The HADDOCK2.4 web server for integrative modeling of biomolecular complexes

Rodrigo V. Honorato[1], Mikael E. Trellet[1,2], Brian Jiménez-García[1,3], Jörg J. Schaarschmidt[1,4], Marco Giulini[1], Victor Reys[1], Panagiotis I. Koukos[1,5], João P. G. L. M. Rodrigues[1,6], Ezgi Karaca[1,7], Gydo C. P. van Zundert[1,6], Jorge Roel-Touris[1,8], Charlotte W. van Noort[1], Zuzana Jandová[1,9], Adrien S. J. Melquiond[1,10] & Alexandre M. J. J. Bonvin[1] ✉

Abstract

Interactions between macromolecules, such as proteins and nucleic acids, are essential for cellular functions. Experimental methods can fail to provide all the information required to fully model biomolecular complexes at atomic resolution, particularly for large and heterogeneous assemblies. Integrative computational approaches have, therefore, gained popularity, complementing traditional experimental methods in structural biology. Here, we introduce HADDOCK2.4, an integrative modeling platform, and its updated web interface (https://wenmr.science.uu.nl/haddock2.4). The platform seamlessly integrates diverse experimental and theoretical data to generate high-quality models of macromolecular complexes. The user-friendly web server offers automated parameter settings, access to distributed computing resources, and pre- and post-processing steps that enhance the user experience. To present the web server's various interfaces and features, we demonstrate two different applications: (i) we predict the structure of an antibody–antigen complex by using NMR data for the antigen and knowledge of the hypervariable loops for the antibody, and (ii) we perform coarse-grained modeling of PRC1 with a nucleosome particle guided by mutagenesis and functional data. The described protocols require some basic familiarity with molecular modeling and the Linux command shell. This new version of our widely used HADDOCK web server allows structural biologists and non-experts to explore intricate macromolecular assemblies encompassing various molecule types.

[1]Computational Structural Biology Group, Bijvoet Centre for Biomolecular Research, Department of Chemistry, Faculty of Science, Utrecht University, Utrecht, The Netherlands. [2]Present address: Fluigent, Le Kremlin-Bicêtre, France. [3]Present address: Zymvol Biomodeling SL, Barcelona, Spain. [4]Present address: Karlsruhe Institute of Technology (KIT), Institute of Nanotechnology, Eggenstein-Leopoldshafen, Germany. [5]Present address: Biomedical Research Foundation, Academy of Athens, Athens, Greece. [6]Present address: Schrödinger Inc., New York, NY, USA. [7]Present address: Izmir Biomedicine and Genome Center, Izmir, Turkey. [8]Present address: Protein Design and Modeling Lab, Department of Structural Biology, Molecular Biology Institute of Barcelona (IBMB-CSIC), Barcelona, Spain. [9]Present address: Boehringer Ingelheim International GmbH, Vienna, Austria. [10]Present address: Utrecht Medical Center, Utrecht, the Netherlands. ✉e-mail: a.m.j.j.bonvin@uu.nl

Introduction

Macromolecules such as proteins and nucleic acids make up most of the machinery of cells and are responsible for performing a plethora of cellular functions and maintaining life. Although many meaningful insights about their functions can be deduced through experimental work, these can fail to provide all the required structural information to model the involvement of biomolecules and their interactions at atomic details. Over the last decade, integrative, information-driven computational approaches for modeling the structure of biomolecules have been increasing in popularity and are by now well established in the toolbox of structural biologists, complementing the classical experimental structural biology techniques such as X-ray crystallography, NMR spectroscopy and cryo-electron microscopy (cryo-EM)[1–3]. In recent years, new valuable artificial intelligence (AI)-based tools have been developed, with AlphaFold2[4] leading the way[5,6]. The related AlphaFold database (https://alphafold.ebi.ac.uk)[4,7] is now providing over 200 million models of proteins, covering almost all of UniProt[8]. This database provides a unique resource, which is further shifting the research focus from single structures to macromolecular assemblies and high-order complexes, often consisting of various molecule types. This effort is seen as a stepping stone toward realizing a structural model of the cell in atomic or near-atomic detail[9].

The complexity of such assemblies requires integrative computational approaches to combine experimental information from various sources with (AI) structures of the individual components to build models of these higher-order complexes. Although the resulting integrative models are not equivalent to experimental structures, their relevance has been recognized by the worldwide Protein Data Bank (PDB)[10], which has developed the PDB-dev repository to collect them (https://pdb-dev.wwpdb.org)[11,12].

Computational docking

Among the various classical computational approaches that can be complementary and alternative to experiments, docking has typically been the method of choice for predicting the 3D structures of biomolecular complexes. Computational docking aims at obtaining the structure of a macromolecular complex given its interacting units. Over the years, methods have been developed to study the interactions of different molecule types such as ligands, nucleic acids, peptides, carbohydrates and mainly proteins. Although these interactions can be obtained via ab initio protocols (i.e., considering only the individual input molecules), the addition of experimental and/or predicted information or both (in the form of restraints) has been shown to greatly increase the quality and accuracy of the generated models[1,13–18]. In the past decade, the field has been moving toward integrative approaches[1,19], with various software adding restraint-based protocols[20–22]. The HADDOCK[23] (**H**igh **A**mbiguity **D**riven **DOCK**ing) software is one of the pioneers in integrative modeling.

AI has also entered the field of protein–protein complex prediction. Although AlphFold2 was originally trained only on single structures, its potential for predicting complexes was instantly recognized, with applications to protein–peptide complex prediction[24] and protein–protein complexes, sometimes in combination with classical docking methods[25]. A new version trained specifically for complexes, AlphaFold-Multimer[26], was released shortly after AlphFold2. AlphaFold2 has demonstrated excellent performance for protein–protein complex prediction in cases in which co-evolution can be used, with applications to interactome-wide predictions[27,28]. Other methods like RoseTTafold have also been applied to the prediction of protein–protein complexes[29]. The use of AI-driven structure prediction has been made accessible through Jupyter notebooks available via ColabFold[30]. Most methods to date, however, have concentrated on protein–protein and protein–peptide complexes and usually require a co-evolution signal to generate successful predictions. There is thus still a need for 'more classical' approaches to model, for example, antibody–antigen complexes (because limited or no co-evolution signal is clearly present in the hypervariable loops of antibody) or large and complex assemblies consisting of various molecule types (mixture of proteins, nucleic acids and other molecules).

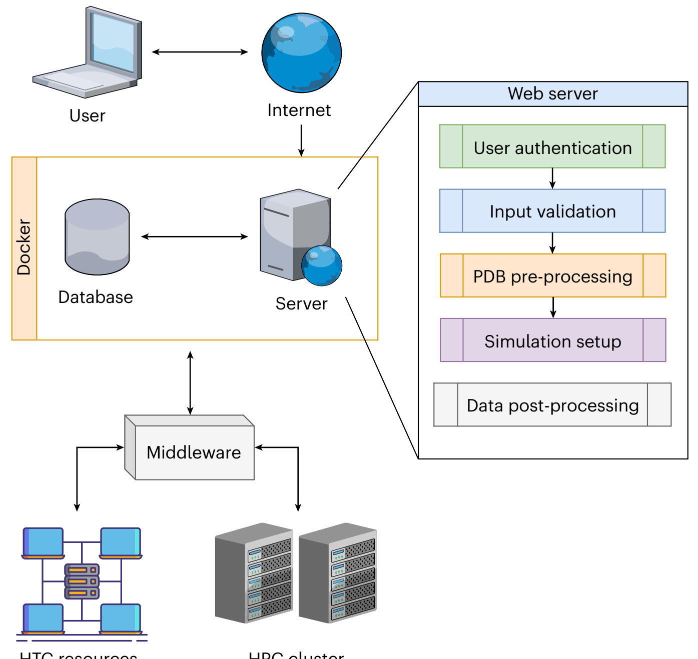


Fig. 1 | Overall structure of the HADDOCK2.4 web server. After authentication on the server (https://wenmr.science.uu.nl/haddock2.4), registered users can submit their data and fine-tune the parameters via the web interface. The server takes care of validating the input data and parameter settings, pre-processing the input files and setting up the docking computation, which are then handled by the middleware (the connecting component between the server and the computational resources). The latter takes care of the execution and monitoring of the jobs, distributing the compute units either to a local high-performance cluster (HPC) or to distributed high-throughput computing (HTC) (grid) resources provided by various sites in Europe, Asia and the US Open Science Grid under the umbrella of EGI (www.egi.eu) and the European Open Science Cloud (https://eosc-portal.eu)[40].

Here, we describe HADDOCK, our integrative modeling platform powered by the Crystallography & NMR System (CNS)[31], and its web interface. Since the original publication of the HADDOCK web server in 2010[32] and subsequent versions[33], additional features have been added, resulting in a new, user-friendly version that is easily accessible by non-expert users and available at https://wenmr.science.uu.nl/haddock2.4. Since its release, the HADDOCK2.4 web server has grown in popularity and now accounts for ~50% of the total number of all docking runs (currently >0.6 million) submitted to the HADDOCK server. We present here new features and usage statistics and, more importantly, illustrate the use of the HADDOCK2.4 server by modeling two types of complexes: (i) an antibody–antigen complex by using knowledge of the hypervariable loops and an NMR-based definition of the epitope binding site (Procedure 1) and (ii) a nucleosome complex illustrating the coarse-graining capabilities of the server to model large complexes[34,35] (Procedure 2).

The HADDOCK2.4 web server

HADDOCK is an integrative modeling platform that can integrate various sources of experimental information such as mutagenesis, mass spectrometry (e.g., cross-links and hydrogen/deuterium exchange), NMR (chemical shift perturbation, residual dipolar couplings, etc.), small angle x-ray scattering, cryo-EM and/or theoretical data such as bioinformatics interface prediction and co-evolution contacts to obtain high-quality models of macromolecular complexes. Here, we present HADDOCK, v2.4, together with an updated version of the web server that builds upon the previously described features[33] by adding support for up to 20-body docking, allowing for coarse-grained representations of proteins and DNA[34,35] by using the MARTINI forcefield[36], supporting cryo-EM maps as restraints[17,37,38] and shape restraints[39] and providing support for glycosylated proteins and various modified amino acids and carbohydrates (https://wenmr.science.uu.nl/haddock2.4/library).

Because of the increased feature set and exposed parameters, the technology behind the server has been completely redesigned to provide a modern, robust and General Data Protection Regulation–compliant web-based graphical user interface to HADDOCK (and all the other services offered by the Computational Structural Biology group (https://wenmr.science.uu.nl). The HADDOCK2.4 web server uses the Flask web framework to serve its content and bridge with the back end software and is tightly linked to a PostgreSQL database to handle both user and processed data (Fig. 1).

