last two selections are the nitrogen and hydrogen atom and the last two values are the measured residual dipolar coupling (in Hz) and its associated error.

In VEAN, an intervector projection angle restraint is defined between pairs of vectors. The use of a tensor is no longer needed. The restraint should contain four atom selections (defining the two pair of atoms for which RDC have been measured) and four numbers (first angle, range, second angle, range) defining the angular range (angle ± range) in which the intervector projection angle should reside.

```
assign (segid A and resid 10 and name N)
      (segid A and resid 10 and name HN)
      (segid B and resid 2 and name N)
      (segid B and resid 2 and name HN) 10.0 4.3 170.0 4.3
```

These restraints can be generated using the `dipolar_segid.py` script kindly provided by Dr. Helen Mott and Dr. Wayne Boucher (Cambridge University) and distributed with the HADDOCK program (in RDCTools).

Diffusion anisotropy restraint files

Diffusion anisotropy restraints can be used in CNS using the DANI energy term²⁸. The restraint is defined between the T1/T2 ratio for the observed pair of nuclei (e.g., N-H) and an alignment tensor. The tensor will be automatically added to the coordinate file by HADDOCK. The restraints should refer to the correct tensor atom name, residue name (DAN) and residue number: they must be numbered between 995 and 999 and contain the atoms OO, Z, X and Y. The format of a diffusion anisotropy restraint is as follows:

```
assign (resid 999 and resname DAN and name OO)
      (resid 999 and resname DAN and name Z)
      (resid 999 and resname DAN and name X)
      (resid 999 and resname DAN and name Y)
      (segid B and resid 2 and name N)
      (segid B and resid 2 and name HN) 10.069 0.200
```

The first four selections contain the alignment tensor, the last two selections are the nitrogen and hydrogen atoms and the last two values are the measured T1/T2 ratio and its associated error.

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 2** (references 33–35).

TABLE 2 | Troubleshooting table.

| Step | Problem and solution |
|------|---|
| 3 | <p>Problem: Can I use homology models with HADDOCK? Solution: Yes, but we only recommend their use if they are of high quality. See Box 3, <i>Starting Structure Quality</i> for details</p> <p>Problem: Can I dock ensembles of structures? Solution: Yes, this is supported automatically. See Box 3, <i>Ensembles of Structures for more details</i></p> <p>Problem: Can I dock more than two molecules? Solution: The HADDOCK standalone version distributed to users can deal with up to six molecules, but the HADDOCK server currently only supports two. However, a web server interface for more molecules has been developed and is currently in the testing phase</p> <p>Problem: I have rigid body solutions from another docking program. Can I refine them with HADDOCK? Solution: HADDOCK can be used for the refinement of complexes, by skipping the rigid-body docking stage and defining restraints automatically from the initial complex. Currently, this protocol requires access to the Guru interface and the adjustment of several parameters; however, we are planning to develop a separate interface to offer this service in a user-friendly manner to all HADDOCK server users. This interface will also be able to refine single structures</p> |
| 4 | <p>Problem: I have experimental data for only one partner molecule. Can I still use HADDOCK? Solution: Yes, if you have active residues for only the first partner, you can determine all surface residues for the second partner and enter them as passive residues. In that case, you do not need to define passive residues for the first partner. See Box 1 for more details. Alternatively, you can perform bioinformatic interface prediction (for a review, see De Vries and Bonvin⁹) on the second partner, and enter these predictions as either active or passive residues</p> <p>Problem: I do not have any experimental data. Can I still use HADDOCK? Solution: It depends on your system. You will need to have very good starting structures (see Box 3, <i>Starting Structure Quality</i>). Even then, the docking results will be highly speculative. There are two strategies that you can follow. The Expert and Guru interfaces offer various modes of <i>ab initio</i> docking (see Step 10 for more details). Alternatively, you could do interface prediction on both partners (for a review, see De Vries and Bonvin⁹), and enter the predictions as active residues. We have developed a consensus interface prediction method called CPORT (De Vries and Bonvin, manuscript in preparation) that is optimized for use with HADDOCK</p> <p>For either strategy, increase the number of rigid body structures to at least 5,000 (Step 11). In case of interface prediction, aim to overpredict the interface and increase the fraction of restraints that is discarded per docking trial (up to 90%; see Box 2 for details). A specialized web interface for interface predictions, based on the Easy interface, that automatically implements these changes in the protocol, is currently under development</p> <p>Finally, you may also consider using a docking program designed for <i>ab initio</i> docking, such as ZDOCK^{33,34}, to see whether the results agree with HADDOCK. In the future, we plan to integrate ZDOCK as an alternative to the rigid-body stage in HADDOCK</p> |
| 17 | <p>Problem: I get an error message ‘PDB contains multiple forms of the same residue. This is not supported in the current form’ Solution: Some crystal structures contain multiple conformations of a residue, with a label in column 17 of the PDB. This is not supported by HADDOCK. You must convert your PDB to a multimodel PDB to support multiple conformations. See Box 3, <i>Ensembles of Structures</i> for more details. Alternatively, if the residue is not important for the interaction, you can just choose one conformation, remove the other conformation and clear the label</p> <p>Problem: I get an error message ‘PDB contains a residue number that is not a number’ Solution: See Box 3, <i>Numerical Residue Identifiers</i></p> <p>Problem: I get an error message ‘PDB contains multiple chains with overlapping residues’ Solution: See Box 3, <i>Nonoverlapping Numbering</i></p> <p>Problem: I get an error message ‘PDB contains multiple residues with number X in chain Y’ Solution: See Box 3, <i>Nonoverlapping Numbering and Numerical Residue Identifiers</i></p> <p>Problem: I get an error message ‘PDB contains elemental ion X without charge/with inconsistent charge’. How do I deal with elemental ions? Solution: See Box 3, <i>Elemental Ions</i></p> |

