

TOOLS FOR PROTEIN SCIENCE

CCBuilder 2.0: Powerful and accessible coiled-coil modeling

Christopher W. Wood ¹ and Derek N. Woolfson ^{1,2,3*}

¹School of Chemistry, University of Bristol, Cantock's Close, Bristol BS8 1TS, United Kingdom

²School of Biochemistry, University of Bristol, Medical Sciences Building, University Walk, Bristol BS8 1TD, United Kingdom

³BrisSynBio, University of Bristol, Life Sciences Building, Tyndall Avenue, Bristol BS8 1TQ, United Kingdom

Received 30 June 2017; Accepted 22 August 2017

DOI: 10.1002/pro.3279

Published online 24 August 2017 proteinscience.org

Abstract: The increased availability of user-friendly and accessible computational tools for biomolecular modeling would expand the reach and application of biomolecular engineering and design. For protein modeling, one key challenge is to reduce the complexities of 3D protein folds to sets of parametric equations that nonetheless capture the salient features of these structures accurately. At present, this is possible for a subset of proteins, namely, repeat proteins. The α -helical coiled coil provides one such example, which represents \approx 3–5% of all known protein-encoding regions of DNA. Coiled coils are bundles of α helices that can be described by a small set of structural parameters. Here we describe how this parametric description can be implemented in an easy-to-use web application, called CCBuilder 2.0, for modeling and optimizing both α -helical coiled coils and polyproline-based collagen triple helices. This has many applications from providing models to aid molecular replacement for X-ray crystallography, *in silico* model building and engineering of natural and designed protein assemblies, and through to the creation of completely *de novo* “dark matter” protein structures. CCBuilder 2.0 is available as a web-based application, the code for which is open-source and can be downloaded freely. <http://coiledcoils.chm.bris.ac.uk/ccbuilder2>.

Lay Summary: We have created CCBuilder 2.0, an easy to use web-based application that can model structures for a whole class of proteins, the α -helical coiled coil, which is estimated to account for 3–5% of all proteins in nature. CCBuilder 2.0 will be of use to a large number of protein scientists engaged in fundamental studies, such as protein structure determination, through to more-applied research including designing and engineering novel proteins that have potential applications in biotechnology.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Abbreviations: BUDE, Bristol University Docking Engine; BUFF, Bristol University docking engine Force Field; ISAMBARD, Intelligent System for Analysis, Model Building And Rational Design

Additional Supporting Information may be found in the online version of this article.

Grant sponsor: European Research Council; Grant number: 340764; Grant sponsor: BBSRC/ERASynBio; Grant number: BB/M005615/1; Grant sponsor: Royal Society Wolfson Research Merit Award; Grant number: WM140008.

*Correspondence to: Derek N. Woolfson, School of Chemistry, University of Bristol, Cantock's Close, Bristol BS8 1TS, United Kingdom. E-mail: D.N.Woolfson@bristol.ac.uk

Keywords: coiled coil; collagen; computational design; structural modeling; structural bioinformatics; web app; parametric design; protein design; synthetic biology

Introduction

Protein design has advanced rapidly over the past decade, with biomolecular modeling making a significant contribution to this development.^{1–6} An exciting and emerging aspect of protein design centers on exploring the “dark matter” of protein fold space; that is, those protein folds that are theoretically possible but have not been observed in or explored by nature.⁷ This offers near unlimited potential for designing new protein structures that could have a range of applications in synthetic biology and biotechnology.^{7,8} However, designing such *truly de novo* structures represents a significant challenge, as no specific sequence or structural data are available as guides. An additional and general challenge in protein design is the combinatorial nature of the problem, which renders unguided searches through sequence and structural space impossible. Computational methods are required to address both of these challenges. Parametric modeling and design of protein structures^{9–12}—i.e., where a fold or set of related folds are described geometrically—offers a route to addressing both these challenges as it effectively guides and reduces the conformational space examined.

In order for parametric modeling to be useful and effective, the fold must have a regular structure that can be described using a small number of mathematical parameters. Repeat proteins are an excellent target for this type of modeling.¹³ The α -helical coiled coil is one such example, which has been fertile ground for initial exploration into parametric modeling and design.^{4,9,10,14,15}

Coiled coils account for between 3% and 5% of all protein-encoding DNA.^{16,17} They perform a diverse range of natural functions including structural roles, DNA binding, and driving protein–protein interactions.^{18–20} This ubiquitous fold consists of two or more α helices that wrap around each other to form a rope-like, superhelical structure.^{21,22} Classical coiled-coil helices are amphipathic with a hydrophobic stripe along their length. This is usually generated through a repeating sequence unit of seven amino acids known as a heptad repeat, although other repeats are possible and observed.^{22,23} Conventionally, the residues of the heptad are assigned the letters *a–g* (Fig. 1). Hydrophobic residues usually occupy the first (*a*) and fourth (*d*) sites of the repeat, and the number, precise position, and type of the hydrophobic side chain strongly influences the oligomeric state, topology, and partnering preference of the coiled coil.^{10,20–22,24–26}

Most of the binding enthalpy that drives assembly of the helices comes from the formation of a

hydrophobic interface, with lesser contributions from salt-bridge and/or hydrogen-bonding interactions (Fig. 1).²⁷ The formation of the hydrophobic interface is facilitated by an intimate mode of helix–helix interaction known as knobs-into-holes packing. This is formed when a series of “knob” residues on one helix projects into complementary “holes” formed by 4-residue diamonds on a partnering helix.²¹

Dimeric, trimeric, and tetrameric coiled coils are the most abundant forms of α -helical coiled coils found in nature, accounting for >98% of all known coiled coils.^{28,29} This wealth of information has led to reliable methods for predicting these low-order, or classical coiled-coil states from sequence.^{22,23,30–32} Moreover, these structures and associated sequences provide rules of thumb for the successful rational *in vivo* design of simple coiled-coil assemblies.^{33,34} In addition, there are examples of both natural and designed coiled coils with higher oligomeric states.^{9,10,35–37} α -Helical barrels—i.e., coiled coils with oligomeric states >4—are of particular interest as they have channels that run along their lengths. These structures have a range of potential applications as scaffolds for *de novo* enzymes, materials, and membrane channels.^{3,38–40} As higher order coiled coils are so rare in nature, atomistic modeling is essential for the design of these structures. This presents a considerable opportunity for *in silico* modeling and design to cover and test a larger sequence space ahead of time-consuming experiments.