Table 1 | Feature comparison between the v2.2 and v2.4 of the HADDOCK web server and source code

HADDOCK	v2.2	v2.4
Development operations		
Docker integration	–	✓
Continuous integration (Jenkins)	–	✓
Single sign on (via EGI Check-in)	✓	✓
Front end		
Interactive result pages	–	✓
Refinement interface	✓	✓
File submission	✓	✓
Back end		
GDPR compliant	✓	✓
Validation of input parameters	–	✓
Molecule type-specific optimal settings	–	✓
User workspace	–	✓
Admin control panel	–	✓
Reporting API	–	✓
Features		
Molecule type: shape	–	✓
Molecule type: glycan	–	✓
Cryo-EM data	–	✓
20-body docking	–	✓
Symmetry restraints	✓	✓
Coarse grain representation	–	✓
Glycosylated proteins	–	✓
Cyclic peptides	–	✓

API, application programming interface; GDPR, General Data Protection Regulation.

To enhance scalability and testability and to accelerate and facilitate the setup process, the portal is deployed via Docker containers on our own servers. We also upgraded the front end and followed modern usability standards. The front end now uses Bootstrap, a popular toolkit framework, and the server has been made significantly more interactive and user friendly. In terms of new capabilities, the HADDOCK2.4 server represents a significant step forward in the integrative modeling of biomolecular complexes (see Table 1 for the main new features and options and a comparison with the previous 2.2 version).

Experience from multiple face-to-face workshops (and a cursory examination of user support requests on the HADDOCK forum at ask.bioexcel.eu) indicates that most users encounter difficulties in the first step of the submission process, during input file submission. To improve the user experience and guide troubleshooting, the server performs extensive validation of the input data and parameter settings before submission, which is done in the back end with a collection of in-house developed modules (see https://wenmr.science.uu.nl/haddock2.4/manual for additional details). Validation of user data and parameter settings is crucial to ensure efficient usage of the server. This is under steady development based on user feedback and unexpected issues encountered in its daily usage; the frequently asked questions are quickly accessible in the interface and are visible on a dedicated page (https://wenmr.science.uu.nl/haddock2.4/faq). In the context of the BioExcel Center of Excellence for Computational Biomolecular Research (https://bioexcel.eu), we have created an in-depth guide to best practices (https://www.bonvinlab.org/software/bpg/).

Furthermore, the result pages have been re-designed to offer a richer presentation, including interactive plots and online visualization of the generated complexes. To increase the

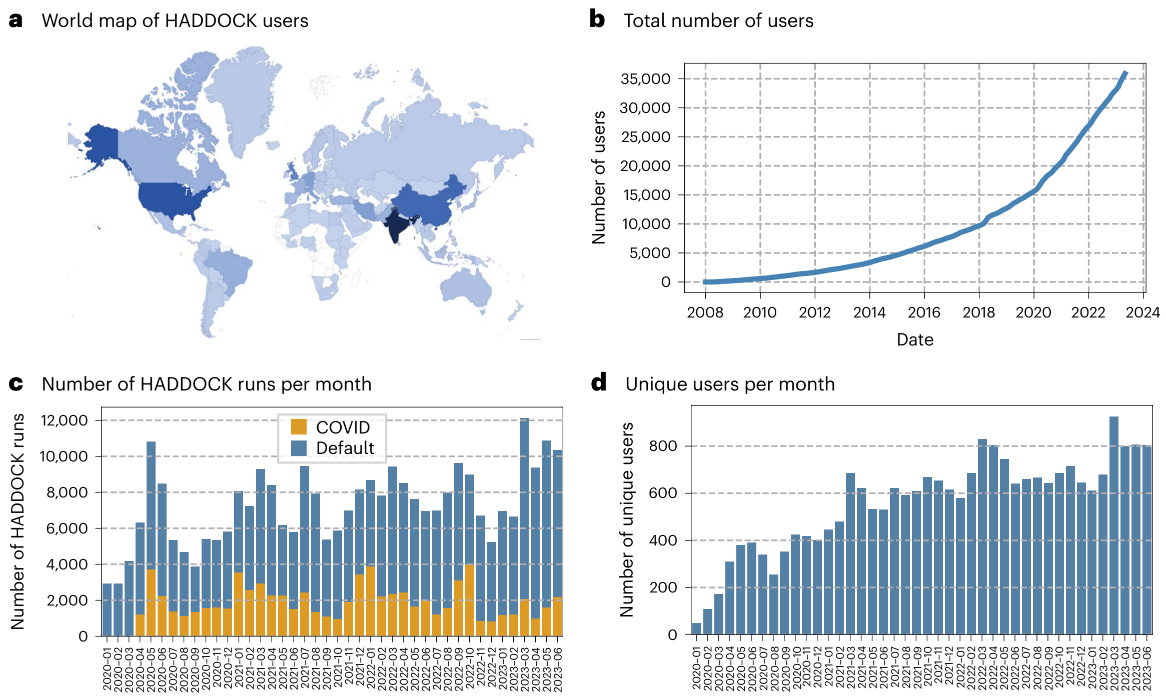


Fig. 2 | HADDOCK server statistics for the period January 2020 to June 2023. **a**, World map of HADDOCK users (colored by number of users). **b**, Cumulative number of HADDOCK users registered. **c**, Number of HADDOCK runs per month, including the fraction of jobs tagged as COVID-19–related since April 2020. **d**, Unique users per month since January 2020.

accessibility and usability of the new features, several online tutorials that use the v2.4 server are available (https://www.bonvinlab.org/education/HADDOCK24/).

Advantages and limitations

Usability and accessibility are cornerstones of HADDOCK, and this has been the focal point of the continuous development and user-driven development of the web portal, which has been online since 2008. The HADDOCK web server is available free of charge for non-profit users upon registration. This registration is necessary because users should be identified for the use of the distributed EGI/European Open Science Cloud (EOSC) high-throughput computing (HTC) resources, so that they can keep track of their submissions via a dedicated workspace and receive support if they encounter errors and to provide anonymized usage statistics to the various funding agencies. Upon registration, users agree to the privacy and terms of use conditions (https://wenmr.science.uu.nl/privacy; https://wenmr.science.uu.nl/conditions). Currently, our database contains >46,000 registered users from 145 different countries (https:// wenmr.science.uu.nl/stats) with a steady yearly growth (Fig. 2a,b). By using the web interface, users gain access to computational resources that would otherwise be unavailable to them, allowing researchers all over the world to gain a deeper understanding of biomolecular interactions in their field of research. The server is also regularly used for educational purposes (e.g., teaching assignments). User submissions typically translate into a few hundred to thousands of individual compute jobs, resulting in ~20 million jobs being processed per year on the EGI/EOSC HTC resources (not counting our local cluster usage), exhausting >55 million CPU h/year. As a result of the coronavirus disease 2019 (COVID)-19 pandemic, we have also experienced almost a tripling in both submissions (Fig. 2c) and active unique users per month (Fig. 2d), with ~25–30% of all submissions being COVID-19–related since April 2020.

One of the limitations of the server is that users must provide 3D structural models of the components as a starting point for the docking. For example, the server does not accept amino

acid sequences for proteins or SMILE (Simplified Molecular-Input Line-Entry System) strings for small ligands. Furthermore, the server is not designed for high-throughput screening of a large number of interactions (e.g., drug screening), because this would become computationally very expensive. It is designed to ensure a fair distribution of resources between users. Therefore, it is better to run large screening efforts on high-performance computing (HPC) resources by using a local installation of HADDOCK. The size of the complexes that can be modeled with HADDOCK is also limited by the number of molecules supported (20) and the number of residues per molecule, which cannot exceed 9,999 because of PDB format issues. For very large assemblies, the computing time might also become a limiting factor causing job failures because CPU time limits are reached. Because the server is offered as a free resource, there are limitations to the computational resources available.

Experimental design

Input structures

The HADDOCK program does not perform any data validation; thus, the web interface is complementary to it, adding several checks to make sure the structures are adequate. The input structures are first checked for formatting issues. These can be experimental structures extracted from the PDB or structural models from simulations or taken from, for example, the AlphaFold database. HADDOCK expects structures to follow the PDB format and be submitted in either .pdb or .cif format. If a structure is submitted with a .cif extension, it will be converted to .pdb, considering only ATOM/HETATM records. No other formats are supported. This effectively limits the size of a molecular component that can be handled by HADDOCK to a maximum of 9,999 residues.
The input structures are further checked for the following:

Category	Notes
Overlapping residue numbers	Residue numbers must be unique within a given input molecule. For example, if providing a homodimer protein as input, the residue numbering of the two chains must be distinct because those will be treated as a single chain. These can be addressed with a PDB processing tool such as pdb-tools[48]
Presence of target chains	The server allows the selection of specific chains from the input PDB and checks if those are present
Presence of insertion codes	Residue numbers must be unique within a given input molecule. This also applies to insertion codes. For example, amino acid insertions in the hypervariable loops of antibodies can be indicated by adding a letter to the residue number (e.g., 101A, 101B, etc.). In case of insertions, the numbering must be corrected to a sequential numbering. These can be addressed with a PDB processing tool such as pdb-tools[48] or PDBFixer[56]
Presence of alternate locations	In high-resolution crystal structure, different conformations of a residue or side chain might be present, with partial occupancies. HADDOCK allows for only a single occurrence of a residue or side chain, and a selection of the most relevant alternate location must be made (e.g., selecting the highest-occupancy conformation), or the confirmation should be split into an ensemble of conformations. These can be addressed with a PDB processing tool such as pdb-tools or PDBFixer
Residues	For all residues defined in the input files, the server checks if they are supported. Refer to https://wenmr.science.uu.nl/haddock2.4/library for a list of supported modified amino acids and glycans
Metal ions	If metal ions are identified, the server checks for the proper naming. The charge state should be defined in both the atom and residue names. Refer to https://wenmr.science.uu.nl/haddock2.4/library for an example

A list of commonly found errors and how to fix them can be found at https://wenmr.science.uu.nl/haddock2.4/faq.

Restraints

Before docking, when using the HADDOCK web interface, it is necessary to provide a list of active residues (residues known to make contact within the complex) and passive residues (residues that potentially make contact) for each molecule involved (unless an ab initio docking protocol is followed). HADDOCK generates ambiguous interaction restraints on the basis of these residues to drive the docking. The passive residues can be defined automatically on the basis of the active residues specified by the user. These passive residues are typically located on the protein surface and within a given radius (default of 6.5 Å) of any

active residue. It is important to note that not all residues should be designated as active, and only residues physically capable of making contact should be included. By default, the server will filter out residues with <15% relative solvent accessibility. Still, this threshold can be modified (or the option turned off) if users have advanced-level access to the server (see below).

Experimental evidence can be used to determine the active residues. For example, in mutagenesis data, residues that, when mutated, result in the loss or reduction of binding are selected as active. In the case of NMR chemical shift perturbation (CSP) data, residues that exhibit significant changes in chemical shifts when the protein is bound to its partner are chosen. However, it is the user's responsibility to interpret the experimental data properly and ensure that the selected residues are on the protein surface and able to make physical contacts.