(continued)



TABLE 2 | Troubleshooting table (continued).

| Step | Problem and solution |
|------|---|
| | <p>Problem: I get an error message 'Mismatch between the atoms in the database and the supplied atoms for ligand X'</p> <p>Possible reason: This is due to a cofactor in your PDB. In the case of cofactors, the PRODRG server³⁵ was contacted for parameters but failed to provide them for some or all atoms. For more details, see Box 3, Cofactors And Modified Amino Acids</p> <p>Problem: I get an error message 'PDB contains an unknown modified amino acid X'</p> <p>Possible reason: Any residue specified as ATOM is assumed to be an amino acid. If your protein contains a modified amino acid that is not recognized by HADDOCK, this error message will be the result. A list of supported modified amino acids can be found at: http://haddock.chem.uu.nl/services/HADDOCK/library.html</p> <p>Alternatively, it could be that you were actually trying to define a cofactor. Cofactors are supported even if they are unknown because the PRODRG server is automatically contacted for parameters. However, in order for your cofactor to be recognized, it must be specified as HETATM. For more details, see Box 3, Cofactors And Modified Amino Acids</p> |
| 18 | <p>Problem: I get an error message 'There was an error in the topology generation step'</p> <p>Possible reason: This error message is relatively rare. It is most likely caused by a low-quality starting structure (see Box 3, Starting Structure Quality) or problems with a cofactor (see Box 3, Cofactors And Modified Amino Acids). The result page will present you with three CNS output files that may contain error messages providing further insight</p> <p>Problem: I get an error message 'There was an error in the rigid body stage of docking'</p> <p>Possible reason: This error message is very rare. You should look at the monitor page ('haddock.out' in same web directory if you forgot to save the link) if all of your structures failed, or only a fraction of them. In the latter case, it would probably be caused by a low-quality starting structure (see Box 3, Starting Structure Quality) or problems with a cofactor (see Box 3, Cofactors And Modified Amino Acids). In case of a cofactor, there was probably a data formatting problem that was not intercepted by the server</p> <p>The results page presents the CNS output file from the first docking trial and may contain error messages providing further insight</p> <p>Problem: I get an error message 'There was an error in the simulated annealing stage'</p> <p>Possible reason: This is the most common stage for a docking to fail, once it has started. This is caused either by a low quality structure (see Box 3, Starting Structure Quality) that has exploded or by cofactors, ions or protein fragments that have drifted away. In the latter case, it is almost always a cofactor or anion, as for cations and fragments, automatic restraints are defined by HADDOCK to keep them together. Segments defined as fully flexible may also occasionally unfold and cause the protein to 'explode'. The results page presents the CNS output file from the first docking trial and may contain error messages providing further insight</p> <p>Solution: Remove all cofactors that are not important for the complex formation. If that does not help, consider docking without any cofactors at all</p> <p>Before docking, refine the structure (e.g., using molecular dynamics) to increase its stability</p> <p>Define intra-molecular restraints to preserve the structure, using a custom restraints file (see Box 2). This requires access to the Expert interface server</p> <p>Problem: I get an error message 'There was an error in the water refinement stage'</p> <p>Solution: This error is relatively rare. The causes are usually the same as for an error in the simulated annealing stage, see the previous entry for details</p> <p>Problem: I get an error message 'There was an error in the analysis of the simulated annealing/water refinement stage'</p> <p>Possible reason: This error is very rare. The most likely cause is a weird PDB format or cofactor, not intercepted by the server, preventing the HADDOCK analysis tools from running successfully. Download the run and look in the structures/it1/analysis and structures/it1/water/analysis directories for more information. If you have HADDOCK installed on your local computer, you might be able to finish the analysis manually (see the HADDOCK manual at http://nmr.chem.uu.nl/haddock for details)</p> <p>Problem: I get an error message 'There was an error in the clustering of the solutions'</p> <p>Solution: The docking finished successfully, but the results did not converge: the final docking solutions were not similar enough to form clusters according to standard criteria. If your system is very large, you could consider loosening the criteria: download the docking results and do a manual clustering analysis (see the HADDOCK manual at http://nmr.chem.uu.nl/haddock for details). However, the best remedy is to obtain more specific restraints</p> <p>Note: if there is internal symmetry in one or both of your molecules, this is not taken into account during the clustering process. In that case, you should download the results and do a manual clustering analysis. This requires that HADDOCK is installed on your local machine. See the HADDOCK manual at http://nmr.chem.uu.nl/haddock for more details</p> |
| 19 | <p>Problem: My docking results have disappeared!</p> <p>Solution: HADDOCK results are deleted after 1 week. If you have kept your parameter file, upload it using the File upload interface to re-do your docking</p> |

