

# <sup>1</sup> gdock: Information-driven protein-protein docking using a genetic algorithm

<sup>3</sup> **Rodrigo V. Honorato**  

<sup>4</sup> 1 Computational Structural Biology Group, Utrecht University, The Netherlands ¶ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: 

Submitted: 01 February 2026

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

## <sup>5</sup> Summary

<sup>6</sup> Proteins carry out most biological functions by interacting with other proteins, and <sup>7</sup> understanding these interactions at the molecular level is essential for drug design and <sup>8</sup> biomedical research. Computational docking predicts how two proteins bind together by <sup>9</sup> searching for arrangements that are both physically plausible and consistent with experimental <sup>10</sup> data.

<sup>11</sup> gdock is a command-line tool that performs protein-protein docking guided by user-supplied <sup>12</sup> restraints—information about which residues are likely at the binding interface. It uses <sup>13</sup> a genetic algorithm to efficiently explore possible orientations of one protein relative to <sup>14</sup> another, scoring each candidate with a physics-based energy function. Written entirely <sup>15</sup> in Rust, gdock compiles to a single executable with no external dependencies, making it <sup>16</sup> straightforward to install and integrate into automated workflows. On a standard workstation, <sup>17</sup> most docking runs complete in under 20 seconds. Documentation and source code are available <sup>18</sup> at <https://github.com/rvhonorato/gdock> and <https://gdock.org>.

## <sup>19</sup> Statement of need

<sup>20</sup> Information-driven docking incorporates experimental data—from mutagenesis, cross-linking <sup>21</sup> mass spectrometry, or NMR—to guide protein complex structure prediction ([Noort et al., 2021](#)). While several tools support this approach, they typically require complex runtime <sup>22</sup> environments or offer limited restraint integration. gdock contributes to this ecosystem as a <sup>23</sup> fast, minimal implementation:

- <sup>25</sup> ▪ **Single binary:** No runtime dependencies or environment setup
- <sup>26</sup> ▪ **Speed:** ~15 seconds per complex on standard hardware
- <sup>27</sup> ▪ **Rust implementation:** Native energy functions, memory-safe, readable
- <sup>28</sup> ▪ **CLI-first:** Designed for scripting and pipeline integration

<sup>29</sup> The tool provides an accessible option for researchers who need restraint-driven docking without <sup>30</sup> the overhead of larger software packages.

## <sup>31</sup> State of the field

<sup>32</sup> Protein-protein docking software spans a range of complexity and capability. For example, <sup>33</sup> ClusPro ([Kozakov et al., 2017](#)) and ZDOCK ([Pierce et al., 2014](#)) provide FFT-based sampling <sup>34</sup> with web interfaces, though restraint integration is limited. HADDOCK ([Dominguez et al., 2003](#)) <sup>35</sup> offers comprehensive information-driven docking with flexible refinement, symmetry handling, <sup>36</sup> and multi-body support; LightDock ([Jiménez-García et al., 2018](#)) uses swarm optimization <sup>37</sup> with restraint support—both require managed Python environments with specific package <sup>38</sup> versions, and HADDOCK additionally depends on CNS (Crystallography and NMR System)

39 (Brünger et al., 1998). A limited Rust implementation of LightDock exists (Jiménez-García,  
 40 2020) and served as one inspiration for gdock.  
 41 gdock occupies a distinct niche: a dependency-free, single-binary tool for restraint-driven rigid-  
 42 body docking. Rather than extending existing software—which would require adapting to their  
 43 architectural constraints—gdock was built from scratch in Rust to prioritize minimal deployment  
 44 overhead and scripting integration. Crucially, the entire scoring function is implemented from  
 45 scratch in modern, readable code, making it fully transparent and easy to verify—unlike tools  
 46 that depend on legacy Fortran engines or opaque external libraries. gdock does not aim to  
 47 replace full-featured docking platforms but provides a lightweight alternative for users with  
 48 reliable interface information who need rapid, reproducible results without environment setup.

## 49 Software design

50 gdock is a Rust rewrite of an earlier Python prototype, compiling to a ~7,000-line statically-  
 51 linked binary with no runtime dependencies.