Experimental data used in HADDOCK can be noisy, leading to false positives in which residues are defined as active although they do not make contact with the partner molecule. To address this, HADDOCK allows for the random removal of a fraction of the restraints (active residues) during the docking process. By default, 50% of the restraints are discarded. HADDOCK performs multiple docking runs with different initial random orientations and sets of discarded restraints. The correct docking solutions are expected to have better scores than those driven by incorrect restraints.

Users with advanced access have more control over the treatment of active and passive residues, being able to modify or disable the random removal of restraints. The percentage of restraints discarded is determined by the number of restraint partitions, which is set to two by default (resulting in a 50% discard rate). However, with advanced access to the server, any percentage of restraints can be discarded by adjusting the number of partitions.

In addition, distance restraint files can be uploaded instead of specifying active and passive residues directly. These files should follow the proper CNS syntax (which is not case sensitive unless a selection is put between double quotes).

Next, to distance (ambiguous or unambiguous) restraints, HADDOCK also supports dihedral angle restraints and various NMR-derived anisotropy restraints. All restraints in CNS consist of one or more atom selections and one or more floating-point values.

A short description of the various restraints supported by the HADDOCK2.4 server is provided in Box 1.

Submission interfaces

The web server provides users with a user-friendly interface for using HADDOCK with the added benefits of pre-processing and validation of the input data, automatic parameter settings depending on the molecule types and docking scenarios, post-processing and analysis of results and access to free computing resources, especially the distributed HTC resources of EGI and the EOSC[40]. Upon registration, users are given access to one of the three available interfaces: Default, File Submission or Refinement.

Default interface (Submit menu). This is the default interface through which users can upload their input data and fine-tune all possible available parameters. The option to fully customize the >500 parameters can be overwhelming, especially to non-expert users. For this reason, we developed two interfaces, the Easy and the Advanced (or Guru) interface, with the former having access to limited options and the latter with full access. Users can easily request Guru access by stating their motivation via their profile (registration) page in the web server.

Users submitting through the Default interface will need to assign a job name to their run and set the number of molecules to dock (≤20). For each molecule, the users should upload a PDB or PDBx/mmCIF structure and specify the type of molecule defined. Currently, the following types of molecules are supported:

- Protein or protein-ligand (e.g., proteins that include a cofactor or other small ligands)
- Ligand (for small-molecule docking)

BOX 1

Restraints

This section illustrates the different type of restraints supported by HADDOCK and their format for use in CNS, the computational engine used by HADDOCK. The provided examples are arbitrary.

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 2 of chain A and residue 10 of chain B should have an effective distance of ≤2 Å. Note that CNS, the computational engine of HADDOCK, uses the segID and not the chainID to distinguish between molecules. segID values are assigned automatically by the HADDOCK server for each submitted molecule. They can, however, be manually modified by the user. In calculating the effective distance, all atoms belonging to residues 2 and 10 will be considered. The example that follows illustrates a more specific distance between two atoms, for example, to form an intermolecular disulfide bridge between the SG atoms of two cysteine residues:

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

Ambiguous restraints are defined by adding 'or' statements within one atom selection. The following example corresponds to the Cys-to-Lys linkage in the nucleosome-modeling procedure described in this article (Procedure 2). It connects the cysteine to two potential candidate lysines to form the linkage:

```
assign (
 (resid 602 and segid A)
 or
 (resid 603 and segid A)
 )
 (resid 85 and segid B) 2.0 2.0 0.0
```

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>.

The following example would restrain the first rotation along the side chain of residue 48 (the rotation around the CA-CB bond) to a value of 180° ± 30°:

```
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

NMR residual dipolar couplings can be defined in two manners in CNS: directly (SANI[57]) or as intervector angle (VEAN[58]) restraints.

In SANI, the residual dipolar coupling restraint is defined between the observed pair of nuclei (e.g., N-H) and an alignment tensor, with its associated axial and rhombic components. The tensor will automatically be 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 991 and 999 and contain 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 N and H atoms and the last two values are the measured residual dipolar coupling (in Hertz) 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 atom pairs for which residual dipolar couplings have been measured) and four numbers (first angle, range, second angle and 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
```

PCS restraint files

PCSs can provide useful information on both the distances and the orientation of the molecules to be docked. They can be used directly as restraints in HADDOCK by using the XPCS energy term[59] in CNS. Note that for this, the standard CNS version needs to be recompiled with the files provided in the cns1.3 directory in the HADDOCK distribution.

For each set of PCS restraints used, a tensor must be defined, together with its axial and rhombic components. Note that the proper units for use in HADDOCK should be: $10^{-28}/(12 \times \pi)$ m^3 (which gives a scaling factor of 265.26 compared to values expressed in 10^{-32} m^3). The position of the tensor also needs to be defined by specifying

(continued from previous page)

distance restraints with respect to the protein to which it is attached. These restraints should be defined by uploading a tensor distance restraint file to the server.

For example, if the tensor is attached to a metal ion with residue number 999, as could be the case for a metallo-protein, the restraint would look like:

```
assi (resid 999 and resn XAN and name OO) (resid 190
and segid A) 0.0 0.0 0.0
```

The distances are thus all set to 0 to keep the tensor on top of the metal ion.

An NMR PCS restraint is defined between the observed nuclei (e.g., N) and the tensor associated with the paramagnetic center. The tensor will automatically be added to the coordinate file by HADDOCK. The proper format for PCS restraints is the following:

```
assign (resid 999 and resname XAN and name OO)
 (resid 999 and resname XAN and name Z)
 (resid 999 and resname XAN and name X)
 (resid 999 and resname XAN and name Y)
 (resid 7 and name N and segid A) -0.119906 0.15
```

where the last two numbers are the PCS value and its associated error, respectively. The restraints should refer to the correct tensor atom name, residue name (XAN) and residue number: they must be numbered between 991 and 999 and contain the atoms OO, Z, X and Y.

Diffusion anisotropy restraint files

Diffusion anisotropy restraints can be used in CNS by using the DANI energy term[60]. The restraint is defined between the T_1/T_2 ratio for the observed pair of nuclei (e.g., N-H) and an alignment tensor. The tensor will automatically be added to the coordinate file by HADDOCK. The restraints should refer to the correct tensor atom name, residue name (DAN) and residue number. The residues must be numbered between 991 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 atom and the last two values are the measured T_1/T_2 ratio and its associated error.

- Peptide (similar in principle to protein, but with peptide-specific docking parameter settings (e.g., increased flexibility))
- Glycan
- Nucleic acids (both DNA and RNA)
- Protein–nucleic acid (a protein–DNA complex (e.g., nucleosome) treated as one molecule)

Furthermore, users can choose several options for each molecule such as the use of coarse graining, defining the molecule as a cyclic peptide, adding charges to the N/C termini or keeping terminal phosphates for nucleic acids and fixing the molecule in its original position. The server supports various post-translational modifications, metal ions and various glycans (see Input structures). Once the users click 'Next', the form and the input data are validated by the back end. The protonation state of each histidine residue is automatically determined with the software *reduce*[41], all-atom structures are converted to coarse-grained representations if this option was selected and parameters and topologies are automatically generated for small molecules by using an in-house version of the software PRODRG[42].

Users are then redirected to the Input parameters tab, where they can access input-specific parameters. Here, they can manually define active (should be at the interaction interface in the final models) and passive (can be at the interaction interface in the final models) residues for each molecule or select those from an interactive sequence display (see Restraints). Buried residues are automatically removed (this can be turned off at the Guru level) from the selection to improve computational efficiency. An option has been added to define the entire surface as passive when no information about the interface for one of the molecules is available (this does, however, require that active residues are defined for the other molecule). An interactive NGL structure viewer[43] allows users to visualize residues defined as active for each molecule. Finally, users can specify cryo-EM restraints and upload an EM density map.

By clicking on 'Next', users are brought to the last submission step, where they can change >500 docking parameters, depending on their access tier. The default parameters are those shown by default in the submission page. These parameters are separated into main categories, with subdivisions. The main categories are:

- *Distance restraints*. This menu is used for uploading user-defined ambiguous and unambiguous distance restraint files, specifying the ab-initio docking setting and defining some restraint-specific parameters
- *Sampling parameters*. This menu is used to change the number of models generated at the various stages of the docking protocol and control some protocol-specific parameters
- *Clustering parameters*. This menu is used to define the method and associated parameters for the clustering of the final models
- *Dihedral and hydrogen bond restraints*. This menu is used to upload and define hydrogen-bond and dihedral angle restraints
- *Non-crystallographic symmetry restraints*. This menu is used to define pairs of non-crystallographic symmetry restraints
- *Symmetry restraints*. This menu is used to define various sets of symmetry restraints (from C2 to C6 symmetries)
- *Restraint energy constants*. This menu is used to define distance and dihedral restraint energy force constants for the various stages of the protocol and also controls when (at which stage) the various distance restraints are used
- *Residual dipolar coupling restraints*. This menu is used to upload and define NMR residual dipolar coupling restraints and their associated energy function and parameters
- *Pseudo contact shift (PCS) restraints*. This menu is used to upload and define PCS restraints and their associated parameters and force constants
- *Relaxation anisotropy restraints*, *energy and interaction parameters*. This menu is used to upload and define relaxation anisotropy restraints and their associated parameters and force constants
- *Scoring parameters*. This menu is used to define the weights of the various terms contributing to the HADDOCK scoring function
- *Advanced sampling parameters*. This menu is used to change parameters associated with the various sampling and refinement stages of the docking protocol, defining, for example, the simulated annealing protocol used in the flexible refinement stage
- *Solvated docking parameters*. This menu is used to change specific parameters associated with the solvated docking mode of HADDOCK. Solvated docking can be activated under the Sampling parameter menu
- *Analysis parameters*. This menu is used to define the type of analysis performed by HADDOCK, the default being cluster analysis only. The full analysis will generate several additional files that are accessible only by downloading the archive of the full run. For details, refer to https://www.bonvinlab.org/software/haddock2.4/analysis/

Once the parameters are edited, users can download the input files (processed by HADDOCK), submit their run, download the parameter file as a Java Script Object Notation–formatted text file or clear the form. It is recommended to save the parameter file, because it provides a complete description of all parameter settings and input data (except the cryo-EM map, which is too large to be included in it) submitted to the server. This file can be re-used to repeat a run (via the File submission interface; see below), edited to modify a few parameters and submit a new run or provided as supporting information in a publication.

Errors or inconsistencies detected by the server validation machinery are reported on top of the page in red boxes, and submission is disabled until those have been resolved. Common error messages and their meaning can be found in the FAQ link (https://wenmr.science.uu.nl/haddock2.4/faq).

File submission interface (Submit File menu). This interface allows users to upload the parameter file containing all simulation parameters and input data via a simplified form that requires the user only to define a run name and upload the parameter file.