(continued)

TABLE 2 | Troubleshooting table (continued).

| Step | Problem and solution |
|------|--|
| | <p>Problem: I want to re-do my docking with just one parameter changed! Do I have to fill in the web form again? What about a batch of docking runs?</p> <p>Solution: HADDOCK parameter files can be edited in a text editor, and they can also be manipulated in Python using the Spyder framework (Box 5). With the File upload interface, you can upload your modified parameter file. There is also a script to upload parameter files automatically, which is available on request. The HADDOCK server uses internal queuing to prevent one user from monopolizing the server; feel free to submit moderately large numbers of docking runs (up to 30) simultaneously</p> |

ANTICIPATED RESULTS

Figure 3 shows an example of a HADDOCK run output, together with an example of visualization with Jmol³². This is the result of a docking of the proteins E2A (1F3G) and HPr (1HDN), involved in the *Escherichia coli* sugar phosphotransferase system. The complex 1GGR has been previously solved by NMR³⁶. However, the HADDOCK docking run was driven by CSP data only. The top structure from the top cluster has an interface RMSD of only 1.1 Å to the first model of the NMR structure, nearly within experimental error. However, the second cluster, in which the HPr protein is 90 degrees rotated, has a HADDOCK score that is only slightly inferior. Therefore, in the case of E2A-HPr, additional experimental information has to be taken into account to completely rule out alternatives to the highest-scoring (and correct) solution.

The E2A-HPr docking results can be viewed at <http://haddock.chem.uu.nl/Haddock/run-example/index.html>. A web form with the docking parameters for E2A-HPr filled in can be viewed at <http://haddock.chem.uu.nl/services/HADDOCK/haddockserver-demo.html>.

Figure 3 | Example of docking results from the HADDOCK server. All docking solutions are clustered and ranked according to their HADDOCK scores. The best structures of each cluster can be downloaded or directly viewed in Jmol (screenshots are shown on the right).