52 **Search algorithm.** A genetic algorithm explores rigid-body transformations of the ligand relative  
 53 to the receptor. Each chromosome encodes six genes: three Euler angles ( $\alpha$ ,  $\beta$ ,  $\gamma$ ) for rotation  
 54 and three displacement values (x, y, z) for translation. A generation consists of a population  
 55 of chromosomes that evolves through tournament selection, uniform crossover, creep mutation  
 56 (Gaussian perturbations for local refinement), and elitism. Fitness evaluation is parallelized  
 57 across the population. The search terminates early upon convergence.

58 **Scoring function.** The energy function combines four terms:

$$E_{total} = w_{vdw}E_{vdw} + w_{elec}E_{elec} + w_{desolv}E_{desolv} + w_{air}E_{air}$$

- 59 ▪  $E_{vdw}$ : Soft-core Lennard-Jones potential that remains finite at short distances, allowing  
   60 the search to explore conformations with minor clashes
- 61 ▪  $E_{elec}$ : Coulombic interactions with distance-dependent dielectric ( $\varepsilon = r$ ) to dampen  
   62 long-range effects
- 63 ▪  $E_{desolv}$ : Empirical atomic solvation parameters penalizing burial of polar atoms and  
   64 rewarding burial of hydrophobic atoms
- 65 ▪  $E_{air}$ : Flat-bottom harmonic potential on C $\alpha$ –C $\alpha$  distances between user-specified residue  
   66 pairs (no penalty within 0–7 Å, quadratic penalty beyond), conceptually inspired by  
   67 HADDOCK's distance restraints (Dominguez et al., 2003) but using a simpler purely  
   68 harmonic form without the linear switching at long distances

69 **Weight calibration.** The weights  $w_{vdw}$ ,  $w_{elec}$ , and  $w_{desolv}$  were calibrated using the  
 70 Dockground decoy set (Gao et al., 2008), which provides 100 decoy structures per complex  
 71 with exactly one near-native conformation. A grid search tested weight combinations by  
 72 re-scoring all decoys and measuring how often the near-native structure ranked in the top  
 73 50. The final weights ( $w_{vdw} = 0.4$ ,  $w_{elec} = 0.05$ ,  $w_{desolv} = 3.4$ ) maximize this ranking  
 74 performance. The restraint weight  $w_{air}$  is fixed at 1.0 since calibration was performed without  
 75 restraints.

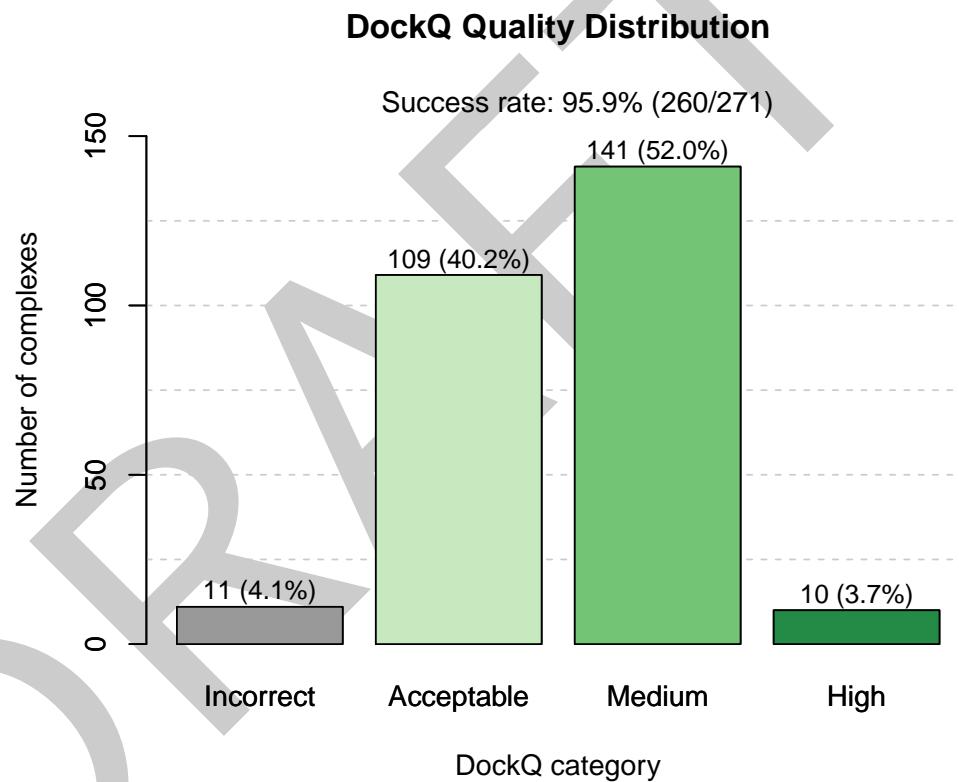
76 **Output.** Final models are clustered using Fraction of Common Contacts (FCC) (Rodrigues et  
 77 al., 2012), re-implemented natively in Rust, and ranked by score, providing both diverse and  
 78 top-scoring solutions.