Refinement interface (Refinement menu). Besides molecular docking, HADDOCK is also capable of performing structure refinement (e.g., to remove clashes at an interface or complete a model with missing side chains). This can be done via the default interface, which requires modifying

several parameters (instructions are available at https://www.bonvinlab.org/software/haddock2.4/tips/advanced_refinement/). However, the server offers a simplified and dedicated interface with predefined parameters specific only to refinement, which increases the usability and accessibility of this protocol. Various refinement protocols can be selected from a menu, allowing for simple energy-minimization protocols to more complex simulated annealing with centroid restraints to refine low- to medium-resolution cryo-EM models[38]. The refinement interface supports both single molecules (i.e., not a complex) and complexes. Users can input those as single conformation or as an ensemble of conformations in PDB format (by using the MODEL/ENDMDL statements). The server accepts any type of molecule supported by HADDOCK as well as combinations of those types.

Workspace and help pages

The server also provides a workspace where users (when logged in) can view their submitted runs and their status (which is color coded). The workspace allows users to access the results pages, download the corresponding parameter file, download the top four models of the top 10 clusters or delete the run from the server. Note that all runs are automatically deleted 2 weeks after completion, and it is the user's responsibility to save their data.

A help page is also provided, and it provides links to the online HADDOCK2.4 software manual (https://www.bonvinlab.org/software/haddock2.4/), the best practice guide (https://www.bonvinlab.org/software/bpg/), the support forum and a server output example (https://www.bonvinlab.org/education/HADDOCK24/).

Result example page

The server provides a link to an example result page for a complex modeled by using NMR CSP data. Details of this modeling can be found in the associated tutorial at https://www.bonvinlab.org/education/HADDOCK24/HADDOCK24-protein-protein-basic/.

The first section of the result page (Fig. 3a) reports the number of clusters (a cluster represents a set of similar models defined on the basis of either the similarity of contacts or the positional root-mean-square deviations) and related statistics (HADDOCK score, cluster size, root mean square deviation (RMSD) from the top ranked model, energies and buried surface area) of the top 10 clusters. The top 10 clusters are selected and sorted according to the average HADDOCK score of the best four models of each cluster, from the lowest (best) HADDOCK score to the highest (worst). The HADDOCK score is calculated as:

$$\mathrm{HADDOCK_{score}} = 1.0 \times \mathrm{E_{vdw}} + 0.2 \times \mathrm{E_{elec}} + 1.0 \times \mathrm{E_{desol}} + 0.1 \times \mathrm{E_{air}}$$

where E_{vdw} is the intermolecular van der Waals energy and E_{elec} is the intermolecular electrostatic energy calculated with the optimized potentials for liquid simulations force field[44]. E_{desol} represents an empirical desolvation energy term adapted from previous studies[45], which estimates the energetic gain or penalty of burying specific side chains upon complex formation, and E_{air} represents the distance restraints energy. The intermolecular energies are calculated by using an 8.5-Å cutoff and the optimized potentials for liquid simulations non-bonded parameters.

The name of the clusters is defined by their size, with the largest cluster numbered as 1. The corresponding models can be visualised online by clicking on the eye icon (Fig. 3b) or downloaded for further analysis.

The bottom of the page presents some interactive graphical representation of the results (Fig. 3c). The plots show the distribution of various measures (including the HADDOCK score, van der Waals energy and other terms) as a function of the fraction of common contact or interface RMSD from the best generated model (the best scoring model). The models are color coded according to the cluster to which they belong. Finally, the distribution of components of the HADDOCK score (E_{vdw}, E_{elec} and E_{desol}) for the various clusters can be visualized in the cluster analysis section (Fig. 3d).

In case the scores of various clusters are within 1 s.d. of each other, all should be considered as a valid solution for the docking. Ideally, some additional independent experimental information should be available to decide on the best solution.

Execution timing

Many factors affect the run time of a docking run, among which the most important ones are the size of the systems to be modeled (both the number of molecules to be docked and the number of residues), the definition of restraints (both in terms of their number and importantly the

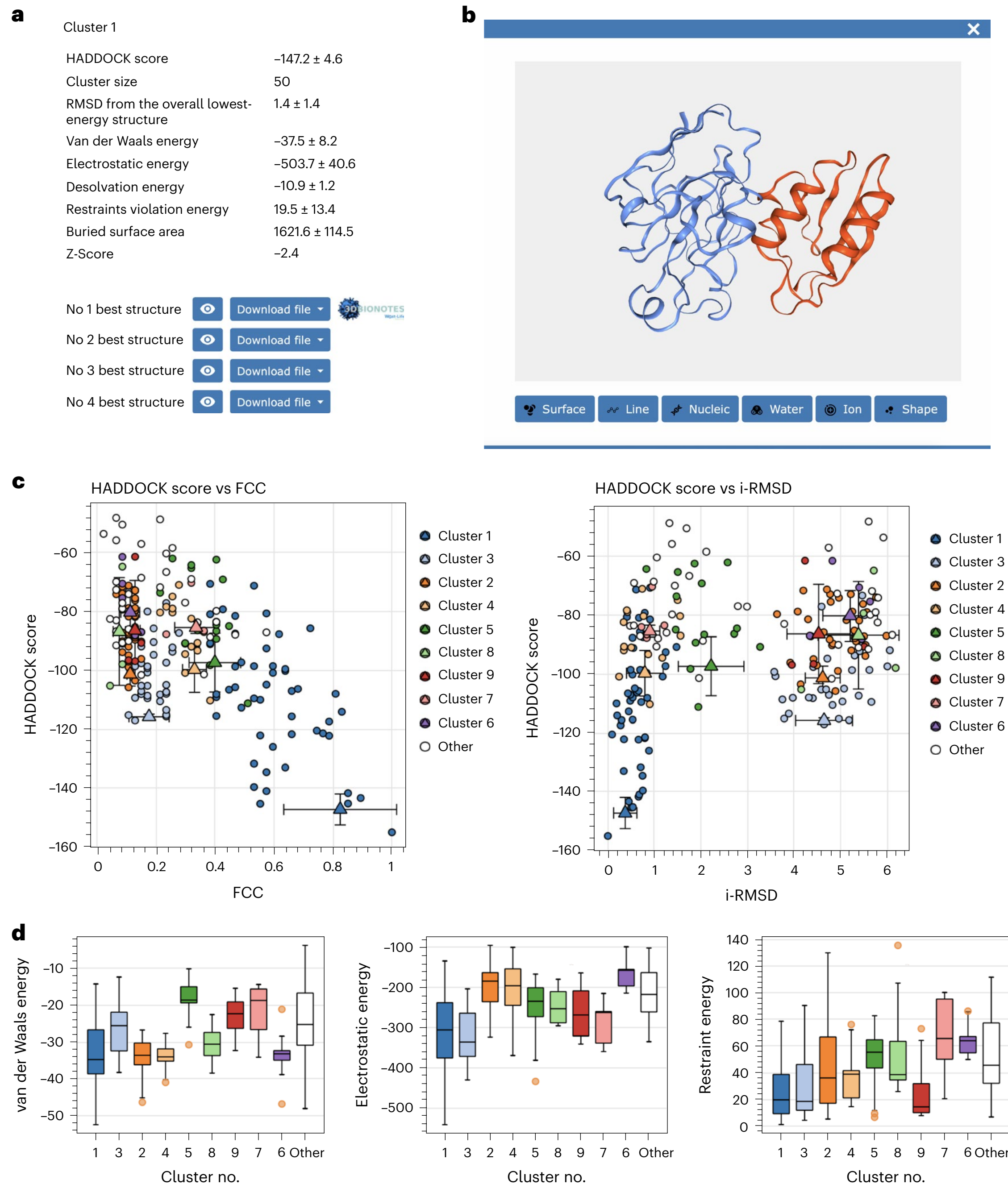


Fig. 3 | Overview of the HADDOCK2.4 result page. a, Cluster 1 statistics. **b**, Online visualization of models. **c**, Model analysis to assess the convergence of the calculations and to visualize the various models and their clustering. The best model is taken as reference for fraction of common contacts (FCC, on the left)) and the interface root mean square deviation (i-RMSD) (on the right) calculations. As can be seen in **c**, cluster 1 also contains the best overall scoring model. **d**, Cluster analysis showing the distributions of various score components of the various clusters, allowing comparison of clusters based on specific energy terms. In this particular example, six clusters were obtained, with cluster 1 (**a**) being the most populated and best-scoring cluster based on the HADDOCK score. The second best cluster (not shown) has a HADDOCK score of −120.5 ± 5.3 a.u., which makes cluster 1 the clear winner. These data are taken from a pre-calculated run for this antibody–antigen protocol available at https://wenmr.science.uu.nl/haddock2.4/result/4242424242/264063-Antibody-Antigen-NMR.

degree of ambiguity in those (i.e., the number of possible atom-atom distances defined by the selections)), the number of models generated and other parameter settings such as the number of refinement steps for the various stages of the protocol. Furthermore, distributing the computation onto worldwide HTC resources also comes with some overhead in terms of timing. Large systems are usually run on our local HPC resources. All those factors make it difficult to predict the wall clock run time for a given system. To give an estimate of timings, we have analyzed the run time from starting a run (i.e., not counting the queued time when the server is very busy) to completion over the years 2020–2023. The resulting distribution is shown in Supplementary Fig. 1. Some runs can complete in less than half an hour, whereas others can take several days. In addition to some of the factors listed above, hardware and middleware problems can also cause delays. Overall, however, the HADDOCK server has been in continuous operation since its early days and has handled over half a million user submissions since its launch in 2008, with ~50% of those processed by the HADDOCK2.4 server.

Materials

Equipment

- Experimental or theoretical information about the biomolecular complex to be modeled (see Equipment setup)
- A registered account for the HADDOCK2.4 server with Guru level access (see Equipment setup)

Software

- HADDOCK server (https://wenmr.science.uu.nl/haddock2.4)
- Python 3+ (https://www.python.org/)

Hardware

- A computer with internet access, a modern graphical web browser (tested on Chrome, Safari, Edge and Firefox) and a MacOS/Linux/Windows subsystem for Linux system terminal

Equipment setup

Models of the biomolecules

The procedure requires atomic-resolution experimental or high-quality models of the biomolecules to be docked in PDB/CIF format, preferably with unnecessary components removed (e.g., a small co-crystalized molecule from the crystallization buffer or disordered regions from AlphaFold2 models).

HADDOCK account

A registered account for the HADDOCK2.4 server with Guru level access is needed. To register an account, users should:

1. Access the registration page at https://wenmr.science.uu.nl/auth/register/.
2. Confirm their email address so that they can be notified once the account has been activated.
3. Log into their registration page and request elevated access.

Phyton installation

To download and edit PDB files during the pre-processing step (Step 1), install pdb-tools as follows:

1. Download Python3+ from https://www.python.org/.
2. Install the software by following the instructions provided in Python's documentation (https://www.python.org/doc/).
3. Install pdb-tools via the python package manager. With python properly configured, type '`pip install pdb-tools`' in the terminal.