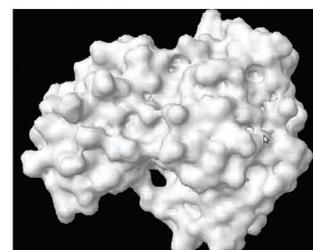
HADDOCK server status for docking run e2ahpr

Status: **FINISHED**

Your HADDOCK run has successfully completed. The complete run can be downloaded as a gzipped tar file [here](#) (unavailable for this example). Your results have been clustered, and the top 10 clusters are shown here. The top cluster is the most reliable according to HADDOCK.

CLUSTER 1

| | |
|---|-----------------|
| HADDOCK score | -154.5 +/- 4.4 |
| Cluster size | 126 |
| RMSD from the overall lowest-energy structure | 0.6 +/- 0.4 |
| Van der Waals energy | -39.0 +/- 1.3 |
| Electrostatic energy | -591.8 +/- 15.3 |
| Desolvation energy | -0.4 +/- 4.8 |
| Restraints violation energy | 32.6 +/- 12.40 |
| Buried Surface Area | 1648.4 +/- 40.3 |

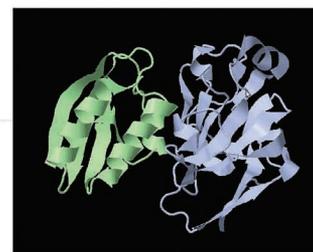


View the docking solutions in a Jmol structure viewer. Your browser must be Java enabled:

- Nr 1 best structure: [View structure](#) [Download structure](#)
- Nr 2 best structure: [View structure](#) [Download structure](#)
- Nr 3 best structure: [View structure](#) [Download structure](#)
- Nr 4 best structure: [View structure](#) [Download structure](#)

CLUSTER 3

| | |
|---|-----------------|
| HADDOCK score | -141.1 +/- 8.7 |
| Cluster size | 20 |
| RMSD from the overall lowest-energy structure | 8.9 +/- 0.1 |
| Van der Waals energy | -41.0 +/- 5.4 |
| Electrostatic energy | -554.1 +/- 26.6 |
| Desolvation energy | 3.6 +/- 5.4 |
| Restraints violation energy | 71.2 +/- 10.83 |
| Buried Surface Area | 1693.0 +/- 78.4 |



View the docking solutions in a Jmol structure viewer. Your browser must be Java enabled:

- Nr 1 best structure: [View structure](#) [Download structure](#)
- Nr 2 best structure: [View structure](#) [Download structure](#)
- Nr 3 best structure: [View structure](#) [Download structure](#)
- Nr 4 best structure: [View structure](#) [Download structure](#)

Note that HADDOCK results are deleted after one week.

BOX 5 | THE SPYDER FRAMEWORK

The Spyder framework (<http://www.spyderware.nl>) is an extensible, Python-based data validation and processing framework. A Spyder data model consists of parameter names and their data types, as well as Python code for data validation. Out of this data model, all the essential parts of a web server can be generated, namely, a web form, a parser for the web form, validation code, a Python class to hold the data and code to read and write parameter files. There is a separate Spyder data model for each of the HADDOCK server web interfaces, as well as for various intermediate stages of data processing. In Spyder, the definition of converter functions causes data models to automatically acquire functionality from each other. In this manner, it does not matter whether a data model instance is constructed from a web form or from a parameter file. Therefore, all four HADDOCK web interfaces are completely auto-generated and processed by a single CGI script written in Python. This architecture is illustrated in **Figure 1**.

The Spyder framework can be freely downloaded at <http://www.spyderware.nl>. We have developed a Spyder module 'haddock' to generate HADDOCK server parameter files by scripting. Documentation on the HADDOCK server is available at <http://www.spyderware.nl/doc/Spyder.HaddockRunParameters-class.html>

This module will also be distributed with future versions of the HADDOCK program, so that the HADDOCK server can also be used just to generate parameter files, which can be downloaded and run on a local cluster or supercomputer.

We recommend that HADDOCK parameter files be deposited along with HADDOCK-generated structures in the Protein Data Bank.