79 **Code quality.** The codebase includes 174 unit tests covering parsing, energy calculations,  
 80 and algorithm behavior. Continuous integration enforces code formatting (rustfmt), linting  
 81 (clippy with warnings as errors), and test passage on every commit. Rust's ownership model  
 82 provides compile-time guarantees against data races and use-after-free errors. The software is  
 83 released under the permissive 0BSD license.

## 84 Research impact statement

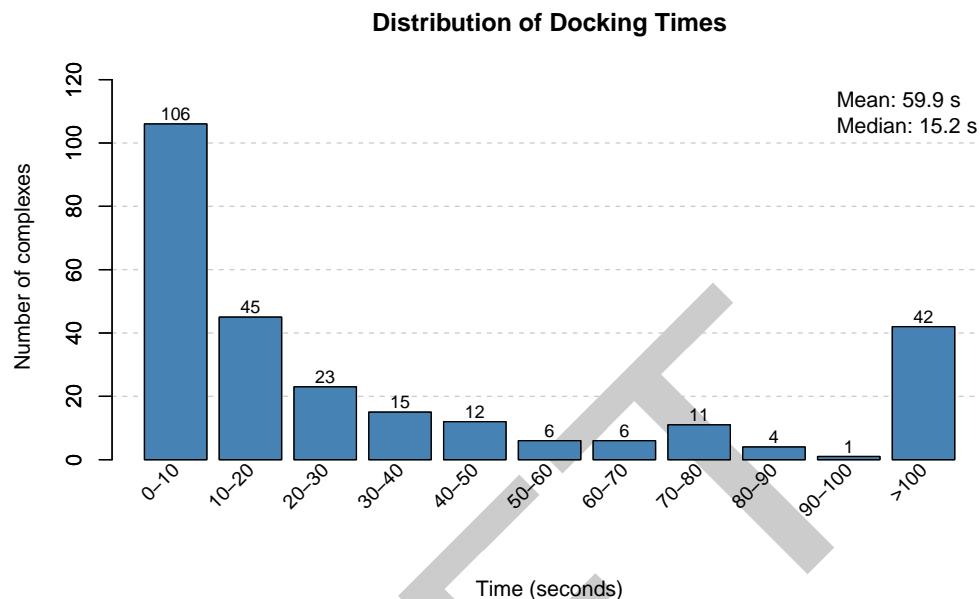
85 gdock was validated on 271 complexes from the Protein-Protein Docking Benchmark v5 (Vreven  
 86 et al., 2015), a standard dataset for assessing methods, using the bound-bound conformations.  
 87 Restraints were derived as explicit C $\alpha$ -C $\alpha$  residue pairs from native interface contacts (within  
 88 5 Å), simulating ideal contact information. Unlike HADDOCK's ambiguous interface restraints,  
 89 each restraint specifies a single receptor-ligand residue pair.

90 Using the DockQ metric (Basu & Wallner, 2016) to assess model quality, gdock achieved a  
 91 95.9% success rate (260/271 complexes with at least one acceptable model, DockQ  $> 0.23$ ).  
 92 Medium-quality models (DockQ  $< 0.49$ ) were obtained for 55.7% of complexes, and high-quality  
 93 models (DockQ  $> 0.80$ ) for 3.7% (Figure 1).



**Figure 1:** Distribution of docking quality across 271 benchmark complexes. Each complex is categorized by its best DockQ score among 10 output models.

94 Performance benchmarks on a 48-core machine show a median docking time of ~15 seconds  
 95 per complex, with 56% of cases completing within 20 seconds (Figure 2).



**Figure 2:** Distribution of docking times across benchmark complexes. Most cases complete within 20 seconds; outliers correspond to larger protein systems.