Procedure 1: modeling antibody–antigen complexes with the HADDOCK web server by using a priori NMR information

CRITICAL Procedure 1 demonstrates the usage of the HADDOCK web server to predict the structure of an antibody–antigen complex by using NMR CSP data to identify the antigen's epitope (binding site) and provide knowledge of the hypervariable loops of the antibody to guide the docking. The components of the complex are IL-1β (PDB ID 4I1B [46]), the antigen and its highly specific monoclonal antibody gevokizumab (PDB ID 4G6K). The objective is to recreate the complex whose structure is known (PDB ID 4G6M [47]) from the unbound components.

Antibody and antigen structure pre-processing

CRITICAL This pre-processing step can be performed by using pdb-tools[48], which are a set of Python scripts to edit PDB files. We describe the use of pdb-tools from the terminal command line, but it can also be accessed as a web service at https://wenmr.science.uu.nl/pdbtools/.

CRITICAL Antibodies consist of two chains, light (L) and heavy (H). The corresponding PDB file of each chain needs to be pre-processed for use in HADDOCK for the molecule to be defined as a single chain.

CRITICAL This section results in two new files. We have done this experiment already for the example antibody and have made these files available on GitHub. These can be used directly in Step 6 (Box 2).

1. Open a Linux terminal to make use of pdb-tools.

CRITICAL STEP The pdb-tools will be used to prepare the antibody: removing hetero atoms, correcting the numbering of possible insertions in the hypervariable loops, assigning a unique chainID and shifting the residue numbering of the light chain to avoid overlap with the heavy chain.

1. Download and prepare chain H (copy and paste in your terminal the full gray block):

```
pdb_fetch 4G6K | pdb_tidy -strict | pdb_selchain -H | pdb_delhetatm |\
pdb_fixinsert | pdb_keepcoord >4G6K_H.pdb
```

CRITICAL STEP It is important to select the first chain (`pdb_selchain`), remove the hetero atoms (`pdb_delhetam`) and correct possible insertions in the hypervariable loops (`pdb_fixinsert`).

TROUBLESHOOTING

3. Download and prepare chain L:

```
pdb_fetch 4G6K | pdb_tidy -strict | pdb_selchain -L | pdb_delhetatm |\
pdb_fixinsert | pdb_shiftres -1000 | pdb_keepcoord >4G6K_L.pdb
```

BOX 2

Data required and generated in Procedure 1

All data required for this procedure are available at https://github.com/haddocking/haddock24-protocol, including the pre-processed input molecules for this protocol, which can be downloaded with the following commands:

`wget` https://github.com/haddocking/haddock24-protocol/raw/main/antibody-antigen/4G6K_clean.pdb

`wget` https://github.com/haddocking/haddock24-protocol/raw/main/antibody-antigen/4I1B_clean.pdb

- *4G6K_clean.pdb*. This is the PDB file of the unbound (free) form of the antibody with the two chains defined as a single chain and with residues renumbered to avoid overlap in numbering between the chains. The structure was further truncated to only keep the two domains involved in binding (to save computational time).
- *4I1B_clean.pdb*. This is the PDB file of the unbound (free) form of the antigen, without heteroatoms.

CRITICAL STEP It is important to shift the residue numbering of the light chain to avoid overlap with the heavy chain (`pdb_shiftres`). In this example, the `pdb_shiftres` command shifts the residue numbering by 1,000, which makes it easier to find the original numbering. If the first chain would have >1,000 residues, this value must be increased to avoid overlap in numbering.

4. Merge the two pre-processed chains:

```
pdb_merge 4G6K_H.pdb 4G6K_L.pdb | pdb_chain -A | pdb_chainxseg |\
pdb_tidy -strict >4G6K_clean.pdb
```

CRITICAL STEP Because the two chains of the antibody were processed separately, they should be merged into one PDB file to represent the full antibody.

5. Prepare the antigen

```
pdb_fetch 4I1B | pdb_tidy -strict | pdb_delhetatm | pdb_keepcoord
>4I1B_clean.pdb
```

Data input

CRITICAL Change only the parameters described in this protocol; leave all others to their default value.

6. Open the HADDOCK 2.4 default submission interface (https://wenmr.science.uu.nl/haddock2.4/submit) and refer to the screenshots of the various steps shown in Extended Data Figs. 1–5.
7. Define a name for your docking run in the field Job name (item 7 in Extended Data Figure 1): 'Antibody-Antigen-NMR'.
8. Select the number of molecules to dock (item 8 in Extended Data Figure 1): '2' (default).
9. Input the first protein PDB file (the pre-processed unbound free form of the antibody (item 9 in Extended Data Figure 1): 'Molecule 1 – input > 4G6K_clean.pdb'.
10. Input the second protein PDB file (the pre-processed unbound free form of the antigen) (item 10 in Extended Data Figure 2): 'Molecule 2 – input > 4I1B_clean.pdb'.
11. Click on the 'Next' button at the bottom left of the interface. This will upload the pre-processed structures from Steps 1–5 to the HADDOCK server, where they will be processed and checked for formatting errors (item 11 in Extended Data Figure 3).

TROUBLESHOOTING

Defining restraints to guide the docking

▲ **CRITICAL** Information used to define the restraints will typically come from the literature, experimental results or bioinformatics predictions. In this example, we describe how users can define the active residues in HADDOCK (i.e., the residues that are expected to participate in the interaction) on the basis of the knowledge of the hypervariable loops for the antibody and a priori NMR information for the antigen. Users should refer to the article describing the crystal structure of the antibody–antigen complex, which also includes a report of the experimental NMR chemical shift titration experiments obtained to map the binding site (epitope) of the antibody on IL-1β[47]. This information must be mapped to the renumbered antigen to identify the experimentally determined epitope of IL-1β; these residues will be defined as active in HADDOCK—residues that are expected to participate in the interaction.

12. Specify the active residues for the first molecule (antibody). To do this, open the 'Molecule1 – parameters' drop-down menu (if not already opened) and manually enter the active residues (in this case, the hypervariable loop residues) as a comma-separated list, making sure to follow the numbering of the pre-processed antibody file):

```
Active residues > 31,32,33,34,35,52,54,55,56,100,101,102,103,104,105,1
06,1031, 1032,1049,1050,1053,1091,1092,1093,1094,1096
```

To visualise the defined active residues online, click on the 'Visualize residues' button (item 12 in Extended Data Figure 3).

13. The server allows the user to define 'passive' residues automatically as the surface neighbors of an active residue. In the case of antibodies, it does not make sense to define surface neighbors as passive, because only hypervariable loop residues are typically involved in the interaction. It is therefore important to turn off the automatic definition of passive residues for the first molecule (antibody). To do this, toggle off the option 'Automatically define passive residues around the active residues' (item 13 in Extended Data Figure 3).
14. Specify active residues for the second molecule (the NMR-identified epitope residues of the antigen) (item 14 in Extended Data Figure 4): 'Molecule2 – parameters'.

```
Active residues > 72,73,74,75,81,83,84,89,90,92,94,96,97,98,115,116,117
```

15. (Optional—alternative to Step 14) If NMR data to define the epitope are not available, the server offers the option to automatically select all solvent-accessible residues as passive. To do this, toggle on the option 'Automatically define surface residues as passive' (Extended Data Figure 4).
16. Keep the 'Automatically define passive residues around the active residues' setting 'on'. For the antigen, it is advisable to use the definition of passive residues, because experiments might not always detect the full binding site. As such, increasing the putative binding interface by defining the surface neighbors (passive residues) of active residues usually helps in getting better docking results.
17. Click the 'Next' button at the bottom left of the interface (item 15 in Extended Data Figure 2).

Docking parameters

18. (Optional) You are now in the 'Docking parameters' tab (Extended Data Figure 5). Open the 'Sampling parameters' menu and consider whether any docking parameters should be changed (not needed here).
19. Many HADDOCK parameters can be modified in this tab. Information about optimal parameters for various setups can be found at the docking section of the HADDOCK Best Practice Guide (https://www.bonvinlab.org/software/bpg/#docking-with-haddock). In the specific example described in this procedure, the current default values are used (no changes needed).

Submission

20. Click the 'Submit' button at the bottom left of the interface (item 20 in Extended Data Figure 5). This step will redirect users to a 'Status' page.
21. From the 'Status' page, you can download the run parameter file and check the status of the run. This page is automatically refreshed. Once the simulation has finished, users will be automatically redirected to the 'Result' page and will be notified by e-mail. This page can be bookmarked for quick reference. The exact run time will depend on the load on the server and on the distributed HTC resources used for the computations. When you get the email, follow the instructions to see your results (and refer to Anticipated results).

Procedure 2: coarse-grained protein docking on a nucleosome particle: PRC1 ubiquitination module bound to the nucleosome

CRITICAL Procedure 2 demonstrates the modeling of PRC1, which is responsible for the ubiquitylation of the nucleosomal histone (H2A) regulating developmental processes[49,50]. The interaction between PRC1 and histone H2A was explored in a previous publication[35] and is used here as an example of how to use coarse-grained atom representation for docking in HADDOCK. The procedure demonstrates how to use mutagenesis data as active residues and how to define specific unambiguous restraints between residues that are known to participate in the interaction.

The HADDOCK 2.4 default submission interface is used (https://wenmr.science.uu.nl/haddock2.4/submit).

Preparing the PDB file for docking

CRITICAL The molecules are prepared for docking with pdb-tools. The following commands should be executed in a terminal window, under the bash shell.

CRITICAL This section will result in two new files ready for docking. These have already been pre-processed and are made available on GitHub. These can be used directly in the Data Input part of the procedure (starting at Step 6) (see Box 3 for details about the files).

1. Open a Linux terminal to make use of the command line of pdb-tools.
2. Download the nucleosome structure PDB ID 4R8P (`pdb_fetch`) and prepare the structure for docking, keeping only the relevant chains (deleting chains K, L, M and N) (`pdb_delchain`) and shifting their residue numbering by increments of 250 to avoid overlap (`pdb_reres`) and assigning them a unique chainID (`pdb_chain`). In this example, the residue numbering is shifted by increments of 250 because each of the chain contains <250 residues. All this can be done with pdb-tools by copying and pasting the following text into the command line.

```
pdb_fetch 4r8p | pdb_tidy -strict | pdb_delhetatm | pdb_keepcoord |\
 pdb_delchain -K,L,M,N >_temp.pdb
pdb_splitchain _temp.pdb
counter=0
cat /dev/null > 4r8p_clean.pdb
chains=(A B C D E F G H I J)
for i in "${chains[@]}"; do
 filename="_temp_${i}.pdb"
 pdb_reres -"${counter}" "${filename}" » 4r8p_clean.pdb
 rm -f "${filename}"
 counter=$((counter + 250))
done
pdb_chain -A 4r8p_clean.pdb >_temp.pdb
mv -f _temp.pdb 4r8p_clean.pdb
```

CRITICAL STEP Each molecule for docking purpose in HADDOCK must consist of a single chain with non-overlapping reside numbers.