Parametric Modeling of Coiled-Coils

Owing to the regular nature of α -helical coiled coils, their structures can be described using a small number of geometric parameters. The first mathematical parameterization was provided by Crick in 1953, who described the super helix of a coiled coil using three simple parameters pitch (*P*) or pitch angle (α), radius of the assembly (*r*), and the interface angle (also known as the Crick angle at *a* or $\varphi C\alpha$) (Fig. 1).⁴¹

A range of software tools have applied or built upon Crick’s original parameterization: In 1995, Offer and Session developed software (MakeCCSC) to model the backbone of the coiled coil, utilizing Crick’s original parameterization.⁴² Around the same time, Harbury *et al.* also implemented Crick’s equations to aid the design of a right-handed coiled-coil tetramer.^{43,44} A generalization of the Crick equations followed in 2002, allowing coiled coils with noncanonical repeats to be modeled.⁴⁵ More recently, Grigoryan and DeGrado have developed methods for

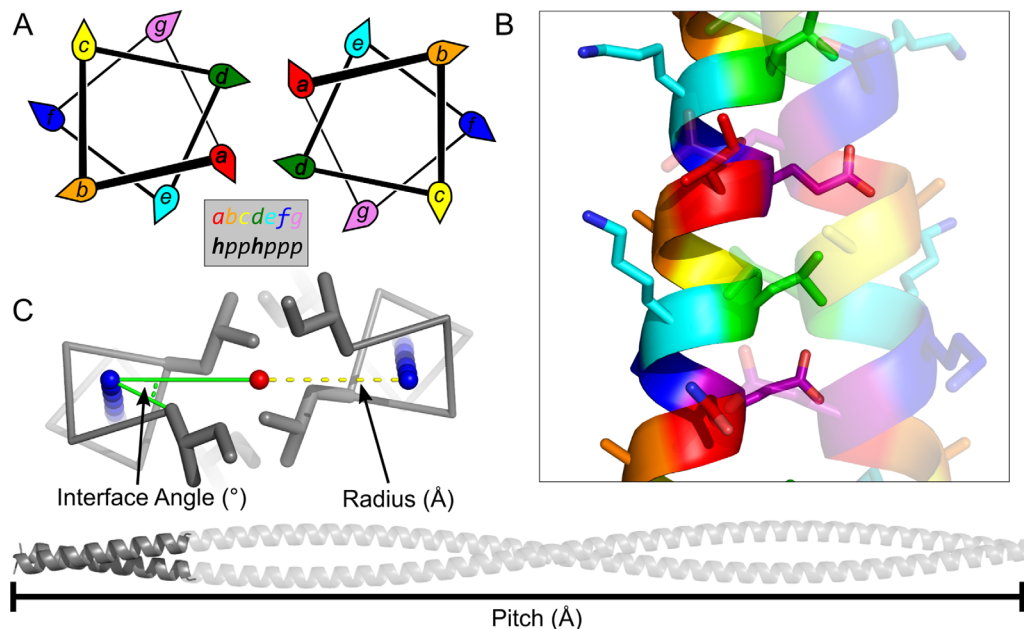


Figure 1. Structure of α -helical coiled coils. (A) Helical-wheel diagrams showing the projection of residues in the heptad repeat. (B) Helices in a coiled coil pack closely together, forming knobs-into-holes interactions. (C) Coiled coils can be described using three geometric parameters: interface angle ($^{\circ}$), radius (\AA), and pitch (\AA).

fitting Crick parameters to structural data of coiled coils, and building backbone models.¹⁵

We have added to this body of work by creating CCBUILDER,¹⁴ a web-based application for generating models of coiled coils and by developing the ISAMBARD software package,⁴⁶ which generalizes parametric model building, and has a range of tools for modeling α -helical coiled coils, helical bundles and, indeed, any parameterizable protein fold.

CCBuilder

CCBuilder was developed with an emphasis on usability and to be accessible to nonspecialist users. Given a sequence and a few structural parameters, it will create a fully atomistic model of a coiled coil. The pipeline for generating a model is straightforward: a modified version of MAKECCSC generates a backbone model with the required structural parameters; the model is passed to Rosetta (using the PyRosetta interface) to pack the side chains⁴⁷; CCBUILDER then runs a range of analysis programs to assess the quality of the model, including geometric evaluation of the backbone, knobs-into-holes analysis with SOCKET,⁴⁸ and interaction energies using two all-atom force fields, BUDE^{49,50} and Rosetta.⁵¹

Previously, we tested the accuracy of CCBUILDER's modeling protocol by recreating the structures of 653 coiled-coil proteins with known structures.¹⁴ These had a range of common and unusual topologies, which accounted for >97% of all experimental determined coiled-coil structures. The models were shown to be highly accurate, with respect to the

experimental structures [0.77 \AA (standard deviation 0.49 \AA) backbone RMSD], demonstrating that a parameterization with only 3 or 4 structural parameters can capture much of the natural complexity of this class of protein fold, including rotameric preference of side chains.¹⁴