96 These results reflect ideal restraint conditions; real-world performance depends on restraint  
 97 quality. As a rigid-body method, gdock is best suited for cases where conformational changes  
 98 upon binding are minimal.  
 99 Scripts to reproduce these experiments are available at: <https://github.com/rvhonorato/gdock-benchmark>.  
 100

## 101 AI usage disclosure

102 Claude (Anthropic) assisted with code review, test generation, and proofreading. All AI-  
 103 generated content was verified by the author through manual review and the continuous  
 104 integration pipeline.

## 105 Acknowledgements

106 The author thanks Prof. Dr. Alexandre Bonvin for computational resources and expertise, and  
 107 Dr. Brian Jiménez-García for early conceptualization.

## 108 References

- 109 Basu, S., & Wallner, B. (2016). DockQ: A quality measure for protein-protein docking models.  
 110 *PloS One*, 11(8), e0161879. <https://doi.org/10.1371/journal.pone.0161879>
- 111 Brünger, A. T., Adams, P. D., Clore, G. M., DeLano, W. L., Gros, P., Grosse-Kunstleve, R. W.,  
 112 Jiang, J.-S., Kuszewski, J., Nilges, M., Pannu, N. S., Read, R. J., Rice, L. M., Simonson,  
 113 T., & Warren, G. L. (1998). Crystallography & NMR system: A new software suite for  
 114 macromolecular structure determination. *Acta Crystallographica Section D: Biological  
 115 Crystallography*, 54(5), 905–921. <https://doi.org/10.1107/S0907444998003254>

- 116 Dominguez, C., Boelens, R., & Bonvin, A. M. (2003). HADDOCK: A protein-protein docking  
117 approach based on biochemical or biophysical information. *Journal of the American  
118 Chemical Society*, 125(7), 1731–1737. <https://doi.org/10.1021/ja026939x>
- 119 Gao, Y., Douguet, D., Tovchigrechko, A., & Vakser, I. A. (2008). DOCKGROUND protein-  
120 protein docking decoy set. *Bioinformatics*, 24(22), 2634–2635. <https://doi.org/10.1093/bioinformatics/btn497>
- 122 Jiménez-García, B. (2020). *Lightdock-rust: Rust implementation of the LightDock  
123 macromolecular docking framework*. <https://github.com/lightdock/lightdock-rust>
- 124 Jiménez-García, B., Roel-Touris, J., Romero-Durana, M., Vidal, M., Jiménez-González, D., &  
125 Fernández-Recio, J. (2018). LightDock: A new multi-scale approach to protein-protein  
126 docking. *Bioinformatics*, 34(1), 49–55. <https://doi.org/10.1093/bioinformatics/btx555>
- 127 Kozakov, D., Hall, D. R., Xia, B., Porter, K. A., Padhorny, D., Yueh, C., Beglov, D., & Vajda,  
128 S. (2017). The ClusPro web server for protein-protein docking. *Nature Protocols*, 12(2),  
129 255–278. <https://doi.org/10.1038/nprot.2016.169>
- 130 Noort, C. W. van, Honorato, R. V., & Bonvin, A. M. (2021). Information-driven modeling  
131 of biomolecular complexes. *Current Opinion in Structural Biology*, 70, 70–77. <https://doi.org/10.1016/j.sbi.2021.05.003>
- 133 Pierce, B. G., Wiehe, K., Hwang, H., Kim, B.-H., Vreven, T., & Weng, Z. (2014). ZDOCK  
134 server: Interactive docking prediction of protein-protein complexes and symmetric trimers.  
135 *Bioinformatics*, 30(12), 1771–1773. <https://doi.org/10.1093/bioinformatics/btu097>
- 136 Rodrigues, J. P., Trellet, M., Schmitz, C., Kastritis, P., Karaca, E., Melquiond, A. S., &  
137 Bonvin, A. M. (2012). Clustering biomolecular complexes by residue contacts similarity.  
138 *Proteins: Structure, Function, and Bioinformatics*, 80(7), 1810–1817. <https://doi.org/10.1002/prot.24078>
- 140 Vreven, T., Moal, I. H., Vangone, A., Pierce, B. G., Kastritis, P. L., Torchala, M., Chaleil,  
141 R., Jimenez-Garcia, B., Bates, P. A., Fernandez-Recio, J., & others. (2015). Updates  
142 to the integrated protein-protein interaction benchmarks: Docking benchmark version 5  
143 and affinity benchmark version 2. *Journal of Molecular Biology*, 427(19), 3031–3041.  
144 <https://doi.org/10.1016/j.jmb.2015.07.016>