TROUBLESHOOTING

BOX 3

Data required and generated in Procedure 2

The pre-processed molecules, restraints needed for this procedure and list of solvent-accessible residues for the histones can be downloaded from github.com/haddocking/haddock24-protocol:

wget https://github.com/haddocking/haddock24-protocol/raw/main/nucleosome/4r8p_clean.pdb
wget https://github.com/haddocking/haddock24-protocol/raw/main/nucleosome/3rpg_clean.pdb
wget https://github.com/haddocking/haddock24-protocol/raw/main/nucleosome/lys-cys-linkage.tbl
wget https://github.com/haddocking/haddock24-protocol/raw/main/nucleosome/histones-passive.list

- *4r8p_clean.pdb*. The raw structure represents the bound form between PRC1 and the nucleosome; four PRC1 chains (K, L, M and N) were removed, and because the nucleosome will be input as one molecule, the remaining chains were renamed by using a unique ID (A), and the residues were renumbered.
- *3rpg_clean.pdb*. This is the unbound form of PRC1 without heteroatoms and defined as chain B.
- *lys-cys-linkage.tbl*. This is the distance restraint file that follows CNS syntax (Box 2) and will be used in Step 3 of the submission.
- *histones-passive.list*. This is a text file containing the comma-separated list of solvent-accessible residues for the histone proteins that will be used to define the passive residues.

3. Download the PRC1 structure (PDB ID 3RPG) and prepare it for docking by using pdb-tools.

```
pdb_fetch 3rpg | pdb_tidy -strict | pdb_delhetatm | pdb_keepcoord |
pdb_reres --1 |\
pdb_chain -B >3rpg_clean.pdb
```

CRITICAL STEP Note the '--1' in the `pdb_reres` command. This means that we are starting the numbering of the molecule at −1 (to match the reference complex).

Preparing the restraints to guide the docking

CRITICAL The information to guide the docking consists of four residues on PRC1, previously identified by mutagenesis to be important for the binding (residue numbers 207, 209, 328 and 329 based on the numbering on the docking-ready PDB file of PCR1—these will be input as active in Step 18), and the knowledge that, in particular, Cys85 of PRC1 forms a covalent bond with either Lys118 or Lys119 (Lys602, Lys603 after renumbering) of histone H2A[35]. The definition of this covalent bond illustrates the use of user-defined distance restraints (the Cys-to-Lys-specific restraint).

CRITICAL STEP When defining restraints it is important the residue numbering and chain ID (segid) are matching those in the input PDB files.

4. Create the unambiguous restraint for the covalent bond between Cys85 of PRC1 and Lys119 or Lys118 (Lys602, Lys603 after renumbering) of H2A[49,51]. For this example, type in the command line the following text to save the restraint in the 'lys-cys-linkage.tbl' restraint file.

```
cat «EOT > lys-cys-linkage.tbl
assign (resid 85 and segid B)
(
(resid 602 and segid A)
or
(resid 603 and segid A)
)
2.0 2.0 0.0
EOT
```

5. Identify the solvent-accessible residues of the histone proteins. In this example, the four residues identified by mutagenesis on PCR1 should interact with residues on the histones on the same side of the nucleosome that the two lysines (Lys 119 and Lys 118) of H2A are located. In the absence of more specific information, it makes sense to define the entire solvent-accessible surface of the histones on the side of the nucleosome where the two lysines of H2A are located as passive residues. The corresponding list of passive residues, which was obtained by manually selecting the solvent-accessible histone residues on the side of the nucleosome where the H2A lysines are located, is provided in the 'histones-passive.txt' file.

Data input

CRITICAL Change only the parameters described in this procedure; leave all others to their default value.

6. Open the HADDOCK 2.4 default submission interface at https://wenmr.science.uu.nl/haddock2.4/submit.
7. Define a name for your docking run in the field Job name: 'Nucleosome-Docking-CG'.
8. Select the number of molecules to dock: '2' (default).
9. Input the nucleosome as the first protein PDB file: 'Molecule 1 – input > 4r8p_clean.pdb'.
10. Under 'What kind of molecule are you docking', change the kind of molecule by selecting 'Protein-Nucleic acid'.
11. Under 'Do you want to coarse-grain your molecule', select 'Toggle to coarse-grain your molecule'.

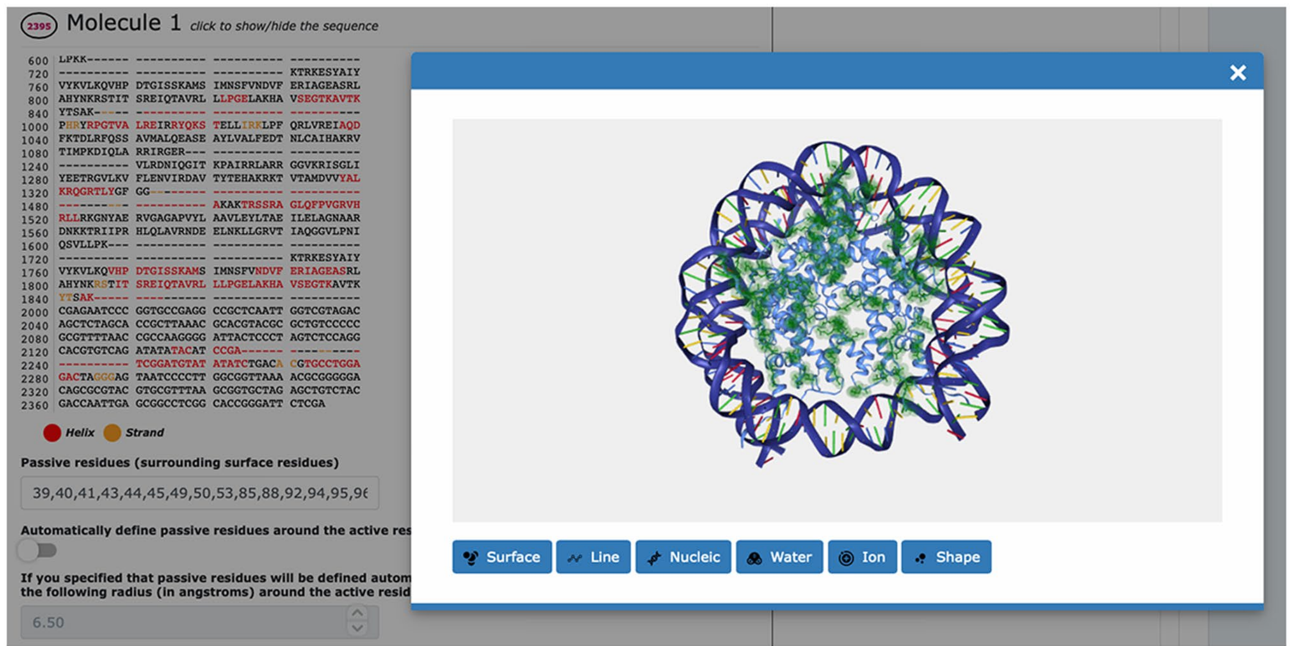


Fig. 4 | Online visualization of active/passive residues. View of the online visualization of the passive residues (shown in green dots) defined for the nucleosome.

12. Input PCR1 as the second protein PDB file: 'Molecule 2 – input > 3rpg_clean.pdb'.
13. Click on the 'Next' button at the bottom left of the interface. This will upload the structures to the HADDOCK server, where they will be processed and checked for formatting errors.

TROUBLESHOOTING

Defining restraints to guide the docking

CRITICAL Here, we use a mix of active-passive residue definition (ambiguous restraints) and specific unambiguous distance restraints. To define the active-passive regions to guide the docking, follow Steps 14–19.

14. Leave the list of active residues empty for the first molecule (nucleosome) in 'Molecule 1 – Parameters'.
15. Turn off the automatic definition of passive residues (surface neighbors of the active one) for the first molecule (nucleosome) to manually define the histone solvent-accessible residues on one side of the nucleosome. For this, toggle off the option 'Automatically define passive residues around the active residues'.
16. Define the passive residues of the nucleosome:
    Passive Residues > 39,40,41,43,44,45,49,50,53,85,88,92,94,95,96,97,250,251,252,253,254,256,257,260,261,264,281,282,285,289,292,293,296,297,300,309,500,503,506,529,530,548,552,555,556,557,558,560,572,573,575,578,579,593,594,595,597,598,600,601,602,603,762,766,767,769,770,775,779,809,810,811,814,818,823,824,827,828,831,832,834,835,838,839,841,842,843,844,1014,1018,1077
17. Visualize the passive residues to check that what was selected is on one side of the nucleosome only by clicking on 'Visualize residues'. Figure 4 shows the corresponding view.
18. Define the active residues of PRC-1 as follows in 'Molecule 2 – Parameters':
    Active Residues > 207,209,328,329
19. Click 'Next'. An alert message will appear on the top of the next page to notify you that some parameter settings have been automatically optimized from default values. This is

done to ensure that the best settings for the type of molecules being docked are used and, in this case, the use of coarse graining. The optimized values are based on our publications and benchmarks and can be viewed in the 'Settings Page' at https://wenmr.science.uu.nl/haddock2.4/settings#optimal.

Docking parameters

20. Input the unambiguous restraints manually generated. You can supply a HADDOCK restraints TBL file with restraints that will always be enforced (unambiguous restraints):> lys-cys-linkage.tbl

Submission

CRITICAL Because we are using the default parameter settings (including those automatically optimized for this particular setup), we can directly proceed with the submission.

21. Click the 'Submit' button at the bottom left of the interface. This step will redirect users to a 'Status' page.
22. From the 'Status' page, you can download the run parameter file and check the status of the run. This page is automatically refreshed. Once the simulation has finished, users will be automatically redirected to the 'Result' page and will be notified by e-mail. This page can be bookmarked for quick reference.
23. The exact run time will depend on the load on the server and on the distributed HTC resources used for the computations. When you get the email, follow the instructions to see your results (and refer to nticipated results). Once the job has finished, you will be redirected to the 'Results' page. Refer to Anticipated results for a description of its content.

Troubleshooting

Troubleshooting advice can be found in Table 2.

Table 2 | Troubleshooting table

Step	Problem	Possible reason	Possible solution
2 (Procedures 1 and 2)	`pdb_xxxx` command not found	pdb-tools is not properly installed	Review the installation instructions under the software section
	Command not found	Not running the commands in a Bash terminal	Make sure that you are using Bash
11 (Procedure 1), 13 (Procedure 2)	'There was an error processing your PDB'	Error when preparing the input structures	Carefully perform again all steps involving pdb-tools and also check https://wenmr.science.uu.nl/haddock2.4/faq for possible error causes
Anticipated results	`wget` command not found	Under MacOSX, `wget` might not be installed by default	Use the `curl` command instead of the `wget` command. When using the `curl` command, add the argument -o and the filename to which the standard output should be written (i.e., either 4G6K_clean.pdb or 4I1B_clean.pdb)

Timing

The procedures shown here use a web service for running the docking and the command line for processing the input data. Therefore, all commands given at the command line in the Linux terminal should have an instantaneous execution. The execution of the two docking runs by the server can take from half an hour to several hours in Procedure 1 and a few hours to a few days in Procedure 2. See Supplementary Fig. 1 for more information.