Our focus on usability of CCBUILDER has enabled wide spread adoption by many users, not just by specialists in molecular modeling or protein design, and for many applications. In a similar vein to Crick's original motivation to parameterize the α -helical coiled coil, we routinely use models generated by CCBUILDER to phase X-Ray crystal structure data during molecular replacement.¹⁰ Similarly, others have used models generated with CCBUILDER to fit SAXS data^{52,53} or model electron microscopy data.^{38,53,54} Various natural coiled-coil sequences, with unknown structures, have been analyzed using models produced by CCBUILDER: models have been used to predict of the sequence register of the heptad repeat⁵⁵; to probe the interaction energy of putative interfaces^{56,57}; and to calculate electrostatic surface potentials.⁵⁸ Furthermore, models from CCBUILDER have been used as the basis for other computational methods, such as homology modeling, using other software packages.^{59,60}

CCScanner

The CCBUILDER protocol was further developed to create CCScanner, which automates model building by fitting coiled coil parameters for a given sequence. To do this, the model-building protocol was extracted from CCBUILDER, and a genetic

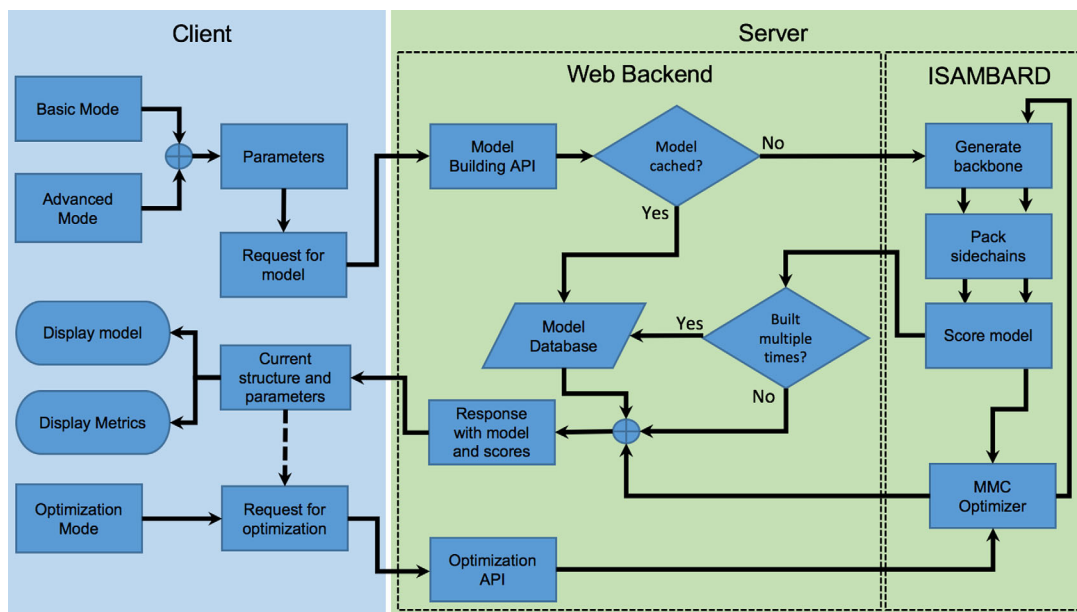


Figure 2. Architecture of the CCBUILDER 2.0 web application. The client side (Elm, JavaScript, HTML, and CSS) is used for submitting parameters and displaying models/metrics. The web backend (Flask, MongoDB, uWSGI, and NGINX) serves the application web pages and also provides an RESTful API to the modeling engine (implemented using ISAMBARD).⁴⁶

algorithm was implemented to optimize the structural parameters used to build the model. While CCScanner is unpublished, we have incorporated its model-building methodology into the ISAMBARD software package.⁴⁶

CCScanner can be used to predict the oligomeric state and coiled-coil parameters for a given sequence, whether natural or designed. We applied it as part of the computational design of α -helical barrels.¹⁰ High-resolution X-ray crystal structures were determined for four peptides—CC-Pent, CC-Hex2, CC-Hex3, and CC-Hept—using models produced by CCScanner to phase the data during molecular replacement. CCScanner accurately predicted the oligomeric state and coiled-coil parameters for all of the structurally characterized α -helical barrels, demonstrating that it is a powerful tool for the design of coiled coils in both observed and possible conformations. These peptides extend a basis set of completely *de novo* coiled coils,²⁷ which now contains a *de novo* designed dimer through to heptamer (Fig. 4).

CCBuilder 2.0

Since the original development of CCBUILDER, the architecture of the application has proven difficult to maintain, so it has become increasingly necessary to update the application. Furthermore, with developments in coiled-coil design, there is demand to generate models on a much higher scale than previously possible, using optimization algorithms similar to those developed for CCScanner. Here we present CCBUILDER 2.0, a complete rewrite of the CCBUILDER web application using modern web technologies and

software design tools/principles, and state-of-the-art parametric modeling software. This brings a range of new features and improvements to robustness, usability, scalability, and portability. The CCBUILDER 2.0 web application can be accessed at: <http://coiled-coils.chm.bris.ac.uk/ccbuilder2>. The source code is open source and available to download from the Woolfson Group GitHub repository (<https://github.com/woolfson-group>).

Architecture

CCBuilder 2.0 has three main software layers: the user interface; an RESTful API backend; and the model building protocols. Each layer has been designed to be as independent as possible (Fig. 2), to make the platform flexible.