ACKNOWLEDGMENTS This work was supported by The Netherlands Organization for Scientific Research (VICI grant no. 700.56.442 to A.B.) and the European Community (FP6 integrated Project SPINE2-COMPLEX, contract no. 032220; FP6 STREP project 'ExtendNMR', contract no. LSHG-CT-2005-018988; and FP7 e-Infrastructure 'e-NMR' I3 project, grant number 213010). The Dutch BiG Grid project with financial support from The Netherlands Organization for Scientific Research (NWO) is acknowledged for the use of the computing and storage facilities. We would like to thank J. Verhaal for critical reading of the manuscript.

AUTHOR CONTRIBUTIONS A.M.J.J.B. is the primary author of the HADDOCK program. S.J.d.v. developed the HADDOCK web server and the Spyder framework. M.v.D. developed the nucleic acid functionality of the server and the graphical design of the server web site. A.M.J.J.B. supervised the project. S.J.d.v. wrote the paper.

COMPETING FINANCIAL INTERESTS The authors declare no competing financial interests.

Published online at <http://www.natureprotocols.com/>.

Reprints and permissions information is available online at <http://npg.nature.com/reprintsandpermissions/>.

- van Dijk, A.D., Boelens, R. & Bonvin, A.M. Data-driven docking for the study of biomolecular complexes. *FEBS. J.* **272**, 293–312 (2005).
- Bonvin, A.M. Flexible protein-protein docking. *Curr. Opin. Struct. Biol.* **16**, 194–200 (2006).
- Janin, J. Assessing predictions of protein-protein interaction: the CAPRI experiment. *Protein Sci.* **14**, 278–283 (2005).
- Méndez, R., Leplae, R., Lensink, M.F. & Wodak, S.J. Assessment of CAPRI predictions in rounds 3–5 shows progress in docking procedures. *Proteins* **60**, 150–169 (2005).
- Dominguez, C., Boelens, R. & Bonvin, A.M.J.J. HADDOCK: A protein-protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.* **125**, 1731–1737 (2003).
- de Vries, S.J. *et al.* HADDOCK versus HADDOCK: new features and performance of HADDOCK2.0 on the CAPRI targets. *Proteins* (2007).
- van Dijk, A.D.J. *et al.* Data-driven docking: HADDOCK's adventures in CAPRI. *Proteins* **60**, 232–238 (2005).
- de Vries, S.J., van Dijk, A.D.J. & Bonvin, A.M.J.J. WHISCY: What information does surface conservation yield? Application to data-driven docking. *Proteins* **63**, 479–489 (2006).
- de Vries, S.J. & Bonvin, A.M. How proteins get in touch: interface prediction in the study of biomolecular complexes. *Curr. Protein Pept. Sci.* **9**, 394–406 (2008).
- van Dijk, M. *et al.* Information-driven protein-DNA docking using HADDOCK: it is a matter of flexibility. *Nucleic Acids Res.* **34**, 3317–3325 (2006).
- Berman, H.M. *et al.* The Protein Data Bank. *Nucleic Acids Res.* **28**, 235–242 (2000).
- Moreira, I.S., Fernandes, P.A. & Ramos, M.J. Protein-protein docking dealing with the unknown. *J. Comput. Chem.* (2009).
- Brunger, A.T. *et al.* Crystallography & NMR system: a new software suite for macromolecular structure determination. *Acta Crystallogr. D Biol. Crystallogr.* **54** (Pt 5): 905–921 (1998).
- Schwieters, C.D., Kuszewski, J.J., Tjandra, N. & Clore, G.M. The Xplor-NIH NMR molecular structure determination package. *J. Magn. Reson.* **160**, 65–73 (2003).
- Vranken, W.F. *et al.* The CCPN data model for NMR spectroscopy: development of a software pipeline. *Proteins* **59**, 687–696 (2005).
- Hennig, J., Hennig, K.D. & Sunnerhagen, M. MTMDAT: automated analysis and visualization of mass spectrometry data for tertiary and quaternary structure probing of proteins. *Bioinformatics* **24**, 1310–1312 (2008).
- Comeau, S.R., Gatchell, D.W., Vajda, S. & Camacho, C.J. ClusPro: an automated docking and discrimination method for the prediction of protein complexes. *Bioinformatics* **20**, 45–50 (2004).