Anticipated results

Modeling antibody–antigen complexes

The computing time for this particular antibody–antigen docking protocol, based on the knowledge of the hypervariable loops and NMR mapping of the epitope on the antigen, was 20 min, running exclusively on our local server resources. Our HADDOCK2.4 run resulted in six clusters (corresponding to 191 of the 200 generated models). The top-ranked cluster (cluster 1) is significantly better in terms of HADDOCK score than the second ranked one: −148.3 ± 2.4 versus −120.5 ± 5.3 a.u. A comparison with the known experimental crystal structure of this complex (PDB ID 4G6M[52]) reveals high similarity (average interface RMSD of 1.65 ± 0.47 Å and DockQ score[53] of 0.68 ± 0.13 for the top four models of cluster 1).

Note that the results might show slight numerical differences with respect to the user run, because the computations are distributed on HTC resources with different hardware, which affects the numerical precision of the computations.

To facilitate the comparison with the experimental structure of this complex, a clean version of the reference complex (4G6M), processed in a similar way as the antibody for docking, is available from https://github.com/haddocking/haddock24-protocol. It can be downloaded with the following command:

`wget` https://github.com/haddocking/haddock24-protocol/raw/main/antibody-antigen/4G6M_clean.pdb

Refer to the Troubleshooting section for more detail.

Nucleosome coarse-grained docking

The nucleosome coarse-grained docking example presented here is based on a few mutagenesis data and the knowledge of a Cys-to-Lys linkage. Our HADDOCK2.4 run resulted in seven clusters (corresponding to 184 out of the 200 generated models). The example run is available at https://wenmr.science.uu.nl/haddock2.4/result/4242424242/264564-Nucleosome-Docking-CG. As an indication of computing time requirements, this run (the modeling of a complex of 1,402 residues (proteins + nucleic acid)) was completed in 76 min. Note that, when using the coarse-graining option, the final models are atomistic ones, because one of the final steps in the protocol is the morphing of the atomistic models of the starting structures onto the coarse-grained models of the complex by following a distance-restrained simulated annealing protocol (for details about how this works, refer to ref. 33).

When you look at the result page, you will notice that the top-ranked cluster (cluster 1) has a substantially better (lower) HADDOCK score than the second ranked one: when we performed the procedure, the result was −162.2 ± 5.2 versus −134.7 ± 11.5 a.u.

You should expect differences with your own run, because the nature of the computations and differences in hardware on which they are running will cause numerical differences. As such, both the scores and the clustering might differ. The top four cluster models in the example run have an average interface RMSD of 2.88 ± 0.06 Å and an average DockQ score of 0.42 ± 0.01.

To facilitate comparisons, a clean version of the reference complex, processed in a similar way as we did above for the nucleosome, is available from https://github.com/haddocking/haddock24-protocol/nucleosome. It can be downloaded with the following command:

`wget` https://github.com/haddocking/haddock24-protocol/raw/main/nucleosome/4r8p_reference.pdb

Refer to the Troubleshooting section for more detail.

Data availability

All data used in the procedures are available at a dedicated GitHub repository[54]: https://github.com/haddocking/haddock24-protocol.

Code availability

pdb-tools is available at https://github.com/haddocking/pdb-tools (Apache 2.0)[55], and the HADDOCK2.4 web service application (used in the described protocols) is available at https://wenmr.science.uu.nl/haddock2.4.

References

1. van Noort, C. W., Honorato, R. V. & Bonvin, A. M. J. J. Information-driven modeling of biomolecular complexes. *Curr. Opin. Struct. Biol.* **70**, 70–77 (2021).
2. Masrati, G. et al. Integrative structural biology in the era of accurate structure prediction. *J. Mol. Biol.* **433**, 167127 (2021).
3. Rout, M. P. & Sali, A. Principles for integrative structural biology studies. *Cell* **177**, 1384–1403 (2019).
4. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
5. Perrakis, A. & Sixma, T. K. AI revolutions in biology. *EMBO Rep.* **22**, e54046 (2021).
6. Li, S. et al. Recent advances in predicting protein–protein interactions with the aid of artificial intelligence algorithms. *Curr. Opin. Struct. Biol.* **73**, 102344 (2022).
7. Varadi, M. et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res.* **50**, D439–D444 (2021).
8. The UniProt Consortium. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Res.* **51**, D523–D531 (2022).
9. Vakser, I. A. & Deeds, E. J. Computational approaches to macromolecular interactions in the cell. *Curr. Opin. Struct. Biol.* **55**, 59–65 (2019).
10. wwPDB Consortium. Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Res.* **47**, D520–D528 (2018).
11. Sali, A. et al. Outcome of the First wwPDB Hybrid/Integrative Methods Task Force Workshop. *Structure* **23**, 1156–1167 (2015).
12. Vallat, B. et al. New system for archiving integrative structures. *Acta Crystallogr. D. Struct. Biol.* **77**, 1486–1496 (2021).
13. Saponaro, A., Maione, V., Bonvin, A. & Cantini, F. Understanding docking complexes of macromolecules using HADDOCK: the synergy between experimental data and computations. *Bio-Protoc.* **10**, e3793 (2020).
14. Roel-Touris, J., Jiménez-García, B. & Bonvin, A. M. J. J. Integrative modeling of membrane-associated protein assemblies. *Nat. Commun.* **11**, 6210 (2020).
15. Ambrosetti, F., Jandova, Z. & Bonvin, A. M. J. J. Information-driven antibody–antigen modelling with HADDOCK. In: *Methods in Molecular Biology* Vol. 2552 (eds Tsumoto, K. & Kuroda, D.) 267–282 (Humana, 2023).
16. Ambrosetti, F., Jiménez-García, B., Roel-Touris, J. & Bonvin, A. M. J. J. Modeling antibody-antigen complexes by information-driven docking. *Structure* **28**, 119–129.e2 (2020).
17. Trellet, M., van Zundert, G. & Bonvin, A. M. J. J. Structural bioinformatics, methods and protocols. *Methods Mol. Biol.* **2112**, 145–162 (2020).
18. Koukos, P. I. & Bonvin, A. M. J. J. Integrative modelling of biomolecular complexes. *J. Mol. Biol.* **432**, 2861–2881 (2020).
19. Rosell, M. & Fernández-Recio, J. Docking approaches for modeling multi-molecular assemblies. *Curr. Opin. Struct. Biol.* **64**, 59–65 (2020).
20. Roel-Touris, J., Bonvin, A. M. J. J. & Jiménez-García, B. LightDock goes information-driven. *Bioinformatics* **36**, 950–952 (2019).
21. Xia, B., Vajda, S. & Kozakov, D. Accounting for pairwise distance restraints in FFT-based protein–protein docking. *Bioinformatics* **32**, 3342–3344 (2016).
22. Echartea, M. E. R., Ritchie, D. W. & de Beauchêne, I. C. Using restraints in EROS-DOCK improves model quality in pairwise and multicomponent protein docking. *Proteins* **88**, 1121–1128 (2020).
23. 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).
24. Tsaban, T. et al. Harnessing protein folding neural networks for peptide–protein docking. *Nat. Commun.* **13**, 176 (2022).
25. Ghani, U. et al. Improved docking of protein models by a combination of Alphafold2 and ClusPro. Preprint at *bioRxiv* https://doi.org/10.1101/2021.09.07.459290 (2022).
26. Evans, R. et al. Protein complex prediction with AlphaFold-Multimer. Preprint at *bioRxiv* https://doi.org/10.1101/2021.10.04.463034 (2022).
27. Bryant, P. et al. Predicting the structure of large protein complexes using AlphaFold and Monte Carlo tree search. *Nat. Commun.* **13**, 6028 (2022).
28. Burke, D. F. et al. Towards a structurally resolved human protein interaction network. *Nat. Struct. Mol. Biol.* **30**, 216–225 (2023).
29. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
30. Mirdita, M. et al. ColabFold: making protein folding accessible to all. *Nat. Methods* **19**, 679–682 (2022).
31. Brünger, A. T. et al. Crystallography & NMR system: a new software suite for macromolecular structure determination. *Acta Crystallogr. D. Biol. Crystallogr.* **54**, 905–921 (1998).
32. de Vries, S. J., van Dijk, M. & Bonvin, A. M. J. J. The HADDOCK web server for data-driven biomolecular docking. *Nat. Protoc.* **5**, 883–897 (2010).
33. van Zundert, G. C. P. et al. The HADDOCK2.2 web server: user-friendly integrative modeling of biomolecular complexes. *J. Mol. Biol.* **428**, 720–725 (2016).
34. Roel-Touris, J., Don, C. G., Honorato, R. V., Rodrigues, J. P. G. L. M. & Bonvin, A. M. J. J. Less is more: coarse-grained integrative modeling of large biomolecular assemblies with HADDOCK. *J. Chem. Theory Comput.* **15**, 6358–6367 (2019).
35. Honorato, R. V., Roel-Touris, J. & Bonvin, A. M. J. J. MARTINI-based protein-DNA coarse-grained HADDOCKing. *Front. Mol. Biosci.* **6**, 102 (2019).
36. Marrink, S. J., Risselada, H. J., Yefimov, S., Tieleman, D. P. & de Vries, A. H. The MARTINI force field: coarse grained model for biomolecular simulations. *J. Phys. Chem. B* **111**, 7812–7824 (2007).
37. van Zundert, G. C. P. & Bonvin, A. M. J. J. Defining the limits and reliability of rigid-body fitting in cryo-EM maps using multi-scale image pyramids. *J. Struct. Biol.* **195**, 252–258 (2016).
38. Neijenhuis, T., Keulen, S. Cvan & Bonvin, A. M. J. J. Interface refinement of low- to medium-resolution cryo-EM complexes using HADDOCK2.4. *Structure* **30**, 476–484 (2021).
39. Koukos, P. I., Réau, M. & Bonvin, A. M. J. J. Shape-restrained modeling of protein–small-molecule complexes with high ambiguity driven DOCKing. *J. Chem. Inf. Model* **61**, 4807–4818 (2021).
40. Honorato, R. V. et al. Structural biology in the clouds: the WeNMR-EOSC ecosystem. *Front. Mol. Biosci.* **8**, 729513 (2021).
41. Chen, V. B. et al. MolProbity: all-atom structure validation for macromolecular crystallography. *Acta Crystallogr. D. Biol. Crystallogr.* **66**, 12–21 (2010).
42. Schüttelkopf, A. W. & van Aalten, D. M. F. PRODRG: a tool for high-throughput crystallography of protein–ligand complexes. *Acta Crystallogr. D. Biol. l Crystallogr.* **60**, 1355–1363 (2004).
43. Rose, A. S. et al. NGL viewer: web-based molecular graphics for large complexes. *Bioinformatics* **34**, 3755–3758 (2018).
44. Jorgensen, W. L. & Tirado-Rives, J. The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.* **110**, 1657–1666 (1988).
45. Fernández-Recio, J., Totrov, M. & Abagyan, R. Identification of protein–protein interaction sites from docking energy landscapes. *J. Mol. Biol.* **335**, 843–865 (2004).
46. Veerapandian, B. et al. Functional implications of interleukin-1β based on the three-dimensional structure. *Proteins* **12**, 10–23 (1992).
47. Blech, M. et al. One target—two different binding modes: structural insights into gevokizumab and canakinumab interactions to interleukin-1β. *J. Mol. Biol.* **425**, 94–111 (2013).
48. Rodrigues, J. P. G. L. M., Teixeira, J. M. C., Trellet, M. & Bonvin, A. M. J. J. pdb-tools: a swiss army knife for molecular structures. *F1000Res.* **7**, 1961 (2018).
49. Mattiroli, F., Uckelmann, M., Sahtoe, D. D., van Dijk, W. J. & Sixma, T. K. The nucleosome acidic patch plays a critical role in RNF168-dependent ubiquitination of histone H2A. *Nat. Commun.* **5**, 3291 (2014).
50. McGinty, R. K., Henrici, R. C. & Tan, S. Crystal structure of the PRC1 ubiquitylation module bound to the nucleosome. *Nature* **514**, 591–596 (2014).
51. Bentley, M. L. et al. Recognition of UbcH5c and the nucleosome by the Bmi1/Ring1b ubiquitin ligase complex. *EMBO J.* **30**, 3285–3297 (2011).
52. Blech, M. & Hoerer, S. Crystal structure of human IL-1beta in complex with therapeutic antibody binding fragment of gevokizumab. Available at https://www.rcsb.org/structure/4g6m (2012).
53. Basu, S. & Wallner, B. DockQ: a quality measure for protein-protein docking models. *PLoS One* **11**, e0161879 (2016).