The user interface to the web application is written in Elm, JavaScript, HTML, and CSS. Elm is a functional programming language specifically for building web applications. It compiles down to native JavaScript and offers excellent performance without runtime exceptions (<http://elm-lang.org/>).

The main user interface provides a molecular viewer for visualizing models and panels for inputting parameters, building example models, running parameter optimizations, displaying model information, downloading the structure, showing build history, and controlling the viewer (Fig. 3).

The front-end collects and validates user input before generating and sending an HTTP request to the backend API (Fig. 2). With this architecture, any service could supply the model as long as the HTTP response is formatted correctly, allowing the backend model building protocols to remain up-to-date with

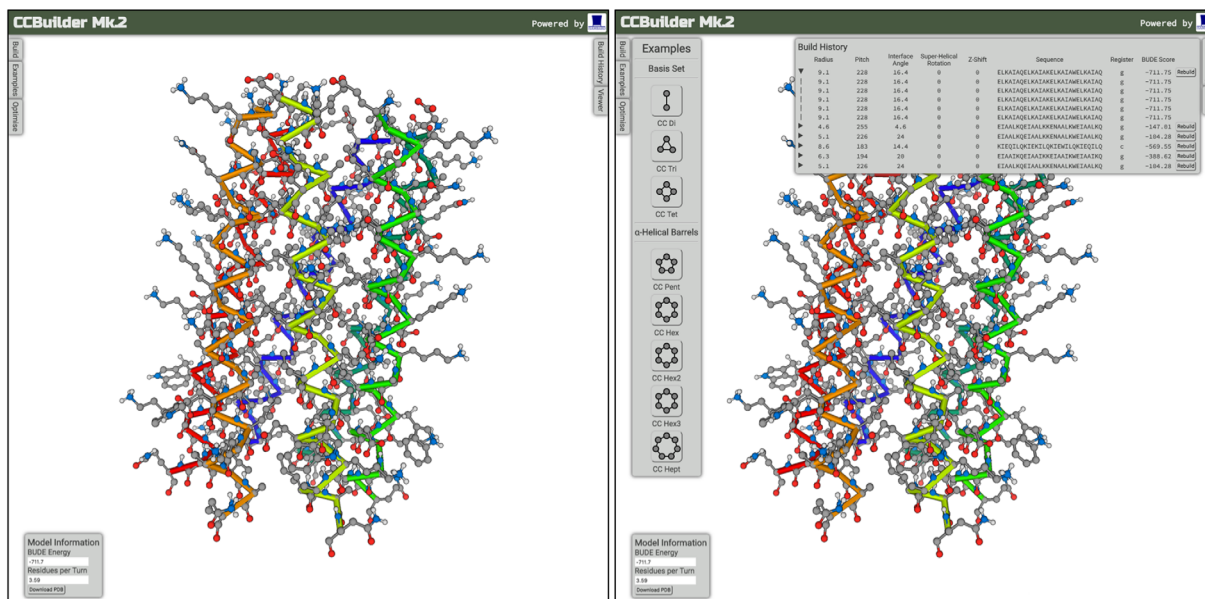


Figure 3. The CCBuilder 2.0 Interface. Panels can be hidden to give a full view of the model in the molecular viewer.

the main ISAMBARD repository. The RESTful API is designed to allow researchers to generate models with the CCBuilder 2.0 server programmatically, increasing the number of models that they can generate.

The backend of the website is written in Python using the Flask web framework (<http://flask.pocoo.org/>). It is very lightweight, providing basic routing, templating for a few web pages and, most importantly, providing an RESTful API for building, saving, and retrieving models. The application is served using uWSGI with NGINX.

The model building protocol takes the parameters provided by the front end, as JSON in the HTTP request body, and passes them to a model building script (Supporting Information). The ISAMBARD software package is used to build the model.⁴⁶ This means that if user requires more complex or a larger number of models they can bypass the web app and use ISAMBARD to run this model building protocol on their local machine.

Once a model has been generated the structure and scoring metrics (BUDE interaction energy, knobs-into-holes analysis, and backbone properties) are returned as JSON in an HTTP response, which is decoded and displayed by the front end. The structure is shown using a molecular viewer embedded in the web app, which is built using PV, a WebGL-based visualizer for proteins and biomolecules.⁶¹

When a model is built, the backend stores the parameters used to create a model in a database (MongoDB), and if a model is requested repeatedly, then the model and scoring data are cached and then subsequently served directly. This reduces server load and improves the overall user experience.

To provide portability and scalability, the whole application has been created to run in three Docker containers, one container for the web and modeling

elements (NGINX, uWSGI, Flask, and ISAMBARD) and one for managing optimization jobs and another for the database (MongoDB). A Docker Compose script is included in the application repository, allowing users to easily run the web application locally.

The interface

The CCBuilder 2.0 interface brings to focus the most important element of the application: the model itself. The window is scalable and almost all of it is dedicated to displaying the model, with floating panels that can be hidden. There are six such panels, which are used for submitting parameters, selecting examples, optimizing models, displaying model information, displaying build history, and tweaking the representation of the model in the molecular viewer. The state of the app is cached in local storage every time a model is built, allowing users to resume from exact point where they left off in the previous sessions.