- Tovchigrechko, A. & Vakser, I.A. GRAMM-X public web server for protein-protein docking. *Nucleic Acids Res.* **34**, W310–314 (2006).
- Andrusier, N., Nussinov, R. & Wolfson, H.J. FireDock: fast interaction refinement in molecular docking. *Proteins* **69**, 139–159 (2007).
- Mashiach, E. *et al.* FireDock: a web server for fast interaction refinement in molecular docking. *Nucleic Acids Res.* **36**, W229–W232 (2008).
- Schneidman-Duhovny, D., Inbar, Y., Nussinov, R. & Wolfson, H.J. PatchDock and SymmDock: servers for rigid and symmetric docking. *Nucleic Acids Res.* **33**, W363–W367 (2005).
- Mustard, D. & Ritchie, D.W. Docking essential dynamics eigenstructures. *Proteins* **60**, 269–274 (2005).
- Ritchie, D.W. & Kemp, G.J.L. Protein docking using spherical polar Fourier correlations. *Proteins* **39**, 178–194 (2000).
- Lyskov, S. & Gray, J.J. The RosettaDock server for local protein-protein docking. *Nucleic Acids Res.* **36**, W233–W238 (2008).
- Vriend, G. WHAT IF: a molecular modeling and drug design program. *J. Mol. Graph* **8**, 52–56, 29 (1990).
- Tjandra, N. *et al.* Use of dipolar H-1-N-15 and H-1-C-13 couplings in the structure determination of magnetically oriented macromolecules in solution. *Nat. Struct. Biol.* **4**, 732–738 (1997).
- Meiler, J., Blomberg, N., Nilges, M. & Griesinger, C. A new approach for applying residual dipolar couplings as restraints in structure elucidation. *J. Biomol. NMR* **16**, 245–252 (2000).
- Tjandra, N. *et al.* Defining long range order in NMR structure determination from the dependence of heteronuclear relaxation times on rotational diffusion anisotropy. *Nat. Struct. Biol.* **4**, 443–449 (1997).
- van Dijk, A.D., Fushman, D. & Bonvin, A.M. 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* **60**, 367–381 (2005).
30. van Dijk, A.D., Kaptein, R., Boelens, R. & Bonvin, A.M. Combining NMR relaxation with chemical shift perturbation data to drive protein-protein docking. *J. Biomol. NMR* **34**, 237–244 (2006).
 31. van Dijk, A.D. & Bonvin, A.M. Solvated docking: introducing water into the modelling of biomolecular complexes. *Bioinformatics* **22**, 2340–2347 (2006).
 32. Jmol: an open-source Java viewer for chemical structures in 3D.
 33. Chen, R., Li, L. & Weng, Z. ZDOCK: an initial-stage protein-docking algorithm. *Proteins* **52**, 80–87 (2003).
 34. Wiehe, K. *et al.* The performance of ZDOCK and ZRANK in rounds 6–11 of CAPRI. *Proteins* **69**, 719–725 (2007).
 35. Schuttelkopf, A.W. & van Aalten, D.M. PRODRG: a tool for high-throughput crystallography of protein-ligand complexes. *Acta. Crystallogr. D. Biol. Crystallogr.* **60**, 1355–1363 (2004).
 36. Wang, G. *et al.* 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 (2000).
 37. Hubbard, S.J. & Thornton, J.M. *NACCESS* (Department of Biochemistry and Molecular biology, University College London, 1993).
 38. Linge, J.P., Habeck, M., Rieping, W. & Nilges, M. ARIA: automated NOE assignment and NMR structure calculation. *Bioinformatics* **19**, 315–316 (2003).
 39. Nilges, M., Gronenborn, A.M., Brunger, A.T. & Clore, G.M. Determination of three-dimensional structures of proteins by simulated annealing with interproton distance restraints. Application to crambin, potato carboxypeptidase inhibitor and barley serine proteinase inhibitor 2. *Protein Eng.* **2**, 27–38 (1988).
 40. van Dijk, M. & Bonvin, A.M. 3D-DART: a DNA structure modelling server. *Nucleic Acids Res.* (2009).
 41. Lu, X.J. & Olson, W.K. 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic Acids Res.* **31**, 5108–5121 (2003).
 42. Lu, X.J. & Olson, W.K. 3DNA: a versatile, integrated software system for the analysis, rebuilding and visualization of three-dimensional nucleic-acid structures. *Nat. Protoc.* **3**, 1213–1227 (2008).