54. HADDOCK2.4 web server protocol data. *GitHub* https://github.com/haddocking/haddock24-protocol (2023).
55. pdb-tools. *GitHub* https://github.com/haddocking/pdb-tools (2024).
56. Eastman, P. et al. OpenMM 7: rapid development of high performance algorithms for molecular dynamics. *PLoS Comput. Biol.* **13**, e1005659 (2017).
57. Tjandra, N., Omichinski, J. G., Gronenborn, A. M., Clore, G. M. & Bax, A. Use of dipolar 1H–15N and 1H–13C couplings in the structure determination of magnetically oriented macromolecules in solution. *Nat. Struct. Biol.* **4**, 732–738 (1997).
58. 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).
59. Banci, L. et al. Paramagnetism-based restraints for Xplor-NIH. *J. Biomol. NMR* **28**, 249–261 (2004).
60. Tjandra, N., Garrett, D. S., Gronenborn, A. M., Bax, A. & Clore, G. M. 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).

Acknowledgements
The authors acknowledge financial support from the European Union Horizon 2020, projects BioExcel (675728, 823830), from the EuroHPC Joint Undertaking and Sweden, Netherlands, Germany, Spain, Finland and Norway (101093290) and from the Dutch Foundation for Scientific Research (NWO) (TOP-PUNT grant 718.015.001). The FP7 WeNMR (261572), H2020 West-Life (675858), EOSC-hub (777536) and EGI-ACE (101017567) European e-Infrastructure projects are acknowledged for the use of the EGI infrastructure with the dedicated support of CESNET-MCC, INFN-LNL-2, NCG-INGRID-PT, TW-NCHC, CESGA, IFCA-LCG2, UA-BITP, TR-FC1-ULAKBIM, CSTCLOUD-EGI, IN2P3-CPPM, CIRMMP, SURFsara and NIKHEF and the additional support of the national GRID Initiatives of Belgium, France, Italy, Germany, the Netherlands, Poland, Portugal, Spain, UK, Taiwan and the US Open Science Grid.

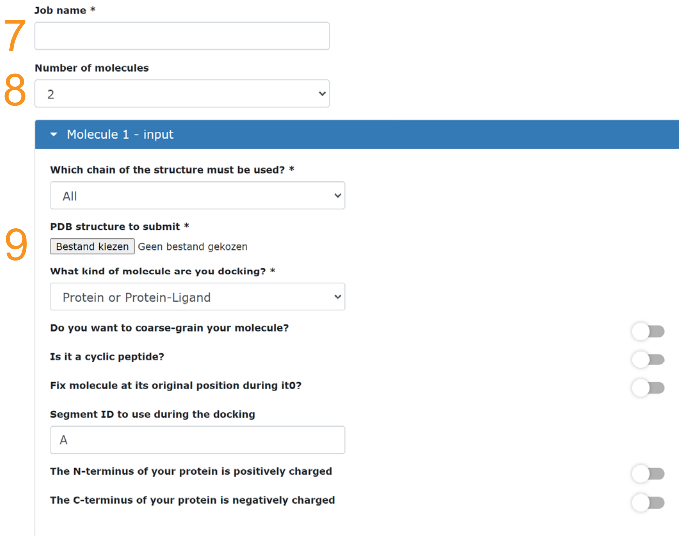
Author contributions
A.M.J.J.B. supervised the project. All authors contributed to the development of HADDOCK2.4 and the various protocols and types of molecules supported. R.V.H., M.E.T, B.J.-G., J.J.S. and P.I.K. contributed to the development of the web interface. A.M.J.J.B. and R.V.H. wrote the manuscript. All authors contributed to the manuscript editing and checking.

Competing interests
The authors declare no competing interests.

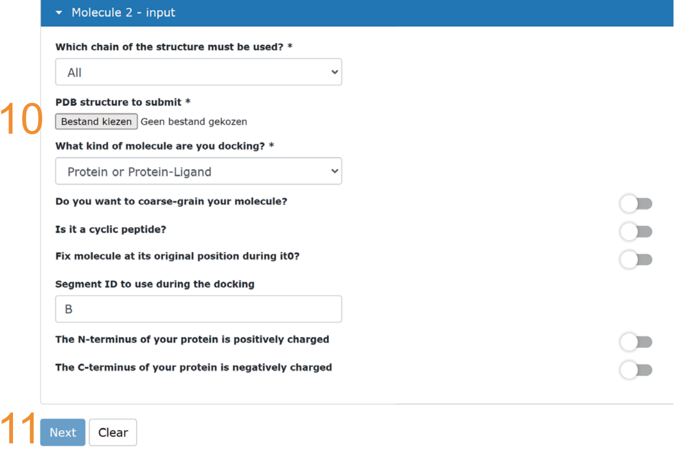
Correspondence and requests for materials should be addressed to Alexandre M. J. J. Bonvin.

Related links
Key references using this protocol

de Vries, S. J. et al. *Nat. Protoc.* **5**, 883–897 (2010): https://doi.org/10.1038/nprot.2010.32
Honorato, R. V. et al. *Front. Mol. Biosci.* **6**, 102 (2019): https://doi.org/10.3389/fmolb.2019.00102
Ambrosetti, F. et al. *Structure* **28**, 119–129.e2 (2020): https://doi.org/10.1016/j.str.2019.10.011

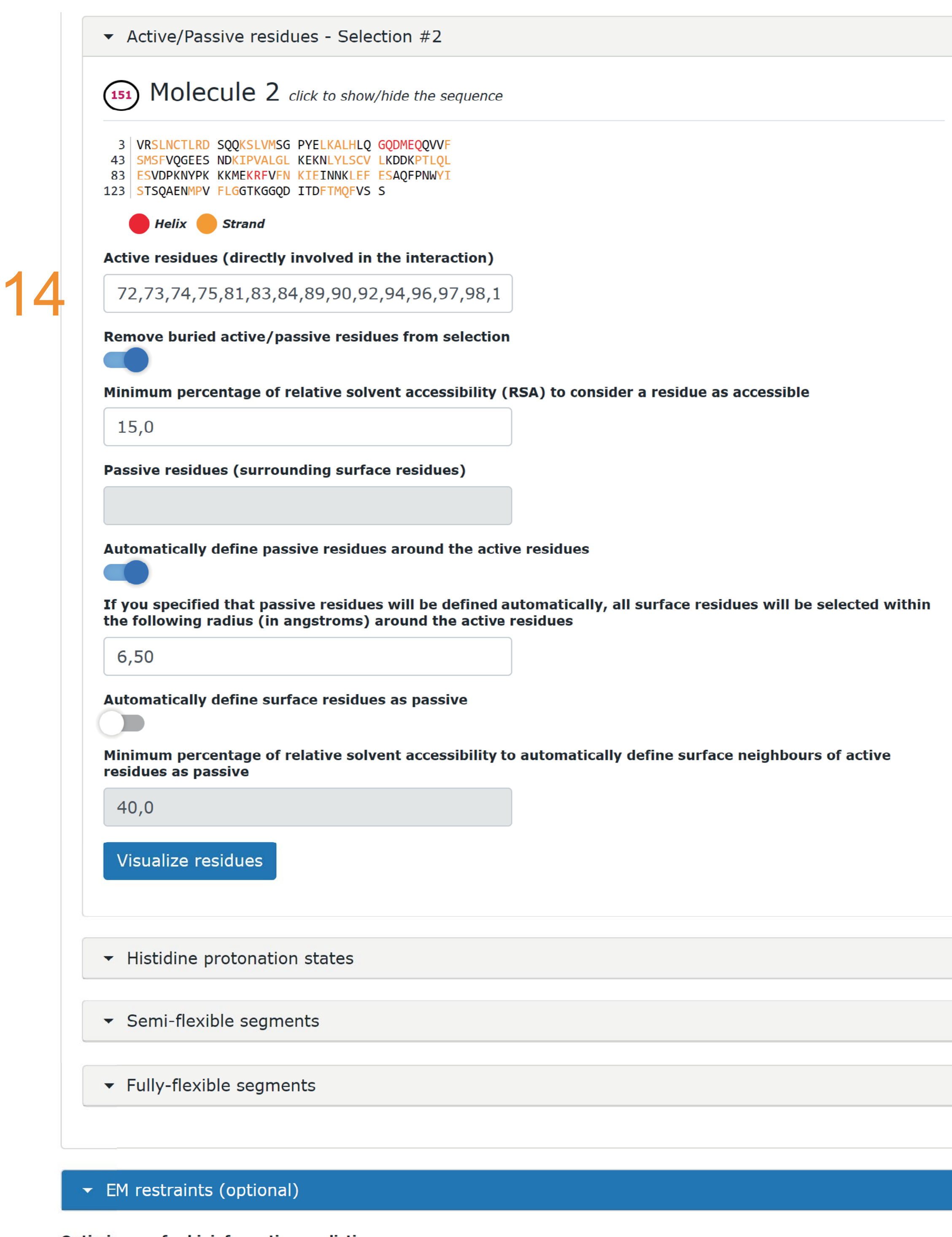


Extended Data Fig. 1 | Screenshots showing Steps 7–9 of Procedure 1.



Extended Data Fig. 2 | Screenshots showing Steps 10–11 of Procedure 1.

Extended Data Fig. 3 | Screenshots showing Steps 12–13 of Procedure 1.



Extended Data Fig. 4 | Screenshots showing Steps 14–15 of Procedure 1.