Building a model

The model building protocol in CCBuilder 2.0 is implemented entirely using the ISAMBARD software package.⁴⁶ It employs a generalized parametric description of a coiled coil, which is much more flexible and offers increased model building accuracy over the methodology employed by the original CCBuilder (*vida supra*). Side chains are packed onto the backbone structure using an interface that ISAMBARD provides to SCWRL4.⁶² There are a range of analysis modules included within ISAMBARD including tools for performing geometric analysis of the backbone and a modern implementation of the SOCKET protocol.⁴⁸ The BUFF module in ISAMBARD provides an

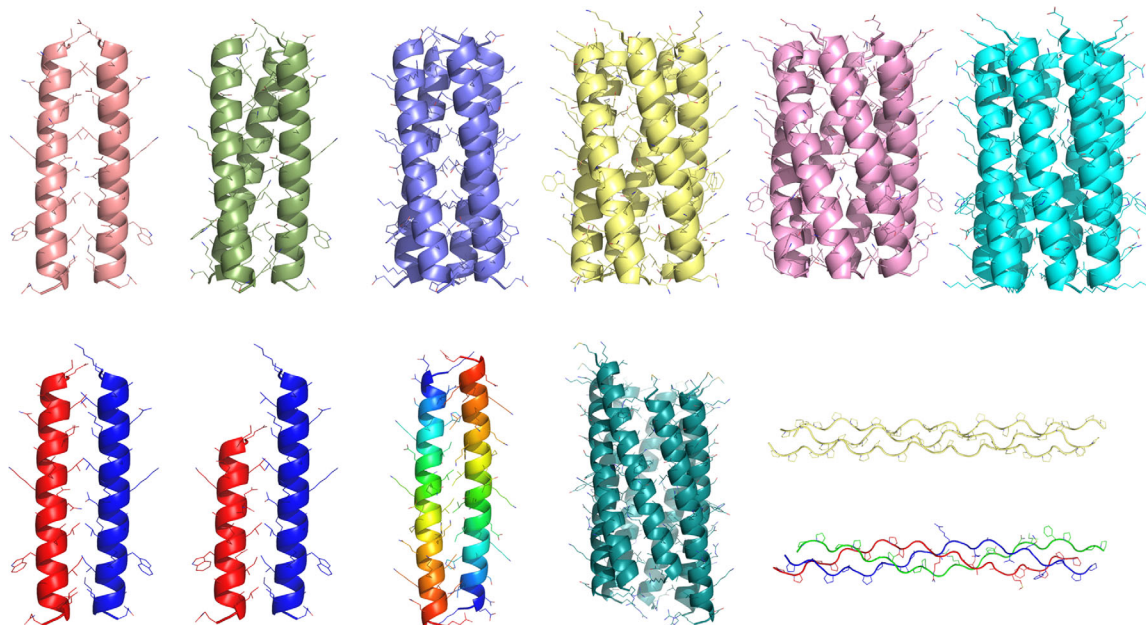


Figure 4. CCBuilder 2.0 can model a diverse range of coiled coils and collagens. Top row: dimer, trimer, tetramer, pentamer, hexamer, and heptamer, which are all homotypic. Bottom row: A4/B4 heterodimer, A3/B4 heterodimer, antiparallel homodimer, slipped heptamer, and homotrimeric and heterotrimeric collagens.

all-atom scoring potential, and is a standalone implementation of the BUDE force field.^{49,50}

To create a model using CCBuilder 2.0, users can get started quickly with a range of example models that recreate known coiled-coil structures. These can be used as a starting point for modeling coiled coils with an unknown structure. From there, the user provides parametric values that describe the coiled-coil architecture. In the basic building mode, the user gives a single value for the oligomeric state, radius, pitch, interface angle, and sequence (with an associated register). As a consequence, this mode can only be used to model parallel, homo-oligomeric coiled coils. The advanced mode allows the user to break symmetry by specifying values for all of these parameters independently for each helix. Furthermore, it allows the user to specify some additional parameters such as helix orientation, z-shift, and super-helical rotation, allowing antiparallel and slipped systems to be modelled. The super-helical rotation value refers to deviation from the ideal value for a symmetric bundle, which are calculated with the following equation:

$$i-1\left(\frac{360}{n}\right)$$

where i is the helix identifier—e.g., helices 1, 2, 3, and 4 in a tetrameric coiled coil—and n is the oligomeric state. For example, in a tetramer, helix 1 would have a default super-helical rotation value of 0°, while helix 3 would have a value of 180°.

All parameters are defined relative to the super-helical axis of the coiled coil, and so are completely

decoupled from one another. This is required for users to generate models of complex coiled coils, where symmetry is broken and the concept of a reference helix becomes meaningless.

CCBuilder 2.0 offers a significant improvement in parameter submission and validation. CCBuilder was restricted to a maximum of 8 helices in the advanced build mode. With the design and discovery of larger coiled-coil assemblies, this restriction has been lifted. Tools have been added to facilitate building models of coiled coils with higher oligomeric states, such as “copy and paste” buttons to allow parameters to be transferred between helices, meaning that repetitive parameter entry is avoided. Users can switch between the basic and advanced building modes at any time, and parameters that have been entered up to that point are carried over. For example, if a user generates a model of a coiled coil using the basic building mode, this can be used as a starting point for building an advanced mode model, as the parameters are preserved without restriction.

Collagen build mode

One major advance of parametric modeling in ISAMBARD, compared to previous software for such modeling of coiled coils, is that the mathematics that describes the parameterization (called a “specification”) is separated from the mathematics that describes the secondary structure. Essentially, in ISAMBARD, the specification contains a mathematical parameterization that creates paths through space, along which secondary structure are built. This means that the coiled-coil specification, which

describes paths that follow a super-helix, can be equally applied to coiled coils or other helical bundles, such as the collagen triple helix.⁴⁶

As ISAMBARD is used as the modeling engine in CCBUILDER 2.0, we have included an option to model the collagen triple helix. Both basic and advanced build modes are available, with parameters for radius, pitch, interface angle, and z-shift in both modes, with the oligomeric state locked to 3. Both homotypic and heterotypic collagens can be modeled. For the collagen build mode only, the character “O” is also allowed in sequence submissions, which inserts the noncanonical amino acid hydroxyproline into the model at the specified positions.

Optimizing a model

In contrast to CCBUILDER, which requires manual tweaking of parameters to optimize a coiled-coil model, CCBUILDER 2.0 utilizes ISAMBARD to implement the CCScanner protocol, allowing automated optimization of structural parameters. CCBUILDER 2.0 optimizes structural parameters based on the BUDE interaction energy between the helices.^{49,50} We have shown previously that this approach can be used to predict the oligomeric state and parameters for a coiled-coil sequence with high accuracy.¹⁰ The optimization tab in CCBUILDER 2.0 can be used to submit a model for parameter optimization, and, on the server side, uses a Metropolis Monte Carlo optimizer, implemented in the ISAMBARD software package, to improve the values iteratively. Owing to restriction on available server-side compute, optimizations can only be submitted for models constructed using the basic build mode, and are limited to a maximum of 300 residues. If more-complex, larger, or high-throughput optimizations are required, they can be performed on a local computing resource using the ISAMBARD software package directly.

Conclusions

If carefully balanced, usability of a biomolecular modeling package does not have to be at the expense of modeling power. With CCBUILDER, our focus on usability has enabled scores of research scientists, many of whom are nonexpert in atomistic modeling, to produce highly accurate models of coiled coils that can be used for a range of real-world applications.^{52–58,60}

With CCBUILDER 2.0 (<http://coiledcoils.chm.bris.ac.uk/ccbuilder2/>), we have modernized the whole application, preserving the core functionality while adding a range of new features and vastly improving the overall user experience. It is now possible to optimize structural parameters to improve the quality of the model with a single click. Users who wish to dig a little deeper can download the modeling API itself, ISAMBARD,⁴⁶ and perform more complex model-building routines offline.

Careful consideration has been made to the portability and extendibility of the code base. Heavy users of CCBUILDER 2.0, or those with unreliable internet connections, are encouraged to download the application and run it inside a Docker container on their local machine. The application is open source and freely available through GitHub (<https://github.com/woolfson-group/ccbmk2>). Users are free to modify or extend CCBUILDER 2.0, and any useful modifications can be submitted to be merged into the public application, allowing all users to benefit. We plan to update the application frequently in a similar manner, continually adding new features and improving stability, usability, and utility.

A wide range of informatics and modeling tools are becoming available for α -helical coiled coils.^{14,28,29,31,63–66} A clear path forward would be to link these tools together to create a unified informatics and modeling service. Such a “CCCentral” facility would allow user to feed data from natural coiled coils directly into design, or structural modeling of putative coiled coils identified using sequence information. We plan to build and present such an environment in the future to benefit a broad swathe of peptide and protein scientists interested in the structural biology, engineering, design, and application of α -helical coiled coils and related parameterizable protein folds.

Supporting Information

The Supporting Information associated with this article contains example Python scripts that the web backend uses for generating models.

References

1. Jacobs TM, Williams B, Williams T, Xu X, Eletsky A, Federizon JF, Szyperski T, Kuhlman B (2016) Design of structurally distinct proteins using strategies inspired by evolution. *Science* 352:687–690.
2. Jiang L, Althoff EA, Clemente FR, Doyle L, Rothlisberger D, Zanghellini A, Gallaher JL, Betker JL, Tanaka F, Barbas CF, Hilvert D, Houk KN, Stoddard BL, Baker D (2008) De novo computational design of retro-aldol enzymes. *Science* 319:1387–1391.
3. Burton AJ, Thomson AR, Dawson WM, Brady RL, Woolfson DN (2016) Installing hydrolytic activity into a completely de novo protein framework. *Nat Chem* 8: 837–844.
4. Joh NH, Wang T, Bhate MP, Acharya R, Wu Y, Grabe M, Hong M, Grigoryan G, DeGrado WF (2014) De novo design of a transmembrane Zn²⁺-transporting four-helix bundle. *Science* 346:1520–1524.
5. Hoersch D, Roh SH, Chiu W, Kortemme T (2013) Reprogramming an ATP-driven protein machine into a light-gated nanocage. *Nat Nanotechnol* 8:928–932.
6. Campeotto I, Goldenzweig A, Davey J, Barford L, Marshall JM, Silk SE, Wright KE, Draper SJ, Higgins MK, Fleishman SJ (2017) One-step design of a stable variant of the malaria invasion protein RH5 for use as a vaccine immunogen. *Proc Natl Acad Sci USA* 114: 998–1002.

7. Taylor WR, Chelliah V, Hollup SM, MacDonald JT, Jonassen I (2009) Probing the “dark matter” of protein fold space. *Structure* 17:1244–1252.
8. Woolfson DN, Bartlett GJ, Burton AJ, Heal JW, Niitsu A, Thomson AR, Wood CW (2015) De novo protein design: how do we expand into the universe of possible protein structures?. *Curr Opin Struct Biol* 33:16–26.
9. Huang PS, Oberdorfer G, Xu CF, Pei XY, Nannenga BL, Rogers JM, DiMaio F, Gonen T, Luisi B, Baker D (2014) High thermodynamic stability of parametrically designed helical bundles. *Science* 346:481–485.
10. Thomson AR, Wood CW, Burton AJ, Bartlett GJ, Sessions RB, Brady RL, Woolfson DN (2014) Computational design of water-soluble α -helical barrels. *Science* 346:485–488.
11. Brunette TJ, Parmeggiani F, Huang PS, Bhabha G, Ekiert DC, Tsutakawa SE, Hura GL, Tainer JA, Baker D (2015) Exploring the repeat protein universe through computational protein design. *Nature* 528:580–584.
12. MacDonald JT, Kabasakal BV, Godding D, Kraatz S, Henderson L, Barber J, Freemont PS, Murray JW (2016) Synthetic beta-solenoid proteins with the fragment-free computational design of a beta-hairpin extension. *Proc Natl Acad Sci USA* 113:10346–10351.
13. Kajava AV (2012) Tandem repeats in proteins: from sequence to structure. *J Struct Biol* 179:279–288.
14. Wood CW, Bruning M, Ibarra AA, Bartlett GJ, Thomson AR, Sessions RB, Brady RL, Woolfson DN (2014) CCBUILDER: an interactive web-based tool for building, designing and assessing coiled-coil protein assemblies. *Bioinformatics* 30:3029–3035.
15. Grigoryan G, DeGrado WF (2011) Probing designability via a generalized model of helical bundle geometry. *J Mol Biol* 405:1079–1100.
16. Rose A, Manikantan S, Schraegle SJ, Maloy MA, Stahlberg EA, Meier I (2004) Genome-wide identification of arabidopsis coiled-coil proteins and establishment of the ARABI-COIL database. *Plant Physiol* 134:927–939.
17. Rackham OJL, Madera M, Armstrong CT, Vincent TL, Woolfson DN, Gough J (2010) The evolution and structure prediction of coiled coils across all genomes. *J Mol Biol* 403:480–493.
18. Strelkov SV, Burkhard P (2002) Analysis of α -helical coiled coils with the program TWISTER reveals a structural mechanism for stutter compensation. *J Struct Biol* 137:54–64.
19. Mason JM, Arndt KM (2004) Coiled coil domains: stability, specificity, and biological implications. *ChemBiochem* 5:170–176.
20. Woolfson DN, Bartlett GJ, Bruning M, Thomson AR (2012) New currency for old rope: from coiled-coil assemblies to α -helical barrels. *Curr Opin Struct Biol* 22:432–441.
21. Crick FHC (1953) The packing of α -helices - simple coiled-coils. *Acta Cryst* 6:689–697.
22. Lupas AN, Gruber M (2005) The structure of α -helical coiled coils. *Adv Prot Chem* 70:37–78.
23. Lupas AN, Basser J (2017) Coiled coils - a model system for the 21st century. *Trends Biochem Sci* 42:130–140.
24. Walshaw J, Woolfson DN (2003) Extended knobs-into-holes packing in classical and complex coiled-coil assemblies. *J Struct Biol* 144:349–361.
25. Harbury PB, Zhang T, Kim PS, Alber T (1993) A switch between 2-stranded, 3-stranded and 4-stranded coiled coils in GCN4 leucine-zipper mutants. *Science* 262:1401–1407.
26. Harbury PB, Kim PS, Alber T (1994) Crystal-structure of an isoleucine-zipper trimer. *Nature* 371:80–83.
27. Fletcher JM, Boyle AL, Bruning M, Bartlett GJ, Vincent TL, Zaccari NR, Armstrong CT, Bromley EHC, Booth PJ, Brady RL, Thomson AR, Woolfson DN (2012) A basis set of de novo coiled-coil peptide oligomers for rational protein design and synthetic biology. *ACS Synth Biol* 1:240–250.
28. Moutevelis E, Woolfson DN (2009) A periodic table of coiled-coil protein structures. *J Mol Biol* 385:726–732.
29. Testa OD, Moutevelis E, Woolfson DN (2009) CC plus: a relational database of coiled-coil structures. *Nucleic Acids Res* 37:D315–D322.
30. Armstrong CT, Vincent TL, Green PJ, Woolfson DN (2011) SCORER 2.0: an algorithm for distinguishing parallel dimeric and trimeric coiled-coil sequences. *Bioinformatics* 27:1908–1914.
31. Vincent TL, Green PJ, Woolfson DN (2013) LOGICOIL-multi-state prediction of coiled-coil oligomeric state. *Bioinformatics* 29:69–76.
32. Wolf E, Kim PS, Berger B (1997) MultiCoil: a program for predicting two- and three-stranded coiled coils. *Protein Sci* 6:1179–1189.
33. Woolfson DN (2005) The design of coiled-coil structures and assemblies. *Adv Prot Sci* 70:79–112.
34. Woolfson DN (2017) Coiled-coil design: updated and upgraded. *Subcell Biochem* 82:35–61.
35. Koronakis V, Sharff A, Koronakis E, Luisi B, Hughes C (2000) Crystal structure of the bacterial membrane protein TolC central to multidrug efflux and protein export. *Nature* 405:914–919.
36. Natri F, Chino M, Maglio O, Bhagi-Damodaran A, Lu Y, Lombardi A (2016) Design and engineering of artificial oxygen-activating metalloenzymes. *Chem Soc Rev* 45:5020–5054.
37. Ozbek S, Engel J, Stetefeld J (2002) Storage function of cartilage oligomeric matrix protein: the crystal structure of the coiled-coil domain in complex with vitamin D-3. *EMBO J* 21:5960–5968.
38. Burgess NC, Sharp TH, Thomas F, Wood CW, Thomson AR, Zaccari NR, Brady RL, Serpell LC, Woolfson DN (2015) Modular design of self-assembling peptide-based nanotubes. *J Am Chem Soc* 137:10554–10562.
39. Mahendran KR, Niitsu A, Kong LB, Thomson AR, Sessions RB, Woolfson DN, Bayley H (2017) A monodisperse transmembrane α -helical peptide barrel. *Nat Chem* 9:411–419.
40. Xu CF, Liu R, Mehta AK, Guerrero-Ferreira RC, Wright ER, Dunin-Horkawicz S, Morris K, Serpell LC, Zuo XB, Wall JS, Conticello VP (2013) Rational design of helical nanotubes from self-assembly of coiled-coil lock washers. *J Am Chem Soc* 135:15565–15578.
41. Crick FHC (1953) The fourier transform of a coiled-coil. *Acta Cryst* 6:685–689.
42. Offer G, Sessions R (1995) Computer modeling of the α -helical coiled-coil - packing of side-chains in the inner-core. *J Mol Biol* 249:967–987.
43. Harbury PB, Tidor B, Kim PS (1995) Repacking protein cores with backbone freedom - structure prediction for coiled coils. *Proc Natl Acad Sci USA* 92:8408–8412.
44. Harbury PB, Plecs JJ, Tidor B, Alber T, Kim PS (1998) High-resolution protein design with backbone freedom. *Science* 282:1462–1467.
45. Offer G, Hicks MR, Woolfson DN (2002) Generalized Crick equations for modeling noncanonical coiled coils. *J Struct Biol* 137:41–53.
46. Wood CW, Heal JW, Thomson AR, Bartlett GJ, Ibarra AA, Leo Brady R, Sessions RB, Woolfson DN (2017) ISAMBARD: an open-source computational environment for biomolecular analysis, modelling and design. *Bioinformatics* DOI: 10.1093/bioinformatics/btx352

47. Chaudhury S, Lyskov S, Gray JJ (2010) PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics* 26: 689–691.
48. Walshaw J, Woolfson DN (2001) SOCKET: A program for identifying and analysing coiled-coil motifs within protein structures. *J Mol Biol* 307:1427–1450.
49. McIntosh-Smith S, Wilson T, Ibarra AA, Crisp J, Sessions RB (2012) Benchmarking energy efficiency, power costs and carbon emissions on heterogeneous systems. *Comput J* 55:192–205.
50. McIntosh-Smith S, Price J, Sessions RB, Ibarra AA (2015) High performance in silico virtual drug screening on many-core processors. *Int J High Perform C* 29: 119–134.
51. Das R, Baker D (2008) Macromolecular modeling with Rosetta. *Annu Rev Biochem* 77:363–382.
52. Hagelueken G, Clarke BR, Huang HX, Tuukkanen A, Danciu I, Svergun DI, Hussain R, Liu HT, Whitfield C, Naismith JH (2015) A coiled-coil domain acts as a molecular ruler to regulate O-antigen chain length in lipopolysaccharide. *Nat Struct Mol Biol* 22:50–56.
53. Scholefield J, Henriques R, Savulescu AF, Fontan E, Boucharlat A, Laplantine E, Smahi A, Israel A, Agou F, Mhlanga MM (2016) Super-resolution microscopy reveals a preformed NEMO lattice structure that is collapsed in incontinentia pigmenti. *Nat Commun* 7:13.
54. Turgay Y, Eibauer M, Goldman AE, Shimi T, Khayat M, Ben-Harush K, Dubrovsky-Gaupp A, Sapra KT, Goldman RD, Medalia O (2017) The molecular architecture of lamins in somatic cells. *Nature* 543:261–264.
55. Mei Y, Su MF, Sanishvili R, Chakravarthy S, Colbert CL, Sinha SC (2016) Identification of BECN1 and ATG14 coiled-coil interface residues that are important for starvation-induced autophagy. *Biochemistry* 55: 4239–4253.
56. Kiss B, Kalmar L, Nyitray L, Pal G (2016) Structural determinants governing S100A4-induced isoform-selective disassembly of nonmuscle myosin II filaments. *FEBS J* 283:2164–2180.
57. Lesne E, Krammer EM, Dupre E, Loch C, Lensink MF, Antoine R, Jacob-Dubuisson F (2016) Balance between coiled-coil stability and dynamics regulates activity of BvgS sensor kinase in *Bordetella*. *MBio* 7: e02089.
58. Wang H, Morita CT (2015) Sensor function for butyrophilin 3A1 in prenyl pyrophosphate stimulation of human V gamma 2V delta 2 T cells. *J Immunol* 195: 4583–4594.
59. Rogozhina Y, Mironovich S, Shestak A, Adyan T, Polyakov A, Podolyak D, Bakulina A, Dzemeshevich S, Zaklyazminskaya E (2016) New intronic splicing mutation in the LMNA gene causing progressive cardiac conduction defects and variable myopathy. *Gene* 595:202–206.
60. Phang JM, Harrop SJ, Duff AP, Sokolova AV, Crossett B, Walsh JC, Beckham SA, Nguyen CD, Davies RB, Glockner C, Bromley EHC, Wilk KE, Curmi PMG (2016) Structural characterization suggests models for monomeric and dimeric forms of full-length ezrin. *Biochem J* 473:2763–2782.
61. 10.5281/zenodo.592834
62. Krivov GG, Shapovalov MV, Dunbrack RL Jr. (2009) Improved prediction of protein side-chain conformations with SCWRL4. *Proteins* 77:778–795.
63. Delorenzi M, Speed T (2002) An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. *Bioinformatics* 18:617–625.
64. Potapov V, Kaplan JB, Keating AE (2015) Data-driven prediction and design of bZIP coiled-coil interactions. *PLoS Comput Biol* 11:28.
65. Fong JH, Keating AE, Singh M (2004) Predicting specificity in bZIP coiled-coil protein interactions. *Genome Biol* 5:10.
66. Lupas A, Vandyke M, Stock J (1991) Predicting coiled coils from protein sequences. *Science* 252:1162–1164.